**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

====================

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE GIỚI THIỆU SẢN PHẨM VÀ QUẢN LÝ BÁN HÀNG**

**CHO CỬA HÀNG XE MÁY VIỆT HƯƠNG**

**CBHD: ThS. Vũ Minh Yến**

**Sinh viên: Nguyễn Văn Hoạt**

**Mã số sinh viên: 2019602947**

***Hà Nội - Năm 2023***

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

====================

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE GIỚI THIỆU SẢN PHẨM VÀ QUẢN LÝ BÁN HÀNG**

**CHO CỬA HÀNG XE MÁY VIỆT HƯƠNG**

**CBHD: ThS. Vũ Minh Yến**

**Sinh viên: Nguyễn Văn Hoạt**

**Mã số sinh viên: 2019602947**

***Hà Nội - Năm 2023***

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc132658411)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc132658412)

[1.1. Tổng quan ASP.NET Core MVC 2](#_Toc132658413)

[1.1.1. Mô hình MVC 3](#_Toc132658414)

[1.1.2. Cơ chế hoạt động của ASP.NET MVC 4](#_Toc132658415)

[1.2. Entity Framework Core 4](#_Toc132658416)

[1.3. Repository Design Patern 6](#_Toc132658417)

[1.4. Dependency Injection 7](#_Toc132658418)

[1.5. Thư viện Identity trong ASP.NET Core 7](#_Toc132658419)

[1.6. Hệ quản trị cơ sở dữ liệu SQL Server 8](#_Toc132658420)

[1.6.1. Hệ quản trị cơ sở dữ liệu SQL Server là gì? 8](#_Toc132658421)

[1.6.2. Các tính năng của SQL Server 8](#_Toc132658422)

[CHƯƠNG 2. KHẢO SÁT 10](#_Toc132658423)

[2.1. Giới thiệu cửa hàng 10](#_Toc132658424)

[2.2. Khảo sát sơ bộ 10](#_Toc132658425)

[2.2.1. Khảo sát công việc của từng vị trí trong cửa hàng 10](#_Toc132658426)

[2.2.2. Các biểu mẫu thu thập được. 11](#_Toc132658427)

[2.3. Đánh giá hiện trạng 13](#_Toc132658428)

[2.4. Mục tiêu của hệ thống 13](#_Toc132658429)

[2.5. Yêu cầu đối với hệ thống mới 14](#_Toc132658430)

[2.5.1. Yêu cầu chức năng 14](#_Toc132658431)

[2.5.2. Yêu cầu phi chức năng 15](#_Toc132658432)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 16](#_Toc132658433)

[3.1. Biểu đồ usecase 16](#_Toc132658434)

[3.1.1. Biểu đồ 16](#_Toc132658435)

[3.1.2. Đặc tả usecase 16](#_Toc132658436)

[3.2. Biểu đồ trình tự 30](#_Toc132658437)

[3.3. Biểu đồ lớp lĩnh vực 48](#_Toc132658438)

[3.4. Thiết kế cơ sở dữ liệu 54](#_Toc132658439)

[3.4.1. Bảng LoaiXe 54](#_Toc132658440)

[3.4.2. Bảng XeMay 54](#_Toc132658441)

[3.4.3. Bảng PhieuNhap 56](#_Toc132658442)

[3.4.4. Bảng ChiTietPhieuNhap 56](#_Toc132658443)

[3.4.5. Bảng LienHeDatHang 57](#_Toc132658444)

[3.4.6. Bảng PhieuBaoHanh 57](#_Toc132658445)

[3.4.7. Bảng User 58](#_Toc132658446)

[3.4.8. Bảng DanhGia 59](#_Toc132658447)

[3.4.9. Bảng DonHang 60](#_Toc132658448)

[3.4.10. Bảng HoaDonBanHang 61](#_Toc132658449)

[KẾT LUẬN 62](#_Toc132658450)

[TÀI LIỆU THAM KHẢO 63](#_Toc132658451)

**DANH MỤC CÁC BẢNG**

[Bảng 1‑1: Các phiên bản của .NET Core 2](#_Toc132048264)

[Bảng 1‑2: Các phiên bản của Entity Framework Core 5](#_Toc132048265)

[Bảng 3‑1: Đặc tả usecase đăng nhập 16](#_Toc132048266)

[Bảng 3‑2: Đặc tả usecase đăng ký 17](#_Toc132048267)

[Bảng 3‑3: Đặc tả usecase xem danh sách xe 18](#_Toc132048268)

[Bảng 3‑4: Đặc tả usecase xem chi tiết xe 19](#_Toc132048269)

[Bảng 3‑5: Đặc tả usecase xem thông tin đơn hàng đã mua 19](#_Toc132048270)

[Bảng 3‑6: Đặc tả usecase đánh giá xe đã mua 20](#_Toc132048271)

[Bảng 3‑7: Đặc tả usecase thêm xe vào danh sách yêu thích 21](#_Toc132048272)

[Bảng 3‑8: Đặc tả usecase liên hệ đặt hàng 21](#_Toc132048273)

[Bảng 3‑9: Đặc tả usecase quản lý hãng xe 22](#_Toc132048274)

[Bảng 3‑10: Đặc tả usecase quản lý thông tin xe 23](#_Toc132048275)

[Bảng 3‑11: Đặc tả usecase quản lý tài kh oản 24](#_Toc132048276)

[Bảng 3‑12: Đặc tả usecase quản lý nhập kho 26](#_Toc132048277)

[Bảng 3‑13: Đặc tả usecase quản lý đơn hàng 27](#_Toc132048278)

[Bảng 3‑14: Đặc tả usecase quản lý hóa đơn 27](#_Toc132048279)

[Bảng 3‑15: Đặc tả usecase quản lý bảo hành 28](#_Toc132048280)

[Bảng 3‑16: Đặc tả usecase báo cáo thống kê 29](#_Toc132048281)

[Bảng 3‑17: Chi tiết bảng LoaiXe 54](#_Toc132048282)

[Bảng 3‑18: Chi tiết bảng XeMay 54](#_Toc132048283)

[Bảng 3‑19: Chi tiết bảng PhieuNhap 56](#_Toc132048284)

[Bảng 3‑20: Chi tiết bảng ChiTietPhieuNhap 56](#_Toc132048285)

[Bảng 3‑21: Chi tiết bảng LienHeDatHang 57](#_Toc132048286)

[Bảng 3‑22: Chi tiết bảng PhieuBaoHanh 57](#_Toc132048287)

[Bảng 3‑23: Chi tiết bảng User 58](#_Toc132048288)

[Bảng 3‑24: Chi tiết bảng DanhGia 59](#_Toc132048289)

[Bảng 3‑25: Chi tiết bảng DonHang 60](#_Toc132048290)

[Bảng 3‑26: Chi tiết bảng HoaDonBanHang 61](#_Toc132048291)

**DANH MỤC HÌNH ẢNH**

[Hình 1‑1: Mô hình MVC 3](#_Toc133150057)

[Hình 1‑2: Cơ chế hoạt động của ASP.NET MVC 4](#_Toc133150058)

[Hình 1‑3: Object Relational Mapping (ORM) 5](#_Toc133150059)

[Hình 1‑4: Hai cách tiếp cận trong Entity FrameWork Core 6](#_Toc133150060)

[Hình 1‑5: Sơ đồ tổ chức của Repository Parten 6](#_Toc133150061)

[Hình 2‑1: Phiếu nhập kho 11](#_Toc133150062)

[Hình 2‑2: Hóa đơn bán hàng 12](#_Toc133150063)

[Hình 2‑3: Phiếu bảo hành 13](#_Toc133150064)

[Hình 3‑1: Biểu đồ UseCase 17](#_Toc133150065)

[Hình 3‑2: Biếu đồ tuần tự usecase đăng nhập 32](#_Toc133150066)

[Hình 3‑3: Biểu đồ tuần tự usecase đăng ký 33](#_Toc133150067)

[Hình 3‑4: Biểu đồ tuần tự xem danh sách xe 34](#_Toc133150068)

[Hình 3‑5: Biểu đồ tuần tự usecase xem chi tiết xe máy 35](#_Toc133150069)

[Hình 3‑6: Biểu đồ tuần tự usecase xem thông tin đơn hàng đã mua 36](#_Toc133150070)

[Hình 3‑7: Biểu đồ tuần tự usecase đánh giá xe đã mua 37](#_Toc133150071)

[Hình 3‑8: Biểu đồ tuần tự usecase thêm xe vào danh sách yêu thích 38](#_Toc133150072)

[Hình 3‑9: Biểu đồ tuần tự usecase liên hệ đặt hàng 39](#_Toc133150073)

[Hình 3‑10: Biểu đồ tuần tự usecase quản lý hãng xe 42](#_Toc133150074)

[Hình 3‑11: Biểu đồ tuần tự usecase quản lý xe 45](#_Toc133150075)

[Hình 3‑12: Biểu đồ tuần tự usecase quản lý tài khoản 45](#_Toc133150076)

[Hình 3‑13: Biểu đồ tuần tự usecase quản lý nhập kho 46](#_Toc133150077)

[Hình 3‑14: Biểu đồ tuần tự usecase quản lý đơn hàng 47](#_Toc133150078)

[Hình 3‑15: Biểu đồ tuần tự usecase quản lý bảo hành 48](#_Toc133150079)

[Hình 3‑16: Biểu đồ tuần tự usecase thống kê 49](#_Toc133150080)

[Hình 3‑17: Biểu đồ lớp lĩnh vực usecase đăng nhập 49](#_Toc133150081)

