

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



ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG SÀN THƯƠNG MẠI GAME BẰNG NUXTJS VÀ LARAVEL

SINH VIÊN THỰC HIỆN : TÒNG TRỌNG HIỆP

MÃ SINH VIÊN : 1451020091

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

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



TÒNG TRỌNG HIỆP

**XÂY DỰNG SẢN THƯƠNG MẠI GAME
BẰNG NUXTJS VÀ LARAVEL**

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN: TS. BÙI HẢI PHONG

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em xin cam đoan rằng đồ án này là kết quả nghiên cứu và làm việc của cá nhân em, được thực hiện dưới sự hướng dẫn của thầy Bùi Hải Phong và tập thể giảng viên Khoa Công nghệ thông tin trường Đại học Đại Nam.

Toàn bộ nội dung trong đồ án, bao gồm ý tưởng, thiết kế, mã nguồn và các phần liên quan khác đều là sản phẩm của sự nỗ lực và sáng tạo cá nhân. Mọi thông tin tham khảo từ các nguồn tài liệu, trang web hay các nghiên cứu khác đã được trích dẫn đầy đủ và rõ ràng theo quy định.

Em xin chịu hoàn toàn trách nhiệm về tính trung thực và chính xác của nội dung đồ án này. Trong trường hợp có bất kỳ khiếu nại hay tranh chấp nào về quyền sở hữu trí tuệ, em xin chịu hoàn toàn trách nhiệm trước nhà trường và pháp luật.

Hà Nội, ngày 31 tháng 5 năm 2024
Trân trọng

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy TS. Bùi Hải Phong, người đã không quản ngại thời gian, công sức và tâm huyết để hướng dẫn, động viên và giúp đỡ em hoàn thành đồ án này. Những lời chỉ bảo và sự hỗ trợ của thầy đã giúp em vượt qua nhiều khó khăn và thử thách, giúp em tiến bộ và hoàn thiện từng bước trong quá trình nghiên cứu và thực hiện đồ án.

Em xin bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy cô trong Khoa Công nghệ Thông tin, trường Đại học Đại Nam. Những kiến thức quý báu và sự hướng dẫn tận tình của các thầy cô trong suốt quá trình học tập tại trường đã trang bị cho em nền tảng vững chắc để phát triển bản thân và hoàn thành tốt đồ án này. Những bài giảng, kinh nghiệm và sự nhiệt tình của các thầy cô không chỉ giúp em trong việc học tập mà còn tạo nên động lực để em phấn đấu và nỗ lực hơn trong tương lai.

Em cũng xin chân thành cảm ơn sự ủng hộ và động viên từ gia đình, bạn bè và các đồng nghiệp. Gia đình luôn là nguồn động lực lớn nhất, là chỗ dựa vững chắc, tạo điều kiện tốt nhất để em có thể tập trung học tập và nghiên cứu. Bạn bè và thầy cô đã luôn ở bên, cùng chia sẻ, góp ý và hỗ trợ em trong suốt quá trình thực hiện đồ án, giúp em có thêm nhiều ý tưởng và kinh nghiệm quý báu.

Em đặc biệt cảm ơn những người bạn đã giúp đỡ em trong quá trình nghiên cứu và hoàn thành đồ án này. Sự động viên, khích lệ từ các bạn đã giúp em có thêm niềm tin và sức mạnh để vượt qua mọi trở ngại.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đã và đang đóng một vai trò rất quan trọng trong cuộc sống, đạt được rất nhiều thành tựu rực rỡ với những bước tiến nhảy vọt. Việc áp dụng những ứng dụng của công nghệ thông tin vào các lĩnh vực đời sống của con người ngày càng tăng và nó đã không ngừng can thiệp vào hầu hết các công việc cũng như mọi lĩnh vực của cuộc sống. Đi đôi với sự phát triển công nghệ chế tạo các thiết bị phần cứng của máy tính thì việc các sản phẩm phần mềm ứng dụng lần lượt ra đời áp dụng mọi yêu cầu công việc của cuộc sống và nó có tính chất quyết định đến sự thành công đối với từng công việc đã được áp dụng. Công nghệ thông tin nay đã trở thành ngành công nghiệp mũi nhọn. Nó là một ngành khoa học kỹ thuật không thể thiếu trong cuộc sống cũng như các hoạt động xã hội như: Quản lý kinh tế, thông tin. Ở nước ta hiện nay, việc áp dụng tin học hóa trong việc quản lý tại các cơ quan, xí nghiệp đang rất phổ biến và trở nên cấp thiết.. Do nhu cầu cấp thiết ở trên em quyết định lựa chọn đề tài “Xây dựng sàn thương mại game bằng NuxtJS và Laravel (PHP)” cho đồ án tốt nghiệp của mình.

Vì đồ án được thực hiện trong phạm vi thời gian hạn hẹp và hạn chế về mặt kiến thức chuyên môn, do đó bài báo cáo của em không thể tránh khỏi những sai sót nhất định. Đồng thời bản thân báo cáo là kết quả của một quá trình tổng kết, thu thập kết quả từ việc khảo sát thực tế, những bài học đúc rút từ trong quá trình học tập và làm đồ án của em. Em rất mong có được những ý kiến đóng góp của thầy, cô để bài báo cáo và bản thân em hoàn thiện hơn.

[illegible]

MỤC LỤC

CHƯƠNG 1. TỔNG QUÁT VỀ ĐỀ TÀI	1
1.1. Tổng quan đề tài	1
1.2. Lý do chọn đề tài	2
1.3. Phương pháp nghiên cứu	3
1.4. Mục tiêu của đề tài.....	4
1.5. Nội dung thực hiện	6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	7
2.1. Tổng quan về NuxtJS	7
2.1.1. Giới Thiệu về NuxtJS	7
2.1.2. Đặc Điểm Chính của NuxtJS.....	8
2.1.3. Ứng dụng của NuxtJS.....	8
2.1.4. Ưu điểm của NuxtJS.....	9
2.1.5. Nhược điểm của NuxtJS.....	9
2.2. API (Application Programming Interface)	10
2.2.1. API là gì.....	10
2.2.2. Ứng dụng của API	10
2.2.3. Web API là gì	11
2.2.4. Cách thức hoạt động của Web API	11
2.2.5. Ưu và nhược điểm của Web API.....	12
2.3. Tổng quan về TailwindCSS.....	13
2.3.1. Các đặc điểm chính.....	13
2.3.2. Ưu điểm của TailwindCSS	13
2.3.3. Nhược điểm của TailwindCSS	14
2.3.4. Ứng dụng của TailwindCSS	14
2.4. Tổng quan về Laravel PHP.....	15
2.4.1. Các đặc điểm chính của Laravel.....	15
2.4.2. Ưu điểm của Laravel	16
2.4.3. Nhược điểm của Laravel.....	17

2.5. Tổng quan về MySQL	17
2.5.1. Ứng dụng của Laravel	17
2.5.2. Các đặc điểm chính của MySQL	18
2.5.3. Ưu điểm	18
2.5.4. Nhược điểm	19
2.5.5. Ứng dụng thực tiễn	19
Tiểu kết	20
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	21
3.1. Tổng quan về bài toán.....	21
3.1.1. Mô tả bài toán.....	21
3.1.2. Các chức năng chính của hệ thống	21
3.2. Phân tích hệ thống	23
3.2.1. Xác định Actor và Usecase.....	23
3.2.2. Biểu đồ Usecase.....	24
3.2.3. Biểu đồ hoạt động.....	37
3.2.4. Biểu đồ lớp phân tích các ca sử dụng	45
3.2.5. Biểu đồ tuần tự.....	48
3.2.6. Biểu đồ trạng thái	53
3.2.7. Biểu đồ lớp thực thể liên kết.....	54
3.3. Thiết kế hệ thống	55
3.3.1. Thiết kế bảng cơ sở dữ liệu	55
3.3.2. Bảng database diagram.....	58
3.4. Thiết kế giao diện	59
3.4.1. Thiết kế giao diện người dùng.....	59
3.4.2. Các nguyên tắc thiết kế giao diện	59
3.5. Cài đặt.....	60
3.5.1. Môi trường phát triển.....	60
3.5.2. Quy trình cài đặt	60
3.6. Kiểm thử phần mềm	61

3.6.1. Các loại kiểm thử.....	61
3.6.2. Kịch bản kiểm thử	61
3.6.3. Kết quả kiểm thử	62
Tiêu kết.....	62
CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC.....	63
4.1. Kết quả đạt được.....	63
4.1.1. Lý thuyết.....	63
4.1.2. Kỹ năng.....	63
4.2. Giao diện của dự án	64
4.2.1. Giao diện trang chủ.....	64
4.2.2. Giao diện chi tiết sản phẩm	65
4.2.3. Giao diện đăng nhập khách hàng.....	66
4.2.4. Giao diện đăng ký khách hàng	67
4.2.5. Giao diện thông tin khách hàng.....	68
4.2.6. Giao diện giỏ hàng.....	69
4.2.7. Giao diện sản phẩm yêu thích.....	70
4.2.8. Giao diện tìm kiếm sản phẩm.....	71
4.2.9. Giao diện danh sách đơn hàng.....	72
4.2.10. Giao diện chi tiết đơn hàng.....	73
4.2.11. Giao diện thanh toán.....	74
4.2.12. Giao diện thống kê.....	75
4.2.13. Giao diện thông tin cửa hàng.....	75
4.2.14. Giao diện danh sách sản phẩm	76
4.2.15. Giao diện thêm sản phẩm	77
4.2.16. Giao diện sửa sản phẩm.....	78
4.2.17. Giao diện quản lý đơn hàng.....	79

DANH MỤC HÌNH ẢNH

Hình 2. 1: Sơ đồ hoạt động của NuxtJS [1].....	7
Hình 2. 2: API (Application Programming Interface) [2].....	10
Hình 2. 3: Mô hình hoạt động của Laravel [4].....	15
Hình 3. 1: Sơ đồ usecase tổng quát trang chủ	24
Hình 3. 2: Sơ đồ usecase tổng quát trang quản trị.....	24
Hình 3. 3: Sơ đồ usecase đăng nhập, đăng xuất của khách hàng	25
Hình 3. 4: Sơ đồ usecase đăng nhập, đăng xuất của chủ cửa hàng	25
Hình 3. 5: Sơ đồ usecase đăng nhập, đăng xuất của quản trị viên	25
Hình 3. 6: Sơ đồ usecase đăng ký tài khoản của khách hàng.....	26
Hình 3. 7: Sơ đồ usecase đăng ký tài khoản của chủ cửa hàng.....	26
Hình 3. 8: Sơ đồ usecase xem sản phẩm	26
Hình 3. 9: Sơ đồ usecase quản lý giỏ hàng.....	27
Hình 3. 10: Sơ đồ usecase tìm kiếm sản phẩm.....	27
Hình 3. 11: Sơ đồ usecase đặt hàng.....	28
Hình 3. 12: Sơ đồ usecase quản lý thông tin khách hàng.....	28
Hình 3. 13: Sơ đồ usecase quản lý sản phẩm	29
Hình 3. 14: Sơ đồ usecase quản lý đơn hàng.....	29
Hình 3. 15: Sơ đồ usecase quản lý thông tin chủ cửa hàng.....	30
Hình 3. 16: Sơ đồ usecase thống kê cửa hàng	30
Hình 3. 17: Sơ đồ hoạt động đăng nhập	37
Hình 3. 18: Sơ đồ hoạt động đăng xuất	37
Hình 3. 19: Sơ đồ hoạt động đăng ký	38
Hình 3. 20: Sơ đồ hoạt động thêm sản phẩm	39
Hình 3. 21: Sơ đồ hoạt động sửa sản phẩm	40
Hình 3. 22: Sơ đồ hoạt động xóa sản phẩm.....	41
Hình 3. 23: Sơ đồ hoạt động đơn hàng.....	42
Hình 3. 24: Sơ đồ hoạt động đặt hàng	43
Hình 3. 25: Sơ đồ hoạt động tìm kiếm sản phẩm	44
Hình 3. 26: Biểu đồ lớp đăng nhập.....	45
Hình 3. 27: Biểu đồ lớp xem sản phẩm	45
Hình 3. 28: Biểu đồ lớp đặt hàng.....	46
Hình 3. 29: Biểu đồ lớp quản lý sản phẩm	47
Hình 3. 30: Sơ đồ tuần tự đăng nhập.....	48
Hình 3. 31: Sơ đồ tuần tự đăng xuất.....	48

Hình 3. 32: Sơ đồ tuần tự đăng ký.....	49
Hình 3. 33: Sơ đồ tuần tự xem sản phẩm	49
Hình 3. 34: Sơ đồ tuần tự quản lý sản phẩm	50
Hình 3. 35: Sơ đồ tuần tự thêm sản phẩm	50
Hình 3. 36: Sơ đồ tuần tự sửa sản phẩm.....	51
Hình 3. 37: Sơ đồ tuần tự xóa sản phẩm	51
Hình 3. 38: Sơ đồ tuần tự xem đơn hàng của khách hàng.....	52
Hình 3. 39: Sơ đồ tuần tự đặt hàng.....	52
Hình 3. 40: Sơ đồ trạng thái đăng nhập.....	53
Hình 3. 41: Sơ đồ trạng thái tìm kiếm sản phẩm.....	53
Hình 3. 42: Sơ đồ trạng thái đặt hàng.....	54
Hình 3. 43: Sơ đồ thực thể liên kết.....	54
Hình 3. 44: Sơ đồ diagram.....	58

Hình 4. 1: Giao diện trang chủ	64
Hình 4. 2: Giao diện chi tiết sản phẩm	65
Hình 4. 3: Giao diện đăng nhập khách hàng	66
Hình 4. 4: Giao diện đăng ký khách hàng	67
Hình 4. 5: Giao diện hồ sơ khách hàng	68
Hình 4. 6: Giao diện giỏ hàng	69
Hình 4. 7: Giao diện sản phẩm yêu thích	70
Hình 4. 8: Giao diện tìm kiếm sản phẩm.....	71
Hình 4. 9: Giao diện quản lý đơn hàng khách hàng	72
Hình 4. 10: Giao diện chi tiết đơn hàng	73
Hình 4. 11: Giao diện thanh toán.....	74
Hình 4. 12: Giao diện thống kê	75
Hình 4. 13: Giao diện thông tin cửa hàng	75
Hình 4. 14: Giao diện quản lý sản phẩm	76
Hình 4. 15: Giao diện thêm sản phẩm	77
Hình 4. 16: Giao diện sửa sản phẩm.....	78
Hình 4. 17: Giao diện quản lý đơn hàng của cửa hàng	79

DANH MỤC BẢNG BIỂU

Bảng 3. 1: Đặc tả usecase đăng nhập	31
Bảng 3. 2: Đặc tả usecase đăng xuất	31
Bảng 3. 3: Đặc tả usecase đăng ký	32
Bảng 3. 4: Đặc tả usecase xem sản phẩm.....	33
Bảng 3. 5: Đặc tả usecase quản lý giỏ hàng	34
Bảng 3. 6: Đặc tả usecase quản lý đơn hàng	34
Bảng 3. 7: Đặc tả usecase quản lý sản phẩm.....	35
Bảng 3. 8: Đặc tả usecase quản lý tình trạng đơn hàng.....	36
Bảng 3. 9: Bảng users	55
Bảng 3. 10: Bảng categories	55
Bảng 3. 11: Bảng images.....	55
Bảng 3. 12: Bảng products	56
Bảng 3. 13: Bảng carts.....	56
Bảng 3. 14: Bảng wishlists	57
Bảng 3. 15: Bảng orders	57
Bảng 3. 16: Bảng order_details	57
Bảng 3. 17: Bảng stores.....	58

CHƯƠNG 1. TỔNG QUÁT VỀ ĐỀ TÀI

1.1. Tổng quan đề tài

Trong kỷ nguyên số hóa hiện nay, thương mại điện tử đã trở thành một phần không thể thiếu trong đời sống và kinh doanh toàn cầu. Việc mua sắm trực tuyến không chỉ giúp tiết kiệm thời gian và công sức mà còn mở rộng cơ hội tiếp cận đến các sản phẩm và dịch vụ từ khắp nơi trên thế giới. Với sự phát triển mạnh mẽ của công nghệ và internet, các nền tảng thương mại điện tử ngày càng được ưa chuộng và trở thành một công cụ đắc lực trong việc thúc đẩy kinh doanh trực tuyến.

Một trong những xu hướng nổi bật trong lĩnh vực thương mại điện tử là sự phát triển của các sàn thương mại trực tuyến (marketplace). Các sàn thương mại này cho phép nhiều người bán cùng tham gia và cung cấp sản phẩm trên một nền tảng duy nhất, mang đến sự lựa chọn đa dạng cho người mua và tối ưu hóa trải nghiệm mua sắm trực tuyến. Với sự hỗ trợ của các công nghệ hiện đại như NuxtJS và Laravel, việc xây dựng và quản lý một sàn thương mại trực tuyến trở nên dễ dàng và hiệu quả hơn.

Mục tiêu của đề tài này là xây dựng một sàn thương mại game trực tuyến sử dụng các công nghệ tiên tiến như NuxtJS và Laravel (PHP). Đây là một giải pháp hoàn chỉnh giúp người dùng có thể dễ dàng tìm kiếm, mua bán và trao đổi các trò chơi, tài khoản game, và các dịch vụ liên quan. NuxtJS sẽ được sử dụng để phát triển giao diện người dùng (frontend) với các tính năng hiện đại, thân thiện và tối ưu hóa trải nghiệm người dùng. Laravel sẽ đóng vai trò làm nền tảng backend, đảm bảo việc quản lý dữ liệu, xử lý giao dịch và bảo mật hệ thống.

Lý do chọn đề tài này xuất phát từ những nhu cầu thực tế và tiềm năng phát triển của thị trường game trực tuyến:

- Nhu cầu cao về giao dịch và trao đổi sản phẩm game: Với sự phát triển không ngừng của ngành công nghiệp game, nhu cầu về mua bán và trao đổi các sản phẩm game như tài khoản, vật phẩm, và dịch vụ hỗ trợ ngày càng tăng cao. Một sàn thương mại chuyên biệt về game sẽ đáp ứng tốt nhu cầu này, cung cấp một nền tảng an toàn và tiện lợi cho cộng đồng game thủ.

- Tận dụng tiềm năng công nghệ mới: Sử dụng NuxtJS và Laravel không chỉ mang lại hiệu quả trong phát triển và quản lý sản phẩm thương mại mà còn giúp tối ưu hóa hiệu suất và bảo mật hệ thống. NuxtJS, với khả năng phát triển ứng dụng web tĩnh và động, kết hợp với Laravel, một framework PHP mạnh mẽ, sẽ tạo nên một giải pháp hoàn chỉnh và linh hoạt.
- Thúc đẩy thương mại điện tử trong lĩnh vực game: Đề tài này không chỉ giới hạn ở việc xây dựng một sản phẩm thương mại mà còn nhằm khám phá và thúc đẩy thương mại điện tử trong lĩnh vực game, một lĩnh vực có tiềm năng lớn nhưng chưa được khai thác đầy đủ.

Đề tài "Xây dựng sản phẩm thương mại game bằng NuxtJS và Laravel (PHP)" không chỉ mang lại giá trị thực tiễn mà còn góp phần thúc đẩy sự phát triển của thương mại điện tử trong lĩnh vực game, mở ra nhiều cơ hội mới cho cả người bán và người mua.

1.2. Lý do chọn đề tài

Ngành công nghiệp game đã và đang phát triển với tốc độ chóng mặt, trở thành một trong những ngành giải trí có doanh thu lớn nhất trên thế giới. Theo báo cáo của Newzoo, doanh thu từ thị trường game toàn cầu đã vượt qua 159 tỷ USD vào năm 2020 và tiếp tục tăng trưởng mạnh mẽ. Game không chỉ là một phương tiện giải trí mà còn là một nền tảng giao lưu, kết nối giữa các game thủ trên toàn thế giới. Với sự phát triển này, nhu cầu về các dịch vụ liên quan đến game, bao gồm mua bán tài khoản, vật phẩm trong game, và dịch vụ hỗ trợ, ngày càng trở nên cấp thiết.

Thương mại điện tử đã thay đổi cách con người mua sắm và giao dịch. Các sản phẩm thương mại điện tử như Amazon, eBay, và Alibaba đã chứng minh hiệu quả của mô hình này trong việc cung cấp một nền tảng thuận tiện và tin cậy cho cả người mua và người bán. Trong bối cảnh đó, việc xây dựng một sản phẩm thương mại chuyên biệt cho lĩnh vực game là một bước đi chiến lược nhằm đáp ứng nhu cầu cụ thể của cộng đồng game thủ.

Sản phẩm thương mại game không chỉ là nơi để mua bán, mà còn là một cộng đồng kết nối các game thủ với nhau. Việc tạo ra một môi trường an toàn, tiện lợi và chuyên nghiệp để giao dịch sẽ thúc đẩy sự phát triển của cộng đồng game, tạo ra các giá trị bền vững và lâu dài. Người dùng có thể dễ dàng trao đổi tài khoản, vật phẩm trong game và tìm kiếm các dịch vụ hỗ trợ, tạo nên một hệ sinh thái đa dạng và phong phú.

Việc lựa chọn NuxtJS và Laravel để phát triển sản phẩm thương mại game không chỉ dựa trên hiệu quả mà các công nghệ này mang lại mà còn bởi sự linh hoạt và tiềm năng mở rộng của chúng:

- **NuxtJS:** Đây là một framework dựa trên Vue.js, cho phép phát triển các ứng dụng web hiện đại với hiệu suất cao và dễ dàng tùy chỉnh. NuxtJS hỗ trợ cả server-side rendering (SSR) và static site generation (SSG), giúp tối ưu hóa tốc độ tải trang và trải nghiệm người dùng. Điều này đặc biệt quan trọng đối với một sản phẩm thương mại trực tuyến, nơi mà tốc độ và trải nghiệm người dùng quyết định sự thành công.
- **Laravel:** Laravel là một framework PHP mạnh mẽ và phổ biến, được biết đến với khả năng xây dựng các ứng dụng web phức tạp một cách nhanh chóng và an toàn. Với các tính năng như routing, middleware, authentication, và hỗ trợ ORM (Eloquent), Laravel cung cấp một nền tảng vững chắc cho việc phát triển backend của sản phẩm thương mại game.

Cuối cùng, việc thực hiện đề tài này cũng đóng góp vào sự phát triển và ứng dụng các công nghệ mới. Bằng cách triển khai các công nghệ tiên tiến như NuxtJS và Laravel, đề tài không chỉ tạo ra một sản phẩm có giá trị thực tiễn mà còn góp phần thúc đẩy sự tiến bộ trong lĩnh vực phát triển web và thương mại điện tử.

Tóm lại, đề tài "Xây dựng sản phẩm thương mại game bằng NuxtJS và Laravel (PHP)" không chỉ mang lại giá trị thực tiễn cho cộng đồng game thủ mà còn mở ra nhiều cơ hội mới cho việc phát triển thương mại điện tử trong lĩnh vực game. Sự kết hợp giữa các công nghệ tiên tiến và nhu cầu thực tế của thị trường sẽ tạo nên một nền tảng mạnh mẽ và hiệu quả, đáp ứng tốt các yêu cầu và kỳ vọng của người dùng.

1.3. Phương pháp nghiên cứu

Trong quá trình nghiên cứu và phát triển sản phẩm thương mại game bằng NuxtJS và Laravel, tôi sẽ áp dụng một loạt các phương pháp nghiên cứu để đảm bảo tính khoa học và hiệu quả của dự án. Cụ thể, chúng tôi sẽ thực hiện các bước sau:

- **Nghiên Cứu Thư Mục:** Đầu tiên, chúng tôi sẽ tiến hành nghiên cứu kỹ lưỡng về các công nghệ NuxtJS và Laravel, cũng như các nguyên tắc thiết kế web và nguyên lý

bảo mật. Việc này giúp chúng tôi hiểu sâu hơn về khả năng và hạn chế của các công nghệ này, từ đó định hình phương hướng phát triển dự án.

- **Phân Tích Yêu Cầu:** Sau đó, chúng tôi sẽ tiến hành phân tích yêu cầu từ khách hàng và người dùng cuối. Qua việc tìm hiểu nhu cầu và mong muốn của họ, chúng tôi có thể xác định rõ mục tiêu và phạm vi của dự án để xây dựng một sản phẩm đáp ứng được mong đợi.
- **Phát Triển Prototype:** Chúng tôi sẽ sử dụng phương pháp phát triển dựa trên Agile để xây dựng một phiên bản prototype của sản phẩm thương mại game. Việc này giúp chúng tôi nhanh chóng thu thập phản hồi từ khách hàng và người dùng, từ đó điều chỉnh và cải tiến sản phẩm theo hướng đúng đắn.
- **Thử Nghiệm và Đánh Giá:** Chúng tôi sẽ tiến hành các bài thử nghiệm và đánh giá để đảm bảo tính ổn định, hiệu suất và tính bảo mật của sản phẩm thương mại game. Thông qua việc kiểm tra và đánh giá kỹ lưỡng, chúng tôi có thể xác định và khắc phục các lỗi và vấn đề trước khi sản phẩm được triển khai vào môi trường thực tế.
- **Tối Ưu Hóa và Bảo Trì:** Cuối cùng, chúng tôi sẽ tiếp tục tối ưu hóa và bảo trì sản phẩm sau khi triển khai, đảm bảo rằng nó luôn hoạt động ổn định và đáp ứng được nhu cầu của người dùng. Việc này bao gồm việc theo dõi hiệu suất, thu thập phản hồi từ người dùng và cập nhật các tính năng mới để cải thiện trải nghiệm người dùng.

Bằng cách áp dụng những phương pháp nghiên cứu này, chúng tôi tin rằng chúng tôi sẽ có thể phát triển một sản phẩm thương mại game chất lượng cao, đáp ứng được nhu cầu của người dùng và mang lại giá trị cho khách hàng.

1.4. Mục tiêu của đề tài

Mục tiêu chính của đề tài "Xây dựng sản phẩm thương mại game bằng NuxtJS và Laravel (PHP)" là phát triển một nền tảng thương mại điện tử chuyên biệt cho cộng đồng game thủ. Đề tài này nhằm cung cấp một giải pháp hoàn chỉnh và hiệu quả cho việc mua bán, trao đổi các sản phẩm và dịch vụ liên quan đến game. Dưới đây là các mục tiêu cụ thể mà đề tài hướng tới:

Mục tiêu đầu tiên và quan trọng nhất là xây dựng một sản phẩm thương mại trực tuyến chuyên biệt cho game. Sản phẩm này sẽ cho phép người dùng đăng ký, đăng nhập, và thực

hiện các giao dịch mua bán tài khoản game và các dịch vụ hỗ trợ khác. Hệ thống sẽ bao gồm các tính năng chính như:

- Đăng ký và đăng nhập người dùng: Tạo tài khoản, quản lý hồ sơ người dùng và xác thực người dùng.
- Danh mục sản phẩm: Cho phép người bán đăng các sản phẩm và dịch vụ game lên sàn với đầy đủ thông tin chi tiết.
- Giỏ hàng và thanh toán: Cung cấp tính năng giỏ hàng, thanh toán trực tuyến an toàn và tiện lợi.
- Đánh giá và bình luận: Cho phép người dùng đánh giá và bình luận về sản phẩm và dịch vụ sau khi giao dịch.

Mục tiêu thứ hai là tối ưu hóa trải nghiệm người dùng (UX/UI) trên sàn thương mại. Điều này bao gồm:

- Giao diện người dùng thân thiện và trực quan: Sử dụng NuxtJS và TailwindCSS để xây dựng giao diện hiện đại, dễ sử dụng và phản hồi nhanh.
- Tối ưu hóa hiệu suất: Đảm bảo sàn thương mại hoạt động mượt mà trên các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động.
- Tính năng tìm kiếm và lọc sản phẩm: Cung cấp công cụ tìm kiếm và bộ lọc hiệu quả để người dùng dễ dàng tìm thấy sản phẩm mong muốn.

Mục tiêu tiếp theo là xây dựng hệ thống với khả năng mở rộng và quản lý dễ dàng. Hệ thống cần được thiết kế sao cho có thể dễ dàng bổ sung thêm các tính năng mới và mở rộng quy mô mà không ảnh hưởng đến hiệu suất:

- Cấu trúc modular: Sử dụng Laravel và NuxtJS để xây dựng hệ thống với cấu trúc modular, dễ dàng bảo trì và mở rộng.
- Tài liệu hóa đầy đủ: Cung cấp tài liệu chi tiết về hệ thống để hỗ trợ việc bảo trì và phát triển sau này.
- Hỗ trợ quản trị viên: Cung cấp các công cụ và giao diện quản trị viên để quản lý sản phẩm, người dùng và các giao dịch trên sàn một cách hiệu quả.

Cuối cùng, mục tiêu của đề tài là không chỉ xây dựng hệ thống mà còn đánh giá hiệu quả của nó và thực hiện các cải tiến liên tục dựa trên phản hồi của người dùng:

- Thu thập và phân tích phản hồi người dùng: Sử dụng các công cụ phân tích để thu thập dữ liệu và phản hồi từ người dùng, từ đó đưa ra các cải tiến cần thiết.
- Kiểm tra và cải tiến định kỳ: Thực hiện các kiểm tra định kỳ và cập nhật hệ thống để đảm bảo nó luôn hoạt động hiệu quả và đáp ứng nhu cầu của người dùng.
- Tóm lại, đề tài "Xây dựng sàn thương mại game bằng NuxtJS và Laravel (PHP)" đặt ra các mục tiêu cụ thể nhằm tạo ra một nền tảng thương mại điện tử chuyên biệt cho game, mang lại trải nghiệm người dùng tốt nhất, đảm bảo tính bảo mật và an toàn, có khả năng mở rộng và dễ dàng quản lý, cùng với sự cải tiến liên tục dựa trên phản hồi từ người dùng.

1.5. Nội dung thực hiện

Để hoàn thành đề tài "Xây dựng sàn thương mại game bằng NuxtJS và Laravel (PHP)", nội dung thực hiện được chia thành các giai đoạn cụ thể. Trước tiên là giai đoạn phân tích yêu cầu, trong đó tiến hành thu thập và phân tích yêu cầu từ người dùng và các bên liên quan thông qua phỏng vấn và khảo sát. Các yêu cầu này sẽ được phân loại và mô tả chi tiết bằng các công cụ như Use Case Diagram và User Stories. Tiếp theo là giai đoạn lựa chọn công nghệ hỗ trợ như cơ sở dữ liệu MySQL và CSS framework (TailwindCSS).

Giai đoạn tiếp theo là giai đoạn phát triển và triển khai hệ thống, với việc xây dựng giao diện người dùng bằng NuxtJS và TailwindCSS để đảm bảo tính tương tác và hiệu suất cao, phát triển backend bằng Laravel để xây dựng các API và logic xử lý trên server, và triển khai hệ thống lên các môi trường thử nghiệm và sản xuất sử dụng Docker và CI/CD.

