

TRƯỜNG ĐẠI HỌC QUY NHƠN
KHOA CÔNG NGHỆ THÔNG TIN

------



**BÁO CÁO CUỐI KỲ
PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐÓI
TUỢNG**

**Website bán xe ô tô
ShopCar**

Sinh viên thực hiện : **Lê Phi Khanh**

Võ Quang Bình

Trần Văn Hoài Nhiên

Lớp : **Công nghệ phần mềm K39**

Giáo viên hướng dẫn : **TS. Nguyễn Thanh Bình**

Quy Nhơn, tháng 12 năm 2019

MỤC LỤC

LỜI CẢM ƠN	3
Phần I : Giới Thiệu và Phân Tích đề tài.....	4
I. TỔNG QUAN.....	4
1. Mục tiêu đề tài	4
1.1. Ý tưởng.....	4
1.2. Phát biểu mục tiêu đề tài.....	4
1.3. Các chức năng chính của hệ thống.....	4
II. PHÂN TÍCH THIẾT KẾ	4
1. Yêu cầu chức năng	5
1.1. Sơ đồ use cases	5
1.2. Đặc tả use cases	7
2. Yêu cầu phi chức năng.....	20
3. Các mô hình phân tích.....	21
4. Lược đồ cơ sở dữ liệu.....	35
5. Sơ đồ lớp.....	35
7. Đặc tả giao diện màn hình	36
Phần II : Các Framework sử dụng và Triển Khai	50
I. FRAMEWORK SỬ DỤNG TRONG PROJECT	50
1. Spring MVC.....	50
1.1. Spring MVC (Spring Web MVC)	50
1.2. Flow trong Spring MVC.....	50
2. Spring Security	51
II. CẤU HÌNH VÀ TRIỂN KHAI	62
1. Cấu hình và triển khai lên host	62
2. Cấu hình và triển khai lên localhost	67
Deploy từ Eclipse	68
III. KẾT QUẢ ĐẠT ĐƯỢC	73
1. Kết quả đạt được	73
3. Hạn chế của đề tài	74
3. Hướng phát triển của đề tài.....	74
4. Giao diện	74
CÔNG CỤ HỖ TRỢ	79
TÀI LIỆU THAM KHẢO	79

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Thầy Nguyễn Thanh Bình đã tận tình giúp đỡ trong suốt quá trình học tập và thực hiện để chúng em có thể hoàn thành được môn học và đề tài cuối kì.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài cuối kì nhưng do sự thiếu kinh nghiệm và kiến thức nên còn gặp sai sót trong cách trình bày và triển khai . Kính mong thầy thông cảm và nhận được ý kiến đánh giá của thầy .

Nhóm chúng em xin cảm ơn !

Phần I : Giới Thiệu và Phân Tích đề tài

I. TỔNG QUAN

1. Mục tiêu đề tài

1.1. Ý tưởng

Với nhiều người trong cuộc sống hiện đại ngày nay, việc mua cho mình một chiếc ô tô là rất cần thiết. thậm chí là muôn bán nó đi nhưng không biết bán ở đâu là tiện nhất. Hoặc một số người tài chính còn hạng hẹp thường có nhu cầu mua lại xe cũ nhưng không biết nơi nào là đáng tin cậy. Chính vì nhu cầu này nên thiết kế website mua bán xe ô tô sẽ tạo nên một kênh mua sắm xe an toàn cho mọi người có nhu cầu.

Nắm bắt được cơ hội này thì nhóm đã nảy sinh ra ý tưởng và tạo nên website bán và mua sắm ô tô trực tuyến.

1.2. Phát biểu mục tiêu đề tài

Xây dựng một website mua bán xe ô tô tiện lợi, nhanh chóng, dễ sử dụng cho khách hàng. Đáp ứng nhu cầu cung – cầu của khách hàng, tạo nên một môi trường phát triển lành mạnh, chất lượng và uy tín

Nghiên cứu và kết hợp framework Spring và Hibernate trên nền tảng Java vào phát triển đề tài.

1.3. Các chức năng chính của hệ thống

Đăng Nhập, Đăng Ký

Quản lý tài khoản xe

Quản lý bán xe

Quản lý xem tài khoản xe

Quản lý mua xe

1.4. Các tác nhân

KHÁCH HÀNG: là người giao dịch với hệ thống qua việc đặt hàng, khách hàng có thể chọn các loại xe, mua xe hoặc bán xe. Khách hàng có thể đăng ký làm thành viên của hệ thống.

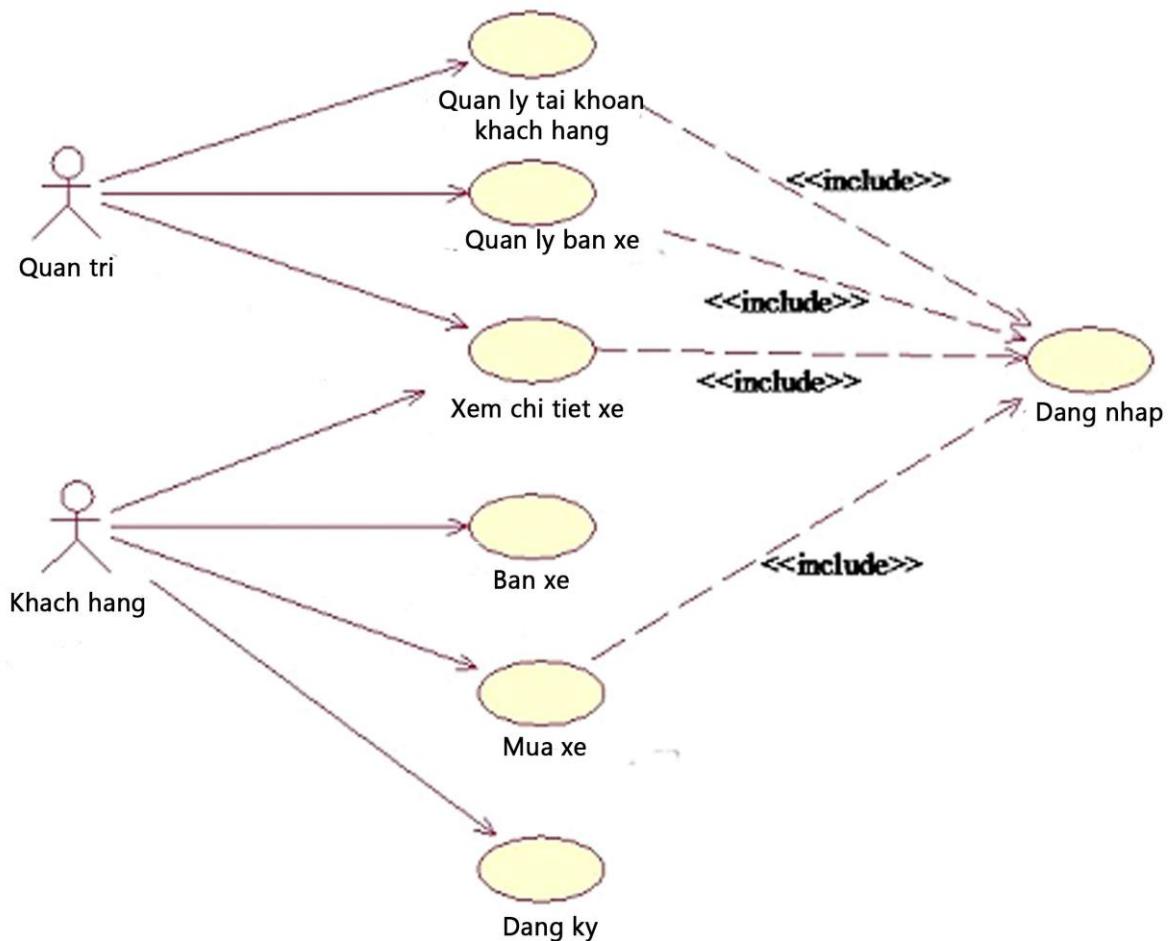
NGƯỜI QUẢN LÝ: là người điều hành, quản lý và đưa ra các hướng giải quyết trên hệ thống

II. PHÂN TÍCH THIẾT KẾ

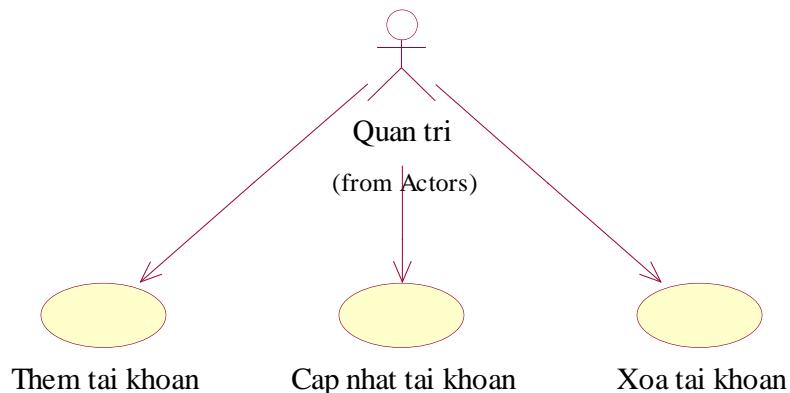
1. Yêu cầu chức năng

1.1. Sơ đồ use cases

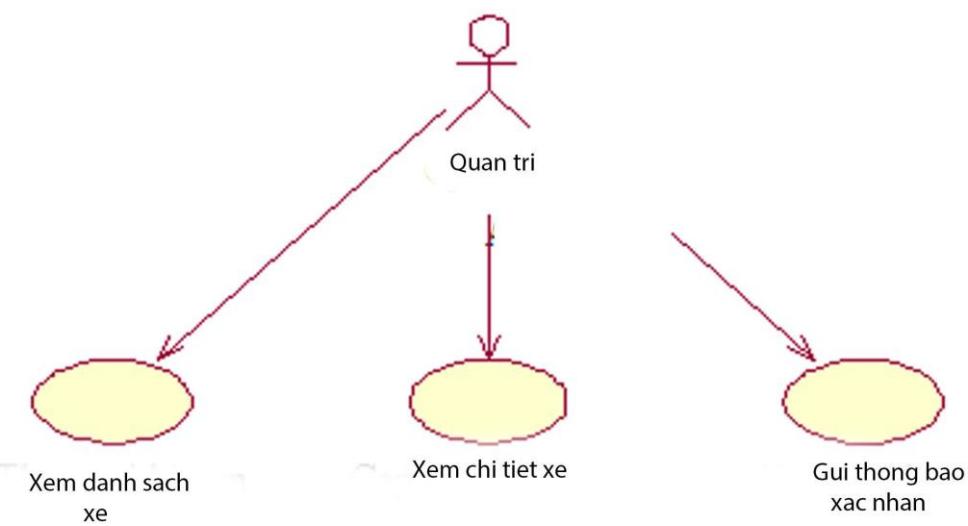
1.1.1. Use case tổng quan hệ thống



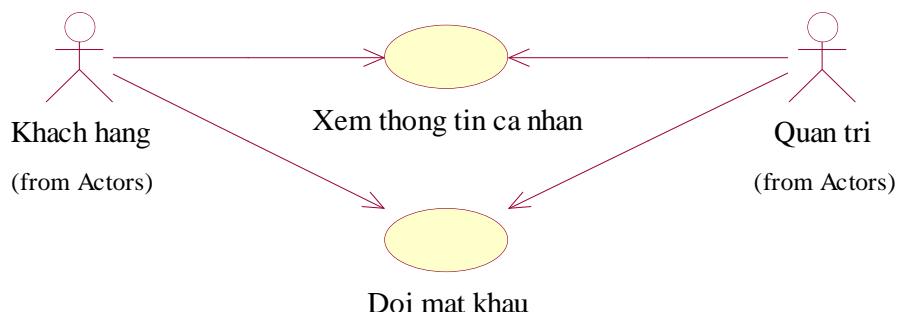
1.1.2. Use case quản lý tài khoản



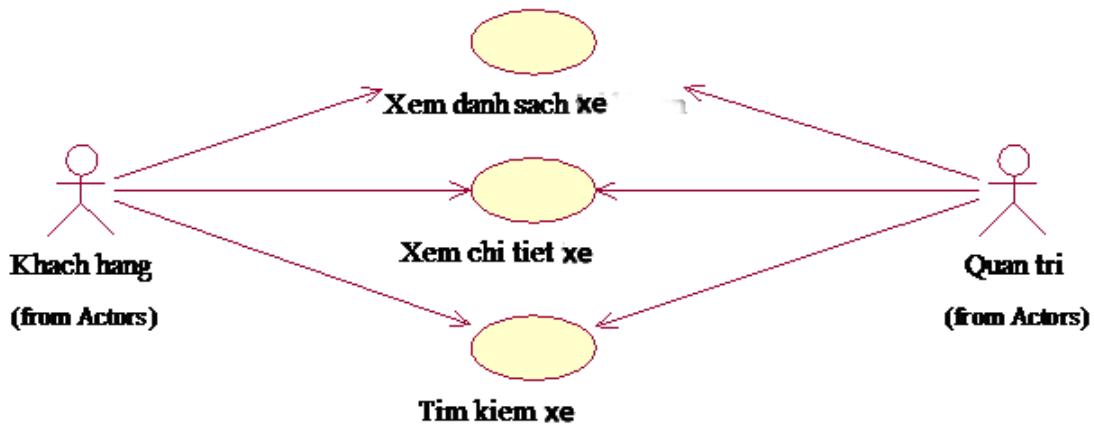
1.1.3. Uses case quản lý bán xe



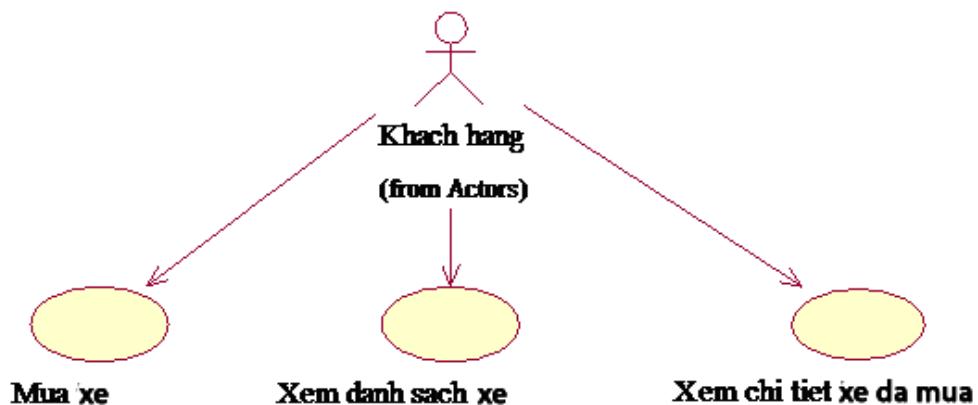
1.1.4. Uses case quản lý thông tin chủ xe



1.1.5. Uses case quản lý xem thông tin xe



1.1.6. Uses case quản lý mua xe



1.2. Đặc tả use cases

1.2.1. Use case đăng ký

Mã use case	UC-01
Tên use case	Đăng ký mới tài khoản
Tác nhân (Actor)	Khách hàng
Level	User goal
Mô tả ngắn (Brief)	Tạo tài khoản để khách hàng đăng nhập và sử dụng các chức năng của hệ thống
Tiền điều kiện(Preconditions)	Khách hàng muốn sử dụng các chức năng của hệ thống
Kết quả(Postconditions)	Khách hàng tạo thành công một tài khoản để đăng nhập và sử dụng. Thông tin được lưu vào CSDL

Điều kiện kích hoạt use case (Triggers – specific business event)	Khách hàng muốn tạo tài khoản của hệ thống
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Khách hàng nhấn nút “ĐĂNG KÝ” trên góc giao diện. 2. Khách hàng điền đầy đủ và đúng các thông tin <ol style="list-style-type: none"> 2.1 Khách hàng nhập tên của họ. 2.2 Khách hàng nhập tên đăng nhập. 2.3 Khách hàng nhập mật khẩu. 2.4 Khách hàng nhập lại mật khẩu. 2.5 Khách hàng nhập email. 2.6 Khách hàng nhập số điện thoại. 3. Nhấn nút “Đăng ký”.
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2:</p> <p>a. Khách hàng không nhập hoặc để trống bất kỳ ô nào => Giao diện hiển thị thông báo “Please fill out this fill” tại ngay ô đó.</p> <p>b. Khách hàng nhập dữ liệu không hợp lệ</p> <p>Ví dụ : Nhập trùng tên đăng nhập => Giao diện hiển thị thông báo “ Tên tài khoản đã được đăng ký, vui lòng nhập lại .“</p> <p>Ví dụ : Số điện thoại sai => Giao diện hiển thị thông báo “ Yêu cầu số điện thoại phải đủ 10 chữ số. “</p>

1.2.2. Use case đăng nhập

Mã use case	UC-02
Tên use case	Đăng nhập
Tác nhân (Actor)	Khách hàng hoặc người quản trị
Level	User goal
Mô tả ngắn (Brief)	Khách hàng hoặc người quản trị đăng nhập vào hệ thống

Tiền điều kiện(Preconditions)	Khách hàng hoặc người quản trị đã có sẵn tài khoản
Kết quả(Postconditions)	Khách hàng hoặc người quản trị đăng nhập được vào hệ thống
Điều kiện kích hoạt use case (Triggers – specific business event)	Khách hàng hoặc người quản trị muốn đăng nhập vào hệ thống
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> Chọn chức năng “ĐĂNG NHẬP” trên góc phải giao diện. Nhập đầy đủ và chính xác tên tài khoản và password Nhấn nút “Đăng nhập”
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2:</p> <p>a. Khách hàng không nhập hoặc bỏ trống bất kỳ ô nào => Giao diện hiển thị thông báo “Please fill out this fill” tại ngay ô đó.</p> <p>b. Nhập sai tên đăng nhập hoặc mật khẩu => Giao diện hiển thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng!!!”.</p>

1.2.3 Use case thêm xe

Mã use case	UC-03
Tên use case	Thêm thông tin bán một chiếc xe
Tác nhân (Actor)	Khách Hàng
Level	User goal
Mô tả ngắn (Brief)	Khách hàng đăng thông tin xe cần bán
Tiền điều kiện(Preconditions)	Khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Một xe mới được gửi lên trang quản lý để quản trị viên xác thực thông tin trước khi đăng lên trang chủ bán xe

Điều kiện kích hoạt use case (Triggers – specific business event)	Muốn thêm mới một xe cần mua hoặc cần bán vào hệ thống
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Khách hàng chọn chức năng “Thêm Xe” 2. Khách hàng cần phải nhập đầy đủ thông tin chi tiết của xe <ol style="list-style-type: none"> 2.1 Khách hàng chọn loại xe trong combobox gồm: <ul style="list-style-type: none"> ➤ XE THỂ THAO ➤ XE GIA ĐÌNH ➤ LUXURY CAR ➤ BOOM CAR 2.2 Khách hàng nhập giá bán, số lượng xe. 2.3 Khách hàng nhập mô tả về xe 2.4 Khách hàng đăng ảnh xe(cần đăng ảnh thực tế để tăng độ xác thực) 2.5 Khách hàng nhập thông tin liên lạc của mình (số điện thoại, email...) 3. Khách hàng chọn “Xác Nhận”
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2:</p> <p>a. Khách hàng không nhập hoặc để trống bất kỳ ô nào =>Giao diện hiển thị thông báo “Please fill out this fill” tại ngay ô đó.</p>

1.2.4. Use case cập nhật tài khoản

Mã use case	UC-04
Tên use case	Cập nhật thông tin xe
Tác nhân (Actor)	Người quản trị
Level	User goal
Mô tả ngắn (Brief)	Người quản trị cập nhật một thông tin xe trong hệ thống

Tiền điều kiện(Preconditions)	Người quản trị phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Một thông tin xe được cập nhật thành công
Điều kiện kích hoạt use case (Triggers – specific business event)	Có yêu cầu xác nhận đăng bán xe từ khách hàng
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Quản trị viên truy cập trang danh sách tài khoản 2. Quản trị viên nhấp nút “XEM YÊU CẦU” 3. Quản trị viên xem xét thông tin của xe đã đầy đủ chưa, và độ tin cậy có cao hay không. 4. Gửi thông báo xác nhận cho chủ xe. 5. Nhấp nút “CẬP NHẬT” trên giao diện.
Luồng sự kiện phụ (Extensions)	<p>Tại bước 5:</p> <p>a. Thông tin chưa đáp ứng yêu cầu =></p>

1.2.5. Use case xóa xe

Mã use case	UC-05
Tên use case	Xóa xe
Tác nhân (Actor)	Người quản trị
Level	User goal
Mô tả ngắn (Brief)	Người quản trị xóa một xe khỏi hệ thống
Tiền điều kiện(Preconditions)	Người quản trị phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Một chiếc xe được xóa thành công
Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên muốn xóa một chiếc xe khỏi hệ thống

Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Quản trị viên truy cập trang danh sách các loại xe 2. Quản trị viên nhấp nút “XÓA” dưới mỗi loại xe 3. Hệ thống hỏi “Bạn có muốn xóa không?” <ul style="list-style-type: none"> ➤ Quản trị viên nhấn “OK”: xóa thành công. ➤ Quản trị viên nhấn “Cancel”: hủy xóa .
Luồng sự kiện phụ (Extensions)	

1.2.6. Use case xem danh sách mua xe

Mã use case	UC-06
Tên use case	Xem danh sách mua xe
Tác nhân (Actor)	Quản trị viên
Level	User goal
Mô tả ngắn (Brief)	Quản trị viên xem danh sách khách hàng mua xe
Tiền điều kiện(Preconditions)	Quản trị viên phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Hiển thị danh sách những khách hàng nào đã mua xe
Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên muốn xem danh sách mua xe
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Quản trị viên nhấp nút “Danh sách mua” trên menu. 2. Hệ thống hiển thị danh sách gồm: <ul style="list-style-type: none"> ➤ STT ➤ Thời gian mua ➤ Tên người mua ➤ Mã loại xe được mua ➤ Trị giá ➤ Chi tiết

Luồng sự kiện phụ (Extensions)	Tại bước 2: <i>a. Không hiển thị danh sách nếu chưa có khách hàng nào mua xe</i>
-----------------------------------	--

1.2.7. Use case xem chi tiết đơn mua

Mã use case	UC-07
Tên use case	Xem chi tiết đơn mua
Tác nhân (Actor)	Quản trị viên
Level	User goal
Mô tả ngắn (Brief)	Quản trị viên xem chi tiết khách hàng mua xe
Tiền điều kiện(Preconditions)	Quản trị viên phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Xem được chi tiết khách hàng nào đã mua
Điều kiện kích hoạt use case (Triggers – specific business event)	Phải tồn tại danh sách mua xe
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Quản trị viên nhấn nút “Chi tiết” trong danh sách mua. 2. Hệ thống hiển thị danh sách gồm: <ul style="list-style-type: none"> ➤ Tên người mua – người bán ➤ Số điện thoại người mua – người bán ➤ Một vài thông tin chi tiết về người mua – người bán
Luồng sự kiện phụ (Extensions)	

1.2.8. Use case xem thông tin cá nhân

Mã use case	UC-08
Tên use case	Xem thông tin cá nhân
Tác nhân (Actor)	Quản trị viên và khách hàng

Level	User goal
Mô tả ngắn (Brief)	Quản trị viên và khách hàng xem thông tin cá nhân
Tiền điều kiện(Preconditions)	Quản trị viên và khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Quản trị viên và khách hàng xem được thông tin cá nhân
Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên và khách hàng muốn xem lại thông tin cá nhân của mình
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> Nhấn nút “Thông tin cá nhân” trên menu. Hệ thống hiển thị danh sách thông tin gồm: <ul style="list-style-type: none"> ➤ Tên đăng nhập ➤ Tên của bạn ➤ Địa chỉ email ➤ Số điện thoại
Luồng sự kiện phụ (Extensions)	

1.2.9, Use case đổi mật khẩu

Mã use case	UC-9
Tên use case	Đổi mật khẩu
Tác nhân (Actor)	Quản trị viên và khách hàng
Level	User goal
Mô tả ngắn (Brief)	Quản trị viên và khách hàng xem đổi mật khẩu của mình
Tiền điều kiện(Preconditions)	Quản trị viên và khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Quản trị viên và khách hàng đổi mật khẩu thành công

Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên và khách hàng muốn đổi lại mật khẩu của mình
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Nhấn nút “Đổi mật khẩu” trên menu. 2. Nhập mật khẩu hiện tại 3. Nhập mật khẩu mới 4. Nhập lại mật khẩu mới 5. Nhấn nút “Đổi mật khẩu”
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2-4:</p> <p>a. Khách hàng không nhập hoặc để trống bất kỳ ô nào =>Giao diện hiển thị thông báo “Please fill out this fill” tại ngay ô đó.</p> <p>Tại bước 2:</p> <p>a. Khách hàng nhập mật khẩu hiện tại không đúng =>Giao diện hiển thị thông báo “Mật khẩu hiện tại không đúng”.</p> <p>Tại bước 4:</p> <p>a. Khách hàng nhập mật khẩu mới không đúng =>Giao diện hiển thị thông báo “Mật khẩu nhập lại không trùng với mật khẩu mới”.</p>

1.2.10. Use case xem danh sách các loại xe

Mã use case	UC-10
Tên use case	Xem danh sách các loại xe
Tác nhân (Actor)	Quản trị viên và khách hàng
Level	User goal
Mô tả ngắn (Brief)	Quản trị viên và khách hàng xem danh sách các loại xe
Tiền điều kiện(Preconditions)	Quản trị viên và khách hàng truy cập vào trang chủ

Kết quả(Postconditions)	Quản trị viên và khách hàng xem được danh sách các loại xe
Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên và khách hàng muốn xem danh sách các loại xe
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Nhấn nút “CAR LIST” trên menu. 2. Giao diện danh sách các loại xe hiện ra với từng loại gồm: <ul style="list-style-type: none"> ➤ Mã xe ➤ Hình ảnh xe ➤ Thông tin đăng ký ➤ Chuyên ➤ Giá bán

1.2.11.. Use case xem chi tiết xe

Mã use case	UC-11
Tên use case	Xem chi tiết xe
Tác nhân (Actor)	Quản trị viên
Level	User goal
Mô tả ngắn (Brief)	Quản trị viên xem chi tiết xe
Tiền điều kiện(Preconditions)	Quản trị viên truy cập vào danh sách xe
Kết quả(Postconditions)	Quản trị viên xem được chi tiết xe
Điều kiện kích hoạt use case (Triggers – specific business event)	Quản trị viên muốn xem danh sách xe
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Nhấn vào hình ảnh của xe .

(Main scenario, basic flow)	<p>2. Giao diện chi tiết của xe đó hiện ra gồm:</p> <ul style="list-style-type: none"> ➤ Mã xe ➤ Những hình ảnh của xe ➤ Thông tin đăng ký ➤ ChuyênGiá bán
Luồng sự kiện phụ (Extensions)	

1.2.12 Use case tìm kiếm xe

Mã use case	UC-12
Tên use case	Tìm kiếm xe
Tác nhân (Actor)	Khách hàng và quản trị viên
Level	User goal
Mô tả ngắn (Brief)	Khách hàng và quản trị viên tìm kiếm xe
Tiền điều kiện(Preconditions)	Khách hàng và quản trị viên truy cập vào danh sách các loại xe
Kết quả(Postconditions)	Khách hàng và quản trị viên tìm kiếm được xe tương ứng .
Điều kiện kích hoạt use case (Triggers – specific business event)	Khách hàng và quản trị viên muốn tìm kiếm xe
Luồng sự kiện chính (Main scenario, basic flow)	<ul style="list-style-type: none"> ➤ Nhập mã số xe cần tìm kiếm ➤ Chọn giá tiền trong combobox với các khoảng giá: <ul style="list-style-type: none"> ➤ Dưới 1 triệu \$ ➤ Từ 1-2 triệu \$ ➤ Trên 2 triệu \$ ➤ Chọn thể loại chuyên trong combobox gồm: <ul style="list-style-type: none"> ➤ TẤT CẢ (mặc định) ➤ XE THÊ THAO

	<ul style="list-style-type: none"> ➤ XE GIA ĐÌNH ➤ LUXURY CAR ➤ BOOM CAR ➤ Nhấn nút “Tìm kiếm”
Luồng sự kiện phụ (Extensions)	<p>Tại bước 4:</p> <p>a. <i>Tìm không thấy xe tương ứng</i> =>Giao diện hiển thị thông báo “Không tìm thấy xe”.</p>

1.2.13 Use case mua xe

Mã use case	UC-13
Tên use case	Mua xe
Tác nhân (Actor)	Khách hàng
Level	User goal
Mô tả ngắn (Brief)	Khách hàng mua một chiếc xe trong hệ thống
Tiền điều kiện(Preconditions)	Khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Khách hàng mua thành công một chiếc xe trong hệ thống
Điều kiện kích hoạt use case (Triggers – specific business event)	<p>Khách hàng muốn mua một chiếc xe trong hệ thống</p> <p>Khách hàng phải truy cập vào trang chi tiết</p>
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Nhấn vào xe cần mua trong danh sách 2. Nhấn nút “MUA NGAY” 3. Thanh toán qua online qua thẻ . 4. Nhấn nút “XÁC NHẬN MUA”
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2:</p> <p>a. <i>Khách hàng chưa đăng nhập</i> =>Giao diện hiển thị thông báo “Bạn chưa đăng nhập. Vui lòng đăng nhập và quay trở lại sau.”. Và có</p>

	<p>nút “ĐĂNG NHẬP” bên cạnh.</p> <p><i>b. Khách hàng không đủ tiền để mua xe</i></p> <p>=> Giao diện hiển thị thông báo “Số tiền của bạn không đủ. Vui lòng nạp tiền và quay lại sau. Hoặc bạn cũng có thể chuyển tiền qua ATM hoặc ví điện tử.”. Và có nút “Chuyển tiền qua ATM – Ví điện tử” bên cạnh.</p>
--	---

1.2.14 Use case xem danh sách xe đã mua

Mã use case	UC-14
Tên use case	Xem danh sách xe đã mua
Tác nhân (Actor)	Khách vàng
Level	User goal
Mô tả ngắn (Brief)	Khách hàng xem danh sách xe mình đã mua
Tiền điều kiện(Preconditions)	Khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Hiển thị danh sách xe khách hàng đã mua
Điều kiện kích hoạt use case (Triggers – specific business event)	Khách hàng muốn xem danh sách xe mà mình đã mua
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Khách hàng nhấn nút “xe đã mua” trên menu. 2. Hệ thống hiển thị danh sách gồm: <ul style="list-style-type: none"> ➤ STT ➤ Thời gian ➤ Mã xe đã mua ➤ Trị giá ➤ Chi tiết
Luồng sự kiện phụ (Extensions)	<p>Tại bước 2:</p> <p><i>a. Không hiển thị danh sách nếu khách hàng chưa mua xe</i></p>

	<i>nào</i>
--	------------

1.2.15 Use case xem chi tiết xe đã mua

Mã use case	UC-15
Tên use case	Xem chi tiết xe đã mua
Tác nhân (Actor)	Khách hàng
Level	User goal
Mô tả ngắn (Brief)	Khách hàng xem chi tiết xe mình đã mua
Tiền điều kiện(Preconditions)	Khách hàng phải đăng nhập vào hệ thống
Kết quả(Postconditions)	Xem được chi tiết xe mà mình đã mua
Điều kiện kích hoạt use case (Triggers – specific business event)	Phải tồn tại danh sách xe đã mua
Luồng sự kiện chính (Main scenario, basic flow)	<ol style="list-style-type: none"> 1. Khách hàng nhấn nút “Chi tiết” trong danh sách các loại xe đã mua. 2. Hệ thống hiển thị danh sách gồm: <ul style="list-style-type: none"> ➤ Tên xe ➤ Chuyên thể loại gì ➤ Thông tin đăng ký ➤ Thông tin của xe về giá xe, người bán,...
Luồng sự kiện phụ (Extensions)	

2. Yêu cầu phi chức năng

2.1. Môi trường

Hệ thống hoạt động trên Google Chrome, FireFox và CocCoc.