[Hình 3‑18: Biểu đồ lớp lĩnh vực usecase đăng ký 50](#_Toc133150082)

[Hình 3‑19: Biểu đồ lớp lĩnh vực usecase xem danh sách xe 50](#_Toc133150083)

[Hình 3‑20: Biểu đồ lớp lĩnh vực usecase xem chi tiết xe 50](#_Toc133150084)

[Hình 3‑21: Biểu đồ lớp lĩnh vực usecase xem thông tin đơn hàng đã mua 51](#_Toc133150085)

[Hình 3‑22: Biểu đồ lớp lĩnh vực usecase đánh giá xe đã mua 51](#_Toc133150086)

[Hình 3‑23: Biểu đồ lớp lĩnh vực usecase thêm xe vào danh sách yêu thích 51](#_Toc133150087)

[Hình 3‑24: Biểu đồ lớp lĩnh vực usecase liên hệ đặt hàng 52](#_Toc133150088)

[Hình 3‑25: Biểu đồ lớp lĩnh vực usecase quản lý hãng xe 52](#_Toc133150089)

[Hình 3‑26: Biểu đồ lớp lĩnh vực usecase quản lý thông tin xe 52](#_Toc133150090)

[Hình 3‑27: Biểu đồ lớp lĩnh vực usecase quản lý tài khoản 53](#_Toc133150091)

[Hình 3‑28: Biểu đồ lớp lĩnh vực usecase quản lý nhập kho 53](#_Toc133150092)

[Hình 3‑29: Biểu đồ lớp lĩnh vực usecase quản lý đơn hàng 54](#_Toc133150093)

[Hình 3‑30: Biểu đồ lớp lĩnh vực usecase quản lý bảo hành 54](#_Toc133150094)

[Hình 3‑31: Biểu đồ lớp lĩnh vực usecase thống kê 54](#_Toc133150095)

[Hình 3‑32: Cơ sở dữ liệu 55](#_Toc133150096)

[Hình 4‑1: Entity sử dụng trong project 66](#_Toc133150097)

# MỞ ĐẦU

# CƠ SỞ LÝ THUYẾT

## Tổng quan ASP.NET Core MVC

ASP.Net là một nền tảng dành cho phát triển web, được Microsoft phát hành và cung cấp lần đầu tiên vào năm 2002. Nền tảng được sử dụng để tạo ra các ứng dụng web-based.

Các ứng dụng ASP.Net có thể được viết bằng nhiều ngôn ngữ .Net khác nhau. Trong đó có các kiểu ngôn ngữ như C #, VB.Net và F#.

Sau đây là các phiên bản của ASP.NET Core

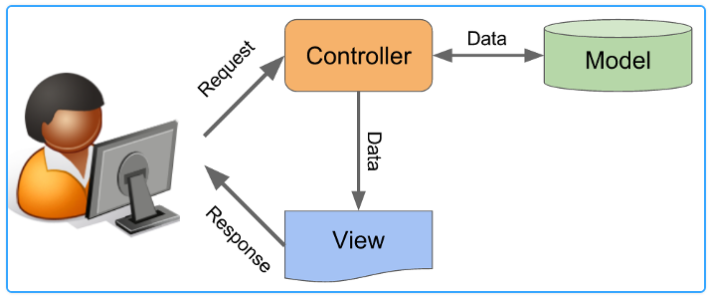
Bảng 1‑1: Các phiên bản của .NET Core

|  |  |  |  |
| --- | --- | --- | --- |
| **Phiên bản** | **Ngày phát hành** | **Phiên bản cuối** | **Ngày cập nhật cuối cùng** |
| .NET Core 1.0 | 27/06/2016 | 1.0.16 | 14 /05/2019 |
| .NET Core 1.1 | 16/11/2016 | 1.1.13 | 14 /05/2019 |
| .NET Core 2.0 | 14/08/2017 | 2.0.9 | 10/07/2018 |
| .NET Core 2.1 | 30/05/2018 | 2.1.30 (LTS) | 19/08/2021 |
| .NET Core 2.2 | 04/12/2018 | 2.2.8 | 19/11/2019 |
| .NET Core 3.0 | 23/09/2019 | 3.0.3 | 18/02/2020 |
| .NET Core 3.1 | 03/12/2019 | 3.1.22 (LTS) | 14/12/2021 |
| .NET 5 | 10/11/2020 | 5.0.13 | 14/12/2021 |
| .NET 6 | 08/11/2021 | 6.0.1 (LTS) | 14/12/2021 |
| .NET 7 | 09/11/2022 | 7.0.1 | 10/01/2023 |
| .NET 8 | 2023-11 (dự kiến) | LTS |  |

### Mô hình MVC

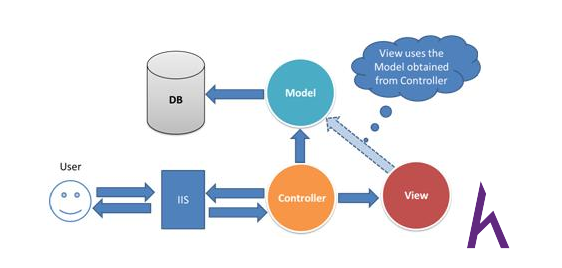
MVC (Model – View - Controller) là một design partern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller.

* Model: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.
* View: Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
* Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.



Hình 1‑1: Mô hình MVC

### Cơ chế hoạt động của ASP.NET MVC



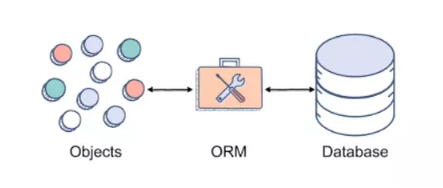
Hình 1‑2: Cơ chế hoạt động của ASP.NET MVC

Dựa vào hình mô phỏng ở *Hình 1-2* thì cơ chế hoạt động của ASP.Net MVC gồm các bước:

* User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser
* Yêu cầu đó được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ chuyển qua tầng model
* Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller
* Controller sẽ giúp dữ liệu được chuyển từ model qua view
* View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho User thông qua tầng View

## Entity Framework Core

Entity Framework Core là một Object Relational Mapping (ORM) framework giúp dễ dàng truy xuất và lưu trữ dữ liệu trong database thông qua việc ánh xạ database thành các objects tương ứng trong code.

****

Hình 1‑3: Object Relational Mapping (ORM)

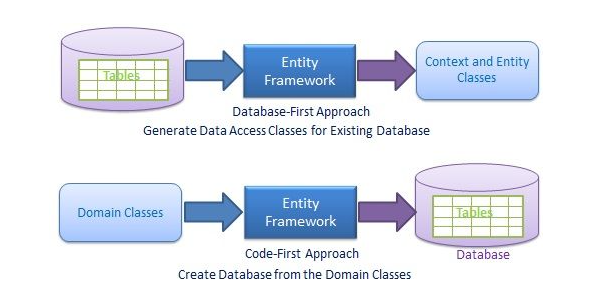
* Các phiên bản Entity FrameWork Core

Bảng 1‑2: Các phiên bản của Entity Framework Core

| **Phiên bản** | **Thời gian phát hành** |
| --- | --- |
| EF Core 2.0 | 8- 2017 |
| EF Core 1.1 | 11- 2016 |
| EF Core 1.0 | 6- 2016 |

EF Core hỗ trợ hai cách tiếp cận phát triển: Code First và Database First. EF Core chủ yếu nhắm vào cách tiếp cận Code First và cung cấp ít hỗ trợ cho cách tiếp cận Database First vì trình thiết kế trực quan hoặc trình hướng dẫn cho mô hình DB không được hỗ trợ kể từ EF Core 2.0.

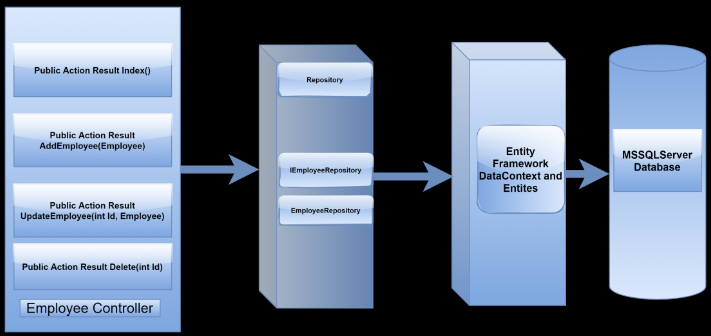
* Theo cách tiếp cận Code First, EF Core API tạo cơ sở dữ liệu và các bảng bằng cách sử dụng chuyển đổi (migration) dựa trên các quy ước và cấu hình được cung cấp trong các lớp thực thể. Cách tiếp cận này rất hữu ích trong thiết kế hướng miền (Domain Driven Design - DDD).
* Theo cách tiếp cận Database First, EF Core API tạo các lớp thực thể và Context dựa trên cơ sở dữ liệu có sẵn bằng cách sử dụng các lệnh của EF Core. Điều này được hỗtrợ hạn chế trong EF Core vì nó không hỗ trợ trình thiết kế trực quan hoặc trình hướng dẫn.



Hình 1‑4: Hai cách tiếp cận trong Entity FrameWork Core

## Repository Design Patern

Repository là một design parten dùng để tạo ra một lớp abstraction trung gian giữa lớp data và lớp business. Lớp này chứa đựng phương thức thao tác mà để giao tiếp với lớp data để phục vụ cho business từ lớp logic. Mục đích tạo ra lớp này để cách ly với việc tiếp cận data sao cho những thay đổi không ảnh hưởng trực tiếp đến lớp logic business.



