



TRƯỜNG ĐẠI HỌC NGOẠI - NGỮ TIN HỌC
THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MẪU THIẾT KẾ PHẦN MỀM
ĐỀ TÀI: WEBSITE QUẢN LÝ XE MOTO

Giảng viên: **Ths.Lý Đoàn Duy Khánh**

Nhóm Sinh viên thực hiện:

Nguyễn Lê Hùng Anh – 19DH110349

Lý Hạnh Phát – 19DH110838

Nguyễn Nhơn Hòa – 19DH111074

TP.HCM, 2022

Mục lục

LỜI MỞ ĐẦU	1
1 Tổng quan về đề tài	2
1.1 Giới thiệu đề tài.....	2
1.2 Khảo sát	2
1.3 Mô tả chức năng.....	3
2 Phân tích và thiết kế hệ thống.....	7
2.1 Sơ đồ phân rã chức năng.....	7
2.1.1 IshowRoom.....	7
2.1.2 Khách Hàng	8
2.1.3 Yêu cầu phi chức năng.....	8
2.2 Sơ đồ	10
2.2.1 Use Case Diagram	11
2.2.2 Activity Diagram	55
2.2.3 Sequence Diagram	58
2.2.4 Statechart Diagram	63
2.2.5 Class Diagram.....	65
2.2.6 ER Diagram	66
3 Kỹ thuật thực hiện.....	67
3.1 Cơ sở lý thuyết.....	67
3.2 Áp dụng.....	68
3.2.1 Model	68
3.2.2 View.....	69

3.2.3	Controller	69
4	Design Patterns	70
4.1	Các mẫu design pattern được áp dụng trong đề tài	70
4.1.1	Singleton Pattern.....	70
4.1.2	Template Method Pattern	71
4.1.3	Prototype Pattern.....	72
4.1.4	Proxy Pattern.....	73
4.1.5	MVC Pattern	73
4.2	Áp dụng.....	75
4.2.1	Singleton Pattern.....	75
4.2.2	Template Method Pattern	78
4.2.3	Prototype Pattern.....	80
4.2.4	Proxy Pattern.....	82
4.2.5	MVC Pattern	83
5	Triển khai – Kết quả	83
5.1	Một số chức năng và màn hình chính	83
5.1.1	Giao diện chung	83
5.1.2	Giao diện User:	84
5.1.3	Giao diện Admin:.....	87
6	Đánh Giá – Hướng Phát Triển.....	89
6.1	Ưu điểm.....	89
6.2	Khuyết điểm	89
6.3	Đánh giá	89
6.4	Hướng phát triển và mở rộng.....	90

7 Kết luận	91
Tài liệu tham khảo	92

Danh mục hình ảnh

Hình 1: Sơ đồ phân ra chức năng - Quản lý	7
Hình 2: Sơ đồ phân rã chức năng-Khách hàng.....	8
Hình 3: Sơ đồ UseCase tổng quan.....	10
Hình 4: Sơ đồ UseCase - Khách hàng	11
Hình 5: Sơ đồ UseCase - Nhân viên.....	21
Hình 6: Sơ đồ UseCase - Quản lý.....	26
Hình 7: Activity Diagram quy trình Đặt lịch hẹn - Khách hàng	55
Hình 8: Activity Diagram quy trình Đăng ký - Khách hàng	55
Hình 9: Activity Diagram quy trình Đăng nhập - Khách hàng	56
Hình 10: Activity Diagram quy trình Quản lý lịch hẹn – Nhân viên	56
Hình 11: Activity Diagram quy trình cập nhật, hủy lịch hẹn - Khách hàng	57
Hình 12: Activity Diagram quy trình Cập nhật thông tin cá nhân - Khách hàng.....	57
Hình 13: Activity Diagram quy trình quản lý mẫu xe - Quản lý.....	57
Hình 14: Sequence Diagram quy trình đăng nhập	58
Hình 15: Sequence Diagram quy trình Đăng ký tài khoản.....	58
Hình 16: Sequence Diagram quy trình Đặt lịch - Khách hàng.....	59
Hình 17: Sequence Diagram quy trình xử lý đặt lịch – Nhân viên	59
Hình 18: Sequence Diagram quy trình quản lý mẫu xe - Quản lý	60
Hình 19: Sequence Diagram quy trình quản lý hãng - Quản lý	61
Hình 20: Sequence Diagram quy trình quản lý nhân viên - Quản lý	62
Hình 21: State Machine Diagram cho đối tượng lịch hẹn - Khách hàng	63
Hình 22: State Machine Diagram cho đối tượng lịch hẹn - Nhân viên.....	63
Hình 23: State Machine Diagram cho đối tượng hãng và mẫu xe - Nhân viên	64
Hình 24: Class Diagram hệ thống	65
Hình 25: ER Diagram hệ thống Database Diagram	66
Hình 26: Database Diagram hệ thống	66
Hình 27: Model.....	68
Hình 28: View	69

Hình 29: Contronller.....	69
Hình 30: Singleton Pattern	70
Hình 31: Template Method Pattern	71
Hình 32: Prototype Pattern	72
Hình 33: Proxy Pattern	73
Hình 34: MVC Pattern.....	74
Hình 35: Giao diện đăng nhập.....	83
Hình 36: Giao diện đăng ký.....	83
Hình 37: Giao diện trang chủ - Khách hàng.....	84
Hình 38: Giao diện tra cứu xe theo hãng - Khách hàng	85
Hình 39: Giao diện đặt lịch hẹn - Khách hàng	86
Hình 40: Giao diện thông tin cá nhân - Khách hàng	86
Hình 41: Giao diện quản lý mẫu xe - Quản lý.....	87
Hình 42: Giao diện quản lý hãng xe - Quản lý.....	87
Hình 43: Giao diện quản lý danh sách xe - Nhân viên, Quản lý	88
Hình 44: Giao diện danh sách khách hàng - Nhân viên	88
Hình 45: Các Usecase hoàn thiện	89

LỜI MỞ ĐẦU

Hiện nay với tốc độ phát triển của công nghệ phần mềm, nhu cầu sử dụng các công cụ, ứng dụng giúp quản lý, thực hiện các nghiệp vụ không chỉ nhanh chóng, tiện lợi, chính xác mà còn tự động hóa các quy trình nghiệp vụ phù hợp, tùy biến theo từng ngành nghề.

Đề tài “Website quản lý xe moto” là một đề tài dựa vào ý tưởng ứng dụng công nghệ thông tin vào việc quản lý, kinh doanh moto cho một showroom bán moto cho khách hàng.

Showroom bán moto là một showroom chuyên cung cấp, buôn bán các loại xe moto phân khối lớn (BMW, Ducati, KTM, Kawasaki, Honda, Yamaha,...) cho khách hàng. Do nhu cầu mua bán moto phân khối lớn ngày càng cao nên dự án xây dựng một trang web “Web Đặt/Bán Xe Moto” giúp giải quyết nhiều vấn đề cho showroom và hỗ trợ tối đa cho khách hàng muốn đến showroom trải nghiệm dịch vụ, đặt mua xe,...

Chúng em rất mong có được sự quan tâm, giúp đỡ của các quý thầy cô.

Xin chân thành cảm ơn!

1 Tổng quan về đề tài

1.1 Giới thiệu đề tài

Đề tài “Website quản lý xe moto” nhằm xây dựng chương trình quản lý bán xe moto. Chương trình giúp quản lý các công việc như: thêm, xóa, cập nhật thông tin về các hãng xe, mẫu xe, giá cả, khuyến mãi hàng tháng cũng như là thông tin nhân viên, thông tin khách hàng; Tạo hóa đơn thanh toán, báo cáo thống kê doanh thu. Về phía khách hàng, họ có thể tra cứu các thông tin về sản phẩm mình muốn như thông số, hình ảnh, bảo hành và gửi feedback về sản phẩm mình đã từng mua. Ngoài ra, chương trình quản lý bán xe moto còn hỗ trợ tính năng đặt lịch hẹn, giúp khách hàng có thể đặt lịch đến showroom tham gia trải nghiệm sản phẩm trước khi mua xe.

Chương trình quản lý bán xe moto được xây dựng trên nền tảng ASP.Net Core. ASP.NET Core là tập hợp thư viện như một Framework mới được xây dựng ứng dụng web khi có kết nối internet, nó là một sản phẩm của Microsoft khá nổi tiếng trong cộng đồng lập trình hiện nay. Đối với chương trình quản lý bán xe moto, ASP.Net Core được dùng để thiết lập chương trình và xử lý các chức năng. Với sự trợ giúp của các phần mềm: Microsoft Visual Studio 2019 giúp xây dựng chương trình và SQL sever 2018 giúp quản lý cơ sở dữ liệu trong chương trình.

1.2 Khảo sát

- Khách hàng:

- + Khi truy cập vào trang web, khách hàng có thể tra cứu thông tin các mẫu xe, khách hàng có thể tra cứu mẫu xe theo từng hãng hoặc theo giá bán. Khách hàng có thể đặt lịch để đến showroom trải nghiệm cũng như được trực tiếp quan sát mẫu xe. Tuy nhiên khách hàng phải thực hiện đăng nhập trước đó.
- + Nếu chưa đăng nhập, hệ thống sẽ chuyển trang đăng nhập và yêu cầu người dùng đăng nhập.
- + Nếu chưa có tài khoản đăng nhập, khách hàng có thể đăng ký.
- + Sau khi đăng nhập, khách hàng có thể thực hiện đặt lịch, xem trang cá nhân, chỉnh sửa thông tin cá nhân, đánh giá mẫu xe, xem lịch sử đặt lịch, chỉnh sửa lịch hẹn với điều kiện lịch hẹn đó chưa được phía showroom xác nhận.

- **Nhân viên:**

- + Sau khi thực hiện đăng nhập và tài khoản thuộc loại tài khoản dành cho nhân viên, người nhân viên có thể: tra cứu thông tin xe, cập nhật trạng thái xe, tra cứu thông tin khách hàng.
- + Tạo hóa đơn thanh toán.
- + Tạo lịch hẹn cho khách, xác nhận lịch hẹn cho khách hàng, tra cứu lịch hẹn.
- + Áp dụng khuyến mãi.

- **Quản lý:**

- + Sau khi đăng nhập và tài khoản thuộc loại tài khoản dành cho quản lý, người quản lý có thể: tra cứu, thêm, xóa, cập nhật thông tin các hãng xe, mẫu xe, giá cả, khuyến mãi, thông tin nhân viên, khách hàng.
- + Cập nhật thông tin, hủy hóa đơn.
- + Xem báo cáo thông kê doanh thu.
- + Ngoài ra người quản lý có thể thực hiện tất cả chức năng của nhân viên.

1.3 Mô tả chức năng

- **Quản lý hãng xe:**

- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến hãng xe
- + **Mức độ ưu tiên:** Trung bình, cần cập nhật chi tiết, chính xác làm tiền đề cho việc quản lý, thay đổi dễ dàng.
- + **Tác nhân:** Người quản lý có thể thêm, cập nhật, bổ sung thay đổi thông tin của hãng xe khi có sự thay đổi trong hệ thống.
- + **Chuỗi đáp ứng:** Quản lý hãng xe cần nhập chính xác các thông tin liên quan đến hãng xe.
- + **Yêu cầu:** Quản lý hãng xe cần nhập chính xác các thông tin vì nó là tiền đề cho quản lý mẫu xe, quản lý xe. Khi cần thay đổi thông tin hãng xe thì tất cả các chức năng quản lý liên quan cũng phải thay đổi theo cho phù hợp. Còn nếu muốn hủy hoặc xóa một hãng xe thì các bảng liên quan đến hãng đó phải trống. Nếu không hệ thống sẽ báo lỗi vào không cho phép thao tác.

- **Quản lý mẫu xe:**

- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến mẫu xe
- + **Mức độ ưu tiên:** Cao, cần cập quản lý chặt chẽ đảm bảo các thông tin không bị sai sót ảnh hưởng đến cả hệ thống.
- + **Tác nhân:** Người quản lý có thể thêm, cập nhật, bổ sung thay đổi thông tin của mẫu xe khi có sự thay đổi.
- + **Chuỗi đáp ứng:** Quản lý mẫu xe cần nhập chính xác các thông tin liên quan đến mẫu xe. Có sẵn tính năng chọn dữ liệu thay cho nhập trực tiếp tránh sai sót dữ liệu ảnh hưởng đến cơ sở dữ liệu,
- + **Yêu cầu:** Quản lý mẫu xe cần nhập chính xác các thông tin vì nó là tiền đề cho quản lý xe và quản lý đặt lịch hoặc hóa đơn. Khi cần thay đổi thông tin mẫu xe thì tất cả các chức năng quản lý liên quan cũng phải thay đổi theo cho phù hợp. Còn nếu muốn hủy hoặc xóa một mẫu xe thì các bảng liên quan đến hãng đó phải trống. Nếu không hệ thống sẽ báo lỗi vào không cho phép thao tác.

- **Quản lý nhân viên:**

- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến nhân viên
- + **Mức độ ưu tiên:** Trung bình, cần cập nhật quản lý chặt chẽ đảm bảo các thông tin không bị sai sót.
- + **Tác nhân:** Người quản lý có thể thêm, cập nhật, bổ sung thay đổi thông tin của nhân viên khi nhân viên có sự thay đổi về các thông tin liên quan.
- + **Chuỗi đáp ứng:** Quản lý nhân viên cần nhập chính xác các thông tin liên quan đến nhân viên. Có các chức năng chọn thông tin thay cho nhập tay để đảm bảo về mặt dữ liệu khi lưu.
- + **Yêu cầu:** Quản lý nhân viên cần nhập chính xác các thông tin.

- **Quản lý khách hàng:**

- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến khách hàng

- + **Mức độ ưu tiên:** Cao, cần cập nhật quản lý chặt chẽ đảm bảo các thông tin không bị sai sót.
 - + **Tác nhân:** Người quản lý có thể thêm, cập nhật, bổ sung thay đổi thông tin của khách hàng khi khách hàng có sự thay đổi về các thông tin liên quan.
 - + **Chuỗi đáp ứng:** Quản lý nhân viên cần nhập chính xác các thông tin liên quan đến khách hàng. Có các chức năng chọn thông tin thay cho nhập tay để đảm bảo về mặt dữ liệu khi lưu.
 - + **Yêu cầu:** Quản lý khách hàng cần nhập chính xác các thông tin vì sẽ ảnh hưởng đến việc quản lý hóa đơn và quản lý lịch hẹn. Nếu cần thay đổi thông tin khách hàng thì các bảng liên quan cũng phải cập nhật lại.
- **Quản lý lịch hẹn:**
- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến lịch hẹn
 - + **Mức độ ưu tiên:** Cao, cần cập nhật quản lý chặt chẽ đảm bảo các thông tin không bị sai sót ảnh hưởng đến khách hàng, nhân viên và các tra cứu truy xuất dữ liệu sau này.
 - + **Tác nhân:** Người quản lý, nhân viên có thể thêm, cập nhật, bổ sung thay đổi thông tin, xác nhận, hủy lịch hẹn khi lịch hẹn có sự thay đổi về các thông tin liên quan.
 - + **Chuỗi đáp ứng:** Quản lý lịch hẹn cần nhập chính xác các thông tin liên quan đến lịch hẹn. Có các chức năng chọn thông tin thay cho nhập tay để đảm bảo chính xác về mặt dữ liệu khi lưu.
 - + **Yêu cầu:** Quản lý đặt lịch cần nhập chính xác các thông tin vì sẽ ảnh hưởng đến quản lý hệ thống và dịch vụ chăm sóc khách hàng.
- **Quản lý hóa đơn:**
- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến hóa đơn
 - + **Mức độ ưu tiên:** Cao, cần cập nhật quản lý chặt chẽ đảm bảo các thông tin không bị sai sót ảnh hưởng nghiêm trọng đến việc quản lý doanh thu, và các

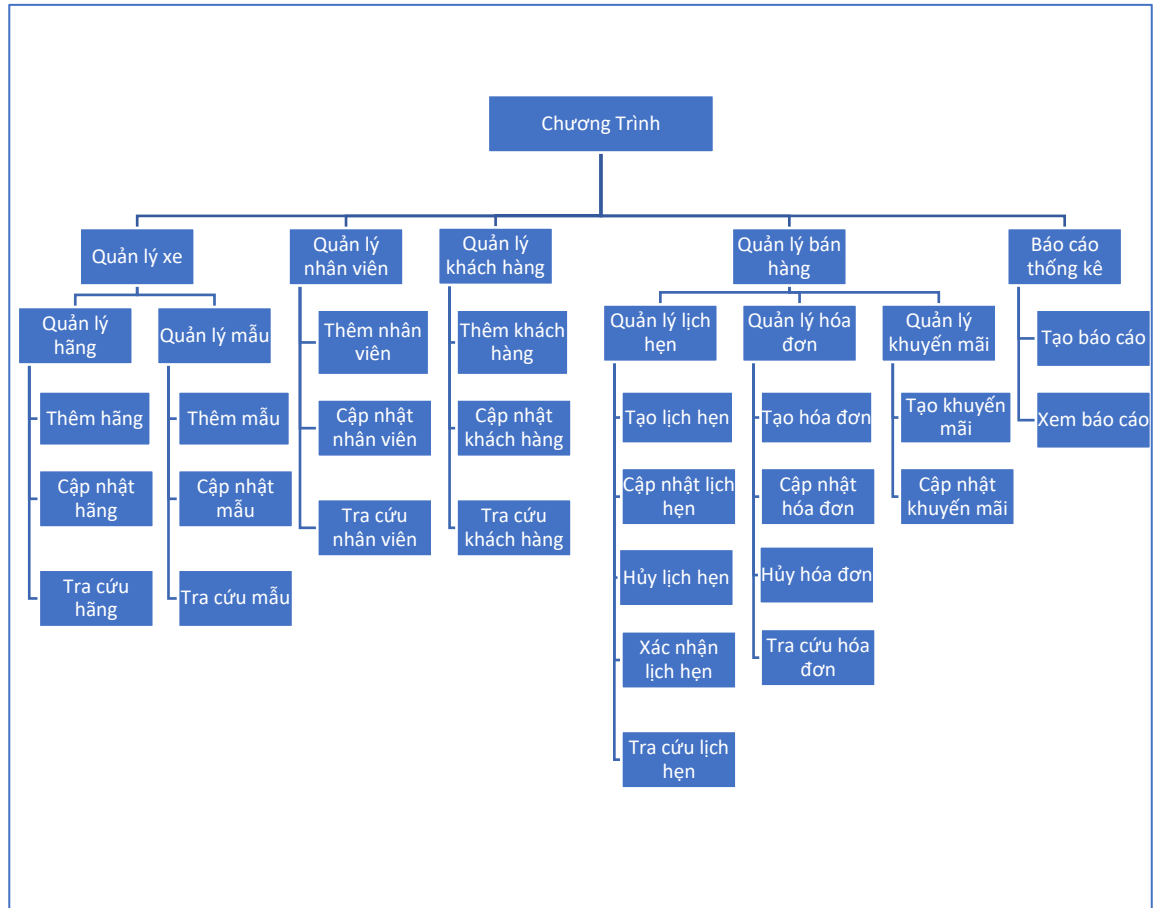
tra cứu truy xuất dữ liệu sau này. Và đảm bảo tính nhất quán về dữ liệu hóa đơn.

- + **Tác nhân:** Người quản lý, nhân viên có thể thêm, cập nhật, bổ sung thay đổi thông tin hóa đơn khi hóa đơn có sự thay đổi về các thông tin liên quan.
 - + **Chuỗi đáp ứng:** Quản lý hóa đơn cần nhập chính xác các thông tin liên quan đến hóa đơn. Có các chức năng chọn thông tin thay cho nhập tay để đảm bảo chính xác về mặt dữ liệu .
 - + **Yêu cầu:** Quản lý hóa đơn cần nhập chính xác đầy đủ các thông tin, số liệu vì sẽ ảnh hưởng đến quản lý hệ thống và dịch vụ chăm sóc khách hàng, nghiêm trọng hơn là doanh thu thất thoát.
- **Quản lý khuyến mãi:**
- + **Mô tả:** Dùng để quản lý các thông tin liên quan đến khuyến mãi.
 - + **Mức độ ưu tiên:** trung bình, cần cập nhật quản lý các thông tin không bị sai sót ảnh hưởng đến việc quản lý doanh thu, và các tra cứu truy xuất dữ liệu sau này. Và đảm bảo tính nhất quán về dữ liệu hóa đơn.
 - + **Tác nhân:** Người quản lý, có thể thêm, cập nhật, bổ sung thay đổi thông tin khuyến mãi khi khuyến mãi có sự thay đổi về các thông tin liên quan.
 - + **Chuỗi đáp ứng:** Quản lý khuyến mãi cần nhập chính xác các thông tin giá trị khuyến mãi.
 - + **Yêu cầu:** Quản lý khuyến mãi cần nhập chính xác đầy đủ các thông tin, số liệu vì sẽ ảnh hưởng đến quản lý hệ thống và doanh thu.

