

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG WEBSITE LUYỆN GỖ BÀN PHÍM
SỬ DỤNG CÔNG NGHỆ ASP.NET MVC
KẾT HỢP CHATBOT AI

Người hướng dẫn: THS. NGUYỄN TẮT PHÚ CƯỜNG

Sinh viên thực hiện: NGUYỄN TRẦN MINH TRIẾT

Mã số sinh viên : 2151220040

Lớp: 21CT1

Khóa : 2021

Đà Nẵng, 12/2025.

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG WEBSITE LUYỆN GỖ BÀN PHÍM
SỬ DỤNG CÔNG NGHỆ ASP.NET MVC
KẾT HỢP CHATBOT AI

Người hướng dẫn: THS. NGUYỄN TẮT PHÚ CƯỜNG

Sinh viên thực hiện: NGUYỄN TRẦN MINH TRIẾT

Mã số sinh viên: 2151220040

Lớp: 21CT1

Khóa: 2021

Đà Nẵng, 12/2025.

TÓM TẮT

Tên đề tài: Xây dựng website luyện gõ bàn phím sử dụng công nghệ ASP.NET MVC kết hợp Chatbot AI.

Sinh viên thực hiện: Nguyễn Trần Minh Triết

Giảng viên hướng dẫn: ThS. Nguyễn Tất Phú Cường

MSSV: 2151220040. Lớp: 21CT1

Đề án “Xây dựng website luyện gõ bàn phím sử dụng công nghệ ASP.NET MVC kết hợp Chatbot AI” được thực hiện nhằm cung cấp một giải pháp luyện tập kỹ năng gõ bàn phím thông minh, tương tác cao và khắc phục sự đơn điệu của các công cụ truyền thống. Hệ thống được phát triển trên nền tảng công nghệ ASP.NET MVC ổn định kết hợp với cơ sở dữ liệu SQL Server, tập trung vào các chức năng cốt lõi là cung cấp bài tập phân cấp, đo lường chính xác tốc độ cùng độ chính xác, và trực quan hóa quá trình tiến bộ của người học thông qua hệ thống báo cáo thống kê chi tiết.

Điểm đột phá của dự án là việc tích hợp công nghệ chatbot trí tuệ nhân tạo để mang lại trải nghiệm cá nhân hóa vượt trội. Tính năng này cung cấp một trợ lý ảo hỗ trợ người dùng mọi lúc, đồng thời cho phép tự động sinh ra các văn bản luyện tập mới mẻ không giới hạn theo chủ đề yêu cầu, giúp duy trì hứng thú học tập lâu dài. Bên cạnh đó, đề án đã xây dựng thành công nhóm chức năng quản lý lớp học chuyên biệt, cho phép giáo viên giao bài và giám sát học sinh hiệu quả, kết hợp với các yếu tố trò chơi hóa và bảng xếp hạng thi đua để tạo động lực cạnh tranh lành mạnh, mang lại giá trị ứng dụng thực tiễn cao cho môi trường giáo dục và cộng đồng.

LỜI NÓI ĐẦU

Giữa làn sóng số hóa khả năng làm chủ bàn phím chính là chìa khóa tối ưu hóa năng suất lao động. Mặc dù nhu cầu rất lớn, nhưng thị trường phần mềm giáo dục trong nước vẫn thiếu vắng những nền tảng luyện gõ có tính tương tác cao và khả năng quản lý lớp học tập trung.

Xuất phát từ nhu cầu thực tiễn đó đề án tốt nghiệp với đề tài “Xây dựng website luyện gõ bàn phím sử dụng công nghệ ASP.NET MVC kết hợp Chatbot AI” ra đời. Mục tiêu của dự án là kiến tạo một môi trường học tập 4.0, kết hợp sự mạnh mẽ của công nghệ .NET với trí tuệ nhân tạo để tự động hóa việc sinh bài tập và hỗ trợ người dùng theo thời gian thực. Đây là bước đi nhằm giải quyết bài toán kép nâng cao hứng thú cho người học và cung cấp công cụ theo dõi tiến độ trực quan cho người dạy, khẳng định giá trị thực tiễn của công nghệ trong giáo dục.

Để có thể hoàn thành đề án tốt nghiệp này em xin được bày tỏ lòng biết ơn sâu sắc đến Ban Giám hiệu và Quý Thầy Cô trong Khoa Công nghệ thông tin của Trường Đại học Kiến trúc Đà Nẵng đã tận tình giảng dạy và trang bị cho em nền tảng kiến thức vững chắc. Em cũng xin tri ân gia đình và bạn bè đã luôn là hậu phương vững chắc, đồng viên em trong suốt quá trình học tập.

Đặc biệt, em xin được phép gửi lời cảm ơn chân thành và sâu sắc nhất đến với Thầy ***ThS. Nguyễn Tất Phú Cường***. Thầy vừa là giáo viên chủ nhiệm vừa là giảng viên hướng dẫn trực tiếp của em trong quá trình làm đề án tốt nghiệp này, Thầy đã tận tình hướng dẫn truyền đạt kiến thức chuyên môn giúp em hoàn thiện đề án tốt nghiệp này.

Cuối cùng, em xin cảm ơn Quý Thầy Cô trong Hội đồng đã dành thời gian xem xét và đánh giá đề án tốt nghiệp này của em. Do kiến thức và kinh nghiệm còn hạn chế, đề án không tránh khỏi thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ Quý Thầy Cô để hoàn thiện bản thân và phát triển sản phẩm tốt hơn trong tương lai.

Một lần nữa em xin chân thành cảm ơn!

CAM ĐOAN

Em xin được cam đoan kết quả đạt được trong đồ án tốt nghiệp là sản phẩm của riêng cá nhân dưới sự hướng dẫn của Thầy ThS. Nguyễn Tất Phú Cường. Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của bản thân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Sinh viên thực hiện

Nguyễn Trần Minh Triết

MỤC LỤC

Tóm tắt	
Lời nói đầu	i
Cam đoan	ii
Mục lục	iii
Danh mục hình ảnh	vi
Danh mục bảng biểu	ix
Danh mục các chữ viết tắt	xii
	Trang
MỞ ĐẦU	1
1. Tổng quan về định hướng và lí do nghiên cứu đề tài	1
1.1. Tính cấp thiết của đề tài.....	1
1.2. Tổng quan các kết quả nghiên cứu trong và ngoài nước	2
1.3. Tính khả thi và khả năng mở rộng.....	3
2. Phạm vi đề tài	4
3. Mục tiêu đề tài	5
3.1. Mục tiêu chung.....	5
3.2. Mục tiêu cụ thể	5
4. Nhiệm vụ của đề tài	6
5. Đối tượng, phạm vi nghiên cứu	7
5.1. Đối tượng nghiên cứu.....	7
5.2. Phạm vi nghiên cứu	7
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	9
1.1. Giới thiệu về nền tảng .NET Framework	9
1.2. Giới thiệu về ngôn ngữ lập trình C#	10
1.3. Giới thiệu về ASP.NET MVC	12
1.4. Giới thiệu về hệ quản trị cơ sở dữ liệu Microsoft SQL Server	14
1.5. Giới thiệu về ADO.NET	16
1.6. Giới thiệu về LINQ to SQL	17
1.7. Giới thiệu về ngôn ngữ JavaScript	19
1.8. Giới thiệu về Bootstrap	21
1.9. Giới thiệu về SignalR	22
CHƯƠNG 2: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	24

2.1. Mô tả bài toán	24
2.2. Khảo sát và xác định yêu cầu của hệ thống.....	25
2.2.1. Khảo sát hệ thống	25
2.2.2. Xác định yêu cầu hệ thống	25
2.2.3. Phân tích yêu cầu hệ thống.....	27
2.3. Sơ đồ Use Case và biểu đồ tuần tự minh hoạ các nghiệp vụ.....	35
2.3.1. Minh họa Sơ đồ Use Case và biểu đồ tuần tự tổng quát	35
2.3.2. Minh họa chức năng Đăng kí tài khoản.....	43
2.3.3. Minh họa chức năng Xác thực tài khoản Email	45
2.3.4. Minh họa chức năng Đăng nhập/Đăng xuất	46
2.3.5. Minh họa chức năng Quên mật khẩu.....	49
2.3.6. Minh họa chức năng Xem và Chọn bài luyện	53
2.3.7. Minh họa chức năng Luyện gõ và Lưu kết quả.....	56
2.3.8. Minh họa chức năng Trò chơi gõ chữ.....	59
2.3.9. Minh họa chức năng AI tạo văn bản tự động	63
2.3.10. Minh họa chức năng Tham gia cuộc thi chung	66
2.3.11. Minh họa chức năng Tạo và Chia sẻ văn bản.....	70
2.3.12. Minh họa chức năng Tạo lớp học.....	74
2.3.13. Minh họa chức năng Gửi lời mời qua Email.....	76
2.3.14. Minh họa chức năng Tham gia lớp học	78
2.3.15. Minh họa chức năng Quản lý bài luyện tập.....	81
2.3.16. Minh họa chức năng Quản lý người dùng.....	83
2.3.17. Minh họa chức năng Quản lý lớp học	85
2.3.18. Minh họa chức năng Theo dõi tiến độ học viên	87
2.3.19. Minh họa chức năng Chatbot AI hỗ trợ người dùng	89
2.3.20. Minh họa chức năng Thống kê và Báo cáo	92
2.4. Sơ đồ lớp	94
2.5. Thiết kế cơ sở dữ liệu.....	100
2.5.1. Mô hình thực thể liên kết	100
2.5.2. Mô hình quan hệ.....	104
2.6. Thiết kế giao diện.....	118
2.6.1. Giao diện của người dùng	118
2.6.2. Giao diện của giáo viên	129
2.6.3. Giao diện của học sinh	133
2.6.4. Giao diện của quản trị viên.....	135
CHƯƠNG 3: CÀI ĐẶT VÀ PHÁT TRIỂN WEBSITE	139

3.1. Cấu trúc các thư viện	139
3.2. Cấu trúc thư mục code.....	141
3.2.1. Cấu trúc thư mục code Back-end	141
3.2.2. Cấu trúc thư mục code Front-end.....	142
3.3. Xử lý thời gian thực.....	143
3.4. Xử lý kiểm tra chữ đúng sai	146
3.5. Xử lý xác thực qua Email.....	150
3.6. Xử lý Chatbot AI hỗ trợ.....	155
3.7. Xử lý chống spam và bảo mật.....	159
3.8. Kiểm thử.....	164
3.8.1. Các phương án kiểm thử	164
3.8.2. Các chức năng cần kiểm thử	165
3.8.3. Test cases chi tiết.....	167
3.8.4. Kế hoạch thực hiện.....	176
3.8.5. Công cụ và tài nguyên	176
KẾT LUẬN.....	178
1. Kết quả đạt được	178
2. Hạn chế.....	179
3. Hướng phát triển	179
DANH MỤC TÀI LIỆU THAM KHẢO.....	181

DANH MỤC HÌNH ẢNH

Hình 2.1. Sơ đồ Use Case tổng quát	35
Hình 2.2. Biểu đồ tuần tự tổng quát.....	38
Hình 2.3. Sơ đồ Use Case chức năng Đăng kí tài khoản	43
Hình 2.4. Biểu đồ tuần tự chức năng Đăng kí tài khoản.....	44
Hình 2.5. Sơ đồ Use Case chức năng Xác thực tài khoản Email.....	45
Hình 2.6. Biểu đồ tuần tự chức năng Xác thực tài khoản Email	45
Hình 2.7. Sơ đồ Use Case chức năng Đăng nhập/Đăng xuất	46
Hình 2.8. Biểu đồ tuần tự chức năng Đăng nhập/Đăng xuất	47
Hình 2.9. Sơ đồ Use Case chức năng Quên mật khẩu	49
Hình 2.10. Biểu đồ tuần tự chức năng Quên mật khẩu.....	51
Hình 2.11. Sơ đồ Use Case chức năng Xem và Chọn bài luyện.....	53
Hình 2.12. Biểu đồ tuần tự chức năng Xem và Chọn bài luyện	54
Hình 2.13. Sơ đồ Use Case chức năng Luyện gõ và Lưu kết quả	56
Hình 2.14. Biểu đồ tuần tự chức năng Luyện gõ và Lưu kết quả.....	58
Hình 2.15. Sơ đồ Use Case chức năng Trò chơi gõ chữ	59
Hình 2.16. Biểu đồ tuần tự chức năng Trò chơi gõ chữ.....	61
Hình 2.17. Sơ đồ Use Case chức năng AI tạo văn bản tự động.....	63
Hình 2.18. Biểu đồ tuần tự chức năng AI tạo văn bản tự động	65
Hình 2.19. Sơ đồ Use Case chức năng Tham gia cuộc thi chung.....	66
Hình 2.20. Biểu đồ tuần tự chức năng Tham gia cuộc thi chung	68
Hình 2.21. Sơ đồ Use Case chức năng Tạo và Chia sẻ văn bản	70
Hình 2.22. Biểu đồ tuần tự chức năng Tạo và Chia sẻ văn bản.....	72
Hình 2.23. Sơ đồ Use Case chức năng Tạo lớp học	74
Hình 2.24. Biểu đồ tuần tự chức năng Tạo lớp học	75
Hình 2.25. Sơ đồ Use Case chức năng Gửi lời mời qua Email	76
Hình 2.26. Biểu đồ tuần tự chức năng Gửi lời mời qua Email	77
Hình 2.27. Sơ đồ Use Case chức năng Tham gia lớp học	78
Hình 2.28. Biểu đồ tuần tự chức năng Tham gia lớp học	80
Hình 2.29. Sơ đồ Use Case chức năng Quản lý bài luyện tập	81
Hình 2.30. Biểu đồ tuần tự chức năng Quản lý bài luyện tập.....	82
Hình 2.31. Sơ đồ Use Case chức năng Quản lý người dùng	83

Hình 2.32. Biểu đồ tuần tự chức năng Quản lý người dùng.....	84
Hình 2.33. Sơ đồ Use Case chức năng Quản lý lớp học.....	85
Hình 2.34. Biểu đồ tuần tự chức năng Quản lý lớp học	86
Hình 2.35. Sơ đồ Use Case chức năng Theo dõi tiến độ học viên.....	87
Hình 2.36. Biểu đồ tuần tự chức năng Theo dõi tiến độ học viên	88
Hình 2.37. Sơ đồ Use Case chức năng Chatbot AI hỗ trợ người dùng.....	89
Hình 2.38. Biểu đồ tuần tự chức năng Chatbot AI hỗ trợ người dùng.....	91
Hình 2.39. Sơ đồ Use Case chức năng Thống kê và Báo cáo.....	92
Hình 2.40. Biểu đồ tuần tự chức năng Thống kê và Báo cáo	93
Hình 2.41. Sơ đồ lớp.....	94
Hình 2.42. Mô hình thực thể liên kết.....	100
Hình 2.43. Mô hình quan hệ	104
Hình 2.44. Màn hình giao diện chính người dùng ở trang chủ.....	118
Hình 2.45. Màn hình giới thiệu về website.....	119
Hình 2.46. Màn hình câu hỏi và trả lời chung	119
Hình 2.47. Màn hình luyện gõ bàn phím cơ bản	120
Hình 2.48. Màn hình luyện gõ hàng phím trên và hàng phím dưới.....	120
Hình 2.49. Màn hình trò chơi gõ chữ nhanh đầu tiên.....	121
Hình 2.50. Màn hình trò chơi gõ chữ nhanh thứ 2	121
Hình 2.51. Màn hình trò chơi gõ chữ nhanh thứ 3	122
Hình 2.52. Màn hình trò chơi gõ chữ nhanh thứ 4	122
Hình 2.53. Màn hình giao diện đăng kí của người dùng	123
Hình 2.54. Màn hình giao diện đăng nhập của người dùng.....	123
Hình 2.55. Màn hình kiểm tra tốc độ luyện gõ bàn phím.....	124
Hình 2.56. Màn hình kết quả kiểm tra luyện gõ bàn phím	124
Hình 2.57. Màn hình kết quả kiểm tra luyện gõ bàn phím	125
Hình 2.58. Màn hình tham gia cuộc thi đánh máy.....	125
Hình 2.59. Màn hình tạo cuộc thi đánh máy công khai.....	126
Hình 2.60. Màn hình tạo cuộc thi đánh máy bí mật	126
Hình 2.61. Màn hình các chế độ luyện tập	127
Hình 2.62. Màn hình các văn bản luyện tập mới.....	127
Hình 2.63. Màn hình các văn bản luyện tập yêu thích	128
Hình 2.64. Màn hình các văn bản luyện tập đã tạo.....	128
Hình 2.65. Màn hình tạo văn bản luyện tập.....	129
Hình 2.66. Màn hình chính giao diện của giáo viên.....	129

Hình 2.67. Màn hình danh sách lớp học của giáo viên.....	130
Hình 2.68. Màn hình mời học sinh vào lớp bằng Email.....	130
Hình 2.69. Màn hình quản lý lớp học của giáo viên.....	131
Hình 2.70. Màn hình tạo bài kiểm tra cho lớp học của giáo viên.....	131
Hình 2.71. Màn hình quản lý các bài kiểm tra của giáo viên	132
Hình 2.72. Màn hình tạo văn bản kiểm tra của giáo viên.....	132
Hình 2.73. Màn hình quản lý danh sách thành viên làm kiểm tra	133
Hình 2.74. Màn hình chính giao diện của học sinh	133
Hình 2.75. Màn hình tham gia lớp học bằng mã lớp	134
Hình 2.76. Màn hình kết quả bài thi của học sinh	134
Hình 2.77. Màn hình trang đăng nhập của quản trị viên	135
Hình 2.78. Màn hình chính giao diện của quản trị viên	135
Hình 2.79. Màn hình quản lý tài khoản của quản trị viên	136
Hình 2.80. Màn hình thêm mới tài khoản của quản trị viên	136
Hình 2.81. Màn hình quản lý văn bản hệ thống.....	137
Hình 2.82. Màn hình thêm mới văn bản hệ thống của quản trị viên	137
Hình 2.83. Màn hình quản lý văn bản cuộc thi chung tranh bảng xếp hạng	138
Hình 2.84. Màn hình quản lý văn bản do người dùng tạo	138

DANH MỤC BẢNG BIỂU

Bảng 2.1. Mô tả Use Case tổng quát	36
Bảng 2.2. Luồng chính của biểu đồ tuần tự tổng quát	39
Bảng 2.3. Luồng phụ của biểu đồ tuần tự tổng quát	42
Bảng 2.4. Mô tả Sơ đồ Use Case chức năng Đăng kí tài khoản	43
Bảng 2.5. Luồng chính chức năng Đăng kí tài khoản	44
Bảng 2.6. Luồng phụ chức năng Đăng kí tài khoản	44
Bảng 2.7. Mô tả Sơ đồ Use Case chức năng Xác thực tài khoản Email	45
Bảng 2.8. Luồng chính chức năng Xác thực tài khoản Email	45
Bảng 2.9. Luồng phụ chức năng Xác thực tài khoản Email	46
Bảng 2.10. Mô tả Use Case chức năng Đăng nhập/Đăng xuất	46
Bảng 2.11. Luồng chính chức năng Đăng nhập/Đăng xuất	48
Bảng 2.12. Luồng phụ chức năng Đăng nhập/Đăng xuất	48
Bảng 2.13. Mô tả Use Case chức năng Quên mật khẩu	50
Bảng 2.14. Luồng chính chức năng Quên mật khẩu	52
Bảng 2.15. Luồng phụ chức năng Quên mật khẩu	53
Bảng 2.16. Mô tả Use Case chức năng Xem và Chọn bài luyện	54
Bảng 2.17. Luồng chính chức năng Xem và Chọn bài luyện	55
Bảng 2.18. Luồng phụ chức năng Xem và Chọn bài luyện	55
Bảng 2.19. Mô tả Use Case chức năng Luyện gõ và Lưu kết quả	57
Bảng 2.20. Luồng chính chức năng Luyện gõ và Lưu kết quả	58
Bảng 2.21. Luồng phụ chức năng Luyện gõ và Lưu kết quả	59
Bảng 2.22. Mô tả Use Case chức năng Trò chơi gõ chữ	60
Bảng 2.23. Luồng chính chức năng Trò chơi gõ chữ	62
Bảng 2.24. Luồng phụ chức năng Trò chơi gõ chữ	62
Bảng 2.25. Mô tả Use Case chức năng AI tạo văn bản tự động	64
Bảng 2.26. Luồng chính chức năng AI tạo văn bản tự động	65
Bảng 2.27. Luồng phụ chức năng AI tạo văn bản tự động	66
Bảng 2.28. Mô tả Use Case chức năng Tham gia cuộc thi chung	67
Bảng 2.29. Luồng chính chức năng Tham gia cuộc thi chung	69
Bảng 2.30. Luồng phụ chức năng Tham gia cuộc thi chung	69
Bảng 2.31. Mô tả Use Case chức năng Tạo và Chia sẻ văn bản	71

Bảng 2.32. Luồng chính chức năng Tạo và Chia sẻ văn bản.....	73
Bảng 2.33. Luồng phụ chức năng Tạo và Chia sẻ văn bản.....	73
Bảng 2.34. Mô tả Use Case chức năng Tạo lớp học.....	74
Bảng 2.35. Luồng chính chức năng Tạo lớp học.....	75
Bảng 2.36. Luồng phụ chức năng Tạo lớp học.....	75
Bảng 2.37. Mô tả Use Case chức năng Gửi lời mời qua Email.....	77
Bảng 2.38. Luồng chính chức năng Gửi lời mời qua Email.....	78
Bảng 2.39. Luồng phụ chức năng Gửi lời mời qua Email.....	78
Bảng 2.40. Mô tả Use Case chức năng Tham gia lớp học.....	79
Bảng 2.41. Luồng chính chức năng Tham gia lớp học.....	80
Bảng 2.42. Luồng phụ chức năng Tham gia lớp học.....	81
Bảng 2.43. Mô tả Use Case chức năng Quản lý bài luyện tập.....	81
Bảng 2.44. Luồng chính chức năng Quản lý bài luyện tập.....	82
Bảng 2.45. Luồng phụ chức năng Quản lý bài luyện tập.....	82
Bảng 2.46. Mô tả Use Case chức năng Quản lý người dùng.....	83
Bảng 2.47. Luồng chính chức năng Quản lý người dùng.....	84
Bảng 2.48. Luồng phụ chức năng Quản lý người dùng.....	84
Bảng 2.49. Mô tả Use Case chức năng Quản lý lớp học.....	85
Bảng 2.50. Luồng chính chức năng Quản lý lớp học.....	86
Bảng 2.51. Luồng phụ chức năng Quản lý lớp học.....	87
Bảng 2.52. Mô tả Use Case chức năng Theo dõi tiến độ học viên.....	87
Bảng 2.53. Luồng chính chức năng Theo dõi tiến độ học viên.....	88
Bảng 2.54. Luồng phụ chức năng Theo dõi tiến độ học viên.....	88
Bảng 2.55. Mô tả Use Case chức năng Chatbot AI hỗ trợ người dùng.....	90
Bảng 2.56. Luồng chính chức năng Chatbot AI hỗ trợ người dùng.....	91
Bảng 2.57. Luồng phụ chức năng Chatbot AI hỗ trợ người dùng.....	91
Bảng 2.58. Mô tả Use Case chức năng Thống kê và Báo cáo.....	92
Bảng 2.59. Luồng chính chức năng Thống kê và Báo cáo.....	93
Bảng 2.60. Luồng phụ chức năng Thống kê và Báo cáo.....	93
Bảng 2.61. Mô tả các tác nhân sơ đồ lớp.....	95
Bảng 2.62. Mô tả các lớp thực thể.....	96
Bảng 2.63. Mô tả quan hệ giữa tác nhân và thực thể của sơ đồ lớp.....	98
Bảng 2.64. Mô tả quan hệ giữa các thực thể.....	101
Bảng 2.65. Mô tả mô hình quan hệ.....	105
Bảng 2.66. Mô tả bảng USER.....	109

Bảng 2.67. Mô tả bảng TYPEACCOUNT	110
Bảng 2.68. Mô tả bảng CLASSROOM	110
Bảng 2.69. Mô tả bảng USERPROGRESS	111
Bảng 2.70. Mô tả bảng EXERCISELANGUAGE	111
Bảng 2.71. Mô tả bảng EXERCISE.....	111
Bảng 2.72. Mô tả bảng EXERCISETEXT	112
Bảng 2.73. Mô tả bảng TEXTTESTEDU.....	112
Bảng 2.74. Mô tả bảng TESTEDU.....	113
Bảng 2.75. Mô tả bảng TYPINGRESULTEDU	114
Bảng 2.76. Mô tả bảng TYPINGRESULTS	114
Bảng 2.77. Mô tả bảng TYPINGRESULTGAME.....	115
Bảng 2.78. Mô tả bảng COMPETITION.....	116
Bảng 2.79. Mô tả bảng TEXTPRACTICE	117
Bảng 2.80. Bảng Test Case Quản lý tài khoản	167
Bảng 2.81. Bảng Test Case Kiểm tra gõ phím.....	170
Bảng 2.82. Bảng Test Case Luyện văn bản và AI.....	171
Bảng 2.83. Bảng Test Case Cuộc thi	172
Bảng 2.84. Bảng Test Case Quản lý lớp học	173
Bảng 2.85. Bảng Test Case Admin	174
Bảng 2.86. Bảng Test Case SignalR	175
Bảng 2.87. Bảng Test Case Bảo mật chung.....	176

DANH MỤC CÁC CHỮ VIẾT TẮT

STT	Từ viết tắt	Giải nghĩa
1	WPM	Words Per Minute (số từ mỗi phút)
2	SSMS	SQL Server Management Studio
3	MVC	Model-View-Controller
4	OTP	One-Time-Password
5	API	Application Programming Interface
6	SMTP	Simple Mail Transfer Protocol
7	AI	Artificial Intelligence
8	UC	UseCase

MỞ ĐẦU

1. Tổng quan về định hướng và lí do nghiên cứu đề tài

Trong bối cảnh của cuộc cách mạng công nghiệp 4.0 và quá trình chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, kỹ năng sử dụng máy tính thành thạo đã trở thành một yêu cầu cơ bản đối với hầu hết các ngành nghề. Trong đó, kỹ năng gõ bàn phím nhanh và chính xác không chỉ là một lợi thế mà còn là một yếu tố nền tảng, ảnh hưởng trực tiếp đến hiệu suất và năng suất làm việc cũng như học tập. Việc gõ bàn phím chậm và thiếu chính xác không chỉ làm lãng phí thời gian mà còn có thể làm gián đoạn dòng suy nghĩ, giảm khả năng tập trung vào nội dung công việc. Nhận thức được tầm quan trọng và tính cấp thiết của vấn đề, đề tài xây dựng website luyện gõ bàn phím được thực hiện với mục tiêu cung cấp một công cụ đào tạo hiện đại, hiệu quả và dễ dàng tiếp cận cho mọi đối tượng người dùng.

Định hướng nghiên cứu của đề tài tập trung vào ba khía cạnh chính: phương pháp sư phạm, trải nghiệm người dùng và giải pháp công nghệ. Về mặt sư phạm, website sẽ xây dựng một lộ trình học tập có hệ thống, từ các bài học cơ bản dành cho người mới bắt đầu (làm quen với vị trí các phím, gõ 10 ngón) đến các bài tập nâng cao giúp cải thiện tốc độ và giảm lỗi sai. Về trải nghiệm người dùng, đề tài chú trọng vào việc thiết kế giao diện (UI/UX) thân thiện, trực quan và tạo động lực, tích hợp các yếu tố trò chơi hóa (gamification) như hệ thống tính điểm, bảng xếp hạng, và theo dõi tiến độ cá nhân. Về công nghệ, đề tài lựa chọn ASP.NET Framework làm nền tảng phát triển phía backend nhờ vào sự ổn định, tính bảo mật cao và khả năng mở rộng mạnh mẽ của nó. Hệ thống sẽ được xây dựng theo kiến trúc MVC (Model-View-Controller) để đảm bảo mã nguồn có cấu trúc rõ ràng, dễ bảo trì và phát triển trong tương lai.

1.1. Tính cấp thiết của đề tài.

Tính cấp thiết của đề tài này không xuất phát từ việc “thiếu” các công cụ luyện gõ bàn phím, nó là một yêu cầu bắt buộc xuất phát từ sự thất bại của các công cụ hiện tại trong việc đáp ứng ba cuộc cách mạng: cách mạng E-learning, cách mạng AI, và cách mạng về trải nghiệm người dùng.

Thất bại lớn nhất và cấp thiết nhất nằm ở lĩnh vực Giáo dục. Chúng ta đang ở trong kỷ nguyên E-learning, nhưng lại tồn tại một nghịch lý không thể chấp nhận: Hàng ngàn giáo viên tin học trên cả nước đang dạy “chay” kỹ năng gõ bàn phím. Họ hoàn

toàn không có một công cụ quản lý chuyên biệt. Họ không thể tạo lớp học, không thể giao bài tập, và không thể theo dõi tiến độ của học sinh. Đây là một khoảng trống nghiệp vụ sư phạm khổng lồ. Vì vậy, sự ra đời của module Lớp học (Classroom) trong đề tài này không phải là một lựa chọn, mà là một giải pháp cấp bách để vũ khí hóa giáo viên, đáp ứng nhu cầu quản lý học tập thực tế.

Thứ hai, các công cụ hiện tại đã lỗi thời về mặt Công nghệ. Người dùng ngày nay, từ sinh viên đến nhân viên văn phòng, đã sống và làm việc cùng Trí tuệ Nhân tạo. Họ sẽ không lãng phí thời gian với các nội dung tĩnh, nhàm chán, và lặp lại từ thập niên trước. Họ yêu cầu một nền tảng thông minh. Tính cấp thiết ở đây là việc phải tích hợp AI. Đây là lý do người dùng nên chọn website này: nó sử dụng Gemini API để cung cấp nội dung luyện tập vô hạn (AI Generation) và một Trợ lý ảo (AI Chatbot) 24/7. Đây là tiêu chuẩn mới mà các công cụ khác không có.

Cuối cùng, các công cụ cũ thất bại trong việc giải quyết bài toán tâm lý. Tại sao đa số người dùng bỏ cuộc? Vì luyện tập một mình rất cô đơn và nhàm chán. Học sinh, sinh viên hay dân văn phòng đều cần động lực. Tính cấp thiết của đề tài này là giải quyết triệt để sự nhàm chán đó. Bằng cách áp dụng Gamification (MiniGame, Cuộc thi, Bảng xếp hạng), dự án này biến việc học từ một nhiệm vụ đơn độc thành một trải nghiệm xã hội có tính cạnh tranh, giữ chân người dùng và đảm bảo họ đạt được mục tiêu.

Do đó, tính cấp thiết của đề tài này không chỉ là việc xây dựng thêm một website luyện gõ bàn phím. Đây là một luận đề về việc các công cụ kỹ năng cơ bản phải phát triển. Nó lập luận rằng tương lai của việc học kỹ năng nằm ở sự hội tụ của AI, Gamification, và Quản lý E-learning. Dự án này là một bước đi tất yếu để giải quyết sự lỗi thời của các công cụ hiện tại và đáp ứng nhu cầu thực tế của một lực lượng lao động số và một nền giáo dục đang thay đổi. Đó cũng là lý do để em chọn tìm hiểu và phát triển đề tài này.

1.2. Tổng quan các kết quả nghiên cứu trong và ngoài nước

Việc nghiên cứu và phát triển các công cụ luyện gõ bàn phím đã được thực hiện từ lâu trên thế giới và ở Việt Nam. Trên thế giới, có nhiều website và phần mềm nổi tiếng như Typing.com, Keybr.com, TypingClub, v.v. Các nền tảng này đã rất thành công trong việc áp dụng các phương pháp sư phạm hiệu quả, tích hợp yếu tố trò chơi hóa (gamification) để tăng tính hấp dẫn và duy trì động lực cho người học. Họ cung cấp các bài học đa dạng, từ cơ bản đến nâng cao, hỗ trợ nhiều ngôn ngữ và có hệ thống phân tích chi tiết kết quả. Các website này thường được xây dựng trên các công nghệ web tiên tiến như React, Angular hoặc Vue.js, kết hợp với các framework backend mạnh mẽ.

Tại Việt Nam, các nghiên cứu và ứng dụng về luyện gõ bàn phím cũng đã được chú trọng. Một số sản phẩm tiêu biểu có thể kể đến như phần mềm Typing Master phiên bản tiếng Việt hoặc các website được phát triển bởi các cá nhân và tổ chức trong nước. Tuy nhiên, phần lớn các sản phẩm này vẫn còn hạn chế về mặt tính năng, giao diện hoặc chưa được cập nhật công nghệ mới, đặc biệt là thiếu một nền tảng toàn diện và chuyên nghiệp như các website quốc tế. Điều này cho thấy tiềm năng lớn cho việc phát triển một website luyện gõ bàn phím chất lượng cao tại Việt Nam, đáp ứng nhu cầu ngày càng tăng của người dùng trong nước.

1.3. Tính khả thi và khả năng mở rộng

Việc đánh giá tính khả thi và khả năng mở rộng của đề tài là một bước quan trọng, khẳng định giá trị thực tiễn và tiềm năng phát triển của hệ thống trong tương lai.

Về tính khả thi, đề tài được xây dựng trên một nền tảng công nghệ hoàn toàn khả thi về mặt kỹ thuật, chi phí và vận hành. Về kỹ thuật, hệ thống sử dụng các công nghệ phổ biến, trưởng thành và có tài liệu hỗ trợ dồi dào từ Microsoft, bao gồm framework ASP.NET MVC, ngôn ngữ C# và Hệ quản trị Cơ sở dữ liệu SQL Server. Em, với kiến thức đã được đào tạo trong chuyên ngành Công nghệ Thông tin, hoàn toàn có đủ năng lực để tiếp cận và làm chủ các công nghệ này. Các công cụ phát triển như Visual Studio 2022 và SQL Server Management Studio cung cấp một môi trường làm việc chuyên nghiệp, đẩy nhanh tiến độ và đảm bảo chất lượng mã nguồn. Về chi phí, việc sử dụng các phiên bản miễn phí như Visual Studio Community và SQL Server Express, cùng với việc API Gemini cung cấp các gói sử dụng cho nhà phát triển, khiến chi phí ban đầu gần như bằng không, rất phù hợp với bối cảnh của một đồ án sinh viên. Về vận hành, bản thân hệ thống đã được thiết kế với một nhóm chức năng Quản trị toàn diện, cho phép quản lý nội dung và người dùng, đảm bảo tính khả thi trong việc duy trì hoạt động sau khi triển khai.

Về khả năng mở rộng, đây là một trong những ưu điểm thiết kế lớn nhất của hệ thống, được đảm bảo ngay từ khi lựa chọn kiến trúc. Việc áp dụng triệt để kiến trúc ASP.NET MVC là yếu tố then chốt. Kiến trúc này thực hiện nguyên tắc phân tách các mối quan tâm một cách rõ ràng, chia logic nghiệp vụ, dữ liệu, và giao diện thành các thành phần độc lập (Controller, Model, và View). Điều này tạo ra một hệ thống có tính lắp ghép cao. Khi có nhu cầu mở rộng một chức năng mới trong tương lai, ví dụ như thêm một trò chơi luyện gõ mới hoặc một hình thức luyện tập chuyên biệt (như gõ mã lập trình), nhà phát triển chỉ cần tạo ra các Lớp điều khiển (Controller) và các Lớp mô hình (Model) cùng các Giao diện (View) mới. Quá trình này hoàn toàn không gây ảnh hưởng hay xung đột với các nhóm chức năng hiện có như quản lý lớp học hay quản trị

hệ thống. Hơn nữa, việc sử dụng cơ sở dữ liệu quan hệ SQL Server cũng cho phép mở rộng cấu trúc dữ liệu một cách linh hoạt bằng cách thêm bảng hoặc thêm cột mới mà không phá vỡ dữ liệu cũ. Chính nhờ kiến trúc này, hệ thống không chỉ đáp ứng các yêu cầu của đề tài hiện tại mà còn sẵn sàng cho việc phát triển, nâng cấp trong tương lai.

2. Phạm vi đề tài

Phạm vi của đề tài bao gồm việc thiết kế và xây dựng một hệ sinh thái luyện gõ bàn phím trực tuyến toàn diện, được cấu thành từ nhiều module nghiệp vụ chuyên sâu, vượt xa các chức năng cơ bản.

Module cốt lõi đầu tiên là Hệ thống Quản lý Người dùng, chịu trách nhiệm cho toàn bộ vòng đời tài khoản. Phạm vi này bao gồm các chức năng đăng ký, đăng nhập an toàn, tích hợp xác thực tài khoản qua Gmail và cơ chế khôi phục mật khẩu. Hệ thống cũng cho phép người dùng quản lý hồ sơ cá nhân của mình.

Tiếp theo là module Luyện tập và Thống kê đây là tính năng cốt lõi của website. Phạm vi này không chỉ dừng lại ở việc cung cấp các bài luyện tập theo cấp độ. Nó bao gồm chức năng cho phép người dùng thực hiện bài gõ, hệ thống sẽ đo lường các chỉ số quan trọng như Tốc độ gõ (WPM) và Độ chính xác (Độ chính xác). Các kết quả này được tự động lưu trữ, liên kết với tài khoản người dùng, hình thành nên Hệ thống Thống kê và Theo dõi. Hệ thống này có khả năng trực quan hóa tiến trình học tập của người dùng qua các biểu đồ lịch sử, giúp họ tự đánh giá sự tiến bộ.

Để nâng cao trải nghiệm và tạo động lực, phạm vi đề tài được mở rộng với các module Tương tác Nâng cao. Các chức năng này bao gồm các trò chơi luyện gõ để rèn luyện phản xạ, module Cuộc thi với Bảng xếp hạng thời gian thực để tạo môi trường cạnh tranh, và một hệ thống đóng góp nội dung cho phép người dùng tự tải lên bài luyện tập, với sự kiểm duyệt của quản trị viên.

Một trọng tâm đặc biệt của đề tài là việc tích hợp Công nghệ Trí tuệ Nhân tạo (AI). Phạm vi này bao gồm hai chức năng chính sử dụng API Gemini: một Chatbot AI hỗ trợ giải đáp thắc mắc của người dùng 24/7, và chức năng Tự động tạo Văn bản, cho phép người dùng tự sinh nội dung luyện tập vô hạn theo chủ đề mong muốn.

Một phạm vi nghiệp vụ quan trọng khác là module Quản lý Lớp học. Chức năng này được thiết kế cho vai trò Giáo viên, cho phép họ tạo lớp học ảo, sinh mã lớp, và gửi lời mời tham gia qua Gmail. Giáo viên có thể gán bài luyện tập cho lớp và sử dụng một bảng điều khiển riêng để theo dõi tiến độ chi tiết của từng học sinh, biến website thành một công cụ hỗ trợ giảng dạy.

Cuối cùng, phạm vi của đề tài bao trùm cả module Quản trị Hệ thống. Đây là khu vực dành cho Quản trị viên, cung cấp toàn quyền CRUD (Tạo, Đọc, Cập nhật, Xóa) đối với nội dung bài luyện tập, quản lý toàn bộ tài khoản người dùng (khóa, xóa, cấp quyền), khóa tài khoản (xóa mềm), và xem các báo cáo thống kê toàn cục của website.

3. Mục tiêu đề tài

3.1. Mục tiêu chung

Mục tiêu chung của đề tài là nghiên cứu, phân tích và xây dựng thành công một hệ sinh thái luyện gõ bàn phím trực tuyến toàn diện. Đây không chỉ là một dự án học thuật mà còn được định hướng là một website hoàn chỉnh, chuyên nghiệp và hiệu quả, có khả năng hoạt động như một sản phẩm thực tế để đáp ứng nhu cầu học tập và nâng cao kỹ năng của người dùng Việt Nam. Hệ thống này không chỉ dừng lại ở việc đo lường kỹ năng, mà còn hướng đến việc tạo ra một môi trường học tập thông minh. Đề tài tập trung vào việc tích hợp công nghệ Trí tuệ Nhân tạo để cung cấp các tính năng hỗ trợ tiên tiến và phát triển các nghiệp vụ chuyên biệt nhằm hỗ trợ công tác giảng dạy, học tập trực tuyến. Mục tiêu cuối cùng là sản phẩm phải hoạt động ổn định, bảo mật và dễ dàng mở rộng, góp phần trực tiếp vào việc nâng cao năng suất làm việc cho các ngành nghề đòi hỏi sử dụng máy tính, đồng thời kiến tạo một môi trường học tập trực tuyến thú vị, có tính tương tác cao và sau khi sử dụng website người dùng có thể cải thiện được kỹ năng gõ bàn phím của bản thân.

3.2. Mục tiêu cụ thể

Để hiện thực hóa mục tiêu chung, đề tài đặt ra các mục tiêu cụ thể, trong đó mục tiêu cốt lõi và quan trọng nhất là xây dựng hoàn chỉnh các chức năng luyện gõ bàn phím. Đây là hạt nhân của toàn bộ hệ thống. Mục tiêu này bao gồm việc phát triển một giao diện luyện tập trực quan, có khả năng đo lường chính xác tuyệt đối các chỉ số hiệu suất quan trọng là tốc độ gõ và độ chính xác. Ngay sau khi người dùng hoàn thành, hệ thống phải có khả năng lưu trữ kết quả này. Nối tiếp trực tiếp, mục tiêu là phải xây dựng một hệ thống thống kê cá nhân, sử dụng các biểu đồ trực quan để hiển thị lịch sử luyện tập và sự tiến bộ, giúp người dùng thấy rõ hiệu quả cải thiện của bản thân.

Để phục vụ cho mục tiêu cốt lõi nêu trên, mục tiêu cụ thể thứ hai là xây dựng một nền tảng kỹ thuật và cơ sở dữ liệu vững chắc. Điều này đòi hỏi phải nghiên cứu và áp dụng nền tảng ASP.NET MVC để xây dựng phần backend mạnh mẽ, ổn định, cùng với các công nghệ frontend hiện đại. Đồng thời, phải thiết kế và hiện thực hóa một cấu trúc cơ sở dữ liệu tối ưu, có khả năng lưu trữ hiệu quả thông tin người dùng, nội dung bài học, và hàng triệu bản ghi kết quả luyện tập. Kèm theo đó là việc phát triển hoàn

chỉnh các chức năng định danh cơ bản như đăng ký, đăng nhập an toàn, xác thực qua thư điện tử và quản lý hồ sơ.

Sau khi hoàn thành lỗi luyện tập, mục tiêu tiếp theo là nâng cao trải nghiệm, tạo động lực và tích hợp trí thông minh cho hệ thống. Mục tiêu này sẽ được thực hiện thông qua việc áp dụng các yếu tố “trò chơi hóa” (gamification), bao gồm việc phát triển các trò chơi nhỏ và tổ chức các cuộc thi có bảng xếp hạng cạnh tranh. Đặc biệt, mục tiêu là phải tích hợp thành công công nghệ Trí tuệ Nhân tạo qua API Gemini để giải quyết sự nhàm chán, bao gồm một trợ lý ảo (chatbot) hỗ trợ và chức năng tự động tạo văn bản luyện tập vô hạn theo yêu cầu người dùng.

Mục tiêu cụ thể cuối cùng là mở rộng hệ thống từ một công cụ cá nhân thành một nền tảng E-learning và quản trị chuyên nghiệp. Phải phát triển hoàn chỉnh nhóm chức năng quản lý lớp học, cho phép Giáo viên tạo lớp, gán bài tập, và sử dụng bảng điều khiển chuyên biệt để theo dõi tiến độ của học sinh. Song song đó, phải xây dựng một hệ thống Quản trị tập trung, cho phép Quản trị viên toàn quyền quản lý nội dung, kiểm soát người dùng và kiểm duyệt dữ liệu, đảm bảo hệ thống vận hành trơn tru và mang lại giá trị thực tiễn cao.

4. Nhiệm vụ của đề tài

Để hoàn thành các mục tiêu cụ thể đã đề ra, đề tài cần thực hiện một chuỗi các nhiệm vụ nghiên cứu và phát triển có trình tự logic.

Nhiệm vụ đầu tiên là khảo sát, phân tích và thiết kế. Giai đoạn này bao gồm việc nghiên cứu lý thuyết về các công nghệ sẽ sử dụng, đặc biệt là ASP.NET MVC và API Gemini. Phải tiến hành phân tích các hệ thống tương tự đang thành công để rút kinh nghiệm. Dựa trên kết quả khảo sát, nhiệm vụ là phải đặc tả hóa yêu cầu hệ thống, từ đó thiết kế kiến trúc phần mềm tổng thể và thiết kế chi tiết cơ sở dữ liệu.

Nhiệm vụ thứ hai và trọng tâm nhất là lập trình và hiện thực hóa hệ thống. Dựa trên bản thiết kế, nhiệm vụ là tiến hành phát triển mã nguồn cho cả backend (sử dụng ASP.NET MVC) và frontend (sử dụng HTML, CSS, JavaScript). Phải lập trình tuần tự và hoàn chỉnh tất cả các nhóm chức năng đã đặc tả, từ cơ bản như đăng nhập, luyện tập, đến các chức năng phức tạp như tích hợp AI, nghiệp vụ lớp học và bảng điều khiển quản trị.

Nhiệm vụ thứ ba là kiểm thử và triển khai. Sau khi hoàn thành việc lập trình các chức năng, phải thực hiện công tác kiểm thử phần mềm. Nhiệm vụ này bao gồm việc kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử chấp nhận người dùng để phát hiện, ghi nhận và sửa chữa các lỗi phát sinh. Mục tiêu là đảm bảo hệ thống hoạt động ổn định,

chính xác, bảo mật trước khi triển khai lên môi trường máy chủ thực tế.

Nhiệm vụ cuối cùng là tổng hợp và hoàn thiện luận văn. Phải tổng kết lại toàn bộ quá trình nghiên cứu, phân tích, thiết kế, phát triển và kiểm thử. Các kết quả đạt được, những hạn chế và hướng phát triển trong tương lai sẽ được trình bày rõ ràng, logic và khoa học trong tài liệu báo cáo luận văn tốt nghiệp hoàn chỉnh.

5. Đối tượng, phạm vi nghiên cứu

5.1. Đối tượng nghiên cứu

Đề tài “Xây dựng website luyện gõ bàn phím sử dụng công nghệ ASP.NET MVC kết hợp Chatbot AI” hướng tới một đối tượng người dùng rộng rãi, bao gồm học sinh, sinh viên, những người làm việc văn phòng, người cần tương tác với máy tính hoặc chỉ đơn giản là người cần cải thiện kỹ năng gõ bàn phím để nâng cao hiệu quả học tập và làm việc. Mặc dù đối tượng chính là những người dùng Việt Nam, tuy nhiên website sẽ được phát triển theo hướng có thể dễ dàng mở rộng để hỗ trợ đa ngôn ngữ trong tương lai.

5.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài được xác định một cách rõ ràng và cụ thể, tập trung vào ba khía cạnh chính:

Thứ nhất, về mặt nội dung và chức năng, đề tài hướng tới việc xây dựng một nền tảng luyện gõ bàn phím toàn diện. Điều này bao gồm việc nghiên cứu và thiết kế các bài học được phân cấp từ cơ bản đến nâng cao, đồng thời xây dựng hệ thống kiểm tra và đánh giá kết quả một cách chính xác. Đặc biệt, việc tích hợp các tính năng gamification như bảng xếp hạng, thống kê tiến độ và biểu đồ trực quan sẽ giúp tăng tính tương tác và tạo động lực cho người dùng.

Thứ hai, về mặt công nghệ, đề tài sẽ tập trung vào việc nghiên cứu và ứng dụng những công nghệ nền tảng của Microsoft. Cụ thể, nghiên cứu sẽ đi sâu vào ASP.NET framework để xây dựng phần backend và cơ sở dữ liệu. Mặc dù có nhiều mã nguồn mở và công cụ hiện đại khác, đề tài ưu tiên sử dụng các giải pháp trong hệ sinh thái của Microsoft để đảm bảo sự đồng bộ, ổn định và bảo mật. Phạm vi nghiên cứu cũng bao gồm việc khảo sát các thư viện và công cụ hỗ trợ cho frontend (HTML5, CSS3, JavaScript) nhằm tối ưu hóa trải nghiệm người dùng trên nhiều loại thiết bị.

Thứ ba, về mặt phương pháp, đề tài không đặt nặng việc kế thừa các mã nguồn mở có sẵn một cách trực tiếp mà tập trung vào việc tự xây dựng các module từ đầu. Cách tiếp cận này giúp sinh viên nắm vững kiến thức, hiểu rõ nguyên lý hoạt động của

hệ thống và chủ động trong việc phát triển, bảo trì sau này. Nghiên cứu cũng sẽ tập trung vào việc phân tích nhu cầu người dùng để thiết kế một sản phẩm đáp ứng tốt nhất các yêu cầu thực tế.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu về nền tảng .NET Framework

.NET Framework là một nền tảng phát triển phần mềm framework của Microsoft, chuyên để xây dựng và chạy các ứng dụng trên hệ điều hành Windows. .NET Framework bao gồm hai thành phần chính: Common Language Runtime (động cơ thực thi quản lý bộ nhớ, đa luồng, an toàn kiểu dữ liệu) và thư viện lớp .NET Framework một tập hợp lớn các lớp, API phục vụ các tác vụ cơ bản như xử lý chuỗi, nhập xuất, kết nối cơ sở dữ liệu, giao diện người dùng,.... .NET Framework hỗ trợ xây dựng và chạy ứng dụng Windows và dịch vụ Web (web services), được thiết kế để cung cấp môi trường lập trình hướng đối tượng nhất quán, giảm thiểu xung đột phiên bản khi triển khai phần mềm, cho phép chạy mã an toàn, và loại bỏ các vấn đề hiệu năng thường gặp của môi trường script/interpreted

Đối với .NET Framework thì ưu điểm nổi bật là cung cấp thư viện lớp phong phú và ổn định cho phép tái sử dụng mã trong nhiều tác vụ chung như xử lý chuỗi hay tập tin và cơ sở dữ liệu cùng giao diện. Microsoft nhấn mạnh đây là một framework rất mạnh mẽ và đã trưởng thành với một thư viện lớp đồ sộ hỗ trợ rất nhiều loại ứng dụng trên Windows. Nền tảng này đã được cộng đồng phát triển rộng rãi nhờ mô hình lập trình nhất quán cùng môi trường khởi tạo dễ dàng với Visual Studio và công cụ gỡ lỗi mạnh mẽ cũng như sự hỗ trợ trực tiếp từ Microsoft giúp việc triển khai và bảo trì ứng dụng trở nên đơn giản. Nó cung cấp một nền tảng OOP nhất quán giúp giảm xung đột phiên bản khi cài đặt phần mềm đồng thời đảm bảo mã chạy ở hiệu năng tối ưu so với các môi trường kịch bản hay thông dịch.

Tuy nhiên .NET Framework cũng có một số hạn chế nhất định theo các tài liệu chính thức của Microsoft. Công nghệ này chỉ dành cho Windows nên không có phiên bản cho Linux hoặc macOS và các ứng dụng chỉ chạy trên Windows. Trên cùng một máy tính chỉ có thể cài một phiên bản .NET Framework 4.x vì các bản này là cập nhật tại chỗ chứ không hỗ trợ cài nhiều phiên bản song song như .NET Core. Nền tảng này tiếp tục phát triển chủ yếu thông qua các bản cập nhật khi phiên bản Windows mới được phát hành. Mặc dù mã nguồn tham chiếu có công khai nhưng .NET Framework không phải là dự án mã nguồn mở chủ động nên khả năng cải tiến nhanh hay hỗ trợ cộng đồng tương đối hạn chế. Microsoft khuyến nghị rằng phát triển ứng dụng mới nên dùng .NET hiện đại như .NET 8 thay vì .NET Framework.

1.2. Giới thiệu về ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình hướng đối tượng đa dụng thuộc họ ngôn ngữ C và được thiết kế để phát triển trên nền tảng .NET. Theo Microsoft thì C# là ngôn ngữ phổ biến nhất cho nền tảng .NET vốn là một môi trường phát triển mã nguồn mở và đa nền tảng. Các chương trình C# có khả năng chạy trên nhiều thiết bị khác nhau từ thiết bị IoT đến đám mây cũng như hỗ trợ viết ứng dụng cho điện thoại và máy tính bảng hay máy tính để bàn và máy chủ. Ngôn ngữ này được tối ưu hóa để giúp lập trình viên tăng năng suất trong khi vẫn đảm bảo hiệu năng cao. Ngoài ra C# hỗ trợ đầy đủ các tính năng của lập trình hướng đối tượng kết hợp nhiều tính năng của lập trình hàm và cho phép xử lý tác vụ hiệu năng thấp mà không cần dùng mã nguồn không an toàn.

Dưới đây là ví dụ “Hello, World” trong C# – một chương trình cơ bản minh họa cách in dòng chữ ra màn hình console trên .NET Framework:

```
// Sử dụng namespace System để truy cập các lớp cơ bản như Console
using System;

namespace HelloWorldApp // Khai báo namespace của chương trình
{
    // Lớp Program chứa điểm vào (entry point) của ứng dụng
    class Program
    {
        // Phương thức Main là nơi chương trình bắt đầu thực thi
        static void Main()
        {
            // In ra màn hình dòng chữ "Hello, World"
            Console.WriteLine("Hello, World");
        }
    }
}
```

Về ưu điểm thì C# là ngôn ngữ đa nền tảng và đa dụng cho phép lập trình viên phát triển nhanh chóng đồng thời đạt hiệu năng cao. Đây là ngôn ngữ phổ biến nhất trong hệ sinh thái .NET với hàng triệu lập trình viên trên toàn cầu và hỗ trợ lập trình hướng đối tượng đầy đủ cùng nhiều tính năng của lập trình hàm giúp viết mã an toàn

mà vẫn xử lý hiệu quả các thao tác cấp thấp khi cần thiết. Do hầu hết runtime và thư viện của .NET cũng được viết bằng C# nên các cải tiến ngôn ngữ này thường mang lại lợi ích cho toàn bộ hệ sinh thái, có thể nói đến các lợi ích sau:

- Đa năng và linh hoạt: C# có thể dùng để phát triển nhiều loại ứng dụng, từ ứng dụng máy tính để bàn, web, di động, đến các ứng dụng thực tế ảo, trí tuệ nhân tạo và máy học.

- Phát triển game: Đây là lựa chọn phổ biến cho việc phát triển game nhờ tính năng mạnh mẽ của nó trong nền tảng Unity.

- Đa nền tảng: Với nền tảng .NET hiện đại, C# cho phép bạn xây dựng ứng dụng chạy trên nhiều hệ điều hành khác nhau, bao gồm cả các giải pháp IoT.

- Dễ học và sử dụng: C# có cú pháp đơn giản, dễ đọc, giúp người mới bắt đầu tiếp cận nhanh chóng. Nó cũng có đường cong học tập tương đối thấp.

- An toàn và hiệu quả: C# là ngôn ngữ "type-safe" (kiểu dữ liệu an toàn), giúp ngăn ngừa nhiều lỗi phổ biến tại thời điểm biên dịch. Nó cũng cung cấp hiệu suất thời gian chạy tuyệt vời và quản lý bộ nhớ tự động thông qua cơ chế thu gom rác.

- Hỗ trợ mạnh mẽ: C# được phát triển và hỗ trợ bởi Microsoft, với một hệ sinh thái rộng lớn, bao gồm thư viện lớn và công cụ phát triển hiệu quả như Visual Studio.

- Cộng đồng lớn: Có một cộng đồng người dùng C# đang phát triển nhanh chóng, cung cấp nhiều tài liệu, hướng dẫn và sự hỗ trợ.

- Khả năng mở rộng và tích hợp: C# có khả năng mở rộng cao cho các chương trình và tích hợp tốt với nhiều cơ sở dữ liệu và công nghệ khác.

Ngoài ra C# có một số hạn chế bao gồm sự phụ thuộc vào .NET, hiệu năng không bằng các ngôn ngữ biên dịch trực tiếp như C++, khó tương tác cấp thấp với phần cứng và có đường cong học tập dốc đối với các khái niệm phức tạp trong hệ sinh thái .NET. Bên cạnh đó, một số công nghệ đặc thù của Windows có thể không hỗ trợ đầy đủ trên các nền tảng khác:

- Phụ thuộc vào .NET Runtime: C# cần .NET Runtime để chạy, điều này có thể tăng kích thước tệp thực thi của ứng dụng so với các ngôn ngữ như C/C++.

- Hiệu năng: Hiệu năng của C# có thể thấp hơn C++ do chạy trên máy ảo (CLR) và cơ chế thu gom rác (Garbage Collection), không phù hợp cho các ứng dụng yêu cầu hiệu năng cao nhất (như ứng dụng đồ họa, âm thanh, hệ thống thời gian thực).

- Khó khăn trong tương tác cấp thấp: C# không được thiết kế để tương tác trực

tiếp với phần cứng hoặc thực hiện các tác vụ cấp thấp. Mặc dù có thể sử dụng các kỹ thuật như P/Invoke, nhưng nó có thể phức tạp và tiềm ẩn rủi ro lỗi.

- Hạn chế về đa nền tảng ban đầu: Mặc dù các phiên bản gần đây như .NET đã cải thiện đáng kể khả năng đa nền tảng, nhưng một số công nghệ cũ của Windows (như WinForms, WPF) vẫn chỉ hoạt động trên Windows.

- Đường cong học tập: Với một hệ sinh thái lớn và nhiều khái niệm phức tạp (như LINQ, async/await), C# có thể khó học đối với người mới bắt đầu, đặc biệt khi phải làm quen với toàn bộ thư viện .NET.

- Tốn tài nguyên hệ thống: Môi trường .NET và Visual Studio có thể tiêu tốn nhiều tài nguyên hệ thống (CPU, RAM) hơn so với một số ngôn ngữ khác.

