|  |  |
| --- | --- |
|  |  |
| NGUYỄN CÔNG THÀNH CÔNG NGHỆ THÔNG TIN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN |
|  |
| **XÂY DỰNG WEBSITE BÁN Ô TÔ**  **SỬ DỤNG VUEJS VÀ ASP.NET CORE** |
|  |
| **CBHD:** ThS. Vũ Duy Giang  **Sinh viên:** Nguyễn Công Thành  **Mã số sinh viên:** 2020603178 |
|  |
| **Hà Nội – Năm 2024** |

LỜI NÓI ĐẦU

*Em xin bày tỏ lòng biết ơn sâu sắc đến Trường Đại học Công nghiệp Hà Nội và Thầy giáo ThS. Vũ Duy Giang đã dành thời gian và kiến thức quý báu để hướng dẫn và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp ngành Công nghệ Thông tin.*

*Em không thể không đề cập đến sự cống hiến và sự hỗ trợ chân thành của Thầy giáo Giang trong suốt quãng thời gian chúng tôi làm việc với ông. Sự kiên nhẫn, kiến thức sâu rộng và sự chỉ dẫn tận tình của Thầy giáo Giang đã giúp em vượt qua những thách thức trong quá trình nghiên cứu và thực hiện đồ án.*

*Em cũng muốn bày tỏ lòng biết ơn đến Ban giám hiệu và các giảng viên của Trường Đại học Công nghiệp Hà Nội đã cung cấp môi trường học tập và nghiên cứu chuyên nghiệp, giúp em phát triển kiến thức và kỹ năng trong suốt thời gian học tập tại trường.*

*Em xin chân thành cảm ơn!*

MỤC LỤC

[LỜI NÓI ĐẦU i](#_Toc167107110)

[DANH MỤC HÌNH ẢNH iii](#_Toc167107111)

[DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT v](#_Toc167107112)

[MỞ ĐẦU](#_Toc167107113)

[1. Lý do chọn đề tài 5](#_Toc167107114)

[2. Mục tiêu đề tài 5](#_Toc167107115)

[3. Đối tượng nghiên cứu 6](#_Toc167107116)

[4. Tính cấp thiết của đề tài 6](#_Toc167107117)

[Chương 1. CƠ SỞ LÝ THUYẾT 7](#_Toc167107118)

[1.1. Các khái niệm cơ bản 7](#_Toc167107119)

[Chương 2. CÔNG NGHỆ SỬ DỤNG ASP.NET VÀ VUEJS 10](#_Toc167107120)

[2.1. Công nghệ sử dụng 10](#_Toc167107121)

[2.2. Công cụ 17](#_Toc167107122)

[Chương 3. XÂY DỰNG WEBSITE BÁN Ô TÔ 18](#_Toc167107123)

[3.1. Nghiên cứu, khảo sát thị trường 18](#_Toc167107124)

[3.2. Phân tích nghiệp vụ 19](#_Toc167107125)

[3.3. Đặc tả yêu cầu 23](#_Toc167107126)

[3.4. Kết quả đạt được 47](#_Toc167107127)

[3.5. Kiểm thử 51](#_Toc167107128)

[KÊT LUẬN 57](#_Toc167107129)

[1. Những kết quả sau quá trình nghiên cứu 57](#_Toc167107130)

[2. Kiến nghị về những nghiên cứu tiếp theo. 58](#_Toc167107131)

[TÀI LIỆU THAM KHẢO 60](#_Toc167107132)

DANH MỤC HÌNH ẢNH

[Hình 2.1: VueJS Framework 11](#_Toc167107084)

[Hình 2.2: Logo Vuetify 12](#_Toc167107085)

[Hình 2.3 : Mô hình Clean Architecture 16](#_Toc167107086)

[Hình 2.4: Mô hình chi tiết Clean Architecture 16](#_Toc167107087)

[Hình 3.1: Biểu đồ use case tổng quát 23](#_Toc167107088)

[Hình 3.2: Biểu đồ tuần tự của quản lý sản phẩm 27](#_Toc167107089)

[Hình 3.3 : Biểu đồ lớp quản lý sản phẩm 28](#_Toc167107090)

[Hình 3.4 : Biểu đồ trình tự dịch vụ lái thử 30](#_Toc167107091)

[Hình 3.5: Biểu đồ lớp của dịch vu lái thử 31](#_Toc167107092)

[Hình 3.6: Biểu đồ trình tự quản lí cửa hàng 34](#_Toc167107093)

[Hình 3.7 : Biểu đồ lớp quản lí của hàng 35](#_Toc167107094)

[Hình 3.8: Biểu đồ trình tự quản lí tài khoản 38](#_Toc167107095)

[Hình 3.9: Biểu đồ lớp quản lí tài khoản 39](#_Toc167107096)

[Hình 3.10: Biểu đồ trình tự quản lí nhân viên 43](#_Toc167107097)

[Hình 3.11: Biểu đồ lớp quản lí nhân viên 44](#_Toc167107098)

[Hình 3.12 : Biểu đồ Entity Relationship 45](#_Toc167107099)

[Hình 3.13: Hình các bảng trong CSDL 46](#_Toc167107100)

[Hình 3.14:Màn hình chính 47](#_Toc167107101)

[Hình 3.15:Màn danh sách xe 47](#_Toc167107102)

[Hình 3.16:Màn thông tin chi tiết xe 48](#_Toc167107103)

[Hình 3.17: Màn liên hệ 48](#_Toc167107104)

[Hình 3.18: Màn bào dưỡng 49](#_Toc167107105)

[Hình 3.19: Màn trang chủ admin 49](#_Toc167107106)

[Hình 3.20 :Màn quản lí hãng xe 50](#_Toc167107107)

[Hình 3.21 : Màn quản lí cửa hàng 50](#_Toc167107108)

[Hình 3.22: Màn quản lí sản phẩm 51](#_Toc167107109)

DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu | Ý nghĩa |
|  |  |  |
|  |  |  |
|  |  |  |

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời buổi công nghệ thông tin và truyền thông ngày càng cải tiến và phát triển một cách thần tốc. Mọi ngành nghề đều dần áp dụng công nghệ thông tin để tăng năng suất làm việc, mức độ hiệu quả và tiết kiệm được chi phí so với sử dụng cách thủ công. Cũng chính vì thế, đối với thương mại thì không còn xa lạ gì với cụm từ “Thương mại điện tử” ngày càng phổ biến đến với mọi trường trong thời gian trở lại đây, và đặc biệt nóng và phát triển mạnh mẽ hơn trong thời kì dịch bệnh vẫn chưa có dấu hiệu hạ nhiệt, mọi người vẫn chọn mua hàng trực tuyến để có thể bảo vệ được sức khỏe bản thân và gia đình, và vẫn hơn hết là sự nhanh chóng và tiện lợi của nó. Tuy nhiên, các sàn thương mại điện tử chỉ phục vụ cho các sản phẩm nhỏ và lẻ hoặc các phẩm mà nhu cầu người mua hướng đến nhiều, đối với các hàng hóa đặc biệt như ô tô và xe máy thì chúng thường có giá trị rất cao và sẽ bị giới hạn khu vực mua tại sàn thương mại điện tử. Một điểm trừ có thể kể đến khác là khi khách hàng trước khi quyết định mua sản phẩm nào đó thì họ thường có một tâm lý là muốn tìm hiểu sản phẩm đó ,được sờ tận tay thấy tận mắt để kiểm tra được chất lượng và có hợp với mình không, thì các sàn thương mại điện tử sẽ thường không có được đặc quyền đó. Chính vì thế, nhóm chúng em quyết định xây dựng một trang web chuyên về lĩnh vực buôn bán ô tô trực tuyến, bao gồm cả hàng mới và hàng đã qua sử dụng. Sử dụng các phương thức kinh doanh đặc biệt để khách hàng có thể mua được sản phẩm mình ưng ý mà không mất quá nhiều thời gian dể đi ra của hàng xem sản phẩm và nghe tư vấn từ nhân viên.Tuy nhiên, sản phẩm này của chúng em về chuyên môn vẫn chưa được hoàn thiện và có thể thiếu sót một số thông tin nhất định về một sản phẩm ô tô. Nhưng em sẽ cố gắng hoàn thiện ứng dụng một cách chỉnh chu và chuyên nghiệp nhất có thể.

1. Mục tiêu đề tài

Đề tài : **Xây dựng website bán ô tô** đáp ứng được những mục tiêu :

* Xây dựng giao diện người dùng với một số chức năng cơ bản : cho phép khách hàng xem, tìm kiếm, đặt lịch lái thử, bảo dưỡng xe .
* Xây dựng giao diện trang quản trị với một số chức năng: quản lý nhân viên, quản lý sản phẩm, quản lý hãng xe , quản lí tài khoản ,… .
* Phân tích thiết kế Database.

1. Đối tượng nghiên cứu

* Website bán ô tô
* Framework VueJS
* ASP.Net Core
* CSDL : MariaDB

1. Tính cấp thiết của đề tài

Với sự bùng nổ mạnh của công nghệ thông tin và sự phát triển của mạng Internet nên việc trao đổi thông tin trở nên dễ dàng hơn bao giờ hết.

Việc phát triển của các sàn thương mại điện tử. Người dùng mạng xã hội ngày càng phát triển. Để đáp ứng nhu cầu giải trí, mua sắm trực tuyến, việc ra đời các website bán hàng ngày cảng phổ biến. Các website ngày càng phát triển đáp ứng cả số lượng và chất lượng của các website này.

Để xây đáp ứng yêu cầu của học phần đề ra, cũng như thấy được sự quan trọng và niềm đam mê thiết kế ra các trang web em đã chọn đề tài “Xây dựng website bán ô tô.”.

## CƠ SỞ LÝ THUYẾT

### Các khái niệm cơ bản

##### Website là gì ?

Web là tên thường gọi của World Wide Web (mạng toàn cầu), một tập hợp con của Internet bao gồm các trang có thể được truy cập bằng trình duyệt Web.

Các trang web được định dạng bằng ngôn ngữ gọi là Ngôn ngữ đánh dấu siêu văn bản (Hypertext Markup Language - HTML). Ngôn ngữ này cho phép người dùng nhấp qua các trang trên Web thông qua các liên kết. Web sử dụng giao thức HTTP để truyền dữ liệu và chia sẻ thông tin. Các trình duyệt như Internet Explorer, Google Chrome hoặc Mozilla Firefox hoạt động như một công cụ để người dùng có thể truy cập các tài liệu Web hoặc các trang Web được kết nối thông qua các liên kết.

Web chỉ là một trong những cách chia sẻ thông tin qua Internet bên cạnh những thứ khác bao gồm email, nhắn tin tức thời và Giao thức truyền tệp (FTP).

Hiểu một cách ngắn gọn thì Web là mạng.

Còn Site là địa điểm. Ví dụ worksite có nghĩa là nơi làm việc, chỉ một địa điểm, địa chỉ cụ thể.

Như vậy Website = Web + Site, tức một địa chỉ cụ thể trong mạng toàn cầu hay còn được gọi là trang mạng.

Điều đó có nghĩa là website phải đảm bảo được các yếu tố: Nằm trong mạng toàn cầu, có thể truy cập bằng các trình duyệt web, sử dụng giao thức HTTP để truyền dữ liệu và chia sẻ thông tin và quan trọng là phải có một địa chỉ cụ thể.

Website thường chứa các nội dung văn bản, hình ảnh, video và rất nhiều định dạng nội dung khác, được lưu trữ trên máy chủ.

Cấu tạo và hoạt động của website :

* Website thường chứa nhiều webpage hay còn thường được gọi là trang con. Tất cả được lưu trữ dưới định dạng html hoặc xhtml (Extensible HyperText Markup Language - mở rộng của html). Chúng sẽ được lưu trên các máy chủ (web server).
* Khi người dùng muốn truy cập các thông tin từ website cần sử dụng các trình duyệt web để truy cập vào địa chỉ của website, đọc các file lưu trữ dưới định dạng html hoặc xhtml và hiển thị dưới dạng trực quan để dễ dàng tiếp nhận nội dung, thao tác.

Một website muốn hoạt động cần có các thành phần:

* Source Code (mã nguồn): Để các nội dung có thể hiển thị, thao tác, tương tác, tự động tối ưu trên từng loạt thiết bị…
* Web hosting (Lưu trữ web): Nếu website là một ngôi nhà thì hosting chính là miếng đất để xây ngôi nhà đó. Nó lưu trữ Source Code, hình ảnh, video, nội dung… của website.
* Tên miền (domain): Khi bạn đã có miếng đất và ngôi nhà, bạn cần có địa chỉ để những người khác có thể tìm đến và truy cập. Vì vậy bạn cần có một tên miền (domain). Mỗi tên miền là duy nhất trên toàn thế giới để đảo bảo mọi người đều có thể đến chính xác nhà của bạn.

##### Website thương mại điện tử

Thương mại điện tử (E-commerce) là hình thức kinh doanh mua bán sản phẩm hoặc dịch vụ thông qua các hệ thống điện tử như Internet và các mạng máy tính. Thương mại điện tử bao gồm một loạt các hoạt động như mua sắm trực tuyến, ngân hàng trực tuyến, vé điện tử, và nhiều hình thức giao dịch khác thông qua các nền tảng kỹ thuật số.

Lợi ích của Thương mại điện tử :

* Tiện lợi: Người tiêu dùng có thể mua sắm bất kỳ lúc nào và ở bất kỳ đâu chỉ cần có kết nối internet.
* Tiếp cận thị trường rộng lớn: Doanh nghiệp có thể tiếp cận với khách hàng toàn cầu mà không bị giới hạn bởi địa lý.
* Chi phí thấp hơn: Việc vận hành một cửa hàng trực tuyến thường tốn ít chi phí hơn so với một cửa hàng truyền thống, giúp tiết kiệm chi phí thuê mặt bằng, nhân viên, và các chi phí khác.
* Cá nhân hóa trải nghiệm mua sắm: Các nền tảng thương mại điện tử có thể thu thập dữ liệu khách hàng để cung cấp các gợi ý mua sắm phù hợp, nâng cao trải nghiệm khách hàng.
* Dễ dàng quản lý và theo dõi: Các hệ thống quản lý đơn hàng và khách hàng tự động giúp doanh nghiệp dễ dàng theo dõi và phân tích các hoạt động kinh doanh.