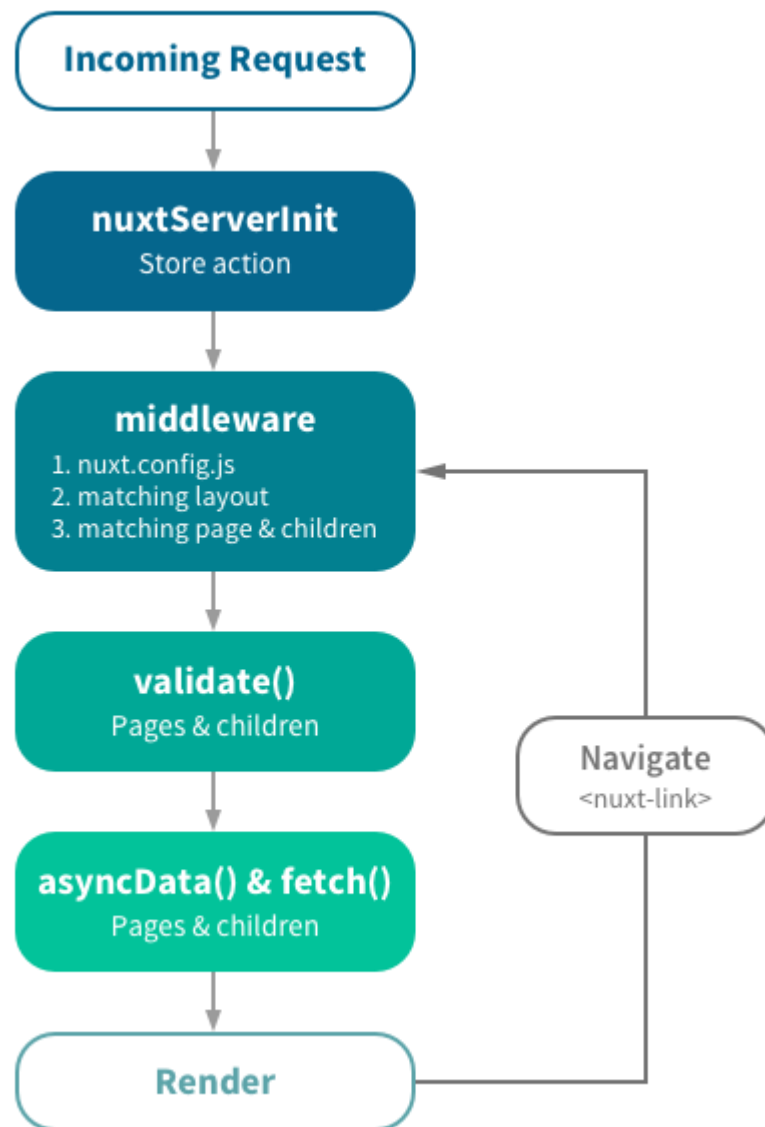
Cuối cùng là giai đoạn tài liệu hóa và bảo trì, viết tài liệu chi tiết về kiến trúc hệ thống, thiết kế cơ sở dữ liệu, giao diện người dùng, API và các chức năng chính, cung cấp hướng dẫn sử dụng và tài liệu kỹ thuật, cùng với việc thực hiện các hoạt động bảo trì định kỳ để đảm bảo hệ thống luôn ổn định và cập nhật các tính năng mới theo nhu cầu người dùng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về NuxtJS

2.1.1. Giới Thiệu về NuxtJS

NuxtJS là một framework phổ biến dựa trên Vue.js, được thiết kế để giúp các nhà phát triển xây dựng các ứng dụng web động và tĩnh với trải nghiệm người dùng tuyệt vời. NuxtJS mang lại nhiều lợi ích, chẳng hạn như cấu trúc thư mục rõ ràng, tích hợp tốt với Vue.js, hỗ trợ các tính năng như Server-Side Rendering (SSR), Static Site Generation (SSG), và khả năng mở rộng mạnh mẽ. Trong phần này, chúng ta sẽ tìm hiểu sâu hơn về các đặc điểm chính và ưu nhược điểm của NuxtJS.



Hình 2. 1: Sơ đồ hoạt động của NuxtJS [1]

2.1.2. Đặc Điểm Chính của NuxtJS

- **Universal Vue Applications:** NuxtJS cho phép phát triển các ứng dụng Vue.js đa nền tảng, có khả năng render trên máy chủ và trình duyệt. Điều này giúp cải thiện hiệu suất và trải nghiệm người dùng bằng cách tối ưu hóa thời gian tải trang và SEO. Bằng cách này, các ứng dụng có thể đạt được hiệu suất tốt hơn và tăng khả năng tương tác.
- **Tối Ưu Hóa SEO:** Một điểm mạnh của NuxtJS là khả năng tạo ra các trang tĩnh tối ưu cho SEO thông qua SSR và SSG. Điều này giúp ứng dụng của bạn có thể được index nhanh chóng và hiệu quả trên các công cụ tìm kiếm, đảm bảo sự hiển thị và tiếp cận của người dùng tốt hơn.
- **Quản Lý Tập Trung:** NuxtJS cung cấp một cấu trúc thư mục mặc định, giúp tổ chức và quản lý các thành phần của ứng dụng một cách hiệu quả. Điều này làm cho việc phát triển ứng dụng trở nên dễ dàng hơn và tăng tính tổ chức của dự án, đặc biệt là trong các dự án lớn và phức tạp.
- **Hỗ Trợ Vue.js Ecosystem:** NuxtJS tích hợp sẵn các tính năng mạnh mẽ của Vue.js như Vuex (quản lý trạng thái), Vue Router (định tuyến), và Vue Server Renderer (SSR). Điều này giúp tăng cường hiệu suất và khả năng mở rộng của ứng dụng, và cho phép bạn sử dụng những công cụ quen thuộc của Vue.js một cách dễ dàng.
- **Plugin và Middleware:** NuxtJS cho phép tích hợp các plugin và middleware một cách dễ dàng để mở rộng chức năng của ứng dụng. Bạn có thể sử dụng các plugin có sẵn hoặc tự viết plugin của riêng mình để đáp ứng nhu cầu cụ thể của dự án, tùy chỉnh và mở rộng tính năng của ứng dụng theo ý muốn.

2.1.3. Ứng dụng của NuxtJS

NuxtJS có thể được sử dụng để phát triển nhiều loại ứng dụng khác nhau, từ các trang web tĩnh, blog cá nhân, đến các ứng dụng web phức tạp cần tương tác động và hiệu suất cao. Với khả năng tích hợp SSR và SSG, NuxtJS đặc biệt hữu ích cho các dự án cần tối ưu SEO và tốc độ tải trang nhanh. Hơn nữa, khả năng mở rộng và tích hợp module giúp NuxtJS phù hợp với các ứng dụng web hiện đại cần tính linh hoạt và dễ bảo trì.

2.1.4. Ưu điểm của NuxtJS

- Tích hợp SSR và SSG: Cho phép cải thiện SEO và tốc độ tải trang, mang lại trải nghiệm người dùng tốt hơn.
- Routing tự động: Giảm thiểu thời gian cấu hình và quản lý các route trong ứng dụng.
- Hệ thống module mạnh mẽ: Dễ dàng tích hợp các tính năng mới mà không cần nhiều cấu hình phức tạp.
- Cộng đồng hỗ trợ lớn: Nhiều tài liệu và nguồn tài nguyên học tập giúp dễ dàng giải quyết các vấn đề gặp phải trong quá trình phát triển.
- Phát triển nhanh chóng: Cấu trúc thư mục rõ ràng và các công cụ hỗ trợ phát triển giúp đẩy nhanh quá trình phát triển ứng dụng.

2.1.5. Nhược điểm của NuxtJS

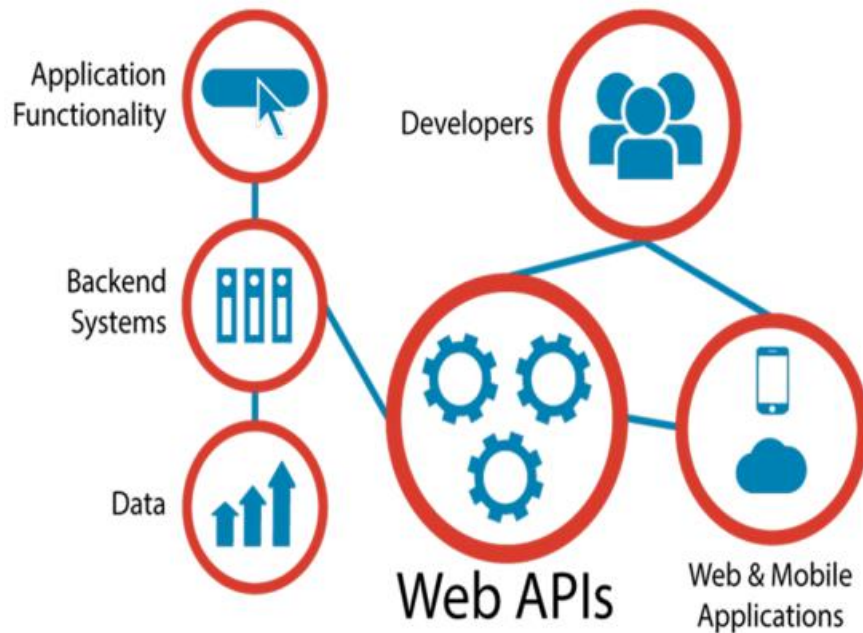
- Độ phức tạp tăng: Với những người mới bắt đầu, việc hiểu và làm chủ được NuxtJS có thể gặp nhiều khó khăn do các khái niệm như SSR, SSG, và các cấu hình phức tạp.
- Thời gian xây dựng lâu hơn: So với việc phát triển một SPA (Single Page Application) thông thường, việc sử dụng SSR hoặc SSG có thể làm tăng thời gian build của dự án.
- Hiệu suất có thể bị ảnh hưởng: Trong một số trường hợp, nếu không được tối ưu hóa đúng cách, việc sử dụng SSR có thể làm giảm hiệu suất của ứng dụng, đặc biệt là với các ứng dụng có nhiều yêu cầu xử lý trên server.

Tóm lại, NuxtJS là một framework mạnh mẽ và linh hoạt, mang lại nhiều lợi ích cho việc phát triển các ứng dụng web hiện đại. Với các tính năng như SSR, SSG, routing tự động, và hệ thống module mạnh mẽ, NuxtJS giúp cải thiện hiệu suất, tối ưu SEO, và giảm thiểu thời gian phát triển, đồng thời mang lại trải nghiệm người dùng tốt nhất. Tuy nhiên, như mọi công nghệ, NuxtJS cũng có những nhược điểm cần được cân nhắc khi áp dụng vào các dự án cụ thể.

2.2. API (Application Programming Interface)

2.2.1. API là gì

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.



Hình 2. 2: API (Application Programming Interface) [2]

2.2.2. Ứng dụng của API

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai

khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

2.2.3. Web API là gì

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Những điểm nổi bật của Web API:

- Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.
- Tự động hóa sản phẩm: Với web API, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.
- Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.
- Cập nhật thông tin thời gian thực: API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.
- Có tiêu chuẩn chung để sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

2.2.4. Cách thức hoạt động của Web API

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

2.2.5. Ưu và nhược điểm của Web API

a) Ưu điểm

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.
- Linh hoạt với các định dạng dữ liệu khi trả về client: Json, XML hay định dạng khác.
- Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.
- Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, Json.
- Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.
- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

b) Nhược điểm

Do web API còn khá mới nên chưa thể đánh giá nhiều về nhược điểm của mô hình này. Tuy nhiên, có hai nhược điểm dễ dàng nhận thấy:

- Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST
- Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt
- Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành
- Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

2.3. Tổng quan về TailwindCSS

TailwindCSS là một framework CSS tiện dụng và linh hoạt, được thiết kế để giúp các nhà phát triển xây dựng các giao diện web hiện đại một cách nhanh chóng và dễ dàng. Khác với các framework CSS truyền thống như Bootstrap hoặc Foundation, TailwindCSS không cung cấp các thành phần UI sẵn có mà thay vào đó cung cấp các lớp tiện ích (utility classes) cho phép bạn xây dựng các thiết kế tùy chỉnh mà không cần phải viết CSS từ đầu. Trong phần này, chúng ta sẽ tìm hiểu sâu hơn về các đặc điểm chính, ưu điểm và nhược điểm của TailwindCSS.

2.3.1. Các đặc điểm chính

- **Utility-First:** TailwindCSS cung cấp một bộ lớp tiện ích phong phú cho phép bạn áp dụng các kiểu CSS trực tiếp trong HTML. Điều này giúp tăng tốc độ phát triển và giảm thiểu việc viết CSS thủ công.
- **Tùy Chỉnh Cao:** TailwindCSS cho phép tùy chỉnh sâu thông qua cấu hình file `tailwind.config.js`, nơi bạn có thể định nghĩa các màu sắc, khoảng cách, và các thông số khác theo ý muốn của mình.
- **Responsive Design:** TailwindCSS hỗ trợ thiết kế responsive một cách dễ dàng với các lớp tiện ích dành riêng cho từng kích thước màn hình, giúp xây dựng các giao diện linh hoạt và thích ứng với nhiều thiết bị khác nhau.
- **Chế Độ JIT (Just-In-Time):** TailwindCSS có chế độ JIT giúp biên dịch CSS chỉ khi cần thiết, giúp giảm thiểu kích thước file CSS và tối ưu hóa hiệu suất.
- **Cộng Đồng Mạnh:** TailwindCSS có một cộng đồng lớn và năng động, với nhiều plugin, template và công cụ hỗ trợ giúp tăng cường khả năng và tính linh hoạt của framework.

2.3.2. Ưu điểm của TailwindCSS

- **Tăng Tốc Độ Phát Triển:** Sử dụng các lớp tiện ích giúp tăng tốc độ phát triển giao diện bằng cách giảm thiểu thời gian viết và duy trì CSS.
- **Tính Nhất Quán Cao:** Sử dụng các lớp tiện ích đồng nhất giúp đảm bảo tính nhất quán trong thiết kế giao diện trên toàn bộ dự án.
- **Tùy Chỉnh Dễ Dàng:** Khả năng tùy chỉnh cao của TailwindCSS cho phép bạn dễ dàng điều chỉnh các thông số thiết kế để phù hợp với yêu cầu cụ thể của dự án.

- **Hỗ Trợ Responsive Tốt:** TailwindCSS giúp tạo ra các thiết kế responsive một cách đơn giản và hiệu quả, với các lớp tiện ích dành riêng cho từng kích thước màn hình.
- **Hiệu Suất Tối Ưu:** Chế độ JIT giúp tối ưu hóa hiệu suất bằng cách biên dịch CSS chỉ khi cần thiết, giúp giảm thiểu kích thước file CSS và tăng tốc độ tải trang.

2.3.3. Nhược điểm của TailwindCSS

- **Tăng Kích Thước HTML:** Việc sử dụng nhiều lớp tiện ích trong HTML có thể làm tăng kích thước file HTML, làm cho nó trở nên khó đọc và duy trì.
- **Khó Bắt Đầu Với Người Mới:** Đối với những người mới bắt đầu, việc làm quen với cách tiếp cận utility-first và việc sử dụng hàng trăm lớp tiện ích có thể gặp nhiều khó khăn.
- **Phụ Thuộc Vào Cấu Hình:** TailwindCSS yêu cầu cấu hình ban đầu thông qua file `tailwind.config.js`, điều này có thể phức tạp đối với những người mới làm quen với framework.
- **Phải Biết CSS Để Tùy Chỉnh:** Mặc dù TailwindCSS cung cấp nhiều lớp tiện ích, để tùy chỉnh sâu hơn, bạn vẫn cần hiểu biết về CSS.

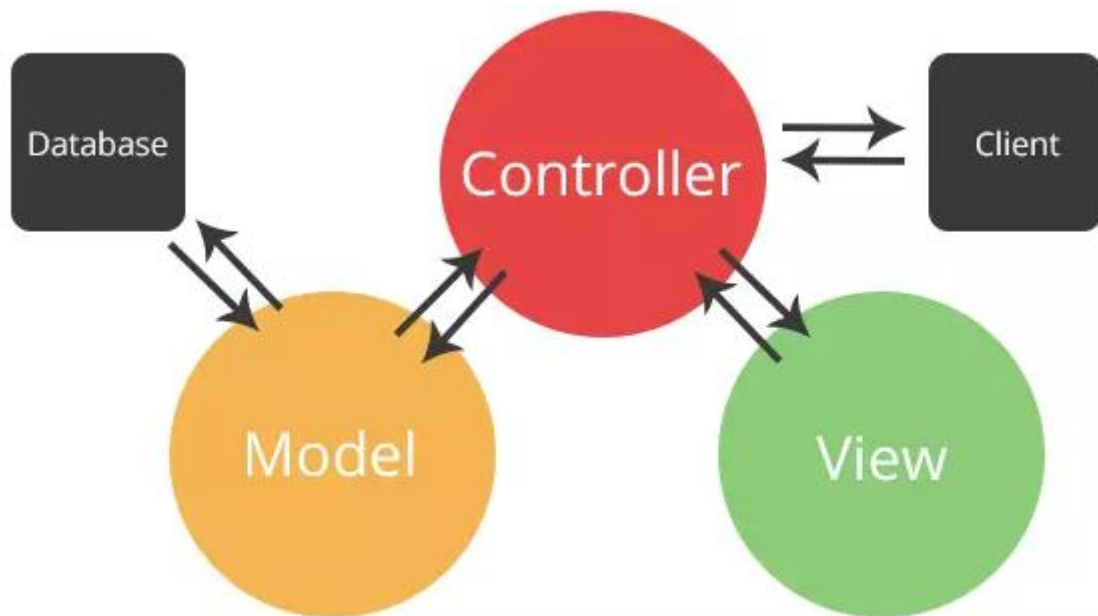
2.3.4. Ứng dụng của TailwindCSS

TailwindCSS có thể được sử dụng trong nhiều loại dự án web khác nhau, từ các trang web cá nhân, blog, đến các ứng dụng web phức tạp. Với khả năng tùy chỉnh cao và hỗ trợ thiết kế responsive, TailwindCSS đặc biệt hữu ích cho các dự án yêu cầu sự linh hoạt và khả năng thích ứng nhanh chóng. Ngoài ra, với sự hỗ trợ mạnh mẽ từ cộng đồng, các nhà phát triển có thể dễ dàng tìm kiếm sự trợ giúp và tài nguyên để nâng cao chất lượng và hiệu suất dự án.

Tóm lại, TailwindCSS là một framework CSS mạnh mẽ và linh hoạt, mang lại nhiều lợi ích cho việc phát triển giao diện web hiện đại. Với triết lý utility-first, khả năng tùy chỉnh cao và sự hỗ trợ mạnh mẽ từ cộng đồng, TailwindCSS giúp tăng tốc độ phát triển, đảm bảo tính nhất quán và tối ưu hóa hiệu suất giao diện web. Tuy nhiên, như mọi công nghệ, TailwindCSS cũng có những nhược điểm cần được cân nhắc khi áp dụng vào các dự án cụ thể.

2.4. Tổng quan về Laravel PHP

Laravel là một framework PHP hiện đại, được xây dựng dựa trên nhiều nguyên tắc và công cụ từ các framework khác như Ruby on Rails, ASP.NET MVC và Sinatra. Laravel cung cấp một bộ công cụ toàn diện bao gồm hệ thống routing mạnh mẽ, cơ chế quản lý cơ sở dữ liệu tiện lợi, hệ thống xác thực và quản lý người dùng tích hợp, cùng nhiều thư viện hữu ích khác. Laravel hướng tới việc tối ưu hóa trải nghiệm phát triển, giúp các nhà phát triển tạo ra các ứng dụng web chất lượng cao một cách nhanh chóng và dễ dàng.



Hình 2. 3: Mô hình hoạt động của Laravel [4]

2.4.1. Các đặc điểm chính của Laravel

- **Eloquent ORM:** Eloquent là một ORM (Object-Relational Mapping) mạnh mẽ và dễ sử dụng của Laravel. Nó cho phép các nhà phát triển tương tác với cơ sở dữ liệu một cách trực quan thông qua các mô hình (models), giúp giảm thiểu mã SQL thủ công.
- **Blade Templating Engine:** Blade là công cụ template mạnh mẽ của Laravel, cho phép các nhà phát triển tạo ra các giao diện web động một cách dễ dàng. Blade hỗ trợ các biểu thức điều kiện, vòng lặp, và khả năng kế thừa template, giúp tổ chức mã HTML một cách sạch sẽ và hiệu quả.
- **Routing Linh Hoạt:** Laravel cung cấp hệ thống routing mạnh mẽ, cho phép các nhà phát triển định nghĩa các route một cách rõ ràng và dễ dàng. Hệ thống

routing hỗ trợ các route RESTful, middleware, và khả năng nhóm route, giúp tổ chức ứng dụng một cách logic và rõ ràng.

- **Hệ Thống Migration và Seeding:** Laravel cung cấp công cụ migration và seeding mạnh mẽ, giúp quản lý và phiên bản hóa cơ sở dữ liệu một cách dễ dàng. Migration cho phép các nhà phát triển định nghĩa các thay đổi trong cấu trúc cơ sở dữ liệu và áp dụng chúng một cách nhất quán trên nhiều môi trường. Seeding cho phép khởi tạo dữ liệu mẫu một cách tiện lợi.
- **Authentication và Authorization:** Laravel tích hợp sẵn các cơ chế xác thực và phân quyền mạnh mẽ, giúp các nhà phát triển xây dựng các hệ thống quản lý người dùng một cách dễ dàng và bảo mật.
- **Task Scheduling:** Laravel cung cấp công cụ task scheduling tiện lợi, cho phép lập lịch thực hiện các tác vụ định kỳ như gửi email, sao lưu cơ sở dữ liệu, hoặc các công việc bảo trì khác.
- **Hỗ Trợ RESTful API:** Laravel hỗ trợ xây dựng các API RESTful một cách dễ dàng, với các công cụ hỗ trợ như Laravel Passport và Laravel Sanctum, giúp quản lý xác thực và phân quyền API một cách bảo mật và hiệu quả.

2.4.2. Ưu điểm của Laravel

- **Tăng Tốc Độ Phát Triển:** Laravel cung cấp nhiều công cụ và thư viện tích hợp, giúp giảm thiểu thời gian viết mã và tăng tốc độ phát triển ứng dụng.
- **Tính Nhất Quán và Dễ Bảo Trì:** Việc tuân theo mô hình MVC và sử dụng các công cụ như Eloquent ORM và Blade templating engine giúp mã nguồn được tổ chức một cách nhất quán và dễ bảo trì.
- **Bảo Mật Cao:** Laravel tích hợp nhiều tính năng bảo mật như mã hóa dữ liệu, bảo vệ chống tấn công CSRF, và hệ thống xác thực mạnh mẽ, giúp bảo vệ ứng dụng khỏi các mối đe dọa bảo mật phổ biến.
- **Cộng Đồng Lớn:** Laravel có một cộng đồng phát triển lớn và năng động, cung cấp nhiều tài liệu, plugin, và hỗ trợ giúp các nhà phát triển dễ dàng tìm kiếm sự giúp đỡ và giải quyết các vấn đề kỹ thuật.
- **Hỗ Trợ Testing Tốt:** Laravel tích hợp sẵn PHPUnit, giúp viết và thực hiện các bài kiểm tra tự động một cách dễ dàng, đảm bảo chất lượng mã nguồn và giảm thiểu lỗi phát sinh.

2.4.3. Nhược điểm của Laravel

- **Hiệu Suất Không Cao:** So với một số framework PHP khác như Lumen (một phiên bản nhỏ gọn của Laravel) hoặc các framework thuần PHP không sử dụng nhiều tính năng tích hợp, Laravel có thể chậm hơn do tích hợp nhiều tính năng phức tạp.
- **Đòi Hỏi Tài Nguyên Hệ Thống:** Laravel yêu cầu nhiều tài nguyên hệ thống hơn để chạy một cách hiệu quả, đặc biệt là khi sử dụng trên các ứng dụng lớn và phức tạp.
- **Độ Phức Tạp Tăng Lên Với Dự Án Lớn:** Mặc dù Laravel rất phù hợp cho các dự án nhỏ và trung bình, việc quản lý một dự án lớn với nhiều module và thành phần có thể trở nên phức tạp và khó khăn nếu không được tổ chức và quản lý một cách tốt.

2.5. Tổng quan về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) dựa trên ngôn ngữ SQL (Structured Query Language), được thiết kế để quản lý và truy vấn dữ liệu một cách hiệu quả. MySQL hỗ trợ nhiều hệ điều hành khác nhau, bao gồm Linux, Windows và macOS, và có thể tích hợp dễ dàng với nhiều ngôn ngữ lập trình như PHP, Java, Python, và nhiều ngôn ngữ khác. Với mô hình client-server, MySQL cho phép người dùng quản lý và truy cập dữ liệu từ xa một cách tiện lợi và bảo mật.

2.5.1. Ứng dụng của Laravel

Laravel có thể được sử dụng để phát triển nhiều loại ứng dụng web khác nhau, từ các trang web cá nhân, blog, đến các ứng dụng thương mại điện tử phức tạp và hệ thống quản lý doanh nghiệp. Với sự hỗ trợ mạnh mẽ cho việc xây dựng RESTful API, Laravel cũng rất phù hợp cho các dự án cần tương tác dữ liệu với nhiều dịch vụ khác nhau. Ngoài ra, Laravel cũng được sử dụng rộng rãi trong việc phát triển các ứng dụng SaaS (Software as a Service) nhờ vào khả năng mở rộng và tích hợp dễ dàng.

Tóm lại, Laravel là một framework PHP mạnh mẽ và linh hoạt, cung cấp nhiều công cụ và tính năng giúp tăng tốc độ phát triển, đảm bảo tính nhất quán và bảo mật cho ứng dụng. Với sự hỗ trợ mạnh mẽ từ cộng đồng và khả năng mở rộng dễ dàng, Laravel là một lựa chọn lý tưởng cho việc phát triển các ứng dụng web hiện đại. Tuy

nhiên, các nhà phát triển cần cân nhắc các nhược điểm của Laravel và áp dụng các biện pháp tối ưu hóa phù hợp để đảm bảo hiệu suất và quản lý hiệu quả các dự án lớn.

2.5.2. Các đặc điểm chính của MySQL

- **Hiệu Suất Cao:** MySQL được tối ưu hóa cho hiệu suất cao và khả năng xử lý lượng dữ liệu lớn một cách nhanh chóng và hiệu quả. Nó hỗ trợ nhiều kiểu lưu trữ (storage engines) như InnoDB và MyISAM, giúp tối ưu hóa việc lưu trữ và truy vấn dữ liệu.
- **Tính Năng Bảo Mật:** MySQL cung cấp nhiều tính năng bảo mật mạnh mẽ như xác thực người dùng, quản lý quyền truy cập, và mã hóa dữ liệu. Điều này giúp bảo vệ dữ liệu khỏi các truy cập trái phép và đảm bảo tính toàn vẹn của dữ liệu.
- **Khả Năng Mở Rộng:** MySQL hỗ trợ việc mở rộng quy mô theo chiều ngang thông qua việc sử dụng các kỹ thuật như replication và clustering, giúp cải thiện khả năng chịu tải và độ tin cậy của hệ thống.
- **Quản Lý Dữ Liệu Linh Hoạt:** MySQL hỗ trợ các giao dịch (transactions) và khóa (locking) để đảm bảo tính nhất quán và toàn vẹn của dữ liệu trong các thao tác cập nhật phức tạp. Hơn nữa, MySQL còn hỗ trợ các chỉ mục (indexes), khóa ngoại (foreign keys), và các tính năng khác giúp tối ưu hóa việc quản lý và truy vấn dữ liệu.
- **Công Cụ Quản Lý:** MySQL cung cấp nhiều công cụ quản lý dữ liệu như MySQL Workbench, một giao diện đồ họa giúp người dùng thiết kế, phát triển, và quản lý cơ sở dữ liệu một cách dễ dàng. Ngoài ra, MySQL còn có các công cụ dòng lệnh (command-line tools) mạnh mẽ cho việc quản trị và bảo trì cơ sở dữ liệu.

2.5.3. Ưu điểm

- **Miễn Phí và Mã Nguồn Mở:** MySQL là phần mềm mã nguồn mở và miễn phí, giúp giảm chi phí bản quyền và cung cấp sự linh hoạt trong việc tùy chỉnh và mở rộng hệ thống theo nhu cầu cụ thể của người dùng.
- **Cộng Đồng Hỗ Trợ Lớn:** MySQL có một cộng đồng người dùng và phát triển rộng lớn, cung cấp nhiều tài liệu, diễn đàn, và tài nguyên học tập để hỗ trợ người dùng trong việc giải quyết các vấn đề kỹ thuật và tối ưu hóa hệ thống.

- **Tích Hợp Dễ Dàng:** MySQL có thể tích hợp dễ dàng với nhiều ngôn ngữ lập trình và công nghệ web, giúp các nhà phát triển xây dựng các ứng dụng web và dịch vụ trực tuyến một cách nhanh chóng và hiệu quả.
- **Tính Tương Thích Cao:** MySQL hỗ trợ nhiều nền tảng và hệ điều hành khác nhau, giúp đảm bảo tính tương thích và linh hoạt trong việc triển khai hệ thống trên nhiều môi trường khác nhau.
- **Hiệu Suất và Khả Năng Mở Rộng:** MySQL cung cấp hiệu suất cao và khả năng mở rộng linh hoạt, giúp đáp ứng các yêu cầu về tốc độ xử lý và lưu trữ dữ liệu của các ứng dụng web và doanh nghiệp hiện đại.

2.5.4. Nhược điểm

- **Hạn Chế Trong Các Tính Năng Nâng Cao:** Mặc dù MySQL rất mạnh mẽ trong việc xử lý các tác vụ cơ bản, nó có thể thiếu một số tính năng nâng cao và phức tạp so với các hệ quản trị cơ sở dữ liệu khác như PostgreSQL.
- **Khó Khăn Trong Quản Trị Hệ Thống Lớn:** Quản lý và tối ưu hóa MySQL trong các hệ thống lớn và phức tạp có thể đòi hỏi kiến thức và kỹ năng chuyên sâu, điều này có thể gây khó khăn cho các quản trị viên cơ sở dữ liệu mới hoặc thiếu kinh nghiệm.
- **Giao Dịch Không Được Hỗ Trợ Tốt Trong Một Số Storage Engines:** Một số kiểu lưu trữ như MyISAM không hỗ trợ giao dịch và khóa ngoại, điều này có thể làm giảm tính nhất quán và toàn vẹn của dữ liệu trong các ứng dụng yêu cầu quản lý giao dịch phức tạp.

2.5.5. Ứng dụng thực tiễn

MySQL được sử dụng rộng rãi trong nhiều loại ứng dụng khác nhau, từ các trang web cá nhân, blog, đến các hệ thống thương mại điện tử và ứng dụng quản lý doanh nghiệp phức tạp. MySQL đặc biệt phổ biến trong cộng đồng phát triển web, với nhiều nền tảng CMS (Content Management Systems) như WordPress, Joomla, và Drupal sử dụng MySQL làm hệ quản trị cơ sở dữ liệu mặc định. Ngoài ra, MySQL cũng được sử dụng trong nhiều ứng dụng doanh nghiệp và hệ thống quản lý dữ liệu lớn, nhờ vào khả năng mở rộng và hiệu suất cao.