1.3. Giới thiệu về ASP.NET MVC

ASP.NET MVC là một framework web của Microsoft dùng để xây dựng ứng dụng web dựa trên mô hình MVC (Model-View-Controller). Mô hình này phân tách ứng dụng thành ba thành phần độc lập và tích hợp với các tính năng hiện có của ASP.NET như master pages và cơ chế xác thực. Lập trình theo MVC giúp tách biệt rõ các phần logic giao diện, xử lý input và dữ liệu, giúp quản lý độ phức tạp và dễ dàng kiểm thử hơn:

- Model: Quản lý dữ liệu và logic nghiệp vụ của ứng dụng.
- View: Chịu trách nhiệm hiển thị giao diện người dùng, thường là dưới dạng HTML.
- Controller: Xử lý yêu cầu từ người dùng, tương tác với Model để lấy/cập nhật dữ liệu, và chọn View phù hợp để hiển thị kết quả.

Ví dụ code minh họa:

```
public class HomeController : Controller
{
    // Action Index sẽ được gọi khi người dùng truy cập vào /Home/Index hoặc
    // /Home

    public ActionResult Index()
    {
        // Gửi một thông điệp sang View thông qua ViewBag

        ViewBag.Message = "Welcome to ASP.NET MVC!";
    }
}
```

```
// Trả về View tên "Index" (mặc định dựa theo tên action)

return View();

}}
```

Ưu điểm:

- Quản lý phức tạp tốt hơn nhờ tách biệt rõ ràng Model, View, Controller. Điều này giúp mỗi phần tập trung vào một nhiệm vụ riêng (UI, xử lý input, logic nghiệp vụ), tăng khả năng bảo trì mã nguồn.

- Không sử dụng ViewState hay các Form server-side, cho phép lập trình viên kiểm soát hoàn toàn luồng dữ liệu và hành vi trang. Kết quả là URL đẹp, thân thiện SEO nhờ router linh hoạt.

- Hỗ trợ tốt phát triển theo kiểu kiểm thử (test-driven development) và mô hình Front Controller, rất phù hợp cho các nhóm phát triển lớn cần phân công nhiệm vụ rõ ràng. Tài liệu Microsoft cũng ghi nhận MVC “làm cho kiểm thử ứng dụng dễ dàng hơn” và “phù hợp cho những ứng dụng web được hỗ trợ bởi các nhóm lớn”.

Hạn chế:

- So với Web Forms, MVC đòi hỏi nhiều mã lệnh hơn vì không có cơ chế quản lý trạng thái tự động như ViewState. Tài liệu so sánh của Microsoft ghi rằng Web Forms “thường ít phức tạp hơn, vì các thành phần tích hợp chặt chẽ và thường yêu cầu ít mã hơn MVC”. Do đó, đối với các ứng dụng nhỏ hoặc những nhà phát triển mới, MVC có thể có đường cong học tập cao hơn.

- Vì mất chế độ bảo lưu trạng thái (view state) và event-driven model, việc xử lý các tác vụ tương tác nặng (như forms có nhiều thành phần server) phải viết thủ công nhiều hơn.

Khi nào nên dùng:

- Khi xây dựng các ứng dụng web cần phân tách rõ các tầng logic và hỗ trợ kiểm thử (TDD), ví dụ ứng dụng quy mô lớn hoặc ứng dụng đa tầng. Tài liệu chính thức khuyến nghị dùng MVC cho “những ứng dụng web do các nhóm lớn phát triển và cần kiểm soát cao”.

- Khi cần URL thân thiện cho SEO và muốn có toàn quyền điều khiển cách dữ liệu được xử lý và hiển thị.

- Nếu ứng dụng đòi hỏi đơn giản hóa việc phát triển front-end/back-end tách biệt (ví dụ, giao diện phức tạp qua Razor view).

- Không nên dùng nếu ứng dụng rất đơn giản, cần phát triển nhanh, hoặc dựa nhiều vào cơ chế event và ViewState của Web Forms. Trong các trường hợp này, Web Forms có thể đơn giản và ít mã hơn so với MVC.

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

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ và toàn diện được phát triển bởi Microsoft. Hệ thống này đóng vai trò là kho lưu trữ trung tâm, chịu trách nhiệm quản lý, truy xuất và bảo vệ toàn bộ dữ liệu của ứng dụng. Trong kiến trúc phần mềm của đề tài, SQL Server đảm nhiệm vai trò lưu trữ vật lý cho thành phần Model, chứa đựng các thông tin quan trọng như hồ sơ người dùng, nội dung bài luyện tập và lịch sử kết quả thi đấu. Ngôn ngữ truy vấn chính được sử dụng để tương tác với SQL Server là Transact-SQL (T-SQL), một phiên bản mở rộng của chuẩn SQL do Microsoft phát triển, cung cấp khả năng xử lý dữ liệu mạnh mẽ, hỗ trợ các thủ tục lưu trữ và các hàm định nghĩa người dùng phức tạp.

Việc lựa chọn SQL Server cho đề tài này dựa trên những ưu điểm vượt trội về hiệu năng và tính bảo mật. Về hiệu năng, SQL Server được tối ưu hóa để xử lý hàng triệu giao dịch mỗi ngày với tốc độ cao, đảm bảo rằng việc lưu trữ kết quả gõ phím thời gian thực của người dùng diễn ra tức thì và không có độ trễ. Về bảo mật, hệ thống cung cấp các cơ chế phân quyền chi tiết, mã hóa dữ liệu và kiểm soát truy cập chặt chẽ, giúp bảo vệ thông tin cá nhân của người dùng khỏi các mối đe dọa. Ngoài ra, sự tương thích tuyệt đối giữa SQL Server và nền tảng .NET Framework (bao gồm ASP.NET MVC và ngôn ngữ C#) giúp quá trình phát triển ứng dụng trở nên liền mạch, giảm thiểu các lỗi tương thích và tối ưu hóa thời gian triển khai dự án.

Công cụ chính để quản trị và thao tác với SQL Server trong quá trình phát triển là SQL Server Management Studio (SSMS). Đây là môi trường làm việc tích hợp cho phép nhà phát triển thiết kế lược đồ cơ sở dữ liệu trực quan, viết và tối ưu hóa các câu lệnh truy vấn cũng như giám sát hiệu suất của hệ thống. Nhờ sự kết hợp giữa sức mạnh lưu trữ của SQL Server và công cụ quản trị tiện lợi SSMS, đề tài đảm bảo được tính toàn vẹn dữ liệu và khả năng mở rộng linh hoạt khi số lượng người dùng tăng lên trong tương lai.

Ví dụ code minh họa:

-- Tạo bảng SinhVien để lưu thông tin sinh viên

```
CREATE TABLE SinhVien
```

```
MaSV INT IDENTITY(1,1) PRIMARY KEY, -- Khóa chính tự tăng bắt đầu từ 1
```

```
HoTen NVARCHAR(100),          -- Họ tên sinh viên (dùng NVARCHAR
                                để hỗ trợ Unicode)
NgaySinh DATE,                 -- Ngày sinh
Lop NVARCHAR(20)               -- Lớp của sinh viên
);
GO -- Kết thúc batch
```

Về ưu điểm Microsoft SQL Server được giới công nghệ đánh giá là một trong những hệ quản trị cơ sở dữ liệu quan hệ hàng đầu thế giới nhờ sự ổn định và khả năng tích hợp mạnh mẽ. Điểm mạnh lớn nhất của SQL Server là tính bảo mật vượt trội với các lớp mã hóa dữ liệu trong suốt và cơ chế phân quyền chi tiết, giúp bảo vệ an toàn tuyệt đối cho các thông tin nhạy cảm của hệ thống. Hiệu năng xử lý cũng là một lợi thế cạnh tranh khi SQL Server sở hữu các thuật toán nén dữ liệu và tối ưu hóa truy vấn thông minh, cho phép xử lý hàng triệu giao dịch mỗi giây mà vẫn duy trì độ trễ thấp. Đặc biệt, đối với các nhà phát triển sử dụng nền tảng của Microsoft, SQL Server cung cấp sự tương thích hoàn hảo với .NET Framework và ngôn ngữ C#, giúp quá trình kết nối và thao tác dữ liệu diễn ra mượt mà thông qua Entity Framework hoặc ADO.NET. Bên cạnh đó, bộ công cụ quản trị SQL Server Management Studio đi kèm mang lại trải nghiệm trực quan, giúp việc thiết kế, truy vấn và sao lưu dữ liệu trở nên dễ dàng và hiệu quả hơn rất nhiều so với các hệ quản trị khác.

Tuy nhiên, SQL Server cũng tồn tại những nhược điểm nhất định mà người sử dụng cần cân nhắc. Hạn chế lớn nhất thường được nhắc đến là chi phí bản quyền. Mặc dù phiên bản Express miễn phí phù hợp cho học tập và các ứng dụng nhỏ, nhưng các phiên bản Enterprise dành cho doanh nghiệp lớn lại có chi phí cấp phép rất cao, tạo ra rào cản tài chính không nhỏ cho các công ty khởi nghiệp. Một nhược điểm khác là vấn đề tiêu tốn tài nguyên phần cứng, SQL Server thường yêu cầu cấu hình máy chủ cao với lượng bộ nhớ RAM lớn để duy trì hiệu suất ổn định, đặc biệt là khi cơ sở dữ liệu phình to. Ngoài ra, mặc dù Microsoft đã bắt đầu hỗ trợ Linux trong những năm gần đây, nhưng SQL Server vẫn hoạt động tối ưu nhất và được hỗ trợ đầy đủ tính năng nhất trên hệ điều hành Windows, điều này phần nào hạn chế sự linh hoạt khi triển khai trên các hạ tầng máy chủ đa nền tảng.

Về khả năng ứng dụng thực tiễn, nhờ sự mạnh mẽ và đa năng, SQL Server được ứng dụng rộng rãi trong hầu hết các lĩnh vực công nghệ thông tin. Ứng dụng phổ biến nhất là làm kho lưu trữ dữ liệu (Back-end) cho các website và ứng dụng thương mại điện tử quy mô lớn, nơi yêu cầu tính toàn vẹn dữ liệu và khả năng xử lý giao dịch phức

tập như hệ thống đặt hàng hay thanh toán trực tuyến. Trong môi trường doanh nghiệp, SQL Server là trái tim của các hệ thống quản trị nguồn lực (ERP), quản lý quan hệ khách hàng (CRM) nhờ khả năng lưu trữ và báo cáo mạnh mẽ. Ngoài ra, với sự phát triển của dữ liệu lớn, SQL Server còn được sử dụng trong các hệ thống Phân tích dữ liệu (Data Warehousing) và Trí tuệ doanh nghiệp (Business Intelligence), giúp các tổ chức tổng hợp và khai thác giá trị từ khối lượng dữ liệu khổng lồ để đưa ra các quyết định kinh doanh chính xác.

1.5. Giới thiệu về ADO.NET

ADO.NET là một tập hợp các lớp giúp ứng dụng .NET tương tác với nhiều nguồn dữ liệu khác nhau, như cơ sở dữ liệu SQL Server, Oracle, MySQL. Ưu điểm chính của nó là kiến trúc dữ liệu không kết nối (disconnected) giúp giảm tải cho cơ sở dữ liệu và tiết kiệm tài nguyên, trong khi nhược điểm là việc xử lý dữ liệu có thể phức tạp hơn so với các công nghệ kết nối trực tiếp. ADO.NET được ứng dụng rộng rãi trong các ứng dụng doanh nghiệp, ứng dụng web và các ứng dụng đòi hỏi khả năng truy cập dữ liệu mạnh mẽ.

Ví dụ code minh họa:

```
using System;

using System.Data.SqlClient;

string connStr = "Data Source=(local);Initial
Catalog=Northwind;Integrated Security=true";

using (SqlConnection conn = new SqlConnection(connStr))
{
    SqlCommand cmd = new SqlCommand(
        "SELECT TOP 5 ProductName FROM Products ORDER BY ProductID",
        conn);

    conn.Open();

    SqlDataReader reader = cmd.ExecuteReader();

    while (reader.Read())
    {
        Console.WriteLine(reader.GetString(0));
    }
}
```

Ưu điểm:

- Kiến trúc không kết nối: Dữ liệu được tải vào bộ nhớ, cho phép ứng dụng thao tác mà không cần kết nối liên tục tới cơ sở dữ liệu, giúp giảm tải cho cơ sở dữ liệu và cải thiện hiệu suất.
- Hỗ trợ nhiều nguồn dữ liệu: Cung cấp một cách tiếp cận đồng nhất để truy cập các nguồn dữ liệu khác nhau, bao gồm cơ sở dữ liệu quan hệ và các tài liệu XML.
- Linh hoạt: Cho phép ứng dụng thao tác với dữ liệu trên bộ nhớ cache và chỉ kết nối lại khi cần cập nhật hoặc lấy thêm dữ liệu.

Hạn chế:

- Độ phức tạp: Việc quản lý dữ liệu không kết nối có thể phức tạp hơn so với các phương thức kết nối trực tiếp, đặc biệt là khi cần xử lý các xung đột dữ liệu.
- Xử lý các thao tác phức tạp: Một số thao tác dữ liệu phức tạp có thể cần được triển khai cẩn thận hơn để đảm bảo tính nhất quán của dữ liệu.

Ứng dụng:

- Ứng dụng doanh nghiệp: Xây dựng các ứng dụng doanh nghiệp đòi hỏi khả năng xử lý lượng lớn dữ liệu và tương tác với cơ sở dữ liệu một cách hiệu quả.
- Ứng dụng web: Phát triển các ứng dụng web có thể xử lý nhiều người dùng đồng thời mà không làm quá tải cơ sở dữ liệu.
- Ứng dụng desktop: Tạo các ứng dụng desktop có khả năng truy cập và cập nhật dữ liệu từ nhiều nguồn khác nhau, bao gồm cả các tệp XML.

1.6. Giới thiệu về LINQ to SQL

LINQ to SQL là một thành phần của .NET Framework (từ phiên bản 3.5) cung cấp cơ sở hạ tầng chạy (runtime) để quản lý dữ liệu quan hệ dưới dạng đối tượng trong .NET. Công cụ này ánh xạ cấu trúc cơ sở dữ liệu (bảng, cột, quan hệ) thành mô hình đối tượng (các lớp và thuộc tính C#/VB). Khi ứng dụng chạy, LINQ to SQL sẽ dịch các truy vấn LINQ (ngôn ngữ truy vấn tích hợp) thành câu lệnh SQL tương đương và gửi đến cơ sở dữ liệu để thực thi. Sau đó, kết quả trả về được chuyển ngược thành các đối tượng .NET để lập trình viên làm việc tiếp. Để khai báo ánh xạ, Visual Studio cung cấp LINQ to SQL Classes (.dbml) hoặc công cụ SqlMetal: chúng trích xuất thông tin lược đồ CSDL vào một tệp XML định dạng DBML, sau đó sinh mã class (DataContext và các entity) dựa trên file DBML này.

Ví dụ code minh họa:

```
// Khởi tạo DataContext, kết nối đến file cơ sở dữ liệu Northwind
DataContext db = new DataContext(@"c:\data\northwind.mdf");

// Lấy bảng Customer từ cơ sở dữ liệu dưới dạng Table<Customer>
Table<Customer> customers = db.GetTable<Customer>();

// Tạo truy vấn LINQ: lấy các khách hàng có City = "London"
var query = from c in customers

              where c.City == "London"

              select c;

// Duyệt từng phần tử trong kết quả truy vấn
foreach (Customer cust in query)
{
    // In ra ID và thành phố của khách hàng
    Console.WriteLine($"ID={cust.CustomerID}, City={cust.City}");
}
```

Ưu điểm:

- LINQ to SQL tích hợp sẵn trong .NET, cho phép lập trình kiểu LINQ thuận tiện khi truy vấn dữ liệu SQL Server dưới dạng đối tượng. Điều này giảm thiểu mã SQL thủ công và tận dụng tính năng IntelliSense của IDE. Nó “phù hợp với các kịch bản cơ bản” và được hỗ trợ trong Visual Studio.

- Thích hợp với các ứng dụng nhỏ hoặc legacy: LINQ to SQL rất dễ thiết lập bằng cách kéo-thả bảng trong Object-Relational Designer (.dbml), tự động sinh class và DataContext. Tài liệu Microsoft nhấn mạnh LINQ to SQL “được dùng khi duy trì ứng dụng kế thừa” hoặc trong “ứng dụng đơn giản sử dụng SQL Server và không cần ánh xạ nhiều bảng”.

Nhược điểm:

- Không còn được phát triển tích cực: LINQ to SQL đã được Microsoft thay thế dần bằng Entity Framework. Tài liệu lưu ý đây là “công nghệ ORM đầu tiên” nhưng “không được phát triển thêm”. Do đó, nó thiếu nhiều tính năng mới của Entity Framework.

- Hạn chế về ánh xạ: Công cụ O/R Designer của LINQ to SQL “chỉ hỗ trợ ánh xạ 1:1” giữa lớp thực thể và bảng cơ sở dữ liệu. Các ánh xạ phức tạp (ví dụ ánh xạ một đối tượng ra nhiều bảng hay một kết quả JOIN) không được hỗ trợ trong trường hợp đó Microsoft khuyên dùng Entity Framework.

Khi nào nên dùng:

- Dùng khi cần phát triển nhanh các ứng dụng tương tác dữ liệu đơn giản với SQL Server, đặc biệt khi đang duy trì những ứng dụng cũ đã dùng LINQ to SQL. Ví dụ, dự án nhỏ không đòi hỏi nhiều mối quan hệ phức tạp giữa các bảng.

- Khi muốn tận dụng LINQ để viết truy vấn trên C# mà không cần sử dụng ORM nặng như EF. Đặc biệt, tài liệu đề xuất LINQ to SQL cho “ứng dụng kế thừa” hoặc “ứng dụng đơn giản” với SQL Server và không yêu cầu ánh xạ nhiều bảng.

- Không nên dùng cho ứng dụng mới đòi hỏi kiến trúc phức tạp: nếu cần ánh xạ nhiều bảng/phức hợp, hoặc muốn hỗ trợ đa nền tảng và cập nhật lâu dài, nên sử dụng Entity Framework.

1.7. Giới thiệu về ngôn ngữ JavaScript

JavaScript là một ngôn ngữ lập trình bậc cao, đa năng và năng động, được xem là một trong ba trụ cột công nghệ cốt lõi của World Wide Web bên cạnh HTML và CSS. Ban đầu được phát triển bởi Netscape với mục đích tạo ra các tương tác động trên trình duyệt web, JavaScript hiện nay đã phát triển vượt bậc và được chuẩn hóa dưới tên gọi ECMAScript. Đặc điểm nổi bật nhất của ngôn ngữ này là khả năng chạy trực tiếp trên trình duyệt của người dùng mà không cần biên dịch trước, giúp biến các website tĩnh thành các ứng dụng web tương tác phong phú. Thông qua việc thao tác với mô hình đối tượng tài liệu JavaScript cho phép thay đổi nội dung hiển thị, xử lý các sự kiện người dùng như nhấp chuột hay gõ phím và cập nhật dữ liệu bất đồng bộ mà không cần tải lại toàn bộ trang.

Về mặt kỹ thuật, JavaScript là một ngôn ngữ đa mẫu hình, hỗ trợ cả lập trình hướng sự kiện, lập trình hàm và lập trình hướng đối tượng dựa trên nguyên mẫu (prototype-based). Một trong những thế mạnh lớn nhất của JavaScript là cơ chế xử lý bất đồng bộ thông qua các khái niệm như Callbacks, Promises và Async/Await. Cơ chế này cho phép thực hiện các tác vụ tốn thời gian như gọi API hoặc truy xuất dữ liệu từ máy chủ mà không làm chặn luồng chính của giao diện, đảm bảo trải nghiệm người dùng luôn mượt mà và không bị gián đoạn. Hơn nữa, với sự ra đời của môi trường thực thi Node.js, JavaScript đã vượt ra khỏi phạm vi trình duyệt để hoạt động mạnh mẽ trên máy chủ (server-side), cho phép các nhà phát triển sử dụng một ngôn ngữ duy nhất cho

toàn bộ hệ thống từ frontend đến backend.

Trong phạm vi của đề án này, JavaScript đóng vai trò không thể thay thế trong việc xây dựng các chức năng tương tác cốt lõi tại phía máy khách. Ngôn ngữ này chịu trách nhiệm bắt các sự kiện bàn phím theo thời gian thực để xử lý logic của bài luyện gõ, tính toán các chỉ số tốc độ gõ và độ chính xác ngay lập tức. Đồng thời, JavaScript kết hợp với các công nghệ như AJAX và SignalR để thực hiện việc truyền tải dữ liệu hai chiều liên tục với máy chủ ASP.NET MVC, đảm bảo các tính năng như cập nhật bảng xếp hạng hay phòng chờ lớp học diễn ra tức thời. Sự linh hoạt và hệ sinh thái thư viện phong phú của JavaScript chính là chìa khóa để tạo nên một giao diện người dùng thân thiện, hiện đại và có tính phản hồi cao cho hệ thống.

Ưu điểm:

- Tốc độ thực thi cao phía máy khách: Mã nguồn JavaScript được chạy trực tiếp ngay trên trình duyệt của người dùng mà không cần phải biên dịch trước hay gửi yêu cầu về máy chủ, giúp giảm thiểu độ trễ và tiết kiệm băng thông đáng kể.
- Tính tương thích và phổ biến: Ngôn ngữ này hoạt động mượt mà trên mọi trình duyệt web hiện đại và dễ dàng tích hợp với các ngôn ngữ khác như HTML hay PHP để tạo nên các ứng dụng web phong phú.
- Hệ sinh thái không lồ: Sở hữu kho thư viện và khung làm việc (framework) phong phú như React, Angular hay Vue, giúp rút ngắn thời gian phát triển sản phẩm.
- Khả năng hoạt động đa năng: Với sự hỗ trợ của môi trường Node.js, JavaScript cho phép xây dựng trọn vẹn cả phần giao diện người dùng và phần máy chủ, tạo nên sự đồng nhất về ngôn ngữ cho toàn bộ dự án.

Nhược điểm:

- Vấn đề bảo mật phía máy khách: Do mã nguồn được tải về và thực thi công khai trên trình duyệt nên rất dễ bị kẻ xấu khai thác hoặc chen mã độc nếu không có các biện pháp kiểm soát chặt chẽ.
- Sự phân mảnh trình duyệt: Mặc dù đã có các chuẩn chung nhưng đôi khi cách thông dịch mã JavaScript vẫn có sự khác biệt nhỏ giữa các trình duyệt khác nhau, dẫn đến việc giao diện hiển thị không đồng nhất.
- Khó khăn trong gỡ lỗi: Vì JavaScript là ngôn ngữ định kiểu động nên các lỗi liên quan đến dữ liệu thường chỉ phát sinh khi chương trình đang chạy thay vì được phát hiện ngay lúc biên dịch.

Ứng dụng:

- Phát triển giao diện web tương tác: Đây là ứng dụng phổ biến nhất để tạo ra các hiệu ứng động, kiểm tra dữ liệu đầu vào và xử lý sự kiện người dùng ngay tại trình duyệt.
- Lập trình máy chủ và thời gian thực: Thông qua Node.js, JavaScript được dùng để xây dựng các hệ thống yêu cầu tốc độ cao như ứng dụng trò chuyện trực tuyến hay các dịch vụ phát trực tiếp.
- Phát triển ứng dụng đa nền tảng: Sử dụng các công nghệ như React Native hay Electron để phát triển ứng dụng di động và phần mềm máy tính để bàn, cho phép viết mã một lần nhưng chạy được trên nhiều hệ điều hành khác nhau.

1.8. Giới thiệu về Bootstrap

Bootstrap là một framework phát triển giao diện người dùng (frontend) mã nguồn mở và miễn phí phổ biến nhất hiện nay. Được phát triển bởi Twitter, Bootstrap cung cấp một bộ sưu tập các công cụ bao gồm HTML, CSS và JavaScript để giúp lập trình viên thiết kế các website có khả năng tương thích với nhiều thiết bị (responsive) và ưu tiên giao diện di động (mobile-first).

Ưu điểm

- Tốc độ phát triển nhanh chóng: Bootstrap cung cấp sẵn một thư viện khổng lồ các thành phần giao diện được thiết kế trước như thanh điều hướng, nút bấm, biểu mẫu và các hộp thoại, giúp lập trình viên tiết kiệm thời gian viết mã từ đầu
- Hệ thống lưới (Grid System) linh hoạt: Đây là tính năng cốt lõi giúp chia bố cục website thành các hàng và cột thông minh, đảm bảo giao diện tự động co giãn và hiển thị đẹp mắt trên mọi kích thước màn hình từ điện thoại đến máy tính để bàn
- Tính tương thích trình duyệt cao: Các thành phần của Bootstrap đã được kiểm thử kỹ lưỡng để đảm bảo hoạt động ổn định và đồng nhất trên tất cả các trình duyệt phổ biến như Chrome, Firefox, Safari và Edge
- Dễ dàng tùy biến và mở rộng: Mặc dù cung cấp giao diện mặc định, Bootstrap cho phép lập trình viên dễ dàng tùy chỉnh màu sắc, kích thước và kiểu dáng thông qua các biến SASS hoặc tập tin cấu hình riêng

Nhược điểm

- Thiết kế dễ bị rập khuôn: Do quá phổ biến và có phong cách thiết kế đặc trưng, các website sử dụng Bootstrap thường có giao diện trông khá giống nhau nếu lập trình

viên không tùy biến kỹ, làm giảm tính độc đáo của thương hiệu

- Dung lượng tệp tin lớn: Bộ mã nguồn đầy đủ của Bootstrap chứa rất nhiều dòng mã CSS và JavaScript, bao gồm cả những thành phần mà dự án có thể không bao giờ dùng đến, gây ảnh hưởng nhẹ đến tốc độ tải trang nếu không được tối ưu hóa

- Khó khăn khi ghi đè kiểu dáng: Việc thay đổi sâu các thiết kế mặc định đôi khi khá phức tạp và đòi hỏi lập trình viên phải sử dụng nhiều quy tắc CSS ghi đè (override) hoặc sử dụng !important, làm mã nguồn trở nên cồng kềnh

Ứng dụng

- Xây dựng trang quản trị: Bootstrap cực kỳ mạnh mẽ trong việc tạo ra các giao diện quản lý dữ liệu với hệ thống bảng biểu, biểu đồ và các thành phần nhập liệu chuẩn mực, rất phù hợp cho phần trang quản trị của đồ án này

- Phát triển sản phẩm mẫu: Nhờ tốc độ triển khai nhanh, Bootstrap là lựa chọn hàng đầu cho các công ty khởi nghiệp muốn nhanh chóng ra mắt phiên bản thử nghiệm của sản phẩm

- Thiết kế website đáp ứng: Được sử dụng rộng rãi cho các trang tin tức, thương mại điện tử và blog cá nhân yêu cầu khả năng hiển thị tốt trên thiết bị di động mà không cần xây dựng phiên bản mobile riêng biệt

1.9. Giới thiệu về SignalR

SignalR là một thư viện mã nguồn mở dành cho các nhà phát triển ASP.NET, được thiết kế với mục đích đơn giản hóa việc thêm các chức năng web thời gian thực vào ứng dụng. Khái niệm thời gian thực ở đây được hiểu là khả năng máy chủ chủ động đẩy nội dung đến các máy khách đã kết nối ngay lập tức thay vì phải chờ máy khách gửi yêu cầu mới cập nhật dữ liệu. Cơ chế hoạt động cốt lõi của SignalR dựa trên việc trừu tượng hóa các kết nối mạng thông qua một API cấp cao gọi là Hub. Hub cho phép máy chủ và máy khách giao tiếp với nhau trực tiếp thông qua cơ chế Gọi thủ tục từ xa, nghĩa là máy chủ có thể gọi hàm JavaScript trên trình duyệt của người dùng và ngược lại, trình duyệt cũng có thể gọi các phương thức C# trên máy chủ một cách mượt mà. Bên dưới lớp trừu tượng này, SignalR tự động quản lý kết nối và lựa chọn phương thức truyền tải tối ưu nhất dựa trên khả năng của trình duyệt và máy chủ, ưu tiên sử dụng WebSockets nếu có hỗ trợ hoặc tự động chuyển sang các kỹ thuật khác như Server-Sent Events hay Long Polling nếu cần thiết.

Ưu điểm:

- Dễ dàng thêm tính năng real-time: SignalR cung cấp API để server gọi hàm trên

client JavaScript, tự động quản lý kết nối và cho phép đẩy dữ liệu tới client mà không cần client hỏi liên tục. Điều này thuận tiện cho các ứng dụng như chat trực tuyến, dashboard cập nhật liên tục, thông báo thời gian thực.

- Hỗ trợ nhiều loại transport và scale-out: SignalR dùng WebSocket khi được hỗ trợ, và tự động fallback sang các công nghệ HTTP long-polling hay Server-Sent Events nếu trình duyệt không hỗ trợ WebSocket. Bên cạnh đó, SignalR có các provider mở rộng (Service Bus, Redis, SQL Server) cho phép scale-out xử lý hàng nghìn kết nối song song.

- API linh hoạt cho broadcast và nhóm: SignalR cho phép gửi tin nhắn tới tất cả client, tới từng client riêng lẻ, hoặc nhóm client theo nhóm tùy ý. Tài liệu ghi rõ SignalR “cho phép broadcast tin nhắn tới tất cả client cùng lúc” như một phòng chat chung.

Nhược điểm:

- Phiên bản ASP.NET SignalR (2.x) chỉ chạy trên .NET Framework cũ và Windows, không còn được phát triển tích cực. Tài liệu Microsoft cảnh báo rằng đây không phải phiên bản mới nhất và khuyến cáo chuyển sang ASP.NET Core SignalR cho ứng dụng mới. Điều này nghĩa là SignalR 2.x không hỗ trợ cross-platform tốt và có khả năng thiếu cập nhật bảo mật hay tính năng mới so với bản Core.

- Phụ thuộc vào WebSocket/long-polling: Mặc dù thuận tiện, nhưng việc phải dùng nhiều cơ chế dự phòng có thể gây phức tạp khi cấu hình, và nếu môi trường máy chủ không hỗ trợ (ví dụ .NET Framework < 4.5 không có WebSocket), sẽ phải dùng cơ chế chậm hơn (long polling).

Ứng dụng:

- Dùng khi ứng dụng web cần tính năng real-time, tức là cần cập nhật dữ liệu ngay lập tức từ server khi có thay đổi. Ví dụ, ứng dụng chat, hệ thống theo dõi (dashboard) hiện số liệu tự động, ứng dụng cộng tác trực tuyến, cập nhật tiến độ công việc, biểu mẫu theo dõi kết quả thời gian thực... Tài liệu nêu rõ mọi tình huống bất cứ khi nào người dùng phải refresh trang để xem dữ liệu mới đều có thể sử dụng SignalR.

- Phù hợp khi cần tần suất cập nhật cao: ví dụ game trực tuyến thời gian thực hay hiển thị dữ liệu cảm biến live. SignalR được thiết kế cho phép các loại ứng dụng mới với cập nhật tần suất cao, như game thời gian thực.

- Không nên dùng nếu chỉ cần cập nhật thưa hoặc nếu xây dựng ứng dụng mới trên .NET Core (khi đó nên dùng ASP.NET Core SignalR).

CHƯƠNG 2: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Mô tả bài toán

Xây dựng một nền tảng luyện gõ bàn phím trực tuyến không chỉ là một dự án công nghệ, mà còn là một cơ hội để thúc đẩy việc nâng cao kỹ năng gõ bàn phím và trải nghiệm học tập. Nền tảng này sẽ trở thành nguồn tài liệu phong phú và đa dạng, giúp người học tiến bộ và tìm thấy niềm vui trong việc cải thiện kỹ năng gõ bàn phím của họ.

Dự án bắt đầu với việc tạo ra một môi trường đăng ký và đăng nhập an toàn và thuận tiện. Tính năng này sẽ giúp người dùng lưu trữ kết quả của họ, quản lý tiến trình học tập và theo dõi tiến độ của bản thân. Khi quên mật khẩu người dùng có thể chọn tính năng quên mật khẩu ở màn hình đăng nhập.

Các bài kiểm tra gõ bàn phím có thể bao gồm nhiều cấp độ khác nhau và các loại bài tập khác nhau, từ những bài tập đơn giản cho người mới bắt đầu đến những thử thách phức tạp và thú vị dành cho những người đã có kinh nghiệm. Tính năng này giúp người học kiểm tra kỹ năng cơ bản và mở rộng kỹ năng của họ thông qua việc thực hành với các loại bài tập khác nhau.

Sau mỗi lần làm bài, người dùng sẽ nhận được phản hồi về kết quả và tiến bộ của mình, đồng thời có thể xem lại các bài tập và câu trả lời. Điều này giúp người học hiểu rõ hơn về những lỗi thường gặp và cách cải thiện kỹ năng của mình.

Người dùng có thể chơi các trò chơi nhỏ để giải trí cũng như là luyện tốc độ phản ứng khi gõ bàn phím giúp nâng cao trải nghiệm người dùng.

Các lớp học được người dùng là Giáo viên tạo ra và có thể mời người dùng khác là Học sinh tham gia vào lớp học thông qua mã lớp hoặc gmail lời mời. Ở trong lớp mọi người có thể cùng luyện gõ với nhau thông qua một đoạn văn bản mà Giáo viên đã tạo sẵn có thể tự nghĩ ra hoặc dùng AI đã được tích hợp để sinh ra nội dung khi nhập chủ đề bất kì được coi là một bài thi dành cho Học sinh khi tham gia vào lớp học.

Người dùng có thành tích tốt khi luyện tập sẽ được vào Bảng xếp hạng theo thứ tự từ thấp đến cao nhằm tăng tính cạnh tranh tích cực cho người dùng.

Ngoài ra, tính năng Chatbot AI được tích hợp sẽ có nhiệm vụ hỗ trợ trả lời các câu hỏi của người dùng. Phân tích các lỗi sai khi luyện gõ của người dùng và đưa ra các bài luyện tập thích hợp cho người dùng để cải thiện những lỗi đó.

Phản quản lý sẽ giúp quản trị viên dễ dàng quản lý dữ liệu, tạo ra các bài tập mới và xem thống kê về tiến độ học tập và thành tích của người học. Quản lý người dùng sẽ giúp quản trị viên tương tác và hỗ trợ người dùng một cách hiệu quả.

Tất cả các tính năng và chức năng này đều hướng tới mục tiêu cuối cùng là giúp người học phát triển kỹ năng gõ bàn phím và tạo ra một môi trường học tập trực tuyến tích cực và an toàn.

2.2. Khảo sát và xác định yêu cầu của hệ thống

2.2.1. Khảo sát hệ thống

Trong quá trình phát triển website gõ bàn phím trực tuyến, việc thiếu một mô hình cụ thể để khảo sát và áp dụng trực tiếp đã thúc đẩy em tiến hành tìm hiểu chuyên sâu các hệ thống tương tự trên mạng Internet. Quá trình này được thực hiện bằng cách nghiên cứu và học hỏi thêm từ hai website hàng đầu trong lĩnh vực luyện gõ bàn phím này là TypingTop.com và 10FastFingers.com.

Việc khảo sát đã tập trung vào ba khía cạnh cốt lõi. Đầu tiên, em nghiên cứu quy trình phát triển và thiết kế website, tìm hiểu về cách thức các nền tảng này được xây dựng từ ý tưởng ban đầu đến khi hoàn thiện, bao gồm cả cách họ thiết kế trải nghiệm người dùng để tạo ra sự hấp dẫn và trực quan.

Thứ hai, em đã khảo sát về công nghệ phát triển web được sử dụng, bao gồm cả các công nghệ phía frontend như HTML, CSS, JavaScript và các công nghệ phía server như Node.js hay Python. Em đặc biệt chú trọng đến việc các hệ thống này đã ứng dụng mô hình MVC (Model-View-Controller) như thế nào để đảm bảo mã nguồn được quản lý một cách có tổ chức và hiệu quả.

Cuối cùng, em tìm hiểu về cách thức triển khai và quản lý nội dung, bao gồm việc cập nhật, duy trì các bài test gõ bàn phím và cách tổ chức cơ sở dữ liệu để người dùng có thể truy cập hàng ngàn bài tập khác nhau.

Bên cạnh việc phân tích các hệ thống, em còn tham khảo kinh nghiệm từ những người đi trước trong cùng lĩnh vực. Những bài học này đã cung cấp những góc nhìn thực tiễn, giúp em tối ưu hóa quy trình làm việc và đảm bảo chất lượng cho dự án của mình. Nhờ đó, em có thể rút ra những ưu điểm nổi bật và tránh những sai lầm phổ biến, từ đó xây dựng một sản phẩm không chỉ đáp ứng yêu cầu kỹ thuật mà còn mang lại giá trị thực tiễn cao cho người dùng.

2.2.2. Xác định yêu cầu hệ thống

Sau khi khảo sát bối cảnh, xác định mục tiêu và phạm vi, phần này đi vào đặc tả

chi tiết các yêu cầu mà hệ thống phải đáp ứng. Yêu cầu hệ thống được chia thành hai nhóm chính là Yêu cầu chức năng và Yêu cầu phi chức năng.

2.2.2.3. Yêu cầu chức năng

Yêu cầu chức năng đặc tả các nghiệp vụ cụ thể mà hệ thống phải thực thi, được phân nhóm logic thành năm nhóm chức năng chính.

Đầu tiên là nhóm nghiệp vụ Quản lý Định danh chịu trách nhiệm cho toàn bộ vòng đời tài khoản. Hệ thống phải đảm bảo các quy trình đăng ký, bao gồm cả việc xác thực qua thư điện tử, đăng nhập, đăng xuất và khôi phục mật khẩu diễn ra một cách an toàn. Đây là nền tảng bắt buộc để cá nhân hóa trải nghiệm và lưu trữ tiến độ của người dùng.

Tiếp theo là nhóm chức năng Luyện tập và Tương tác đây là chức năng cốt lõi của hệ thống. Yêu cầu cốt lõi là cho phép người dùng chọn bài gõ, thực hiện đo lường chính xác tốc độ gõ và độ chính xác, sau đó tự động lưu kết quả. Để duy trì động lực, nhóm chức năng này phải tích hợp các yếu tố trò chơi hóa như các trò chơi nhỏ, các cuộc thi gõ có bảng xếp hạng, cùng khả năng cho phép người dùng đóng góp nội dung luyện tập.

Nhóm chức năng tích hợp Chatbot sử dụng Trí tuệ nhân tạo thể hiện tính hiện đại của đề tài. Bằng cách sử dụng các công nghệ trí tuệ nhân tạo tiên tiến, hệ thống phải cung cấp một trợ lý ảo thông minh để hỗ trợ người dùng liên tục. Đồng thời, hệ thống phải có chức năng đề xuất tự tạo văn bản bằng Trí tuệ nhân tạo, cho phép người dùng sinh nội dung luyện tập vô hạn từ các chủ đề tùy chỉnh mà người dùng nhập.

Nhóm chức năng Quản lý Lớp học mở rộng hệ thống thành một công cụ hỗ trợ học tập trực tuyến. Yêu cầu này cho phép Giáo viên tạo lớp học, nhận mã lớp, và mời học viên thông qua thư điện tử hoặc mã lớp. Giáo viên phải có quyền gán bài tập và sử dụng một bảng điều khiển riêng để theo dõi, giám sát chi tiết lớp học.

Cuối cùng, nhóm chức năng Quản trị Hệ thống đảm bảo khả năng vận hành và kiểm soát. Quản trị viên phải có toàn quyền quản lý tài khoản người dùng, bao gồm cả việc khóa hoặc thêm mới, chỉnh sửa, cập nhật và xóa tài khoản người dùng. Hơn nữa, Quản trị viên phải có khả năng quản lý toàn bộ nội dung, thực hiện các thao tác như thêm mới, chỉnh sửa, xóa bài luyện tập, cuộc thi, văn bản và phải thực hiện nghiệp vụ kiểm duyệt các nội dung do người dùng đóng góp trước khi cho phép hiển thị công khai.

2.2.2.4. Yêu cầu phi chức năng

Những yêu cầu này đảm bảo chất lượng, hiệu suất và tính ổn định của hệ thống:

Bảo mật: HTTPS bắt buộc, CSRF/XSS/SQLi protection, Hash mật khẩu (BCrypt ≥ 10 rounds). RBAC (User/Teacher/Admin), throttle đăng nhập, audit log.

Hiệu năng: P95 response < 800ms cho trang nghiệp vụ, < 2s cho truy vấn báo cáo.

Cache tĩnh (bundling/minify), paginate list.

Khả dụng & ổn định: Xử lý lỗi AI/Gmail graceful, retry queue cho Email. Log và giám sát (Serilog/ELK).

Khả mở rộng: Dịch vụ AI và gửi Email tách thành service, hàng đợi gửi mail. DB index cho bảng kết quả/nhật ký.

Khả bảo trì: Kiến trúc MVC rõ: Controller mỏng, Service/Repository. Unit Test cho logic tính điểm, anti-cheat.

Quy định & dữ liệu cá nhân: Tuân thủ bảo mật thông tin người dùng, cho phép xóa tài khoản theo yêu cầu. Ẩn Email khi public (masking).

Quản lý quyền hạn: Quyền hạn người dùng được phân định rõ ràng, đảm bảo mỗi người dùng chỉ có quyền truy cập vào dữ liệu và chức năng phù hợp với vai trò và nhiệm vụ của họ. Việc tạo mới và cập nhật dữ liệu chỉ được thực hiện bởi những người có quyền hạn tương ứng, để ngăn chặn sự truy cập trái phép hoặc sửa đổi dữ liệu không phù hợp.

2.2.3. Phân tích yêu cầu hệ thống

Sau khi khảo sát và xác định được các yêu cầu của hệ thống, có thể phân tích để đưa ra các quy trình nghiệp vụ chung của hệ thống bao gồm các phần:

2.2.3.1. Chức năng Đăng ký tài khoản

Mục tiêu của chức năng là cho phép người dùng vắng lai (Guest) tạo tài khoản mới và đảm bảo tính sở hữu Email thông qua quy trình kích hoạt.

Đầu vào là thông tin người dùng (Username, Email, Mật khẩu) từ biểu mẫu đăng ký.

Đầu ra bao gồm một bản ghi người dùng mới (chưa xác thực) trong CSDL, một Email chứa liên kết kích hoạt được gửi đến người dùng, và thông báo yêu cầu kiểm tra Email.

Chức năng này khởi tạo luồng nghiệp vụ đăng ký. Hệ thống thực thi validation (kiểm tra định dạng Email, độ mạnh mật khẩu) và đảm bảo tính duy nhất của Email. Khi

thông tin hợp lệ, một bản ghi người dùng mới được tạo với trạng thái `IsVerified=false` và mật khẩu được mã hóa (hash). Đồng thời, hệ thống sinh một token xác thực duy nhất (có giới hạn thời gian) và gửi một Email chứa liên kết kích hoạt đến địa chỉ người dùng cung cấp qua Gmail Service.

2.2.3.2. Chức năng Xác thực Tài khoản qua Email

Mục tiêu là kích hoạt tài khoản người dùng khi họ xác nhận sở hữu địa chỉ Email đã đăng ký.

Đầu vào là token xác thực (thường nằm trên tham số của URL mà người dùng nhập vào).

Đầu ra là trạng thái tài khoản người dùng được cập nhật (`IsVerified=true`), token bị vô hiệu hóa, và thông báo Xác thực thành công hiển thị cho người dùng.

Đây là nghiệp vụ tiếp nối của quá trình đăng ký. Khi người dùng nhấp vào liên kết kích hoạt, hệ thống tiếp nhận yêu cầu chứa token. Máy chủ thực hiện tra cứu và xác thực token kiểm tra tính tồn tại và thời hạn hiệu lực. Nếu token hợp lệ, trạng thái tài khoản người dùng được cập nhật và token bị vô hiệu hoá để đảm bảo chỉ sử dụng một lần.

2.2.3.3. Chức năng Đăng nhập và Đăng xuất

Mục tiêu của chức năng này là xác thực danh tính người dùng đã đăng ký và quản lý phiên làm việc của họ.

Đầu vào Đăng nhập là thông tin xác thực username/Email, mật khẩu. Đầu vào Đăng xuất là yêu cầu đăng xuất.

Đầu ra Đăng nhập là một phiên làm việc Session/Cookie được tạo, và người dùng được điều hướng đến trang cá nhân. Đầu ra (Đăng xuất là phiên làm việc bị hủy, và người dùng được điều hướng về trang chủ.

Khi đăng nhập, hệ thống tiến hành so khớp mật khẩu đã mã hóa (BCrypt). Nếu xác thực thành công và tài khoản không bị khóa (`IsLocked=false`), hệ thống sẽ tạo phiên làm việc và gán vai trò (Role) tương ứng. Chức năng Đăng xuất cho phép người dùng chủ động hủy phiên làm việc và xóa cookie xác thực.

2.2.3.4. Chức năng Quên mật khẩu

Mục tiêu là cung cấp quy trình an toàn cho phép người dùng đặt lại mật khẩu khi họ quên.

Đầu vào bao gồm địa chỉ Email đã đăng ký (để yêu cầu reset) và Mật khẩu mới

cùng Mật khẩu xác nhận.

Đầu ra là một Email chứa liên kết đặt lại mật khẩu (gửi cho người dùng), mật khẩu mới được cập nhật trong CSDL, và thông báo “Đặt lại mật khẩu thành công”.

Quy trình này gồm hai bước: (1) Người dùng nhập Email, hệ thống xác thực và gửi ResetToken (dùng một lần, thời hạn ngắn) qua Email. (2) Người dùng truy cập liên kết, nhập mật khẩu mới. Hệ thống validate token và cập nhật PasswordHash trong cơ sở dữ liệu.

2.2.3.5. Chức năng Duyệt và Chọn bài luyện tập

Mục tiêu là cung cấp giao diện cho người dùng tìm kiếm, lọc và chọn bài gõ bàn phím từ thư viện nội dung.

Đầu vào là các tham số lọc (ngôn ngữ, độ khó, thể loại) và thao tác chọn bài luyện tập.

Đầu ra là danh sách bài luyện tập đã được lọc và phân trang, cùng với nội dung chi tiết của bài luyện được chọn.

Chức năng này cho phép người dùng (bao gồm cả Khách, nếu được cho phép) duyệt thư viện bài luyện. Khi người dùng chọn một bài cụ thể, hệ thống truy vấn và hiển thị nội dung của bài luyện đó, chuẩn bị cho phiên gõ bàn phím.

2.2.3.6. Chức năng Luyện gõ và Lưu kết quả

Dữ liệu đầu vào là một gói tin được gửi từ client-side sau khi người dùng kết thúc phiên luyện tập. Gói tin này chứa các chỉ số hiệu suất đã được tính toán (bao gồm WPM, Độ chính xác, Sai sót, Thời gian hoàn thành), ExerciseId, và UserId (được lấy từ phiên đăng nhập hiện tại của người dùng).

Đầu ra là một bản ghi mới được khởi tạo và lưu trữ thành công trong bảng PracticeResults của cơ sở dữ liệu, liên kết chính xác UserId và ExerciseId. Hệ thống đồng thời trả về một phản hồi (HTTP Response) cho client để xác nhận việc lưu trữ thành công hoặc thông báo lỗi nếu dữ liệu không vượt qua được bước validation (ví dụ: bị phát hiện bởi cơ chế anti-cheat).

Đây là chức năng cốt lõi, cho phép đo lường và ghi nhận hiệu suất luyện tập. Khi người dùng thực hiện bài gõ, giao diện phía client-side sẽ tính toán các chỉ số thời gian thực (WPM, Độ chính xác, Sai sót). Sau khi hoàn thành, dữ liệu tổng hợp được gửi lên máy chủ. Hệ thống thực hiện validation dữ liệu (áp dụng cơ chế anti-cheat để loại bỏ kết quả bất thường) và lưu trữ vào bảng PracticeResults, liên kết chặt chẽ với UserId và

ExerciseId để phục vụ thống kê tiến độ sau này.

2.2.3.7. Chức năng Trò chơi luyện gõ

Dữ liệu đầu vào là kết quả cuối cùng của phiên chơi được gửi từ client khi trò chơi kết thúc. Dữ liệu này bao gồm điểm số đạt được, thời gian chơi, userId, và gametype.

Đầu ra là một bản ghi mới được tạo và lưu trữ trong bảng GameSessions của cơ sở dữ liệu. Hệ thống trả về phản hồi xác nhận việc lưu điểm thành công. Dữ liệu Bảng xếp hạng liên quan đến trò chơi này được cập nhật và có thể được gửi lại cho client để hiển thị.

Nhằm đa dạng hóa trải nghiệm, hệ thống cung cấp các trò chơi luyện gõ (ví dụ: gõ từ/ký tự rơi). Chức năng này tập trung vào phản xạ và tính điểm dựa trên thời gian. Phía client chịu trách nhiệm chính trong việc chạy vòng lặp game. Khi kết thúc hết giờ hoặc thua, điểm số và thời gian được gửi về máy chủ. Hệ thống lưu kết quả vào bảng GameSessions và có thể hiển thị trên bảng xếp hạng riêng của trò chơi. Các cơ chế chống gian lận (debounce, giới hạn tần suất gửi) được áp dụng.

2.2.3.8. Chức năng Tạo văn bản bằng AI

Mục tiêu của chức năng là tích hợp mô hình ngôn ngữ lớn tiên tiến để tự động sinh nội dung luyện tập mới, độc đáo dựa trên yêu cầu (prompt) của người dùng.

Đầu vào là Chủ đề và các tham số tùy chọn (độ dài, ngôn ngữ, mức độ phức tạp) do người dùng nhập.

Đầu ra là một đoạn văn bản do AI tạo ra một bài luyện tập mới được lưu từ nội dung này.

Chức năng này thực hiện một lệnh gọi API bất đồng bộ đến dịch vụ Gemini (Google AI). Hệ thống gửi prompt của người dùng đến mô hình. Văn bản do AI trả về sẽ được hệ thống kiểm duyệt (lọc nội dung nhạy cảm) trước khi hiển thị. Chức năng này tuân thủ các ràng buộc về chi phí (quota API), chính sách an toàn, và cung cấp tùy chọn cho phép người dùng lưu nội dung AI tạo ra thành một bài luyện tập cá nhân.

2.2.3.9. Chức năng Tham gia Cuộc thi

Mục tiêu là cho phép người dùng thi đấu kỹ năng gõ bàn phím trong một sự kiện có tổ chức, có giới hạn thời gian và được xếp hạng.

Đầu vào là yêu cầu ghi danh tham gia cuộc thi và kết quả bài thi (WPM, Độ chính xác) khi người dùng nộp bài.

Đầu ra là bản ghi CompetitionParticipant (lưu kết quả thi) và Bảng xếp hạng của cuộc thi được cập nhật.

Hệ thống cho phép Admin tạo các cuộc thi. Người dùng có thể ghi danh vào các cuộc thi đang diễn ra. Khi nộp bài, hệ thống tiếp nhận kết quả, áp dụng luật của cuộc thi (ví dụ: giới hạn số lần nộp) và cập nhật bảng xếp hạng.

2.2.3.10. Chức năng Tạo và Chia sẻ văn bản

Mục tiêu là xây dựng tính cộng đồng bằng cách cho phép người dùng tạo văn bản và đóng góp nội dung bài luyện tập, với sự kiểm duyệt của Admin.

Đầu vào là Tiêu đề, Nội dung, Ngôn ngữ, Thẻ loại, người dùng có thể nhập chủ đề yêu thích để sử dụng trí tuệ nhân tạo tự tạo văn bản.

Đầu ra là một bản ghi nội dung mới với trạng thái chờ duyệt (Status=pending) trong CSDL và thông báo gửi thành công cho người dùng.

Văn bản sẽ được tạo. Bài chia sẻ mới sẽ chờ Quản trị viên thực hiện quy trình kiểm duyệt trước khi được xuất bản công khai (Status=published) hoặc bị từ chối (Status=rejected).

2.2.2.11. Chức năng Tạo Lớp học

Mục tiêu là cho phép người dùng có vai trò Giáo viên (Teacher) tạo ra các lớp học ảo để quản lý học viên.

Đầu vào là Tên lớp (Class Name) và Mô tả từ biểu mẫu tạo lớp.

Đầu ra là một bản ghi Classroom mới trong CSDL và một Mã lớp duy nhất được sinh ra, hiển thị cho Giáo viên.

Chức năng này khởi tạo một không gian học tập riêng. Mã lớp (ClassCode) được thiết kế ngắn gọn, dễ chia sẻ, đóng vai trò là khóa truy cập chính cho học viên.

2.2.3.12. Chức năng Mời học viên qua Gmail

Mục tiêu là tự động hóa việc gửi lời mời tham gia lớp học đến danh sách Email của học viên.

Đầu vào là danh sách địa chỉ Email của học viên cần mời.

Đầu ra là các Email mời (chứa InviteToken duy nhất) được gửi đến học viên, đồng thời các bản ghi ClassInvites được tạo để theo dõi trạng thái.

Sau khi tạo lớp, Giáo viên có thể mời học viên bằng cách nhập Email. Hệ thống sẽ sinh các Token mời duy nhất, có thời hạn, và sử dụng Dịch vụ Gmail để gửi hàng loạt

Email mời chứa liên kết tham gia.

2.2.3.13. Chức năng Tham gia Lớp học

Mục tiêu là cho phép học viên gia nhập một lớp học bằng Mã lớp hoặc qua Liên kết mời.

Đầu vào là Mã lớp do học viên nhập, hoặc Token mời từ liên kết Email.

Đầu ra là một bản ghi ClassMembers mới (liên kết học viên và lớp học) và thông báo “Tham gia thành công”.

Hệ thống sẽ xác thực tính hợp lệ của Mã lớp hoặc Token. Nếu hợp lệ và người dùng chưa phải là thành viên, hệ thống sẽ chính thức ghi danh học viên vào lớp học.

2.2.3.14. Chức năng Quản lý Bài luyện tập

Mục tiêu là cung cấp cho Admin toàn quyền CRUD (Tạo, Đọc, Cập nhật, Xóa) trên toàn bộ thư viện bài luyện tập của hệ thống.

Đầu vào là các thao tác CRUD (dữ liệu từ form tạo/sửa bài, lệnh xóa).

Đầu ra là dữ liệu bài luyện tập (Exercises) trong CSDL được cập nhật, thêm mới hoặc xóa bỏ, và giao diện quản lý được làm mới.

Đây là chức năng quản trị nội dung cốt lõi. Admin có thể gắn nhãn, phân loại, và thiết lập trạng thái hiển thị (Công khai/Riêng tư) cho các bài luyện tập.

2.2.3.15. Chức năng Quản lý Người dùng

Mục tiêu là cung cấp cho Admin quyền quản trị vòng đời, trạng thái và vai trò của các tài khoản người dùng trong hệ thống.

Đầu vào là các thao tác quản trị (lệnh Khóa/Mở khóa, lệnh Xóa, form Sửa/Thêm, lệnh Reset Mật khẩu).

Đầu ra là dữ liệu người dùng trong CSDL được cập nhật (ví dụ: IsLocked=true, thay đổi vai trò).

Chức năng này cho phép Admin Khóa (ngăn chặn đăng nhập) hoặc Mở khóa tài khoản, Xóa (ưu tiên xóa mềm), và Đặt lại mật khẩu cho người dùng khi cần thiết.

2.2.3.16. Chức năng Quản lý Lớp học

Mục tiêu là cung cấp giao diện quản trị chi tiết cho Giáo viên (hoặc Admin) đối với các lớp học họ sở hữu/quản lý.

Đầu vào là các yêu cầu sửa thông tin lớp, thêm/xóa thành viên, hoặc gán bài tập

(ClassAssignments).

Đầu ra là dữ liệu lớp học, thành viên và bài tập được cập nhật trong CSDL.

Giáo viên có thể quản lý danh sách thành viên (thêm/xóa thủ công) và quan trọng nhất là gán các bài luyện tập cụ thể cho cả lớp hoặc cho từng học viên.

2.2.3.17. Chức năng Theo dõi Tiến độ Học sinh

Mục tiêu của chức năng là cung cấp cho Giáo viên (Teacher) một công cụ giám sát hiệu suất học tập của lớp, cho phép xem tiến bộ WPM/Độ chính xác theo thời gian, theo bài, hoặc theo học sinh.

Đầu vào là các tham số lọc (lớp, học viên, khoảng thời gian).

Đầu ra là các biểu đồ (thể hiện tiến độ) hoặc bảng thống kê chi tiết đã được tổng hợp.

Giao diện cho phép lọc dữ liệu. Hệ thống sẽ tổng hợp (aggregate) dữ liệu từ bảng PracticeResults của các thành viên trong lớp, tính toán các chỉ số (WPM, Độ chính xác trung bình) và trực quan hóa dữ liệu.

2.2.3.18. Chức năng Thống kê và Báo cáo

Mục tiêu là cung cấp dashboard phân tích dữ liệu tổng quan cho Admin và Teacher, hỗ trợ khả năng xuất báo cáo.

Đầu vào là các bộ lọc và yêu cầu xuất file.

Đầu ra là giao diện Dashboard và các file báo cáo được tải về.

Đây là module phân tích dữ liệu tổng quan. Admin có thể xem dashboard toàn hệ thống. Teacher có thể xem báo cáo của các lớp mình phụ trách. Chức năng này thực hiện các truy vấn tổng hợp phức tạp, áp dụng cơ chế cache và hỗ trợ xuất dữ liệu ra các định dạng phổ biến.

2.2.3.19. Chức năng Hỗ trợ người dùng qua Chatbot AI

Mục tiêu là cung cấp một trợ lý ảo thông minh, luôn sẵn sàng 24/7 để giải đáp thắc mắc, hướng dẫn sử dụng và hỗ trợ người dùng trong quá trình luyện tập.

Đầu vào là câu hỏi hoặc yêu cầu (prompt) của người dùng nhập vào cửa sổ chat.

Đầu ra là câu trả lời do AI tạo ra dựa trên câu hỏi của người dùng dựa trên các lỗi sai của người dùng để đưa ra bài luyện tập thích hợp được hiển thị trong giao diện hội thoại.