Khó khăn của Thương mại điện tử :

* Bảo mật thông tin: Việc bảo vệ thông tin cá nhân và tài chính của khách hàng là một thách thức lớn do nguy cơ bị tấn công mạng và lừa đảo trực tuyến.
* Tin cậy và uy tín: Người tiêu dùng có thể lo ngại về chất lượng sản phẩm và dịch vụ, nhất là khi mua hàng từ các nhà cung cấp ít tên tuổi hoặc không có uy tín.
* Cạnh tranh cao: Thương mại điện tử là một lĩnh vực có sự cạnh tranh rất khốc liệt, đòi hỏi các doanh nghiệp phải không ngừng đổi mới và nâng cao chất lượng dịch vụ.
* Vận chuyển và giao nhận: Đảm bảo hàng hóa được giao đúng thời gian và trong tình trạng tốt là một thách thức, đặc biệt là với các sản phẩm dễ vỡ hoặc có giá trị cao.
* Pháp lý và quy định: Doanh nghiệp phải tuân thủ các quy định pháp lý của từng quốc gia về thương mại điện tử, thuế, quyền riêng tư, và các vấn đề khác, điều này có thể phức tạp và tốn kém.

**Kết luận chương**

Sau khi tìm hiểu về các khái niệm website và website thương mại, ta hiểu rõ hơn về cách thức hoạt động và tầm quan trọng của chúng trong bối cảnh kinh doanh hiện đại. Website, với vai trò là nền tảng cung cấp thông tin và dịch vụ, đã trở thành công cụ không thể thiếu trong việc kết nối doanh nghiệp với khách hàng. Đặc biệt, website thương mại, với các chức năng hỗ trợ mua bán trực tuyến, không chỉ mở rộng phạm vi tiếp cận khách hàng mà còn tối ưu hóa quy trình kinh doanh, giúp tiết kiệm chi phí và nâng cao hiệu quả hoạt động. Đồng thời, chúng ta cũng nhận thức được những thách thức đáng kể mà các doanh nghiệp phải đối mặt, như đảm bảo an ninh mạng, duy trì uy tín, và cạnh tranh gay gắt trong môi trường số hóa. Qua đó, có thể thấy rằng sự kết hợp giữa hiểu biết về công nghệ và chiến lược kinh doanh là chìa khóa để thành công trong thời đại thương mại điện tử phát triển mạnh mẽ.

## CÔNG NGHỆ SỬ DỤNG ASP.NET VÀ VUEJS

### Công nghệ sử dụng

#### Xây dựng giao diện người dùng bằng VueJS

##### FrameWork VueJS

 VueJS là một framework mã nguồn mở của JavaScript được sử dụng để phát triển các giao diện web tương tác. Nó là một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. VueJS tập trung vào view layer. Nó có thể dễ dàng tích hợp vào các dự án lớn để phát triển front-end mà không gặp bất kỳ sự cố nào.



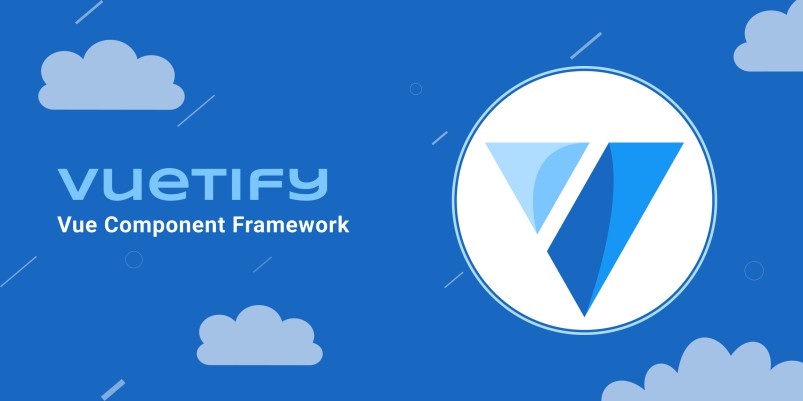
Hình .: VueJS Framework

Trong VueJS, có một số tính năng quan trọng bao gồm:

* Xử lý sự kiện: VueJS sử dụng thuộc tính v-on để lắng nghe các sự kiện từ các phần tử DOM.
* Component: Đây là một tính năng quan trọng của VueJS giúp tạo ra các phần tử tùy chỉnh và có thể tái sử dụng trong HTML.
* Hoạt ảnh/chuyển tiếp: VueJS cung cấp nhiều cách khác nhau để áp dụng chuyển đổi sang các phần tử HTML khi chúng được thêm, cập nhật hoặc xóa khỏi DOM.
* Thuộc tính được tính toán: Đây là một tính năng quan trọng giúp lắng nghe những thay đổi được thực hiện đối với các thành phần giao diện người dùng và thực hiện các tính toán cần thiết.
* Mẫu: VueJS cung cấp các mẫu dựa trên HTML liên kết DOM với dữ liệu đối tượng Vue để hiển thị dữ liệu và tương tác với người dùng.
* Chỉ thị: VueJS có các chỉ thị tích hợp như v-if, v-else, v-show, v-on, v-bind và v-model, được sử dụng để thực hiện các hành động khác nhau trên giao diện người dùng. Chúng giúp thực hiện các tác vụ như điều kiện hiển thị, liên kết dữ liệu, xử lý sự kiện và tạo các phần tử tùy chỉnh.

##### Vuetify framework

Vuetify là một Vue UI framework mà các thành phần của nó dựa trên [Material](https://material.io/design) [Design](https://material.io/design), một ngôn ngữ thiết kế phổ biến do Google phát triển. Nó bao gồm các nguyên tắc giao diện người dùng cho thẻ, hình dạng, tương tác, hiệu ứng chiều sâu như ánh sáng và bóng, v.v.



Hình .: Logo Vuetify

Vuetify giúp xây dựng các trang web và ứng dụng có giao diện chuyên nghiệp mà không cần bất kỳ kỹ năng thiết kế nào. Sử dụng các thành phần được tạo sẵn, sẽ nhanh chóng xây dựng các trang web phù hợp với chất lượng thiết kế của các sản phẩm của Google như Google Analytics và Gmail.

Tại sao nên chọn Vuetify :

* Vuetify là mã nguồn mở có sẵn miễn phí theo [MIT cấp phép](http://opensource.org/licenses/MIT). Ngoài ra, mã nguồn của Vuetify có sẵn trên GitHub, cho phép nhà phát triển sửa đổi và đóng góp vào sự phát triển của nó nếu chọn làm như vậy.
* Tính linh hoạt: Mọi thành phần trong Vuetify đều được làm thủ công dưới vỏ bọc của Google [Material Design specification](https://material.io/) và à đi kèm với hàng trăm tùy chọn tùy chỉnh phù hợp với mọi phong cách hoặc thiết kế.
* Vuetify có một hệ sinh thái rộng lớn gồm các công cụ hỗ trợ làm phong phú thêm trải nghiệm phát triển, từ tạo dự án đến thiết kế bộ giao diện người dùng.

#### Xây dựng API bằng ASP.NET core

##### Framework ASP.NET core

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.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

Ưu điểm:

+ Đa nền tảng: ASP.NET Core hỗ trợ chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux, giúp cho việc triển khai ứng dụng linh hoạt hơn.

+ Hiệu suất cao: ASP.NET Core được tối ưu hóa để cung cấp hiệu suất tốt hơn so với các phiên bản trước đó của ASP.NET.

+ Modular và linh hoạt: Nền tảng này được thiết kế theo kiểu modular, cho phép lập trình viên chỉ sử dụng những thành phần cần thiết cho ứng dụng của mình, giúp giảm kích thước ứng dụng và tối ưu hóa bộ nhớ.

+ Hỗ trợ Dependency Injection tích hợp: ASP.NET Core tích hợp sẵn hệ thống Dependency Injection, giúp quản lý và triển khai các dịch vụ và thành phần trong ứng dụng một cách dễ dàng và linh hoạt.

+ Hỗ trợ các công nghệ hiện đại: ASP.NET Core hỗ trợ các công nghệ mới như Docker, Kubernetes, và các công nghệ cloud computing, giúp dễ dàng tích hợp và triển khai ứng dụng trên các môi trường điện toán đám mây.

Nhược điểm:

+ Thư viện hỗ trợ: Mặc dù ASP.NET Core đã có sự phát triển mạnh mẽ, nhưng vẫn có một số thư viện bên thứ ba không có sẵn hoặc hỗ trợ kém so với các nền tảng khác như Node.js hoặc Python.

+ Học đòi: Do ASP.NET Core có một số khái niệm mới và cách tiếp cận khác so với các phiên bản trước đó của ASP.NET, việc học và làm quen với nền tảng này có thể mất thời gian đối với những người mới bắt đầu.

+ Hạn chế trong môi trường nhất quán: Mặc dù ASP.NET Core đã cải thiện khả năng chạy trên các nền tảng khác nhau, nhưng vẫn có một số hạn chế về môi trường phát triển và triển khai so với các công nghệ mã nguồn mở phổ biến khác.

+ Cộng đồng: Mặc dù cộng đồng ASP.NET Core ngày càng phát triển, nhưng vẫn có thể không lớn bằng một số cộng đồng mã nguồn mở khác như PHP hoặc Python, dẫn đến việc tìm kiếm hỗ trợ và tài liệu có thể gặp khó khăn hơn.

##### Xây dựng theo chuẩn Restful API

Restful **API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. Restful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. **RESTful** không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API.**

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

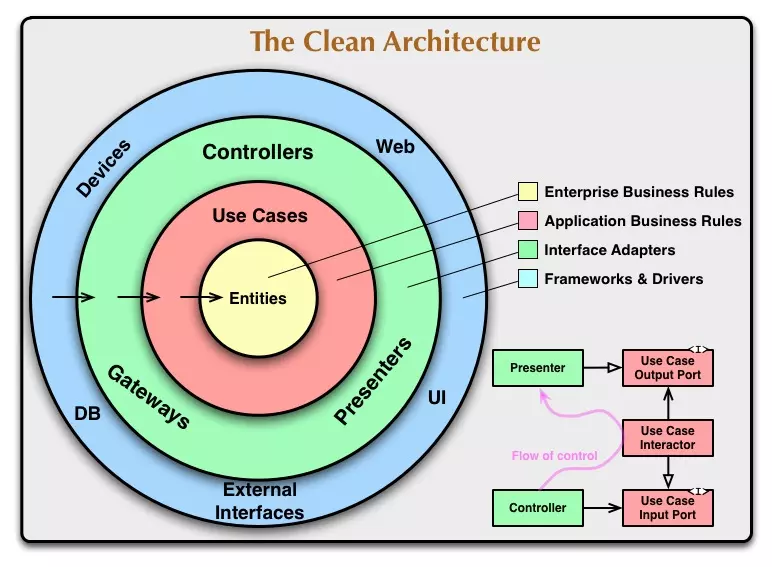
##### Mô hình xây dựng : Mô hình đa tầng

Kiến Trúc đa tầng (Clean Architecture) là một mô hình thiết kế phần mềm được giới thiệu bởi Robert C. Martin (còn được gọi là Uncle Bob). Mục tiêu của kiến trúc này là tạo ra các hệ thống dễ bảo trì, dễ mở rộng và dễ kiểm tra, đồng thời tách biệt rõ ràng giữa các mối quan tâm khác nhau của ứng dụng.

Nguyên lý cơ bản :

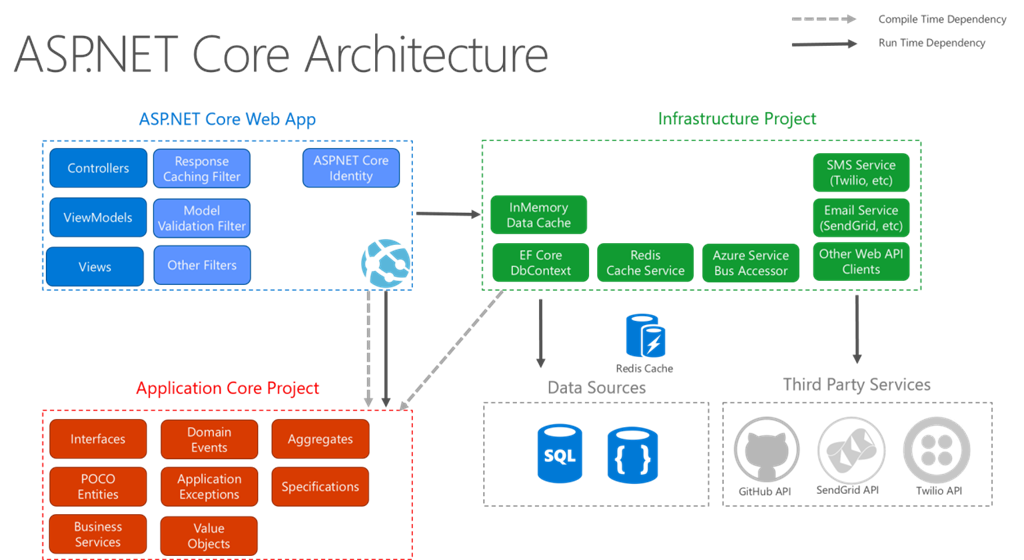
Kiến Trúc đa tầng dựa trên một số nguyên lý cơ bản, trong đó nổi bật nhất là Nguyên Tắc Đảo Ngược Phụ Thuộc (Dependency Inversion Principle - DIP) và Thiết Kế Hướng Miền (Domain-Driven Design - DDD). Các nguyên lý này nhấn mạnh rằng:

* Các mô-đun cấp cao không nên phụ thuộc vào các mô-đun cấp thấp. Cả hai nên phụ thuộc vào các giao diện trừu tượng.
* Các chi tiết cụ thể (cơ sở dữ liệu, giao diện người dùng, thư viện) nên phụ thuộc vào các trừu tượng chứ không phải ngược lại



Hình . : Mô hình Clean Architecture

Các phụ thuộc trong Clean Architecture luôn hướng về phía trong. Điều này có nghĩa là các lớp ngoài cùng phụ thuộc vào các lớp bên trong chứ không phải ngược lại. Các giao diện (interfaces) được định nghĩa trong các lớp bên trong và được triển khai trong các lớp bên ngoài.



Hình .: Mô hình chi tiết Clean Architecture

### Công cụ

* Visual Studio Code : phục vụ code giao diện người dùng
* Visual Studio: phục vụ code API
* MariaDB : cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* Postman: sử dụng để thử nghiệm API.
* Chrome DevTools: gỡ lỗi khi phát triển web.
* Diagram : thiết kế cơ sở dữ liệu.
* Rational Rose: thiết kế hệ thống.