2 Phân tích và thiết kế hệ thống

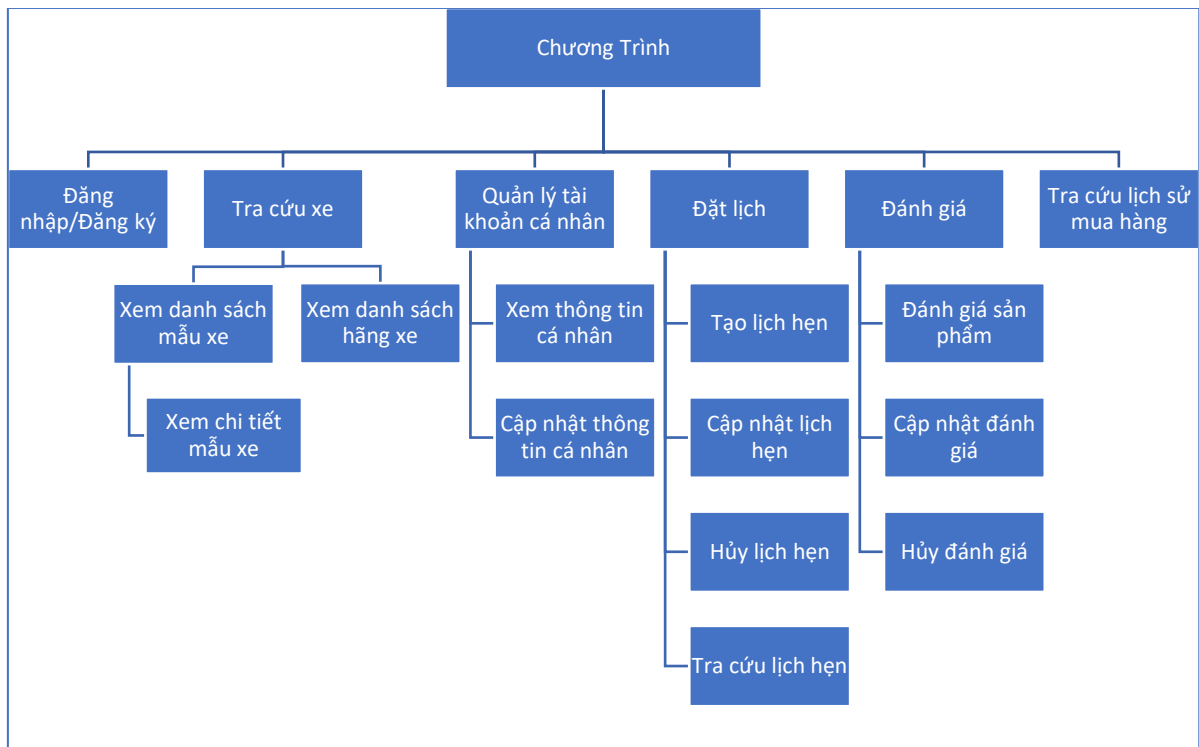
2.1 Sơ đồ phân rã chức năng

2.1.1 IshowRoom



Hình 1: Sơ đồ phân rã chức năng - Quản lý

2.1.2 Khách Hàng



Hình 2: Sơ đồ phân rã chức năng-Khách hàng

2.1.3 Yêu cầu phi chức năng

- **Yêu cầu về an toàn:**

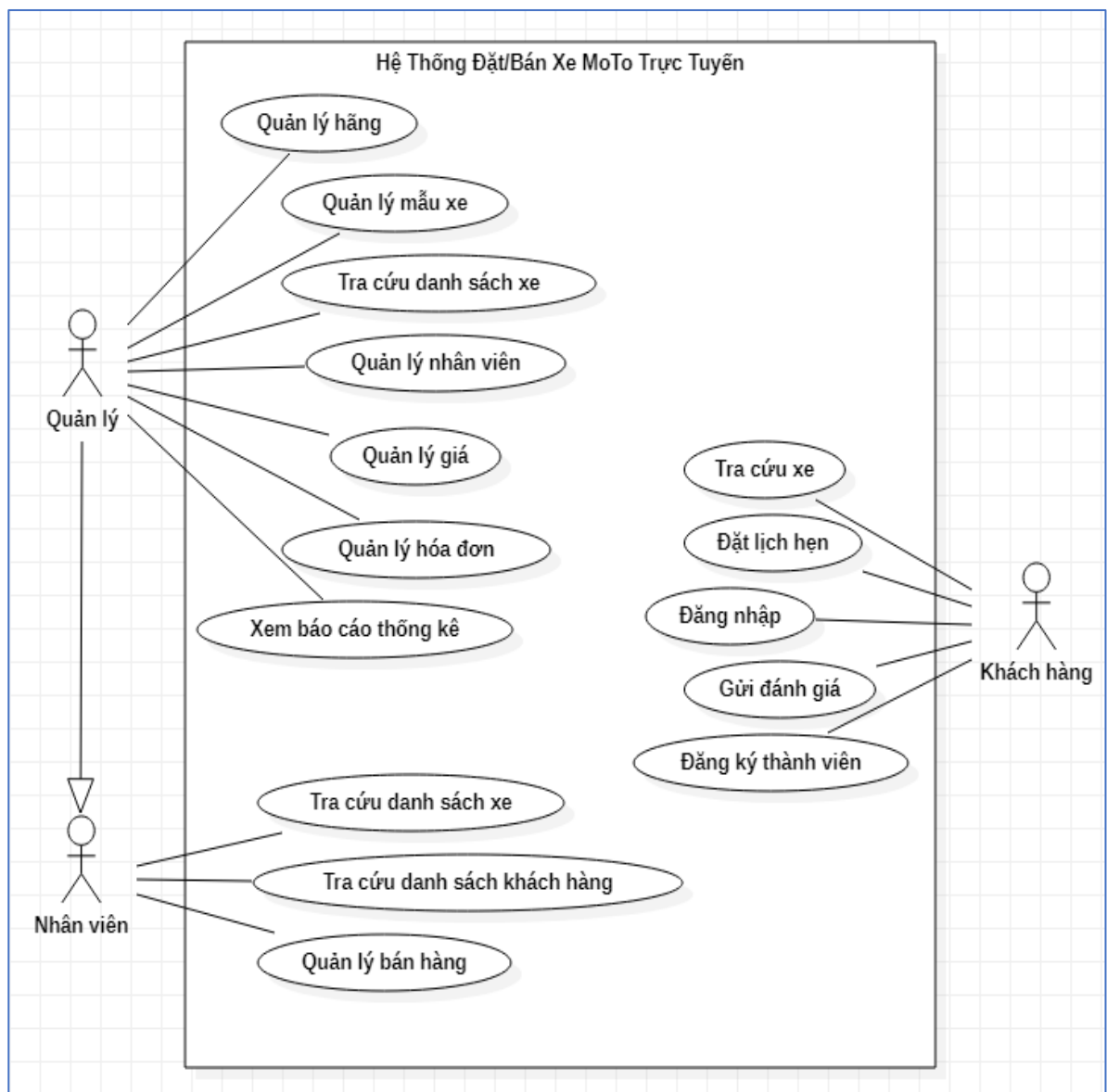
- + Hệ thống không cho xóa tất cả các thông tin mà chỉ cập nhật trạng thái hoặc hủy trên giao diện người dùng.

- **Yêu cầu về bảo mật:**

- + Quản lý mẫu xe: Quản lý
- + Quản lý hãng xe: Quản lý
- + Tra cứu danh sách xe : Quản lý, nhân viên
- + Quản lý giá: Quản lý
- + Tạo hóa đơn: Quản lý, nhân viên
- + Chỉnh sửa, hủy hóa đơn: Quản lý
- + Quản lý nhân viên: Quản lý
- + Xem báo cáo thống kê: Quản lý
- + Đặt, hủy, sửa, tra cứu thông tin lịch hẹn: Nhân viên, khách hàng

- + Xác nhận lịch hẹn: Nhân viên.
- **Yêu cầu về chất lượng phần mềm:**
 - + Dễ sửa lỗi:
 - Xác định lỗi nhanh.
 - Note: Khi sửa thì sửa 1 lỗi chức năng không ảnh hưởng tới các chức năng khác. Không ảnh hưởng tới các chức năng đã có. Với cùng các yêu cầu nghiệp vụ.
 - + Dễ bảo trì:
 - Thêm hoặc thay đổi chức năng mới nhanh.
 - + Tái sử dụng:
 - Xây dựng phần mềm quản lí mới nhanh phát triển từ phần mềm cũ.
 - + Dễ thích ứng:
 - Tức là khi đổi sang hệ quản trị mới hệ thống vẫn hoạt động tốt.
 - + Dễ sử dụng:
 - Phần mềm dễ sử dụng, giao diện thân thiện với người dùng
 - Note: Có phần hướng dẫn sử dụng phần mềm
 - + Tính chính xác:
 - Dữ liệu trong hệ thống và dữ liệu xuất ra cần phải chính xác
 - Note: Dữ liệu được kiểm tra chặt chẽ khi thực hiện các thao tác trên chúng.
 - + Tính linh hoạt:
 - Thao tác giữa các thành phần trong cơ sở dữ liệu cần phải được tương thích với nhau
 - + Tính khả chuyển:
 - Khả năng chạy được trên nhiều hệ thống máy tính hay trên nhiều hệ điều hành
 - Note: Có thể chuyển từ hệ thống nọ sang hệ thống kia mà gần như không phải sửa đổi gì

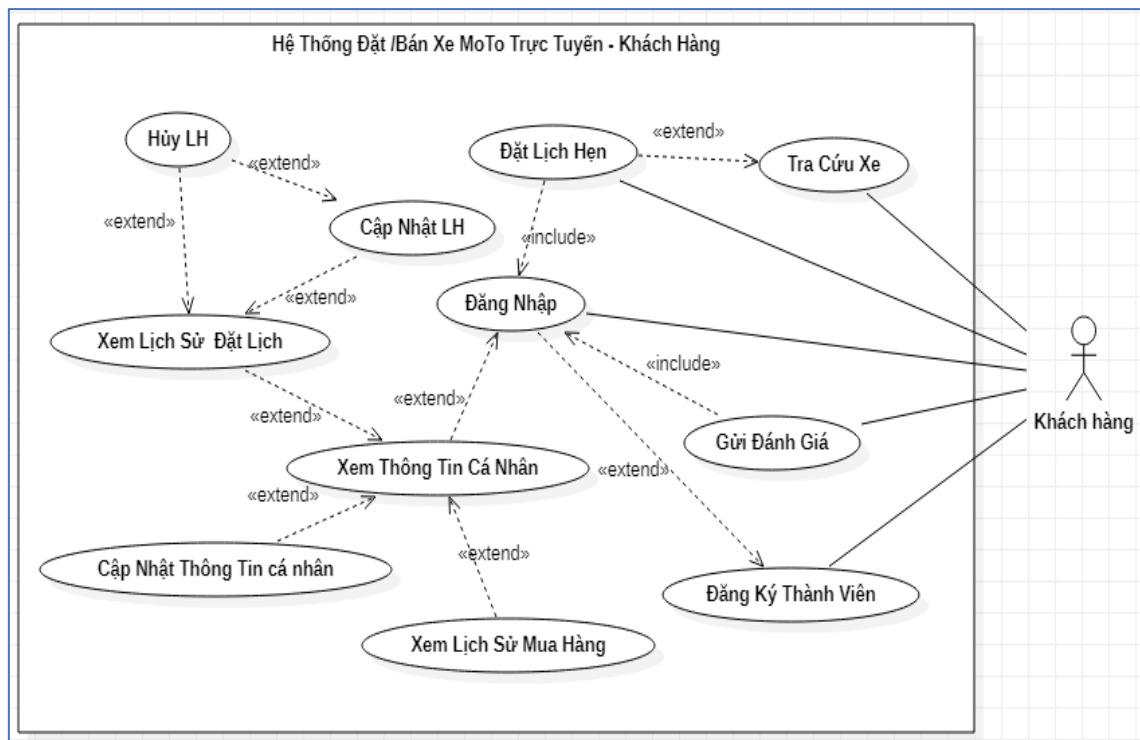
2.2 Sơ đồ



Hình 3: Sơ đồ UseCase tổng quan

2.2.1 Use Case Diagram

2.2.1.1. Mô tả usecase Khách hàng:



Hình 4: Sơ đồ UseCase - Khách hàng

UC01: Tra cứu danh sách xe

Name	Tra cứu danh sách xe		Code	UC01
Description	Dùng để tra cứu danh sách xe hiện có của hệ thống			
Actor	Nhân viên, quản lý	Trigger	Khi actor đăng nhập vào hệ thống hoặc nhấn vào tab Danh sách xe trên menu	
Pre condition	Phải có tài khoản và đăng nhập vào hệ thống			
Post condition	Hiển thị danh sách các xe lên màn hình người dung			
Standard flow/ Process	- Actor đăng nhập vào hệ thống			

	<ul style="list-style-type: none"> - Hệ thống load và hiển thị danh sách xe lên màn hình người dùng
Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm thấy danh sách xe - Lỗi không load được lên view - Không kết nối được với server
System state in error situations	Lỗi khi load danh sách xe

UC02: Cập nhật trạng thái xe

Name	Cập nhật trạng thái xe	Code	UC02
Description	Dùng để cập nhật trạng thái xe thành “đã bán”		
Actor	Nhân viên	Trigger	Khi nhân viên chọn bán và xác nhận bán một xe trong danh sách
Pre condition	Ở màn hình tra cứu danh sách xe và load được danh sách xe		
Post condition	Cập nhật trạng thái xe đã bán		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn một chiếc xe trong danh sách xe - Nhân viên chọn nút bán xe - Nhân viên chọn xác nhận đã hoàn thành thủ tục và bán chiếc này - Hệ thống cập nhật lại trạng thái xe là “đã bán” 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn hủy hoặc không xác nhận 		

	<ul style="list-style-type: none"> - Hiện thị thông báo và trở về màn hình tra cứu, không cập nhật trạng thái xe
Error situations	<ul style="list-style-type: none"> - Không tìm thấy xe trong danh sách xe - Không kết nối được với server - Không cập nhật được trạng thái xe khi xác nhận
System state in error situations	Cập nhật trạng thái xe thất bại

UC03: Quản lý bán hàng

Name	Quản lý bán hàng	Code	UC03
Description	Giúp quản lý các công việc liên quan đến bán hàng		
Actor	Nhân viên	Trigger	Nhân viên chọn vào tab Bán hàng trên hệ thống
Pre condition			
Post condition	Chuyển đến trang quản lý bán hàng		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn vào tab bán hàng trên menu - Hệ thống hiển thị các thông tin, chức năng liên quan 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Lỗi không Load được trang quản lý bán hàng - Không kết nối được với server 		
Error situations	<ul style="list-style-type: none"> - Không tìm thấy xe trong danh sách xe - Không kết nối được với server - Không cập nhật được trạng thái xe khi xác nhận 		

System state in error situations	Lỗi khi load trang quản lý bán hàng
---	-------------------------------------

UC04: Quản lý hóa đơn

Name	Quản lý hóa đơn	Code	UC04
Description	Dùng để thêm xóa sửa tra cứu các thông tin liên quan đến hóa đơn		
Actor	Nhân viên	Trigger	Khi nhân viên chọn mục quản lý hóa đơn trong trang quản lý bán hàng
Pre condition	Đang ở trang quản lý bán hàng		
Post condition	Chuyển sang màn hình quản lý hóa đơn		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên đang ở trang quản lý bán hàng và chọn mục quản lý hóa đơn - Hệ thống chuyển sang trang quản lý hóa đơn - Hệ thống load các thông tin liên quan và hiển thị lên view 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không load được các thông tin hóa đơn - Lỗi không chuyển sang trang quản lý hóa đơn khi nhấn nút - Không kết nối được với server 		
System state in error situations	Quản lý bán hàng không thành công		

UC05: Quản lý lịch hẹn

Name	Quản lý lịch hẹn	Code	UC05
Description	Dùng để thêm, hủy, cập nhật, xác nhận lịch hẹn cho khách hàng		
Actor	Nhân viên	Trigger	Khi nhân viên chọn vào mục lịch hẹn
Pre condition	Phải ở trang quản lý bán hàng		
Post condition	Chuyển sang trang Quản lý lịch hẹn		
Standard flow/ Process	<ul style="list-style-type: none"> - Chọn quản lý lịch hẹn ở trang quản lý bán hàng - Chuyển sang trang quản lý lịch hẹn - Hệ thống load các lịch hẹn hiện có và hiện lên view 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm được lịch hẹn - Hệ thống không chuyển sang màn hình quản lý lịch hẹn - Không kết nối được với server 		
System state in error situations	Quản lý lịch hẹn không thành công.		

UC06: Tạo lịch hẹn

Name	Tạo lịch hẹn	Code	UC06
Description	Dùng để tạo một lịch hẹn mới theo yêu cầu của khách hàng		

Actor	Nhân viên	Trigger	Khi nhân viên chọn tạo lịch hẹn ở quản lý lịch hẹn
Pre condition	Phải ở trang quản lý lịch hẹn		
Post condition	Hiển thị form đặt lịch để điền thông tin		
Standard flow/ Process	<ul style="list-style-type: none"> - Chọn nút tạo lịch hẹn - Hệ thống hiển thị form đặt lịch - Nhân viên điền thông tin và nhấn nút xác nhận - Hệ thống lưu xuống CSDL - Hệ thống trở về màn hình quản lý lịch hẹn - Hệ thống cập nhật lại danh sách lịch hẹn và load lại view 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Nhân viên hủy tác vụ đặt lịch - Hệ thống trở về màn hình quản lý đặt lịch 		
Error situations	<ul style="list-style-type: none"> - Lỗi không hiển thị được form đặt lịch - Lỗi không lưu được lịch đặt - Không kết nối được với server 		
System state in error situations	Đặt lịch thất bại		

UC07: Xem lịch sử lịch hẹn

Name	Xem lịch sử lịch hẹn	Code	UC07
Description	Dùng để tra cứu lịch sử lịch hẹn		
Actor	Nhân viên	Trigger	Khi nhân viên chọn nút lịch sử trong

			giao diện quản lí đặt lịch
Pre condition	Xem lịch sử lịch hẹn		
Post condition	Chuyển sang màn hình lịch sử lịch hẹn		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn nút lịch sử trong giao diện quản lí đặt lịch - Hệ thống load tất cả các lịch hẹn đã hoàn thành - Hiện thị danh sách lịch hẹn 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm được lịch sử đặt lịch hẹn - Lỗi không load được lịch hẹn - Không kết nối được với server 		
System state in error situations	Xem lịch sử thất bại		

UC08: Xác nhận lịch hẹn

Name	Xác nhận lịch hẹn	Code	UC08
Description	Dùng để xác nhận lịch hẹn của khách hàng		
Actor	Nhân viên	Trigger	Khi nhân viên nhấn xác nhận lịch hẹn
Pre condition	Phải ở trang quản lý lịch hẹn và có lịch hẹn ở trạng thái mới		
Post condition	Thay đổi trạng thái lịch hẹn		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn vào lịch hẹn mới 		

	<ul style="list-style-type: none"> - Nhấn xác nhận lịch hẹn - Hệ thống thay đổi trạng thái lịch hẹn “đã xác nhận”
Alternative flow/ Process	<ul style="list-style-type: none"> - Nhân viên hủy tác vụ - Hệ thống trở về trang quản lý lịch hẹn và không thay đổi trạng thái lịch hẹn.
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm được lịch hẹn - Lỗi không thay đổi được trạng thái lịch hẹn - Không kết nối được với server
System state in error situations	Xác nhận lịch hẹn thất bại

UC09: Tra cứu Lịch hẹn

Name	Tra cứu lịch hẹn	Code	UC09
Description	Dùng để tra cứu các lịch hẹn theo trạng thái, ngày tháng, vv		
Actor	Nhân viên	Trigger	Nhân viên tra cứu lịch hẹn
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình lịch sử mua hàng		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn xem lịch sử mua hàng - Chọn mẫu xe muốn đánh giá - Chọn gửi đánh giá - Nhập đánh giá và xác nhận gửi đánh giá 		

	<ul style="list-style-type: none"> - Hệ thống thông báo đánh giá thành công và chuyển màn hình lịch sử mua hàng
Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Chọn gửi đánh giá nhưng hệ thống không chuyển màn hình đánh giá - Không kết nối được với server
System state in error situations	Không đánh giá được sản phẩm

UC10: Xem Thông Tin Cá Nhân

Name	Xem thông tin cá nhân	Code	UC10
Description	Dùng để xem thông tin cá nhân của khách thành viên		
Actor	Khách thành viên	Trigger	Khi Khách thành viên chọn xem thông tin cá nhân
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình tra cứu xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn xem thông tin cá nhân 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Chọn xem thông tin cá nhân nhưng hệ thống không chuyển màn hình thông tin cá nhân khách hàng 		

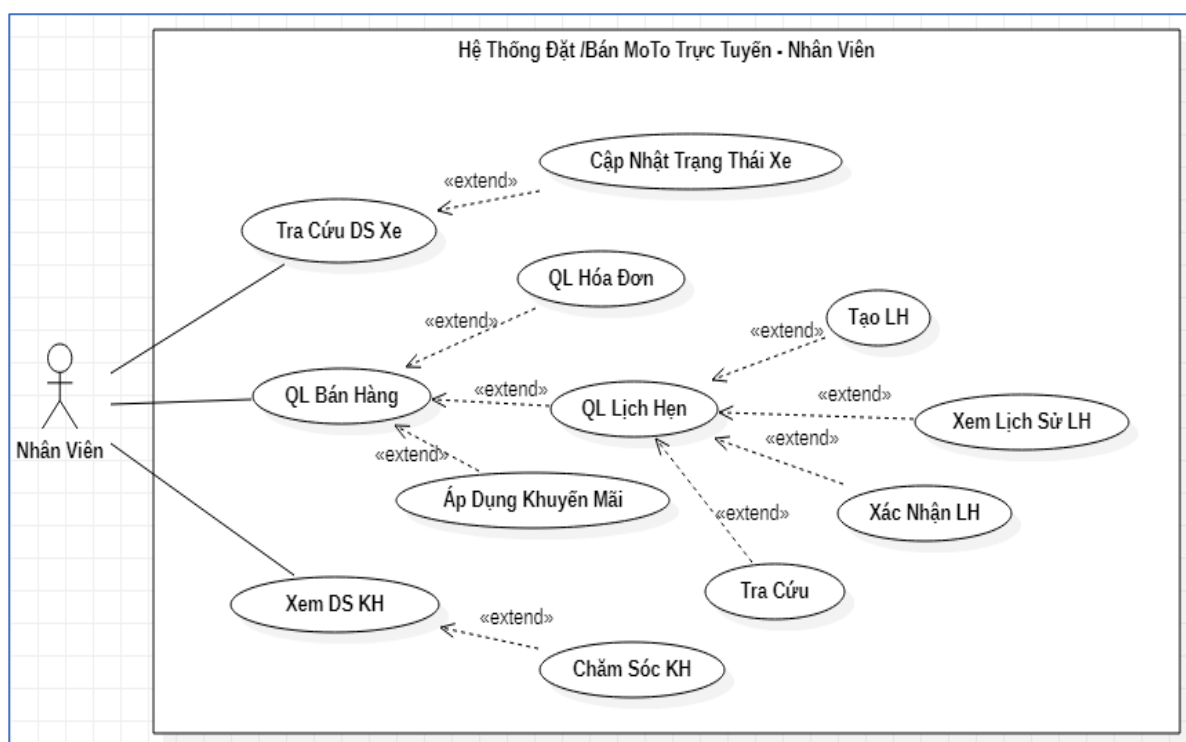
	- Không kết nối được với server
System state in error situations	Không xem được thông tin cá nhân

UC11: Cập Nhật Thông Tin Cá Nhân

Name	Cập nhật thông tin cá nhân	Code	UC11
Description	Dùng để cập nhật thông tin cá nhân của khách thành viên		
Actor	Khách thành viên	Trigger	Khi Khách thành viên chọn cập nhật thông tin cá nhân
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình thông tin khách thành viên		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn xem thông tin cá nhân - Chọn cập nhật thông tin cá nhân - Khách điền thông tin cập nhật thông tin cá nhân - Khách xác nhận cập nhật thông tin - Hệ thống thông báo cập nhật thông tin thành công và chuyển sang màn hình thông tin cá nhân 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Chọn cập nhật thông tin cá nhân nhưng hệ thống không chuyển màn hình cập nhật 		

	<ul style="list-style-type: none"> - Xác nhận cập nhật thông tin cá nhân nhưng hệ thống không thay đổi thông tin mới cập nhật - Không kết nối được với server
System state in error situations	Không cập nhật được thông tin cá nhân

2.2.2.2. Mô tả usecase Nhân viên:



Hình 5: Sơ đồ UseCase - Nhân viên

UC12: Thêm hóa đơn

Name	Thêm hóa đơn	Code	UC12
Description	Dùng để thêm hóa đơn mới		
Actor	Nhân viên	Trigger	Khi nhân viên chọn thêm hóa đơn

Pre condition	Đang ở trang quản lý hóa đơn
Post condition	Thêm một hóa đơn mới vào hệ thống và trở về trang quản lý hóa đơn
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên nhấn thêm hóa đơn - Hệ thống hiện form - Nhân viên điền form thêm hóa đơn mới - Nhân viên nhấn xác nhận thêm - Hệ thống kiểm tra Validation - Hệ thống thêm hóa đơn mới xuống DB - Trở về trang quản lý hóa đơn
Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép tạo hóa đơn mới
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được hóa đơn mới xuống DB
System state in error situations	Thêm hóa đơn mới không thành công

UC13: Tra cứu hóa đơn

Name	Tra cứu hóa đơn	Code	UC13
Description	Dùng để tra cứu, tìm kiếm hóa đơn		
Actor	Nhân viên	Trigger	Nhân viên nhập thông tin muốn tìm kiếm của hóa đơn

Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản nhân viên hoặc quản lý
Post condition	Hiển thị danh sách hóa đơn thỏa yêu cầu
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên nhập thông tin muốn tìm vào search box - Hệ thống lọc danh sách theo thông tin nhập vào - Hệ thống hiển thị danh sách hóa đơn đã lọc
Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Lỗi không lọc được dữ liệu - Lỗi không hiển thị danh sách hóa đơn sau khi lọc - Không kết nối được với server
System state in error situations	Tra cứu hóa đơn thất bại

