

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



ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG ĐẶT SÂN
THỂ THAO ONLINE

Sinh viên thực hiện : **HỒ THANH NHA**
Mã sinh viên : **2151220160**
Lớp : **21CT6**
Người hướng dẫn : **TS. LÂM TÙNG GIANG**

Đà Nẵng, tháng 12/2025

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



ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG ĐẶT SÂN
THỂ THAO ONLINE

Sinh viên thực hiện : **HỒ THANH NHA**
Mã sinh viên : **2151220160**
Lớp : **21CT6**
Người hướng dẫn : **TS. LÂM TÙNG GIANG**

Đà Nẵng, tháng 12/2025

[illegible]

Giảng viên hướng dẫn

This image shows a full page of a document template. It consists of approximately 20 evenly spaced, thin grey horizontal lines running across the width of the page. The background is white, and there are no margins, text, or other markings present.

Đà Nẵng, ngày tháng năm 2025
Giảng viên phản biện

TÓM TẮT

Tên đề tài: Xây dựng hệ thống đặt sân thể thao online

Sinh viên thực hiện: Hồ Thanh Nha

Mã số SV: 2151220160 Lớp: 21CT6

Đề tài tập trung hướng đến việc xây dựng một hệ thống quản lý đặt sân thể thao trực tuyến nhằm hỗ trợ các cơ sở cho thuê sân tự động hóa quá trình đặt sân và nâng cao trải nghiệm người dùng. Hệ thống cho phép khách hàng tra cứu thông tin sân, xem chi tiết loại sân, lựa chọn khung giờ đặt (gồm: ngày chơi và giờ chơi), chọn sân mong muốn và thanh toán trực tuyến một cách nhanh chóng.

1. Khách vãng lai.

- Khách vãng lai tìm kiếm, xem thông tin sân, xem giá sân bãi , xem các khuyến mãi của sân, và có thể đăng kí tài khoản mới để đặt sân.

2. Khách hàng

- Khách hàng nhập thông tin cá nhân để đăng kí/dăng nhập tài khoản đặt sân.
- Khách hàng đặt sân mình mong muốn.
- Khách hàng tìm kiếm, xem thông tin sân, xem giá mỗi loại sân, và đặt sân phù hợp với nhu cầu sử dụng...
- Tra cứu sân đã đặt.
- Quản lí tài khoản cá nhân.

3. Quản trị viên

- Quản trị viên(Admin): Có quyền thêm sửa xóa khuyến mãi, loại sân, và lịch sử của khách hàng, quản lý tài khoản, quản lý đơn đặt sân.
- Xem được doanh thu.
- Quản lí mã giảm giá.

Đề tài áp dụng công nghệ web ReactJS (frontend), NodeJS và ExpressJS (backend) kết hợp cơ sở dữ liệu MongoDB nhằm đảm bảo tính chính xác, an toàn và tốc độ xử lý cao. Việc xây dựng hệ thống không chỉ giúp giảm tải nhân sự, hạn chế sai sót trong quá trình đặt sân thủ công mà còn mang đến sự tiện lợi, minh bạch và chuyên nghiệp cho cả người dùng lẫn đơn vị vận hành sân thể thao

LỜI NÓI ĐẦU

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc và lòng tôn trọng đến tất cả quý thầy cô giáo đã đồng hành và hỗ trợ em trong suốt quá trình thực hiện đề tài này. Sự giúp đỡ, sự chỉ dẫn và kiến thức quý báu mà quý thầy cô đã chia sẻ với em là điều không thể định giá được và đã góp phần lớn trong thành công của đồ án này.

Đặc biệt, em muốn bày tỏ lòng biết ơn chân thành đến **TS. Lâm Tùng Giang** người đã tận tình và không ngừng động viên, hỗ trợ em trong suốt quá trình thực hiện đề tài tốt nghiệp này. Nhờ những kiến thức và kinh nghiệm của thầy, em đã học hỏi được nhiều điều quý giá và trải qua một quá trình học tập và nghiên cứu thực sự đáng nhớ.

Đồng thời, em xin gửi lời cảm ơn chân thành tới toàn thể quý thầy cô thuộc khoa Công Nghệ Thông Tin _ Trường Đại Học Kiến Trúc Đà Nẵng. Suốt gần 5 năm học tập tại trường, em đã được trang bị những kiến thức chuyên môn, những kỹ năng quan trọng và cả những lời khuyên quý báu từ quý thầy cô. Đặc biệt, em muốn gửi lời cảm ơn đến **TS. Lâm Tùng Giang** thầy đã trở thành nguồn động viên và nguồn sự hỗ trợ vô cùng quan trọng trong quá trình học tập và công việc của em.

Tuy nhiên trong quá trình thực hiện đề tài này, do kiến thức và thời gian có hạn, em không thể tránh khỏi những sai sót. Vì vậy, em mong quý thầy, cô thông cảm và góp ý để em có thể hoàn thiện hơn và tránh được những sai lầm trong tương lai.

CAM ĐOAN

Tôi Hồ Thanh Nha sinh viên khóa 2021, chuyên ngành Công nghệ thông tin, xin cam đoan rằng:

1. Đồ án tốt nghiệp này là kết quả nghiên cứu và thực hiện của riêng tôi, được tiến hành trên cơ sở kiến thức, tư duy khoa học và sự hướng dẫn của giảng viên hướng dẫn.
2. Đồ án không sao chép hoặc đạo văn từ bất kỳ công trình, tài liệu, hay nguồn thông tin nào khác mà không có trích dẫn, tham khảo rõ ràng và hợp lệ.
3. Các nguồn tài liệu, dữ liệu, và thông tin sử dụng trong quá trình thực hiện đồ án đều được chọn lọc từ những nguồn chính thống, đáng tin cậy và tuân thủ quy định về bản quyền.
4. Các số liệu, kết quả nghiên cứu và kết luận trong đồ án là trung thực, chính xác, không bị chỉnh sửa, bóp méo hay làm sai lệch kết quả nghiên cứu.
5. Tôi nhận thức rõ tầm quan trọng của liêm chính học thuật và chịu trách nhiệm hoàn toàn trước nhà trường, giảng viên hướng dẫn và pháp luật về tính trung thực trong nội dung của đồ án.

Tôi xin cam kết thực hiện nghiêm túc các quy định của nhà trường về đạo đức và liêm chính học thuật, và chịu hoàn toàn trách nhiệm nếu có bất kỳ vi phạm nào xảy ra.

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

Hồ Thanh Nha

Mục Lục

Tóm Tắt	
Lời Nói Đầu Và Cảm Ơn	i
Cam Đoan	ii
Mục Lục	iii
Danh Sách Các Bảng Và Hình Vẽ	iv
Danh Sách Ký Hiệu, Chữ Viết Tắt	v
Chương 1: TỔNG QUAN VỀ ĐỀ TÀI	1
1.1 Mục đích thực hiện đề tài	1
1.2 Đối tượng nghiên cứu	2
1.2.1 Khách vãng lai (chưa đăng ký tài khoản)	2
1.2.2 Khách hàng (User đã đăng kí)	2
1.2.3 Quản trị viên (admin)	2
1.3 Phạm vi nghiên cứu	3
1.4 Phương pháp nghiên cứu	4
1.4.1 Nghiên cứu tài liệu	4
1.4.2 Phân tích và thiết kế hệ thống	4
1.4.3 Phương pháp thực nghiệm	4
1.4.4 Phương pháp kiểm thử	4
Chương 2: CƠ SỞ LÝ THUYẾT	5
2.1 Giới thiệu mô hình	5
2.1.1 Mô hình Client – Server trong phát triển website	5
2.1.2 Vai trò và chức năng của Client	6
2.1.3 Vai trò và chức năng của Server	6
2.1.4 Ưu điểm của mô hình Client – Server	7
2.2 Giới thiệu công nghệ	7
2.2.1 Giới thiệu về NodeJS	7
2.2.2 Ưu điểm của Nodejs	8
2.2.3 Nhược điểm của Nodejs	9
2.3 Giới thiệu về Reacjs	9
2.3.1 Ưu điểm của Reacjs	10
2.3.2 Nhược điểm của Reacjs	10
2.4 Giới thiệu về Visual studio code	11
2.4.1 Ưu điểm của Visual studio code	12
2.4.2 Nhược điểm của Visual studio code	13
2.5 Giới thiệu về hệ quản trị cơ sở dữ liệu MongoDB	13
2.5.1 Ưu điểm của hệ quản trị cơ sở dữ liệu MongoDB	14
2.5.2 Nhược điểm của hệ quản trị cơ sở dữ liệu MongoDB	15
Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	16
3.1 Mô tả bài toán	16
3.1.1 Bối cảnh và mục tiêu	16
3.1.2 Nhóm người dùng	16
3.1.3 Quy trình nghiệp vụ	18
3.1.4 Các yêu cầu của hệ thống	18
3.2 Mô hình cơ sở dữ liệu	19
3.2.1 Mô hình phân rã chức năng(Business functional diagram)	19
3.2.2 Biểu đồ dòng dữ liệu	23
3.2.3 Biểu đồ usecase	30

3.2.4 Kịch bản hoạt động chức năng	33
3.2.5 Sơ đồ hoạt động chức năng	40
3.2.6 Biểu đồ tuần tự	45
3.3 Sơ đồ lớp.....	52
3.4 Thiết kế cơ sở dữ liệu	52
3.4.1 Sơ đồ ERD.....	52
3.4.1 Tất Cả các bảng	53
Chương 4: KẾT QUẢ THỰC NGHIỆM.....	61
4.1 Xây dựng website demo	61
4.1.1 Giao diện trang chủ website	61
4.1.2 Giao diện trang đăng nhập.....	61
4.1.3 Giao diện trang đăng ký.....	63
4.1.4 Giao diện trang quên mật khẩu.....	64
4.1.5 Giao diện trang đặt sân	65
4.1.6 Giao diện trang chi tiết sân	65
4.1.7 Giao diện trang đặt sân	66
4.1.8 Giao diện trang xác nhận thanh toán	67
4.1.9 Giao diện thanh toán trực tuyến momo	68
4.1.10 Giao diện trang khuyến mãi.....	69
4.1.11 Giao diện trang lịch sử hủy sân	69
4.1.12 Giao diện trang cập nhập lại mật khẩu	70
4.1.13 Giao diện trang chủ admin.....	71
4.1.14 Giao diện trang quản lý tài khoản.....	72
4.1.15 Giao diện trang quản lý danh sách sân	73
4.1.16 Giao diện trang thêm sân.....	74
4.1.17 Giao diện trang chỉnh sửa sân.....	75
4.1.18 Giao diện trang quản lý đánh giá.....	76
4.1.19 Giao diện trang thống kê doanh thu	77
4.1.20 Giao diện trang thêm mới khuyến mãi	77
4.1.21 Giao diện trang xem lịch sử hủy	78
4.1.22 Giao diện trang quản lý thông tin nổi bật và khuyến mãi.....	79
Kết luận và đề xuất	81
1 Kết luận	81
2 Đề xuất.....	82
Tài liệu tham khảo	83

Danh Sách Các Bảng Và Hình Vẽ

Danh Mục Hình Vẽ

Hình 2.1 Mô Hình Client - Server	5
Hình 2.2 Logo Nodejs	7
Hình 2.3 Logo reacjs	9
Hình 2.4 Logo visual studio code	11
Hình 2.5 Logo MongoB	13
Hình 3.1 Sơ đồ phân rã chức năng	19
Hình 3.2 Sơ đồ ngữ cảnh	24
Hình 3.3 Biểu đồ phân rã chức năng quản lý tìm kiếm đặt hàng	26
Hình 3.4 Biểu đồ phân rã chức năng quản lý sân	27
Hình 3.5 Biểu đồ phân rã chức năng quản lý khách hàng	28
Hình 3.6 Biểu đồ phân rã chức năng quản lý đơn đặt	29
Hình 3.7 Biểu đồ phân rã chức năng thống kê báo cáo	30
Hình 3.8 Biểu đồ usecase tổng quát	31
Hình 3.9 Biểu đồ usecase tác nhân khách vắng lai.....	32
Hình 3.10 Biểu đồ usecase tác nhân người dùng.....	32
Hình 3.11 Biểu đồ usecase tác nhân quản trị viên.....	33
Hình 3.12 Sơ đồ hoạt động chức năng đăng ký.....	40
Hình 3.13 Sơ đồ hoạt động chức năng đăng nhập.....	40
Hình 3.14 Sơ đồ hoạt động chức năng thay đổi mật khẩu.....	41
Hình 3.15 Sơ đồ hoạt động chức năng tìm kiếm	41
Hình 3.16 Sơ đồ hoạt động chức năng đánh giá và bình luận	42
Hình 3.17 Sơ đồ hoạt động chức năng Lịch sử đặt	43
Hình 3.18 Sơ đồ hoạt động chức năng đặt sân	43
Hình 3.19 Sơ đồ hoạt động chức năng Khuyến mãi.....	44
Hình 3.20 Sơ đồ hoạt động chức năng quản lý sân	45
Hình 3.21 Sơ đồ tuần tự chức năng đăng ký	45
Hình 3.22 Sơ đồ tuần tự chức năng đăng nhập.....	46
Hình 3.23 Sơ đồ tuần tự chức năng thay đổi mật khẩu	46
Hình 3.24 Sơ đồ tuần tự chức năng tìm kiếm.....	47
Hình 3.25 Sơ đồ tuần tự chức năng đánh giá và bình luận.....	47
Hình 3.26 Sơ đồ tuần tự chức năng hủy sân.....	47
Hình 3.27 Sơ đồ tuần tự chức năng đặt sân	48
Hình 3.28 Sơ đồ tuần tự chức năng xóa tài khoản.....	48
Hình 3.29 Sơ đồ tuần tự chức năng thêm Khuyến mãi	49
Hình 3.30 Sơ đồ tuần tự chức năng sửa Khuyến mãi	49
Hình 3.31 Sơ đồ tuần tự chức năng xóa khuyến mãi.....	50
Hình 3.32 Sơ đồ tuần tự chức năng thêm sân.....	51
Hình 3.33 Sơ đồ tuần tự chức năng sửa sân	51
Hình 3.34 Sơ đồ tuần tự chức năng xóa sân	51
Hình 3.35 Sơ đồ lớp.....	52
Hình 3.36 Sơ đồ ERD.....	53
Hình 4.1 Giao diện trang chủ khách hàng	61
Hình 4.2 Giao diện trang đăng nhập.....	62
Hình 4.3 source code trang đăng nhập	62
Hình 4.4 Giao diện trang đăng ký.....	63
Hình 4.5 source code trang đăng kí	63

Hình 4.6 Giao diện trang quên mật khẩu.....	64
Hình 4.7 source code trang quên mật khẩu	64
Hình 4.8 Giao diện trang đặt sân	65
Hình 4.9 Giao diện trang chi tiết sân.....	66
Hình 4.10 Giao diện trang đặt sân	66
Hình 4.11 source code trang đặt sân.....	66
Hình 4.12 Giao diện trang xác nhận thanh toán	67
Hình 4.13 source code trang thanh toán	67
Hình 4.14 Giao diện trang thanh toán trực tuyến momo	68
Hình 4.15 Giao diện trang khuyến mãi.....	69
Hình 4.16 Giao diện trang lịch sử hủy sân	70
Hình 4.17 source code trang hủy sân.....	70
Hình 4.18 Giao diện trang cập nhập mật khẩu	70
Hình 4.19 source code trang lấy lại mật khẩu	71
Hình 4.20 Giao diện trang chủ admin.....	72
Hình 4.21 Giao diện trang quản lý tài khoản.....	72
Hình 4.22 source code trang quản lý tài khoản	73
Hình 4.23 Giao diện trang quản lý danh sách sân	73
Hình 4.24 source code trang danh sách sân.....	74
Hình 4.25 Giao diện trang thêm sân.....	74
Hình 4.26 source code trang thêm sân.....	74
Hình 4.27 Giao diện tranh chỉnh sửa sân.....	75
Hình 4.28 source code trang chỉnh sửa sân	75
Hình 4.29 Giao diện trang quản lý đánh giá.....	76
Hình 4.30 source code trang đánh giá	76
Hình 4.31 Giao diện trang thống kê doanh thu.....	77
Hình 4.32 Giao diện trang thêm mới khuyến mãi	78
Hình 4.33 source code trang tạo khuyến mãi	78
Hình 4.34 Giao diện trang xem lịch sử hủy.....	79
Hình 4.35 source code trang lịch sử hủy	79
Hình 4.36 Giao diện trang quản lý thông tin nổi bật và khuyến mãi.....	80

Danh Mục Bảng Biểu

Bảng 1 Kịch bản chức năng đăng ký.....	33
Bảng 2 Kịch bản chức năng đăng nhập	34
Bảng 3 Kịch bản chức năng tìm kiếm	34
Bảng 4 Kịch bản chức năng đánh giá và bình luận	35
Bảng 5 Kịch bản chức năng lịch sử đặt	36
Bảng 6 Kịch bản chức năng đặt sân	36
Bảng 7 Kịch bản chức năng quản lý tài khoản	37
Bảng 8 Kịch bản chức năng quản lý khuyến mãi	38
Bảng 9 Kịch bản chức năng quản lý sân	39
Bảng 10 Tất cả các bảng trong cơ sở dữ liệu.....	53
Bảng 11 Bảng user.....	54
Bảng 12 Bảng LoyaltyPoint	54
Bảng 13 FootballField	54
Bảng 14 Bảng FootballTimeSlot	55
Bảng 15 Bảng BasketballField	55
Bảng 16 Bảng BasketballTimeSlot.....	55
Bảng 17 Bảng TennisField	56

Bảng 18 Bảng TennisTimeSlot.....	56
Bảng 19 Bảng Field	57
Bảng 20 Bảng Booking	57
Bảng 21 Bảng CancelledBooking	57
Bảng 22 Bảng Payment	58
Bảng 23 Bảng Promotion	58
Bảng 24 Bảng Review	59
Bảng 25 Bảng phân tích các mối quan hệ	60

Danh Sách Ký Hiệu, Chữ Viết Tắt

Stt	Chữ viết tắt	Giải nghĩa	Nghĩa tiếng Việt
1	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
2	DOM	Document Object Model	Mô hình đối tượng tài liệu
3	API	Application Programming Interface	Các phương thức, giao thức kết nối với các thư viện và ứng dụng khác
4	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
5	RAM	Random Access Memory	Bộ nhớ khả biến cho phép đọc-ghi
6	ERD	Entity-Relationship Diagram	Sơ đồ mô hình thực thể quan hệ
7	CPU	Central Processing Unit	Bộ xử lý trung tâm
8	SOCKET	Network Socket	Socket hoạt động như một cổng kết nối cho phép server và client gửi

Chương 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Mục đích thực hiện đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh và nhu cầu trải nghiệm dịch vụ trực tuyến ngày càng tăng, lĩnh vực thể thao – giải trí cũng ghi nhận sự chuyển dịch mạnh mẽ sang môi trường số. Việc đặt sân thể thao theo hình thức truyền thống thường gặp nhiều hạn chế như: thiếu thông tin về sân, khó kiểm tra tình trạng trống, tốn thời gian liên hệ qua điện thoại, dễ xảy ra sai sót trong quá trình ghi nhận đặt sân và khó quản lý lịch đặt khi lượng khách tăng cao. Điều này đặt ra nhu cầu cấp thiết về một hệ thống đặt sân trực tuyến hiện đại, tiện lợi và chính xác.

Trước thực trạng đó, đề tài “Xây dựng hệ thống đặt sân thể thao trực tuyến sử dụng ReactJS và NodeJS” được thực hiện nhằm xây dựng một nền tảng giúp khách hàng dễ dàng tra cứu thông tin sân, xem khuyến mãi, chọn khung giờ phù hợp và đặt sân nhanh chóng. Về phía nhà quản lý, hệ thống hỗ trợ quản lý sân, khách hàng, khuyến mãi, đánh giá và doanh thu một cách trực quan, khoa học.

Mục đích chính của đề tài là phát triển một hệ thống đặt sân thể thao online hiện đại – minh bạch – dễ sử dụng, đáp ứng nhu cầu đặt sân mọi lúc mọi nơi của khách hàng, đồng thời hỗ trợ cơ sở kinh doanh tối ưu hóa quy trình vận hành, giảm sai sót và nâng cao hiệu quả kinh doanh.

Hệ thống mang mục tiêu nâng cao trải nghiệm người dùng thông qua giao diện trực quan, khả năng phản hồi nhanh, xử lý thông tin thời gian thực, và hỗ trợ đặt sân một cách chủ động – thuận tiện. Bên cạnh đó, hệ thống cũng đem lại một quy trình quản lý khép kín dành cho admin nhằm cải thiện khả năng giám sát và điều hành.

Mục tiêu cụ thể của đề tài

- Xây dựng website đặt sân thể thao online cho phép khách truy cập xem thông tin sân, xem khuyến mãi, và thực hiện đặt sân nhanh chóng sau khi đăng ký tài khoản.
- Ứng dụng ReactJS cho frontend nhằm xây dựng giao diện tương tác, tốc độ cao và thân thiện với người dùng trên cả PC, tablet và smartphone.
- Sử dụng NodeJS – ExpressJS để xây dựng backend xử lý logic nghiệp vụ, quản lý yêu cầu đặt sân, đánh giá, khuyến mãi, tài khoản và doanh thu.
- Lưu trữ dữ liệu bằng MongoDB, đảm bảo khả năng mở rộng, tốc độ truy vấn tốt và phù hợp với hệ thống có dữ liệu linh hoạt.

- Xây dựng các chức năng quản trị: quản lý sân, lịch đặt, khách hàng, khuyến mãi, đánh giá và thống kê doanh thu giúp admin vận hành hệ thống hiệu quả.
- Tối ưu hóa trải nghiệm người dùng bằng giao diện trực quan, quy trình đặt sân mạch lạc, bảo mật thông tin và phân quyền truy cập rõ ràng.

1.2 Đối tượng nghiên cứu

Hệ thống được xây dựng hướng đến 3 nhóm tác nhân chính:

1.2.1 Khách vắng lai (chưa đăng ký tài khoản)

Có thể thực hiện:

- Xem danh sách sân thể thao (bóng đá, bóng rổ, tennis...)
- Xem chi tiết sân, bảng giá, tiện ích
- Xem các chương trình khuyến mãi
- Xem đánh giá từ khách hàng khác

Không có quyền đặt sân cho đến khi đăng ký và đăng nhập.

1.2.2 Khách hàng (user đã đăng ký)

Có thể thực hiện:

- Đăng ký, đăng nhập, quản lý thông tin cá nhân
- Xem danh sách sân, xem khuyến mãi, lựa chọn ngày và giờ đặt sân
- Đặt sân, thanh toán online
- Xem lịch sử đặt sân
- Đánh giá sân sau khi sử dụng dịch vụ
- Nhận thông báo xác nhận đặt sân

1.2.3 Quản trị viên (admin)

Có thể thực hiện:

- Quản lý thông tin sân (thêm/sửa/xóa sân, khung giờ, giá thuê)
- Quản lý khách hàng
- Quản lý khuyến mãi
- Quản lý đánh giá, xử lý các phản hồi
- Xem thống kê doanh thu theo ngày/tháng/sân
- Quản lý đơn đặt sân, kiểm tra lịch trống

Công nghệ sử dụng

- ReactJS: xây dựng giao diện web, tăng tính tương tác, tối ưu UX.
- NodeJS – ExpressJS: xây dựng server backend, xử lý API, xác thực người dùng.
- MongoDB: lưu trữ các dữ liệu liên quan đến sân, khuyến mãi, người dùng, lịch đặt và đánh giá.

1.3 Phạm vi nghiên cứu

Đề tài tập trung nghiên cứu và xây dựng hệ thống đặt sân thể thao cho mô hình **cơ sở cho thuê sân quy mô vừa và nhỏ**, với các phạm vi sau:

- Phân tích các hệ thống đặt sân trực tuyến hiện có: Sân 5, Sân Nhanh, Sân Bóng 24h để tham khảo giao diện, quy trình và hạn chế.
- Nghiên cứu mô hình responsive design giúp website sử dụng tốt trên nhiều thiết bị.
- Phát triển các chức năng chính:

Hệ thống dành cho khách hàng

- Đăng ký, đăng nhập, cập nhật thông tin cá nhân
- Xem thông tin sân, tiện ích, giá
- Xem khuyến mãi
- Đặt sân theo ngày – giờ
- Đánh giá sân
- Xem lịch sử đặt sân

Hệ thống dành cho admin

- Thêm/sửa/xóa sân thể thao
- Quản lý lịch đặt sân và khung giờ trống
- Quản lý khách hàng
- Quản lý khuyến mãi
- Xem đánh giá và xử lý vi phạm
- Thống kê doanh thu theo sân/khung giờ

Đề tài không tập trung phát triển tích hợp cổng thanh toán quốc tế mà chỉ xây dựng ở mức cơ bản (ví dụ: thanh toán khi đến sân hoặc mô phỏng cổng thanh toán). Chưa triển khai các chuẩn bảo mật nâng cao như OTP, mã hóa nâng cấp hoặc hệ thống chống bot.

1.4 Phương pháp nghiên cứu

1.4.1 Nghiên cứu tài liệu

- Tham khảo các hệ thống đặt sân hiện có để phân tích quy trình nghiệp vụ thực tế.
- Nghiên cứu công nghệ ReactJS, NodeJS, MongoDB, ZocKet, Nodemailer
- Tìm hiểu các mô hình UI/UX phù hợp với dịch vụ đặt sân.

1.4.2 Phân tích và thiết kế hệ thống

- Xác định yêu cầu chức năng và phi chức năng cho 3 tác nhân.
- Mô hình hóa hệ thống bằng:
- Use Case Diagram (quyền của khách vắng lai – khách hàng – admin)
- Activity Diagram (quy trình đặt sân)
- Sequence Diagram (giao tiếp client–server khi đặt sân)
- ERD/MongoDB Document Schema (sân, người dùng, khuyến mãi, lịch đặt)

1.4.3 Phương pháp thực nghiệm

- Xây dựng giao diện web bằng ReactJS.
- Xây dựng server API bằng NodeJS – ExpressJS.
- Thiết kế cấu trúc collection trong MongoDB.

1.4.4 Phương pháp kiểm thử

- Kiểm thử chức năng: đăng nhập, đặt sân, đánh giá, quản lý sân.
- Kiểm thử giao diện: độ trực quan, dễ sử dụng, tương thích nhiều thiết bị.
- Kiểm thử hiệu năng: tốc độ tải trang, phản hồi API, truy vấn MongoDB.
- Kiểm thử với người dùng thử nghiệm (beta test) để đánh giá UX.

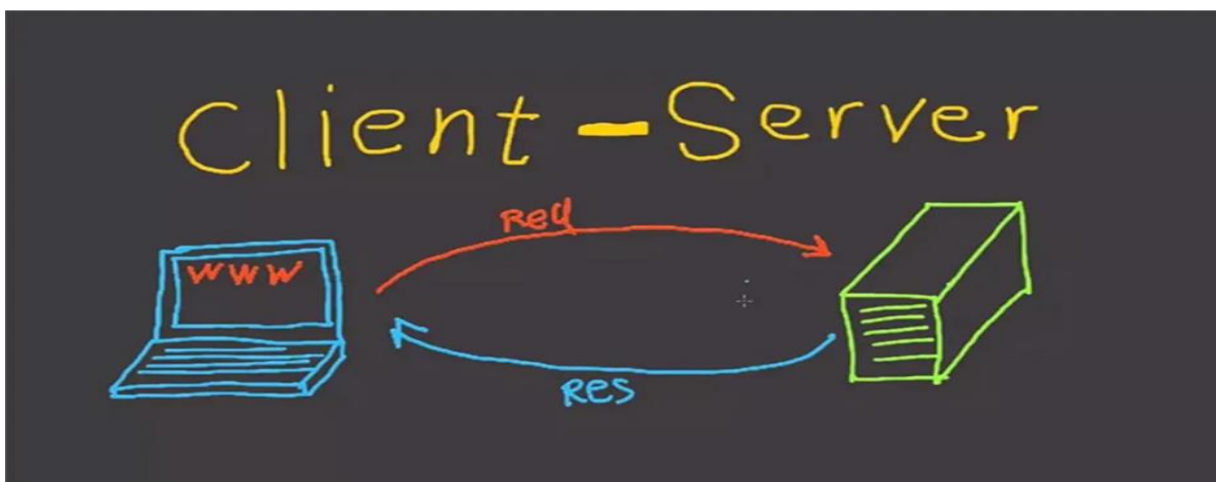
Chương 2: CƠ SỞ LÝ THUYẾT

Để xây dựng một hệ thống website đặt sân thể thao trực tuyến hoàn chỉnh và hoạt động hiệu quả, việc nghiên cứu và vận dụng các cơ sở lý thuyết phù hợp là yếu tố vô cùng quan trọng. Các kiến thức lý thuyết không chỉ cung cấp nền tảng khoa học – công nghệ cho quá trình phát triển hệ thống mà còn giúp định hướng trong việc giải quyết các vấn đề thực tiễn nảy sinh trong quá trình triển khai. Trong phần này, chúng ta sẽ tìm hiểu các kiến thức nền tảng liên quan như công nghệ lập trình web, phương pháp thiết kế giao diện người dùng, mô hình thiết kế và quản lý cơ sở dữ liệu, cũng như các nguyên lý về thương mại dịch vụ trực tuyến. Đây chính là cơ sở quan trọng để đảm bảo hệ thống được phát triển tối ưu, mang lại trải nghiệm thuận tiện cho người dùng và đáp ứng đúng nhu cầu thực tế.

2.1 Giới thiệu mô hình

2.1.1 Mô hình Client – Server trong phát triển website

Trong lĩnh vực phát triển phần mềm nói chung và lập trình web nói riêng, mô hình Client – Server là một kiến trúc cơ bản và phổ biến nhất được sử dụng để xây dựng các ứng dụng web hiện đại. Kiến trúc này phân chia rõ ràng vai trò giữa hai thành phần chính: phía Client (máy khách) và Server (máy chủ), giúp hệ thống hoạt động hiệu quả, linh hoạt và dễ dàng mở rộng. Client (máy khách): Là phía người dùng tương tác trực tiếp, thường là trình duyệt web. Client gửi yêu cầu (request) tới Server thông qua các giao thức như HTTP hoặc HTTPS. Giao diện này có thể được phát triển bằng các công nghệ như HTML, CSS, JavaScript, và các thư viện/framework như React, Vue hoặc Angular.



Hình 2.1 Mô Hình Client - Server

Mô hình Client – Server là một mô hình phân tán, trong đó Client là bên gửi yêu cầu dịch vụ và Server là bên cung cấp dịch vụ đó. Khi người dùng thao tác trên một website (ví dụ như xem phim, đăng nhập, đặt hàng), các hành động đó sẽ được gửi dưới dạng yêu cầu HTTP tới Server. Sau khi xử lý yêu cầu, Server sẽ gửi phản hồi về cho Client, thường dưới dạng dữ liệu JSON hoặc HTML, và dữ liệu này sẽ được Client hiển thị cho người dùng.

Đặc điểm của mô hình này là Client không trực tiếp truy cập vào cơ sở dữ liệu, mà toàn bộ tương tác với dữ liệu đều phải thông qua Server, đảm bảo tính bảo mật và quản lý dữ liệu tập trung.

2.1.2 Vai trò và chức năng của Client

Client là phần giao diện người dùng (UI – User Interface), nơi người dùng tương tác trực tiếp với hệ thống. Trong chuyên đề này, phần Client được xây dựng bằng React.js một thư viện JavaScript mạnh mẽ chuyên dùng để xây dựng các giao diện người dùng tương tác nhanh và mượt mà.

Các chức năng chính của Client bao gồm:

- Hiển thị giao diện thân thiện, dễ sử dụng.
- Gửi các yêu cầu HTTP đến Server thông qua các phương thức như GET, POST, PUT, DELETE.
- Nhận dữ liệu phản hồi từ Server và cập nhật giao diện tương ứng.
- Thực hiện điều hướng trang, quản lý trạng thái giao diện (state), và xử lý sự kiện người dùng.

Với việc sử dụng công nghệ Single Page Application (SPA), ứng dụng React chỉ tải một lần duy nhất và sau đó cập nhật nội dung động mà không cần tải lại toàn bộ trang, mang đến trải nghiệm mượt mà cho người dùng.

2.1.3 Vai trò và chức năng của Server

Server là nơi tiếp nhận, xử lý các yêu cầu từ Client và gửi phản hồi tương ứng. Trong đồ án này, Server được xây dựng bằng Node.js kết hợp với Express.js – một framework mạnh mẽ và linh hoạt cho ứng dụng web.

Các chức năng chính của Server bao gồm:

- Xử lý logic nghiệp vụ (business logic) như đăng nhập, xử lý giỏ hàng, đặt hàng...
- Quản lý và tương tác với cơ sở dữ liệu (thêm, sửa, xóa, truy vấn dữ liệu).

- Cung cấp API RESTful để các client có thể dễ dàng giao tiếp.
- Bảo mật hệ thống bằng các phương pháp xác thực (authentication) và phân quyền (authorization).
- Xử lý các tệp tin (uploads), mã hóa thông tin, gửi email, v.v.
- Server thường hoạt động trên một cổng (port) riêng, lắng nghe các yêu cầu đến và phản hồi dữ liệu phù hợp với định dạng JSON hoặc HTML.

2.1.4 Ưu điểm của mô hình Client – Server

Tính phân tách rõ ràng: Tách biệt giao diện và xử lý logic, dễ bảo trì và nâng cấp

Tái sử dụng cao: Backend có thể dùng chung cho nhiều loại client (web, mobile, desktop).