Hình 1‑5: Sơ đồ tổ chức của Repository Parten

## Dependency Injection

Dependency injection (DI) là một kỹ thuật trong lập trình, nó là một hình thức cụ thể của Inverse of Control (Dependency Inverse). DI thiết kế sao cho các dependency (phụ thuộc) của một đối tượng có thể được đưa vào, tiêm vào đối tượng đó (Injection) khi nó cần tới (khi đối tượng khởi tạo). Cụ thể cần làm:

* Xây dựng các lớp (dịch vụ) có sự phụ thuộc nhau một cách lỏng lẻo, và dependency có thể tiêm vào đối tượng (injection) - thường qua phương thức khởi tạo constructor, property, setter
* Xây dựng được một thư viện có thể tự động tạo ra các đối tượng, các dependency tiêm vào đối tượng đó, thường là áp dụng kỹ thuật Reflection của C# (xem thêm lớp type): Thường là thư viện này quá phức tạp để tự phát triển nên có thể sử dụng các thư viện có sẵn như: Microsoft.Extensions.DependencyInjection hoặc thư viện bên thứ ba như Windsor, Unity Ninject ...
* Sử dụng thư viện Identity để xác thực và phân quyền

ASP.NET Core Identity là một thành phần (built-in) của ASP.NET Core, nó cung cấp các tính năng đầy đủ và đa dạng về authentication. Có thể như: Tạo tài khoản, login với user name và password, cập nhật profile. Hoặc cũng có thể sử dụng những provider bên ngoài giống như: Facebook, Google, Twitter, Microsoft account và nhiều cái khác nữa. Cũng có thể cấu hình ASP.NET Core để sử dụng với SQL Server nhằm lưu trữ tên đăng nhập, mật khẩu và dữ liệu của tài khoản. Bài lược dịch này tôi sẽ giới thiệu với các bạn cách tạo, cấu hình project có sử dụng Identity, đồng thời có thể chỉnh sửa một số thông tin của người dùng.

## Thư viện Identity trong ASP.NET Core

ASP.NET Core Identity là một thành phần (built-in) của ASP.NET Core, nó cung cấp các tính năng đầy đủ và đa dạng về authentication. Có thể như: Tạo tài khoản, login với user name và password, cập nhật profile. Hoặc cũng có thể sử dụng những provider bên ngoài giống như: Facebook, Google, Twitter, Microsoft account và nhiều cái khác nữa. Cũng có thể cấu hình ASP.NET Core để sử dụng với SQL Server nhằm lưu trữ user name, password và dữ liệu profile. Bài lược dịch này tôi sẽ giới thiệu với các bạn cách tạo, cấu hình project có sửu dụng Identity, đồng thời custom một số thông tin của người dùng.

## Hệ quản trị cơ sở dữ liệu SQL Server

### Hệ quản trị cơ sở dữ liệu SQL Server là gì?

SQL Server (viết tắt của cụm từ Structured Query Language) là một hệ thống quản lý cơ sở dữ liệu quan hệ (Relational Database Management System, viết tắt là RDBMS). SQL Server có khả năng hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty hoạt động trong lĩnh vực IT.

Cũng giống như các hệ thống quản lý cơ sở dữ liệu qua hệ khác, SQL Server được xây dựng trên lớp SQL – là ngôn ngữ lập trình tiêu chuẩn hoá được quản trị viên cơ sở dữ liệu (DBAs) và các chuyên gia IT sử dụng để quản lý cơ sở dữ liệu và truy vấn các dữ liệu nằm bên trong.

### Các tính năng của SQL Server

**Tính năng doanh nghiệp chuyên biệt**

Microsoft hiện nay đã ra mắt thêm các tính năng quản lý dữ liệu đa dạng dành cho doanh nghiệp và các công cụ phân tích SQL Server. Đi cùng với các dịch vụ Machine Learning được tích hợp lần đầu tiên trong phiên bản SQL Server 2016, các dịch vụ phân tích dữ liệu, công cụ phân tích dữ liệu, các ứng dụng trực quan hoá dữ liệu và các dịch vụ SQL Server Reporting nhằm hỗ trợ và phân phối các báo cáo BI.

**Tính năng quản lý**

Về phương diện quản trị, Microsoft SQL Server gồm các dịch vụ tích hợp SQL Server, dịch vụ SQL Server Data Quality và dịch vụ SQL Server Master. Hai bộ công cụ dành riêng cho quản trị viên cơ sở dữ liệu và lập trình viên (SQL Server Data Tools) sử dụng trong việc phát triển hệ thống cơ sở dữ liệu cho SQL Server Management Studio, thực hiện nhiệm vụ triển khai, giám sát và quản lý các cơ sở dữ liệu.

* Tại sao nên sử dụng SQL Server trong thiết kế website?

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập. Nó là thành phần với vai trò ngôn ngữ làm công cụ giao tiếp của cơ sở dữ liệu và người dùng. Bởi vậy, nó được sử dụng trong các dịch vụ thiết kế web được tích hợp tính năng giao tiếp với người dùng với các vai trò sau:

* SQL là một ngôn ngữ đòi hỏi sự tương tác cao. Người dùng có thể dễ dàng trao đổi với tiện ích bằng câu lệnh SQL đến cơ sở dữ liệu và nhận lại kết quả từ đó.
* SQL là một ngôn ngữ lập trình cơ sở dữ liệu. Bằng cách nhúng các câu lệnh SQL trong ngôn ngữ lập trình, các lập trình viên có thể xây dựng được các chương trình ứng dụng giao tiếp với cơ sở dữ liệu
* SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu. Quản trị viên cơ sở dữ liệu có thể quản lý và điều khiển các truy cập tới cơ sở dữ liệu thông qua SQL
* SQL được sử dụng như một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách – chủ.
* SQL được sử dụng với vai trò tương tác với các dữ liệu trong các máy chủ web và máy chủ Internet
* SQL có vai trò giao tiếp với các hệ thống trên mạng, làm nhiệm vụ gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

# KHẢO SÁT

## Giới thiệu cửa hàng

Công ty TNHH Xe Máy Việt Hương được thành lập vào ngày 10 tháng 10 năm 2009, định hướng phát triển của doanh nghiệp là chuyên cung cấp các loại xe máy từ thương hiệu Honda, Yamaha.

Công ty TNHH Xe Máy Việt Hương trang bị những thiết bị hiện đại, tân tiến nhất cùng đội ngũ kỹ thuật có trình độ chuyên môn cao và kỹ năng nghiệp vụ chuyên nghiệp hứa hẹn sẽ mang đến những sản phẩm có chất lượng cao với mức giá cạnh tranh, phù hợp cho mọi đối tượng. Hướng đến việc đảm bảo đáp ứng nhu cầu của khách hàng, và mang đến trải nghiệm tốt nhất khi sử dụng các sản phẩm và dịch vụ tại Việt Hương Moto.

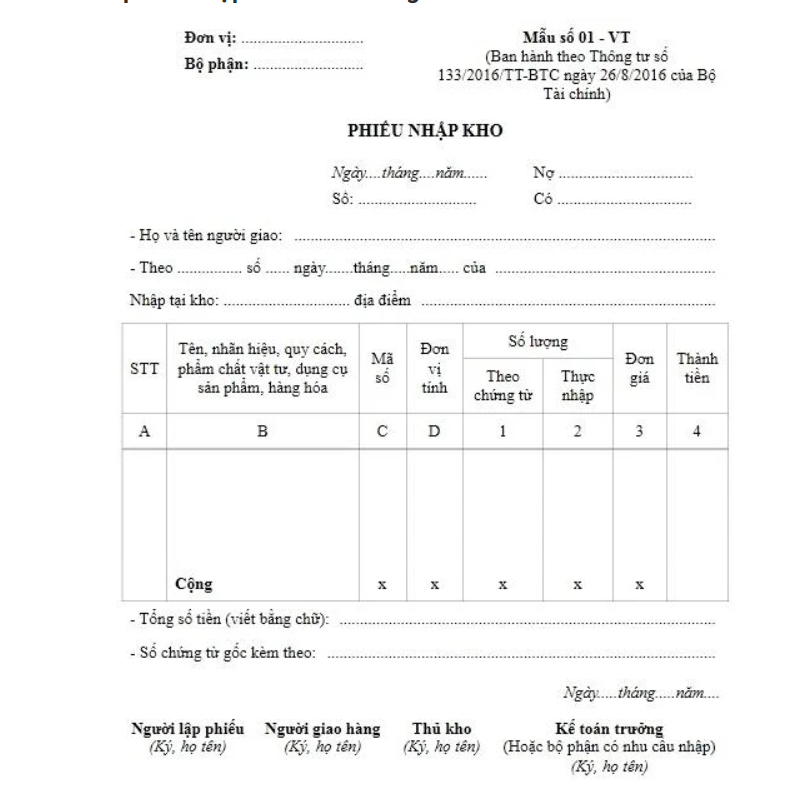
## Khảo sát sơ bộ

### Khảo sát công việc của từng vị trí trong cửa hàng

* Chủ cửa hàng
  + Quản lý toàn bộ hoạt động của cửa hàng
  + Nhận báo cáo thống kê định kỳ của nhân viên kế toán và nhân viên bán hàng
  + Kiểm tra các hoá đơn để nhập xuất tiền.
* Nhân viên bán hàng.
  + Phụ trách việc thu giữ, thống kê, tổng hợp các loại giấy tờ, sổ sách của công ty
  + Phụ trách việc nhập và xuất hàng
  + Báo cáo tài chính cho chủ cửa hàng
  + Bán hàng, giúp khách hàng lựa chọn sản phẩm và giúp khách hàng làm thủ tục thanh toán khi quyết định mua hàng.
  + Thường xuyên nghe ngóng và tìm hiểu về mức độ hài lòng của khách về cả sản phẩm và phong cách bán hàng để đưa ra cách khắc phục những hạn chế.