2.2. Hiệu suất

Chức năng tìm kiếm xe trả kết quả trong vòng tối đa 2 giây.

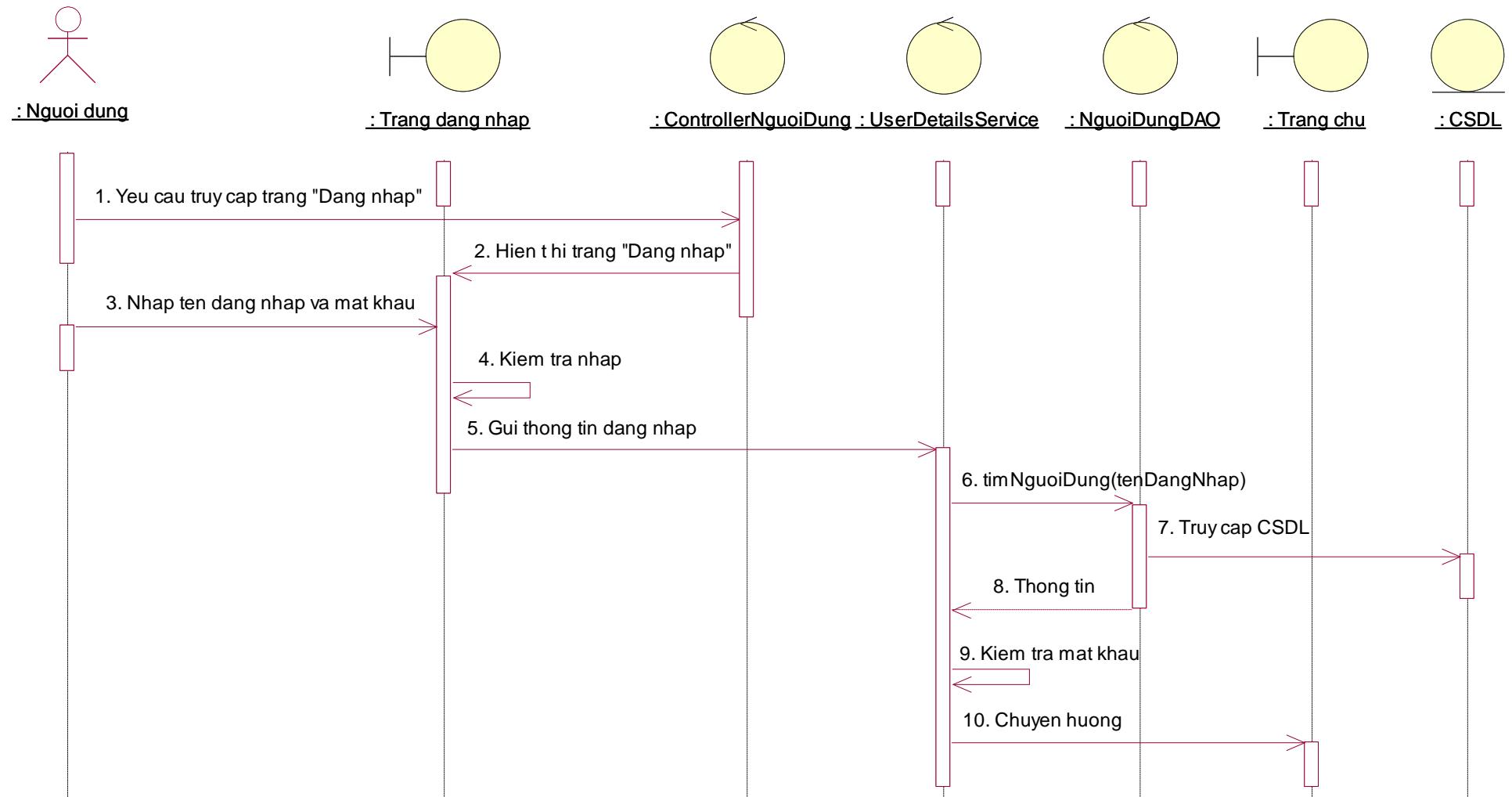
Thời gian hiển thị thông tin chi tiết của một chiếc xe trong vòng 2 giây.

Hệ thống có thể xử lý tối đa 1000 người dùng cùng lúc.

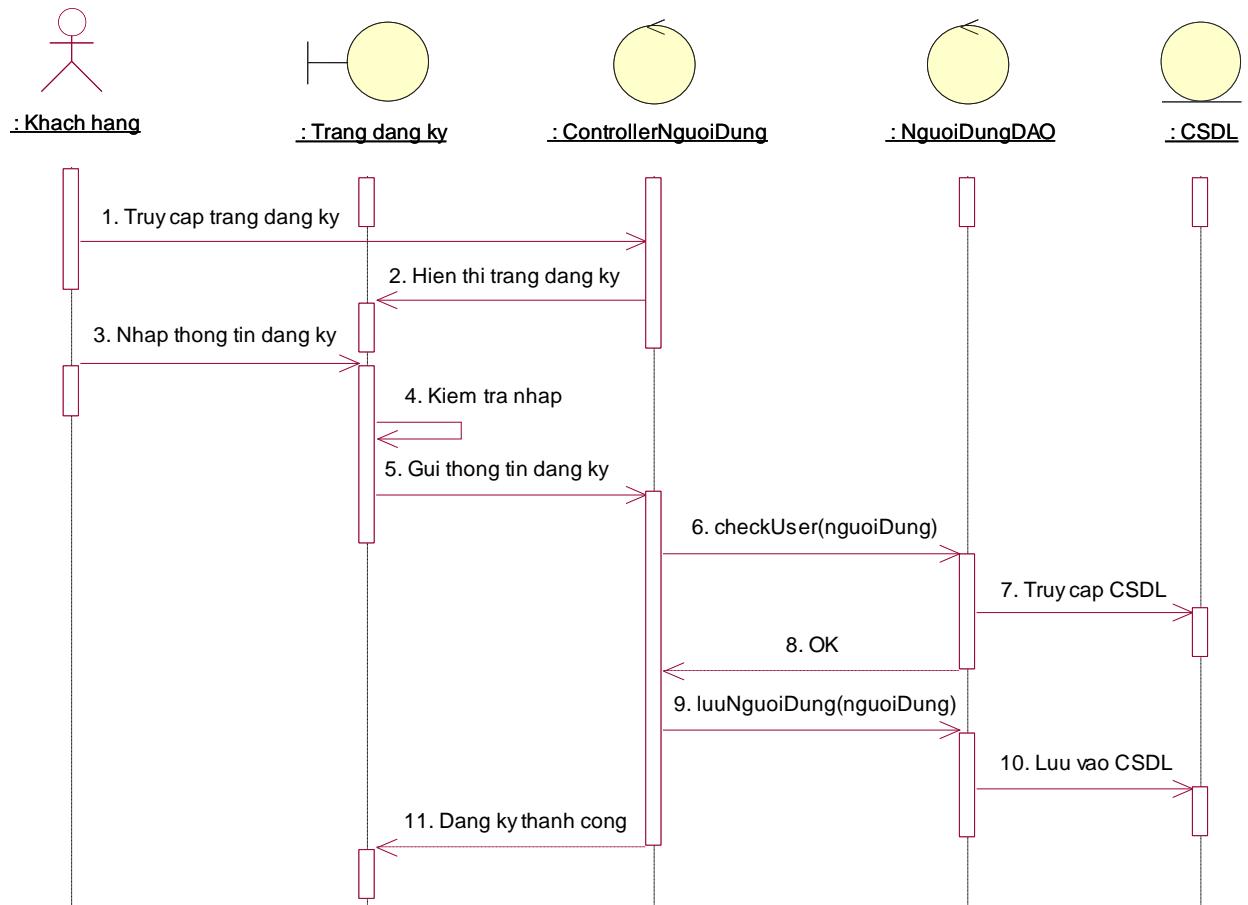
3. Các mô hình phân tích

3.1. Sơ đồ Activity và Sequence

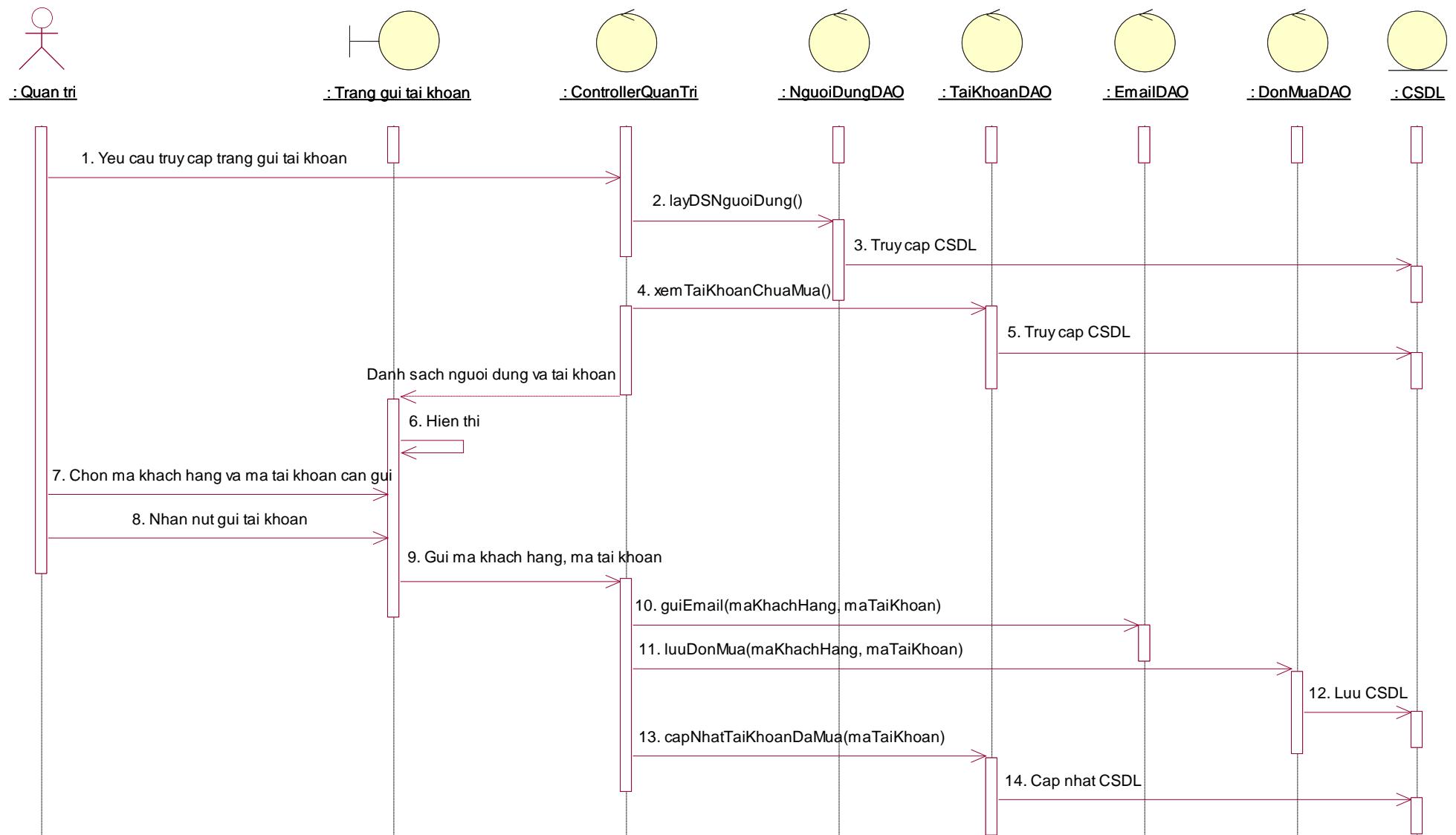
3.1.1 Chức năng đăng nhập



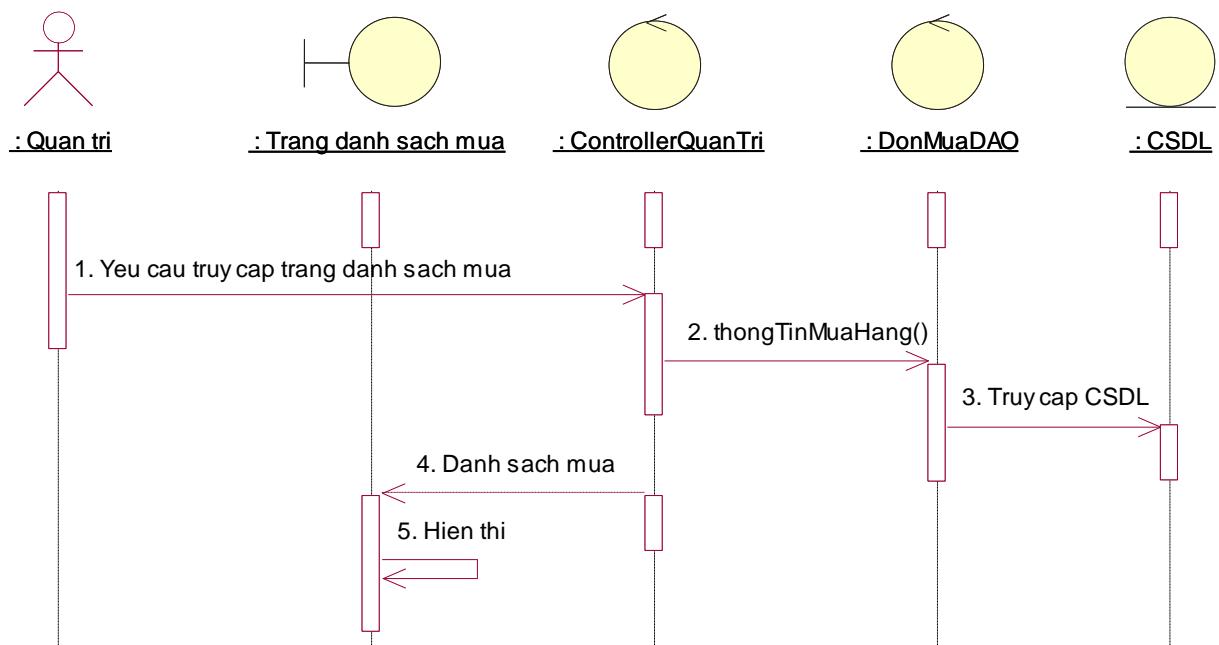
3.1.2 Chức năng đăng ký



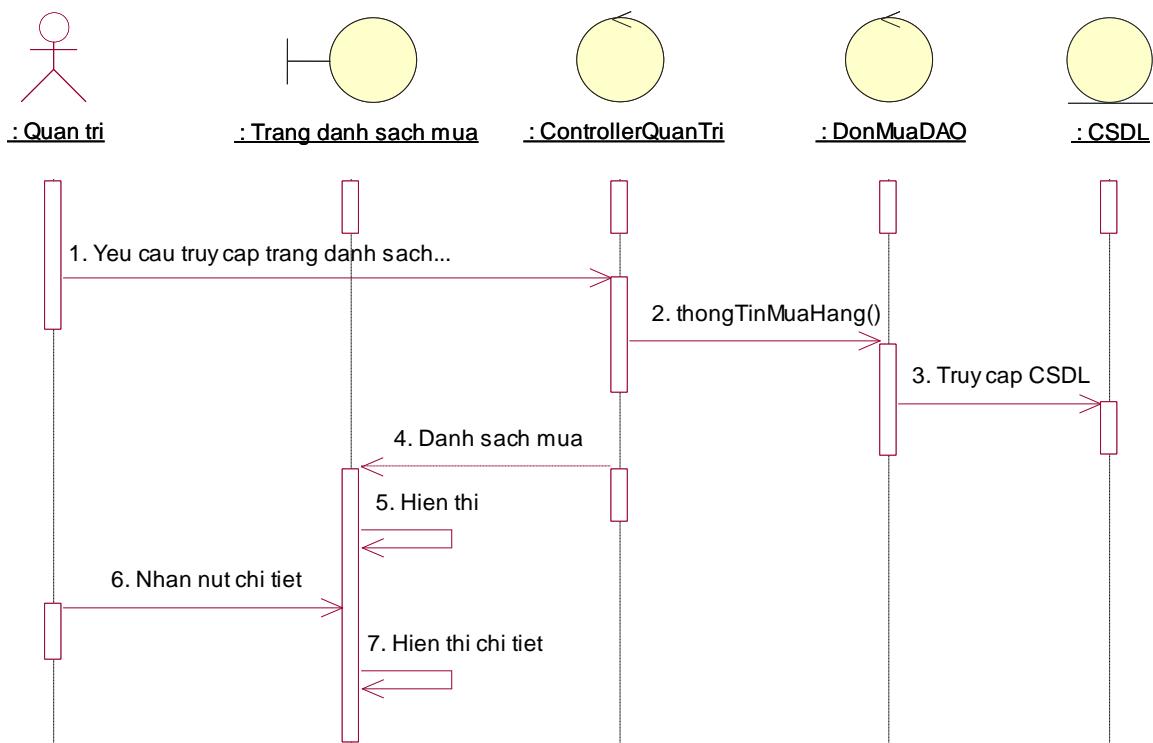
3.1.3. Chức năng gửi thông tin xe cho khách hàng



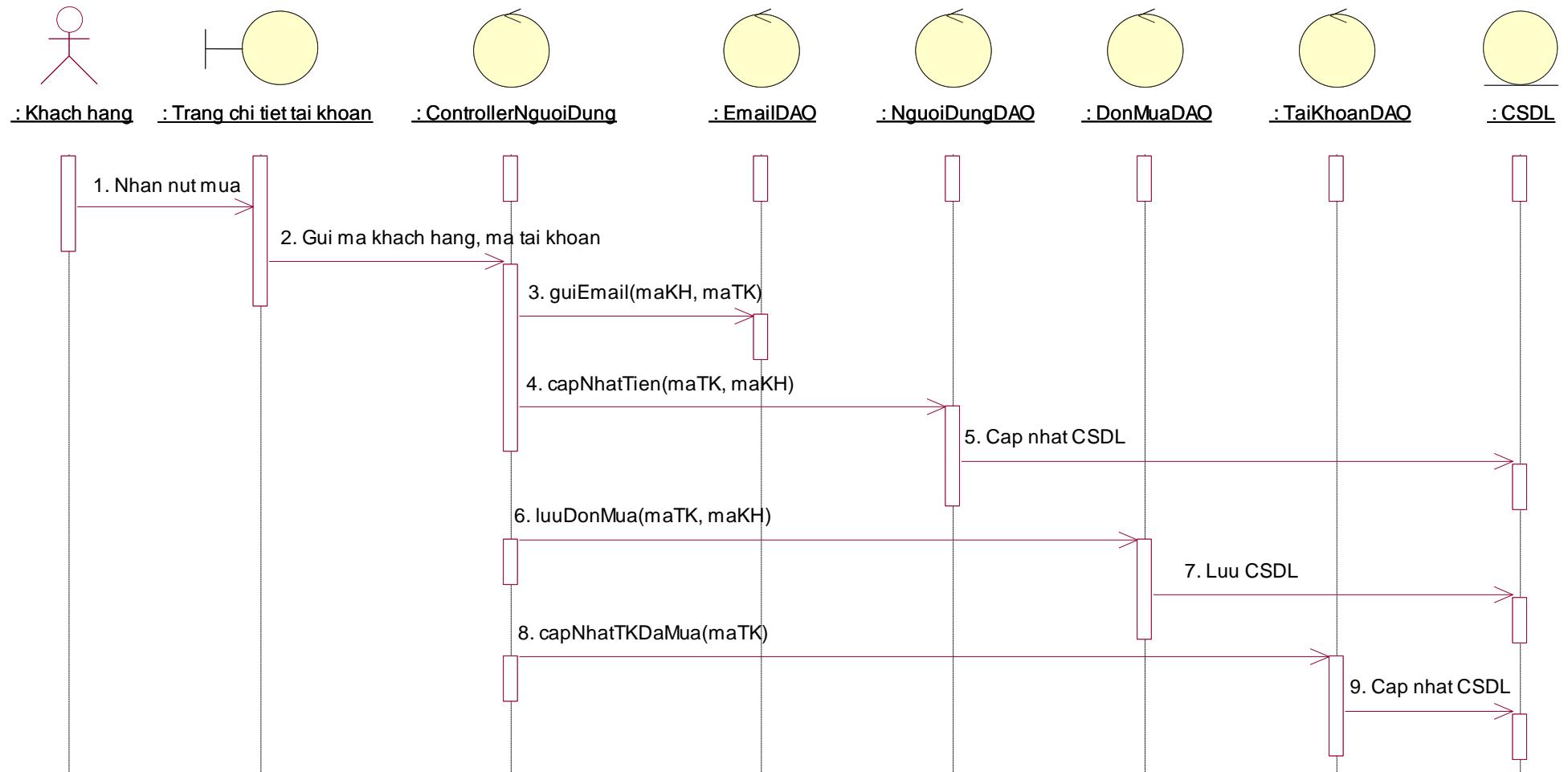
3.1.4. Chức năng xem danh sách mua cho quản trị viên



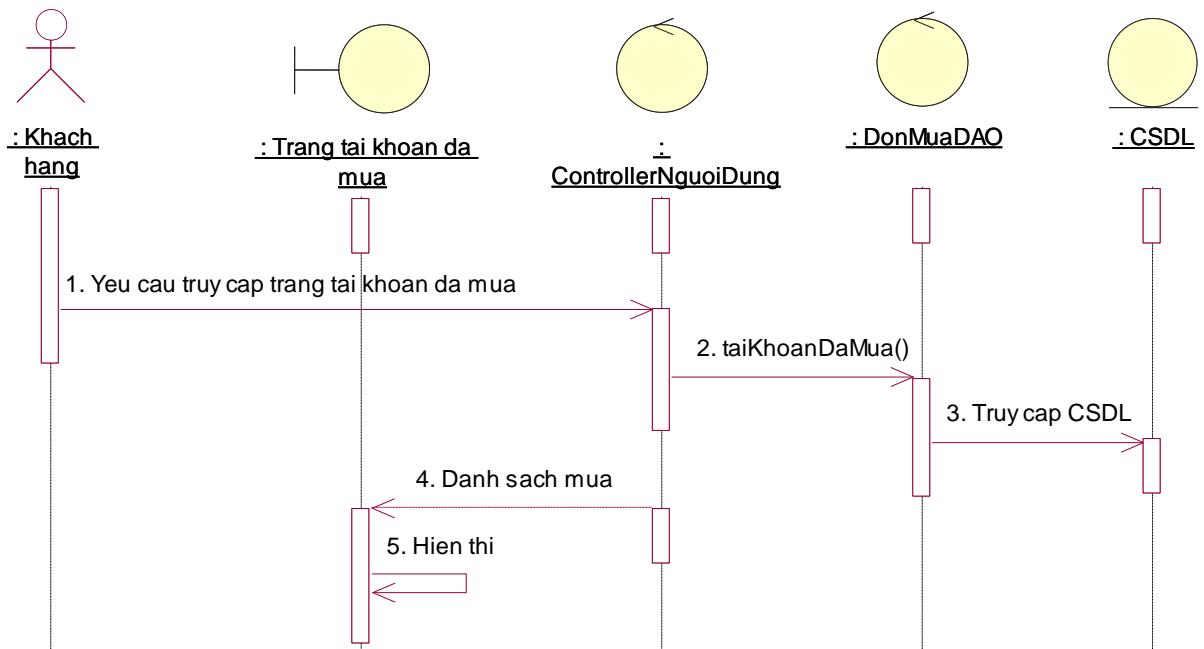
3.1.5. Chức năng xem chi tiết người mua xe cho quản trị viên



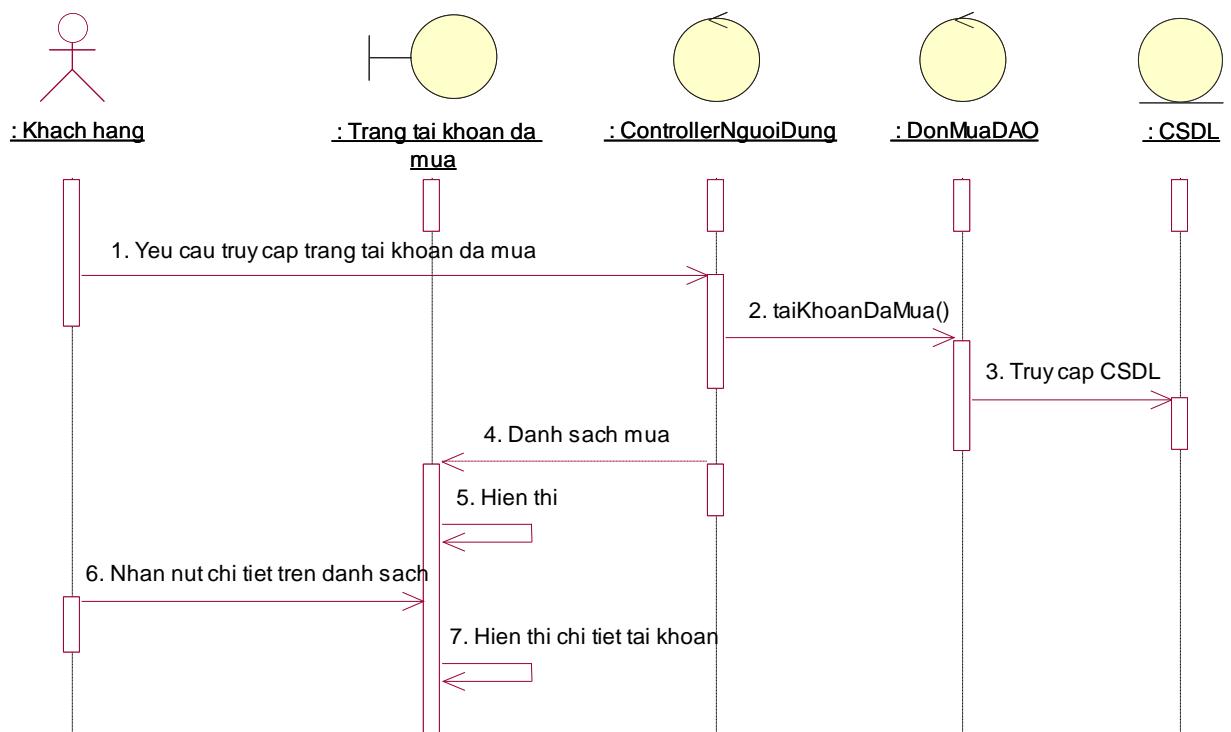
3.1.6. Chức năng mua xe



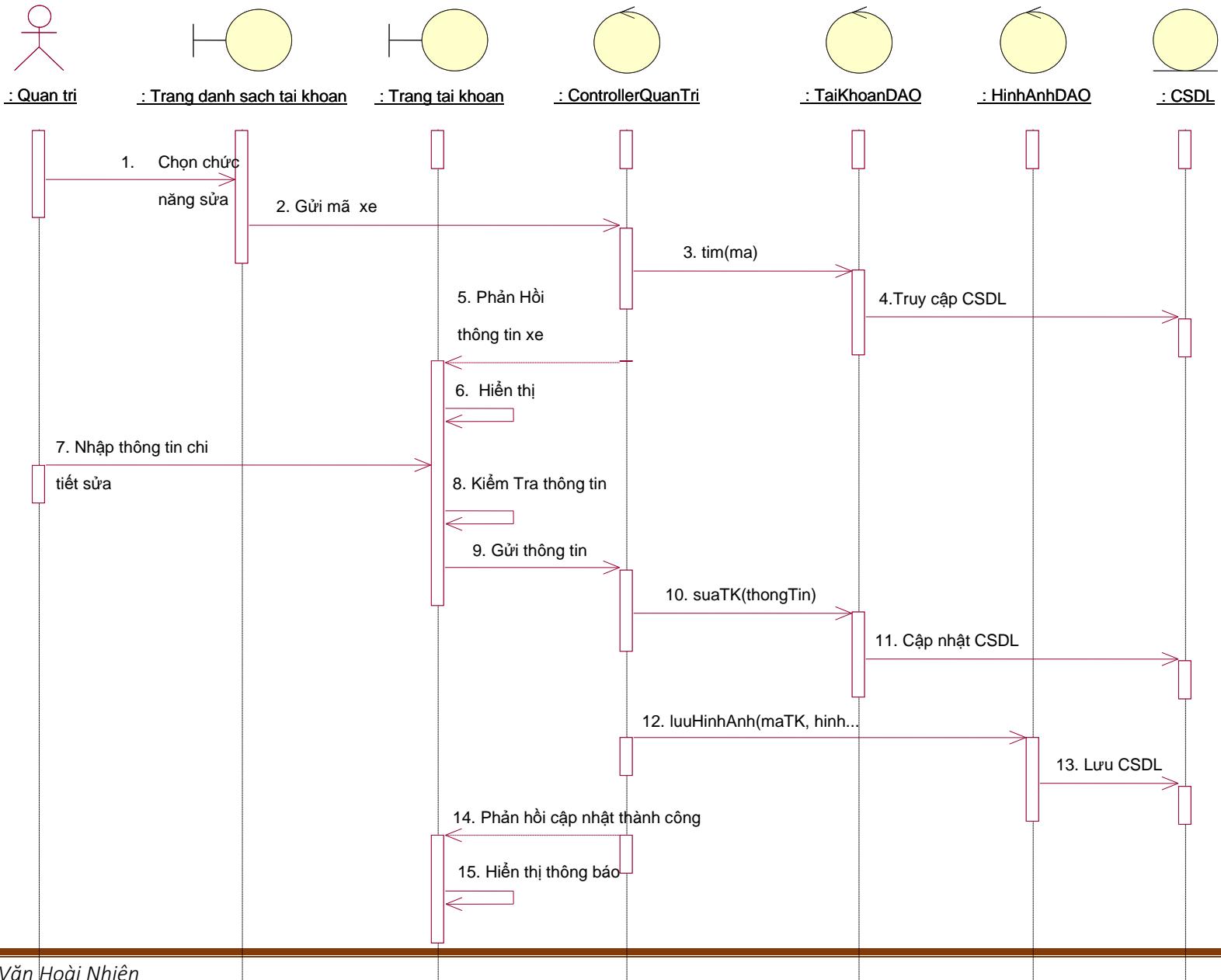
3.1.7. Chức năng xem danh sách xe đã mua