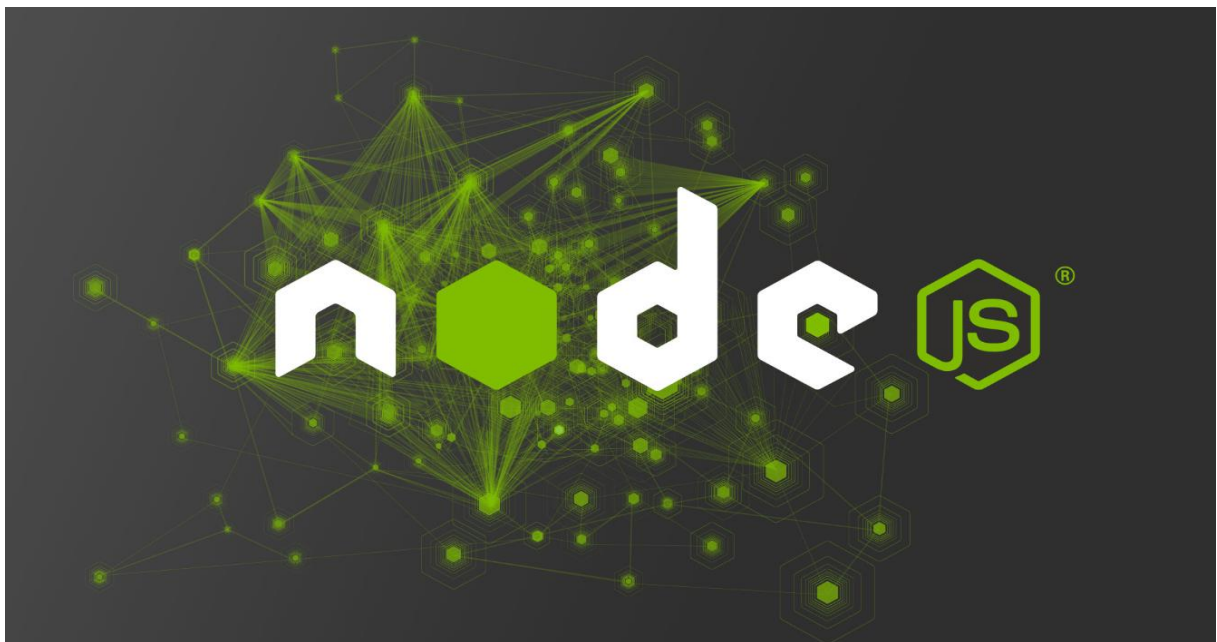
Dễ mở rộng: Có thể tăng hiệu năng Server hoặc triển khai thêm nhiều Client dễ dàng.

Bảo mật: Dữ liệu được quản lý tập trung tại Server, dễ kiểm soát và bảo vệ.

2.2 Giới thiệu công nghệ

2.2.1 Giới thiệu về NodeJS

NodeJS là một nền tảng chạy phía máy chủ (server-side platform) được xây dựng dựa trên ngôn ngữ lập trình JavaScript. NodeJS sử dụng mô hình lập trình hướng sự kiện (event-driven) và bất đồng bộ (non-blocking I/O), giúp xử lý nhiều yêu cầu cùng lúc một cách hiệu quả và tiết kiệm tài nguyên hệ thống. NodeJS hiện được sử dụng rộng rãi trong phát triển các ứng dụng web, API, dịch vụ thời gian thực, hệ thống IoT.



Hình 2.2 Logo Nodejs

NodeJS được phát triển bởi Ryan Dahl và chính thức ra mắt vào năm 2009. Ban đầu, NodeJS được tạo ra nhằm giúp JavaScript có thể hoạt động ngoài trình duyệt, đặc biệt là xử lý các tác vụ phía backend. Nhờ việc sử dụng Google V8 Engine của Chrome, NodeJS mang lại tốc độ xử lý cao và hiệu suất mạnh mẽ. Từ khi ra đời, NodeJS đã nhanh chóng phát triển và trở thành nền tảng rất phổ biến, được hỗ trợ và phát triển bởi cộng đồng lập trình viên rộng khắp trên thế giới.

Không giống như các công nghệ backend truyền thống sử dụng mô hình xử lý theo luồng (thread) và đồng bộ, NodeJS hoạt động theo cơ chế “async – non-blocking”, cho phép xử lý đa tác vụ mà không bị chặn, giúp hệ thống hoạt động hiệu quả hơn đặc biệt đối với các ứng dụng yêu cầu khả năng mở rộng cao. NodeJS hỗ trợ chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, macOS và có thể tích hợp với nhiều công nghệ, hệ quản trị cơ sở dữ liệu khác nhau, phù hợp cho các ứng dụng web hiện đại và dịch vụ thời gian thực.

2.2.2 Ưu điểm của Nodejs

- NodeJS sử dụng ngôn ngữ JavaScript – một ngôn ngữ quen thuộc và phổ biến, giúp lập trình viên có thể sử dụng chung một ngôn ngữ cho cả frontend và backend, giảm độ phức tạp trong phát triển ứng dụng.
- NodeJS hoạt động dựa trên mô hình xử lý bất đồng bộ (non-blocking I/O) và hướng sự kiện (event-driven), giúp tăng hiệu suất và khả năng xử lý nhiều yêu cầu cùng lúc mà không cần sử dụng nhiều luồng, hạn chế tình trạng nghẽn kết nối.
- NodeJS có thể chạy trên nhiều môi trường và hệ điều hành khác nhau như Windows, Linux, macOS,... mà không yêu cầu thay đổi mã nguồn, phù hợp cho nhiều kiến trúc triển khai.
- Java là ngôn ngữ độc lập với nền tảng, có nghĩa là chương trình Java có thể chạy trên nhiều hệ điều hành khác nhau mà không cần phải chỉnh sửa lại mã nguồn. Chỉ cần chuyển đổi code java sang file jar hoặc file war là chúng ta có thể chạy trên nhiều nền tảng
- Bộ xử lý JavaScript V8 Engine được tích hợp trong NodeJS giúp tối ưu tốc độ thực thi mã nguồn, tăng hiệu quả xử lý và phù hợp cho các ứng dụng thời gian thực.
- NodeJS có thể chạy trên nhiều môi trường và hệ điều hành khác nhau như Windows, Linux, macOS,... mà không yêu cầu thay đổi mã nguồn, phù hợp cho nhiều kiến trúc triển khai.

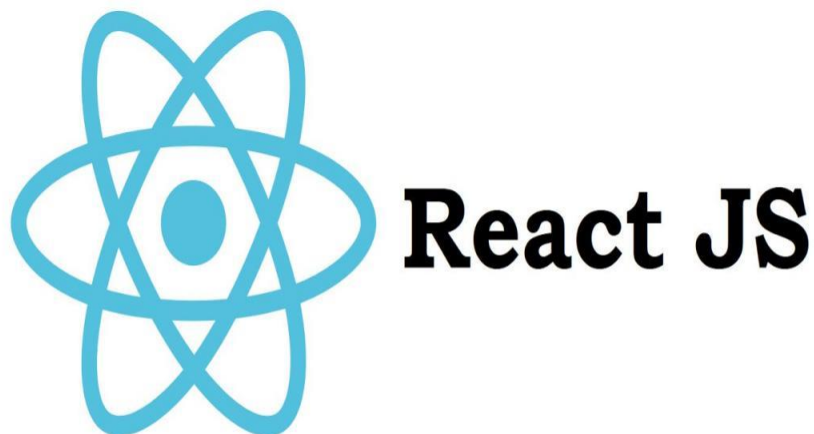
- NodeJS linh hoạt và dễ mở rộng nhờ vào mô hình lập trình module, giúp mã nguồn rõ ràng, dễ bảo trì, nâng cấp và tích hợp thêm nhiều tính năng mới.
- NodeJS có một hệ sinh thái thư viện và module phong phú thông qua npm (Node Package Manager), cho phép lập trình viên dễ dàng tìm kiếm thư viện hỗ trợ, rút ngắn thời gian phát triển ứng dụng.

2.2.3 Nhược điểm của Nodejs

- Do sử dụng cơ chế single-thread và mô hình xử lý bất đồng bộ, việc quản lý các tác vụ phức tạp có nhiều thao tác xử lý đồng thời có thể gây khó khăn trong lập trình, bảo trì và kiểm soát luồng xử lý, đặc biệt với các lập trình viên mới.
- Không phù hợp cho các tác vụ tính toán nặng như xử lý thuật toán đồ họa, mã hóa hay phân tích dữ liệu lớn vì có thể làm chậm toàn bộ tiến trình xử lý do NodeJS chỉ chạy trên một luồng chính.
- Với mô hình callback và xử lý bất đồng bộ liên tiếp, nếu không tổ chức mã nguồn hợp lý có thể dẫn đến tình trạng “callback hell”, gây khó khăn cho việc đọc, debug và bảo trì mã nguồn.

2.3 Giới thiệu về Reactjs

React (còn được gọi là Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dùng nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dung các component mà chứa các component cụ thể hoặc các thẻ HTML.



Hình 2.3 Logo reactjs

Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.

Nó cũng sử dụng khái niệm là Virtual DOM (DOM ảo). Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là Reactjs chỉ render những component hay subcomponent nào thực sự thay đổi.

2.3.1 Ưu điểm của Reactjs

- Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
- Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
- Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
- Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

2.3.2 Nhược điểm của Reactjs

ReactJS chỉ đảm nhiệm tầng View trong mô hình ứng dụng, do đó bản chất React không phải là một framework MVC hoàn chỉnh mà chỉ là thư viện xây dựng giao diện người dùng (UI library). Điều này dẫn đến việc React không cung cấp sẵn các thành phần như Model và Controller, không hỗ trợ đầy đủ cơ chế quản lý dữ liệu hay xử lý logic nghiệp vụ. Khi sử dụng ReactJS, lập trình viên buộc phải tích hợp thêm các thư viện hỗ trợ như Redux, Zustand, MobX, Axios, React Query... tùy theo nhu cầu, từ đó làm tăng độ phức tạp của dự án.

- Việc tích hợp ReactJS vào các framework MVC truyền thống như Laravel, Spring, Django hay .NET cần trải qua nhiều bước cấu hình, đặc biệt liên quan đến bundler, transpiler và hệ thống routing. Điều này khiến quá trình triển khai trở nên mất thời gian hơn so với việc sử dụng các công nghệ frontend tích hợp sẵn trong framework.
- React khá nặng nề so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
- Dung lượng thư viện React tương đối lớn nếu so sánh với một số framework hoặc thư viện frontend khác. Mặc dù dung lượng React không phải quá nặng nhưng vẫn có mức kích thước tương đương với Angular (khoảng 35kb so với 39kb của Angular), trong khi Angular lại là một framework hoàn chỉnh cung cấp đầy đủ tính năng từ kiến trúc đến xử lý logic. Điều này khiến React trở nên ít tối ưu hơn ở những dự án cần tải nhanh, đặc biệt trên mạng tốc độ thấp.

2.4 Giới thiệu về Visual studio code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn nhỏ gọn và mạnh mẽ, phát triển bởi Microsoft. Được thiết kế để đáp ứng nhu cầu lập trình của cộng đồng đa dạng, VS Code mang đến cho nhà phát triển một môi trường làm việc hiện đại và linh hoạt.



Hình 2.4 Logo visual studio code

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

2.4.1 Ưu điểm của Visual studio code

- Hỗ trợ nhiều ngôn ngữ lập trình: VSCode hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, TypeScript, Python, Java, C++, và nhiều ngôn ngữ khác. Điều này làm cho nó trở thành một IDE đa nhiệm và đa ngôn ngữ.
- Hỗ trợ Git tích hợp: VSCode tích hợp sẵn công cụ quản lý phiên bản Git, giúp bạn theo dõi thay đổi mã nguồn, thực hiện commit, và quản lý nhánh dễ dàng.
- Hiện thị IntelliSense: VSCode cung cấp IntelliSense, một tính năng tự động hoàn thành mã nguồn, giúp bạn nhanh chóng và dễ dàng gõ mã và truy cập vào các phương thức và biến.
- Mở rộng và Cộng đồng: VSCode có hệ thống mở rộng mạnh mẽ, cho phép bạn cài đặt các tiện ích mở rộng (extensions) từ thư viện lớn trên Visual Studio Code Marketplace. Cộng đồng người dùng cũng đóng góp nhiều extensions hữu ích.
- Gỡ lỗi (Debugging): VSCode hỗ trợ chức năng gỡ lỗi cho nhiều ngôn ngữ lập trình, với khả năng thiết lập các điểm dừng (breakpoints), xem biến, và theo dõi thực thi chương trình.
- Terminal tích hợp: VSCode tích hợp một terminal trực tiếp trong giao diện người dùng, giúp bạn thực hiện các lệnh hệ thống và quản lý dự án mà không cần chuyển đến cửa sổ terminal riêng biệt.
- Theme và tùy chỉnh: VSCode cho phép bạn tùy chỉnh giao diện và theme theo ý muốn của mình. Nó cũng hỗ trợ nhiều theme màu sắc phổ biến.
- Integrations và Snippets: Hỗ trợ nhiều tích hợp và snippets, giúp tăng tốc quá trình phát triển bằng cách cung cấp các đoạn mã mẫu và tích hợp với các dịch vụ phổ biến.
- Tích hợp với các công cụ môi trường phát triển: VSCode có thể tích hợp với nhiều công cụ và môi trường phát triển khác nhau, tạo điều kiện thuận lợi cho quá trình phát triển ứng dụng.

2.4.2 Nhược điểm của Visual studio code

- VS Code tiêu tốn nhiều tài nguyên hệ thống, đặc biệt khi cài đặt nhiều tiện ích mở rộng (extensions). Trên máy tính có cấu hình thấp, hiệu suất có thể giảm, gây ra tình trạng giật, lag hoặc phản hồi chậm.
- VS Code có tính năng mặc định khá cơ bản. Để sử dụng hiệu quả cho các ngôn ngữ hoặc công nghệ cụ thể, người dùng phải cài đặt các extensions.
- Mặc dù hỗ trợ đa ngôn ngữ, nhưng VS Code không tối ưu chuyên sâu như các IDE chuyên dụng khác. Ví dụ:
- VS Code không tích hợp sẵn các công cụ nâng cao như quản lý vòng đời dự án, trình biên dịch, hay mô phỏng môi trường phức tạp (như Visual Studio hoặc Eclipse).
- VS Code không tích hợp sẵn các công cụ nâng cao như quản lý vòng đời dự án, trình biên dịch, hay mô phỏng môi trường phức tạp (như Visual Studio hoặc Eclipse).
- Với người mới, cấu hình VSCode ban đầu hơi phức tạp về môi trường làm việc.

2.5 Giới thiệu về hệ quản trị cơ sở dữ liệu MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL (Not Only SQL) dạng Document-Oriented (hướng tài liệu). Thay vì lưu trữ dữ liệu dưới dạng bảng (table) như các hệ quản trị CSDL quan hệ truyền thống, MongoDB lưu dữ liệu dưới dạng JSON/BSON, linh hoạt và dễ mở rộng. MongoDB được thiết kế để xử lý dữ liệu lớn, phi cấu trúc hoặc bán cấu trúc, đồng thời hỗ trợ khả năng mở rộng ngang (scale-out) rất hiệu quả.



Hình 2.5 Logo MongoB

Vai trò của hệ quản trị cơ sở dữ liệu:

- Cung cấp môi trường lưu trữ dữ liệu tập trung, giúp quản lý dữ liệu hiệu quả, thay thế cho việc lưu trữ bằng file hoặc các phương thức rời rạc, đảm bảo dữ liệu được tổ chức, bảo mật và dễ dàng truy cập khi cần.
- Hỗ trợ đầy đủ các thao tác xử lý dữ liệu như: thêm mới, cập nhật, xóa và truy vấn dữ liệu. Các thao tác này tương ứng với CRUD (Create – Read – Update – Delete). MongoDB sử dụng cú pháp Mongo Query Language (MQL) thay vì SQL như các hệ quản trị cơ sở dữ liệu quan hệ.
- MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được sử dụng rộng rãi trong các ứng dụng web hiện đại nhờ khả năng linh hoạt trong lưu trữ dữ liệu dạng tài liệu (document), đặc biệt phù hợp với các hệ thống yêu cầu tốc độ truy xuất nhanh và khả năng mở rộng cao.
- MongoDB hỗ trợ lưu trữ dữ liệu dưới dạng JSON/BSON, giúp mô hình dữ liệu trở nên linh hoạt, dễ thay đổi cấu trúc mà không ảnh hưởng đến toàn bộ hệ thống như các cơ sở dữ liệu quan hệ.
- MongoDB có thể được quản lý và thao tác thông qua nhiều công cụ trực quan như MongoDB Compass, hoặc Mongo Shell, hỗ trợ triển khai trên nhiều môi trường từ máy tính cá nhân, máy chủ doanh nghiệp đến các nền tảng cloud như Atlas.
- Giao diện trực quan, dễ sử dụng và cú pháp truy vấn thân thiện giúp người mới tiếp cận dễ dàng, đồng thời vẫn đáp ứng tốt nhu cầu mở rộng của doanh nghiệp.

2.5.1 Ưu điểm của hệ quản trị cơ sở dữ liệu MongoDB

- Miễn phí và mã nguồn mở, người dùng có thể cài đặt và sử dụng dễ dàng, tương tự như các công cụ phổ biến trong SQL.
- Cung cấp nhiều chức năng linh hoạt, đặc biệt phù hợp với các ứng dụng web hiện đại cần xử lý dữ liệu lớn và thay đổi liên tục.
- Hỗ trợ nhiều giao diện quản lý trực quan như MongoDB Compass, Robo 3T, Mongo Atlas,... giúp việc thao tác trở nên dễ dàng.
- Hoạt động tốt trên nhiều nền tảng hệ điều hành như Windows, macOS, Linux,... mà không yêu cầu cấu hình phức tạp.
- Linh hoạt trong lưu trữ dữ liệu, không yêu cầu schema cố định, cho phép thay đổi cấu trúc dữ liệu linh hoạt theo yêu cầu phát triển.

- Khả năng mở rộng (scaling) rất tốt, đặc biệt là phương pháp sharding, phù hợp với hệ thống lớn có lượng truy cập cao.
- Hỗ trợ tìm kiếm và truy vấn hiệu quả, đặc biệt với dữ liệu phi cấu trúc, bán cấu trúc, dữ liệu thời gian thực.
- Dễ dàng đồng bộ và triển khai trên môi trường cloud, đặc biệt khi sử dụng MongoDB Atlas.

2.5.2 Nhược điểm của hệ quản trị cơ sở dữ liệu MongoDB

- Không hỗ trợ tốt các giao dịch phức tạp như trong SQL, mặc dù đã cải thiện từ phiên bản mới nhưng vẫn chưa tối ưu bằng hệ quản trị quan hệ.
- Tốn nhiều dung lượng lưu trữ hơn do lưu trữ dữ liệu dưới dạng BSON dẫn đến kích thước document có thể lớn.
- Yêu cầu người dùng phải có kiến thức về NoSQL, cách tổ chức document, collection để tránh dư thừa dữ liệu và giảm hiệu suất.
- Không đảm bảo tính nhất quán dữ liệu tuyệt đối (eventual consistency) trong một số trường hợp phân tán dữ liệu.
- Nếu thiết kế dữ liệu không tối ưu, hiệu suất sẽ suy giảm nhanh khi hệ thống mở rộng hoặc số lượng truy vấn tăng cao.
- Một số tính năng nâng cao, dịch vụ Cloud và bảo mật chuyên sâu yêu cầu người dùng trả phí khi sử dụng trên MongoDB Atlas.

Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Mô tả bài toán

3.1.1 Bối cảnh và mục tiêu

Trong bối cảnh xu hướng số hóa và thương mại điện tử đang phát triển mạnh mẽ tại Việt Nam, nhu cầu đặt dịch vụ trực tuyến ngày càng trở nên phổ biến, không chỉ trong lĩnh vực mua sắm mà còn lan rộng sang các hoạt động giải trí và thể thao. Đặc biệt, sau thời kỳ dịch Covid-19, thói quen quản lý thời gian và ưu tiên sự tiện lợi của người dùng đã thay đổi rõ rệt, dẫn đến nhu cầu tìm kiếm – đặt lịch – thanh toán dịch vụ theo hình thức online ngày càng tăng cao.

Tại các thành phố lớn, trong đó có Đà Nẵng, nhu cầu sử dụng và thuê sân thể thao như sân bóng đá, bóng chuyền, cầu lông, tennis,... ngày càng nhiều, đặc biệt đối với các nhóm đối tượng như học sinh, sinh viên, nhân viên văn phòng, câu lạc bộ thể thao, hội nhóm bạn bè,... Tuy nhiên, việc đặt sân truyền thống vẫn còn nhiều hạn chế như phải trực tiếp liên hệ qua điện thoại, thiếu tính minh bạch về giá cả – thời gian trống sân, dễ xảy ra tình trạng nhầm lẫn hoặc trùng lịch.

Xuất phát từ thực tiễn trên, đề tài “Xây dựng hệ thống đặt sân thể thao trực tuyến” được nghiên cứu và phát triển nhằm mang đến một giải pháp công nghệ hiện đại, giúp người dùng có thể dễ dàng tìm kiếm, xem lịch trống, đặt sân và thanh toán trực tuyến nhanh chóng, tiện lợi mọi lúc mọi nơi. Ngoài ra, hệ thống còn hỗ trợ chủ sân quản lý sân bãi, lịch đặt, khách hàng, doanh thu và các hoạt động kinh doanh một cách hiệu quả.

Mục tiêu của đề tài là xây dựng một website đặt sân thể thao trực tuyến với giao diện thân thiện, dễ sử dụng, hoạt động ổn định, cung cấp đầy đủ các tính năng cần thiết cho người dùng và quản trị viên, góp phần tối ưu hóa trải nghiệm khách hàng và hỗ trợ quá trình số hóa dịch vụ cho các cơ sở kinh doanh sân thể thao.

3.1.2 Nhóm người dùng

Khách hàng(customer):

Người dùng ẩn danh (Khách vắng lai – Anonymous User)

- Không cần đăng nhập vẫn có thể truy cập hệ thống để xem thông tin sân, giá thuê, vị trí, tiện ích đi kèm và khung giờ trống.

- Người dùng có thể đăng ký tài khoản bằng email và xác thực qua mã nhận gửi qua email để trở thành khách hàng đã đăng nhập.

Người dùng đã đăng ký:

Sau khi đăng ký và đăng nhập thành công, người dùng sẽ có toàn quyền sử dụng các tính năng của hệ thống bao gồm:

- Tìm kiếm và lọc sân theo loại (bóng đá, cầu lông, bóng chuyền, tennis,...), địa điểm, khung giờ và mức giá.
- Xem chi tiết sân bao gồm: diện tích, loại sân, số lượng người chơi phù hợp, bảng giá theo giờ, hình ảnh thực tế, tiện ích kèm theo và đánh giá từ người dùng khác.
- Đặt sân trực tuyến ngay tại phần thông tin chi tiết hoặc thông qua giỏ đặt sân.
- Chọn hình thức thanh toán: thanh toán khi đến sân (offline) hoặc thanh toán trực tuyến (online).
- Theo dõi lịch sử đặt sân: xem chi tiết đơn đặt sân, trạng thái đơn (đang chờ xác nhận, đã xác nhận, đã hoàn thành, đã hủy), có thể hủy đơn theo điều kiện quy định.
- Cập nhật thông tin cá nhân gồm họ tên, số điện thoại, email, avatar, thông tin liên hệ.
- Xem tin tức và bài viết liên quan đến thể thao, sức khỏe, giải đấu, hướng dẫn đặt sân, khuyến mãi,...

Quản trị viên(admin)

Sau khi đăng nhập bằng tài khoản quản trị, Admin có quyền quản lý toàn bộ dữ liệu và tính năng của hệ thống, bao gồm:

- Quản lý tài khoản người dùng: tìm kiếm, xem chi tiết thông tin, khóa/mở khóa tài khoản khi vi phạm;
- Quản lý thông tin sân: thêm mới, cập nhật hoặc xóa sân trong hệ thống; điều chỉnh bảng giá, mô tả, tiện ích hoặc các thông tin liên quan đến sân.
- Quản lý đơn đặt sân: lọc, xem chi tiết đơn theo thời gian, loại sân, hình thức thanh toán, trạng thái đơn; cập nhật trạng thái đơn đặt (chờ xác nhận, đã xác nhận, đang sử dụng, hoàn thành, hủy).
- Quản lý danh mục loại sân: thêm mới, cập nhật và xóa danh mục sân (VD: bóng đá mini 5,7 người, sân bóng rổ, sân tennis)

- Quản lý bài viết: tạo, sửa, xóa bài viết liên quan đến hướng dẫn đặt sân, mẹo chơi thể thao, tin tức sự kiện, khuyến mãi.
- Quản lý doanh thu – thống kê: hệ thống hỗ trợ thống kê doanh thu theo ngày, tuần, tháng, quý, năm; thống kê sân được đặt nhiều nhất và khung giờ phổ biến.

3.1.3 Quy trình nghiệp vụ

Quy trình nghiệp vụ của khách hàng (User Flow)

- Khách hàng truy cập vào website và có thể tìm kiếm sân theo nhiều tiêu chí như: loại sân (bóng đá, cầu lông,...), địa điểm, giá thuê, thời gian trống, tiện ích đi kèm.
- Khách hàng chọn sân muốn đặt và tiến hành kiểm tra lịch trống theo ngày/giờ mong muốn.
- Khách hàng thực hiện đặt sân trực tuyến bằng cách: đặt trực tiếp hoặc thêm vào giỏ đặt và xử lý sau.
- Khách hàng thanh toán trực tuyến (Online) qua ví điện tử hoặc cổng thanh toán.
- Hệ thống gửi thông báo xác nhận đặt sân qua email hoặc giao diện tài khoản.
- Khách hàng có thể gửi đánh giá và phản hồi sau khi sử dụng dịch vụ.

Quy trình nghiệp vụ của người quản trị:

- Admin đăng nhập vào hệ thống với quyền quản trị.
- Admin quản lý thông tin sân: thêm mới, chỉnh sửa hoặc xóa sân, cập nhật hình ảnh, tiện ích và giá thuê.
- Admin quản lý đơn đặt sân: xem danh sách, lọc theo tiêu chí (ngày, loại sân, trạng thái, phương thức thanh toán) và cập nhật trạng thái đơn (chờ xác nhận, đã xác nhận, đang sử dụng, hoàn thành, hủy).
- Admin quản lý tài khoản người dùng: xem chi tiết, khóa/mở khóa tài khoản khi xuất hiện vi phạm.
- Admin quản lý danh mục sân: thêm mới, sửa đổi, sắp xếp hoặc xóa danh mục liên quan.
- Admin quản lý bài viết về hướng dẫn đặt sân, tin tức thể thao, giải đấu, kinh nghiệm chơi, hoặc các chương trình ưu đãi – khuyến mãi.
- Admin có thể theo dõi thống kê doanh thu theo ngày, tuần, tháng, quý, năm và xem báo cáo phân tích liên quan.

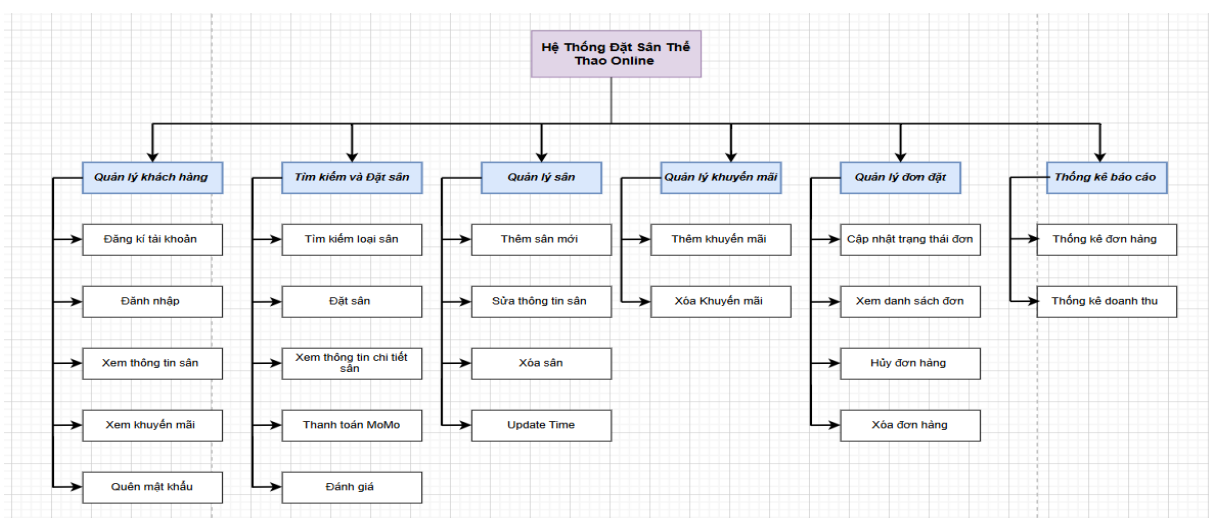
3.1.4 Các yêu cầu của hệ thống

- **Giao diện dễ sử dụng và thân thiện** với người dùng, phù hợp cho cả khách hàng và quản trị viên,
- **Tính năng tìm kiếm và lọc sân linh hoạt**, cho phép người dùng lọc theo nhiều tiêu chí như loại sân, khu vực, giá thuê, thời gian trống, tiện ích đi kèm.
- Cổng thanh toán MoMo tích hợp hỗ trợ thanh toán trực tuyến nhanh chóng và bảo mật.
- **Tích hợp cổng thanh toán trực tuyến** (ví điện tử, chuyển khoản, quét QR,...) nhằm hỗ trợ giao dịch nhanh chóng, an toàn và bảo mật;
- **Hệ thống quản lý tài khoản người dùng bảo mật**, giúp quản lý thông tin cá nhân, ngăn chặn truy cập trái phép và hỗ trợ quản lý quyền theo cấp (khách hàng – quản trị viên).
- **Hệ thống quản lý sân và đơn đặt sân khoa học**, giúp admin và chủ sân dễ dàng điều chỉnh thông tin sân, lịch trống, giá thuê, đồng thời theo dõi và xử lý trạng thái đơn đặt sân.
- **Tính năng quản lý tin tức và khuyến mãi**, nhằm đăng tải các bài viết liên quan đến thể thao, hướng dẫn sử dụng sân, sự kiện, giải đấu, chương trình ưu đãi,... để tăng tương tác và thu hút khách hàng.

3.2 Mô hình cơ sở dữ liệu

3.2.1 Mô hình phân rã chức năng(Business functional diagram)

BDF là biểu đồ phân rã có thứ bậc các chức năng của hệ thống thông tin từ tổng thể thể đến chi tiết. Mỗi chức năng có thể có một hoặc nhiều chức năng con, tất cả được thể hiện trong một khung của biểu đồ.



Hình 3.1 Sơ đồ phân rã chức năng

Miêu tả chi tiết chức năng hệ thống đặt sân thể thao online:

1. Quản lý khách hàng

Đăng ký tài khoản

- Người dùng mới có thể tạo tài khoản để sử dụng hệ thống.
- Hệ thống yêu cầu nhập thông tin cơ bản: họ tên, email, số điện thoại, mật khẩu.
- Email xác nhận được gửi đến người dùng để kích hoạt tài khoản.
- Tài khoản chỉ có hiệu lực khi người dùng xác thực thành công.

Đăng nhập

- Người dùng đăng nhập bằng email và mật khẩu đã đăng ký.
- Hệ thống kiểm tra thông tin xác thực trong cơ sở dữ liệu.
- Thông báo lỗi khi thông tin không chính xác.

Xem thông tin cá nhân

- Người dùng xem được thông tin cá nhân và lịch sử hoạt động.
- Các thông tin hiển thị: tên, email, số điện thoại, danh sách sân đã đặt.

Xem khuyến mãi

- Người dùng có thể xem các chương trình khuyến mãi đang áp dụng.
- Hiển thị chi tiết: tên khuyến mãi, mô tả, thời gian áp dụng, điều kiện sử dụng.

Quên mật khẩu

- Hỗ trợ lấy lại mật khẩu qua email đăng ký.
- Hệ thống gửi mã OTP hoặc link đổi mật khẩu.

2. Tìm kiếm và đặt sân

Tìm kiếm loại sân

- Người dùng có thể tìm theo loại sân:
 - Sân bóng đá
 - Sân bóng rổ
 - Sân tennis
- Hỗ trợ tìm theo từ khóa và định dạng theo bộ lọc (giờ rảnh, giá tiền...).

Đặt sân

- Người dùng chọn:
 - Loại sân
 - Ngày đặt
 - Khung giờ
 - Hệ thống kiểm tra slot giờ còn trống trước khi cho đặt.
 - Nếu trùng giờ → thông báo và yêu cầu chọn giờ khác.

Xem thông tin chi tiết sân

- Hiển thị đầy đủ thông tin từng sân:
 - Tên sân
 - Loại sân
 - Hình ảnh
 - Giá thuê
 - Tình trạng giờ
- Hiển thị lịch giờ theo ngày để dễ chọn giờ trống.

Thanh toán MoMo

- Người dùng chọn thanh toán qua ví MoMo.
- Hệ thống tạo liên kết thanh toán.
- Sau khi giao dịch thành công → tự động xác nhận đơn đặt sân.

Đánh giá

- Người dùng có thể đánh giá sân sau khi đã sử dụng dịch vụ.
- Nội dung đánh giá gồm: số sao, nhận xét.
- Đánh giá hiển thị công khai cho mọi người xem.

3. Quản lý sân (Admin)

Thêm sân mới

- Admin thêm sân mới:
 - Tên sân
 - Loại sân
 - Thông tin chi tiết
 - Hình ảnh
 - Giá thuê
- Sân được lưu trong cơ sở dữ liệu và hiển thị trên hệ thống.

Sửa thông tin sân

- Admin chỉnh sửa hình ảnh, giá thuê, mô tả.

- Các thay đổi được cập nhật ngay lập tức.

Xóa sân

- Xóa sân không còn hoạt động.
- Sân bị xóa sẽ không hiển thị và không thể đặt.

Update Time (Cập nhật khung giờ)

- Admin cập nhật giờ mở cửa, giờ đóng cửa, hoặc thêm slot giờ mới.
- Hệ thống tự động áp dụng cho toàn bộ lịch đặt sân.