Tóm lại, MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ và linh hoạt, cung cấp nhiều tính năng và công cụ hữu ích cho việc quản lý và truy vấn dữ liệu. Với hiệu suất cao, tính bảo mật mạnh mẽ, và khả năng mở rộng linh hoạt, MySQL là một lựa chọn lý tưởng cho nhiều loại ứng dụng web và doanh nghiệp. Tuy nhiên, các nhà phát triển và quản trị viên cơ sở dữ liệu cần cân nhắc các nhược điểm của MySQL và áp dụng các biện pháp tối ưu hóa phù hợp để đảm bảo hiệu suất và tính nhất quán của hệ thống.

Tiểu kết

Chương 2 đã trình bày tổng quan về các công nghệ chính được sử dụng trong việc xây dựng sản phẩm thương mại game, bao gồm NuxtJS, Laravel, TailwindCSS và MySQL. NuxtJS được giới thiệu như một framework mạnh mẽ cho phát triển ứng dụng web với tính năng SSR và static site generation, giúp tối ưu hóa hiệu suất và trải nghiệm người dùng. Laravel, một framework PHP nổi bật, cung cấp các công cụ mạnh mẽ như Eloquent ORM và hệ thống routing linh hoạt, giúp đơn giản hóa quá trình phát triển và quản lý ứng dụng. TailwindCSS, với cách tiếp cận tiện ích-first, cho phép tùy chỉnh giao diện người dùng một cách nhanh chóng và dễ dàng, trong khi MySQL đảm bảo quản lý dữ liệu hiệu quả và ổn định. Những kiến thức nền tảng này sẽ hỗ trợ việc thiết kế và triển khai sản phẩm thương mại game trong các chương tiếp theo, đảm bảo quá trình phát triển diễn ra suôn sẻ và hiệu quả.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Tổng quan về bài toán

3.1.1. Mô tả bài toán

Trong bối cảnh phát triển mạnh mẽ của ngành công nghiệp game và sự bùng nổ của các nền tảng thương mại điện tử, việc xây dựng một sàn thương mại điện tử chuyên biệt cho việc mua các sản phẩm game trở nên cấp thiết hơn bao giờ hết. Mục tiêu của bài toán này là thiết kế và phát triển một nền tảng trực tuyến cho phép người dùng mua các tựa game. Hệ thống cần có chức năng quản lý sản phẩm và dịch vụ, cho phép người dùng đăng bán, thêm, sửa, xóa và quản lý các sản phẩm của mình một cách dễ dàng.

Để đảm bảo giao dịch an toàn, hệ thống phải hỗ trợ các phương thức thanh toán trực tuyến phổ biến và cung cấp cơ chế xác nhận giao dịch để bảo vệ quyền lợi của cả người mua và người bán. Chức năng tìm kiếm và gợi ý sản phẩm giúp người dùng dễ dàng tìm kiếm sản phẩm dựa trên nhiều tiêu chí và nhận được gợi ý dựa trên lịch sử tìm kiếm và mua hàng. Hệ thống cũng cho phép người dùng đánh giá và bình luận về các sản phẩm, giúp người mua tiềm năng có thêm thông tin để ra quyết định.

Hệ thống sẽ sử dụng NuxtJS để xây dựng giao diện người dùng tương tác và thân thiện, Laravel để phát triển các API và xử lý logic nghiệp vụ phức tạp, MySQL để lưu trữ dữ liệu và TailwindCSS để xây dựng giao diện người dùng nhanh chóng và hiệu quả. Lợi ích của hệ thống này là mang lại sự tiện lợi cho người dùng, mở rộng thị trường cho người bán và đảm bảo các giao dịch an toàn và tin cậy. Bằng cách sử dụng các công nghệ hiện đại như NuxtJS và Laravel, hệ thống sẽ cung cấp trải nghiệm người dùng tốt, hiệu suất cao và khả năng mở rộng dễ dàng.

3.1.2. Các chức năng chính của hệ thống

Xây dựng website gồm các chức năng sau:

Trang Chủ:

- Hiển thị sản phẩm nổi bật, đề xuất và khuyến mãi.
- Tìm kiếm nhanh các sản phẩm.

Danh Mục Sản Phẩm:

- Phân loại theo từ, sắp xếp và loại sản phẩm
- Hiện thị danh sách sản phẩm với hình ảnh và giá.

Trang Chi Tiết Sản Phẩm:

- Mô tả chi tiết về sản phẩm, bao gồm đặc điểm.
- Hiện thị ảnh nhiều góc của sản phẩm.

Giỏ Hàng và Thanh Toán:

- Thêm sản phẩm vào giỏ hàng.
- Hiện thị sản phẩm trong giỏ hàng.
- Tính tổng tiền.
- Lựa chọn phương thức thanh toán (thẻ ngân hàng, chuyển khoản).

Tài Khoản Người Dùng:

- Đăng ký và đăng nhập người dùng.
- Quản lý thông tin cá nhân.
- Xem lịch sử mua hàng và theo dõi đơn hàng.

Quản Lý Đơn Hàng (dành cho chủ cửa hàng):

- Hiện thị đơn hàng.
- Xem, xác nhận, và xử lý đơn hàng từ khách hàng.

Tìm Kiếm và Lọc Danh Sách Sản Phẩm (dành cho chủ cửa hàng):

- Hỗ trợ tìm kiếm nâng cao với từ khóa và bộ lọc.
- Lọc theo danh trạng thái của sản phẩm.

Thống kê doanh số bán hàng và lượt truy cập.

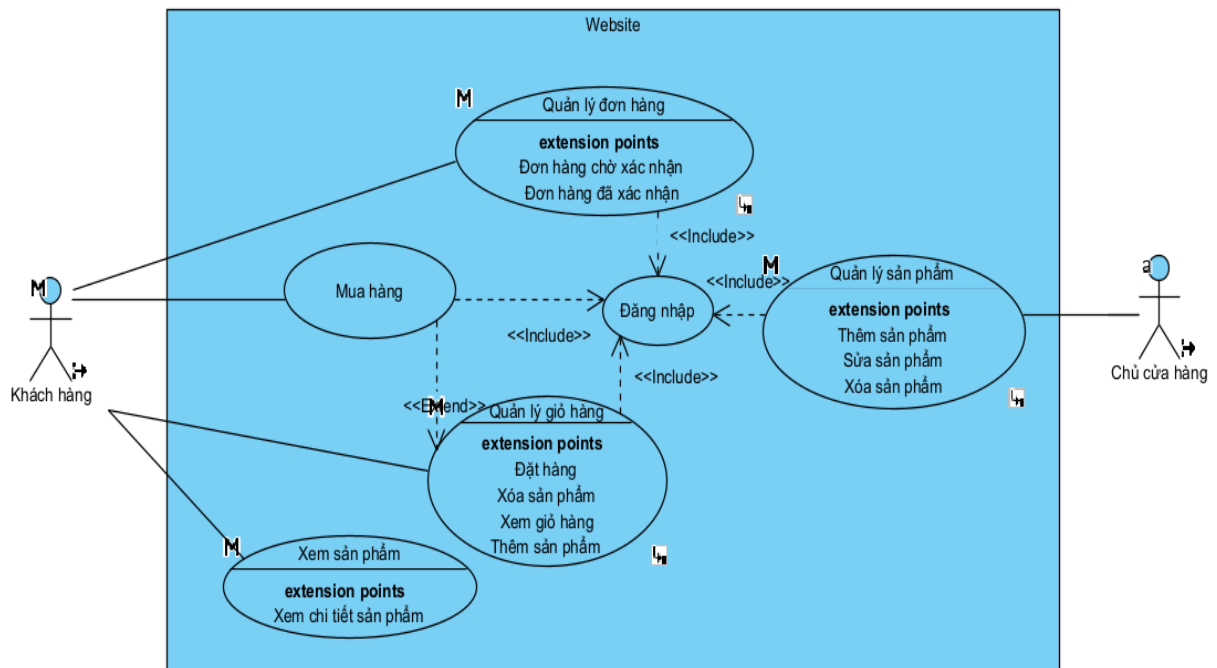
3.2. Phân tích hệ thống

3.2.1. Xác định Actor và Usecase

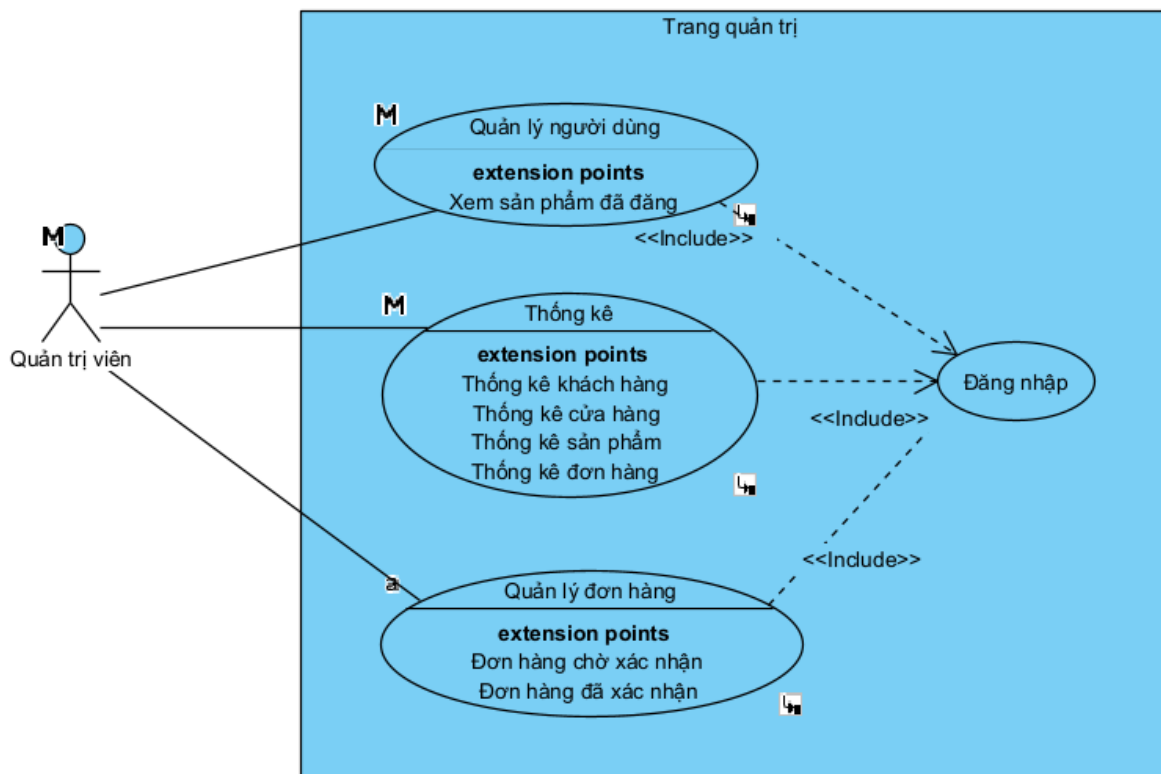
STT	ACTOR	USECASE
1	Chủ cửa hàng	<ul style="list-style-type: none"> - Quản lý sản phẩm: Thêm sửa xóa - Thống kê: Doanh thu và dữ liệu
2	Khách hàng	<ul style="list-style-type: none"> - Đăng ký tài khoản - Đăng nhập, đăng xuất - Quản lý giỏ hàng: thêm và xóa sản phẩm - Quản lý sản phẩm yêu thích: thêm và xóa sản phẩm yêu thích - Tìm kiếm sản phẩm - Đặt hàng - Quản lý tài khoản cá nhân: <ul style="list-style-type: none"> + Đổi mật khẩu + Thay đổi thông tin + Xem lịch sử mua hàng
3	Admin	<ul style="list-style-type: none"> - Đăng nhập, đăng xuất - Quản lý đơn hàng: duyệt - Quản lý tài khoản: khách hàng và chủ cửa hàng - Thống kê: Doanh thu và dữ liệu

3.2.2. Biểu đồ Usecase

a) Usecase tổng quát



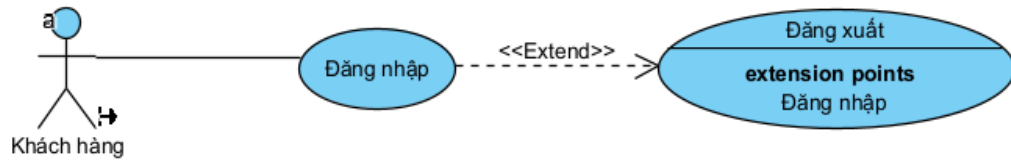
Hình 3. 1: Sơ đồ usecase tổng quát trang chủ



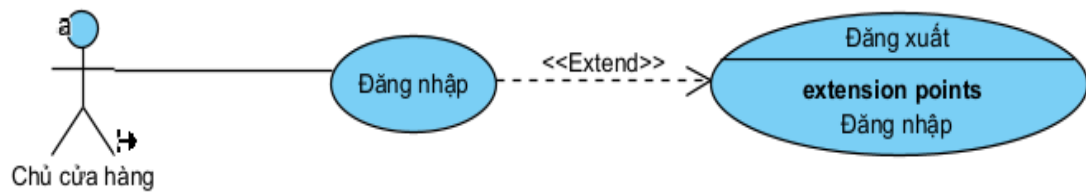
Hình 3. 2: Sơ đồ usecase tổng quát trang quản trị

b) Sơ đồ phân rã Usecase

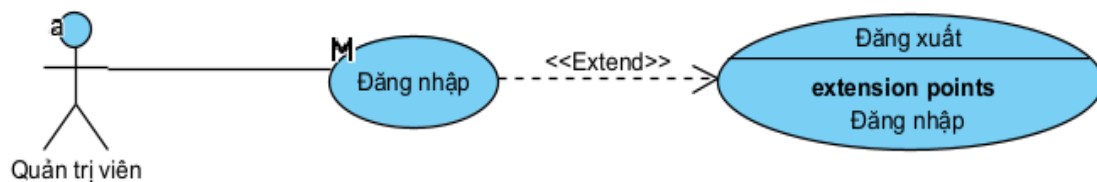
- Usecase đăng nhập, đăng xuất



Hình 3. 3: Sơ đồ usecase đăng nhập, đăng xuất của khách hàng



Hình 3. 4: Sơ đồ usecase đăng nhập, đăng xuất của chủ cửa hàng



Hình 3. 5: Sơ đồ usecase đăng nhập, đăng xuất của quản trị viên

- Usecase đăng ký tài khoản

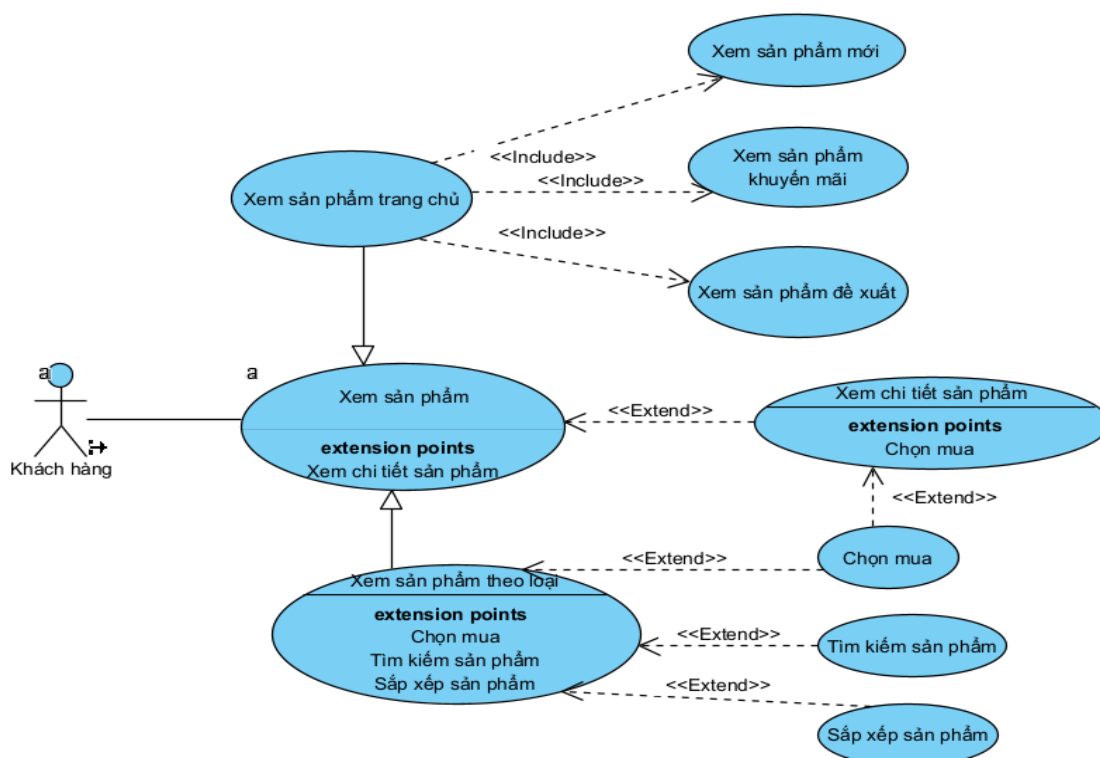


Hình 3. 6: Sơ đồ usecase đăng ký tài khoản của khách hàng



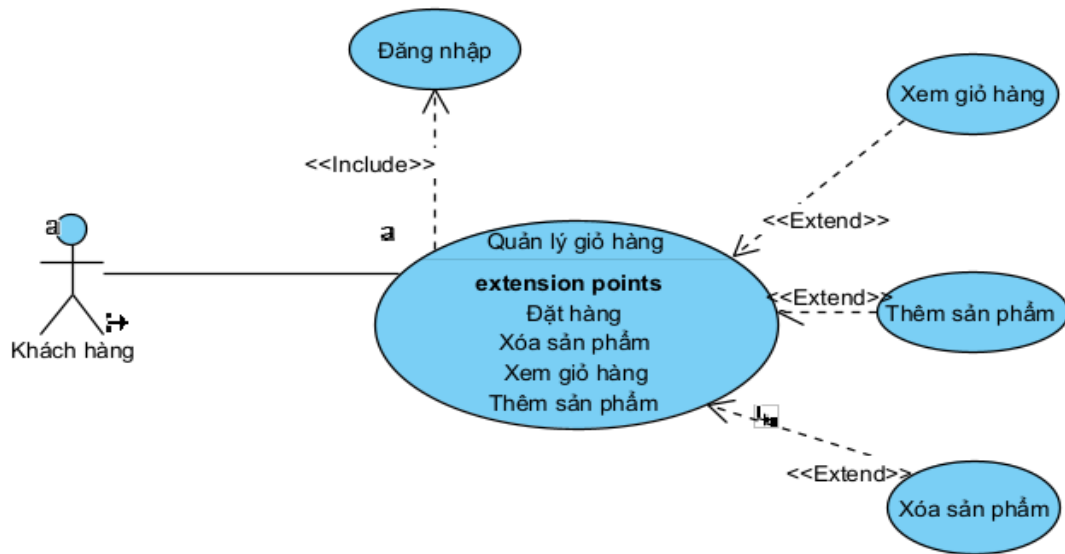
Hình 3. 7: Sơ đồ usecase đăng ký tài khoản của chủ cửa hàng

- Usecase xem sản phẩm



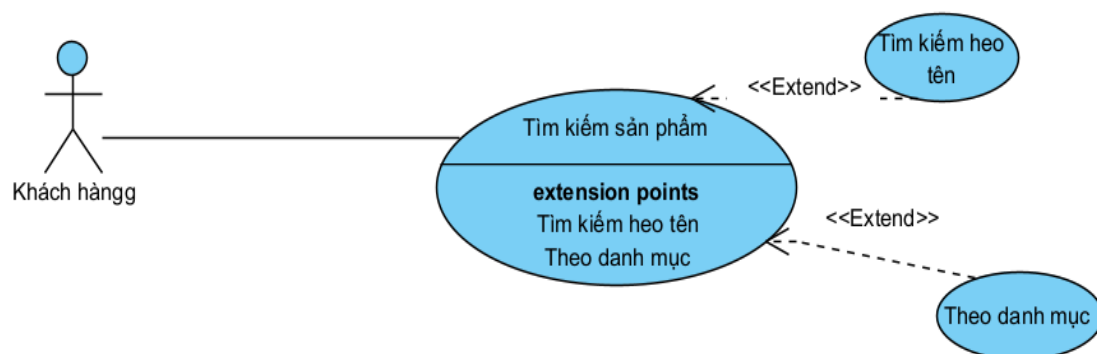
Hình 3. 8: Sơ đồ usecase xem sản phẩm

– Usecase quản lý giỏ hàng



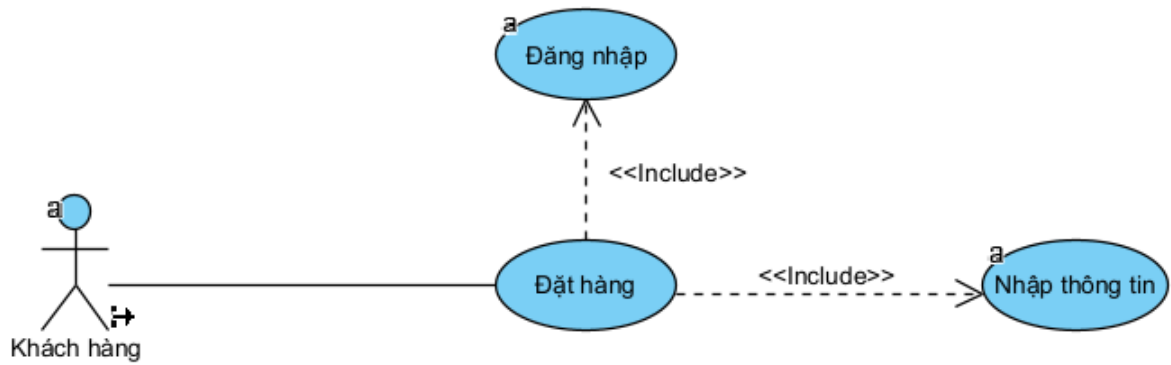
Hình 3. 9: Sơ đồ usecase quản lý giỏ hàng

– Usecase tìm kiếm sản phẩm



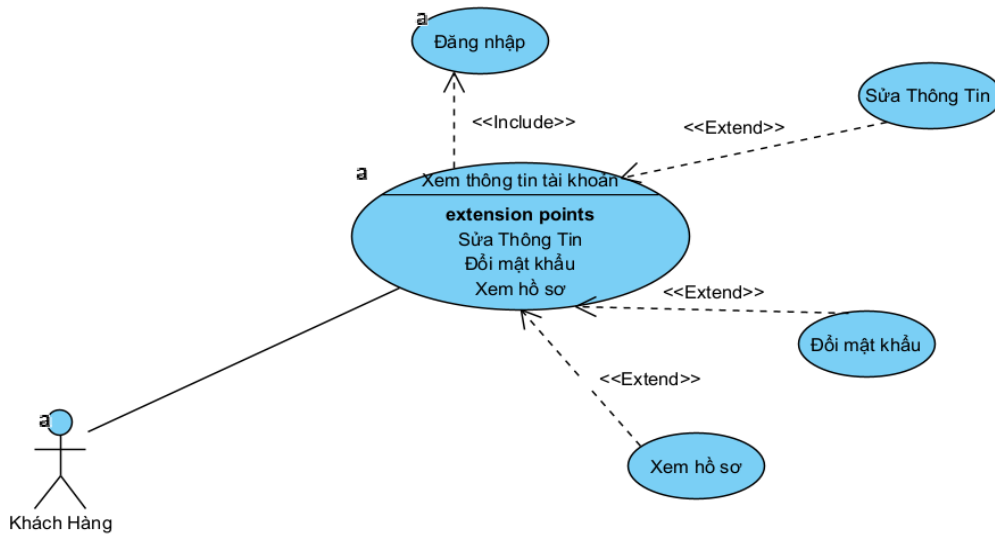
Hình 3. 10: Sơ đồ usecase tìm kiếm sản phẩm

- Usecase đặt hàng



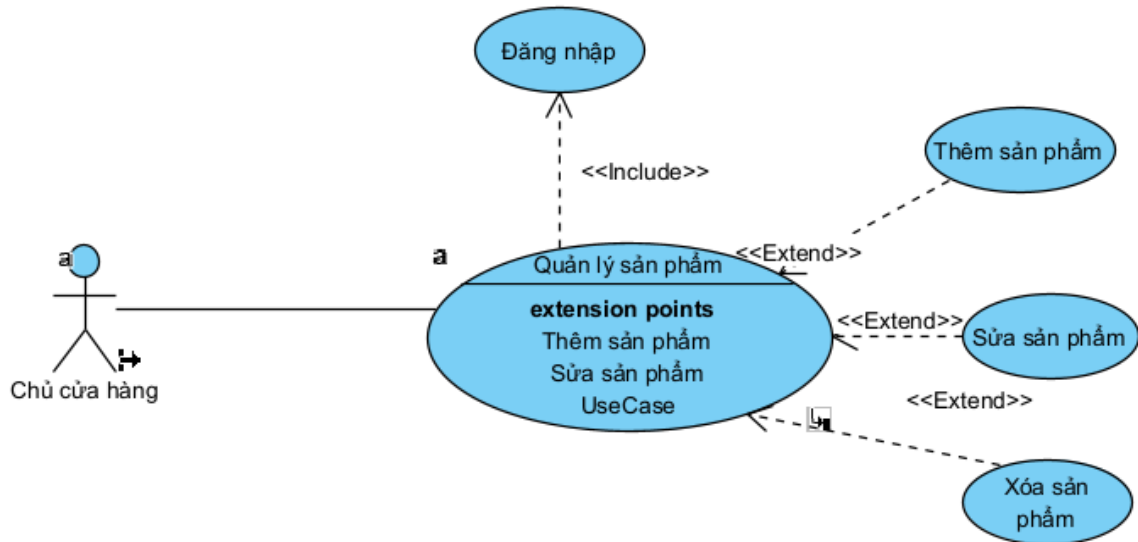
Hình 3. 11: Sơ đồ usecase đặt hàng

- Usecase quản lý thông tin tài khoản khách hàng



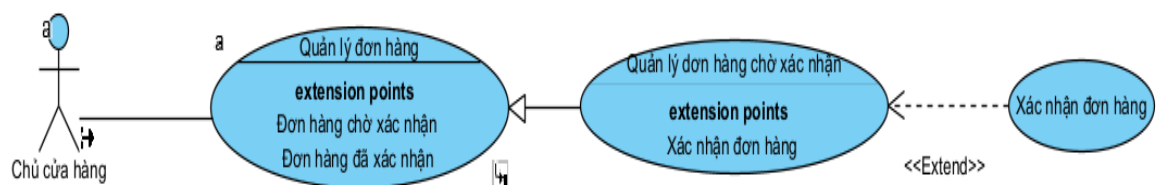
Hình 3. 12: Sơ đồ usecase quản lý thông tin khách hàng

– Usecase quản lý sản phẩm



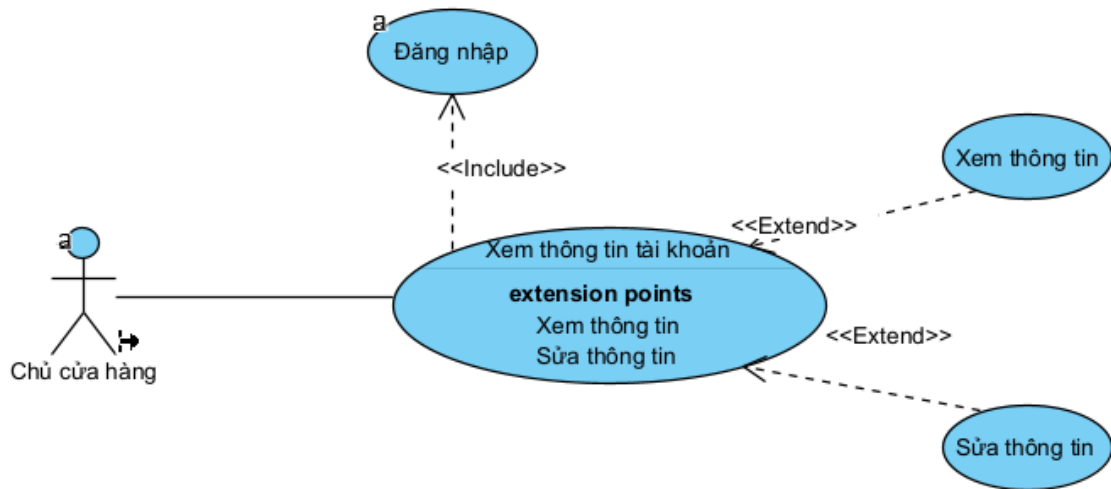
Hình 3. 13: Sơ đồ usecase quản lý sản phẩm

– Usecase quản lý đơn hàng



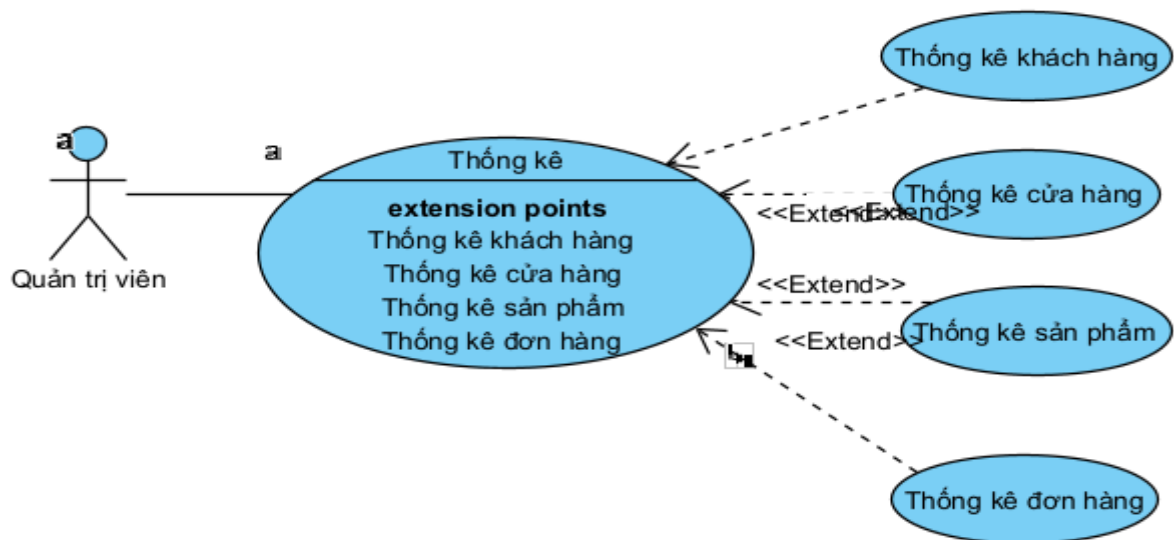
Hình 3. 14: Sơ đồ usecase quản lý đơn hàng

- Usecase quản lý thông tin tài khoản chủ cửa hàng



Hình 3. 15: Sơ đồ usecase quản lý thông tin chủ cửa hàng

- Usecase thống kê



Hình 3. 16: Sơ đồ usecase thống kê cửa hàng

c) Đặc tả Usecase

– Đặc tả Usecase đăng nhập

Bảng 3. 1: Đặc tả usecase đăng nhập

Tên Use Case	Đăng nhập tài khoản
Mô tả sơ lược	Người dùng, chủ cửa hàng đăng nhập vào hệ thống bán hàng.
Actor chính	Khách hàng, chủ cửa hàng
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Khách hàng, chủ cửa hàng chọn đăng nhập.
Điều kiện sau - Post-Condition(s)	Hệ thống thêm mới tài khoản.
Dòng sự kiện chính (main flow)	
Actor	System
1. Người sử dụng chọn chức năng “Đăng nhập”.	2. Hệ thống hiển thị trang đăng nhập.
3. Người sử dụng nhập thông tin vào form đăng nhập.	4. Hệ thống kiểm tra thông tin khách hàng, chủ cửa hàng và chuyển sang trang chủ.
Dòng sự kiện phụ (secondary flow)	
7. Nhập sai định dạng Email, password. 9. Nhập thiếu thông tin: email, username, password, confirm password. 11. Nhập password và confirm password không trùng nhau.	8. Hệ thống yêu cầu nhập lại. 10. Hệ thống yêu cầu nhập đầy đủ. 12. Hệ thống yêu cầu nhập lại
13. Thông tin tài khoản không tồn tại trong CSDL.	14. Hệ thống thông báo tài khoản chưa được đăng ký

– Đặc tả Usecase đăng xuất

Bảng 3. 2: Đặc tả usecase đăng xuất

Tên Use Case	Đăng xuất tài khoản
Mô tả sơ lược	Người dùng, chủ cửa hàng đăng xuất khỏi hệ thống bán hàng.
Actor chính	Khách hàng, chủ cửa hàng

Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Khách hàng, chủ cửa hàng chọn đăng xuất.
Dòng sự kiện chính (main flow)	
Actor	System
1. Người sử dụng chọn chức năng “Đăng xuất”.	2. Hệ thống hiển thị xác nhận đăng xuất.
3. Người sử dụng xác nhận.	4. Hệ thống đăng xuất khỏi hệ thống.