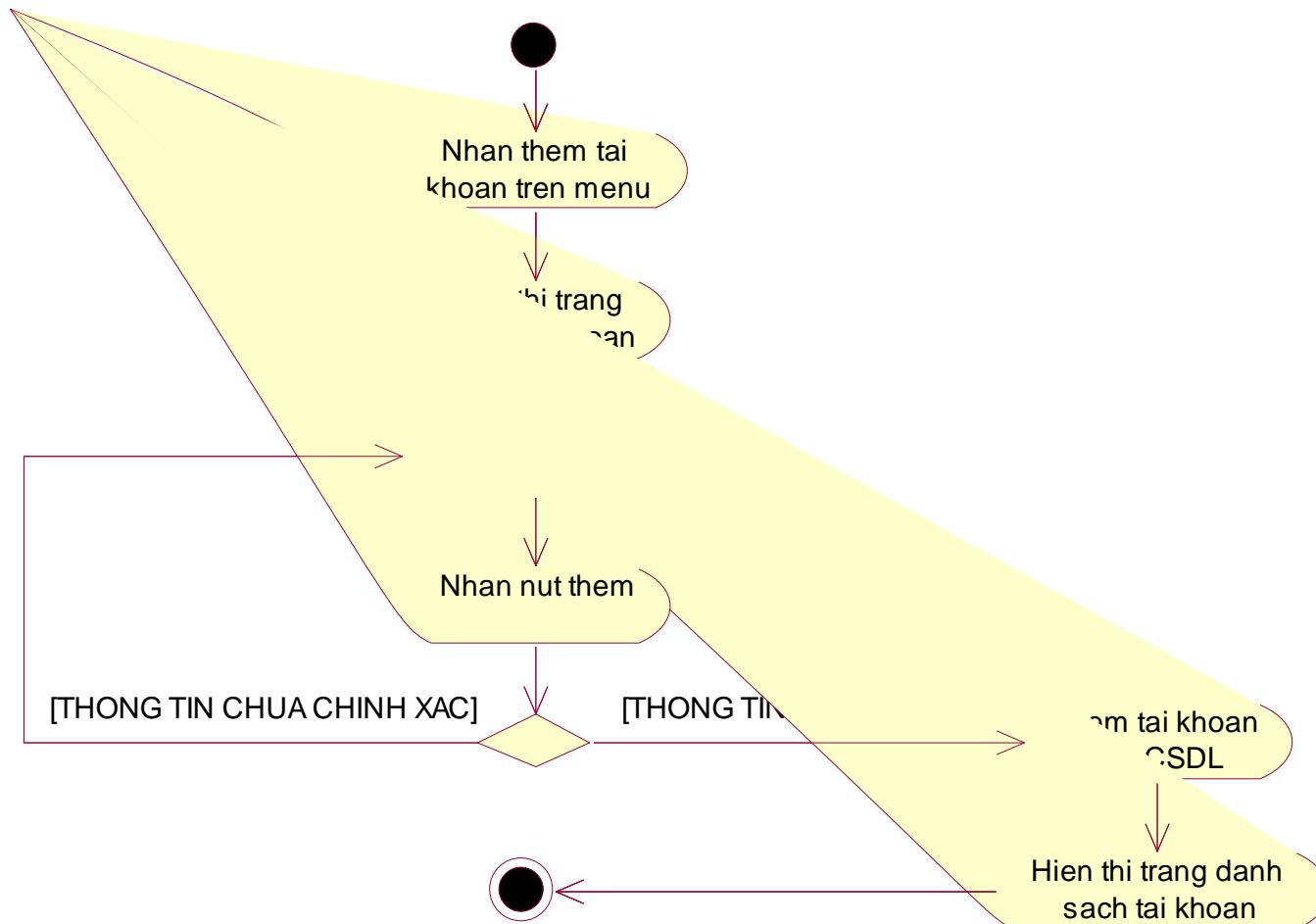
3.1.8. Chức năng xem chi tiết xe đã mua



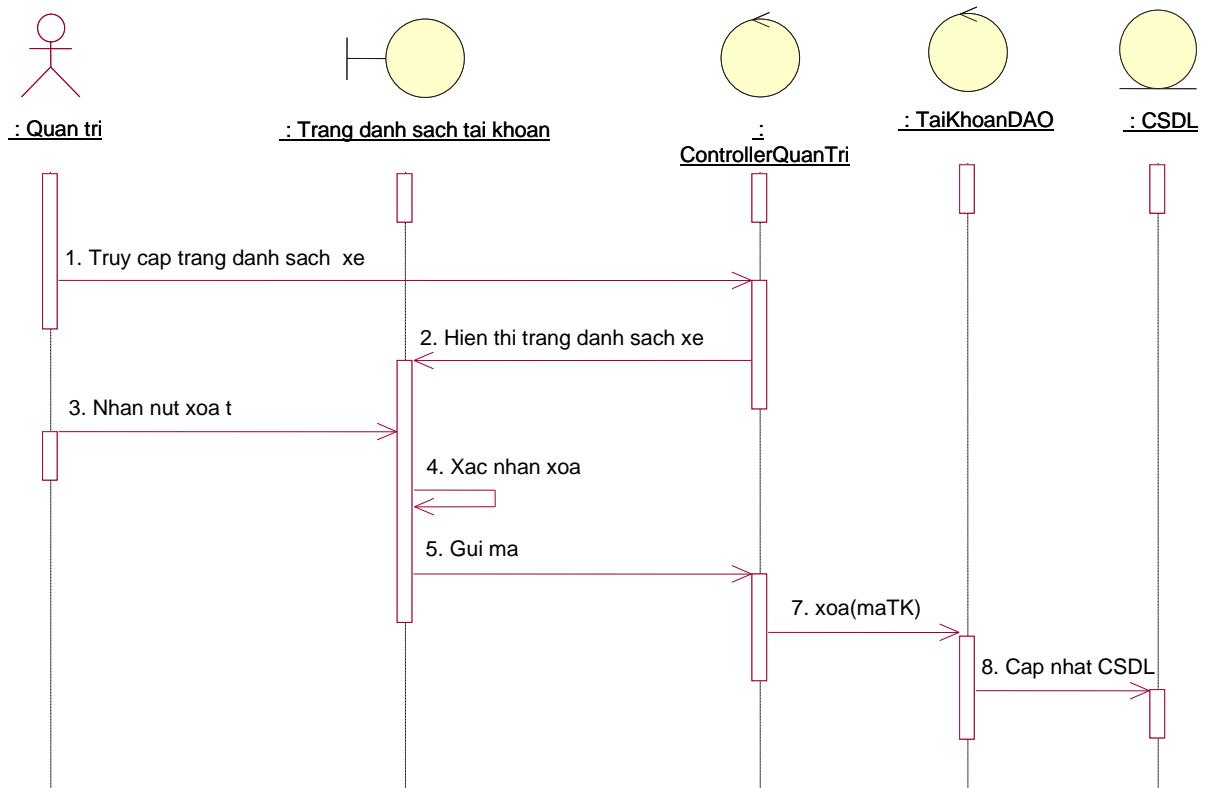
3.1.9. Chức năng cập nhật thông tin xe



3.1.10. Chức năng thêm xe

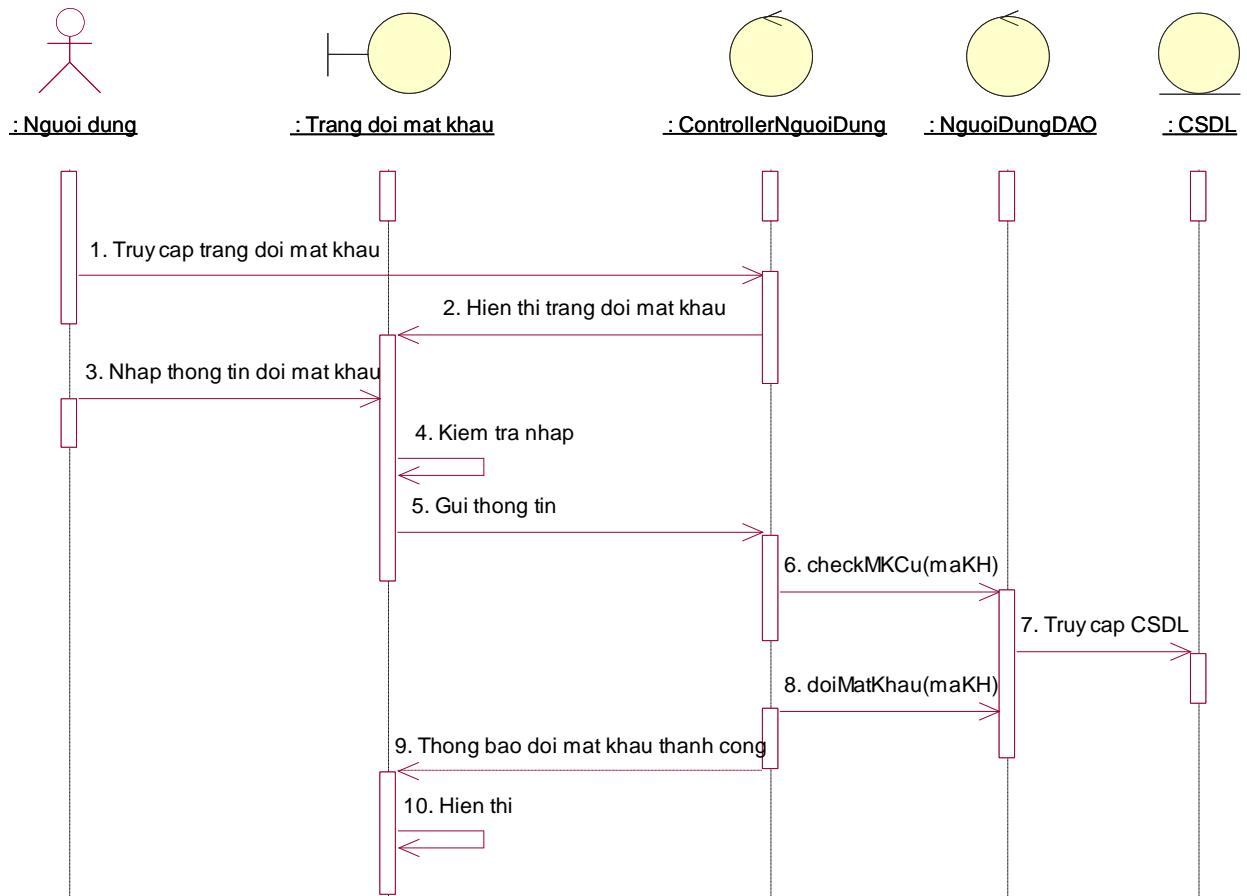


3.1.11. Chức năng xóa xe

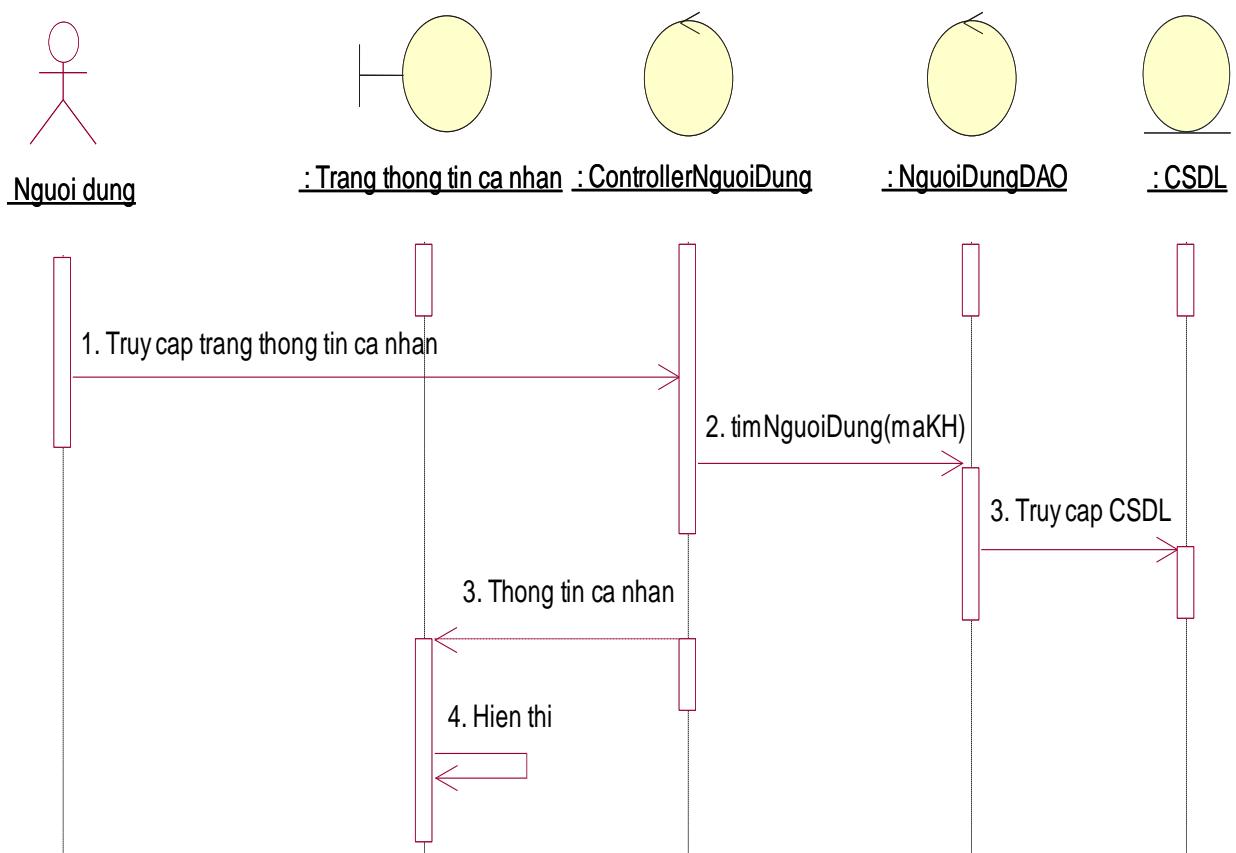


3.1.12. Chức năng đổi mật khẩu

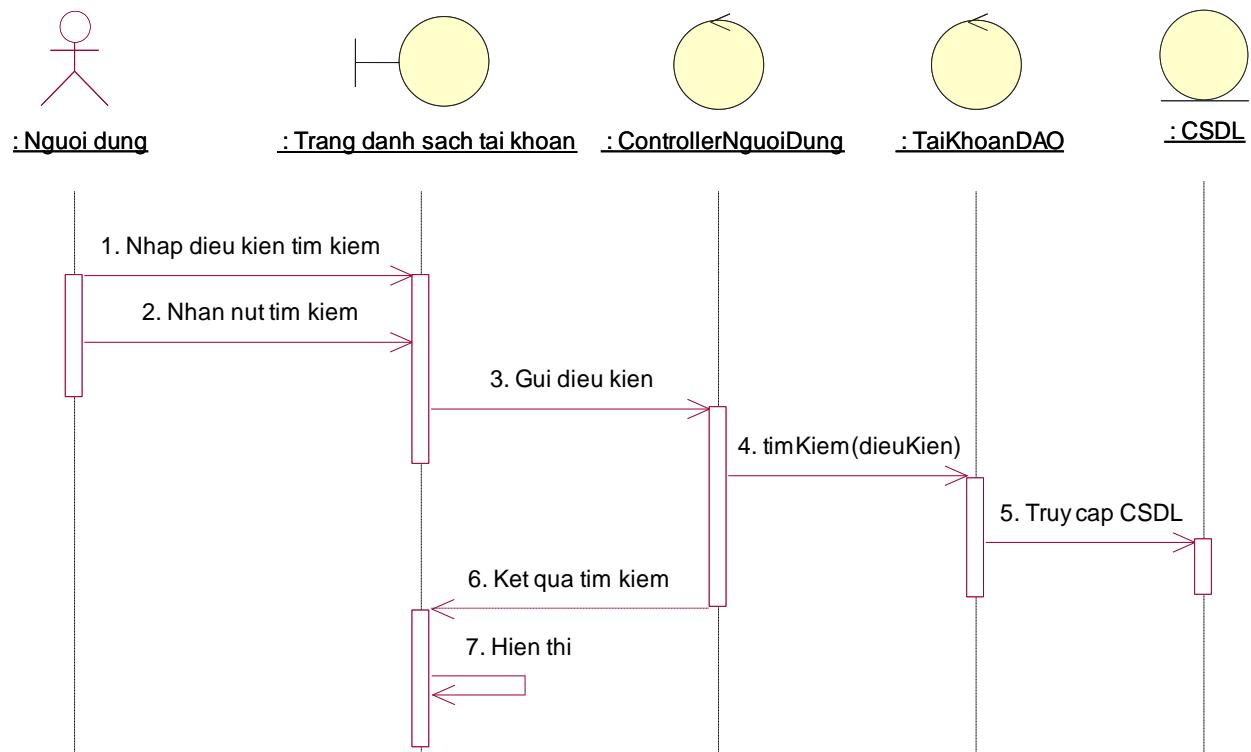
Website bán xe ô tô trực tuyến



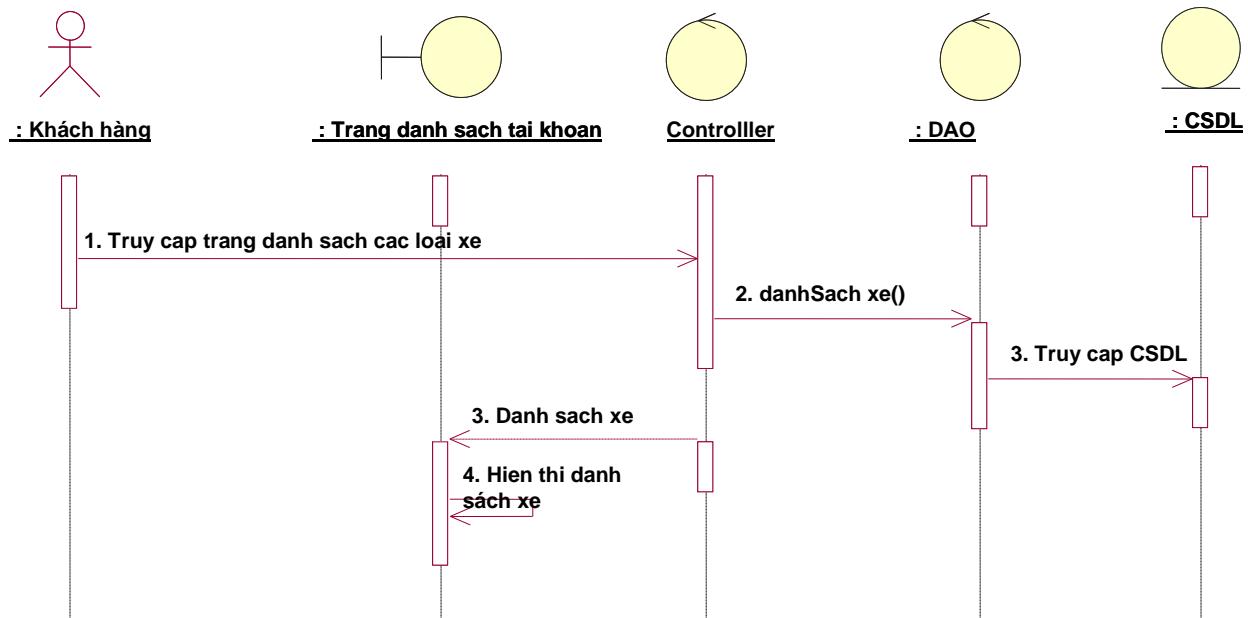
3.1.13. Chức năng xem thông tin cá nhân



1.1.14. Chức năng tìm kiếm xe

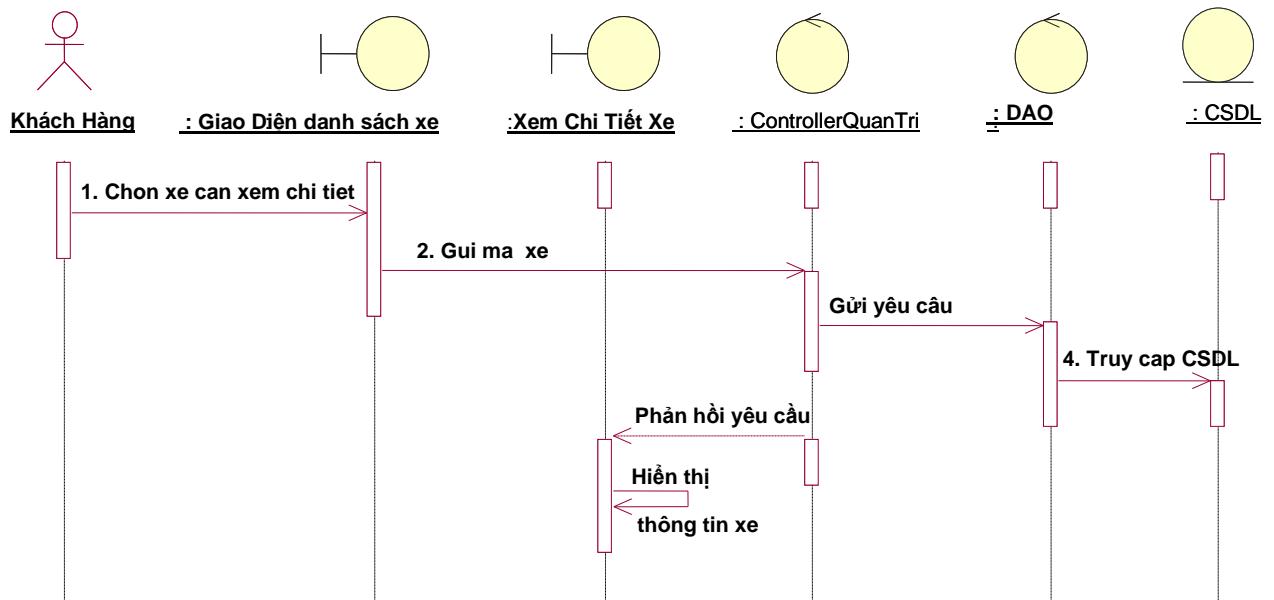


3.1.15. Chức năng xem danh sách các loại xe

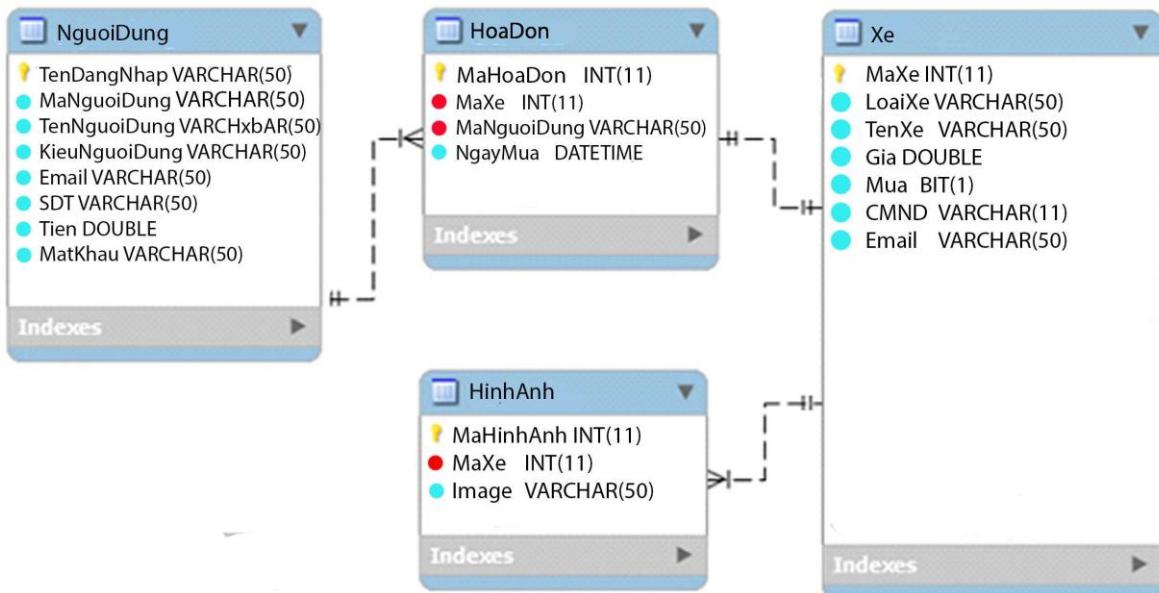


3.1.16. Chức năng xem thông tin chi tiết xe

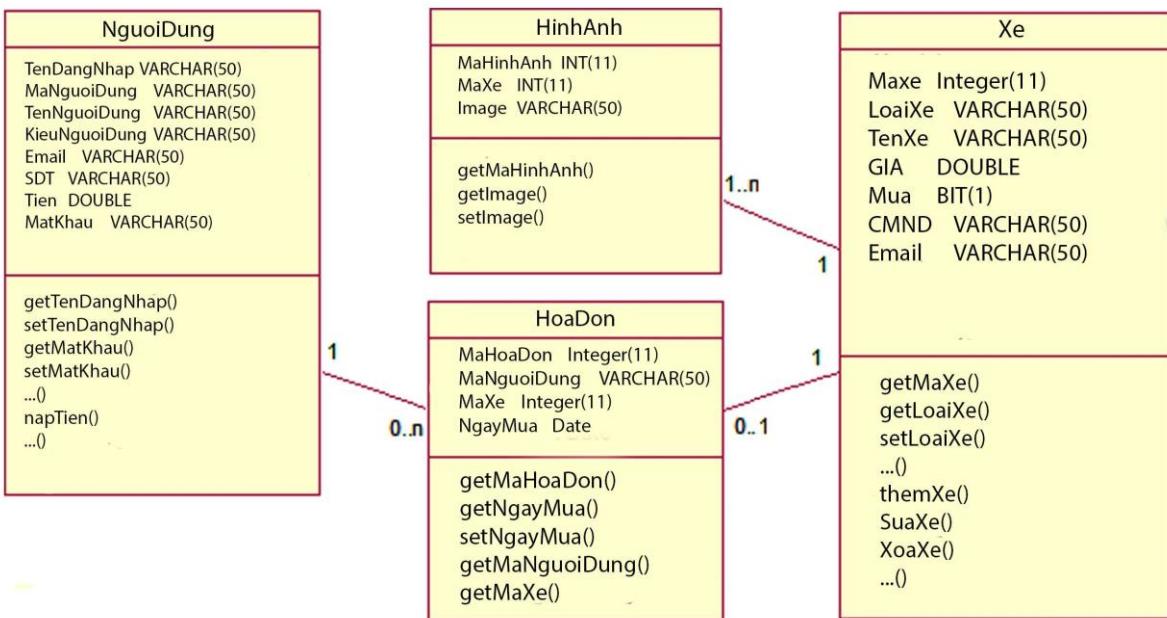
Website bán xe ô tô trực tuyến



4. Lược đồ cơ sở dữ liệu



5. Sơ đồ lớp



7. Đặc tả giao diện màn hình

7.1. Đặc tả màn hình trang chủ

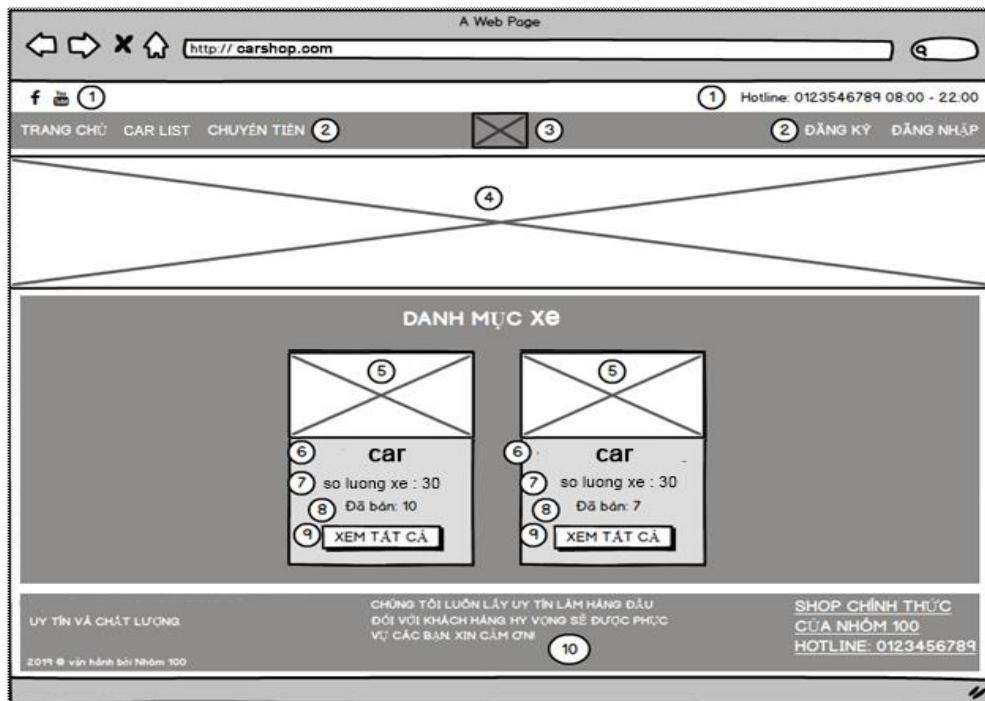
Mã màn hình	SCR-01	Tên chức năng	Trang chủ	Ngày tạo	24/10/2019
				Người tạo	Lê Phi Khanh

7.1.1. Tổng quan

Màn hình hiển thị trang chủ của website cho người dùng.