4. Quản lý khuyến mãi (Admin)

Thêm khuyến mãi

- Admin tạo khuyến mãi mới:
- Tên chương trình
- Mức giảm giá
- Thời gian áp dụng
- Điều kiện sử dụng

Xóa khuyến mãi

- Admin có thể xóa khuyến mãi hết hạn hoặc không còn áp dụng.
- Khuyến mãi bị xóa sẽ không còn hiện trên giao diện người dùng.

5. Quản lý đơn đặt sân (Admin)

Cập nhật trạng thái đơn

- Trạng thái gồm:
 - Đã xác nhận
 - Đã hủy
- Admin cập nhật thủ công cho từng đơn sân.

Xem danh sách đơn

- Hiển thị toàn bộ danh sách đơn đặt:
 - Tên khách hàng
 - Sân đặt
 - Ngày giờ
 - Trạng thái
 - Số tiền
- Hỗ trợ lọc theo thời gian và trạng thái.

Hủy đơn hàng

- Admin có quyền hủy đơn nếu:
 - Khách yêu cầu
 - Sân gặp sự cố
 - Hủy đơn cập nhật ngay lập tức.

Xóa đơn

- Admin xóa những đơn không còn cần thiết để lưu trữ.

6. Thống kê báo cáo (Admin)

Thống kê đơn hàng

- Xem tổng số đơn theo:
 - Ngày
 - Tháng
 - Năm
- Biểu đồ hiển thị trực quan.

Thống kê doanh thu

- Tổng doanh thu theo khoảng thời gian.
- Hỗ trợ lọc theo loại sân, phương thức thanh toán.

3.2.2 Biểu đồ dòng dữ liệu

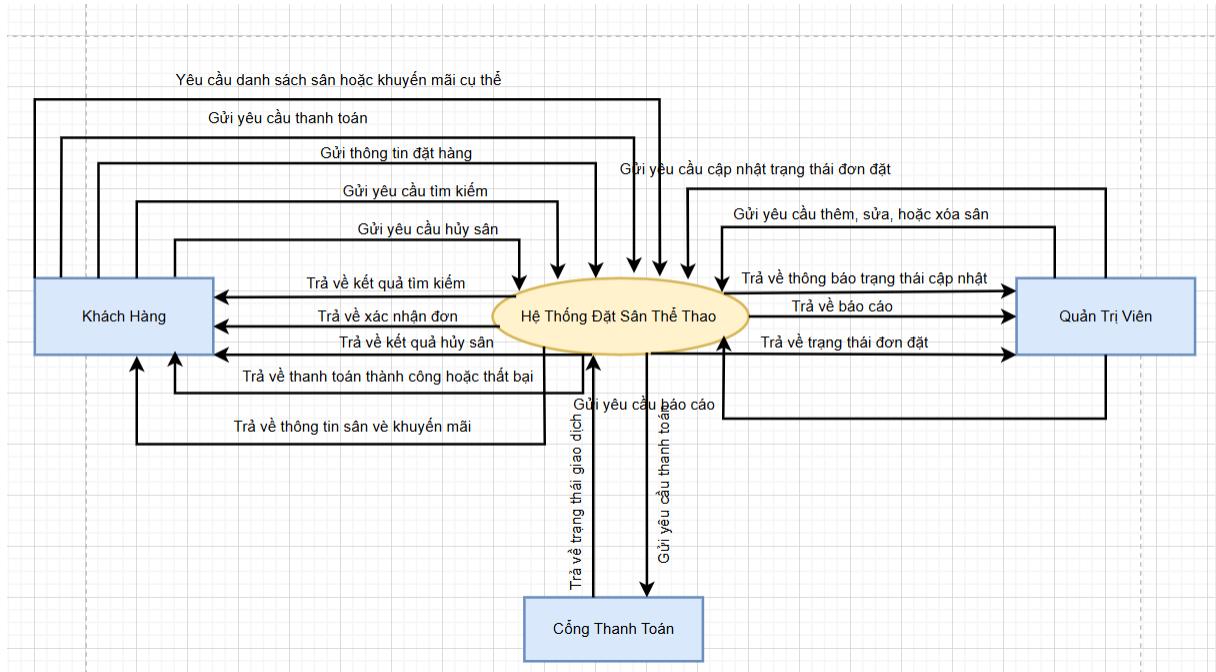
DFD là một công cụ đồ họa để mô tả luồng dữ liệu luân chuyển trong một hệ thống thông tin và những hoạt động xử lý được thực hiện bởi hệ thống thông tin đó. Biểu đồ dòng dữ liệu còn có tên gọi khác là biểu đồ bọt, biểu đồ biến đổi và mô hình chức năng. Bộ ký hiệu DFD của Gane & Sarson sử dụng trong phân tích hệ thống của website bán áo quần thời trang qua mạng.

+ Xử lý : là một hoạt động được thực hiện trên dòng dữ liệu vào để tạo một dòng dữ liệu ra, là chức năng được thực hiện bởi hệ thống thông tin để đáp ứng lại các dòng dữ liệu vào ra.

+ Dòng dữ liệu: biểu diễn một sự di chuyển của dữ liệu, giữa các quá trình hoặc kho dữ liệu.

+ Kho dữ liệu: Nơi lưu trữ dữ liệu, là các bảng đã được xác định trong quá trình phân tích dữ liệu.

+ Tác nhân ngoài: Một tác nhân ngoài là một nguồn cung cấp hoặc nhận thông tin dữ liệu của hệ thống thông tin.



Hình 3.2 Sơ đồ ngữ cảnh

Mô tả chi tiết

1. Thành phần chính

Hệ thống Đặt Sân Thể Thao

- Là **trung tâm của sơ đồ**, nơi xử lý mọi yêu cầu và luồng thông tin giữa các tác nhân.

- Cung cấp các dịch vụ:
- Tìm kiếm sân thể thao (bóng đá, bóng rổ, tennis...)
- Đặt sân, hủy sân
- Xác nhận trạng thái đơn đặt
- Quản lý lịch đặt sân
- Xử lý thanh toán trực tuyến qua cổng thanh toán
- Gửi/nhận báo cáo thống kê cho quản trị viên
- Hiện thị danh sách sân, thời gian trống, khuyến mãi

2. Các tác nhân bên ngoài

Khách hàng

- Bao gồm:
 - Người dùng chưa đăng ký (Guest)
 - Người dùng đăng ký tài khoản
- Các chức năng chính:
 - Tìm kiếm sân theo loại, vị trí, giá, thời gian trống
 - Đặt sân & nhận xác nhận đơn
 - Hủy đặt sân
 - Thanh toán trực tuyến qua MoMo
 - Xem danh sách khuyến mãi và thông tin sân
 - Nhận thông báo về trạng thái đơn đặt hoặc kết quả giao dịch

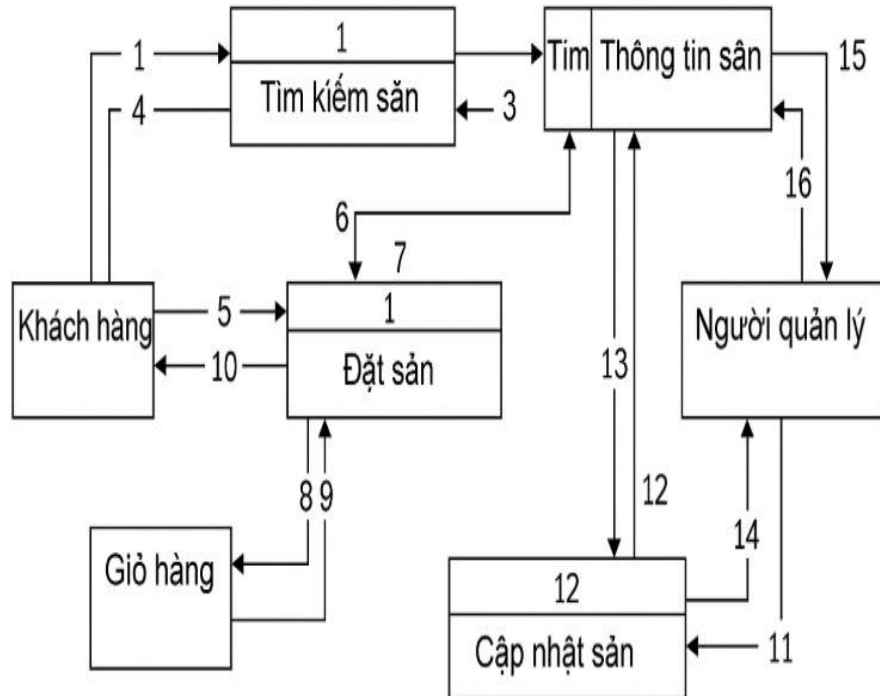
Quản trị viên

- Người vận hành hệ thống
 - Chịu trách nhiệm:
 - Quản lý sân (thêm, sửa, xóa, cập nhật trạng thái)
 - Quản lý đơn đặt sân
 - Quản lý khuyến mãi
 - Quản lý lịch đặt và khả dụng của sân
 - Xem và phân tích báo cáo thống kê
 - Theo dõi trạng thái thanh toán từ khách hàng

Cổng thanh toán

- Hỗ trợ thanh toán trực tuyến (MoMo)
 - Chức năng:
 - Nhận yêu cầu thanh toán từ hệ thống
 - Xử lý giao dịch
 - Trả về kết quả thanh toán (thành công / thất bại)
 - Gửi mã giao dịch để hệ thống cập nhật trạng thái đặt sân

3.2.2.1 Biểu đồ phân rã chức năng quản lý tìm kiếm và đặt sân

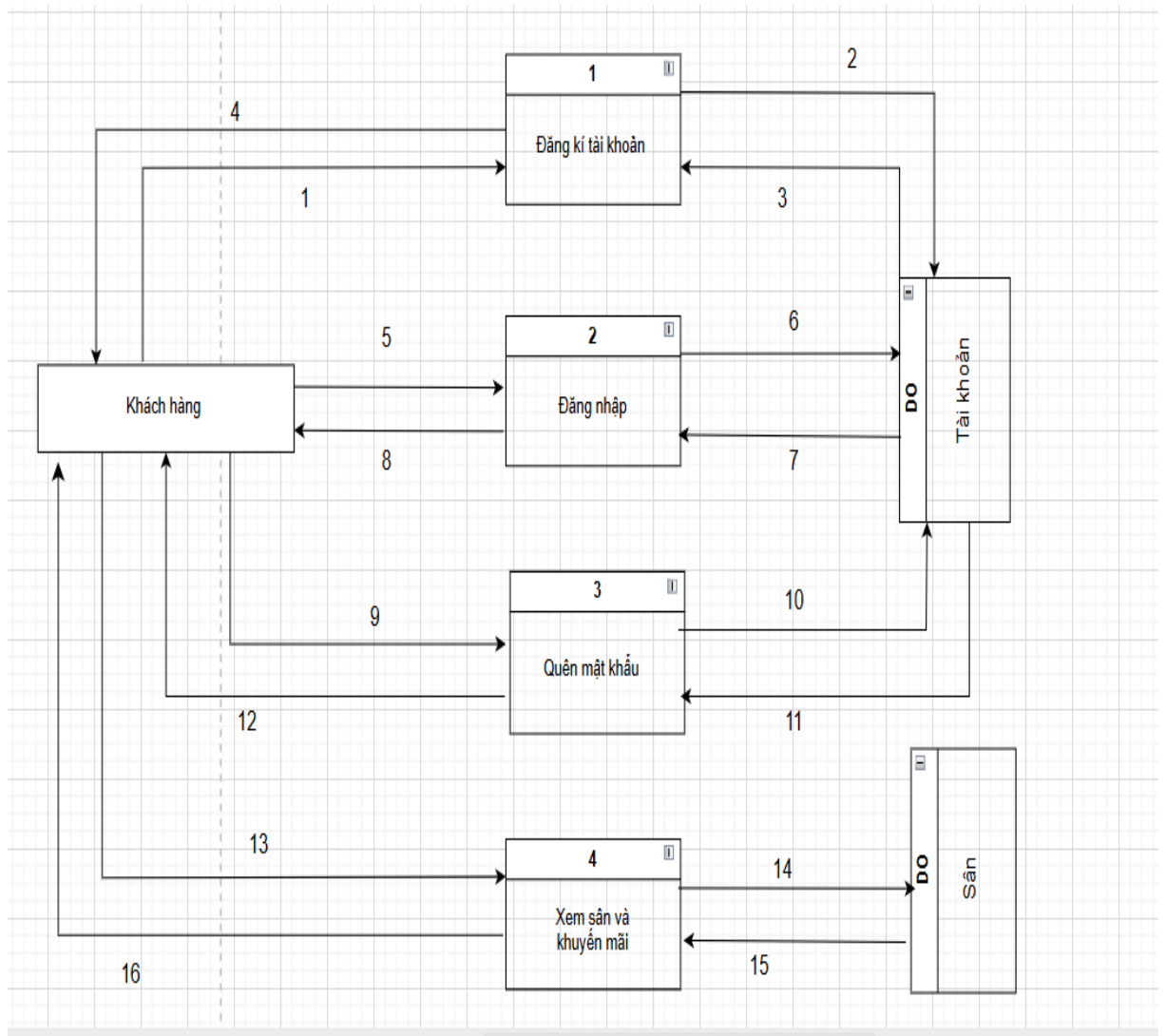


Hình 3.3 Biểu đồ phân rã chức năng quản lý tìm kiếm đặt hàng

Chú thích :

1. Yêu cầu tìm kiếm 2DS. Xác nhận thông tin 12CN. Trả về thông tin 4. Thông tin tìm sân 5. Yêu cầu lựa chọn, đặt sân 6. Gửi yêu cầu 7. Thông tin Sân 8. Lưu thông tin sân vào giỏ hàng	9. Thông tin đã đặt 10. Kết quả trả về yêu cầu 11. Gửi yêu cầu cập nhật thông tin sân 12. Lưu thông tin 13. Thông tin trả về 14. Lưu thông tin sân 15. Kết quả trả về 16. Yêu cầu
--	---

3.2.2.3 Biểu đồ phân rã chức năng quản lý khách hàng

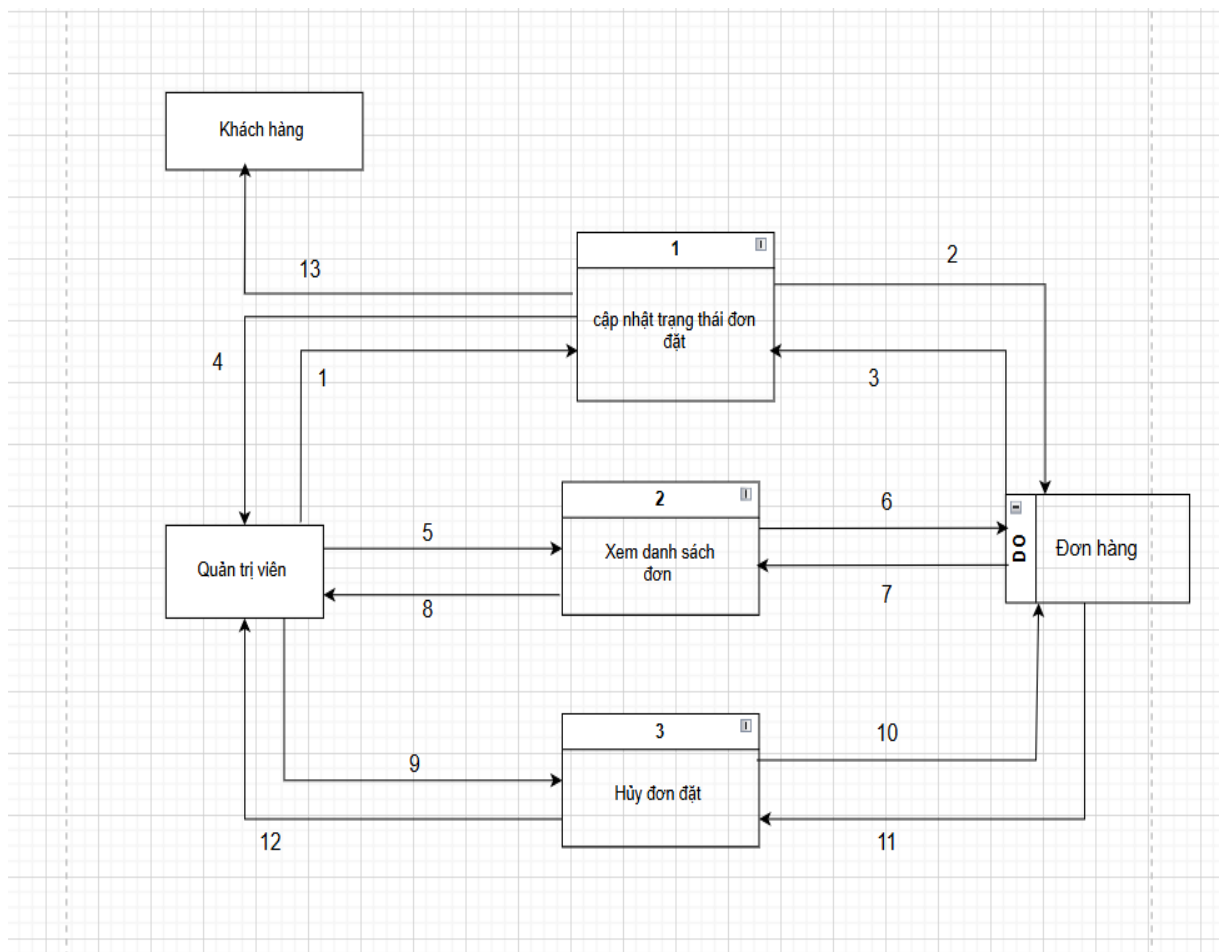


Hình 3.5 Biểu đồ phân rã chức năng quản lý khách hàng

Chú thích

1. Yêu cầu quên mật khẩu
2. Thông tin cần đổi
3. Thông tin đã đổi
4. Kết quả đổi
5. Yêu cầu xem sân và khuyến mãi
6. Lấy thông tin cần xem
7. Thông tin cần xem
8. Thông tin chi tiết

3.2.2.4 Biểu đồ phân rã chức năng quản lý đơn đặt sân

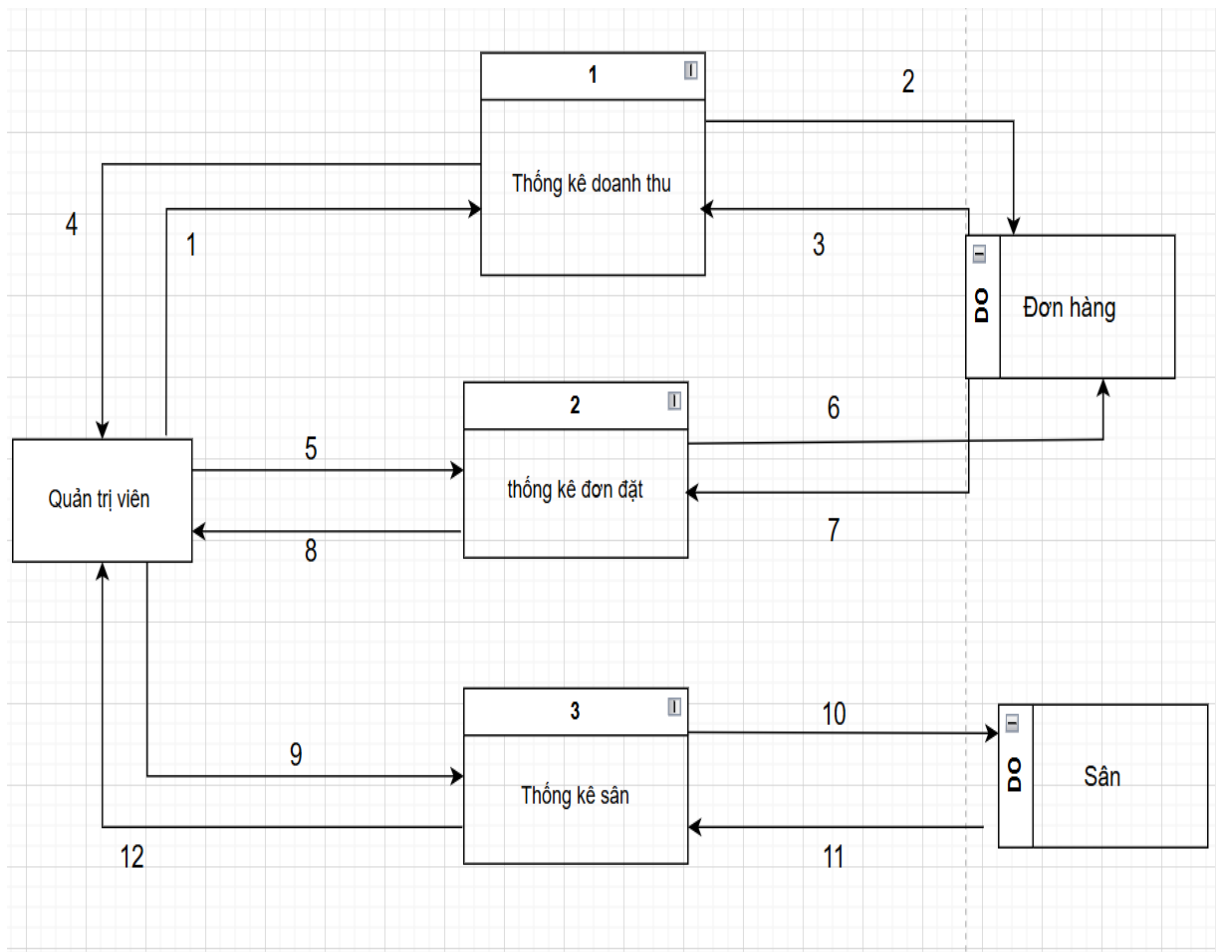


Hình 3.6 Biểu đồ phân rã chức năng quản lý đơn đặt

Chú thích

1. Yêu cầu cập nhật trạng thái đơn đặt 2. Trạng thái đơn đặt cập nhật 3. Đơn đặt đã cập nhật trạng thái 4. Kết quả cập nhật đơn đặt 5. Yêu cầu xem danh sách đơn đặt 6. Lấy danh sách đơn	7. Danh sách đơn đặt 8. Kết quả danh sách đơn 9. Yêu cầu hủy đơn đặt 10. Đơn hàng hủy 11. Đơn hàng đã hủy 12. Kết quả hủy đơn 13. Thông báo cập nhật trạng thái đơn đặt
---	--

3.2.2.5 Biểu đồ phân rã chức năng thống kê báo cáo



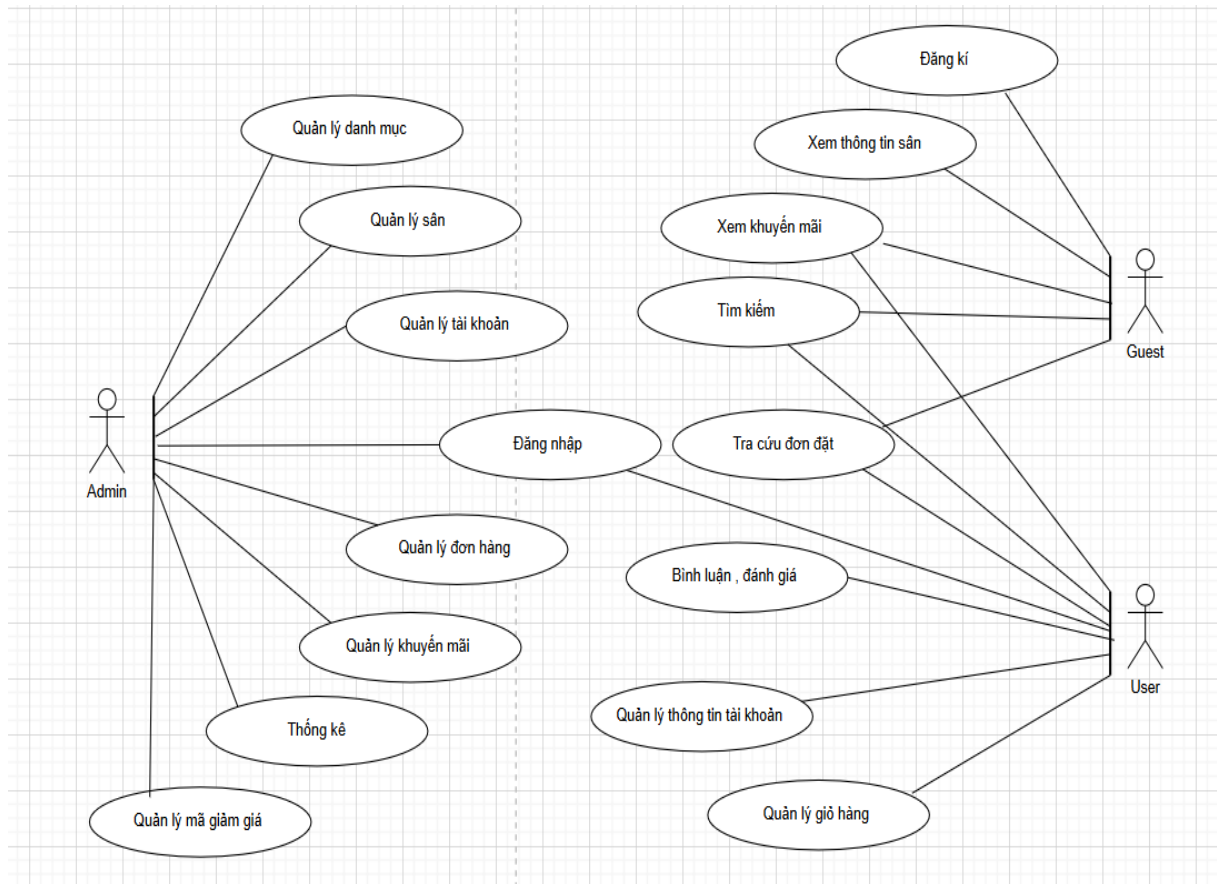
Hình 3.7 Biểu đồ phân rã chức năng thống kê báo cáo

Chú thích

1. Yêu cầu thống kê doanh thu 2. Lấy thông tin đơn đặt. 3. Thông tin doanh thu 4. Doanh thu 5. Yêu cầu thống kê đơn đặt 6. Lấy thông tin đơn	7. Thông tin đơn 8. Báo cáo thông tin đơn đặt 9. Yêu cầu thống kê sản phẩm 10. Lấy thông tin sản phẩm 11. Thông tin sản phẩm 12. Báo cáo thống kê sản phẩm
--	--

3.2.3 Biểu đồ usecase

3.2.3.1 Biểu đồ usecase tổng quát



Hình 3.8 Biểu đồ usecase tổng quát

- **Khách vãng lai (Guest):** là những người truy cập vào website nhưng chưa đăng nhập. Mục đích của hệ thống hướng tới những tác nhân này chỉ nhằm quảng cáo các sân bãi đến người dùng, do đó những tác nhân này chỉ được sử dụng các chức năng cơ bản như: tìm kiếm, xem sân, khuyến mãi, xem bình luận của những người dùng đăng nhập, và có thể bình luận dưới dạng khách.

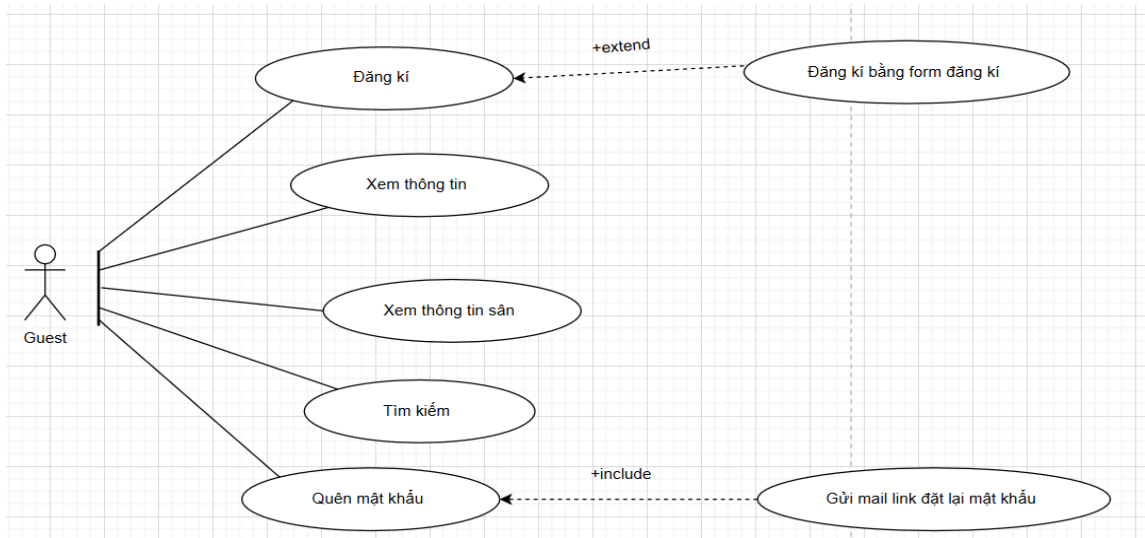
- **Người dùng (User):** là các thành viên đã đăng ký tài khoản. Tác nhân này bao gồm các chức năng cơ bản của Guest và còn có thêm nhiều chức năng nâng cao hơn: quản lý khuyến mãi, quản lý đơn đặt, bình luận, đặt sân.

- **Quản trị viên (Admin):** giữ vai trò chính trong hệ thống website. Là người quản lý và theo dõi toàn bộ hệ thống. Quản trị viên có thể thực hiện được các chức năng: quản lý sản phẩm, quản lý khuyến mãi, quản lý tài khoản, quản lý đơn đặt, thống kê, quản lý đánh giá .

3.2.3.2 Biểu đồ usecase từng tác nhân

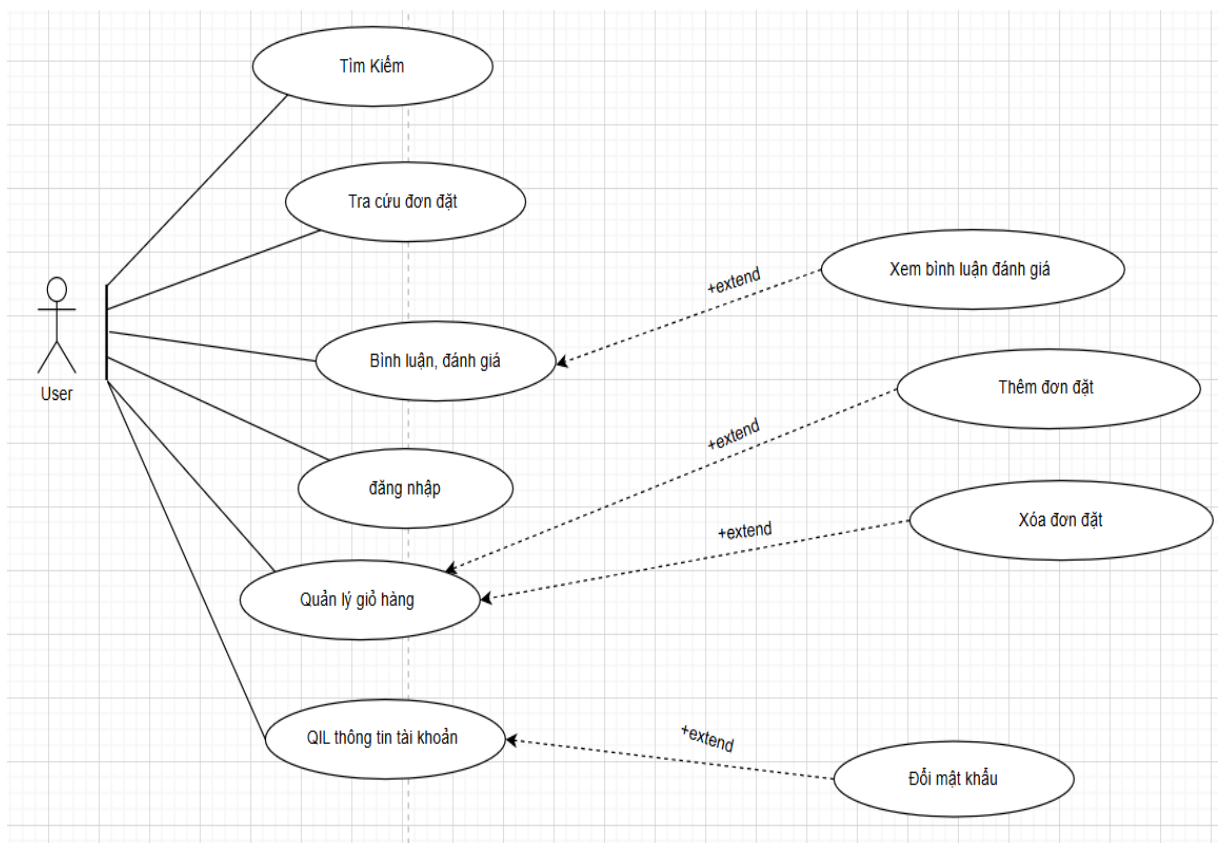
- Biểu đồ usecase tác nhân khách vãng lai

Xây dựng hệ thống đặt sân thể thao online



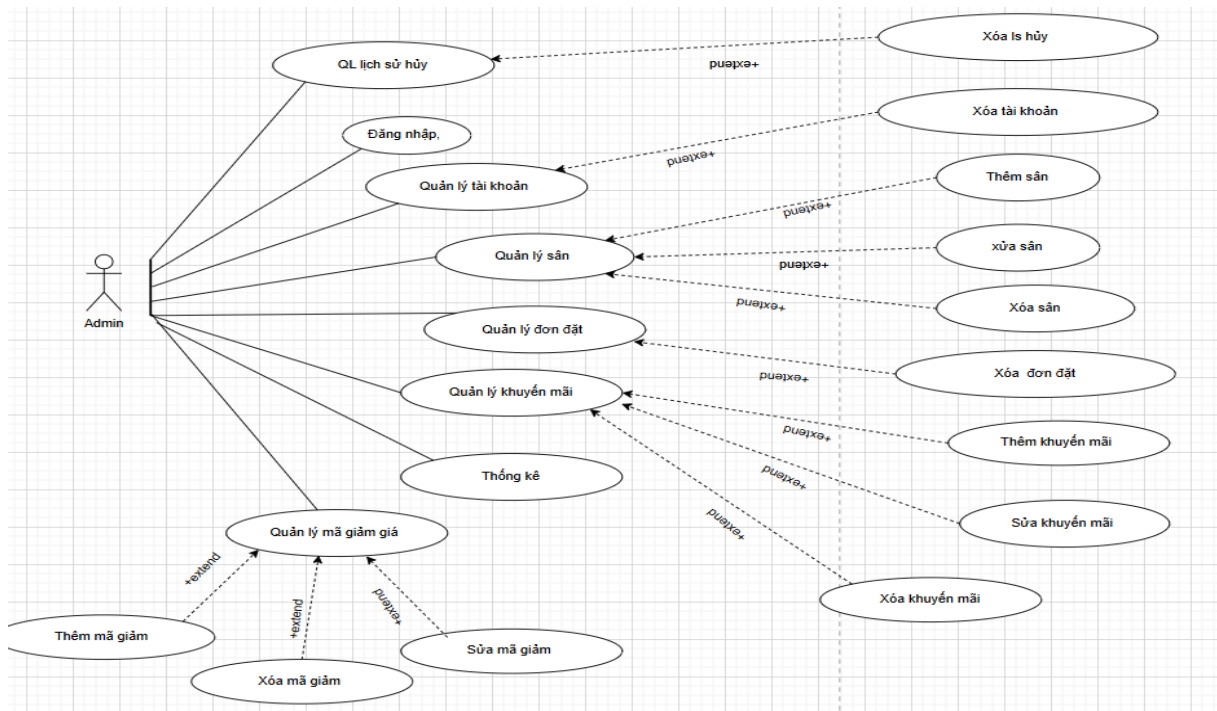
Hình 3.9 Biểu đồ usecase tác nhân khách vãng lai

- Biểu đồ usecase tác nhân người dùng



Hình 3.10 Biểu đồ usecase tác nhân người dùng

- Biểu đồ usecase tác nhân quản trị viên



Hình 3.11 Biểu đồ usecase tác nhân quản trị viên

3.2.4 Kịch bản hoạt động chức năng

3.2.4.1 Kịch bản chức năng đăng ký

Bảng 1 Kịch bản chức năng đăng ký

Tên UseCase	Đăng ký
Mô Tả	Cho phép đăng ký tài khoản mới.
Tác Nhân	Guest
Điều kiện đầu vào	Chưa có tài khoản chọn chức năng đăng ký.
Điều kiện đầu ra	Đăng ký thành công hoặc hủy đăng ký.
Dòng sự kiện chính	Người dùng chọn chức năng đăng ký: Người dùng chọn đăng ký: sau khi điền đầy đủ thông tin và ấn nút “Đăng ký”, hệ thống thông báo đăng ký thành công hay thất bại và chuyển sang trang đăng nhập.