UC14: Xem danh sách khách hàng

Name	Xem danh sách khách hàng	Code	UC14
Description	Dùng để xem thông tin, danh sách khách hàng		
Actor	Nhân viên	Trigger	Khi nhân viên chọn vào tab khách hàng trên hệ thống
Pre condition	Đăng nhập vào hệ thống và đã có khách hàng		
Post condition	Chuyển sang trang danh sách khách hàng		

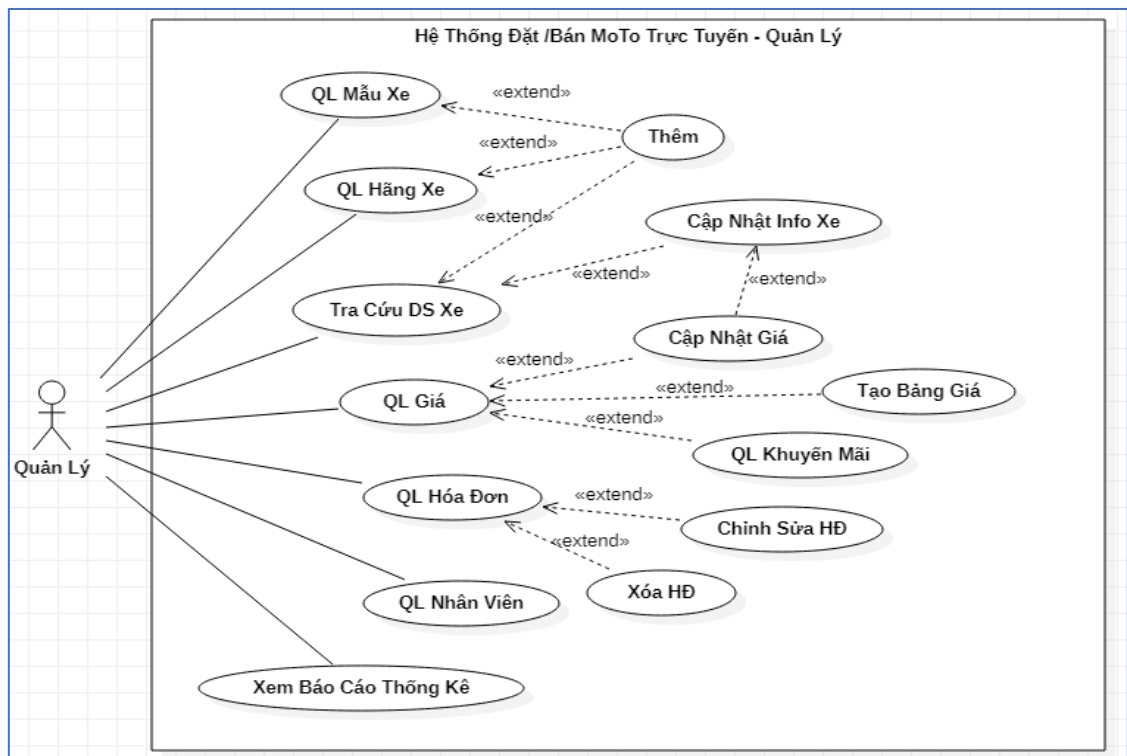
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn tab khách hàng - Hệ thống load thông tin khách hàng cần thiết - Hệ thống hiển thị danh sách khách hàng lên view
Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm thấy khách hàng - Lỗi không toad được thông tin khách hàng - Lỗi không hiển thị được danh sách khách hàng - Không kết nối được với server
System state in error situations	Xem danh sách khách hàng thất bại

UC15: Cập Nhật Loại khách hàng

Name	Cập nhật loại khách hàng	Code	UC15
Description	Dùng để cập nhật loại khách hàng khi khách hàng đủ tiêu chí		
Actor	Nhân viên	Trigger	Khi nhân viên chọn vào 1 khách hàng và chọn cập nhật loại khách hàng
Pre condition	Phải ở màn hình Xem danh sách khách hàng		
Post condition	Cập nhật loại khách hàng và trở về trang danh sách khách hàng		
Standard flow/ Process	<ul style="list-style-type: none"> - Nhân viên chọn khách hàng 		

	<ul style="list-style-type: none"> - Chọn cập nhật loại khách hàng - Chọn loại khách hàng muốn cập nhật - Hệ thống kiểm tra tất cả điều kiện và tiêu chí - Hệ thống cập nhật lại loại khách hàng - Hệ thống thông báo cập nhật loại khách hàng thành công và trở về trang danh sách khách hàng
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống kiểm tra khách hàng không đủ tiêu chí để cập nhật loại khách hàng đó. - Thông báo khách hàng không đủ điều kiện - Hủy tác vụ và trở về trang danh sách khách hàng
Error situations	<ul style="list-style-type: none"> - Lỗi khi cập nhật loại khách hàng - Lỗi khi kiểm tra điều kiện khách hàng - Không kết nối được với server - Lỗi không tìm thấy thông tin khách hàng
System state in error situations	Lỗi khi cập nhật loại khách hàng

2.2.2.3. Mô tả usecase Quản lý



Hình 6: Sơ đồ UseCase - Quản lý

UC16: Thêm mẫu xe

Name	Thêm mẫu xe	Code	UC16
Description	Dùng để thêm mẫu xe mới		
Actor	Quản lý	Trigger	Quản lý chọn thêm mẫu xe ở trang QL mẫu xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Thêm một mẫu xe mới vào hệ thống và trở về trang quản lý mẫu xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn thêm mẫu xe mới - Hệ thống hiện form - Quản lý điền form thêm mẫu xe mới 		

	<ul style="list-style-type: none"> - Quản lý nhấn xác nhận thêm - Hệ thống kiểm tra Validation - Hệ thống thêm mẫu xe mới xuống DB - Trở về trang quản lý mẫu xe
Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép tạo mẫu xe mới
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được mẫu xe mới xuống DB
System state in error situations	Thêm Mẫu xe mới thất bại

UC17: Xóa Mẫu xe

Name	Xóa mẫu xe	Code	UC17
Description	Dùng để xóa mẫu xe không còn kinh doanh		
Actor	Quản lý	Trigger	Quản lý chọn thêm mẫu xe ở trang QL mẫu xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Xóa Mẫu xe đã chọn		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn xóa mẫu xe trong danh sách - Hệ thống kiểm tra validation - Quản lý nhấn xác nhận xóa - Hệ thống xóa mẫu xe đã chọn 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Xóa mẫu xe không hợp lệ - Hiện thị thông báo lỗi và không cho xóa mẫu xe.
Error situations	<ul style="list-style-type: none"> - Không tìm thấy mẫu xe - Không kết nối được với server - Không xóa được mẫu xe - Không kiểm tra được điều kiện
System state in error situations	Xóa mẫu xe thất bại.

UC18: Cập nhật mẫu xe

Name	Cập nhật mẫu xe	Code	UC18
Description	Giúp cập nhật thông tin mẫu xe		
Actor	Quản lý	Trigger	Quản lý chọn Cập nhật trong giao diện quản lý mẫu
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Cập nhật lại thông tin mẫu xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn cập nhật mẫu xe trong danh sách - Hệ thống hiện form và load thông tin mẫu xe lên form - Quản lý sửa các thông tin muốn cập nhật - Quản lý nhấn xác nhận cập nhật - Hệ thống kiểm tra Validation - Hệ thống cập nhật mẫu xe xuống DB - Trở về trang quản lý mẫu xe 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép cập nhật mẫu xe
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không load được dữ liệu cũ lên form - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được cập nhật mới xuống DB
System state in error situations	Cập nhật mẫu xe không thành công

UC19: Tra cứu mẫu xe

Name	Tra cứu mẫu xe	Code	UC19
Description	Dùng để tra cứu, tìm kiếm mẫu xe		
Actor	Quản lý	Trigger	Quản lý nhập thông tin muốn tìm kiếm của mẫu xe
Pre condition	<ul style="list-style-type: none"> - Phải đăng nhập vào hệ thống bằng tài khoản quản lý 		
Post condition	<ul style="list-style-type: none"> - Hiện thị danh sách mẫu xe thỏa yêu cầu 		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhập thông tin muốn tìm vào search box - Hệ thống lọc danh sách theo thông tin nhập vào - Hệ thống hiện thị danh sách mẫu xe đã lọc 		

Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Lỗi không lọc được dữ liệu - Lỗi không hiển thị danh sách mẫu xe sau khi lọc - Không kết nối được với server
System state in error situations	Tra cứu mẫu xe thất bại

UC20: Thêm hãng xe

Name	Thêm hãng xe	Code	UC20
Description	Dùng để thêm hãng xe mới		
Actor	Quản lý	Trigger	Quản lý chọn thêm hãng xe ở trang QL hãng xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Thêm một hãng xe mới vào hệ thống và trở về trang quản lý hãng xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn thêm hãng xe mới - Hệ thống hiện form - Quản lý điền form thêm hãng xe mới - Quản lý nhấn xác nhận thêm - Hệ thống kiểm tra Validation - Hệ thống thêm hãng xe mới xuống DB - Trở về trang quản lý mẫu xe 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép tạo hãng xe mới
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được hãng xe mới xuống DB
System state in error situations	Thêm hãng xe mới thất bại

UC21: Xóa hãng xe

Name	Xóa hãng xe	Code	UC21
Description	Dùng để xóa hãng xe không còn kinh doanh		
Actor	Quản lý	Trigger	Khi Quản lý chọn xóa hãng xe trong giao diện quản lý hãng
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Xóa hãng xe đã chọn		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn xóa hãng xe trong danh sách - Hệ thống kiểm tra validation - Quản lý nhấn xác nhận xóa - Hệ thống xóa hãng xe đã chọn 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Xóa hãng xe không hợp lệ 		

	<ul style="list-style-type: none"> - Hiện thị thông báo lỗi và không cho xóa hãng xe.
Error situations	<ul style="list-style-type: none"> - Không tìm thấy hãng xe - Không kết nối được với server - không xóa được hãng xe - Không kiểm tra được điều kiện
System state in error situations	Xóa hãng xe thất bại.

UC22: Cập nhật hãng xe

Name	Cập nhật hãng xe	Code	UC22
Description	Giúp cập nhật thông tin hãng xe		
Actor	Quản lý	Trigger	Quản lý chọn Cập nhật trong giao diện quản lý hãng
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Cập nhật lại thông tin hãng xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn cập nhật hãng xe trong danh sách - Hệ thống hiện form và load thông tin hãng xe lên form - Quản lý sửa các thông tin muốn cập nhật - Quản lý nhấn xác nhận cập nhật - Hệ thống kiểm tra Validation - Hệ thống cập nhật hãng xe xuống DB - Trở về trang quản lý mẫu xe 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép cập nhật hãng xe
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không load được dữ liệu cũ lên form - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được cập nhật mới xuống DB
System state in error situations	Cập nhật hãng xe không thành công

UC23: Tra cứu hãng xe

Name	Tra cứu hãng xe	Code	UC23
Description	Dùng để tra cứu, tìm kiếm hãng xe		
Actor	Quản lý	Trigger	Quản lý nhập thông tin muốn tìm kiếm của hãng xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Hiện thị danh sách hãng xe thỏa yêu cầu		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhập thông tin muốn tìm vào search box - Hệ thống lọc danh sách theo thông tin nhập vào - Hệ thống hiện thị danh sách hãng xe đã lọc 		
Alternative flow/ Process			

Error situations	<ul style="list-style-type: none"> - Lỗi không lọc được dữ liệu - Lỗi không hiển thị danh sách hãng xe sau khi lọc - Không kết nối được với server
System state in error situations	Tra cứu hãng xe thất bại

UC24: Thêm giá xe

Name	Thêm giá xe	Code	UC24
Description	Dùng để thêm giá xe mới		
Actor	Quản lý	Trigger	Quản lý chọn thêm giá xe ở trang QL giá xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Thêm một giá xe mới vào hệ thống và trở về trang quản lý giá xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn thêm giá xe mới - Hệ thống hiện form - Quản lý điền form thêm giá xe mới - Quản lý nhấn xác nhận thêm - Hệ thống kiểm tra Validation - Hệ thống thêm giá xe mới xuống DB - Trở về trang quản lý giá xe 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép tạo giá xe mới 		
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không kiểm tra được Validation - Ko kết nối được với server - Lỗi không lưu được giá xe mới xuống DB 		
System state in error situations	Thêm giá xe mới thất bại		

UC25: Xóa giá xe

Name	Xóa giá xe	Code	UC25
Description	Dùng để xóa giá xe không còn phù hợp		
Actor	Quản lý	Trigger	Khi Quản lý chọn xóa giá xe trong giao diện quản lý giá xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Xóa giá xe đã chọn		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn xóa giá xe trong danh sách - Hệ thống kiểm tra validation - Quản lý nhấn xác nhận xóa - Hệ thống xóa giá xe đã chọn 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Xóa giá xe không hợp lệ - Hiện thị thông báo lỗi và không cho xóa giá xe. 		
Error situations	<ul style="list-style-type: none"> - Không tìm thấy giá xe - Không kết nối được với server - Không xóa được giá xe - Không kiểm tra được điều kiện 		
System state in error situations	Xóa giá xe thất bại.		

UC26: Cập nhật giá xe

Name	Cập nhật giá xe	Code	UC26
Description	Giúp cập nhật thông tin giá xe		
Actor	Quản lý	Trigger	Quản lý chọn Cập nhật trong giao diện quản lý giá
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Cập nhật lại thông tin giá xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn cập nhật giá xe trong danh sách - Hệ thống hiện form và load thông tin giá xe lên form - Quản lý sửa các thông tin muốn cập nhật - Quản lý nhấn xác nhận cập nhật - Hệ thống kiểm tra Validation - Hệ thống cập nhật giá xe xuống DB - Trở về trang quản lý giá xe 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép cập nhật giá xe 		
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không load được dữ liệu cũ lên form - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được cập nhật mới xuống DB 		
System state in error situations	Cập nhật giá xe không thành công		

UC27: Tra cứu giá xe

Name	Tra cứu giá xe	Code	UC27
Description	Dùng để tra cứu, tìm kiếm giá xe		
Actor	Quản lý	Trigger	Quản lý nhập giá muốn tìm kiếm của xe
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Hiện thị danh sách hãng xe thỏa yêu cầu		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhập giá muốn tìm vào search box - Hệ thống lọc danh sách theo giá nhập vào - Hệ thống hiện thị danh sách xe có giá tương ứng 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không lọc được dữ liệu - Lỗi không hiện thị danh sách xe sau khi lọc - Không kết nối được với server 		
System state in error situations	Tra cứu giá xe thất bại		

UC28: Tạo bảng giá

Name	Tạo bảng giá	Code	UC28
Description	Dùng để tạo bảng giá xe thời điểm hiện tại		

Actor	Quản lý	Trigger	Quản lý nhấn nút tạo bảng giá trong giao diện quản lý giá
Pre condition	Phải đăng nhập vào hệ thống bằng tài khoản quản lý		
Post condition	Hiển thị danh sách, bảng giá		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn nút tạo bảng giá - Hệ thống tổng hợp các giá xe theo mẫu, hãng - Hệ thống hiển thị một bảng giá tổng hợp 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi khi tổng hợp dữ liệu - Lỗi không hiển thị bảng giá - Không kết nối được với server 		
System state in error situations	Tạo bảng giá thất bại		

UC29: Cập nhật giá

Name	Cập nhật giá	Code	UC29
Description	Dùng để cập nhật lại giá xe		
Actor	Quản lý	Trigger	Quản lý nhấn nút cập nhật giá
Pre condition	Ở trang quản lý giá hoặc trang cập nhật info xe		
Post condition	Thay đổi, cập nhật lại giá xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn nút cập nhật giá 		

	<ul style="list-style-type: none"> - Hệ thống hiển thị form và load giá cũ lên form - Quản lý điền giá mới - Hệ thống kiểm tra validation - Hệ thống cập nhật lại giá mới xuống CSDL
Alternative flow/ Process	<ul style="list-style-type: none"> - Dữ liệu không phù hợp validation - Thông báo lỗi và không cho lưu giá mới
Error situations	<ul style="list-style-type: none"> - Lỗi không load được giá cũ - Lỗi không hiển thị được form - Không kết nối được với server - Lỗi không lưu được giá mới xuống CSDL
System state in error situations	Cập nhật giá thất bại

UC30: Cập nhật info xe

Name	Cập nhật info xe	Code	UC30
Description	Dùng để cập nhật lại info xe		
Actor	Quản lý	Trigger	Quản lý nhấn nút cập nhật info xe
Pre condition	Ở trang tra cứu danh sách xe		
Post condition	Thay đổi, cập nhật lại info xe		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn nút cập nhật info xe - Hệ thống hiển thị form và load info xe cũ lên form - Quản lý điền info xe mới - Hệ thống kiểm tra validation - Hệ thống cập nhật lại info xe mới xuống CSDL 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Dữ liệu không phù hợp validation - Thông báo lỗi và không cho lưu info xe mới
Error situations	<ul style="list-style-type: none"> - Lỗi không load được info xe cũ - Lỗi không hiển thị được form - Không kết nối được với server - Lỗi không lưu được info xe mới xuống CSDL
System state in error situations	Cập nhật info xe thất bại

UC31: Thêm xe

Name	Thêm xe	Code	UC31
Description	Dùng để thêm xe mới		
Actor	Quản lý	Trigger	Quản lý chọn thêm xe
Pre condition	Phải ở trang QL mẫu xe hoặc trang QL hãng xe hoặc trang tra cứu ds xe		
Post condition	Thêm một xe mới vào hệ thống và trở về trang quản lý trước đó		
Standard flow/ Process	<ul style="list-style-type: none"> - Quản lý nhấn thêm xe mới - Hệ thống hiện form - Quản lý điền form thêm xe mới - Quản lý nhấn xác nhận thêm - Hệ thống kiểm tra Validation - Hệ thống thêm xe mới xuống DB - Trở về trang quản lý trước đó 		

Alternative flow/ Process	<ul style="list-style-type: none"> - Validation không hợp lệ - Hiện thông báo lên người dùng và không cho phép tạo xe mới
Error situations	<ul style="list-style-type: none"> - Lỗi không hiện form nhập dữ liệu - Lỗi không kiểm tra được Validation - Không kết nối được với server - Lỗi không lưu được xe mới xuống DB
System state in error situations	Thêm xe mới thất bại

UC32: Tra cứu nhân viên

Name	Tra cứu nhân viên	Code	UC32
Description	Hiển thị danh sách nhân viên		
Actor	Quản lý	Trigger	Khi quản lý chọn tra cứu danh sách nhân viên
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình thông tin chi tiết nhân viên		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Quản lý chọn xem danh sách nhân viên - Hệ thống hiển thị danh sách cùng thông tin cơ bản của nhân viên - Quản lý chọn nhân viên cần xem - Hệ thống chuyển sang màn hình chi tiết nhân viên 		

Alternative flow/ Process	
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm thấy danh sách nhân viên - Chọn xem chi tiết nhân viên nhưng không chuyển màn hình thông tin chi tiết nhân viên - Không kết nối được với server
System state in error situations	Không tra cứu được danh sách nhân viên

UC33: Thêm nhân viên

Name	Thêm nhân viên	Code	UC33
Description	Dùng để thêm nhân viên mới		
Actor	Quản Lý	Trigger	Khi quản lý chọn thêm nhân viên
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách nhân viên		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách nhân viên - Chọn thêm nhân viên - Quản lý điền thông tin cho nhân viên mới - Quản lý xác nhận thêm - Hệ thống thông báo thêm nhân viên thành công và chuyển sang màn hình danh sách nhân viên 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo thêm thất bại - Hệ thống chuyển màn hình danh sách nhân viên 		

Error situations	<ul style="list-style-type: none"> - Chọn thêm nhân viên nhưng hệ thống không chuyển màn hình thêm - Xác nhận thêm nhân viên nhưng hệ thống không thêm nhân viên - Không kết nối được với server
System state in error situations	Thêm xe nhân viên mới thất bại

UC34: Xóa nhân viên

Name	Xóa nhân viên	Code	UC34
Description	Dùng để xóa nhân viên mới		
Actor	Quản Lý	Trigger	Khi quản lý chọn xóa nhân viên
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách nhân viên		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách nhân viên - Chọn xóa nhân viên - Quản lý xác nhận xóa - Hệ thống thông báo xóa nhân viên thành công và chuyển sang màn hình danh sách nhân viên 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo xóa thất bại - Hệ thống chuyển màn hình danh sách nhân viên 		
Error situations	<ul style="list-style-type: none"> - Chọn xóa nhân viên nhưng hệ thống không hiện thông báo xác nhận 		

	<ul style="list-style-type: none"> - Xác nhận thêm nhân viên nhưng hệ thống không thêm nhân viên - Không kết nối được với server
System state in error situations	Xóa nhân viên thất bại

UC35: Cập nhật thông tin nhân viên

Name	Cập nhật thông tin nhân viên	Code	UC35
Description	Dùng để cập nhật thông tin cá nhân của nhân viên		
Actor	Quản Lý	Trigger	Khi quản lý chọn cập nhật thông tin nhân viên
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách nhân viên		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách nhân viên - Chọn cập nhật thông tin nhân viên - Quản lý điền thông tin cập nhật cho nhân viên - Quản lý xác nhận cập nhật thông tin - Hệ thống thông báo cập nhật thông tin thành công và chuyển sang màn hình thông tin nhân viên 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo cập nhật thất bại - Hệ thống chuyển màn hình danh sách nhân viên 		

Error situations	<ul style="list-style-type: none"> - Chọn cập nhật thông tin nhân viên nhưng hệ thống không chuyển màn hình cập nhật - Xác nhận cập nhật thông tin nhân viên nhưng hệ thống không thay đổi thông tin mới cập nhật - Không kết nối được với server
System state in error situations	Không cập nhật được thông tin nhân viên

UC36: Tra cứu khuyến mãi

Name	Tra cứu khuyến mãi	Code	UC36
Description	Hiển thị danh sách khuyến mãi		
Actor	Quản lý	Trigger	Khi quản lý chọn tra cứu danh sách khuyến mãi
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình thông tin chi tiết khuyến mãi		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Quản lý chọn xem danh sách khuyến mãi - Hệ thống hiển thị danh sách khuyến mãi - Quản lý chọn khuyến mãi cần xem - Hệ thống chuyển sang màn hình chi tiết khuyến mãi 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm thấy danh sách khuyến mãi - Chọn xem chi tiết khuyến mãi nhưng không chuyển màn hình thông tin chi tiết khuyến mãi - Không kết nối được với server 		
System state in error situations	Không tra cứu được danh sách khuyến mãi		

UC37: Thêm khuyến mãi

Name	Thêm khuyến mãi	Code	UC37
Description	Dùng để thêm khuyến mãi		
Actor	Quản Lý	Trigger	Khi quản lý chọn thêm khuyến mãi
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách khuyến mãi		
Standard flow/ Process	<ul style="list-style-type: none">- Đăng nhập thành công vào hệ thống- Chọn tra cứu danh sách khuyến mãi- Chọn thêm khuyến mãi- Quản lý điền thông tin cho khuyến mãi mới- Quản lý xác nhận thêm- Hệ thống thông báo thêm khuyến mãi thành công và chuyển sang màn hình danh sách khuyến mãi		
Alternative flow/ Process	<ul style="list-style-type: none">- Hệ thống báo thêm thất bại- Hệ thống chuyển màn hình danh sách khuyến mãi		
Error situations	<ul style="list-style-type: none">- Chọn thêm khuyến mãi nhưng hệ thống không chuyển màn hình thêm khuyến mãi- Xác nhận thêm nhân viên nhưng hệ thống không thêm khuyến mãi- Không kết nối được với server		