– Đặc tả Usecase đăng ký

Bảng 3. 3: Đặc tả usecase đăng ký

Tên Use Case	Đăng ký tài khoản
Mô tả sơ lược	Người dùng đăng ký thành viên của hệ thống bán hàng.
Actor chính	Khách hàng, chủ cửa hàng
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Người dùng chọn đăng ký tài khoản.
Điều kiện sau - Post-Condition(s)	Hệ thống thêm mới tài khoản.
Dòng sự kiện chính (main flow)	
Actor	System
1. Người sử dụng chọn chức năng “Đăng ký”.	2. Hệ thống hiển thị trang đăng ký.
3. Người sử dụng nhập thông tin vào form đăng ký tài khoản.	4. Hệ thống lưu thông tin người dùng vào CSDL và chuyển sang trang chủ.
Dòng sự kiện phụ (secondary flow)	
7. Nhập sai định dạng Email, password. 9. Nhập thiếu thông tin: email, username, password, confirm password. 11. Nhập password và confirm password không trùng nhau.	8. Hệ thống yêu cầu nhập lại. 10. Hệ thống yêu cầu nhập đầy đủ. 12. Hệ thống yêu cầu nhập lại
13. Thông tin tài khoản đã tồn tại trong CSDL.	14. Hệ thống thông báo tài khoản đã tồn tại.

– Đặc tả Usecase xem sản phẩm

Bảng 3. 4: Đặc tả usecase xem sản phẩm

Tên Use Case	Xem sản phẩm
Mô tả sơ lược	Xem sản phẩm cho phép người dùng xem sản phẩm trực tuyến và chọn mua sản phẩm.
Actor chính	Khách hàng (Client)
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Truy cập website.
Điều kiện sau - Post-Condition(s)	Hệ thống hiển thị danh sách sản phẩm cho khách hàng.
Dòng sự kiện chính (main flow)	
Actor	System
1. Khách hàng xem sản phẩm ở trang chủ.	2. Hệ thống hiển thị trang chủ.
3. Khách hàng xem các sản phẩm theo danh mục.	4. Hệ thống hiển thị trang danh mục sản phẩm.
5. Khách hàng tìm kiếm sản phẩm.	6. Hệ thống hiển thị danh sách sản phẩm.

– Đặc tả Usecase quản lý giỏ hàng

Bảng 3. 5: Đặc tả usecase quản lý giỏ hàng

Tên Use Case	Quản lý giỏ hàng
Mô tả sơ lược	Giỏ hàng cho phép người dùng thuận lợi cho việc mua sản phẩm. Tính năng này mang đến trải nghiệm tốt hơn với khách hàng.
Actor chính	Khách hàng (Client)
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Truy cập website.
Điều kiện sau - Post-Condition(s)	Hệ thống hiển thị danh sách sản phẩm cho khách hàng, khách hàng có thể chọn sản phẩm để thêm vào giỏ.
Dòng sự kiện chính (main flow)	
Actor	System
1. Khách hàng xem sản phẩm ở trang chủ.	2. Hệ thống hiển thị trang chủ.
3. Khách hàng chọn sản phẩm mình thích và thêm vào giỏ hàng.	4. Hệ thống lưu vào localStorage và hiển thị lên giỏ hàng, nếu khách hàng đã đăng nhập thì thêm sản phẩm vào giỏ hàng trong CSDL.
5. Khách hàng xóa và sửa số lượng sản phẩm trong giỏ hàng.	6. Hệ thống hiển thị danh sách sản phẩm trong giỏ hàng, khách hàng có thể xóa và thay đổi số lượng sản phẩm và cập nhập vào CSDL.

– Đặc tả Usecase quản lý đơn hàng

Bảng 3. 6: Đặc tả usecase quản lý đơn hàng

Tên Use Case	Đơn hàng
Mô tả sơ lược	Đơn hàng cho phép khách hàng đặt đơn trực tuyến và theo dõi tình trạng đơn hàng trên website.
Actor chính	Khách hàng (Client)
Actor phụ	Không

Điều kiện trước - Pre-Condition(s)	Truy cập website, đăng nhập và mua hàng.
Điều kiện sau - Post-Condition(s)	Hệ thống hiển thị danh sách sản phẩm khách hàng mua hàng và hiển thị đơn hàng khi khách hàng mua hàng thành công.
Dòng sự kiện chính (main flow)	
Actor	System
1. Khách hàng xem sản phẩm ở trang chủ thêm vào giỏ hàng hoặc mua trực tiếp.	2. Hệ thống hiển thị trang xác nhận đặt đơn và lưu vào CSDL.
3. Khách hàng chọn xem đơn hàng đã mua.	4. Hệ thống hiển thị danh sách đơn hàng đã mua.

– Đặc tả Usecase quản lý sản phẩm

Bảng 3. 7: Đặc tả usecase quản lý sản phẩm

Tên Use Case	Quản lý sản phẩm
Mô tả sơ lược	Chức năng quản lý sản phẩm là chức năng cho phép chủ cửa hàng có thể tạo ra những sản phẩm mới.
Actor chính	Chủ cửa hàng
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Chủ cửa hàng thêm, sửa, xoá và thực thi thành công.
Điều kiện sau - Post-Condition(s)	Hệ thống cập nhật sản phẩm sau khi thực hiện thêm, sửa, xoá thành công. Hệ thống cập nhật trạng thái sản phẩm sau khi thực thi thành công.
Dòng sự kiện chính (main flow)	
Actor	System
1. Chủ cửa hàng nhập từ khoá để tìm kiếm tên sản phẩm theo ý muốn.	2. Hệ thống hiển thị các sản phẩm chứa từ khoá đã tìm kiếm.
3. Chủ cửa hàng chọn thêm mới.	4. Hệ thống hiển thị giao diện thêm mới sản phẩm.
5. Chủ cửa hàng chọn một sản phẩm muốn chỉnh sửa sau đó chọn chỉnh sửa.	6. Hệ thống hiển thị giao diện chỉnh sửa của sản phẩm đó.

7. Chủ cửa hàng chọn một hoặc nhiều sản phẩm muốn xóa sau đó chọn xóa.	8. Hệ thống hiển thị thông báo “Bạn có chắc chắn muốn xóa sản phẩm này”.
--	--

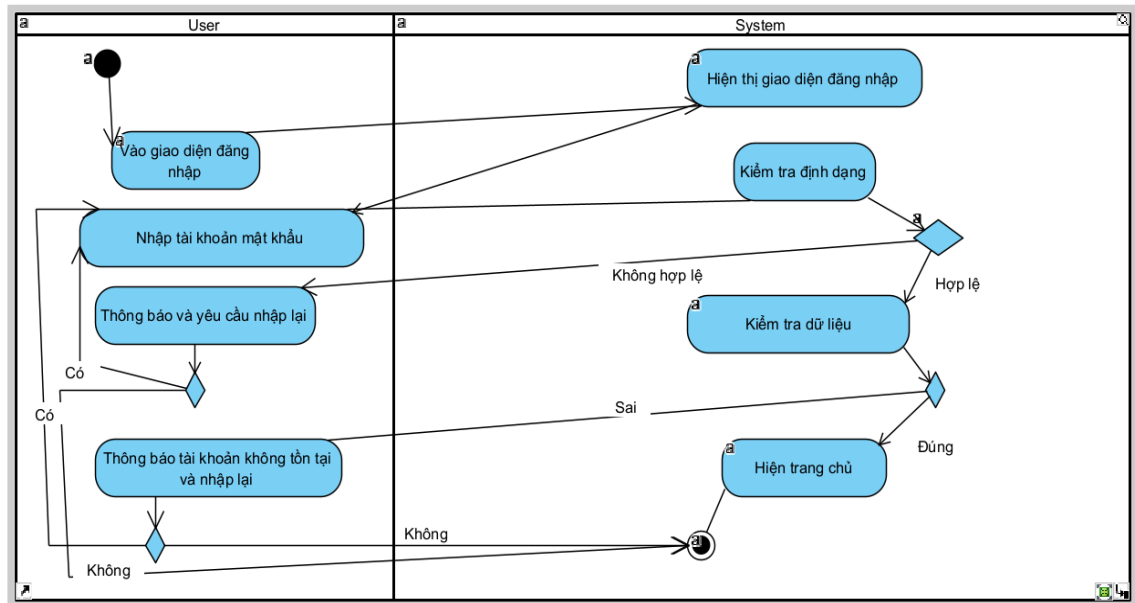
– Đặc tả Usecase quản lý tình trạng đơn hàng

Bảng 3. 8: Đặc tả usecase quản lý tình trạng đơn hàng

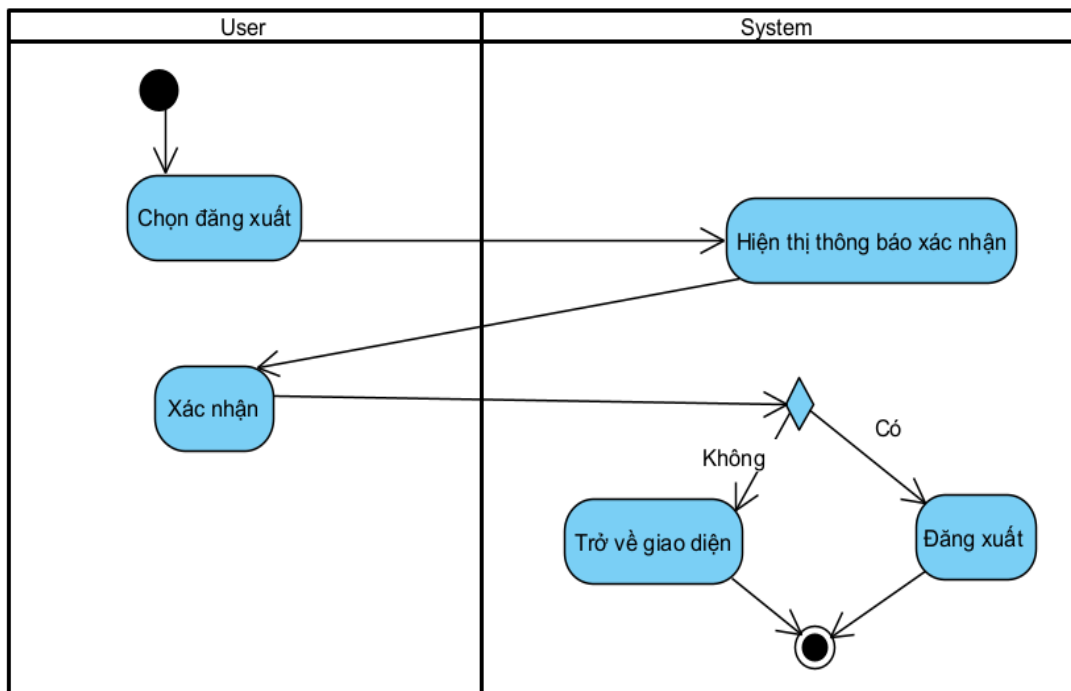
Tên Use Case	Quản lý danh sách đơn hàng
Mô tả sơ lược	Chức năng quản lý danh sách đơn hàng là chức năng cho phép chủ cửa hàng có thể thay đổi trạng thái đơn hàng .
Actor chính	Chủ cửa hàng
Actor phụ	Không
Điều kiện trước - Pre-Condition(s)	Chủ cửa hàng thay đổi trạng thái và thực thi thành công.
Điều kiện sau - Post-Condition(s)	Hệ thống cập nhật danh sách đơn hàng sau khi thực hiện thành công
Dòng sự kiện chính (main flow)	
Actor	System
1. Chủ cửa hàng nhập từ khoá để tìm kiếm tên người dùng có đơn hàng theo ý muốn.	2. Hệ thống hiển thị các đơn hàng chứa từ khoá đã tìm kiếm.
3. Chủ cửa hàng chọn thay đổi trạng thái đơn hàng.	4. Hệ thống hiển thị thông báo thành công.

3.2.3. Biểu đồ hoạt động

a) Biểu đồ hoạt động đăng nhập, đăng xuất

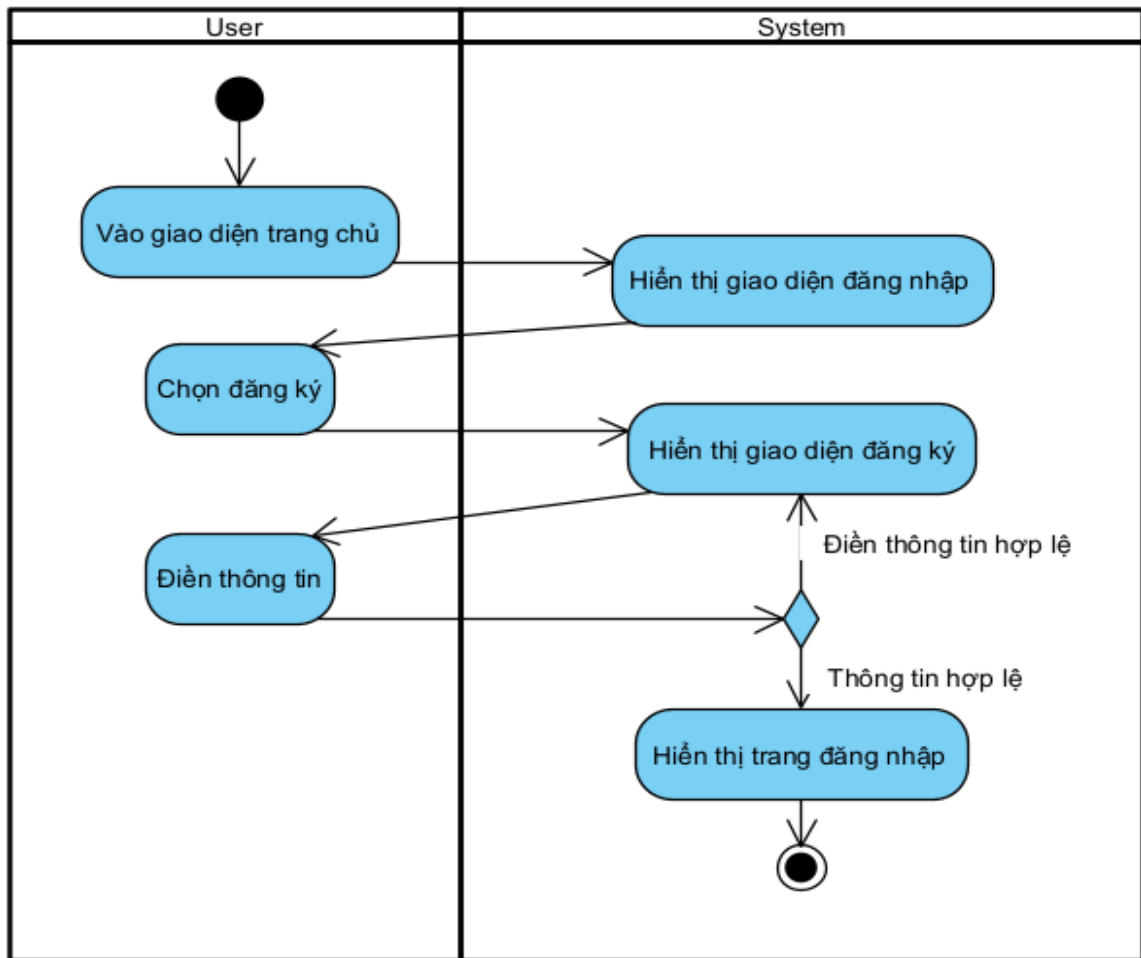


Hình 3. 17: Sơ đồ hoạt động đăng nhập



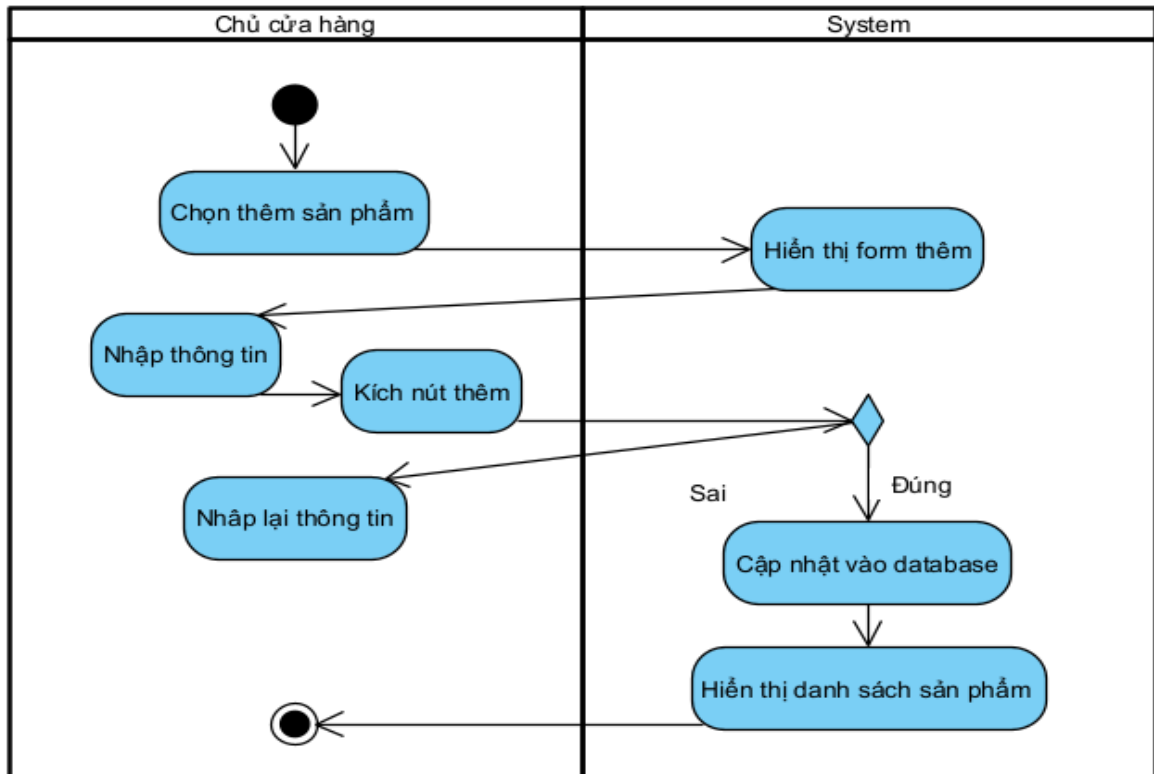
Hình 3. 18: Sơ đồ hoạt động đăng xuất

b) Biểu đồ hoạt động đăng ký tài khoản

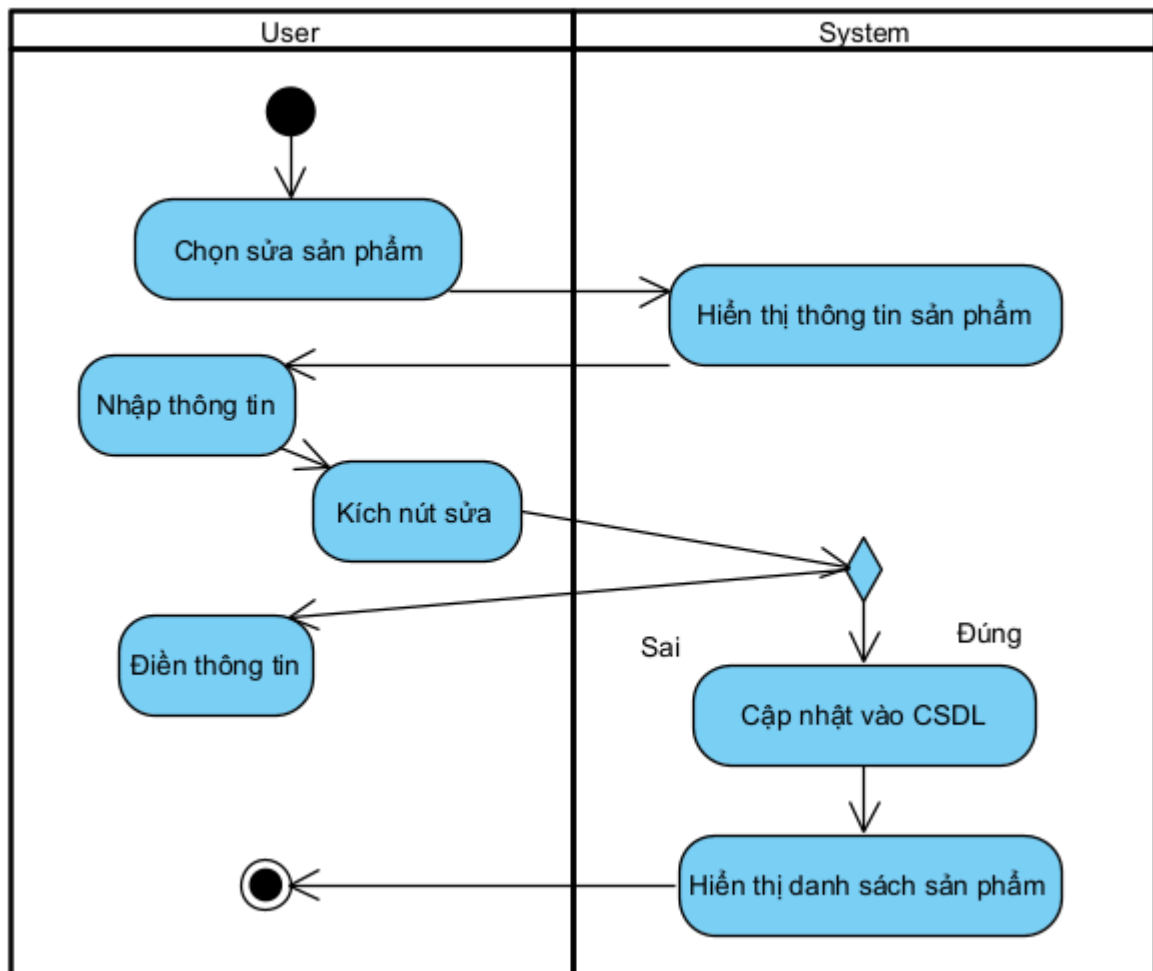


Hình 3. 19: Sơ đồ hoạt động đăng ký

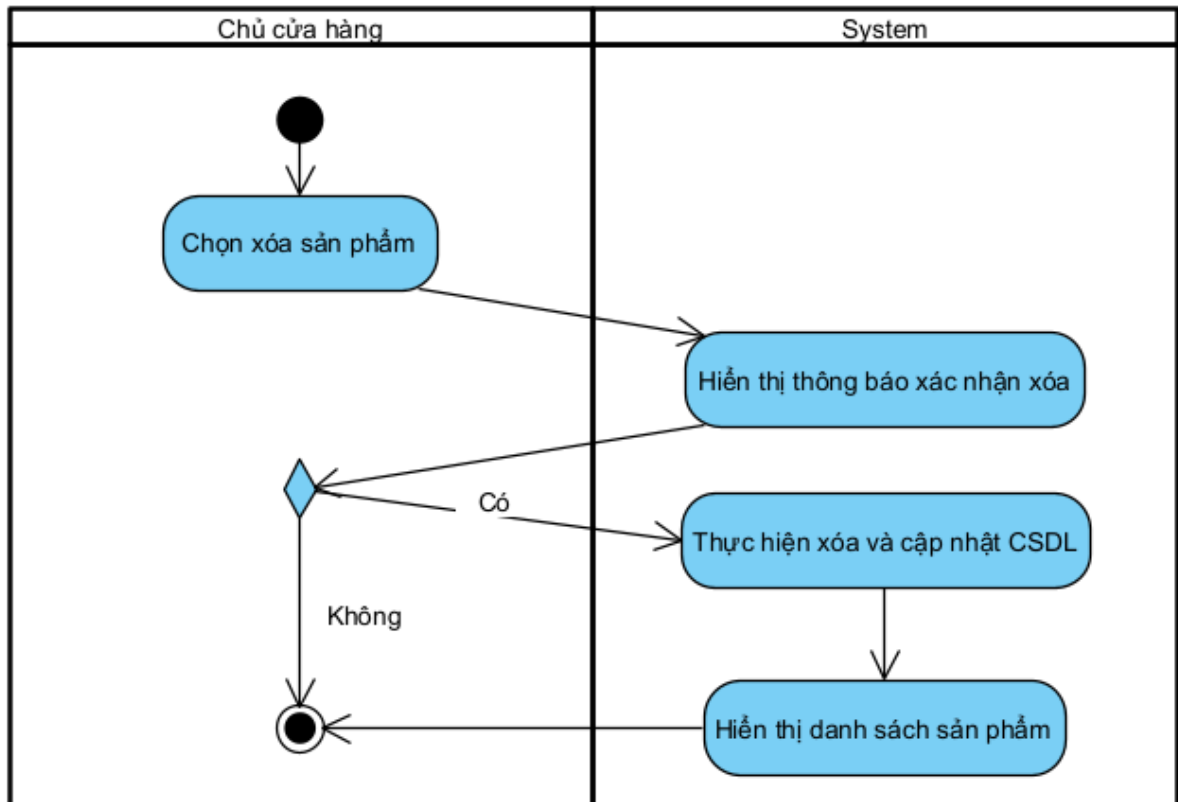
c) Biểu đồ hoạt động quản lý sản phẩm



Hình 3. 20: Sơ đồ hoạt động thêm sản phẩm

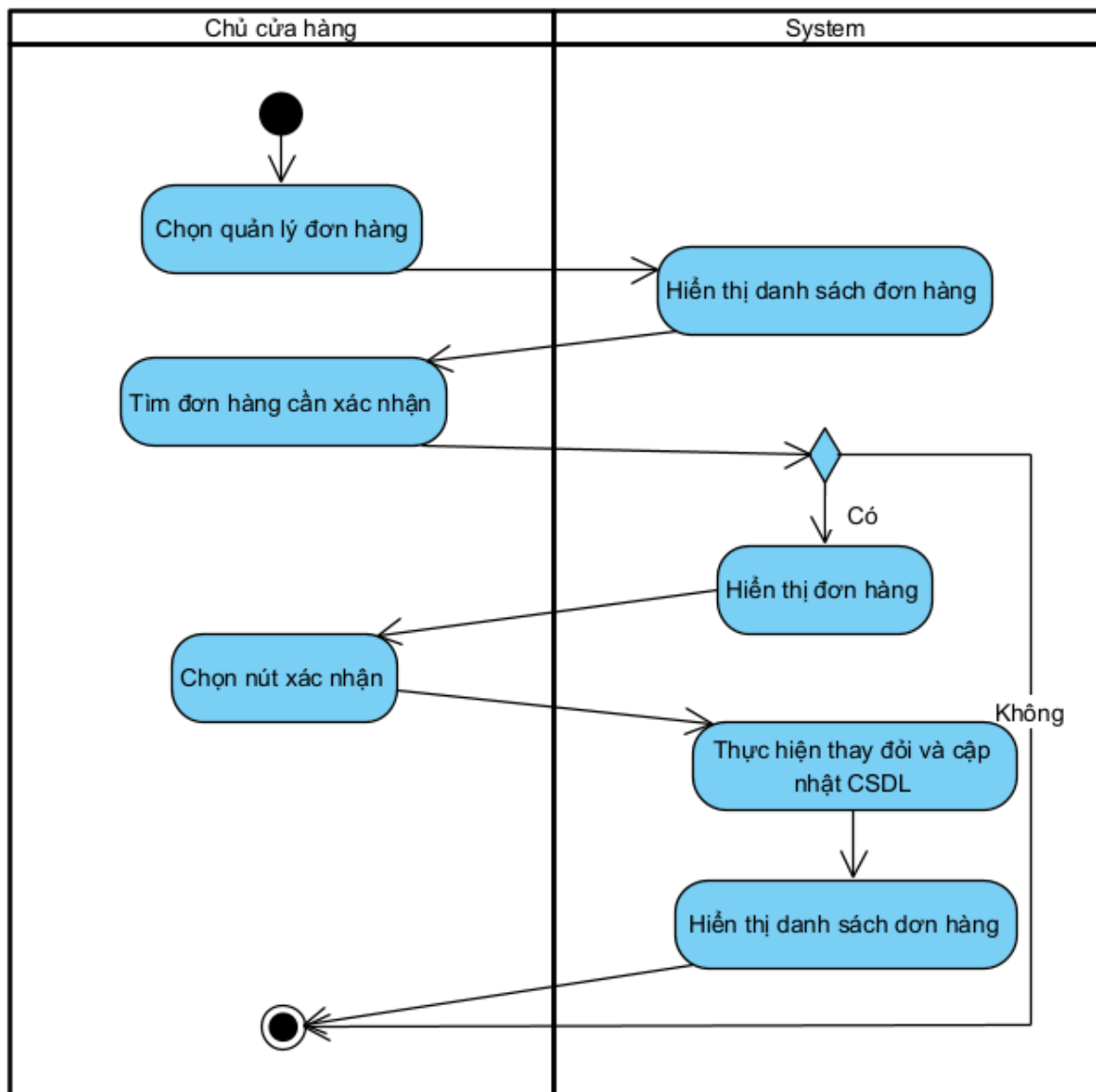


Hình 3. 21: Sơ đồ hoạt động sửa sản phẩm



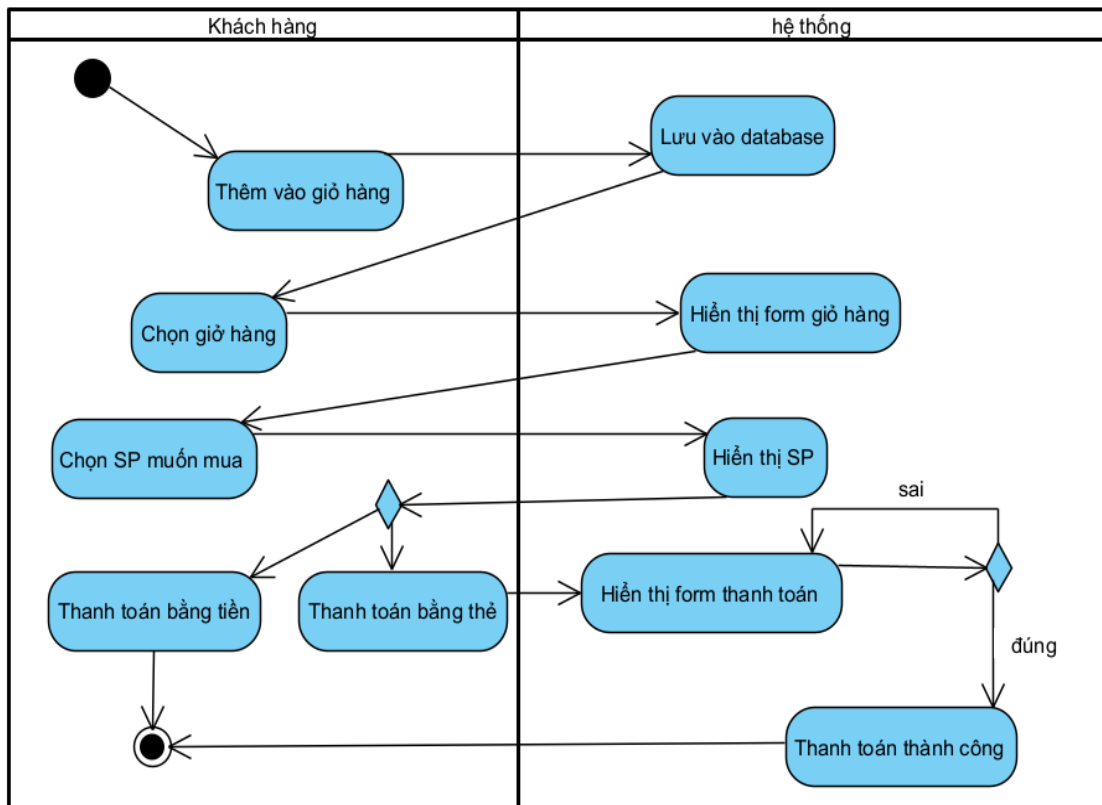
Hình 3. 22: Sơ đồ hoạt động xóa sản phẩm