Dòng sự kiện phụ	Nếu tài khoản gmail không chính xác hoặc đã đăng ký, hệ thống sẽ hiển thị thông báo email không đúng hoặc đã tồn tại .
------------------	--

3.2.4.2 Kịch bản chức năng đăng nhập

Bảng 2 Kịch bản chức năng đăng nhập

Tên UseCase	Đăng Nhập
Mô Tả	Cho phép thành viên đăng nhập tài khoản.
Tác Nhân	User
Điều kiện đầu vào	Đã đăng ký và chưa đăng nhập.
Điều kiện đầu ra	Đăng nhập thành công hoặc hủy đăng nhập.
Dòng sự kiện chính	Khi người dùng chọn chức năng đăng nhập: - Người dùng chọn chức năng đăng nhập: sau khi điền gmail, mật khẩu và ấn nút đăng nhập, hệ thống sẽ kiểm tra, nếu thành công sẽ chuyển hướng đến trang chủ hoặc trang quản trị(admin) theo quyền.
Dòng sự kiện phụ	- Nếu người dùng nhập sai gmail và mật khẩu, hệ thống sẽ hiển thị thông báo và quay lại trang đăng nhập.

3.2.4.3 Kịch bản chức năng tìm kiếm

Bảng 3 Kịch bản chức năng tìm kiếm

Tên UseCase	Tìm Kiếm
Mô Tả	Cho phép khách hàng tìm kiếm sân

Tác Nhân	Cho tất cả người dùng (User và Guest)
Điều kiện đầu vào	Khi người dùng nhập keyword tìm kiếm trên khung tìm kiếm.
Điều kiện đầu ra	Kết quả tìm kiếm của loại sân.
Dòng sự kiện chính	Khi người dùng nhập keyword tìm kiếm và click vào nút tìm kiếm trên đầu trang, hệ thống sẽ chuyển hướng đến trang kết quả tìm kiếm của sản phẩm và kết thúc ca sử dụng chức năng tìm kiếm.
Dòng sự kiện phụ	Khi keyword tìm kiếm không có kết quả, hệ thống sẽ không hiển thị và quay lại trang chủ.

3.2.4.4 Kịch bản chức năng đánh giá và bình luận

Bảng 4 Kịch bản chức năng đánh giá và bình luận

Tên UseCase	Đánh giá và bình luận
Mô Tả	Cho phép khách hàng đánh giá và bình luận về sản phẩm.
Tác Nhân	Người dùng đã đăng nhập (User) và Khách vãng lai (Guest)
Điều kiện đầu vào	Khi người dùng truy cập vào trang chi tiết sân và điền số sao đánh giá và bình luận.
Điều kiện đầu ra	Kết quả đánh giá và bình luận về sản phẩm đó được hiển thị trên trang chi tiết đánh giá
Dòng sự kiện chính	Khi người dùng đăng nhập truy cập trang đánh giá và nhập nội dung đánh giá và bình luận rồi ấn nút gửi. Sau khi gửi, bình luận và đánh giá sẽ được hiển thị ngay trên trang chi tiết đánh giá
Dòng sự kiện phụ	Không có

3.2.4.5 Kịch bản chức năng lịch sử đặt

Bảng 5 Kịch bản chức năng lịch sử đặt

Tên Usecase	Lịch sử đặt
Mô tả	Cho phép khách hàng quản lý lịch sử đặt
Tác nhân	User
Điều kiện đầu vào	Khi người dùng chọn các chức năng Lịch sử
Điều kiện đầu ra	Kết quả lịch sử đặt của khách hàng sẽ hiển thị

3.2.4.6 Kịch bản chức năng đặt sân

Bảng 6 Kịch bản chức năng đặt sân

Tên Usecase	Đặt sân
Mô tả	Cho phép khách hàng đăng nhập và tiến hành đặt sân trên hệ thống
Tác nhân	Người dùng đã đăng nhập (user)
Điều kiện đầu vào	Đã đăng nhập vào hệ thống và có sân đã đặt trong lịch sử đặt
Điều kiện đầu ra	Đặt sân thành công
Dòng sự kiện chính	Khi người dùng đăng nhập và chọn loại sân muốn đặt, hệ thống sẽ hiển thị form thông tin người mua và phương thức thanh toán 1. Khách hàng chọn “ loại sân ” 2. Hệ thống hiển thị tất cả loại sân mà

	khách hàng chọn - Khách hàng chọn loại sân mà mình muốn đặt - Khách hàng xem thông tin sân và chọn đặt sân - Hệ thống chuyển qua trang cho đặt - Khách hàng chọn ngày giờ và xác nhận thanh toán - Hệ thống thông báo thanh toán thành công hoặc thất bại - Sau khi thanh toán thành công thì gửi về email
Dòng sự kiện phụ	Nếu khách hàng chọn ngày cùng với khách hàng khác mà khách hàng khác đã đặt trước thì sẽ hiển thị thông báo không được chọn khung giờ đó nữa

3.2.4.7 Kịch bản chức năng quản lý tài khoản

Bảng 7 Kịch bản chức năng quản lý tài khoản

Tên UseCase	Quản lý tài khoản
Mô Tả	Cho phép người quản trị quản lý tất cả các tài khoản người dùng trên hệ thống.
Tác Nhân	Người quản trị (Admin).
Điều kiện đầu vào	Người quản trị truy cập vào trang quản trị và chọn chức năng quản lý tài khoản.
Điều kiện đầu ra	Kết quả thao tác với các chức năng quản lý tài khoản người dùng.

Dòng sự kiện chính	Khi người quản trị truy cập vào trang quản trị của hệ thống, tại đây người quản trị chọn chức năng quản lý tài khoản, hệ thống sẽ hiển thị toàn bộ danh sách tài khoản. Người quản trị có thể thao tác với các chức năng quản trị như: xóa tài khoản
Dòng sự kiện phụ	Không có

3.2.4.8 Kịch bản chức năng quản lý khuyến mãi

Bảng 8 Kịch bản chức năng quản lý khuyến mãi

Tên UseCase	Quản lý khuyến mãi
Mô Tả	Cho phép người quản trị quản lý tất cả các Khuyến mãi trên hệ thống.
Tác Nhân	Người quản trị (Admin).
Điều kiện đầu vào	Người quản trị truy cập vào trang quản trị và chọn chức năng khuyến mãi
Điều kiện đầu ra	Kết quả thao tác với các chức năng quản lý khuyến mãi trên hệ thống.
Dòng sự kiện chính	Khi người quản trị truy cập vào trang quản trị của hệ thống, tại đây người quản trị chọn chức năng khuyến mãi, hệ thống sẽ hiển thị toàn bộ danh sách khuyến mãi trên hệ thống. Người quản trị có thể thao tác với các chức năng quản trị như: xem chi tiết, thêm khuyến mãi mới, sửa, xóa,
Dòng sự kiện phụ	Không có

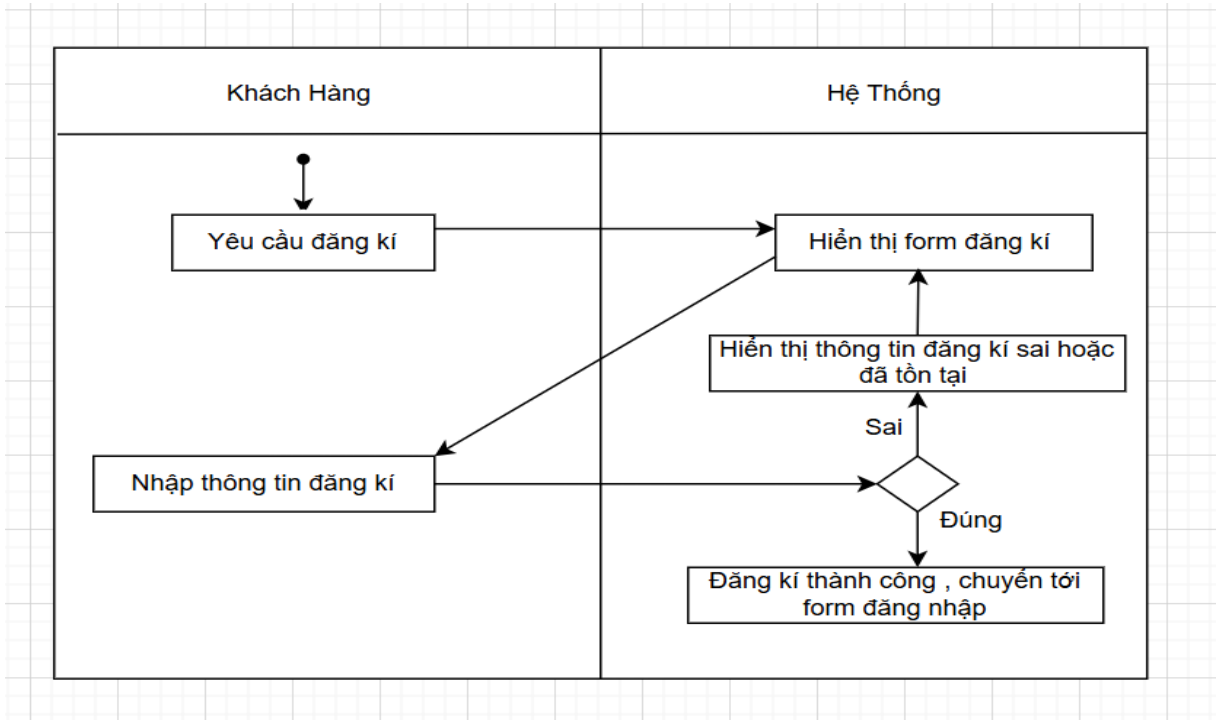
3.2.4.9 Kịch bản chức năng quản lý sân

Bảng 9 Kịch bản chức năng quản lý sân

Tên usecase	Quản lý sản phẩm
Mô tả	Cho phép người quản trị quản lý tất cả các sân của hệ thống
Tác nhân	Người quản trị (Admin)
Điều kiện đầu vào	Người quản trị truy cập vào trang quản trị và truy cập vào chức năng quản lý sân
Điều kiện đầu ra	Kết quả thao tác với các chức năng quản lý trên hệ thống
Dòng sự kiện chính	Khi người quản trị truy cập vào trang quản trị của hệ thống tại đây người quản trị chọn chức năng quản lý sân hệ thống sẽ tự động hiện thị toàn bộ sân của cửa hàng. Tại đây người quản trị có thể thao tác tất cả các chức năng như thêm sửa xóa, tìm kiếm sân
Dòng sự kiện phụ	Không có

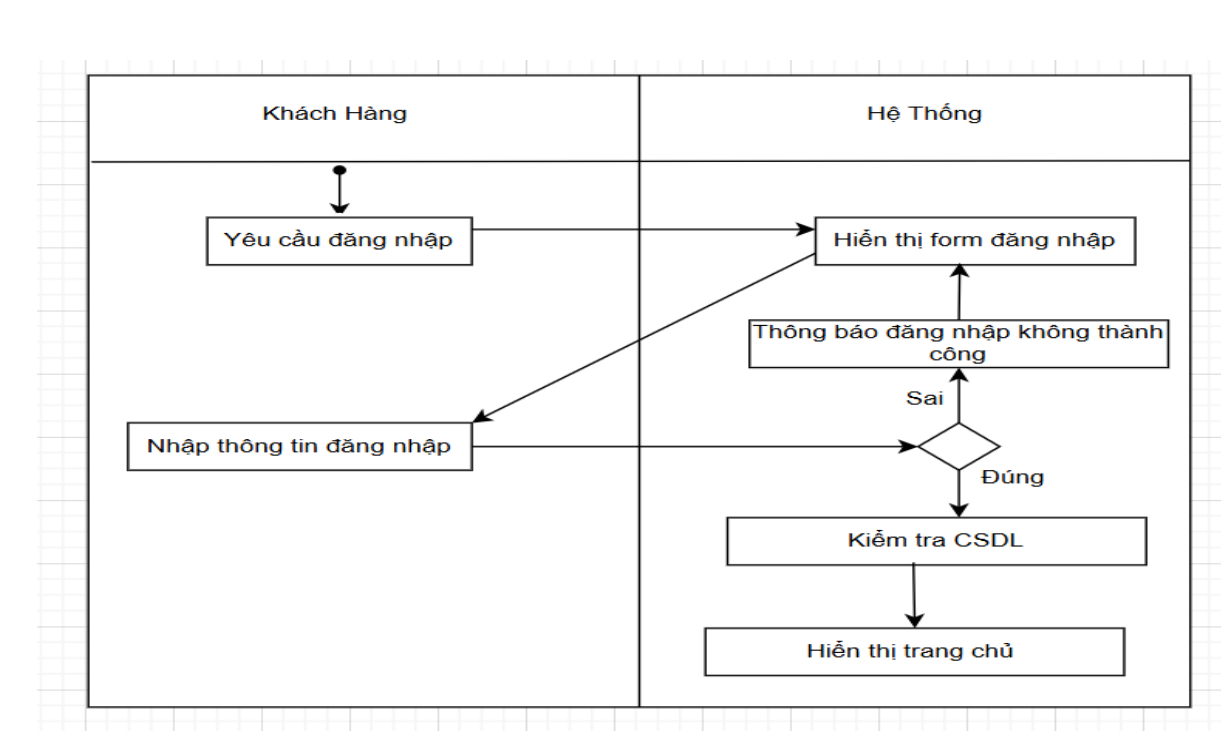
3.2.5 Sơ đồ hoạt động chức năng

3.2.5.1 Sơ đồ hoạt động chức năng đăng ký



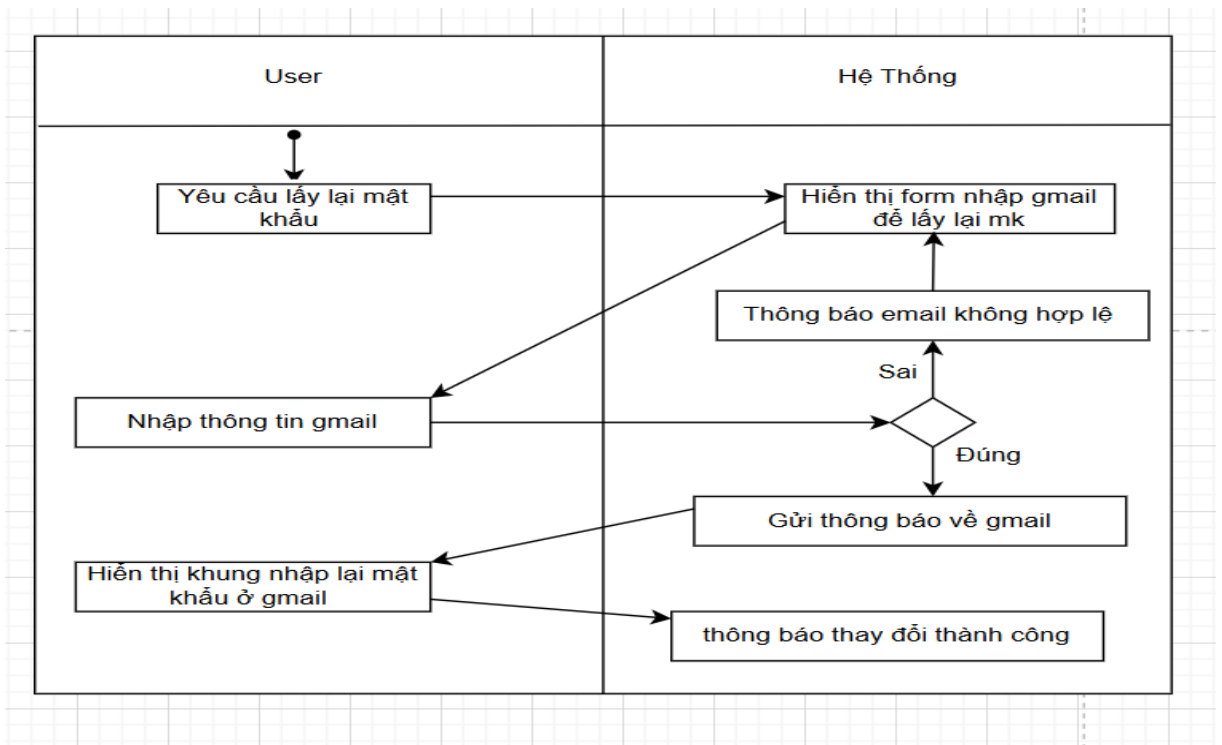
Hình 3.12 Sơ đồ hoạt động chức năng đăng ký

3.2.5.2 Sơ đồ hoạt động chức năng đăng nhập



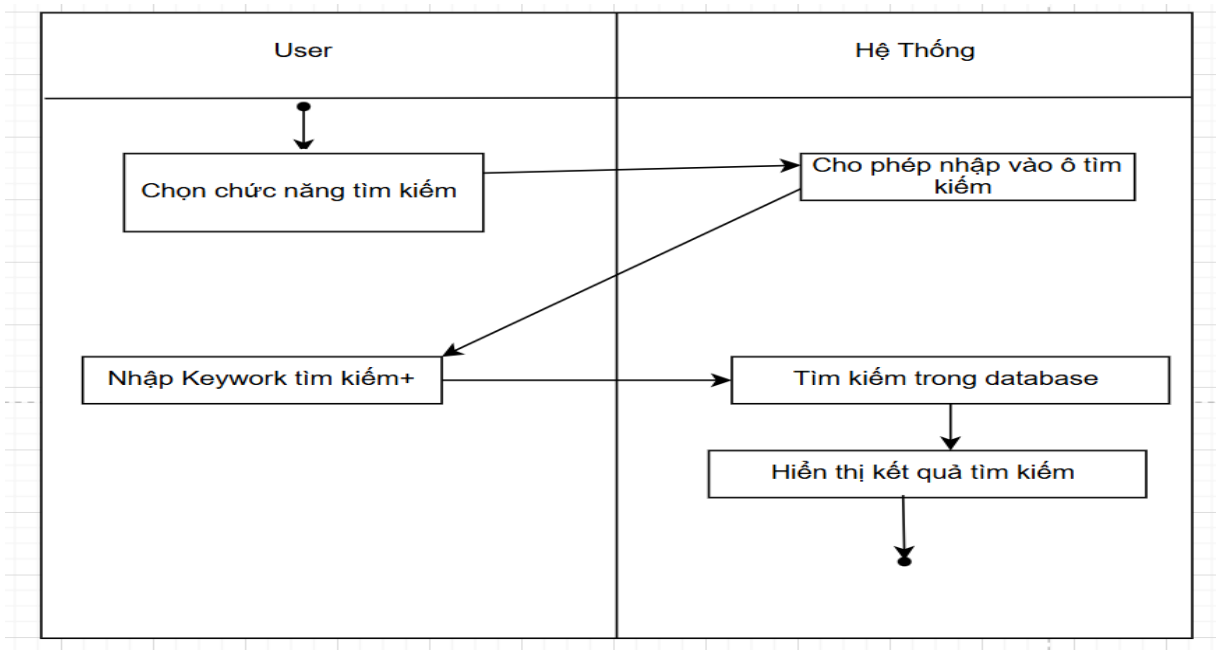
Hình 3.13 Sơ đồ hoạt động chức năng đăng nhập

3.2.5.3 Sơ đồ hoạt động chức năng quên mật khẩu



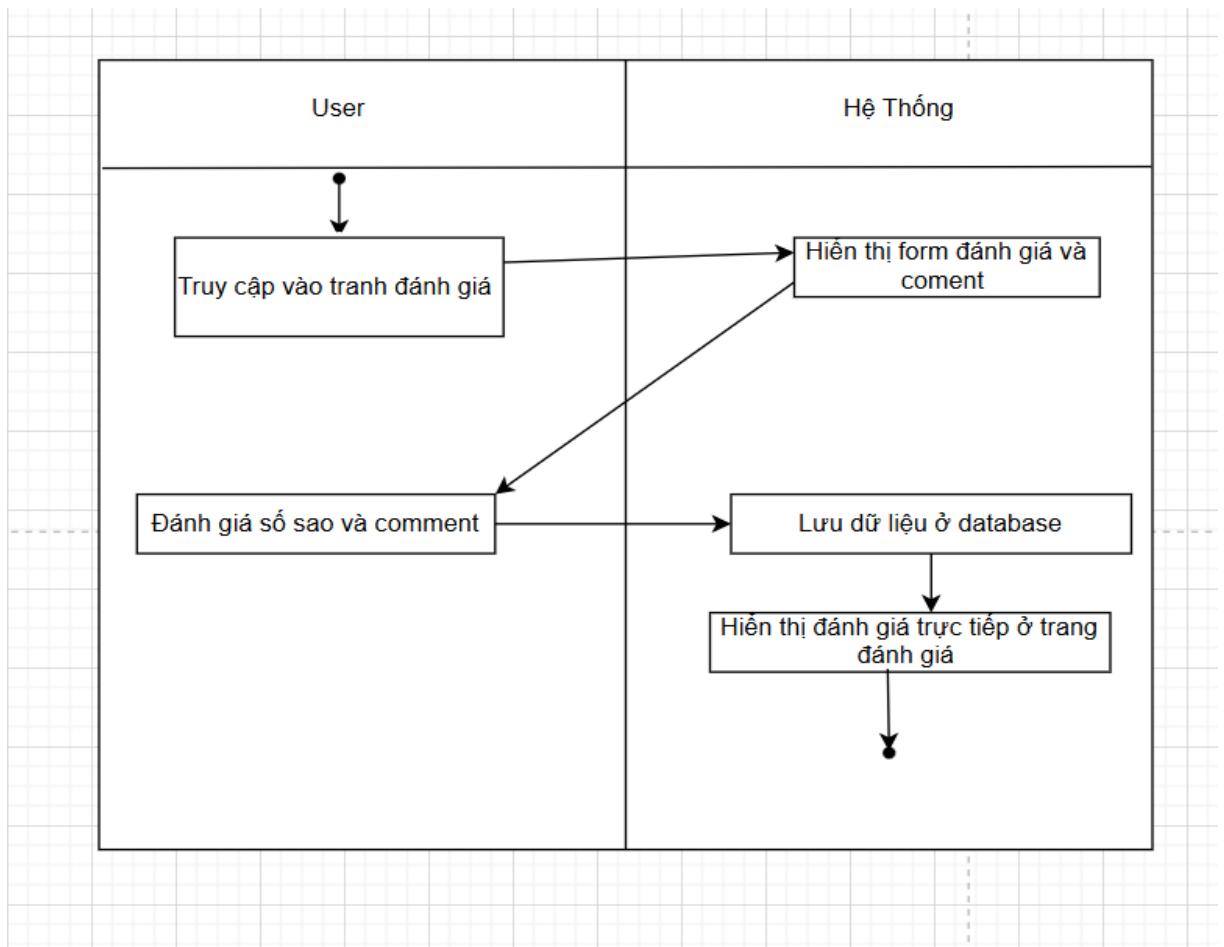
Hình 3.14 Sơ đồ hoạt động chức năng thay đổi mật khẩu

3.2.5.4 Sơ đồ hoạt động chức năng tìm kiếm



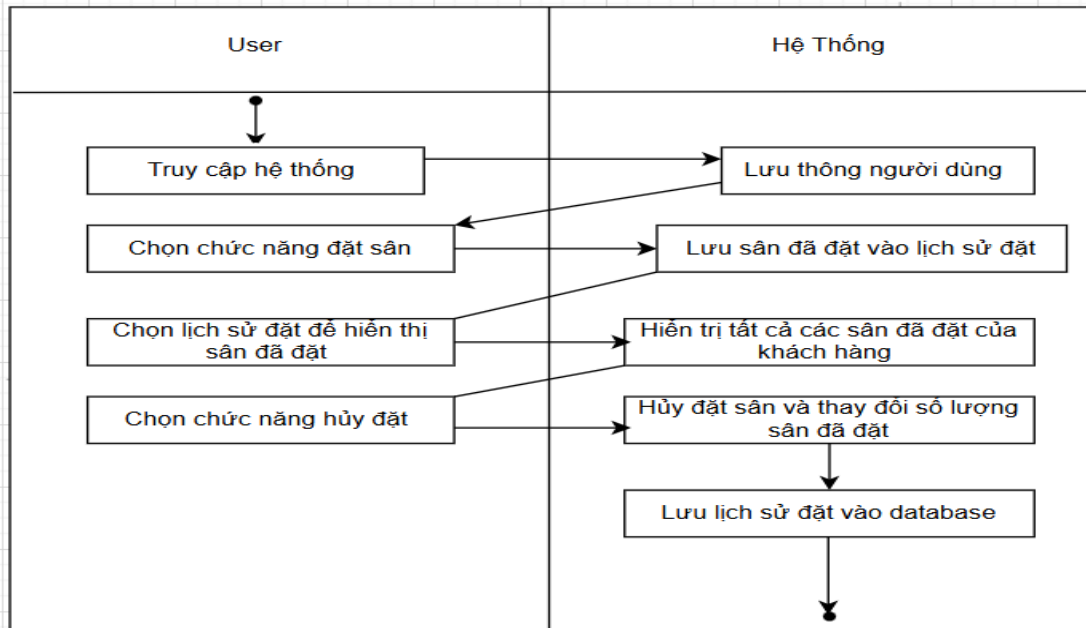
Hình 3.15 Sơ đồ hoạt động chức năng tìm kiếm

3.2.5.5 Sơ đồ hoạt động chức năng đánh giá và bình luận



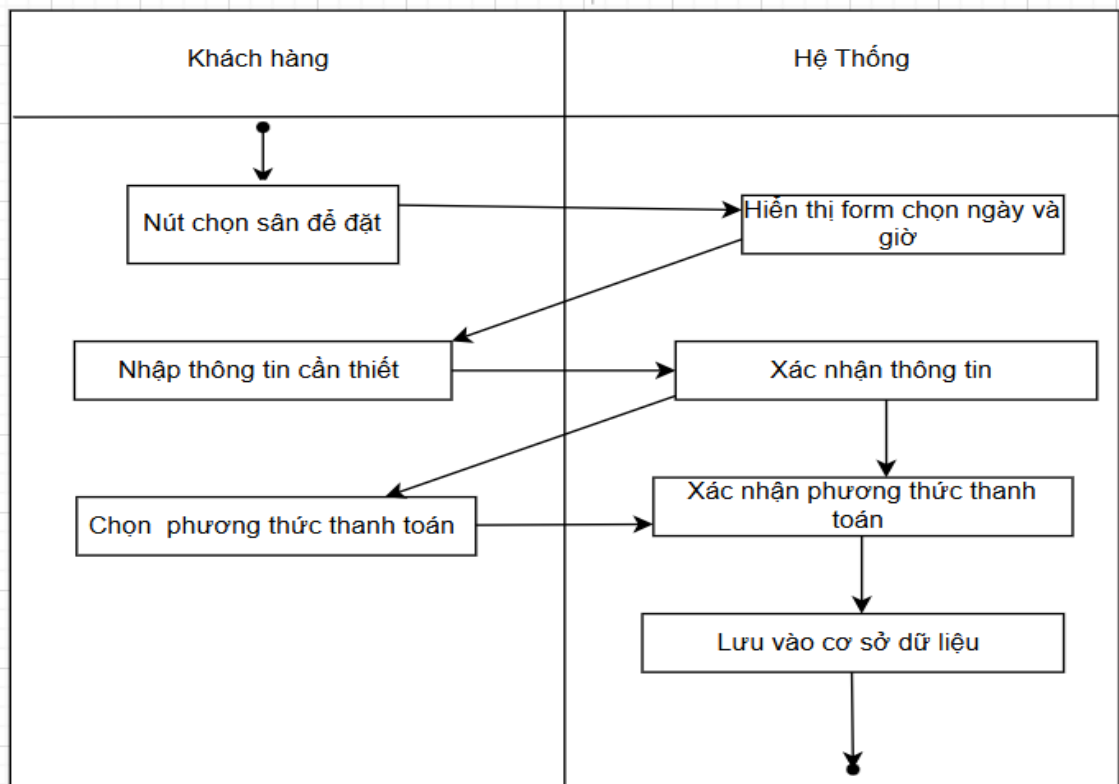
Hình 3.16 Sơ đồ hoạt động chức năng đánh giá và bình luận

3.2.5.6 Sơ đồ hoạt động chức năng lịch sử đặt



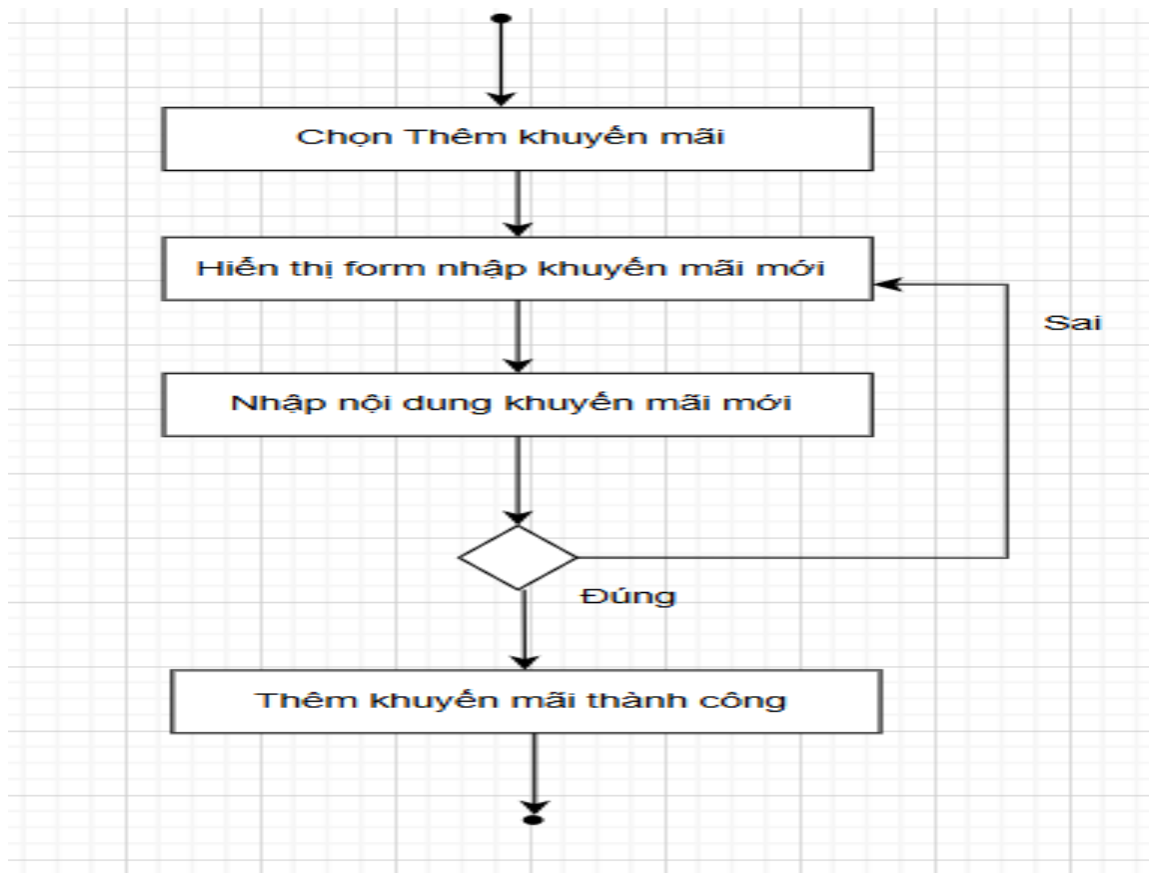
Hình 3.17 Sơ đồ hoạt động chức năng Lịch sử đặt