**Kết luận chương**

Là kết quả của quá trình tìm hiểu, phân tích và tổng hợp các nguồn tài liệu và nghiên cứu liên quan đến đề tài "Xây dựng website bán xe ô tô sử dụng Vue.js và ASP.NET". Trong chương này, chúng tôi đã trình bày các khái niệm và nguyên lý cơ bản về hai công nghệ chính được sử dụng trong dự án của chúng tôi.

Đầu tiên, chúng tôi đã giới thiệu về Vue.js - một framework JavaScript phổ biến được sử dụng cho việc phát triển giao diện người dùng tương tác. Chúng tôi đã trình bày về các đặc điểm và ưu điểm của Vue.js như tính linh hoạt, dễ dàng học và sử dụng, cũng như cách nó hoạt động trong việc tạo ra các component, quản lý trạng thái ứng dụng và xử lý sự kiện.

Tiếp theo, chúng tôi đã tập trung vào ASP.NET - một framework phổ biến cho việc phát triển ứng dụng web và dịch vụ web. Chúng tôi đã giới thiệu về kiến trúc và các thành phần chính của ASP.NET, cũng như cách nó được sử dụng trong việc xây dựng backend cho trang web của chúng tôi. Chúng tôi đã tóm tắt về việc xử lý yêu cầu HTTP, truy cập cơ sở dữ liệu, xác thực và quản lý phiên trong ASP.NET.

Từ việc hiểu biết sâu rộng về Vue.js và ASP.NET qua các nội dung được trình bày trong chương cơ sở lý thuyết, chúng tôi sẽ tiếp tục triển khai và phát triển ứng dụng website bán xe ô tô của mình, hy vọng rằng sẽ đem lại trải nghiệm tuyệt vời cho người dùng cuối.

## XÂY DỰNG WEBSITE BÁN Ô TÔ

### Nghiên cứu, khảo sát thị trường

#### Xác định mục tiêu nghiên cứu:

* Hiểu rõ về hành vi mua sắm và sở thích của khách hàng khi mua xe ô tô trực tuyến.
* Xác định các yếu tố quyết định khi khách hàng quyết định mua xe ô tô qua trang web.
* Tìm hiểu về các dịch vụ và tính năng mà khách hàng mong đợi từ một trang web bán xe ô tô.

#### Thu thập dữ liệu cơ bản:

* Thu thập thông tin về các đại lý xe ô tô và trang web bán xe ô tô trực tuyến đang hoạt động trên thị trường.
* Xác định số lượng và loại xe ô tô được bán ra trên thị trường.
* Phân tích xu hướng giá cả và các chương trình khuyến mãi hiện đang diễn ra.

#### Nghiên cứu về người tiêu dùng:

* Tiến hành cuộc khảo sát hoặc phỏng vấn với một mẫu ngẫu nhiên các người tiêu dùng để hiểu rõ hơn về hành vi mua sắm và yêu cầu của họ khi mua xe ô tô trực tuyến.
* Phân tích dữ liệu từ các trang web xã hội hoặc diễn đàn ô tô để tìm hiểu về những ý kiến và phản hồi của người dùng về các trang web bán xe ô tô hiện tại.

#### Đánh giá đối thủ:

* Phân tích các trang web bán xe ô tô hàng đầu trên thị trường để hiểu về các dịch vụ và tính năng mà họ cung cấp.
* Đánh giá điểm mạnh và điểm yếu của các đối thủ trong lĩnh vực bán xe ô tô trực tuyến.

#### Phân tích xu hướng và tiềm năng thị trường:

* Tìm hiểu về các xu hướng mới trong lĩnh vực mua bán xe ô tô trực tuyến, như sự phát triển của thị trường xe điện, hoặc xu hướng mua sắm trực tuyến tăng cao do ảnh hưởng của dịch COVID-19.
* Xem xét các chính sách pháp lý và các yếu tố văn hóa xã hội có thể ảnh hưởng đến thị trường bán xe ô tô.

#### Tổng hợp và phân tích dữ liệu:

* Tổng hợp và phân tích dữ liệu thu thập được từ các nguồn khác nhau để đưa ra các kết luận và hướng đi cho chiến lược kinh doanh của bạn.

#### Xây dựng chiến lược:

* Dựa trên những phân tích và kết luận từ nghiên cứu và khảo sát, xây dựng một chiến lược tiếp thị và kinh doanh hiệu quả để thu hút khách hàng và tạo ra lợi ích cho doanh nghiệp của bạn.

### Phân tích nghiệp vụ

#### Xác định yêu cầu hệ thống

##### Quản lý sản phẩm:

* Hiển thị thông tin chi tiết về các loại xe ô tô bao gồm hãng xe, mẫu xe, năm sản xuất, màu sắc, giá cả, và các tính năng kỹ thuật.
* Cho phép quản trị viên thêm, sửa đổi, và xóa sản phẩm từ hệ thống quản trị.

##### Tìm kiếm và lọc sản phẩm:

* Cung cấp công cụ tìm kiếm và lọc sản phẩm dựa trên các tiêu chí như hãng xe, mẫu xe, năm sản xuất, màu sắc, và giá cả.
* Hỗ trợ tìm kiếm theo từ khoá và tìm kiếm nâng cao.

##### Giao diện người dùng thân thiện:

* Thiết kế giao diện người dùng đẹp mắt, dễ sử dụng và thân thiện với người dùng.
* Tối ưu hóa trải nghiệm người dùng trên cả máy tính và thiết bị di động.
* Tích hợp dịch vụ bên thứ ba:
* Tích hợp các dịch vụ như Google Maps để hiển thị thông tin vị trí và hướng dẫn đường đến các đại lý hoặc showroom.
* Tích hợp công cụ xem trước xe ô tô, bảng giá, và tính năng so sánh.

##### Bảo mật và quản lý dữ liệu:

* Bảo vệ thông tin cá nhân của người dùng và giao dịch thanh toán bằng cách sử dụng mã hóa và các biện pháp bảo mật khác.
* Sao lưu và phục hồi dữ liệu định kỳ để đảm bảo tính toàn vẹn và sẵn sàng của hệ thống.

##### Quản lý Nhân viên:

* Tạo một hệ thống quản lý nhân viên để lưu trữ thông tin cá nhân của nhân viên như tên, địa chỉ, thông tin liên lạc, v.v.
* Cho phép quản trị viên thêm, sửa đổi và xóa nhân viên từ hệ thống.
* Phân quyền truy cập để quản trị viên có thể quản lý thông tin nhân viên.

##### Quản lý Cửa hàng:

* Lưu trữ thông tin về các cửa hàng hoặc đại lý bán xe ô tô như tên, địa chỉ, số điện thoại, v.v.
* Hiển thị thông tin vị trí của các cửa hàng trên bản đồ để khách hàng dễ dàng tìm kiếm.
* Cho phép quản trị viên thêm, sửa đổi và xóa thông tin về cửa hàng từ hệ thống.

##### Quản lý Hình Ảnh:

* Lưu trữ hình ảnh của các loại xe ô tô và cửa hàng.
* Tạo một thư viện hình ảnh để quản lý và tìm kiếm hình ảnh một cách dễ dàng.
* Cho phép người dùng xem hình ảnh chi tiết của các loại xe ô tô.

##### Quản lý Liên Hệ:

* Lưu trữ thông tin liên hệ của khách hàng như tên, địa chỉ email, số điện thoại, v.v.
* Tạo một biểu mẫu liên hệ để người dùng có thể gửi câu hỏi hoặc yêu cầu thông tin.

##### Quản lý Hóa Đơn:

* Lưu trữ thông tin về các đơn hàng của khách hàng, bao gồm thông tin về sản phẩm đã mua, giá cả, v.v.
* Cho phép quản trị viên xem và quản lý các đơn hàng từ hệ thống quản trị.

##### Quản lý Dịch vụ:

* Lưu trữ thông tin về các dịch vụ của khách hàng , bao gồm : dịch vụ về đặt lịch lái thử , bảo dưỡng , …
* Cho phép quản trị viên xem và quản lý các dịch từ hệ thống quản trị.

##### Quản lý hãng xe :

* Lưu trữ thông tin về các hãng xe ô tô như tên, v.v.
* Hiển thị thông tin sản phẩm theo hãng xe của cửa hàng để khách hàng dễ dàng tìm kiếm.
* Cho phép quản trị viên thêm, sửa đổi thông tin về hãng xe từ hệ thống.

### Đặc tả yêu cầu

#### Biểu đồ Use Case tổng quát



Hình .: Biểu đồ use case tổng quát

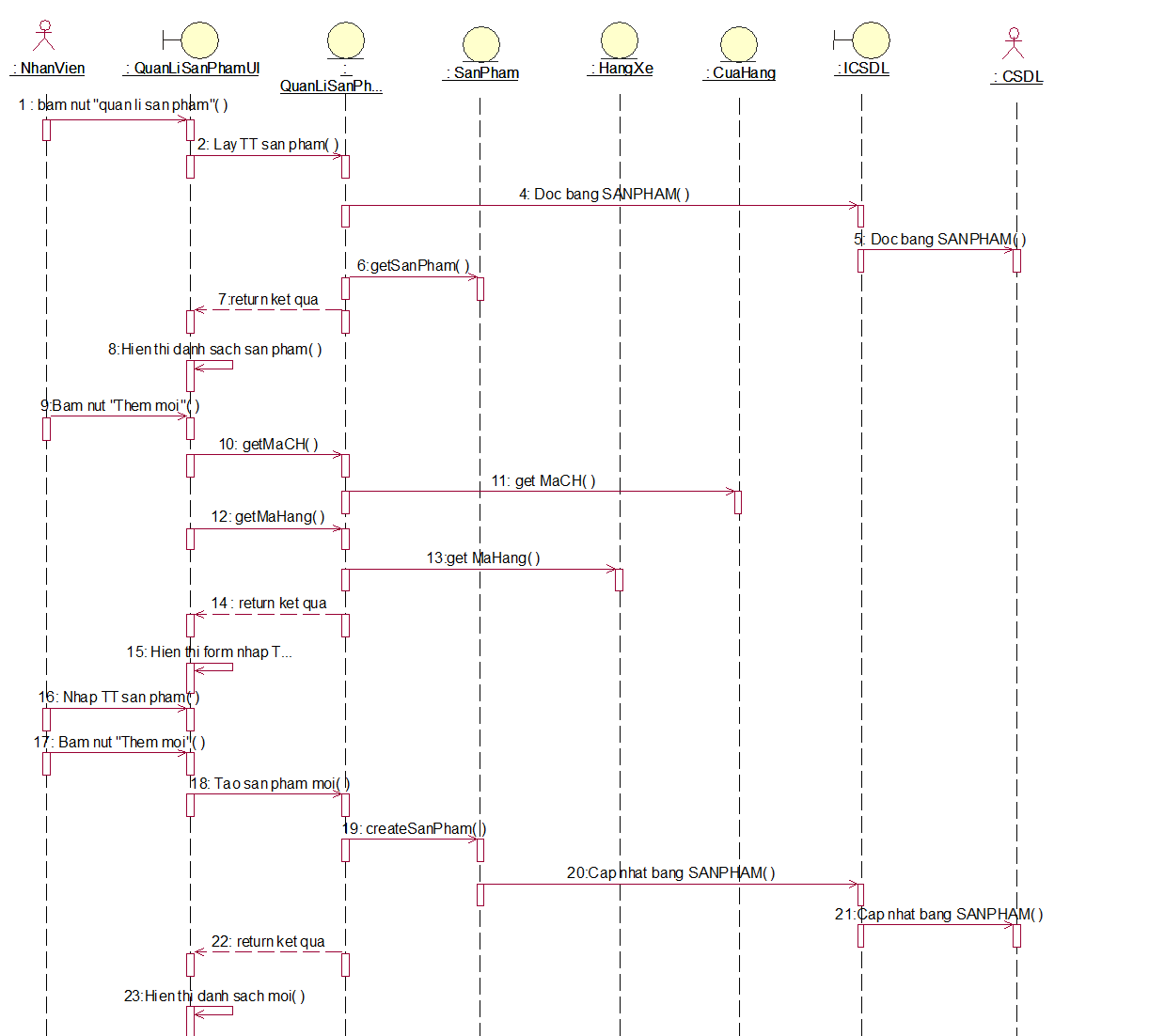
#### Mô tả chi tiết Use Case

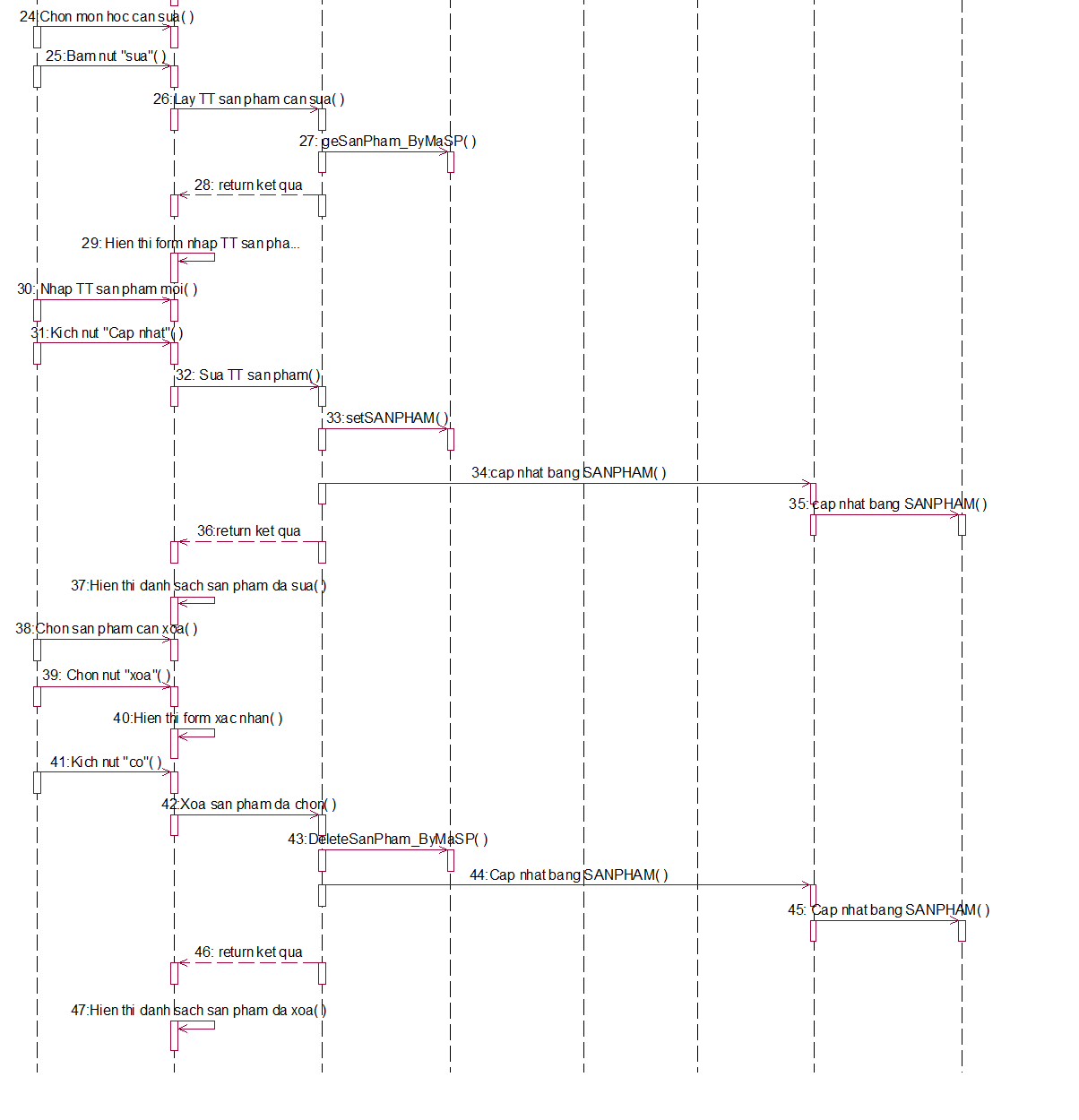
##### Use case : Quản lí sản phẩm

* Mô tả use case :