d) Biểu đồ hoạt động quản lý đơn hàng



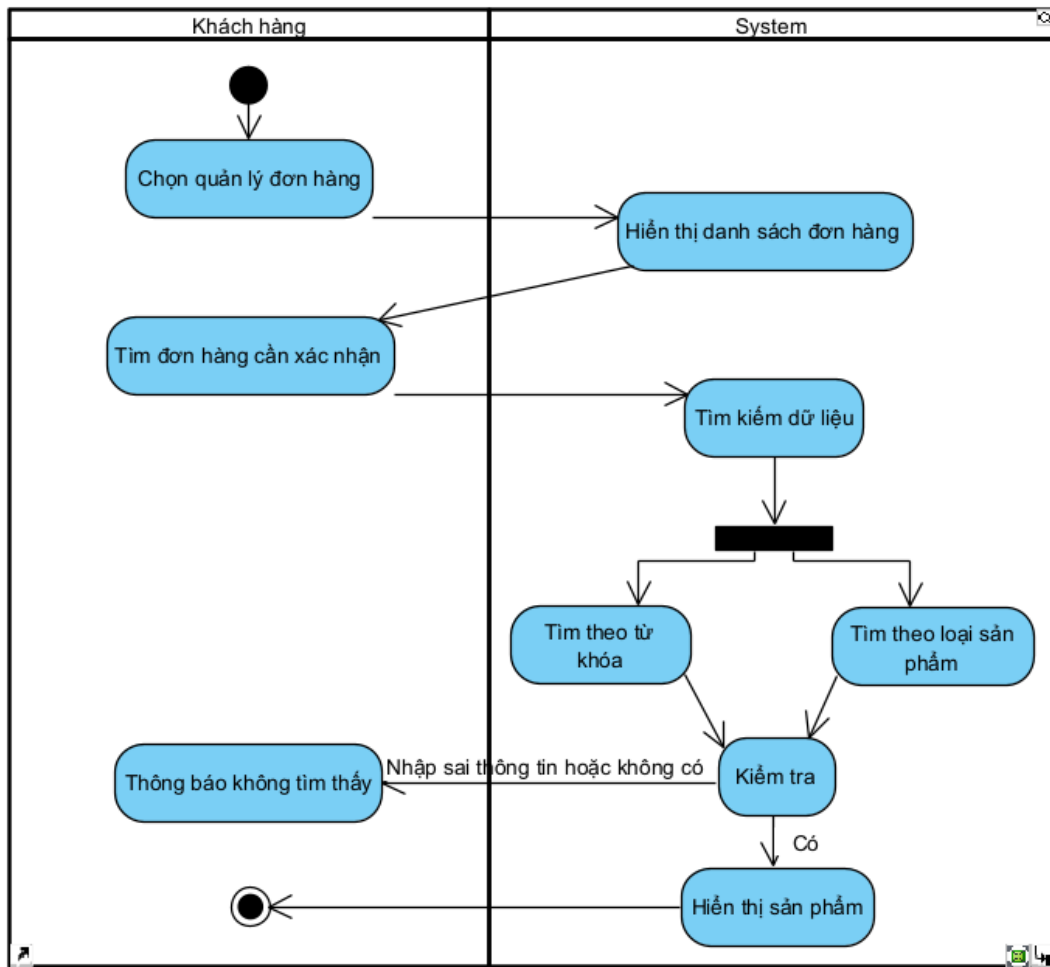
Hình 3. 23: Sơ đồ hoạt động đơn hàng

e) Biểu đồ hoạt động đặt hàng



Hình 3. 24: Sơ đồ hoạt động đặt hàng

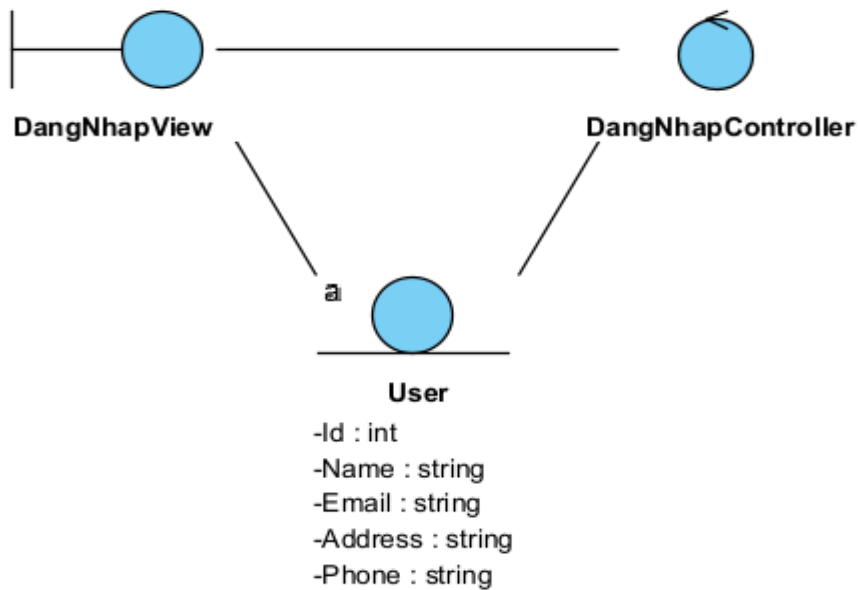
f) Biểu đồ hoạt động tìm kiếm sản phẩm



Hình 3. 25: Sơ đồ hoạt động tìm kiếm sản phẩm

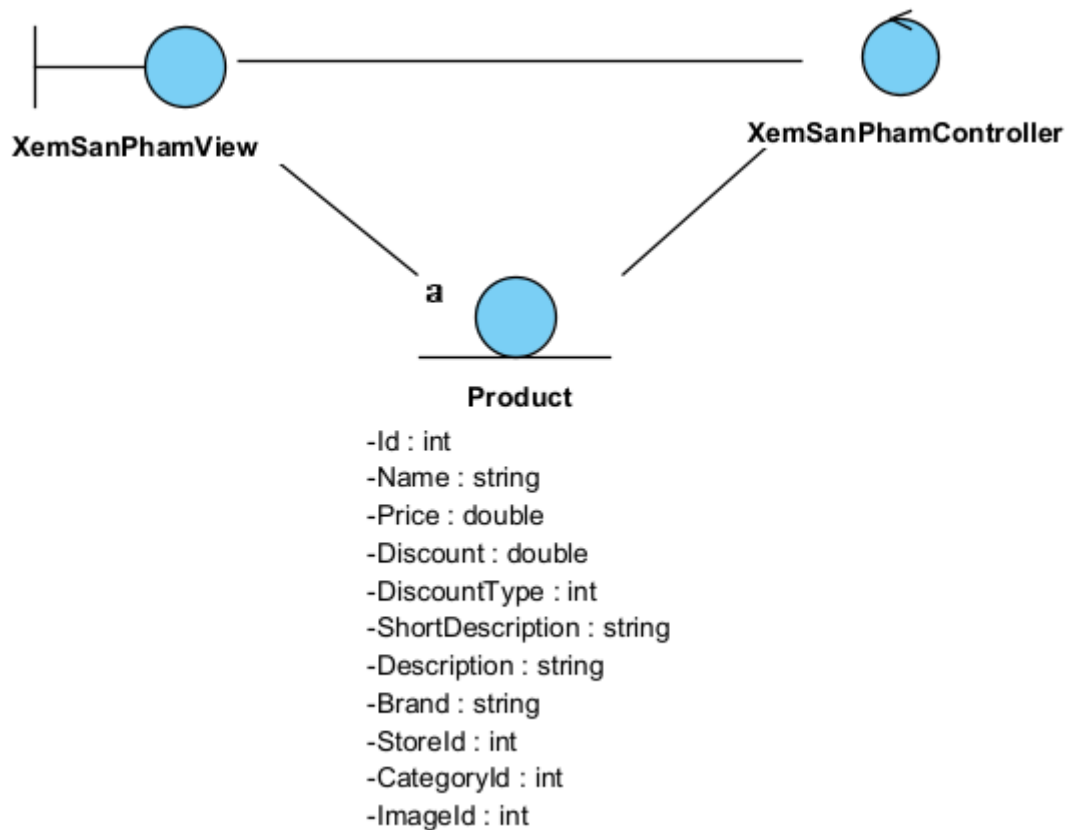
3.2.4. Biểu đồ lớp phân tích các ca sử dụng

a. Biểu đồ lớp đăng nhập



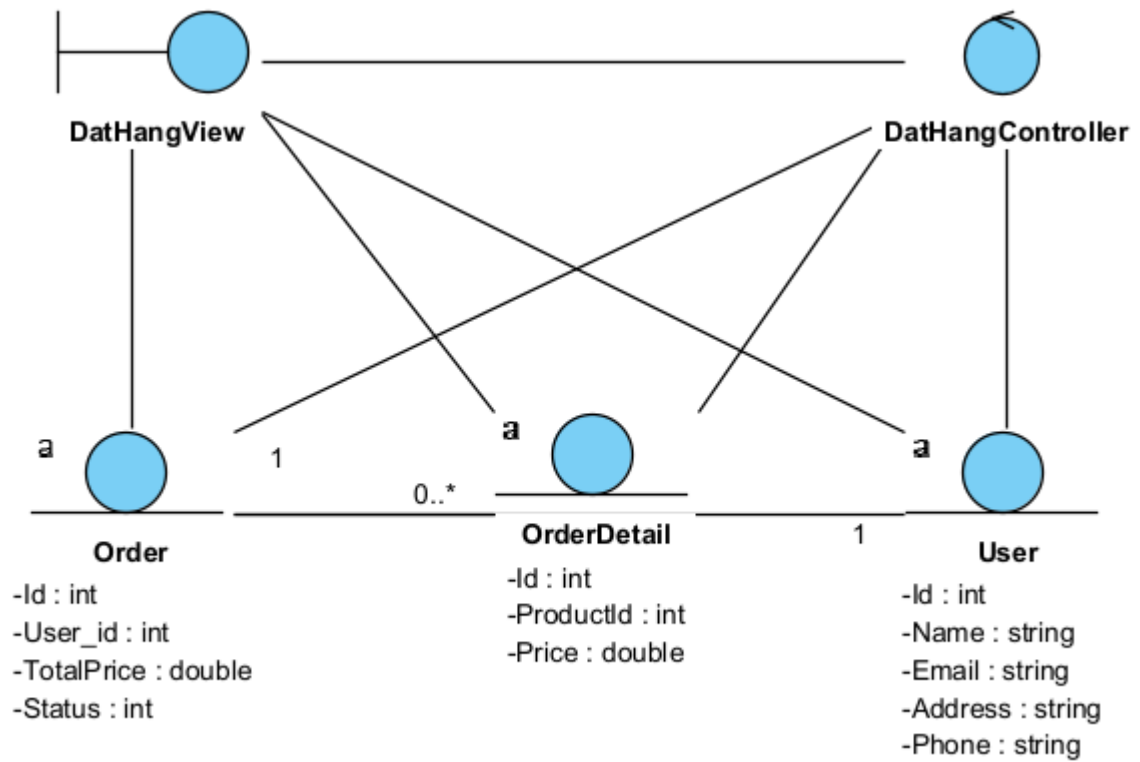
Hình 3. 26: Biểu đồ lớp đăng nhập

b. Biểu đồ lớp xem sản phẩm



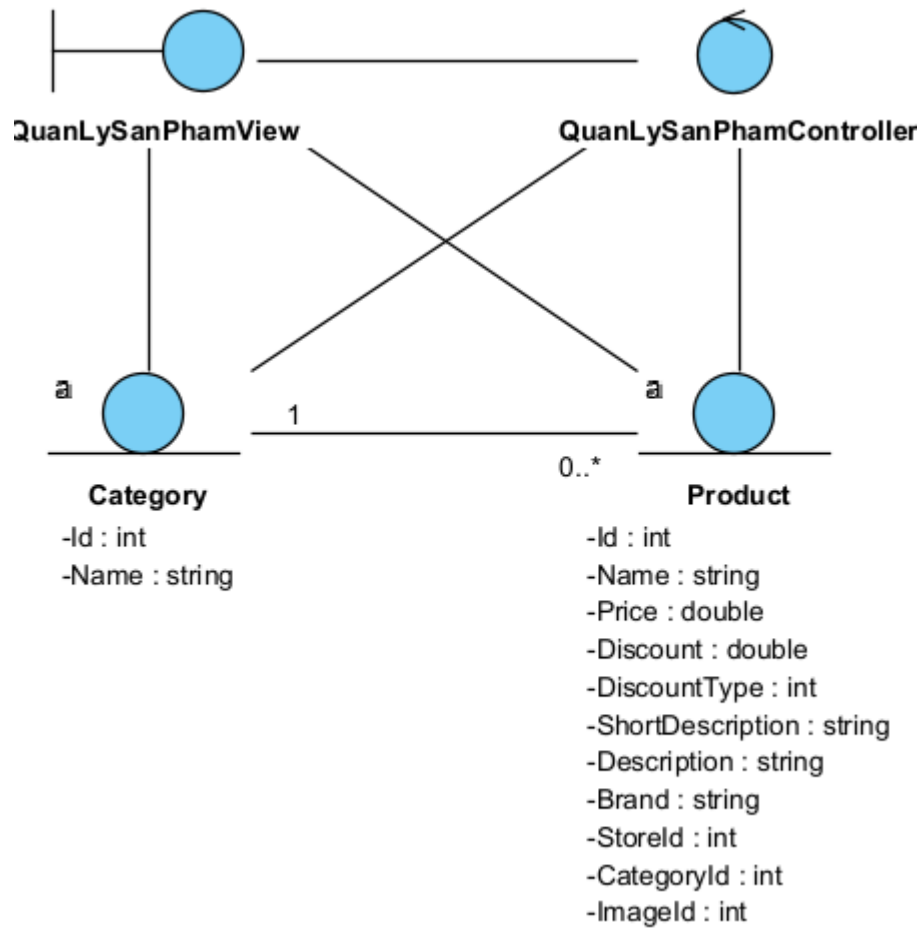
Hình 3. 27: Biểu đồ lớp xem sản phẩm

c. Biểu đồ lớp đặt hàng



Hình 3. 28: Biểu đồ lớp đặt hàng

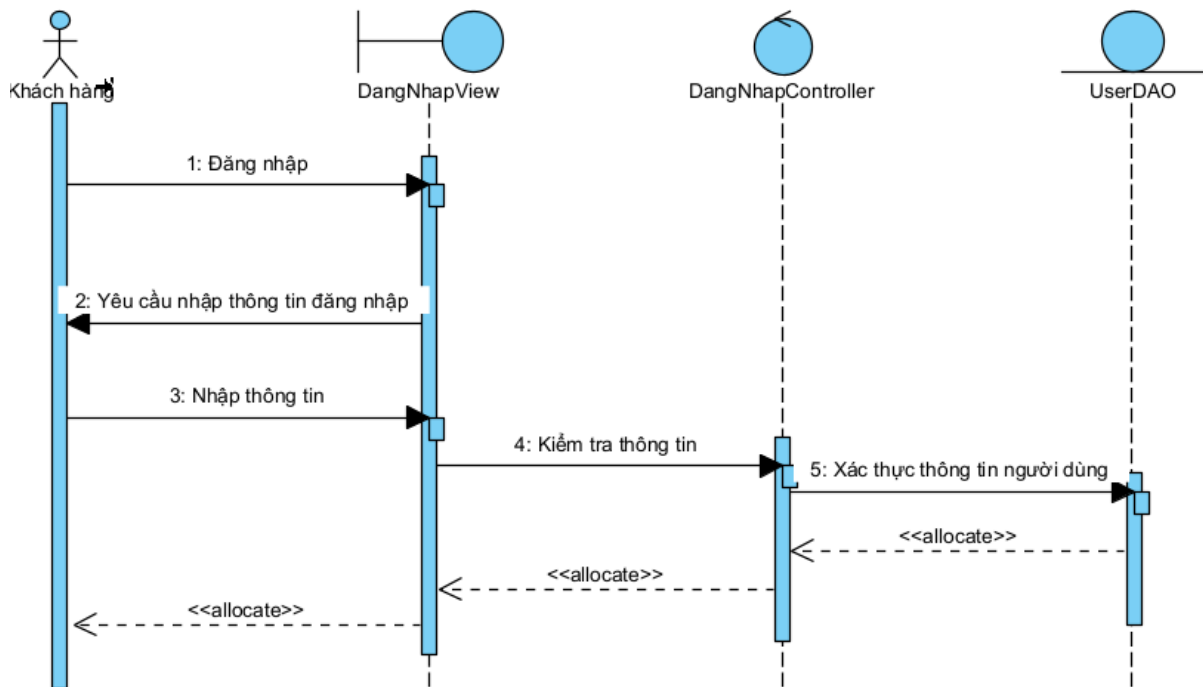
d. Biểu đồ lớp quản lý sản phẩm



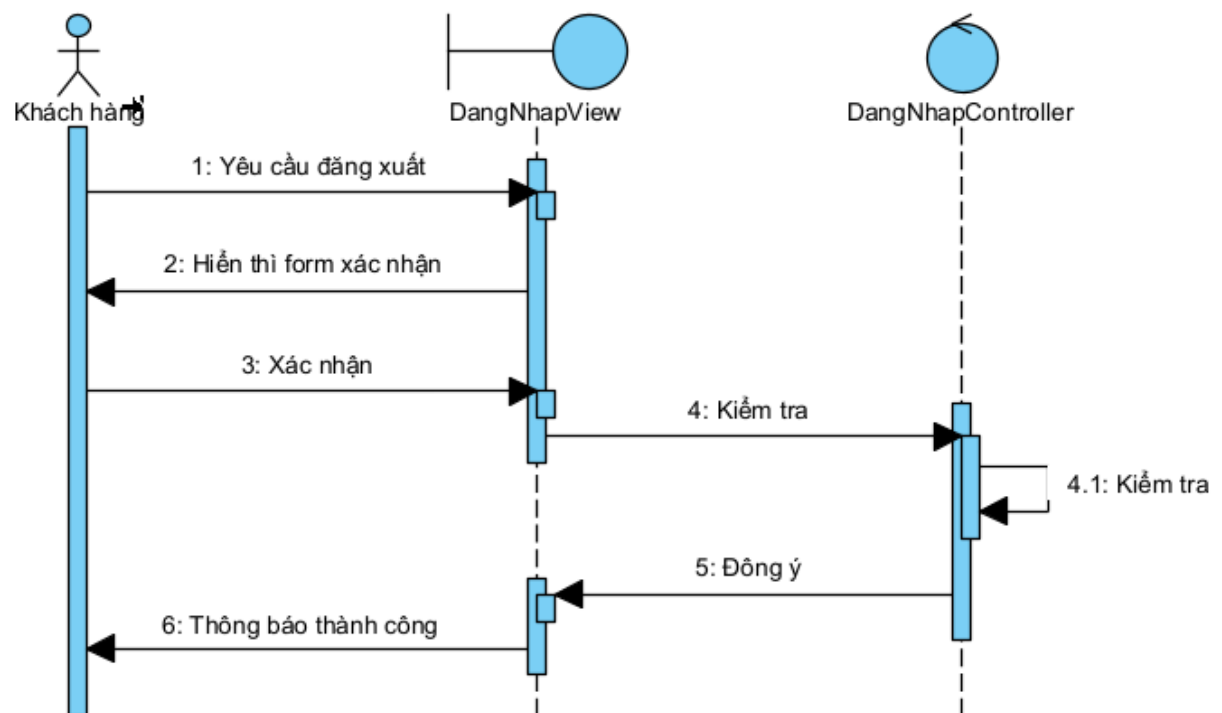
Hình 3. 29: Biểu đồ lớp quản lý sản phẩm

3.2.5. Biểu đồ tuần tự

a. Biểu đồ tuần tự đăng nhập và đăng xuất

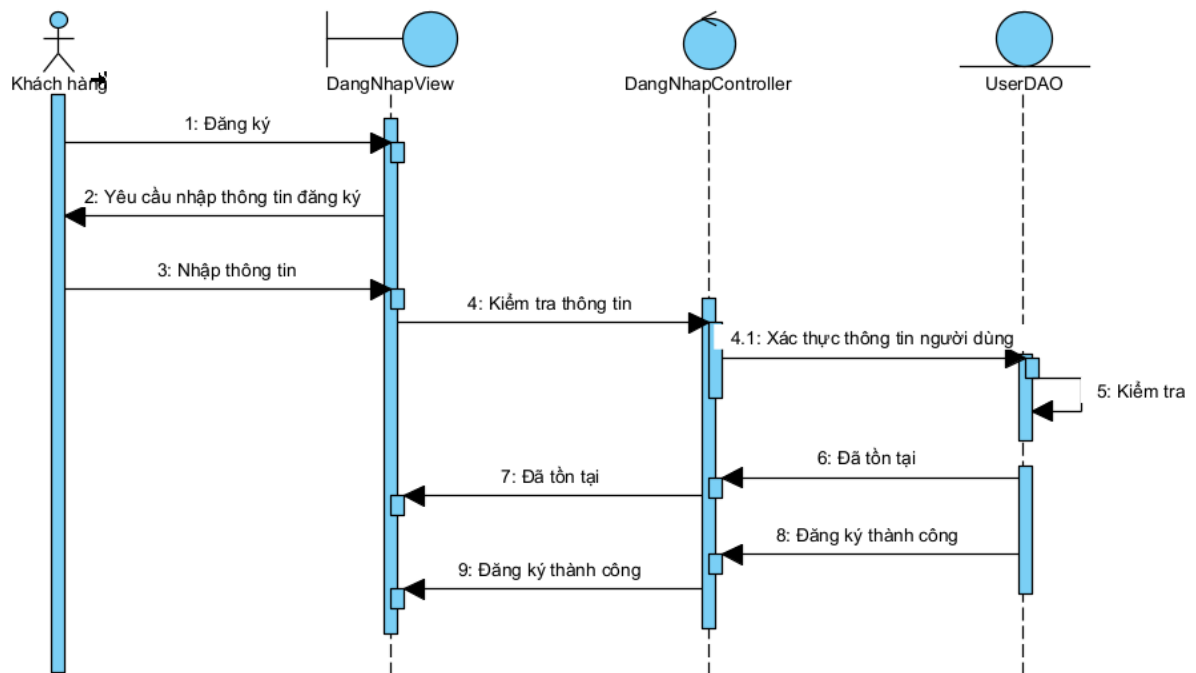


Hình 3. 30: Sơ đồ tuần tự đăng nhập



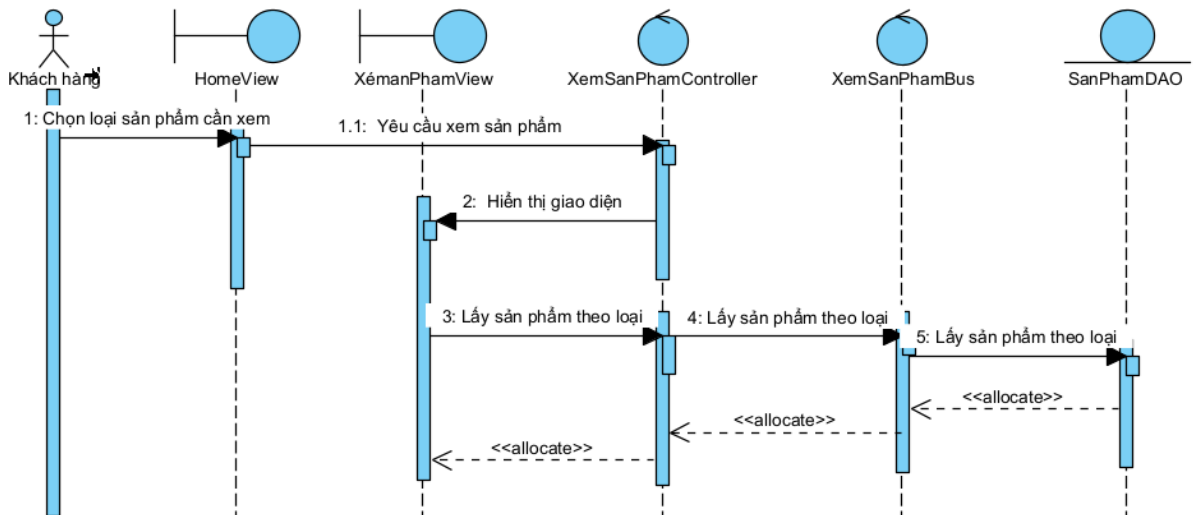
Hình 3. 31: Sơ đồ tuần tự đăng xuất

b. Biểu đồ tuần tự đăng ký tài khoản



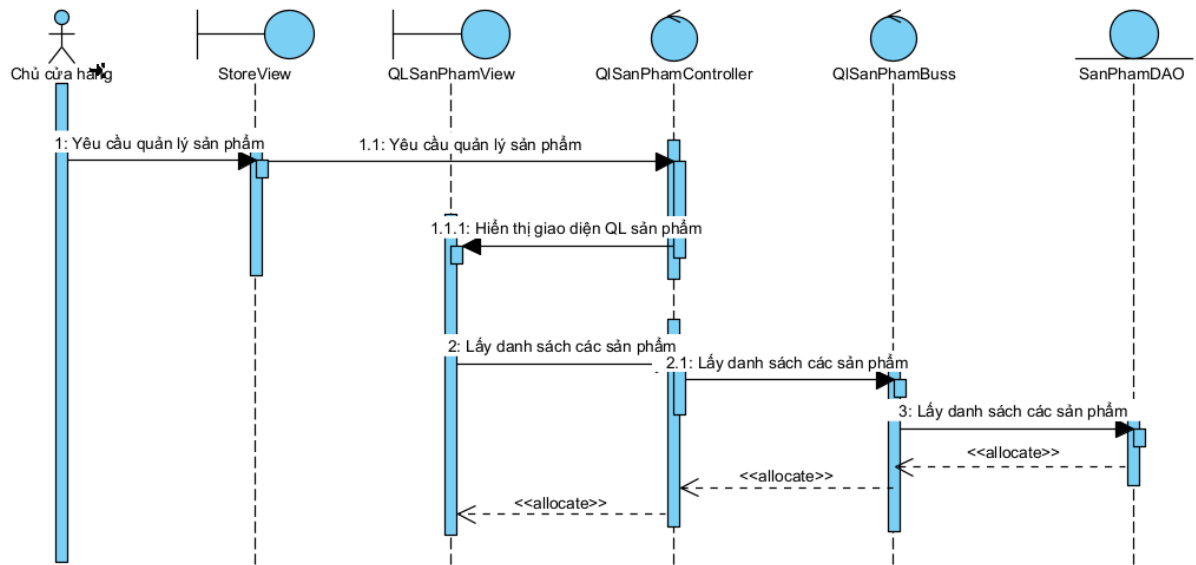
Hình 3. 32: Sơ đồ tuần tự đăng ký

c. Biểu đồ tuần tự xem sản phẩm



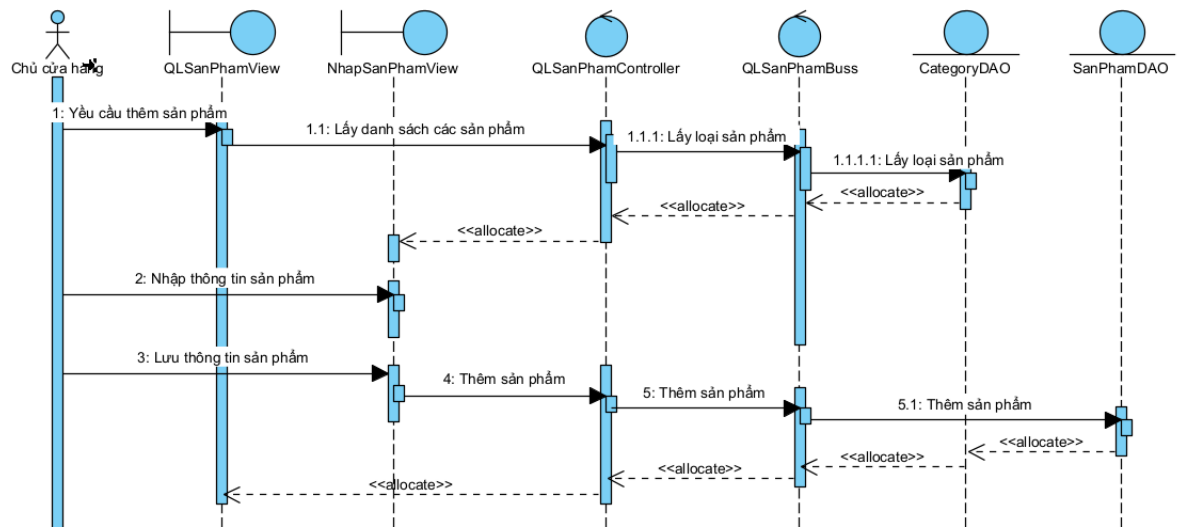
Hình 3. 33: Sơ đồ tuần tự xem sản phẩm

d. Biểu đồ tuần tự quản lý sản phẩm



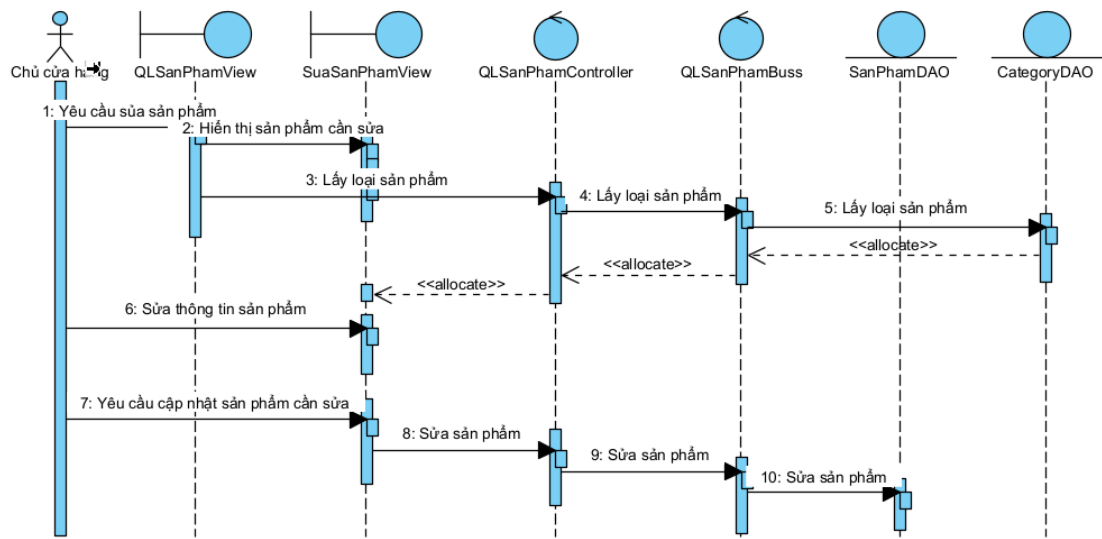
Hình 3. 34: Sơ đồ tuần tự quản lý sản phẩm