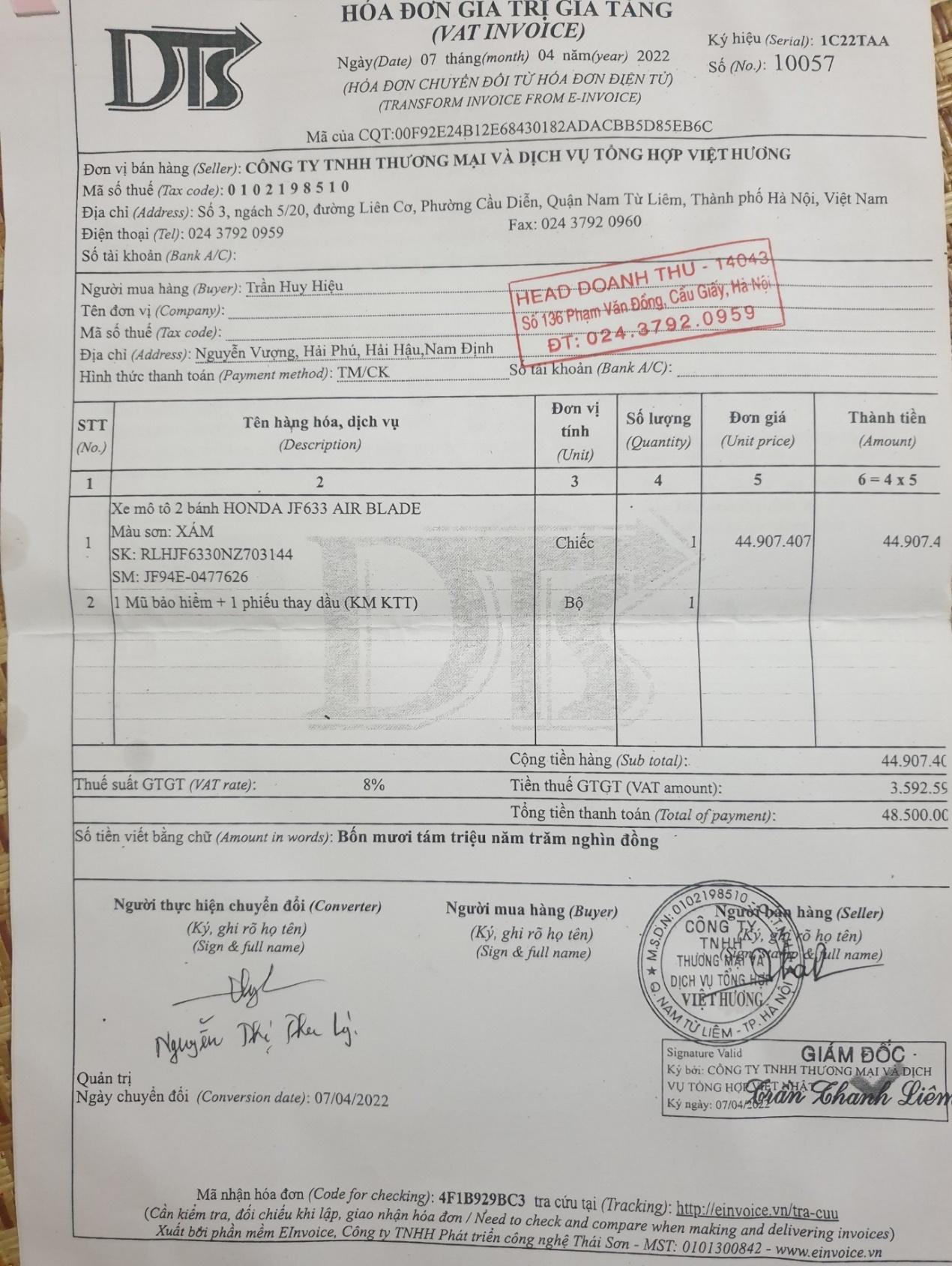
### Các biểu mẫu thu thập được.

* Phiếu nhập kho



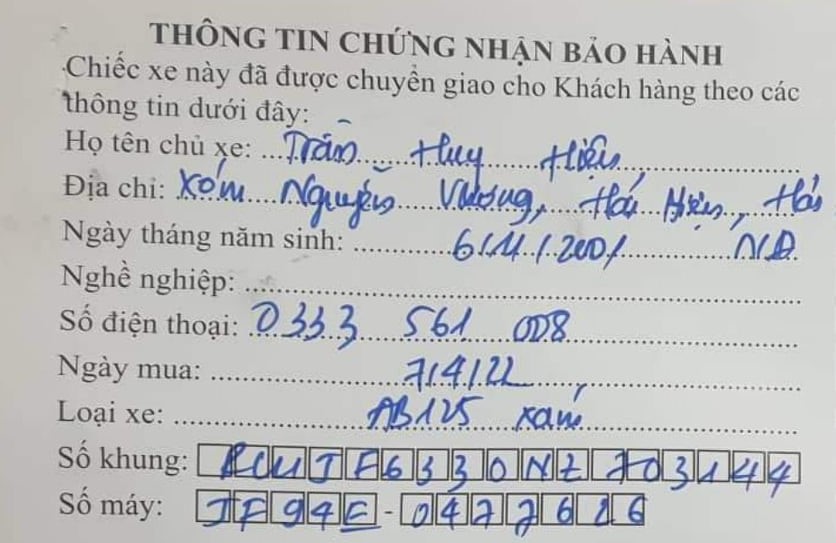
Hình 2‑1: Phiếu nhập kho

* Hoá đơn bán hàng



Hình 2‑2: Hóa đơn bán hàng

* Phiếu bảo hành



Hình 2‑3: Phiếu bảo hành

## Đánh giá hiện trạng

Khi không sử dụng hệ thống quản lý bán hàng:

* Thông tin sản phẩm mới, giá sản phẩm và những chương trình khuyến mãi chưa được tiếp cận nhanh chóng tới khách hàng
* Quản lý cồng kềnh, làm cho người quản lý khó khăn trong việc lưu trữ, quản lý và tra cứu thông tin sản phẩm.
* Còn nhiều công việc chưa tự động vẫn thủ công, chủ cửa hàng vẫn phải sử dụng giấy tờ nhiều dẫn đến những sai sót trong quá trình làm việc.

## Mục tiêu của hệ thống

Website quản lý bán hàng sẽ giúp tiết kiệm được nhiều chi phí, thời gian, giảm tối đa áp lực công việc cho các bộ phận bán hàng của cửa hàng Việt Hương, đồng thời tăng tính chính xác và quản lý thông tin toàn diện về cửa hàng như thông tin tài khoản, hãng xe, số lượng xe, doanh thu của cửa hàng. Từ đó giúp cửa hàng tăng doanh thu, giảm chí phí và đưa các sản phẩm dễ dàng tiếp cần với khách hàng.

## Yêu cầu đối với hệ thống mới

### Yêu cầu chức năng

Đưa ra một website giúp cho việc quản lý bán hàng dễ dàng và tiện lợi hơn:

* Đối với người quản trị hệ thống
* Quản lý hãng xe
  + Nhập các hãng xe cần thêm vào hệ thống.
  + Với những hãng xe cần thay đổi, quản trị viên bấm sửa hãng xe hoặc xóa hãng xe.
* Quản lý thông tin xe máy
  + Nhập thông tin các xe máy mới vào hệ thống.
  + Với những xe cần thay đổi thông tin, quản trị viên bấm sửa thông tin xe hoặc xóa xe.
  + Quản lý giá xe máy
* Quản lý tài khoản
  + Nhập thông tin tài khoản mới vào hệ thống.
  + Với những tài khoản cần thay đổi thông tin, quản trị viên bấm sửa thông tin tài khoản hoặc xóa tài khoản.
* Đăng nhập
  + Quản trị viên cần đăng nhập bằng tài khoản admin để sử dụng hệ thống.
* Đối với nhân viên bán hàng
* Quản lý nhập kho
  + Quản lý thông tin về các lô hàng nhập vào cửa hàng.
* Quản lý đơn hàng
  + Quản trị viên có thể tạo đơn hàng mới cho khách hàng dựa trên liên hệ mua hàng của khách.
  + Sau khi lập đơn hàng mới có thể xuất hóa đơn bán hàng.
  + Với những đơn hàng cần xóa thì quản trị viên bấm vào xóa đơn hàng.
* Quản lý hóa đơn
  + Quản trị viên có thể xem các hóa đơn có trong hệ thống.
* Quản lý bảo hành
  + Quản trị viên có thể lập phiếu bảo hành cho chiếc xe được bán ra
  + Quản trị viên có thể sửa thông tin bảo hành hoặc xóa phiếu bảo hành
* Báo cáo – thống kê
  + Báo cáo số lượng xe nhập, số lượng xe được bán ra và doanh thu của cửa hàng trong một tháng.
* Đăng nhập
  + Quản trị viên cần đăng nhập bằng tài khoản admin để sử dụng hệ thống.
* Đối với người dùng
* Xem danh sách hãng xe.
* Xem danh sách xe.
* Xem chi tiết xe.
* Thêm xe vào danh sách yêu thích (Yêu cầu người dùng đăng nhập).
* Xem thông tin xe đã xem trước đó (Yêu cầu người dùng đăng nhập).
* Đánh giá xe đã mua.
* Xem thông tin đơn hàng đã mua.
* Liên hệ đặt hàng.
* Đặt hàng
* Xem thông tin đơn hàng đã mua
* Huỷ liên hệ đặt hàng.
* Đăng nhập.
* Đăng ký