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lí sản phẩm |
| **Mô tả** | Use case này cho phép nhân viên của cửa hàng thêm , cập nhật , xóa sản phẩm trong cửa hàng |
| **Người thực hiện** | Nhân viên của cửa hàng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case này bắt đầu khi người dùng kích vào nút quản lí sản phẩm trên menu trang chủ và hệ thống hiển thị bảng sản phẩm trong bảng SANPHAM , HangXe , CuaHang gồm : TenSP , TenCH , TenHang , MoTa, MauSac , NamSX, NamDKy , SoDam , DongCo ,SoGhe , LoaiHopSo, DungTich , TinhTrang.   * Thêm sản phẩm : * Khi nhân viên nhấn chọn vào nút “Thêm mới”. Hệ thống sẽ hiển thị form nhập thông tin sản phẩm mới . * Nhân viên sẽ nhập vào các thông tin chi tiết của sản phẩm . Sau khi nhập xong , nhân viên sẽ nhấn nút thêm , hệ thống sẽ thêm mới 1 bản ghi vào bảng SANPHAM . Hệ thống sẽ hiển thị bảng sản phẩm lên màn hình * Cập nhật thông tin sản phẩm : * Khi nhân viên nhấn vào nút “ Sửa ” của 1 sản phẩm , trên màn hình sẽ hiện ra form chi tiết thông tin của sản phẩm đó . Nhân viên có thể sửa các trường nhập dữ liệu . Sau khi sửa thông tin xong , nhân viên bấm nút “ Cập nhật” . Hệ thống cập nhật thông tin sản phẩm vào bảng SANPHAM và hiện thị lại danh sách bảng sản phẩm lên màn hình. * Xóa sản phẩm : * Nhân viên bấm vào nút “Xóa “ của 1 sản phẩm cụ thể trên bảng danh sách sản phẩm . Sau đó màn hình hiện lên 1 form xác nhận để xác nhận . Nhân viên bấm vào nút “có “ , hệ thống sẽ xóa sản phẩm đã chọn khỏi bảng SANPHAM và sẽ hiện thị lại danh sách sản phẩm sau khi đã xóa . Usecase kết thúc . |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu nhân viên nhập thông tin không hợp lệ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. Nhân viên có thể nhập lại để tiếp tục hoặc kích vào mục khác để bỏ qua thao tác.  Tại bước 2 khi xóa sản phẩm , nếu mà nhân viên nhấn vút “Hủy bỏ” , hệ thống sẽ giữ nguyên và hiện thị danh sách sản phẩm cũ.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng SANPHAM được cập nhật |

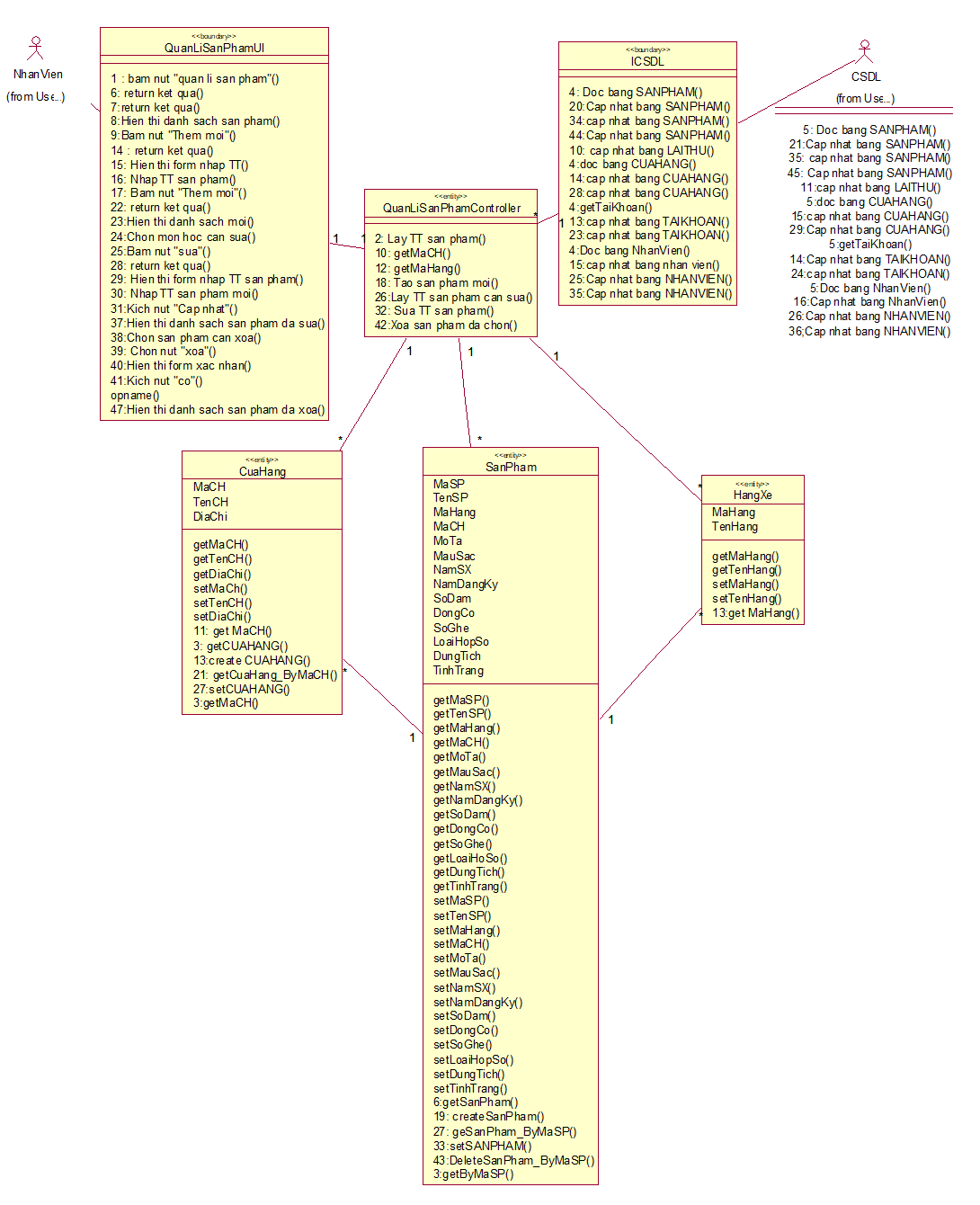
* Biểu đồ trình tự :





Hình .: Biểu đồ tuần tự của quản lý sản phẩm

* Biểu đồ lớp :



Hình . : Biểu đồ lớp quản lý sản phẩm

##### Use case : Dich vụ lái thử

* Mô tả use case :

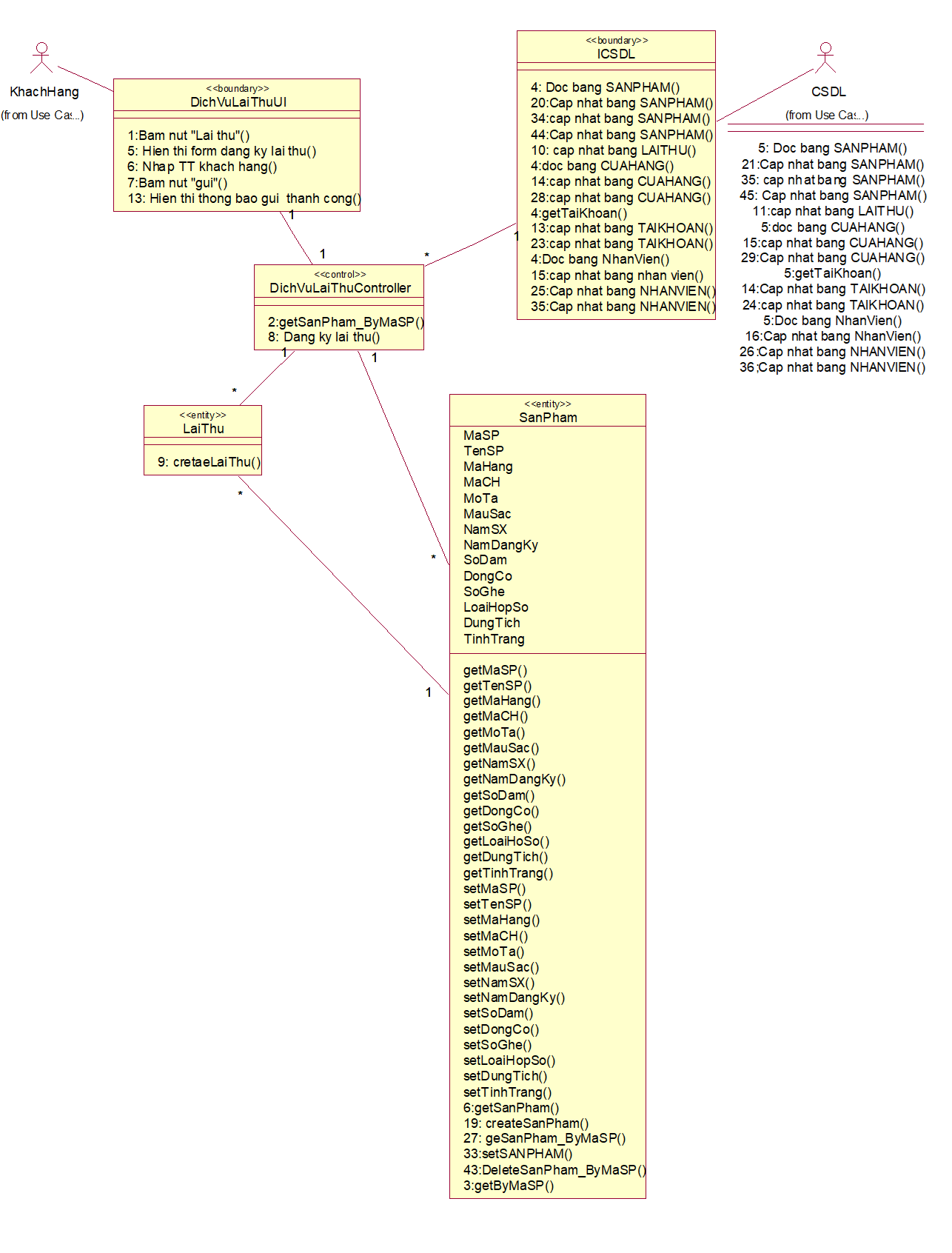
|  |  |
| --- | --- |
| **Tên Use Case** | Dịch vụ lái thử |
| **Mô tả** | Use case này cho phép khách hàng đăng ký lái thử xe của cửa hàng |
| **Người thực hiện** | Khách hàng |
| **Tiền điều kiện** | Không |
| **Luồng cơ bản** | Use case này bắt đầu khi người dùng kích vào nút dịch vụ lái thử trên từng sản phẩm và hệ thống hiển thị ra bảng đăng ký lái thử . Hệ thống sẽ lấy ra thông tin chi tiết xe như : Tên xe , màu sắc , số dặm , năm sx ,.. ở bảng SANPHAM hiện ra ở bảng đăng ký. Khách hàng sẽ điền nốt thông tin cơ bản như : Tên khách hàng , số điện thoại , email , địa chỉ , ngày lái thử ,… Khách hàng sẽ nhấn nút “Gửi” . Use case kết thúc |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu khách hàng nhập thông tin không hợp lệ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. khách hàng có thể nhập lại để tiếp tục hoặc kích vào mục khác để bỏ qua thao tác.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng LaiThu được cập nhật |

* Biểu đồ trình tự :



Hình . : Biểu đồ trình tự dịch vụ lái thử

* Biểu đồ lớp :