3.2.5.7 Sơ đồ hoạt động chức năng đặt sân



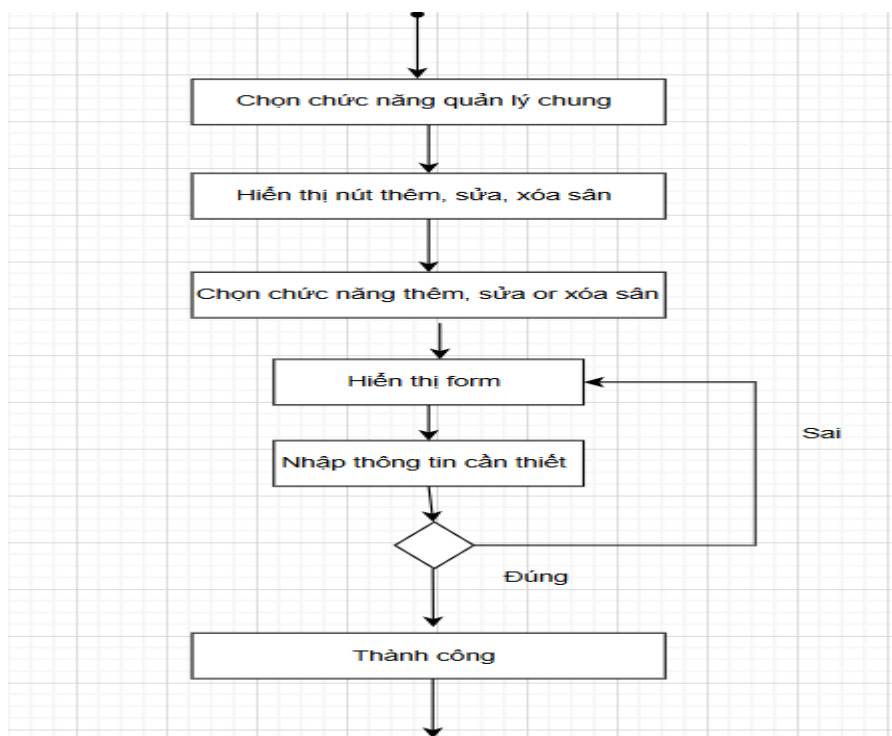
Hình 3.18 Sơ đồ hoạt động chức năng đặt sân

3.2.5.8 Sơ đồ hoạt động chức năng Khuyến mãi



Hình 3.19 Sơ đồ hoạt động chức năng Khuyến mãi

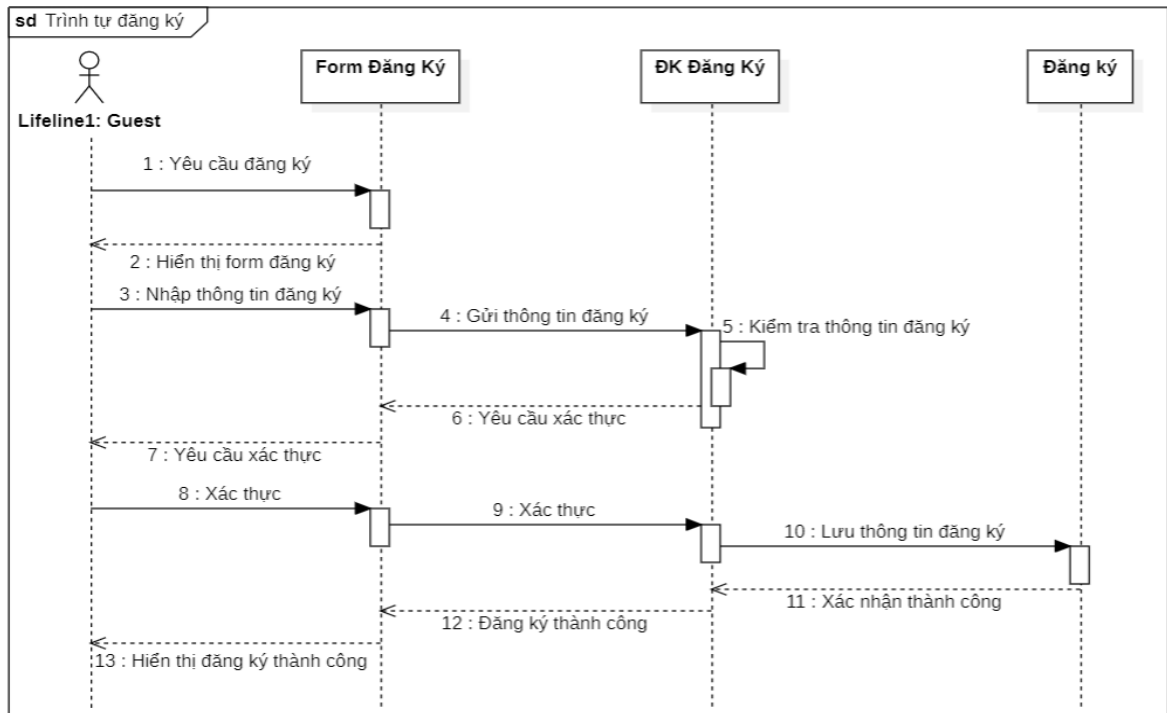
3.2.5.9 Sơ đồ hoạt động chức năng quản lý sân



Hình 3.20 Sơ đồ hoạt động chức năng quản lý sân

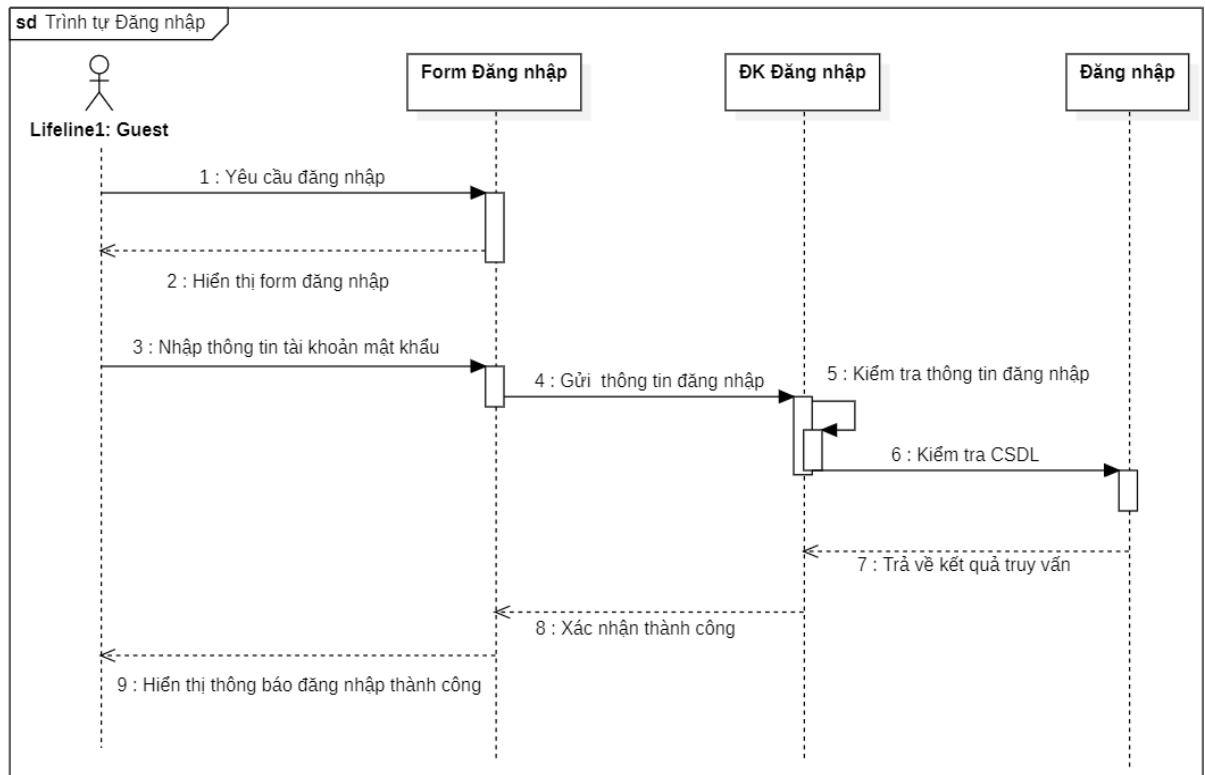
3.2.6 Biểu đồ tuần tự

3.2.6.1 Biểu đồ tuần tự chức năng đăng kí



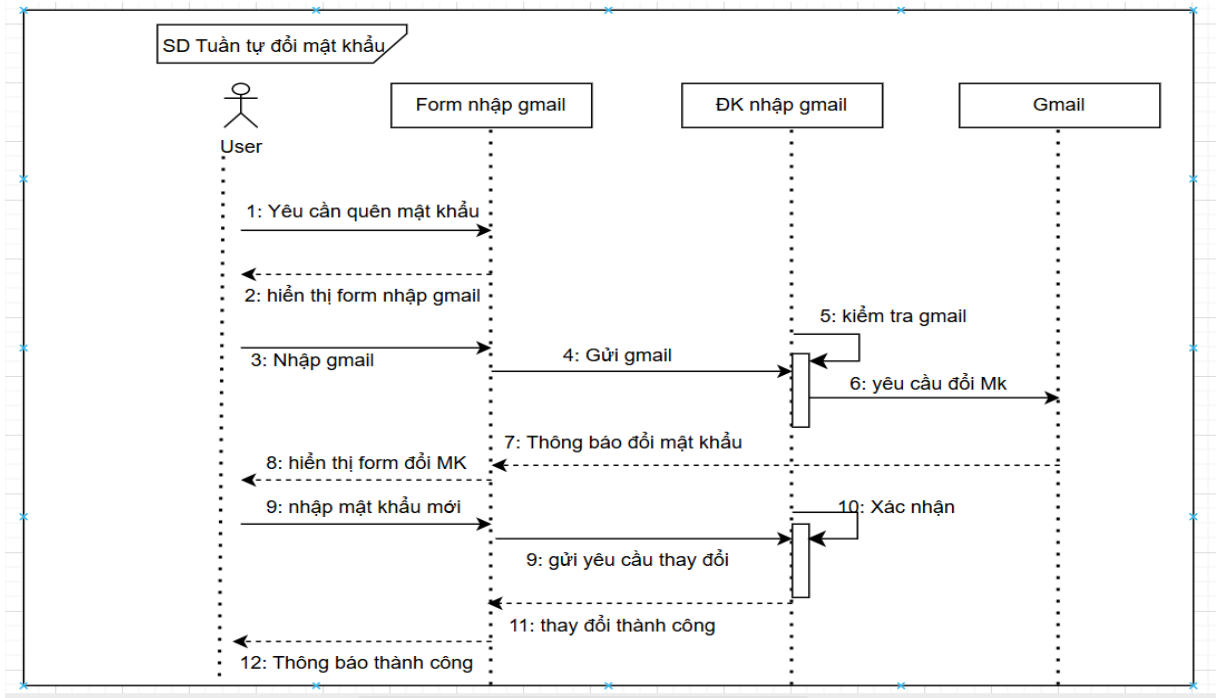
Hình 3.21 Sơ đồ tuần tự chức năng đăng ký

3.2.6.2 Biểu đồ tuần tự chức năng đăng nhập



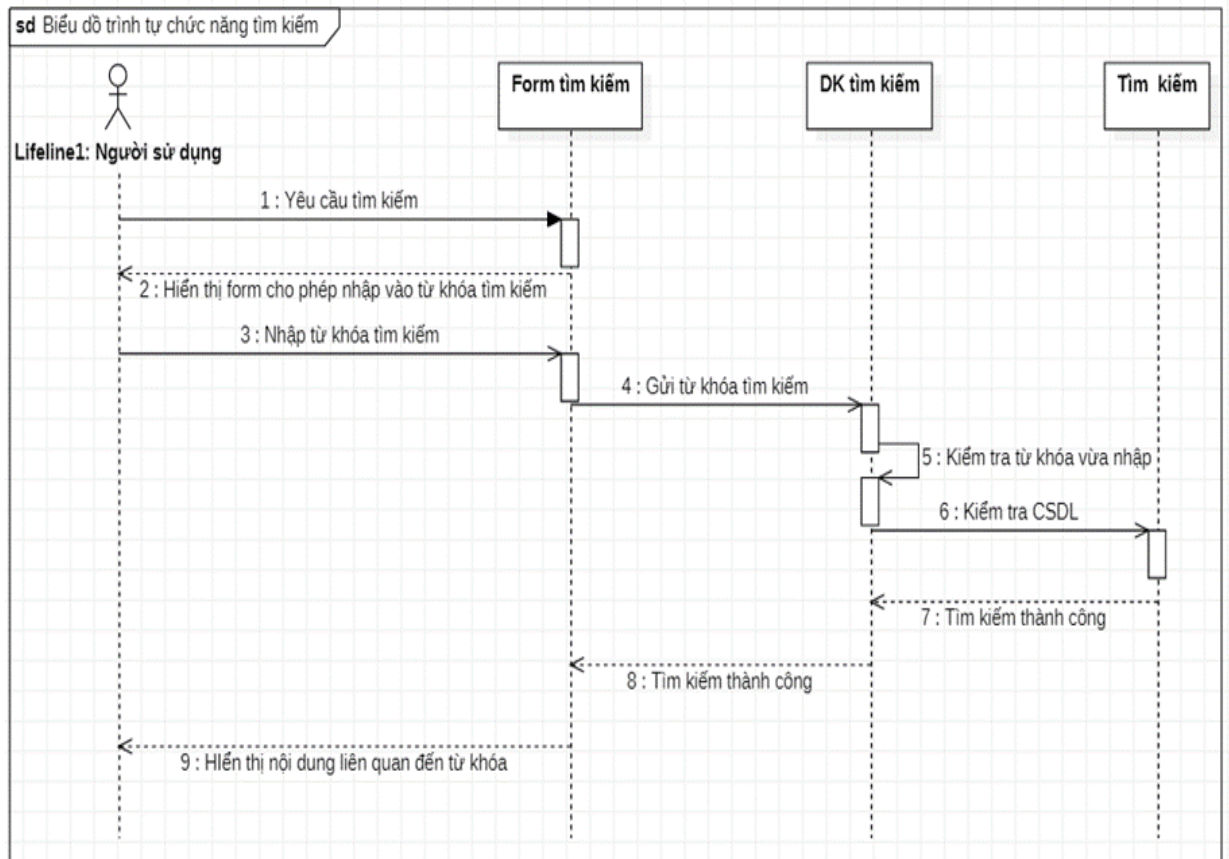
Hình 3.22 Sơ đồ tuần chức năng đăng nhập

3.2.6.3 Biểu đồ tuần tự chức năng thay đổi mật khẩu



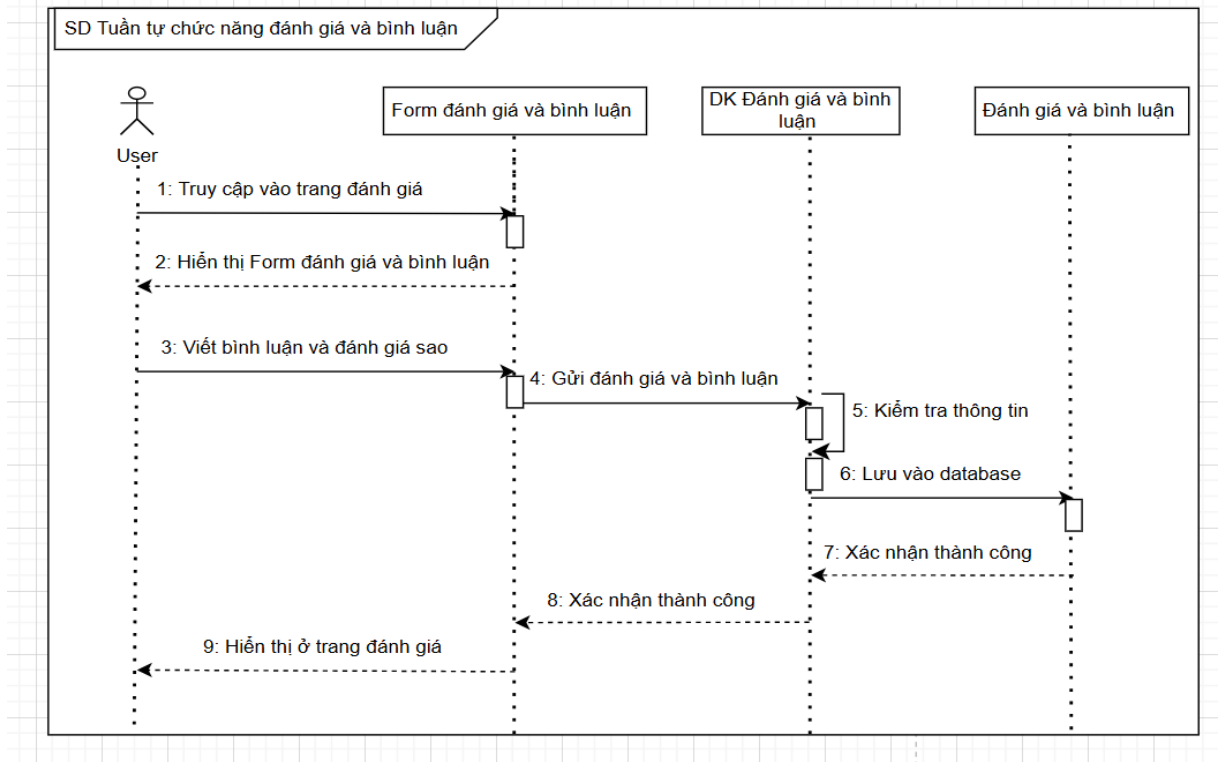
Hình 3.23 Sơ đồ tuần tự chức năng thay đổi mật khẩu

3.2.6.4 Biểu đồ tuần tự chức năng tìm kiếm



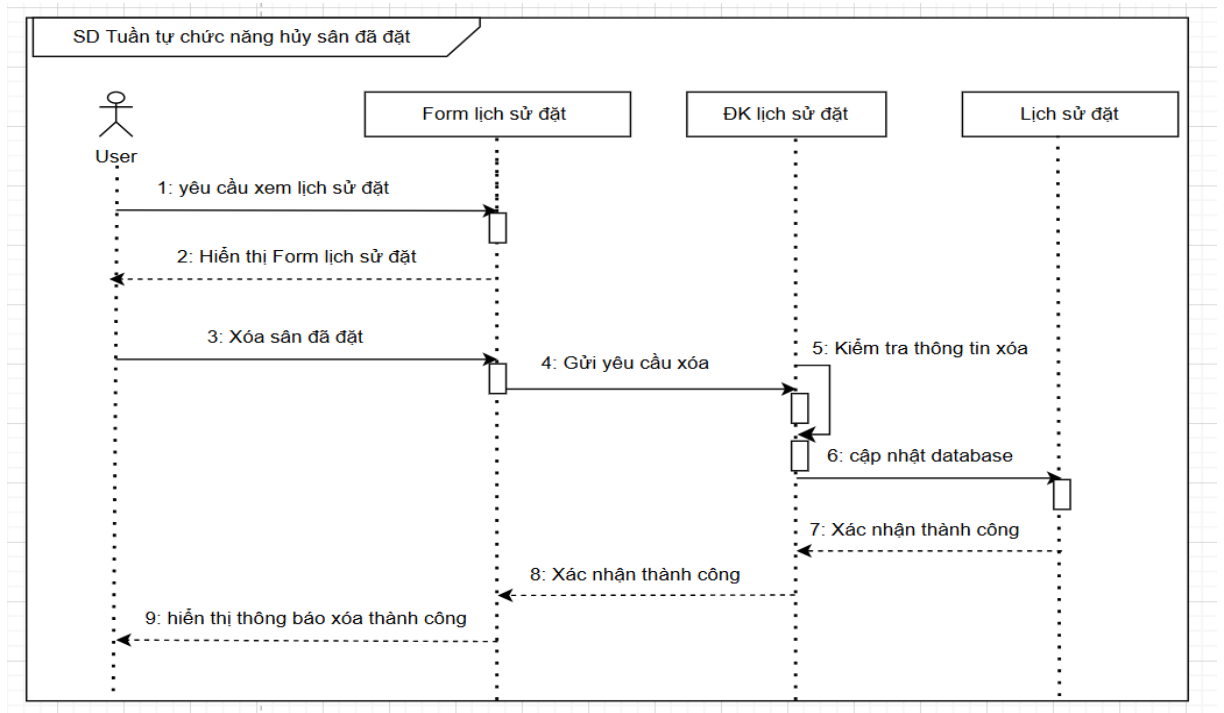
Hình 3.24 Sơ đồ tuần tự chức năng tìm kiếm

3.2.6.5 Biểu đồ tuần tự chức năng bình luận và đánh giá



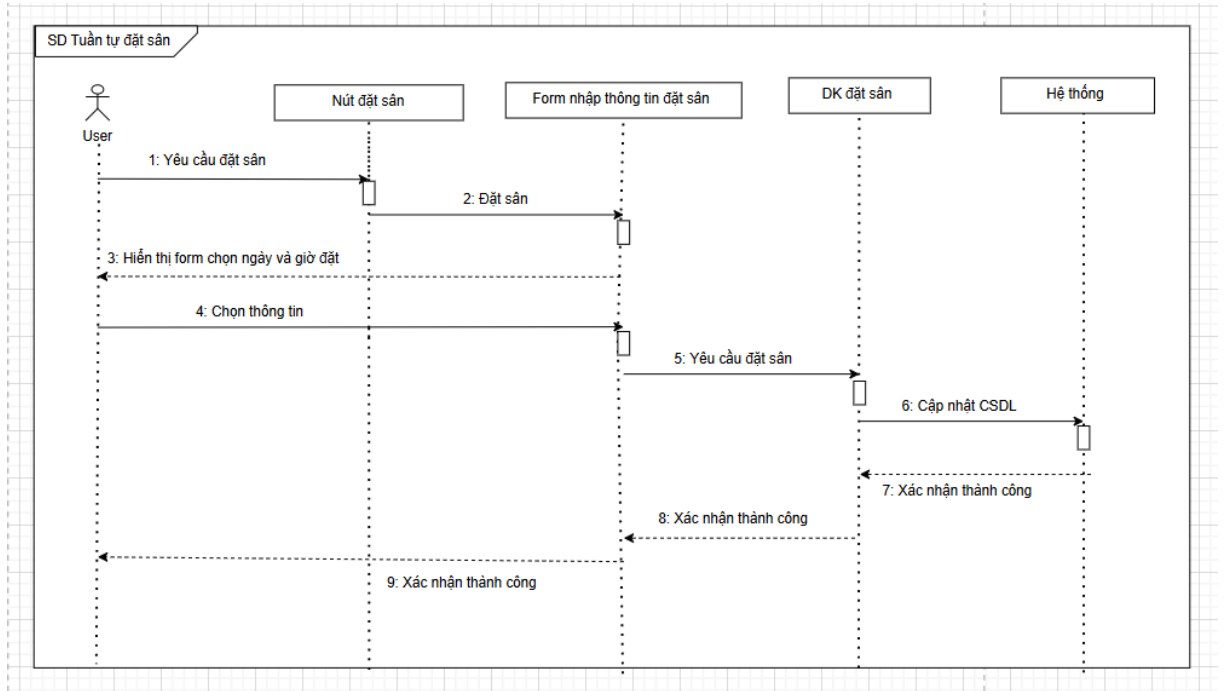
Hình 3.25 Sơ đồ tuần tự chức năng đánh giá và bình luận

3.2.6.6 Biểu đồ tuần tự chức năng hủy sân



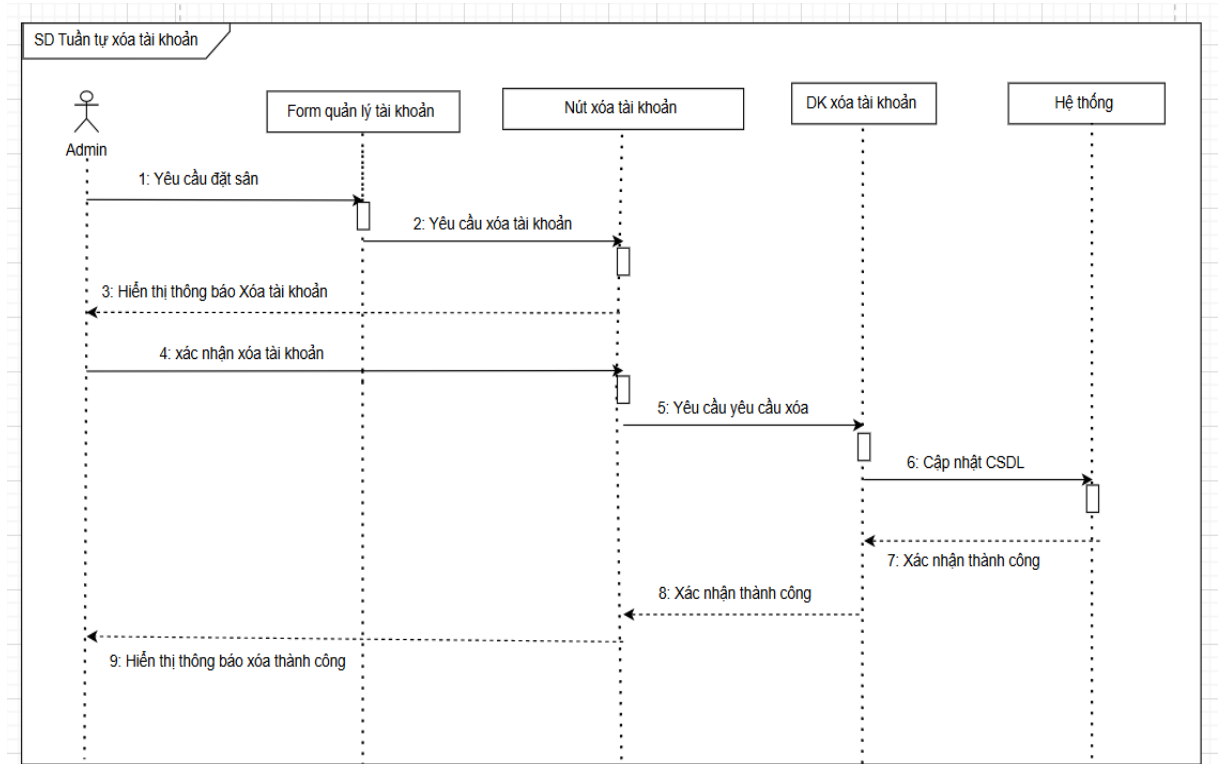
Hình 3.26 Sơ đồ tuần tự chức năng hủy sân

3.2.6.7 Biểu đồ tuần tự chức năng đặt sân



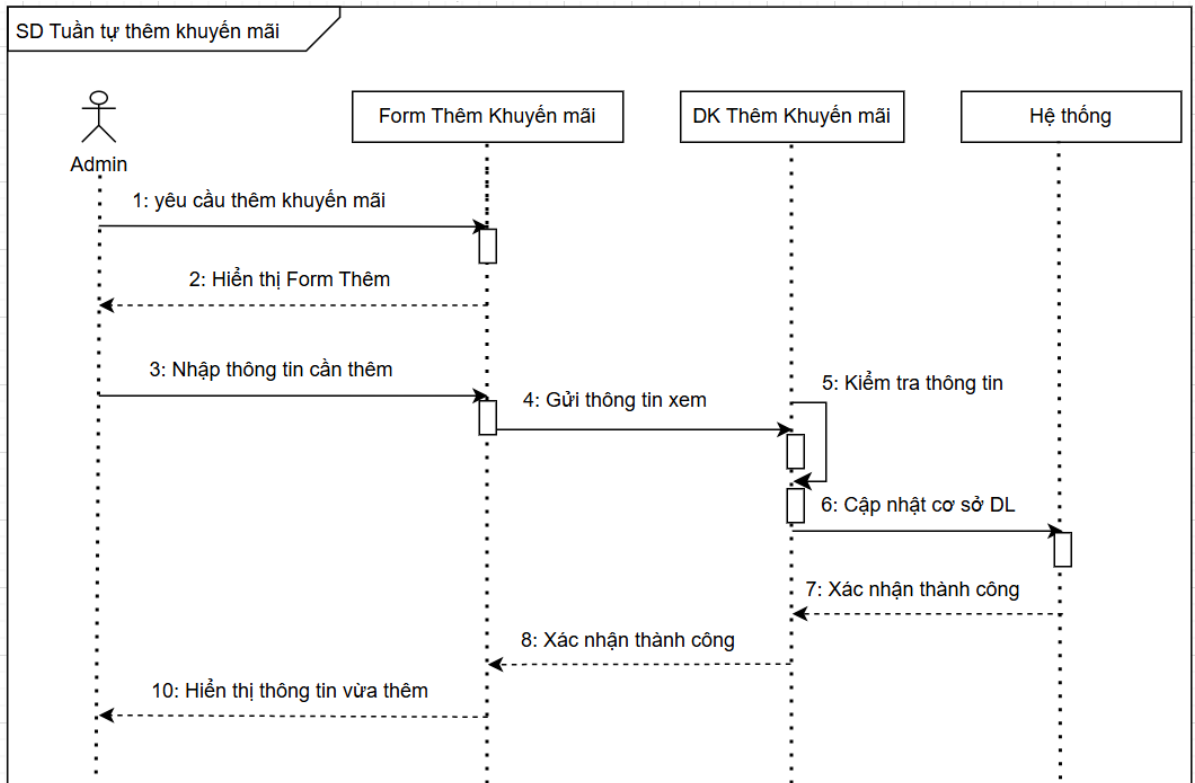
Hình 3.27 Sơ đồ tuần tự chức năng đặt sân

3.2.6.8 Biểu đồ tuần tự chức năng xóa tài khoản



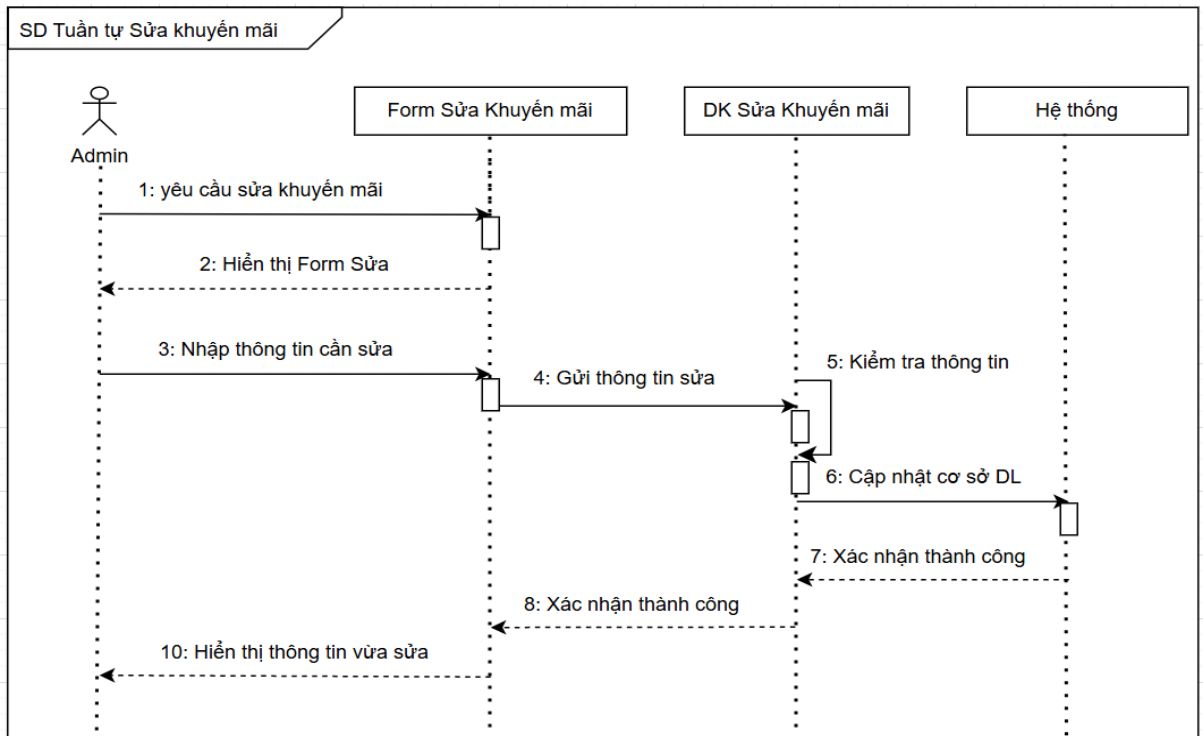
Hình 3.28 Sơ đồ tuần tự chức năng xóa tài khoản