Màn hình này hiện lên khi người dùng nhấp vào nút “TRANG CHỦ” trên giao diện chính hoặc khi website được truy cập.

7.1.2. Giao diện màn hình



7.1.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Header	Icon, label	Hiển thị thông tin
2	Menu	Menu	Hiển thị menu
3	Logo	Image	Logo website
4	Quảng cáo	Image	Quảng cáo cho website
5	Hình ảnh xe	Image	Hiển thị hình ảnh của loại xe
6	Tên xe	Label	Hiển thị tên xe
7	Số lượng xe	Label	Hiển thị tổng số lượng của từng xe
8	Xe đã bán	Label	Hiển thị số xe đã bán
9	Xem tất cả	Button	Chuyển đến trang danh sách tài khoản
10	Footer	Label	Hiển thị thông tin website

7.1.4. Những thao tác chính với màn hình

❖ Đăng ký:

Người dùng nhấn nút “ĐĂNG KÝ” trên menu, sẽ chuyển đến trang đăng ký.

❖ Đăng nhập:

Người dùng nhấn nút “ĐĂNG NHẬP” trên menu, website sẽ chuyển đến trang đăng nhập.

❖ Xem danh sách các loại xe

Website bán xe ô tô trực tuyến

Người dùng nhấn vào nút “XEM TẤT CẢ” ở danh mục xe hoặc nhấn nút “LIST CAR” ở menu. Website sẽ chuyển đến trang danh sách các loại xe.

7.2. Đặc tả màn hình danh sách các loại xe

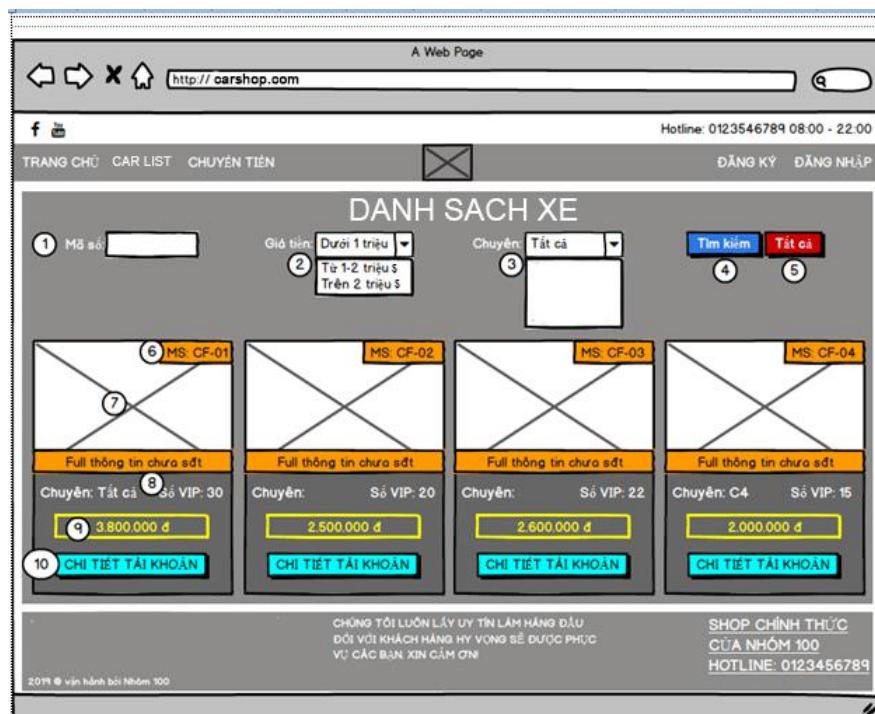
Mã màn hình	SCR-02	Tên chức năng	Danh sách xe	Ngày tạo	24/10/2019
				Người tạo	Võ Quang Bình

7.2.1. Tổng quan

Màn hình cho phép người dùng xem danh sách các loại xe và tìm kiếm chúng.

Màn hình này hiện lên khi người dùng nhấn vào nút “LIST CAR” trên menu hoặc nút “XEM TẤT CẢ” trong danh mục xe.

7.2.2. Giao diện màn hình



7.2.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Mã số xe	Textbox	Nhập mã số xe để tìm kiếm
2	Giá tiền	Combobox	Chọn giá tiền để tìm kiếm
3	Chuyên	Combobox	Chọn thể loại chuyên để tìm kiếm
4	Nút tìm kiếm	Button	Nhấn để tìm kiếm xe
5	Nút tất cả	Button	Nhấn để hiện thị lại tất cả các loại xe
6	Mã số xe	Label	Hiển thị mã số xe

7	Hình ảnh	Image	Hiển thị ảnh đại diện cho xe
8	Thông tin xe	Label	Hiển thị các thông tin của xe
9	Giá xe	Label	Hiển thị giá bán của xe
10	Nút chi tiết	Button	Nhấn để xem chi tiết xe

7.2.4. Những thao tác chính với màn hình

❖ Xem chi tiết xe

Khách hàng nhấn vào ảnh đại diện của xe hoặc nhấn nút “CHI TIẾT”.

Website sẽ chuyển sang trang chi tiết xe.

❖ Tìm kiếm xe

Người dùng nhập mã xe, chọn giá tiền, chọn thẻ loại mà xe chuyên và nhấn nút “Tìm kiếm”.

7.3. Đặc tả màn hình chi tiết xe

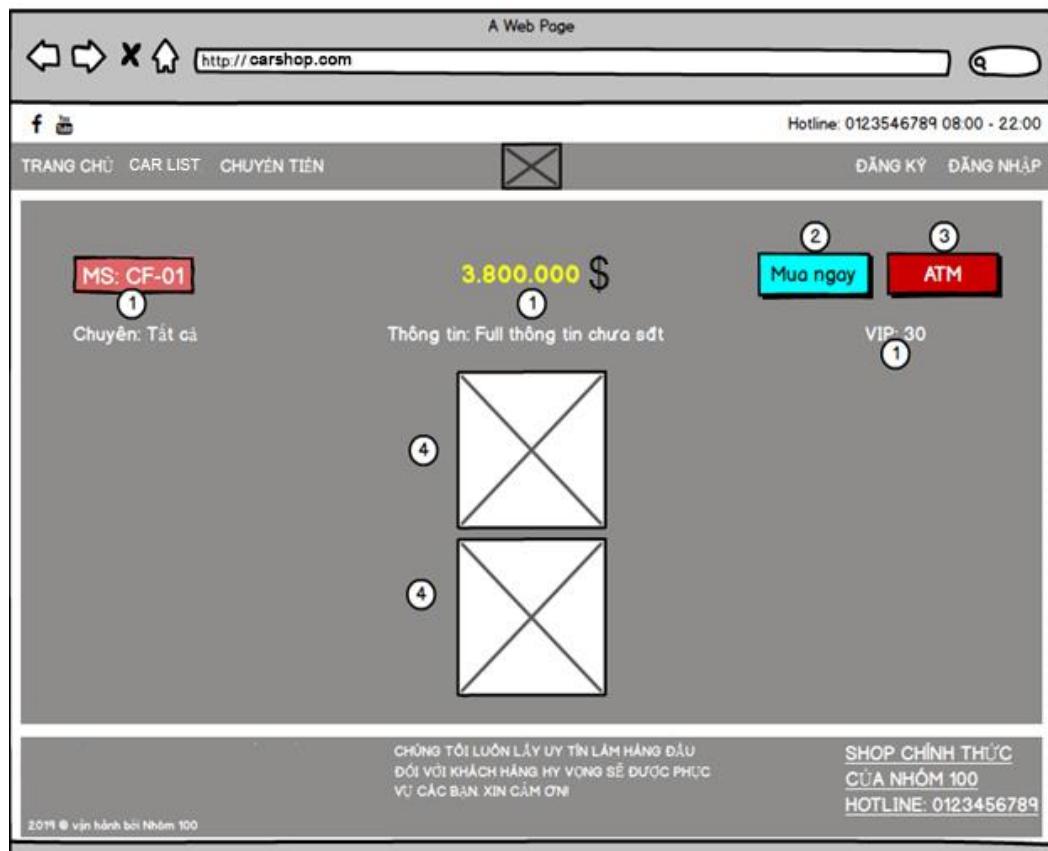
Mã màn hình	SCR-03	Tên chức năng	Chi tiết tài khoản	Ngày tạo	24/10/2019
				Người tạo	Trần Văn Hoài Nhiên

7.3.1. Tổng quan

Màn hình hiển thị các thông tin về xe như: mã xe, giá, số lượng vip, thông tin đăng ký, thẻ loại chuyên và các hình ảnh của xe.

Màn hình này hiện lên khi khách hàng nhấn vào hình hoặc nút “CHI TIẾT” trên trang danh sách các loại xe.

7.3.2. Giao diện màn hình



7.3.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Thông tin xe	Label	Hiển thị những thông tin của xe
2	Nút mua ngay	Button	Nhấn để mua xe
3	Nút ATM	Button	Nhấn để hiện thông tin chuyển khoản
4	Hình xe	Image	Những hình ảnh của xe

7.3.4. Những thao tác chính với màn hình

❖ Mua xe

Khách hàng nhấn vào nút “Mua xe”. Thông báo xác nhận mua xe sẽ hiện lên.

❖ Chuyển tiền qua ATM

Khách hàng nhấn vào nút “ATM”. Hộp thoại chứa thông tin để chuyển tiền sẽ hiện lên.

7.4. ĐẶC TẢ MÀN HÌNH CHUYỂN TIỀN

Mã màn hình	SCR-05	Tên chức năng	Chuyển tiền	Ngày tạo	24/10/2019
				Người tạo	Trần Văn Hoài Nhiên

7.4.1. Tổng quan

Màn hình hiển thị thông tin tài khoản của người dùng để chuyển khoản.

Màn hình này hiện lên khi người dùng nhấn vào nút “CHUYỂN TIỀN” trên menu hoặc nút “ATM” trên trang chi tiết.

7.4.2. Giao diện màn hình

Tên ngân hàng	Số tài khoản	Chi nhánh
Vietcombank	① 0123456789	Quy Nhơn
BIDV	0123456789	Quy Nhơn

2 Nội dung thanh toán: {ID tài khoản muốn mua và tên tài khoản đăng nhập vào shop}
Chuyển xong liên hệ fb: <https://www.facebook.com/abc> hoặc Hotline 0123456789 để được xử lý.

3 Đóng

7.4.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Thông tin ngân hàng	Table	Hiển thị thông tin các ngân hàng để chuyển khoản
2	Nội dung thanh toán	Label	Hiển thị nội dung thanh toán khi chuyển khoản
3	Nút đóng	Button	Nhấn để thoát hộp thoại

7.5. ĐẶC TẢ MÀN HÌNH XE ĐÃ MUA

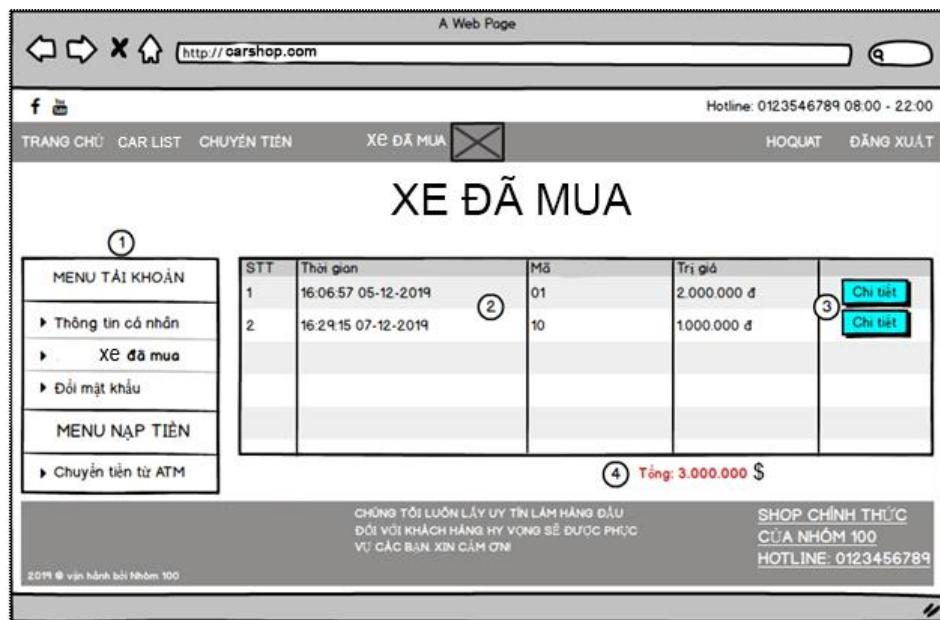
Mã màn hình	SCR-06	Tên chức năng	Tài khoản đã mua	Ngày tạo	24/10/2019
				Người tạo	Lê Phi Khanh

7.5.1. Tổng quan

Màn hình hiển thị danh sách các loại xe người dùng đã mua.

Màn hình này hiện lên khi khách hàng nhấn vào nút “XE ĐÃ MUA” trên menu.

7.5.2. Giao diện màn hình



7.5.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Menu	Menu	Hiển thị các chức năng
2	Thông tin xe	Table	Hiển thị thông tin các xe đã mua
3	Chi tiết xe	Button	Nhấn để hiển thị chi tiết xe đã mua
4	Tổng tiền mua	Label	Hiển thị tổng tiền khách hàng đã mua

7.5.4. Những thao tác chính với màn hình

❖ Xem chi tiết xe đã mua

Khách hàng nhấn vào nút “Chi tiết”. Một hộp thoại chưa thông tin chi tiết của xe đã mua sẽ hiện ra.

7.6. Đặc tả màn hình đổi mật khẩu

Mã màn hình	SCR-08	Tên chức năng	Đổi mật khẩu	Ngày tạo	24/10/2019
				Người tạo	Trần Văn Hoài Nhiên

7.6.1. Tổng quan

Màn hình cho phép người dùng đổi mật khẩu hiện tại.

Màn hình này hiện lên khi người dùng nhấn vào nút “Đổi mật khẩu” trên menu.

7.6.2. Giao diện màn hình

The screenshot shows a web browser window for 'carshop.com'. At the top, there are navigation icons, a search bar, and a URL bar. Below the header, there's a navigation bar with links like 'TRANG CHỦ', 'CAR LIST', 'CHUYỂN TIỀN', 'XE ĐÃ MUA', 'HOQUAT', and 'ĐĂNG XUẤT'. The main content area has a title 'ĐỔI MẬT KHẨU' (Change Password). On the left, there's a sidebar with two sections: 'MENU TÀI KHOẢN' containing 'Thông tin cá nhân', 'Xe đã mua', and 'Đổi mật khẩu'; and 'MENU NẠP TIỀN' containing 'Chuyển tiền từ ATM'. The main form on the right has three input fields labeled ①: 'Mật khẩu cũ', 'Mật khẩu mới', and 'Nhập lại mật khẩu mới'. Below these is a blue button labeled ② 'Đổi mật khẩu'. At the bottom of the page, there's a footer with text about service quality, a shop address 'SHOP CHÍNH THỨC CỦA NHÓM 100', a hotline 'HOTLINE: 0123456789', and a copyright notice '2019 © vận hành bởi Nhóm 100'.

7.6.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Nhập mật khẩu	Textbox	Nhập mật khẩu để đổi mật khẩu
2	Nút đổi mật khẩu	Button	Nhấn để tiến hành đổi mật khẩu

7.6.4. Những thao tác chính với màn hình

❖ Đổi mật khẩu

Người dùng nhấn vào nút “Đổi mật khẩu”. Thông báo đổi mật khẩu thành công.

7.7. Đặc tả màn hình đăng nhập

Mã màn hình	SCR-09	Tên chức năng	Đăng nhập	Ngày tạo	24/10/2019
				Người tạo	Võ Quang Bình

7.7.1. Tổng quan

Màn hình cho phép người dùng đăng nhập vào hệ thống.

Màn hình này hiện lên khi người dùng nhấn vào nút “Đăng nhập” trên menu.

7.7.2. Giao diện màn hình



7.7.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Nhập tên đăng nhập và mật khẩu	Textbox	Nhập tên đăng nhập và mật khẩu để đăng nhập
2	Nút đăng nhập	Button	Nhấn để đăng nhập vào hệ thống
3	Nút tạo tài khoản	Button	Nhấn để đăng ký mới tài khoản

7.7.4. Những thao tác chính với màn hình

❖ Đăng nhập

Người dùng nhấn vào nút “Đăng nhập, đăng nhập thành công, website chuyển đến trang chủ với tên người dùng hiển thị ở trên menu.

❖ Tạo tài khoản

Người dùng nhấn vào nút “Tạo tài khoản”, website sẽ chuyển đến trang tạo tài khoản.

7.8. Đặc tả màn hình đăng ký

Mã màn hình	SCR-10	Tên chức năng	Đăng ký	Ngày tạo	24/10/2019
				Người tạo	Võ Quang Bình

7.8.1. Tổng quan

Website bán xe ô tô trực tuyến

Màn hình cho phép khách hàng đăng ký mới một tài khoản trong hệ thống.

Màn hình này hiện lên khi khách hàng nhấp vào nút “Đăng ký” trên menu.

7.8.2. Giao diện màn hình

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://ShopGame". The top navigation bar includes social media icons (Facebook, YouTube), a search bar, and links for "Hotline: 0123546789 08:00 - 22:00", "TRANG CHỦ", "ĐỘT KÍCH", "CHUYỂN TIỀN", "ĐĂNG KÝ", and "ĐĂNG XUẤT". The main content area is titled "Đăng ký tài khoản". It features six input fields labeled ①: "Nhập tên của bạn" (User name), "Nhập tên đăng nhập" (Login name), "Nhập mật khẩu" (Password), "Nhập lại mật khẩu" (Re-enter password), "Nhập email" (Email), and "Nhập số điện thoại" (Phone number). Below these fields are three buttons labeled ②: a red "Đăng ký" button, a blue "Đăng nhập" button, and a separator "- hoặc -". At the bottom left, there is a footer note: "HỆ THỐNG BÁN ACC GAME CAO CẤP - ĐÀM BẢO UY TÍN VÀ CHẤT LƯỢNG" and "2019 © vận hành bởi Nhóm 100". The bottom right contains a footer note: "CHUNG TỎI LUÔN LẤY UY TÍN LÀM HÀNG ĐẦU ĐỐI VỚI KHÁCH HÀNG, HY VỌNG SẼ ĐƯỢC PHỤC VỤ CÁC BẠN XIN CẨM ƠN!" and "SHOP CHÍNH THỨC CỦA NHÓM 100 HOTLINE: 0123456789".

7.8.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Nhập các thông tin đăng ký	Textbox	Khách hàng nhập các thông tin để đăng ký
2	Nút đăng ký	Button	Nhấn để đăng ký một tài khoản
3	Nút đăng nhập	Button	Nhấn để chuyển sang trang đăng nhập

7.8.4. Những thao tác chính với màn hình

❖ Đăng nhập

Khách hàng nhấp vào nút “Đăng nhập”, website chuyển đến trang đăng nhập.

❖ Tạo tài khoản

Khách hàng nhấp vào nút “Đăng ký”, hệ thống thông báo đăng ký thành công.

7.9. Đặc tả màn hình danh sách tài khoản cho quản trị viên

Website bán xe ô tô trực tuyến

Mã màn hình	SCR-11	Tên chức năng	Danh sách tài khoản	Ngày tạo	24/10/2019
				Người tạo	Lê Phi Khanh

7.9.1. Tổng quan

Màn hình cho phép hiển thị danh sách các loại xe cho quản trị viên

Màn hình này hiện lên khi quản trị viên nhấn vào nút “CAR LIST” trên menu.

7.9.2. Giao diện màn hình

The screenshot shows a user interface titled "DANH SÁCH XE". At the top, there are search filters: "Mã số" (License Plate), "Giá tiền" (Price) set to "Dưới 1 triệu" (Under 1 million), "Chuyên" (Category) set to "Tất cả" (All), and buttons for "Tim kiem" (Search) and "Tất cả" (All). Below the filters, four car entries are listed in a grid:

MS: CF-01	MS: CF-02	MS: CF-03	MS: CF-04
Full thông tin chưa sđt	Full thông tin chưa sđt	Full thông tin chưa sđt	Full thông tin chưa sđt
Chuyên: Tất cả Số VIP: 30	Chuyên: Số VIP: 20	Chuyên: Số VIP: 22	Chuyên: C4 Số VIP: 15
3.800.000 đ	2.500.000 đ	2.600.000 đ	2.000.000 đ
XÓA SỬA	XÓA SỬA	XÓA SỬA	XÓA SỬA

7.9.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Nút xóa	Button	Nhấn để xóa xe khỏi hệ thống
2	Nút sửa	Button	Nhấn để sửa các thông tin của xe

7.9.4. Những thao tác chính với màn hình

❖ Xóa tài khoản

Quản trị viên nhấn vào nút “Xóa”, website sẽ xuất hiện thông báo chấp nhận xóa. Tài khoản sẽ bị xóa khỏi hệ thống khi quản trị viên nhấn “OK”.

❖ Sửa tài khoản

Quản trị viên nhấn vào nút “Sửa”, website sẽ chuyển đến trang sửa tài khoản.

7.9. ĐẶC TẢ MÀN HÌNH THÊM XE

Mã màn hình	SCR-13	Tên chức năng	Thêm tài khoản	Ngày tạo	24/10/2019
				Người tạo	Lê Phi Khanh

7.10.1. Tổng quan

Màn hình cho phép quản trị viên thêm mới một chiếc xe.

Màn hình này hiện lên khi quản trị viên nhấp vào nút “Thêm xe” trên menu.

7.10.2. Giao diện màn hình

THEM XE

Loại	<input type="text" value="Tất cả"/>
Giá:	<input type="text" value="Nhập giá tài khoản"/>
Tên đăng nhập:	<input type="text" value="Nhập tên đăng nhập"/>
Mật khẩu tài khoản:	<input type="text" value="Nhập mật khẩu tài khoản"/>
Email	<input type="text" value="Nhập email tài khoản"/>
CMND	<input type="text" value="Nhập CMND tài khoản"/>
Tài ảnh lên:	<input type="text" value="Nhấn để tải hình ảnh"/>
THÊM	

(1) (2)

7.10.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Thông tin xe	Label, combobox	Nhập, chọn thông tin cho xe
2	Nút thêm	Button	Nhấn để thêm mới

7.10.4. Những thao tác chính với màn hình

- ❖ Thêm tài khoản

Quản trị viên nhấp vào nút “Thêm”, website sẽ thông báo thêm xe thành công.

7.11. ĐẶC TẢ MÀN HÌNH THÔNG TIN CÁ NHÂN

<i>Mã màn hình</i>	<i>SCR-14</i>	<i>Tên chức năng</i>	<i>Thông tin cá nhân</i>	<i>Ngày tạo</i>	<i>24/10/2019</i>
				<i>Người tạo</i>	<i>Lê Phi Khanh</i>

7.11.1. Tổng quan

Màn hình hiển thị thông tin cá nhân của người dùng.

Màn hình này hiện lên khi người dùng nhấn vào nút “Thông tin cá nhân” trên menu.

7.11.2. Giao diện màn hình

THÔNG TIN CÁ NHÂN

Mục	1	Thông tin
Tên đăng nhập:		lephikhanh
Tên của bạn:		Lê Phi Khanh
Số dư của bạn:		2.000.000 đ
Địa chỉ Email:		lephikhanh@gmail.com
Số điện thoại:		0330300303
Loại tài khoản:		NGUOI_DUNG

7.11.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Thông tin cá nhân	Table	Hiển thị các thông tin của người dùng

7.12. Đặc tả màn hình danh sách mua xe

Mã màn hình	SCR-15	Tên chức năng	Danh sách mua tài khoản	Ngày tạo	24/10/2019
				Người tạo	Trần Văn Hoài Nhiên

7.12.1. Tổng quan

Màn hình cho phép quản trị viên xem những khách hàng đã mua xe.

Màn hình này hiện lên khi quản trị viên nhấn vào nút “Danh sách mua” trên menu.

7.12.2. Giao diện màn hình

DANH SÁCH ĐÃ MUA XE

STT	Thời gian	Tên người mua	Mã tài khoản	Trị giá
1	16:06:57 05-12-2019	Khánh	01	2.000.000 đ
2	16:29:15 07-12-2019	Nhiên	10	1.000.000 đ

② Tổng: 3.000.000 đ

7.12.3. Các thành phần của màn hình

STT	Tên	Kiểu	Mô tả
1	Danh sách khách hàng mua xe	Table	Hiển thị danh sách khách hàng đã mua xe
2	Tổng tiền bán	Label	Hiển thị tổng tiền đã bán được

Phần II : Các Framework sử dụng và Triển Khai

I. FRAMEWORK SỬ DỤNG TRONG PROJECT

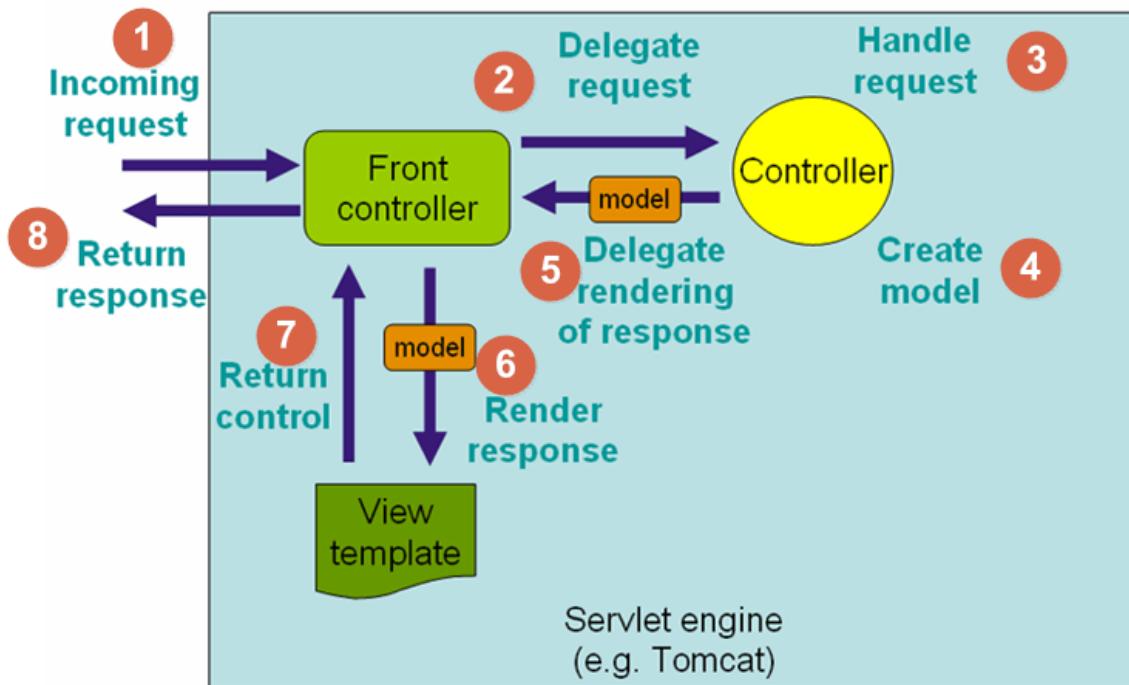
1. Spring MVC

1.1. Spring MVC (Spring Web MVC)

Spring MVC là một Framework / 1 Project mã nguồn mở của Spring.