UC38: Xóa khuyến mãi

Name	Xóa khuyến mãi	Code	UC38
Description	Dùng để xóa nhân viên mới		
Actor	Quản Lý	Trigger	Khi quản lý chọn xóa khuyến mãi
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách khuyến mãi		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách khuyến mãi - Chọn xóa khuyến mãi - Quản lý xác nhận xóa - Hệ thống thông báo xóa khuyến mãi thành công và chuyển sang màn hình danh sách khuyến mãi 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo xóa thất bại - Hệ thống chuyển màn hình danh sách khuyến mãi 		
Error situations	<ul style="list-style-type: none"> - Chọn xóa nhưng hệ thống không hiện thông báo xác nhận xóa - Xác nhận xóa khuyến mãi nhưng hệ thống không xóa khuyến mãi - Không kết nối được với server 		

UC39: Cập nhật khuyến mãi

Name	Cập nhật khuyến mãi	Code	UC39
Description	Dùng để cập nhật giá trị khuyến mãi		
Actor	Quản Lý	Trigger	Khi quản lý chọn cập nhật khuyến mãi
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách khuyến mãi		
Standard flow/ Process	<ul style="list-style-type: none">- Đăng nhập thành công vào hệ thống- Chọn tra cứu danh sách khuyến mãi- Chọn cập nhật khuyến mãi- Quản lý điền thông tin cập nhật cho khuyến mãi- Quản lý xác nhận cập nhật- Hệ thống thông báo cập nhật khuyến mãi thành công và chuyển sang màn hình danh sách khuyến mãi		
Alternative flow/ Process	<ul style="list-style-type: none">- Hệ thống báo cập nhật thất bại- Hệ thống chuyển màn hình danh sách khuyến mãi		
Error situations	<ul style="list-style-type: none">- Chọn cập nhật khuyến mãi nhưng hệ thống không chuyển màn hình cập nhật- Xác nhận cập nhật khuyến mãi nhưng hệ thống không thay đổi thông tin mới cập nhật- Không kết nối được với server		

System state in error situations	Không cập nhật được khuyến mãi
---	--------------------------------

UC40: Tra cứu hóa đơn

Name	Tra cứu hóa đơn	Code	UC40
Description	Hiển thị danh sách khuyến mãi		
Actor	Quản lý	Trigger	Khi quản lý chọn tra cứu danh sách hóa đơn
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình thông tin chi tiết hóa đơn		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Quản lý chọn xem danh sách hóa đơn - Hệ thống hiển thị danh sách hóa đơn - Quản lý chọn hóa đơn cần xem - Hệ thống chuyển sang màn hình chi tiết hóa đơn 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Lỗi không tìm thấy danh sách hóa đơn - Chọn xem chi tiết hóa đơn nhưng không chuyển màn hình thông tin chi tiết hóa đơn - Không kết nối được với server 		
System state in error situations	Không tra cứu được danh sách hóa đơn		

UC41: Xóa hóa đơn

Name	Xóa hóa đơn	Code	UC41
Description	Dùng để xóa nhân viên mới		
Actor	Quản Lý	Trigger	Khi quản lý chọn xóa hóa đơn
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách hóa đơn		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách hóa đơn - Chọn xóa hóa đơn - Quản lý xác nhận xóa - Hệ thống thông báo xóa hóa đơn thành công và chuyển sang màn hình danh sách hóa đơn 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo xóa thất bại - Hệ thống chuyển màn hình danh sách hóa đơn 		
Error situations	<ul style="list-style-type: none"> - Chọn xóa hóa đơn nhưng hệ thống không hiện thông báo xác nhận - Xác nhận xóa hóa đơn nhưng hệ thống không xóa hóa đơn - Không kết nối được với server 		

UC42: Cập nhật thông tin hóa đơn

Name	Cập nhật thông tin hóa đơn	Code	UC42
Description	Dùng để cập nhật thông tin cá nhân của nhân viên		
Actor	Quản Lý	Trigger	Khi quản lý chọn cập nhật thông tin hóa đơn
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình danh sách hóa đơn		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn tra cứu danh sách hóa đơn - Chọn cập nhật thông tin hóa đơn - Quản lý điền thông tin cập nhật cho hóa đơn - Quản lý xác nhận cập nhật hóa đơn - Hệ thống thông báo cập nhật hóa đơn thành công và chuyển sang màn hình thông tin hóa đơn 		
Alternative flow/ Process	<ul style="list-style-type: none"> - Hệ thống báo cập nhật thất bại - Hệ thống chuyển màn hình danh sách hóa đơn 		
Error situations	<ul style="list-style-type: none"> - Chọn cập nhật thông tin hóa đơn nhưng hệ thống không chuyển màn hình cập nhật - Xác nhận cập nhật thông tin hóa đơn nhưng hệ thống không thay đổi thông tin mới cập nhật - Không kết nối được với server 		

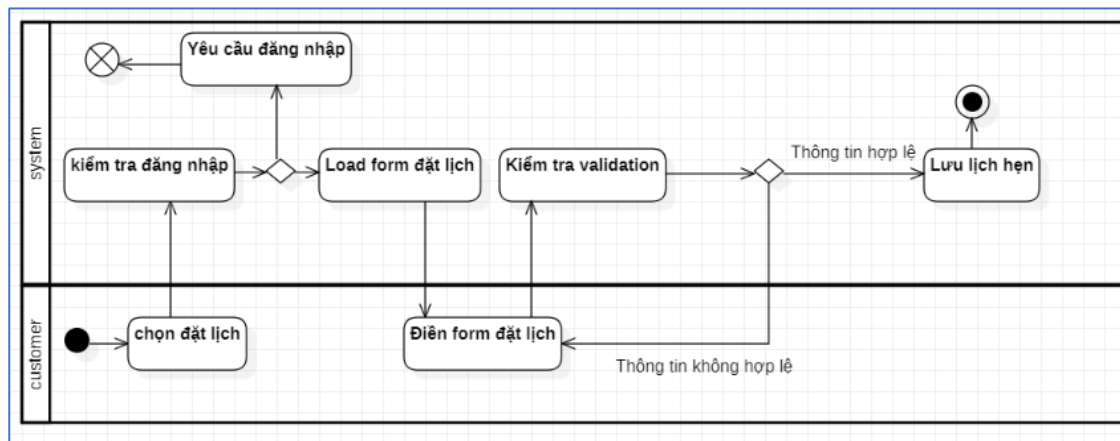
System state in error situations	Không cập nhật được thông tin nhân viên
---	---

UC43: Xem Báo Cáo Thống Kê

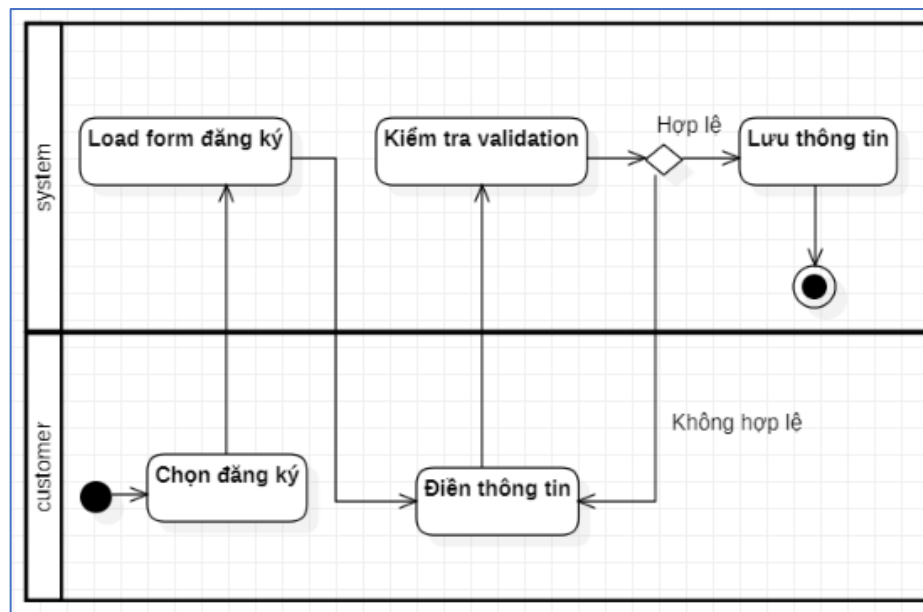
Name	Xem Báo Cáo Thống Kê	Code	UC43
Description	Dùng để xem báo cáo thống kê doanh thu, lượng xe bán ra trong tháng		
Actor	Quản Lý	Trigger	Khi quản lý chọn xem báo cáo thống kê
Pre condition	Đăng nhập vào hệ thống		
Post condition	Chuyển sang màn hình chi tiết báo cáo		
Standard flow/ Process	<ul style="list-style-type: none"> - Đăng nhập thành công vào hệ thống - Chọn xem báo cáo thống kê - Hệ thống hiển thị báo cáo doanh thu - Quản lý chọn xem chi tiết báo cáo - Hệ thống chuyển sang màn hình chi tiết báo cáo 		
Alternative flow/ Process			
Error situations	<ul style="list-style-type: none"> - Chọn xem báo cáo thống kê hệ thống không chuyển màn hình báo cáo thống kê - Chọn xem chi tiết báo cáo thống kê nhưng không chuyển màn hình thông tin chi tiết báo cáo - Không kết nối được với server 		

System state in error situations	Không xem được báo cáo thống kê
---	---------------------------------

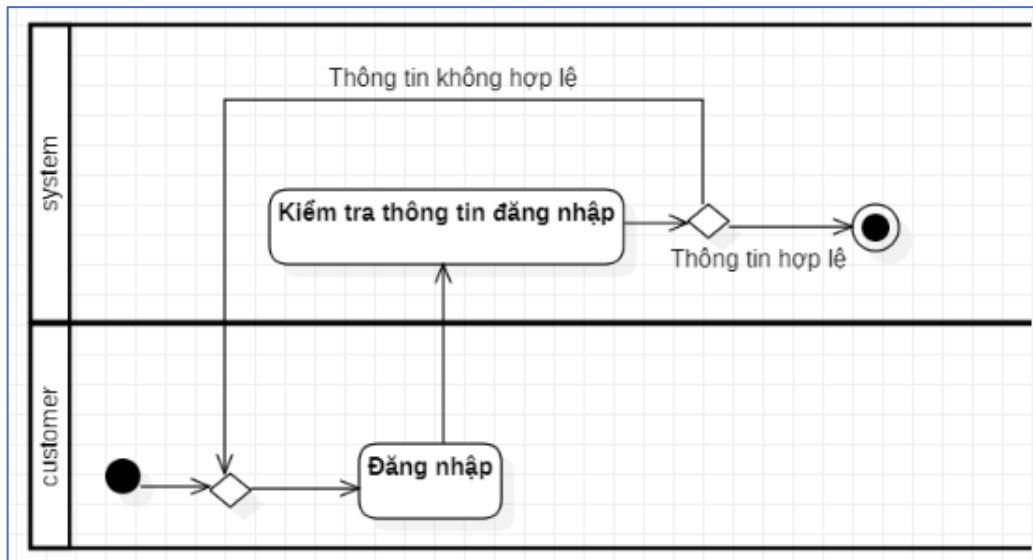
2.2.2 Activity Diagram



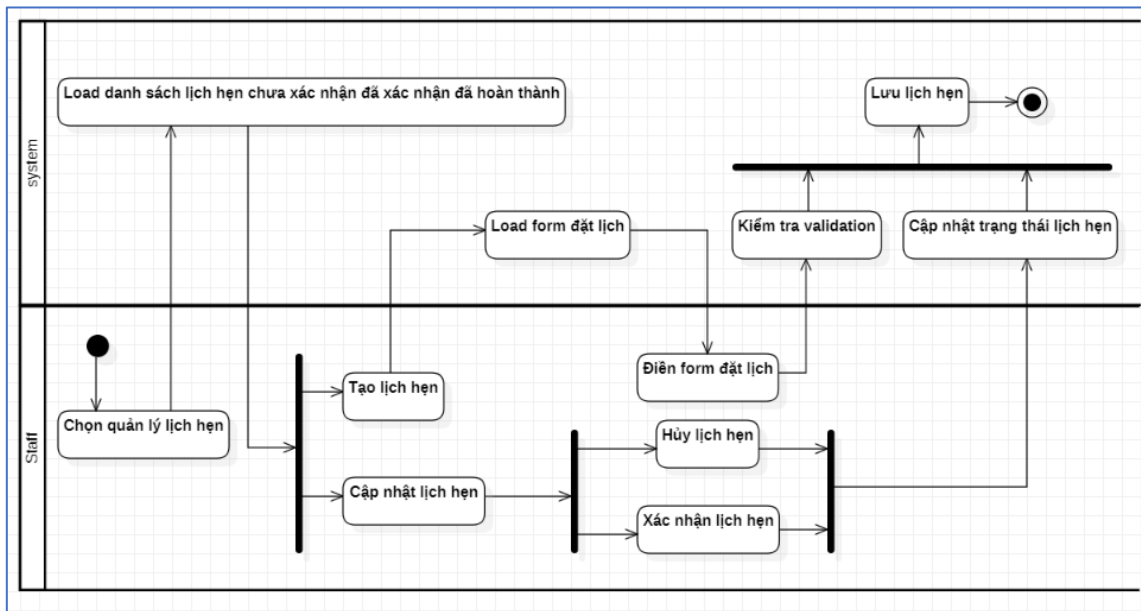
Hình 7: Activity Diagram quy trình Đặt lịch hẹn - Khách hàng



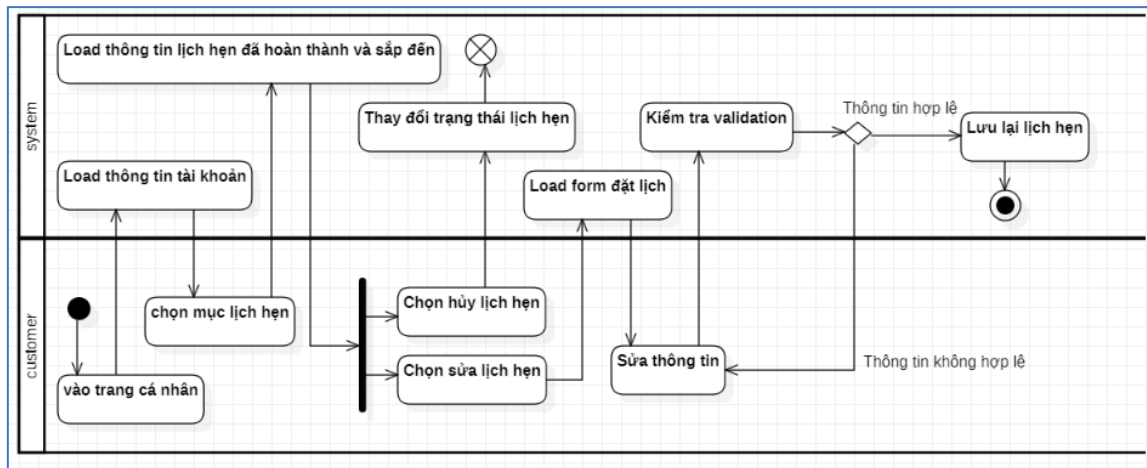
Hình 8: Activity Diagram quy trình Đăng ký - Khách hàng



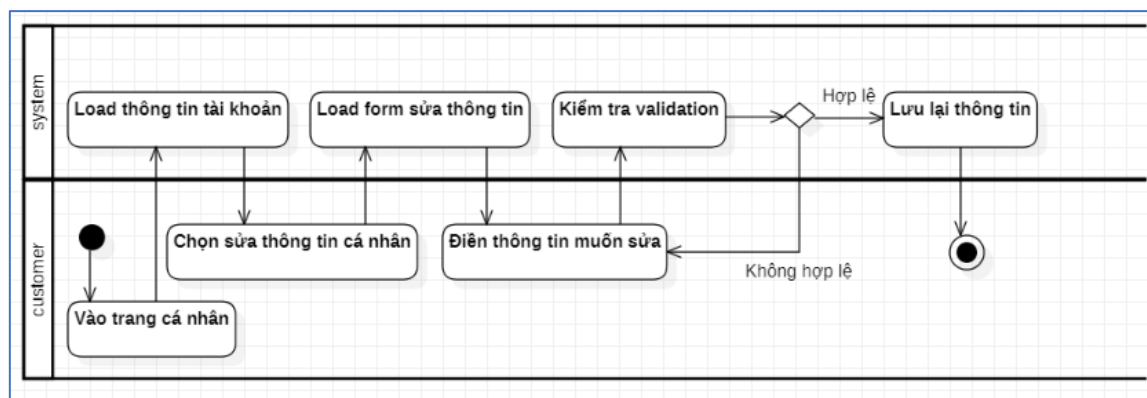
Hình 9: Activity Diagram quy trình Đăng nhập - Khách hàng



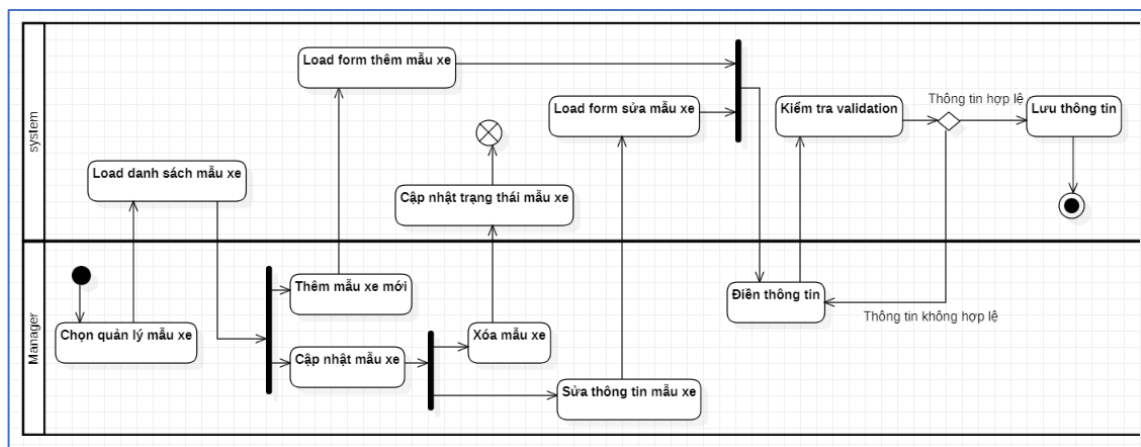
Hình 10: Activity Diagram quy trình Quản lý lịch hẹn – Nhân viên