3.2.6.9 Biểu đồ tuần tự chức năng thêm Khuyến mãi



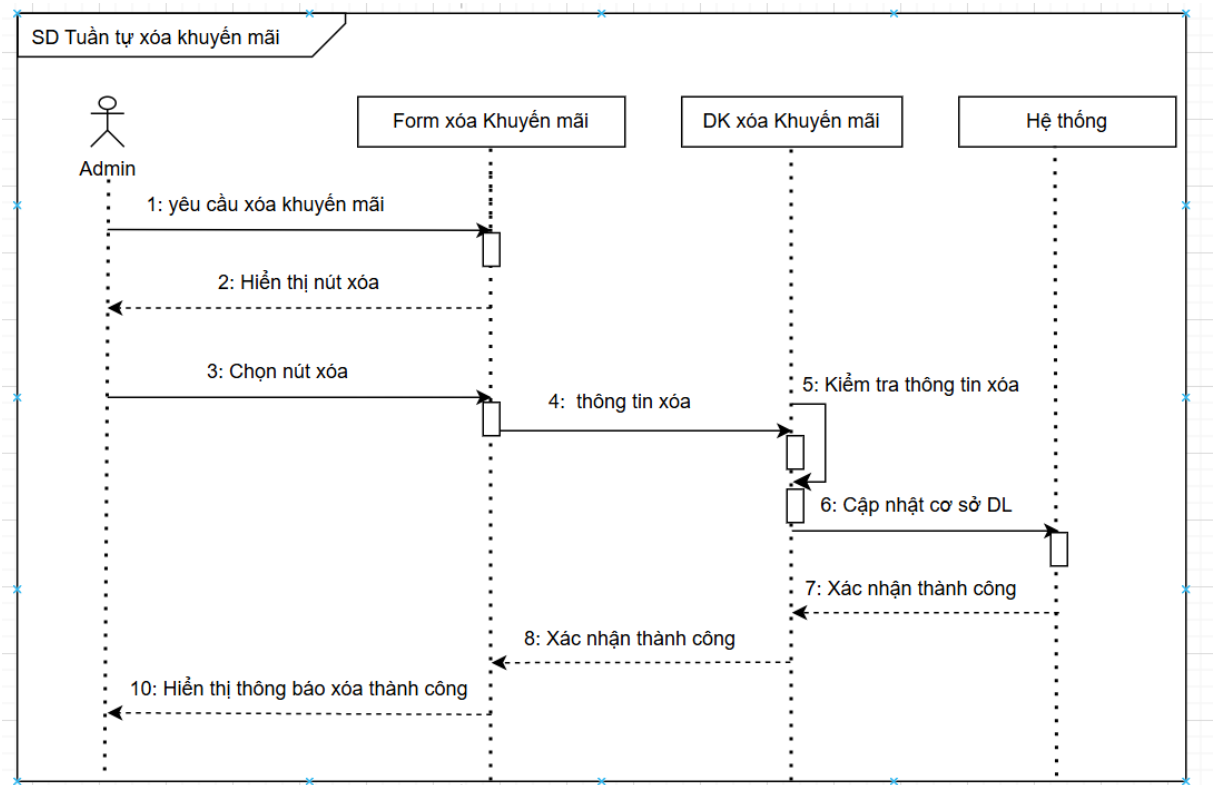
Hình 3.29 Sơ đồ tuần tự chức năng thêm Khuyến mãi

3.2.6.10 Biểu đồ tuần tự chức năng sửa Khuyến mãi



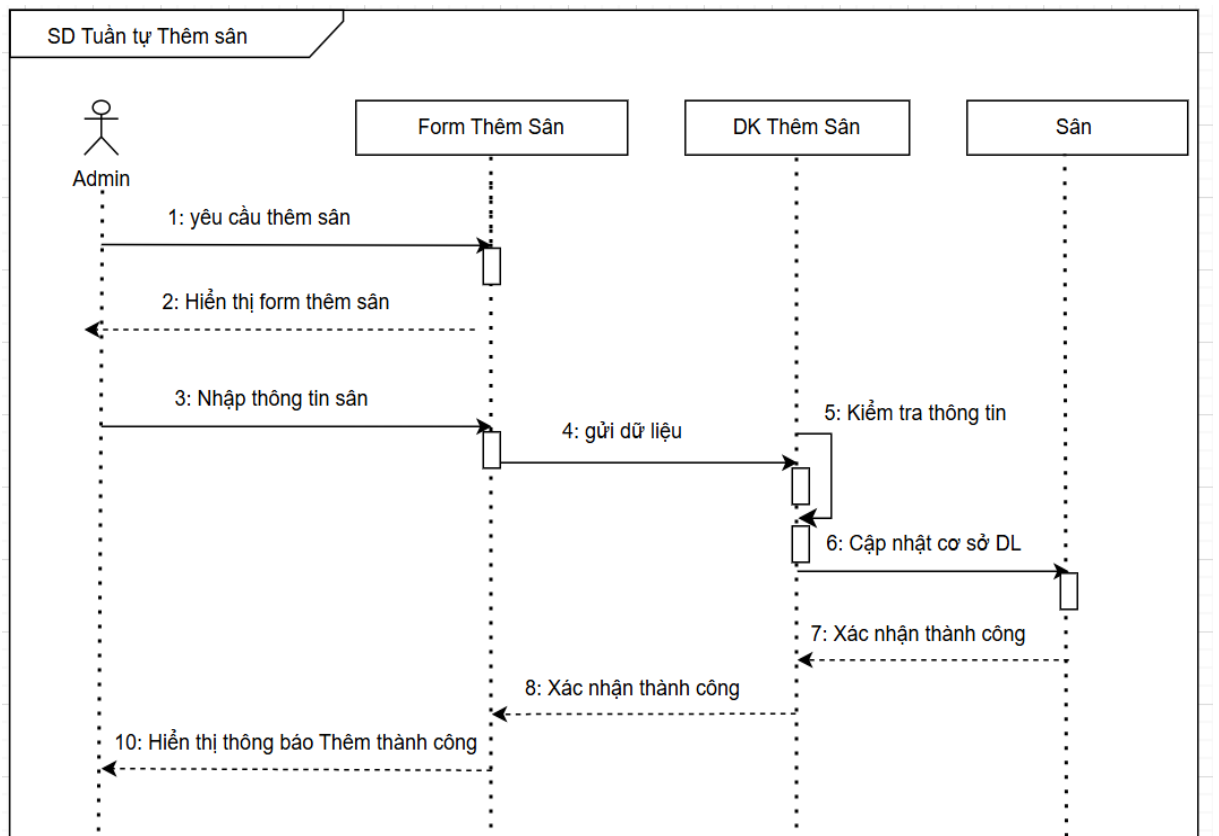
Hình 3.30 Sơ đồ tuần tự chức năng sửa Khuyến mãi

3.2.2.11 Biểu đồ tuần tự chức năng xóa khuyến mãi



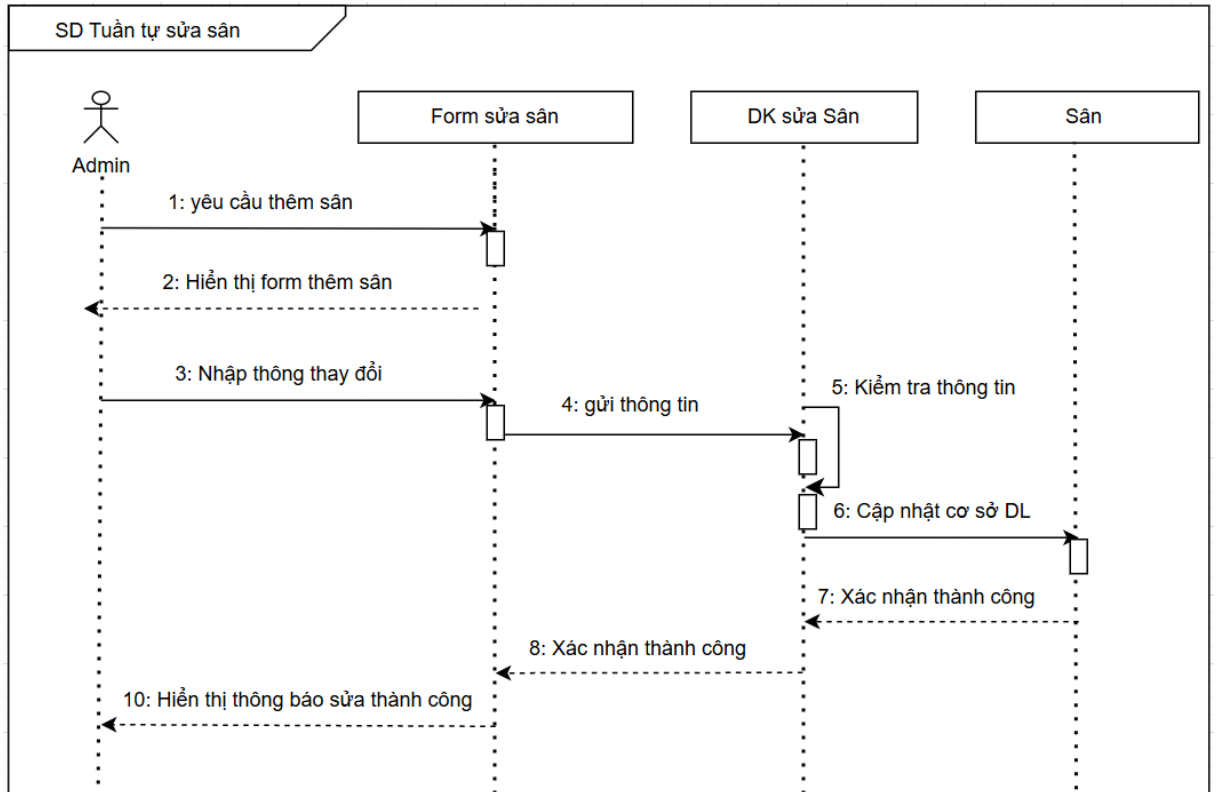
Hình 0.31 Sơ đồ tuần tự chức năng xóa khuyến mãi

3.2.6.12 Biểu đồ tuần tự chức năng thêm sân



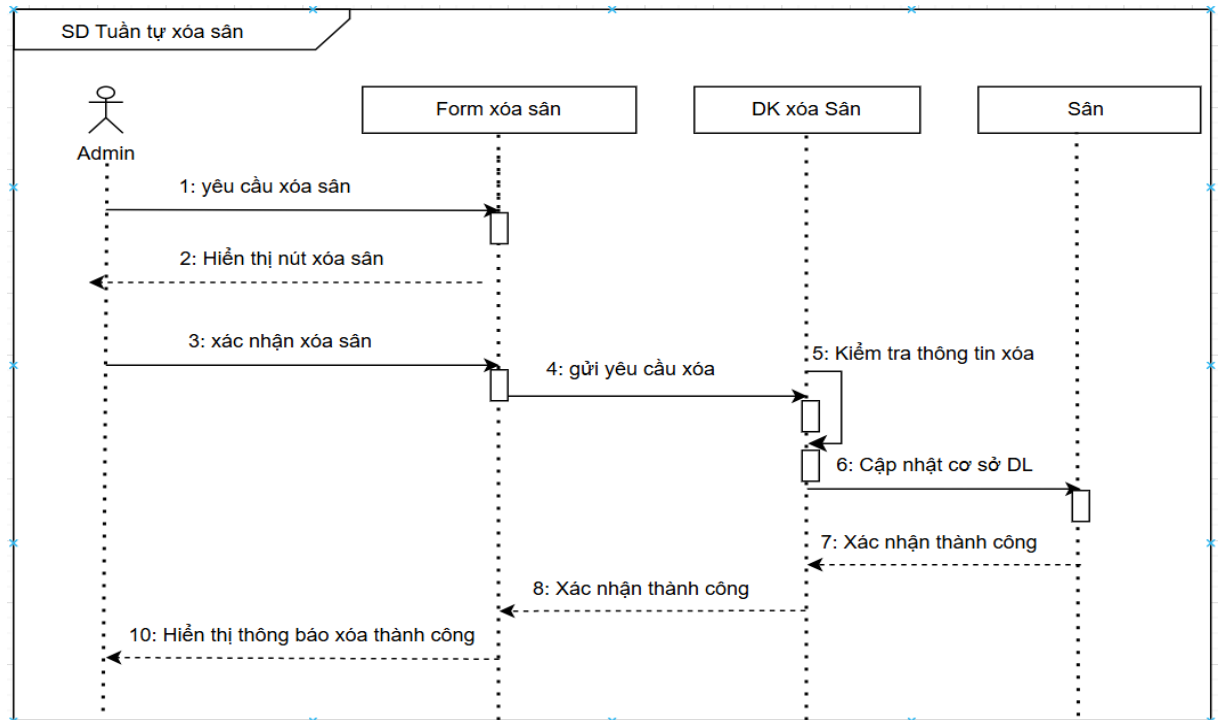
Hình 3.32 Sơ đồ tuần tự chức năng thêm sân

3.2.6.13 Biểu đồ tuần tự chức năng sửa sân



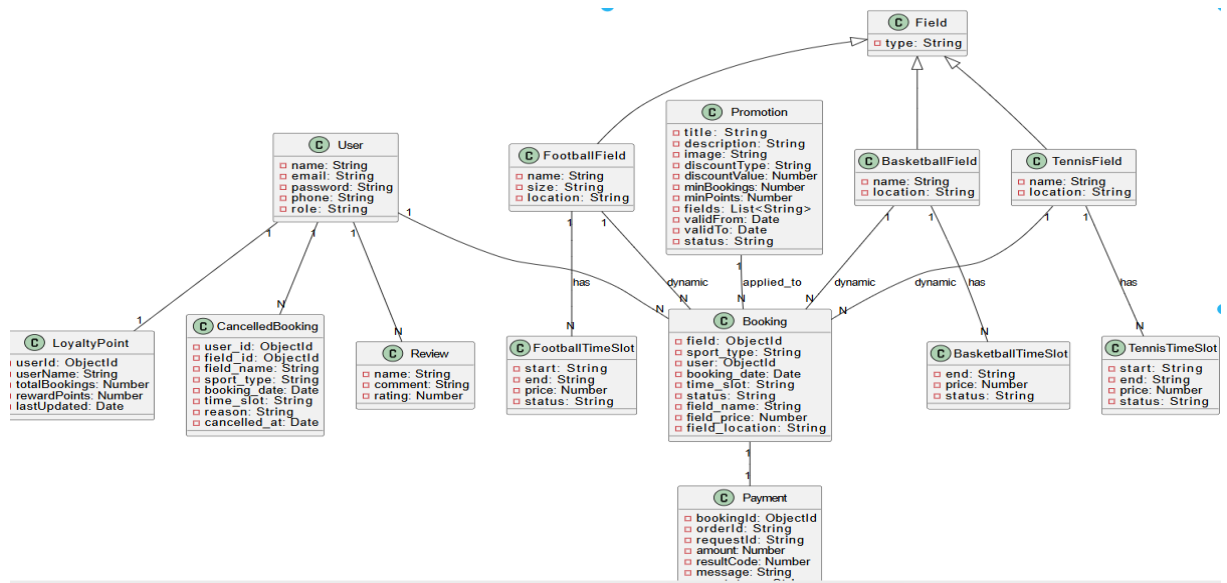
Hình 3.33 Sơ đồ tuần tự chức năng sửa sân

3.2.6.14 Biểu đồ tuần tự chức năng xóa sân



Hình 3.34 Sơ đồ tuần tự chức năng xóa sân

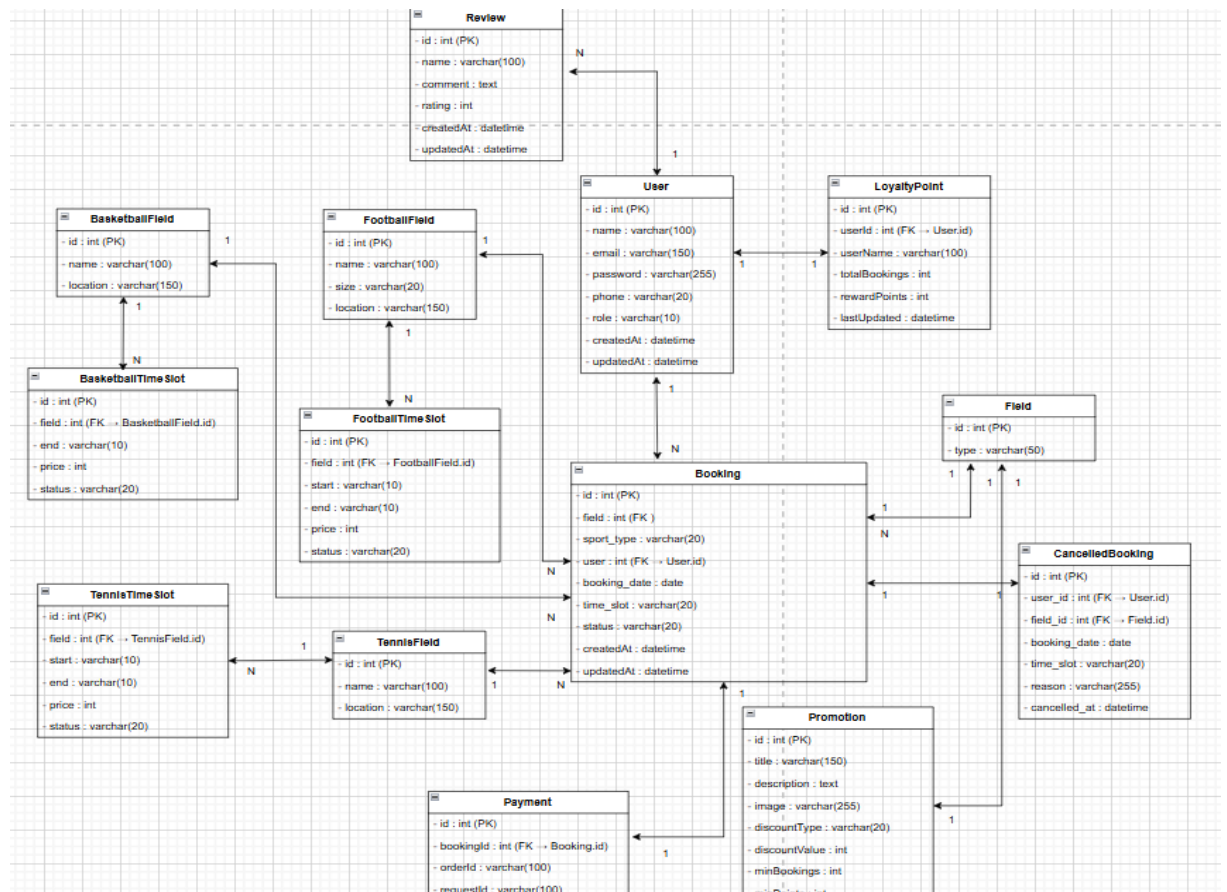
3.3 Sơ đồ lớp



Hình 3.35 Sơ đồ lớp

3.4 Thiết kế cơ sở dữ liệu

3.4.1 Sơ đồ ERD



3.4.1 Tất Cả các bảng

Bảng 10 Tất cả các bảng trong cơ sở dữ liệu

STT	Tên các bảng	Mô tả
1	User	Tài khoản người dùng & admin
2	FootballField	Bảng Sân bóng đá
3	BasketballField	Bảng Sân bóng rổ
4	TennisField	Bảng Sân tennis
5	Field	Loại sân (Bóng đá / Bóng rổ / Tennis)
6	FootballTimeSlot	Khung giờ sân bóng đá
7	BasketballTimeSlot	Khung giờ sân bóng rổ
8	TennisTimeSlot	Khung giờ sân tennis
9	Booking	Lưu đơn đặt sân
10	CancelledBooking	Lịch sử đặt sân bị hủy
11	Payment	Lịch sử thanh toán
12	Promotion	Khuyến mãi / Voucher
13	LoyaltyPoint	Điểm thưởng người dùng

14	Review	Đánh giá & bình luận sân
----	--------	--------------------------

Bảng Bảng User

Bảng 11 Bảng user

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	name	String	Tên người dùng
2	email	String, unique, required	Email đăng nhập
3	password	String, required	Mật khẩu
4	phone	String	Số điện thoại
5	role	String (user/admin)	Quyền
6	createdAt	datetime	Ngày tạo
7	updatedAt	datetime	Ngày cập nhật

Bảng LoyaltyPoint

Bảng 12 Bảng LoyaltyPoint

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	userId	ObjectId → User	Khóa ngoại
2	userName	String	Tên người dùng
3	totalBookings	Number	Tổng lượt đặt sân
4	rewardPoints	Number	Điểm thưởng
5	lastUpdated	Date	Lần cập nhật cuối

Bảng FootballField

Bảng 13 FootballField

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	name	String, required	Tên sân bóng đá
2	size	String (5 người / 7 người)	Kích thước sân
3	location	String	Vị trí sân
4	createdAt	datetime	Ngày tạo
5	updatedAt	datetime	Ngày cập nhật

Bảng FootballTimeSlot

Bảng 14 Bảng FootballTimeSlot

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	start	String, required	Giờ bắt đầu
2	end	String, required	Giờ kết thúc
3	price	Number, required	Giá thuê
4	status	String (available/booked)	Trạng thái
5	createdAt	datetime	Ngày tạo
6	updatedAt	datetime	Ngày cập nhật

Bảng BasketballField

Bảng 15 Bảng BasketballField

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	name	String, required	Tên sân bóng rổ
2	location	String	Vị trí sân
3	createdAt	datetime	Ngày tạo
4	updatedAt	datetime	Ngày cập nhật

Bảng BasketballTimeSlot

Bảng 16 Bảng BasketballTimeSlot

STT	Tên trường	Kiểu dữ liệu	Mô tả
-----	------------	--------------	-------

1	start	String, required	Giờ bắt đầu
2	end	String, required	Giờ kết thúc
3	price	Number	Giá thuê (mọi giờ giống nhau)
4	status	String	available/booked
5	createdAt	datetime	Ngày tạo
6	updatedAt	datetime	Ngày cập nhật

Bảng TennisField

Bảng 17 Bảng TennisField

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	name	String, required	Tên sân tennis
2	location	String	Vị trí sân
3	createdAt	datetime	Ngày tạo
4	updatedAt	datetime	Ngày cập nhật

Bảng TennisTimeSlot

Bảng 18 Bảng TennisTimeSlot

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	start	String	Giờ bắt đầu
2	end	String	Giờ kết thúc
3	price	Number	Giá thuê
4	status	String (available/booked)	Trạng thái
5	createdAt	datetime	Ngày tạo
6	updatedAt	datetime	Ngày cập nhật

Bảng Field

Bảng 19 Bảng Field

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	type	String	Loại môn thể thao
2	createdAt	datetime	Ngày tạo
3	updatedAt	datetime	Ngày cập nhật

Bảng Booking

Bảng 20 Bảng Booking

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	field_id	String	Mã sân
2	sport_type	String	Loại môn: football/basketball/tennis
3	user	ObjectId → User	Người đặt
4	booking_date	Date	Ngày đặt sân
5	time_slot	String	Giờ đặt
6	status	String (booked)	Trạng thái
7	field_name	String	Tên sân
8	field_price	Number	Giá sân
9	field_location	String	Vị trí sân
10	createdAt	datetime	Ngày tạo
11	updatedAt	datetime	Ngày cập nhật

Bảng CancelledBooking

Bảng 21 Bảng CancelledBooking

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	user_id	ObjectId → User	Người hủy
2	field_id	ObjectId → Field	Sân

3	field_name	String	Tên sân
4	sport_type	String	Loại môn
5	booking_date	Date	Ngày đặt
6	time_slot	String	Giờ
7	reason	String	Lý do hủy
8	cancelled_at	Date	Ngày hủy
9	createdAt	datetime	Ngày tạo
10	updatedAt	datetime	Ngày cập nhật

Bảng Payment

Bảng 22 Bảng Payment

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	bookingId	ObjectId Booking →	Mã đơn đặt
2	orderId	String	Mã đơn MoMo
3	requestId	String	requestId MoMo
4	amount	Number	Số tiền thanh toán
5	transId	String	Mã giao dịch MoMo
6	resultCode	Number	Mã kết quả
7	message	String	Tin nhắn phản hồi
8	extraData	Object	Dữ liệu thêm
9	sport_type	String	Loại sân
10	createdAt	datetime	Ngày tạo

Bảng Promotion

Bảng 23 Bảng Promotion

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	title	String	Tiêu đề

2	description	String	Mô tả
3	image	String	Ảnh khuyến mãi
4	discountType	String (percent/fixed/none)	Kiểu giảm giá
5	discountValue	Number	Giá trị giảm
6	minBookings	Number	Số lượt đặt tối thiểu
7	minPoints	Number	Điểm thưởng tối thiểu
8	fields	Array(String)	Loại sân áp dụng
9	validFrom	Date	Hiệu lực từ
10	validTo	Date	Hiệu lực đến
11	status	String (active/inactive)	Trạng thái
12	createdAt	datetime	Ngày tạo
13	updatedAt	datetime	Ngày cập nhật

Bảng Review

Bảng 24 Bảng Review

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	name	String	Tên người đánh giá
2	comment	String	Nội dung đánh giá
3	rating	Number (1–5)	Số sao
4	createdAt	datetime	Ngày tạo
5	updatedAt	datetime	Ngày cập nhật

Phân tích các mối quan hệ

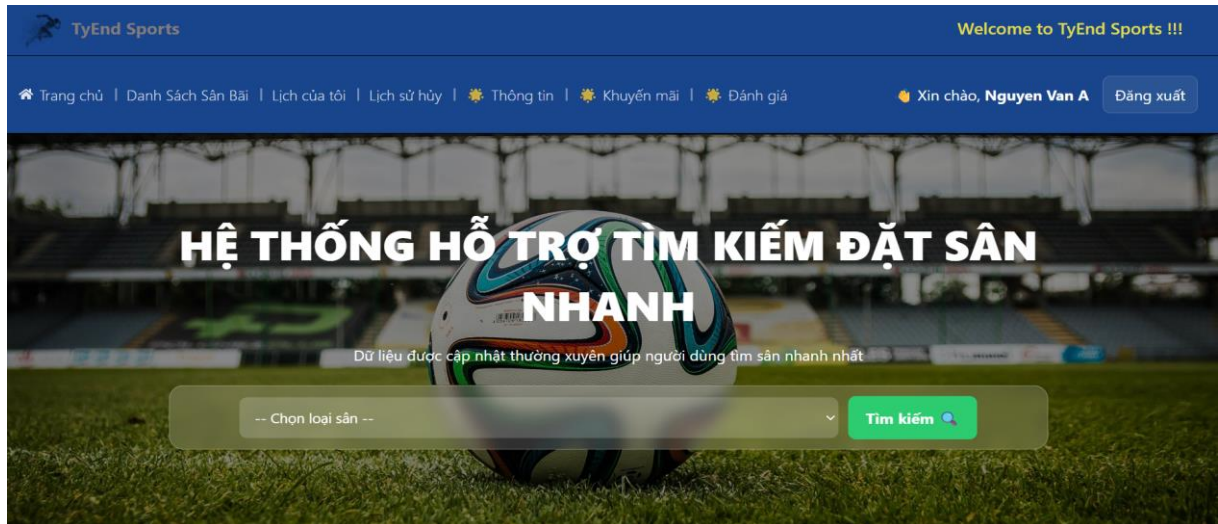
Bảng 25 Bảng phân tích các mối quan hệ

1	User – LoyaltyPoint: 1 – 1
2	User – Booking: 1 – N
3	FootballField – Booking: 1 - N
4	User – Review: 1 – N
5	FootballField – FootballTimeSlot: 1 – N
6	BasketballField – BasketballTimeSlot: 1 – N
7	TennisField – TennisTimeSlot: 1 – N
8	Booking – Payment: 1 – 1
9	Booking – CancelledBooking: 1 – 1
10	Promotion – Field: 1 – N
11	Booking – Field: N – 1
12	BasketballField – Booking: 1-N
13	TennisField – Booking: 1-N

Chương 4: KẾT QUẢ THỰC NGHIỆM

4.1 Xây dựng website demo

4.1.1 Giao diện trang chủ website

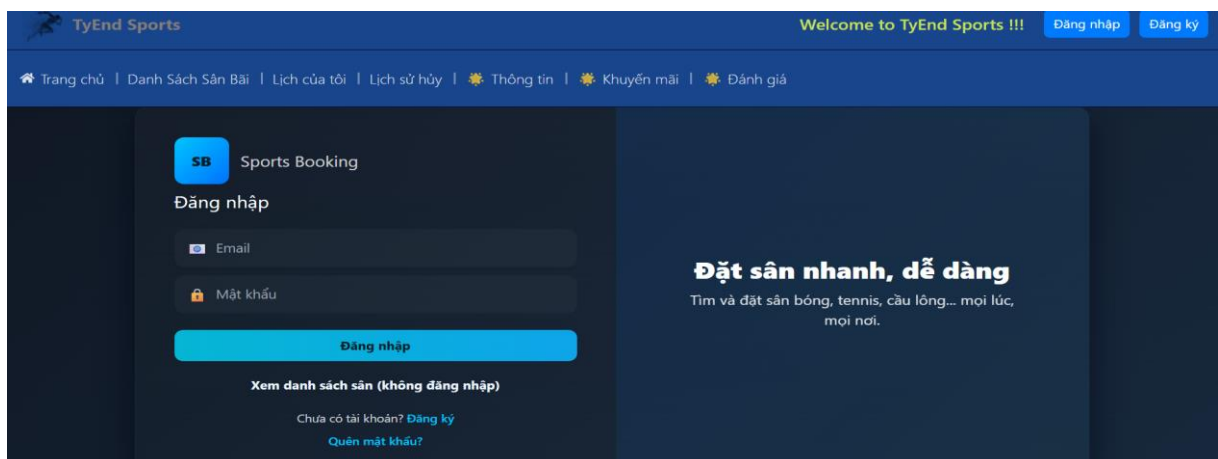


Hình 4.1 Giao diện trang chủ khách hàng

Khách vãng lai có thể xem sân, tìm kiếm sản phẩm theo bộ lọc, danh mục sân, xem thông tin về các thông tin chi tiết sân, khuyến mãi liên quan đến sân bãi, ở đây khách vãng lai cũng có thể đăng kí tài khoản mới để sử dụng mua hàng bằng cách kích vào mục đăng ký trên trang chủ.

Khách hàng đã có tài khoản cũng có thể thực hiện thao tác tương tự khách vãng lai và ấn vào mục đăng nhập để đăng nhập tài khoản sẵn có để sử dụng đầy đủ các tính năng và dịch vụ của hệ thống.

4.1.2 Giao diện trang đăng nhập



Hình 4.2 Giao diện trang đăng nhập

```
router.post("/login", async (req, res) => {
  try {
    const { email, password } = req.body;

    const user = await User.findOne({ email });
    if (!user)
      return res.status(400).json({ message: "Email hoặc mật khẩu không đúng" });

    const match = await bcrypt.compare(password, user.password);
    if (!match)
      return res.status(400).json({ message: "Email hoặc mật khẩu không đúng" });

    const token = jwt.sign({ id: user._id, role: user.role }, process.env.JWT_SECRET, {
      expiresIn: "1d",
    });

    res.json({
      token,
      role: user.role,
      user: {
        id: user._id,
        name: user.name,
        email: user.email,
        role: user.role,
      },
    });
  } catch (err) {
    console.error("Lỗi đăng nhập:", err);
    res.status(500).json({ message: "Lỗi server" });
  }
});
```

Hình 4.3 source code trang đăng nhập

Khách vắng lai chưa có tài khoản hợp lệ không thể đăng nhập vào hệ thống

Ở đây khách hàng điền đầy đủ thông tin tài khoản bao gồm email đăng nhập điền vào mục tên đăng nhập và mật khẩu tương ứng ở mục mật khẩu sau đó nhấp đăng nhập để đăng nhập vào hệ thống, người dùng không được bỏ trống hay điều sau thông tin tên đăng nhập hoặc mật khẩu nếu các trường hợp điền sai hay bỏ trống hệ thống sẽ xử lý thông tin và thông báo lỗi đăng nhập hệ thống thất bại, nếu điền tên đăng nhập và mật khẩu đúng hệ thống xử lý thông báo đăng nhập thành công.

Ở đây người quản trị đăng điền đầy đủ thông tin đăng nhập được cấp quyền quản trị để đăng nhập hệ thống với quyền quản trị viên , nếu để trống hay điền sai các trường thông tin hệ thống xử lý và báo lỗi nếu điền đúng hệ thống xử lý thông báo đăng nhập hệ thống thành công.

4.1.3 Giao diện trang đăng ký

Hình 4.4 Giao diện trang đăng ký

```
router.post("/register", async (req, res) => {
  try {
    const { name, email, password, phone } = req.body;

    const existing = await User.findOne({ email });
    if (existing) {
      return res.status(400).json({ message: "Email đã tồn tại" });
    }

    const hashed = await bcrypt.hash(password, 10);

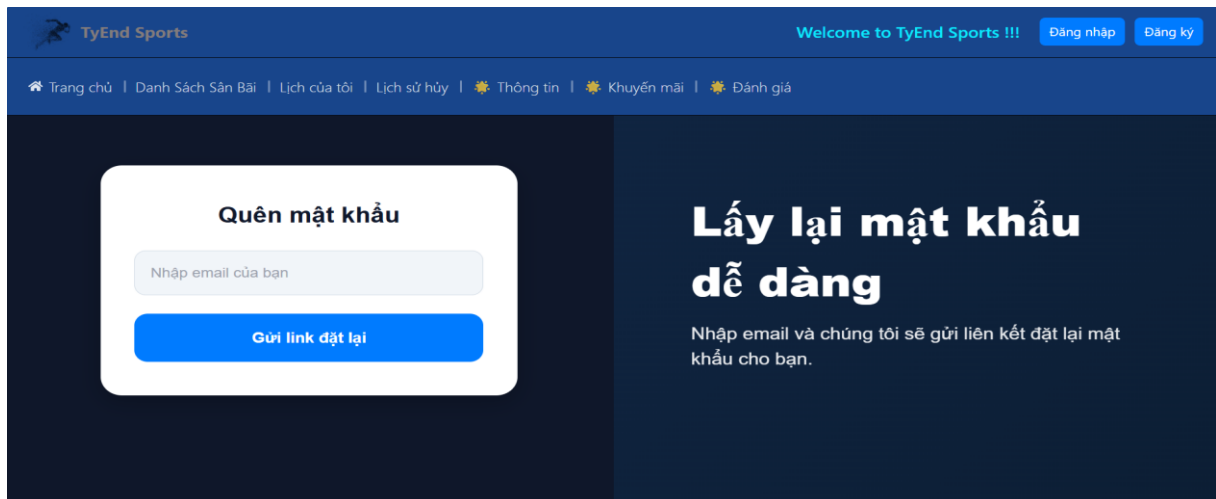
    const user = new User({
      name,
      email,
      password: hashed,
      phone,
      role: "user",
    });
    await user.save();

    res.json({ message: "Đăng ký thành công" });
  } catch (err) {
    console.error("Lỗi đăng ký:", err);
    res.status(500).json({ message: "Lỗi server" });
  }
});
```

Hình 4.5 source code trang đăng kí

Khách muốn sử dụng đầy đủ các tính năng và dịch vụ của hệ thống cần đăng kí tài khoản tại trang đăng kí người dùng cần điền đầy đủ thông tin hợp lệ vào các trường thông tin theo form có sẵn, nếu bỏ trống hay điền thông tin không hợp lệ hệ thống xử lý thông tin và thông báo đăng nhập thất bại, thông báo đăng ký tài khoản thành công khách hàng đi đến trang xem và đặt sân của hệ thống.