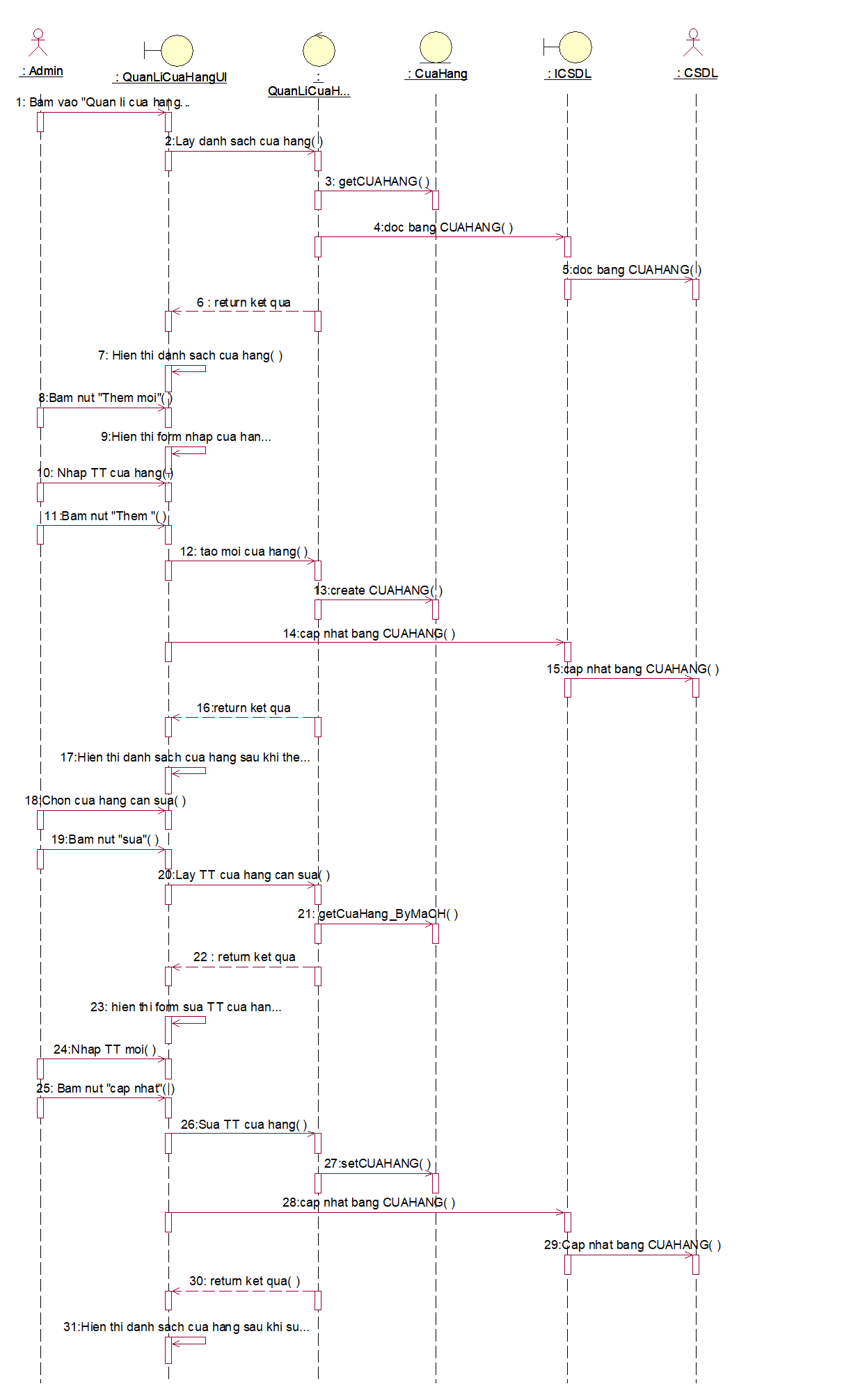
Hình .: Biểu đồ lớp của dịch vu lái thử

##### Use case : Quản lí cửa hàng

##### Mô tả use case :

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lí cửa hàng |
| **Mô tả** | Use case này cho phép người quản trị quản lí cửa hàng , bao gồm : Thêm , sửa cửa hàng. |
| **Người thực hiện** | Người quản trị |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case này bắt đầu khi người quản trị vào nút quản lí cửa hàng trên menu và hệ thống hiển thị ra bảng cửa hàng . Hệ thống sẽ lấy ra thông tin chi tiết xe như : Tên cửa hàng địa chỉ ở bảng. Người quản trị có thể thêm , sửa cửa hàng .   * Thêm cửa hàng : * Khi người quản trị nhấn nút “Thêm mới cửa hàng” . Màn hình sẽ hiển thị ra bảng để người quản trị nhập thông tin cửa hàng. * Sau khi nhập xong , người quản trị bấm nút thêm mới . Hệ thống sẽ thêm mới cửa hàng vào bảng CUAHANG . Màn hình sẽ hiện thị danh sách cửa hàng sau khi đã thêm mới . * Sửa cửa hàng : * Người quản trị chọn cửa hàng cần sửa , sau đó người quản trị bấm nút “Sửa “ . Hệ thống sẽ lấy ra thông tin chi tiết cửa hàng hiện lên bảng sửa . Người quản trị có thể sửa những thông tin của cửa hàng . Sau khi sửa , người quản trị bấm nút “Cập nhật” hệ thống sẽ cập nhật thông tin mới vào bảng cửa hàng và màn hình hiện thị danh sách cửa hàng sau khi đã cập nhật .   . Use case kết thúc |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu người quản trị nhập thông tin không hợp lệ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. người quản trị có thể nhập lại để tiếp tục hoặc kích vào mục khác để bỏ qua thao tác.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng CuaHang được cập nhật |

##### Biểu đồ trình tự:



Hình .: Biểu đồ trình tự quản lí cửa hàng

##### Biểu đồ lớp :



Hình . : Biểu đồ lớp quản lí của hàng

##### Use case : Quản lí tài khoản

* Mô tả use case

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lí tài khoản |
| **Mô tả** | Use case này cho phép người quản trị quản lí tài khoản của nhân viên, bao gồm : Thêm , sửa tài khoản. |
| **Người thực hiện** | Người quản trị |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case này bắt đầu khi người quản trị vào nút quản lí tài khoản trên menu và hệ thống hiển thị ra bảng tài khoản . Hệ thống sẽ lấy ra thông tin chi tiết như : Tên tài khoản . mật khẩu , mã nhân viên , mã nhóm từ bảng TAIKHOAN. Người quản trị có thể thêm , sửa tài khoản .   * Thêm tài khoản: * Khi người quản trị nhấn nút “Thêm mới tài khoản” . Màn hình sẽ hiển thị ra bảng để người quản trị nhập thông tin tài khoản. * Sau khi nhập xong , người quản trị bấm nút thêm mới . Hệ thống sẽ thêm mới tài khoản vào bảng TAIKHOAN . Màn hình sẽ hiện thị danh sách tài khoản sau khi đã thêm mới . * Sửa tài khoản: * Người quản trị chọn tài khoản cần sửa , sau đó người quản trị bấm nút “Sửa “ . Hệ thống sẽ lấy ra thông tin chi tiết tài khoản hiện lên bảng sửa . Người quản trị có thể sửa những thông tin của tài khoản. Sau khi sửa , người quản trị bấm nút “Cập nhật” hệ thống sẽ cập nhật thông tin mới vào bảng tài khoản và màn hình hiện thị danh sách tài khoản sau khi đã cập nhật .   . Use case kết thúc |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu người quản trị nhập thông tin không hợp lệ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. người quản trị có thể nhập lại để tiếp tục hoặc kích vào mục khác để bỏ qua thao tác.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng TAIKHOAN được cập nhật |

* Biểu đồ trình tự :



Hình .: Biểu đồ trình tự quản lí tài khoản

* Biểu đồ lớp :

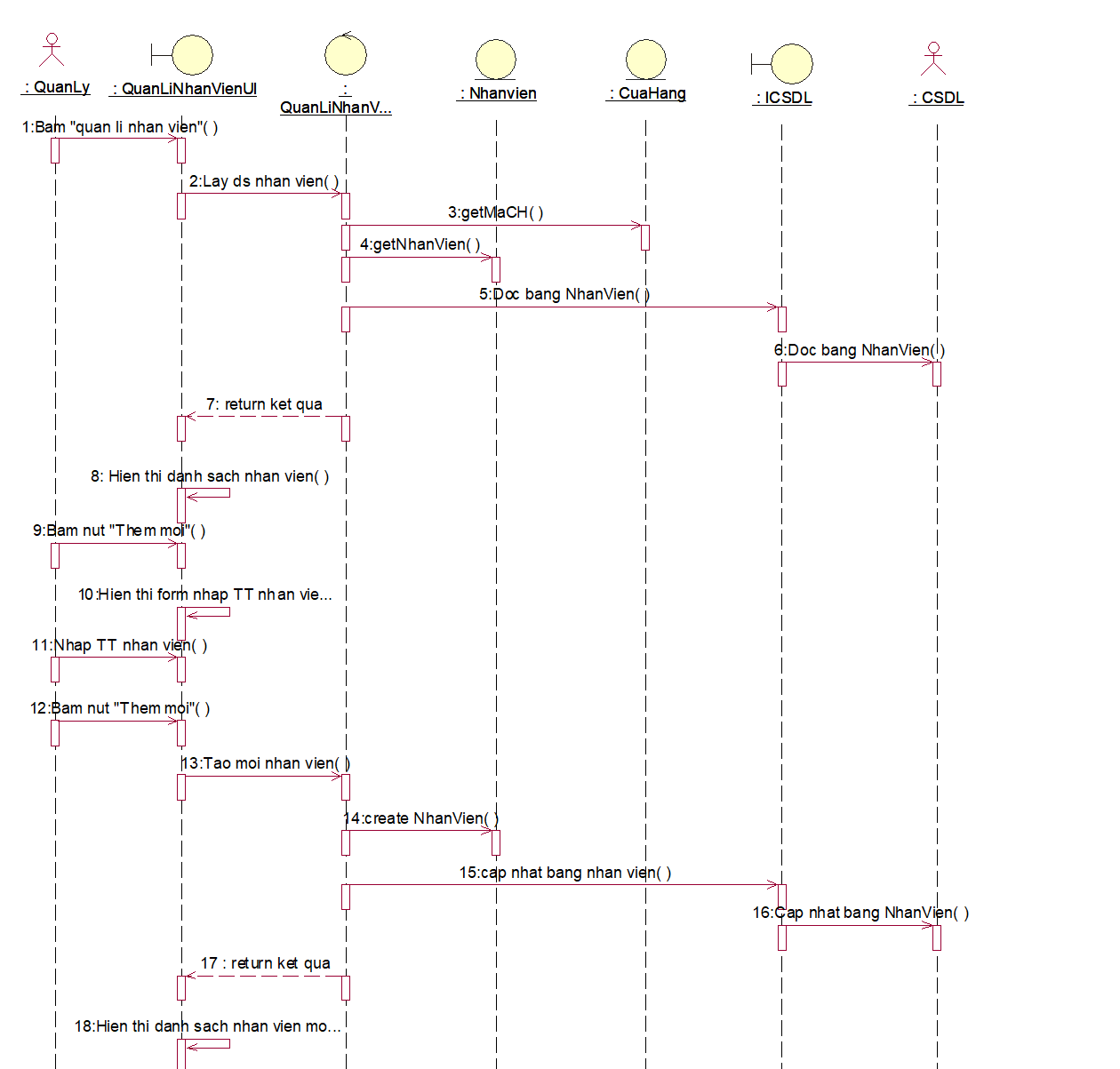


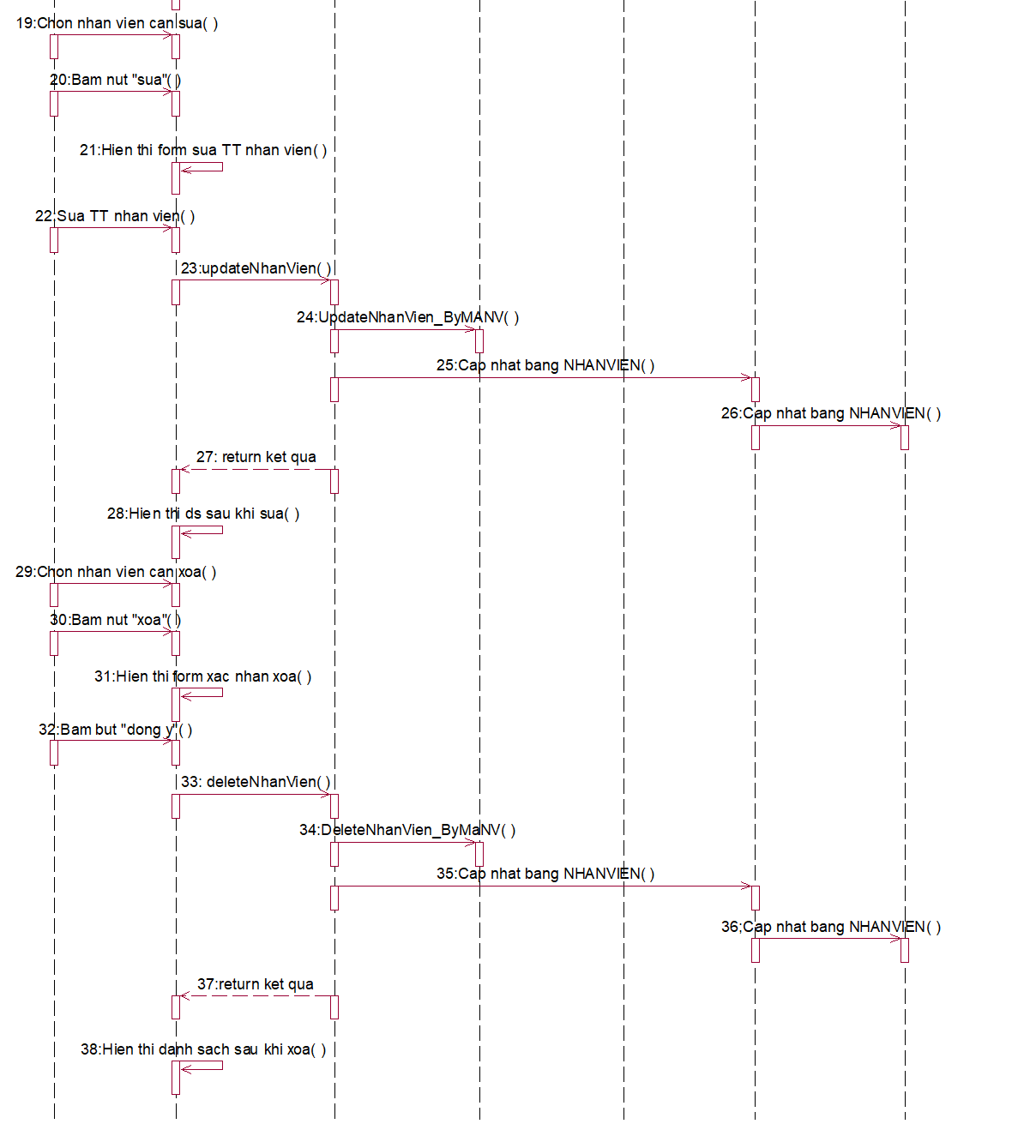
Hình .: Biểu đồ lớp quản lí tài khoản

##### Use case : Quản lí nhân viên

* Mô tả use case :