Hình 11: Activity Diagram quy trình cập nhật, hủy lịch hẹn - Khách hàng

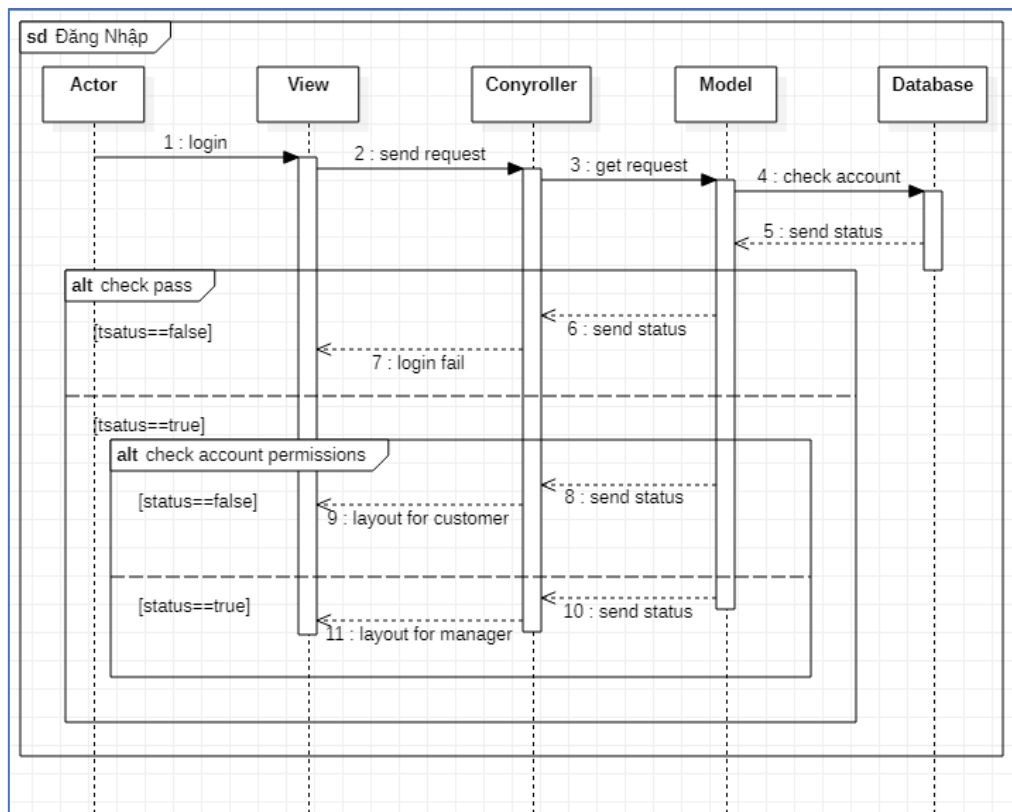


Hình 12: Activity Diagram quy trình Cập nhật thông tin cá nhân - Khách hàng

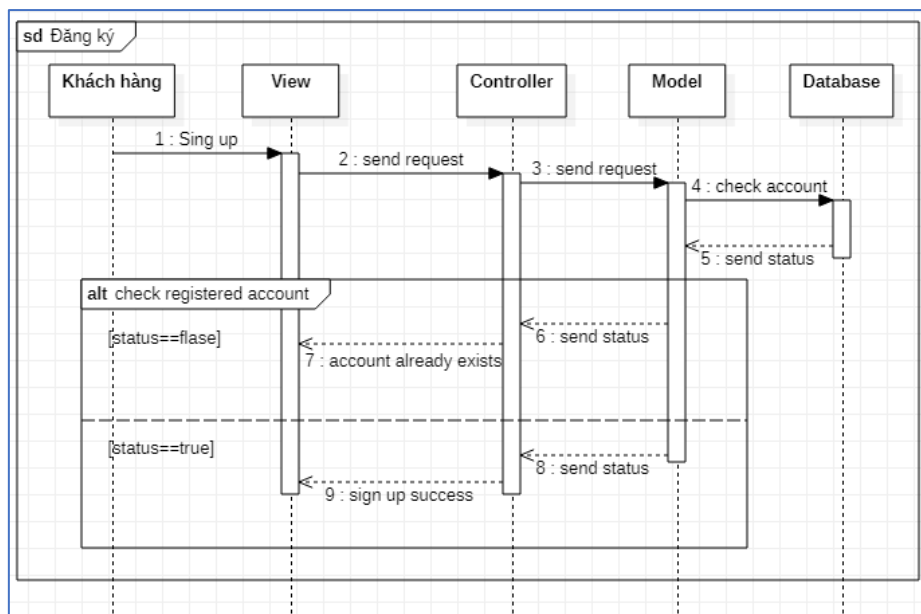


Hình 13: Activity Diagram quy trình quản lý mẫu xe - Quản lý

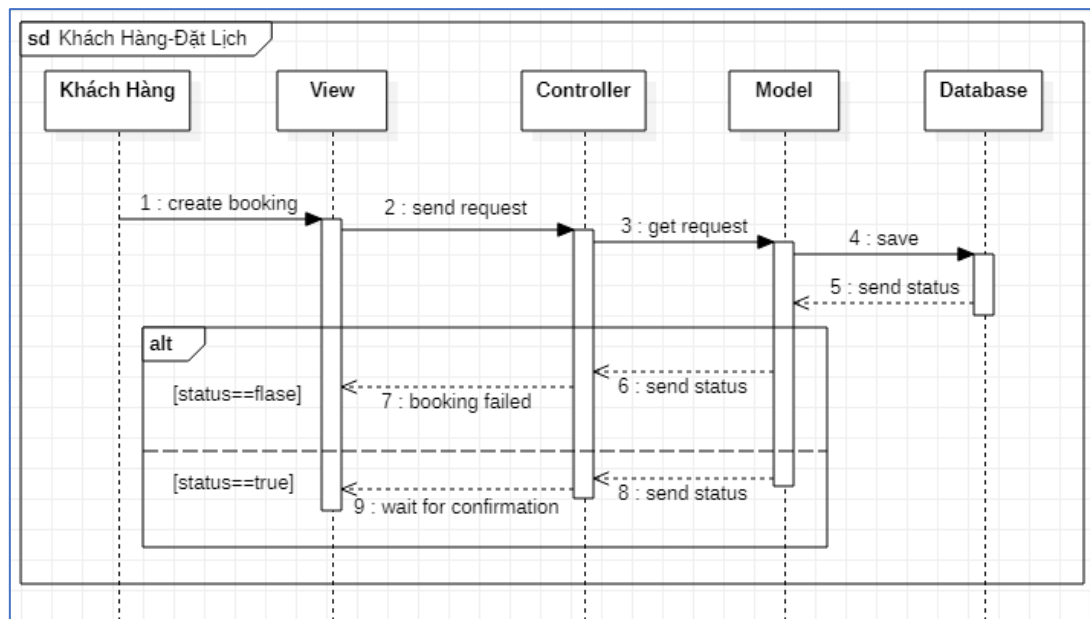
2.2.3 Sequence Diagram



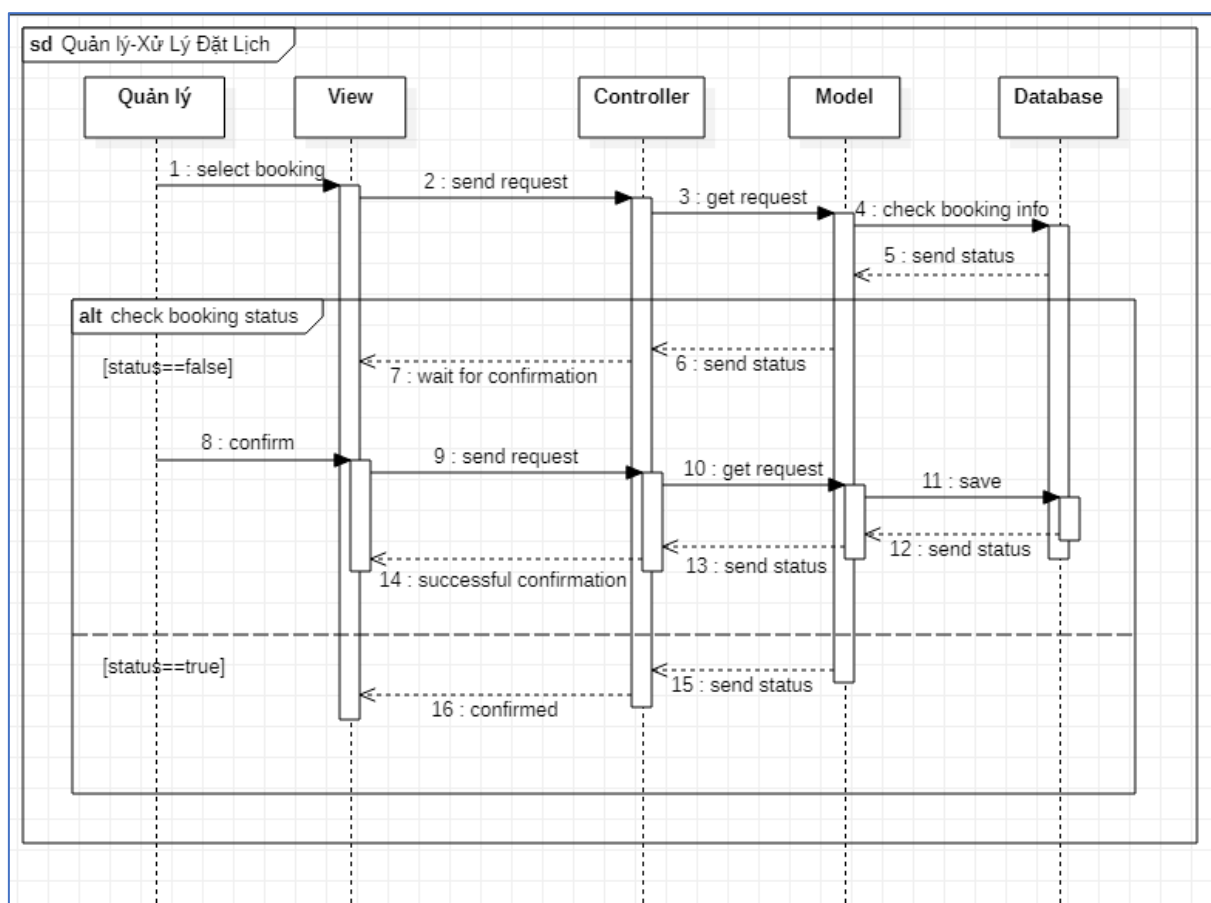
Hình 14: Sequence Diagram quy trình đăng nhập



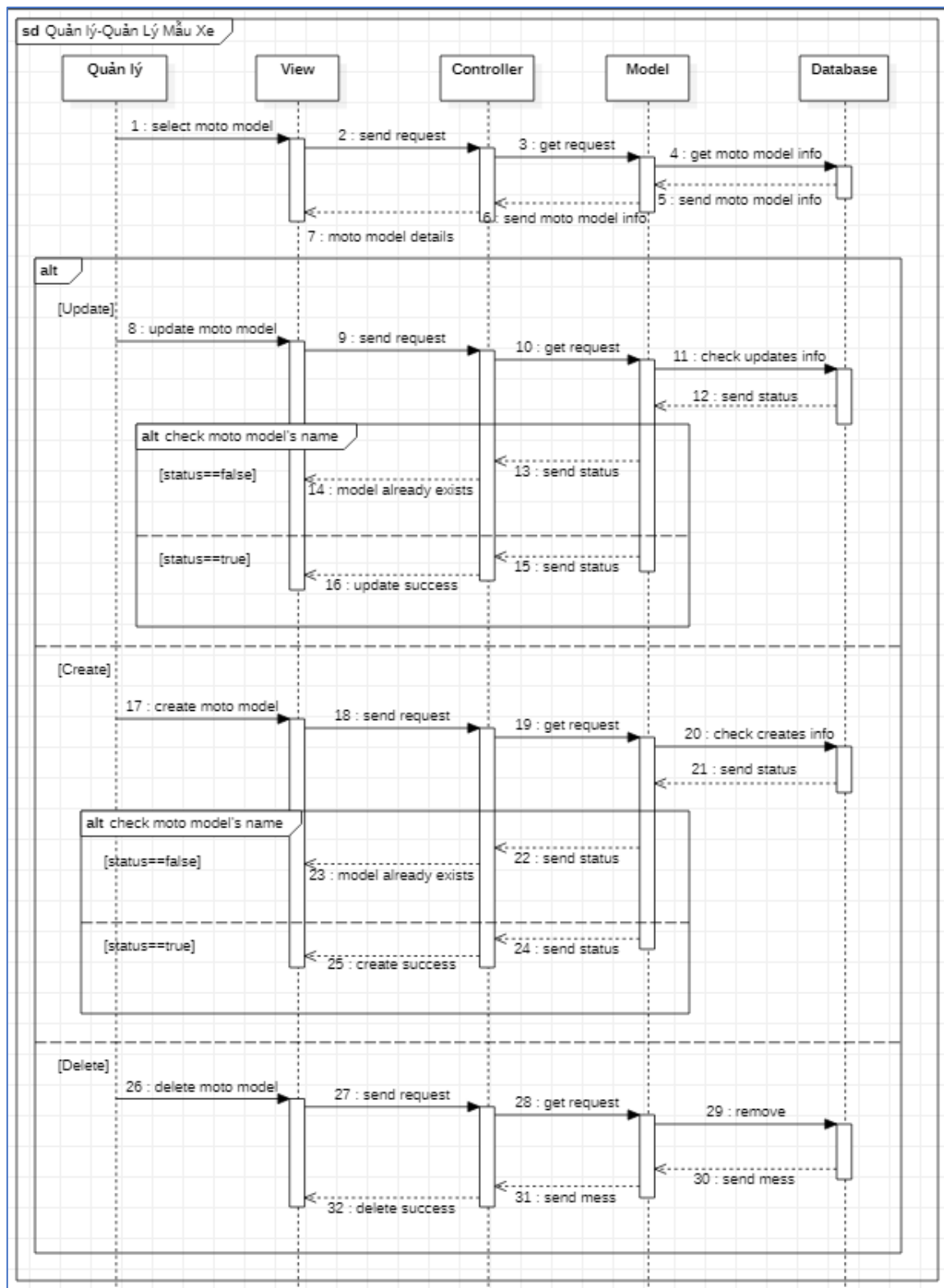
Hình 15: Sequence Diagram quy trình Đăng ký tài khoản



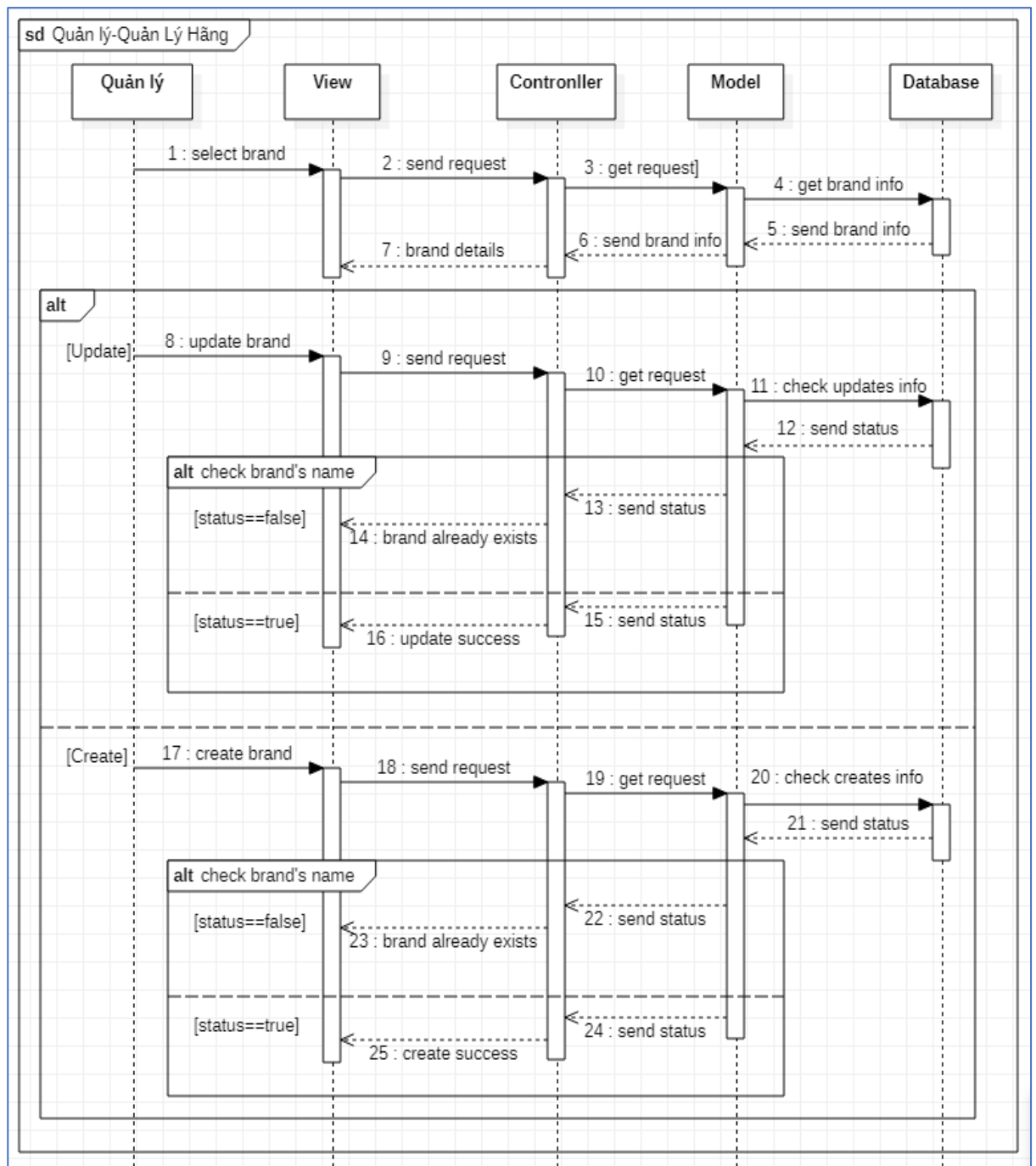
Hình 16: Sequence Diagram quy trình Đặt lịch - Khách hàng



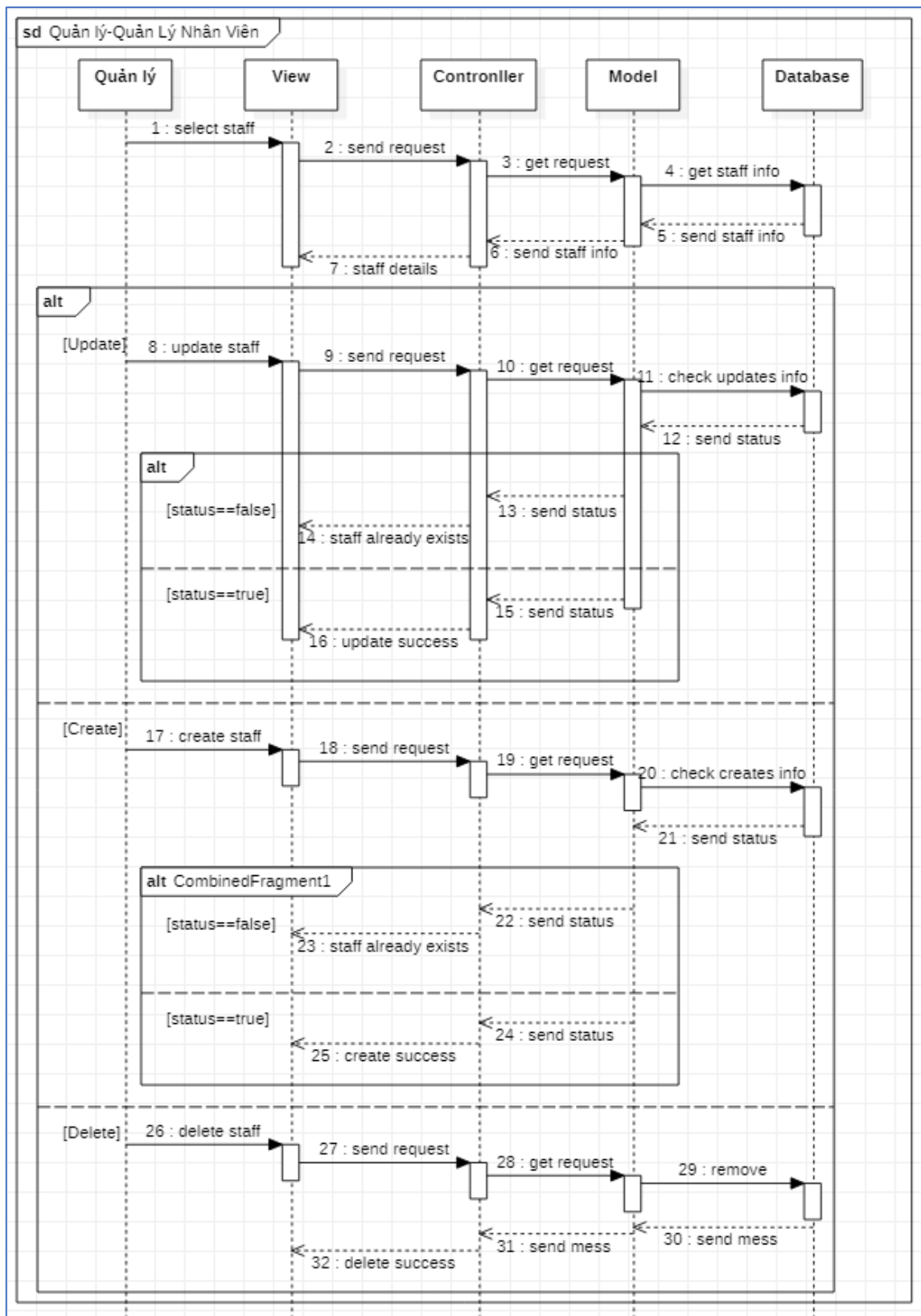
Hình 17: Sequence Diagram quy trình xử lý đặt lịch – Nhân viên