Chức năng này sử dụng Gemini API làm bộ não xử lý ngôn ngữ tự nhiên. Chatbot

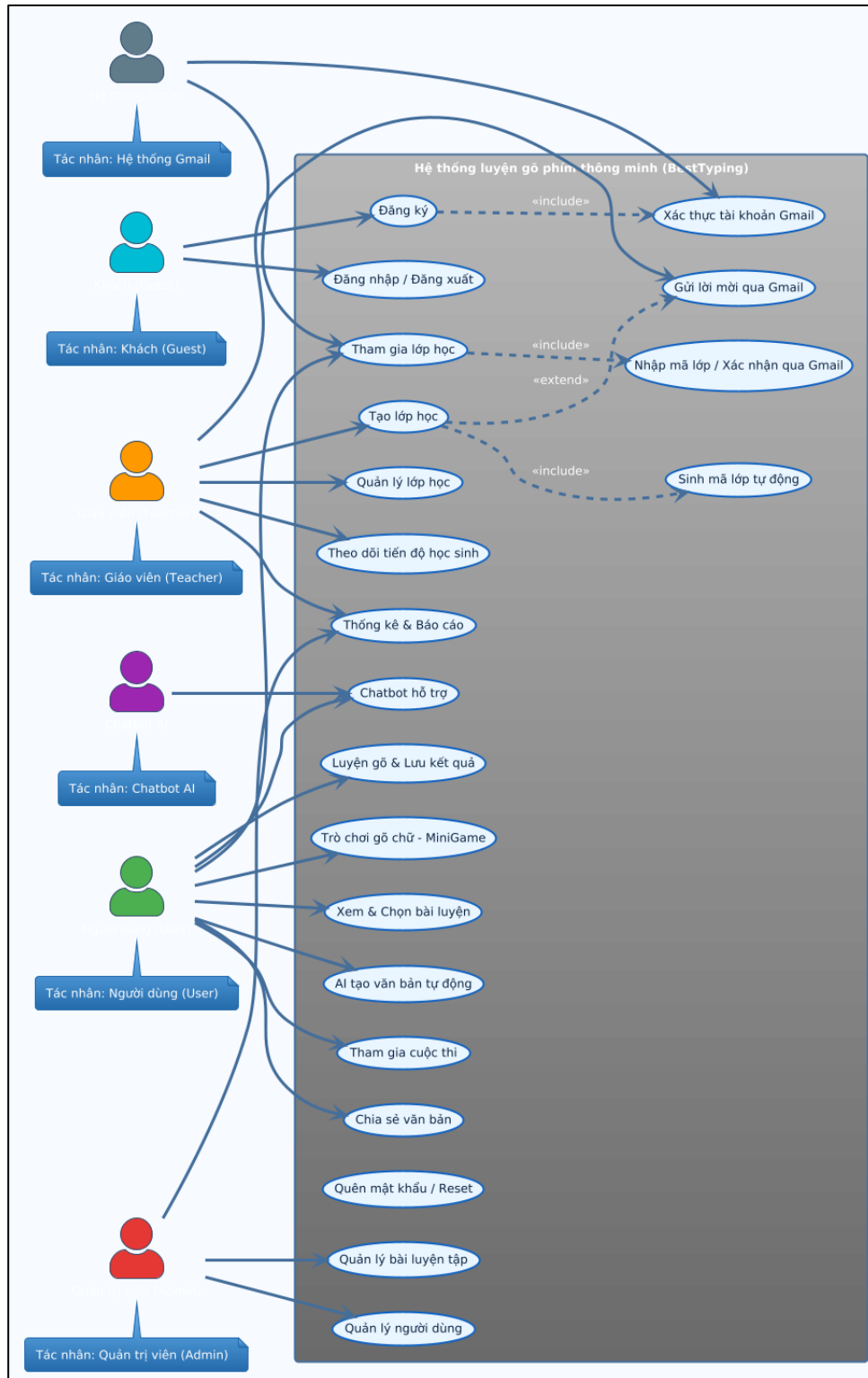
được cung cấp bối cảnh về các tính năng của website ví dụ: “WPM là gì?”, “Tôi hay sai những lỗi gì, cho tôi bài luyện tập để cải thiện?”,.... Hệ thống nhận prompt từ người dùng, gọi đến Gemini API để phân tích, hiểu ý định và tạo ra phản hồi. Kết quả trả về cho người dùng một cách tự nhiên giúp nâng cao trải nghiệm và giảm tải hỗ trợ thủ công. Phân tích những chữ hay gõ sai để đưa ra bài luyện tập cải thiện cho người sử dụng.

2.3. Sơ đồ Use Case và biểu đồ tuần tự minh họa các nghiệp vụ

2.3.1. Minh họa Sơ đồ Use Case và biểu đồ tuần tự tổng quát

2.3.1.1. Sơ đồ Use Case tổng quát

- Sơ đồ Use Case tổng quát



Hình 2.1. Sơ đồ Use Case tổng quát

- Mô tả sơ đồ Use Case tổng quát

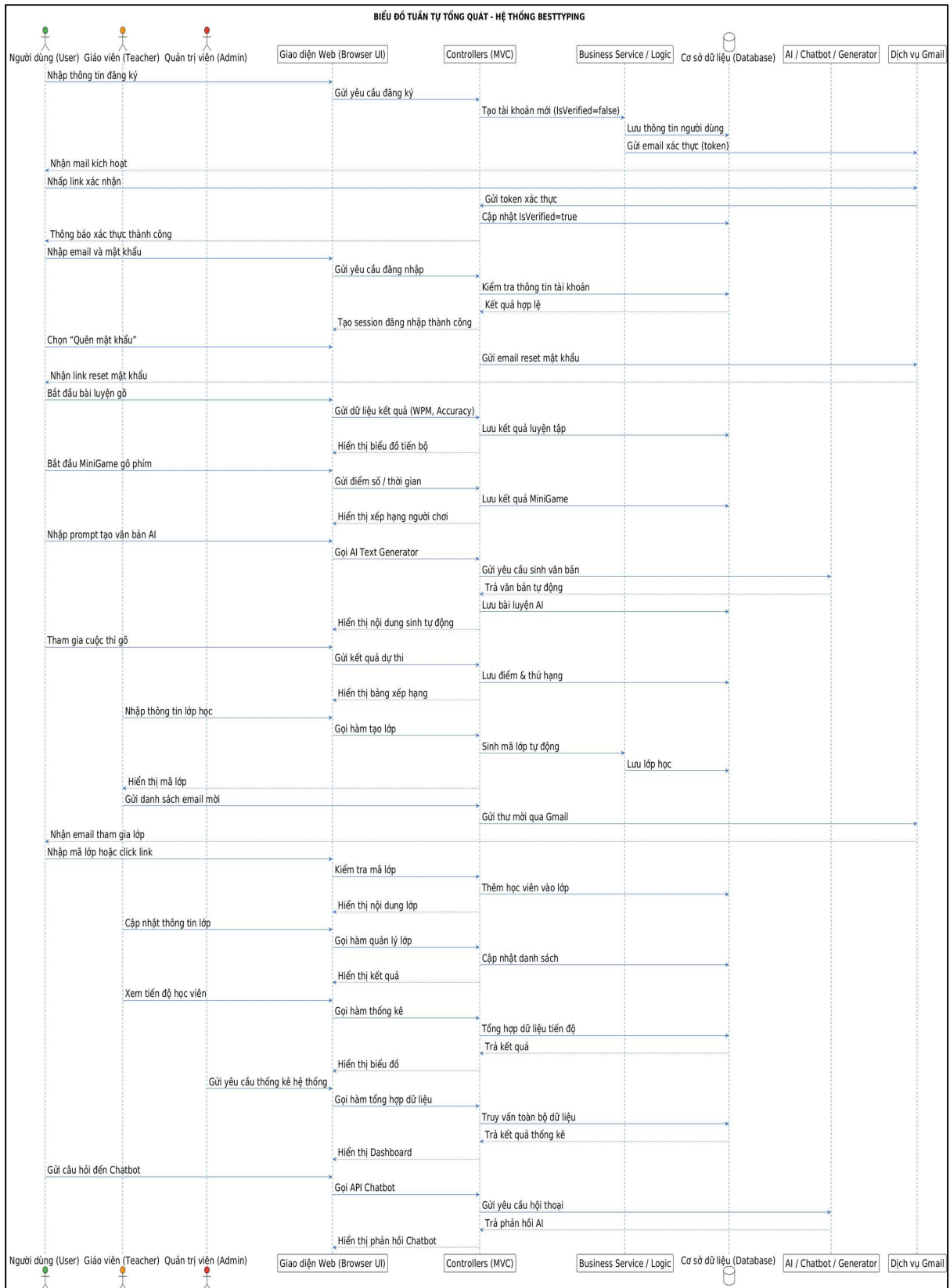
Bảng 2.1. Mô tả Use Case tổng quát

STT	Mã và Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	UC1:Đăng ký	Guest	Tạo tài khoản mới	Có Email hợp lệ	Tạo user & gửi Email xác thực
2	UC2:Xác thực tài khoản Gmail	User, Gmail	Kích hoạt tài khoản qua link Email	Có token hợp lệ	IsVerified = true
3	UC3:Đăng nhập / Đăng xuất	User	Truy cập/thoát hệ thống	Tài khoản tồn tại, không bị khóa	Tạo/Hủy session
4	UC4:Quên mật khẩu / Reset	User, Gmail	Đặt lại mật khẩu qua Email	Email tồn tại	Cập nhật mật khẩu
5	UC5:Xem & Chọn bài luyện	User	Duyệt/chọn bài luyện	Có dữ liệu bài luyện	Mở bài luyện
6	UC6:Luyện gõ & Lưu kết quả	User	Thực hành & lưu WPM/Độ chính xác	Đã chọn bài	Lưu kết quả vào DB
7	UC7:Trò chơi gõ chữ - MiniGame	User	Gõ ký tự hiển thị trong thời gian	Cấu hình game sẵn có	Lưu điểm & thời gian
8	UC8:AI tạo văn bản tự động	User, AI	Sinh văn bản luyện tập bằng AI	API AI hoạt động	Nhận văn bản & (tùy chọn) lưu
9	UC9:Tham gia cuộc thi	User	Tham gia & nộp kết quả	Cuộc thi đang mở	Cập nhật xếp hạng

STT	Mã và Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
10	UC10: Chia sẻ văn bản	User, Admin	Gửi & duyệt nội dung luyện tập	User đăng nhập	Nội dung được public
11	UC11: Tạo lớp học	Teacher	Tạo lớp và sinh mã	Teacher đăng nhập	Lưu lớp & classCode
12	UC12: Gửi lời mời qua Gmail	Teacher, Gmail	Mời học viên	Lớp tồn tại	Gửi Email + token
13	UC13: Tham gia lớp học	User, Gmail	Vào lớp bằng mã/link invite	ClassCode/Token hợp lệ	Thêm ClassMember
14	UC14: Quản lý bài luyện tập	Admin	CRUD bài luyện tập	Admin login	DB cập nhật
15	UC15: Quản lý người dùng	Admin	Thêm/sửa/khóa/xóa	Admin login	User cập nhật
16	UC16: Quản lý lớp học	Teacher / Admin	Quản lý thành viên/luyện tập	Lớp tồn tại	Cập nhật lớp
17	UC17: Theo dõi tiến độ học sinh	Teacher	Xem WPM/Độ chính xác/tiến bộ	Có kết quả	Hiển thị báo cáo
18	UC18: Thống kê & Báo cáo	Admin / Teacher	Dashboard, export	Có dữ liệu	Hiển thị báo cáo
19	UC19: Chatbot hỗ trợ	User/Chatbot	Hỏi đáp/tư vấn	Chatbot online	Phản hồi & lưu log

2.3.1.2. Biểu đồ tuần tự tổng quát

- Biểu đồ tuần tự tổng quát



Hình 2.2. Biểu đồ tuần tự tổng quát

- Luồng chính của biểu đồ tuần tự tổng quát

Bảng 2.2 Luồng chính của biểu đồ tuần tự tổng quát

Bước	Tác nhân	Mô tả hành động / Xử lý chính
1	Người dùng (User)	Nhập thông tin đăng ký tài khoản.
2	Giao diện Web (UI)	Gửi dữ liệu đăng ký đến Controller.
3	Controller (MVC)	Tiếp nhận yêu cầu, gọi Service để tạo tài khoản mới.
4	Service	Lưu tài khoản người dùng vào cơ sở dữ liệu, sinh token xác thực.
5	Gmail Service	Gửi Email xác thực đến người dùng.
6	User	Nhấn vào liên kết xác nhận trong Email.
7	Gmail → Controller	Gửi lại token xác thực đến hệ thống.
8	Controller → DB	Cập nhật trạng thái “IsVerified = true”.
9	UI → User	Hiển thị thông báo xác thực thành công.
10	User	Thực hiện đăng nhập với tài khoản đã kích hoạt.
11	Controller	Kiểm tra thông tin đăng nhập, tạo session.
12	User	Bắt đầu luyện gõ hoặc tham gia MiniGame.
13	UI → Controller	Gửi kết quả tốc độ gõ, độ chính xác (WPM, Độ chính xác).
14	Controller → DB	Lưu kết quả luyện tập, cập nhật tiến độ học tập.
15	User	Gửi yêu cầu tạo văn bản tự động bằng AI.
16	Controller → AI	Gửi prompt và yêu cầu sinh văn bản.
17	AI → Controller	Trả về nội dung văn bản sinh tự động.

Bước	Tác nhân	Mô tả hành động / Xử lý chính
18	Controller → DB	Lưu bài luyện AI vào hệ thống.
19	UI	Hiển thị kết quả văn bản sinh tự động trên giao diện.
20	User	Tham gia cuộc thi gõ bàn phím.
21	Controller → DB	Lưu kết quả cuộc thi, cập nhật bảng xếp hạng.
22	Teacher	Tạo lớp học mới, nhập thông tin lớp.
23	Controller → Service	Sinh mã lớp tự động, lưu vào DB.
24	Teacher	Gửi danh sách học viên để mời qua Gmail.
25	Controller → Gmail	Gửi Email mời học viên tham gia lớp học.
26	User	Nhận Email và tham gia lớp bằng mã lớp hoặc link.
27	Controller → DB	Thêm học viên vào danh sách lớp.
28	Teacher	Cập nhật thông tin lớp học, bài luyện.
29	Controller → DB	Lưu thông tin cập nhật.
30	Teacher	Theo dõi tiến độ học viên, yêu cầu thống kê.
31	Controller → DB	Truy vấn dữ liệu thống kê (WPM, Độ chính xác).
32	UI	Hiển thị biểu đồ tiến độ học viên.
33	Admin	Gửi yêu cầu thống kê toàn hệ thống.
34	Controller → DB	Lấy dữ liệu tổng hợp hệ thống (người dùng, bài luyện, lớp học).
35	UI	Hiển thị Dashboard thống kê tổng hợp.
36	Admin	Quản lý người dùng (thêm, xóa, khóa, mở).

Bước	Tác nhân	Mô tả hành động / Xử lý chính
37	Controller → DB	Cập nhật thông tin người dùng.
38	Admin	Quản lý bài luyện tập (CRUD).
39	Controller → DB	Lưu thay đổi bài luyện.
40	User	Gửi câu hỏi hoặc yêu cầu hỗ trợ qua Chatbot.
41	Controller → AI	Gửi truy vấn hội thoại đến Chatbot.
42	AI → Controller	Trả phản hồi trợ giúp.
43	UI	Hiển thị phản hồi Chatbot AI.

- Luồng phụ của biểu đồ tuần tự tổng quát

Bảng 2.3 Luồng phụ của biểu đồ tuần tự tổng quát

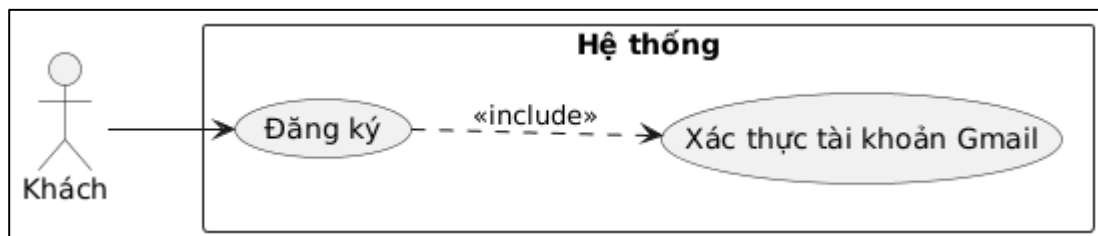
Mã	Tình huống phát sinh	Mô tả xử lý / Phản hồi của hệ thống
LP1	Người dùng nhập Email đã tồn tại khi đăng ký	Hiển thị thông báo lỗi “Email đã tồn tại” và yêu cầu nhập Email khác.
LP2	Email xác thực không hợp lệ hoặc hết hạn	Gửi lại Email xác thực mới, hiển thị thông báo “Vui lòng xác thực lại”.
LP3	Sai mật khẩu khi đăng nhập	Hiển thị cảnh báo “Sai mật khẩu” và gợi ý chức năng “Quên mật khẩu”.
LP4	Không kết nối được Gmail API	Hiển thị cảnh báo “Không thể gửi Email, vui lòng thử lại sau”.
LP5	Người dùng quên mật khẩu	Hệ thống gửi Email reset mật khẩu, hiển thị thông báo “Kiểm tra hộp thư Gmail của bạn”.
LP6	Kết nối đến AI bị lỗi hoặc quá tải	Thông báo “Hệ thống AI tạm ngừng, vui lòng thử lại sau”.
LP7	Người dùng nhập mã lớp sai	Hiển thị lỗi “Mã lớp không tồn tại hoặc đã hết hạn”.
LP8	Lớp học không có học viên	Hiển thị thông báo “Chưa có học viên tham gia lớp này”.
LP9	Dữ liệu thống kê rỗng	Hiển thị thông báo “Không có dữ liệu thống kê phù hợp”.
LP10	Quản trị viên xóa bài luyện không tồn tại	Hiển thị cảnh báo “Bài luyện không tồn tại trong hệ thống”.
LP11	API Chatbot không phản hồi	Thông báo “Chatbot hiện không khả dụng, vui lòng thử lại sau”.
LP12	Dữ liệu mạng không ổn định	Tự động thử gửi lại yêu cầu hoặc lưu cục bộ tạm thời.
LP13	Học viên bị xóa khỏi lớp	Hiển thị thông báo “Bạn không còn trong lớp này”.

Mã	Tình huống phát sinh	Mô tả xử lý / Phản hồi của hệ thống
LP14	Mất kết nối cơ sở dữ liệu	Hiển thị lỗi hệ thống và ghi log vào file nhật ký.
LP15	Người dùng chưa xác thực Email khi đăng nhập	Cảnh báo “Tài khoản chưa được xác thực, vui lòng kiểm tra Email”.
LP16	Trùng mã lớp khi tạo lớp học	Sinh lại mã lớp mới và lưu lại vào cơ sở dữ liệu.
LP17	Dữ liệu luyện gõ bị lỗi (thiếu thông tin)	Hiển thị cảnh báo và không lưu kết quả vào DB.
LP18	Thời gian phản hồi AI quá lâu	Hiển thị “Đang xử lý yêu cầu...” và cho phép hủy thao tác.
LP19	Lỗi truy vấn thống kê hệ thống	Ghi log lỗi và thông báo “Không thể truy xuất dữ liệu thống kê”.

2.3.2. Minh họa chức năng Đăng kí tài khoản

2.3.2.1. Sơ đồ Use Case chức năng Đăng kí tài khoản

- Sơ đồ Use Case chức năng Đăng kí tài khoản



Hình 2.3. Sơ đồ Use Case chức năng Đăng kí tài khoản

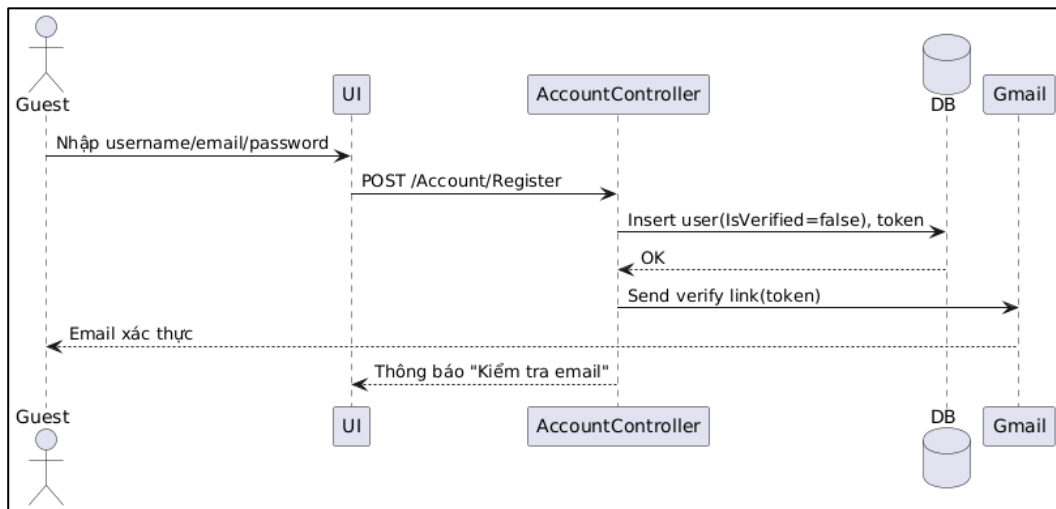
- Mô tả Sơ đồ Use Case chức năng Đăng kí tài khoản

Bảng 2.4. Mô tả Sơ đồ Use Case chức năng Đăng kí tài khoản

STT	Tên Use Case	Tác nhân (Actor)	Mô tả	Tiền điều kiện	Hậu điều kiện
1	UC1:Đăng ký	Khách	Tạo tài khoản mới, gửi Email xác thực	Email hợp lệ, chưa tồn tại	Tạo user & token xác thực

2.3.2.2. Biểu đồ tuần tự chức năng Đăng kí tài khoản

- Biểu đồ tuần tự chức năng Đăng kí tài khoản



Hình 2.4. Biểu đồ tuần tự chức năng Đăng kí tài khoản

- Luồng chính chức năng Đăng kí tài khoản

Bảng 2.5. Luồng chính chức năng Đăng kí tài khoản

Bước	Tác nhân	Mô tả hành động
1	Guest	Nhập thông tin đăng ký
2	UI	Gửi yêu cầu đăng ký
3	AC	Tạo user & token
4	Gmail	Gửi Email xác thực
5	UI	Thông báo kiểm tra Email

- Luồng phụ chức năng Đăng kí tài khoản

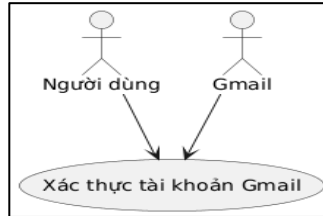
Bảng 2.6. Luồng phụ chức năng Đăng kí tài khoản

Mã	Tình huống	Mô tả
LP1	Email tồn tại	Báo lỗi trùng Email
LP2	Dữ liệu không hợp lệ	Báo lỗi validate form

2.3.3. Minh họa chức năng Xác thực tài khoản Email

2.3.2.3. Sơ đồ Usecasse chức năng Xác thực tài khoản Email

- Sơ đồ Use Case chức năng Xác thực tài khoản Email



Hình 2.5. Sơ đồ Use Case chức năng Xác thực tài khoản Email

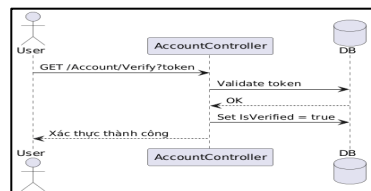
- Mô tả Sơ đồ Use Case chức năng Xác thực tài khoản Email

Bảng 2.7. Mô tả Sơ đồ Use Case chức năng Xác thực tài khoản Email

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	UC2: Xác thực tài khoản Gmail	User, Gmail	Kích hoạt tài khoản qua link	Token tồn tại & hợp lệ	IsVerified=true

2.3.3.2. Biểu đồ tuần tự chức năng Xác thực tài khoản Email

- Biểu đồ tuần tự chức năng Xác thực tài khoản Email



Hình 2.6. Biểu đồ tuần tự chức năng Xác thực tài khoản Email

- Luồng chính chức năng Xác thực tài khoản Email

Bảng 2.8. Luồng chính chức năng Xác thực tài khoản Email

Bước	Tác nhân	Mô tả
1	User	Click link
2	AC	Kiểm tra token
3	DB	Cập nhật xác thực
4	AC	Thông báo thành công

- Luồng phụ chức năng Xác thực tài khoản Email

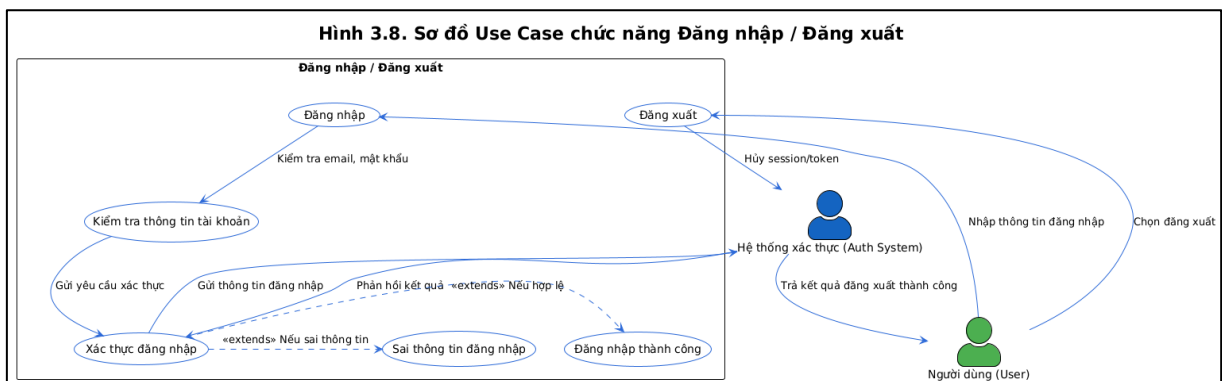
Bảng 2.9. Luồng phụ chức năng Xác thực tài khoản Email

Mã	Tình huống	Mô tả
LP1	Token hết hạn	Gợi ý gửi lại Email
LP2	Token sai	Báo lỗi không hợp lệ

2.3.4. Minh họa chức năng Đăng nhập/Đăng xuất

2.3.4.1. Sơ đồ Use Case chức năng Đăng nhập/Đăng xuất

- Sơ đồ Use Case chức năng Đăng nhập/Đăng xuất



Hình 2.7. Sơ đồ Use Case chức năng Đăng nhập/Đăng xuất

- Mô tả Use Case chức năng Đăng nhập/Đăng xuất

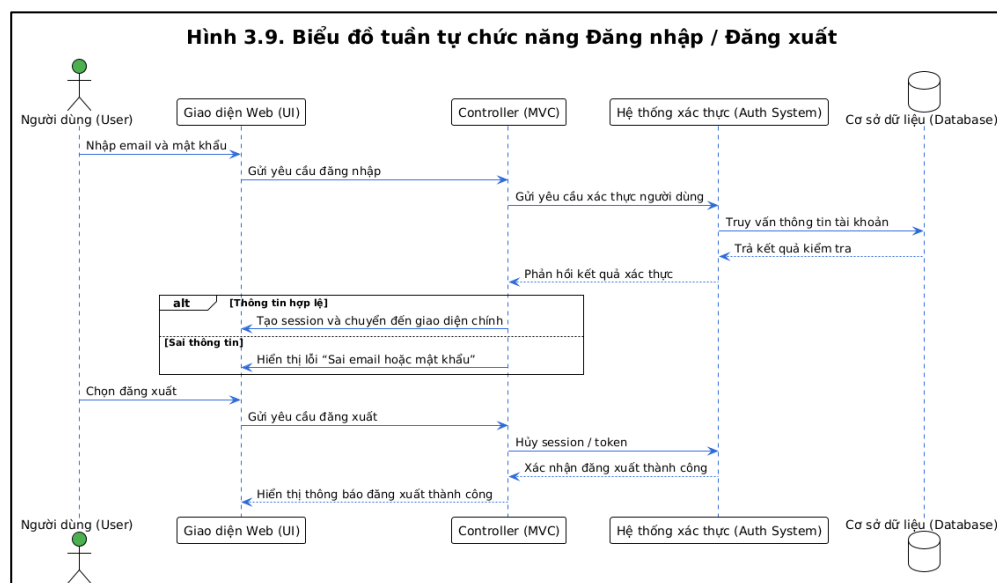
Bảng 2.10. Mô tả Use Case chức năng Đăng nhập/Đăng xuất

STT	Tên Use Case	Tác nhân (Actor)	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Đăng nhập	Người dùng	Người dùng nhập Email, mật khẩu để đăng nhập hệ thống.	Người dùng đã đăng ký tài khoản và xác thực Gmail thành công.	Người dùng đăng nhập vào hệ thống nếu thông tin hợp lệ.
2	Kiểm tra thông tin tài khoản	Hệ thống xác thực	Kiểm tra Email và mật khẩu có tồn tại trong hệ thống không.	Dữ liệu người dùng tồn tại trong cơ sở dữ liệu.	Xác nhận thông tin hợp lệ hoặc sai.

STT	Tên Use Case	Tác nhân (Actor)	Mô tả	Tiền điều kiện	Hậu điều kiện
3	Xác thực thông tin đăng nhập	Hệ thống xác thực	Gửi thông tin tới Auth System để xác thực người dùng.	Thông tin đăng nhập được nhập đầy đủ.	Nhận phản hồi thành công hoặc thất bại.
4	Đăng nhập thành công	Người dùng, Hệ thống xác thực	Khi thông tin hợp lệ, hệ thống tạo session cho người dùng.	Tài khoản hợp lệ.	Chuyển đến giao diện chính của ứng dụng.
5	Sai thông tin đăng nhập	Người dùng	Hệ thống hiển thị thông báo lỗi khi thông tin không hợp lệ.	Nhập sai Email hoặc mật khẩu.	Hiển thị thông báo lỗi, yêu cầu đăng nhập lại.
6	Đăng xuất	Người dùng	Người dùng chọn chức năng đăng xuất để kết thúc phiên làm việc.	Người dùng đang trong trạng thái đăng nhập.	Phiên đăng nhập kết thúc, token/session bị xóa.

2.3.4.2. Biểu đồ tuần tự chức năng Đăng nhập/Đăng xuất

- Biểu đồ tuần tự chức năng Đăng nhập/Đăng xuất



Hình 2.8. Biểu đồ tuần tự chức năng Đăng nhập/Đăng xuất

- Luồng chính chức năng Đăng nhập/Đăng xuất

Bảng 2.11. Luồng chính chức năng Đăng nhập/Đăng xuất

Bước	Tác nhân	Mô tả hành động
1	Người dùng	Nhập thông tin đăng nhập gồm Email và mật khẩu.
2	Giao diện Web	Gửi yêu cầu đăng nhập đến Controller.
3	Controller	Gọi hệ thống xác thực để kiểm tra tài khoản.
4	Auth System	Gửi truy vấn đến cơ sở dữ liệu kiểm tra thông tin.
5	CSDL	Trả kết quả xác thực (hợp lệ / không hợp lệ).
6	Controller	Nếu hợp lệ → tạo session và phản hồi giao diện chính.
7	Người dùng	Chọn chức năng đăng xuất.
8	Controller	Hủy session / token đăng nhập.
9	Giao diện Web	Hiển thị thông báo “Đăng xuất thành công”.

- Luồng phụ chức năng Đăng nhập/Đăng xuất

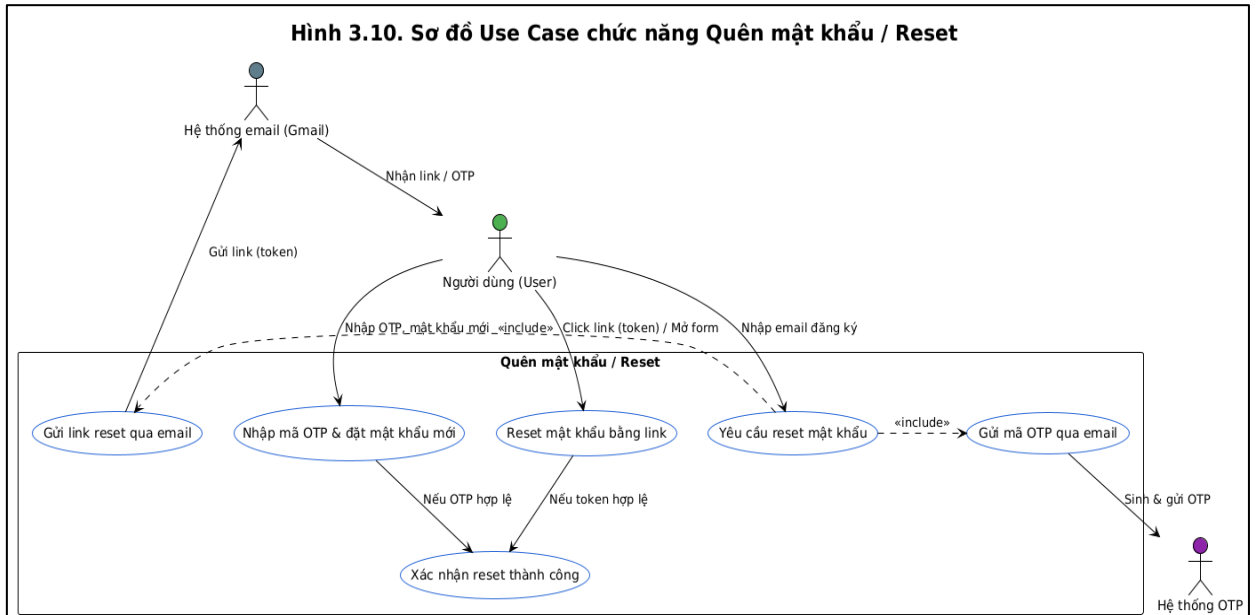
Bảng 2.12. Luồng phụ chức năng Đăng nhập/Đăng xuất

Mã	Tình huống phát sinh	Mô tả xử lý / Phản hồi hệ thống
LP1	Email hoặc mật khẩu không hợp lệ	Hiển thị thông báo “Sai thông tin đăng nhập”.
LP2	Tài khoản chưa xác thực Gmail	Thông báo “Vui lòng xác thực Email trước khi đăng nhập”.
LP3	Tài khoản bị khóa bởi quản trị viên	Hiển thị cảnh báo “Tài khoản bị khóa, liên hệ quản trị viên”.
LP4	Lỗi kết nối cơ sở dữ liệu	Hiển thị thông báo lỗi và ghi log hệ thống.
LP5	Người dùng quên mật khẩu	Hiển thị gợi ý “Chọn chức năng Quên mật khẩu để đặt lại”.
LP6	Người dùng đóng trình duyệt khi chưa đăng xuất	Hệ thống tự động hủy session sau thời gian timeout.

2.3.5. Minh họa chức năng Quên mật khẩu

2.3.5.1. Sơ đồ Use Case chức năng Quên mật khẩu

- Sơ đồ Use Case chức năng Quên mật khẩu



Hình 2.9. Sơ đồ Use Case chức năng Quên mật khẩu

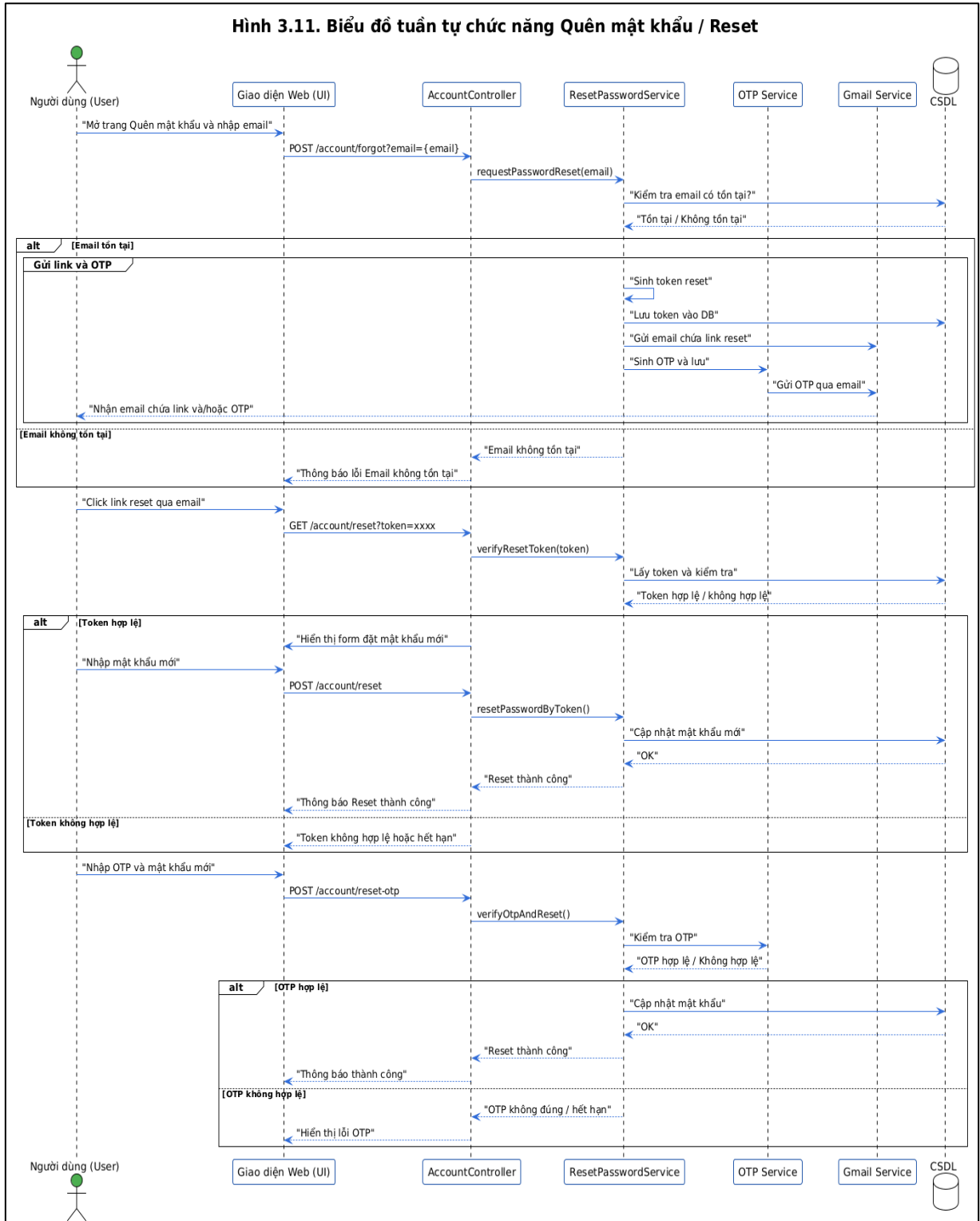
- Mô tả Use Case chức năng Quên mật khẩu

Bảng 2.13. Mô tả Use Case chức năng Quên mật khẩu

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Yêu cầu reset mật khẩu	Người dùng	Người dùng yêu cầu đặt lại mật khẩu bằng Email đã đăng ký.	Người dùng đã có Email đăng ký.	Hệ thống gửi link reset &/hoặc OTP tới Email.
2	Gửi link reset qua Email	Hệ thống	Tạo token 1 lần, gửi link reset có token kèm hạn dùng.	Token được sinh và lưu tạm.	Email chứa link gửi tới người dùng.
3	Gửi mã OTP qua Email	Hệ thống	Sinh mã OTP (6 chữ số), gửi mail, lưu OTP có expiry.	OTP được sinh & lưu.	Người dùng nhận OTP qua Email.
4	Reset mật khẩu bằng link	Người dùng	Người dùng click link chứa token → mở form đặt mật khẩu mới.	Token hợp lệ, chưa hết hạn.	Mật khẩu được cập nhật, token vô hiệu.
5	Nhập mã OTP & đặt mật khẩu mới	Người dùng	Người dùng nhập OTP nhận được và mật khẩu mới.	OTP hợp lệ và chưa hết hạn.	Mật khẩu được cập nhật, OTP vô hiệu.
6	Xác nhận reset thành công	Hệ thống	Hệ thống trả về thông báo thành công.	Mật khẩu đã được cập nhật.	Người dùng có thể đăng nhập bằng mật khẩu mới.

2.3.5.2. Biểu đồ tuần tự chức năng Quên mật khẩu

- Biểu đồ tuần tự chức năng Quên mật khẩu



Hình 2.10. Biểu đồ tuần tự chức năng Quên mật khẩu

- Luồng chính chức năng Quên mật khẩu

Bảng 2.14. Luồng chính chức năng Quên mật khẩu

Bước	Tác nhân	Mô tả hành động
1	User	Chọn “Quên mật khẩu” & nhập Email đã đăng ký.
2	UI	Gửi yêu cầu đến Controller (POST /account/forgot).
3	Controller	Gọi AuthService để xử lý reset.
4	Service	Kiểm tra Email tồn tại, sinh token reset & OTP, lưu DB.
5	Service -> Gmail/OTPSys	Gửi Email chứa link reset &/hoặc OTP.
6	User	Nhận Email, chọn 1 trong 2 cách: click link hoặc nhập OTP.
7	Nếu link: User click link → hệ thống verify token → hiện form đặt mật khẩu mới → User submit → cập nhật mật khẩu trong DB → thông báo thành công.	
8	Nếu OTP: User nhập OTP + mật khẩu mới → hệ thống verify OTP → cập nhật mật khẩu → thông báo thành công.	

- Luồng phụ chức năng Quên mật khẩu

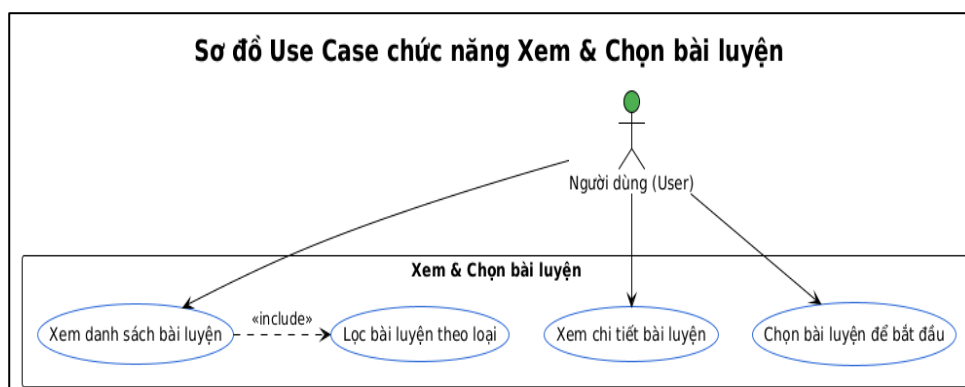
Bảng 2.15. Luồng phụ chức năng Quên mật khẩu

Mã	Tình huống	Xử lý / Phản hồi
LP1	Email không tồn tại	Trả lời “Email chưa đăng ký” → không gửi mail.
LP2	Gmail gửi thất bại (bounced)	Ghi log, retry theo chính sách, thông báo “Không thể gửi Email, thử lại sau”.
LP3	Token reset hết hạn / đã dùng	Thông báo “Link không hợp lệ hoặc đã hết hạn”, đề nghị gửi lại yêu cầu.
LP4	OTP sai hoặc hết hạn	Thông báo “OTP không hợp lệ hoặc đã hết hạn”, cho phép yêu cầu gửi lại OTP (limit).
LP5	Lỗi cơ sở dữ liệu khi cập nhật mật khẩu	Ghi log lỗi, hiển thị thông báo lỗi chung, yêu cầu thử lại sau.
LP6	Lạm dụng gửi nhiều yêu cầu reset	Áp dụng rate limit, tạm khoá tính năng cho IP/Email trong một khoảng thời gian.
LP7	Người dùng không nhận được mail	Hướng dẫn kiểm tra spam / whitelist, cho phép gửi lại sau xác thực.

2.3.6. Minh họa chức năng Xem và Chọn bài luyện

2.3.6.1. Sơ đồ Use Case chức năng Xem và Chọn bài luyện

- Sơ đồ Use Case chức năng Xem và Chọn bài luyện



Hình 2.11. Sơ đồ Use Case chức năng Xem và Chọn bài luyện

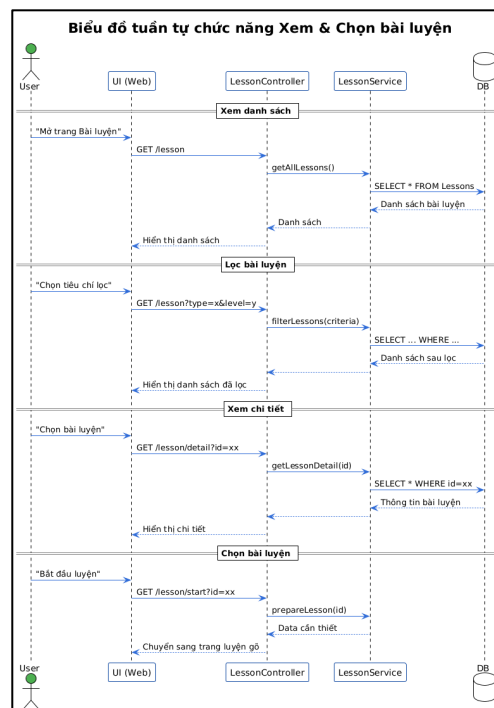
- Mô tả Use Case chức năng Xem và Chọn bài luyện

Bảng 2.16. Mô tả Use Case chức năng Xem và Chọn bài luyện

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Xem danh sách bài luyện	User	Hiển thị danh sách các bài luyện có sẵn.	Người dùng đã đăng nhập.	Danh sách bài luyện được hiển thị.
2	Lọc bài luyện theo loại	User	Người dùng lọc theo độ khó, loại văn bản, chủ đề.	Đang trong màn hình danh sách bài luyện.	Danh sách được lọc theo tiêu chí.
3	Xem chi tiết bài luyện	User	Hiển thị nội dung bài luyện trước khi chọn.	Bài luyện tồn tại.	Hiển thị nội dung bài luyện.
4	Chọn bài luyện	User	Người dùng quyết định chọn bài để luyện.	Nội dung bài luyện hợp lệ.	Hệ thống chuyển sang giao diện luyện gõ.

2.3.6.2. Biểu đồ tuần tự chức năng Xem và Chọn bài luyện

- Biểu đồ tuần tự chức năng Xem và Chọn bài luyện



Hình 2.12. Biểu đồ tuần tự chức năng Xem và Chọn bài luyện

- Luồng chính chức năng Xem và Chọn bài luyện

Bảng 2.17. Luồng chính chức năng Xem và Chọn bài luyện

Bước	Tác nhân	Mô tả
1	User	Mở trang danh sách bài luyện.
2	UI	Gửi yêu cầu lên Controller.
3	Controller	Gọi Service để lấy danh sách.
4	Service	Truy vấn DB lấy danh sách.
5	DB	Trả kết quả danh sách bài luyện.
6	UI	Hiển thị cho người dùng.
7	User	Lọc hoặc xem chi tiết bài luyện.
8	User	Chọn bài để bắt đầu luyện.
9	UI	Điều hướng sang trang luyện gõ.

- Luồng phụ chức năng Xem và Chọn bài luyện

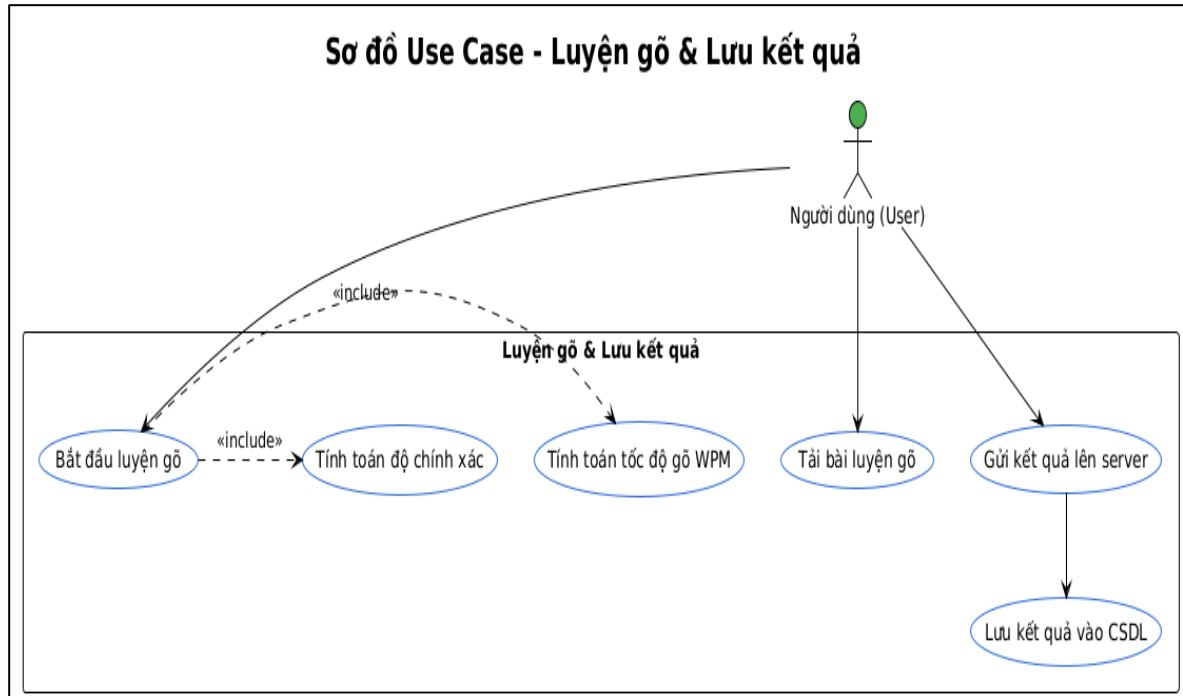
Bảng 2.18. Luồng phụ chức năng Xem và Chọn bài luyện

Mã	Tình huống	Xử lý
LP1	CSDL không có bài luyện	Hiển thị “Chưa có bài luyện nào”.
LP2	Lỗi kết nối DB	Hiển thị lỗi, ghi log hệ thống.
LP3	Bài luyện không tồn tại	Chuyển về danh sách và báo lỗi.
LP4	Người dùng chưa đăng nhập	Điều hướng đến giao diện đăng nhập.
LP5	Bộ lọc không có kết quả	Hiển thị “Không tìm thấy bài phù hợp”.

2.3.7. Minh họa chức năng Luyện gõ và Lưu kết quả

2.3.7.1. Sơ đồ Use Case chức năng Luyện gõ và Lưu kết quả

- Sơ đồ Use Case chức năng Luyện gõ và Lưu kết quả



Hình 2.13. Sơ đồ Use Case chức năng Luyện gõ và Lưu kết quả

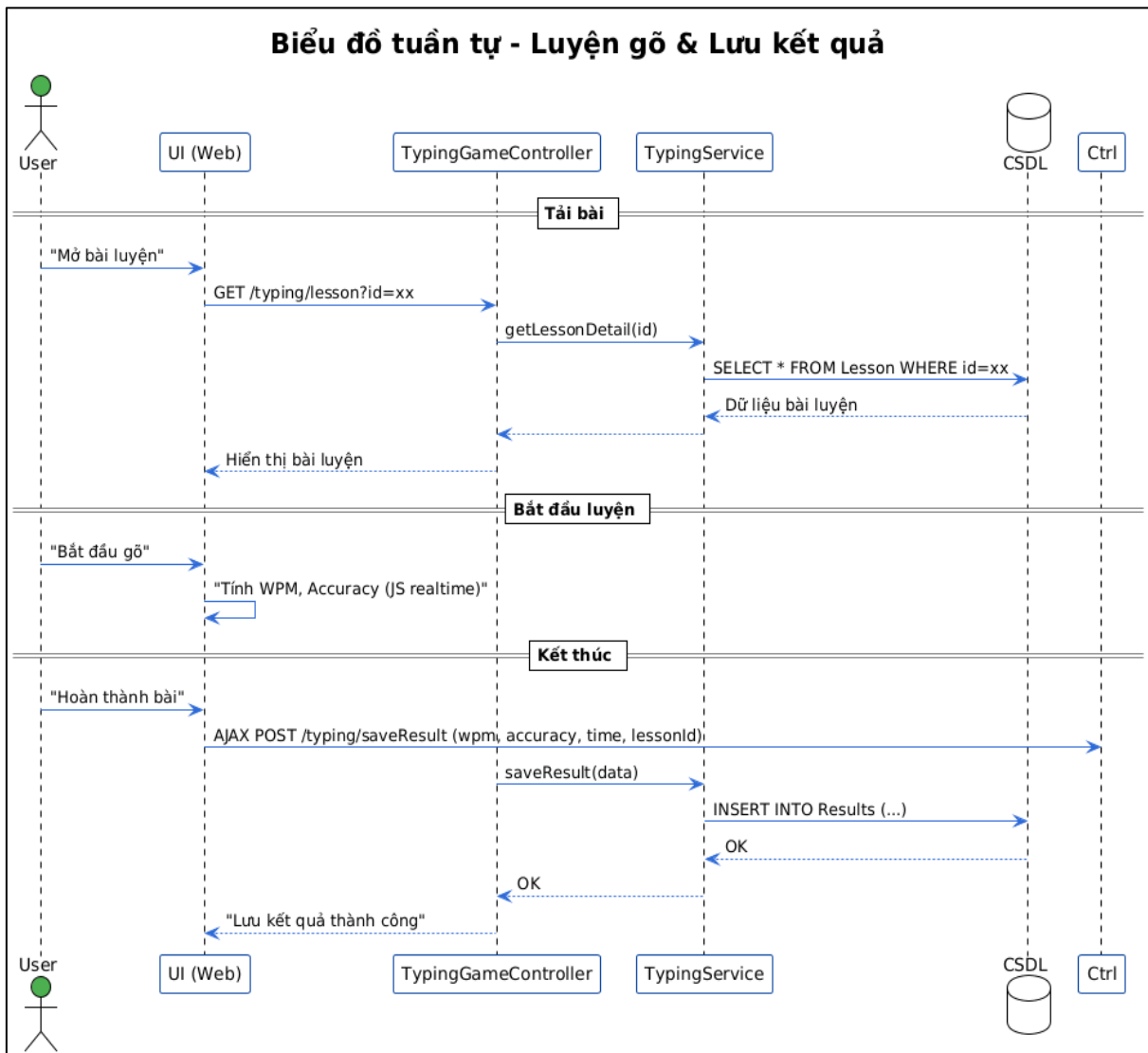
- Mô tả Use Case chức năng Luyện gõ và Lưu kết quả

Bảng 2.19. Mô tả Use Case chức năng Luyện gõ và Lưu kết quả

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Tải bài luyện gõ	User	Người dùng mở một bài luyện.	Bài luyện tồn tại.	Bài luyện hiển thị trên UI.
2	Bắt đầu luyện gõ	User	Người dùng bắt đầu gõ văn bản hiển thị.	Nội dung bài luyện đã tải.	Bắt đầu đo thời gian và tốc độ gõ.
3	Tính toán WPM	Hệ thống (UI/JS)	Tính tốc độ gõ từ thời gian và số từ đúng.	Người dùng đang gõ.	WPM được cập nhật realtime / cuối bài.
4	Tính toán Độ chính xác	Hệ thống	So sánh ký tự đúng/sai.	Người dùng đang gõ.	Độ chính xác được cập nhật.
5	Gửi kết quả	User/UI	Gửi kết quả về server qua AJAX.	Bài luyện hoàn thành.	Dữ liệu đợi lưu.
6	Lưu vào CSDL	Server	Ghi WPM, Độ chính xác, thời gian vào DB.	Dữ liệu hợp lệ.	Kết quả được lưu và hiển thị trong lịch sử.

2.3.7.2. Biểu đồ tuần tự chức năng Luyện gõ và Lưu kết quả

- Biểu đồ tuần tự chức năng Luyện gõ và Lưu kết quả



Hình 2.14. Biểu đồ tuần tự chức năng Luyện gõ và Lưu kết quả

- Luồng chính chức năng Luyện gõ và Lưu kết quả

Bảng 2.20. Luồng chính chức năng Luyện gõ và Lưu kết quả

Bước	Tác nhân	Mô tả
1	User	Chọn bài luyện để mở.
2	UI	Gửi yêu cầu lấy dữ liệu bài luyện.
3	Controller	Lấy dữ liệu từ Service.
4	Service	Lấy nội dung bài luyện từ CSDL.

Bước	Tác nhân	Mô tả
5	UI	Hiển thị bài luyện và cho người dùng bắt đầu gõ.
6	UI	Tính toán WPM và độ chính xác ngay trên trình duyệt.
7	User	Kết thúc bài luyện và gửi kết quả.
8	Server	Lưu kết quả vào DB.

- Luồng phụ chức năng Luyện gõ và Lưu kết quả

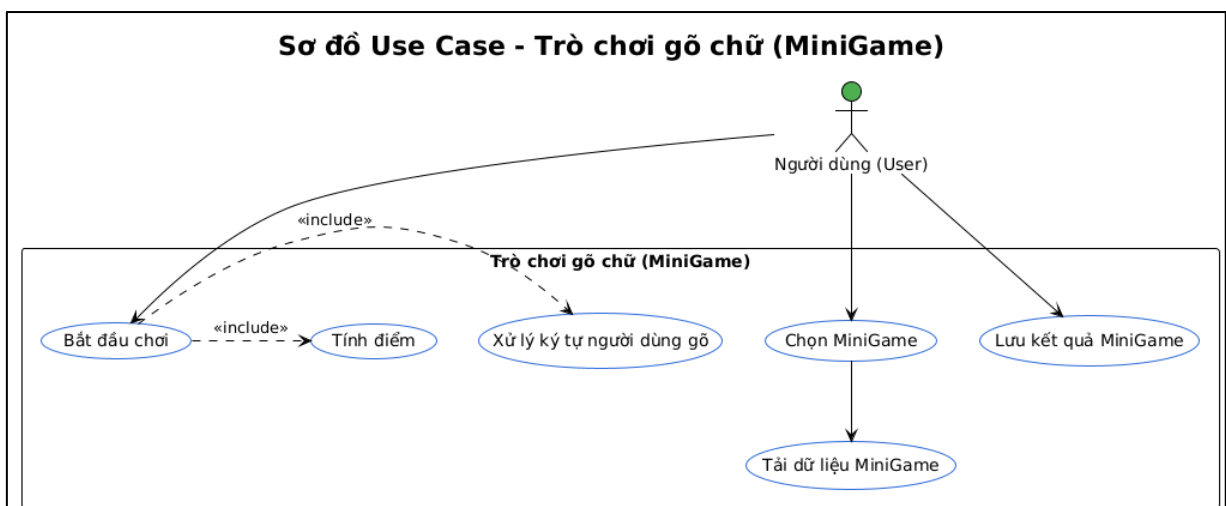
Bảng 2.21. Luồng phụ chức năng Luyện gõ và Lưu kết quả

Mã	Tình huống	Xử lý
LP1	Bài luyện không tồn tại	Hiển thị lỗi và trở về danh sách bài luyện.
LP2	Lỗi tải bài từ DB	Hiển thị thông báo lỗi chung.
LP3	Lỗi mạng khi gửi kết quả	Hiển thị “Không thể lưu, thử lại”.
LP4	Dữ liệu gửi lên không đầy đủ	Server trả lỗi và yêu cầu gửi lại.
LP5	User thoát giữa chừng	Không lưu kết quả, ghi log event.