e. Biểu đồ tuần tự thêm sản phẩm



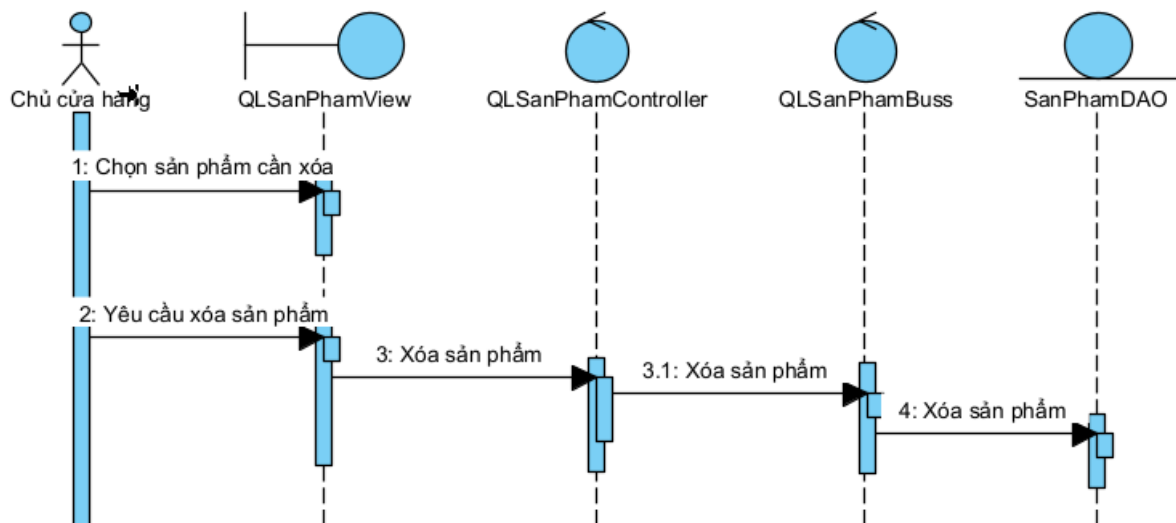
Hình 3. 35: Sơ đồ tuần tự thêm sản phẩm

f. Biểu đồ tuần tự sửa sản phẩm



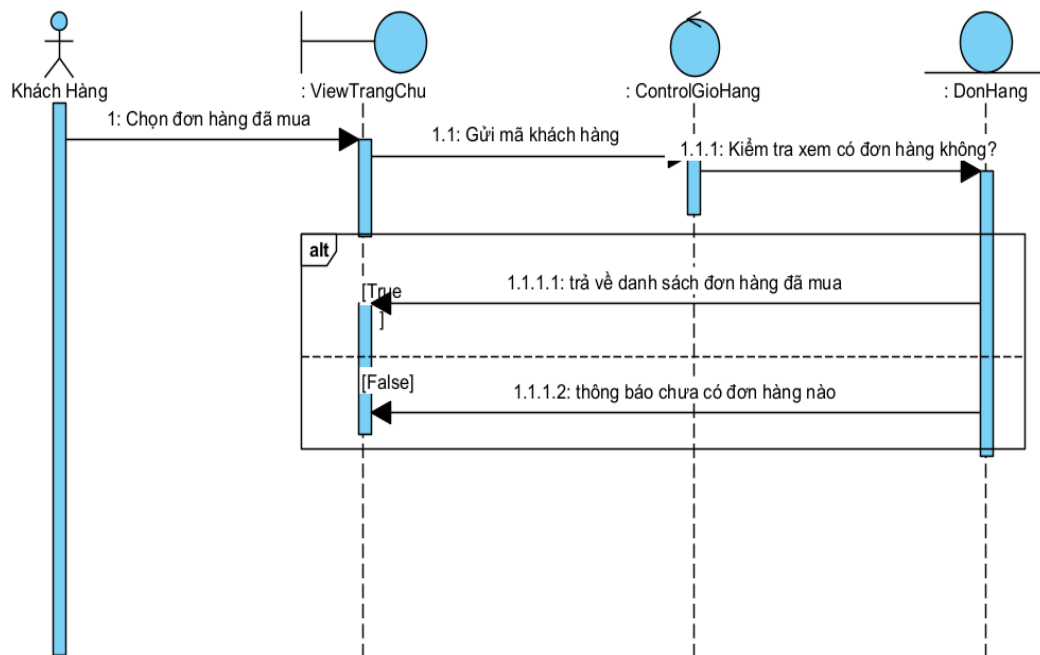
Hình 3. 36: Sơ đồ tuần tự sửa sản phẩm

g. Biểu đồ tuần tự xóa sản phẩm



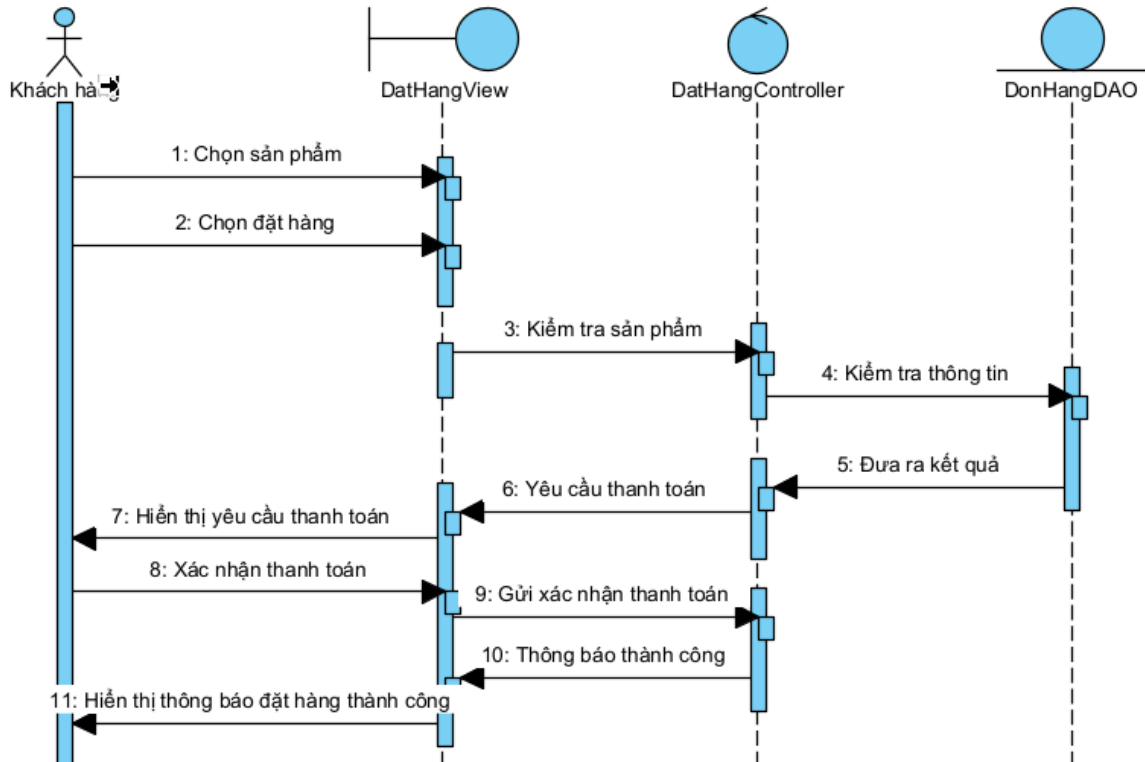
Hình 3. 37: Sơ đồ tuần tự xóa sản phẩm

h. Biểu đồ tuần tự xem đơn hàng của khách hàng



Hình 3. 38: Sơ đồ tuần tự xem đơn hàng của khách hàng

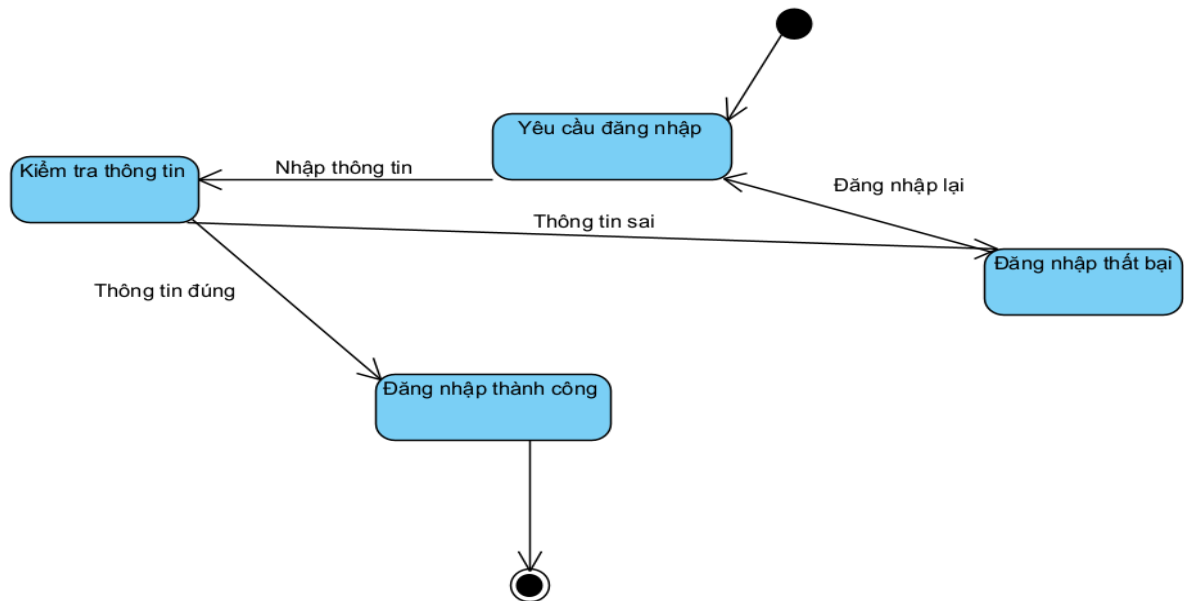
i. Biểu đồ tuần tự đặt hàng



Hình 3. 39: Sơ đồ tuần tự đặt hàng

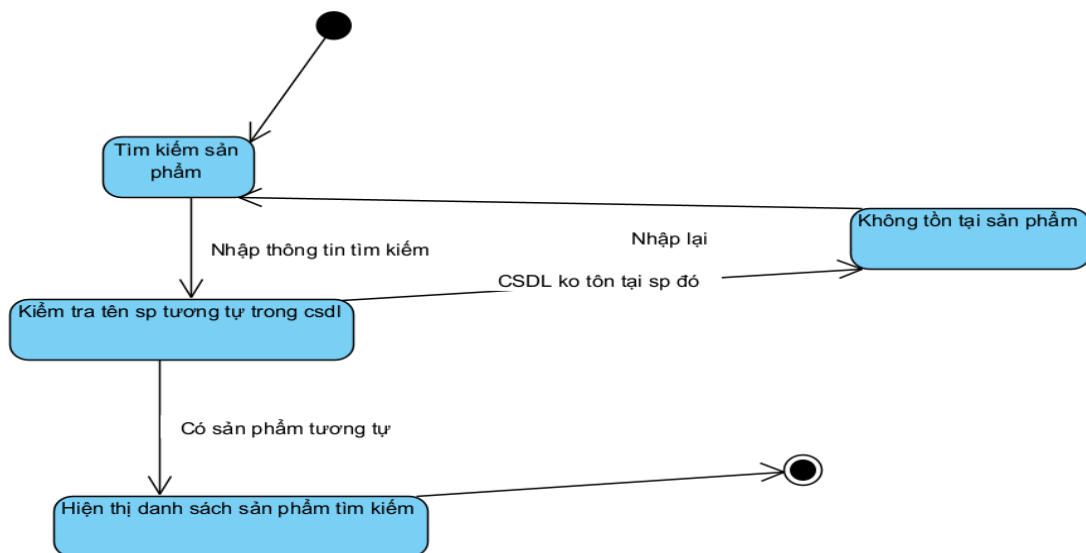
3.2.6. Biểu đồ trạng thái

a. Biểu đồ trạng thái đăng nhập



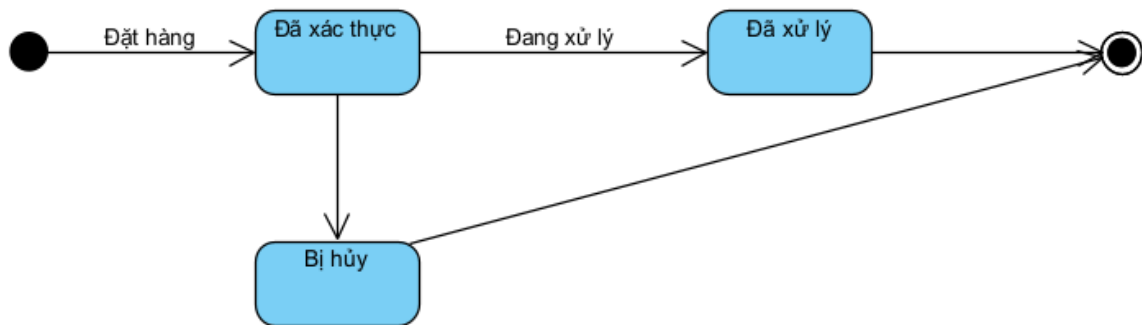
Hình 3. 40: Sơ đồ trạng thái đăng nhập

b. Biểu đồ trạng thái tìm kiếm sản phẩm



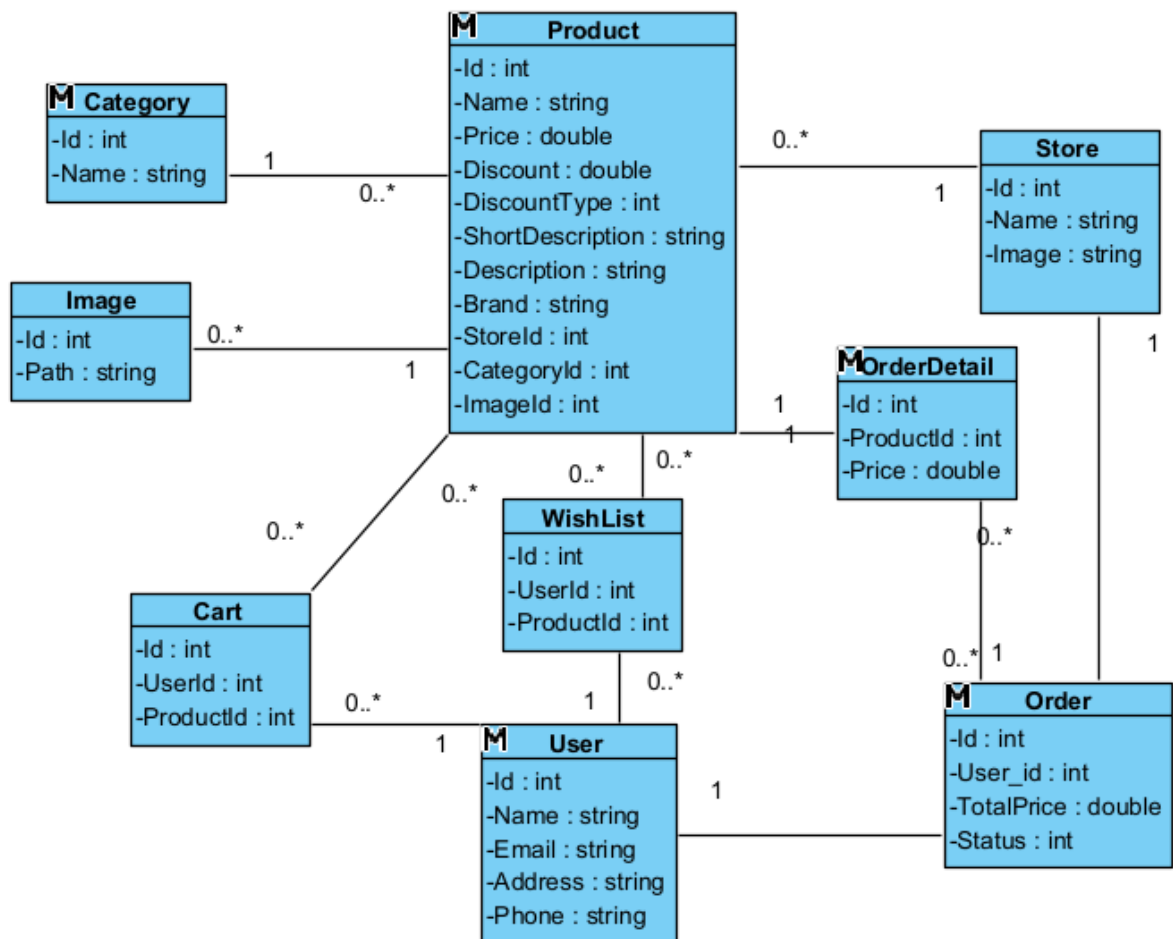
Hình 3. 41: Sơ đồ trạng thái tìm kiếm sản phẩm

c. Biểu đồ trạng thái đặt hàng



Hình 3. 42: Sơ đồ trạng thái đặt hàng

3.2.7. Biểu đồ lớp thực thể liên kết



Hình 3. 43: Sơ đồ thực thể liên kết

3.3. Thiết kế hệ thống

3.3.1. Thiết kế bảng cơ sở dữ liệu

Bảng 3. 9: Bảng users

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã tài khoản	Int	Primary Key,
2	Name	Họ và tên	Nvarchar(50)	
3	Phone	Số điện thoại	Nvarchar(50)	
4	Address	Địa chỉ	Nvarchar(50)	
5	Email	Email	Nvarchar(50)	
6	Password	Mật khẩu	Nvarchar(50)	

Bảng 3. 10: Bảng categories

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã danh mục	Int	Primary Key,
2	Name	Tên danh mục	Nvarchar(50)	

Bảng 3. 11: Bảng images

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã ảnh	Int	Primary Key,
2	Path	Tên nguồn ảnh	Nvarchar(50)	

Bảng 3. 12: Bảng products

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã sản phẩm	Int	Primary Key,
2	Name	Tên sản phẩm	Nvarchar(50)	
3	Category_id	Mã danh mục	Int	Nối với bảng Categories
4	Discount	Giảm giá	Float	
5	Discount_type	Loại giảm giá	Int	
6	Price	Giá	Float	
7	Description	Mô tả	Nvarchar(50)	
8	Short_description	Mô tả ngắn	Nvarchar(50)	
9	Brand	Nhà phát hành	Nvarchar(50)	
10	Image_id	Mã danh mục	Int	Nối với bảng Images
11	Store_id	Mã danh mục	Int	Nối với bảng Stores

Bảng 3. 13: Bảng carts

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã giỏ hàng	Int	Primary Key,
2	User_id	Mã tài khoản	Int	Nối với bảng User

3	Product_id	Mã sản phẩm	Int	Nối với bảng Product
---	------------	-------------	-----	----------------------

Bảng 3. 14: Bảng wishlists

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã yêu thích	Int	Primary Key,
2	User_id	Mã tài khoản	Int	Nối với bảng User
3	Product_id	Mã sản phẩm	Int	Nối với bảng Product

Bảng 3. 15: Bảng orders

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã đơn hàng	Int	Primary Key,
2	User_id	Mã tài khoản	Int	Nối với bảng User
3	Total	Tổng tiền	Float	
4	Status	Trạng thái	Int	

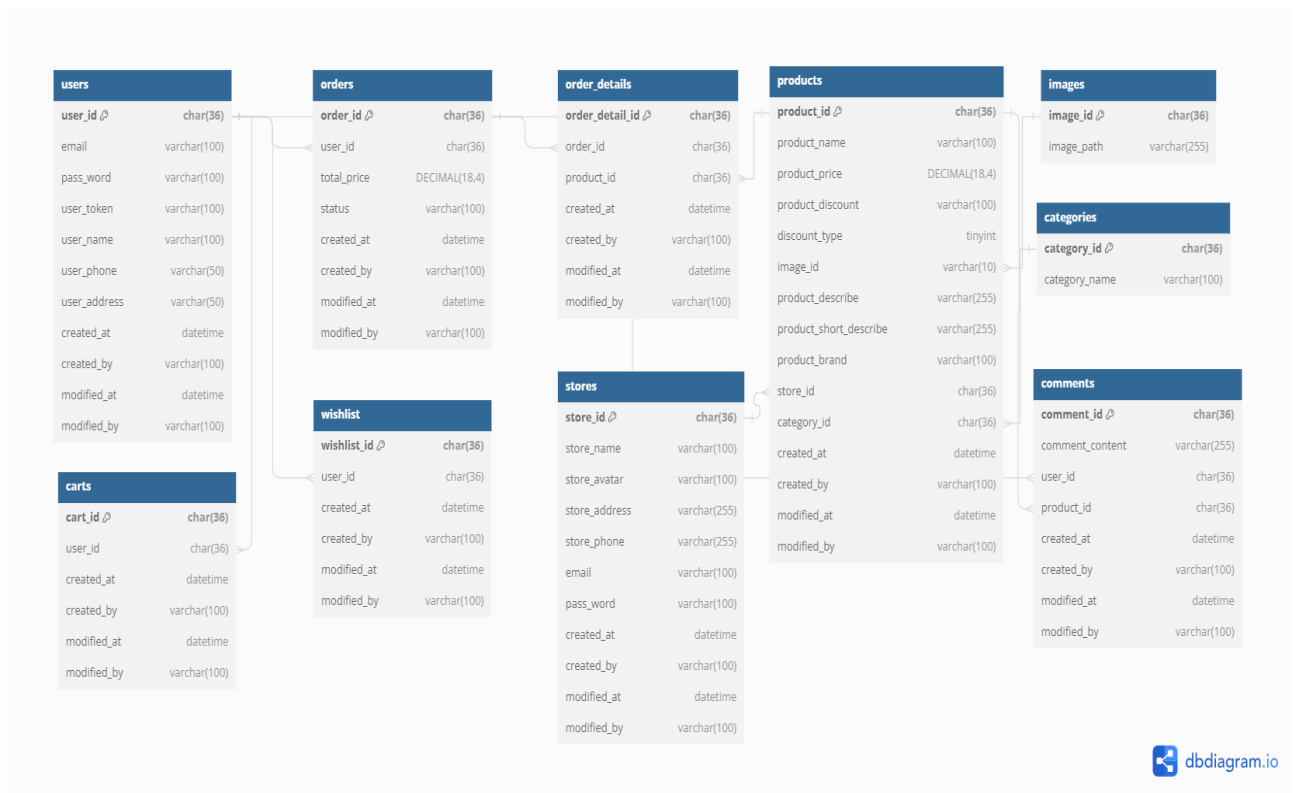
Bảng 3. 16: Bảng order_details

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã chi tiết đơn hàng	Int	Primary Key,
2	Order_id	Mã đơn hàng	Int	Nối với bảng Order
3	Product_id	Mã sản phẩm	Int	Nối với bảng Product

Bảng 3. 17: Bảng stores

STT	Tên trường	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã cửa hàng	Int	Primary Key,
2	Name	Họ và tên	Nvarchar(50)	
3	Email	Email	Nvarchar(50)	
4	Password	Mật khẩu	Nvarchar(50)	
5	Image	Ảnh cửa hàng	Nvarchar(50)	

3.3.2. Bảng database diagram



Hình 3. 44: Sơ đồ diagram

3.4. Thiết kế giao diện

3.4.1. Thiết kế giao diện người dùng

Thiết kế giao diện người dùng (UI) là một bước quan trọng trong quá trình phát triển sản phẩm thương mại game. Giao diện người dùng được thiết kế với mục tiêu chính là đảm bảo tính thân thiện, dễ sử dụng, và hấp dẫn. Quá trình thiết kế giao diện bao gồm việc tạo ra các mockup và wireframe chi tiết để hình dung rõ ràng về giao diện người dùng cuối cùng. Các yếu tố chính của giao diện bao gồm:

- Trang chủ: Trang chủ được thiết kế để trình bày các danh mục game nổi bật, các chương trình khuyến mãi, và tin tức mới nhất. Thiết kế tập trung vào việc thu hút sự chú ý của người dùng với các hình ảnh và video hấp dẫn, cũng như cung cấp các đường dẫn nhanh đến các danh mục game phổ biến.
- Trang danh mục game: Trang danh mục game liệt kê các game theo thể loại, nhà sản xuất, và các tiêu chí tìm kiếm khác. Người dùng có thể dễ dàng tìm kiếm và lọc các game dựa trên các tiêu chí như giá cả, đánh giá của người dùng, và ngày phát hành.
- Trang chi tiết game: Trang chi tiết game hiển thị thông tin chi tiết về từng game, bao gồm mô tả, hình ảnh, video giới thiệu, và đánh giá từ người dùng. Giao diện được thiết kế để cung cấp đầy đủ thông tin cần thiết giúp người dùng đưa ra quyết định mua hàng.
- Giỏ hàng và thanh toán: Giao diện giỏ hàng và quy trình thanh toán được thiết kế đơn giản và dễ hiểu, giúp người dùng dễ dàng hoàn thành giao dịch. Các bước trong quy trình thanh toán được tối giản và rõ ràng, giảm thiểu các bước không cần thiết để cải thiện trải nghiệm người dùng.
- Trang quản lý: Dành cho nhân viên quản lý, bao gồm các chức năng như quản lý sản phẩm, đơn hàng, và người dùng. Giao diện quản lý được thiết kế sao cho dễ sử dụng và hiệu quả, giúp nhân viên dễ dàng thực hiện các nhiệm vụ quản lý hàng ngày.

3.4.2. Các nguyên tắc thiết kế giao diện

Trong quá trình thiết kế giao diện, các nguyên tắc sau đã được áp dụng để đảm bảo tính thẩm mỹ và chức năng:

- Nguyên tắc về màu sắc: Màu sắc được chọn lọc kỹ lưỡng để phù hợp với chủ đề của trang web và tạo cảm giác hài hòa. Sử dụng màu sắc nhất quán trong toàn bộ giao diện giúp tạo nên sự liên kết và dễ nhận diện.

- Nguyên tắc về bố cục: Bố cục trang web được thiết kế rõ ràng và trực quan, giúp người dùng dễ dàng tìm thấy thông tin cần thiết. Các phần tử trên trang được sắp xếp hợp lý, tạo nên một luồng thông tin mạch lạc và dễ theo dõi.
- Nguyên tắc về điều hướng: Thanh điều hướng và các liên kết được sắp xếp hợp lý, giúp người dùng dễ dàng di chuyển giữa các trang và tìm kiếm thông tin. Điều hướng rõ ràng và dễ sử dụng là một yếu tố quan trọng để cải thiện trải nghiệm người dùng.
- Nguyên tắc về tương tác: Các yếu tố tương tác như nút bấm, biểu mẫu, và thông báo được thiết kế để dễ dàng sử dụng và phản hồi nhanh chóng. Các yếu tố này giúp người dùng thực hiện các tác vụ một cách hiệu quả và giảm thiểu lỗi sử dụng.

3.5. Cài đặt

3.5.1. Môi trường phát triển

Môi trường phát triển được thiết lập để hỗ trợ quá trình phát triển và kiểm thử hệ thống một cách hiệu quả. Các công cụ và công nghệ chính bao gồm:

- NuxtJS: Framework mạnh mẽ cho phát triển ứng dụng web tĩnh và động, dựa trên Vue.js.
- Laravel: Framework PHP để xây dựng API và xử lý backend, nổi tiếng với cú pháp dễ hiểu và nhiều tính năng hữu ích.
- TailwindCSS: Framework CSS tiện dụng, cho phép thiết kế giao diện nhanh chóng và dễ dàng.
- MySQL: Hệ quản trị cơ sở dữ liệu phổ biến và mạnh mẽ.

3.5.2. Quy trình cài đặt

Quy trình cài đặt bao gồm các bước từ thiết lập môi trường phát triển đến triển khai hệ thống trên môi trường sản xuất:

- Cài đặt các công cụ và framework cần thiết: Bao gồm NuxtJS, Laravel, TailwindCSS, MySQL. Việc cài đặt được thực hiện thông qua các trình quản lý gói như npm, Composer, và Docker Compose.
- Thiết lập cơ sở dữ liệu: Tạo các bảng và quan hệ cần thiết trong MySQL. Các lệnh migration của Laravel được sử dụng để tạo ra cấu trúc cơ sở dữ liệu.
- Cấu hình các tệp môi trường: Thiết lập các biến môi trường cho cả môi trường phát triển và sản xuất, bao gồm thông tin kết nối cơ sở dữ liệu, cấu hình máy chủ, và các khóa API.

3.6. Kiểm thử phần mềm

3.6.1. Các loại kiểm thử

Để đảm bảo chất lượng của hệ thống, nhiều loại kiểm thử đã được thực hiện, bao gồm:

- **Kiểm thử chức năng:** Đảm bảo các chức năng của hệ thống hoạt động đúng như mong đợi. Các kịch bản kiểm thử chức năng bao gồm kiểm tra các thao tác cơ bản như đăng ký, đăng nhập, thêm vào giỏ hàng, thanh toán, và quản lý sản phẩm.
- **Kiểm thử hiệu suất:** Đánh giá khả năng chịu tải và hiệu suất của hệ thống khi có nhiều người dùng truy cập. Các công cụ kiểm thử hiệu suất như Apache JMeter được sử dụng để mô phỏng lượng truy cập lớn và đo lường thời gian phản hồi của hệ thống.
- **Kiểm thử bảo mật:** Kiểm tra và đánh giá các lỗ hổng bảo mật có thể tồn tại trong hệ thống. Các bài kiểm thử bảo mật bao gồm kiểm tra tấn công SQL Injection, Cross-Site Scripting (XSS), và kiểm tra cấu hình bảo mật của máy chủ.
- **Kiểm thử khả dụng:** Đánh giá mức độ thân thiện và dễ sử dụng của giao diện người dùng. Các bài kiểm thử khả dụng bao gồm kiểm tra giao diện trên các thiết bị và trình duyệt khác nhau, và thu thập phản hồi từ người dùng thử nghiệm.