Hình 18: Sequence Diagram quy trình quản lý mẫu xe - Quản lý

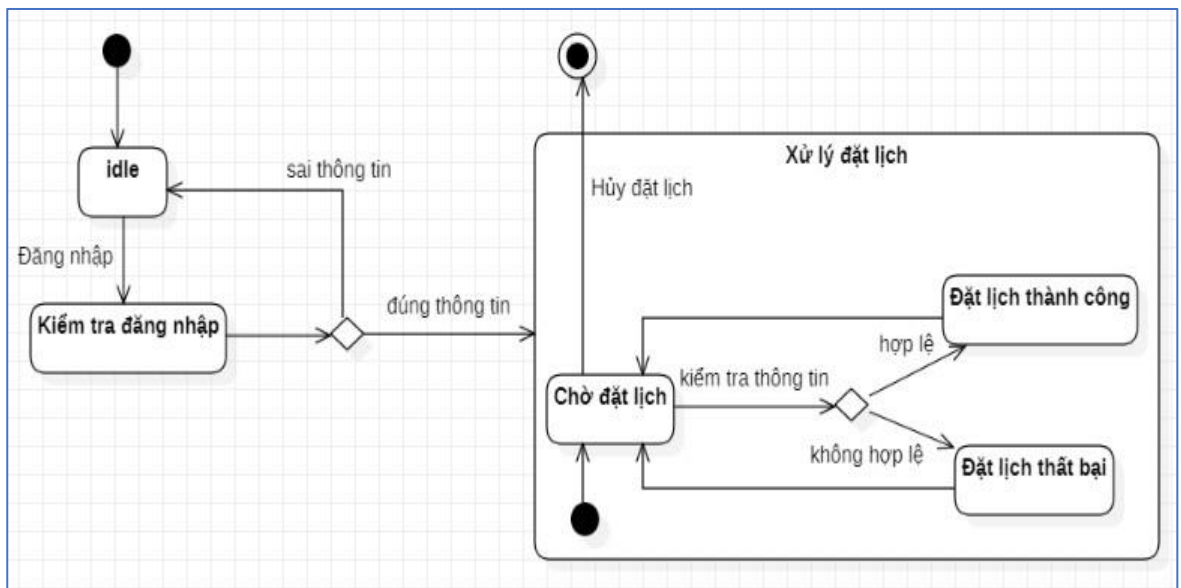


Hình 19: Sequence Diagram quy trình quản lý hãng - Quản lý

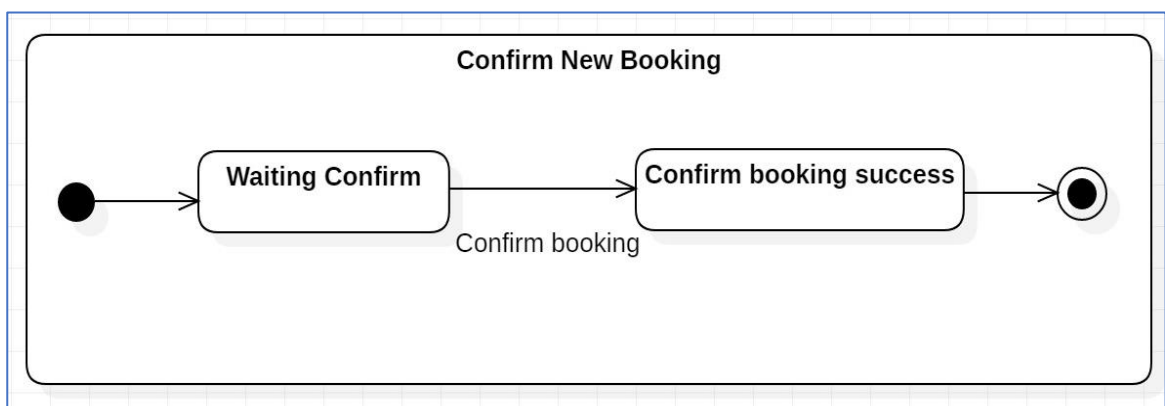


Hình 20: Sequence Diagram quy trình quản lý nhân viên - Quản lý

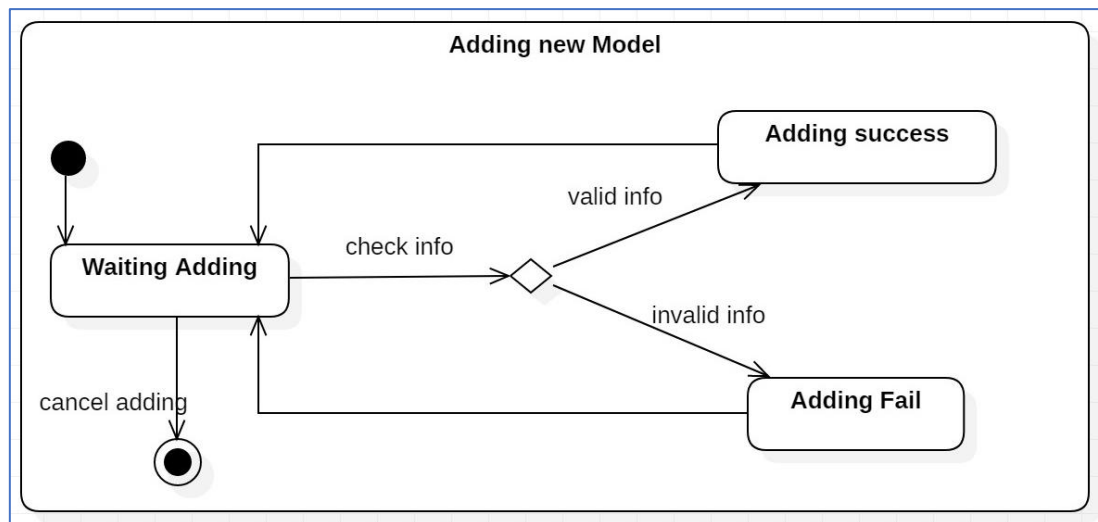
2.2.4 Statechart Diagram



Hình 21: State Machine Diagram cho đối tượng lịch hẹn - Khách hàng

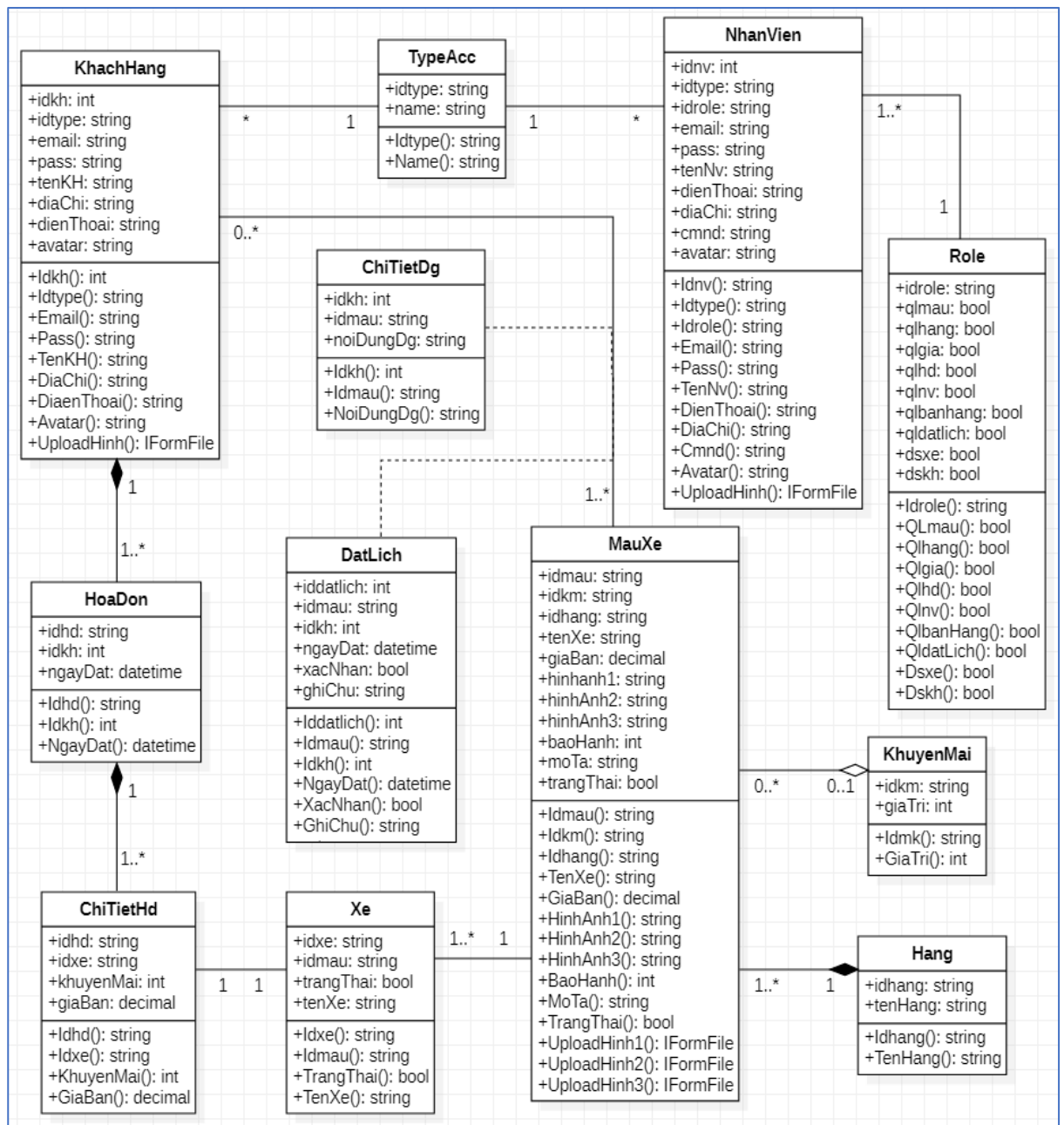


Hình 22: State Machine Diagram cho đối tượng lịch hẹn - Nhân viên



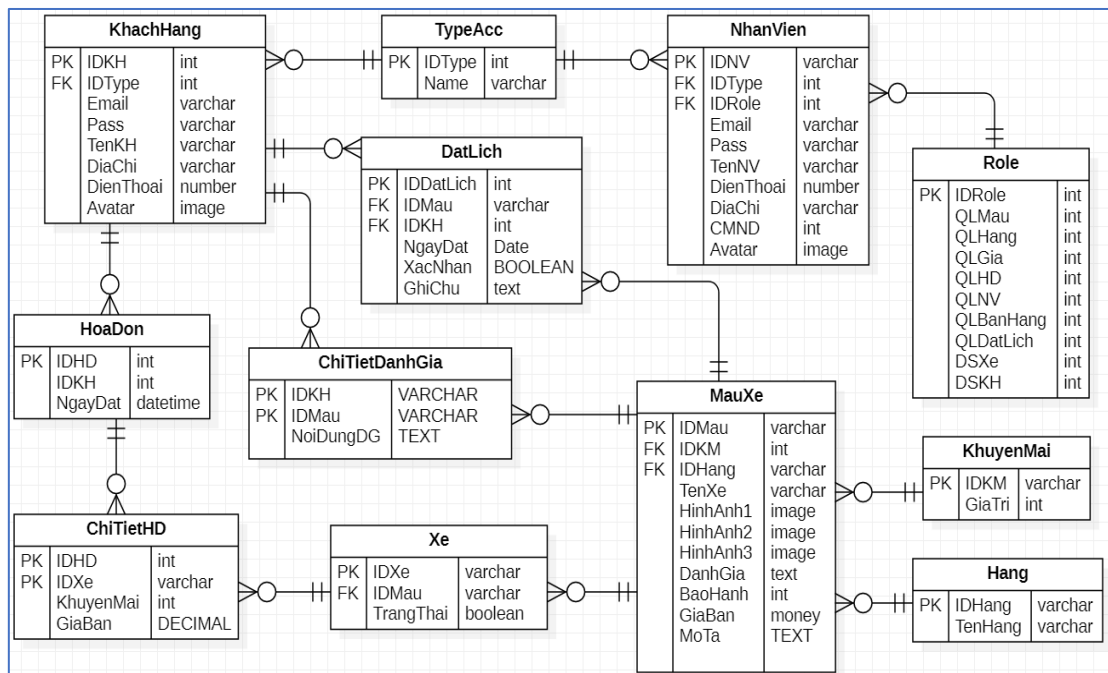
Hình 23: State Machine Diagram cho đối tượng hãng và mẫu xe - Nhân viên

2.2.5 Class Diagram

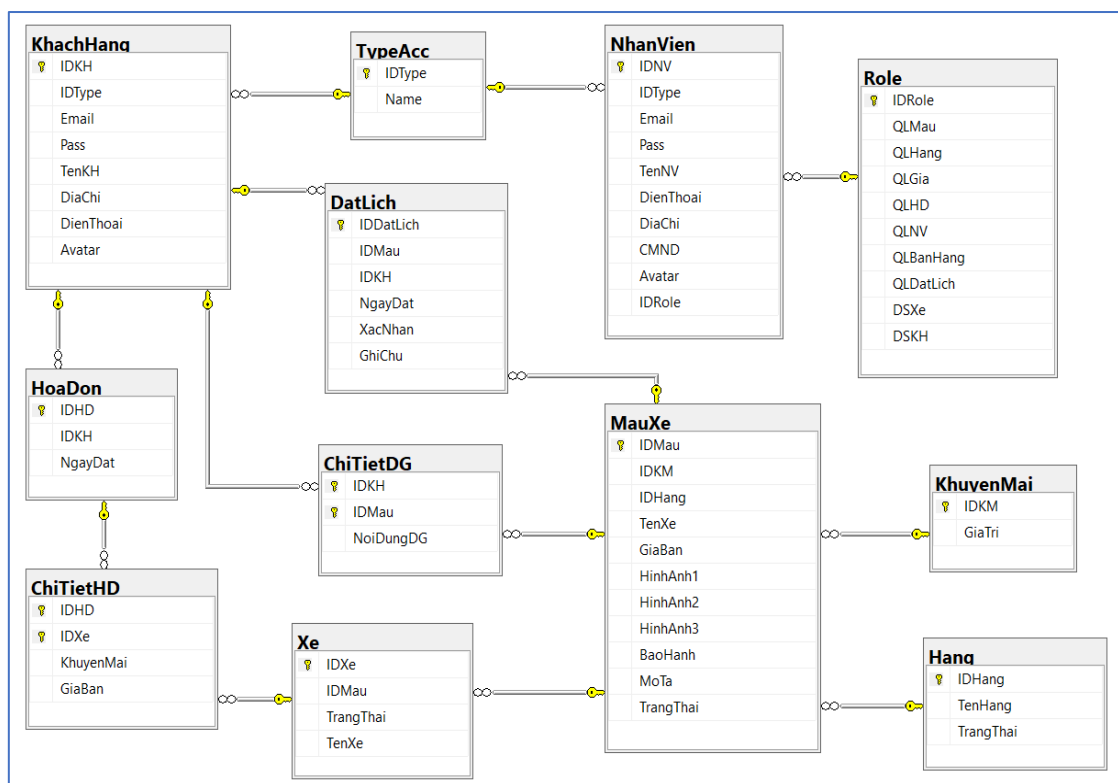


Hình 24: Class Diagram hệ thống

2.2.6 ER Diagram



Hình 25: ER Diagram hệ thống Database Diagram



Hình 26: Database Diagram hệ thống

3 Kỹ thuật thực hiện

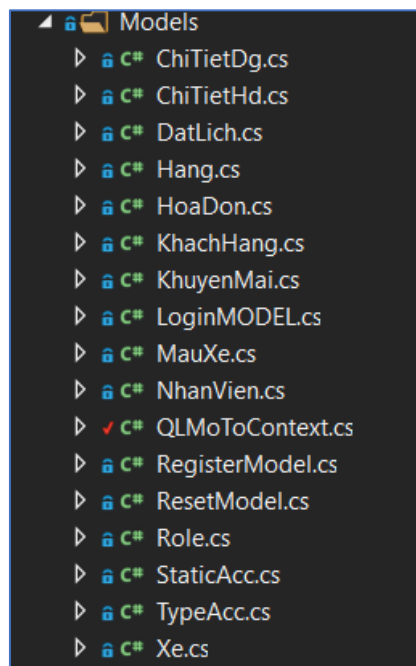
3.1 Cơ sở lý thuyết

- Công nghệ: ASP.NET Core, Entity Framework Core, SQL Server
- Ngôn ngữ: C#, HTML, CSS, JavaScript, Bootstrap, JQuery
- + Mô hình MVC
- + Trình biên dịch: Visual Studio 2019
- + Công cụ quản lý code: Github
- **Giải thích:**
 - **ASP.NET Core** là gì? Nó là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.
 - **ASP.NET Core** đã đạt được những nền tảng cải tiến dưới đây:
 - *Hợp nhất việc xây dựng web UI và web APIs*
 - *Tích hợp những client-side frameworks hiện đại và những luồng phát triển*
 - *Hệ thống cấu hình dựa trên môi trường đám mây thật sự*
 - *Dependency injection được xây dựng sẵn*
 - *HTTP request được tối ưu nhẹ hơn*
 - *Có thể host trên IIS hoặc self-host trong Process của riêng bạn*
 - *Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning*
 - *Chuyển các thực thể, thành phần, module như những NuGet packages*
 - *Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại*
 - *Xây dựng và chạy đa nền tảng (Windows, Mac và Linux)*
 - *Mã nguồn mở và tập trung vào cộng đồng*
 - Sử dụng ASP.NET CORE MVC để xây dựng Web UI và Web API:
 - *Bạn có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC (Model-View-Controller)*
 - *Bạn có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về*

- *Razor cung cấp một ngôn ngữ hiệu quả để tạo Views*
- *Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTM.*
- *Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action*
- *Model Validation tự động thực hiện validate client và server*

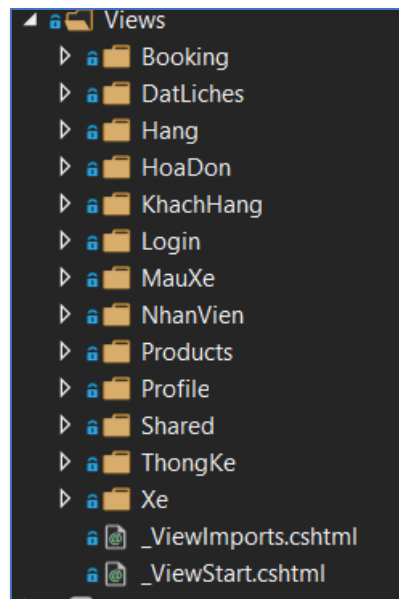
3.2 Áp dụng

3.2.1 Model



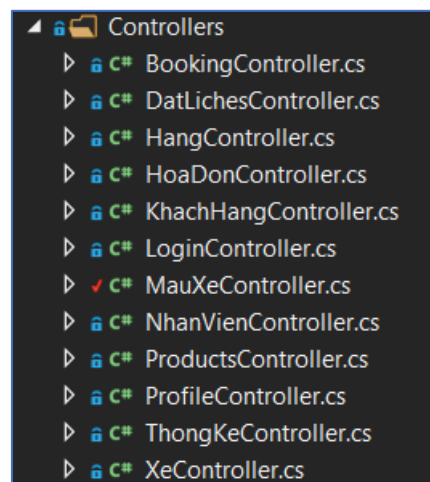
Hình 27: Model

3.2.2 View



Hình 28: View

3.2.3 Controller



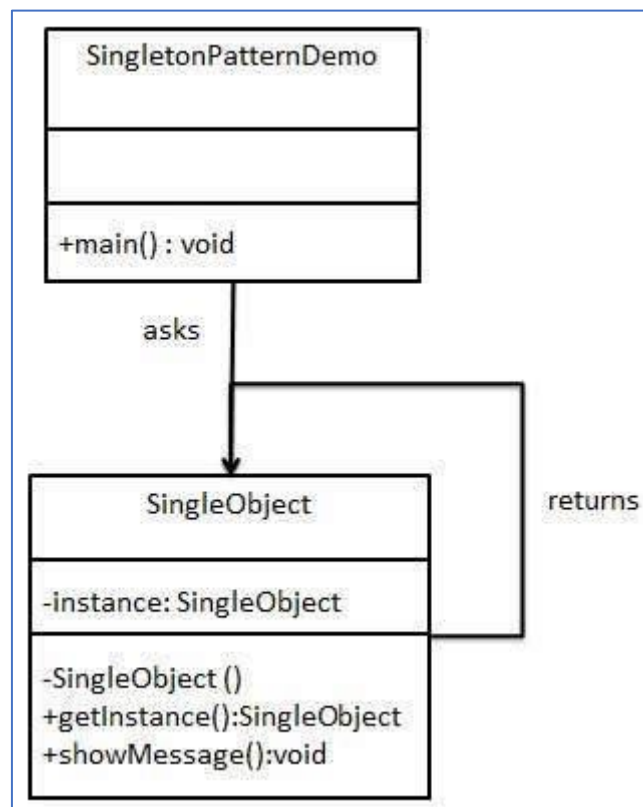
Hình 29: Contronller

4 Design Patterns

4.1 Các mẫu design pattern được áp dụng trong đề tài

4.1.1 Singleton Pattern

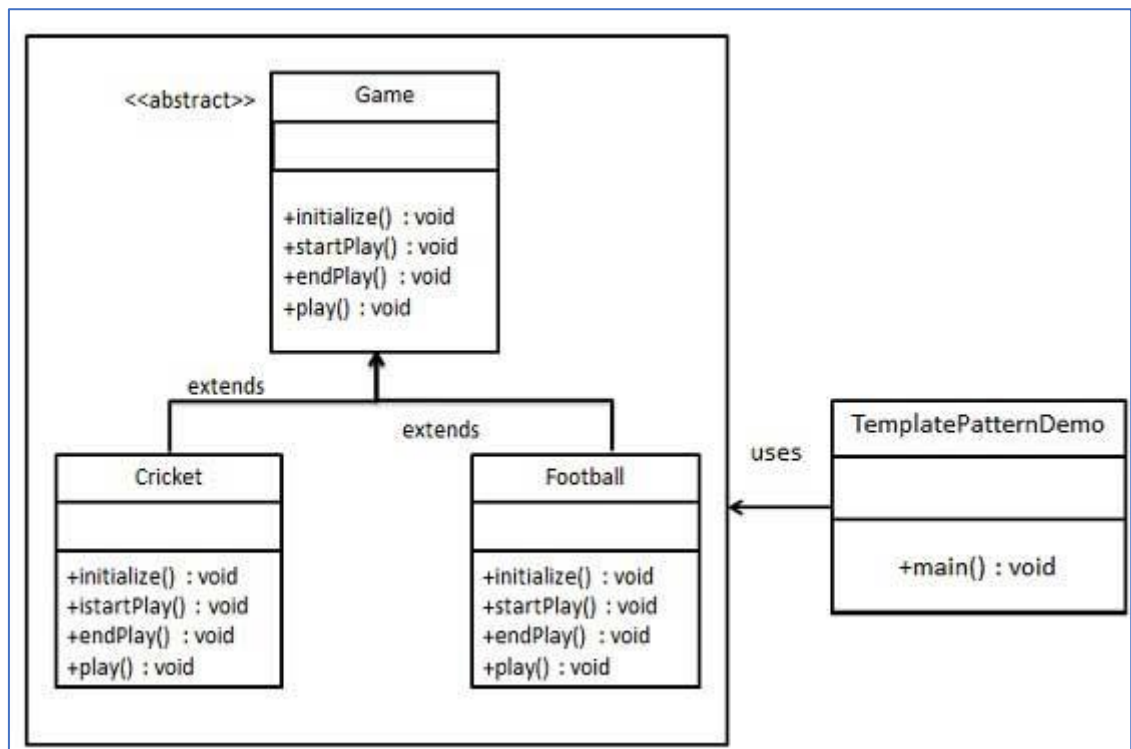
- Singleton Pattern là một design pattern trong số các design pattern thuộc nhóm Creational Design Pattern gồm: Abstract factory, Builder, Factory, Prototype, **Singleton** và Flyweight.
- Singleton Pattern giúp:
 - + Đảm bảo rằng một class chỉ có duy nhất một instance (khởi tạo)
 - + Và cung cấp một cách toàn cầu để truy cập tới instance đó.
- Cấu trúc:



Hình 30: Singleton Pattern

4.1.2 Template Method Pattern

- Template Method pattern là 1 mẫu thiết kế giúp định nghĩa một bộ khung của một thuật toán trong một chức năng, chuyển giao việc thực hiện nó cho các lớp con.
- Template Method Pattern cho phép lớp con định nghĩa lại cách thực hiện của một thuật toán, mà không phải thay đổi cấu trúc thuật toán.
- Cấu trúc:



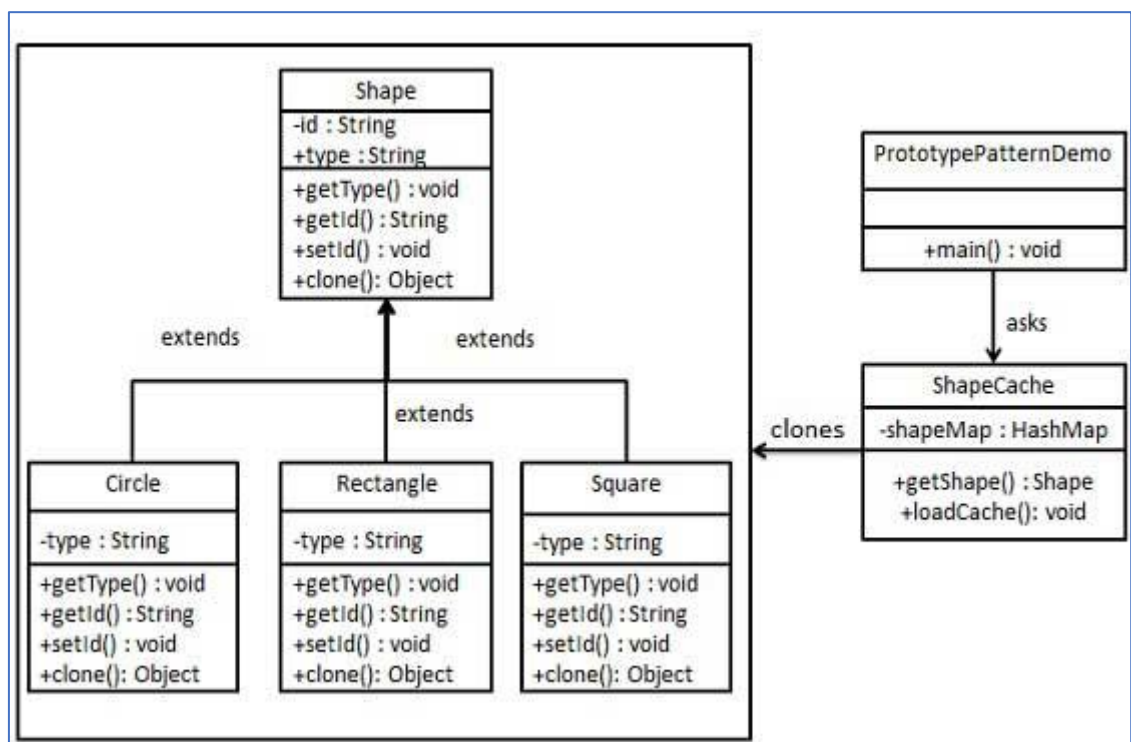
Hình 31: Template Method Pattern

4.1.3 Prototype Pattern

- Prototype Pattern ứng dụng để tạo ra các object lặp trong khi vẫn duy trì được performance. Prototype pattern thuộc loại Creational pattern.
- Prototype Pattern liên quan tới việc xây dựng một interface mẫu để tạo ra các bản sao của một object. Pattern này sử dụng khi việc tạo ra object một cách trực tiếp gây tốn kém tài nguyên.

Ví dụ: Một object chỉ có thể được tạo ra sau một loạt những thao tác phức tạp trên database. Chúng ta có thể cache object đó, trả về bản sao của nó trong request tiếp theo và update database khi cần thiết do đó có thể giảm thiểu được những lần gọi lệnh database.