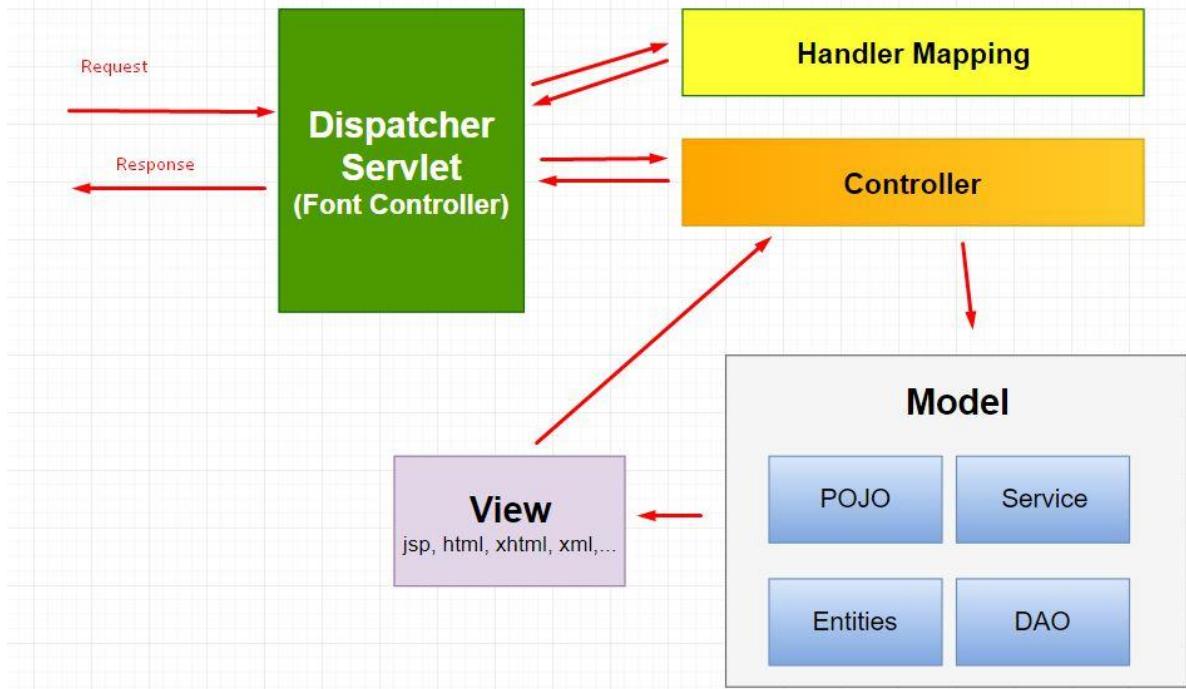
Spring MVC Framework cung cấp kiến trúc MVC (Model-View-Controller) và các component được sử dụng để phát triển các ứng dụng web một cách linh hoạt và giảm sự

1.2. Flow trong Spring MVC.



- Bất kỳ request nào tới ứng dụng web đều sẽ được gửi tới Front Controller (Dispatcher Servlet)
- Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó
- Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
- Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
- Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet)

- Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
- View template, model, view page được build và gửi trả lại Front Controller
- Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.



Trong Mô hình MVC thì:

- **Model:** là các file POJO, Service, DAO thực hiện truy cập database, xử lý business
- **View:** là các file JSP, html...
- **Control:** là Dispatcher Controller, Handler Mapping, Controller – thực hiện điều hướng các request.

1.3. Lợi ích của Spring MVC

Các tầng trong Spring MVC độc lập nên việc unit test dễ dàng hơn.

Phần view có thể tích hợp với nhiều framework về UI như JSF, Freemarker, Thymeleaf...

Spring MVC base trên các POJO class nên các hành động của nó khá đơn giản.

Hỗ trợ cả Annotation và XML config giúp việc phát triển nhanh hơn và sạch hơn.

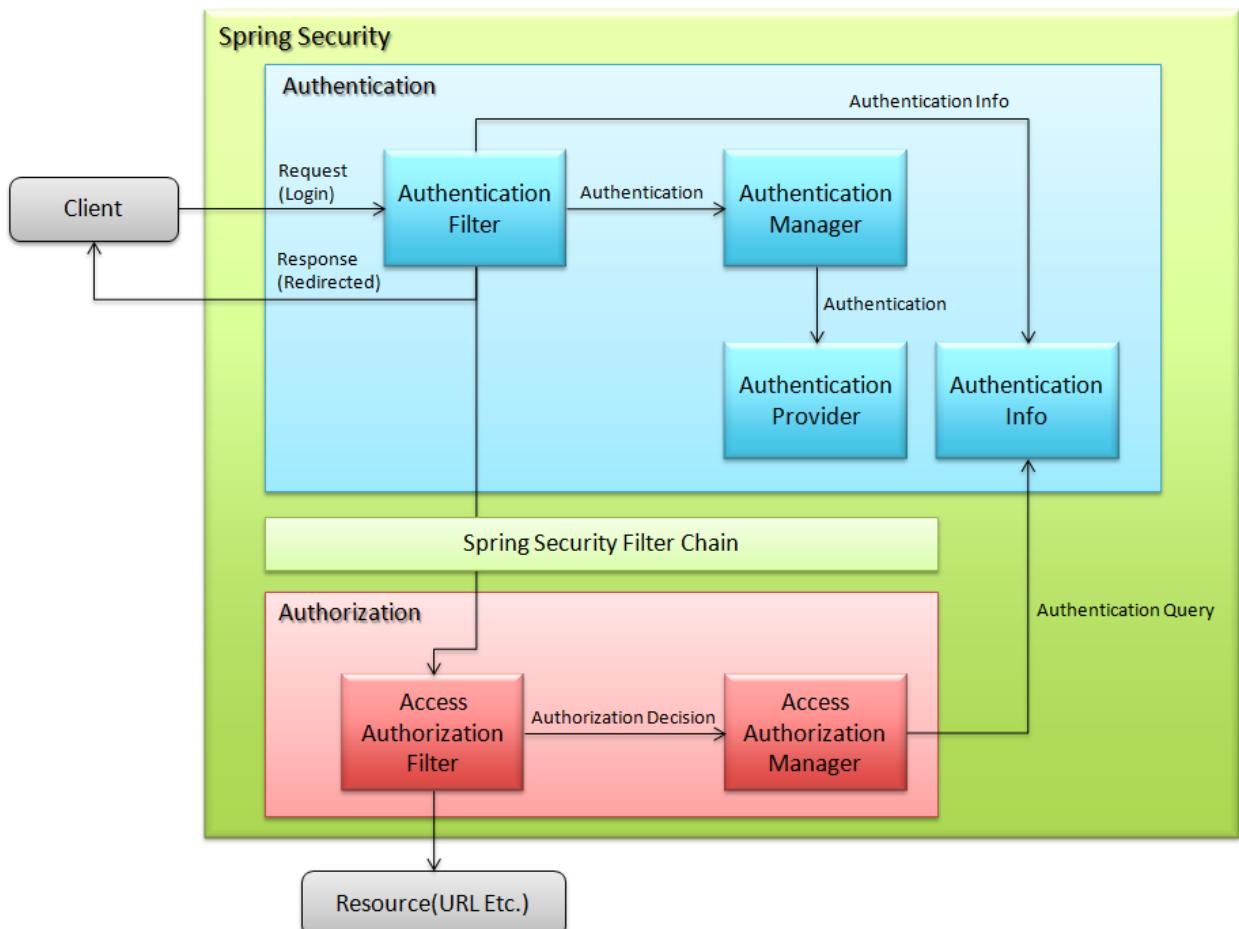
Cung cấp việc phân chia một cách rõ ràng, linh hoạt giữa controller, service, DAO.

2. Spring Security

2.1. Giới thiệu

Spring security: nó là một framework hay đúng hơn là một công cụ cung cấp và xử lý các vấn đề về xác thực và phân quyền cho các ứng dụng web nó là một công cụ luôn song hành và luôn được ưu tiên sử dụng trong các dự án về java trong đó nổi bật nhất là spring boot.

Cơ chế :



Spring security cung cấp cho ta 2 cơ chế cơ bản

- **Authentication:** cơ chế xác thực xem bạn có được quyền vào ứng dụng của bạn không túm váy như kiểu đăng nhập bằng username và password đúng hay sai vậy đây là ải đầu tiên của spring security.
- **Authorization:** cung cấp cho chúng ta cơ chế phân quyền xem user này có được thao tác hành động nào đó không, đơn giản nhất là admin có quyền thêm xóa thành viên có quyền xem bài vậy tất nhiên tới ải này được thì chúng ta phải vượt qua được ải thứ nhất.

2.2. Các thành phần

2.2.1. SecurityContext

SecurityContext là interface cốt lõi của Spring Security, lưu trữ tất cả các chi tiết liên quan đến bảo mật trong ứng dụng. Khi chúng ta kích hoạt Spring Security trong ứng dụng thì SecurityContext cũng sẽ được kích hoạt theo.

2.2.2. SecurityContextHolder

Chúng ta sẽ không truy cập trực tiếp vào SecurityContext, thay vào đó sẽ sử dụng lớp **SecurityContextHolder**. Lớp này lưu trữ security context hiện tại của ứng dụng, bao gồm chi tiết của principal đang tương tác với ứng dụng. Spring Security sẽ dùng một đối tượng Authentication để biểu diễn thông tin này.

2.2.3. UserDetails

UserDetails là một interface cốt lõi của Spring Security. Nó đại diện cho một principal nhưng theo một cách mở rộng và cụ thể hơn. UserDetails bao gồm các method sau:

- ❖ getAuthorities(): trả về danh sách các quyền của người dùng
- ❖ getPassword(): trả về password đã dùng trong quá trình xác thực
- ❖ getUsername(): trả về username đã dùng trong quá trình xác thực
- ❖ isAccountNonExpired(): trả về true nếu tài khoản của người dùng chưa hết hạn
- ❖ isAccountNonLocked(): trả về true nếu người dùng chưa bị khóa
- ❖ isCredentialsNonExpired(): trả về true nếu chứng thực (mật khẩu) của người dùng chưa hết hạn
- ❖ isEnabled(): trả về true nếu người dùng đã được kích hoạt

Đoạn code sau là kết hợp giữa **SecurityContextHolder** và method **getUsername()** trong **UserDetails**, để lấy tên người dùng bằng chính username đó (username ở đây ta nên hiểu là username trong cặp username - password mà người dùng nhập vào khi đăng nhập):

```
83@ @RequestMapping(value = { "/thongtin" }, method = RequestMethod.GET)
84 public String thongTin(Model model) {
85
86     UserDetails nguoiDung = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
87
88     NguoiDung thongtin =  nguoiDungService.timNguoiDung(nguoiDung.getUsername());
89
90     model.addAttribute("thongtin", thongtin);
91
92     return "thongtin";
93 }
```

2.2.4. UserDetailsService

UserDetailsService là một interface có duy nhất một phương thức. Tham số truyền vào chỉ gồm có username của người dùng. Ta sẽ tìm kiếm trong CSDL, phải tồn tại username. Nếu không tìm thấy, ta sẽ ném ra ngoại lệ UsernameNotFoundException.

Phương thức đó là **loadUserByUsername()**, có thể trả về một implementation của **UserDetails**. Có thể là **org.springframework.security.core.userdetails.User**.

Và ta cần phải tạo một lớp để implements UserDetailsService.

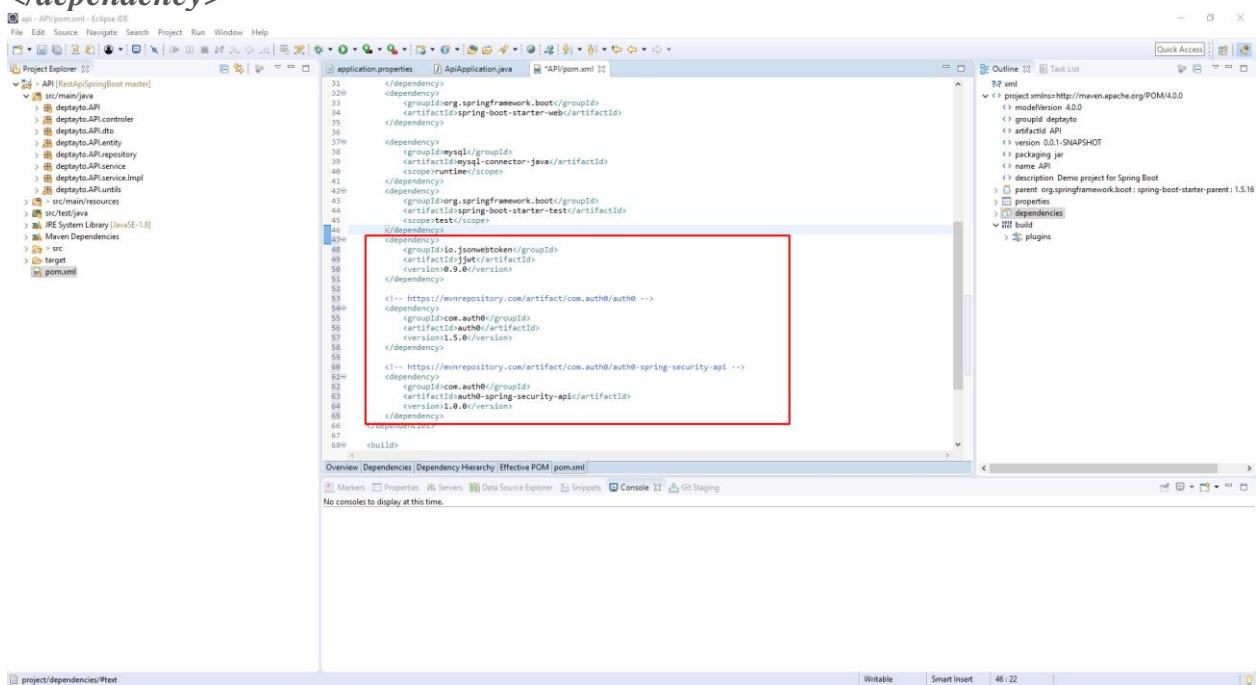
2.2.5. GrantedAuthority

Ở phần trên, đã đề cập đến phương thức getAuthorities(). Phương thức này sẽ trả về một tập hợp các đối tượng **GrantedAuthority**. Một GrantedAuthority là một quyền được cấp cho principal. Các quyền đều có tiền tố là ROLE_, ví dụ như ROLE_ADMIN, ROLE_MEMBER...

2.3. Về Spring Security

Đầu tiên mình sẽ thêm những dependency cần thiết dưới đây vào file pom.xml

```
<dependency>
<groupId>io.jsonwebtoken</groupId>
<artifactId>jjwt</artifactId>
<version>0.9.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.auth0/auth0 -->
<dependency>
<groupId>com.auth0</groupId>
<artifactId>auth0</artifactId>
<version>1.5.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.auth0/auth0-spring-security-api -->
<dependency>
<groupId>com.auth0</groupId>
<artifactId>auth0-spring-security-api</artifactId>
<version>1.0.0</version>
</dependency>
```



Website bán xe ô tô trực tuyến

- Đầu tiên để bắt add authentication vào source ta cần tạo một đối tượng user để chứa thông tin của các user login ở đây mình tạo 1 cái đơn giản chỉ gồm username và password

```
springsecurity - API/src/main/java/deptayo/API/entity/Users.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer [2] API (RestApiSpringBoot master)
src/main/java
deptayo.API
deptayo.API.controller
deptayo.API.dto
deptayo.API.entity
deptayo.API.repository
deptayo.API.service
deptayo.API.service.impl
deptayo.API.utils
src/main/resources
src/test/java
JRE System Library [javaSE-1.8]
Maven Dependencies
src
target
pom.xml

Users.java
1 package deptayo.API.entity;
2 import javax.persistence.GeneratedValue;
3 import javax.persistence.GenerationType;
4 import javax.persistence.Id;
5 import javax.persistence.Entity;
6 import javax.persistence.Table;
7 import java.util.Date;
8
9 @Entity
10 @Table(name = "users")
11 public class Users {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private long id;
16
17     @Column(name = "username", unique = true)
18     private String userName;
19
20     @Column(name = "password")
21     private String password;
22
23     public long getId() {
24         return id;
25     }
26
27     public void setId(long id) {
28         this.id = id;
29     }
30
31     public String getUserName() {
32         return userName;
33     }
34
35     public void setUserName(String userName) {
36         this.userName = userName;
37     }
38
39     public String getPassword() {
40         return password;
41     }
42
43     public void setPassword(String password) {
44         this.password = password;
45     }
46
47 }
48
```

Markers Properties Servers Data Source Explorer Snippets

0 errors, 16 warnings, 0 others

Description	Resource	Path	Location	Type
Java Problems (16 items)				

- Tiếp tới ta tạo 1 class UsersDto.java

```
springsecurity - API/src/main/java/deptayo/API/dto/UsersDto.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer [2] API (RestApiSpringBoot master)
src/main/java
deptayo.API
deptayo.API.controller
deptayo.API.dto
deptayo.API.entity
deptayo.API.repository
deptayo.API.service
deptayo.API.service.impl
deptayo.API.utils
src/main/resources
src/test/java
JRE System Library [javaSE-1.8]
Maven Dependencies
src
target
pom.xml

UsersDto.java
1 package deptayo.API.dto;
2 public class UserDto {
3
4     private long id;
5
6     private String userName;
7
8     private String password;
9
10    public long getId() {
11        return id;
12    }
13
14    public void setId(long id) {
15        this.id = id;
16    }
17
18    public String getUserName() {
19        return userName;
20    }
21
22    public void setUserName(String userName) {
23        this.userName = userName;
24    }
25
26    public String getPassword() {
27        return password;
28    }
29
30    public void setPassword(String password) {
31        this.password = password;
32    }
33
34 }
```

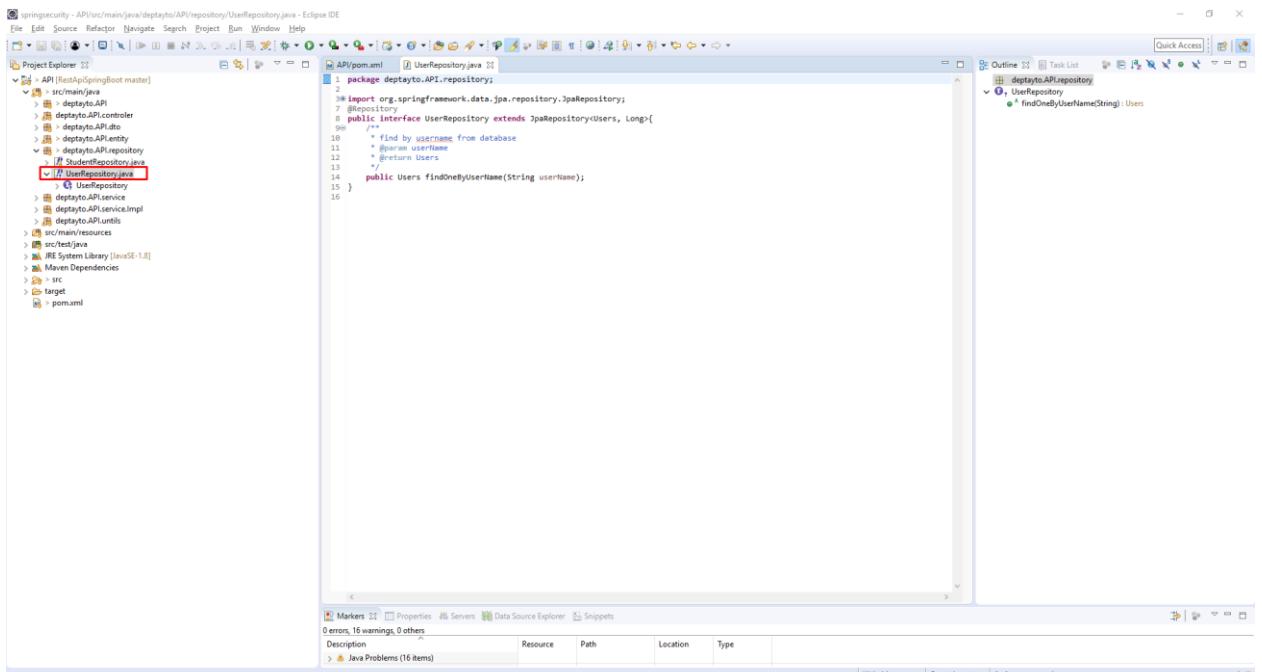
Markers Properties Servers Data Source Explorer Snippets

0 errors, 16 warnings, 0 others

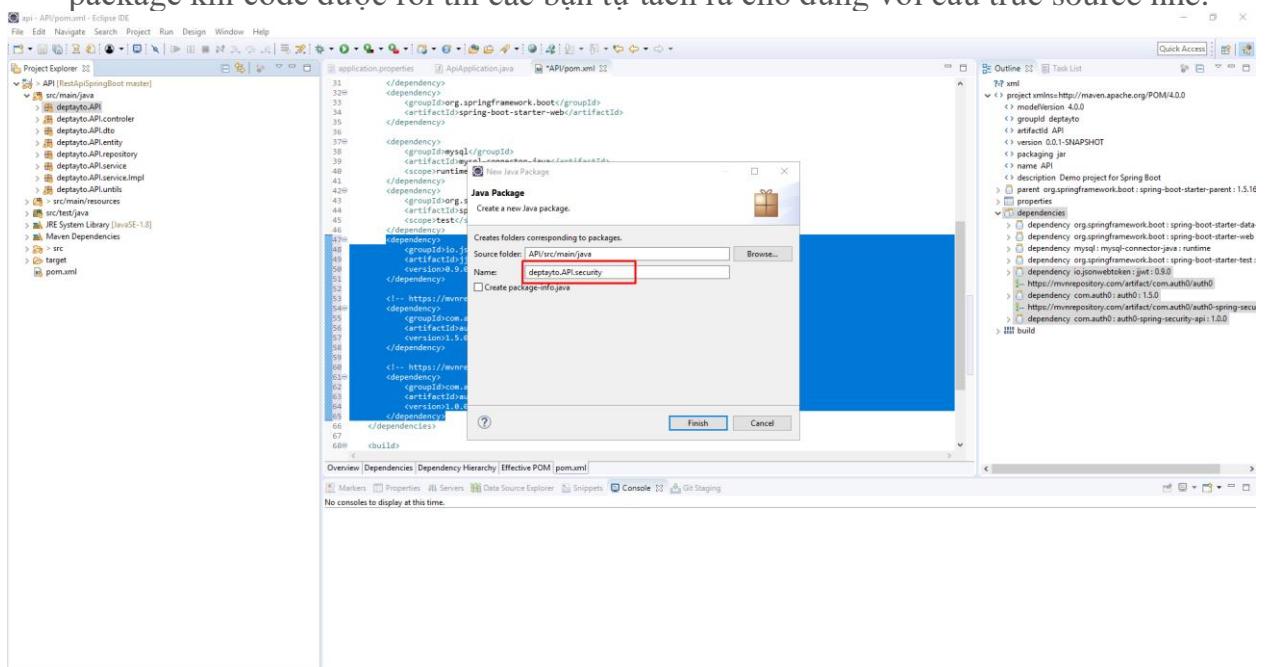
Description	Resource	Path	Location	Type
Java Problems (16 items)				

- Tiếp tới mình tạo thêm 1 class UserRepository.java để lấy thông tin User từ database phục vụ cho việc authentication.

Website bán xe ô tô trực tuyến



- Tiếp theo chúng ta tạo thêm 1 package security như hình vì để cho các bạn dễ theo dõi cũng như dễ hiểu mình sẽ để tất cả các class liên quan tới authentication trong package khi code được rồi thì các bạn tự tách ra cho đúng với cấu trúc source nhé.



- Đầu tiên ta sẽ tạo 1 file TokenAuthenticationService.java để xử lý nhiệm vụ generate ra jwt (json web token)

Website bán xe ô tô trực tuyến

```

package deptato.API.security;
import java.util.Collections;
public class TokenAuthenticationService {
    static final long EXPIRATIONTIME = 1000*60*2; // expire later 2 minute
    static final String SECRET = "thisIsSecret"; // key using decode token
    static final String TOKEN_PREFIX = "Bearer"; // prefix token
    static final String HEADER_STRING = "Authorization"; // key input from header request
    public static void addAuthentication(HttpServletRequest res, String username) {
        String jwt = Jwts.builder()
            .setSubject(username)
            .setIssuedAt(new Date(System.currentTimeMillis()))
            .signWith(SignatureAlgorithm.RS256, SECRET).compact();
        res.addHeader(HEADER_STRING, TOKEN_PREFIX + " " + jwt);
    }
    public static Authentication getAuthentication(HttpServletRequest request) {
        String header = request.getHeader(HEADER_STRING);
        if (header != null) {
            String token = header.substring(TOKEN_PREFIX.length());
            String user = Jwts.parser().setSigningKey(SECRET).parseClaimsJws(token.replace(TOKEN_PREFIX, "")).getBody()
                .getSubject();
            return user != null ? new UsernamePasswordAuthenticationToken(user, null, Collections.emptyList()) : null;
        }
        return null;
    }
}

```

Markers: 0 errors, 16 warnings, 0 others
Description Resource Path Location Type
Java Problems (16 items)

- Ở đây vì mình sẽ lấy thông tin user từ database để authentication nên mình phải có 1 service để tìm user được nhập vào khi authentication xem nó có trong database không vì thế ở đây mình sẽ tạo tiếp 1 class UserService.java và implements lại UserDetailsService.java mà spring cung cấp để có thể so sánh được thông tin của user trong database và thông tin user được nhập vào

```

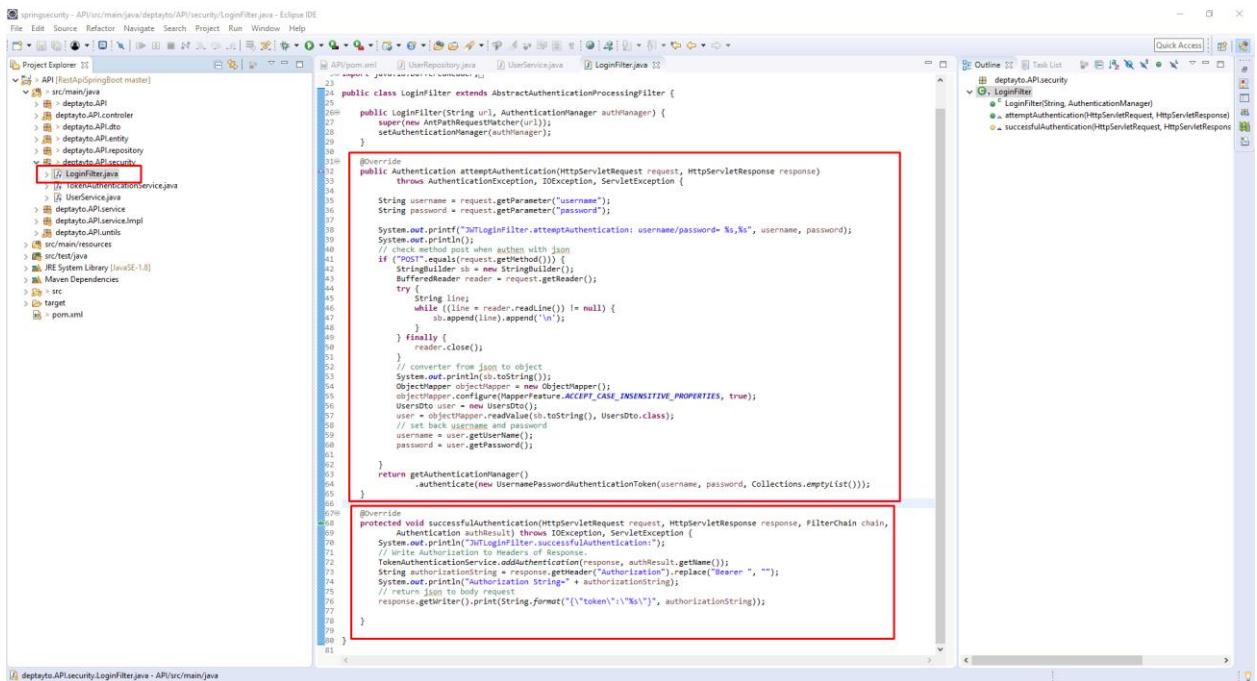
package deptato.API.security;
import java.util.Collections;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.userdetails.User;
import org.springframework.core.userdetails.UserDetails;
import org.springframework.stereotype.Service;
@Service
public class UserService implements UserDetailsService {
    @Autowired
    private UserRepository userRepository;
    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Users user = userRepository.findOneByUsername(username);
        if (user == null) {
            throw new UsernameNotFoundException("User " + username + " not found");
        }
        return org.springframework.security.core.userdetails.User.withUsername(username).password(user.getPassword())
            .authorities(Collections.emptyList()).accountExpired(false).accountLocked(false)
            .credentialsExpired(false).disabled(false).build();
    }
}

```

Markers: 0 errors, 16 warnings, 0 others
Description Resource Path Location Type
Java Problems (16 items)

- Khi authentication chúng ta cần nhập vào username và password để authentication chính vì vậy chúng ta cần tạo 1 class để hứng thông tin nhập vào và tiến hành trả lại token ra cho người dùng nếu thành công vì thế mình sẽ tạo thêm 1 class LoginFilter.java kế thừa lại AbstractAuthenticationProcessingFilter.java của spring nhằm mục đích custom lại 2 hàm attemptAuthentication có nhiệm vụ **authen** và successfulAuthentication có nhiệm vụ xử lý trả ra token cho người dùng nếu authen thành công.