2.3.8. Minh họa chức năng Trò chơi gõ chữ

2.3.8.1. Sơ đồ Use Case chức năng Trò chơi gõ chữ

- Sơ đồ Use Case chức năng Trò chơi gõ chữ



Hình 2.15. Sơ đồ Use Case chức năng Trò chơi gõ chữ

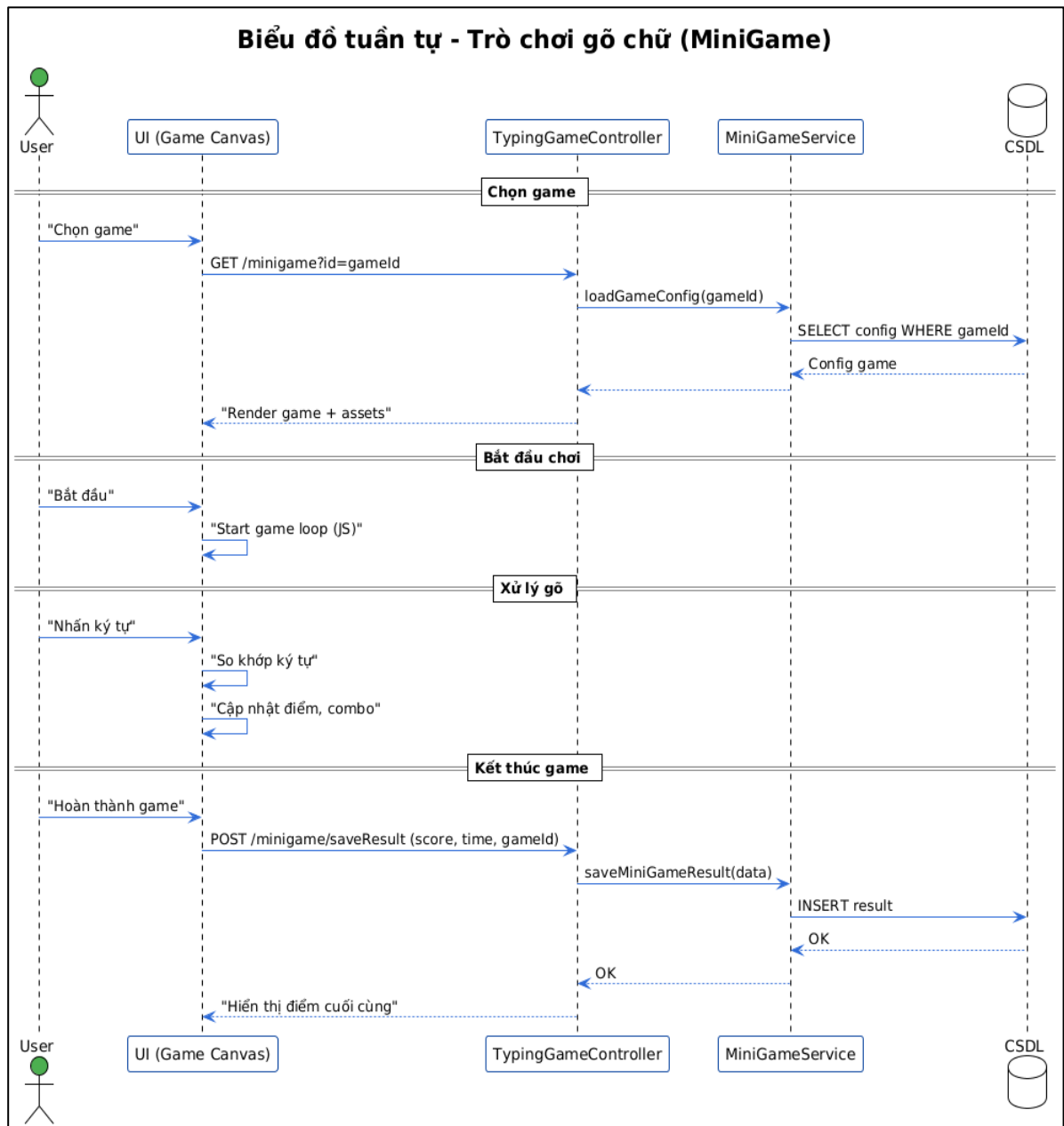
- Mô tả Use Case chức năng Trò chơi gõ chữ

Bảng 2.22. Mô tả Use Case chức năng Trò chơi gõ chữ

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Chọn MiniGame	User	Người dùng chọn loại trò chơi (gõ ký tự rơi xuống, phản xạ ký tự...).	Đăng nhập, MiniGame tồn tại.	Trò chơi được tải lên UI.
2	Tải dữ liệu MiniGame	Hệ thống	Tải danh sách ký tự, tốc độ rơi, thời gian, độ khó.	GameId hợp lệ.	Giao diện sẵn sàng bắt đầu.
3	Bắt đầu chơi	User	Bắt đầu phiên chơi, hiển thị ký tự.	Dữ liệu MiniGame đã tải.	Hệ thống kích hoạt vòng lặp trò chơi.
4	Xử lý ký tự gõ	UI/JS	Nhận phím người dùng và kiểm tra đúng hay sai.	Game đang chạy.	Tính điểm & cập nhật giao diện.
5	Tính điểm	Hệ thống	Tính điểm theo số ký tự đúng, combo, thời gian.	Người dùng gõ bàn phím.	Điểm hiển thị realtime.
6	Lưu kết quả MiniGame	Hệ thống	Gửi điểm cuối cùng lên server.	GameEnd	Kết quả lưu trong CSDL.

2.3.8.2. Biểu đồ tuần tự chức năng Trò chơi gõ chữ

- Biểu đồ tuần tự chức năng Trò chơi gõ chữ



Hình 2.16. Biểu đồ tuần tự chức năng Trò chơi gõ chữ

- Luồng chính chức năng Trò chơi gõ chữ

Bảng 2.23. Luồng chính chức năng Trò chơi gõ chữ

Bước	Tác nhân	Mô tả
1	User	Chọn MiniGame từ danh sách.
2	UI	Gửi yêu cầu tải cấu hình trò chơi.
3	Controller	Lấy dữ liệu game theo gameId.
4	Service	Lấy cấu hình từ DB.
5	UI	Hiển thị game và chờ người dùng bắt đầu.
6	User	Bắt đầu phiên chơi.
7	UI	Vòng lặp game chạy: hiển thị ký tự, nhận phím gõ.
8	UI	Tính điểm theo logic MiniGame.
9	User	Hoàn thành và gửi kết quả lên server.
10	Server	Lưu kết quả vào DB.

- Luồng phụ chức năng Trò chơi gõ chữ

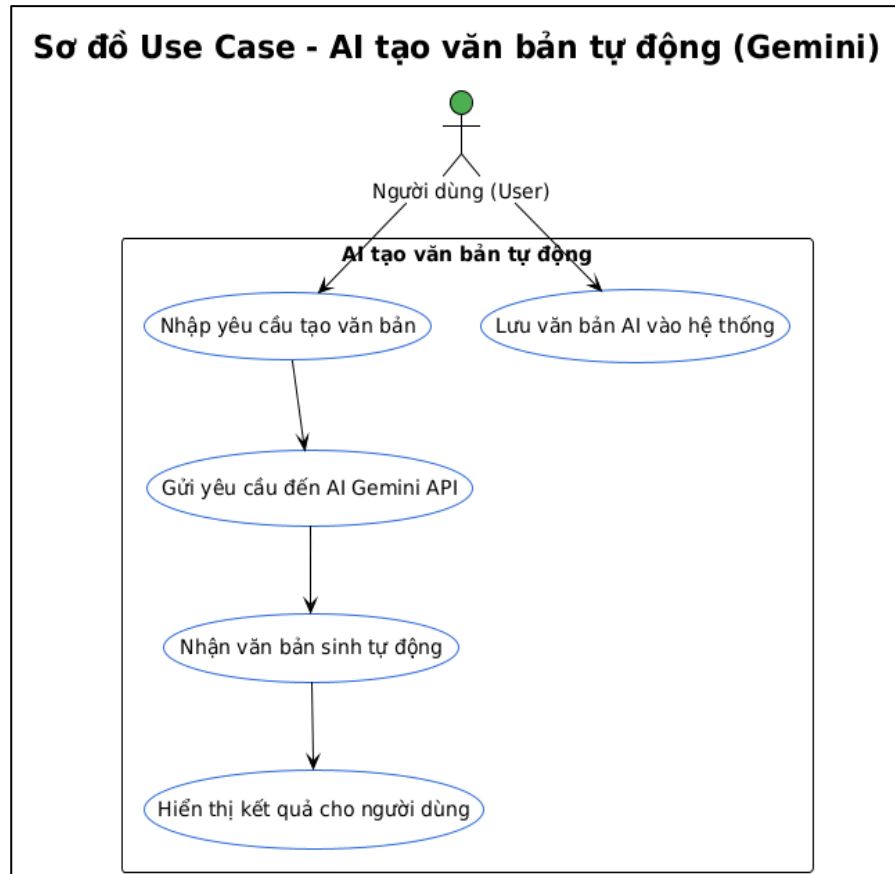
Bảng 2.24. Luồng phụ chức năng Trò chơi gõ chữ

Mã	Tình huống	Mô tả xử lý
LP1	gameId không tồn tại	Trả lỗi và quay lại danh sách Game.
LP2	Lỗi tải cấu hình MiniGame	Thông báo “Không thể khởi tạo game”.
LP3	Mất kết nối khi đang chơi	Game chạy offline nhưng không lưu kết quả.
LP4	Gửi kết quả thất bại	Hiển thị “Không thể lưu kết quả”, cho phép thử lại.
LP5	User thoát giữa chừng	Không lưu kết quả, hủy vòng lặp trò chơi.
LP6	Lỗi JS / canvas	Hiển thị popup lỗi và reset game.

2.3.9. Minh họa chức năng AI tạo văn bản tự động

2.3.9.1. Sơ đồ Use Case chức năng AI tạo văn bản tự động

- Sơ đồ Use Case chức năng AI tạo văn bản tự động



Hình 2.17. Sơ đồ Use Case chức năng AI tạo văn bản tự động

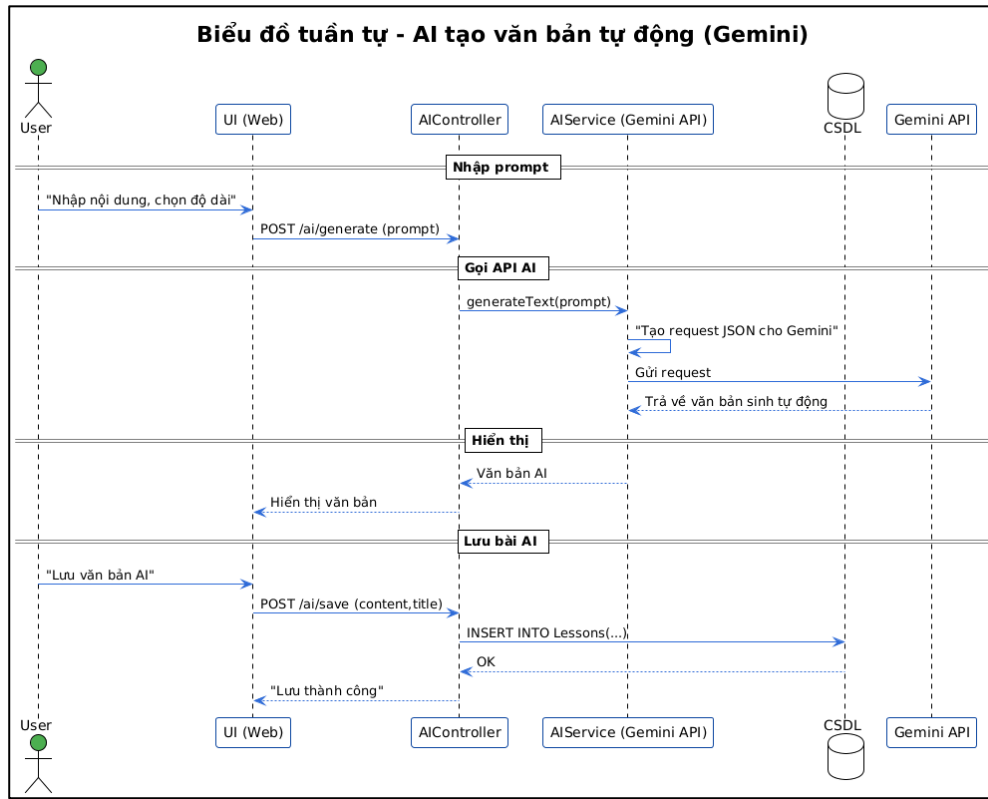
- Mô tả Use Case chức năng AI tạo văn bản tự động

Bảng 2.25. Mô tả Use Case chức năng AI tạo văn bản tự động

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Nhập yêu cầu tạo văn bản	User	Người dùng nhập nội dung mô tả hoặc chủ đề muốn AI tạo.	Đăng nhập hệ thống.	Prompt được gửi đến server.
2	Gọi Gemini API	Backend	Gửi prompt đến Gemini để sinh văn bản.	Có kết nối API.	Nhận lại văn bản sinh tự động.
3	Nhận văn bản AI	Backend	Nhận text trả về từ AI.	API phản hồi hợp lệ.	Dữ liệu sẵn sàng cho UI hiển thị.
4	Hiển thị kết quả	UI	Hiển thị văn bản tự động tạo lên màn hình.	Có dữ liệu AI.	Người dùng xem, có thể chỉnh sửa.
5	Lưu văn bản AI	User/Server	Người dùng quyết định lưu văn bản làm bài luyện mới.	Nội dung hợp lệ.	Bài luyện được lưu vào DB.

2.3.9.2. Biểu đồ tuần tự chức năng AI tạo văn bản tự động

- Biểu đồ tuần tự chức năng AI tạo văn bản tự động



Hình 2.18. Biểu đồ tuần tự chức năng AI tạo văn bản tự động

- Luồng chính chức năng AI tạo văn bản tự động

Bảng 2.26. Luồng chính chức năng AI tạo văn bản tự động

Bước	Tác nhân	Mô tả
1	User	Nhập prompt và gửi yêu cầu tạo văn bản.
2	UI	Gửi dữ liệu lên AIController.
3	Controller	Gọi AIService để xử lý.
4	AIService	Gửi request đến Gemini API.
5	Gemini	Sinh văn bản và trả về.
6	Controller	Trả văn bản lại UI.
7	UI	Hiển thị văn bản cho người dùng.
8	User	Quyết định lưu văn bản.
9	Controller	Lưu văn bản vào DB.

- Luồng phụ chức năng AI tạo văn bản tự động

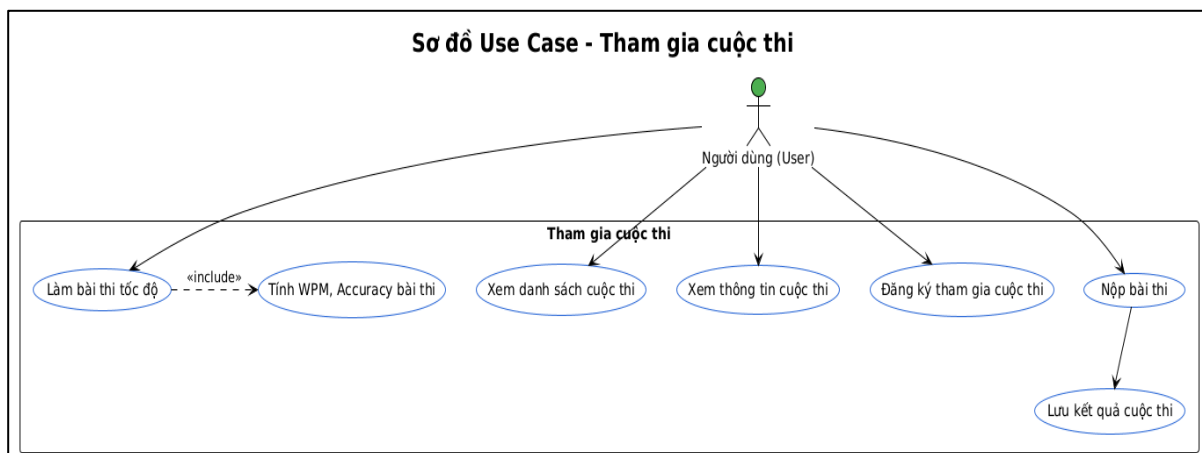
Bảng 2.27. Luồng phụ chức năng AI tạo văn bản tự động

Mã	Tình huống	Mô tả / Xử lý
LP1	API Gemini không phản hồi	Hiển thị “Không thể kết nối AI. Vui lòng thử lại.”
LP2	Prompt quá dài / thiếu	UI cảnh báo người dùng điều chỉnh.
LP3	AI trả về nội dung rỗng	Thông báo lỗi, yêu cầu nhập prompt khác.
LP4	Lỗi mạng	Retry tự động hoặc báo lỗi.
LP5	Lỗi lưu DB	Hiển thị lỗi “Không thể lưu văn bản”, ghi log.

2.3.10. Minh họa chức năng Tham gia cuộc thi chung

2.3.10.1. Sơ đồ Use Case chức năng Tham gia cuộc thi chung

- Sơ đồ Use Case chức năng Tham gia cuộc thi chung



Hình 2.19. Sơ đồ Use Case chức năng Tham gia cuộc thi chung

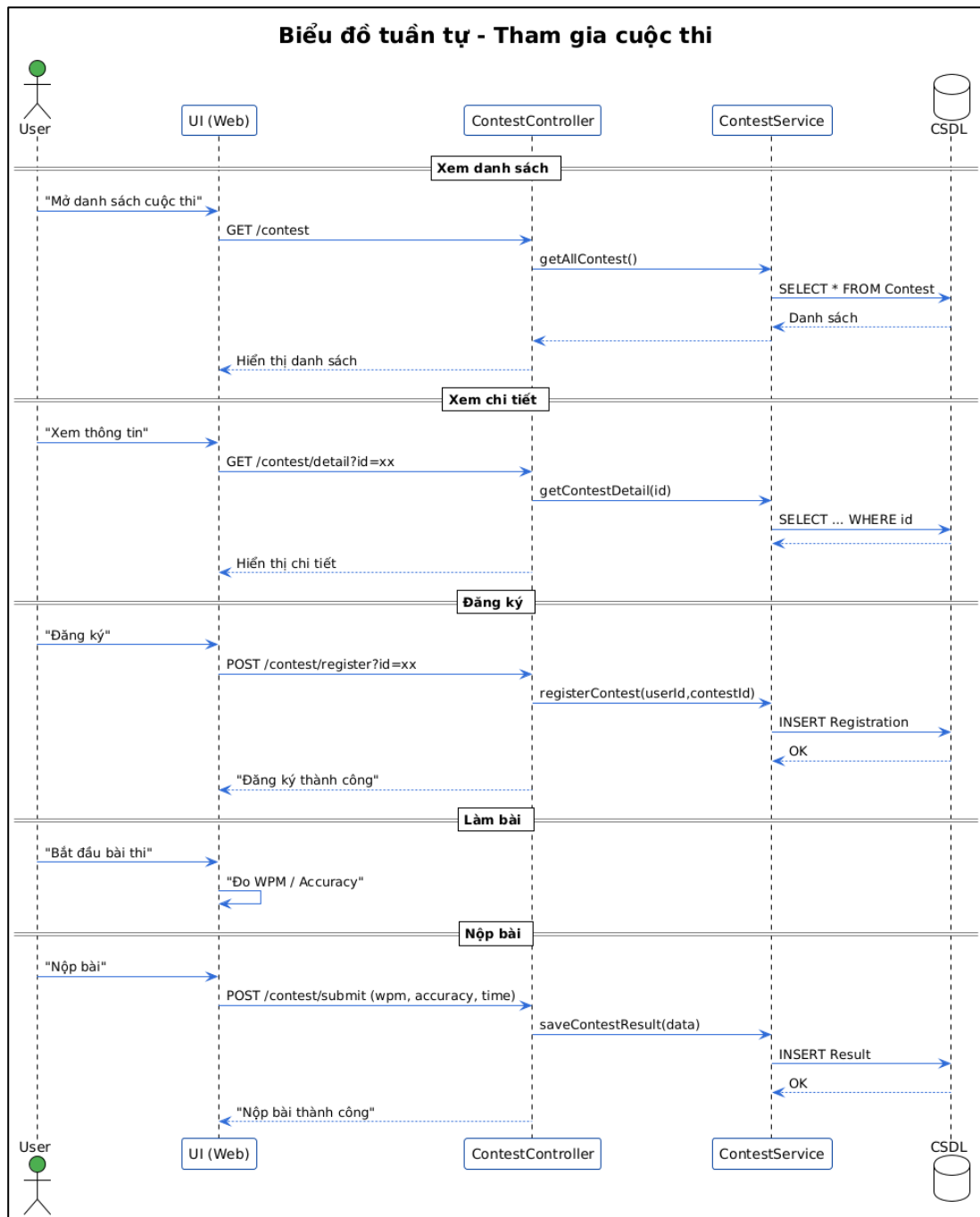
- Mô tả Use Case chức năng Tham gia cuộc thi chung

Bảng 2.28. Mô tả Use Case chức năng Tham gia cuộc thi chung

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Xem danh sách cuộc thi	User	Hiển thị các cuộc thi đang mở.	Người dùng đã đăng nhập.	Danh sách cuộc thi xuất hiện.
2	Xem thông tin cuộc thi	User	Xem chi tiết: thể lệ, thời gian, nội dung.	Cuộc thi tồn tại.	Chi tiết hiển thị.
3	Đăng ký tham gia	User	Đăng ký cuộc thi.	Đã đăng nhập.	User được thêm vào danh sách thi.
4	Làm bài thi tốc độ	User	Thực hiện bài thi gõ trong thời gian quy định.	Cuộc thi đang mở.	Hệ thống đo WPM, Độ chính xác.
5	Tính điểm	Hệ thống	Tính WPM, Độ chính xác từ dữ liệu gõ.	User hoàn thành bài thi.	Điểm sẵn sàng để nộp.
6	Nộp bài thi	User	Gửi kết quả lên server.	Bài thi hoàn thành.	Tiến hành lưu.
7	Lưu kết quả	Server	Ghi điểm WPM, Độ chính xác, thời gian.	Dữ liệu hợp lệ.	Kết quả được lưu vào CSDL.

2.3.10.2. Biểu đồ tuần tự chức năng Tham gia cuộc thi chung

- Biểu đồ tuần tự chức năng Tham gia cuộc thi chung



Hình 2.20. Biểu đồ tuần tự chức năng Tham gia cuộc thi chung

- Luồng chính chức năng Tham gia cuộc thi chung

Bảng 2.29. Luồng chính chức năng Tham gia cuộc thi chung

Bước	Tác nhân	Mô tả
1	User	Mở danh sách cuộc thi.
2	UI	Gửi yêu cầu lấy danh sách.
3	Controller	Lấy danh sách từ Service.
4	Service	Truy vấn DB danh sách cuộc thi.
5	User	Xem chi tiết cuộc thi.
6	User	Đăng ký tham gia.
7	System	Ghi đăng ký vào CSDL.
8	User	Làm bài thi tốc độ.
9	UI	Tính WPM, Độ chính xác bằng JavaScript.
10	User	Nộp kết quả.
11	System	Lưu kết quả vào DB.

- Luồng phụ chức năng Tham gia cuộc thi chung

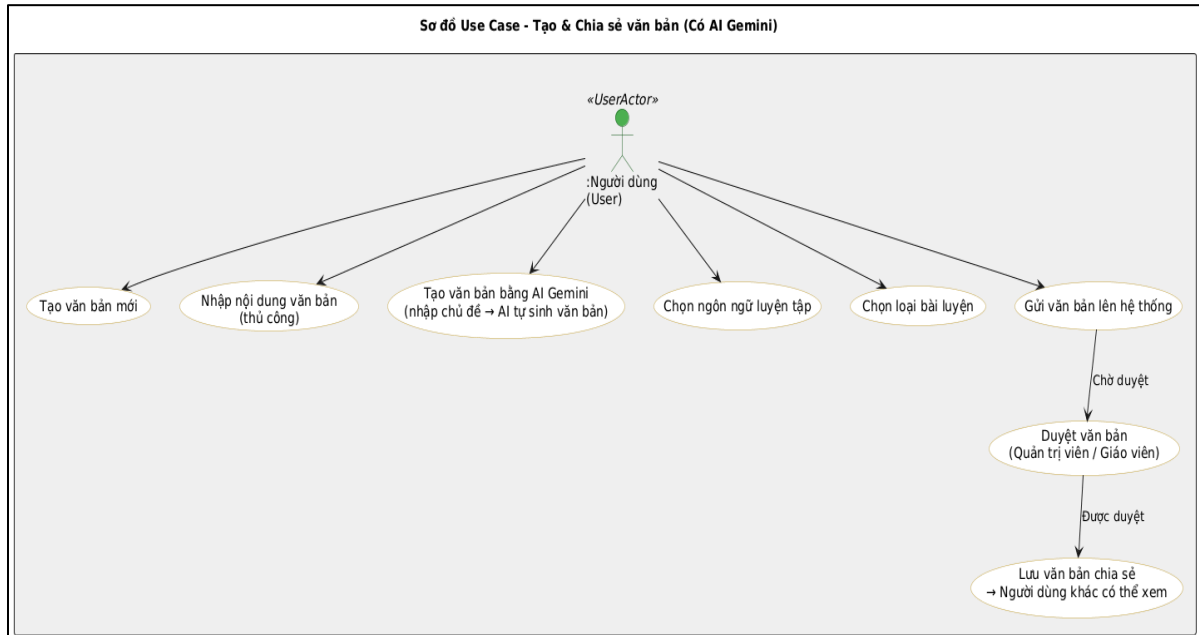
Bảng 2.30. Luồng phụ chức năng Tham gia cuộc thi chung

Mã	Tình huống	Xử lý
LP1	Cuộc thi không tồn tại	Hiển thị lỗi và chuyển về danh sách.
LP2	Cuộc thi đã đóng đăng ký	Thông báo “Cuộc thi đã kết thúc”.
LP3	User đã đăng ký trước đó	Trả thông báo “Bạn đã tham gia”.
LP4	Lỗi tải chi tiết cuộc thi	Hiển thị lỗi chung.
LP5	Lỗi khi lưu kết quả	Hiển thị “Không thể lưu kết quả”.
LP6	User thoát khi đang thi	Không lưu kết quả, ghi log.
LP7	Dữ liệu WPM, Độ chính xác không hợp lệ	Server từ chối cho tham gia.

2.3.11. Minh họa chức năng Tạo và Chia sẻ văn bản luyện tập

2.3.11.1. Sơ đồ Use Case Tạo và Chia sẻ văn bản luyện tập

- Sơ đồ Use Case chức năng Tạo và Chia sẻ văn bản luyện tập



Hình 2.21. Sơ đồ Use Case chức năng Tạo và Chia sẻ văn bản luyện tập

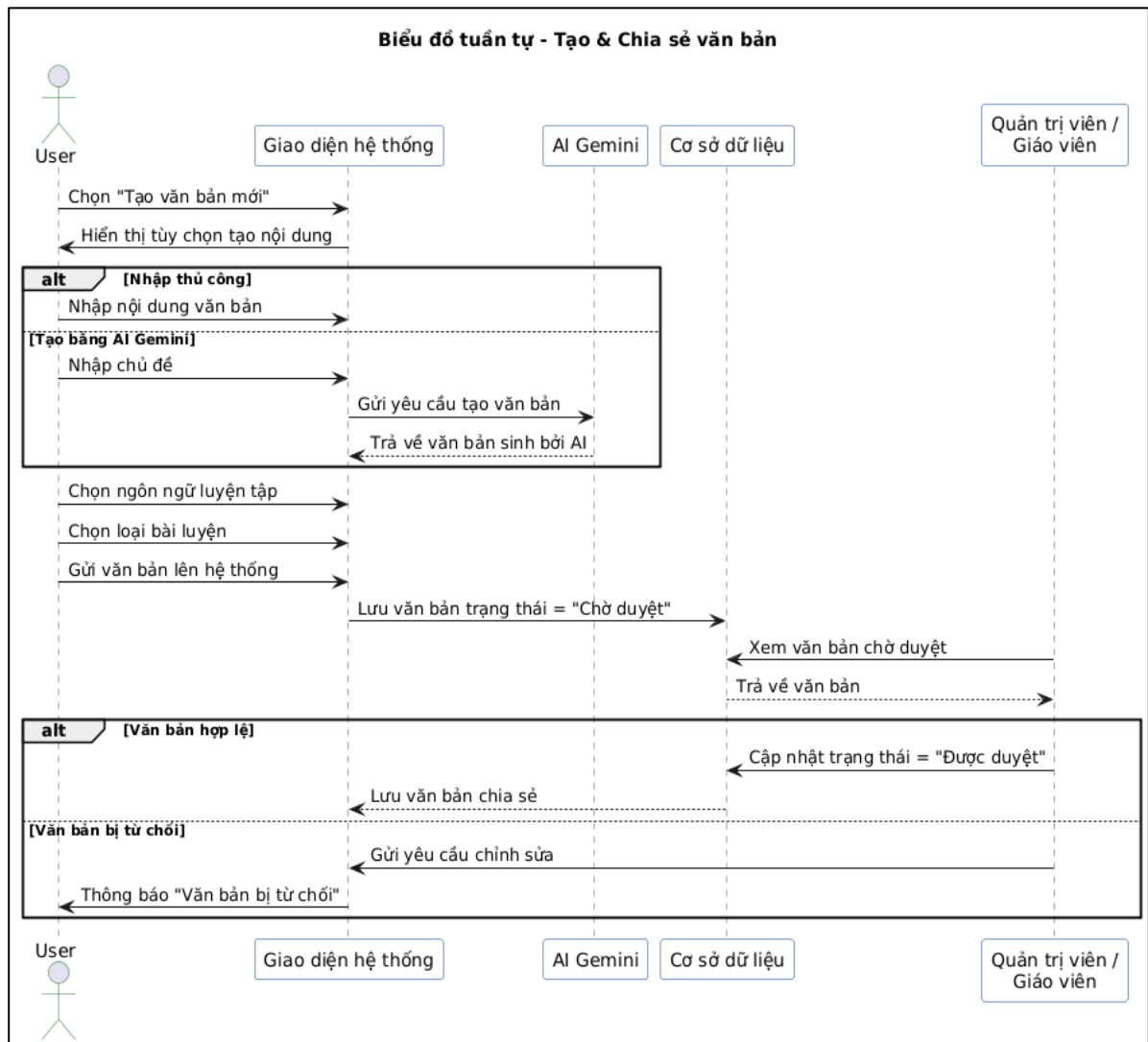
- Mô tả Use Case chức năng Tạo và Chia sẻ văn bản luyện tập

Bảng 2.31. Mô tả Use Case chức năng Tạo và Chia sẻ văn bản luyện tập

STT	Tên Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Nhập văn bản	Người dùng (User)	Người dùng nhập nội dung văn bản muốn chia sẻ. Có thể nhập thủ công hoặc dùng AI để tạo nội dung.	Người dùng đã đăng nhập vào hệ thống.	Nội dung văn bản đã được tạo và sẵn sàng để xử lý tiếp.
2	Chọn loại bài	Người dùng (User)	Người dùng chọn thể loại văn bản: văn bản luyện tập, test, truyện, đoạn code...	Người dùng đã tạo hoặc nhập nội dung văn bản.	Loại bài được chọn và gắn vào văn bản.
3	Gửi văn bản	Người dùng (User)	Người dùng submit văn bản lên hệ thống để chờ duyệt.	Nội dung không bị trống, không vi phạm quy định.	Hệ thống lưu văn bản và chuyển sang trạng thái “Chờ duyệt”.
4	Duyệt văn bản	Quản trị viên / Giáo viên (Admin / Teacher)	Admin/Giáo viên duyệt nội dung văn bản trước khi đưa lên hệ thống.	Người dùng đã gửi văn bản lên hệ thống.	Trạng thái văn bản chuyển thành “Được duyệt” hoặc “Từ chối”.
5	Lưu văn bản	Hệ thống	Hệ thống lưu văn bản đã được duyệt vào danh sách chia sẻ công khai.	Văn bản đã được duyệt.	Văn bản trở thành bài luyện chính thức và hiển thị cho mọi người dùng khác.

2.3.11.2. Biểu đồ tuần tự chức năng Tạo và Chia sẻ văn bản

- Biểu đồ tuần tự chức năng Tạo và Chia sẻ văn bản



Hình 2.22. Biểu đồ tuần tự chức năng Tạo và Chia sẻ văn bản

- Luồng chính chức năng Tạo và Chia sẻ văn bản

Bảng 2.32. Luồng chính chức năng Tạo và Chia sẻ văn bản

Bước	Tác nhân	Mô tả
1	User	Nhập nội dung văn bản cần chia sẻ (thủ công hoặc sử dụng AI Gemini để sinh văn bản).
2	UI (Giao diện người dùng)	Gửi nội dung văn bản lên server thông qua API.
3	Server	Kiểm tra tính hợp lệ của văn bản: độ dài, ký tự đặc biệt, nội dung vi phạm, ngôn ngữ...
4	Server	Lưu văn bản vào bảng ShareTemp dưới trạng thái <i>Chờ duyệt</i> .
5	Admin / Giáo viên	Xem danh sách văn bản chờ duyệt và thực hiện thao tác duyệt hoặc từ chối.
6	Server	Nếu được duyệt: Lưu vào bảng VanBanChiaSe (bài luyện chính thức). Nếu bị từ chối: ghi log và thông báo cho người dùng.

- Luồng phụ chức năng Tạo và Chia sẻ văn bản luyện tập

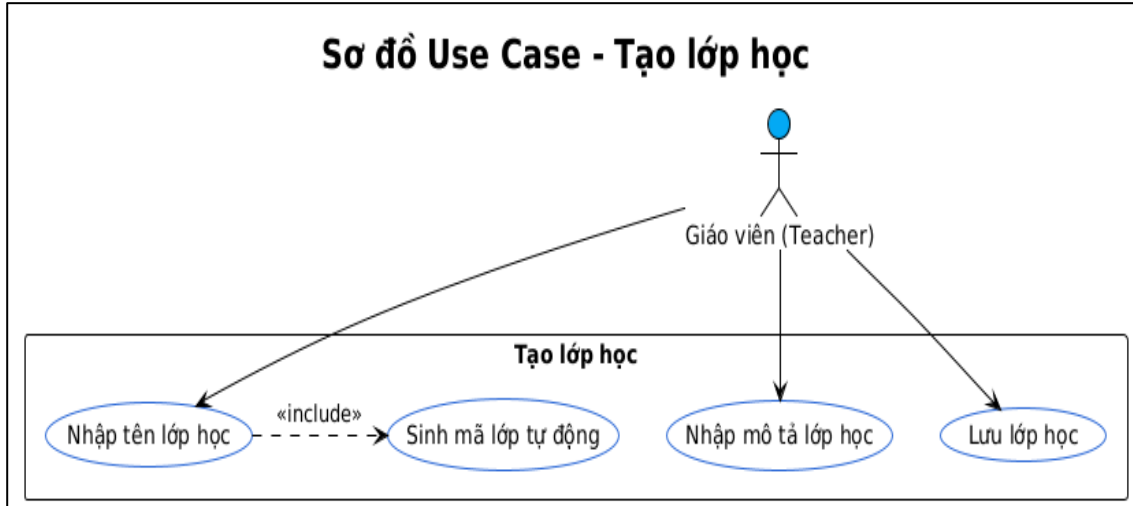
Bảng 2.33. Luồng phụ chức năng Tạo và Chia sẻ văn bản luyện tập

Mã	Tình huống	Xử lý
LP1	Văn bản quá ngắn hoặc rỗng	Hệ thống trả về lỗi: “Nội dung không hợp lệ”. Người dùng phải nhập lại.
LP2	Văn bản chứa từ khóa bị cấm hoặc nội dung vi phạm	Hệ thống từ chối, yêu cầu người dùng chỉnh sửa lại nội dung.
LP3	Admin / Giáo viên từ chối duyệt văn bản	Hệ thống gửi thông báo đến người dùng kèm lý do bị từ chối.
LP4	Lỗi trong quá trình lưu cơ sở dữ liệu (DB)	Hệ thống báo lỗi chung, rollback dữ liệu và yêu cầu thao tác lại.

2.3.12. Minh họa chức năng Tạo lớp học

2.3.12.1. Sơ đồ Use Case chức năng Tạo lớp học

- Sơ đồ Use Case chức năng Tạo lớp học



Hình 2.23. Sơ đồ Use Case chức năng Tạo lớp học

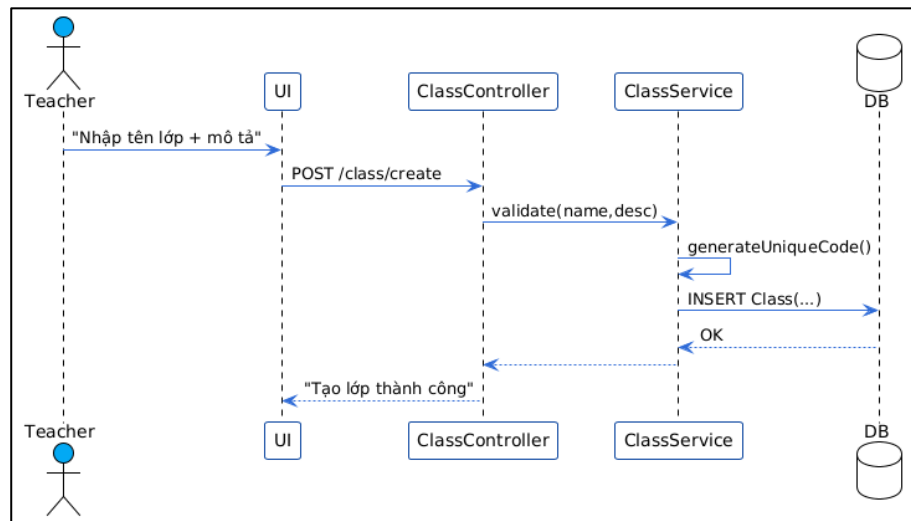
- Mô tả Use Case chức năng Tạo lớp học

Bảng 2.34. Mô tả Use Case chức năng Tạo lớp học

STT	Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Nhập tên lớp	Teacher	Nhập tên lớp học.	Đã đăng nhập.	Tên hợp lệ.
2	Nhập mô tả	Teacher	Thêm mô tả cho lớp.	Đã đăng nhập.	Mô tả hợp lệ.
3	Sinh mã lớp	System	Sinh mã duy nhất (ClassCode).	Đã đăng nhập.	Có mã lớp duy nhất.
4	Lưu lớp	System	Lưu lớp học vào DB.	Tên, mô tả hợp lệ.	Lớp học được tạo.

2.3.12.2. Biểu đồ tuần tự chức năng Tạo lớp học

- Biểu đồ tuần tự chức năng Tạo lớp học



Hình 2.24. Biểu đồ tuần tự chức năng Tạo lớp học

- Luồng chính chức năng Tạo lớp học

Bảng 2.35. Luồng chính chức năng Tạo lớp học

Bước	Tác nhân	Mô tả
1	Teacher	Nhập tên lớp, mô tả.
2	UI	Gửi request lên server.
3	Service	Sinh mã lớp.
4	DB	Lưu lớp học.
5	UI	Hiển thị thông báo tạo thành công.

- Luồng phụ chức năng Tạo lớp học

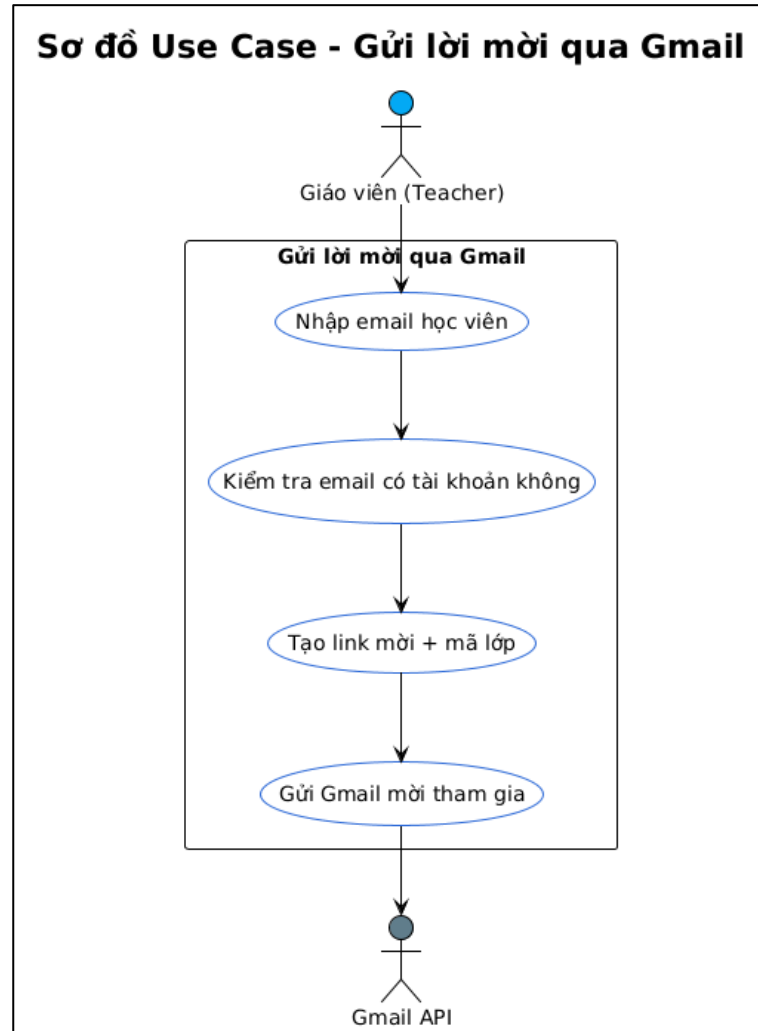
Bảng 2.36. Luồng phụ chức năng Tạo lớp học

Mã	Tình huống	Xử lý
LP1	Tên lớp trùng	Báo lỗi.
LP2	Sinh mã lớp bị trùng	Sinh lại mã mới.
LP3	Lỗi lưu DB	Báo lỗi chung.

2.3.13. Minh họa chức năng Gửi lời mời qua Email

2.3.12.3. Sơ đồ Use Case chức năng Gửi lời mời qua Email

- Sơ đồ Use Case chức năng Gửi lời mời qua Email



Hình 2.25. Sơ đồ Use Case chức năng Gửi lời mời qua Email

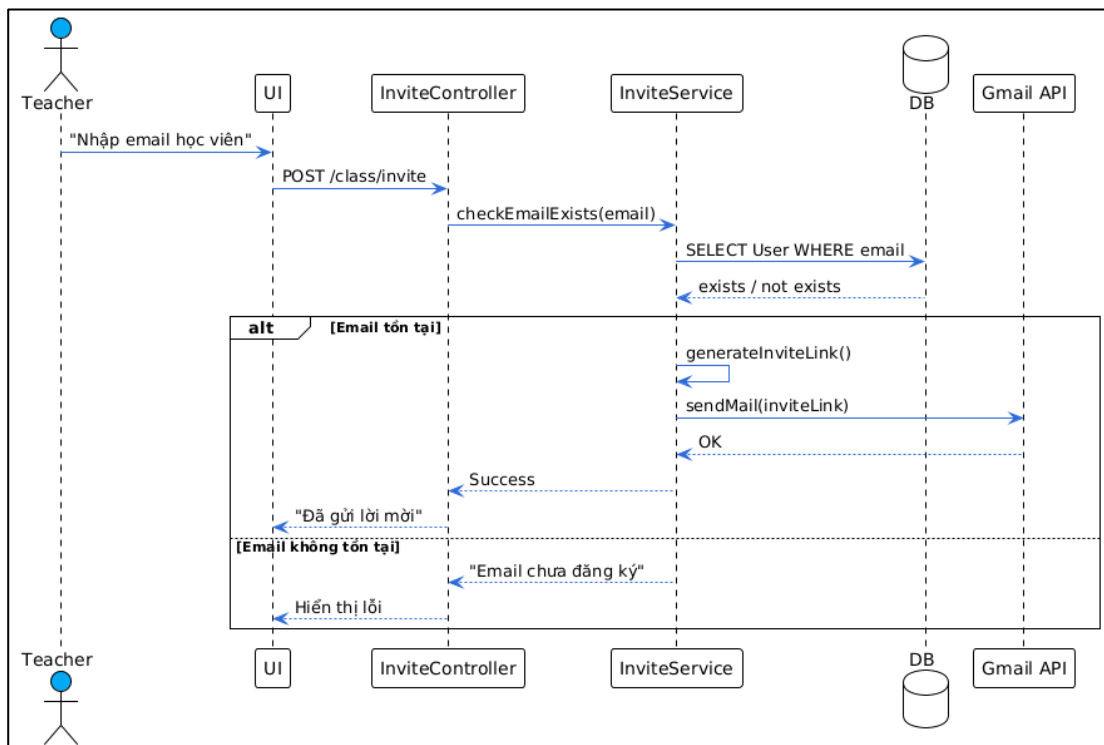
- Mô tả Use Case chức năng Gửi lời mời qua Email

Bảng 2.37. Mô tả Use Case chức năng Gửi lời mời qua Email

STT	Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Nhập Email	Teacher	Nhập Email học viên cần mời.	Lớp đã tồn tại.	Email hợp lệ.
2	Kiểm tra tài khoản	System	Kiểm tra Email đã đăng ký trong hệ thống.	Email nhập vào.	Xác nhận tồn tại.
3	Tạo link mời	System	Tạo link: /join?classCode=xxx	Email hợp lệ.	Link sẵn sàng gửi.
4	Gửi Gmail	System (Gmail API)	Gửi mail mời tham gia lớp.	Có link mời.	Học viên nhận Email.

2.3.13.2. Biểu đồ tuần tự chức năng Gửi lời mời qua Email

- Biểu đồ tuần tự chức năng Gửi lời mời qua Email



Hình 2.26. Biểu đồ tuần tự chức năng Gửi lời mời qua Email

- Luồng chính chức năng Gửi lời mời qua Email

Bảng 2.38. Luồng chính chức năng Gửi lời mời qua Email

Bước	Tác nhân	Mô tả
1	Teacher	Nhập Email học viên cần mời
2	System	Kiểm tra Email.
3	System	Tạo link mời.
4	Gmail API	System (Gmail API)
5	Teacher	Nhận thông báo gửi thành công.

- Luồng phụ chức năng Gửi lời mời qua Email

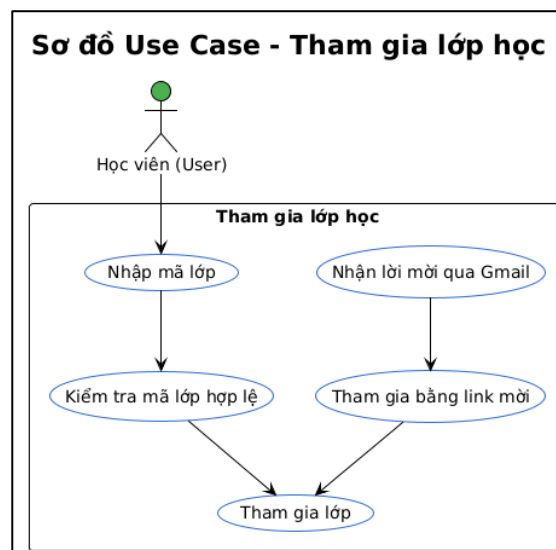
Bảng 2.39. Luồng phụ chức năng Gửi lời mời qua Email

Mã	Tình huống	Xử lý
LP1	Email không có tài khoản	Thông báo “Email chưa đăng ký hệ thống”.
LP2	Gmail API lỗi	Cho phép gửi lại sau.
LP3	Link mời lỗi	Sinh mới link.
LP4	Spam rate limit Gmail	Thử lại sau X phút.

2.3.14. Minh họa chức năng Tham gia lớp học

2.3.14.1. Sơ đồ Use Case chức năng Gửi lời mời qua Tham gia lớp học

- Sơ đồ Use Case chức năng Gửi lời mời qua Tham gia lớp học



Hình 2.27. Sơ đồ Use Case chức năng Tham gia lớp học

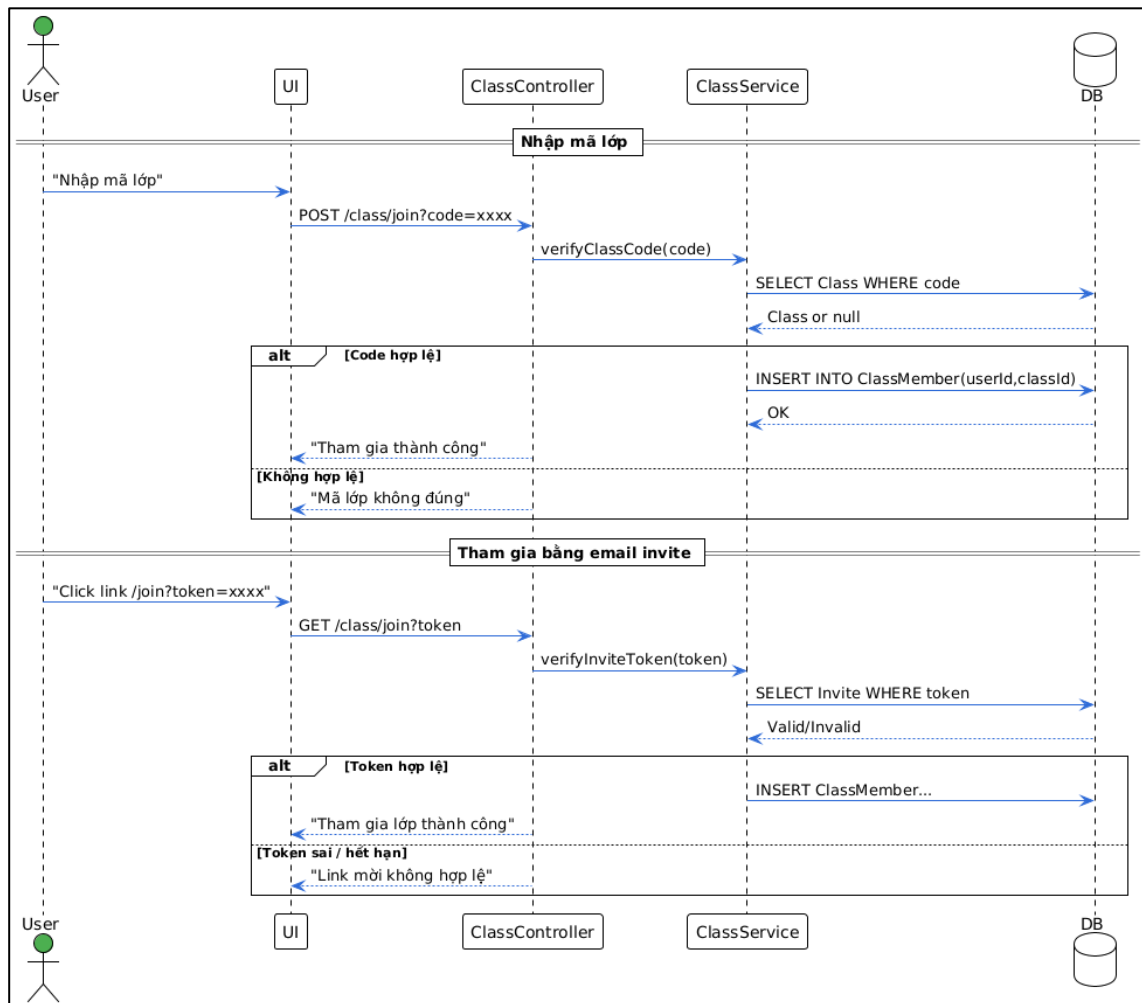
- Mô tả Use Case chức năng Tham gia lớp học

Bảng 2.40. Mô tả Use Case chức năng Tham gia lớp học

STT	Use Case	Tác nhân	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Nhập mã lớp	User	Người dùng nhập mã lớp để tham gia.	Mã lớp phải tồn tại.	Chuyển sang bước kiểm tra mã.
2	Kiểm tra mã lớp	System	Hệ thống kiểm tra mã lớp có hợp lệ hay không.	Mã lớp hợp lệ và lớp đang hoạt động.	Cho phép người dùng tham gia lớp.
3	Tham gia lớp	System / User	Hệ thống ghi vào bảng JoinClass nếu người dùng chưa tham gia trước đó.	Người dùng chưa tham gia lớp trước đó.	Người dùng được thêm vào danh sách lớp.
4	Nhận Email mời	Gmail API (System)	Hệ thống gửi Email mời tham gia lớp thông qua Gmail API.	Email đăng ký tồn tại và hợp lệ.	Người dùng nhận được Email mời.
5	Tham gia bằng link	User	Người dùng nhấn link trong Email → hệ thống tự động thêm vào lớp.	Link hợp lệ, chưa hết hạn.	Người dùng được thêm vào lớp thành công.