### Yêu cầu phi chức năng

* Hiệu năng sử dụng cao
* Thời gian thực hiện nhanh chóng
* Bảo mật thông tin tốt
* Giao diện thân thiện, dễ sử dụng, tính ổn định cao.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Biểu đồ usecase

### Biểu đồ



Hình 3‑1: Biểu đồ UseCase

### Đặc tả usecase

* Đăng nhập

Bảng 3‑1: Đặc tả usecase đăng nhập

|  |  |
| --- | --- |
| Tên Usecase | Đăng nhập |
| Actor | Quản trị viên hệ thống, Khách hàng, Nhân viên cửa hàng |
| Tiền điều kiện | Đã được cung cấp tài khoản |
| Hậu điều kiện | Đăng nhập được vào hệ thống và sử dụng các chức năng |
| Chuỗi sự kiện chính:   1. Actor chọn chức năng đăng nhập 2. Hệ thống hiển thị form đăng nhập và yêu cầu Actor nhập tên tài khoản và mật khẩu 3. Actor nhập tên tài khoản và mật khẩu sau đó ấn đăng nhập 4. Hệ thống kiểm tra thông tin Actor vừa nhập và xác nhận thông tin hợp lệ 5. Quản trị thoát khỏi chức năng đăng nhập | |
| Ngoại lệ:   1. Actor nhập thông tin sai hệ thống trả về thông báo. Actor nhập lại thông tin hoặc kích nút “Hủy” để thoát khỏi chức năng. 2. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Đăng ký

Bảng 3‑2: Đặc tả usecase đăng ký

|  |  |
| --- | --- |
| Tên Usecase | Đăng ký |
| Actor | Khách hàng |
| Tiền điều kiện | Không |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách kích nút “Đăng ký” ở trang chủ. Hệ thống sẽ hiển thị form Đăng ký lên màn hình. 2. Khách hàng nhập đầy đủ thông tin theo hướng dẫn, sau đó kích vào nút “Đăng ký”. Hệ thống kiểm tra và lưu thông tin khách hàng vào CSDL. | |
| Ngoại lệ:   1. Tại bước 2 trong sự kiện chính, nếu khách hàng không nhập đầy đủ thông tin, hệ thống thông báo thiếu thông tin cần nhập và yêu cầu khách hàng nhập. 2. Tại bước 2 trong sự kiện chính, tên tài khoản nhập trùng với tên tài khoản đã tồn tại, hệ thống sẽ đưa ra thông báo và yêu cầu khách hàng nhập lại. 3. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Xem danh sách xe

Bảng 3‑3: Đặc tả usecase xem danh sách xe

|  |  |
| --- | --- |
| Tên Usecase | Xem danh sách xe |
| Actor | Khách hàng |
| Tiền điều kiện | Không |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào nút “Danh sách xe”. Hệ thống hiển thị thông tin danh sách các xe gồm tên xe, hình ảnh, giá. 2. Khách hàng tìm và xem mẫu xe muốn xem. Use case kết thúc. | |
| Ngoại lệ:   1. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Xem chi tiết xe

Bảng 3‑4: Đặc tả usecase xem chi tiết xe

|  |  |
| --- | --- |
| Tên Usecase | Xem chi tiết xe |
| Actor | Khách hàng |
| Tiền điều kiện | Không |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào tên xe hoặc ảnh xe trên màn hình. Hệ thống hiển thị thông tin chi tiết xem lên màn hình gồm tên xe, hình ảnh, giá, mô tả, màu sắc. 2. Khách hàng xem chi tiết xe và có thể thực hiện thêm liên hệ đặt hàng. Use case kết thúc. | |
| Ngoại lệ:   1. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Xem thông tin đơn hàng đã mua

Bảng 3‑5: Đặc tả usecase xem thông tin đơn hàng đã mua

|  |  |
| --- | --- |
| Tên Usecase | Xem thông tin đơn hàng đã mua |
| Actor | Khách hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào nút xem thông tin đơn hàng đã mua trên màn hình. Hệ thống lấy thông tin lịch hẹn và hiển thị danh sách lịch hẹn trên màn hình. 2. . Khách hàng kích vào một lịch hẹn để xem chi tiết lịch hẹn. Usecase kết thúc. | |
| Ngoại lệ:   1. Nếu khách hàng chưa đăng nhập thì hiển thị thông báo lỗi và usecase kết thúc. 2. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Đánh giá xe đã mua

Bảng 3‑6: Đặc tả usecase đánh giá xe đã mua

|  |  |
| --- | --- |
| Tên Usecase | Đánh giá xe đã mua |
| Actor | Khách hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào nút thêm đánh giá xe đã mua trên màn hình. 2. Khách hàng nhập đánh giá và nhấn “Đánh giá”. Hệ thống sẽ thêm đánh giá vào cơ sở dữ liệu và thiển thị thông báo thành công. | |
| Ngoại lệ:   1. Nếu người dùng chưa đăng nhập thì hiển thị thông báo chưa đăng nhập. 2. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Thêm xe vào danh sách yêu thích

Bảng 3‑7: Đặc tả usecase thêm xe vào danh sách yêu thích

|  |  |
| --- | --- |
| Tên Usecase | Thêm xe vào danh sách yêu thích |
| Actor | Khách hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào nút trái tim trên thông tin xe. 2. Hệ thống sẽ thêm xe vào danh sách yêu thích của tài khoản đang đăng nhập và thiển thị thông báo thành công. | |
| Ngoại lệ:   1. Nếu người dùng chưa đăng nhập thì hiển thị thông báo chưa đăng nhập và usecase kết thúc. 2. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Liên hệ đặt hàng

Bảng 3‑8: Đặc tả usecase liên hệ đặt hàng

|  |  |
| --- | --- |
| Tên Usecase | Liên hệ đặt hàng |
| Actor | Khách hàng |
| Tiền điều kiện | Không |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi khách hàng kích vào nút liên hệ đặt hàng trong màn chi tiết xe. 2. Khách hàng nhập thông tin vào form và kích vào nút đặt hàng 3. Hệ thống sẽ thêm thông tin liên hệ đặt hàng và thiển thị thông báo thành công. | |
| Ngoại lệ:   1. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Quản lý hãng xe

Bảng 3‑9: Đặc tả usecase quản lý hãng xe

|  |  |
| --- | --- |
| Tên use case | Quản lý hãng xe |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đăng nhập vào hệ thống |
| Hậu điều kiện | Thông tin hãng được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:   1. Xem danh sách hãng 2. Quản trị viên kích chọn “Hãng Xe” trên thanh menu. 3. Hệ thống hiển thị danh sách tất cả các hãng xe lên màn hình. 4. Thêm hãng 5. Quản trị viên chọn chức năng thêm hãng. Hệ thống hiển thị form thêm hãng và yêu cầu quản trị viên đăng nhập thông tin hãng. 6. Quản trị viên nhập thông tin về hãng mới và nhấn lưu. Hệ thống kiểm tra thông tin hãng và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiện thị lại danh sách hãng. 7. Xóa hãng 8. Quản trị viên chọn chức năng xóa hãng 9. Quản trị viên xóa hãng. Hệ thống xõa hãng và hiển thị lại danh sách hãng. 10. Sửa hãng 11. Quản trị viên chọn chức năng sửa hãng. Hệ thống hiển thị thông tin hãng cũ và yêu cầu quản trị viên đăng nhập thông tin hãng cần sửa. 12. Quản trị viên nhập thông tin hãng mới và nhấn lưu. Hệ thống kiểm tra thông tin hãng và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiển thị lại danh sách hãng. | |
| Ngoại lệ:   1. Tại bước 2b, 4b trong chuỗi sự kiện khi người quản trị nhập thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích chọn “Hủy bỏ” để kết thúc. 2. Tại bước 2b, 4b trong luồng cơ bản khi người quản lý kích nút “Hủy bỏ”. Hệ thống bỏ qua thao tác và hiện thị danh sách các hãng. 3. Tại bước 3b trong luồng cơ bản khi người quản lý kích vào nút “Không đồng ý”. Hệ thống bỏ qua thao tác xóa và hiện thị danh sách các hãng. 4. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Quản lý thông tin xe