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lí nhân viên |
| **Mô tả** | Use case này cho phép người quản lí của cửa hàng thêm , cập nhật , xóa nhân viên trong cửa hàng |
| **Người thực hiện** | Quản lí của cửa hàng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case này bắt đầu khi người quản lí kích vào nút quản lí nhân viên trên menu trang chủ và hệ thống hiển thị bảng nhân viên trong bảng NHANVIEN , CuaHang gồm : TenNV , NgaySinh , GioiTinh , DiaChi, SoDT , ChucVu, MaCH .   * Thêm sản phẩm : * Khi quản lí nhấn chọn vào nút “Thêm mới”. Hệ thống sẽ hiển thị form nhập thông tin nhân viên mới . * Quản lí sẽ nhập vào các thông tin chi tiết của nhân viên . Sau khi nhập xong , quản lí sẽ nhấn nút thêm , hệ thống sẽ thêm mới 1 bản ghi vào bảng NHANVIEN . Hệ thống sẽ hiển thị bảng nhân viên lên màn hình * Cập nhật thông tin nhân viên : * Khi quản lí nhấn vào nút “ Sửa ” của 1 nhân viên , trên màn hình sẽ hiện ra form chi tiết thông tin của nhân viên đó . Quản lí có thể sửa các trường nhập dữ liệu . Sau khi sửa thông tin xong , nhân viên bấm nút “ Cập nhật” . Hệ thống cập nhật thông tin nhân viên vào bảng NhanVien và hiện thị lại danh sách bảng nhân viên lên màn hình. * Xóa nhân viên : * Quản lí bấm vào nút “Xóa “ của 1 nhân viên cụ thể trên bảng danh sách nhân viên . Sau đó màn hình hiện lên 1 form xác nhận để xác nhận . Quản lí bấm vào nút “có “ , hệ thống sẽ xóa nhân viên đã chọn khỏi bảng NHANVIEN và sẽ hiện thị lại danh sách sản phẩm sau khi đã xóa . Usecase kết thúc . |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu quản lí nhập thông tin không hợp lệ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. Quản lí có thể nhập lại để tiếp tục hoặc kích vào mục khác để bỏ qua thao tác.  Tại bước 2 khi xóa nhân viên , nếu mà quản lí nhấn vút “Hủy bỏ” , hệ thống sẽ giữ nguyên và hiện thị danh sách nhân viên cũ.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng NHANVIEN được cập nhật |

* Biểu đồ trình tự:
* 



Hình .: Biểu đồ trình tự quản lí nhân viên

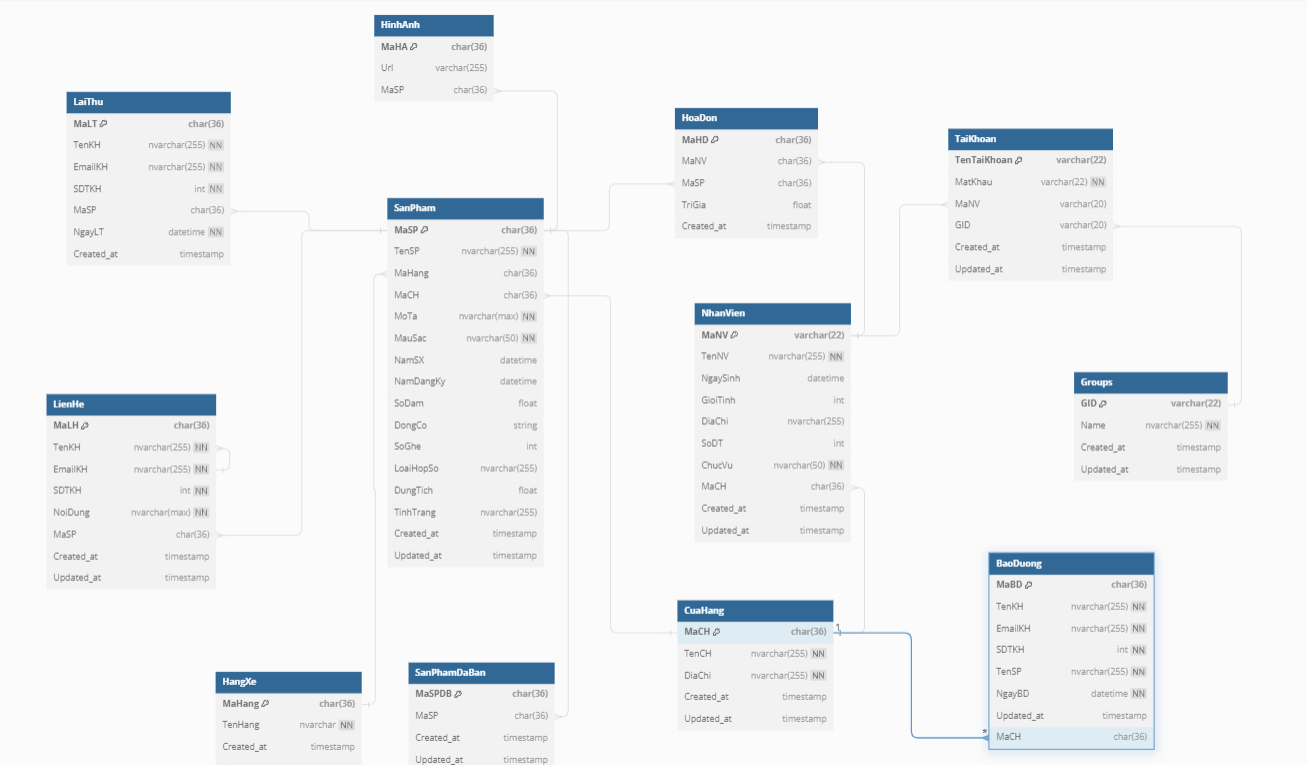
* Biểu đồ lớp :



Hình .: Biểu đồ lớp quản lí nhân viên

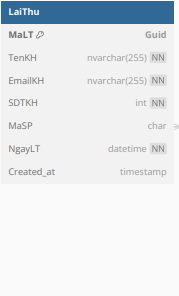
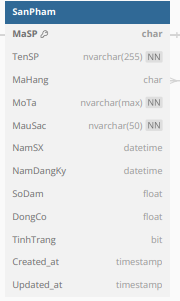
#### Xây dựng cơ sở dữ liệu

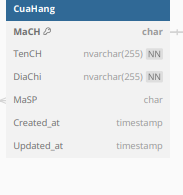
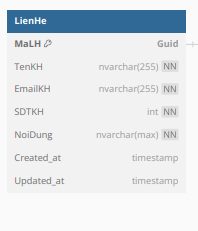
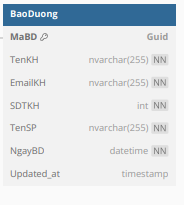
##### Biểu đồ Entity RelationShip

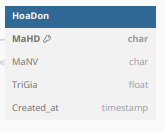
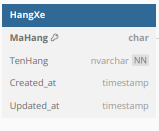
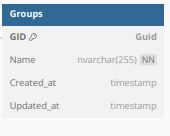