2.3.14.2. Biểu đồ tuần tự chức năng Tham gia lớp học

- Biểu đồ tuần tự chức năng Tham gia lớp học



Hình 2.28. Biểu đồ tuần tự chức năng Tham gia lớp học

- Luồng chính chức năng Tham gia lớp học

Bảng 2.41. Luồng chính chức năng Tham gia lớp học

Bước	Tác nhân	Mô tả
1	User	Nhập mã lớp hoặc mở link invite
2	UI	Gửi yêu cầu join
3	Server	Kiểm tra mã lớp / token
4	DB	Kiểm tra hợp lệ
5	Server	Ghi vào bảng ClassMember
6	User	Vào lớp thành công

- Luồng phụ chức năng Tham gia lớp học

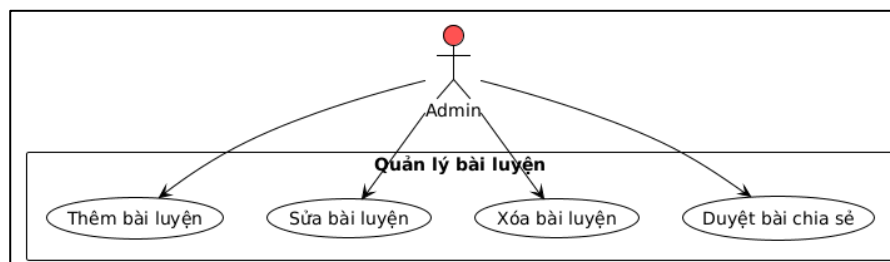
Bảng 2.42. Luồng phụ chức năng Tham gia lớp học

Mã	Tình huống	Xử lý
LP1	Mã lớp sai	Báo lỗi
LP2	Người dùng đã tham gia lớp trước đó	Báo “Bạn đã ở trong lớp”
LP3	Link invite hết hạn	Báo lỗi
LP4	DB lỗi	Thông báo chung
LP5	Lớp đã đóng	Không cho tham gia

2.3.15. Minh họa chức năng Quản lý bài luyện tập

2.3.15.1. Sơ đồ Use Case chức năng Quản lý bài luyện tập

- Sơ đồ Use Case chức năng Quản lý bài luyện tập



Hình 2.29. Sơ đồ Use Case chức năng Quản lý bài luyện tập

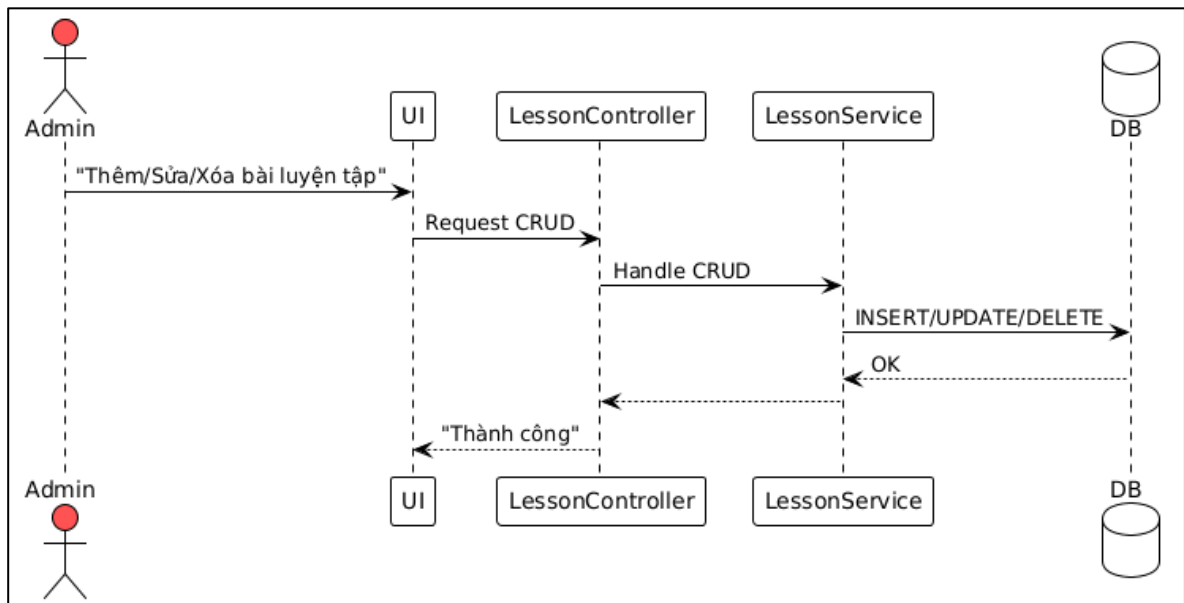
- Mô tả Use Case chức năng Quản lý bài luyện tập

Bảng 2.43. Mô tả Use Case chức năng Quản lý bài luyện tập

STT	Use Case	Actor	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Thêm bài luyện	Admin	Thêm bài luyện mới	Admin đăng nhập	Bài luyện được tạo
2	Sửa bài luyện	Admin	Cập nhật nội dung	Bài tồn tại	Cập nhật thành công
3	Xóa bài luyện	Admin	Xóa bài luyện	Kiểm tra bài luyện	Bài biến mất khỏi hệ thống
4	Duyệt bài chia sẻ	Admin	Duyệt bài từ người dùng	User gửi bài chia sẻ	Lưu thành bài luyện chính thức

2.3.15.2. Biểu đồ tuần tự chức năng Quản lý bài luyện tập

- Biểu đồ tuần tự chức năng Quản lý bài luyện tập



Hình 2.30. Biểu đồ tuần tự chức năng Quản lý bài luyện tập

- Luồng chính chức năng Quản lý bài luyện tập

Bảng 2.44. Luồng chính chức năng Quản lý bài luyện tập

Bước	Actor	Mô tả
1	Admin	Gửi yêu cầu quản lý
2	UI	Gửi request đến Controller
3	Service	Thực hiện CRUD
4	DB	Lưu dữ liệu
5	UI	Báo thành công

- Luồng phụ chức năng Quản lý bài luyện tập

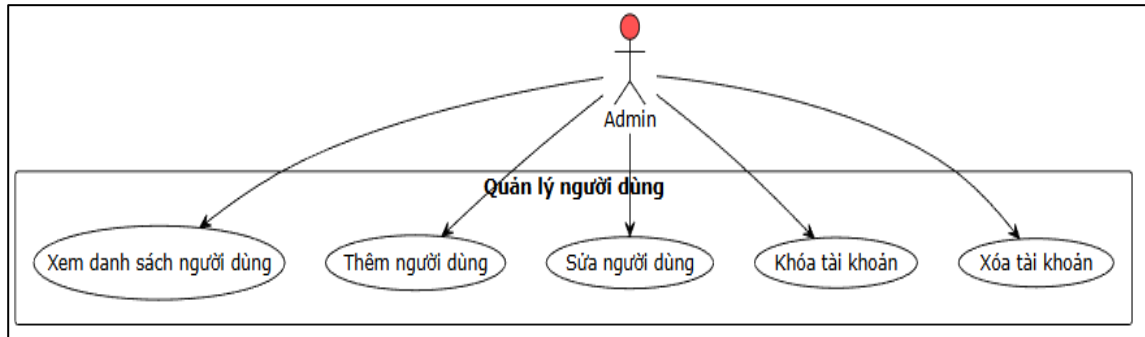
Bảng 2.45. Luồng phụ chức năng Quản lý bài luyện tập

Mã	Tình huống	Xử lý
LP1	Nội dung rỗng	Báo lỗi
LP2	Sửa bài không tồn tại	Báo “Không tìm thấy bài luyện”
LP3	Lỗi DB	Thông báo chung

2.3.16. Minh họa chức năng Quản lý người dùng

2.3.16.1. Sơ đồ Use Case chức năng Quản lý người dùng

- Sơ đồ Use Case chức năng Quản lý người dùng



Hình 2.31. Sơ đồ Use Case chức năng Quản lý người dùng

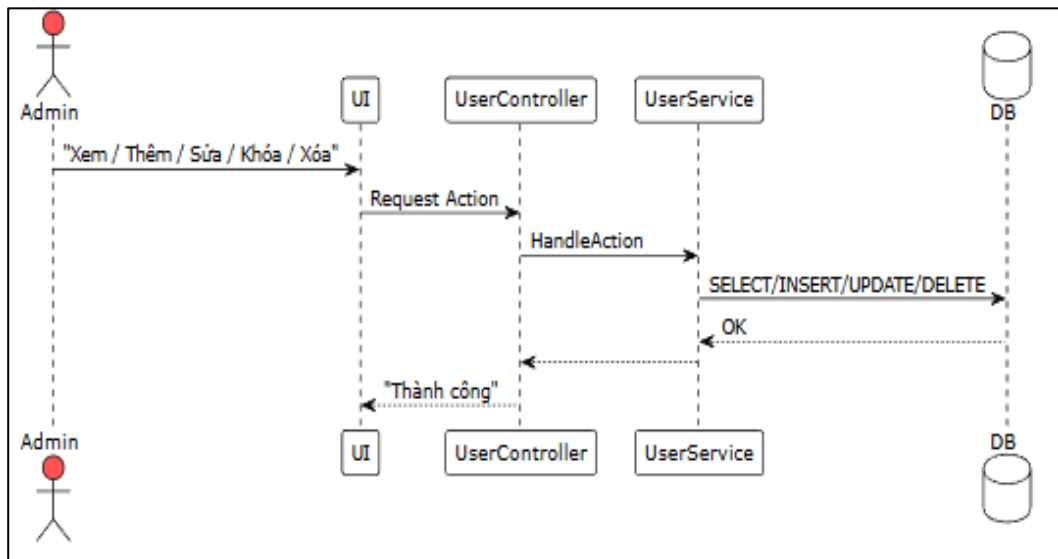
- Mô tả Use Case chức năng Quản lý người dùng

Bảng 2.46. Mô tả Use Case chức năng Quản lý người dùng

STT	Use Case	Actor	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Xem danh sách	Admin	Xem toàn bộ người dùng	Admin đăng nhập	Hiển thị DS
2	Thêm người dùng	Admin	Tạo account thủ công	Tạo mới	Người dùng mới
3	Sửa thông tin	Admin	Cập nhật thông tin user	User tồn tại	Thông tin cập nhật
4	Khóa tài khoản	Admin	Ngăn người dùng đăng nhập	User tồn tại	Account bị khóa
5	Xóa tài khoản	Admin	Xóa user khỏi hệ thống	Kiểm tra	User bị xóa

2.3.16.2. Biểu đồ tuần tự chức năng Quản lý người dùng

- Biểu đồ tuần tự chức năng Quản lý người dùng



Hình 2.32. Biểu đồ tuần tự chức năng Quản lý người dùng

- Luồng chính chức năng Quản lý người dùng

Bảng 2.47. Luồng chính chức năng Quản lý người dùng

Bước	Actor	Mô tả
1	Admin	Chọn thao tác
2	UI	Gửi yêu cầu
3	Controller	Điều hướng Service
4	Service	Thực hiện CRUD
5	DB	Lưu hoặc cập nhật
6	UI	Hiển thị kết quả

- Luồng phụ chức năng Quản lý người dùng

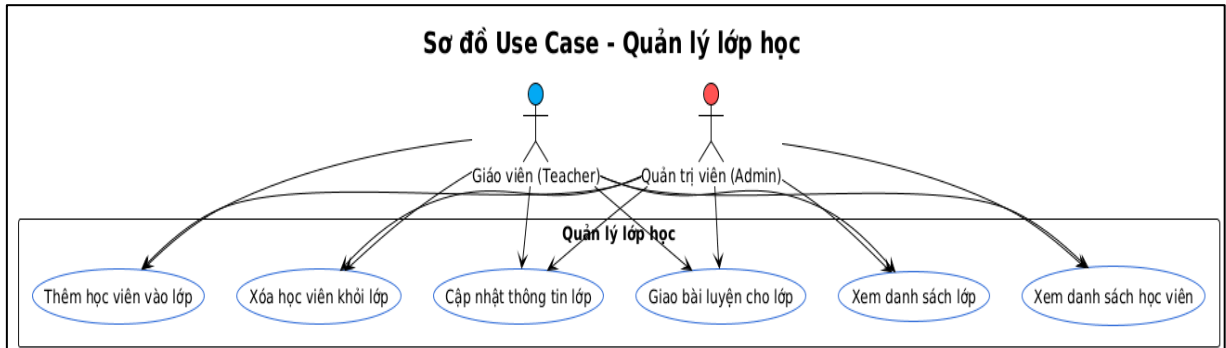
Bảng 2.48. Luồng phụ chức năng Quản lý người dùng

Mã	Tình huống	Xử lý
LP1	Khóa user đã khóa	Báo lỗi
LP2	Email trùng	Không tạo được user mới
LP3	Không tìm thấy user	Thông báo lỗi

2.3.17. Minh họa chức năng Quản lý lớp học

2.3.17.1. Sơ đồ Use Case chức năng Quản lý lớp học

- Sơ đồ Use Case chức năng Quản lý lớp học



Hình 2.33. Sơ đồ Use Case chức năng Quản lý lớp học

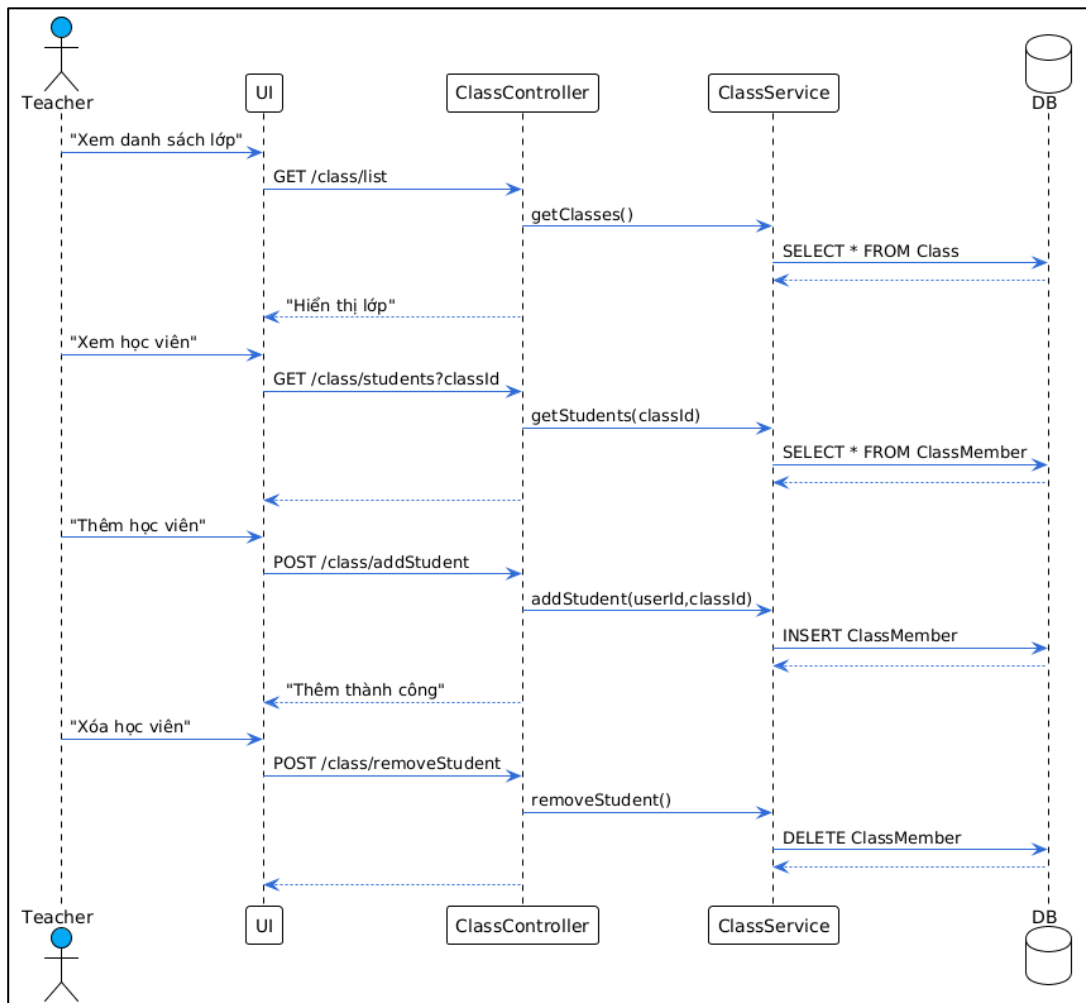
- Mô tả Use Case chức năng Quản lý lớp học

Bảng 2.49. Mô tả Use Case chức năng Quản lý lớp học

STT	Use Case	Actor	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Xem danh sách lớp	Teacher/Admin	Xem các lớp đang quản lý	Đăng nhập	Hiển thị danh sách lớp
2	Xem học viên	Teacher/Admin	Xem học viên trong lớp	Lớp tồn tại	Hiển thị danh sách học viên
3	Thêm học viên	Teacher/Admin	Thêm học viên qua Email hoặc mã	User tồn tại	Học viên được thêm
4	Xóa học viên	Teacher/Admin	Loại học viên khỏi lớp	Học viên tồn tại	Học viên bị xóa
5	Cập nhật lớp	Teacher/Admin	Cập nhật tên, mô tả lớp	Lớp tồn tại	Thông tin cập nhật
6	Giao bài luyện	Teacher/Admin	Giao bài cho cả lớp	Bài luyện tồn tại	Bài giao thành công

2.3.17.2. Biểu đồ tuần tự chức năng Quản lý lớp học

- Biểu đồ tuần tự chức năng Quản lý lớp học



Hình 2.34. Biểu đồ tuần tự chức năng Quản lý lớp học

- Luồng chính chức năng Quản lý lớp học

Bảng 2.50. Luồng chính chức năng Quản lý lớp học

Bước	Actor	Mô tả
1	Teacher/Admin	Xem danh sách lớp
2	System	Lấy danh sách lớp từ DB
3	Teacher/Admin	Xem danh sách học viên
4	Teacher/Admin	Thêm / xóa học viên
5	System	Cập nhật DB
6	Teacher/Admin	Giao bài luyện

- Luồng phụ chức năng Quản lý lớp học

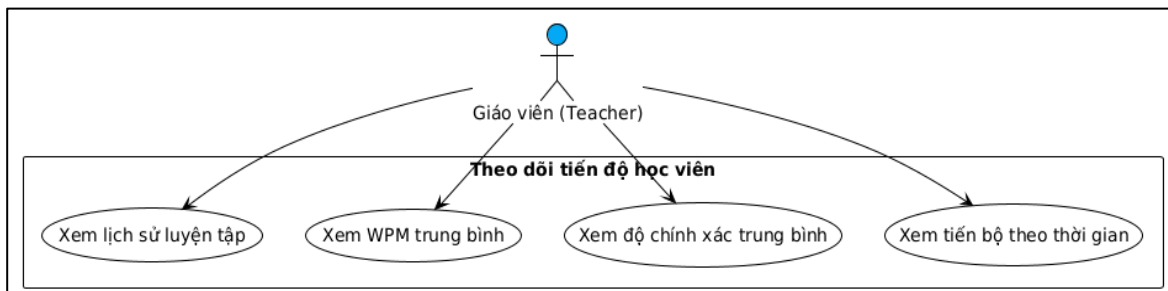
Bảng 2.51. Luồng phụ chức năng Quản lý lớp học

Mã	Tình huống	Xử lý
LP1	User không tồn tại	Báo lỗi
LP2	Lớp không tồn tại	Trả lỗi
LP3	Học viên đã có trong lớp	Báo lỗi
LP4	Lỗi DB	Báo lỗi chung

2.3.18. Minh họa chức năng Theo dõi tiến độ học viên

2.3.18.1. Sơ đồ Use Case chức năng Theo dõi tiến độ học viên

- Sơ đồ Use Case chức năng Theo dõi tiến độ học viên



Hình 2.35. Sơ đồ Use Case chức năng Theo dõi tiến độ học viên

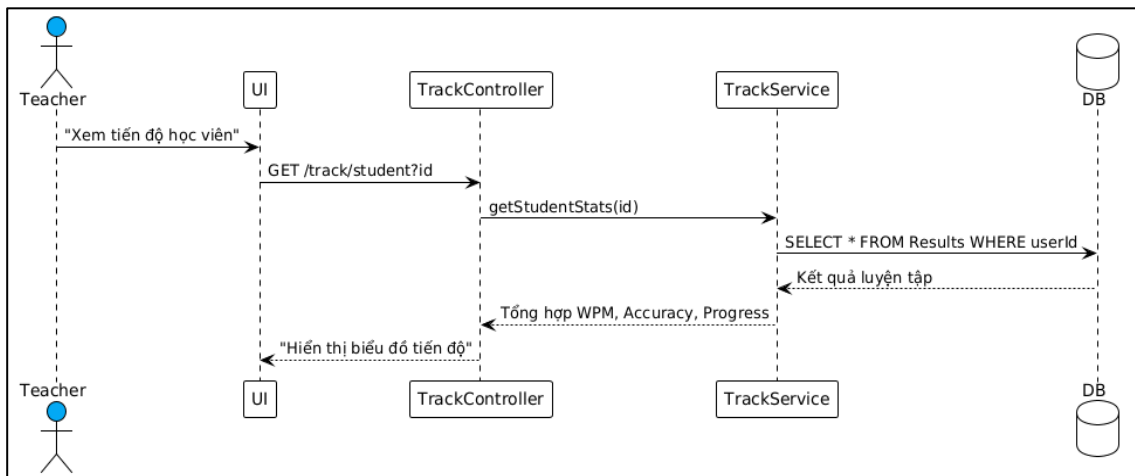
- Mô tả Use Case chức năng Theo dõi tiến độ học viên

Bảng 2.52. Mô tả Use Case chức năng Theo dõi tiến độ học viên

STT	Use Case	Actor	Mô tả
1	Xem lịch sử luyện tập	Teacher	Xem kết quả từng bài luyện
2	Xem WPM trung bình	Teacher	Tính WPM trung bình từ các bài luyện
3	Xem độ chính xác trung bình	Teacher	Tính Độ chính xác trung bình
4	Xem tiến bộ	Teacher	Biểu đồ WPM/Độ chính xác theo thời gian

2.3.18.2. Biểu đồ tuần tự chức năng Theo dõi tiến độ học viên

- Biểu đồ tuần tự chức năng Theo dõi tiến độ học viên



Hình 2.36. Biểu đồ tuần tự chức năng Theo dõi tiến độ học viên

- Luồng chính chức năng Theo dõi tiến độ học viên

Bảng 2.53. Luồng chính chức năng Theo dõi tiến độ học viên

Bước	Mô tả
1	Giáo viên chọn học viên
2	Hệ thống truy vấn kết quả luyện tập
3	Tổng hợp WPM, Độ chính xác
4	Vẽ biểu đồ tiến bộ
5	Hiển thị lên UI

- Luồng phụ chức năng Theo dõi tiến độ học viên (Teacher)

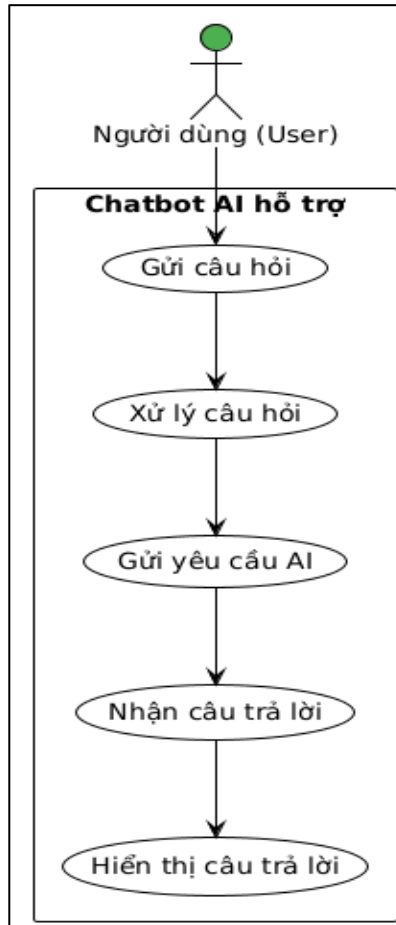
Bảng 2.54. Luồng phụ chức năng Theo dõi tiến độ học viên

Mã	Tình huống	Xử lý
LP1	Học viên chưa luyện tập	Hiển thị “Chưa có dữ liệu”
LP2	Lỗi DB	Báo lỗi
LP3	Không tìm thấy người dùng	Thông báo lỗi

2.3.19. Minh họa chức năng Chatbot AI hỗ trợ người dùng

2.3.19.1. Sơ đồ Use Case chức năng Chatbot AI hỗ trợ người dùng

- Sơ đồ Use Case chức năng Chatbot AI hỗ trợ người dùng



Hình 2.37. Sơ đồ Use Case chức năng Chatbot AI hỗ trợ người dùng

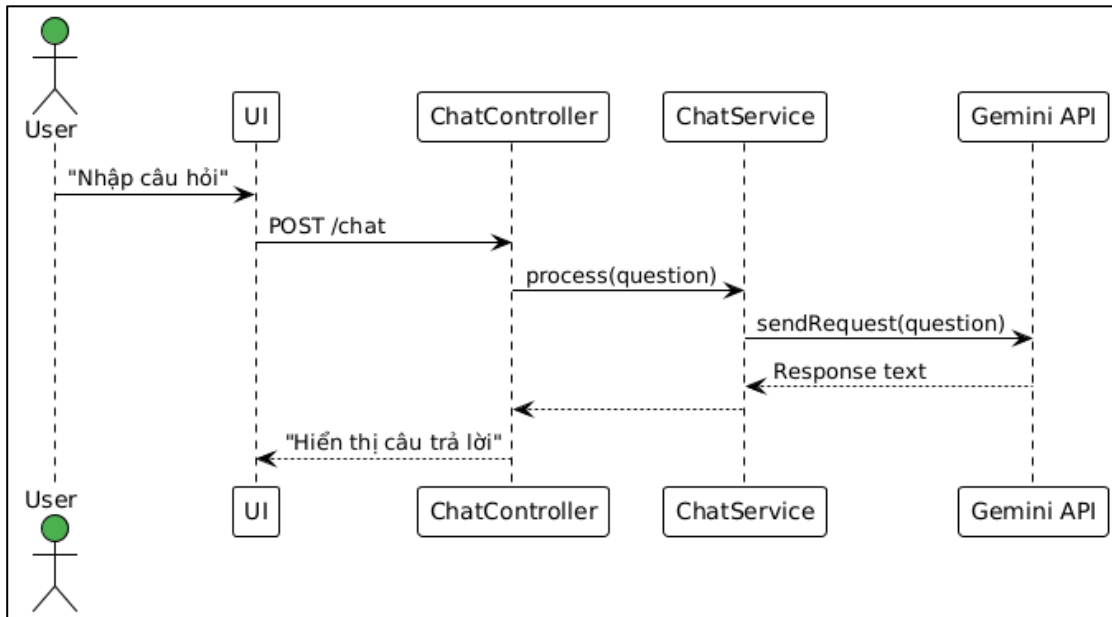
- Mô tả Use Case chức năng Chatbot AI hỗ trợ người dùng

Bảng 2.55. Mô tả Use Case chức năng Chatbot AI hỗ trợ người dùng

STT	Tên Use Case	Tác nhân (Actor)	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Gửi câu hỏi	Người dùng	Người dùng nhập câu hỏi liên quan đến luyện gõ, lớp học, bài luyện, lỗi...	Người dùng đã truy cập giao diện chatbot.	Hệ thống nhận câu hỏi và gửi đi xử lý.
2	Xử lý câu hỏi	Hệ thống	Chuẩn hóa, lọc nội dung, nhận dạng mục đích câu hỏi trước khi gửi sang AI.	Người dùng đã nhập câu hỏi.	Câu hỏi đã được chuẩn hóa để gửi lên AI.
3	Gửi yêu cầu đến AI (Gemini API)	Hệ thống	Tạo request API và gửi câu hỏi lên mô hình Gemini.	Câu hỏi hợp lệ, kết nối API hoạt động.	Nhận được phản hồi từ AI hoặc báo lỗi.
4	Nhận câu trả lời từ AI	Hệ thống	Nhận text từ Gemini API, tổng hợp lại để dễ đọc.	API phản hồi thành công.	Câu trả lời được xử lý và chuẩn bị hiển thị.
5	Hiển thị câu trả lời	UI	Trả câu trả lời lên giao diện người dùng.	Có dữ liệu phản hồi từ AI.	Người dùng nhìn thấy câu trả lời của chatbot.

2.3.19.2. Biểu đồ tuần tự chức năng Chatbot AI hỗ trợ người dùng

- Biểu đồ tuần tự chức năng Chatbot AI hỗ trợ người dùng



Hình 2.38. Biểu đồ tuần tự chức năng Chatbot AI hỗ trợ người dùng

- Luồng chính chức năng Chatbot AI hỗ trợ người dùng

Bảng 2.56. Luồng chính chức năng Chatbot AI hỗ trợ người dùng

Bước	Mô tả
1	Người dùng nhập câu hỏi
2	Hệ thống gửi yêu cầu đến ChatService
3	Gửi câu hỏi lên Gemini API
4	Nhận trả lời
5	Hiển thị câu trả lời

- Luồng phụ chức năng Chatbot AI hỗ trợ người dùng

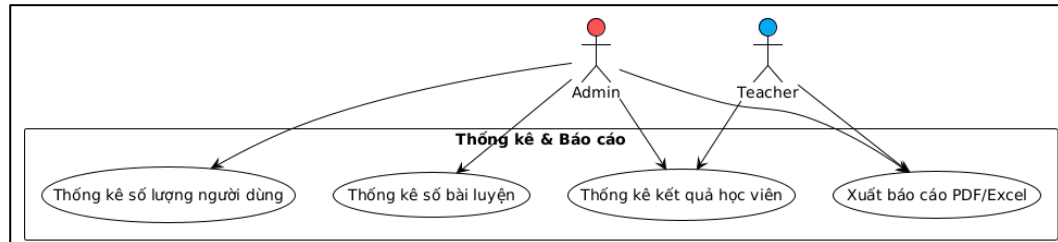
Bảng 2.57. Luồng phụ chức năng Chatbot AI hỗ trợ người dùng

Mã	Tình huống	Xử lý
LP1	AI không phản hồi	Hiển thị “AI đang bận, thử lại”
LP2	Lỗi mạng	Cho phép gửi lại
LP3	Input rỗng	Yêu cầu nhập câu hỏi

2.3.20. Minh họa chức năng Thống kê và Báo cáo

2.3.20.1. Sơ đồ Use Case chức năng Thống kê & Báo cáo

- Sơ đồ Use Case chức năng Thống kê & Báo cáo



Hình 2.39. Sơ đồ Use Case chức năng Thống kê và Báo cáo

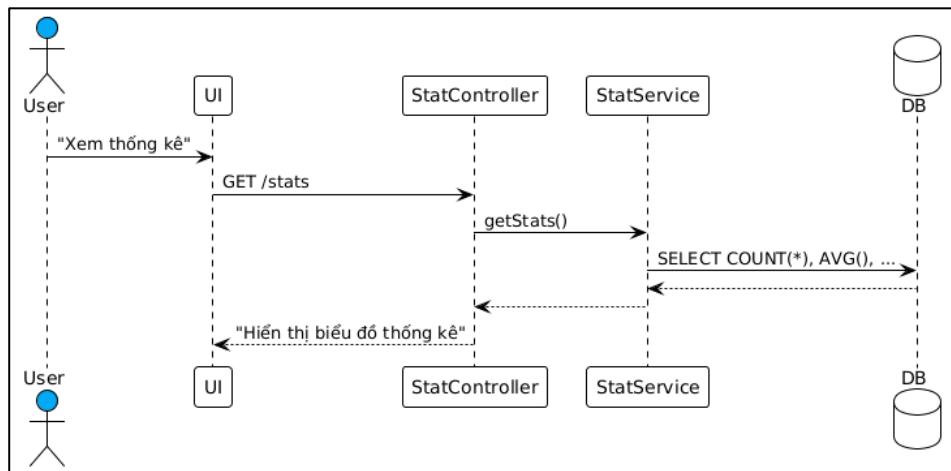
- Mô tả Use Case chức năng Thống kê và Báo cáo

Bảng 2.58. Mô tả Use Case chức năng Thống kê và Báo cáo

STT	Tên Use Case	Tác nhân (Actor)	Mô tả	Tiền điều kiện	Hậu điều kiện
1	Thống kê số lượng người dùng	Admin	Hệ thống hiển thị tổng số người dùng, người mới đăng ký, tài khoản bị khóa.	Admin đã đăng nhập hệ thống.	Các số liệu được hiển thị trên dashboard.
2	Thống kê số lượng bài luyện	Admin	Hệ thống tổng hợp số bài luyện đang hoạt động, bài mới, bài bị ẩn.	Admin đã đăng nhập.	Thông tin thống kê được hiển thị.
3	Thống kê kết quả học viên	Admin/Teacher	Xem WPM, Độ chính xác, số bài đã làm của tất cả học viên hoặc từng lớp.	Giáo viên hoặc Admin đăng nhập, lớp học tồn tại.	Hiển thị thống kê theo từng học viên/bài luyện.
4	Xuất báo cáo (PDF/Excel)	Admin/Teacher	Xuất bản báo cáo tổng hợp kết quả luyện tập hoặc thống kê hệ thống.	Có dữ liệu thống kê, user có quyền.	File báo cáo được lưu hoặc tải xuống.

2.3.20.2. Biểu đồ tuần tự chức năng Thống kê và Báo cáo

- Biểu đồ tuần tự chức năng Thống kê và Báo cáo



Hình 2.40. Biểu đồ tuần tự chức năng Thống kê và Báo cáo

- Luồng chính chức năng Thống kê và Báo cáo

Bảng 2.59. Luồng chính chức năng Thống kê và Báo cáo

Bước	Mô tả
1	Người dùng yêu cầu xem thống kê
2	Server truy vấn DB
3	Tổng hợp số liệu
4	Trả về bảng thống kê
5	Xuất báo cáo (nếu cần)

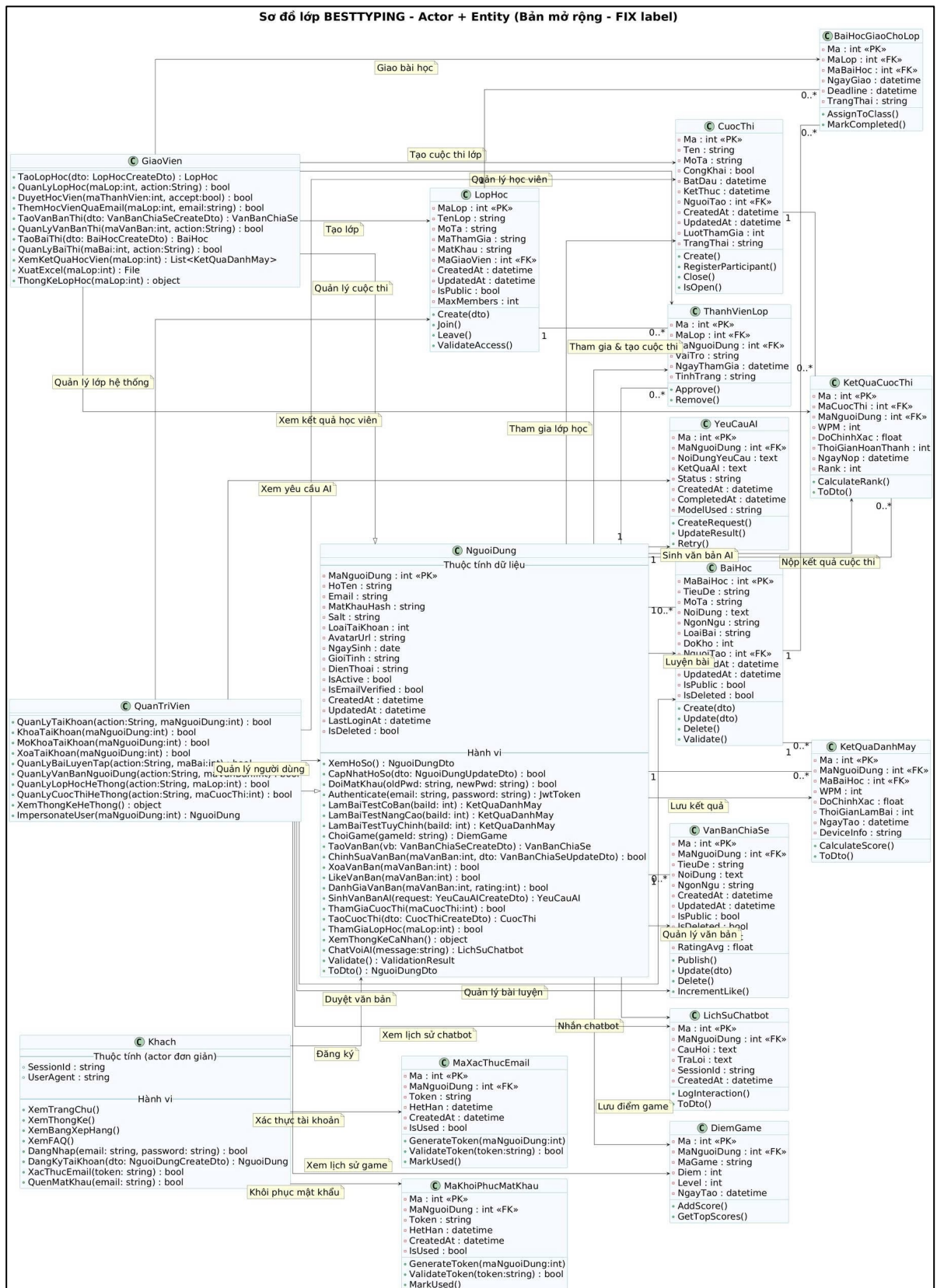
- Luồng phụ chức năng Thống kê và Báo cáo

Bảng 2.60. Luồng phụ chức năng Thống kê và Báo cáo

Mã	Tình huống	Xử lý
LP1	Không có dữ liệu	Hiển thị “Không có số liệu”
LP2	Lỗi DB	Báo lỗi

2.4. Sơ đồ lớp

- Sơ đồ lớp website luyện gõ bàn phím



Hình 2.41. Sơ đồ lớp

- Mô tả các tác nhân sơ đồ lớp

Bảng 2.61. Mô tả các tác nhân sơ đồ lớp

Lớp (Actor)	Thuộc tính	Phương thức	Mô tả
Khach	SessionId, UserAgent	XemTrangChu(), XemThongKe(), XemBangXepHang(), XemFAQ(), DangNhap(), DangKyTaiKhoan(), XacThucEmail(), QuenMatKhau()	Người dùng chưa đăng nhập. Có thể xem nội dung công khai và thực hiện đăng ký, đăng nhập, xác thực Email, khôi phục mật khẩu.
NguoIDung	MaNguoiDung, HoTen, Email, MatKhauHash, LoaiTaiKhoan, AvatarUrl, IsActive, IsEmailVerified,...	XemHoSo(), CapNhatHoSo(), DoiMatKhau(), Authenticate(), LamBaiTest..., TaoVanBan(), ChinhSuaVanBan(), LikeVanBan(), DanhGiaVanBan(), TaoCuocThi(), SinhVanBanAI(), ChatVoiAI(),...	Người dùng hệ thống, có đầy đủ chức năng: luyện tập, quản lý hồ sơ, gửi yêu cầu AI, tham gia lớp và cuộc thi.

Lớp (Actor)	Thuộc tính	Phương thức	Mô tả
GiaoVien	(kế thừa NguoiDung)	TaoLopHoc(), QuanLyLopHoc(), DuyetHocVien(), TaoBaiThi(), GiaoBaiHoc(), XemKetQuaHocVien(), XuatExcel()	Giáo viên có quyền quản lý lớp học, giao bài, xem kết quả của học viên.
QuanTriVien	(kế thừa NguoiDung)	QuanLyTaiKhoan(), KhoaTaiKhoan(), QuanLyVanBanNguoiDung(), QuanLyLopHocHeThong(), QuanLyCuocThiHeThong(), XemThongKeHeThong(), ImpersonateUser()	Quản trị viên có quyền quản lý toàn bộ hệ thống, người dùng, lớp học, nội dung, cuộc thi.

- Mô tả các lớp thực thể

Bảng 2.62. Mô tả các lớp thực thể

Lớp	Thuộc tính chính	Mô tả
MaXacThucEmail	Ma, MaNguoiDung, Token, HetHan, IsUsed, CreatedAt	Lưu token xác thực Email của người dùng khi đăng ký.
MaKhoiPhucMatKhau	Ma, MaNguoiDung, Token, HetHan, IsUsed, CreatedAt	Lưu token phục hồi mật khẩu khi người dùng quên mật khẩu.

Lớp	Thuộc tính chính	Mô tả
BaiHoc	MaBaiHoc, TieuDe, NoiDung, LoaiBai, DoKho, NguoiTao, CreatedAt, IsPublic	Bài học/bài luyện đánh máy. Có thể công khai hoặc riêng tư.
KetQuaDanhMay	Ma, MaNguoiDung, MaBaiHoc, WPM, DoChinhXac, ThoiGianLamBai, NgayTao	Kết quả một lần làm bài của người dùng.
VanBanChiaSe	Ma, MaNguoiDung, TieuDe, NoiDung, NgonNgu, LikeCount, RatingAvg, CreatedAt	Văn bản do người dùng tạo và chia sẻ.
LopHoc	MaLop, TenLop, MoTa, MaThamGia, MatKhau, MaGiaoVien, MaxMembers	Lớp học trực tuyến để giáo viên quản lý học viên và giao bài.
ThanhVienLop	Ma, MaLop, MaNguoiDung, VaiTro, NgayThamGia, TinhTrang	Thông tin thành viên trong lớp học (học viên / giám sát).
BaiHocGiaoChoLop	Ma, MaLop, MaBaiHoc, NgayGiao, Deadline, TrangThai	Một bài học được giao cho lớp với hạn nộp và trạng thái.
CuocThi	Ma, Ten, MoTa, CongKhai, BatDau, KetThuc, NguoiTao, TrangThai	Cuộc thi đánh máy cho người dùng hoặc lớp học.
KetQuaCuocThi	Ma, MaCuocThi, MaNguoiDung, WPM, DoChinhXac, ThoiGianHoanThanh, NgayNop, Rank	Kết quả nộp bài của người dự thi.
DiemGame	Ma, MaNguoiDung, MaGame, Diem, Level, NgayTao	Lưu điểm và cấp độ của người chơi trong các mini game.

Lớp	Thuộc tính chính	Mô tả
YeuCauAI	Ma, MaNguoiDung, NoiDungYeuCau, KetQuaAI, Status, ModelUsed	Lưu yêu cầu AI sinh văn bản từ người dùng.
LichSuChatbot	Ma, MaNguoiDung, CauHoi, TraLoi, SessionId, CreatedAt	Lưu lịch sử tương tác chatbot/AI.

- Mô tả quan hệ giữa tác nhân và thực thể của sơ đồ lớp

Bảng 2.63. Mô tả quan hệ giữa tác nhân và thực thể của sơ đồ lớp

Tác nhân	Thực thể liên quan	Loại quan hệ	Mô tả
Khách	NguoiDung	Tạo / Khởi tạo	Đăng ký tài khoản mới.
Khách	MaXacThucEmail	Đọc / Xác thực	Sử dụng token để xác thực Email.
Khách	MaKhoiPhucMatKhau	Tạo / Xác thực	Gửi yêu cầu quên mật khẩu và nhập token khôi phục.
NguoiDung	BaiHoc	Thực hiện / Xem	Làm bài luyện; xem danh sách bài học.
NguoiDung	KetQuaDanhMay	Tạo	Lưu kết quả mỗi lần luyện.
NguoiDung	VanBanChiaSe	CRUD	Tạo, sửa, xóa, like văn bản chia sẻ.
NguoiDung	CuocThi	Tạo / Tham gia	Tạo cuộc thi hoặc tham gia thi.
NguoiDung	KetQuaCuocThi	Tạo	Nộp bài thi và lưu kết quả.
NguoiDung	DiemGame	Tạo	Lưu điểm game khi chơi.
NguoiDung	YeuCauAI	Tạo	Gửi prompt AI và nhận kết quả.

Tác nhân	Thực thể liên quan	Loại quan hệ	Mô tả
NguoiDung	LichSuChatbot	Tạo	Gửi tin nhắn chatbot và lưu lịch sử AI trả lời.
NguoiDung	ThanhVienLop	Tạo	Gửi yêu cầu tham gia lớp học.
GiaoVien	LopHoc	Tạo / Quản lý	Tạo lớp và quản lý học viên.
GiaoVien	BaiHocGiaoChoLop	Tạo	Giao bài luyện cho lớp.
GiaoVien	KetQuaCuocThi	Đọc	Xem kết quả của học viên.
QuanTriVien	NguoiDung	Quản trị	Khóa / mở khóa / xóa người dùng.
QuanTriVien	BaiHoc	Quản trị	Quản lý nội dung bài luyện.
QuanTriVien	VanBanChiaSe	Quản trị	Kiểm duyệt văn bản chia sẻ.
QuanTriVien	LopHoc	Quản trị	Quản lý lớp học toàn hệ thống.
QuanTriVien	CuocThi	Quản trị	Quản lý cuộc thi hệ thống.
QuanTriVien	YeuCauAI	Giám sát	Xem yêu cầu AI của mọi người dùng.
QuanTriVien	LichSuChatbot	Giám sát	Xem lịch sử chatbot toàn hệ thống.

- Mô tả quan hệ giữa các thực thể

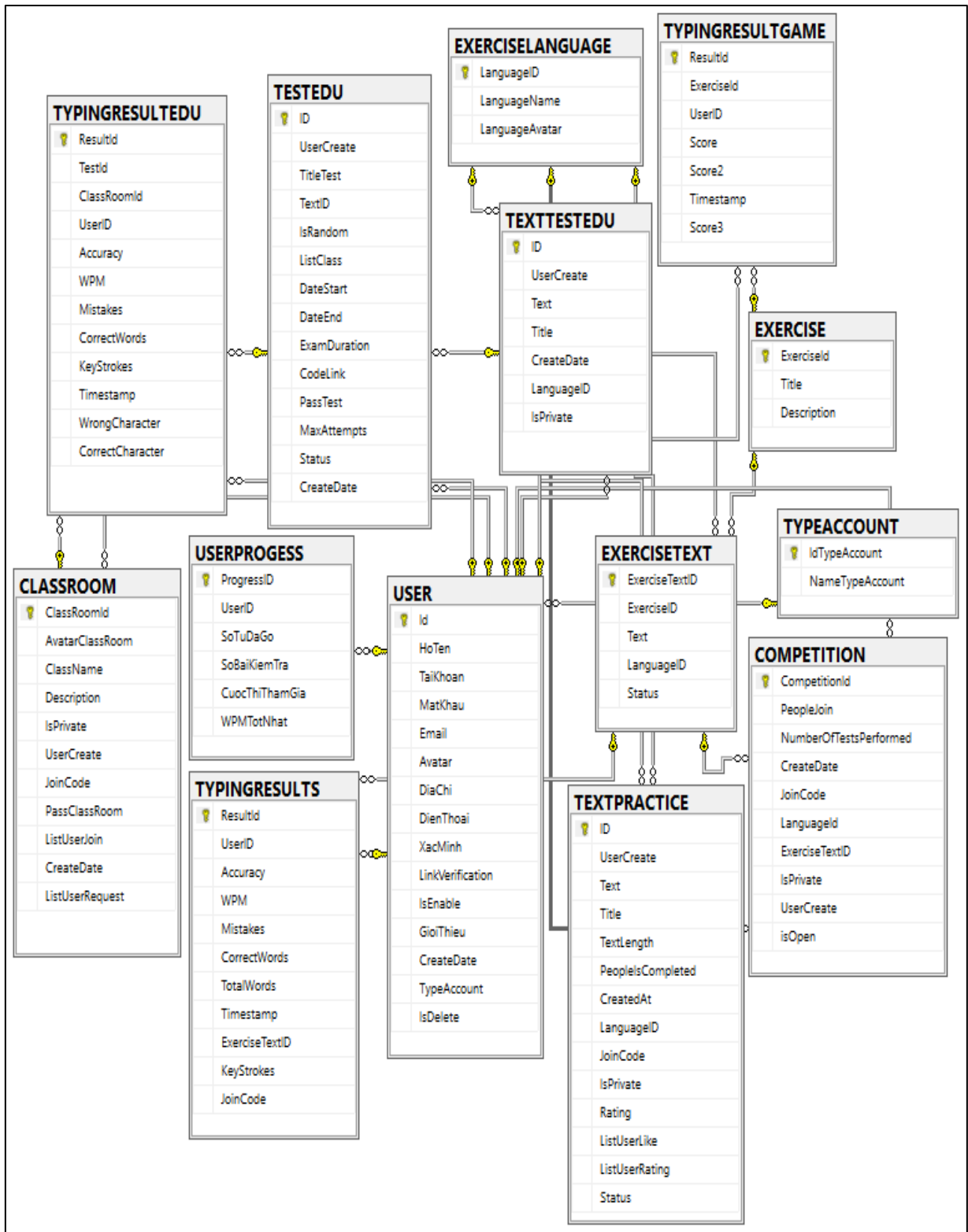
Bảng 2.64. Mô tả quan hệ giữa các thực thể

STT	Quan hệ giữa các thực thể	Kiểu quan hệ (1-1 / 1-N / N-N)	Thuộc tính khóa ngoại	Mô tả quan hệ
1	Người Dùng ↔ Lớp Học	1 – N	Lớp Học.Mã Giáo Viên → Người Dùng.Mã Người Dùng	Mỗi giáo viên quản lý nhiều lớp học; mỗi lớp học chỉ có một giáo viên.
2	Người Dùng ↔ Thành Viên Lớp	1 – N	Thành Viên Lớp.Mã Người Dùng → Người Dùng	Một người dùng tham gia nhiều lớp; mỗi bản ghi thuộc 1 người dùng.
3	Lớp Học ↔ Thành Viên Lớp	1 – N	Thành Viên Lớp.Mã Lớp → Lớp Học.Mã Lớp	Một lớp có nhiều thành viên; một thành viên lớp thuộc một lớp.
4	Người Dùng ↔ Yêu Cầu AI	1 – N	Yêu Cầu AI.Mã Người Dùng → Người Dùng	Người dùng gửi nhiều yêu cầu AI; mỗi yêu cầu thuộc 1 người.
5	Người Dùng ↔ Lịch Sử Chatbot	1 – N	Lịch Sử Chatbot.Mã Người Dùng → Người Dùng	Người dùng có nhiều lần tương tác chatbot; mỗi bản ghi thuộc 1 người.
6	Người Dùng ↔ Bài Học	1 – N	Bài Học.Mã Người Tạo → Người Dùng.Mã Người Dùng	Giáo viên tạo nhiều bài học; bài học chỉ có 1 người tạo.

STT	Quan hệ giữa các thực thể	Kiểu quan hệ (1-1 / 1-N / N-N)	Thuộc tính khóa ngoại	Mô tả quan hệ
7	Bài Học ↔ Kết Quả Đánh Máy	1 – N	Kết Quả Đánh Máy.Mã Bài Học → Bài Học	Một bài học có nhiều kết quả đánh máy của học viên.
8	Người Dùng ↔ Kết Quả Đánh Máy	1 – N	Kết Quả Đánh Máy.Mã Người Dùng → Người Dùng	Một người dùng có nhiều kết quả luyện tập.
9	Người Dùng (Giáo viên) ↔ Giao Bài (qua Lớp Học)	1 – N	Lớp Học.Mã Giáo Viên	Giáo viên giao nhiều bài cho lớp; lớp thuộc 1 giáo viên.
10	Người Dùng ↔ Văn Bản Chia Sẻ	1 – N	Văn Bản Chia Sẻ.Mã Người Dùng	Người dùng chia sẻ nhiều văn bản; mỗi văn bản thuộc 1 người.
11	Người Dùng ↔ Cuộc Thi	1 – N	Cuộc Thi.Mã Người Tạo → Người Dùng	Một giáo viên có thể tạo nhiều cuộc thi.
12	Cuộc Thi ↔ Kết Quả Cuộc Thi	1 – N	Kết Quả Cuộc Thi.Mã Cuộc Thi → Cuộc Thi.Mã	Một cuộc thi có nhiều kết quả của học viên.
13	Người Dùng ↔ Kết Quả Cuộc Thi	1 – N	Kết Quả Cuộc Thi.Mã Người Dùng → Người Dùng	Một người tham gia nhiều cuộc thi, mỗi kết quả được ghi riêng.
14	Người Dùng ↔ Mã Xác Thực Email	1 – N	Mã Xác Thực Email.Mã Người Dùng → Người Dùng	Một người nhận nhiều mã OTP xác thực Email.

STT	Quan hệ giữa các thực thể	Kiểu quan hệ (1-1 / 1-N / N-N)	Thuộc tính khóa ngoại	Mô tả quan hệ
15	Người Dùng ↔ Mã Khôi Phục Mật Khẩu	1 – N	Mã Khôi Phục. Mã Người Dùng → Người Dùng	Một user có thể yêu cầu nhiều mã khôi phục.
16	Người Dùng ↔ Điểm Game	1 – N	Điểm Game. Mã Người Dùng → Người Dùng	Một người có nhiều kết quả game.
17	Người Dùng ↔ Phòng Trò Chuyện	N – N	Qua bảng Thành Viên Phòng	Một người tham gia nhiều phòng; một phòng có nhiều người.
18	Phòng Trò Chuyện ↔ Thành Viên Phòng	1 – N	Thành Viên Phòng. Mã Phòng → Phòng	Một phòng có nhiều thành viên.
19	Người Dùng ↔ Thành Viên Phòng	1 – N	Thành Viên Phòng. Mã Người Dùng → Người Dùng	Ghi lại các phòng mà người dùng tham gia.

2.5.2. Mô hình quan hệ



Hình 2.43. Mô hình quan hệ

- Mô tả mô hình quan hệ

Bảng 2.65. Mô tả mô hình quan hệ

Nhóm thực thể	Tên bảng	Chức năng chính	Quan hệ quan trọng	Mô tả tổng quát
1. Người dùng và phân quyền	USER	Lưu thông tin tài khoản, hồ sơ, xác minh, trạng thái hoạt động.	1–n với hầu hết bảng nội dung: TEXTPRACTICE, TESTEDU, COMPETITION, TYPINGRESULTS, TYPINGRESULTGAME, ...	Đây là trung tâm của hệ thống. Mọi hoạt động luyện tập, tham gia thi, tạo nội dung đều gắn với USER.
	TYPEACCOUNT	Định nghĩa loại tài khoản: Guest, User, Teacher, Admin.	1–n với USER.	Giúp phân quyền và kiểm soát chức năng của từng loại người dùng.
2. Văn bản luyện tập	TEXTPRACTICE	Văn bản luyện tập do người dùng tạo (AI hoặc thủ công).	n–1 với USER; n–1 với EXERCISELANGUAGE.	Lưu nguồn nội dung chính cho việc luyện gõ, người dùng có thể chia sẻ, đánh giá, hoặc đề riêng tư.

Nhóm thực thể	Tên bảng	Chức năng chính	Quan hệ quan trọng	Mô tả tổng quát
và ngôn ngữ	EXERCISELANGUAGE	Danh sách ngôn ngữ.	1–n với TEXTPRACTICE, EXERCISETEXT, TEXTTESTEDU, COMPETITION.	Một văn bản hoặc bài luyện tập luôn gắn với đúng một ngôn ngữ.
3. Bài tập hệ thống và văn bản bài tập	EXERCISE	Bài luyện tạo sẵn trong hệ thống (định nghĩa chung).	1–n với EXERCISETEXT.	Không chứa nội dung, chỉ là cấu trúc bài tập.
	EXERCISETEXT	Text nội dung cho một bài EXERCISE.	n–1 với EXERCISE; n–1 với EXERCISELANGUAGE; n–1 với COMPETITION và TYPINGRESULTS.	Là nơi chứa nội dung thực tế dùng để luyện tập, thi đấu hoặc mini-game.
4. Kiểm tra giáo dục	TEXTTESTEDU	Văn bản của bài kiểm tra giáo dục.	n–1 với USER; n–1 với EXERCISELANGUAGE; 1–n với TESTEDU.	Cho phép giáo viên hoặc user tạo văn bản để dùng trong các bài test.

Nhóm thực thể	Tên bảng	Chức năng chính	Quan hệ quan trọng	Mô tả tổng quát
(Education Test)	TESTEDU	Bài kiểm tra giáo dục, có thời gian, số lần làm, trạng thái.	n-1 với USER; n-1 với TEXTTESTEDU; 1-n với TYPINGRESULTEDU.	Được sử dụng trong lớp học hoặc cá nhân để đánh giá năng lực gõ bàn phím.
	TYPINGRESULTEDU	Kết quả làm bài TestEdu.	n-1 với TESTEDU, USER, CLASSROOM.	Ghi nhận kết quả kiểm tra của học sinh hoặc người dùng.
5. Lớp học (Classroom)	CLASSROOM	Lớp học do giáo viên tạo.	n-1 với USER; 1-n với TYPINGRESULTEDU.	Hỗ trợ làm bài tập theo lớp, kiểm tra theo lớp, quản lý học viên.
	USERPROGRESS	Tiến trình luyện tập của người dùng.	n-1 với USER.	Thống kê tóm tắt quá trình học/luyện tập.
6. Kết quả luyện tập /	TYPINGRESULTS	Kết quả luyện tập (exercise text).	n-1 với USER; n-1 với EXERCISETEXT.	Ghi lại dữ liệu tập luyện: Độ chính xác,

Nhóm thực thể	Tên bảng	Chức năng chính	Quan hệ quan trọng	Mô tả tổng quát
mini-game				WPM, keystrokes.
	TYPINGRESULT GAME	Kết quả mini-game (speed/Độ chính xác challenges).	n-1 với USER; n-1 với EXERCISE.	Lưu điểm game theo nhiều cấp độ (Score/Score2/Score3).
7. Cuộc thi	COMPETITION	Cuộc thi gõ bàn phím cá nhân hoặc theo lớp.	n-1 với USER; n-1 với EXERCISETEXT; n-1 với EXERCISELANGUAGE.	Cho phép người dùng tạo cuộc thi công khai/riêng tư với mã tham gia.