3.6.2. Kịch bản kiểm thử

Các kịch bản kiểm thử được xây dựng chi tiết để kiểm tra từng chức năng của hệ thống:

- **Kiểm thử đăng ký và đăng nhập:** Đảm bảo quy trình đăng ký và đăng nhập hoạt động chính xác. Các kịch bản bao gồm kiểm tra các trường hợp như tài khoản đã tồn tại, mật khẩu không hợp lệ, và xác thực email.
- **Kiểm thử giỏ hàng và thanh toán:** Kiểm tra chức năng thêm sản phẩm vào giỏ hàng và thanh toán. Các kịch bản bao gồm kiểm tra tính toán tổng giá, áp dụng mã giảm giá, và xác thực thông tin thanh toán.
- **Kiểm thử tìm kiếm và duyệt sản phẩm:** Đảm bảo người dùng có thể tìm kiếm và duyệt sản phẩm một cách hiệu quả. Các kịch bản bao gồm kiểm tra tính năng tìm kiếm theo từ khóa, lọc sản phẩm theo danh mục, và sắp xếp kết quả tìm kiếm.
- **Kiểm thử quản lý sản phẩm:** Kiểm tra các chức năng quản lý sản phẩm của nhân viên quản lý. Các kịch bản bao gồm thêm, sửa, xóa sản phẩm, và quản lý tồn kho.

3.6.3. Kết quả kiểm thử

Kết quả kiểm thử cho thấy hệ thống hoạt động ổn định và đáp ứng các yêu cầu chức năng:

- Chức năng hoạt động đúng như mong đợi: Không có lỗi nghiêm trọng được phát hiện. Các bài kiểm thử chức năng cho thấy tất cả các tính năng chính đều hoạt động chính xác.
- Hiệu suất hệ thống đạt yêu cầu: Hệ thống có thể xử lý nhiều yêu cầu đồng thời mà không gặp vấn đề về hiệu suất. Các bài kiểm thử hiệu suất cho thấy thời gian phản hồi của hệ thống nằm trong giới hạn chấp nhận được.
- Bảo mật được đảm bảo: Không có lỗ hổng bảo mật nghiêm trọng được phát hiện. Các bài kiểm thử bảo mật cho thấy hệ thống được bảo vệ tốt chống lại các loại tấn công phổ biến.
- Khả dụng cao: Giao diện người dùng được đánh giá cao về tính thân thiện và dễ sử dụng. Phản hồi từ người dùng thử nghiệm cho thấy họ dễ dàng tìm thấy thông tin cần thiết và thực hiện các thao tác trên trang web.

Qua quá trình kiểm thử, các lỗi nhỏ đã được phát hiện và khắc phục kịp thời, đảm bảo hệ thống sẵn sàng cho việc triển khai chính thức. Các bước kiểm thử đã giúp đảm bảo rằng hệ thống không chỉ hoạt động đúng chức năng mà còn đáp ứng các yêu cầu về hiệu suất, bảo mật, và khả dụng.

Tiểu kết

Chương 3 đã tập trung vào việc thiết kế và triển khai cơ sở dữ liệu cũng như các thành phần kỹ thuật cần thiết cho sàn thương mại game. Đầu tiên, chương này mô tả chi tiết về cơ sở dữ liệu, bao gồm việc thiết kế các bảng, quan hệ giữa các bảng và các ràng buộc dữ liệu nhằm đảm bảo tính toàn vẹn và nhất quán của hệ thống. Tiếp theo, các use case chính đã được phân tích và trình bày qua sơ đồ tuần tự, giúp làm rõ luồng hoạt động của hệ thống từ khi người dùng tương tác cho đến khi các yêu cầu được xử lý và phản hồi. Bên cạnh đó, lớp đối tượng cũng được thiết kế chi tiết, đảm bảo rằng mỗi thành phần trong hệ thống đều được mô tả và tương tác một cách rõ ràng và hợp lý. Những nỗ lực này tạo ra một nền tảng vững chắc cho việc phát triển và triển khai hệ thống sàn thương mại game, đảm bảo rằng tất cả các chức năng và quy trình đều hoạt động hiệu quả và mượt mà. Qua việc hoàn thiện các thiết kế và phân tích kỹ thuật này, chương 3 đã đặt nền móng quan trọng cho các bước phát triển và thử nghiệm tiếp theo của dự án.

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

4.1. Kết quả đạt được

4.1.1. Lý thuyết

Trong quá trình thực hiện đề tài xây dựng sàn thương mại điện tử dành cho các sản phẩm và dịch vụ liên quan đến game, em nghiên cứu đã đạt được nhiều kết quả quan trọng về mặt lý thuyết.

Đầu tiên, em đã tìm hiểu sâu về các công nghệ chủ đạo được sử dụng trong dự án, bao gồm NuxtJS - một framework mạnh mẽ cho phát triển ứng dụng web tĩnh và động dựa trên Vue.js, Laravel - một framework PHP phổ biến dành cho phát triển ứng dụng web với khả năng xây dựng các API mạnh mẽ, và TailwindCSS - một CSS framework tiện dụng giúp tạo giao diện người dùng nhanh chóng và hiệu quả. Ngoài ra, kiến thức về MySQL cũng được củng cố thông qua việc thiết kế và quản lý cơ sở dữ liệu, đảm bảo tính nhất quán và hiệu quả truy vấn dữ liệu.

Qua đó, em đã có cái nhìn toàn diện và sâu sắc về cách xây dựng một hệ thống thương mại điện tử hoàn chỉnh, từ việc thiết kế giao diện người dùng đến quản lý backend và cơ sở dữ liệu.

4.1.2. Kỹ năng

Về mặt kỹ năng, em nghiên cứu đã đạt được nhiều tiến bộ đáng kể thông qua quá trình thực hiện đề tài này.

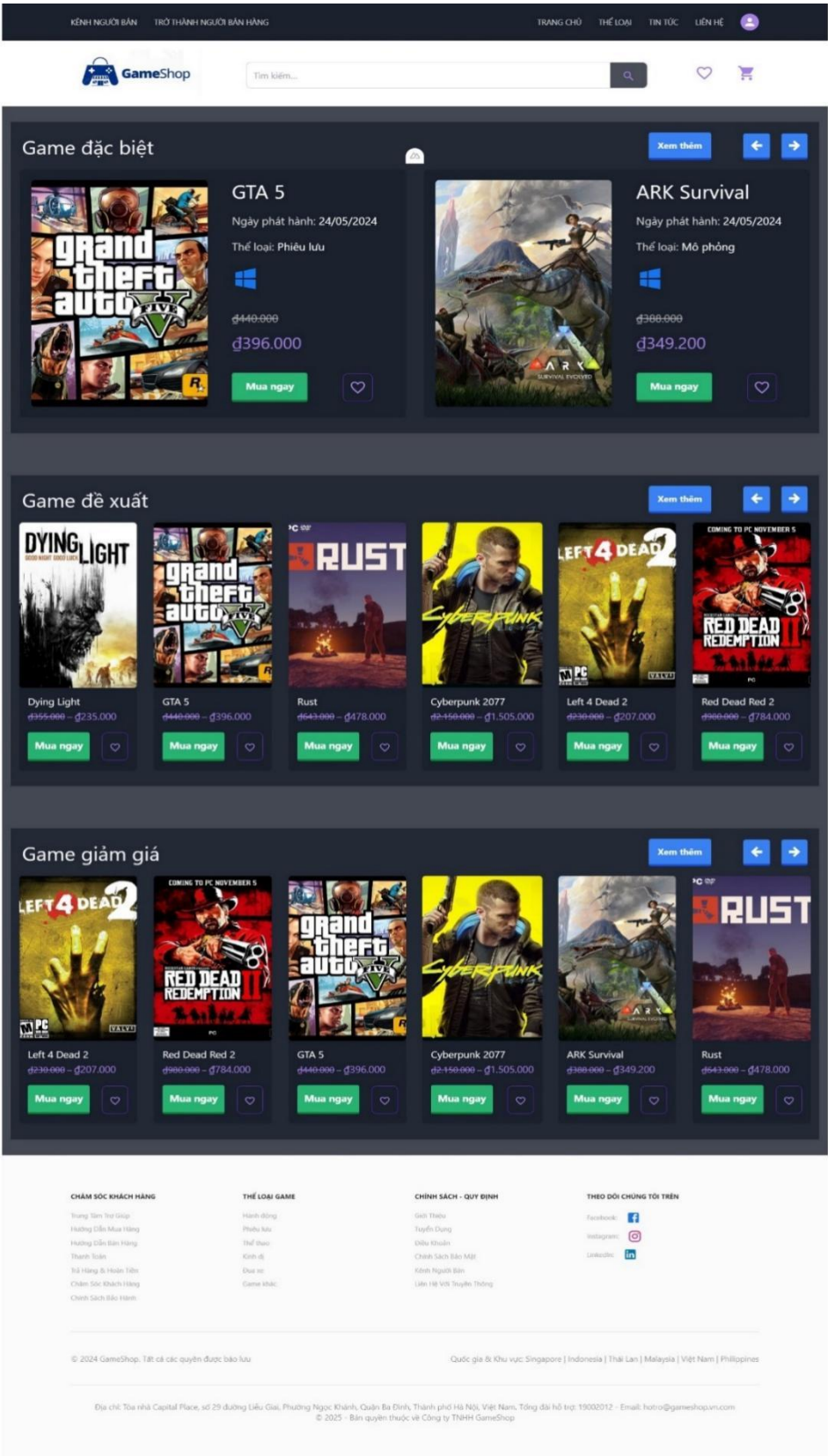
Thứ nhất, kỹ năng lập trình đã được nâng cao rõ rệt, đặc biệt là trong việc sử dụng các framework và công nghệ như NuxtJS, Laravel và TailwindCSS. Em đã học cách xây dựng các thành phần giao diện người dùng phức tạp, phát triển và tiêu chuẩn hóa các API, cũng như tối ưu hóa hiệu suất ứng dụng.

Thứ hai, kỹ năng làm việc em về quản lý dự án đã được cải thiện. Thông qua việc phân chia công việc hợp lý và hiệu quả, em đã đảm bảo tiến độ dự án và chất lượng sản phẩm cuối cùng.

Cuối cùng, kỹ năng về thiết kế cơ sở dữ liệu và quản lý dữ liệu cũng được củng cố, giúp em hiểu rõ hơn về cách thiết kế, tối ưu hóa và bảo mật cơ sở dữ liệu trong một hệ thống thương mại điện tử.

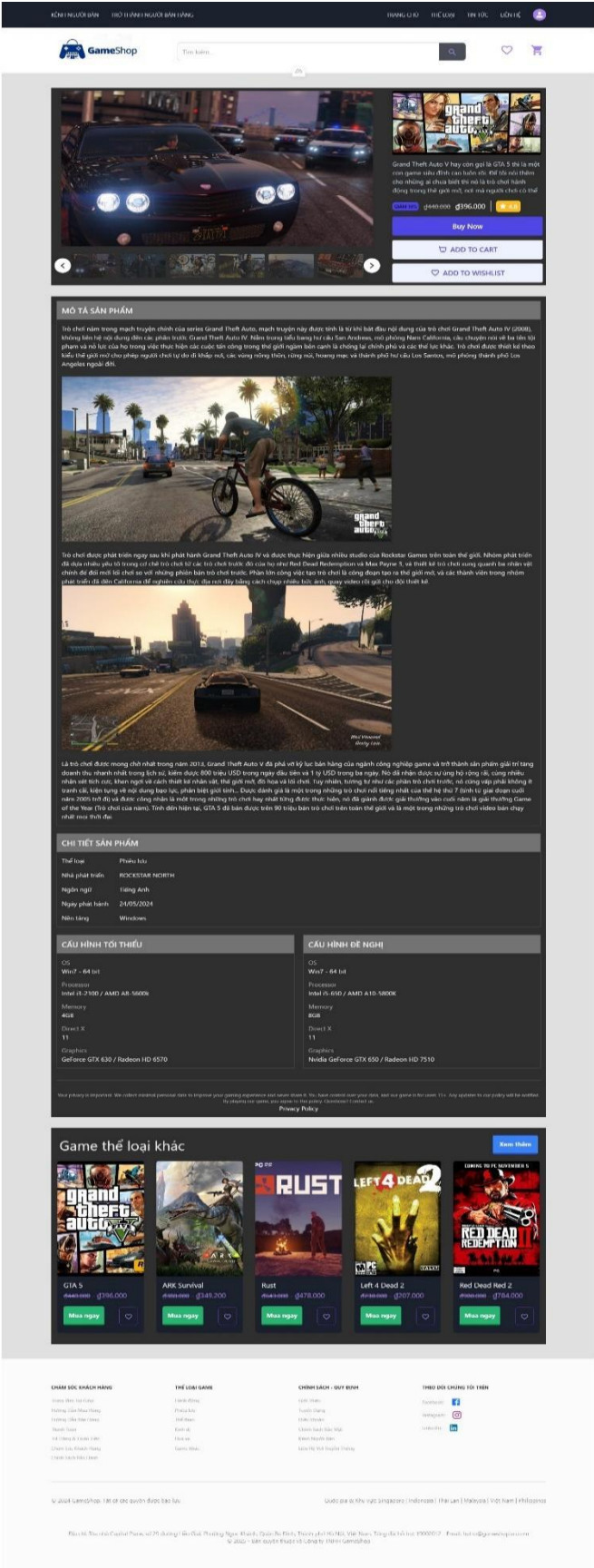
4.2. Giao diện của dự án

4.2.1. Giao diện trang chủ



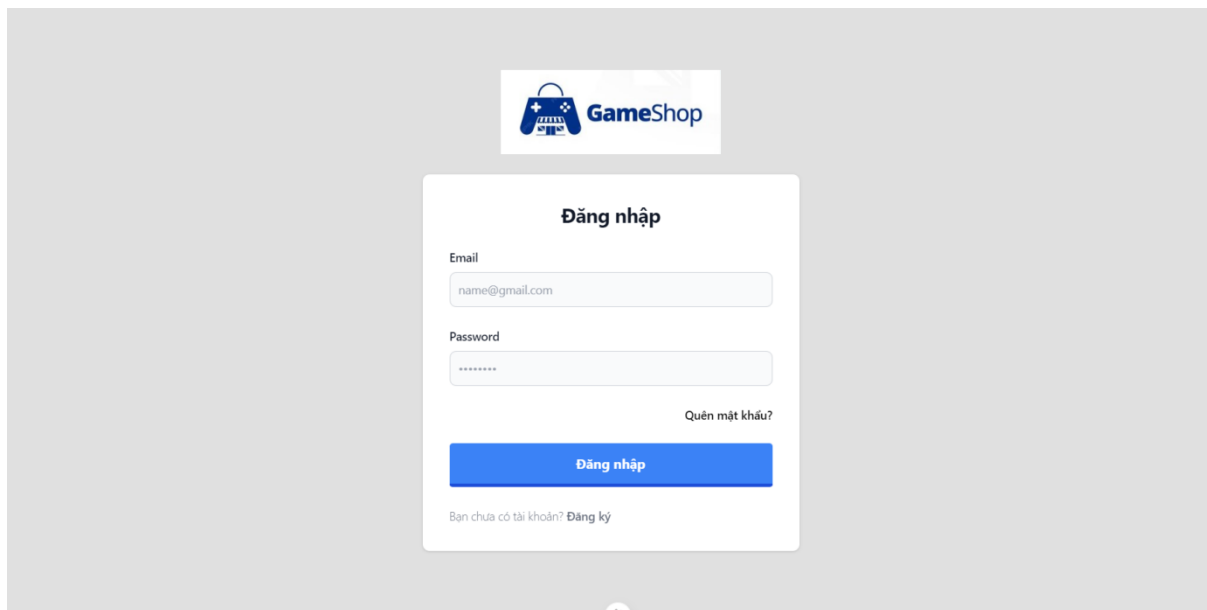
Hình 4. 1: Giao diện trang chủ

4.2.2. Giao diện chi tiết sản phẩm



Hình 4. 2: Giao diện chi tiết sản phẩm

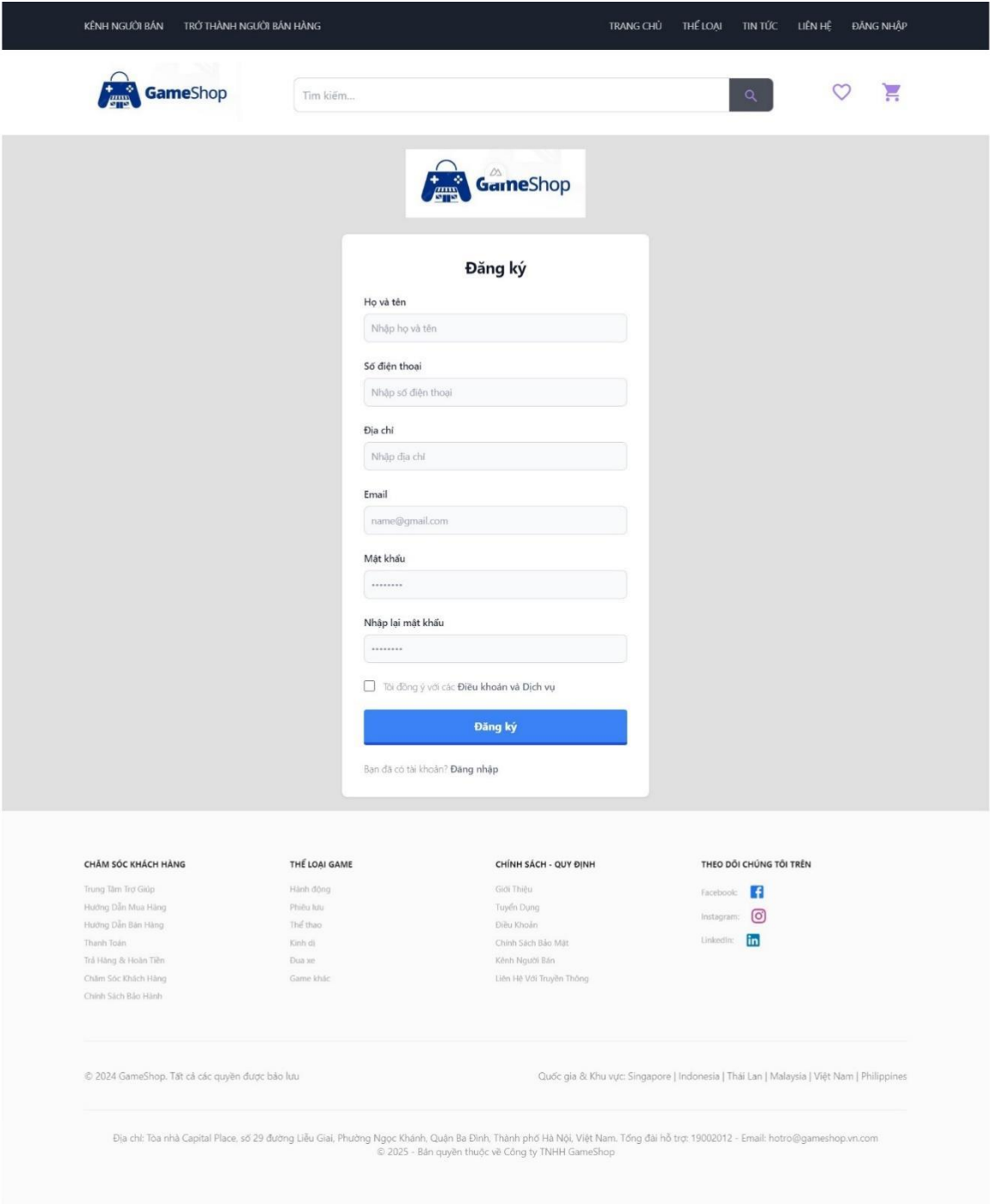
4.2.3. Giao diện đăng nhập khách hàng



The screenshot displays the login page for GameShop. At the top center, there is a logo featuring a blue game controller icon and the text "GameShop". Below the logo, the title "Đăng nhập" (Login) is centered. The form contains two input fields: "Email" with the placeholder "name@gmail.com" and "Password" with masked characters "*****". To the right of the password field is a link "Quên mật khẩu?" (Forgot password?). A prominent blue button labeled "Đăng nhập" (Login) is positioned below the fields. At the bottom, a link "Bạn chưa có tài khoản? Đăng ký" (Don't have an account? Register) is provided. The entire interface is set against a light gray background.

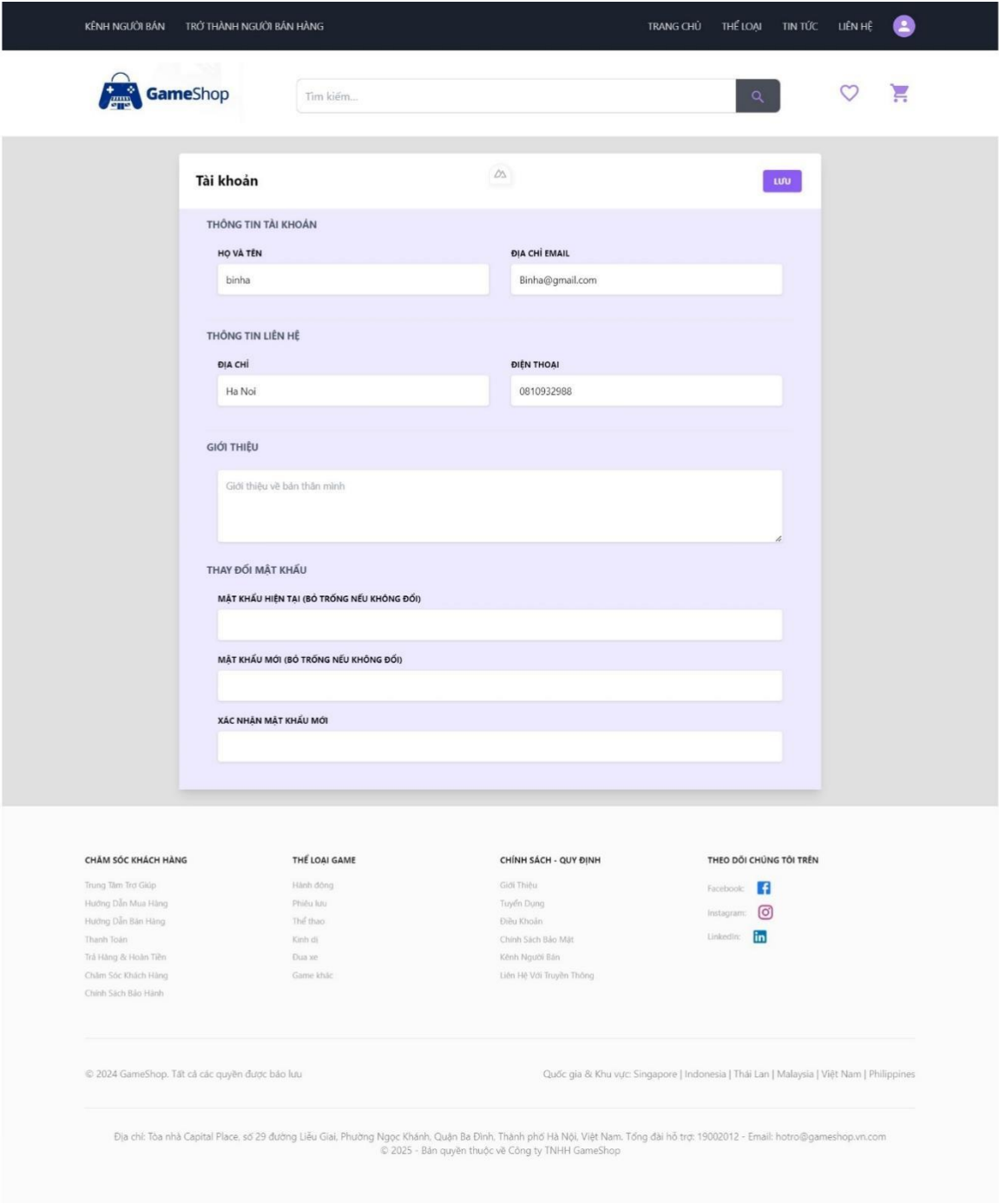
Hình 4. 3: Giao diện đăng nhập khách hàng