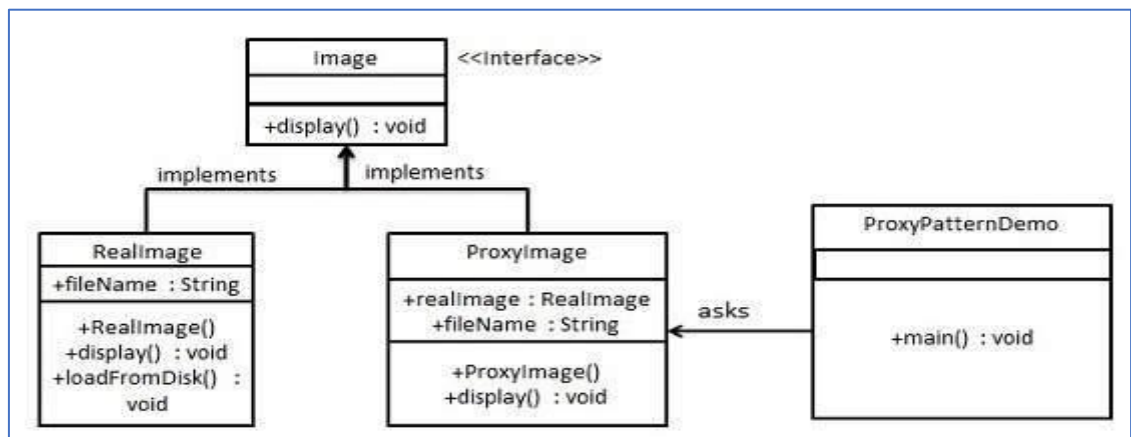
- Cấu trúc:



Hình 32: Prototype Pattern

4.1.4 Proxy Pattern

- Proxy Pattern là một Design Pattern thuộc nhóm Structural Design Pattern liên quan đến cấu trúc và kết cấu các đối tượng trong lập trình hướng đối tượng.
- Proxy là một lớp đại diện cho chức năng của một lớp khác. Loại mô hình thiết kế này thuộc mô hình cấu trúc.
- Trong mẫu proxy, chúng ta có thể tạo đối tượng có đối tượng gốc để giao tiếp chức năng của nó với các đối tượng bên ngoài.
- Cấu trúc:

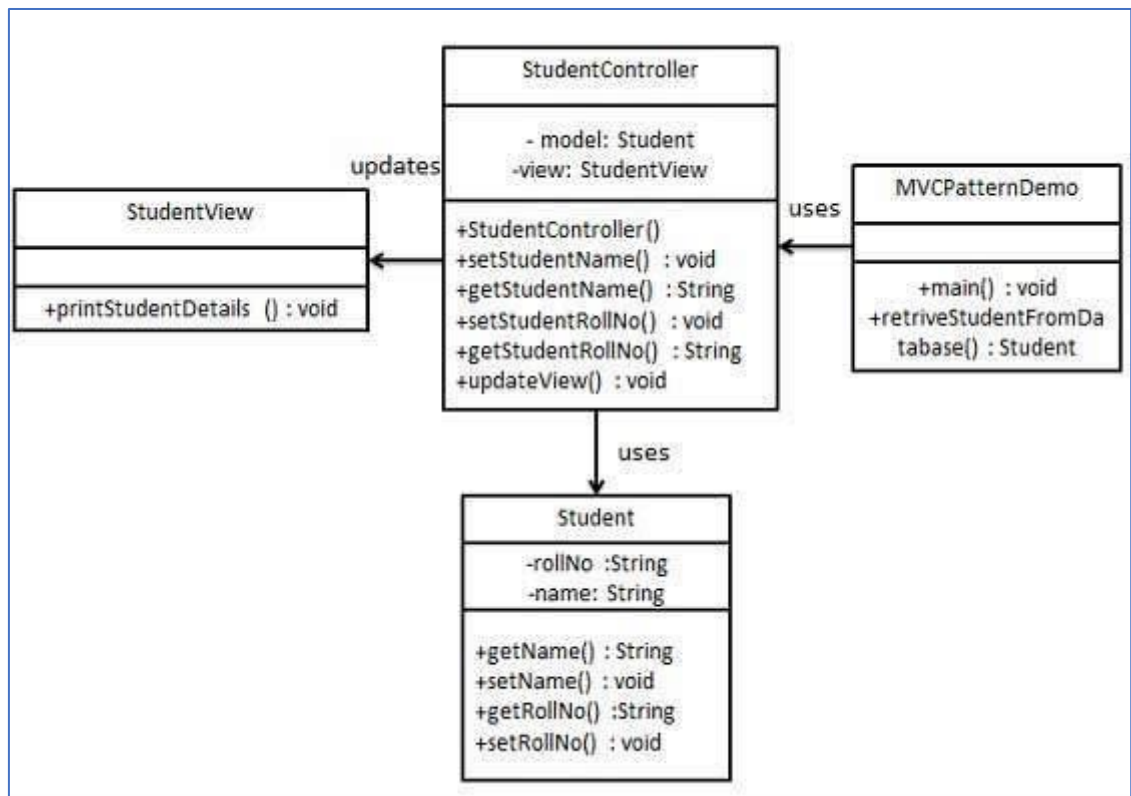


Hình 33: Proxy Pattern

4.1.5 MVC Pattern

- MVC Pattern là viết tắt của Model-View-Controller Pattern. Mẫu này được sử dụng để phân tách các mối quan tâm của ứng dụng.
- + **Mô hình** - Mô hình đại diện cho một đối tượng hoặc JAVA POJO mang dữ liệu. Nó cũng có thể có logic để cập nhật bộ điều khiển nếu dữ liệu của nó thay đổi.
- + **Chế độ xem** - Chế độ xem biểu thị sự trực quan hóa của dữ liệu mà mô hình chứa.
- + **Bộ điều khiển** - Bộ điều khiển hoạt động trên cả mô hình và chế độ xem. Nó kiểm soát luồng dữ liệu vào đối tượng mô hình và cập nhật chế độ xem bất cứ khi nào dữ liệu thay đổi. Nó giữ cho chế độ xem và mô hình tách biệt.

- Cấu trúc:



Hình 34: MVC Pattern

4.2 Áp dụng

4.2.1 Singleton Pattern

a) Class HangSingleton

```
public sealed class HangSingleton
{
    14 references
    public static HangSingleton Instance { get; } = new HangSingleton();
    13 references
    public List<Hang> listHang { get; } = new List<Hang>();

    1 reference
    private HangSingleton() { }

    //1 lần duy nhất
    2 references
    public void Init(QLMoToContext database)
    {
        if (listHang.Count == 0)
        {
            var model = new ViewModel();
            model.ListHang = database.Hangs.ToArray();

            foreach (var item in model.ListHang)
            {
                listHang.Add(item);
            }
        }
    }

    3 references
    public void Update(QLMoToContext database)
    {
        listHang.Clear();
        Init(database);
    }
}
```

b) Class HangController

```
public HangController(QLMoToContext db, ILogger<MauXeController> logger)
{
    database = db;
    HangSingleton.Instance.Init(database);
    this.logger = logger;

    PrintInfomation();
}

public IActionResult Index()
{
    return View(HangSingleton.Instance.listHang);
}
```

```

public IActionResult Create()
{
    ViewBag.Role = TempData["Role"];

    var model = new Hang();

    return View(model);
}

[HttpPost]
0 references
public IActionResult Create(Hang hang)
{
    if (ModelState.IsValid)
    {
        database.Add(hang);
        database.SaveChanges();

        HangSingleton.Instance.Update(database);

        return RedirectToAction("Index");
    }

    return View(hang);
}

public IActionResult Edit(string id)
{
    ViewBag.Role = TempData["Role"];

    var model = new Hang();
    model = database.Hangs.Where(x => x.Idhang == id).FirstOrDefault();

    return View(model);
}

[HttpPost]
0 references
public IActionResult Edit(Hang hang)
{
    var model = hang;
    if (ModelState.IsValid)
    {
        database.Update(model);
        database.SaveChanges();

        HangSingleton.Instance.Update(database);

        return RedirectToAction("Index");
    }

    return View(model);
}

```

c) Class MauController

```
public IActionResult Index(string sortOrder, string currentFilter, string searchString, int? page)
{
    -----
    int pageSize = 10;
    //set page to one is there is no value, ?? is called the null
    int pageNumber = (page ?? 1);
    //return the Model data with paged
    var modelv = new ViewModel
    {
        ListHang = HangSingleton.Instance.listHang.ToArray(),
        ListXe = database.Xes.ToArray(),
        ListMauXes = Modellist.ToPagedList(pageNumber, pageSize),
        ListKhuyenMai = database.KhuyenMais.ToArray()
    };
    return View(modelv);
}

public IActionResult Create()
{
    ViewBag.Role = TempData["Role"];

    var model = new ViewModel();
    //model.ListHang = database.Hangs.ToArray();
    model.ListHang = HangSingleton.Instance.listHang.ToArray();
    model.ListKhuyenMai = database.KhuyenMais.ToArray();
    return View(model);
}

[HttpPost]
0 references
public async Task<IActionResult> Create(MauXe mauXe)
{
    var model = new ViewModel();
    //model.ListHang = database.Hangs.ToArray();
    model.ListHang = HangSingleton.Instance.listHang.ToArray();
    model.ListKhuyenMai = database.KhuyenMais.ToArray();
    if (ModelState.IsValid)
    {
        
        database.Add(mauXe);
        await database.SaveChangesAsync();
        return RedirectToAction("Index");
    }
    return View(model);
}

public IActionResult Details(string id)
{
    ViewBag.Role = TempData["Role"];

    var model = new ViewModel();
    model.ListHang = HangSingleton.Instance.listHang.ToArray();
    model.ListKhuyenMai = database.KhuyenMais.ToArray();
    model.mauXe = database.MauXes.Where(x => x.Idmau == id).FirstOrDefault();
    return View(model);
}
```

4.2.2 Template Method Pattern

a) Class TemplateMethodController

```
public abstract class TemplateMethodController : Controller
{
    3 references
    protected abstract void PrintRoutes();
    3 references
    protected abstract void PrintDIIs();//dependenddeintraction

    //template Method
    2 references
    public void PrintInfomation()
    {
        PrintRoutes();
        PrintDIIs();
    }
}
```

b) Class HangController

```
public class HangController : TemplateMethodController
{
    private readonly QLMoToContext database;
    private readonly ILogger<MauXeController> logger;

    0 references
    public HangController(QLMoToContext db, ILogger<MauXeController> logger)
    {
        database = db;
        HangSingleton.Instance.Init(database);
        this.logger = logger;

        PrintInfomation();
    }

    protected override void PrintRoutes()
    {
        logger.LogDebug($"{GetType().Name}
        Routes:
        GET: MauXe/Index
        GET: MauXe/Create
        POST: MauXe/Create
        GET: MauXe/Delete/id
        POST: MauXe/Delete/id
        GET: MauXe/Edit/id
        POST: MauXe/Edit/id
        ");
    }

    3 references
    protected override void PrintDIIs()
    {
        logger.LogDebug($"{
        Dependencies:
        QLMoToContext database
        ILogger<MauXeController> logger
        ");
    }
}
```

c) Class MauController

```

public class MauXeController : TemplateMethodController
{
    private readonly QLMoToContext database;
    private readonly ILogger<MauXeController> logger;
    private readonly IWebHostEnvironment hostEnvironment;

    0 references
    public MauXeController(QLMoToContext db, IWebHostEnvironment hostEnvironment, ILogger<MauXeController> logger)
    {
        database = db;
        this.hostEnvironment = hostEnvironment;
        this.logger = logger;

        PrintInfomation();
    }

    protected override void PrintRoutes()
    {
        logger.LogDebug($"{GetType().Name}
        Routes:
        GET: MauXe/Index
        GET: MauXe/Create
        POST: MauXe/Create
        GET: MauXe/Details/id
        GET: MauXe/Delete/id
        POST: MauXe/Delete/id
        GET: MauXe/Edit/id
        POST: MauXe/Edit/id
        ");
    }

    3 references
    protected override void PrintDIs()
    {
        logger.LogDebug($"{
        Dependencies:
        QLMoToContext database
        ILogger<MauXeController> logger
        IWebHostEnvironment hostEnvironment
        ");
    }
}

```

4.2.3 Prototype Pattern

```
public interface MauXePrototype
{
    2 references
    MauXePrototype Clone(int s1);
}
```

a) Class MauXe

```
public partial class MauXe:MauXePrototype
{

    public MauXePrototype Clone(int s1)
    {
        MauXe newmau = new MauXe();
        newmau.Idmau = Idmau + s1;
        newmau.TenXe =TenXe;
        newmau.Idhang = Idhang;
        newmau.Idkm = Idkm;
        newmau.BaoHanh = BaoHanh;
        newmau.GiaBan = GiaBan;
        newmau.MoTa = MoTa;
        newmau.TrangThai = TrangThai;
        newmau.HinhAnh1 = HinhAnh1;
        newmau.HinhAnh2 = HinhAnh2;
        newmau.HinhAnh3 = HinhAnh3;

        return newmau;
    }
}
```

b) Class MauXeController

```
public class MauXeController : TemplateMethodController
{
```

```

public IActionResult Duplicate(string id)
{
    ViewBag.Role = TempData["Role"];

    var model = new ViewModel();
    model.ListHang = HangSingleton.Instance.listHang.ToArray();
    model.ListKhuyenMai = database.KhuyenMais.ToArray();
    model.mauXe = database.MauXes.Where(x => x.Idmau == id).FirstOrDefault();
    return View(model);
}

[HttpPost]
0 references
public async Task<IActionResult> Duplicate(string id, MauXe mauXe)
{
    var model = new ViewModel();
    model.ListHang = HangSingleton.Instance.listHang.ToArray();
    model.ListKhuyenMai = database.KhuyenMais.ToArray();
    model.ListMauXe = database.MauXes.ToArray();
    model.mauXe = database.MauXes.Where(x => x.Idmau == id).FirstOrDefault();

    int sl = 0;
    for (int i = 0; i <= model.ListMauXe.Length; i++)
    {
        sl += 1;
    }

    var clonemau = mauXe.Clone(sl);
    database.Add(clonemau);
    await database.SaveChangesAsync();

    return RedirectToAction("Index");
}
}

```

4.2.4 Proxy Pattern

a) Class ProxyKhachHang

```
public enum CodeKhachHang
{
    InvalidTenKhachHang,
    InvalidSoDienThoai,
    Valid
}
2 references
interface MVCEntity
{
    5 references
    CodeKhachHang UpdateDataKhachHang(QLMoToContext database);
}

public class ProxyKhachHang : MVCEntity
{
    private KhachHang khachHang;
    2 references
    public ProxyKhachHang(KhachHang khachHang)
    {
        this.khachHang = khachHang;
    }

    5 references
    public CodeKhachHang UpdateDataKhachHang(QLMoToContext database)
    {
        StringComparison comp = StringComparison.OrdinalIgnoreCase;
        string notallow = "Hồ Chí Minh";

        if(khachHang.TenKh.Contains(notallow, comp))
        {
            return CodeKhachHang.InvalidTenKhachHang;
        }
        else if (khachHang.DienThoai.Length != 10)
        {
            return CodeKhachHang.InvalidSoDienThoai;
        }
        else
        {
            return khachHang.UpdateDataKhachHang(database);
        }
    }
}
```

b) Class KhachHang

```
public partial class KhachHang:MVCEntity
{

    public CodeKhachHang UpdateDataKhachHang(QLMoToContext database)
    {
        database.Update(this);
        return CodeKhachHang.Valid;
    }
}
```

c) Class ProfileController


```

[Authorize(Roles = "customer")]
[HttpPost]
0 references
public async Task<ActionResult> ProfileKhachHang(KhachHang khachHang)
{
    if (ModelState.IsValid)
    {
        var model = await database.KhachHangs.Where(x => x.Idkh == khachHang.Idkh).FirstOrDefaultAsync();
        if(model.Pass == khachHang.Pass)
        {
            model.TenKh = khachHang.TenKh;
            model.DienThoai = khachHang.DienThoai;
            model.DiaChi = khachHang.DiaChi;

            #region Save Image from wwwroot/img
            string wwwRootPath = hostEnvironment.WebRootPath;
            string fileName, extension;

            if (khachHang.UploadHinh != null)
            {
                fileName = Path.GetFileNameWithoutExtension(khachHang.UploadHinh.FileName);
                extension = Path.GetExtension(khachHang.UploadHinh.FileName);
                model.Avatar = fileName + extension;
                string path1 = Path.Combine(wwwRootPath + "/img/", fileName);
                using (var fileStream = new FileStream(path1, FileMode.Create))
                {
                    await khachHang.UploadHinh.CopyToAsync(fileStream);
                }
            }
        }

        ProxyKhachHang proxyKhachHang = new ProxyKhachHang(model);
        var codeupdate = proxyKhachHang.UpdateDataKhachHang(database);
        if (codeupdate == CodeKhachHang.InvalidTenKhachHang)
        {
            khachHang.Avatar = StaticAcc.Avatar;
            ViewBag.ErrorPass = "Tên Đặc Biệt! Không được đặt tên này!";
            return View(khachHang);
        }
        else if (codeupdate == CodeKhachHang.InvalidSoDienThoai)
        {
            khachHang.Avatar = StaticAcc.Avatar;
            ViewBag.ErrorPass = "Số điện thoại phải gồm 10 chữ số!";
            return View(khachHang);
        }
        else
        {
            ViewBag.Message = "Cập nhật thành công!";
            await database.SaveChangesAsync();
        }

        StaticAcc.Avatar = model.Avatar;
        StaticAcc.Name = model.TenKh;
        StaticAcc.TypeAcc = await database.TypeAccs
            .Where(x => x.Idtype == model.Idtype)
            .Select(x => x.Name)
            .FirstOrDefaultAsync();

        return View(model);
    }
    khachHang.Avatar = StaticAcc.Avatar;
    ViewBag.ErrorPass = "Mật khẩu sai!";
    return View(khachHang);
}
khachHang.Avatar = StaticAcc.Avatar;
return View(khachHang);
}

```

4.2.5 MVC Pattern

a) Model

- Models
 - ChiTietDg.cs
 - ChiTietHd.cs
 - DatLich.cs
 - Hang.cs
 - HoaDon.cs
 - KhachHang.cs
 - KhuyenMai.cs
 - LoginMODEL.cs
 - MauXe.cs
 - NhanVien.cs
 - QLMoToContext.cs
 - RegisterModel.cs
 - ResetModel.cs
 - Role.cs
 - StaticAcc.cs
 - TypeAcc.cs
 - Xe.cs

b) View

- Views
 - Booking
 - DatLiches
 - Hang
 - HoaDon
 - KhachHang
 - Login
 - MauXe
 - NhanVien
 - Products
 - Profile
 - Shared
 - ThongKe
 - Xe
 - _ViewImports.cshtml
 - _ViewStart.cshtml

c) Controller

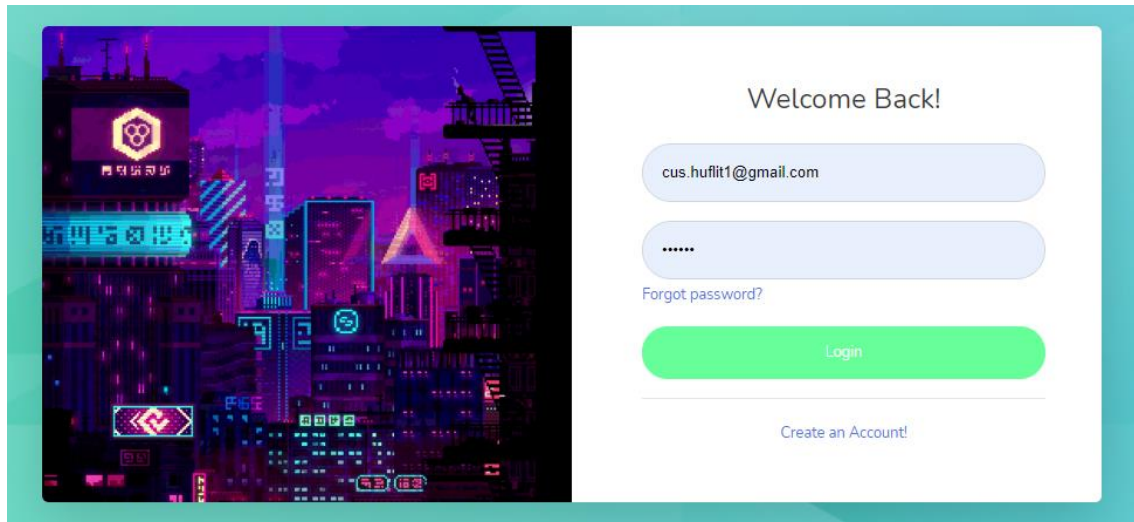
- Controllers
 - BookingController.cs
 - DatLichesController.cs
 - HangController.cs
 - HoaDonController.cs
 - KhachHangController.cs
 - LoginController.cs
 - MauXeController.cs
 - NhanVienController.cs
 - ProductsController.cs
 - ProfileController.cs
 - TemplateMethodController.cs
 - ThongKeController.cs
 - XeController.cs

5 Triển khai – Kết quả

5.1 Một số chức năng và màn hình chính

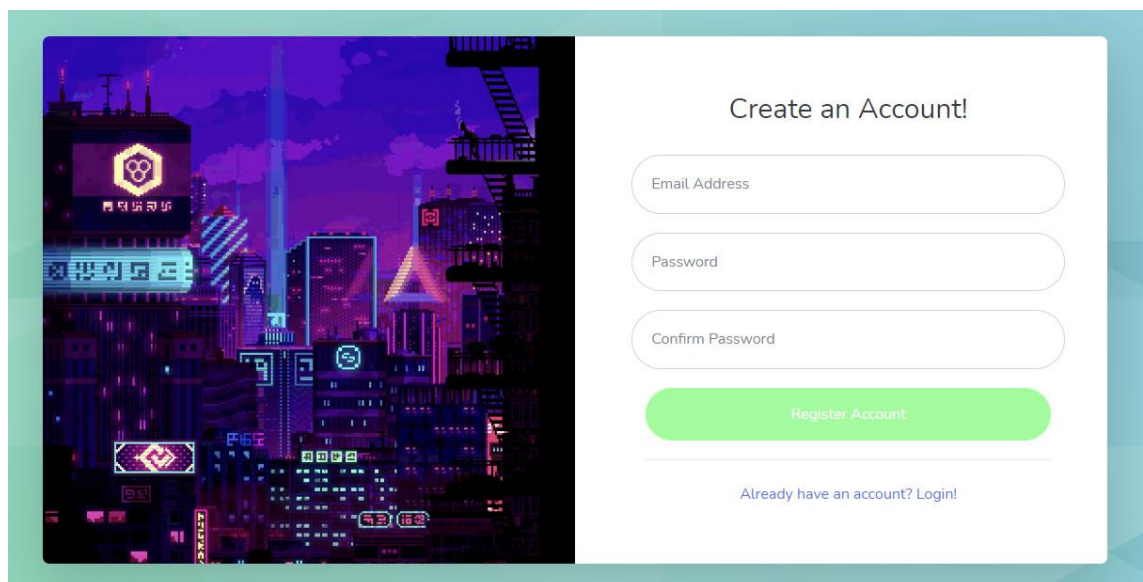
5.1.1 Giao diện chung

5.1.1.1 Giao diện đăng nhập



Hình 35: Giao diện đăng nhập

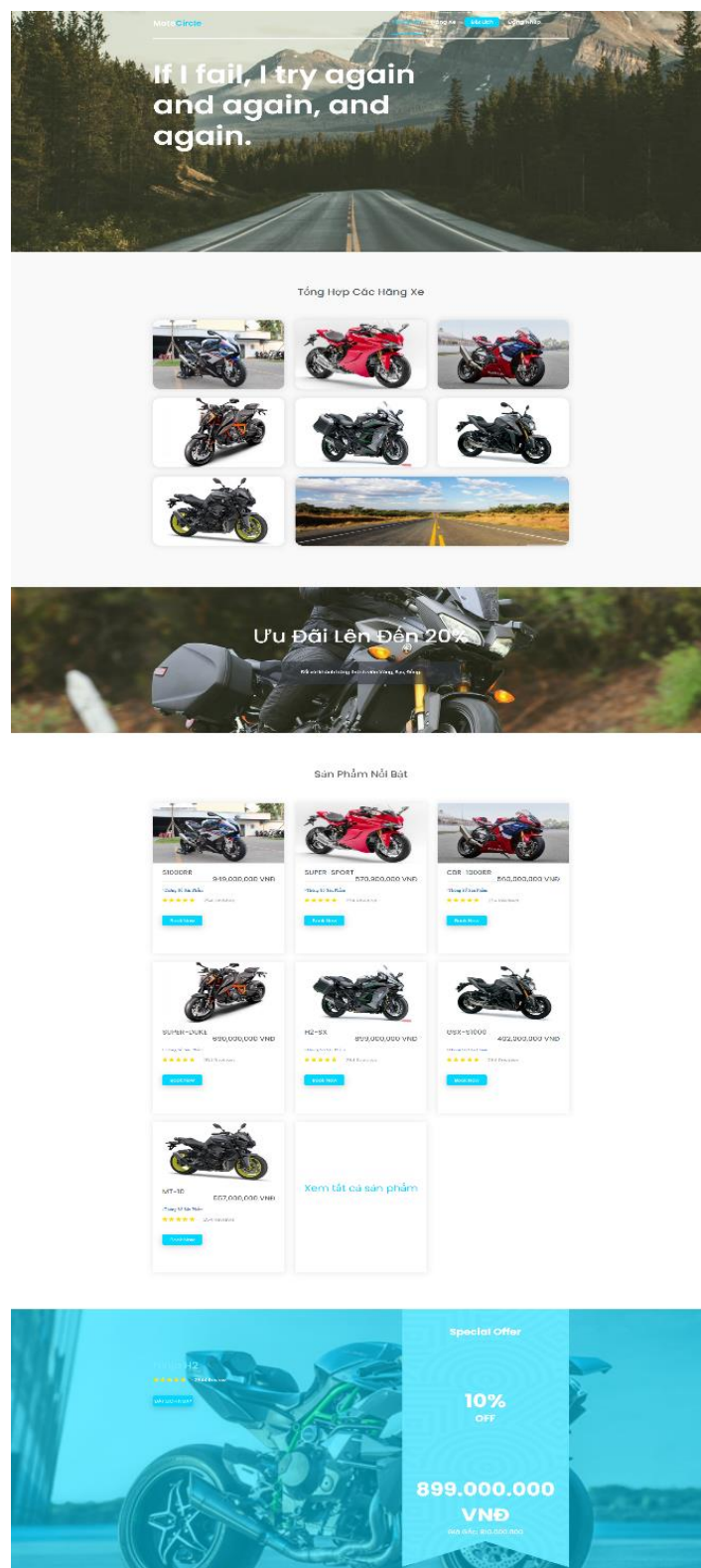
5.1.1.2 Giao diện đăng ký:



Hình 36: Giao diện đăng ký

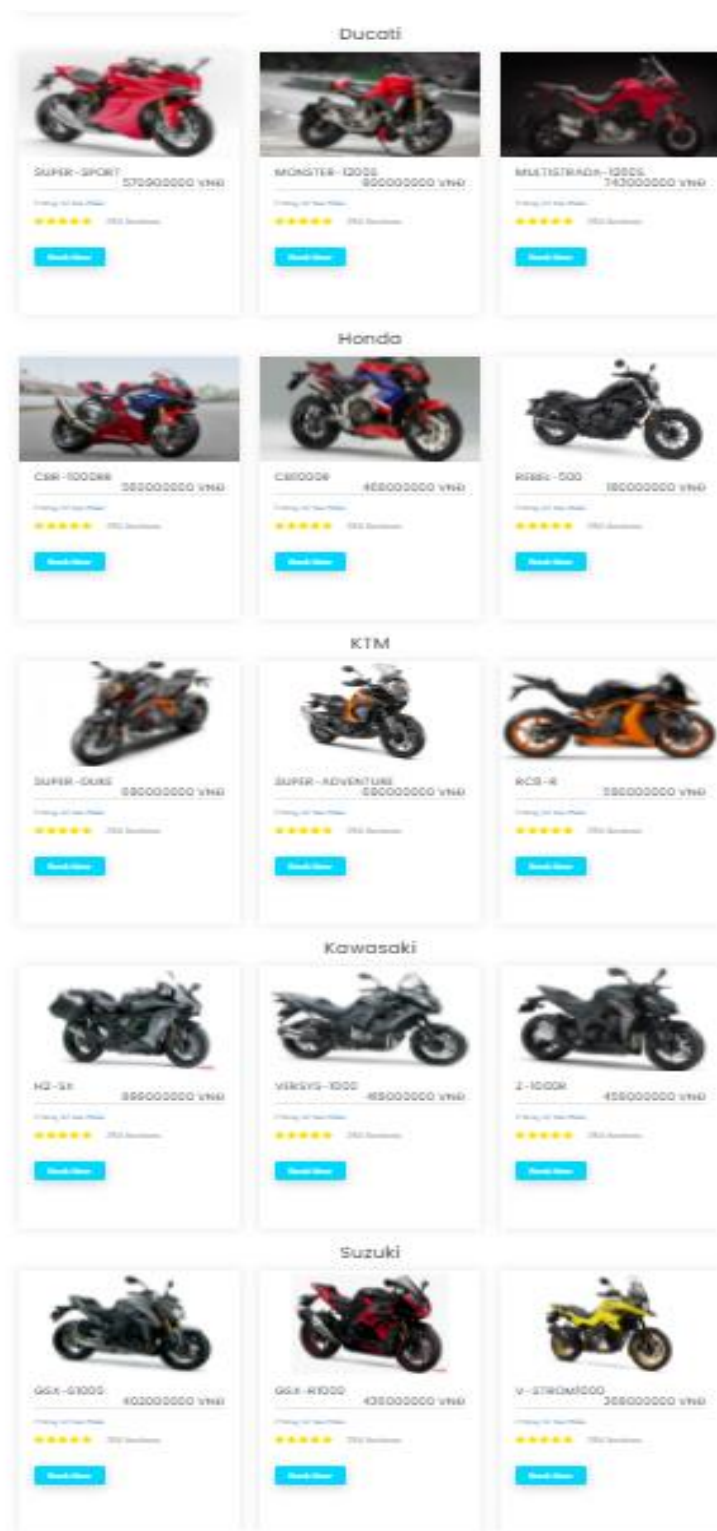
5.1.2 Giao diện User:

5.1.2.1 Giao diện trang chủ



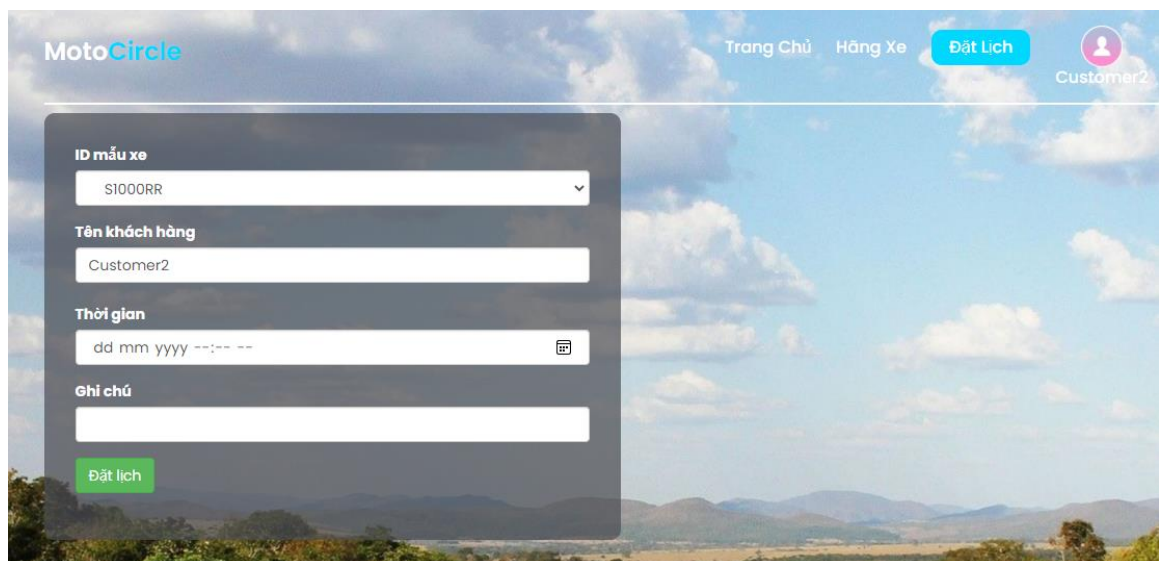
Hình 37: Giao diện trang chủ - Khách hàng

5.1.2.2 Giao diện tra cứu mẫu xe theo hãng xe



Hình 38: Giao diện tra cứu xe theo hãng - Khách hàng

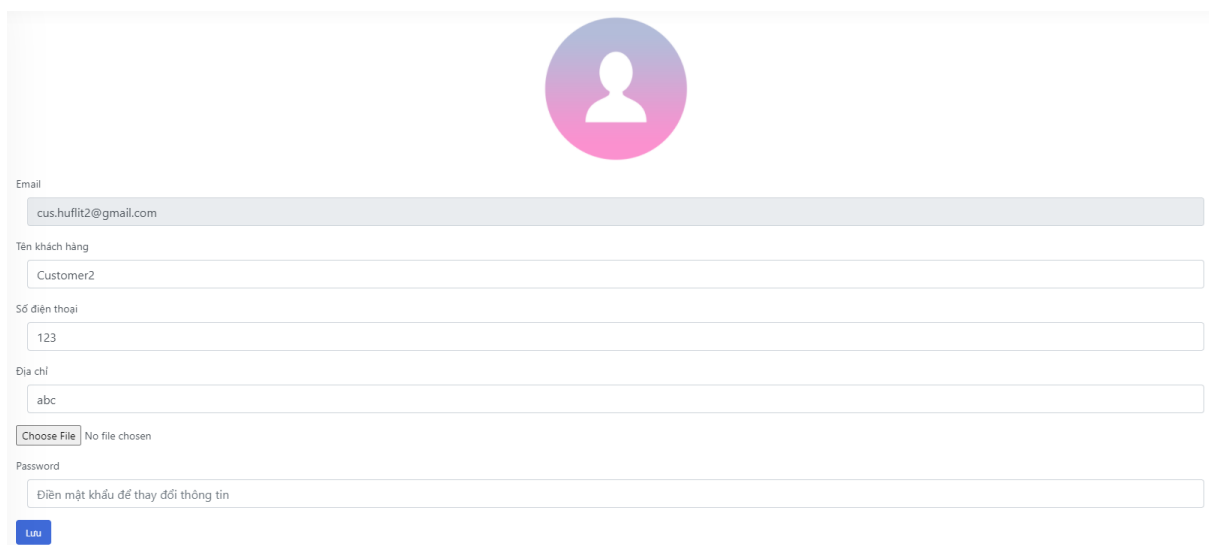
5.1.2.3 Giao diện đặt lịch hẹn



The screenshot shows the MotoCircle website with a booking form overlay. The form includes fields for 'ID mẫu xe' (set to S1000RR), 'Tên khách hàng' (set to Customer2), 'Thời gian' (date and time picker), and 'Ghi chú'. A green 'Đặt lịch' button is at the bottom of the form. The background is a scenic landscape with mountains and a blue sky with clouds. The top navigation bar includes 'Trang Chủ', 'Hãng Xe', and a 'Đặt Lịch' button next to a user profile icon labeled 'Customer2'.

Hình 39: Giao diện đặt lịch hẹn - Khách hàng

5.1.2.4 Giao diện thông tin cá nhân



The screenshot shows a personal information form for a customer. At the top is a circular profile picture placeholder. Below it are input fields for 'Email' (cus.huffit2@gmail.com), 'Tên khách hàng' (Customer2), 'Số điện thoại' (123), and 'Địa chỉ' (abc). There is a 'Choose File' button for a profile picture and a 'Password' field with a hint 'Điền mật khẩu để thay đổi thông tin'. A blue 'Lưu' button is at the bottom left.

Hình 40: Giao diện thông tin cá nhân - Khách hàng

5.1.3 Giao diện Admin:

5.1.3.1 Giao diện quản lý mẫu xe:

ID Mẫu	Hình	Tên xe	Hãng	Giá bán	Bảo hành	Khuyến mãi	Số lượng	
BMW4		BMW-HP4	BMW	110000000 VND	2	0%	6	Chi tiết Sửa
HD2		CB1000R	Honda	468000000 VND	12	0%	5	Chi tiết Sửa
HD1		CBR-1000RR	Honda	560000000 VND	12	0%	5	Chi tiết Sửa
SZ2		GSX-R1000	Suzuki	436000000 VND	12	0%	5	Chi tiết Sửa

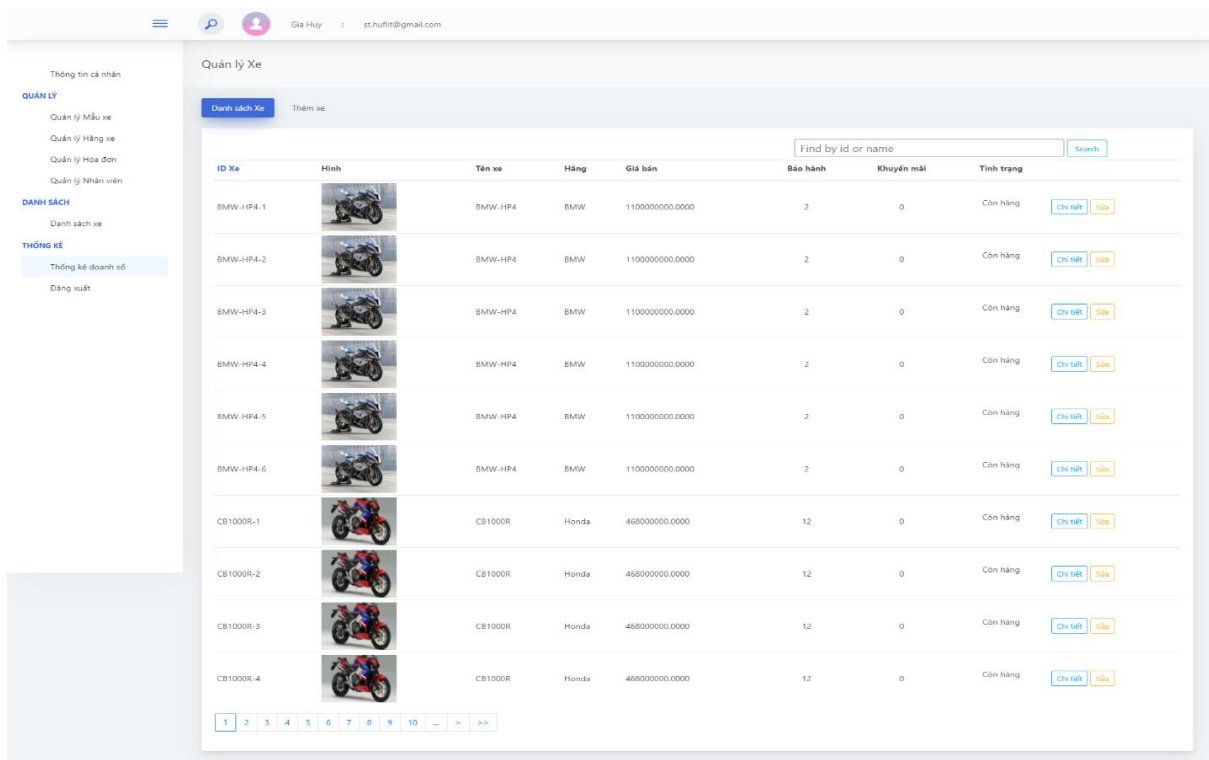
Hình 41: Giao diện quản lý mẫu xe - Quản lý

5.1.3.2 Giao diện quản lý hãng xe:

ID Hãng	Hãng	Số lượng	
BMW	BMW	4	Sửa
DCT	Ducati	3	Sửa
HD	Honda	3	Sửa
KTM	KTM	3	Sửa
KW	Kawasaki	3	Sửa
SZ	Suzuki	3	Sửa
YM	Yamaha	3	Sửa

Hình 42: Giao diện quản lý hãng xe - Quản lý

5.1.3.3 Giao diện quản lý danh sách xe:

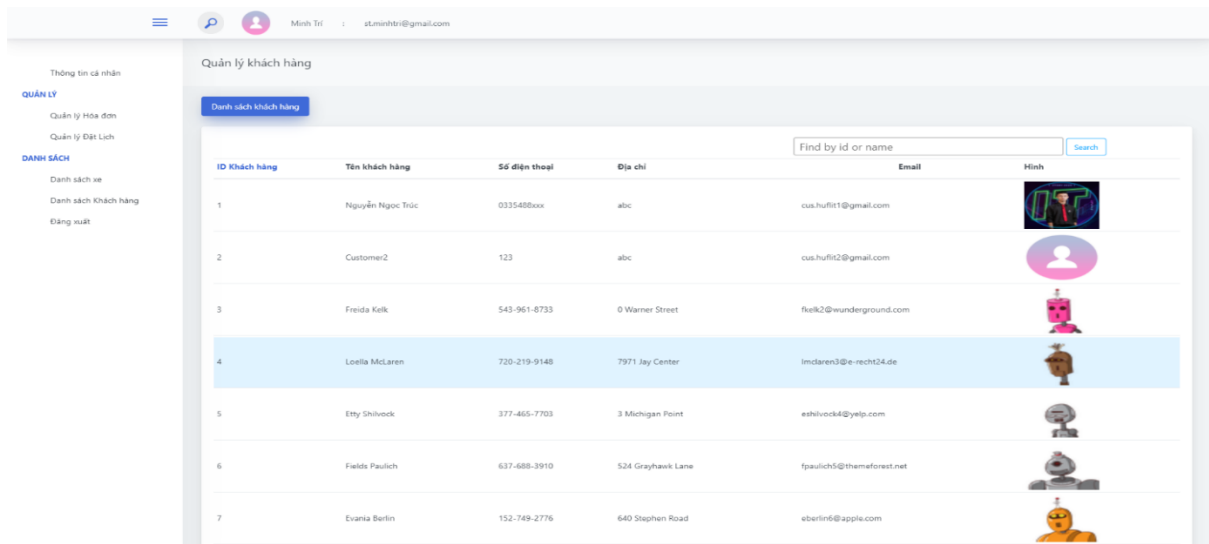


The screenshot displays a web application for managing a vehicle inventory. The interface includes a sidebar with navigation options like 'Quản lý' (Management) and 'Danh sách' (List). The main area is titled 'Quản lý Xe' (Vehicle Management) and features a table listing various motorcycles. Each row represents a vehicle with details such as its ID, image, name, brand, price, warranty, discount, and status. The table is paginated, showing items 1 through 10.

ID Xe	Hình	Tên xe	Hãng	Giá bán	Bảo hành	Khuyến mãi	Tình trạng
BMW-HP4-1		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
BMW-HP4-2		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
BMW-HP4-3		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
BMW-HP4-4		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
BMW-HP4-5		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
BMW-HP4-6		BMW-HP4	BMW	110000000.0000	2	0	Còn hàng
CB1000R-1		CB1000R	Honda	46800000.0000	12	0	Còn hàng
CB1000R-2		CB1000R	Honda	46800000.0000	12	0	Còn hàng
CB1000R-3		CB1000R	Honda	46800000.0000	12	0	Còn hàng
CB1000R-4		CB1000R	Honda	46800000.0000	12	0	Còn hàng

Hình 43: Giao diện quản lý danh sách xe - Nhân viên, Quản lý

5.1.3.4 Giao diện danh sách khách hàng:



The screenshot displays a web application for managing a customer list. The interface includes a sidebar with navigation options like 'Quản lý' (Management) and 'Danh sách' (List). The main area is titled 'Quản lý khách hàng' (Customer Management) and features a table listing various customers. Each row represents a customer with details such as their ID, name, phone number, address, email, and profile picture. The table is paginated, showing items 1 through 7.

ID Khách hàng	Tên khách hàng	Số điện thoại	Địa chỉ	Email	Hình
1	Nguyễn Ngọc Trúc	0335488xxx	abc	cus.huffit1@gmail.com	
2	Customer2	123	abc	cus.huffit2@gmail.com	
3	Freida Kelk	543-961-8733	0 Warner Street	fkell2@wunderground.com	
4	Loella McLaren	720-219-9148	7971 Jay Center	lmclaren3@e-recht24.de	
5	Ethy Shilvork	377-465-7703	3 Michigan Point	eshilvork4@yelp.com	
6	Fields Paulich	637-688-3910	524 Grayhawk Lane	fpaulich5@themaforest.net	
7	Evania Berlin	152-749-2776	640 Stephen Road	eberlin6@apple.com	

Hình 44: Giao diện danh sách khách hàng - Nhân viên

6 Đánh Giá – Hướng Phát Triển

6.1 Ưu điểm

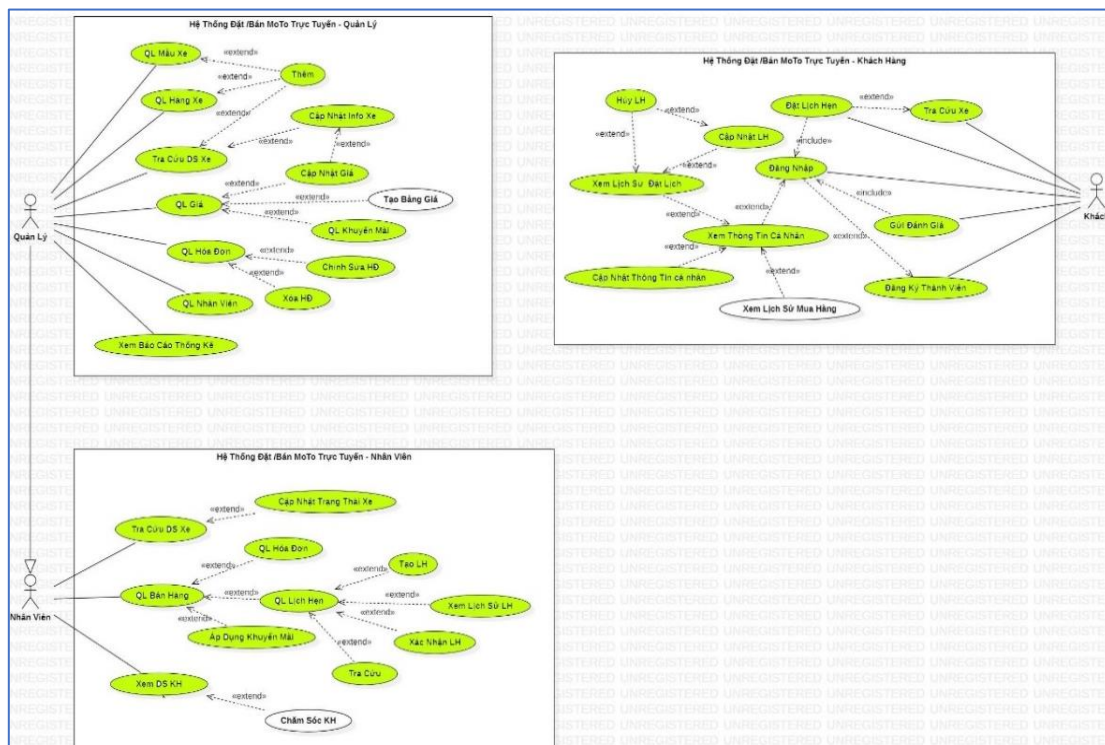
- Giao diện trang web dễ sử dụng
- Bố cục trang web hợp lý, làm được nhiều hiệu ứng
- Ứng dụng được nhiều tính năng của công nghệ vào đề tài
- Code dễ bảo trì, sửa chữa, mở rộng và phát triển

6.2 Nhược điểm

- Chưa tư vấn online được cho khách hàng
- Chưa tự động cập nhật bảng giá theo ngày tháng chỉ định
- Chưa đảm bảo chắc chắn được các yêu cầu về bảo mật

6.3 Đánh giá

- Các usecase đã hoàn thành:



Hình 45: Các Usecase hoàn thiện

- Nhóm tự đánh giá hoàn thiện được 90% các yêu cầu ban đầu đặt ra.

- Do chưa có kinh nghiệm thực tế nên có thể nhóm còn nhiều thiếu sót về mặt nghiệp vụ.
- Kỹ năng, kinh nghiệm xử lý vấn đề còn kém nên vài tính năng nhóm đã có ý tưởng nhưng chưa thực hiện được.

6.4 Hướng phát triển và mở rộng

- Xây dựng tính năng tự động cập nhật bảng giá theo ngày tháng quy định.
- Xây dựng tính năng tư vấn khách hàng online.
- Xây dựng tính năng khuyến mãi theo sự kiện.

7 Kết luận

Kết luận lại, trang web đã được hoàn thiện như yêu cầu ban đầu đặt ra, có những tính năng được bổ sung thêm. Nhưng bên cạnh đó đồ án cũng còn nhiều thiếu sót và khuyết điểm. Thông qua đồ án này, cả nhóm đã cùng nhau học và nghiên cứu được một công nghệ phổ biến mới là ASP.NET CORE và các cách, kỹ năng phân tích vấn đề cũng như cách làm việc nhóm và phát hiện được các lỗi hổng kiến thức chưa vững, cách thức làm việc nhóm cần được cải thiện để tiến độ hoàn thành dự án được tốt hơn.

Tài liệu tham khảo

- [1] E. F. Tutorial, “Entity Framework Core,” [Trực tuyến]. Available: <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>.
- [2] V. Sarcar, “Design Patterns in C__ A Hands-on Guide with Real-world Examples,” 2020. [Trực tuyến]. Available: libgen.li.
- [3] xuanthulab, “(ASP.NET Core) Gửi Mail trong ứng dụng Web Asp.net,” [Trực tuyến]. Available: <https://xuanthulab.net/asp-net-core-gui-mail-trong-ung-dung-web-asp-net.html>.