4.1.4 Giao diện trang quên mật khẩu



Hình 4.6 Giao diện trang quên mật khẩu

```
/* ----- QUÊN MẬT KHẨU ----- */
router.post("/forgot-password", async (req, res) => {
  try {
    const { email } = req.body;
    const user = await User.findOne({ email });
    if (!user) return res.status(400).json({ message: "Email không tồn tại" });
    const token = jwt.sign({ id: user._id }, process.env.JWT_SECRET, {
      expiresIn: "15m",
    });

    const resetLink = `http://localhost:3000/reset-password/${token}`;

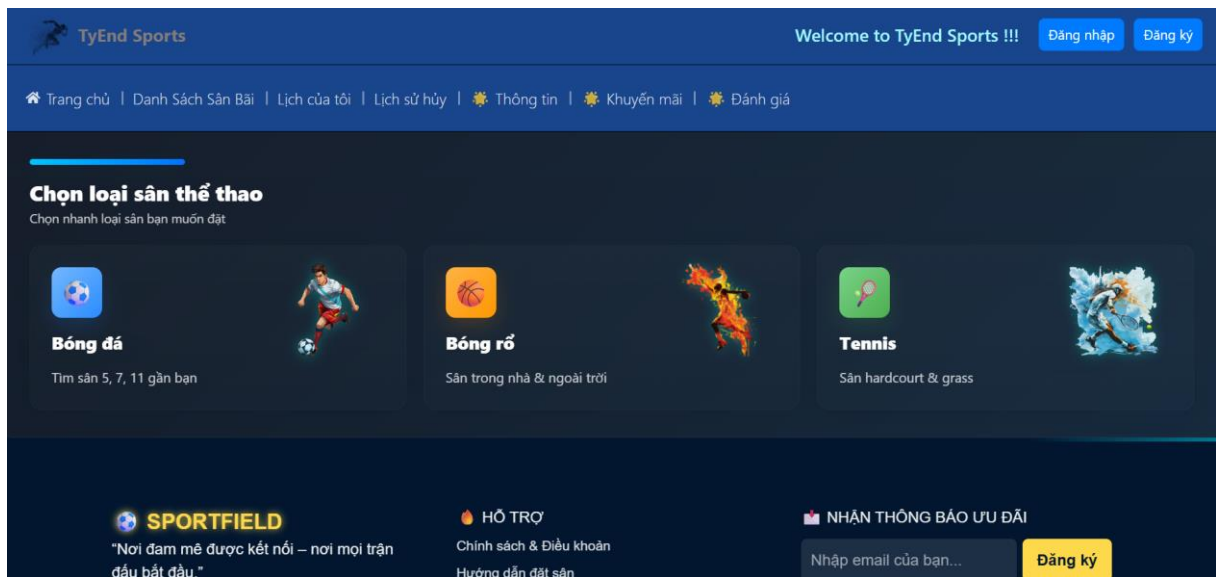
    const transporter = nodemailer.createTransport({
      service: "gmail",
      auth: {
        user: process.env.EMAIL_USER, // Gmail của bạn
        pass: process.env.EMAIL_PASS, // App Password
      },
    });

    await transporter.sendMail({
      from: `Booking App <${process.env.EMAIL_USER}>`,
      to: email,
      subject: "Đặt lại mật khẩu",
      html: `
        <h3>Xin chào ${user.name},</h3>
        <p>Bạn đã yêu cầu đặt lại mật khẩu.</p>
        <p>Nhấn vào link dưới đây để đặt lại (hết hạn sau 15 phút):</p>
        <a href="${resetLink}" target="_blank">${resetLink}</a>
      `,
    });
  }
});
```

Hình 4.7 source code trang quên mật khẩu

Khi quên mật khẩu tài khoản người dùng có thể ẩn quên mật khẩu ở trang đăng nhập hoặc đăng kí , yêu cầu khách hàng nhập gmail để lấy lại mật khẩu lúc này hệ thống sẽ gửi link đặt lại mật khẩu về gmail của khách hàng , và nếu người dùng đến gmail và nhấp vào link thì sẽ về trang giao diện trang quên mật khẩu, ở đây người dùng nhập lại mật khẩu mới và xác nhận để tạo lại mật khẩu của mình để hệ thống xử lí và quay lại trang đăng nhập.

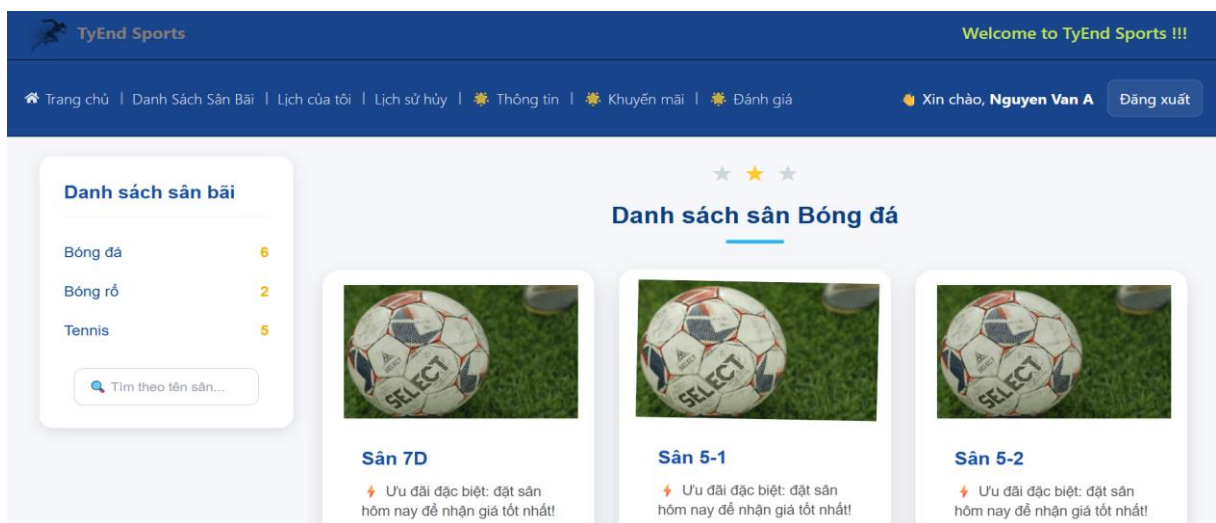
4.1.5 Giao diện trang đặt sân



Hình 4.8 Giao diện trang đặt sân

Tại đây người dùng có thể tìm kiếm sản phẩm thêm bộ môn của mình ở hệ thống như: Bóng Đá ,Bóng Rổ ,Tennit , và sau đó là chọn khung giờ để chơi.

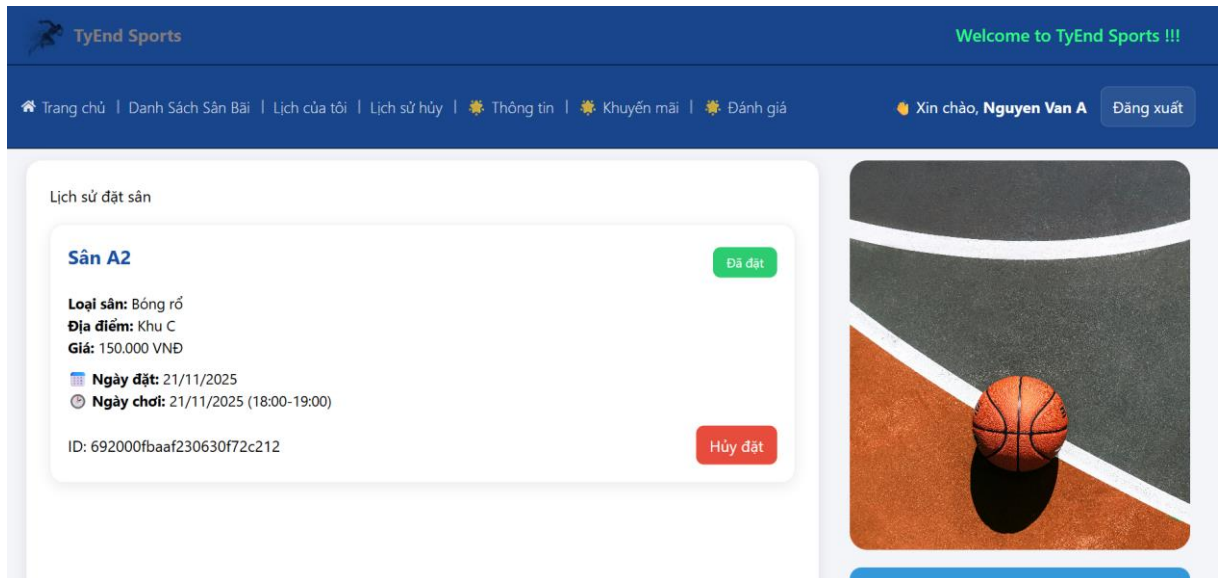
4.1.6 Giao diện trang chi tiết sân



Hình 4.9 Giao diện trang chi tiết sân

Sau khi kích chọn loại sân thì giao diện trang chi tiết sân xuất hiện, tại đây khách hàng có thể xem chi tiết về sân cũng như mô tả sân, tìm kiếm theo tên sân, danh mục sân, hệ thống xử lý thông tin và trả về yêu cầu của khách.

4.1.7 Giao diện trang đặt sân



Hình 4.10 Giao diện trang đặt sân

```
router.post("/", auth, async (req, res) => {
  try {
    const { field_id, booking_date, time_slot, sport_type, usedVoucher, final_price } = req.body;
    let fieldModel, timeSlotModel;
    if (sport_type === "Bóng đá") {
      fieldModel = FootballField;
      timeSlotModel = FootballTimeSlot;
    } else if (sport_type === "Bóng rổ") {
      fieldModel = BasketballField;
      timeSlotModel = BasketballTimeSlot;
    } else if (sport_type === "Tennis") {
      fieldModel = TennisField;
      timeSlotModel = TennisTimeSlot;
    }

    const field = await fieldModel.findById(field_id);
    if (!field) return res.status(404).json({ message: "Không tìm thấy sân" });

    const normalizedDate = new Date(booking_date);
    normalizedDate.setHours(0, 0, 0, 0);

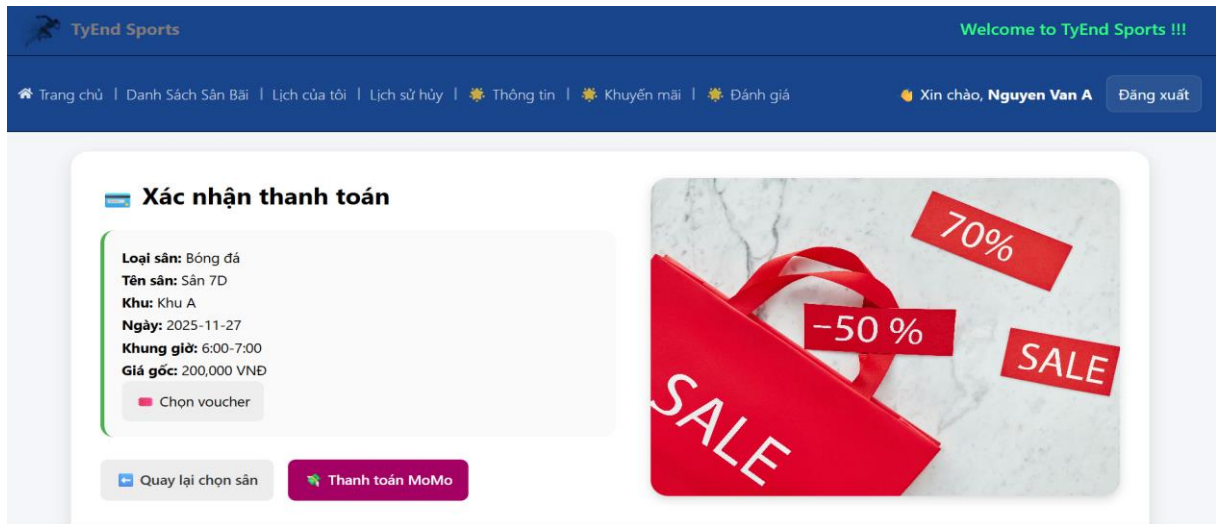
    const exists = await Booking.findOne({
      field_id,
      booking_date: normalizedDate,
      time_slot,
      status: { $ne: "cancelled" },
    });
    if (exists) {
      return res.status(400).json({ message: "⚠️ Khung giờ này đã được đặt." });
    }

    const [start, end] = time_slot.split("-").map((s) => s.trim());
    const slot = await timeSlotModel.findOne({ start, end });
```

Hình 4.11 source code trang đặt sân

Tại đây khách có thể thấy được các sân theo thời gian và sân gần nhất sẽ hiển thị trên đầu, khách hàng có thể hủy sân đã đặt từ trước, và khách hàng có thể đặt sân khách ở nút đặt thêm sân mới.

4.1.8 Giao diện trang xác nhận thanh toán



Hình 4.12 Giao diện trang xác nhận thanh toán

```
router.post("/momo/ipn", async (req, res) => {
  try {
    console.log("📬 Nhận callback từ MoMo:", req.body);

    const {
      orderId,
      requestId,
      amount,
      resultCode,
      message,
      transId,
      extraData,
    } = req.body;

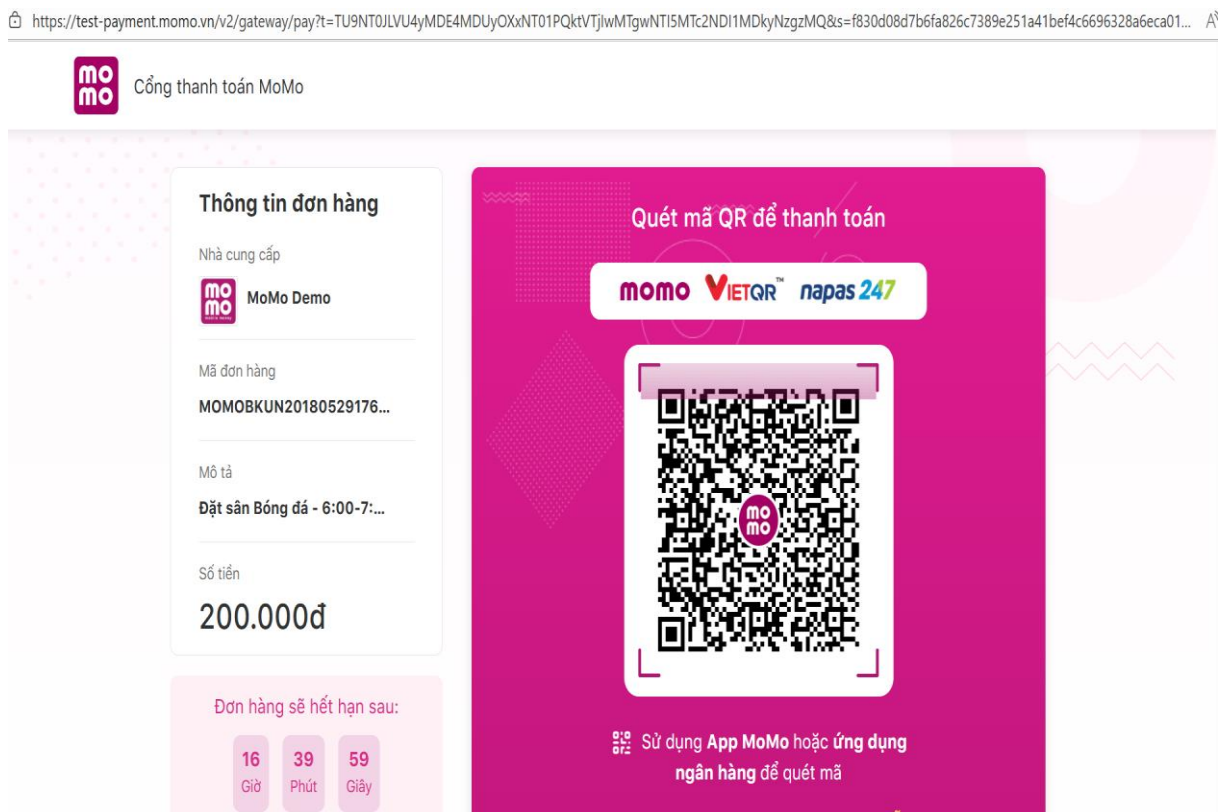
    // Cập nhật thông tin giao dịch trong DB
    await Payment.findOneAndUpdate(
      { orderId },
      {
        transId,
        resultCode,
        message,
        extraData,
      },
      { new: true }
    );

    res.sendStatus(204); // MoMo yêu cầu trả về 204
  } catch (err) {
    console.error("❌ Lỗi xử lý IPN:", err);
    res.status(500).json({ message: "Lỗi xử lý IPN" });
  }
});
```

Hình 4.13 source code trang thanh toán

Ở bước này, hệ thống hiển thị toàn bộ thông tin đặt sân mà khách hàng đã chọn ở các bước trước, bao gồm loại sân, tên sân, khu vực, ngày đặt, khung giờ và giá sân. Khách hàng chỉ cần kiểm tra lại các thông tin này và lựa chọn phương thức thanh toán (MoMo hoặc các hình thức khác tùy hệ thống hỗ trợ). Sau khi nhấn nút “Thanh toán”, hệ thống sẽ tiến hành xử lý giao dịch và thông báo kết quả. Nếu quá trình thanh toán thành công, hệ thống xác nhận đặt sân thành công; trường hợp giao dịch lỗi hoặc không hợp lệ, hệ thống sẽ thông báo thanh toán thất bại để khách hàng thử lại.

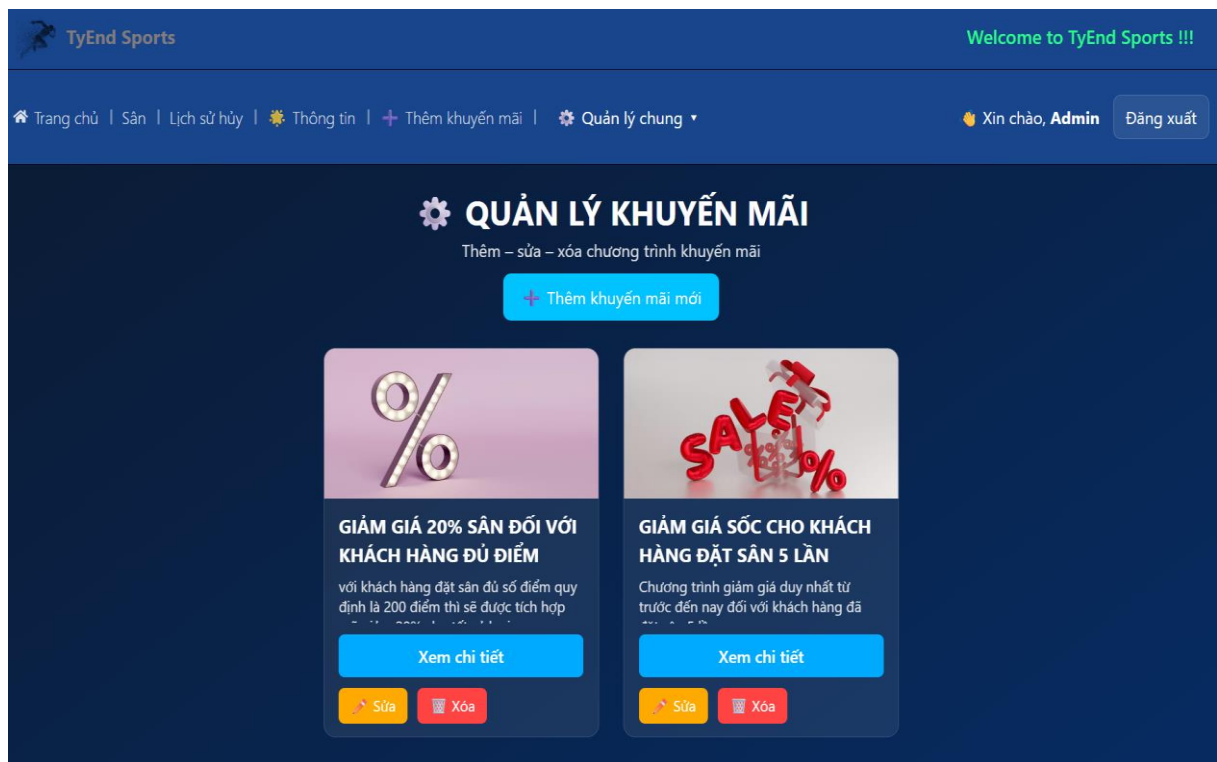
4.1.9 Giao diện thanh toán trực tuyến momo



Hình 4.14 Giao diện trang thanh toán trực tuyến momo

Sau khi khách hàng chọn phương thức thanh toán trực tuyến MoMo, hệ thống sẽ chuyển hướng người dùng sang cổng thanh toán thử nghiệm (MoMo Test Pay). Tại đây, MoMo hiển thị đầy đủ thông tin đơn hàng như mã đơn, mô tả giao dịch và số tiền thanh toán. Người dùng chỉ cần quét mã QR bằng ứng dụng MoMo hoặc ứng dụng ngân hàng hỗ trợ VietQR để tiến hành thanh toán. Nếu giao dịch được xác nhận thành công, MoMo sẽ trả về kết quả thanh toán thành công cho hệ thống và đơn đặt sân được ghi nhận. Ngược lại, nếu giao dịch bị hủy, hết hạn hoặc không hợp lệ, hệ thống sẽ nhận thông báo lỗi và hiển thị thông báo thanh toán thất bại để khách hàng thực hiện lại.

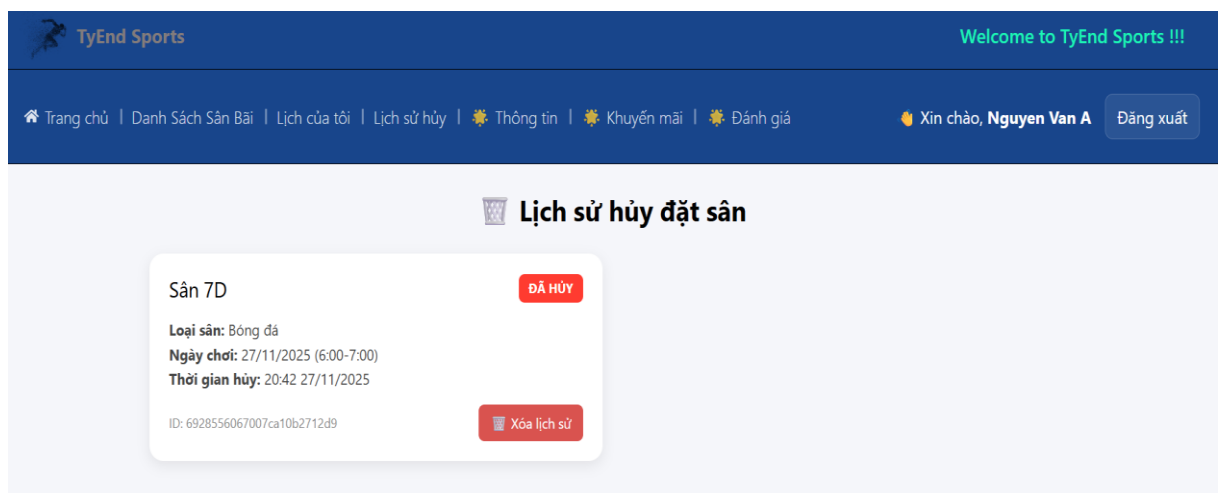
4.1.10 Giao diện trang khuyến mãi



Hình 4.15 Giao diện trang khuyến mãi

Tại trang ưu đãi, hệ thống hiển thị danh sách các khuyến mãi hiện có để người dùng dễ dàng xem và lựa chọn. Khi người dùng nhấn vào nút “Xem chi tiết”, hệ thống sẽ chuyển đến trang hiển thị đầy đủ thông tin của chương trình khuyến mãi, bao gồm mô tả, điều kiện áp dụng, thời gian hiệu lực và mức giảm giá. Ở cuối trang, hệ thống cung cấp nút “Đặt sân ngay” để người dùng có thể nhanh chóng chuyển sang bước chọn sân và áp dụng ưu đãi nếu phù hợp.

4.1.11 Giao diện trang lịch sử hủy sân



Hình 4.16 Giao diện trang lịch sử hủy sân

```
router.get("/cancelled/history", auth, async (req, res) => {
  try {
    let history;
    if (req.user.role === "admin") {
      history = await CancelledBooking.find()
        .populate("user_id", "name email")
        .sort({ cancelled_at: -1 });
    } else {
      history = await CancelledBooking.find({ user_id: req.user.id }).sort({ cancelled_at: -1 });
    }
    res.json(history);
  } catch (err) {
    console.error("❌ Lỗi khi tải lịch sử hủy sân:", err);
    res.status(500).json({ message: "Lỗi server khi tải lịch sử hủy sân", error: err.message });
  }
});
```

Hình 4.17 source code trang hủy sân

Tại mục “Lịch sử hủy đặt sân”, hệ thống hiển thị danh sách các sân mà người dùng đã thực hiện thao tác hủy. Mỗi mục bao gồm các thông tin: tên sân, loại sân, ngày chơi, khung giờ và thời gian người dùng đã hủy đặt sân. Trạng thái của đơn hủy được hiển thị rõ ràng để người dùng dễ dàng theo dõi. Ngoài ra, hệ thống cung cấp nút “Xóa lịch sử” để người dùng chủ động xóa bỏ bản ghi hủy đặt sân khi không còn nhu cầu lưu trữ thông tin này.

4.1.12 Giao diện trang cập nhập lại mật khẩu

Hình 4.18 Giao diện trang cập nhập mật khẩu

```

/* ----- RESET MẬT KHẨU ----- */
router.post("/reset-password/:token", async (req, res) => {
  try {
    const { token } = req.params;
    const { password } = req.body;
    const decoded = jwt.verify(token, process.env.JWT_SECRET);
    const hashed = await bcrypt.hash(password, 10);

    await User.findByIdAndUpdate(decoded.id, { password: hashed });

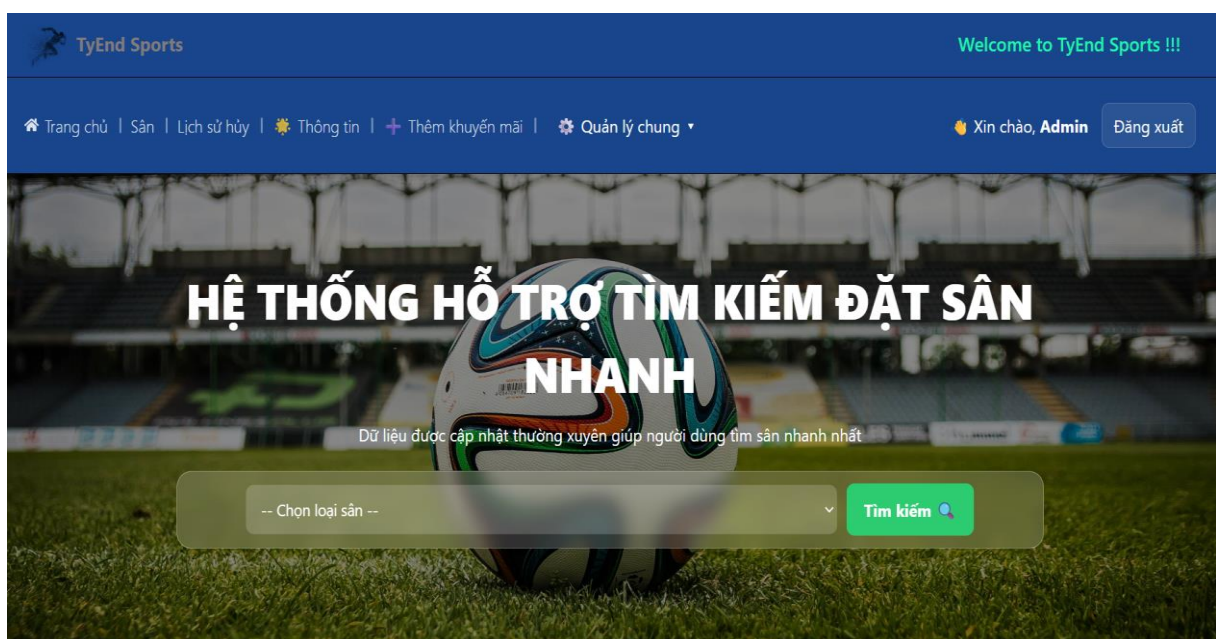
    res.json({ message: "Đặt lại mật khẩu thành công" });
  } catch (err) {
    console.error("Lỗi reset mật khẩu:", err);
    res.status(400).json({ message: "Token không hợp lệ hoặc đã hết hạn" });
  }
});

```

Hình 4.19 source code trang lấy lại mật khẩu

Khi người dùng quên mật khẩu và yêu cầu đặt lại qua email, hệ thống sẽ gửi một liên kết đặt lại mật khẩu đến địa chỉ email đã đăng ký. Sau khi người dùng nhấn vào liên kết trong email, họ được chuyển đến trang “Đặt lại mật khẩu”. Tại đây, hệ thống yêu cầu người dùng nhập mật khẩu mới và xác nhận thay đổi. Nếu mật khẩu hợp lệ và được nhập đúng định dạng, hệ thống sẽ cập nhật mật khẩu mới và thông báo đặt lại thành công. Ngược lại, nếu mật khẩu không hợp lệ hoặc để trống, hệ thống sẽ hiển thị thông báo lỗi để người dùng thực hiện lại.

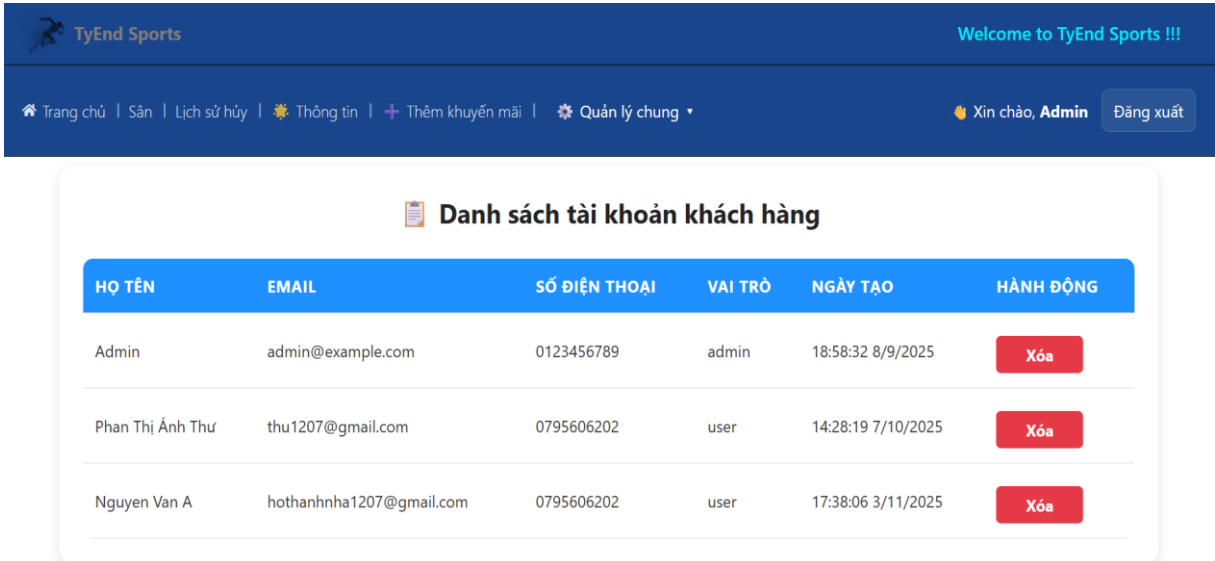
4.1.13 Giao diện trang chủ admin



Hình 4.20 Giao diện trang chủ admin

Đây là giao diện trang chủ dành cho quản trị viên (Admin). Tại trang này, hệ thống hiển thị thanh điều hướng với đầy đủ các chức năng quản lý như quản lý sân, lịch sử hủy, thông tin hệ thống, quản lý khuyến mãi và các mục thiết lập chung. Ngoài ra, giao diện trang chủ cũng cung cấp một công cụ tìm kiếm nhanh, cho phép quản trị viên lựa chọn loại sân và thực hiện tìm kiếm dữ liệu một cách thuận tiện. Thông tin trên trang được cập nhật liên tục nhằm hỗ trợ admin theo dõi và quản lý hệ thống đặt sân thể thao một cách hiệu quả.

4.1.14 Giao diện trang quản lý tài khoản



The screenshot shows the admin dashboard for TyEnd Sports. The header includes the logo, navigation links (Trang chủ, Sân, Lịch sử hủy, Thông tin, Thêm khuyến mãi, Quản lý chung), and a welcome message for Admin. The main content area displays a table titled "Danh sách tài khoản khách hàng" (Customer Account List) with columns for Name, Email, Phone Number, Role, Creation Date, and Action. The table lists three users: Admin, Phan Thị Ánh Thư, and Nguyen Van A, each with a delete button.