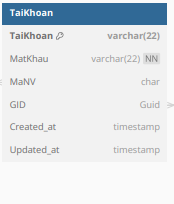
Hình . : Biểu đồ Entity Relationship

##### Các bảng trong CSDL



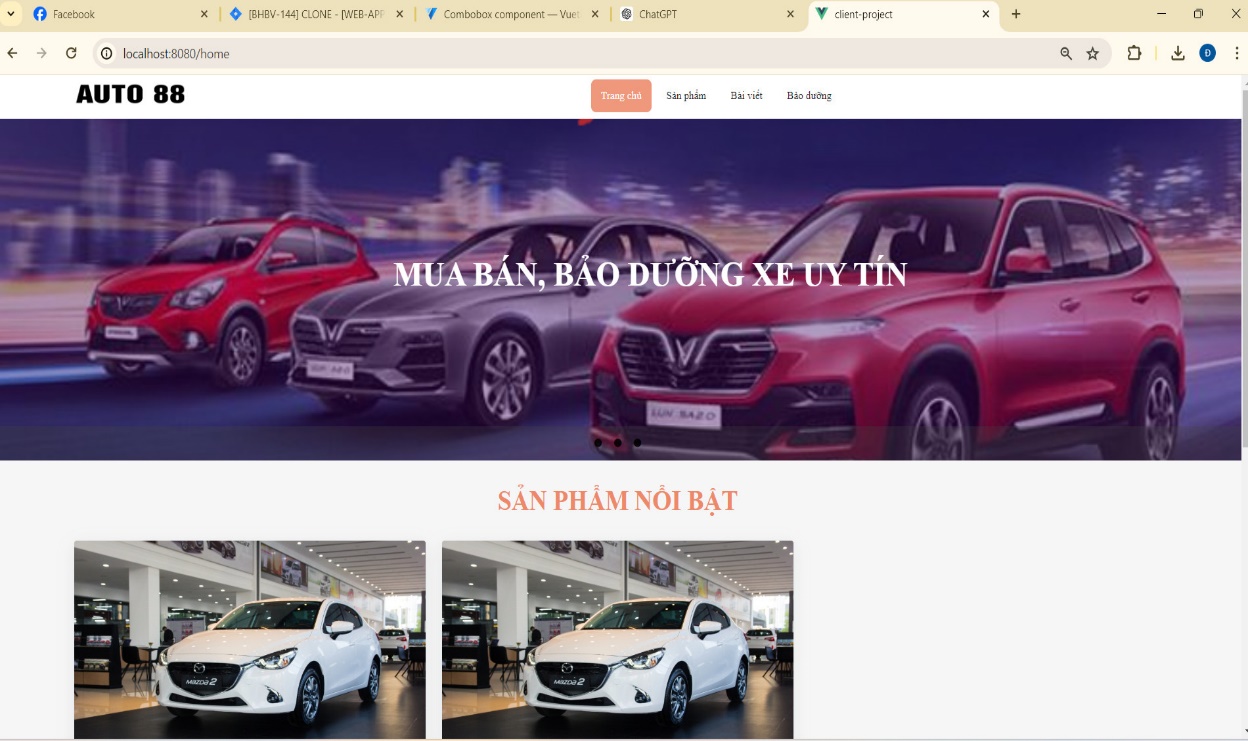




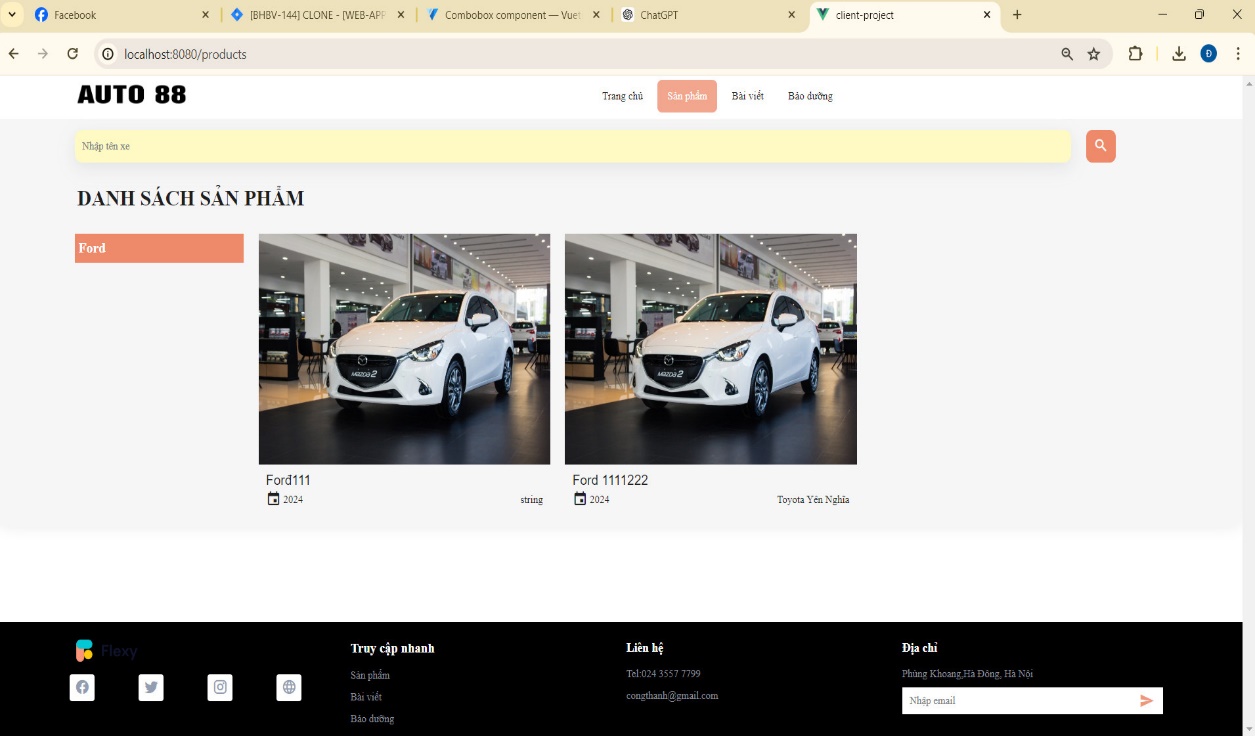


Hình .: Hình các bảng trong CSDL

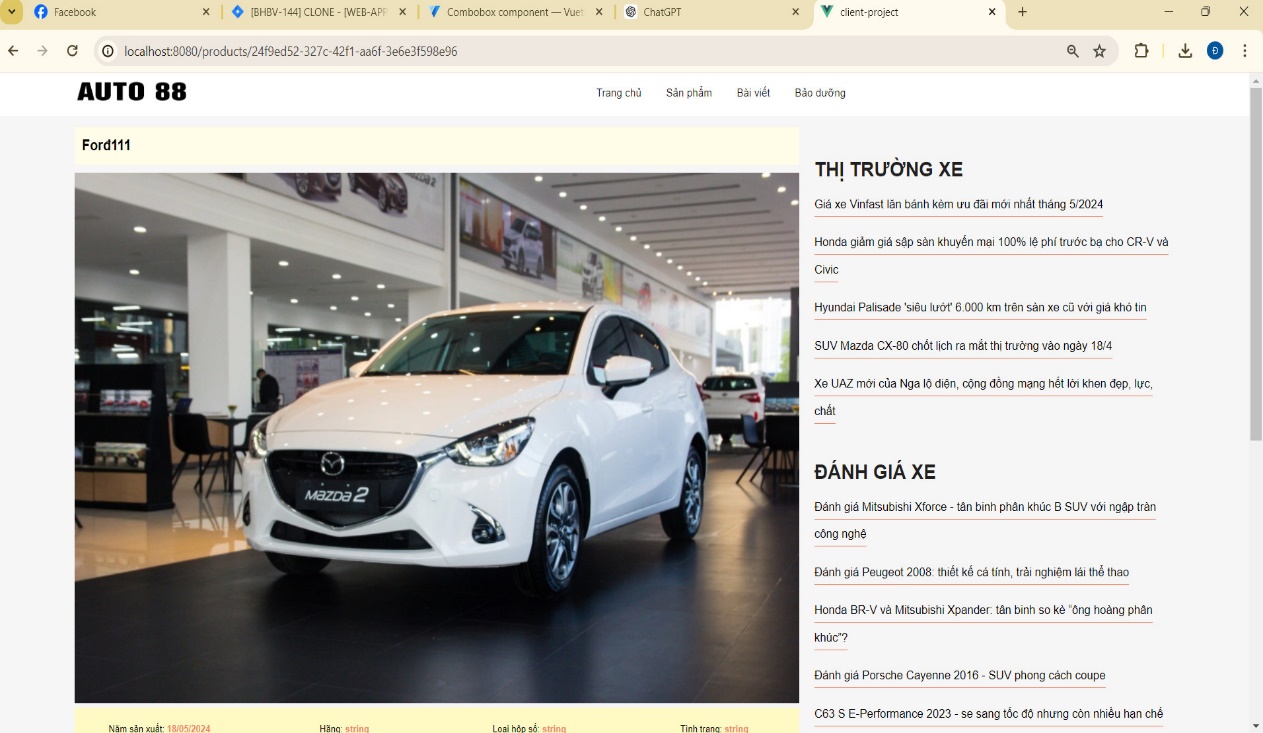
### Kết quả đạt được



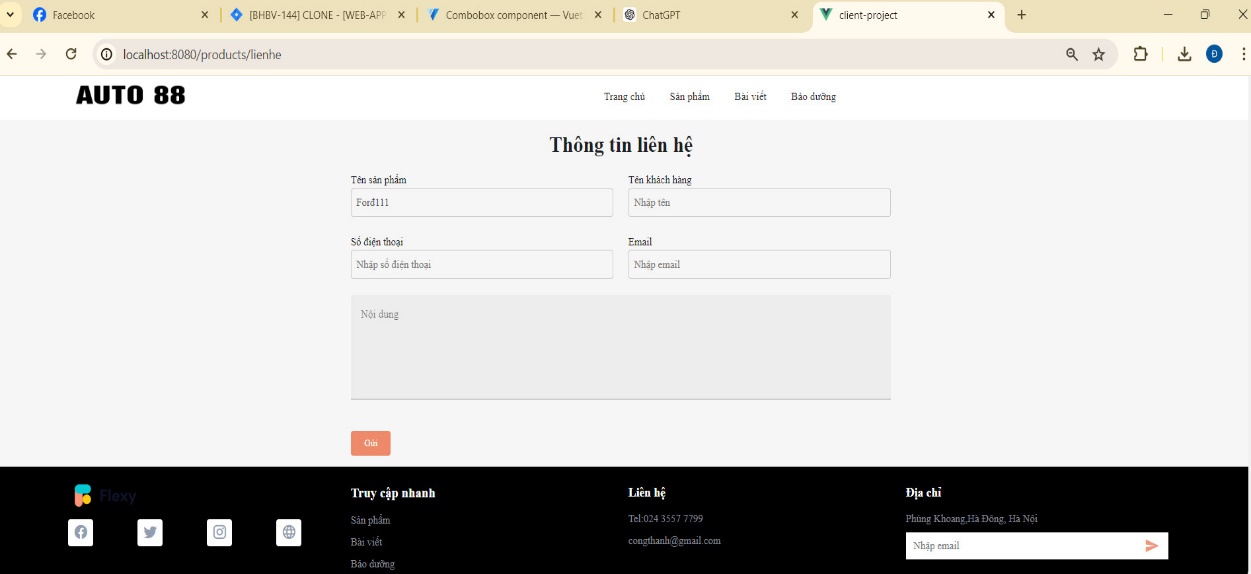
Hình .:Màn hình chính



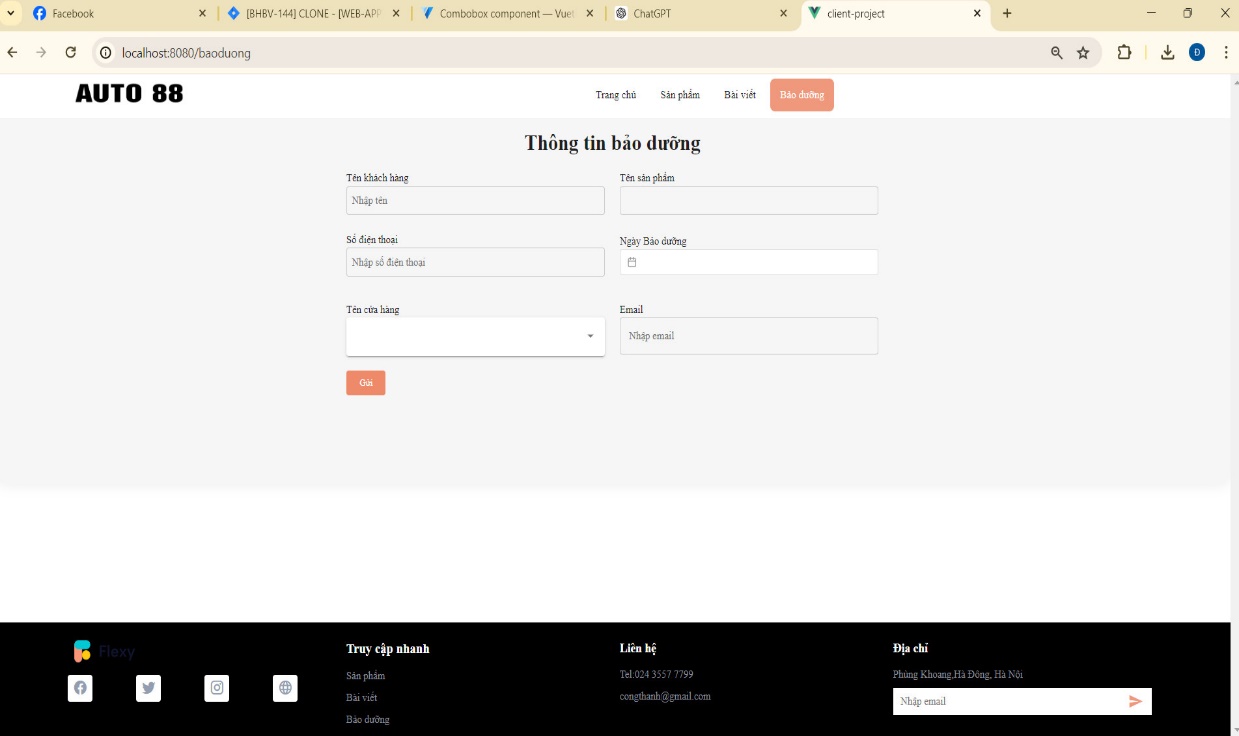
Hình .:Màn danh sách xe



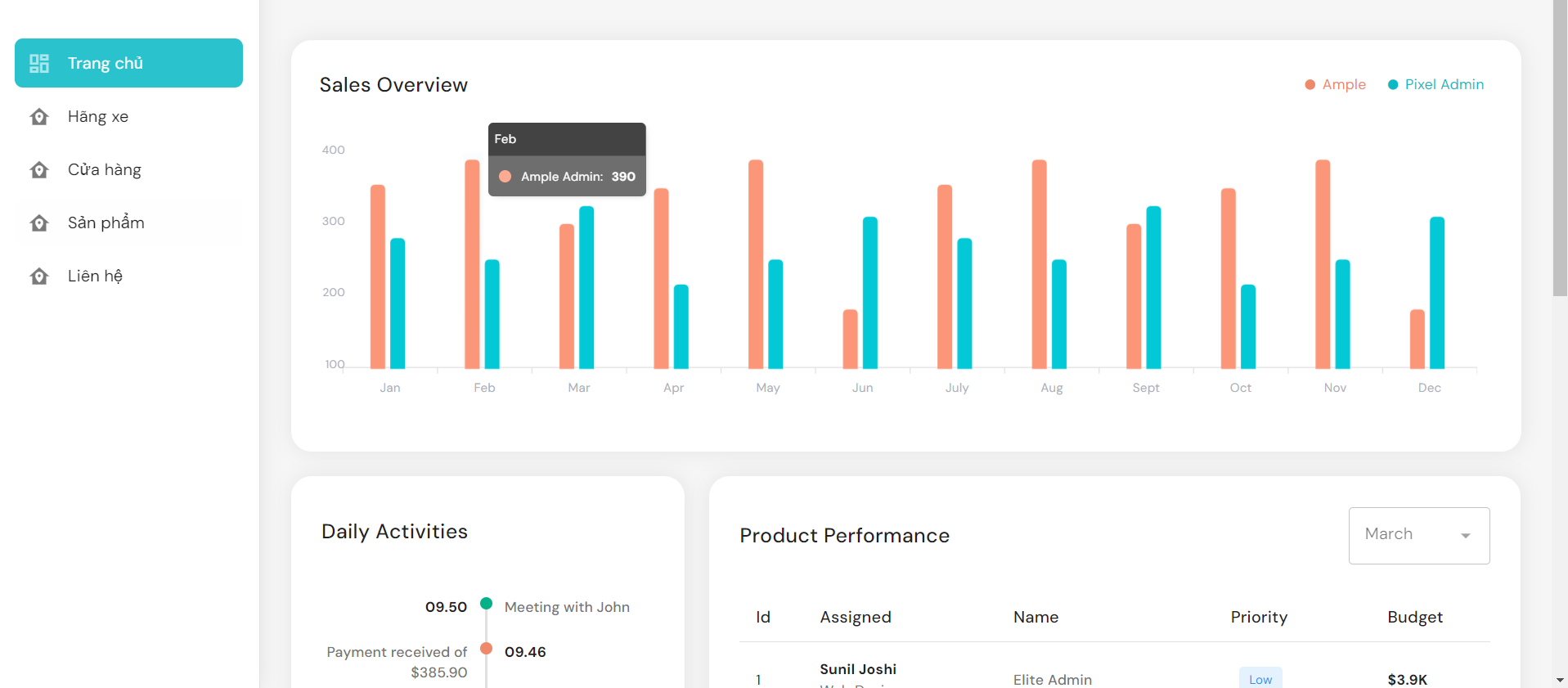
Hình .:Màn thông tin chi tiết xe



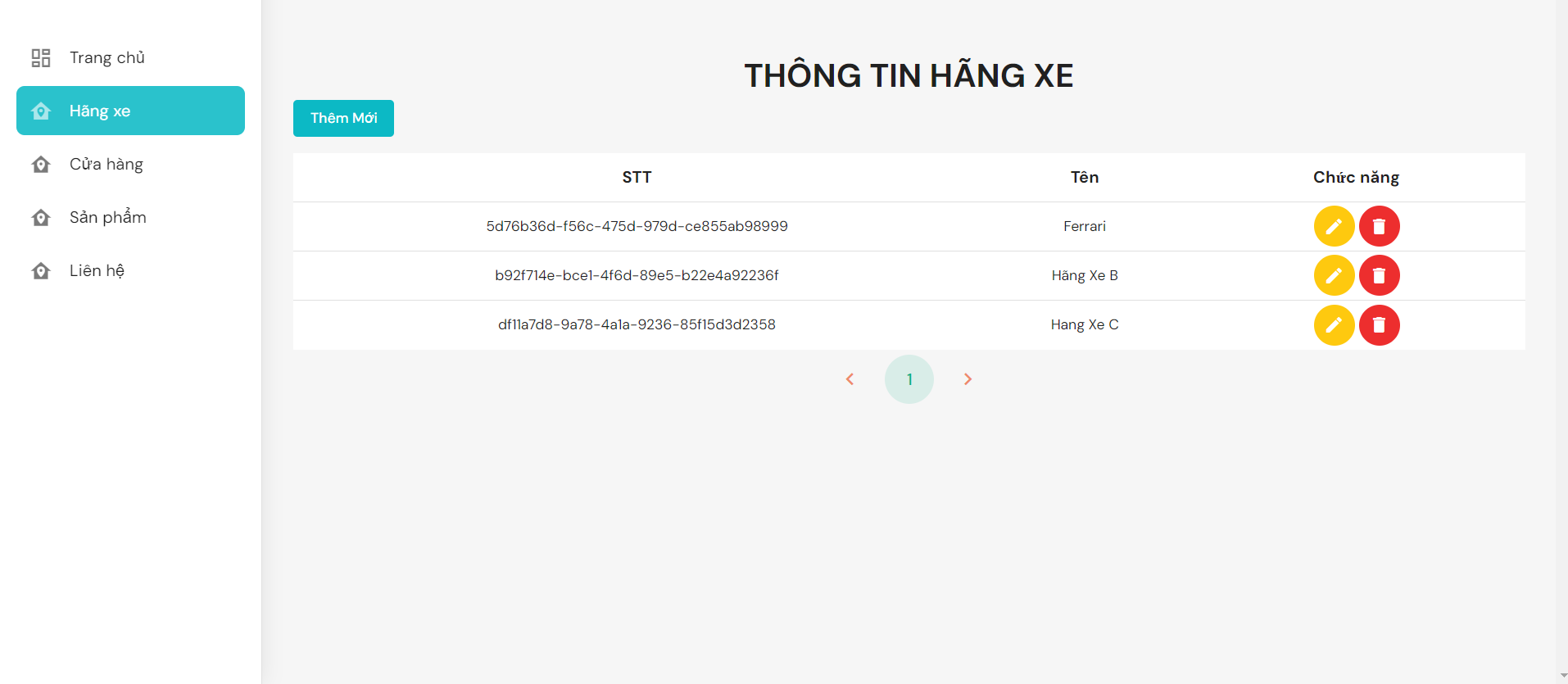
Hình .: Màn liên hệ



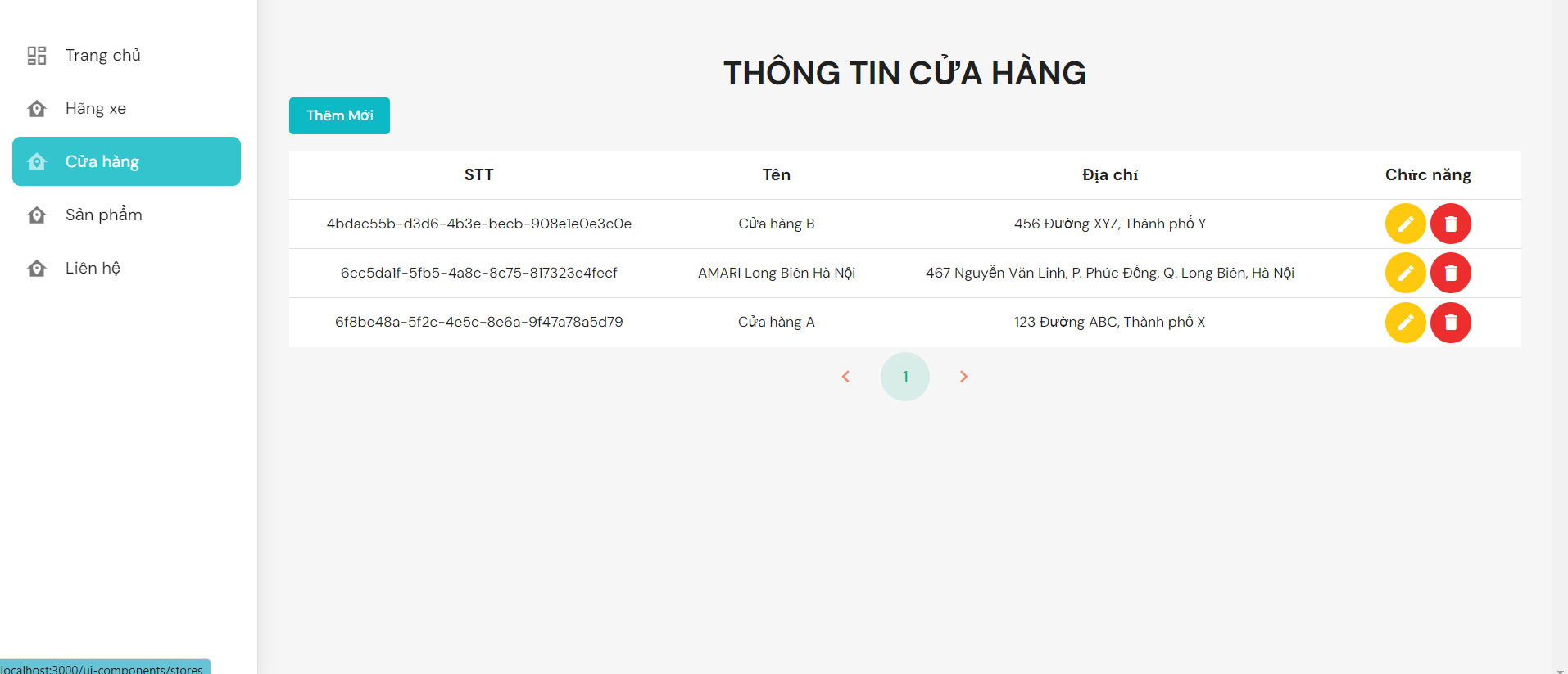
Hình .: Màn bào dưỡng



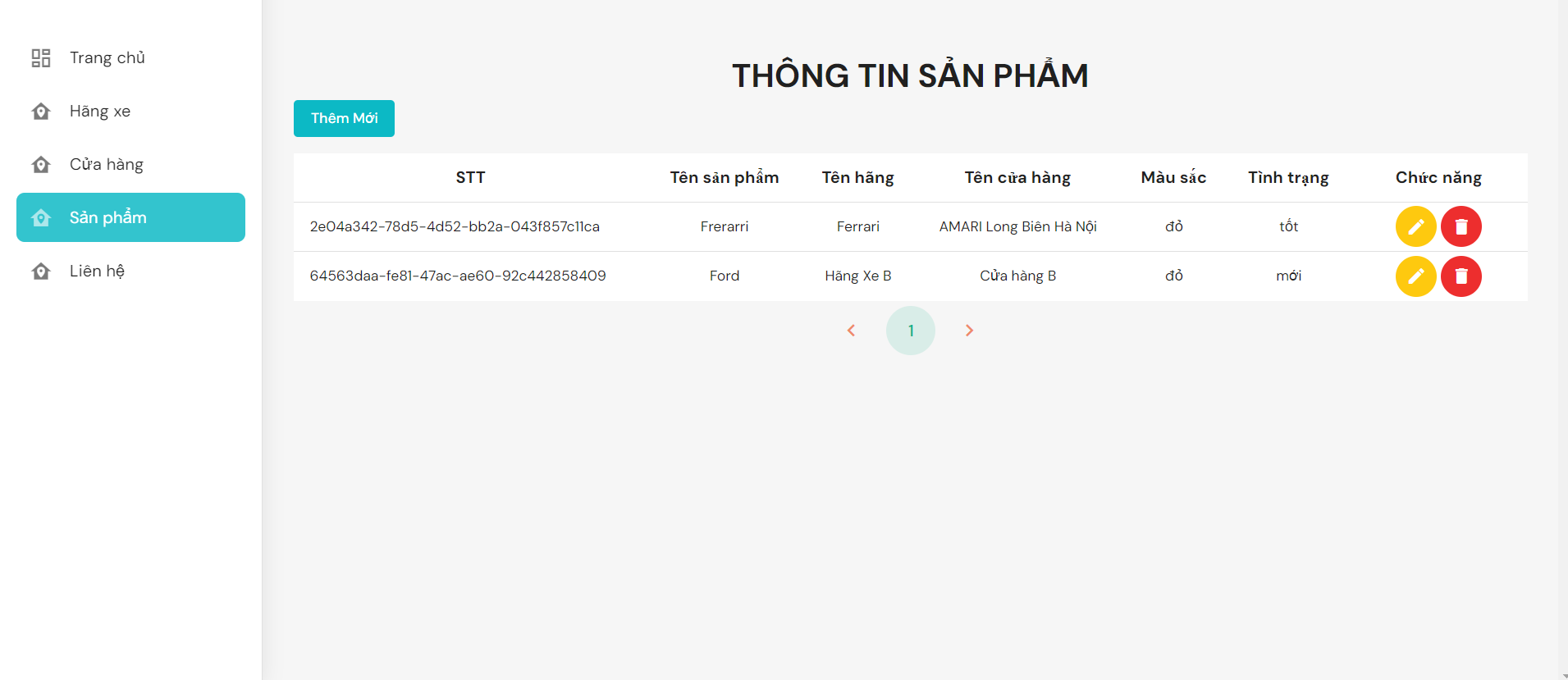
Hình .: Màn trang chủ admin



Hình . :Màn quản lí hãng xe



Hình . : Màn quản lí cửa hàng



Hình .: Màn quản lí sản phẩm

### Kiểm thử

#### Kế hoạch kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| Tên dự án | [www.auto88car.vn](http://www.auto88car.vn) | Người lập | Nguyễn Công Thành |
| Mã dự án | SSC | Người kiểm tra/ xét duyệt | Nguyễn Công Thành |
| Mã tài liệu | SSC\_TL\_01 | Ngày lập | 1/05/2024 |
| Phiên bản | 1 |

Bảng : Thông tin kiểm thử

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Kiểm thử chức năng | Kiểm thử chức năng:   * Đặt lịch lái thử * Đăng nhập |
| 2 | Kiểm thử giao diện | Kiểm thử chức năng:   * Tra cứu xe * Đăng nhập |

Bảng : Nội dung kiểm thử

#### Thực thi kiểm thử

##### Kiểm thử chức năng :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên test case** | **Kịch bản kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Pass(X)**  **/Fail** |
| **Đăng nhập** | | | | | |
| 1 | Đăng nhập thành công | Người dùng nhập đúng tên đăng nhập và mật khẩu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình chính của người dùng sau khi đã đăng nhập thành công. | X |
| 2 | Đăng nhập không thành công 1 | Người dùng nhập sai tên hoặc bỏ trống tên đăng nhập | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập sai hoặc bỏ trống tên đăng nhập, nhập mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình thông báo lỗi do sai tên đăng nhập và quay lại màn hình đăng nhập. | X |
| 3 | Đăng nhập không thành công 2 | Người dùng nhập sai hoặc bỏ trống mật khẩu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, nhập sai hoặc bỏ trống mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình thông báo lỗi do sai mật khẩu và quay lại màn hình đăng nhập. | X |
| 4 | Đăng nhập không thành công 3 | Người dùng nhập sai tên hoặc mật khẩu x lần | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập sai tên đăng nhập và mật khẩu và kích vào nút đăng nhập x lần. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị mà hình thông báo lỗi và khóa màn hình, người dùng không thể đăng nhập lại. | X |
| 5 | Đăng nhập không thành công 4 | Không kết nối được với cơ sở dữ liệu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị thông báo Không kết nối được với cơ sở dữ liệu. | X |
| **Đặt lịch lái thử** | | | | | |
| 6 | Đặt lịch thành công | Người dùng nhập thông tin  đúng | 1. Khách hàng bấm vào dịch vụ lái thử.  2. Khách hàng nhập thông tin rồi bấm nút “Gửi” | 1. Hệ thống hiển thị thông báo đã gửi lịch thành công . | X |
| 7 | Đặt lịch không thành công  1 | Khách hàng không nhập những thông tin yêu cầu . | 1. Khách hàng bấm vào dịch vụ lái thử.  2. Khách hàng nhập thông tin rồi bấm nút “Gửi” | 1. Hệ thống hiển thị thông báo “khách hàng chưa nhập thông tin yêu cầu”. | X |

Bảng : Kiểm thử chức năng hệ thống