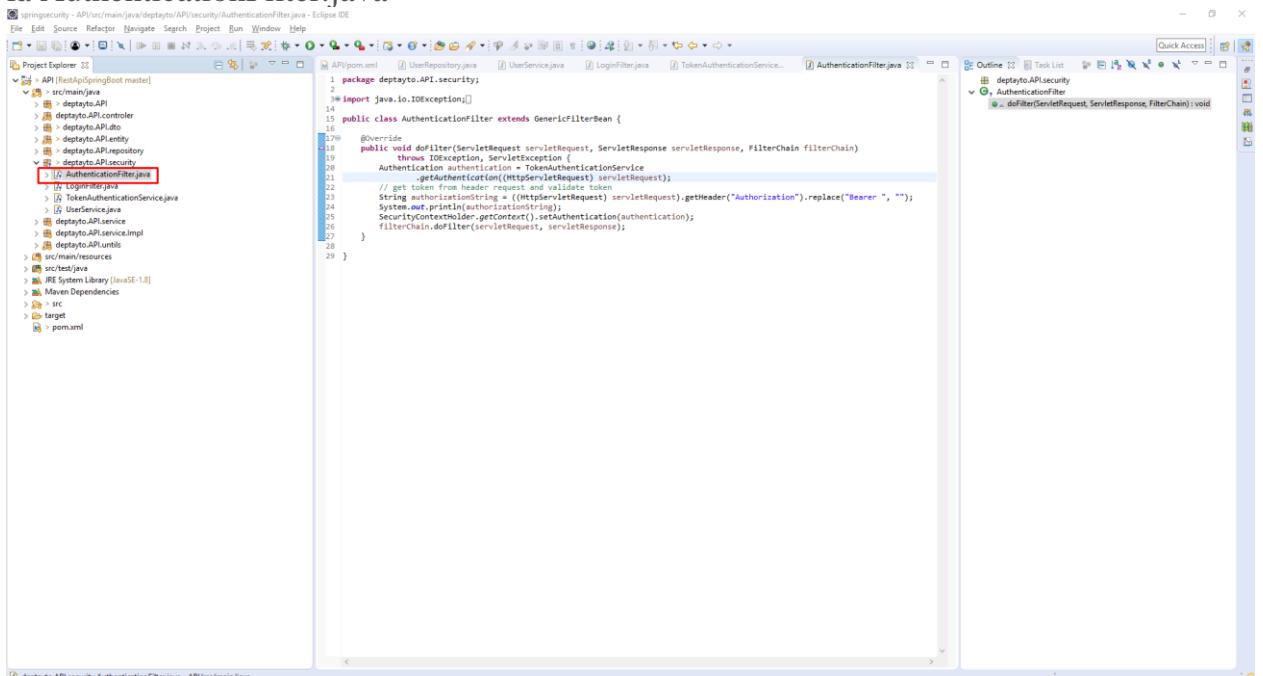
Website bán xe ô tô trực tuyến



```
public class LoginFilter extends AbstractAuthenticationProcessingFilter {
    public LoginFilter(String url, AuthenticationManager authManager) {
        super(new AntPathRequestMatcher(url));
        setAuthenticationManager(authManager);
    }
    @Override
    public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response)
            throws AuthenticationException, IOException, ServletException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        System.out.printf("LoginFilter.attemptAuthentication: username/password= %s,%s", username, password);
        // check method post when putten with json
        if ("POST".equals(request.getMethod())) {
            StringReader reader = new StringReader(request.getReader());
            try {
                String line;
                while ((line = reader.readLine()) != null) {
                    sb.append(line).append("\n");
                }
            } finally {
                reader.close();
            }
        } // converter from json to object
        ObjectMapper mapper = new ObjectMapper();
        mapper.configure(MapperFeature.ACCEPT_CASE_INSENSITIVE_PROPERTIES, true);
        User user = mapper.readValue(sb.toString(), UserDto.class);
        // set back username and password
        username = user.getUsername();
        password = user.getPassword();
    }
    @Override
    protected void successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain chain,
                                           Authentication authentication) throws IOException, ServletException {
        System.out.println("LoginFilter.successfulAuthentication");
        // write Authorization to headers of Response
        TokenAuthenticationResponse authenticationResponse = authentication.getAuthentication();
        String authorizationString = response.getHeader("Authorization").replace("Bearer ", "");
        String authorizationStringing = authenticationResponse.getAuthentication().replace("Bearer ", "");
        System.out.println("Authorization String=" + authorizationString);
        // return json to body request
        response.getWriter().print(String.format("{\"token\":\"%s\"}", authorizationString));
    }
}
```

Như chúng ta đã biết khi authen thành công thì chúng ta sẽ dùng token để có thể truy cập các API khác, vậy thì class xử lý token đó xem có đúng hay là sai ? vì vậy chúng ta cần phải tạo ra 1 file filter để hứng tất cả các request trước khi cho nó có quyền vào bất kì 1 API nào được chỉ định là phải có token mới được vào.

Ở đây mình sẽ tạo thêm 1 class để hứng các request đó và đặt tên là AuthenticationFilter.java



```
public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain)
        throws IOException, ServletException {
    Authentication authentication = TokenAuthenticationService
            .getAuthentication((HttpServletRequest) servletRequest);
    // get token from header request and validate token
    String authorizationString = ((HttpServletRequest) servletRequest).getHeader("Authorization").replace("Bearer ", "");
    System.out.println("Authorization String=" + authorizationString);
    SecurityContextHolder.getContext().setAuthentication(authentication);
    filterChain.doFilter(servletRequest, servletResponse);
}
```

Vậy là tạm xong giờ chúng ta chỉ cần tạo thêm 1 class để config xem API nào sẽ bị chặn và API nào sẽ được truy cập mà không cần qua class authen ở trên nên mình sẽ tạo thêm 1 class WebSecurityConfig.java .

Ở đây mình sẽ Override 2 method configure với configure đầu dùng để lọc API và configure 2 dùng để chỉ định sẽ lấy user trong DB để authen

Website bán xe ô tô trực tuyến

```
1 package deptato.API.security;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.http.HttpMethod;
6 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
7 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
8 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
9 import org.springframework.security.core.userdetails.UserDetailsService;
10 import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
11
12 @Configuration
13 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
14     @Autowired
15     private UserService userService;
16
17     @Override
18     protected void configure(HttpSecurity http) throws Exception {
19         http.csrf().disable();
20         http.headers().frameOptions().sameOrigin();
21         http.authorizeRequests()
22             .antMatchers(HttpMethod.POST, "/authen").permitAll()
23             .antMatchers(HttpMethod.GET, "/authen").permitAll();
24         http.formLogin()
25             .and()
26             .addFilterBefore(authenticationFilter("authen", authenticationManager()),
27                             UsernamePasswordAuthenticationFilter.class)
28             .addFilterBefore(new AuthenticationFilter(), UsernamePasswordAuthenticationFilter.class);
29     }
30
31     @Override
32     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
33         auth.userDetailsService(userService);
34     }
35 }
```

Ok vậy là xong chạy thử chương trình lên ta được như hình

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure under "API (RestAPISpringBoot master)". It includes packages like "src/main/java/deptayo/API" containing "DeptayoApplication.java", "DeptayoController.java", "DeptayoAPIController.java", "DeptayoAPIEntity.java", "DeptayoAPIRepository.java", "DeptayoAPISecurity.java", "DeptayoAPIService.java", and "DeptayoAPIServiceImpl.java". Other folders like "src/main/resources" and "target" are also listed.
- Code Editor View:** Displays the content of "DeptayoApplication.java". The code defines a main class "DeptayoApplication" that extends "SpringApplication". It includes annotations like "@SpringBootApplication" and "@EnableWebSecurity". The "main" method calls "SpringApplication.run(DeptayoApplication.class, args);".
- Outline View:** Shows the class "DeptayoApplication" with its methods, including the annotated "main" method.
- Toolbars and Menus:** Standard Eclipse toolbars and menus like File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.

Ở đây mình đang để port là 6913 lên ta sẽ mở postman test thử API vừa tạo

Website bán xe ô tô trực tuyến

The screenshot shows the Postman interface with a red box highlighting the JSON response body. The response code is 401 Unauthorized, and the message is "Authentication Failed: Bad credentials".

```
[{"timestamp": 1538231222197, "status": 401, "error": "Unauthorized", "message": "Authentication Failed: Bad credentials", "path": "/authen"}]
```

Mởxampp bắt localhost và MySQL nên để add thử 1 user vào sau đó tiến hành authen

The screenshot shows the Eclipse IDE interface with a Java project named "RestAppSpringRoot" open. The code editor displays a Java class with annotations for security. The XAMPP Control Panel window is also visible, showing services like Apache, MySQL, and Tomcat running.

truy cập vào <http://localhost/phpmyadmin/> ở đây trong application.xml mình đang sử dụng DB test ok vào sửa thôi. ở đây mình sẽ add username = hoainhien và pass: 123

Website bán xe ô tô trực tuyến

The screenshot shows the phpMyAdmin interface for a MySQL database named 'test'. The 'users' table is selected. An SQL query is entered in the query editor:

```
INSERT INTO `users`(`password`, `username`) VALUES ('123', 'devtayto')
```

The results pane on the right shows the columns: id, password, and username.

The screenshot shows the Postman application interface. A POST request is being prepared to the endpoint `http://localhost:6913/authen`. The Body tab is selected, showing the following JSON payload:

```
{"username": "devtayto", "password": "123"}
```

The request is highlighted with a red box.

Kết quả ta được

Website bán xe ô tô trực tuyến

The screenshot shows the Postman interface with a successful POST request to `http://localhost:6913/authen`. The JSON body is `{ "username": "dientotd", "password": "123" }`. The response status is 200 OK, time 37 ms, size 605 B, and the token is displayed in the response body.

Đã lấy được token giờ ta thử dùng để truy cập các API khác, thêm Bearer ở đầu cách ra xong điền token như hình để authen

The screenshot shows the Postman interface with a successful GET request to `http://localhost:6913/students`. The Authorization header is set to `Bearer eyJhbGciOiUzUzUmIj9eyJzdWIjOiZXZ0YXbyImlmV4cC16MTUzODizMjQ1Nh0.or...` and the Content-Type header is set to `application/json`. The response status is 200 OK, time 70 ms, size 389 B.

II. CÁU HÌNH VÀ TRIỂN KHAI

1. Cấu hình và triển khai lên host

1.1. Chuẩn bị

Project java đã chạy được trên localhost, đã được push lên github.

Hosting để triển khai ứng dụng web java. Nhóm dùng host free có tên Heroku.

➤ Link: <https://www.heroku.com/>

Website bán xe ô tô trực tuyến

Vì hostHeroku không hỗ trợ MySQL nên phải cần database online, ở đây nhóm dùng Freemysqlhosting.

- Link: <https://www.freemysqlhosting.net/>

Nếu có thẻ VISA hoặc MasterCard thì có thể dùng host AWS của Amazon vì cái này nó sẽ chạy nhanh hơn và nó hỗ trợ cả MySQL

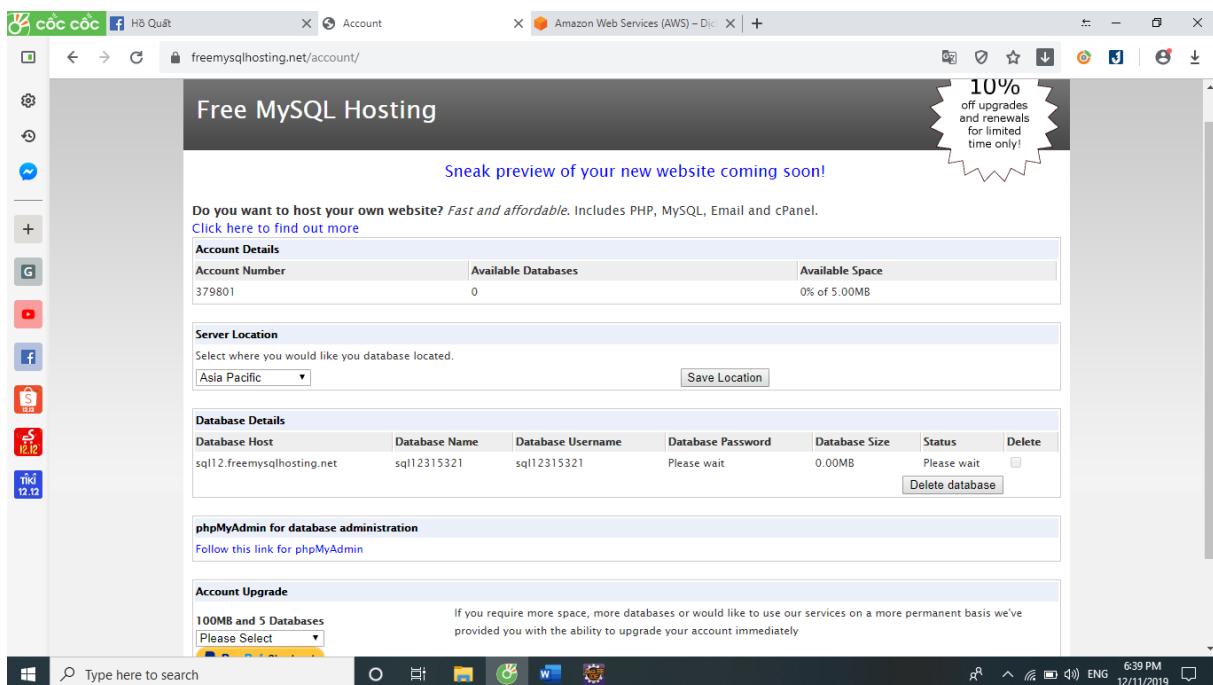
- Link: <https://aws.amazon.com/vi/>

1.2. Triển khai

1.2.1. Triển khai database lên database online

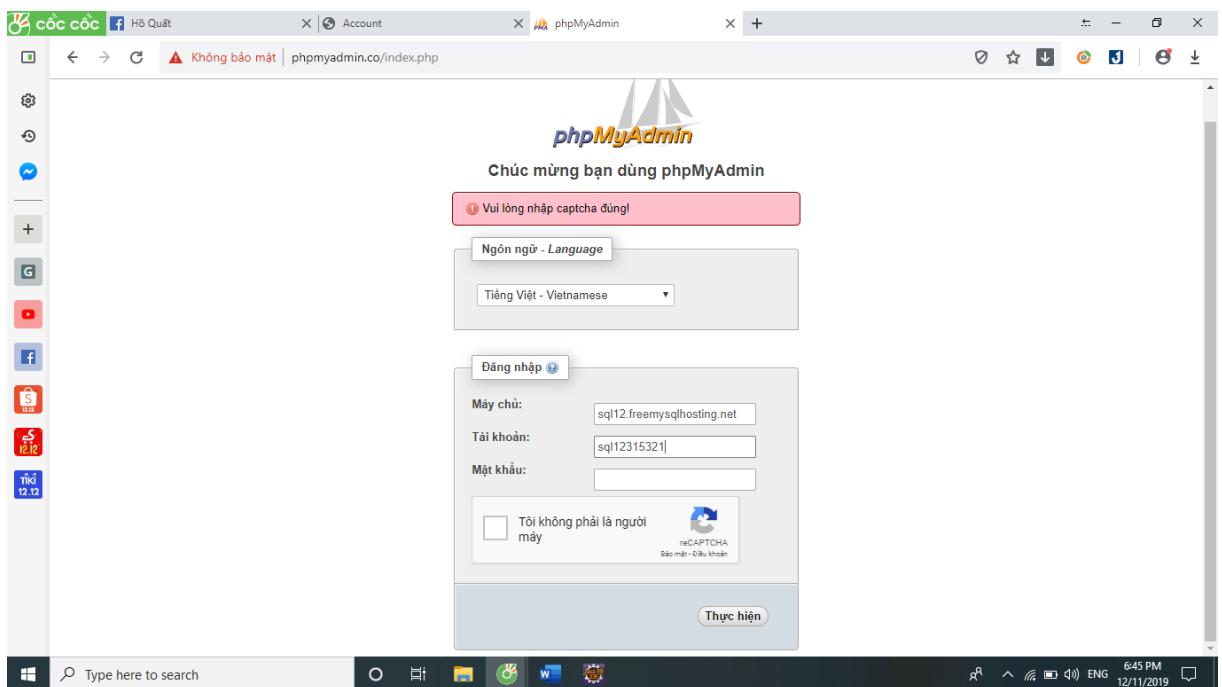
Bước 1: Truy cập vào link ở trên để vào trang chủ Freemysqlhosting.

Bước 2: Đăng ký tài khoản để sử dụng để đăng nhập vào Freemysqlhosting. Dưới đây là giao diện khi đăng nhập thành công.



Bước 3: Nhấn vào Follow this link for phpMyAdmin để truy cập vào phpmyadmin.

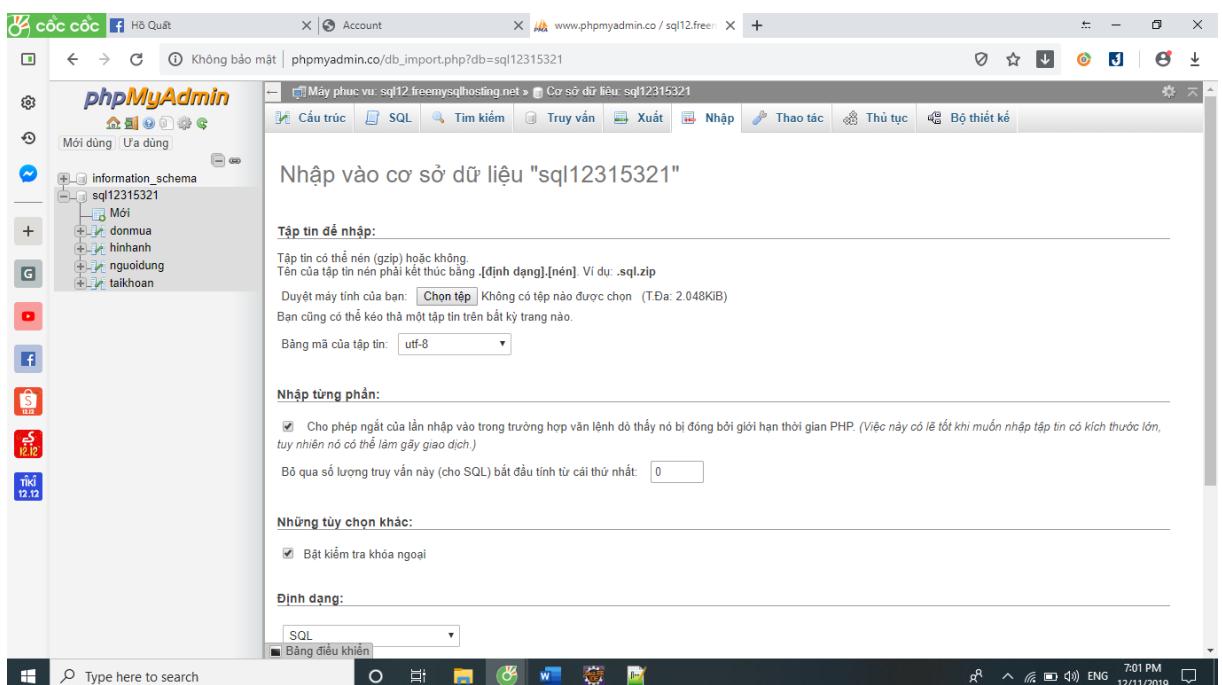
Website bán xe ô tô trực tuyến



Bước 4: Thông tin cần điền:

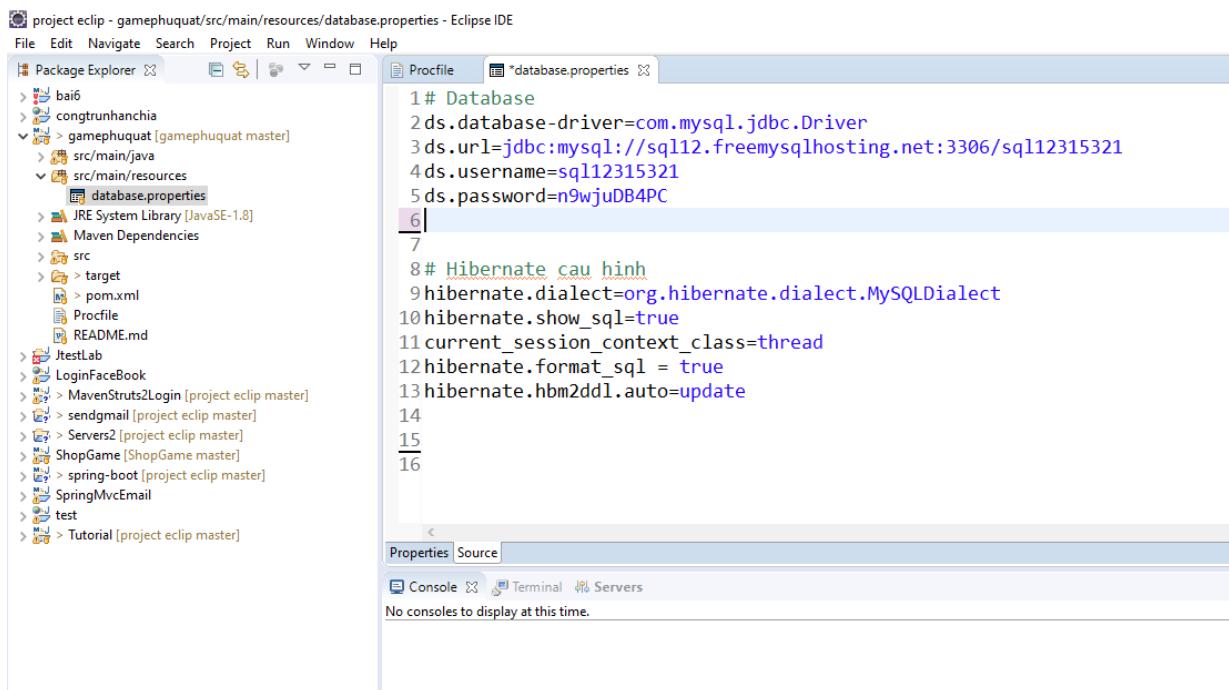
- Máy chủ và tài khoản: có trong giao diện khi đăng nhập thành công.
- Mật khẩu: sẽ gửi vào email (cái này nó gửi về hơi lâu).

Bước 5: Sau khi đăng nhập thành công, vào phần import => chọn file database (.sql)



Bước 6: Sau khi hoàn tất tất cả thì việc còn lại là vào project cấu hình lại đường dẫn trong database (nhớ push lên github).

Website bán xe ô tô trực tuyến



The screenshot shows the Eclipse IDE interface with the 'project eclips - gamephuquat/src/main/resources/database.properties - Eclipse IDE' window active. On the left is the Package Explorer view showing various Java projects and files. The main area displays the contents of the database.properties file:

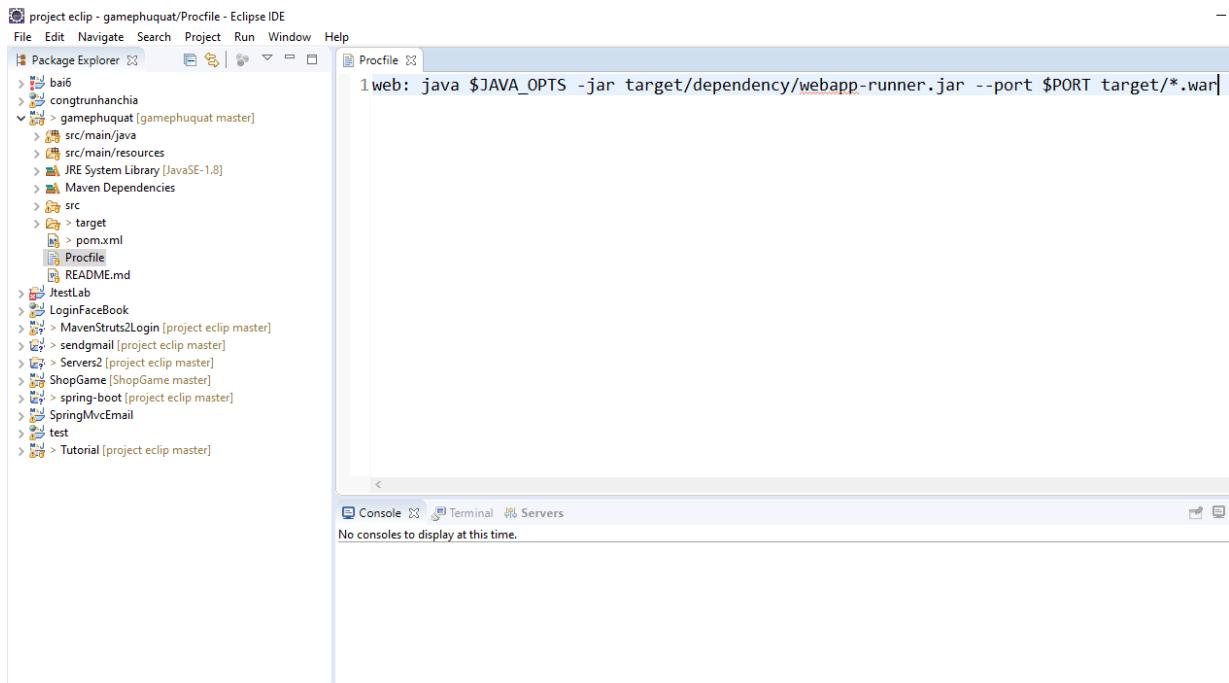
```
1 # Database
2 ds.database-driver=com.mysql.jdbc.Driver
3 ds.url=jdbc:mysql://sql112.freemysqlhosting.net:3306/sql12315321
4 ds.username=sql12315321
5 ds.password=n9wjuDB4PC
6
7
8 # Hibernate cau hinh
9 hibernate.dialect=org.hibernate.dialect.MySQLDialect
10 hibernate.show_sql=true
11 current_session_context_class=thread
12 hibernate.format_sql = true
13 hibernate.hbm2ddl.auto=update
14
15
16
```

Below the code editor, there are tabs for 'Properties' and 'Source'. At the bottom, there are tabs for 'Console', 'Terminal', and 'Servers'.

1.2.2. Triển khai ứng dụng lên host

Bước 1:Cần cấu hình một chút trong project:

- Thêm 1 procfile vào trong project với nội dung như sau:



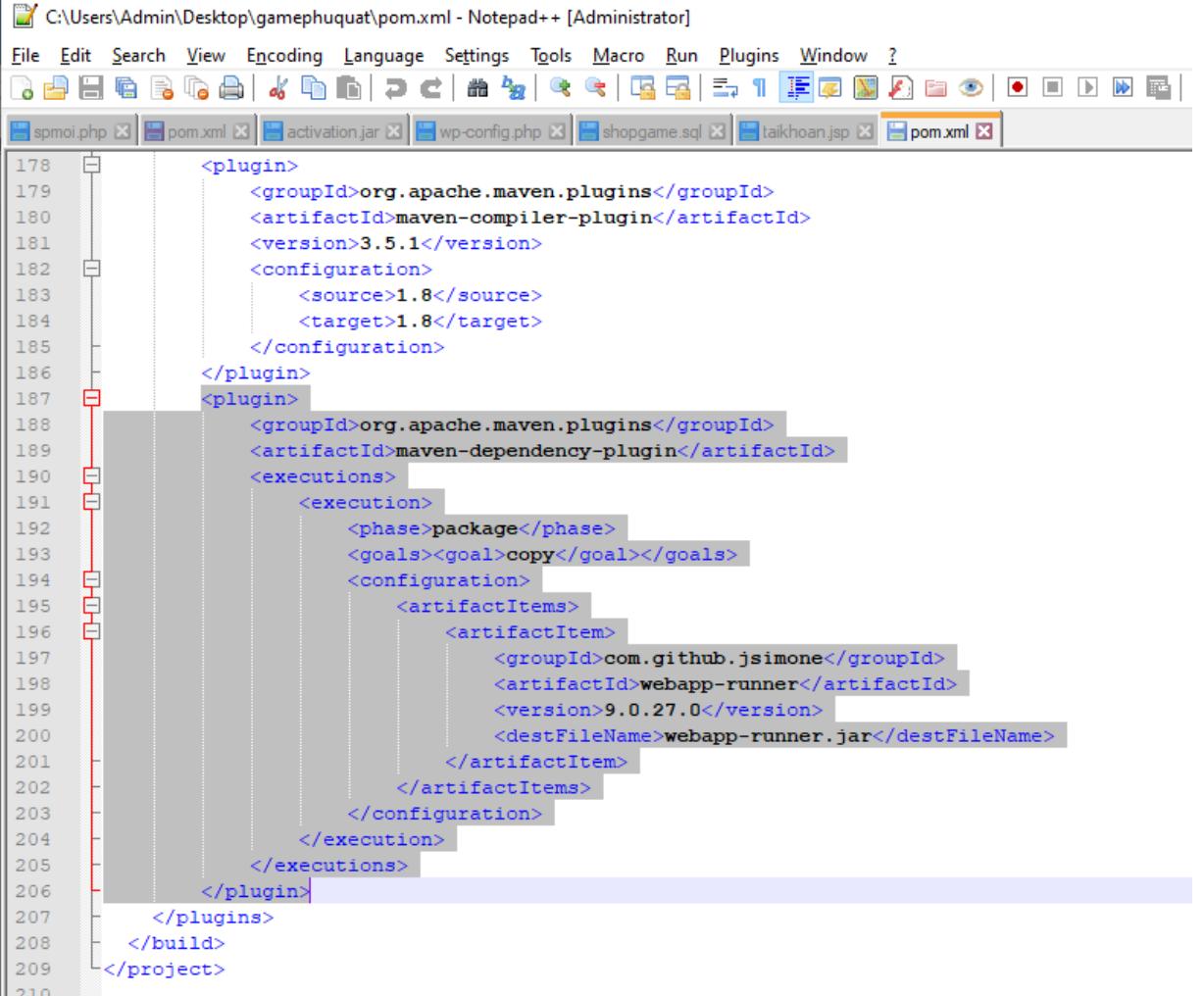
The screenshot shows the Eclipse IDE interface with the 'project eclips - gamephuquat/Procfile - Eclipse IDE' window active. The Procfile contains the following command:

```
1 web: java $JAVA_OPTS -jar target/dependency/webapp-runner.jar --port $PORT target/*.war
```

Below the code editor, there are tabs for 'Console', 'Terminal', and 'Servers'.