4.2.4. Giao diện đăng ký khách hàng



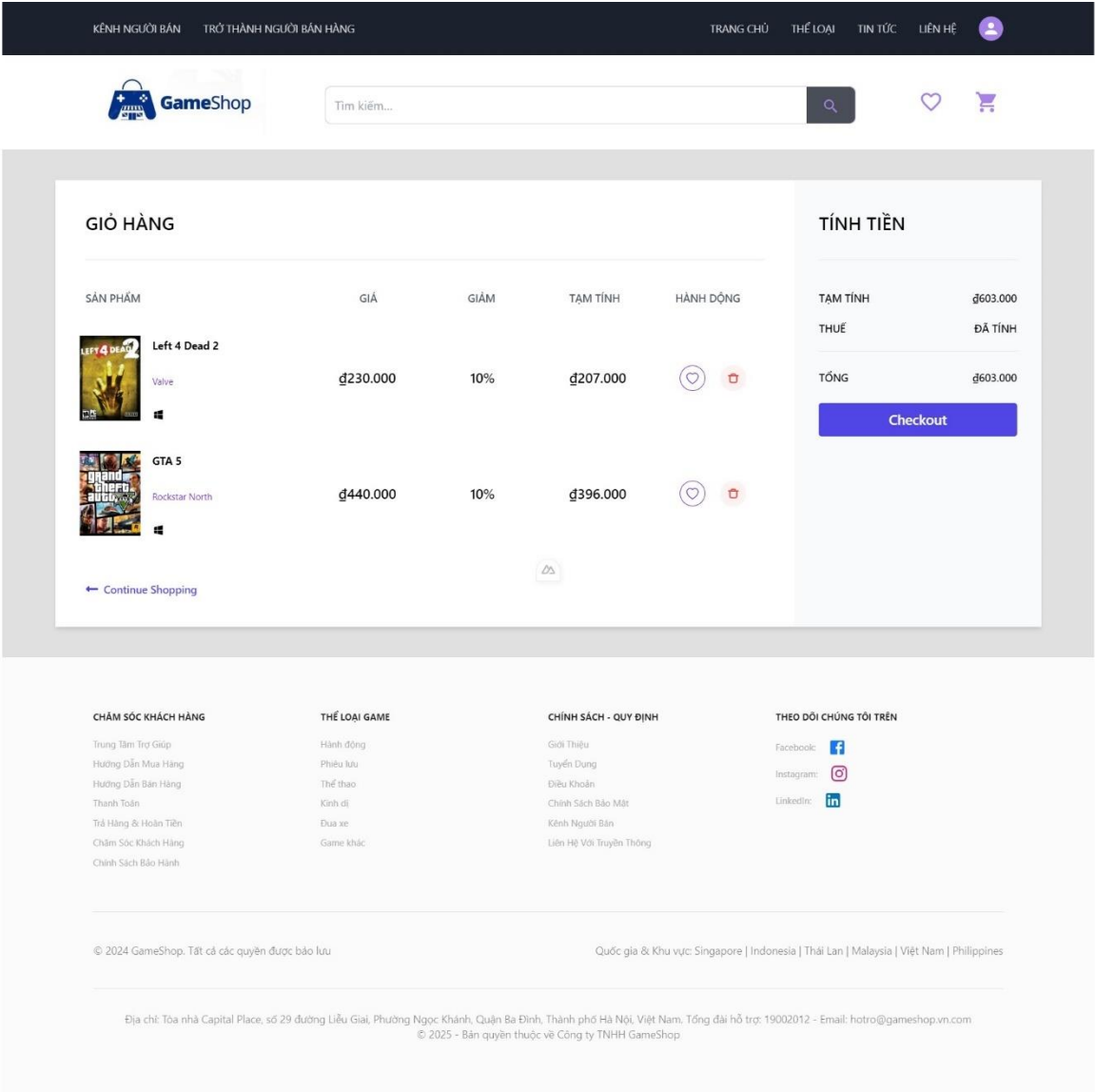
Hình 4. 4: Giao diện đăng ký khách hàng

4.2.5. Giao diện thông tin khách hàng



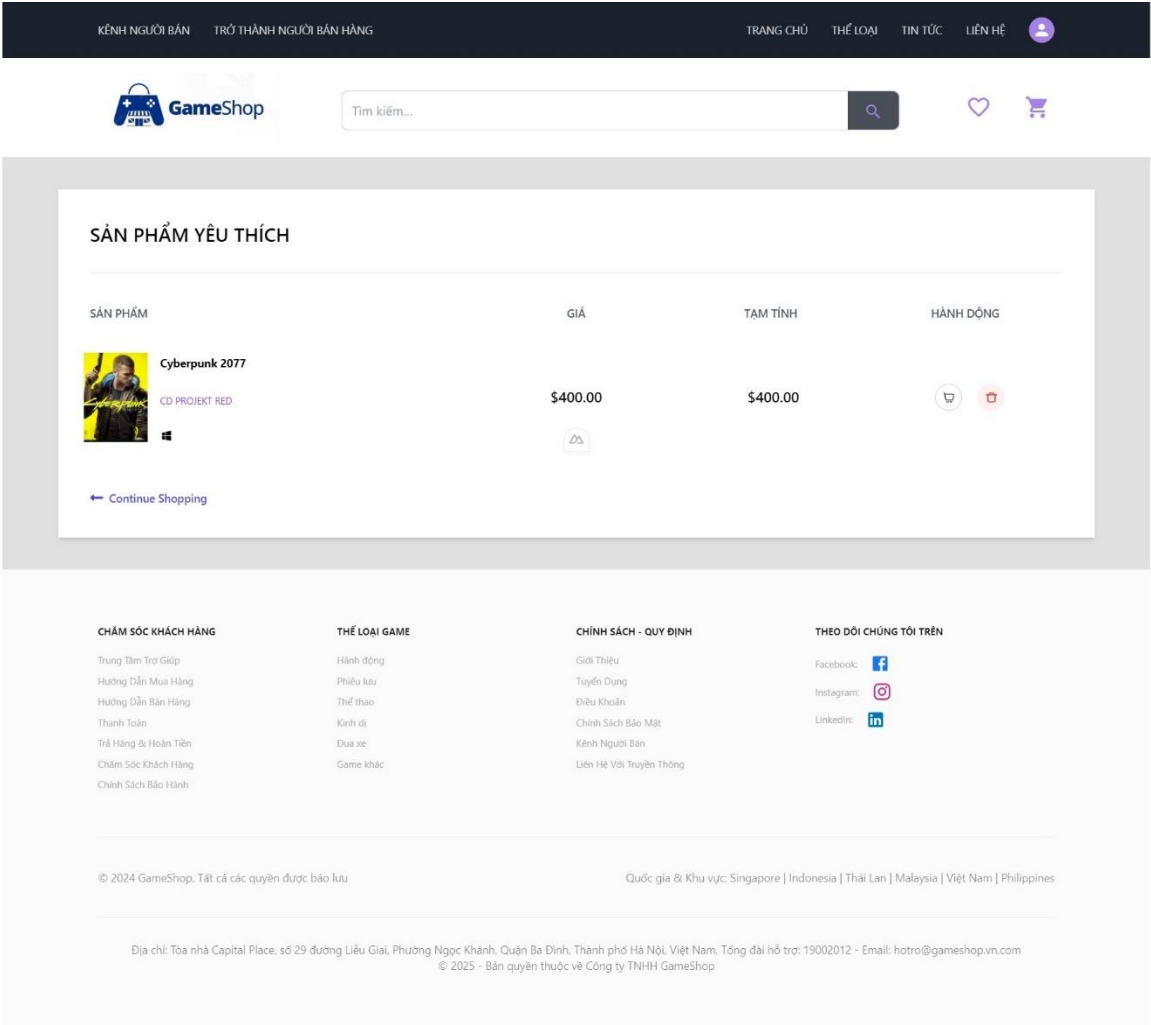
Hình 4. 5: Giao diện hồ sơ khách hàng

4.2.6. Giao diện giỏ hàng



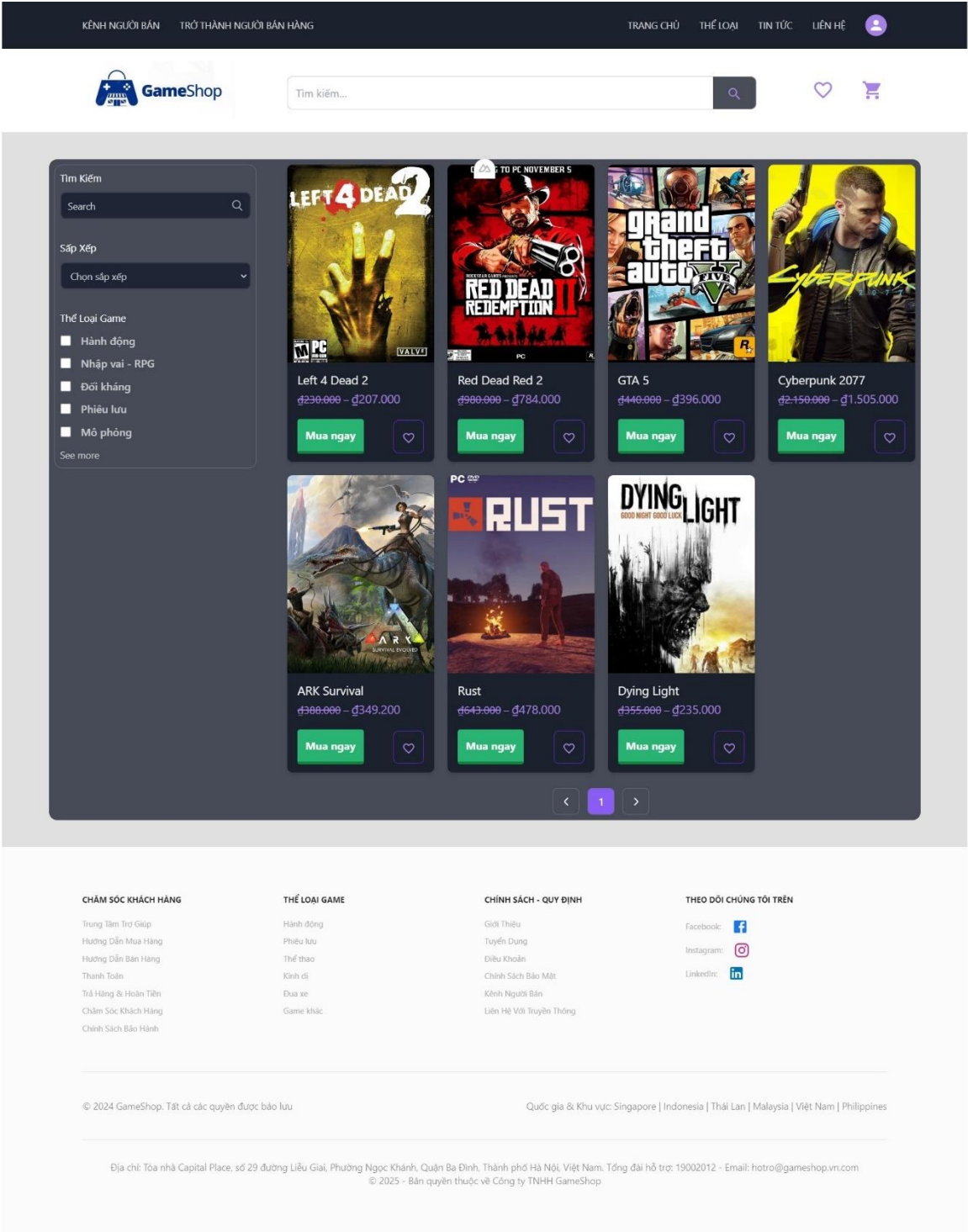
Hình 4. 6: Giao diện giỏ hàng

4.2.7. Giao diện sản phẩm yêu thích



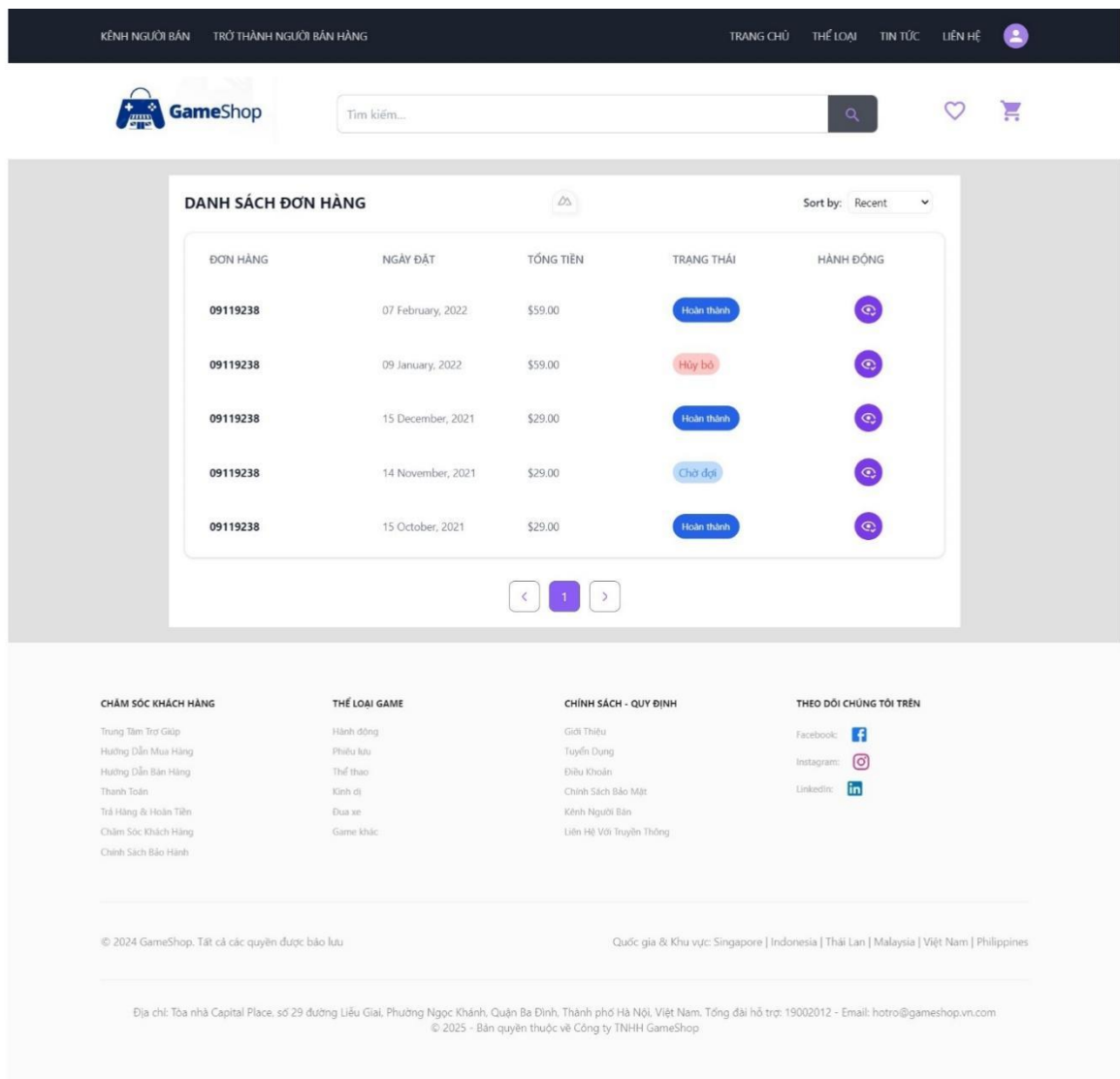
Hình 4. 7: Giao diện sản phẩm yêu thích

4.2.8. Giao diện tìm kiếm sản phẩm



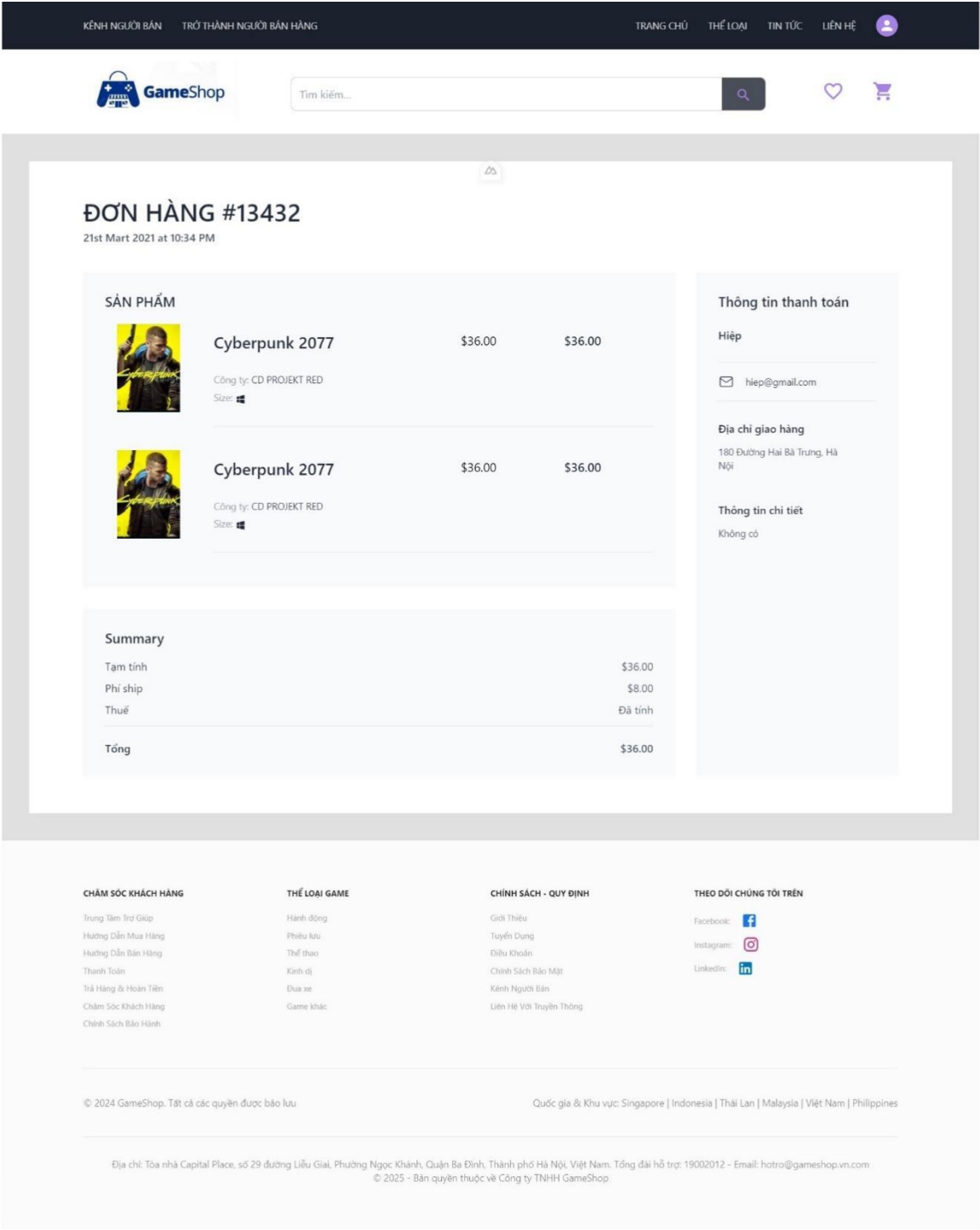
Hình 4. 8: Giao diện tìm kiếm sản phẩm

4.2.9. Giao diện danh sách đơn hàng



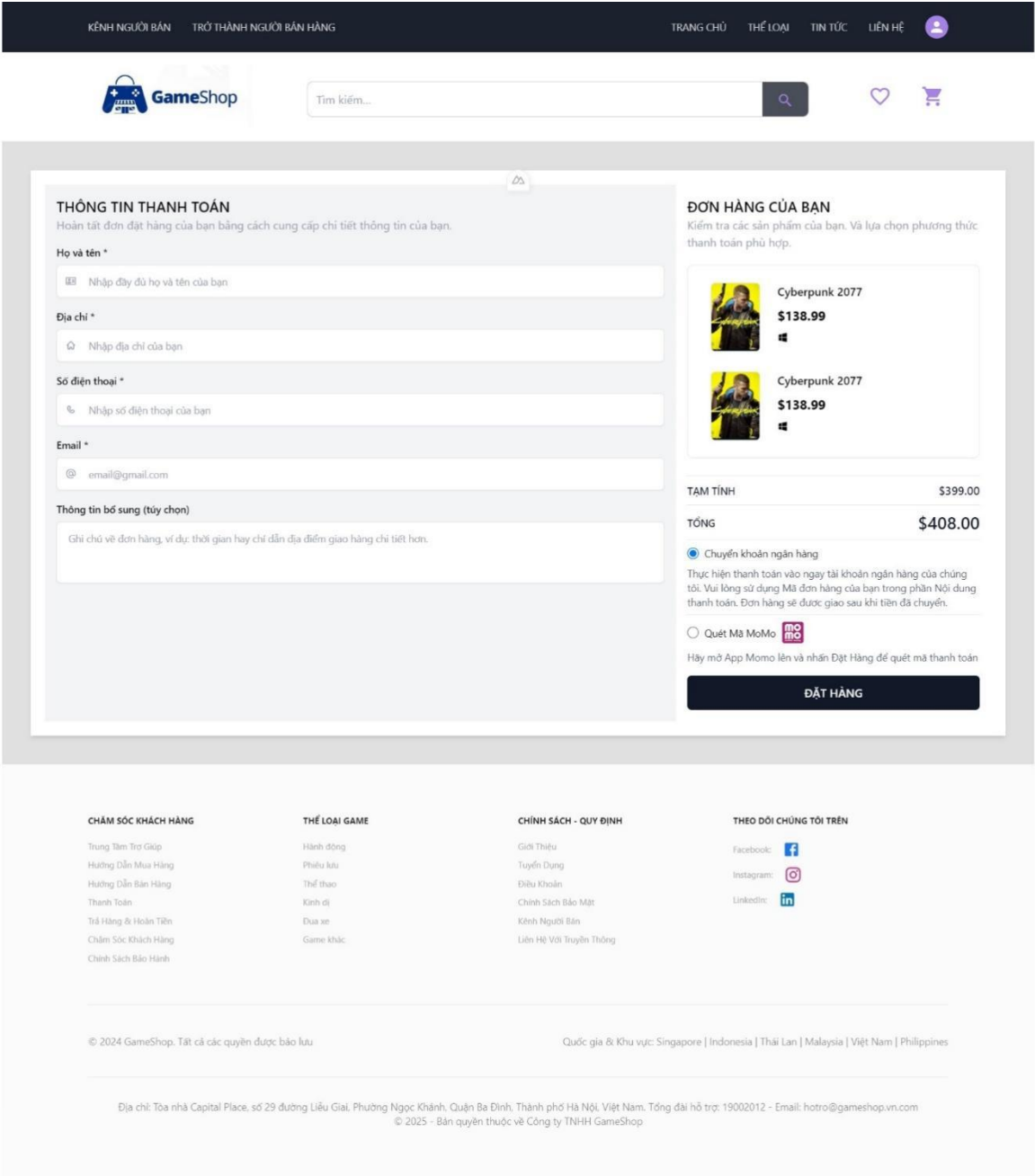
Hình 4. 9: Giao diện quản lý đơn hàng khách hàng

4.2.10. Giao diện chi tiết đơn hàng



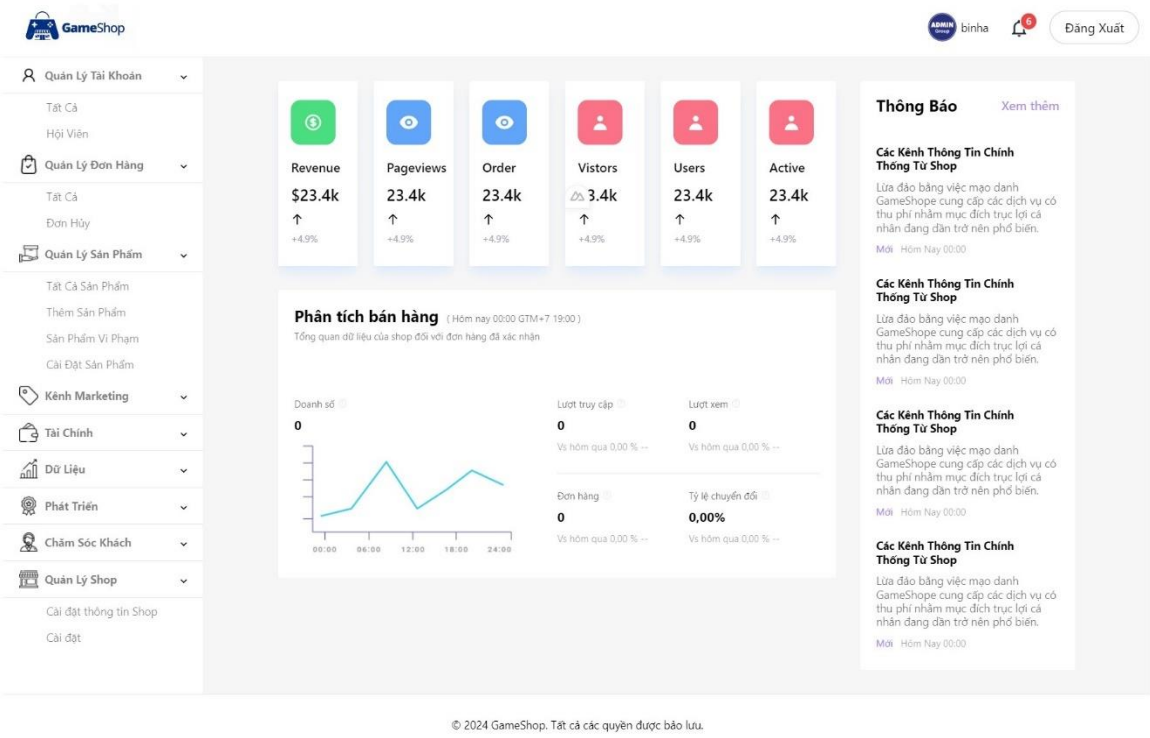
Hình 4. 10: Giao diện chi tiết đơn hàng

4.2.11. Giao diện thanh toán



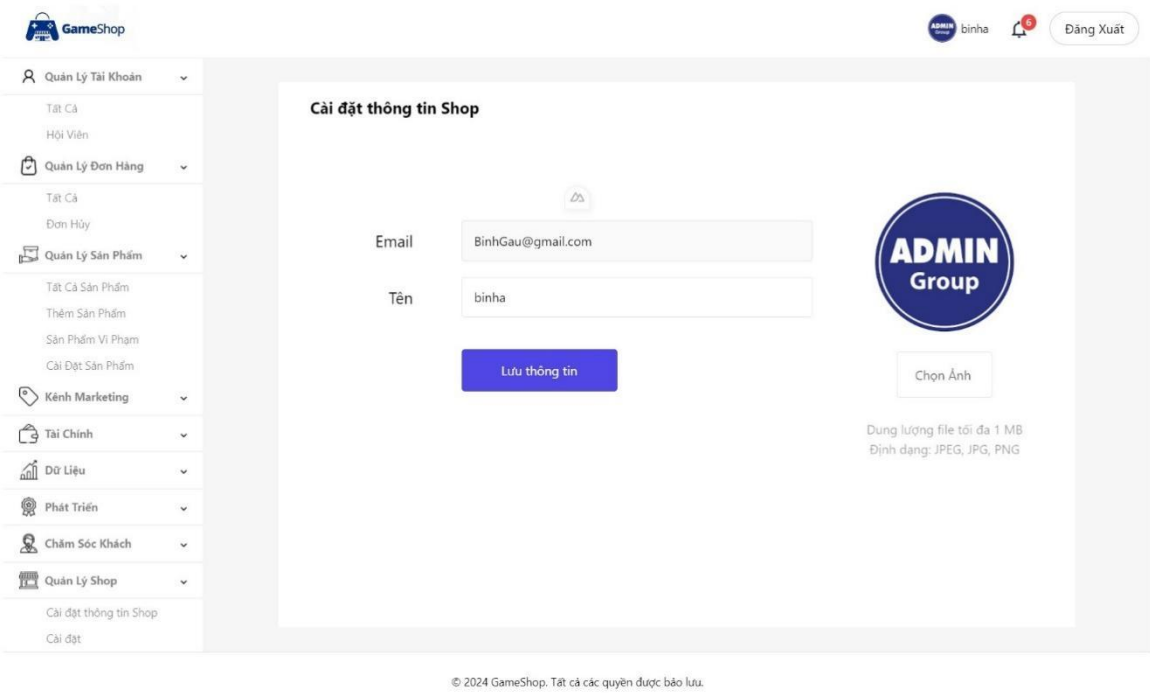
Hình 4. 11: Giao diện thanh toán

4.2.12. Giao diện thống kê



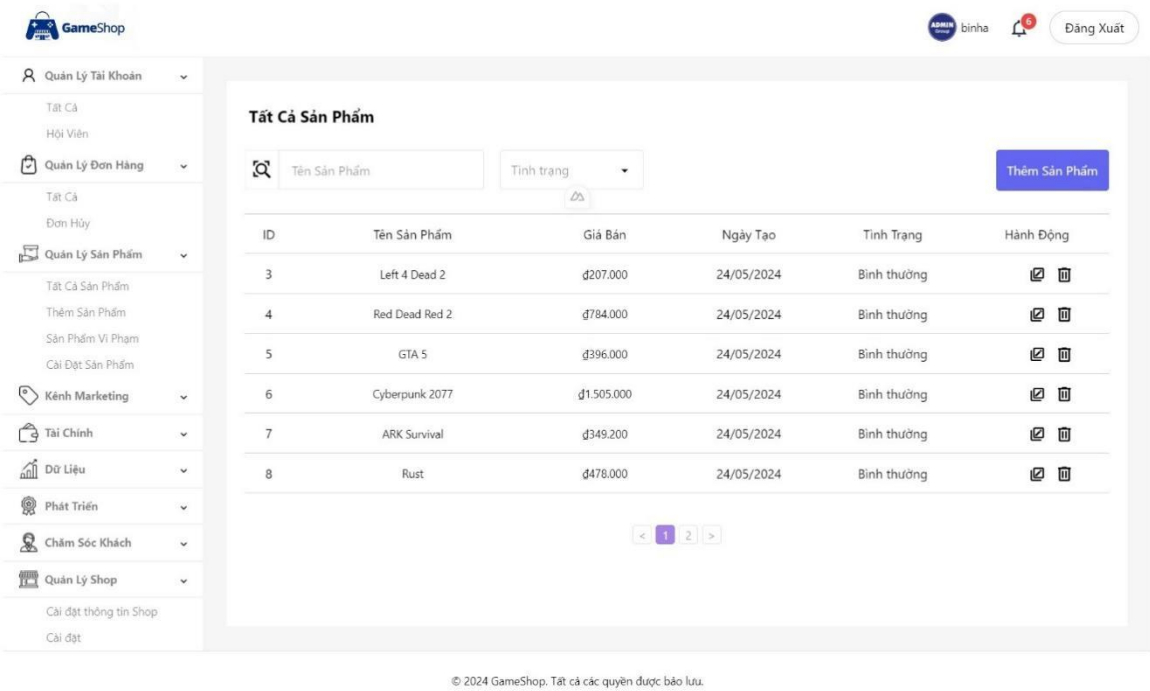
Hình 4. 12: Giao diện thống kê

4.2.13. Giao diện thông tin cửa hàng



Hình 4. 13: Giao diện thông tin cửa hàng

4.2.14. Giao diện danh sách sản phẩm



Hình 4. 14: Giao diện quản lý sản phẩm

4.2.15. Giao diện thêm sản phẩm

GameShop

ADMIN

bình

Đang Xuất

Quản Lý Tài Khoản

Tất Cả

Hội Viên

Quản Lý Đơn Hàng

Tất Cả

Đơn Hủy

Quản Lý Sản Phẩm

Tất Cả Sản Phẩm

Thêm Sản Phẩm

Sản Phẩm Vi Phạm

Cài Đặt Sản Phẩm

Kênh Marketing

Tài Chính

Dữ Liệu

Phát Triển

Chăm Sóc Khách

Quản Lý Shop

Cài đặt thông tin Shop

Cài đặt

Thêm Sản Phẩm

Thông Tin Cơ Bản

Danh mục *

Danh Mục

Tên sản phẩm *

Tên Sản Phẩm

Giá bán *

Giá Bán

Giảm giá

Giảm giá

%☒

đồng☐

Hình Ảnh

Tải ảnh lên

Thông Tin Mô Tả Cơ Bản

Paragraph

B I @ :: := ==

Thông Tin Mô Tả

Paragraph

B I @ :: := ==

Thông Tin Chi Tiết

Nhà phát hành

Nhà phát hành

Hủy Bỏ

Lưu

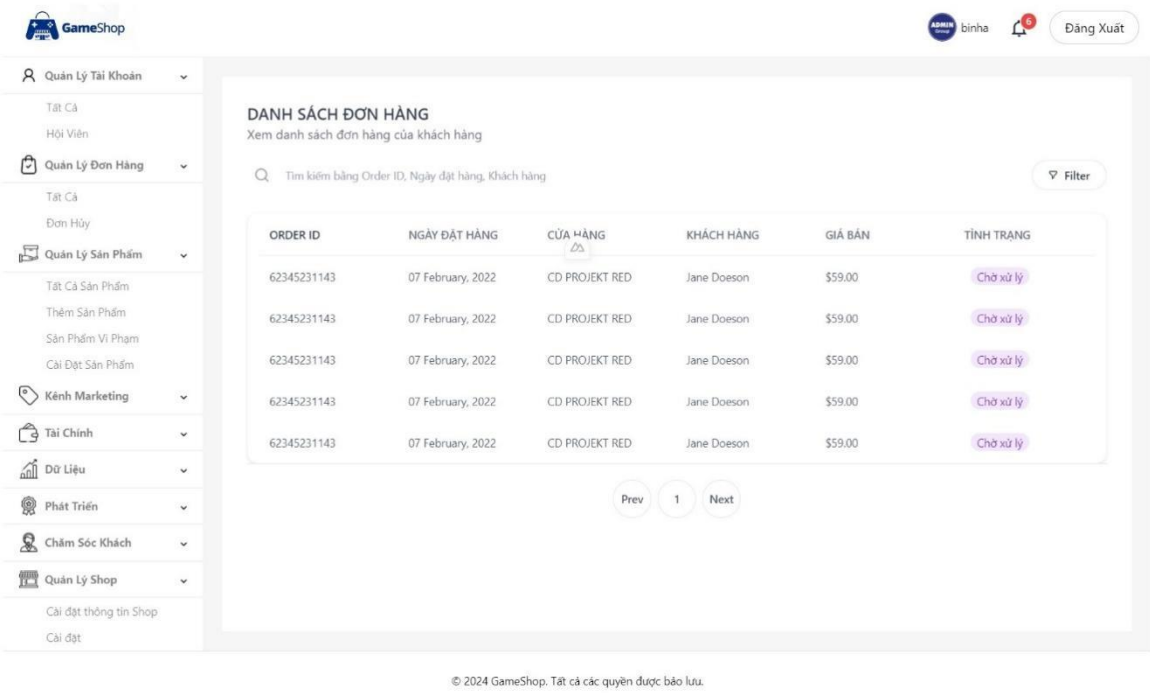
Hình 4. 15: Giao diện thêm sản phẩm

4.2.16. Giao diện sửa sản phẩm

[illegible]

Hình 4. 16: Giao diện sửa sản phẩm

4.2.17. Giao diện quản lý đơn hàng



Hình 4. 17: Giao diện quản lý đơn hàng của cửa hàng

KẾT LUẬN

Sau một quá trình nghiên cứu và triển khai, đồ án “Xây dựng sàn thương mại game bằng NuxtJS và Laravel (PHP)” đã hoàn thành với những kết quả đáng khích lệ. Việc áp dụng các công nghệ hiện đại như NuxtJS và Laravel đã giúp em xây dựng một trang web mua game trực tuyến có giao diện thân thiện, hiệu suất cao và đáp ứng được các yêu cầu đề ra. Trang web đã tích hợp đầy đủ các chức năng cơ bản như mua trực tuyến, quản lý tài khoản người dùng, quản lý danh sách sản phẩm và tìm kiếm game. Mặc dù đạt được nhiều kết quả tích cực, cũng có những nhược điểm cần cải thiện trong tương lai, hiệu năng của trang web vẫn có thể được nâng cao hơn, và một số lỗi nhỏ cần được khắc phục dựa trên phản hồi của người dùng.

Để tiếp tục phát triển dự án này trong tương lai, em định hướng nâng cao trải nghiệm người dùng thông qua tối ưu hóa hiệu năng và cải thiện giao diện dựa trên phản hồi thực tế. Bổ sung các tính năng mới như gợi ý thông minh, chế độ chơi offline và tích hợp với các dịch vụ game khác sẽ là ưu tiên hàng đầu. Bên cạnh đó, việc phát triển ứng dụng di động và mở rộng web thông qua hợp tác với các nền tảng khác cũng sẽ được chú trọng.

Cuối cùng, đồ án này không chỉ giúp em nâng cao kiến thức và kỹ năng lập trình mà còn cung cấp một cái nhìn toàn diện về quy trình phát triển một sản phẩm công nghệ từ giai đoạn ý tưởng đến khi triển khai thực tế.

DANH MỤC THAM KHẢO

Danh mục các Website tham khảo

- [1] <https://nuxtjs.org/docs>, truy cập ngày 14/05/2024
- [2] <https://topdev.vn/blog/api-la-gi/>, truy cập ngày 17/05/2024
- [3] <https://www.hostinger.vn/huong-dan/mysql-la-gi/>, truy cập ngày 19/05/2024
- [4] <https://laravel.com/docs>, truy cập ngày 24/05/2024
- [5] <https://vuejs.org/guide>, truy cập ngày 21/05/2024
- [6] <https://tailwindcss.com/docs>, truy cập ngày 10/05/2024