##### Kiểm thử giao diện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên test case** | **Kịch bản kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Pass(X)**  **/Fail** |
| 1 | Màn hình chung. | Đảm bảo có thể dùng chuột, các phím tab, lên, xuống, trái, phải...để điều hướng các từ màn hình này sang màn hình khác, từ trường này qua trường khác. | 1. Người dùng bật màn hình giao diện người dùng.  2. Chọn di chuyển giữa các trường bằng các phím tab, phím điều hướng (lên xuống trái phải)" | 1. Hệ thống hiển thị giao diện người dùng đang bật  2. Trỏ chuột di chuyển đúng yêu cầu nghiệp vụ. | X |
| 2 | Màn hình đăng nhập | Đảm bảo trường tên đăng nhập và trường mật khẩu bắt buộc nhập phải được đánh dấu. | Nhìn vào các giao diện đang mở và đối chiếu với Figma | 1. Hệ thống hiển thị những trường cần nhập dữ liệu có dấu \*. | X |
| 3 | Màn hình tra cứu xe | Kiểm tra vị trí hiển thị các trường dữ liệu trên các trình duyệt khác nhau | Chạy tất cả các test case trên các trình duyệt được yêu cầu. | 1. Không bị vỡ form.  2. Trỏ chuột đặt đúng chỗ  3. Có thể điều hướng bằng tất cả các nút điều hướng hoặc trỏ chuột | X |

Bảng : Bảng mô tả kiểm thử giao diện

##### Kết quả kiểm thử :

* Tỉ lệ test case đạt (passed): 100%
* Tỉ lệ test case không đạt (failed): 0%

KÊT LUẬN

1. Những kết quả sau quá trình nghiên cứu

Đề tài “*Xây dựng website bán ô tô bằng VueJS và ASP.NET Core*” xuất phát từ những nhu cầu thực tế trong ngành kinh doanh ô tô hiện nay. Việc mua bán ô tô qua mạng đang trở thành xu hướng, giúp khách hàng dễ dàng tìm kiếm, so sánh và mua sắm các mẫu ô tô mà họ quan tâm một cách tiện lợi và nhanh chóng.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ mới.

**Những kết quả đạt được:**

* **Về công nghệ:**
* Hiểu được quá trình thiết kế một website thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, thực hiện, kiểm thử, triển khai, bảo trì.
* Xây dựng thành công website đáp ứng nhu cầu bán ô tô.
* Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript.
* Nắm vững về cách xây dựng website thông qua thư viện, framework: VueJS trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, Visual Studio .
* Sử dụng công cụ thiết kế figma.
* **Về cài đặt chương trình:**
* Cho phép khách hàng có thể xem thông tin sản phẩm và có thể đặt lịch trải nghiệm lái thử xe , bảo dưỡng chăm sóc xe
* Các mặt hàng được cập nhật liên tục theo nhu cầu của khách hàng.
* Trang quản trị phân quyền đầy đủ .
* **Tính năng khác:**
* Giao diện đẹp gây ấn tượng thân thiện với khách hàng.
* Tính bảo mật.

1. Kiến nghị về những nghiên cứu tiếp theo.

* *Tính Năng Tham Khảo và So Sánh:*

Cho phép người dùng so sánh nhiều mẫu xe cùng lúc để đưa ra quyết định mua sắm thông minh hơn.

Cung cấp các bài viết và bài đánh giá về các mẫu xe để giúp người dùng có cái nhìn tổng quan và chi tiết về sản phẩm.

* *Tính Năng Thanh Toán An Toàn:*

Tích hợp các phương thức thanh toán trực tuyến an toàn và đa dạng như thẻ tín dụng, chuyển khoản ngân hàng, và ví điện tử.

Cung cấp tính năng lưu trữ thông tin thanh toán để người dùng có thể thanh toán một cách nhanh chóng và thuận tiện.

* *Tính Năng Hỗ Trợ Khách Hàng:*

Tạo tính năng trò chuyện trực tuyến hoặc hỗ trợ qua điện thoại để giúp người dùng giải đáp thắc mắc và giải quyết vấn đề.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Tham khảo từ Microsoft : “ <https://docs.asp.net/en/latest/tutorials/first-mvc-app/index.html>” |
| [2] | Tham khảo từ :” https://vuejs.org/guide/introduction” |
| [3] | Tham khảo từ :”Fullstack.edu.vn” |
| [4] | Phùng Đức Hòa (Chủ biên); Hoàng Quang Huy; Trịnh Bá Quý, Giáo trình Nhập môn công nghệ phần mềm, Hà Nội: NXB Thống Kê, 2019. |
| [5] | Vũ Thị Dương; Phùng Đức Hòa; Nguyễn Thị Hương Lan, Giáo trình phân tích thiết kế hướng đối tượng, Hà Nội: Nhà Xuất Bản Khoa Học Và Kĩ Thuật, 2015. |