- Thêm đoạn plugin này vào file pom.xml

Website bán xe ô tô trực tuyến



```
C:\Users\Admin\Desktop\gamephuquat\pom.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
spmoi.php pom.xml activation.jar wp-config.php shopgame.sql taikhoan.jsp pom.xml

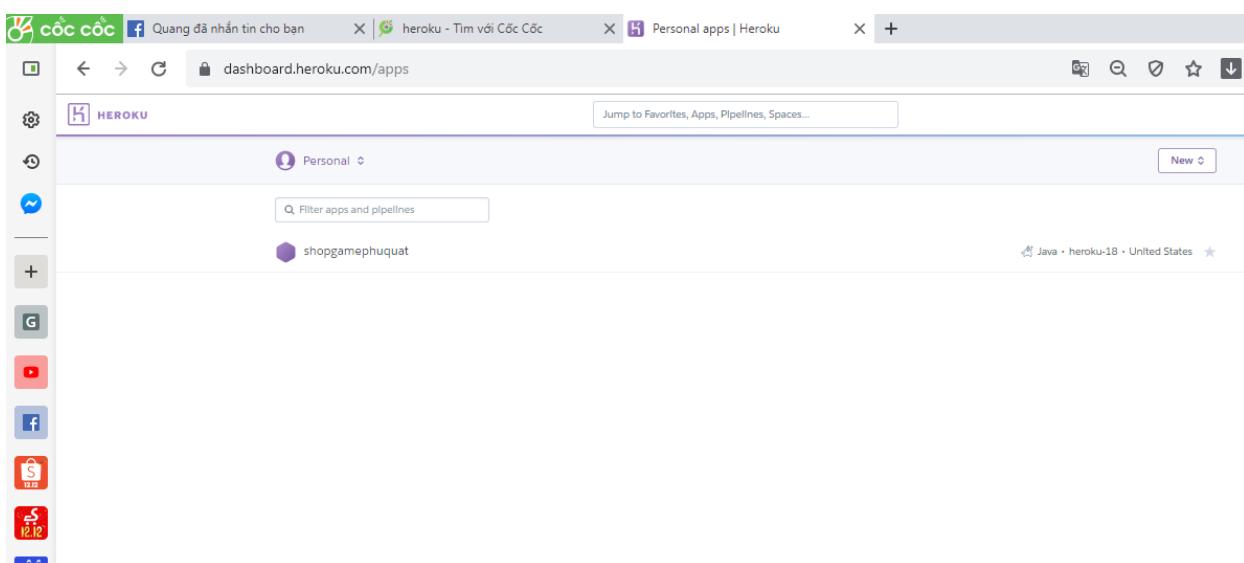
178     <plugin>
179         <groupId>org.apache.maven.plugins</groupId>
180         <artifactId>maven-compiler-plugin</artifactId>
181         <version>3.5.1</version>
182         <configuration>
183             <source>1.8</source>
184             <target>1.8</target>
185         </configuration>
186     </plugin>
187     <plugin>
188         <groupId>org.apache.maven.plugins</groupId>
189         <artifactId>maven-dependency-plugin</artifactId>
190         <executions>
191             <execution>
192                 <phase>package</phase>
193                 <goals><goal>copy</goal></goals>
194             <configuration>
195                 <artifactItems>
196                     <artifactItem>
197                         <groupId>com.github.jsimone</groupId>
198                         <artifactId>webapp-runner</artifactId>
199                         <version>9.0.27.0</version>
200                         <destFileName>webapp-runner.jar</destFileName>
201                     </artifactItem>
202                 </artifactItems>
203             </configuration>
204         </execution>
205     </executions>
206 </plugin>
207 </plugins>
208 </build>
209 </project>
210
```

Bước 2: Truy cập vào đường link Heroku ở trên.

Bước 3: Đăng ký tài khoản và sau đó đăng nhập.

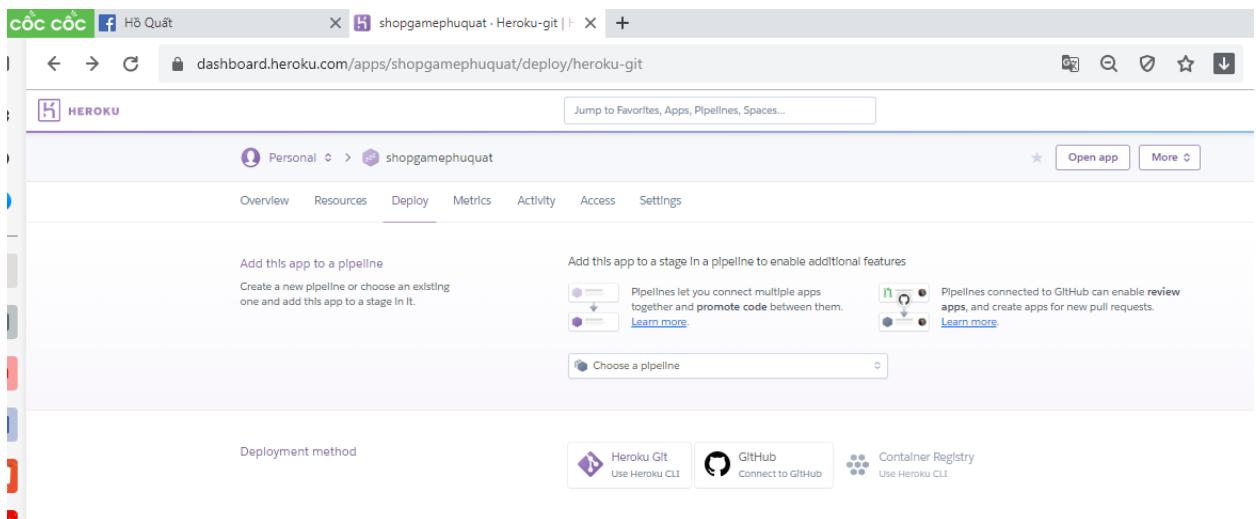
Bước 4: Khi đăng nhập thành công, giao diện trang chủ sẽ hiện ra.

Bước 5: Nhấn New và đặt tên cho website.



Website bán xe ô tô trực tuyến

Bước 6: Nhấn vào tên website mới tạo ra để tiến hành deploy



Bước 7:Nhấn vào deloy phần Deployment methodta chọn **biểu tượng github** => chọn **connect to github**.

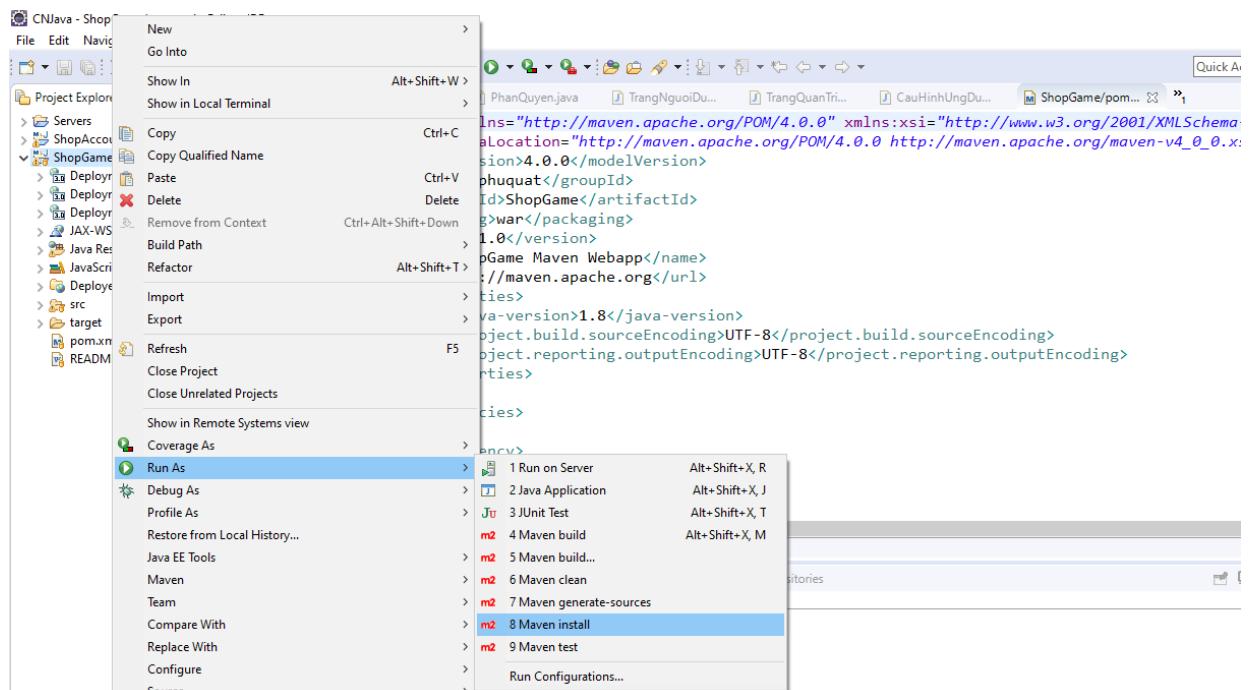
Bước 8:Sau khi đăng nhập github xong chọn project cần đưa lên host và nhấn **connect** => nhấn tiếp **Deploy Branch**

Bước 9:Sau khi hệ thống build xong nhấn vào **View** nếu hiện lên giao diện trang chủ thì đã thành công.

2. Cấu hình và triển khai lên localhost

2.1. Chuẩn bị

Bước 1:File .war của project: Chuột phải vào project => Run As => Maven install



Bước 2: Nếu build file .war thành công thì sẽ có thông báo ở màn hình console như sau:

```
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  16.342 s
[INFO] Finished at: 2019-12-12T09:04:41+07:00
[INFO] -----
```

Để triển khai được file .war lên tomcat thì phải có tài khoản đăng nhập vào Tomcat Web Application Manager.

Bước 3: Vào thư mục tomcat => thư mục conf => mở file “tomcat-users.xml” => thêm đoạn code sau vào trong thẻ <tomcat-users>, với password và username tùy ý.

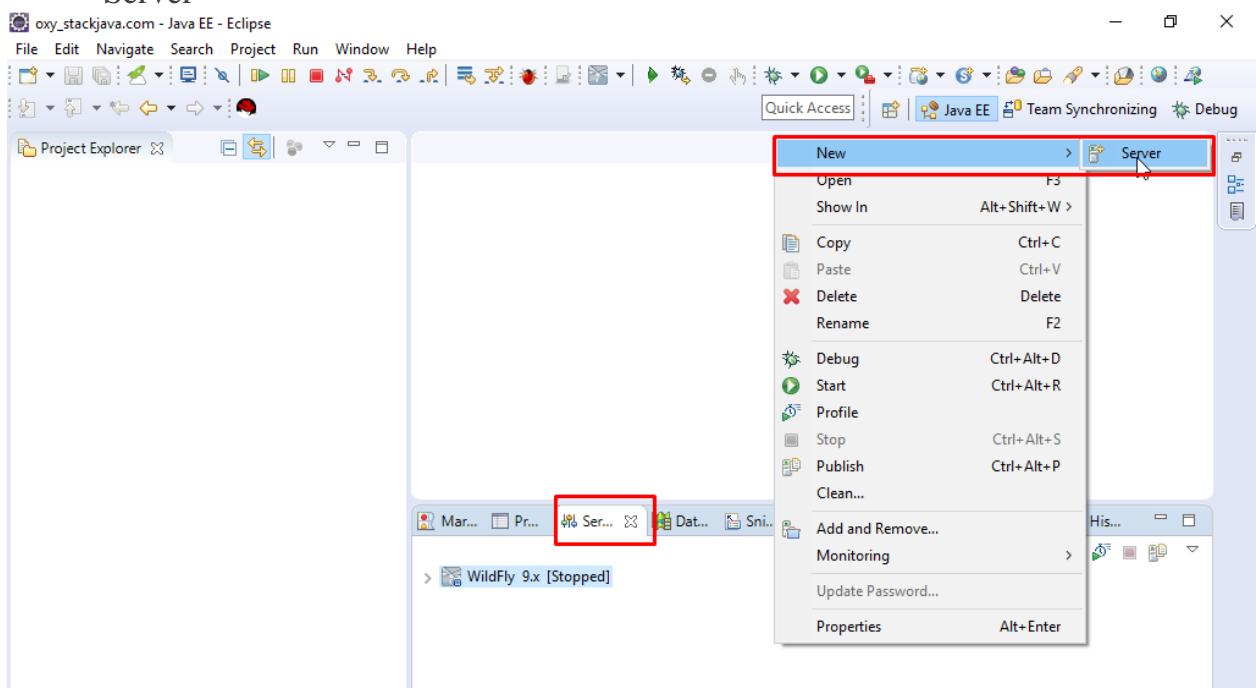
```
<role rolename="manager-gui"/>
<user password="admin" roles="manager-gui,manager-script,admin" username="admin"/>
```

2.2. Triển khai

Deploy từ Eclipse

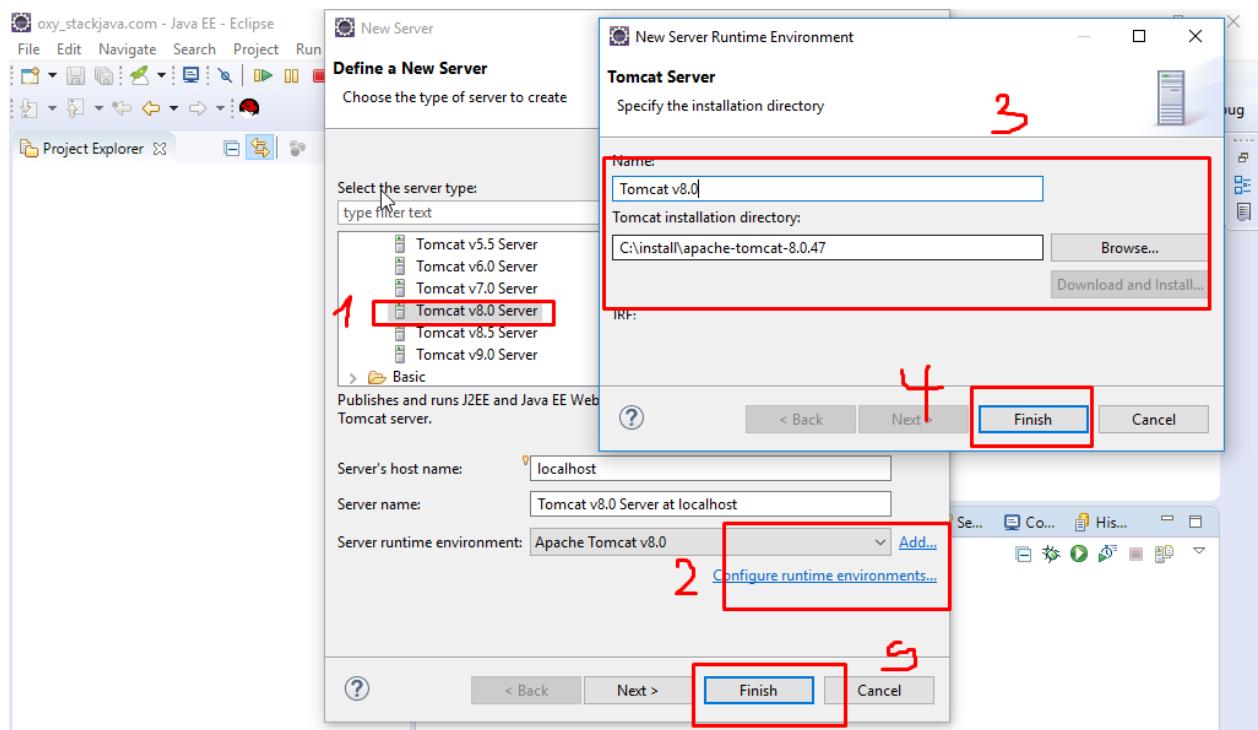
Trước tiên bạn cài tomcat vào eclipse:

- Chọn tab Servers > click chuột phải vào vùng trống của tab Servers và chọn new > Server

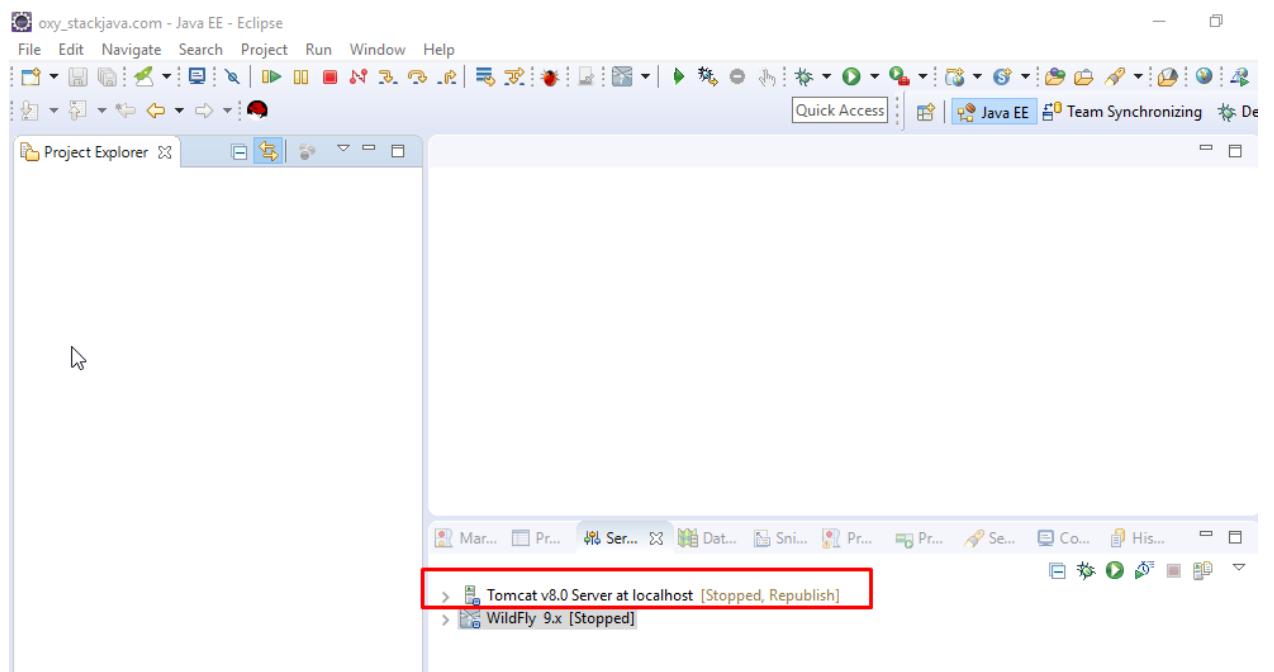


Chọn Apache > Tomcat v8.0 Server > click vào Add Configure runtime environments để trả lời folder tomcat.

Website bán xe ô tô trực tuyến

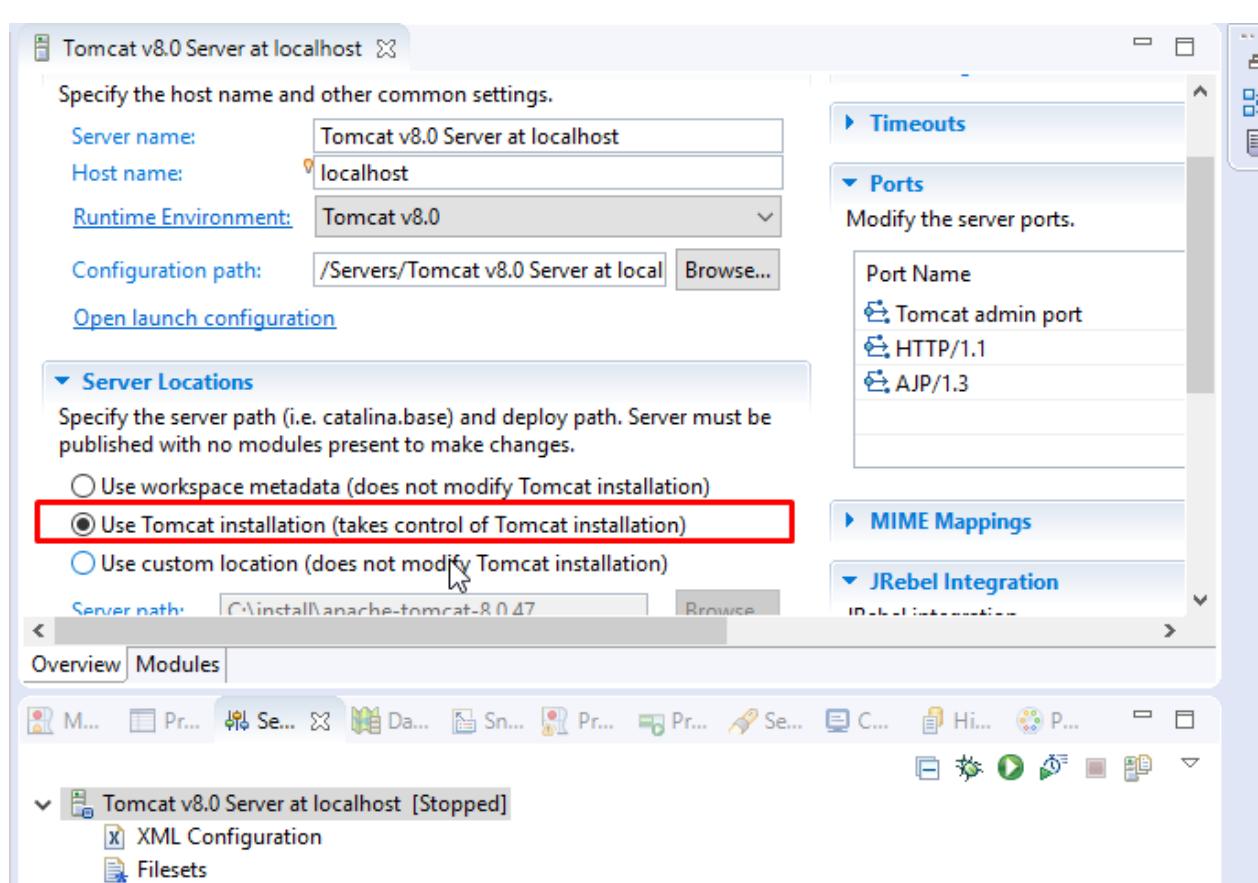


Kết quả sau khi cài thành công:

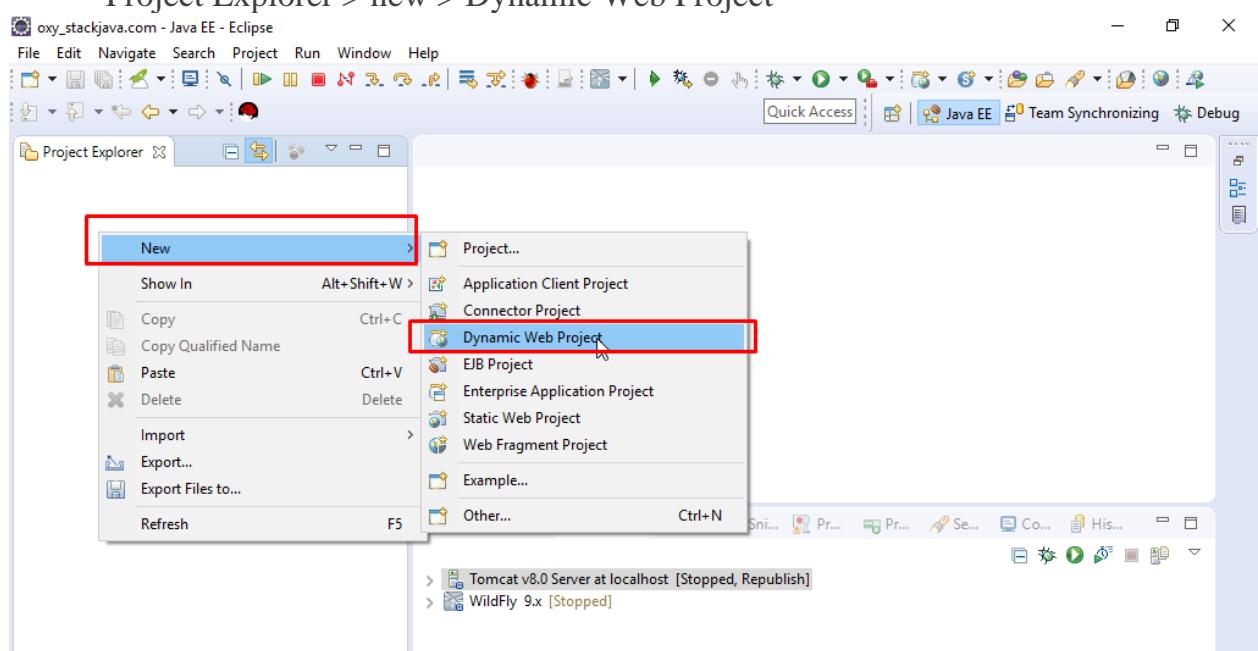


Trò cài đặt tomcat tới folder tomcat để có thể thực hiện truy cập trang quản lý của tomcat.

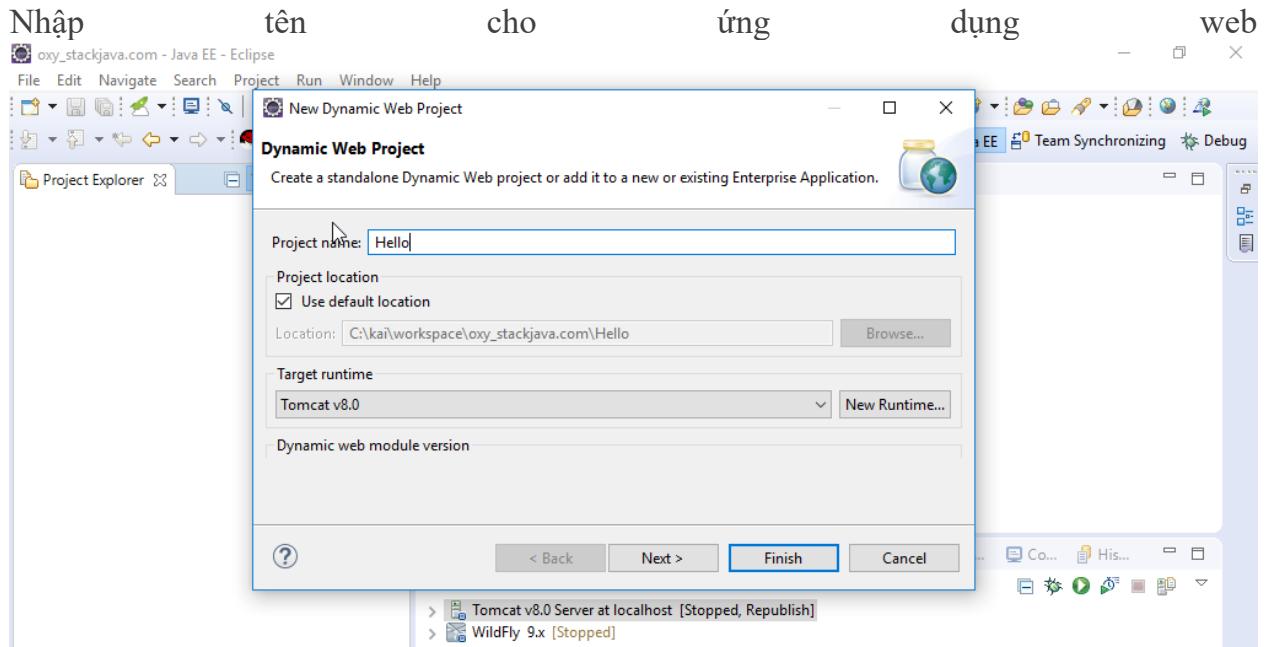
Website bán xe ô tô trực tuyến



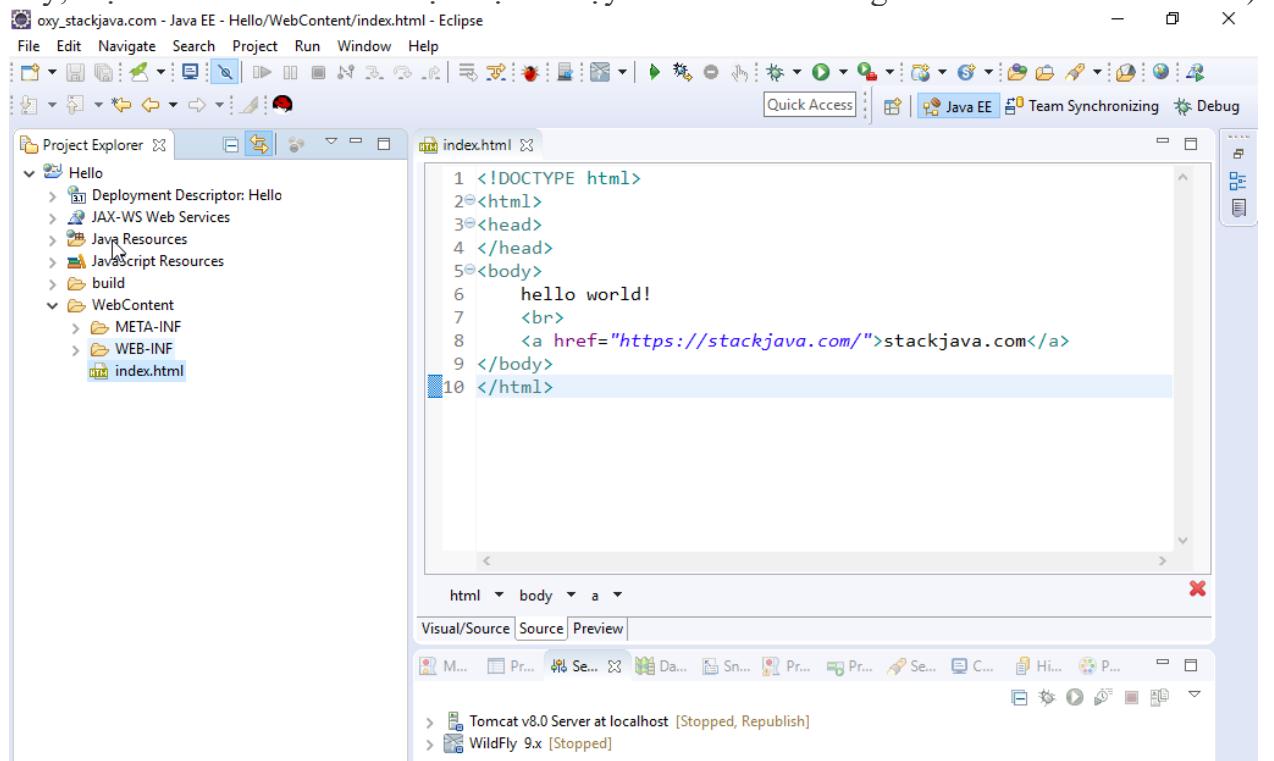
- Tạo 1 ứng dụng web đơn giản trên eclipse: click chuột phải vào vùng trống của Project Explorer > new > Dynamic Web Project