Bảng 3‑10: Đặc tả usecase quản lý thông tin xe

|  |  |
| --- | --- |
| Tên use case | Quản lý xe |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đăng nhập vào hệ thống |
| Hậu điều kiện | Thông tin xe được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:   1. Xem danh sách xe 2. Quản trị viên kích chọn “Xe” trên thanh menu. 3. Hệ thống hiển thị danh sách tất cả các xe lên màn hình. 4. Thêm xe 5. Quản trị viên chọn chức năng thêm xe. Hệ thống hiển thị form thêm xe và yêu cầu quản trị viên đăng nhập thông tin xe. 6. Quản trị viên nhập thông tin về xe mới và nhấn lưu. Hệ thống kiểm tra thông tin xe và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiện thị lại danh sách xe. 7. Xóa xe 8. Quản trị viên chọn chức năng xóa xe 9. Quản trị viên xóa xe. Hệ thống xõa xe và hiển thị lại danh sách xe. 10. Sửa xe 11. Quản trị viên chọn chức năng sửa xe. Hệ thống hiển thị thông tin xe cũ và yêu cầu quản trị viên đăng nhập thông tin xe cần sửa. 12. Quản trị viên nhập thông tin xe mới và nhấn lưu. Hệ thống kiểm tra thông tin xe và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiển thị lại danh sách xe. | |
| Ngoại lệ:   1. Tại bước 2b, 4b trong chuỗi sự kiện khi người quản trị nhập thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích chọn “Hủy bỏ” để kết thúc. 2. Tại bước 2b, 4b trong luồng cơ bản khi người quản lý kích nút “Hủy bỏ”. Hệ thống bỏ qua thao tác và hiện thị danh sách các xe. 3. Tại bước 3b trong luồng cơ bản khi người quản lý kích vào nút “Không đồng ý”. Hệ thống bỏ qua thao tác xóa và hiện thị danh sách các xe. 4. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Quản lý tài khoản

Bảng 3‑11: Đặc tả usecase quản lý tài kh oản

|  |  |
| --- | --- |
| Tên use case | Quản lý tài khoản |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đăng nhập vào hệ thống |
| Hậu điều kiện | Thông tin tài khoản được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:   1. Xem danh sách tài khoản 2. Quản trị viên kích chọn “Tài Khoản” trên thanh menu. 3. Hệ thống hiển thị danh sách tất cả các hãng xe lên màn hình. 4. Tạo tài khoản 5. Quản trị viên chọn chức năng “Tạo tài khoản”. Hệ thống hiển thị form thêm tài khoản và yêu cầu quản trị viên đăng nhập thông tin hãng. 6. Quản trị viên nhập thông tin về tài khoản mới và nhấn tạo. Hệ thống kiểm tra thông tin tài khoản và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiện thị lại danh sách tài khoản. 7. Xóa tài khoản 8. Quản trị viên chọn chức năng xóa tài khoản 9. Quản trị viên xóa tài khoản. Hệ thống xõa tài khoản và hiển thị lại danh sách tài khoản. 10. Sửa tài khoản 11. Quản trị viên chọn chức năng sửa tài khoản. Hệ thống hiển thị thông tin tài khoản cũ và yêu cầu quản trị viên đăng nhập thông tin tài khoản cần sửa. 12. Quản trị viên nhập thông tin tài khoản mới và nhấn lưu. Hệ thống kiểm tra thông tin hãng và xác nhận thông tin hợp lệ. Hệ thống thông báo đã nhập thành công và hiển thị lại danh sách tài khoản. | |
| Ngoại lệ:   1. Tại bước 2b, 4b trong chuỗi sự kiện khi người quản trị nhập thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích chọn “Hủy bỏ” để kết thúc. 2. Tại bước 2b, 4b trong luồng cơ bản khi người quản lý kích nút “Hủy bỏ”. Hệ thống bỏ qua thao tác và hiện thị danh sách các tài khoản. 3. Tại bước 3b trong luồng cơ bản khi người quản lý kích vào nút “Không đồng ý”. Hệ thống bỏ qua thao tác xóa và hiện thị danh sách các tài khoản. 4. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Quản lý nhập kho

Bảng 3‑12: Đặc tả usecase quản lý nhập kho

|  |  |
| --- | --- |
| Tên Usecase | Quản lý nhập kho |
| Actor | Nhân viên cửa hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Không |
| Chuỗi sự kiện chính:   1. Use case bắt đầu khi nhân viên cửa hàng kích vào nút quản lý nhập kho. 2. Nhân viên cửa hàng nhập thông tin vào form và kích vào nút tạo phiếu nhập. 3. Hệ thống sẽ thêm thông tin phiếu nhập kho và thiển thị thông báo thành công. | |
| Ngoại lệ:   1. Tại bất kì thời điểm nào trong quá trình thực hiện usecace nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc. | |

* Quản lý đơn hàng

Bảng 3‑13: Đặc tả usecase quản lý đơn hàng

|  |  |
| --- | --- |
| Tên usecase | Quản lý đơn hàng |
| Actor | Nhân viên cửa hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Thông tin đơn hàng được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:   1. Xem danh sách đơn hàng 2. Nhân viên cửa hàng kích chọn “Đơn hàng” trên thanh menu 3. Hệ thống hiển thị danh sách đơn hàng lên màn hình. 4. Tạo đơn hàng 5. Nhân viên cửa hàng kích chọn “Tạo đơn hàng” trên thanh menu. 6. Hệ thống hiển thị ra bảng đơn hàng và hiển thị lên màn hình. 7. Nhân viên cửa hàng nhập thông tin và kích chọn “Xuất đơn” để hoàn thành. 8. Xóa đơn hàng: 9. Nhân viên cửa hàng chọn chức năng xóa đơn hàng. 10. Nhân viên cửa hàng xóa đơn hàng. Hệ thống xóa đơn hàng và hiện thị lại danh sách đơn hàng. | |
| Ngoại lệ:   1. Tại bước 1b,2b,3b trong luồng cơ bản khi người quản lý kích vào nút “Hủy bỏ”. Hệ thống bỏ qua thao tác hiển thị danh sách, xóa và tạo các đơn hàng. 2. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Quản lý hóa đơn

Bảng 3‑14: Đặc tả usecase quản lý hóa đơn

|  |  |
| --- | --- |
| Tên use case | Quản lý hóa đơn |
| Actor | Nhân viên cửa hàng |
| Tiền điều kiện | Đăng nhập |
| Hậu điều kiện | Không có |
| Chuỗi sự kiện chính:   1. Xem hóa đơn: 2. Nhân viên cửa hàng kích chọn “Hóa Đơn” trên thanh menu. 3. Hệ thống hiển thị danh sách hóa đơn lên màn hình. | |
| Ngoại lệ:   1. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Quản lý bảo hành

Bảng 3‑15: Đặc tả usecase quản lý bảo hành

|  |  |
| --- | --- |
| Tên use case | Quản lý bảo hành |
| Actor | Nhân viên cửa hàng |
| Tiền điều kiện | Nhân viên cửa hàng đăng nhập vào hệ thống |
| Hậu điều kiện | Thông tin báo cáo được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:  1. Thêm thông tin bảo hành  a. Nhân viên cửa hàng kích chọn “Tạo phiếu bảo hành” trên màn hình  b. Nhân viên cửa hàng nhập thông tin bảo hành sau đó kích nút tạo.  c. Hệ thống nhập thông tin bảo hành vào cơ sở dữ liệu.  2. Cập nhật thông tin bảo hành.  a. Nhân viên cửa hàng tìm thông tin phiếu bảo hành cần sửa.  b. Nhân viên cửa hàng nhập thông tin cần sửa vào phiếu bảo hành, sau đó kích nút lưu. Hệ thống lưu thông tin nhân viên vừa nhập vào cơ sở dữ liệu. | |
| Ngoại lệ:  1. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

* Báo cáo thống kê

Bảng 3‑16: Đặc tả usecase báo cáo thống kê

|  |  |
| --- | --- |
| Tên use case | Báo cáo thông kê |
| Actor | Nhân viên cửa hàng |
| Tiền điều kiện | Nhân viên cửa hàng đăng nhập vào hệ thống |
| Hậu điều kiện | Thông tin báo cáo được cập nhật vào cơ sở dữ liệu |
| Chuỗi sự kiện chính:  1. Xem báo cáo thông kê  a. Nhân viên cửa hàng kích chọn “Báo cáo thông kê” trên thanh menu  b.Hệ thống hiển thị thông tin bảng báo cáo lên màn hình gồm tên xe, hãng xe, ngày đầu, ngày cuối, số lượng nhập, số lượng bán, doanh thu.  c. Nhân viên cửa hàng nhập dữ liệu vào bảng báo cáo và lưu lại để thống kê số lượng nhập , bán và doanh thu trong tháng vừa qua. | |
| Ngoại lệ:  1. Tại bước 1b nếu quản trị nhập sai thông tin thì có thể nhập lại để tiếp tục hoặc kích nút “Hủy bỏ” để kết thúc.  2. Tại bất kỳ thời điểm nào trong qua trình thực hiện usecase nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |

## Biểu đồ trình tự

* Usecase đăng nhập



Hình 3‑2: Biếu đồ tuần tự usecase đăng nhập

* Usecase đăng ký



Hình 3‑3: Biểu đồ tuần tự usecase đăng ký

* Usecase xem danh sách xe



Hình 3‑4: Biểu đồ tuần tự xem danh sách xe

* Usecase xem chi tiết xe



Hình 3‑5: Biểu đồ tuần tự usecase xem chi tiết xe máy

* Usecase xem thông tin đơn hàng đã mua



Hình 3‑6: Biểu đồ tuần tự usecase xem thông tin đơn hàng đã mua

* Usecase đánh giá xe đã mua



Hình 3‑7: Biểu đồ tuần tự usecase đánh giá xe đã mua

* Usecase thêm xe vào danh sách yêu thích



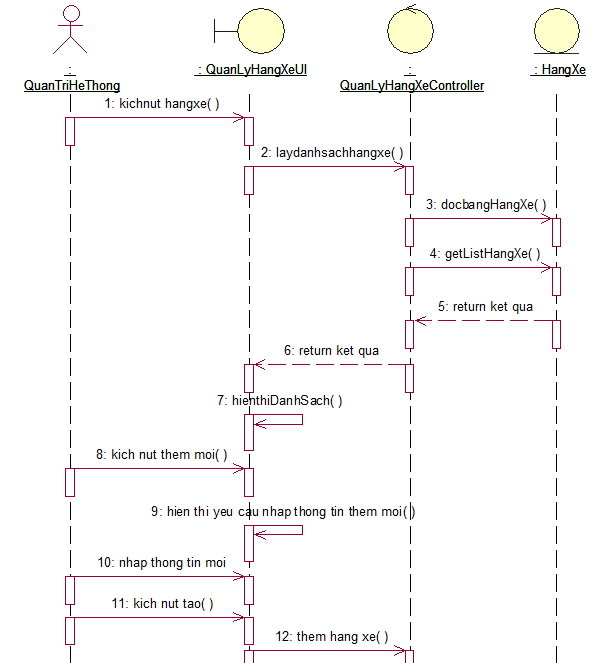
Hình 3‑8: Biểu đồ tuần tự usecase thêm xe vào danh sách yêu thích