HỌ TÊN	EMAIL	SỐ ĐIỆN THOẠI	VAI TRÒ	NGÀY TẠO	HÀNH ĐỘNG
Admin	admin@example.com	0123456789	admin	18:58:32 8/9/2025	Xóa
Phan Thị Ánh Thư	thu1207@gmail.com	0795606202	user	14:28:19 7/10/2025	Xóa
Nguyen Van A	hothanhnha1207@gmail.com	0795606202	user	17:38:06 3/11/2025	Xóa

Hình 4.21 Giao diện trang quản lý tài khoản

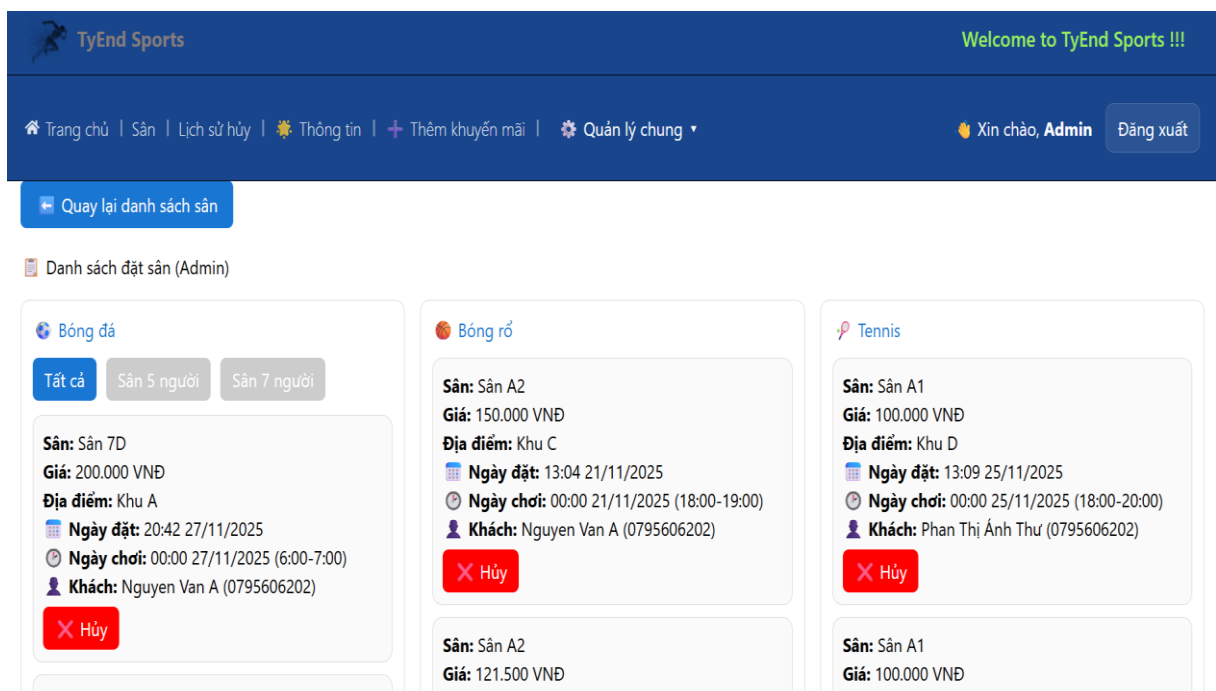
```
function isAdmin(req, res, next) {
  if (!req.user || req.user.role !== "admin") {
    return res.status(403).json({ message: "❌ Bạn không có quyền admin" });
  }
  next();
}

router.get("/users", auth, isAdmin, async (req, res) => {
  try {
    const users = await User.find().select("-password");
    res.json(users);
  } catch (err) {
    console.error("❌ Lỗi lấy danh sách user:", err);
    res.status(500).json({ message: "Lỗi server", error: err.message });
  }
});
```

Hình 4.22 source code trang quản lý tài khoản

Tại mục “Quản lý tài khoản khách hàng”, hệ thống hiển thị danh sách toàn bộ tài khoản đã đăng ký trên hệ thống bao gồm họ tên, email, số điện thoại, vai trò và thời gian tạo tài khoản. Giao diện cho phép quản trị viên dễ dàng theo dõi thông tin người dùng và thực hiện các thao tác quản lý cần thiết. Đối với mỗi tài khoản, hệ thống cung cấp nút “Xóa” để admin có thể xóa tài khoản không còn sử dụng hoặc vi phạm quy định. Khi thao tác xóa được thực hiện, hệ thống sẽ xử lý và cập nhật lại danh sách tài khoản tương ứng.

4.1.15 Giao diện trang quản lý danh sách sân



Hình 4.23 Giao diện trang quản lý danh sách sân

```
router.get("/my", auth, async (req, res) => {
  try {
    const bookings = await Booking.find({ user: req.user.id }).sort({ createdAt: -1 });

    const result = bookings.map((b) => ({
      _id: b._id,
      field_name: b.field_name,
      field_price: b.field_price,
      field_location: b.field_location,
      sport_type: b.sport_type,
      booking_date: b.booking_date,
      time_slot: b.time_slot,
      status: b.status || "booked",
      createdAt: b.createdAt,
    }));

    res.json(result);
  } catch (err) {
    console.error("Lỗi khi lấy danh sách booking:", err);
    res.status(500).json({ message: "✖ Lỗi khi lấy danh sách booking", error: err.message });
  }
});
```

Hình 4.24 source code trang danh sách sân

Tại mục “Danh sách đặt sân (Admin)”, hệ thống hiển thị toàn bộ các lượt đặt sân của khách hàng theo từng môn thể thao như bóng đá, bóng rổ và tennis. Mỗi mục hiển thị đầy đủ thông tin bao gồm tên sân, giá thuê, địa điểm, thời gian đặt, ngày giờ chơi và thông tin khách hàng đã đặt sân. Nhờ đó, quản trị viên có thể theo dõi, kiểm tra và quản lý tình trạng đặt sân một cách thuận tiện. Bên cạnh mỗi lượt đặt sân, hệ thống cung cấp nút “Hủy” cho phép admin thực hiện thao tác hủy đặt sân trong các trường hợp cần thiết. Khi thao tác được thực hiện, hệ thống sẽ xử lý cập nhật lại trạng thái và ghi vào lịch sử hủy nếu áp dụng.

4.1.16 Giao diện trang thêm sân

Hình 4.25 Giao diện trang thêm sân

```
const router = express.Router();

router.post("/", async (req, res) => {
  try {
    const { type, name, size, location } = req.body;

    let newField;

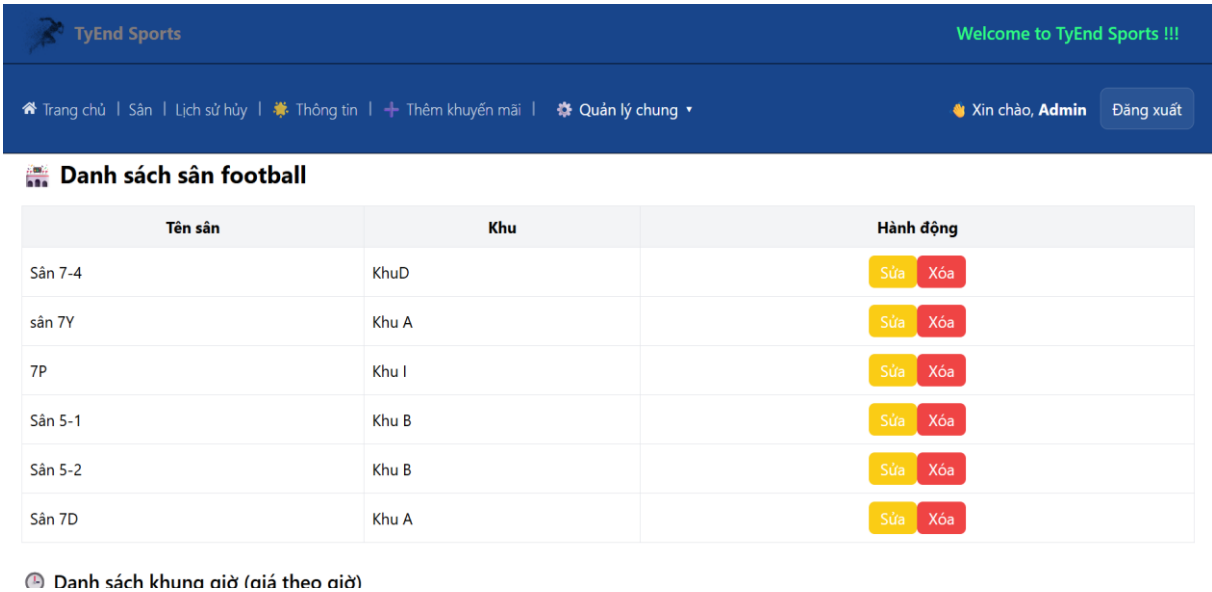
    if (type === "football") {
      if (!size) {
        return res.status(400).json({ message: "Thiếu size cho sân bóng đá" });
      }
      newField = new FootballField({ name, size, location });
    } else if (type === "basketball") {
      newField = new BasketballField({ name, location });
    } else if (type === "tennis") {
      newField = new TennisField({ name, location });
    } else {
      return res.status(400).json({ message: "Loại sân không hợp lệ" });
    }

    await newField.save();
    res.status(201).json({ message: "Thêm sân thành công", field: newField });
  } catch (err) {
    console.error("❌ Lỗi khi thêm sân:", err);
    res.status(500).json({ message: "Lỗi server", error: err.message });
  }
});
```

Hình 4.26 source code trang thêm sân

Tại mục “Thêm sân mới”, hệ thống cho phép quản trị viên nhập các thông tin cơ bản để tạo mới một sân thi đấu, bao gồm tên sân, địa điểm và loại sân. Giao diện được thiết kế trực quan với các trường nhập liệu hướng dẫn rõ ràng giúp admin dễ dàng thao tác. Sau khi nhập đầy đủ thông tin, quản trị viên có thể nhấn nút “Thêm sân” để hệ thống lưu dữ liệu và cập nhật sân mới vào danh sách quản lý. Ngoài ra, nút “Đặt lại” được cung cấp để admin có thể nhanh chóng xóa các dữ liệu đã nhập và thao tác lại khi cần thiết.

4.1.17 Giao diện trang chỉnh sửa sân



Tên sân	Khu	Hành động
Sân 7-4	KhuD	Sửa Xóa
sân 7Y	Khu A	Sửa Xóa
7P	Khu I	Sửa Xóa
Sân 5-1	Khu B	Sửa Xóa
Sân 5-2	Khu B	Sửa Xóa
Sân 7D	Khu A	Sửa Xóa

[Danh sách khu vực chơi \(aiá theo aiờ\)](#)

Hình 4.27 Giao diện trang chỉnh sửa sân

```
// Sửa sân
router.put("/fields/:type/:id", auth, isAdmin, async (req, res) => {
  try {
    const { type, id } = req.params;
    let Model;

    switch (type) {
      case "football":
        Model = FootballField;
        break;
      case "basketball":
        Model = BasketballField;
        break;
      case "tennis":
        Model = TennisField;
        break;
      default:
        return res.status(400).json({ message: "✗ Loại sân không hợp lệ" });
    }

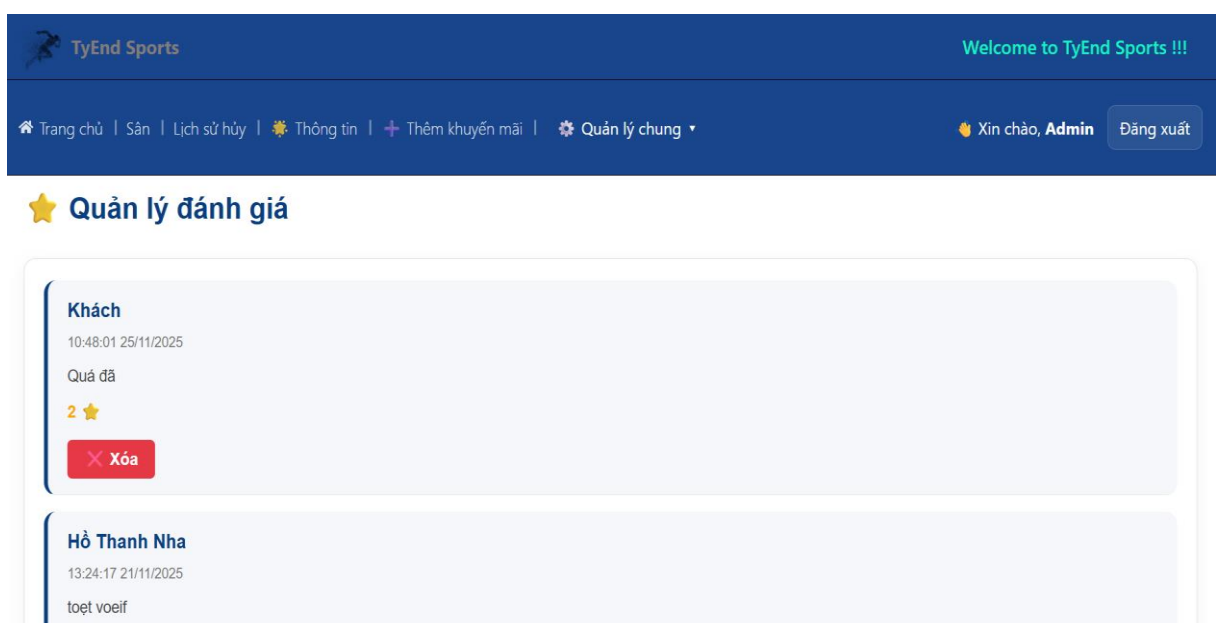
    const updatedField = await Model.findByIdAndUpdate(id, req.body, { new: true });
    if (!updatedField) {
      return res.status(404).json({ message: "Không tìm thấy sân để cập nhật" });
    }

    res.json({ message: "✅ Đã cập nhật sân", field: updatedField });
  } catch (err) {
    res.status(500).json({ message: "Lỗi khi cập nhật sân", error: err.message });
  }
});
```

Hình 4.28 source code trang chỉnh sửa sân

Tại mục “Danh sách sân football”, hệ thống hiển thị toàn bộ các sân bóng thuộc nhóm môn thể thao bóng đá. Mỗi sân được trình bày với các thông tin cơ bản như tên sân và khu vực để quản trị viên dễ dàng theo dõi. Bên cạnh mỗi sân, hệ thống cung cấp hai chức năng quản lý gồm “Sửa” và “Xóa”. Chức năng “Sửa” cho phép admin điều chỉnh thông tin sân khi có sự thay đổi về tên sân, khu vực hoặc các thuộc tính liên quan. Chức năng “Xóa” cho phép loại bỏ những sân không còn hoạt động hoặc cần được thay thế. Sau khi thao tác chỉnh sửa hoặc xóa hoàn tất, hệ thống sẽ cập nhật lại danh sách một cách tự động, đảm bảo dữ liệu luôn chính xác và đồng bộ.

4.1.18 Giao diện trang quản lý đánh giá



Hình 4.29 Giao diện trang quản lý đánh giá

```
router.get("/", async (req, res) => {
  try {
    const reviews = await Review.find().sort({ createdAt: -1 });
    res.json(reviews);
  } catch (err) {
    res.status(500).json({ message: "Lỗi server", error: err.message });
  }
});

// ★ Tính trung bình rating
```

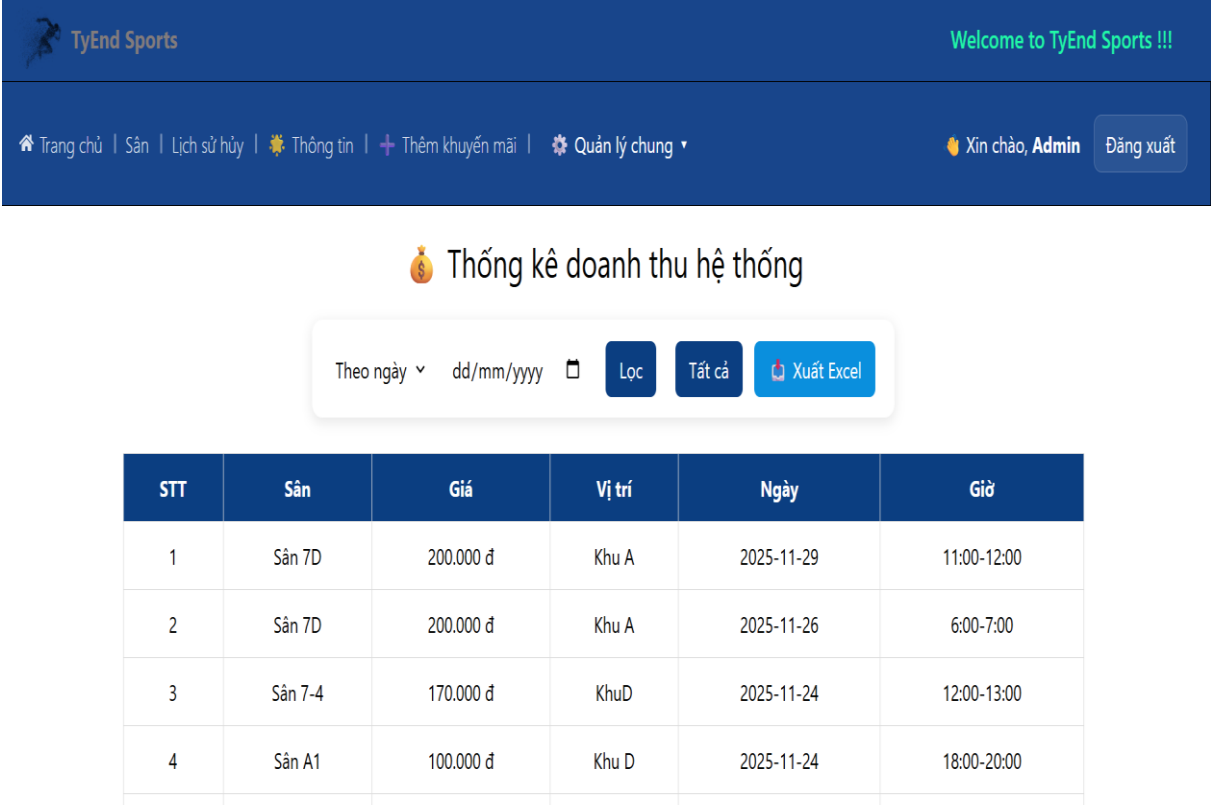
Hình 4.30 source code trang đánh giá

Tại mục “Quản lý đánh giá”, hệ thống hiển thị danh sách toàn bộ các đánh giá mà khách hàng đã gửi về sân. Mỗi đánh giá được trình bày theo từng khối riêng biệt với các

Sinh viên thực hiện: Hồ Thanh Nha Giảng Viên Hướng Dẫn: TS. Lâm Tùng Giang

thông tin gồm tên người đánh giá, thời gian gửi đánh giá, nội dung phản hồi và số sao mà người dùng đã chấm. Điều này giúp quản trị viên dễ dàng theo dõi mức độ hài lòng của khách hàng theo từng thời điểm khác nhau. Trong mỗi khối đánh giá, hệ thống cung cấp nút “Xóa” để admin có thể loại bỏ các đánh giá không phù hợp, spam hoặc chứa nội dung vi phạm. Sau khi xóa, danh sách được cập nhật lại ngay lập tức nhằm đảm bảo dữ liệu phản ánh đúng và sạch sẽ trên hệ thống.

4.1.19 Giao diện trang thống kê doanh thu

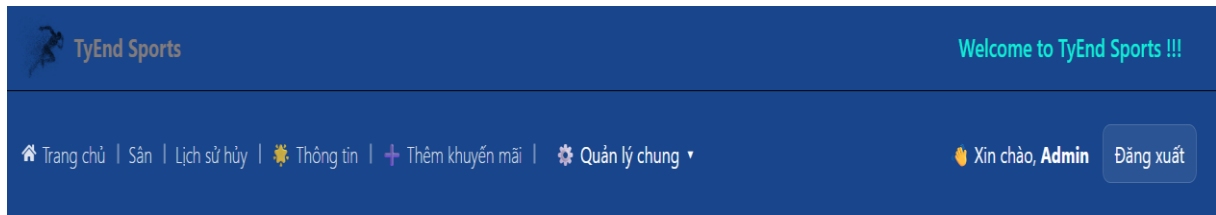


STT	Sân	Giá	Vị trí	Ngày	Giờ
1	Sân 7D	200.000 đ	Khu A	2025-11-29	11:00-12:00
2	Sân 7D	200.000 đ	Khu A	2025-11-26	6:00-7:00
3	Sân 7-4	170.000 đ	Khu D	2025-11-24	12:00-13:00
4	Sân A1	100.000 đ	Khu D	2025-11-24	18:00-20:00

Hình 4.31 Giao diện trang thống kê doanh thu

Chức năng “Thống kê doanh thu hệ thống” cho phép quản trị viên theo dõi toàn bộ doanh thu phát sinh từ hoạt động đặt sân trên hệ thống. Giao diện được thiết kế trực quan, tập trung vào tính tiện dụng và khả năng lọc dữ liệu theo thời gian. Tại phần đầu trang, admin có thể lựa chọn hình thức thống kê theo ngày bằng cách chọn mốc thời gian cụ thể thông qua bộ lọc lịch. Nút “Lọc” giúp hiển thị kết quả theo đúng điều kiện thời gian mà quản trị viên yêu cầu, trong khi nút “Tất cả” dùng để hiển thị toàn bộ dữ liệu doanh thu không giới hạn thời điểm. Ngoài ra, hệ thống tích hợp chức năng “Xuất Excel”, hỗ trợ admin tải toàn bộ dữ liệu thống kê dưới dạng file Excel để phục vụ cho việc đối soát, lưu trữ hoặc báo cáo tài chính.

4.1.20 Giao diện trang thêm mới khuyến mãi



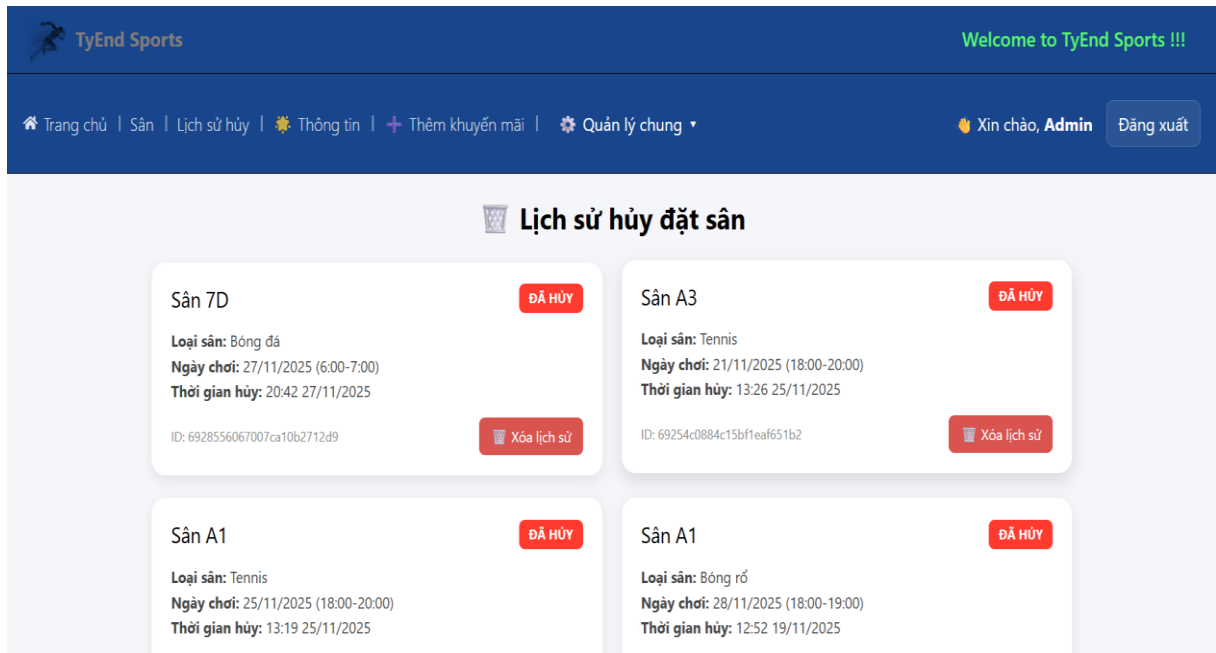
Hình 4.32 Giao diện trang thêm mới khuyến mãi

```
router.post("/", async (req, res) => {
  try {
    const newPromo = await Promotion.create(req.body);
    res.json({ message: "Tạo khuyến mãi thành công", data: newPromo });
  } catch (err) {
    res.status(500).json({ message: "Lỗi khi tạo", error: err.message });
  }
});
```

Hình 4.33 source code trang tạo khuyến mãi

Chức năng “Thêm khuyến mãi mới” cho phép quản trị viên tạo mới các chương trình ưu đãi dành cho người dùng trên hệ thống. Trang được thiết kế với giao diện đơn giản, rõ ràng và dễ thao tác để đảm bảo admin có thể nhanh chóng tạo khuyến mãi khi cần.

4.1.21 Giao diện trang xem lịch sử hủy



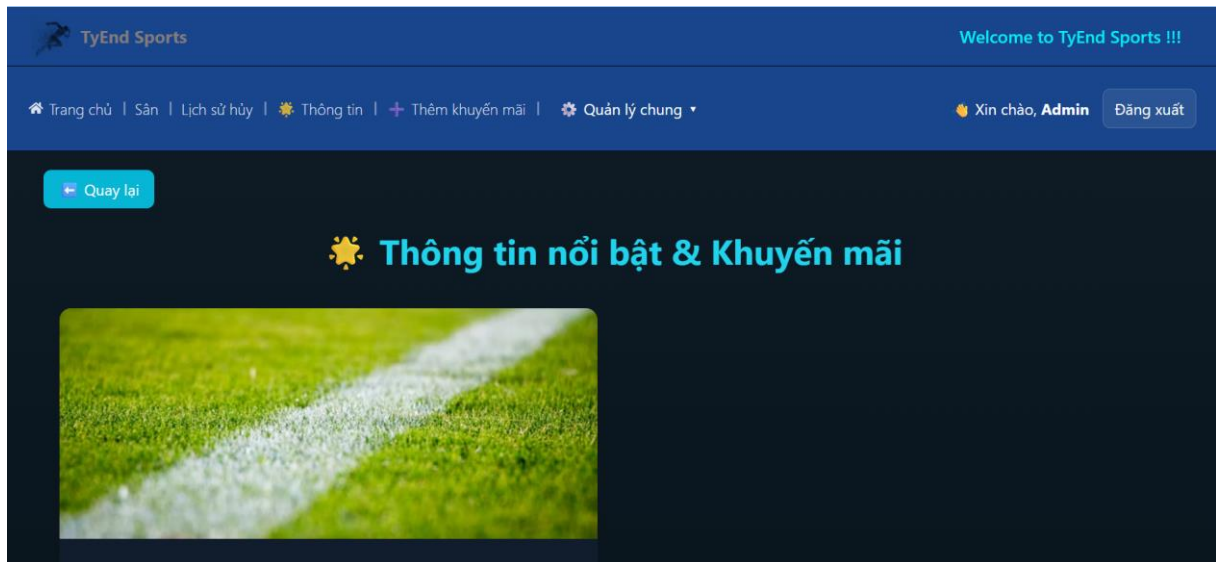
Hình 4.34 Giao diện trang xem lịch sử hủy

```
router.get('/cancelled/history', auth, async (req, res) => {
  try {
    let history;
    if (req.user.role === "admin") {
      history = await CancelledBooking.find()
        .populate("user_id", "name email")
        .sort({ cancelled_at: -1 });
    } else {
      history = await CancelledBooking.find({ user_id: req.user.id }).sort({ cancelled_at: -1 });
    }
    res.json(history);
  } catch (err) {
    console.error("✗ Lỗi khi tải lịch sử hủy sân:", err);
    res.status(500).json({ message: "Lỗi server khi tải lịch sử hủy sân", error: err.message });
  }
});
```

Hình 4.35 source code trang lịch sử hủy

Trang Lịch Sử Hủy Đặt Sân giúp Admin theo dõi tất cả các lượt đặt sân đã bị hủy, bao gồm tên sân, loại sân, thời gian chơi, thời điểm hủy và mã lịch sử. Mỗi lượt hủy được hiển thị dưới dạng thẻ rõ ràng với trạng thái “ĐÃ HỦY” nổi bật, giúp Admin dễ dàng quản lý và kiểm soát thông tin.

4.1.22 Giao diện trang quản lý thông tin nổi bật và khuyến mãi



Hình 4.36 Giao diện trang quản lý thông tin nổi bật và khuyến mãi

Trang Thông Tin Nổi Bật & Khuyến Mãi hiển thị các chương trình ưu đãi và cập nhật quan trọng dành cho người dùng. Admin có thể xem chi tiết từng khuyến mãi kèm hình ảnh minh họa, giúp quản lý nội dung dễ dàng và trực quan.

Kết luận và đề xuất

1 Kết luận

Sau quá trình tìm hiểu và phân tích, em đã xây dựng một Hệ thống đặt sân thể thao online cho đồ án của mình. Sử dụng những kỹ năng lập trình và kiến thức đã học được, như ngôn ngữ JavaScript, ReactJS, NodeJS (Express) và cơ sở dữ liệu MongoDB, em đã tạo ra một hệ thống hoàn chỉnh.

Hệ thống này có thể mang lại nhiều lợi ích cho người dùng, cho phép mọi người dễ dàng tìm kiếm thông tin về các loại sân thể thao, xem giá thuê, thời gian trống, cũng như xem các bài viết liên quan đến thể thao. Đặc biệt, người dùng có thể đặt sân trực tuyến nhanh chóng, giúp tiết kiệm thời gian và công sức so với cách đặt sân truyền thống. Bên cạnh đó, hệ thống cho phép người dùng quản lý các lượt đặt sân của bản thân và hủy đặt sân khi không còn nhu cầu.

Việc áp dụng các kỹ năng lập trình và công nghệ hiện đại như NodeJS và MongoDB, cùng với thư viện ReactJS, đã giúp em tạo ra một hệ thống đáng tin cậy và hiệu quả. Em đã sử dụng khả năng xử lý dữ liệu và logic của ExpressJS để xây dựng các API hỗ trợ việc tìm kiếm sân, quản lý thời gian trống, đặt sân và xử lý hoạt động của người dùng. Cùng với đó, MongoDB đã giúp em lưu trữ và truy vấn dữ liệu linh hoạt, đảm bảo hệ thống hoạt động mượt mà và ổn định.

Ngoài ra, việc thực hiện đồ án đã giúp em áp dụng và rèn luyện các kỹ năng như lập trình, quản lý dự án và kiểm thử. Em đã học cách xây dựng một dự án phức tạp từ đầu đến cuối, từ việc phân tích yêu cầu, thiết kế hệ thống, triển khai backend – frontend và kiểm thử toàn bộ chức năng. Điều này giúp em phát triển khả năng làm việc độc lập, quản lý thời gian và giải quyết các vấn đề phát sinh trong quá trình phát triển.

Tuy nhiên, như mọi dự án phát triển, hệ thống của em vẫn còn một số hạn chế và nhiều tiềm năng cải thiện. Trong tương lai, em sẽ tiếp tục mở rộng và nâng cấp hệ thống để đáp ứng tốt hơn nhu cầu của người dùng, như cải thiện giao diện, tối ưu hiệu suất API và bổ sung các tính năng như tích hợp thanh toán online, gửi thông báo tự động hoặc gợi ý sân phù hợp.

Em tin rằng những kiến thức và kỹ năng mà em tích lũy được thông qua dự án này sẽ tiếp tục hỗ trợ em trong quá trình phát triển bản thân và sự nghiệp trong lĩnh vực phát triển phần mềm.

2 Đề xuất

Do năng lực của em còn hạn chế và thời gian thực hiện đồ án cũng tương đối ngắn, hệ thống đặt sân thể thao online hiện tại mới chỉ đáp ứng được các chức năng cơ bản nhất phục vụ quá trình tìm kiếm sân, xem thông tin sân và đặt sân. So với một hệ thống đặt sân chuyên nghiệp đang được sử dụng rộng rãi trên thị trường, đề tài của em vẫn còn thiếu một số tính năng mở rộng quan trọng nhằm nâng cao trải nghiệm người dùng, cải thiện hiệu quả quản lý và tăng tính ứng dụng thực tế.

Em xin đề xuất phát triển thêm một số tính năng khác:

- Liên kết tài khoản facebook, google, ..
- Thêm chức năng chat bot để hỗ trợ người dùng hoặc chat qua zalo.
- Gửi thông báo qua SMS hoặc Zalo khi admin cập nhật trạng thái đặt sân.

Tài Liệu Tham Khảo

1. Programming with Mosh – “Node.js Tutorial for Beginners” *Nguồn: YouTube*
- 2 Traversy Media – “MERN Stack Tutorial” *Nguồn: YouTube*
3. Net Ninja – “React Tutorial for Beginners” Series *Nguồn: YouTube*
4. MongoDB Documentation – “MongoDB CRUD Operations” *Nguồn: Google*
5. ExpressJS Official Guide – “Express.js Documentation *Nguồn: Google*
6. W3Schools – “ReactJS Tutorial *Nguồn: Google*

Các trang web tham khảo

<https://react.dev/learn>

<https://reactrouter.com/en/main/start/tutorial>

<https://expressjs.com/en/guide/routing.html>

<https://expressjs.com/en/guide/using-middleware.html>

<https://www.mongodb.com/docs/manual/crud/>

<https://www.mongodb.com/docs/manual/schema/>

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Promises>

<https://mongoosejs.com/docs/guide.html>

<https://mongoosejs.com/docs/models.html>