- Mô tả bảng USER

Bảng 2.66. Mô tả bảng USER

Tên trường	Kiểu dữ liệu	Null	Khóa	Mô tả
Id	int	Không	Khóa chính	Mã người dùng
HoTen	nvarchar(100)	Có		Họ tên
TaiKhoan	nvarchar(100)	Không		Tài khoản đăng nhập
MatKhau	nvarchar(255)	Không		Mật khẩu đã bấm
Email	nvarchar(255)	Không		Địa chỉ Email
Avatar	nvarchar(255)	Có		Ảnh đại diện
DiaChi	nvarchar(255)	Có		Địa chỉ
DienThoai	nvarchar(20)	Có		Số điện thoại
XacMinh	bit	Không		Trạng thái xác minh Email
LinkVerification	nvarchar(255)	Có		Link xác thực
IsEnable	bit	Không		Kích hoạt tài khoản
GioiThieu	nvarchar(max)	Có		Giới thiệu bản thân
CreateDate	datetime	Không		Ngày tạo
TypeAccount	int	Không	Khóa ngoại → TYPEACCOUNT. IdTypeAccount	Loại tài khoản
IsDelete	bit	Không		Xóa mềm

- Mô tả bảng TYPEACCOUNT

Bảng 2.67. Mô tả bảng TYPEACCOUNT

Tên trường	Kiểu dữ liệu	Null	Khóa	Mô tả
IdTypeAccount	int	Không	Khóa chính	Mã loại tài khoản
NameTypeAccount	nvarchar(100)	Không		Tên loại tài khoản

- Mô tả bảng CLASSROOM

Bảng 2.68. Mô tả bảng CLASSROOM

Tên trường	Kiểu dữ liệu	Null	Khóa	Mô tả
ClassRoomId	int	Không	Khóa chính	Mã lớp học
AvatarClassRoom	nvarchar(255)	Có		Avatar của lớp
ClassName	nvarchar(255)	Không		Tên lớp
Description	nvarchar(max)	Có		Mô tả lớp
IsPrivate	bit	Không		Lớp riêng tư hay công khai
UserCreate	int	Không	Khóa ngoại → USER.Id	Người tạo lớp
JoinCode	nvarchar(50)	Có		Mã tham gia lớp
PassClassRoom	nvarchar(50)	Có		Mật khẩu lớp
ListUserJoin	nvarchar(max)	Có		Danh sách người tham gia
CreateDate	datetime	Không		Ngày tạo lớp
ListUserRequest	nvarchar(max)	Có		Danh sách yêu cầu tham gia

- Mô tả bảng USERPROGRESS

Bảng 2.69. Mô tả bảng USERPROGRESS

Trường	Kiểu	Null	Khóa	Mô tả
ProgressID	int	Không	Khóa chính	Mã tiến trình
UserID	int	Không	Khóa ngoại → USER.Id	Người dùng
SoTuDaGo	int	Không		Số từ đã gõ
SoBaiKiemTra	int	Không		Số bài kiểm tra
CuocThiThamGia	int	Không		Số cuộc thi
WPMTotNhat	int	Không		Tốc độ cao nhất

- Mô tả bảng EXERCISELANGUAGE

Bảng 2.70. Mô tả bảng EXERCISELANGUAGE

Trường	Kiểu	Null	Khóa	Mô tả
LanguageID	int	Không	Khóa chính	Mã ngôn ngữ
LanguageName	nvarchar(100)	Không		Tên ngôn ngữ
LanguageAvatar	nvarchar(255)	Có		Icon hiển thị

- Mô tả bảng EXERCISE

Bảng 2.71. Mô tả bảng EXERCISE

Trường	Kiểu	Null	Khóa	Mô tả
ExerciseId	int	Không	Khóa chính	Mã bài luyện tập
Title	nvarchar(255)	Không		Tiêu đề
Description	nvarchar(max)	Có		Mô tả

- Mô tả bảng EXERCISETEXT

Bảng 2.72. Mô tả bảng EXERCISETEXT

Trường	Kiểu	Null	Khóa	Mô tả
ExerciseTextID	int	Không	Khóa chính	Mã văn bản bài tập
ExerciseID	int	Không	Khóa ngoại → EXERCISE.ExerciseId	Bài luyện tập
Text	nvarchar (max)	Không		Nội dung
LanguageID	int	Không	Khóa ngoại → EXERCISELANGUAGE.LanguageID	Ngôn ngữ
Status	int	Không		Trạng thái

- Mô tả bảng TEXTTESTEDU

Bảng 2.73. Mô tả bảng TEXTTESTEDU

Trường	Kiểu	Null	Khóa	Mô tả
ID	int	Không	Khóa chính	Mã văn bản test
UserCreate	int	Không	Khóa ngoại → USER.Id	Người tạo
Text	nvarchar (max)	Không		Nội dung bài test
Title	nvarchar (255)	Không		Tiêu đề
CreateDate	datetime	Không		Ngày tạo
LanguageID	int	Không	Khóa ngoại → EXERCISELANGUAGE.LanguageID	Ngôn ngữ
IsPrivate	bit	Không		Riêng tư hay công khai

- Mô tả bảng TESTEDU

Bảng 2.74. Mô tả bảng TESTEDU

Tên trường	Kiểu	Null	Khóa	Mô tả
ID	int	Không	Khóa chính	Mã bài thi
UserCreate	int	Không	Khóa ngoại → USER.Id	Người tạo
TitleTest	nvarchar (255)	Không		Tên bài thi
TextID	int	Không	Khóa ngoại → TEXTTESTED U.ID	Văn bản làm bài
IsRandom	bit	Không		Ngẫu nhiên hay không
ListClass	nvarchar (max)	Có		Các lớp được giao
DateStart	datetime	Không		Thời gian bắt đầu
DateEnd	datetime	Không		Thời gian kết thúc
ExamDuration	int	Không		Thời gian làm bài
CodeLink	nvarchar (50)	Có		Mã link truy cập
PassTest	int	Có		Điểm yêu cầu
MaxAttempts	int	Có		Số lần làm tối đa
Status	int	Không		Trạng thái
CreateDate	datetime	Không		Ngày tạo

- Mô tả bảng TYPINGRESULTEDU

Bảng 2.75. Mô tả bảng TYPINGRESULTEDU

Tên trường	Kiểu	Null	Khóa	Mô tả
ResultId	int	Không	Khóa chính	Mã kết quả
TestId	int	Không	Khóa ngoại → TESTEDU.ID	Mã bài thi
ClassRoomId	int	Có	Khóa ngoại → CLASSROOM.ClassRoomId	Lớp
UserID	int	Không	Khóa ngoại → USER.Id	Người nộp
Độ chính xác	float	Không		Độ chính xác
WPM	int	Không		Tốc độ
Mistakes	int	Không		Số lỗi
CorrectWords	int	Không		Từ đúng
KeyStrokes	int	Không		Tổng lần gõ
Timestamp	datetime	Không		Thời gian
WrongCharacter	int	Không		Ký tự sai
CorrectCharacter	int	Không		Ký tự đúng

- Mô tả bảng TYPINGRESULTS

Bảng 2.76. Mô tả bảng TYPINGRESULTS

Tên trường	Kiểu	Null	Khóa	Mô tả
ResultId	int	Không	Khóa chính	Mã kết quả
UserID	int	Không	Khóa ngoại → USER.Id	Người dùng
Độ chính xác	float	Không		Độ chính xác
WPM	int	Không		Tốc độ
Mistakes	int	Không		Số lỗi

Tên trường	Kiểu	Null	Khóa	Mô tả
CorrectWords	int	Không		Từ đúng
TotalWords	int	Không		Tổng số từ
Timestamp	datetime	Không		Thời gian
ExerciseTextID	int	Không	Khóa ngoại → EXERCISETEXT.ExerciseTextID	Văn bản tập
KeyStrokes	int	Không		Lần gõ
JoinCode	nvarchar(50)	Có		Mã tham gia

- Mô tả bảng TYPINGRESULTGAME

Bảng 2.77. Mô tả bảng TYPINGRESULTGAME

Trường	Kiểu	Null	Khóa	Mô tả
ResultId	int	Không	Khóa chính	Mã kết quả game
ExerciseId	int	Không	Khóa ngoại → EXERCISE.ExerciseId	Chế độ game
UserID	int	Không	Khóa ngoại → USER.Id	Người chơi
Score	int	Không		Điểm level 1
Score2	int	Không		Điểm level 2
Score3	int	Không		Điểm level 3
Timestamp	datetime	Không		Thời gian tạo

- Mô tả bảng COMPETITION

Bảng 2.78. Mô tả bảng COMPETITION

Tên trường	Kiểu	Null	Khóa	Mô tả
CompetitionId	int	Không	Khóa chính	Mã cuộc thi
PeopleJoin	int	Không		Số người tham gia
NumberOfTestsPerformed	int	Không		Tổng lượt làm
CreateDate	datetime	Không		Ngày tạo
JoinCode	nvarchar (50)	Có		Mã tham gia
LanguageId	int	Không	Khóa ngoại → EXERCISELANGUAGE .LanguageID	Ngôn ngữ
ExerciseTextID	int	Không	Khóa ngoại → EXERCISETEXT .ExerciseTextID	Văn bản bài thi
IsPrivate	bit	Không		Công khai hay riêng tư
UserCreate	int	Không	Khóa ngoại →USER.Id	Người tạo
IsOpen	bit	Không		Trạng thái mở

- Mô tả bảng TEXTPRACTICE

Bảng 2.79. Mô tả bảng TEXTPRACTICE

Tên trường	Kiểu dữ liệu	Null	Khóa	Mô tả
ID	int	Không	Khóa chính	Mã văn bản luyện tập
UserCreate	int	Không	Khóa ngoại → USER.Id	Id người tạo văn bản
Text	nvarchar (max)	Không		Nội dung văn bản luyện tập
Title	nvarchar (255)	Không		Tiêu đề văn bản
TextLength	int	Không		Độ dài văn bản (số ký tự / từ)
PeoplesCompleted	int	Có		Số người đã hoàn thành văn bản (có thể 0)
CreatedAt	datetime	Không		Thời điểm tạo văn bản
LanguageID	int	Không	Khóa ngoại → EXERCISELANGUAGE.LanguageID	Ngôn ngữ của văn bản
JoinCode	nvarchar (50)	Có		Mã chia sẻ / mã bài (nếu có)
IsPrivate	bit	Không		1 = riêng tư, 0 = công khai
Rating	float	Có		Điểm trung bình đánh giá (nếu có)

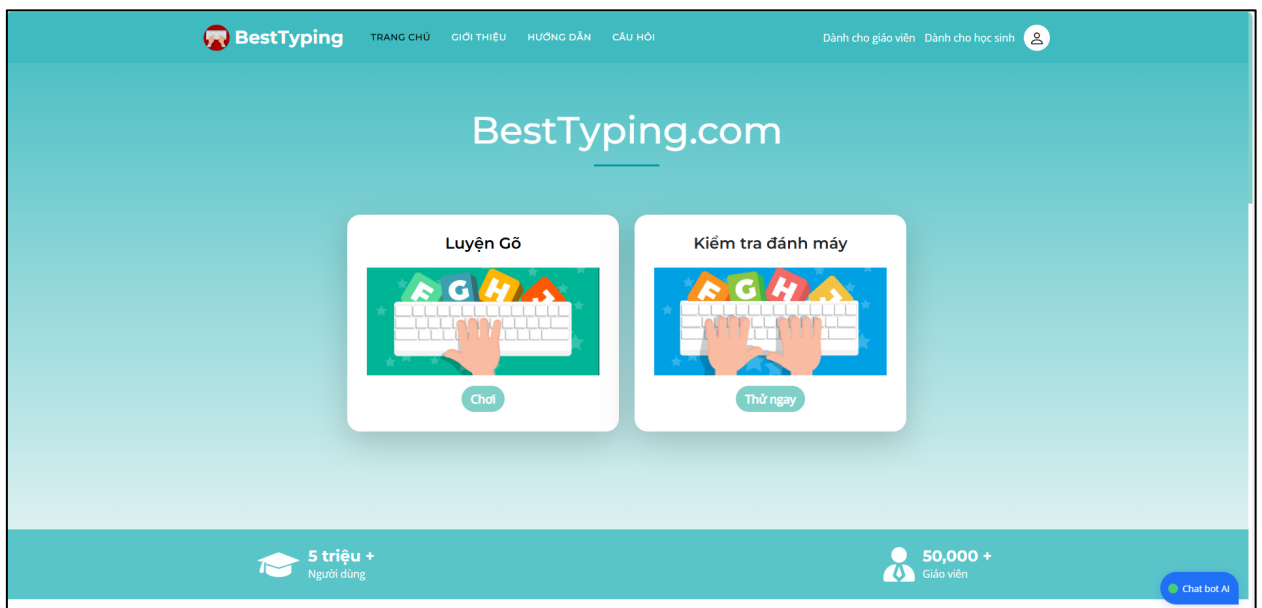
Tên trường	Kiểu dữ liệu	Null	Khóa	Mô tả
ListUserLike	nvarchar (max)	Có		(Hiện lưu tạm) danh sách id người like — khuyến nghị thay bằng bảng liên kết
ListUserRating	nvarchar (max)	Có		(Hiện lưu tạm) danh sách rating chi tiết — khuyến nghị thay bằng bảng liên kết
Status	int	Không		Trạng thái (ví dụ: 0 = draft, 1 = active, 2 = blocked)

2.6. Thiết kế giao diện

2.6.1. Giao diện của người dùng

- Màn hình giao diện chính người dùng ở trang chủ

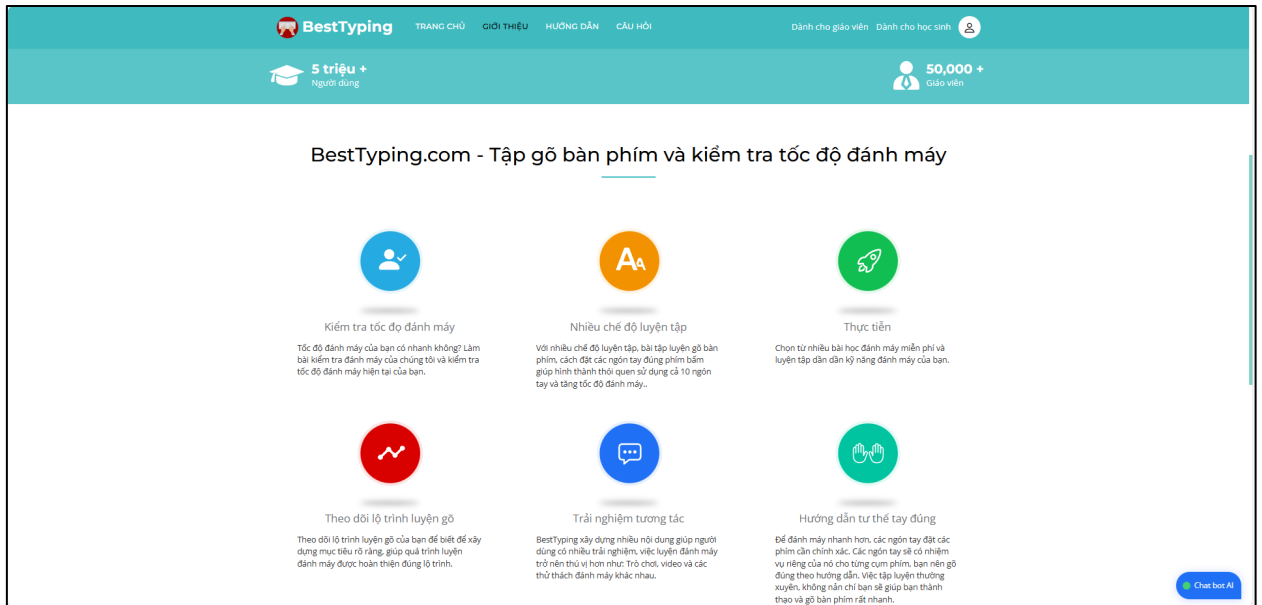
Giao diện trang chủ đóng vai trò là trung tâm điều hướng, giúp người dùng tiếp cận nhanh các chức năng cốt lõi. Giao diện còn tích hợp trợ lý ảo Chatbot ở góc phải để hỗ trợ giải đáp thắc mắc kịp thời cho người dùng.



Hình 2.44. Màn hình giao diện chính người dùng ở trang chủ

- Màn hình giới thiệu về website

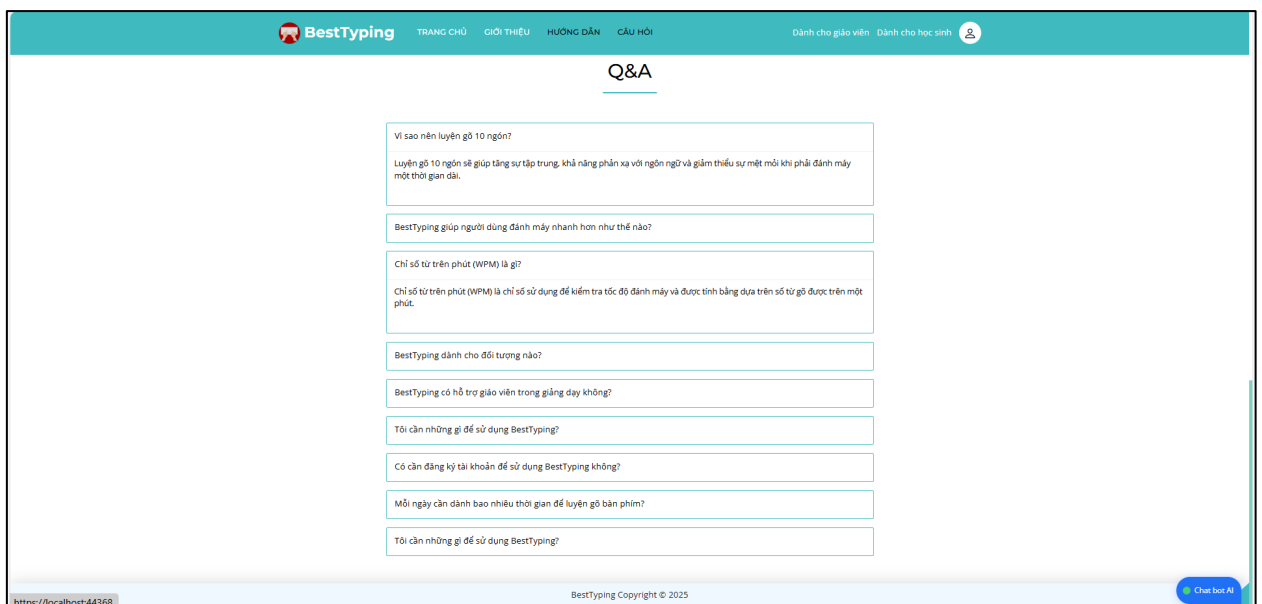
Giao diện này có chức năng giúp người dùng mới nhanh chóng nắm bắt được các công cụ hỗ trợ luyện tập và giá trị thực tiễn mà website mang lại trước khi bắt đầu sử dụng.



Hình 2.45. Màn hình giới thiệu về website

- Màn hình câu hỏi và trả lời chung

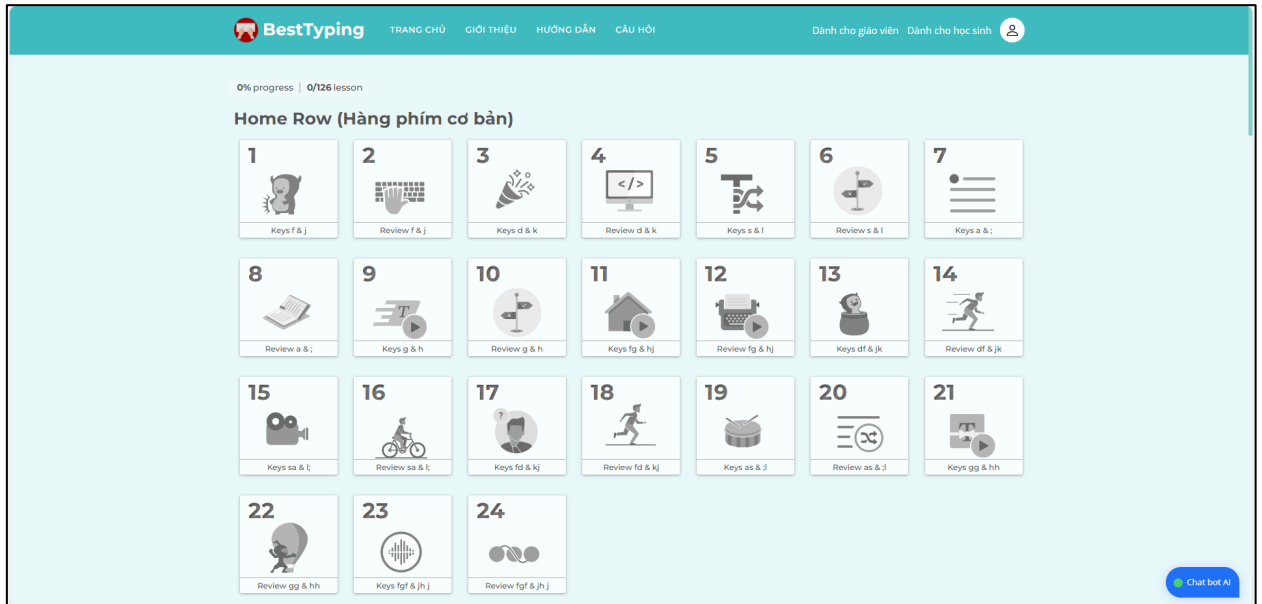
Website hiện các câu hỏi và trả lời chung để vừa mô tả vừa giới thiệu cho người sử dụng hiểu thêm về website.



Hình 2.46. Màn hình câu hỏi và trả lời chung

- Màn hình luyện gõ bàn phím cơ bản

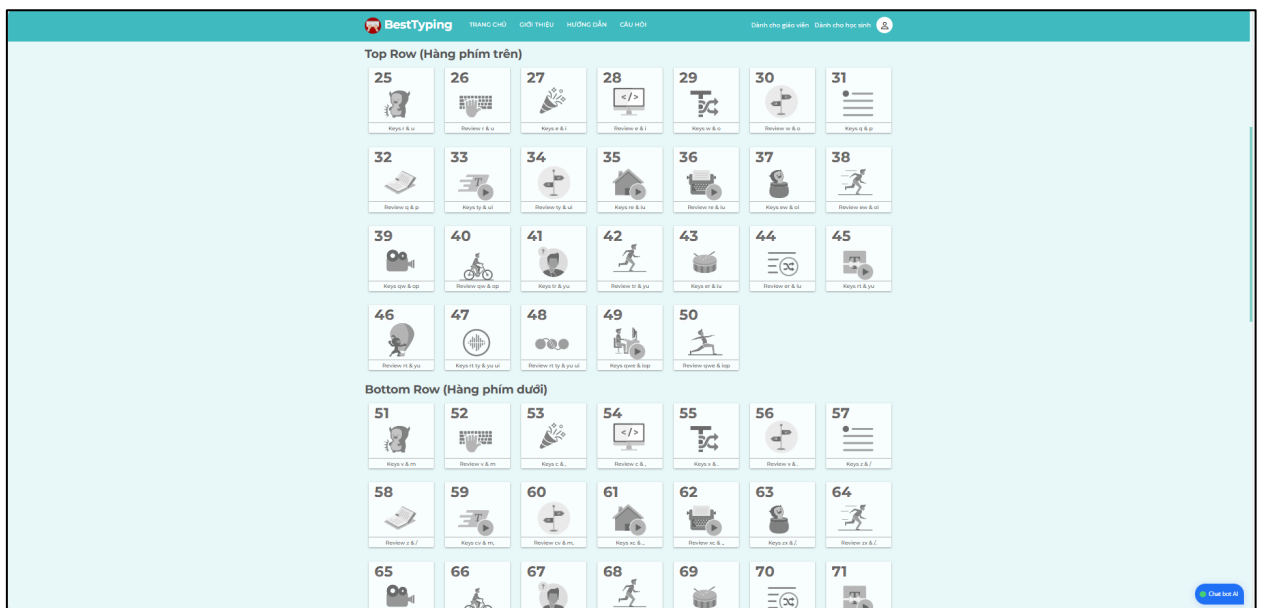
Đây là các bài luyện tập cơ bản cho người mới bắt đầu muốn luyện gõ bàn phím để người làm quen với các hàng phím và luyện gõ từng ngón từng chữ cái.



Hình 2.47. Màn hình luyện gõ bàn phím cơ bản

- Màn hình luyện gõ hàng phím trên và hàng phím dưới

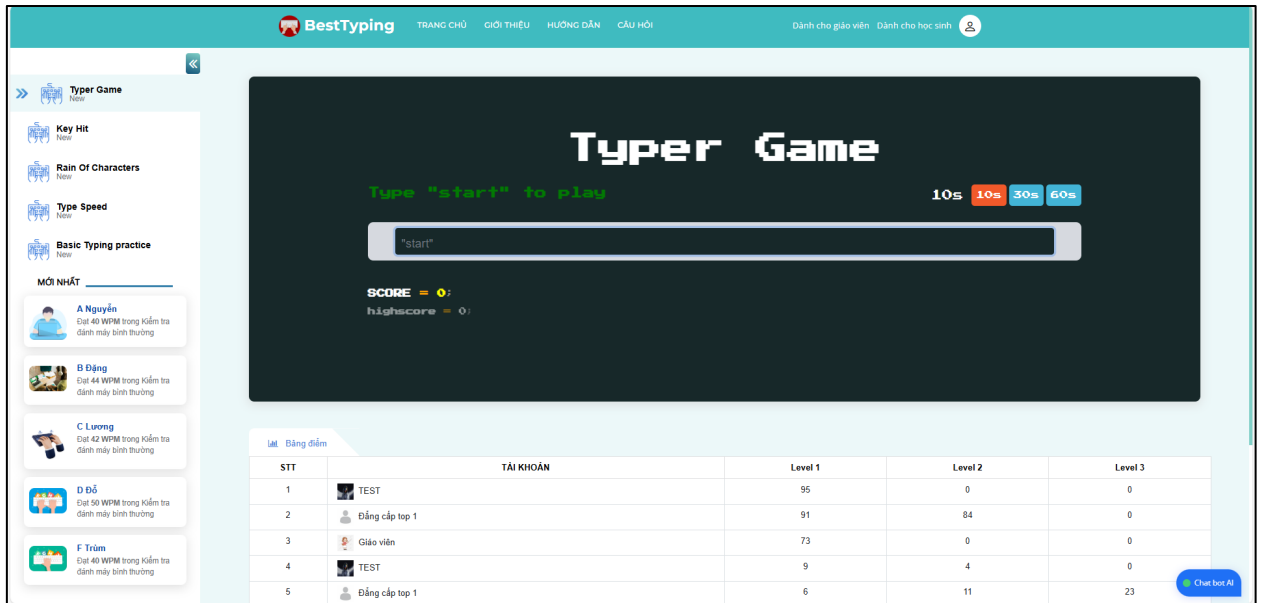
Đây là các bài luyện tập cơ bản cho người mới bắt đầu muốn luyện gõ bàn phím để người làm quen với các hàng phím trên và hàng phím dưới.



Hình 2.48. Màn hình luyện gõ hàng phím trên và hàng phím dưới

- Màn hình trò chơi gõ chữ nhanh đầu tiên

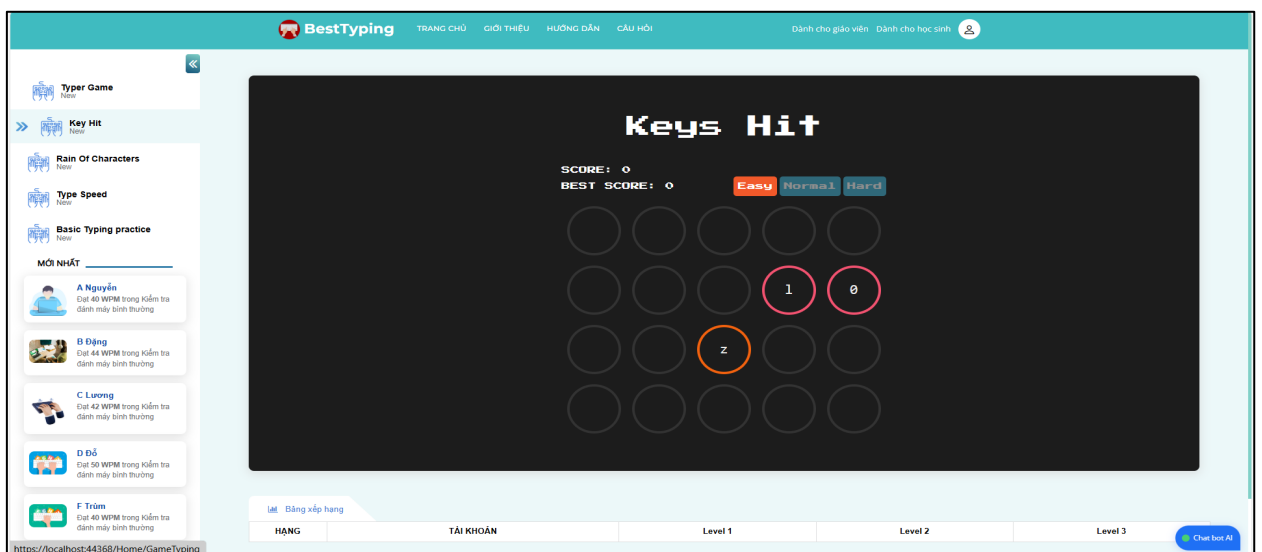
Trò chơi đầu tiên sẽ bắt đầu khi người dùng gõ đúng chữ “start”. Sau đó các chữ khác sẽ xuất hiện và khi người dùng gõ đúng sẽ hiện chữ khác. Ở bên dưới sẽ hiện bảng điểm xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.49. Màn hình trò chơi gõ chữ nhanh đầu tiên

- Màn hình trò chơi gõ chữ nhanh thứ 2

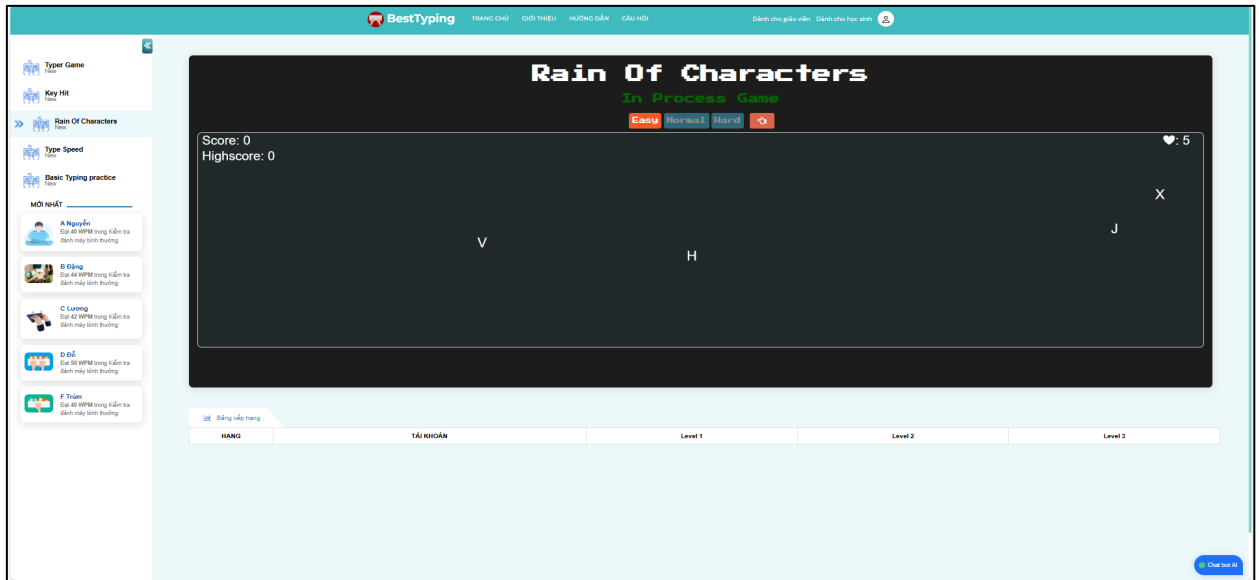
Trò chơi này giúp người dùng tăng tốc độ gõ bàn phím, khi mỗi chữ xuất hiện người dùng sẽ phải gõ đúng chữ đó để chữ đó mất đi sau đó sẽ hiện ra chữ mới khác trong thời gian quy định. Ở bên dưới sẽ hiện bảng điểm xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.50. Màn hình trò chơi gõ chữ nhanh thứ 2

- Màn hình trò chơi gõ chữ nhanh thứ 3

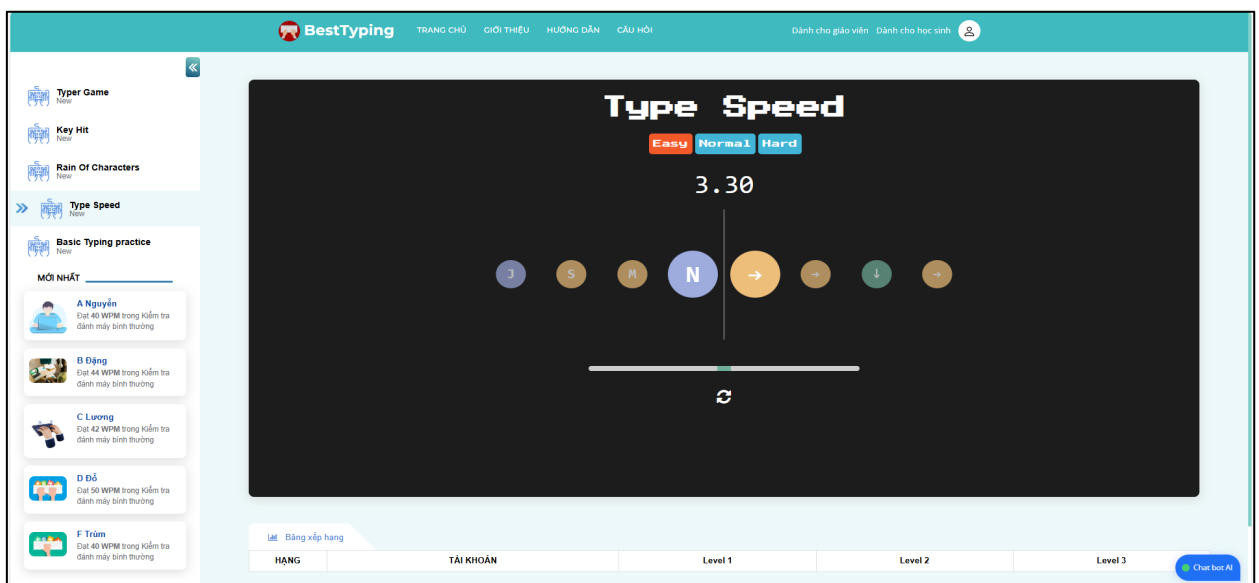
Mỗi chữ sẽ rơi ngẫu nhiên từ trên xuống người dùng sẽ phải gõ đúng chữ rơi để chữ đó biến mất đi và chữ khác sẽ rơi xuống tiếp, người dùng được chơi thua 5 lần rồi sẽ tính kết quả. Ở bên dưới sẽ hiện bảng điểm xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.51. Màn hình trò chơi gõ chữ nhanh thứ 3

- Màn hình trò chơi gõ chữ nhanh thứ 4

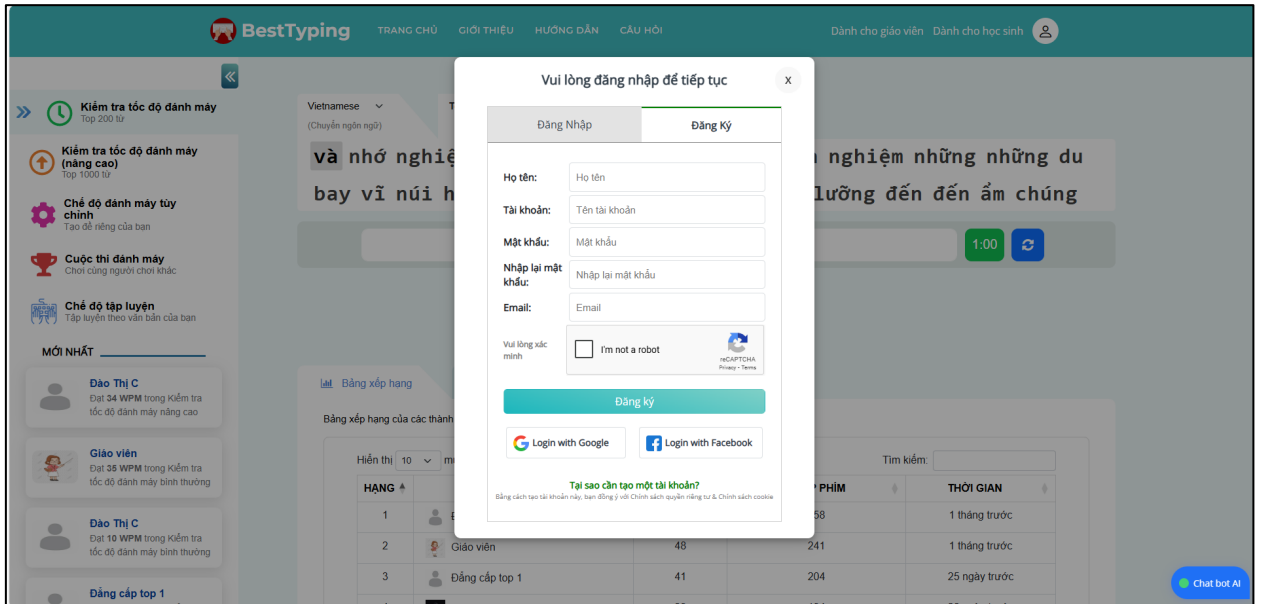
Trong thời gian quy định người dùng sẽ gõ các phím hiện hiện màn hình để luyện gõ nhanh. Ở bên dưới sẽ hiện bảng điểm xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.52. Màn hình trò chơi gõ chữ nhanh thứ 4

- Màn hình giao diện đăng kí của người dùng

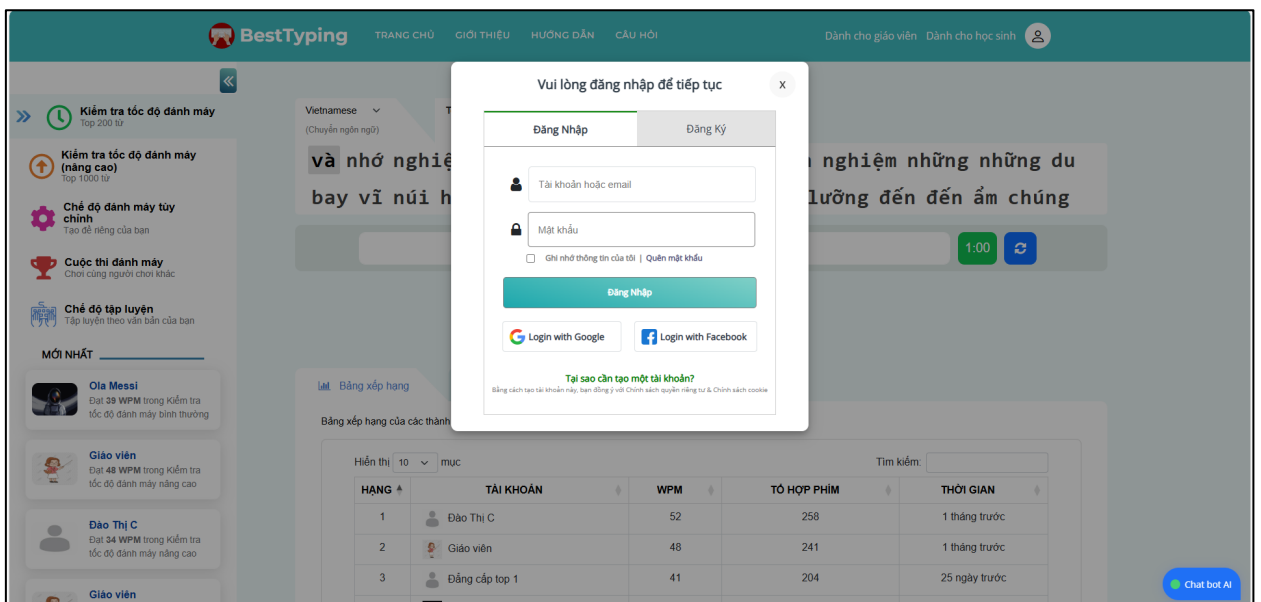
Trang giao diện đăng kí cho phép người dùng đăng kí tài khoản để sử dụng các chức năng mở rộng của website.



Hình 2.53. Màn hình giao diện đăng kí của người dùng

- Màn hình giao diện đăng nhập của người dùng

Trang giao diện đăng nhập cho phép người dùng đã đăng kí tài khoản có thể đăng nhập vào website.



Hình 2.54. Màn hình giao diện đăng nhập của người dùng

- Màn hình kiểm tra tốc độ luyện gõ bàn phím

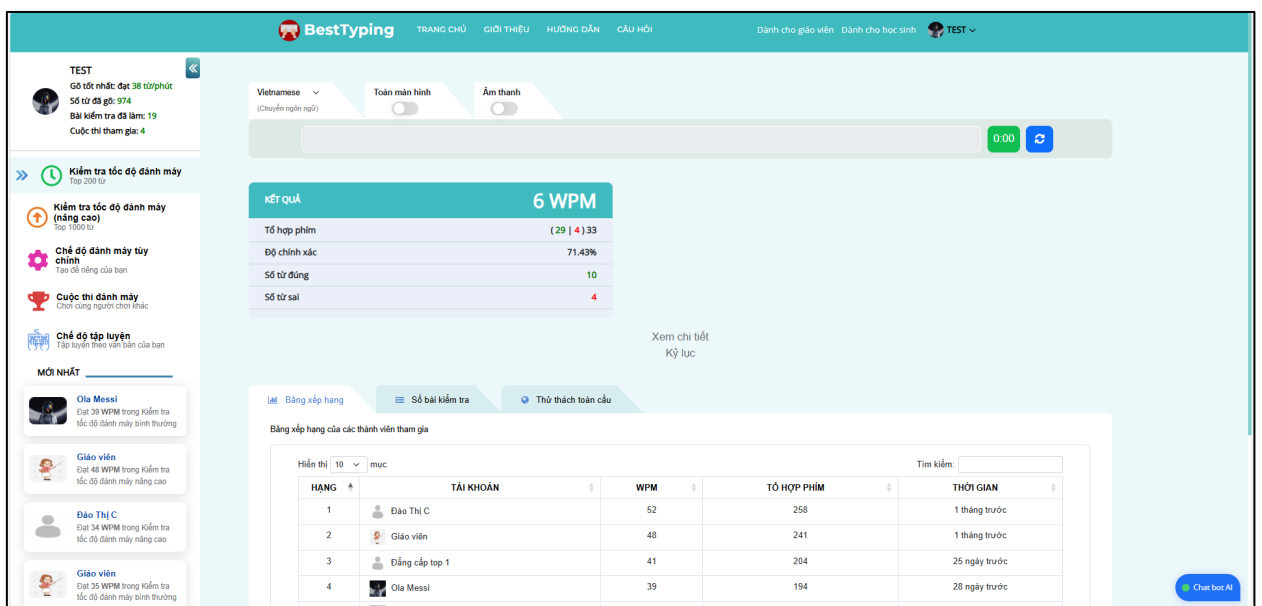
Trang này cho phép người dùng muốn kiểm tra tốc độ, độ chính xác khi gõ bàn phím của mình. Ở bên dưới sẽ hiện bảng xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.55. Màn hình kiểm tra tốc độ luyện gõ bàn phím

- Màn hình kết quả kiểm tra tốc độ luyện gõ bàn phím

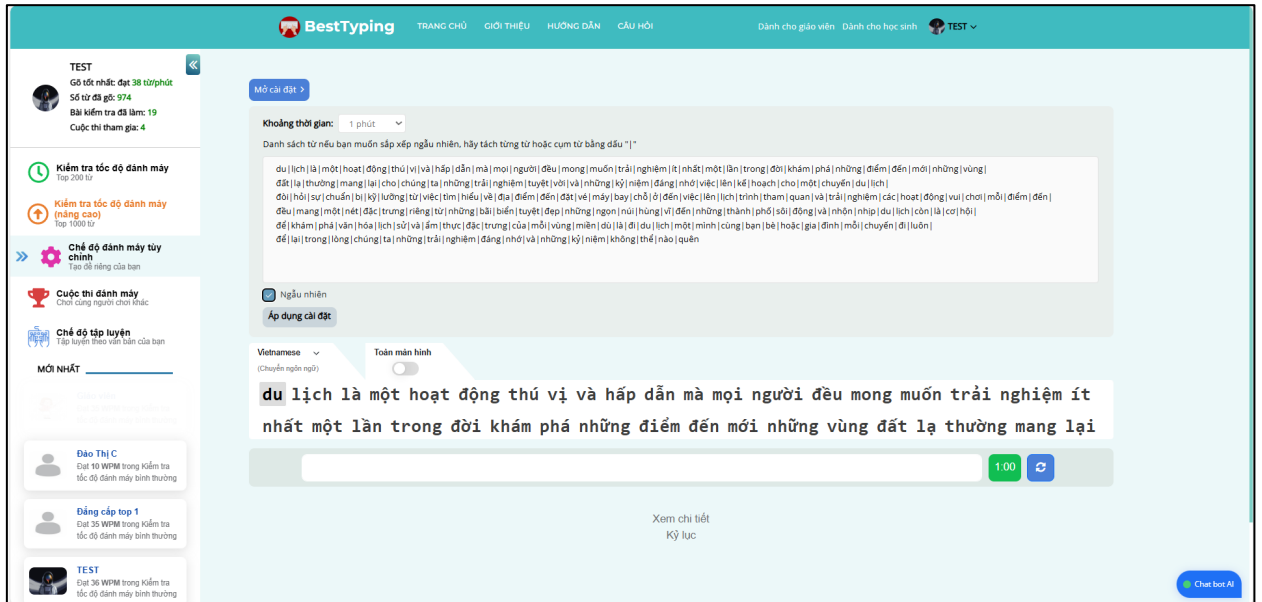
Sau khi người dùng sử dụng chức năng kiểm tra tốc độ, độ chính xác luyện gõ bàn phím xong bảng kết quả của người dùng sẽ hiện ra bên dưới. Ở bên dưới sẽ hiện bảng xếp hạng người chơi theo thứ tự từ cao đến thấp từ trên xuống.



Hình 2.56. Màn hình kết quả kiểm tra luyện gõ bàn phím

- Màn hình chế độ đánh máy tùy chỉnh

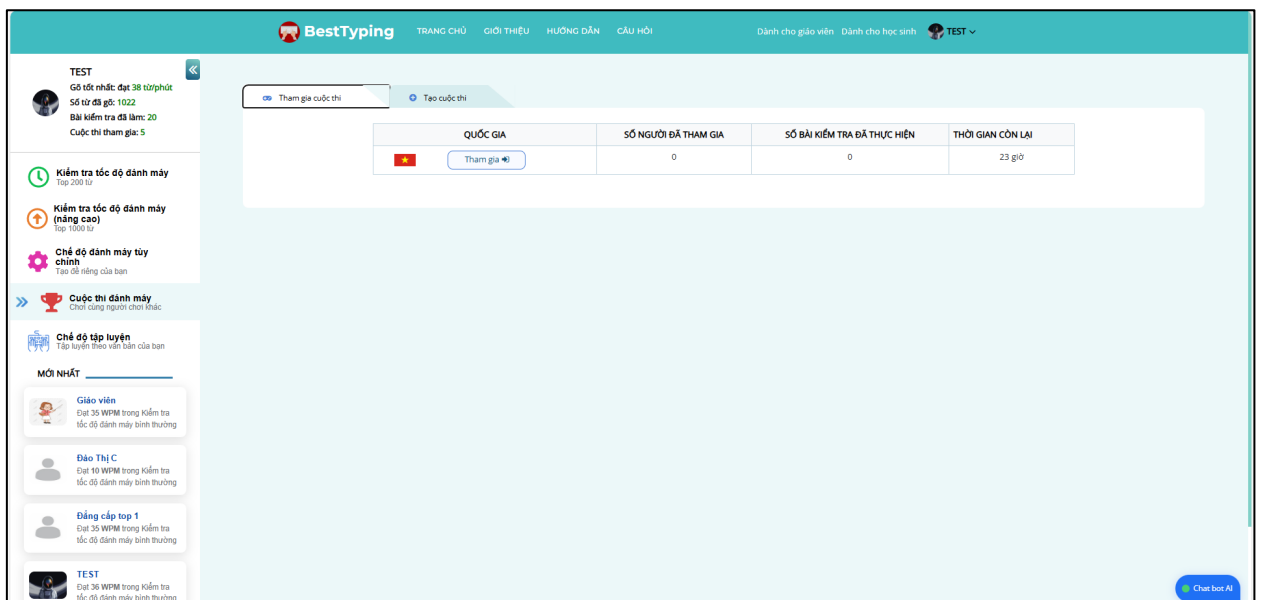
Người dùng có thể tạo trực tiếp trong cài đặt, có thể chọn ngạo trộn từ trong đoạn văn thành ngẫu nhiên để đánh máy tốt hơn.



Hình 2.57. Màn hình kết quả kiểm tra luyện gõ bàn phím

- Màn hình tham gia cuộc thi đánh máy công khai

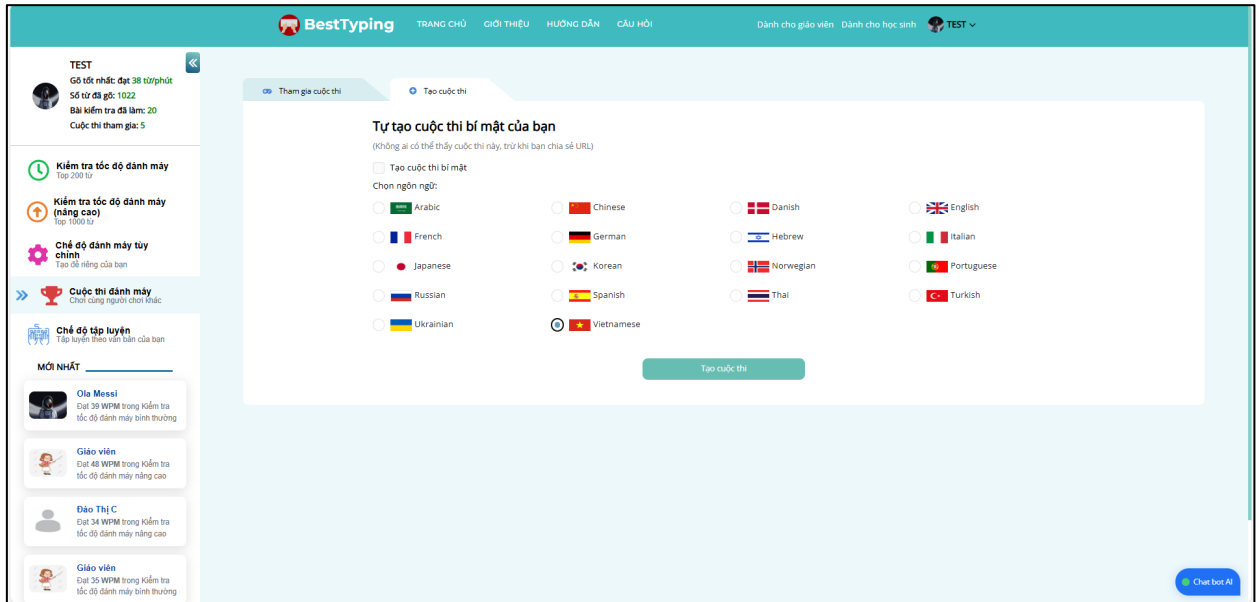
Người dùng có thể tham gia các cuộc thi đánh máy đã được tạo trước đó chung cho tất cả mọi người sử dụng website.



Hình 2.58. Màn hình tham gia cuộc thi đánh máy

- Màn hình tạo cuộc thi đánh máy công khai

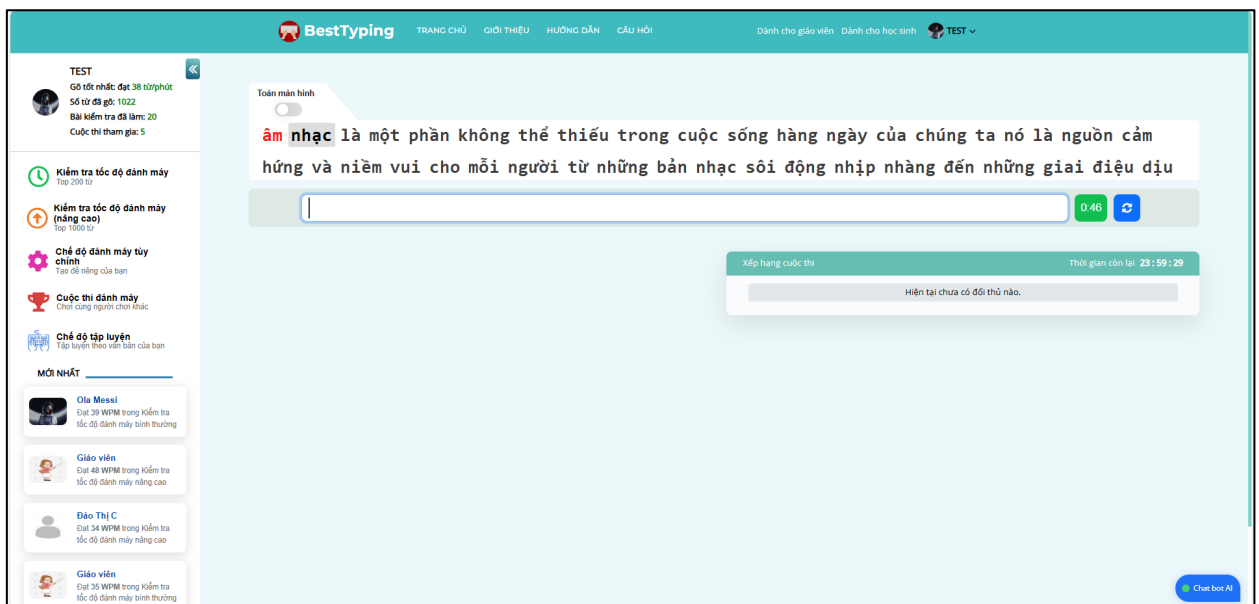
Người dùng tự tạo cuộc thi đánh máy chung cho tất cả người dùng khi sử dụng website



Hình 2.59. Màn hình tạo cuộc thi đánh máy công khai

- Màn hình tạo cuộc thi đánh máy đánh máy bí mật

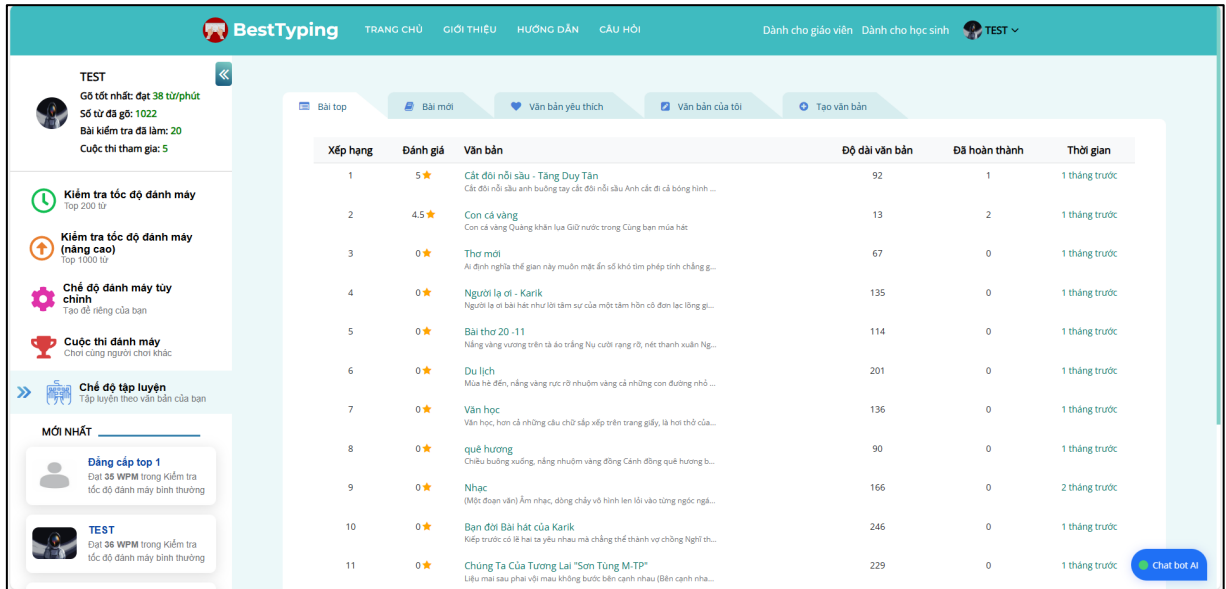
Người dùng có thể tạo cuộc thi đánh máy bí mật để sử dụng, người khác không thể tham gia vào.



Hình 2.60. Màn hình tạo cuộc thi đánh máy bí mật

- Màn hình các chế độ luyện tập

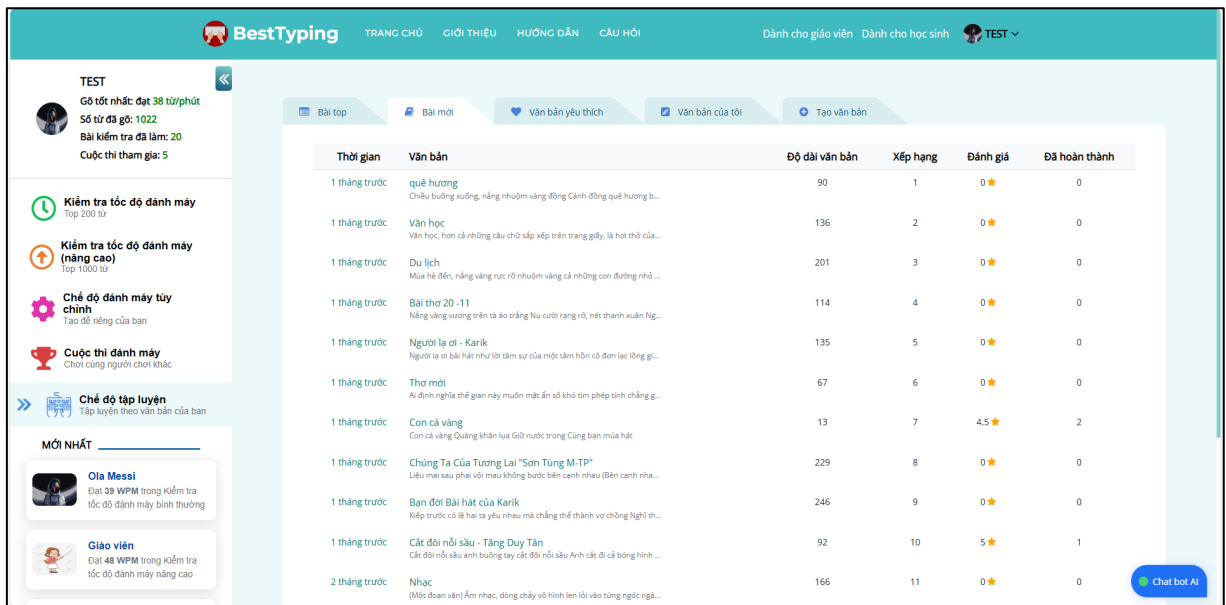
Các chế độ luyện tập đa dạng gồm các bài top, bài mới, văn bản yêu thích, văn bản của người dùng, tạo văn bản.