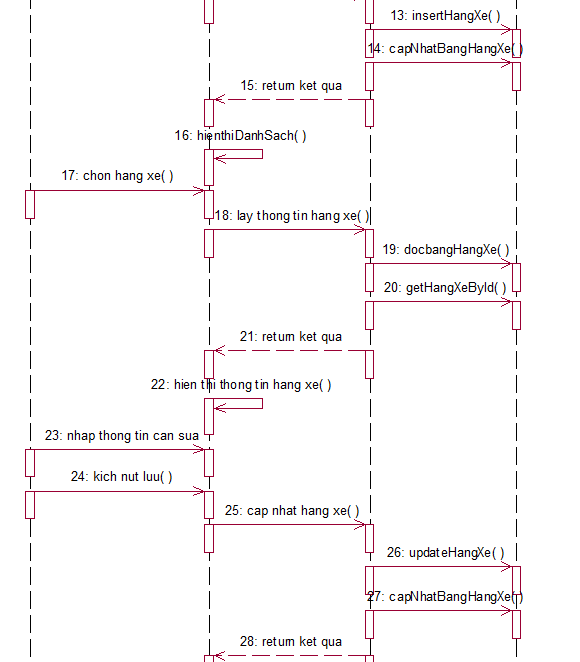
* Usecase liên hệ đặt hàng

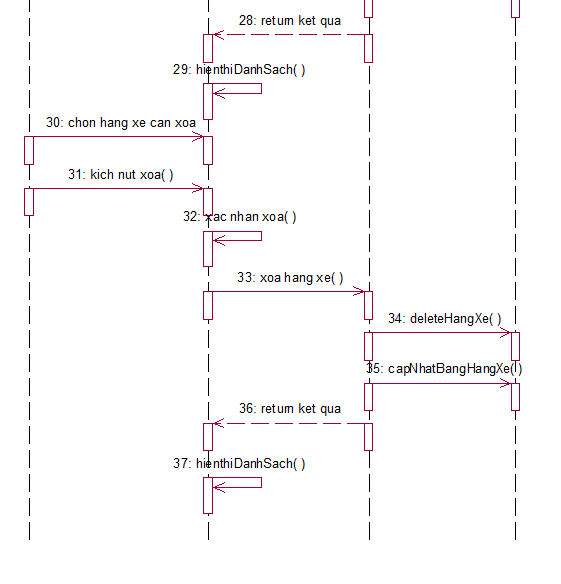


Hình 3‑9: Biểu đồ tuần tự usecase liên hệ đặt hàng

* Usecase quản lý hãng xe

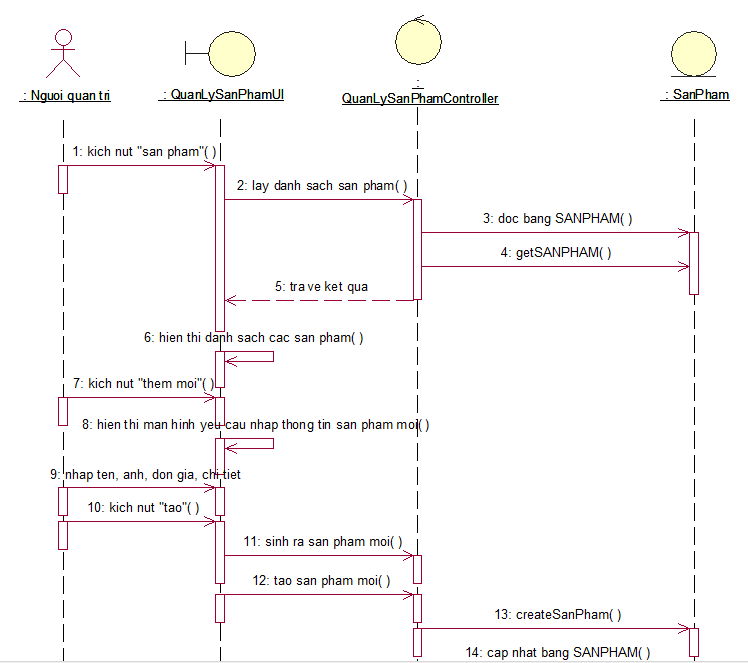


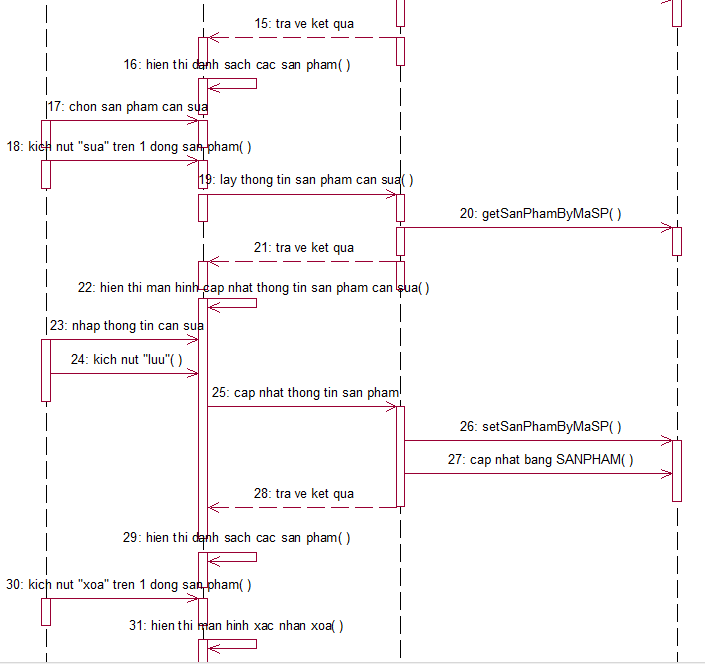


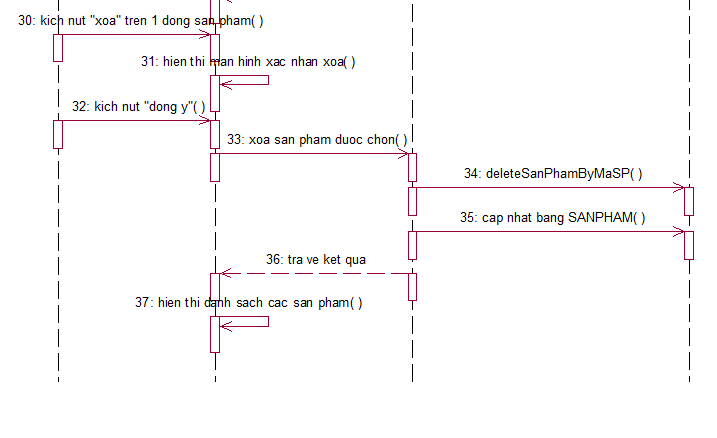


Hình 3‑10: Biểu đồ tuần tự usecase quản lý hãng xe

* Usecase quản lý thông tin xe







Hình 3‑11: Biểu đồ tuần tự usecase quản lý xe

* Usecase quản lý tài khoản



Hình 3‑12: Biểu đồ tuần tự usecase quản lý tài khoản

* Usecase quản lý nhập kho



Hình 3‑13: Biểu đồ tuần tự usecase quản lý nhập kho

* Usecase quản lý đơn hàng



Hình 3‑14: Biểu đồ tuần tự usecase quản lý đơn hàng

* Usecase quản lý bảo hành



Hình 3‑15: Biểu đồ tuần tự usecase quản lý bảo hành

* Usecase báo cáo thống kê



Hình 3‑16: Biểu đồ tuần tự usecase thống kê

## Biểu đồ lớp lĩnh vực

* Usecase đăng nhập



Hình 3‑17: Biểu đồ lớp lĩnh vực usecase đăng nhập

* Usecase đăng ký



Hình 3‑18: Biểu đồ lớp lĩnh vực usecase đăng ký

* Usecase xem danh sách xe



Hình 3‑19: Biểu đồ lớp lĩnh vực usecase xem danh sách xe

* Usecase xem chi tiết xe



Hình 3‑20: Biểu đồ lớp lĩnh vực usecase xem chi tiết xe

* Usecase xem thông tin đơn hàng đã mua



Hình 3‑21: Biểu đồ lớp lĩnh vực usecase xem thông tin đơn hàng đã mua

* Usecase đánh giá xe đã mua



Hình 3‑22: Biểu đồ lớp lĩnh vực usecase đánh giá xe đã mua

* Usecase thêm xe vào danh sách yêu thích



Hình 3‑23: Biểu đồ lớp lĩnh vực usecase thêm xe vào danh sách yêu thích

* Usecase liên hệ đặt hàng



Hình 3‑24: Biểu đồ lớp lĩnh vực usecase liên hệ đặt hàng

* Usecase quản lý hãng xe



Hình 3‑25: Biểu đồ lớp lĩnh vực usecase quản lý hãng xe

* Usecase quản lý thông tin xe



Hình 3‑26: Biểu đồ lớp lĩnh vực usecase quản lý thông tin xe

* Usecase quản lý tài khoản



Hình 3‑27: Biểu đồ lớp lĩnh vực usecase quản lý tài khoản

* Usecase quản lý nhập kho



Hình 3‑28: Biểu đồ lớp lĩnh vực usecase quản lý nhập kho

* Usecase quản lý đơn hàng



Hình 3‑29: Biểu đồ lớp lĩnh vực usecase quản lý đơn hàng

* Usecase quản lý bảo hành



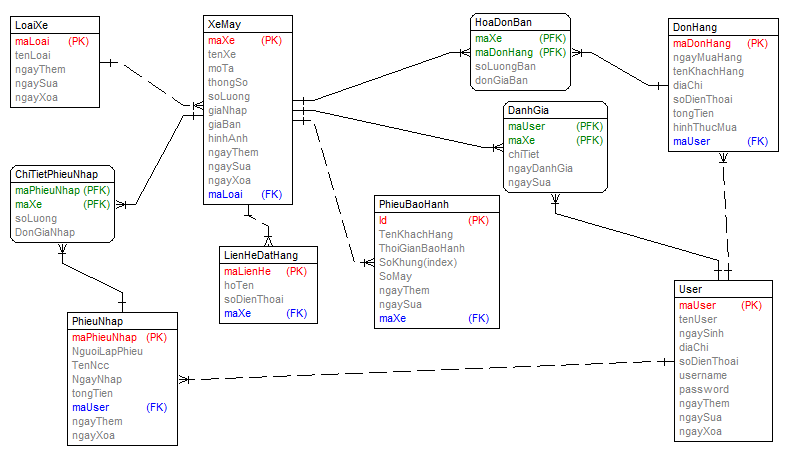
Hình 3‑30: Biểu đồ lớp lĩnh vực usecase quản lý bảo hành

* Usecase báo cáo thống kê



Hình 3‑31: Biểu đồ lớp lĩnh vực usecase thống kê

## Thiết kế cơ sở dữ liệu



Hình 3‑32: Cơ sở dữ liệu

### Bảng LoaiXe

Bảng 3‑17: Chi tiết bảng LoaiXe

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maLoai | Mã loại xe | int |  |
| 2 |  |  | tenLoai | Tên loại xe | Nvarchar(50) |  |

### Bảng XeMay

Bảng 3‑18: Chi tiết bảng XeMay

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maXe | Mã xe máy | Int |  |
| 2 |  | x | maLoaiXe | Mã loại xe | Nvarchar(20) |  |
| 3 |  |  | tenXe | Tên xe máy | Nvarchar(50) |  |
| 4 |  |  | moTa | Mô tả | NText |  |
| 5 |  |  | thongSo | Thông số | NText |  |
| 6 |  |  | soLuong | Số lương | Int |  |
| 7 |  |  | giaNhap | Giá nhập | Decimal |  |
| 8 |  |  | giaBan | Giá bán | Decimal |  |
| 9 |  |  | hinhAnh | Hình ảnh | Nvarchar(100) |  |
| 10 |  |  | ngayThem | Ngày thêm | Datetime |  |
| 11 |  |  | ngaySua | Ngày sửa | Datetime |  |
| 12 |  |  | ngayXoa | Ngày xóa | Datetime |  |

### Bảng PhieuNhap

Bảng 3‑19: Chi tiết bảng PhieuNhap

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maPhieuNhap | Mã phiếu nhập | int |  |
| 2 |  |  | nguoiLap | Người lập phiếu | Nvarchar(50) |  |
| 3 |  |  | tenNCC | Tên nhà cung cấp | Nvarchar(50) |  |
| 4 |  |  | tongTien | Tổng tiền | Decimal |  |
| 5 |  |  | ngayThem | Ngày thêm | Datetime |  |
| 6 |  |  | ngaySua | Ngày sửa | Datetime |  |
| 7 |  |  | ngayXoa | Ngày xóa | Datetime |  |
| 8 |  | x | maUser | Mã user | int |  |

### Bảng ChiTietPhieuNhap

Bảng 3‑20: Chi tiết bảng ChiTietPhieuNhap

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x | x | maPhieuNhap | Mã loại xe | int |  |