Website bán xe ô tô trực tuyến

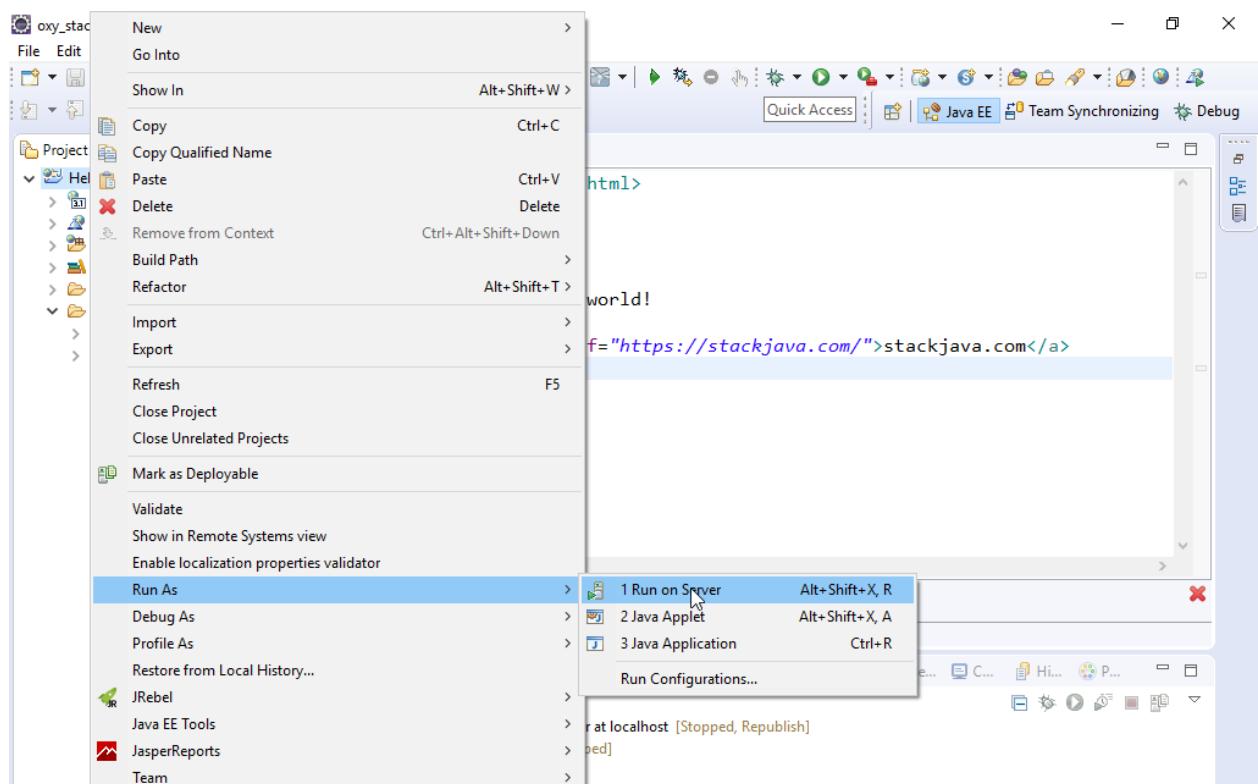


Tạo một file index.html (mặc định ban đầu chạy ứng dụng web nó sẽ chạy file index.html này, bạn có thể sửa file mặc định chạy ban đầu ở trong file WEB-INF/web.xml)

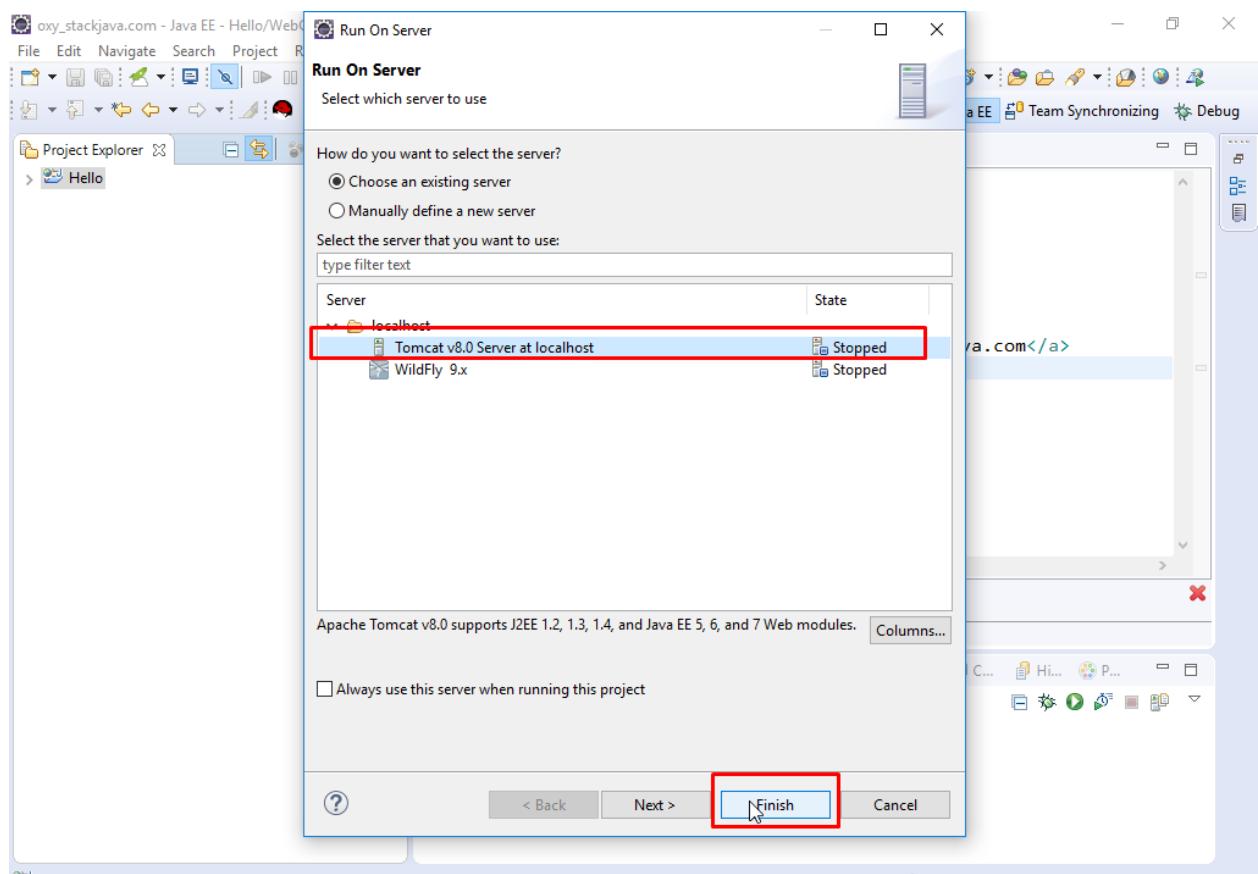


Chạy ứng dụng web vừa tạo trên tomcat: chuột phải vào project > Run As > Run on Server

Website bán xe ô tô trực tuyến



Chọn server Tomcat và click Finish



Website bán xe ô tô trực tuyến

Sau khi deploy bạn sẽ thấy ứng dụng Hello hiện lên trên trang manager của tomcat.

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/Hello	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Kết quả:

localhost:8080/Hello/

hello world!
stackjava.com

III. KẾT QUẢ ĐẠT ĐƯỢC

1. Kết quả đạt được

Sau thời gian thực hiện đề tài, hệ thống cũng đã hoàn thành và đạt được một số kết quả sau:

- Hiểu rõ quy trình mua – bán xe thông qua hệ thống
- Xây dựng được hệ thống mua – bán xe đáp ứng 1 số nhu cầu nhất định của mọi người
- Tìm hiểu và nắm rõ về các công cụ để xây dựng hệ thống như: Eclipse, MySQL, Git.
- Học được cách thức xây dựng một website sử dụng framework Spring MVC và Hibernate kết hợp với một số công nghệ khác.
- Giao diện của website phù hợp với mọi người, thu hút và dễ sử dụng

- Hệ thống đã giúp khách hàng tiết kiệm thời gian, công sức để có được những chiếc xe ưng ý
- Thực hiện kiểm thử phần mềm ở mức độ lập trình.

3. Hạn chế của đề tài

Chưa thể cài đặt chức năng nạp thẻ trực tuyến cho khách hàng.

Chưa thể cập nhật nhiều loại thẻ thanh toán điện tử.

Có thể hình ảnh đăng lên không chính xác 100% với đời thực.

3. Hướng phát triển của đề tài

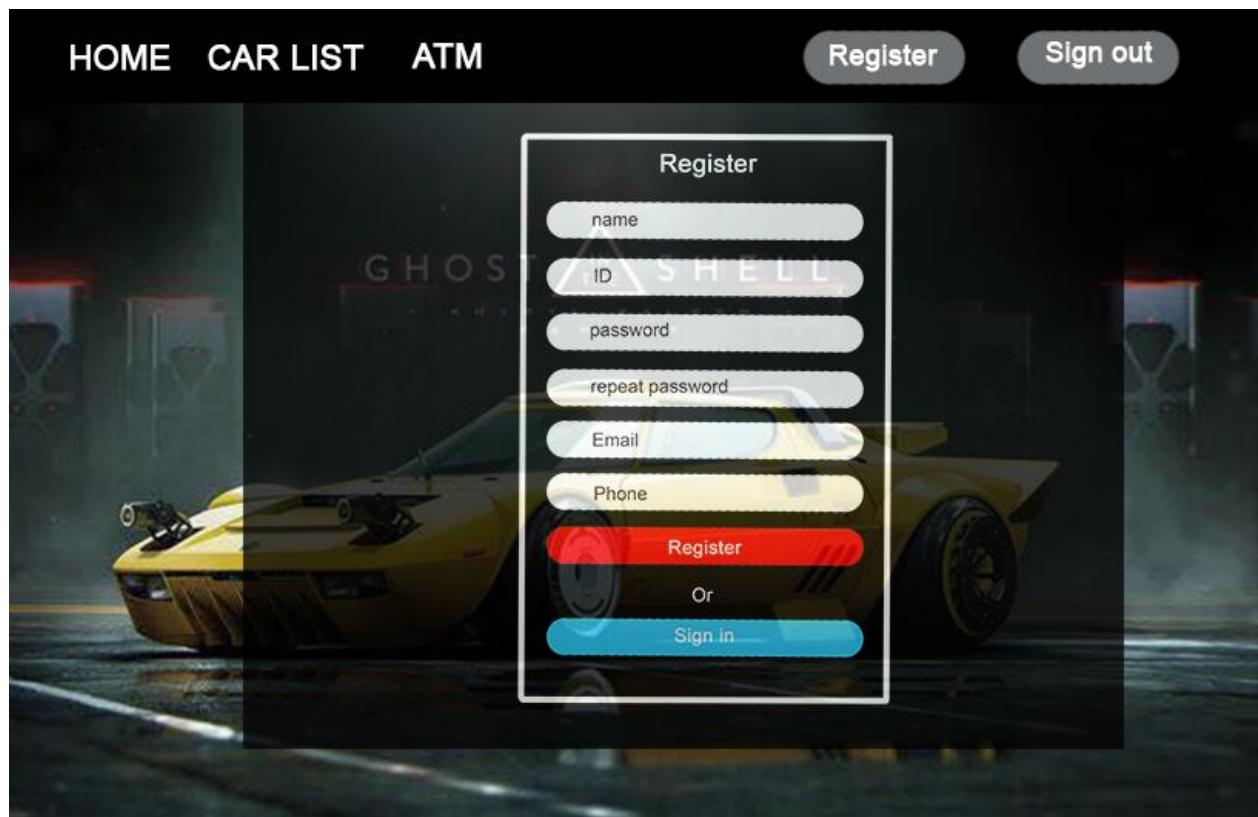
Vì thời gian có hạn nên đề tài còn sai sót cũng như thiếu sót về nhiều mặc chức năng và phi chức năng, giao diện....

Trong khoản thời gian tiếp theo, nhóm chúng em sẽ cố gắng hoàn thiện hệ thống và được mọi người sử dụng rộng rãi. Đồng thời cung cấp các chuyên ngành khác để đảm bảo nhu cầu của khách hàng. Và trong tương lai gần nhóm sẽ hoàn thiện các chức năng :

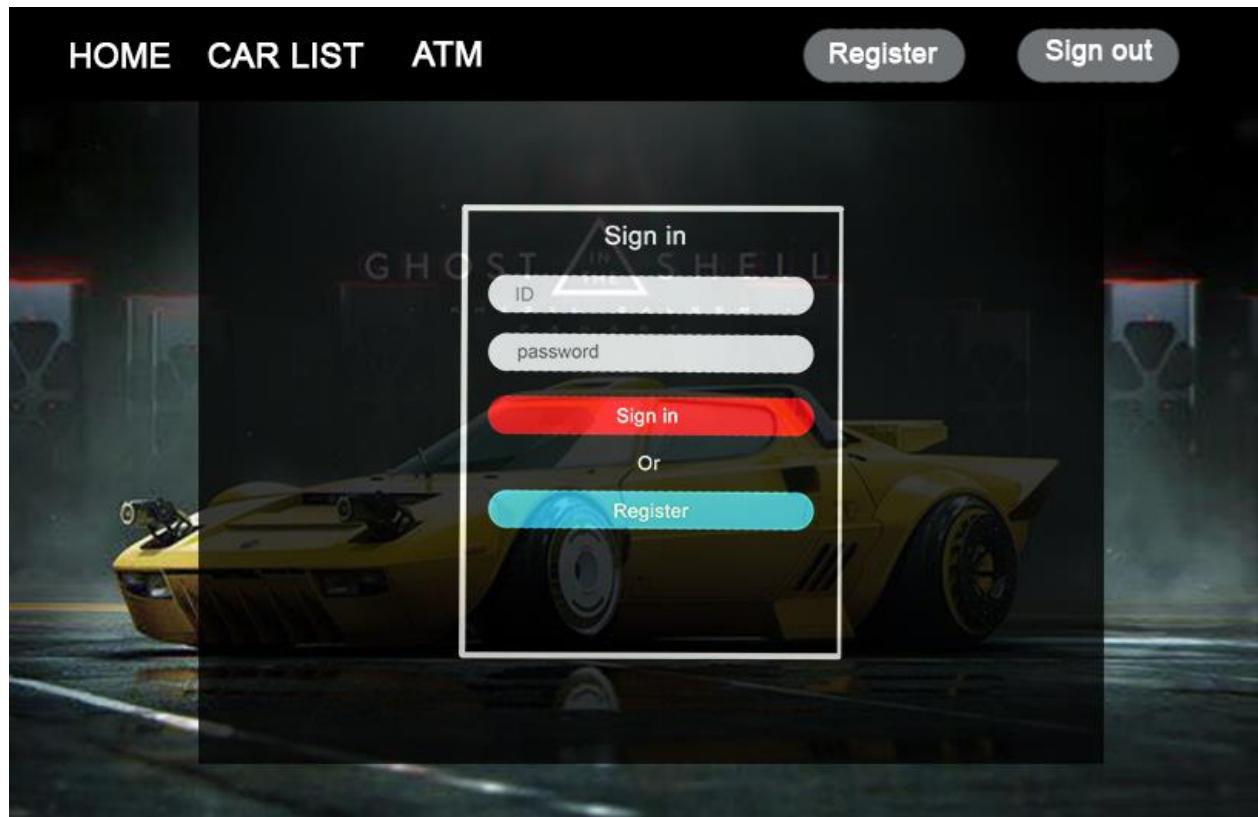
- Thêm chức năng gửi SMS báo thông tin tài khoản đã mua cho khách hàng.
- Cài đặt chức năng nạp thẻ trực tuyến cho khách hàng.
- Thêm chức năng chat cho khách hàng và người quản trị.
- Tăng độ tin cậy cho khách hàng mua bằng cách xác minh rõ các đối tượng bán.

4. Giao diện

4.1. Đăng ký



4.2. Đăng nhập



4.3. Trang chủ

HOME CAR LIST ATM

Register

Sign out

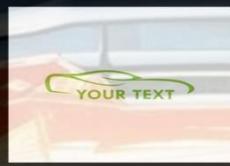
DMC #MAKING GOOD THINGS BETTER
HTTP://WWW.DMC.AG



Best quality

• • •

ACCOUNTS



Web's Address



Your account

CATEGORIES



BOOM CAR
Amount : 10
Sold : 2



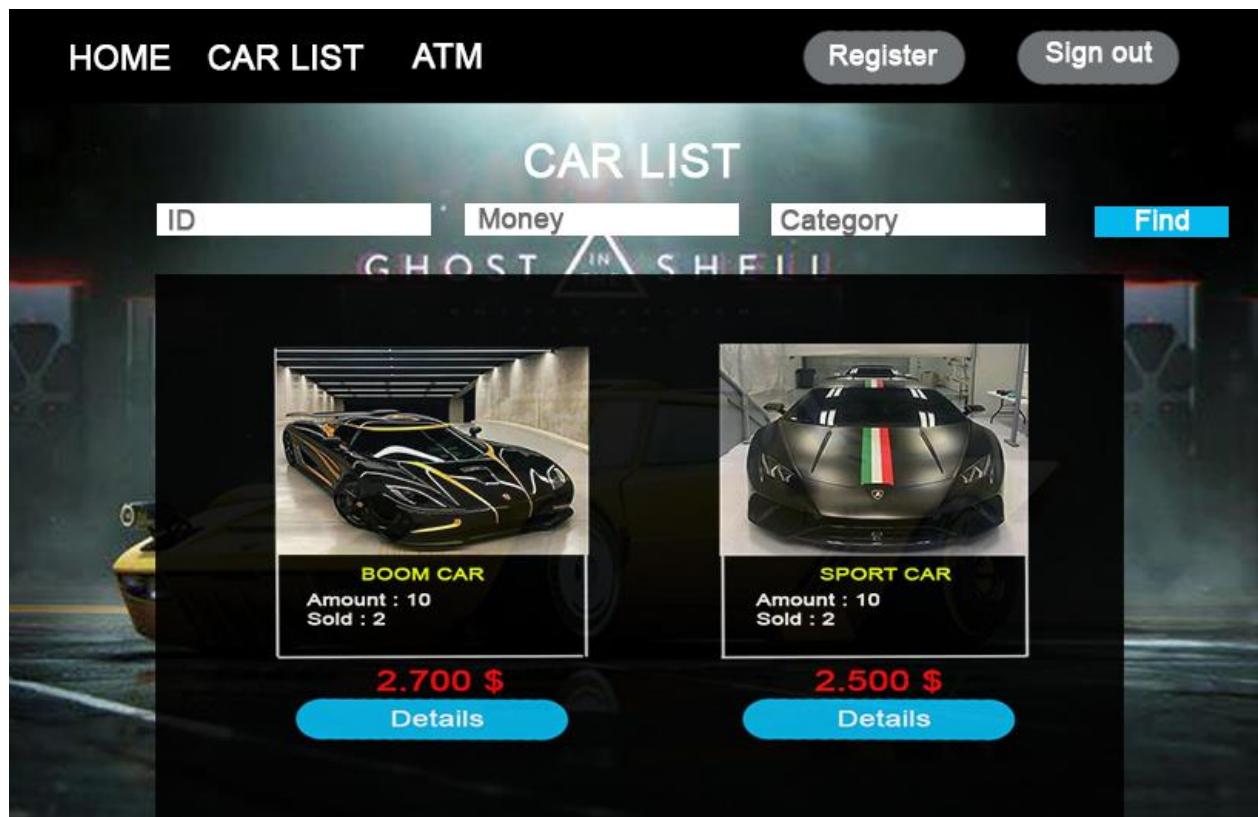
SPORT CAR
Amount : 10
Sold : 2

www.ShopCar.Com

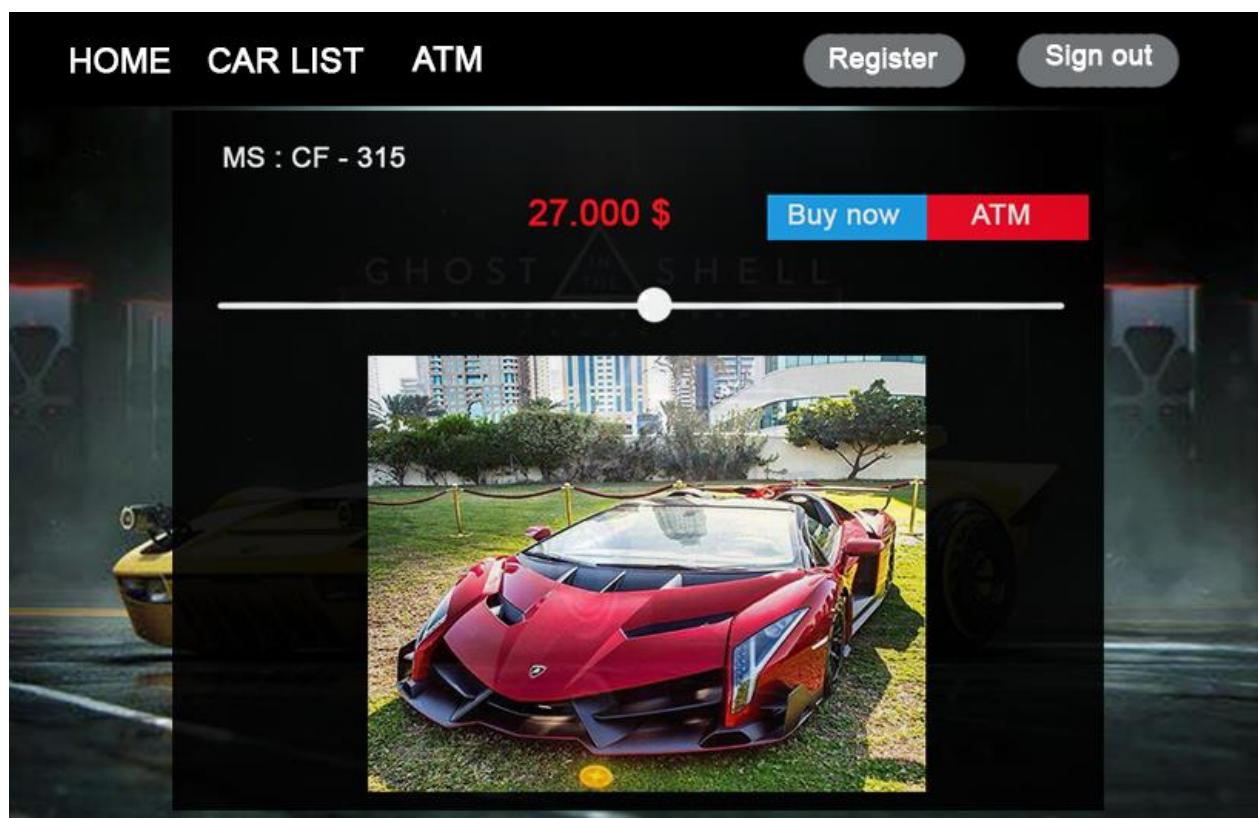
Hotline : **0327549906**

۷۳۳۳ ۷۳۳۳
7333 7333
VKJ VKJ

4.4. Danh sách xe dành cho khách hàng



4.5. Chi tiết xe



4.6. Chuyển tiền từ ATM - Ví điện tử

Website bán xe ô tô trực tuyến

Nạp tiền từ ATM hoặc ví điện tử

Nạp tiền từ ATM hoặc ví điện tử

HƯỚNG DẪN CHI TIẾT CHUYỂN TIỀN TỪ ATM - VÍ ĐIỆN TỬ

ATM	Ví điện tử	
CHUYỂN TIỀN TỪ NGÂN HÀNG (Tên chủ khoán: TRAN VAN A)		
Tên ngân hàng	Số tài khoản	Chi nhánh
Vietcombank	0123456789	VIỆT NAM
DongABank	0123456789	VIỆT NAM
BIDV	0123456789	VIỆT NAM

Nội dung thanh toán: (ID tài khoản muốn mua và tên tài khoản đăng nhập vào shop)

Chuyển xong liên hệ fb: <https://www.facebook.com/abc> hoặc Hotline 0123456789 để được xử lý.

NAP TIỀN TỪ VÍ ĐIỆN TỬ (Tên chủ ví: TRAN VAN A)

ATM	Ví điện tử
Tên ví	Số ví
MoMo	0335623523
Viettelpay	0335623523
Zalopay	0335623523

Nội dung thanh toán: (ID tài khoản muốn mua và tên tài khoản đăng nhập vào shop)

Chuyển xong liên hệ fb: <https://www.facebook.com/abc> hoặc Hotline 0123456789 để được xử lý.

ĐÓNG

ĐÓNG

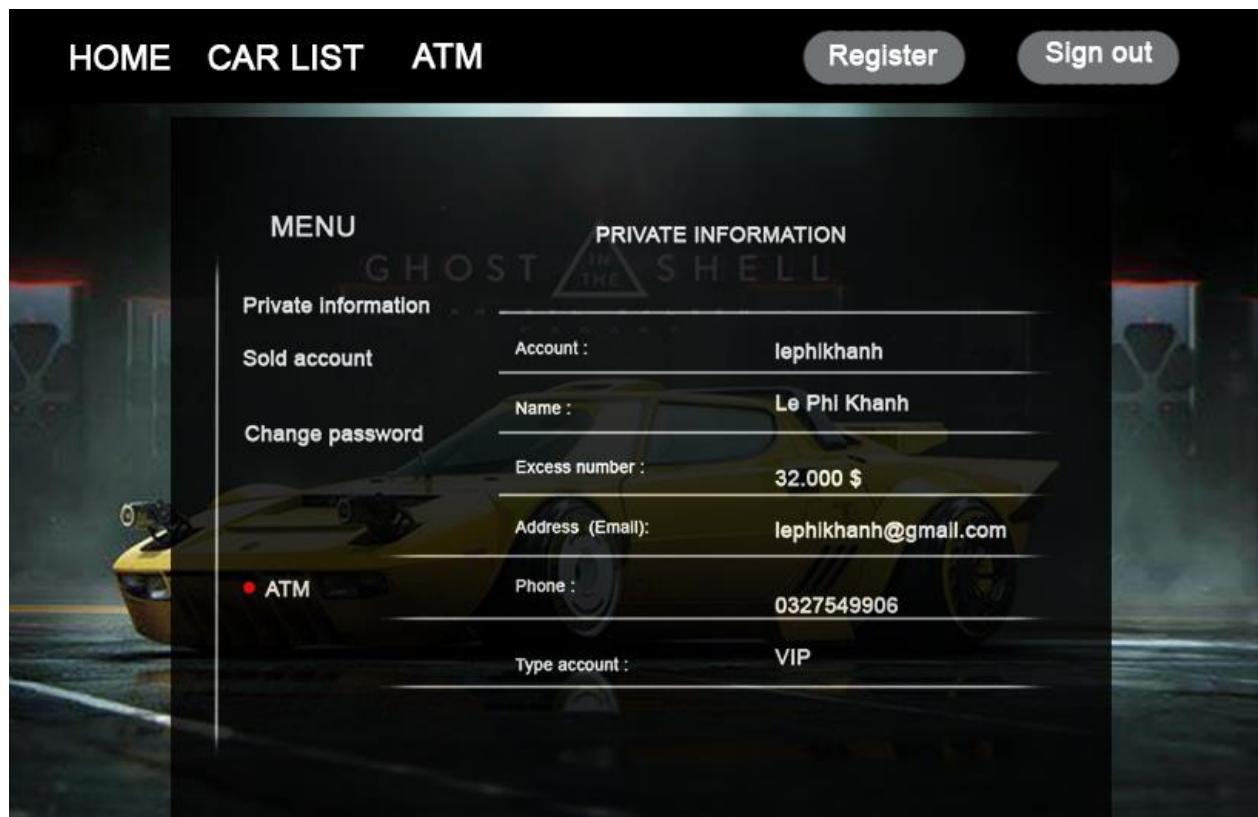
4.7. Danh sách xe đã mua của khách hàng

The screenshot shows a dark-themed website interface. At the top, there are navigation links: HOME, CAR LIST, ATM, Register, and Sign out. Below these, a large image of a yellow sports car is displayed with the text "GHOST IN THE SHELL" overlaid. On the left side, there is a vertical sidebar with a menu and some account information. The menu includes "MENU", "Private Information", "Sold account", and "Change password". The "Sold account" section is currently selected and displays a table titled "SOLD ACCOUNT" with two rows of data. The table columns are STT, TIME, ID, and VALUE. The data is as follows:

STT	TIME	ID	VALUE
1	14:01:14 24/07/2019	90	27.000 \$
2	15:01:14 25/07/2019	90	27.000 \$
			54.000 \$

At the bottom of the sidebar, there is a link labeled "• ATM".

4.8. Thông tin cá nhân



CÔNG CỤ HỖ TRỢ

- Eclipse IDE for Enterprise Java Developers – version: 2019-06 (4.12.0)
- MySQL Workbench – version: 8.0.18
- Tomcat Server – version: 8.5.45

TÀI LIỆU THAM KHẢO

- <https://o7planning.org/vi/11281/huong-dan-spring-mvc-security-voi-hibernate>
- <https://www.javatpoint.com/spring-mvc-crud-example>
- <https://giasutinhoc.vn/website/spring-framework/huong-dan-xu-ly-database-trong-spring-mvc-5-voi-hibernate/>
- <https://stackjava.com/spring/code-vi-du-spring-mvc-spring-jdbc.html>
- <https://stackoverflow.com/>
- <https://www.w3schools.com/>
- <https://getbootstrap.com/>