Hình 2.61. Màn hình các chế độ luyện tập

- Màn hình các văn bản luyện tập mới

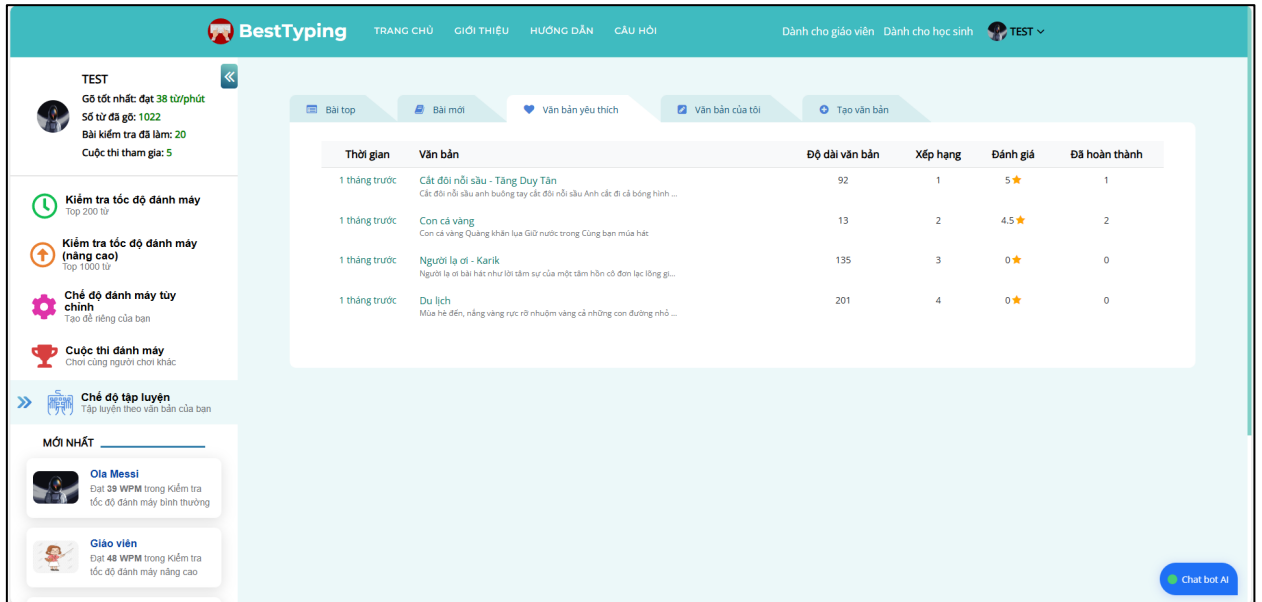
Người dùng có thể chọn các văn bản trong danh sách để luyện gõ, văn bản hệ thống



Hình 2.62. Màn hình các văn bản luyện tập mới

- Màn hình các văn bản luyện tập yêu thích

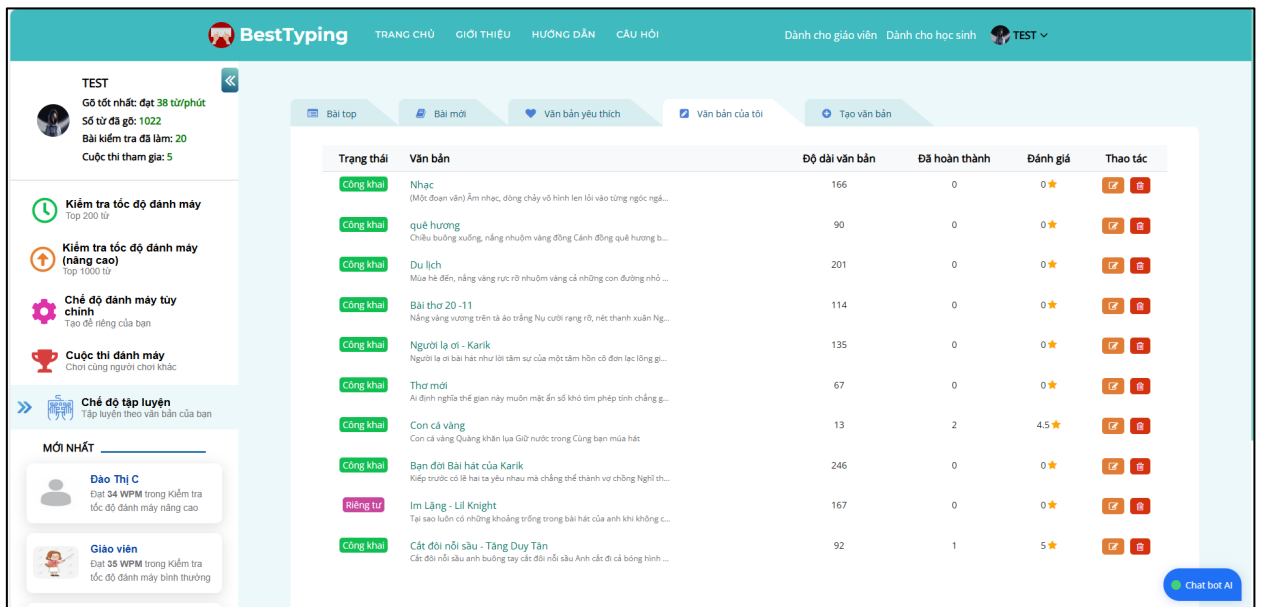
Người dùng có thể thêm các văn bản luyện gõ vào danh sách yêu thích của riêng mình.



Hình 2.63. Màn hình các văn bản luyện tập yêu thích

- Màn hình các văn bản luyện tập đã tạo

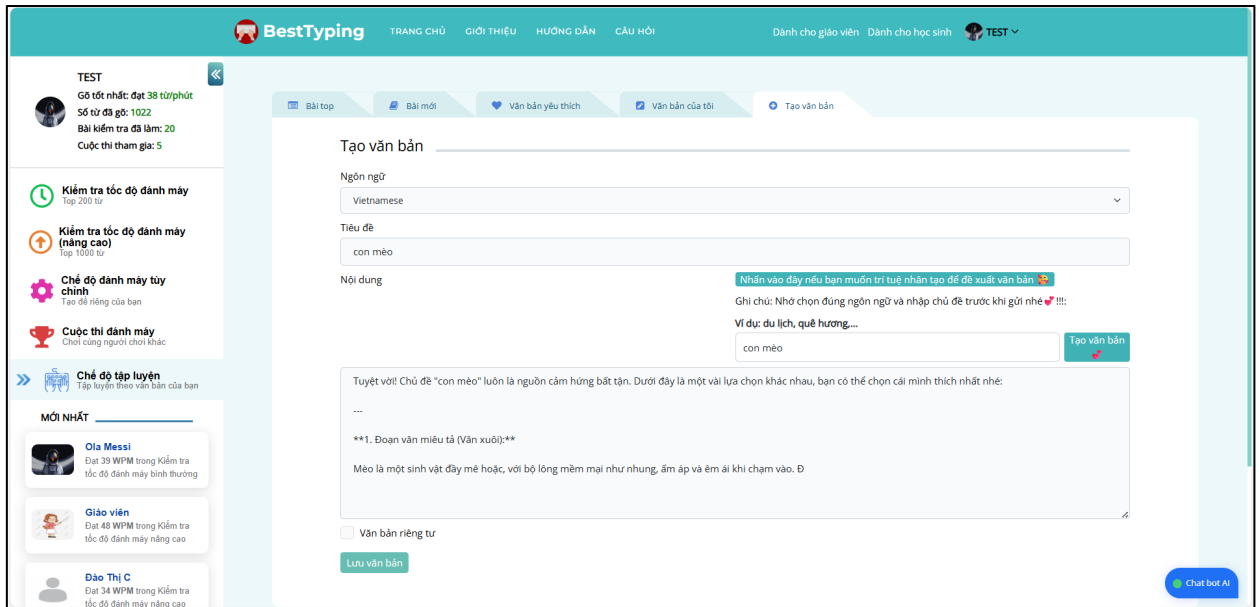
Người dùng có thể xem, chỉnh sửa, xóa các văn bản do mình tạo ra.



Hình 2.64. Màn hình các văn bản luyện tập đã tạo

- Màn hình tạo văn bản luyện tập

Người dùng có thể tự tạo văn bản cho riêng mình, có thể dùng trí tuệ nhân tạo để tự tạo văn bản khi nhập chủ đề, người dùng có thể chia sẻ văn bản mình đã tạo hoặc chọn văn bản riêng tư để sử dụng riêng tư.

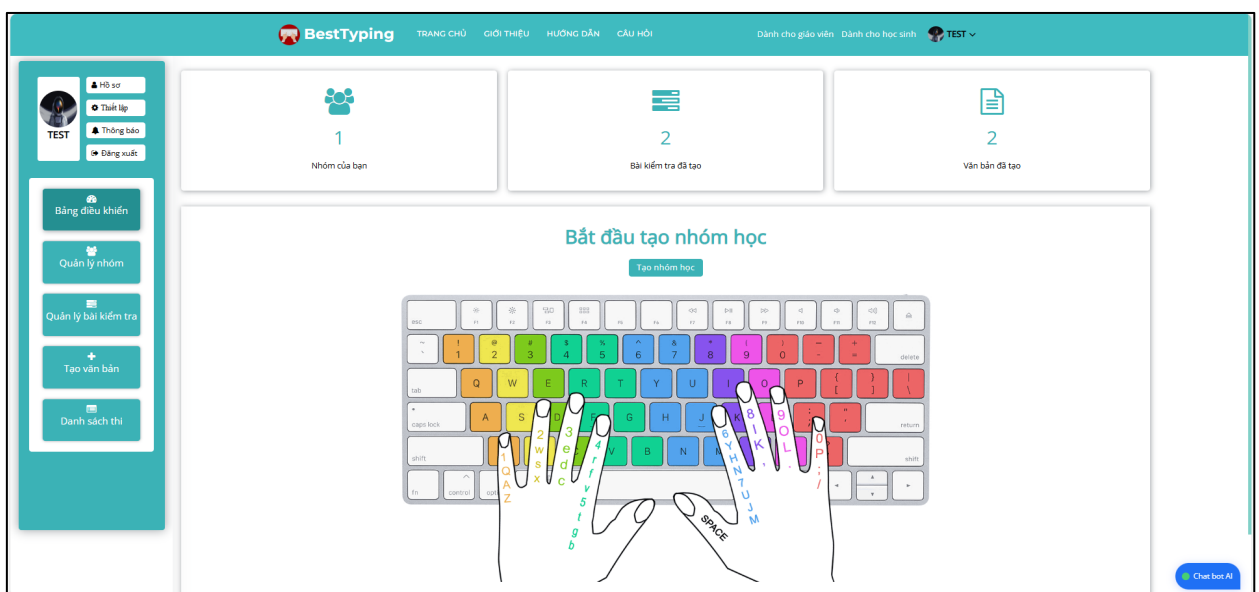


Hình 2.65. Màn hình tạo văn bản luyện tập

2.6.2. Giao diện của giáo viên

- Màn hình chính giao diện của giáo viên

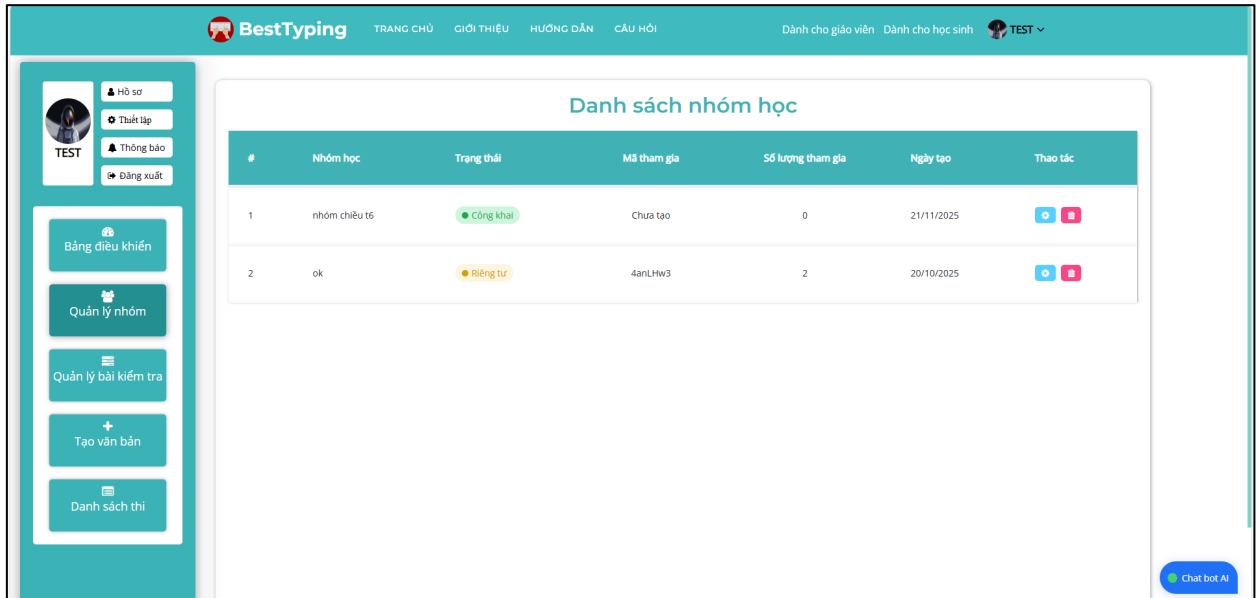
Trang giao diện chính của giáo viên có thể xem tổng số nhóm, các bài kiểm tra và văn bản đã tạo, và tạo nhóm học.



Hình 2.66. Màn hình chính giao diện của giáo viên

- Màn hình danh sách các lớp học của giáo viên

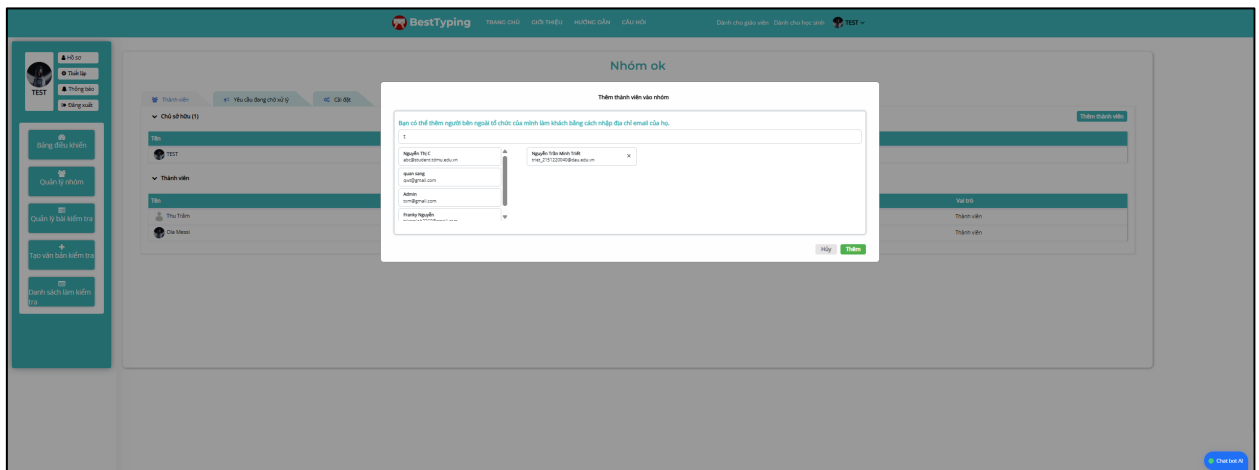
Trang quản lý danh sách các lớp học để giáo viên có thể xem, xóa lớp học của mình đã tạo.



Hình 2.67. Màn hình danh sách lớp học của giáo viên

- Màn hình mời học sinh vào lớp bằng Email

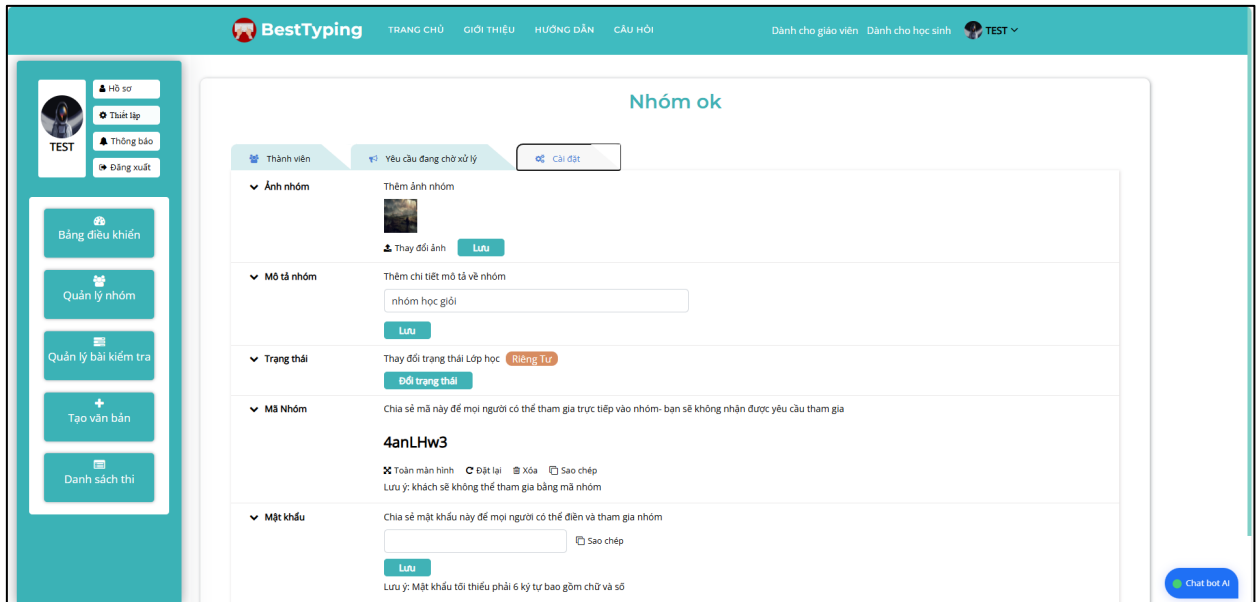
Giáo viên có thể gửi lời mời qua Email để mời học sinh vào lớp học của mình



Hình 2.68. Màn hình mời học sinh vào lớp bằng Email

- Màn hình quản lý lớp học của giáo viên

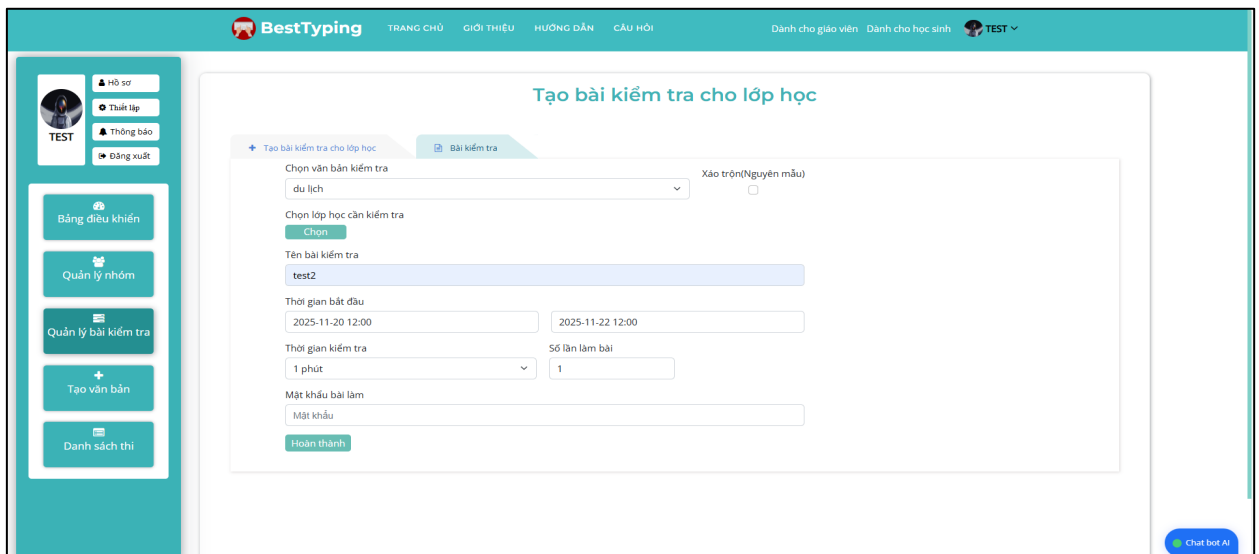
Giáo viên có thể tùy chỉnh cài đặt lớp học của mình tạo ra, xét duyệt các yêu cầu tham gia lớp học, xem danh sách các thành viên trong lớp học, sao chép mã lớp để gửi cho học sinh hoặc mời học sinh tham gia vào lớp thông qua mã lớp, gửi lời mời qua Email.



Hình 2.69. Màn hình quản lý lớp học của giáo viên

- Màn hình tạo bài kiểm tra cho lớp học của giáo viên

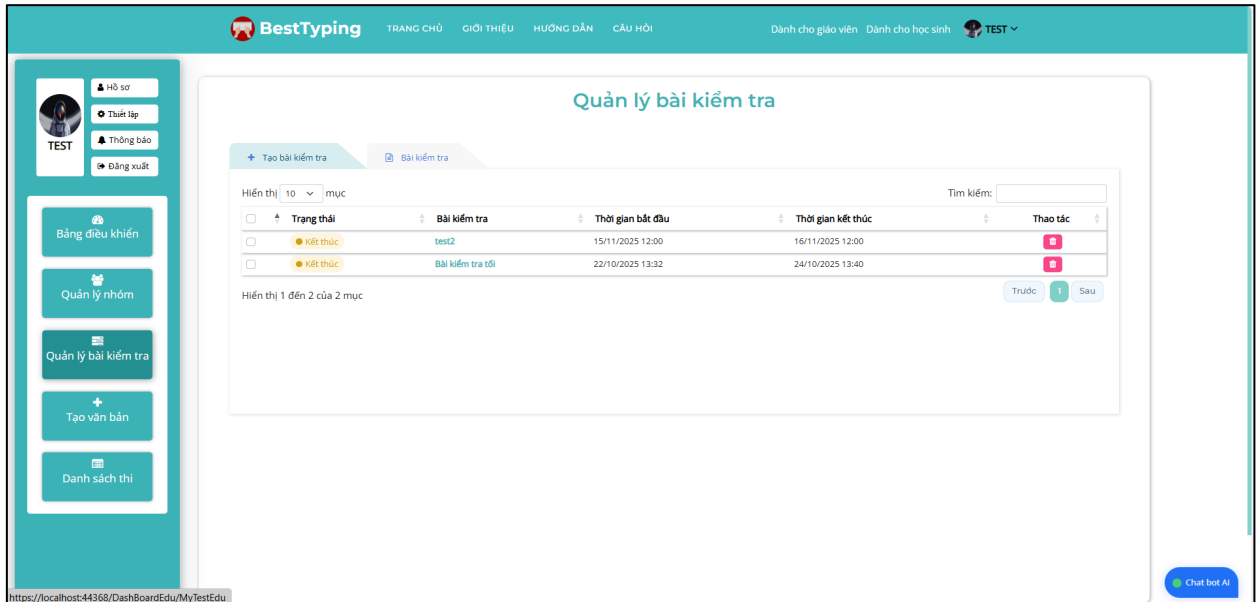
Giáo viên có thể tạo bài kiểm tra cho lớp học bằng cách chọn văn bản kiểm tra đã tạo trước đó từ phân tạo văn bản kiểm dùng để kiểm tra, chọn lớp học cần kiểm tra, đặt tên cho bài kiểm tra, chọn thời gian và đặt mật khẩu cho bài kiểm tra.



Hình 2.70. Màn hình tạo bài kiểm tra cho lớp học của giáo viên

- Màn hình quản lý các bài kiểm tra của giáo viên

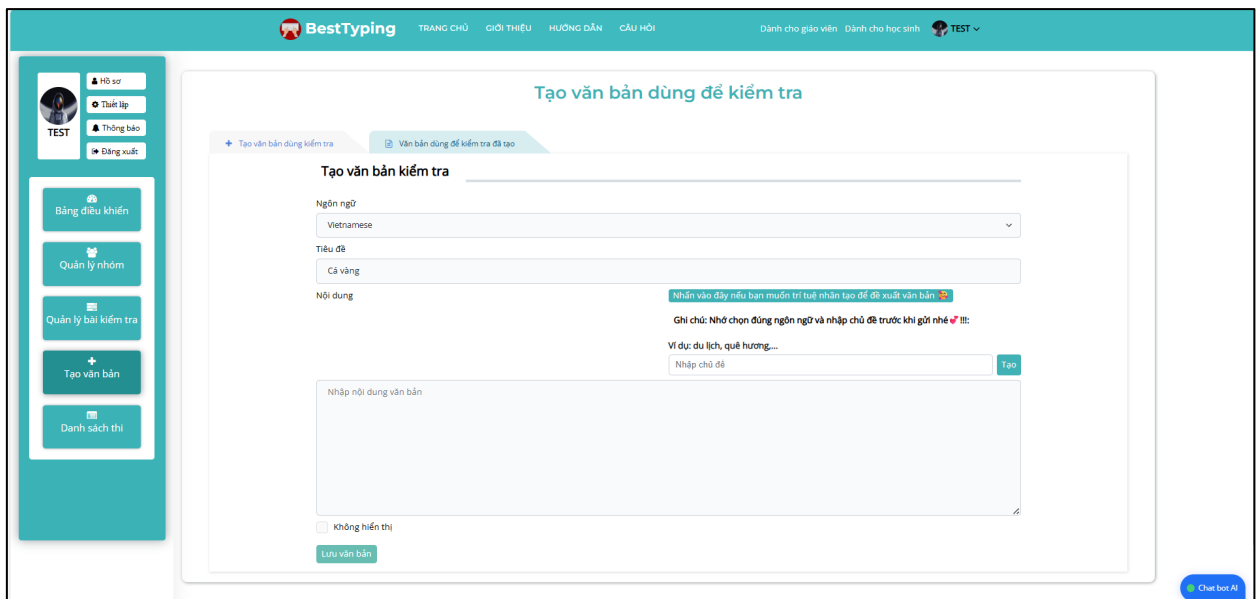
Các bài kiểm tra do giáo viên tạo sẽ hiện ở trang quản lý các bài kiểm tra, có thể xóa bài kiểm tra đã tạo.



Hình 2.71. Màn hình quản lý các bài kiểm tra của giáo viên

- Màn hình tạo văn bản kiểm tra của giáo viên

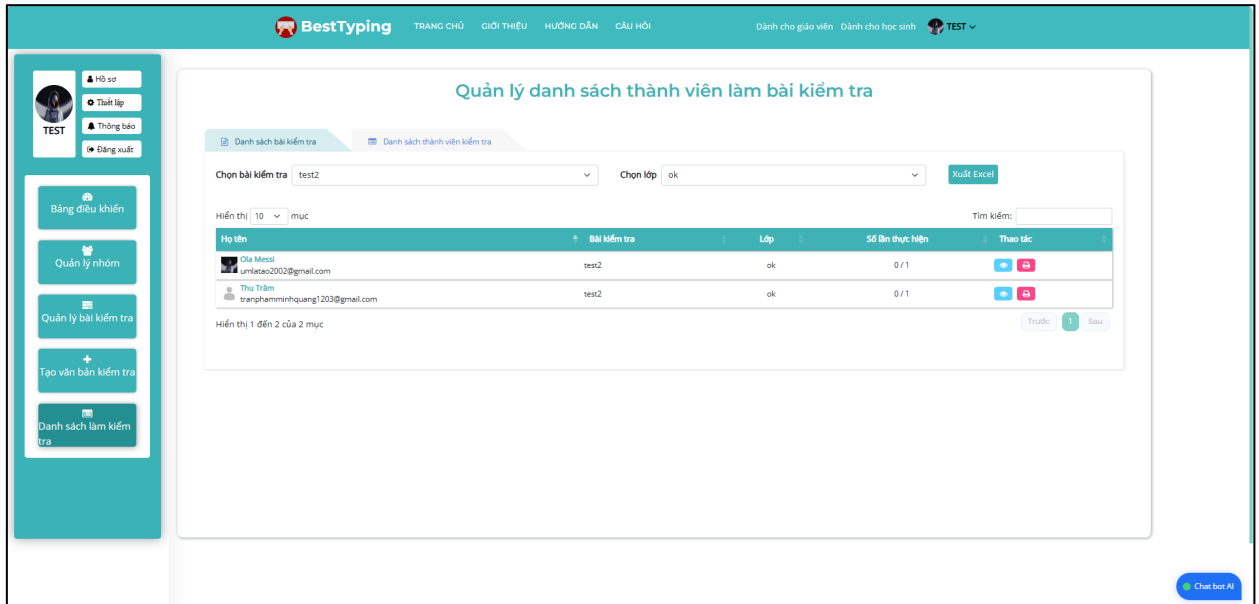
Chức năng tạo văn bản dùng để kiểm tra giúp giáo viên có thể tự tạo văn bản để sử dụng trong các bài kiểm tra, có thể nhập chủ đề để trí tuệ nhân tạo tự tạo văn bản theo chủ đề đã nhập.



Hình 2.72. Màn hình tạo văn bản kiểm tra của giáo viên

- Màn hình quản lý danh sách thành viên làm kiểm tra

Giáo viên có thể chọn lớp để xem danh sách thành viên làm bài kiểm tra, xuất ra file Excel để quản lý.

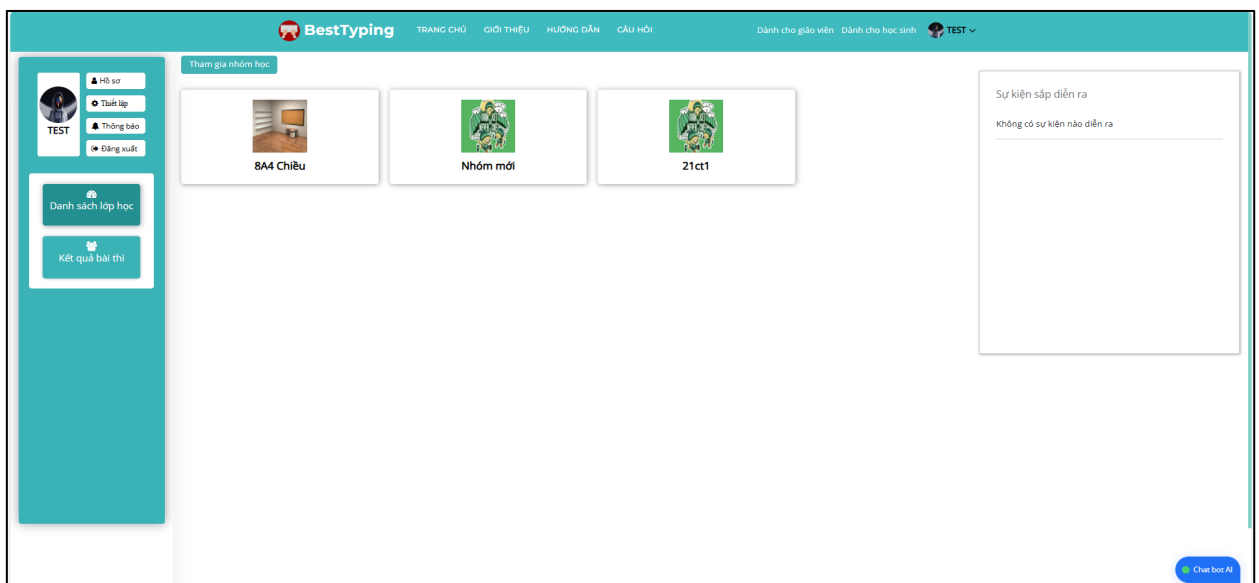


Hình 2.73. Màn hình quản lý danh sách thành viên làm kiểm tra

2.6.3. Giao diện của học sinh

- Màn hình chính giao diện của học sinh

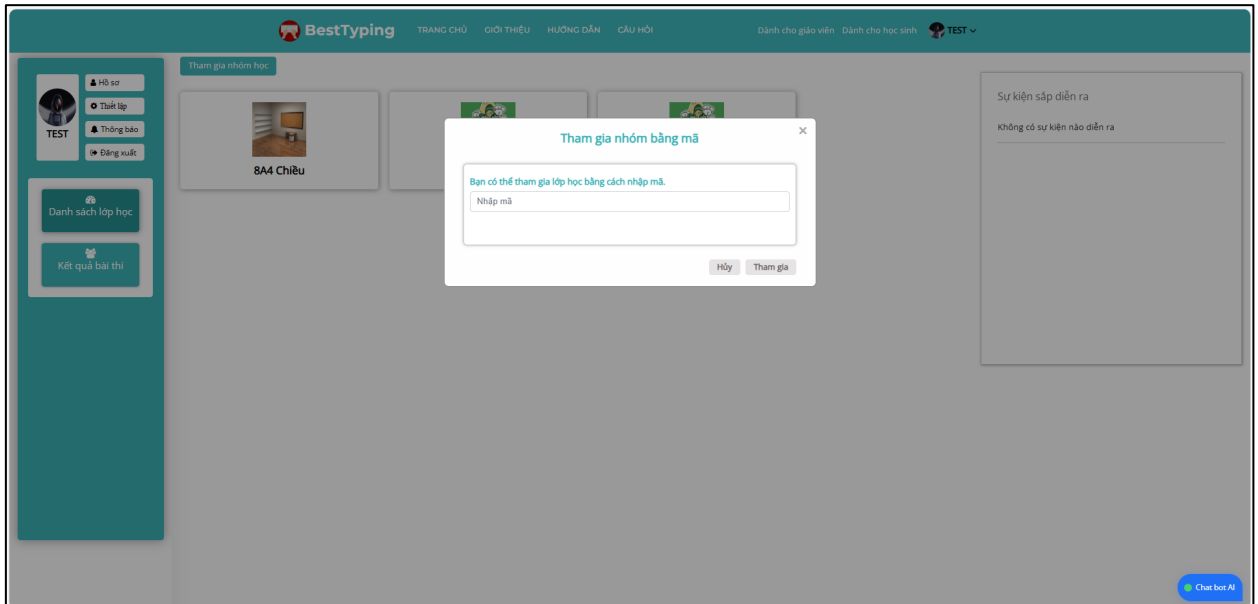
Giao diện chính trang chủ học sinh giúp học sinh có thể xem được các lớp mà mình đã tham gia, xem các sự kiện sắp diễn ra, vào tham gia lớp học bằng cách nhập mã lớp do giáo viên cung cấp



Hình 2.74. Màn hình chính giao diện của học sinh

- Màn hình tham gia lớp học bằng mã lớp

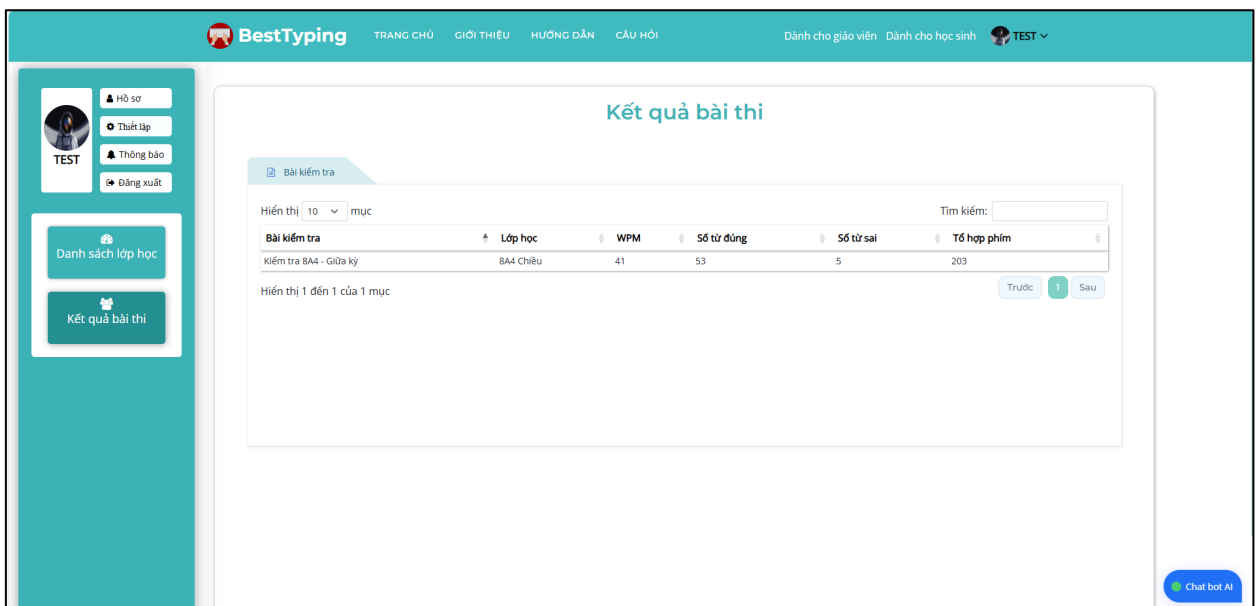
Học sinh có thể chọn chức năng tham gia vào lớp học bằng cách nhập đúng mã lớp mà giáo viên cung cấp để tham gia vào đúng lớp học



Hình 2.75. Màn hình tham gia lớp học bằng mã lớp

- Màn hình kết quả bài thi của học sinh

Học sinh sau khi làm bài kiểm tra do giáo viên tạo xong có thể kiểm tra kết quả ngay sau đó, về WPM, số từ đúng, số từ sai, tổ hợp phím, tên bài kiểm tra, tên lớp.

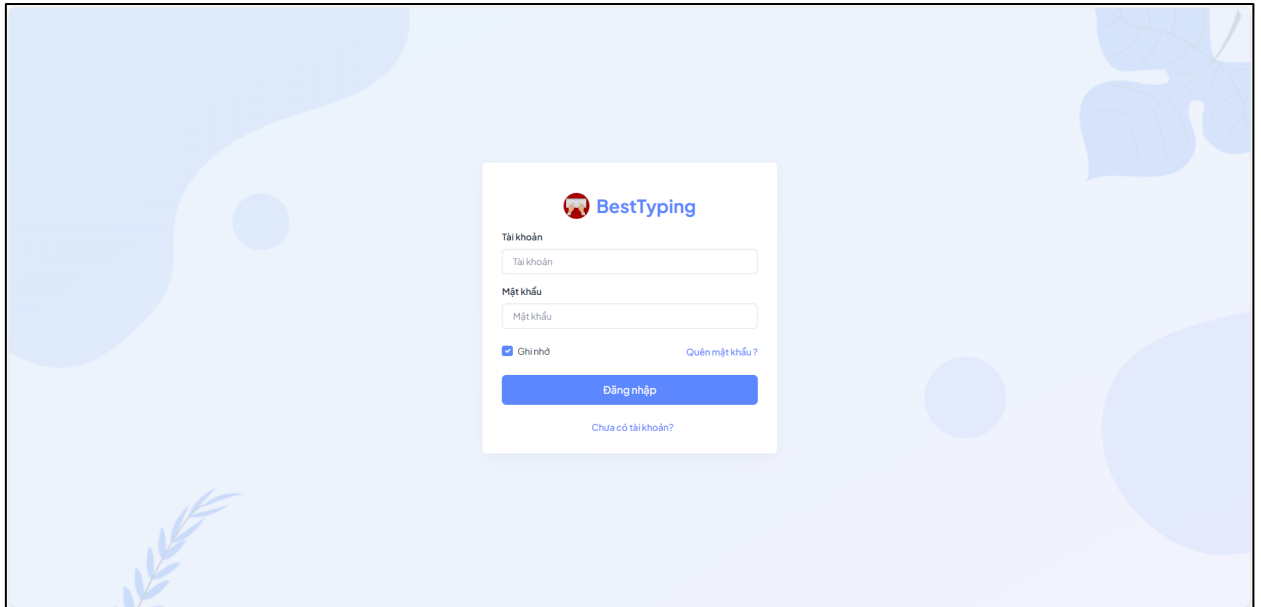


Hình 2.76. Màn hình kết quả bài thi của học sinh

2.6.4. Giao diện của quản trị viên

- Màn hình trang đăng nhập của quản trị viên

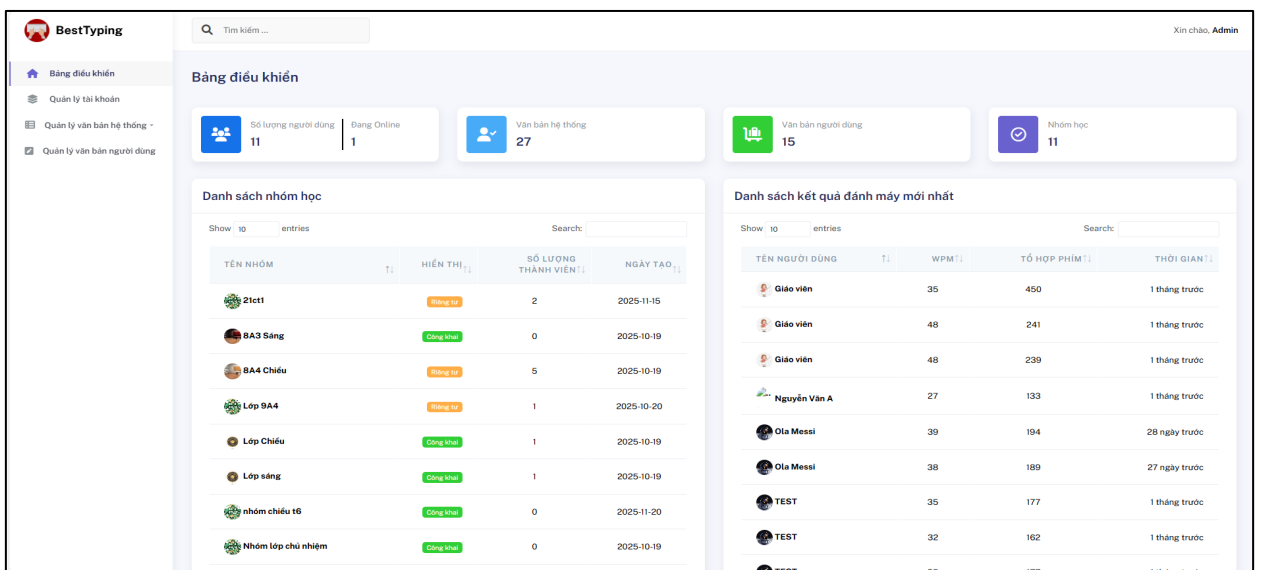
Quản trị viên đăng nhập dùng tài khoản và mật khẩu đăng nhập qua đường liên kết: <https://localhost:44368/admin/auth> để vào trang quản lý dành cho quản trị viên.



Hình 2.77. Màn hình trang đăng nhập của quản trị viên

- Màn hình chính giao diện của quản trị viên

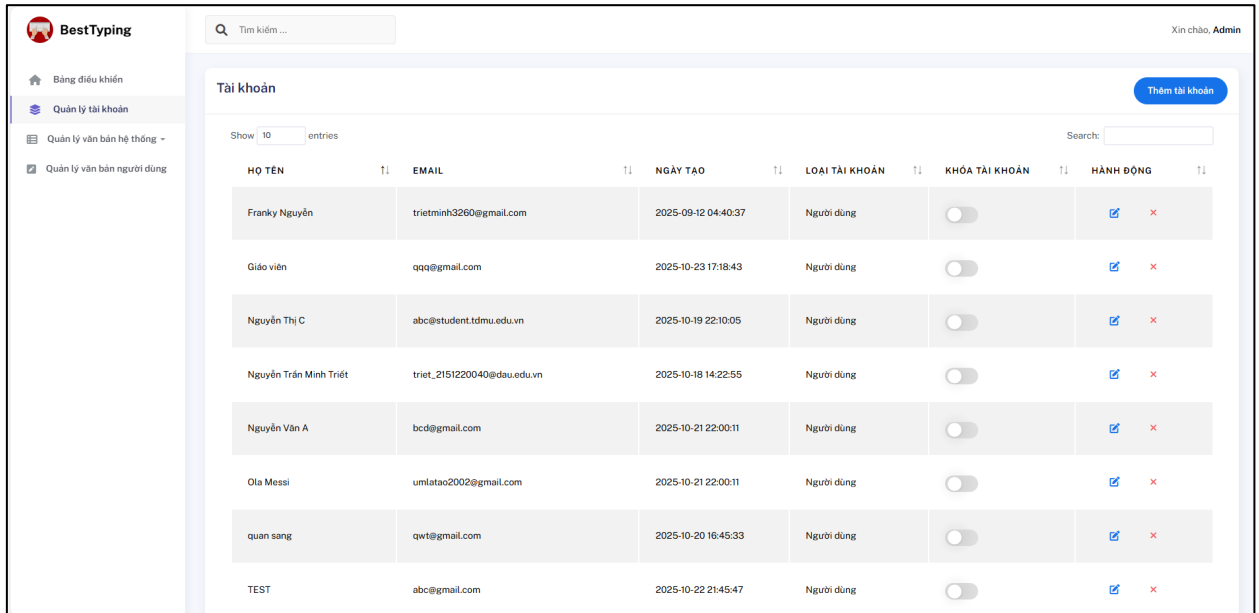
Ở màn hình chính giao diện của quản trị viên có thể xem bảng điều khiển chính, trang quản lý tài khoản, trang quản lý văn bản của hệ thống, trang quản lý văn bản do người dùng tạo.



Hình 2.78. Màn hình chính giao diện của quản trị viên

- Màn hình quản lý tài khoản của quản trị viên

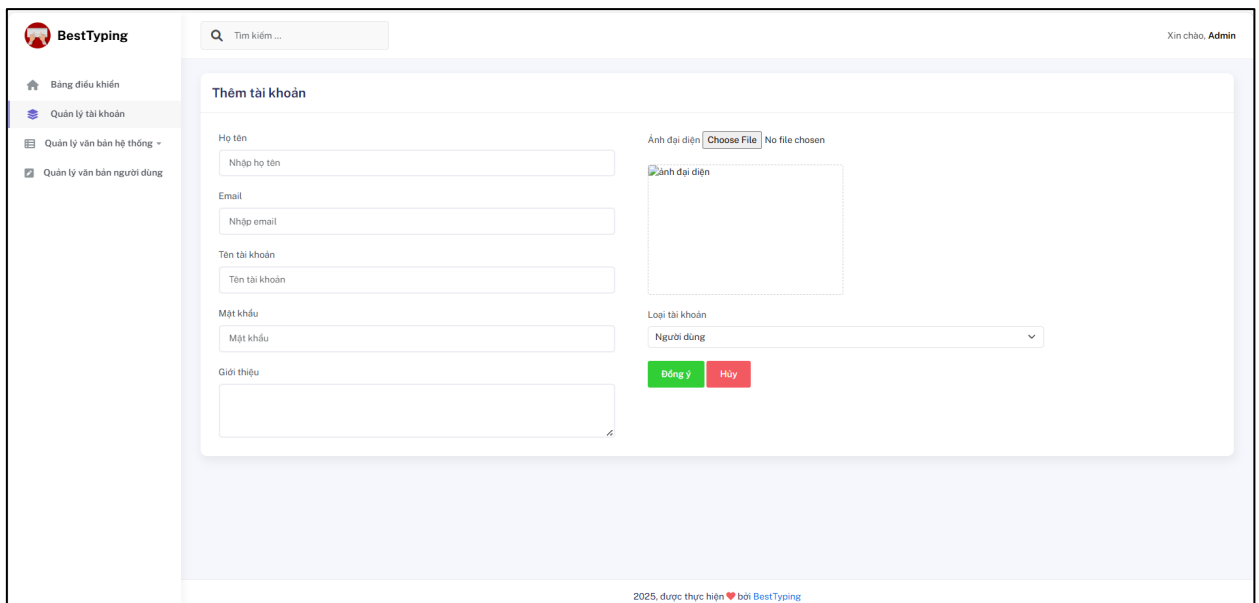
Quản trị viên có thể xem thêm mới, cập nhập, sửa, xóa tài khoản và dùng tính năng khóa tài khoản của người sử dụng ở màn hình quản lý tài khoản.



Hình 2.79. Màn hình quản lý tài khoản của quản trị viên

- Màn hình thêm mới tài khoản của quản trị viên

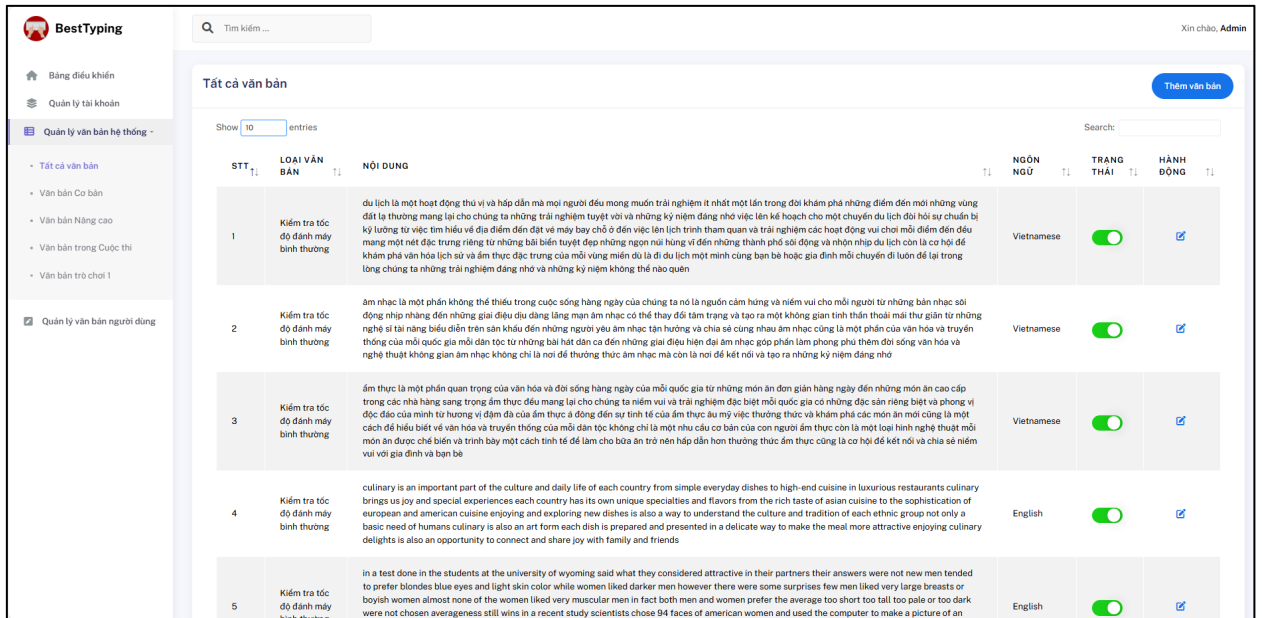
Quản trị viên có thể thêm mới tài khoản người dùng bao gồm đầy đủ thông tin cá nhân của người dùng: họ tên, Email, tên tài khoản, mật khẩu, ảnh đại diện, loại tài khoản, giới thiệu.



Hình 2.80. Màn hình thêm mới tài khoản của quản trị viên

- Màn hình quản lý văn bản hệ thống

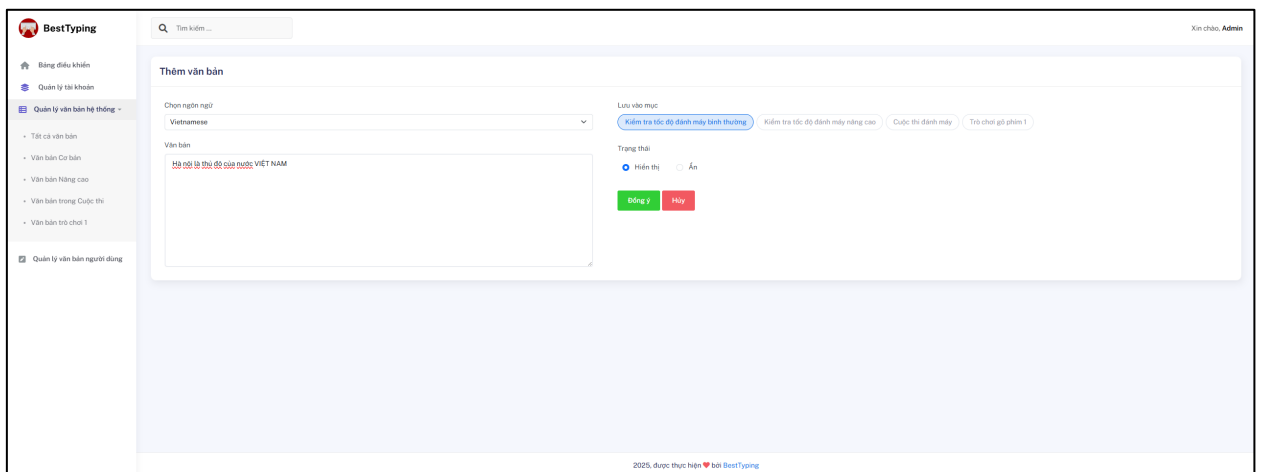
Tất cả các văn bản của hệ thống sẽ được hiện ở màn hình quản lý văn bản hệ thống của quản trị viên bao gồm: văn bản cơ bản, văn bản nâng cao, văn bản dùng trong cuộc thi chung tranh bảng xếp hạng của người dùng, văn bản trong trò chơi, quản trị viên có quyền thêm mới, chỉnh sửa, mở văn bản, khóa hoặc xóa văn bản.



Hình 2.81. Màn hình quản lý văn bản hệ thống

- Màn hình thêm mới văn bản hệ thống của quản trị viên

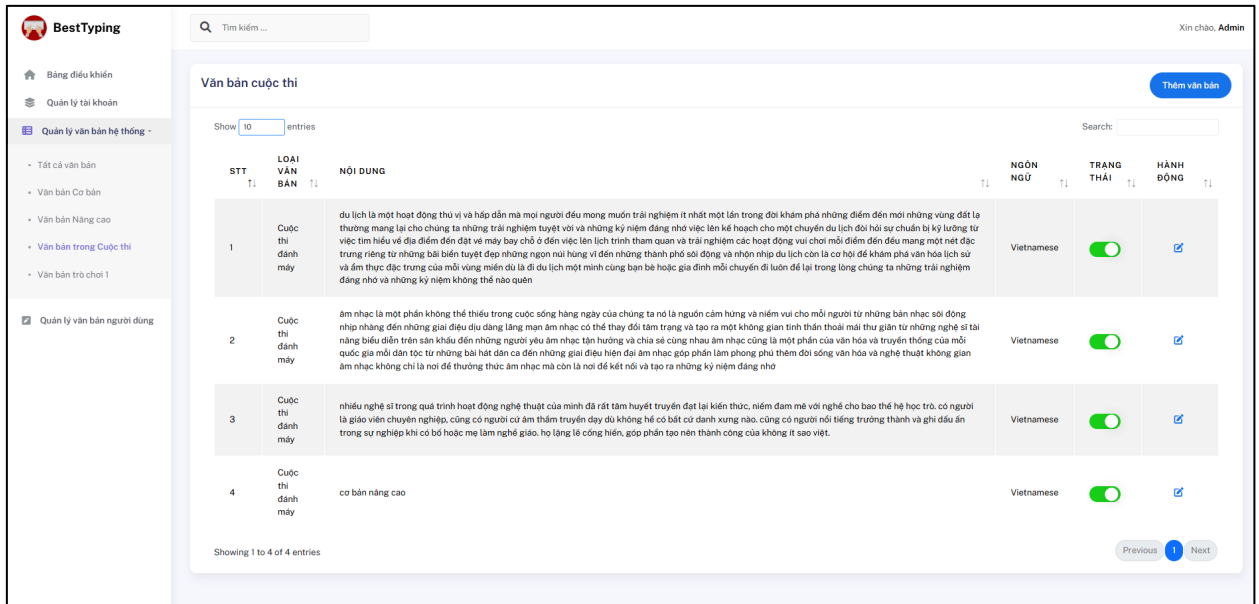
Ở màn hình thêm mới văn bản hệ thống quản trị viên có thể thêm mới bất kỳ một văn bản nào để có thể sử dụng trong hệ thống bao gồm: ngôn ngữ, nội dung văn bản, chọn lưu vào mục nào.



Hình 2.82. Màn hình thêm mới văn bản hệ thống của quản trị viên

- Màn hình quản lý văn bản trong cuộc thi chung tranh bảng xếp hạng của người dùng

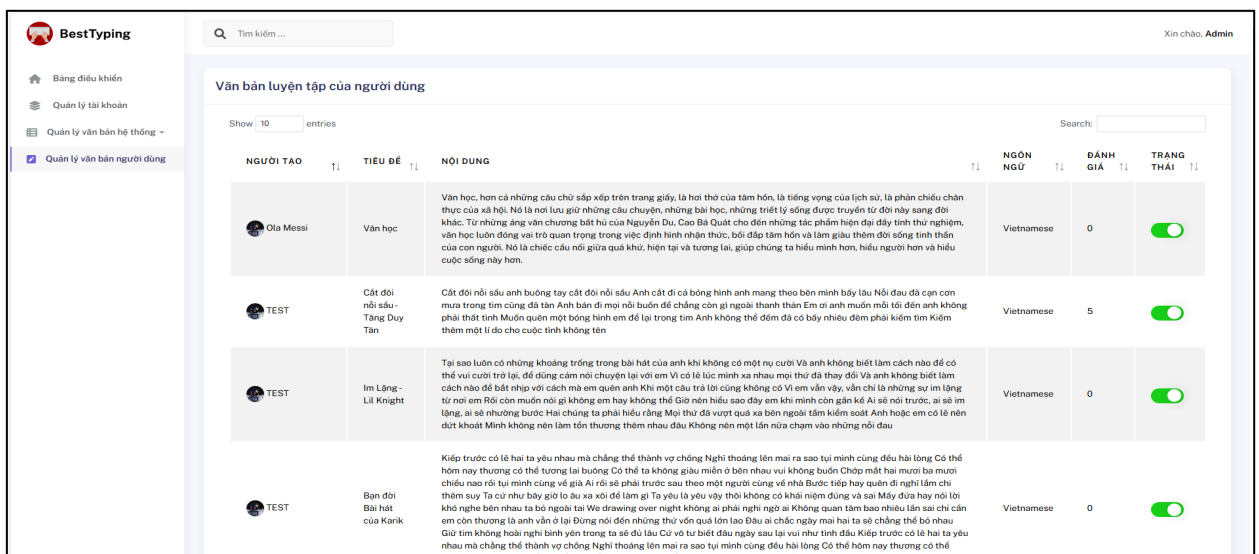
Văn bản trong cuộc thi chung tranh bảng xếp hạng của người dùng do hệ thống tạo hoặc người dùng tạo, quản trị viên có thể thêm mới, khóa, chỉnh sửa hoặc xóa văn bản.



Hình 2.83. Màn hình quản lý văn bản cuộc thi chung tranh bảng xếp hạng

- Màn hình quản lý văn bản do người dùng tạo

Quản trị viên có thể xem văn bản do người dùng tạo bao gồm: người tạo, tiêu đề, nội dung, ngôn ngữ và có thể khóa văn bản do người dùng tạo.



Hình 2.84. Màn hình quản lý văn bản do người dùng tạo

CHƯƠNG 3: CÀI ĐẶT VÀ PHÁT TRIỂN WEBSITE

3.1. Cấu trúc các thư viện

packages/	
├─ Antlr.3.5.0.2/	# Bộ parser/lexer được MVC phụ thuộc
│ └─ lib/	
│ └─ Antlr.3.5.0.2.nupkg	
├─ BCrypt.Net-Next.4.0.3/	# Hash/verify mật khẩu bằng BCrypt
│ └─ lib/	
│ └─ ico.png	
│ └─ readme.md	
│ └─ BCrypt.Net-Next.4.0.3.nupkg	
├─ bootstrap.3.4.1/...	# Framework CSS/JS responsive cho UI
├─ jQuery.3.4.1/...	# Thư viện DOM/AJAX nền tảng
├─ jQuery.Validation.1.17.0/...	# Plugin kiểm tra form client-side
├─ Microsoft.AspNet.Mvc.5.2.7/...	# Khung ASP.NET MVC chính
├─ Microsoft.AspNet.Razor.3.2.7/...	# Engine Razor render view .cshtml
├─ Microsoft.AspNet.SignalR.2.4.3/...	# SignalR server cho realtime hub
├─ Microsoft.AspNet.SignalR.Core.2.4.3/...	# Lõi chung SignalR
├─ Microsoft.AspNet.SignalR.JS.2.4.3/...	# SignalR client JavaScript
├─ Microsoft.AspNet.SignalR.SystemWeb.2.4.3/...	# Tích hợp SignalR vào System.Web
├─ Microsoft.AspNet.Web.Optimization.1.1.3/...	# Bundling/minify CSS/JS

└─ Microsoft.AspNet.WebPages.3.2.7/...	# Nền WebPages cho Razor helpers
└─ Microsoft.CodeDom.Providers.DotNetCompilerPlatform.2.0.1/...	# Roslyn compile view
└─ Microsoft.jQuery.Unobtrusive.Validation.3.2.11/...	# Liên kết data-val với jQuery Validation
└─ Microsoft.Owin.2.1.0/...	# Nền OWIN middleware
└─ Microsoft.Owin.Host.SystemWeb.2.1.0/...	# Host OWIN trong IIS/System.Web
└─ Microsoft.Owin.Security.2.1.0/...	# Middleware xác thực/claims
└─ Microsoft.Web.Infrastructure.1.0.0.0/...	# Tiện ích runtime ASP.NET (file change, cache)
└─ Modernizr.2.8.3/...	# Phát hiện hỗ trợ HTML5/CSS3
└─ Newtonsoft.Json.12.0.2/...	# Serialize/deserialize JSON (API, chatbot)
└─ Owin.1.0/...	# Giao diện chuẩn OWIN
└─ ReCaptcha-AspNet.1.9.0/...	# Tích hợp Google reCAPTCHA phía server
└─ System.Buffers.4.5.1/...	# ArrayPool/Span helper (dependency)
└─ System.Memory.4.5.4/...	# ReadOnlyMemory/Span cho .NET Framework
└─ System.Numerics.Vectors.4.5.0/...	# Vector math (cần bởi System.Memory)
└─ System.Runtime.CompilerServices.Unsafe.4.5.3/...	# API low-level cho Span/Memory
└─ WebGrease.1.6.0/...	# Công cụ tối ưu CSS/JS phục vụ bundling

3.2. Cấu trúc thư mục code

3.2.1. Cấu trúc thư mục code Back-end

BestTyping/BestTyping/

— App_Start/	# Cấu hình khởi động (Bundle, Filter, Route)
— Areas/	# Khu vực thư mục chính
— Admin/	# Phần quản trị viên
— AdminAreaRegistration.cs	# Khai báo area và route riêng cho admin
— Controllers/	# Controller quản trị
— Views/	# Razor giao diện admin +
Web.config	
— Controllers/	# Controller phục vụ người dùng
— Models/	# Middleware xử lý các thao tác cơ sở dữ liệu
— DataBestTyping.dbml)	# Ánh xạ LINQ to SQL tới DB
— DTO/	# ViewModel/DTO cho các chức năng
— Views/	# Razor view các file giao diện
— Theme/	# Giao diện nâng cao: âm thanh, css
— Content/	# CSS lỗi (bootstrap*.css, Site.css)
— Scripts/	# JS lỗi (bootstrap, jquery, signalR, validate, modernizr)
— fonts/	# Bộ glyphicons dùng trong layout
— Hub/RequestJoinHub.cs	# SignalR hub quản lý vào phòng thi, phát thông điệp
— Events/ChatScope.cs	# Phạm vi sự kiện chat/broadcast realtime
— Helper/GeneralHelper.cs	# Hàm tiện ích chung

— Security/CustomAuthorizeAttribute.cs	# Thuộc tính kiểm tra quyền tùy chỉnh
— Global.asax / Global.asax.cs	# Điểm vào ứng dụng, vòng đời ASP.NET
— Startup.cs	# Cấu hình OWIN và ánh xạ SignalR
— Web.config / Web.Debug.config / Web.Release.config	# Cấu hình IIS từng môi trường
— packages.config	# Danh sách thư viện NuGet
— Properties/	# Thông tin assembly + cấu hình publish
— BestTyping.csproj(.user)	# Định nghĩa dự án ASP.NET MVC
— favicon.ico	# Biểu tượng website

3.2.2. Cấu trúc thư mục code Front-end

BestTyping/BestTyping/

— Content/	# CSS phục vụ client (Bootstrap, Site.css)
— Scripts/	# JS chạy trên trình duyệt (jQuery, SignalR, Validate...)
— Theme/	# Kho giao diện nâng cao: animation, css, datatable,
	fonts, images, js hiệu ứng, âm thanh
— fonts/	# Font glyphsicons dùng cho giao diện
— Views/	
— Account/	# Màn hình đăng ký, đăng nhập, đổi mật khẩu
— Competition/	# Trang xem và tham gia cuộc thi
— DashBoardEdu/	# Dashboard giáo viên
— DashBoardStudent/	# Dashboard học sinh
— Home/	# Trang chủ, giới thiệu, landing
— Shared/	# Layout, navbar, footer, modal chung
— _ViewStart.cshtml	# Chỉ định layout mặc định cho toàn bộ view

— Areas/	
— Admin/	# Trang quản trị viên
— Views/	# Giao diện quản trị (Account, Competition,
	Dashboard, Setting, TypingTest, Web.config)
— favicon.ico	# Icon hiển thị trên tab trình duyệt
— ../img/*.jpg	# Ảnh

3.3. Xử lý thời gian thực

Vấn đề: Giáo viên phải thấy yêu cầu tham gia lớp ngay khi học viên gửi, ngược lại học viên cần biết trạng thái hệ thống như số người online mà không phải tải lại trang. Nếu chỉ dùng HTTP truyền thống, dữ liệu luôn trễ và hay xung đột khi nhiều người gửi cùng lúc.

Cách giải quyết:

- Dùng SignalR để mở kết nối realtime giữa trình duyệt và server.
- RequestJoinHub trên server quản lý broadcast danh sách yêu cầu và đếm người online.
- Dashboard học viên/giáo viên đăng ký lắng nghe sự kiện hub để đẩy/gấp dữ liệu lập tức.
- Khi một học sinh gửi request, trang student gọi `hub.server.sendRequsetJoinList` để server phát lại danh sách mới cho toàn bộ thiết bị đang mở lớp đó.

Module code xử lý tại RequestJoinHub.cs:

```
public class RequestJoinHub : Hub
{
    // Biến static dùng để đếm số lượng người dùng đang online trong Hub
    private static int _onlineUsers = 0;

    // Gửi danh sách yêu cầu tham gia lớp (real-time)
    public void SendRequsetJoinList(int idroom)
    {
        // Truy vấn phòng theo id từ Database
    }
}
```

```
var room = db.CLASSROOMS.FirstOrDefault(r => r.ClassRoomId ==
idroom);

// Nếu phòng tồn tại → đọc danh sách yêu cầu từ JSON trong DB
// Nếu không tồn tại → trả về danh sách rỗng
var data = room != null
    ?
JsonConvert.DeserializeObject<List<USERROOM>>(room.ListUserRequest)
    : new List<USERROOM>();

// Gửi danh sách yêu cầu xuống toàn bộ client đang kết nối
// Client sẽ gọi hàm updateRequestJoinTable(data, idroom) để cập nhật
UI

Clients.All.updateRequestJoinTable(data, idroom);
}

// Tăng số lượng người dùng online khi có client mới kết nối
public override Task OnConnected()
{
    _onlineUsers++; // Tăng số lượng người dùng online
    // Gửi số lượng người dùng online mới xuống toàn bộ client
    Clients.All.updateOnlineCount(_onlineUsers);
    return base.OnConnected();
}

// Giảm số lượng khi client ngắt kết nối
public override Task OnDisconnected(bool stopCalled)
{
    _onlineUsers--; // Giảm người dùng online
    // Gửi số lượng người dùng online mới xuống toàn bộ client
    Clients.All.updateOnlineCount(_onlineUsers);
    return base.OnDisconnected(stopCalled);
}}
```

Phía máy khách việc tích hợp được thực hiện thông qua JavaScript để lắng nghe và phản hồi các sự kiện từ Hub.

Tại giao diện của học viên ngay khi nhận được phản hồi thành công từ thao tác gửi yêu cầu, hệ thống sẽ lập tức gọi phương thức `hub.server.sendRequetJoinList` để kích hoạt việc cập nhật dữ liệu trên toàn hệ thống.

Module code xử lý tại `DashboardStudent.cshtml`:

```
if (response.code == 200) {  
    // Hiển thị thông báo thành công cho người dùng  
    toastr.success(response.msg);  
    // kích hoạt hub nhằm thông báo real-time cho giáo viên  
    // - Khởi động SignalR Hub (nếu chưa chạy)  
    // - Gọi hàm sendRequetJoinList bên server để cập nhật danh sách  
    $.connection.hub.start().done(function () {  
        // Gọi phương thức sendRequetJoinList từ phía server (Hub)  
        // → Giáo viên sẽ thấy yêu cầu tham gia lớp được cập nhật ngay lập tức  
        hub.server.sendRequetJoinList(idroom);  
    });  
}
```

Module code xử lý tại `SettingClassRoom.cshtml` và `_LayoutUser.cshtml`:

```
// SettingClassRoom.cshtml: Lắng nghe và cập nhật bảng yêu cầu  
hub.client.updateRequestJoinTable = function (dataUserRequest, roomId)  
{  
    if (roomId === dataid) {  
        var $table = $("#listrequest > tbody").empty();  
        // ... Logic render lại danh sách yêu cầu mới ...  
    }  
};  
  
// _LayoutUser.cshtml: Khởi tạo kết nối SignalR toàn cục  
<script src="~/Scripts/jquery.signalR-2.4.3.min.js"></script>
```

```
<script src=~/.signalr/hubs"></script>

var hub = $.connection.requestJoinHub;

hub.client.updateOnlineCount = function (count) { console.log(count); };

$.connection.hub.start().done(function () { console.log("Connected to
SignalR Hub"); });
```

Kết quả: Yêu cầu tham gia lớp xuất hiện ngay trên màn hình giáo viên mà không reload, đồng thời bộ đếm người online luôn chính xác. Điều này giảm độ trễ xử lý, hạn chế nhầm lẫn khi nhiều học viên cùng gửi, và cải thiện trải nghiệm vì hệ thống phản hồi tức thì.

3.4. Xử lý kiểm tra chữ đúng sai

Vấn đề: Trong việc cung cấp phản hồi trực quan theo thời gian thực là vô cùng cần thiết cụ thể ngay khi người dùng gõ từng ký tự, hệ thống cần lập tức hiển thị màu xanh cho các thao tác đúng nhằm khích lệ và màu đỏ cho các thao tác sai để cảnh báo, giúp người dùng nhanh chóng nhận biết và sửa lỗi kịp thời.

Cách giải quyết:

- Render từng từ thành các `<span class="char">` riêng biệt.
- So sánh từng ký tự real-time khi người dùng gõ bàn phím.
- Áp dụng class CSS:
 - + `.char-correct` → màu xanh (`#11be52`)
 - + `.char-wrong` → màu đỏ (`#ff1818`)
 - + `.char-current` → highlight ký tự tiếp theo

Module code xử lý tại `Theme/js/character-typing-validator.js`

```
javascript

config: {

    correctClass: 'char-correct', // Màu xanh

    wrongClass: 'char-wrong',    // Màu đỏ

    currentClass: 'char-current', // Highlight

    wordSelector: '.word',

    inputSelector: '#inputfield',
```

```
        containerSelector: '#quote'
    }
}
```

Module code xử lý tại Theme/css/character-typing-validator.css

```
css

/* Ký tự đúng - màu xanh */
.char-correct {
    color: #11be52;
    background-color: rgba(17, 190, 82, 0.1);
}

/* Ký tự sai - màu đỏ */
.char-wrong {
    color: #ff1818;
    background-color: rgba(255, 24, 24, 0.1);
    text-decoration: underline;
}

/* Ký tự đang chờ gõ */
.char-current {
    background-color: rgba(221, 221, 221, 0.5);
    border-bottom: 2px solid #4a90e2;
    animation: blink-cursor 1s infinite;
}
```

Module code xử lý tại Theme/js/character-typing-integration.js

```
$(document).ready(function () {
    // Kiểm tra xem module có sẵn không
    if (typeof CharacterTypingValidator === 'undefined') {
        console.error('CharacterTypingValidator module not loaded!');
        return;
    }
}
```

```
// Override hàm renderWords để sử dụng Character Typing Validator
var originalRenderWords = window.renderWords;

if (typeof originalRenderWords === 'function') {
    // Lưu hàm gốc
    window._originalRenderWords = originalRenderWords;

    // Override với version mới
    window.renderWords = function (wordsArray) {
        // Sử dụng Character Typing Validator để render
        CharacterTypingValidator.renderTextWithCharacters(wordsArray);

        // Vẫn gọi hàm highlight từ gốc nếu cần
        if (typeof highlightNextWord === 'function') {
            highlightNextWord();
        }
    };
}

// Override hàm checkCurrentWord để sử dụng Character Typing
Validator
var originalCheckCurrentWord = window.checkCurrentWord;

if (typeof originalCheckCurrentWord === 'function') {
    window._originalCheckCurrentWord = originalCheckCurrentWord;
    window.checkCurrentWord = function () {
        var inputText = $("#inputfield").val();

        CharacterTypingValidator.checkCurrentWordCharacters(inputText);
    };
}

// Override hàm checkWord để sử dụng Character Typing Validator
var originalCheckWord = window.checkWord;
```

```
if (typeof originalCheckWord === 'function') {
    window._originalCheckWord = originalCheckWord;
    window.checkWord = function () {
        var inputText = $("#inputfield").val().trim();
        CharacterTypingValidator.completeWord(inputText);

        // Gọi hàm gốc để xử lý logic khác (như lưu vào localStorage, đếm từ...)
        if (window._originalCheckWord) {
            window._originalCheckWord();
        }
    };
}

// Override hàm resetGame để reset Character Typing Validator
var originalResetGame = window.resetGame;
if (typeof originalResetGame === 'function') {
    window._originalResetGame = originalResetGame;
    window.resetGame = function () {
        CharacterTypingValidator.reset();
        if (window._originalResetGame) {
            window._originalResetGame();
        }
    };
}

console.log('Character Typing Validator integrated successfully');
});
```

Module code xử lý tại CheckTyping.cshtml

```
<!-- CSS -->
<link href="~/Theme/css/character-typing-validator.css" rel="stylesheet"
```

/>

// JavaScript (sau jQuery)

<script src="~/Theme/js/character-typing-validator.js"></script>

<script src="~/Theme/js/character-typing-integration.js"></script>

Kết quả:

- Phản hồi trực quan real-time.
- Người dùng thấy ngay ký tự đúng sẽ hiện màu xanh và chữ sai sẽ hiện màu đỏ.
- Cải thiện độ chính xác khi luyện tập.
- Hỗ trợ tiếng Việt có dấu.
- Tương thích với hệ thống hiện có.

3.5. Xử lý xác thực qua Email

Vấn đề: Cần đảm bảo Email đăng ký hợp lệ, kích hoạt tài khoản trước khi dùng, hỗ trợ quên mật khẩu an toàn và giới hạn truy cập đối với tài khoản chưa xác nhận. Nếu không có cơ chế xác thực, hệ thống dễ bị spam tài khoản, người dùng không kích hoạt được, và không thể khôi phục mật khẩu.

Cách giải quyết:

- Khi đăng ký, generate GUID lưu vào USER.LinkVerification, gửi Email chứa link /Account/VerifyEmail?token=....
- Khi người dùng click link hệ thống đối chiếu token-Email, bật IsEnable=true để kích hoạt.
- Đăng nhập kiểm tra IsEnable chưa xác thực thì từ chối và hướng dẫn kiểm tra Email.
- Quy trình quên mật khẩu: kiểm tra tài khoản/Email, phát OTP 6 ký tự, gửi qua Email, lưu vào USER.XacMinh. Người dùng nhập OTP hợp lệ mới được đổi mật khẩu (hash bằng BCrypt).
- Tất cả Email sử dụng hàm SendEmail với template HTML riêng cho kích hoạt và OTP.

Module code xử lý tại AccountController.cs (thông báo cho người dùng)

[HttpGet]

```
public ActionResult VerifyEmail(string token, string Email)
{
    // Tìm user khớp Email và token
    var user = db.USERS.FirstOrDefault(u => u.Email.Equals(Email) &&
u.LinkVerification.Equals(token));
    if (user != null)
    {
        user.IsEnabled = true; // Kích hoạt tài khoản
        db.SubmitChanges(); // Lưu vào DB

        ViewBag.IsChecked = true;
        ViewBag.Notification = "Xác thực Email thành công!";
        return View();
    }
    // Xử lý nếu xác thực lỗi (tùy chọn)
    return View();
}

[HttpPost]
public JsonResult RegisterAccount(string fullname, string username,
string password, string Email)
{
    // Tạo token và link xác thực
    var verifyToken = Guid.NewGuid().ToString();

    var confirmationLink = Url.Action("VerifyEmail", "Account", new { token =
verifyToken, Email = Email }, Request.Url.Scheme);

    // Gửi Email
    SendEmail(Email, "Xác thực tài khoản BestTyping", confirmationLink, 0);

    // Khởi tạo user (Code gốc thiếu đoạn này)
    var user = new USER();
```

```
user.FullName = fullname;

user.UserName = username;

user.Password = password; // Lưu ý: Nên mã hóa password

user.Email = Email;

user.LinkVerification = verifyToken; // Gán token

user.IsEnabled = false; // Mặc định chưa kích hoạt

// Thêm mới và lưu DB

db.USERS.InsertOnSubmit(user);

db.SubmitChanges();

return Json(new { code = 200, msg = "Vui lòng kiểm tra Email để xác minh tài khoản" });

}
```

Module code xử lý tại AccountController.cs (nhấn kích hoạt tài khoản và quên mật khẩu ở gmail)

```
[HttpPost]

public JsonResult CheckLogin(string usernamesignin, string passsignin)

{

    // Tìm user theo username

    var checkUser = db.USERS.FirstOrDefault(u => u.TaiKhoan == usernamesignin);

    // Kiểm tra user tồn tại & khớp mật khẩu (dùng BCrypt)

    if (checkUser != null && BCrypt.Net.BCrypt.Verify(passsignin, checkUser.MatKhau))

    {

        // Kiểm tra đã kích hoạt Email chưa

        if (checkUser.IsEnabled == false)

            return Json(new { code = 400, msg = "Vui lòng kiểm tra Email để kích hoạt tài khoản" });

        // Đăng nhập thành công (Thường sẽ lưu Session ở đây)
```

```
        return Json(new { code = 200, msg = "Đăng nhập thành công" });
    }

    // Sai thông tin
    return Json(new { code = 500, msg = "Tài khoản hoặc mật khẩu không
đúng" });
}

[HttpPost]
public JsonResult SendVerificationCode(string recipientEmail)
{
    var checkUser = db.USERS.FirstOrDefault(u => u.Email ==
recipientEmail);

    if (checkUser != null)
    {
        // Tạo mã ngẫu nhiên
        string verificationCode = GenerateVerificationCode();

        // Lưu mã vào DB để đối chiếu sau
        checkUser.XacMinh = verificationCode;

        db.SubmitChanges();

        // Gửi Email
        SendEmail(recipientEmail, "Mã xác minh đổi mật khẩu",
verificationCode, 1);

        return Json(new { code = 200, msg = "Mã xác nhận đã gửi" });
    }

    return Json(new { code = 500, msg = "Email không tồn tại!" });
}

[HttpPost]
public JsonResult ResetPassWord(string username, string password,
string Email, string sendcode)
{

```

```
// Tìm user khớp Username & Email

var checkUser = db.USERS.SingleOrDefault(n => n.Email == Email &&
n.TaiKhoan == username);

// Kiểm tra mã xác minh

if (checkUser != null && checkUser.XacMinh == sendcode)
{
    // Hash mật khẩu mới & lưu

    checkUser.MatKhau = BCrypt.Net.BCrypt.HashPassword(password);

    db.SubmitChanges();

    return Json(new { code = 200, msg = "Đổi mật khẩu thành công" });
}

return Json(new { code = 500, msg = "Mã xác minh không hợp lệ!" });
}
```

Module code xử lý tại AccountController.cs (form xác nhận được gửi về gmail)

```
private void SendEmail(string recipientEmail, string subject, string
message, int select)
{
    // Chọn template: 0 = Link xác thực, 1 = Mã số to

    string template = select == 0

        ? $"...<a href=\"{message}\">Xác thực Email</a>..."
        : $"...<p style=\"font-size:60px;\">{message}</p>...";

    using (MailMessage mail = new MailMessage())
    {
        // Cấu hình người gửi

        mail.From = new MailAddress("pingvocuc555@gmail.com",
"BestTyping");

        mail.To.Add(recipientEmail);

        mail.Subject = subject;
```

```
mail.IsBodyHtml = true;

mail.Body = template;

using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
{
    // Cấu hình SMTP Gmail

    // LƯU Ý: Phải dùng 'Mật khẩu ứng dụng' (App Password), không phải
    mật khẩu đăng nhập gmail

    smtp.Credentials = new
    NetworkCredential("pingvocuc555@gmail.com", "your_app_password_here");

    smtp.EnableSsl = true;

    smtp.Send(mail);
}
}
```

Kết quả: Người dùng chỉ đăng nhập được sau khi xác thực Email, tài khoản trùng Email/tên được chặn từ đầu, quên mật khẩu cần OTP gửi đến Email đã đăng ký. Quá trình này giảm spam, đảm bảo tính xác thực và giúp khôi phục mật khẩu an toàn, đồng thời cung cấp trải nghiệm rõ ràng thông qua toast thông báo ở giao diện Theme/js/main.js.

3.6. Xử lý Chatbot AI hỗ trợ

Vấn đề: người dùng cần tư vấn luyện gõ nhanh, gợi ý bài tập và phân tích lỗi ngay trong giao diện chính mà không phải rời trang. Nếu chỉ dùng FAQ tĩnh, trải nghiệm nghèo nàn và không cá nhân hóa theo dữ liệu luyện tập.

Cách giải quyết:

- Tích hợp hộp chat cố định trong _LayoutUser.cshtml, lưu lịch sử hội thoại + lỗi ký tự/từ trong localStorage.
- Khi người dùng gửi câu hỏi, thì hệ thống client gửi AJAX JSON lên /Home/GenerateAnswer, kèm thêm dữ liệu lỗi charErrors, textErrors.
- Server dùng ChatScope.GenerateAnswer tạo prompt động (nhấn mạnh vai trò BestTypingBot, cách phân tích lỗi, mapping bài tập) rồi gọi API Gemini để lấy phản hồi

tiếng Việt.

- Phản hồi được định dạng lại trên client: tự động gắn link đến các tuyến /typing-test, /Home/GameTyping, hiển thị chữ đậm/list, rồi thêm vào lịch sử.

Module code xử lý tại `_LayoutUser.cshtml`

```
<div class="chat-box">...</div>

<script>

function sendMessage() {

    // Lấy tin nhắn và kiểm tra rỗng

    const messageText = $('#chat-input-field').val().trim();

    if (!messageText) return;

    // Lưu tin nhắn người dùng vào LocalStorage

    const chatHistory = JSON.parse(localStorage.getItem('chatHistory') ||

'[]');

    chatHistory.push({ text: messageText, fromUser: true });

    localStorage.setItem('chatHistory', JSON.stringify(chatHistory));

    // Cập nhật giao diện ngay lập tức

    loadChatHistory();

    // Lấy thông tin lỗi gõ phím (nếu có) để gửi kèm context

    const charErrors = JSON.parse(localStorage.getItem('charErrors')) || [];

    const textErrors = JSON.parse(localStorage.getItem('textErrors')) || [];

    // Gọi AJAX gửi dữ liệu về Controller

    $.ajax({

        url: '@Url.Action("GenerateAnswer", "Home")', // Lưu ý: Dùng dấu

ngoặc kép chuẩn

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify({

            question: messageText,

            chars: charErrors,
```

```
        texts: textErrors

    }},

    success: function (response) {

        // Lưu phản hồi của AI/Bot vào LocalStorage

        chatHistory.push({ text: response, fromUser: false });

        localStorage.setItem('chatHistory', JSON.stringify(chatHistory));

        // Cập nhật lại giao diện khi có phản hồi

        loadChatHistory();

    }

});

}

</script>
```

Module code xử lý tại HomeController.cs

```
[HttpPost]

public async Task<ActionResult> GenerateAnswer(Conversation request)

{

    // Kiểm tra rỗng

    if (string.IsNullOrEmpty(request.Question))

        return Json("Vui lòng nhập câu hỏi!");

    // Giới hạn độ dài câu hỏi (<= 30 từ)

    if (GeneralHelper.GetTotalWords(request.Question) > 30)

        return Json("Hỏi ngắn thôi (dưới 30 từ ấy) ...");

    try

    {

        // Gọi xử lý logic (ChatScope)

        var result = await ChatScope.GenerateAnswer(request);

        return Json(result);

    }

}
```

```
catch
{
    // Xử lý lỗi ngoại lệ
    return Json("Nhấn từ từ thôi \nChờ tôi suy nghĩ nữa.");
}
}
```

Module code xử lý tại ChatScope.cs

```
public class ChatScope
{
    private static readonly HttpClient Client = new HttpClient();

    public static async Task<string> GenerateAnswer(Conversation
conversation)
    {
        // Cắt lịch sử nếu quá dài (> 30 message)
        if (conversation.ChatHistory.Count > 30)
            conversation.ChatHistory =
conversation.ChatHistory.Skip(Math.Max(0, conversation.ChatHistory.Count -
20)).ToList();

        // Kiểm tra từ khóa hỏi về lỗi sai
        string lowerQ = conversation.Question?.ToLower() ?? "";
        bool isErrorCase = lowerQ.Contains("lỗi") && lowerQ.Contains("tôi");
        if (isErrorCase)
        {
            // Kèm dữ liệu lỗi vào câu hỏi để Bot phân tích
            if ((conversation.Chars?.Any() ?? false) || (conversation.Texts?.Any()
?? false))
                conversation.Question += "\n\n[errorStats]\nChars: ...";
            else
                conversation.Question += "\n\n[errorStats] Hôm nay bạn chưa luyện
```

```
tập...";

    }

    // Tạo System Prompt

    var promptBuilder = new StringBuilder();

    promptBuilder.AppendLine("Bạn là BestTypingBot...");


    // Cấu hình Request

    var requestBody = new

    {

        SystemInstruction = new { Parts = new { Text =

promptBuilder.ToString() } },

        Contents = conversation.ChatHistory

    };

    // Gửi API

    var content = new

StringContent(JsonConvert.SerializeObject(requestBody), Encoding.UTF8,

"application/json");

    var response = await Client.PostAsync("YOUR_API_ENDPOINT",

content);

    // Parse kết quả

    var jsonResponse = await response.Content.ReadAsStringAsync();

    var dto = JsonConvert.DeserializeObject<Response>(jsonResponse);

    return dto?.Candidates?[0].Content.Parts[0].Text;

}

}
```

3.7. Xử lý chống spam và bảo mật

Vấn đề: Trang đăng ký dễ bị bot spam, tài khoản giả mạo, dữ liệu nhạy cảm như mật khẩu có nguy cơ lộ, Email chưa xác thực có thể truy cập hệ thống, quên mật khẩu cần bảo đảm đúng chủ sở hữu.

Cách giải quyết:

- Dùng reCAPTCHA phiên bản v2 trước khi gửi form đăng ký, để cho client gọi /Account/CheckCaptcha để xác nhận token, từ chối request không vượt captcha.
- Mã hóa mật khẩu: dùng BCrypt.Net-Next để hash mật khẩu khi tạo/đổi, login dùng BCrypt.Verify.
- Xác thực Email bắt buộc: tài khoản tạo mới IsEnable=false và gửi link kích hoạt, login sẽ chặn đến khi xác thực.
- OTP reset password: yêu cầu kết hợp username+Email, gửi mã xác minh 6 ký tự qua Email, bắt buộc trùng mã trước khi đổi mật khẩu.
- SMTP có SSL: gửi mail qua Gmail SMTP 587, SSL bật.
- Kiểm tra trùng thông tin: trước khi đăng ký, tìm Email/tài khoản đã tồn tại.
- Chặn truy cập tài khoản bị khóa (IsDelete).
- Giới hạn độ dài câu hỏi chatbot (≤ 30 từ) để tránh lạm dụng.

Module code xử lý tại AccountController.cs

```
[HttpPost]

public JsonResult CheckCaptcha(string captcharesponse)

{

    // Validate Google ReCaptcha

    bool valid = ReCaptcha.ValidateCaptcha(captcharesponse);

    return Json(new { result = valid ? "PASS" : "FAIL" });

}

[HttpPost]

public JsonResult RegisterAccount(string fullname, string username,
string password, string Email)

{

    // Kiểm tra trùng lặp

    if (db.USERS.Any(u => u.Email == Email || u.TaiKhoan == username))

        return Json(new { code = 400, msg = "Tài khoản hoặc Email đã tồn tại!"

});
```

```
// Khởi tạo User

var verifyToken = Guid.NewGuid().ToString();

var user = new USER();

user.FullName = fullname;

user.TaiKhoan = username;

user.Email = Email;

user.MatKhau = BCrypt.Net.BCrypt.HashPassword(password); // Hash
pass

user.IsEnabled = false;

user.LinkVerification = verifyToken;

// Lưu DB

db.USERS.InsertOnSubmit(user);

db.SubmitChanges();

// Gửi Email (Giả định hàm Url.Action tạo link ở đây)

string link = Url.Action("VerifyEmail", "Account", new { token =
verifyToken, Email = Email }, Request.Url.Scheme);

SendEmail(Email, "Xác thực tài khoản", link, 0);

return Json(new { code = 200, msg = "Đăng ký thành công" });
}

[HttpPost]

public JsonResult CheckLogin(string usernamesignin, string passsignin)
{
    var user = db.USERS.FirstOrDefault(u => u.TaiKhoan == usernamesignin);

    // Check pass và tồn tại

    if (user != null && BCrypt.Net.BCrypt.Verify(passsignin, user.MatKhau))
    {
        if (user.IsEnabled == false) return Json(new { code = 400, msg = "Vui
lòng kiểm tra Email để kích hoạt" });

        if (user.IsDelete == true) return Json(new { code = 400, msg = "Tài
```

khoản đã bị khóa" });

```
    Session["User"] = user; // Lưu session
```

```
    return Json(new { code = 200, msg = "Đăng Nhập thành công" });
```

```
}
```

```
    return Json(new { code = 500, msg = "Tài khoản hoặc mật khẩu không  
hợp lệ!" });
```

```
}
```

```
[HttpPost]
```

```
public JsonResult SendVerificationCode(string recipientEmail)
```

```
{
```

```
    var user = db.USERs.FirstOrDefault(u => u.Email == recipientEmail);
```

```
    if (user == null) return Json(new { code = 500, msg = "Email không tồn  
tại!" });
```

```
    // Tạo code verify
```

```
    string code = GenerateVerificationCode();
```

```
    user.XacMinh = code;
```

```
    db.SubmitChanges();
```

```
    SendEmail(recipientEmail, "Mã xác minh đổi mật khẩu", code, 1);
```

```
    return Json(new { code = 200, msg = "Mã xác nhận đã gửi" });
```

```
}
```

```
private void SendEmail(string recipientEmail, string subject, string  
message, int select)
```

```
{
```

```
    // 0: Link, 1: Code to
```

```
    string template = select == 0
```

```
        ? $"...<a href=\"{message}\">Xác thực Email</a>..."
```

```
        : $"...<p style=\"font-size:60px;\">{message}</p>...";
```

```
    using (MailMessage mail = new MailMessage())
```

```
{
```

```
mail.From = new MailAddress("pingvocuc555@gmail.com",
"BestTyping");

mail.To.Add(recipientEmail);

mail.Subject = subject;

mail.IsBodyHtml = true;

mail.Body = template;

using (SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587))
{
    // App Password đã cung cấp

    smtp.Credentials = new
NetworkCredential("pingvocuc555@gmail.com", "bkxflmlmnyxxrzrz");

    smtp.EnableSsl = true;

    smtp.Send(mail);
}
}
```

Module code xử lý tại _LayoutUser.cshtml

```
$.ajax({

    // Lưu ý: @Url.Action chỉ chạy trong file .cshtml. Nếu file .js riêng thì
dùng "/Home/GenerateAnswer"

    url: '@Url.Action("GenerateAnswer", "Home")',
    method: 'POST',
    contentType: 'application/json',
    // Đóng gói dữ liệu gửi đi
    data: JSON.stringify({
        question: messageText,
        chars: charErrors,
        texts: textErrors
    }),
    },
```

```
success: function(response) {  
    // Xử lý kết quả trả về  
    console.log("Success:", response);  
},  
error: function(err) {  
    console.error("Lỗi:", err);  
}  
});
```

Module code xử lý tại HomeController.cs

```
// Kiểm tra độ dài câu hỏi (Giới hạn 30 từ)  
if (GeneralHelper.GetTotalWords(request.Question) > 30)  
    return Json("Hỏi ngắn thôi (dưới 30 từ ấy) ...");
```

Kết quả: bot spam đăng ký khó vượt vì phải qua reCAPTCHA và Email kích hoạt, mật khẩu lưu ở dạng hash, người dùng bị khóa hoặc chưa xác thực không thể đăng nhập, reset mật khẩu yêu cầu OTP gửi về Email hợp lệ. Nhờ đó, hệ thống giảm đáng kể tài khoản giả mạo và bảo vệ dữ liệu người dùng.

3.8. Kiểm thử

3.8.1. Các phương án kiểm thử

- Unit Testing (Kiểm thử đơn vị)

Mục đích: Kiểm thử từng phương thức, hàm riêng lẻ trong các lớp.

Công cụ: NUnit / xUnit, Moq / NSubstitute.

Phạm vi: Controllers (Account, TypingTest...), Helper, Security, Events, Business logic.

- Integration Testing (Kiểm thử tích hợp)

Mục đích: Kiểm thử tương tác Controller ↔ Model ↔ Database.

Công cụ: NUnit + In-Memory DB / LocalDB.

Phạm vi: LINQ to SQL, DataContext, Session, SignalR.

- System Testing (End-to-end)

Mục đích: Kiểm thử toàn bộ luồng người dùng.

Công cụ: Selenium WebDriver, Postman, BrowserStack.

Kịch bản chính: đăng ký→xác thực→đăng nhập→làm bài→xem kết quả; giáo viên tạo lớp→học sinh tham gia→giao đề→xem kết quả; tạo & tham gia cuộc thi.

- **Security Testing**

Mục đích: Kiểm tra lỗ hổng (OWASP).

Công cụ: OWASP ZAP, Burp Suite.

Phạm vi: Auth/Authorization, SQLi, XSS, CSRF, session, password hashing (BCrypt), input validation, file upload.

- **Performance Testing**

Mục đích: Kiểm thử hiệu suất, tải đồng thời.

Công cụ: Apache JMeter, LoadRunner.

Mục tiêu: Response time < 2s, hỗ trợ ≥ 100 concurrent users.

- **Usability Testing**

Mục đích: Kiểm tra trải nghiệm người dùng (UAT).

Phương pháp: Beta test, UAT sessions, A/B test.

- **API Testing**

Công cụ: Postman, Swagger/OpenAPI, REST Assured.

Phạm vi: JSON endpoints, status codes, error handling.

3.8.2. Các chức năng cần kiểm thử

- **Quản lý tài khoản (AccountController):** Đăng ký, Xác thực Email, Đăng nhập, Quên mật khẩu, Quản lý profile, Upload avatar, Đổi mật khẩu, Đăng xuất (TC từ TC_ACC_001 → TC_ACC_033).

- **Kiểm tra luyện gõ bàn phím (TypingTestController):** Kiểm tra cơ bản/nâng cao/tùy chỉnh, lưu kết quả, cập nhật UserProgress, tính accuracy, lấy bài theo ngôn ngữ (TC_TYP_001 → TC_TYP_015).

- **Game gõ bàn phím nhanh (TypingGameController):** Lưu điểm, bảng xếp hạng, levels (TC_GAM_001 → TC_GAM_004).

- **Luyện tập văn bản (TextPracticeController):** CRUD văn bản, Like, Rating, Tạo bằng AI, Chat AI (TC_TXT_001 → TC_TXT_012).

- **Cuộc thi (CompetitionController):** Tạo, Tham gia, Lưu kết quả, Bảng xếp hạng (TC_COM_001 → TC_COM_009).

- **Quản lý lớp học - Giáo viên (DashboardEduController):** Tạo/chỉnh sửa/xóa lớp, mã tham gia, mật khẩu, duyệt thành viên, quản lý bài thi, xuất Excel (TC_EDU_001 → TC_EDU_026).

- **Học sinh (DashboardStudentController):** Tham gia lớp, làm bài, xem kết quả (TC_STU_001 → TC_STU_013).

- **Trang chủ & Thống kê (HomeController):** Bảng xếp hạng, thống kê, bài học (TC_HOM_001 → TC_HOM_011).

- **SignalR Real-time (RequestJoinHub):** Kết nối với hub, gửi và nhận thời gian thực (TC_SIG_001 → TC_SIG_005).

3.8.3. Test cases chi tiết

- Bảng Test Case Quản lý tài khoản

Bảng 2.80. Bảng Test Case Quản lý tài khoản

TC_ID	Tên Test Case	Tác nhân	Mục đích kiểm thử	Tiền điều kiện	Dữ liệu test (ví dụ)	Các bước thực hiện	Kết quả mong đợi	Ưu tiên
TC_ACC_001	Đăng ký tài khoản hợp lệ	User	Kiểm tra đăng ký thành công	Email /User name chưa tồn tại	Fullname, Username, Password, Email hợp lệ	1. Mở form đăng ký 2. Nhập dữ liệu hợp lệ 3. Xác thực reCAPTCHA 4. Submit	Trả về 200; gửi Email xác thực; tài khoản tạo với IsEnable = false	High
TC_ACC_002	Đăng ký với Email đã tồn tại	User	Kiểm tra lỗi trùng Email	Email đã tồn tại trong DB	Email trùng	1. Mở form đăng ký 2. Nhập Email trùng 3. Submit	Hiển thị lỗi "Email đã tồn tại"; không tạo tài khoản	High
TC_ACC_003	Xác thực Email thành công	User	Kích hoạt tài khoản bằng	Đã gửi Email xác thực; token	Token hợp lệ	1. Click link xác thực hoặc gọi API verify	Tài khoản IsEnable = true; thông	High

TC_ID	Tên Test Case	Tác nhân	Mục đích kiểm thử	Tiền điều kiện	Dữ liệu test (ví dụ)	Các bước thực hiện	Kết quả mong đợi	Ưu tiên
			token hợp lệ	hợp lệ			báo xác thực thành công	
TC_ACC_004	Xác thực Email hết hạn/token không hợp lệ	User	Kiểm tra token lỗi/hết hạn	Token hết hạn hoặc không hợp lệ	Token expired	1. Click link xác thực với token hết hạn/sai	Hiện thị lỗi token hết hạn/không hợp lệ; không kích hoạt tài khoản	Medium
TC_ACC_005	Đăng nhập thành công	User	Kiểm tra xác thực thành công	Tài khoản đã kích hoạt	Username + Password đúng	1. Mở form login 2. Nhập thông tin đúng 3. Submit	Đăng nhập thành công; tạo session; chuyển hướng dashboard	High

TC_ID	Tên Test Case	Tác nhân	Mục đích kiểm thử	Tiền điều kiện	Dữ liệu test (ví dụ)	Các bước thực hiện	Kết quả mong đợi	Ưu tiên
TC_ACC_006	Đăng nhập sai mật khẩu	User	Kiểm tra xử lý lỗi đăng nhập	Tài khoản tồn tại	Username + Password sai	1. Mở form login 2. Nhập mật khẩu sai 3. Submit	Hiện thị lỗi "Sai mật khẩu"; không tạo session	High
TC_ACC_007	Quên mật khẩu – gửi mã đặt lại	User	Gửi Email reset mật khẩu	Email tồn tại trong hệ thống	Email hợp lệ	1. Mở form quên mật khẩu 2. Nhập Email 3. Submit	Gửi Email chứa token/OTP đặt lại mật khẩu; trả về 200	High
TC_ACC_008	Đặt lại mật khẩu bằng token hợp lệ	User	Kiểm tra reset mật khẩu	Token còn hạn; user đã yêu cầu reset	Token + mật khẩu mới	1. Mở link reset 2. Nhập mật khẩu mới 3. Submit	Mật khẩu được thay đổi; đăng nhập thành công	High

- Bảng Test Case Kiểm tra gõ phím

Bảng 2.81. Bảng Test Case Kiểm tra gõ phím

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Dữ liệu test	Các bước	Kết quả mong đợi	Ưu tiên
TC_TYP_001	Lấy bài tập ngẫu nhiên (Basic)	User	Kiểm tra lấy đề	Đã đăng nhập	language=en, type=basic	Vào màn test → Click "Bắt đầu"	Trả nội dung Top200; start timer	High
TC_TYP_002	Lưu kết quả nếu WPM đủ điều kiện	User	Lưu kết quả đúng logic	Đã đăng nhập, WPM trước = 50	WPM hiện = 48 (> 50-5)	Hoàn thành test → Lưu	Kết quả lưu vào TYPINGRESULT; UserProgress cập nhật	High
TC_TYP_003	Không lưu khi WPM thấp	User	Bảo vệ không lưu kết quả xấu	Đã đăng nhập	WPM < threshold	Hoàn thành test → Lưu	Hệ thống trả thông báo "không đủ điều kiện", không lưu	Medium
TC_TYP_004	Lưu kết quả khi chưa đăng nhập	Guest	Kiểm tra quyền	Chưa đăng nhập	-	Thực hiện test → Save	Yêu cầu login/hiển thị lỗi; không lưu	High
TC_TYP_005	Tính accuracy đúng	User	Kiểm tra logic tính accuracy	Đã đăng nhập	input/expected	Làm test mẫu, so sánh output	Accuracy = (# correct chars / total) *100 chính xác	High

- Bảng Test Case Luyện văn bản và AI

Bảng 2.82. Bảng Test Case Luyện văn bản và AI

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Dữ liệu	Các bước	Kết quả mong đợi	Ưu tiên
TC_TXT_001	Tạo văn bản thủ công	User	CRUD văn bản	Đã đăng nhập	title, content, language	Vào form tạo, submit	Bản ghi được tạo, trạng thái = pending (nếu cần duyệt)	High
TC_TXT_002	Tạo văn bản bằng AI (Gemini)	User	Tạo nhanh bài bằng API	Đã đăng nhập, API key cấu hình	chủ đề: "lịch sử"	Chọn "AI tạo", nhập chủ đề, submit	Hệ thống gọi Gemini, nhận văn bản, hiển thị preview, lưu temp	High
TC_TXT_003	Gửi văn bản lên hệ thống (chờ duyệt)	User	Kiểm tra luồng duyệt	Tạo văn bản xong	-	Click "Gửi"	Trạng thái = pending; notification admin	Medium
TC_TXT_004	Admin duyệt văn bản thành công	Admin	Duyệt bài người dùng	Có văn bản pending	-	Admin mở list, approve	Văn bản public, xuất hiện danh sách	High
TC_TXT_005	Văn bản chứa từ cấm bị từ chối	System/Admin	Kiểm soát nội dung	Nội dung chứa từ cấm	-	Submit	Hiển thị lỗi/Reject, gửi thông báo cho user	High

- Bảng Test Case Cuộc thi

Bảng 2.83. Bảng Test Case Cuộc thi

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Dữ liệu	Các bước	Kết quả mong đợi	Ưu tiên
TC_CO M_001	Tạo cuộc thi công khai	Teacher /User	Tạo competition	Đã đăng nhập (teacher)	name , start, end, language	Fill form → Submit	Competition tạo, status=open	High
TC_CO M_002	Tham gia bằng joinCode	User	Kiểm tra join flow	JoinCode tồn tại	joinCode	Nhập joinCode → Join	User được thêm; tham gia competition	High
TC_CO M_003	Lưu kết quả cuộc thi	User	Lưu performance	Đã tham gia, hoàn thành test	WPM, accuracy	Hoàn thành test → Submit	Kết quả lưu vào KetQuaCuocThi	High
TC_CO M_004	Xem bảng xếp hạng	User	Hiển thị leaderboard	Competition có kết quả	-	Mở page ranking	Hiển thị list sắp xếp theo WPM	Medium

- Bảng Test Case Quản lý lớp học

Bảng 2.84. Bảng Test Case Quản lý lớp học

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Dữ liệu	Các bước	Kết quả mong đợi	Ưu tiên
TC_EDU_01	Tạo lớp học mới	Teacher	Tạo Class	Đã đăng nhập teacher	TenLop, MaTham Gia	Fill form → Submit	LopHoc tạo; Teacher là owner	High
TC_EDU_002	Thêm học viên bằng Email	Teacher	Invite student	Email user tồn tại	Email	Nhập Email → Gửi invite	InviteToken tạo; Email gửi; user nhận	Medium
TC_EDU_003	Học viên tham gia bằng mã	Student	Join class	Có MaTham Gia	MaTham Gia	Nhập mã → Join	ThanhVienLop record tạo	High
TC_EDU_004	Duyệt yêu cầu tham gia (real-time)	Teacher	Duyệt/Reject	Có request pending	-	Teacher approve via UI	User được thêm; notification gửi	High

- Bảng Test Case Admin

Bảng 2.85. Bảng Test Case Admin

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Dữ liệu	Các bước	Kết quả mong đợi	Ưu tiên
TC_AD MIN_001	Quản lý tài khoản (CRUD)	Admin	Tạo/sửa/xóa/khóa tài khoản	Admin đã login	user data	Vào Admin Panel → Thực hiện	Thay đổi phản ánh DB; log hoạt động	High
TC_AD MIN_002	Duyệt văn bản người dùng	Admin	Kiểm soát nội dung	Có văn bản pending	-	Mở list → Approve/Reject	Văn bản public hoặc bị từ chối; thông báo user	High
TC_AD MIN_003	Quản lý bài luyện hệ thống	Admin	CRUD exercise text	-	exercise data	CRUD	Dữ liệu thay đổi đúng trong hệ thống	Medium
TC_AD MIN_004	Xem báo cáo hệ thống	Admin	Dashboard	Data có sẵn	-	Mở dashboard	Thống kê số user, bài, lớp... hiển thị	Medium

- Bảng Test Case SignalR

Bảng 2.86. Bảng Test Case SignalR

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Các bước	Kết quả mong đợi	Ưu tiên
TC_SIG_001	Kết nối SignalR Hub	User	Kiểm tra realtime connect	User logged	Mở client connect	Kết nối thành công, connection id trả	High
TC_SIG_002	Gửi yêu cầu tham gia lớp realtime	User/Teacher	Gửi request → Teacher nhận	Connect cả 2	User gửi request → Teacher nhận	Teacher nhận ngay notification	High
TC_SIG_003	Cập nhật danh sách online	System	Theo dõi online	Nhiều client	Connect/Disconnect client	Số người online cập nhật	Medium

- Bảng Test Case Bảo mật chung

Bảng 2.87. Bảng Test Case Bảo mật chung

TC_ID	Tên Test Case	Tác nhân	Mục đích	Tiền điều kiện	Các bước	Kết quả mong đợi	Ưu tiên
TC_SEC_001	SQL Injection	Attacker	Kiểm tra bảo vệ SQLi	Endpoint public	Gửi payload SQLi ở input	Hệ thống sanitize/không lỗi DB	High
TC_SEC_002	XSS trong nội dung văn bản	Attacker	Kiểm tra escaping	Submit nội dung chứa script	Submit → View page	Script không chạy; nội dung được escape	High
TC_PERF_001	Load test 100 concurrent	System	Performance under load	Staging env	Chạy JMeter 100 users	Response time < 2s; no errors > 1%	High

3.8.4. Kế hoạch thực hiện

- Giai đoạn 1 (Tuần 1-2): Unit tests (NUnit/xUnit) — mục tiêu 70% coverage.
- Giai đoạn 2 (Tuần 3-4): Integration tests + test DB.
- Giai đoạn 3 (Tuần 5-6): System E2E (Selenium), cross-browser, responsive.
- Giai đoạn 4 (Tuần 7-8): Security (OWASP ZAP) & Performance (JMeter).
- Giai đoạn 5 (Tuần 9-10): UAT & Regression & Bug fixing.

3.8.5. Công cụ và tài nguyên

- Unit: NUnit/xUnit, Moq
- API: Postman, Newman
- E2E: Selenium WebDriver (C#) / Playwright
- Perf: JMeter

- Security: OWASP ZAP, Burp Suite
- CI: GitHub Actions / Azure DevOps
- DB: SQL Server (staging/test), LocalDB/In-Memory for integration

KẾT LUẬN

Trên đây là toàn bộ báo cáo đồ án tốt nghiệp sau một quá trình học tập, nghiên cứu và triển khai của bản thân em dưới sự hướng dẫn trực tiếp của ThS. Nguyễn Tất Phú Cường, đồ án tốt nghiệp với đề tài “Xây dựng website luyện gõ bàn phím sử dụng công nghệ ASP.NET MVC kết hợp Chatbot AI” đã hoàn thành các mục tiêu đề ra ban đầu về việc xây dựng một hệ thống luyện gõ bàn phím hiện đại, thông minh và an toàn. Tổng kết lại toàn bộ quá trình thực hiện đồ án tốt nghiệp em xin được phép trình bày các kết quả đạt được, những hạn chế còn tồn tại và định hướng phát triển trong tương lai của đồ án cụ thể như sau:

1. Kết quả đạt được

Chức năng cốt lõi của website luyện gõ bàn phím đã hoàn thiện trọn vẹn nền tảng luyện gõ bàn phím trực tuyến với đầy đủ các chế độ đa dạng, bao gồm bài kiểm tra cơ bản, nâng cao, tùy chỉnh văn bản và các trò chơi nhỏ giúp tăng hứng thú. Người dùng được hỗ trợ tối đa bởi hệ thống thông kê tiến trình chi tiết và bảng xếp hạng thi đua, những tính năng này giúp tăng trải nghiệm người dùng hơn so với các hệ thống luyện gõ bàn phím khác.

Ở khía cạnh an toàn thông tin, hệ thống đảm bảo quy trình bảo mật chặt chẽ ngay từ khâu đăng ký với tích hợp reCAPTCHA chống người máy, kích hoạt tài khoản qua thư điện tử, lưu trữ mật khẩu an toàn bằng thuật toán băm (hash) BCrypt và hỗ trợ khôi phục mật khẩu qua mã xác thực một lần OTP. Giao diện người dùng cũng đạt được sự thân thiện và trực quan nhờ việc kết hợp thư viện Bootstrap và SweetAlert.

Đặc biệt đồ án đã phát triển thành công nhóm chức năng chuyên sâu về Lớp học và Cuộc thi chung giúp tăng tính tương tác người dùng với nhau. Giáo viên có thể tạo phòng học, quản lý yêu cầu tham gia và xem thống kê kết quả thi một cách dễ dàng, trong khi học viên tham gia nhanh chóng bằng mã hoặc mật khẩu. Công nghệ SignalR đã được ứng dụng hiệu quả để đảm bảo dữ liệu trong lớp học được cập nhật tức thời mà không cần tải lại trang.

Ngoài ra, việc tích hợp trực tiếp trợ lý ảo AI vào website là một điểm sáng cho phép người dùng có thể trao đổi trực tiếp những thắc mắc 24/7 khi cần hỗ trợ và hệ thống sử dụng dữ liệu lỗi gõ cá nhân để tư vấn lộ trình luyện tập phù hợp nhất cho từng người dùng, người dùng có thể dựa vào các bài luyện tập mà trợ lý ảo AI đưa ra để luyện

tập từ đó giúp cải thiện tối đa kỹ năng luyện gõ bàn phím và tăng tính tương tác giữa người dùng và website cũng như là tối ưu hóa trải nghiệm người dùng một cách cao nhất.

2. Hạn chế

Bên cạnh những thành công đạt được trong bước đầu hệ thống vẫn còn tồn tại một số hạn chế về kỹ thuật và nghiệp vụ cần được khắc phục:

Bảo mật cấu hình các thông tin nhạy cảm như tài khoản gửi thư điện tử SMTP và khóa bí mật API Gemini hiện vẫn đang được nhúng cứng trong mã nguồn thay vì sử dụng cơ chế quản lý bảo mật chuyên dụng. Hệ thống cũng chưa thiết lập các giới hạn về số lần gửi thư điện tử hay mã OTP, đồng thời chưa có cơ chế hết hạn mã thông báo rõ ràng, dẫn đến nguy cơ tiềm ẩn về việc bị lợi dụng để tấn công gửi thư rác.

Công nghệ tính năng thời gian thực sử dụng SignalR mới chỉ dừng lại ở việc hiển thị số lượng kết nối qua giao diện dòng lệnh của trình duyệt mà chưa trực quan hóa số người trực tuyến thực tế lên giao diện người dùng cũng như thiếu cơ chế ghi nhật ký hoạt động nâng cao. Mã nguồn hiện tại phụ thuộc khá nhiều vào các công nghệ cũ như jQuery và Razor chưa có các bài kiểm tra đơn vị hay kiểm tra tự động gây khó khăn cho việc bảo trì và mở rộng sang các nền tảng giao diện hiện đại.

Chức năng chatbot AI chỉ mới lưu ở localstorage chưa lưu được cơ sở dữ liệu ở trong cơ sở dữ liệu SQL Server.

Về nghiệp vụ, module đặt lịch theo trạng thái như dự kiến ban đầu chưa được triển khai trong phiên bản này nên nghiệp vụ đặt lịch chưa được triển khai.

3. Hướng phát triển

Dựa trên kết quả và các hạn chế đã phân tích ở phía trên em xin được phép đề xuất các hướng phát triển cụ thể để hoàn thiện sản phẩm trong tương lai như sau:

Ưu tiên hàng đầu là nâng cấp bảo mật bằng cách tách biệt các cấu hình nhạy cảm sang các tệp quản lý chuyên biệt như appsettings hoặc quản lý các khóa bí mật (Secret Manager), đồng thời bổ sung cơ chế làm mới mã thông báo và giới hạn tần suất gửi thư điện tử để ngăn chặn tấn công.

Về mặt tính năng hệ thống sẽ mở rộng khả năng thời gian thực để hiển thị danh sách người trực tuyến trực quan hơn, tích hợp khung trò chuyện trong lớp học và hệ thống thông báo tức thời khi giáo viên duyệt yêu cầu.

Còn phần công nghệ định hướng dài hạn là nâng cấp giao diện sang các thư viện

hiện đại như React hoặc Vue, hoặc chuyển đổi sang nền tảng ASP.NET Core để dễ dàng bảo trì và hỗ trợ ứng dụng đa nền tảng, nâng cao tính trải nghiệm người dùng.

Trong giai đoạn phát triển tiếp theo hệ thống sẽ triển khai thu phí bằng cách người dùng sẽ được sử dụng miễn phí các tính năng cơ bản kèm theo quảng cáo ở màn hình, người dùng sẽ cần trả phí thuê bao để truy cập không giới hạn các tính năng cao cấp tiêu tốn tài nguyên như tạo văn bản bằng AI, chatbot AI và mở khóa tính năng tạo lớp học.

Thực hiện cải tiến nâng cấp chatbot AI hiện đại hơn, lưu trữ cơ sở dữ liệu chuyên sâu và bảo mật hơn có thể dùng SQL Server để lưu trữ cơ sở dữ liệu.

Cuối cùng, việc bổ sung các bài kiểm tra cho các lớp dịch vụ quan trọng và tích hợp công cụ giám sát lỗi chi tiết sẽ được thực hiện để đảm bảo hệ thống vận hành ổn định khi triển khai thực tế, đồng thời xem xét bổ sung module đặt lịch nếu nhu cầu nghiệp vụ yêu cầu. Viết test cho controller/service quan trọng, thêm monitoring & logging chi tiết để theo dõi lỗi khi triển khai thực tế.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Visual Studio Microsoft. “Visual Studio IDE”.
URL: <https://visualstudio.microsoft.com/> (Truy cập: 01/10/2025).
2. Microsoft.NET. “ASP.NET MVC Pattern”.
URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>
(Truy cập 1/10/2025)
3. Microsoft Learn. “C# language documentation”.
URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
(Truy cập: 03/11/2025).
4. Microsoft Learn. “SQL Server Management Studio (SSMS)”.
URL: <https://learn.microsoft.com/en-us/ssms/> (Truy cập: 10/03/2025).
5. Typingtop. “Tập gõ bàn phím, kiểm tra tốc độ đánh máy”.
URL: <https://typingtop.com/vietnamese> (Truy cập: 15/10/2025).
6. YouTube. “Asp.net sql server database connection tutorial”.
URL: <https://youtu.be/yBsl84hDtUg> (Truy cập: 01/11/2025).
7. GitHub. “aspnet-mvc-angular-crud-example”.
URL: <https://github.com/oktadev/aspnet-mvc-angular-crud-example>
(Truy cập: 08/11/2025).
8. Bootstrap. “Bootstrap 3.4 Documentation”.
URL: <https://getbootstrap.com/docs/3.4/> (Truy cập: 20/11/2025).
9. w3schools. “JavaScript Tutorial”
URL: <https://www.w3schools.com/js/> (Truy cập: 20/11/2025).
10. Stack overflow. “How to display user input immediately in a chatbox UI C#”
URL: <https://stackoverflow.com/questions/78962417/>
(Truy cập: 22/11/2025).

11. Microsoft.com. “Real-time ASP.NET with SignalR”

URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/signalr>

(Truy cập: 15/11/2025).

12. Microsoft Learn. “LINQ to SQL”

URL: <https://learn.microsoft.com/enus/dotnet/framework/data/adonet/sql/linq/>

(Truy cập: 16/11/2025).

13. Microsoft Learn. “ADO.NET”

URL: <https://learn.microsoft.com/vi-vn/dotnet/framework/data/adonet/>

(Truy cập: 16/11/2025).

---Hết---