| 2 | x | x | maXe | Tên loại xe | Nvarchar(50) |  |
|  |  |  | soLuong | Số lượng | int |  |
|  |  |  | donGiaNhap | Đơn giá nhập | decimal |  |

### Bảng LienHeDatHang

Bảng 3‑21: Chi tiết bảng LienHeDatHang

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maLienHe | Mã loại xe | int |  |
| 2 |  |  | hoTen | Tên loại xe | Nvarchar(50) |  |
| 3 |  |  | soDienThoai | Số điện thoại | Varchar(10) |  |
| 4 |  | x | maXe | Mã xe máy | int |  |

### Bảng PhieuBaoHanh

Bảng 3‑22: Chi tiết bảng PhieuBaoHanh

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maPhieu | Mã loại xe | int |  |
| 2 |  |  | tenKhach | Tên khách hàng | Nvarchar(50) |  |
| 3 |  |  | thoiGianBaoHanh | Thời gian bảo hành |  |  |
| 4 |  |  | soKhung | Số khung |  |  |
| 5 |  |  | soMay | Số máy |  |  |
| 6 |  |  | ngayThem | Ngày tạo |  |  |
| 7 |  |  | ngaySua | Ngày sửa |  |  |
| 8 |  | x | maXe | Mã xe máy |  |  |

### Bảng User

Bảng 3‑23: Chi tiết bảng User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maUser | Mã loại xe | int |  |
| 2 |  |  | tenUser | Họ tên | Nvarchar(50) |  |
| 4 |  |  | ngaySinh | Ngày sinh | Datetime |  |
| 5 |  |  | diaChi | Địa chỉ | Nvarchar(100) |  |
| 6 |  |  | soDienThoai | Số điện thoại | Varchar(10) |  |
| 7 |  |  | username | Tên đăng nhập | Varchar(50) |  |
| 8 |  |  | Password | Mật khẩu | Varchar(100) |  |
| 9 |  |  | ngayThem | Ngày thêm | Datetime |  |
| 10 |  |  | ngaySua | Ngày sửa | Datetime |  |
| 11 |  |  | ngayXoa | Ngày xóa | Datetime |  |

### Bảng DanhGia

Bảng 3‑24: Chi tiết bảng DanhGia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x | x | maUser | Mã user | int |  |
| 2 | x | x | maXe | Mã xe | int |  |
| 3 |  |  | chiTiet | Chi tiết đánh giá | NText |  |
| 4 |  |  | ngayDanhGia | Ngày đánh giá | Datetime |  |
| 5 |  |  | ngaySua | Ngày sửa | Datetime |  |
| 6 |  |  | ngayXoa | Ngày xóa | Datetime |  |

### Bảng DonHang

Bảng 3‑25: Chi tiết bảng DonHang

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x |  | maDonHang | Mã loại xe | int |  |
| 2 |  |  | ngayMuaHang | Tên loại xe | Datetime |  |
| 3 |  |  | tenKhachHang | Tên khách hàng | Nvarchar(50) |  |
| 4 |  |  | diaChi | Địa chỉ | Nvarchar(100) |  |
| 5 |  |  | soDienThoai | Số điện thoại | Nvarchar(10) |  |
| 6 |  |  | tongTien | Tổng tiền | Decimal |  |
| 7 |  |  | ngayThem | Ngày thêm | Datetime |  |
| 8 |  |  | ngayXoa | Ngày xóa |  |  |
| 9 |  | x | maUser | Mã user | int |  |

### Bảng HoaDonBanHang

Bảng 3‑26: Chi tiết bảng HoaDonBanHang

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Khóa chính** | **Khóa ngoại** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | x | x | maXe | Mã xe | int |  |
| 2 | x | x | maDonHang | Mã đơn hàng | int |  |
| 3 |  |  | soLuong | Số lượng bán | int |  |
| 4 |  |  | donGia | Đơn giá | Decimal |  |

# CÀI ĐẶT CHƯƠNG TRÌNH

## Tổ chức source code

### Cài đặt Nuget Package

Các package được sử dụng trong đồ án:

* AutoMapper.Extensions.Microsoft.DependencyInjection
* Microsoft.AspNetCore.Identity.EntityFrameworkCore
* Microsoft.AspNetCore.Identity.UI
* Microsoft.AspNetCore.Mvc.Abstractions
* Microsoft.EntityFrameworkCore
* Microsoft.EntityFrameworkCore.Tools
* Microsoft.NET.Test.Sdk
* Microsoft.VisualStudio.Web.CodeGeneration.Design

### Cài đặt Generic Repository Pattern

Source code được tổ chức theo mô hình MVC với cấu trúc Generic Repository Parten nhằm:

* Tách lớp truy cập cơ sở dữ liệu (data access layer) và lớp business (business logic layer): Mô hình này giúp cho việc phân chia và quản lý code trở nên dễ dàng hơn, từ đó giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn.
* Tăng tính tái sử dụng của code: Generic Repository Interface được thiết kế để trừu tượng hóa các thao tác truy vấn dữ liệu cơ bản, từ đó giúp cho code có thể tái sử dụng được trong nhiều ứng dụng khác nhau.
* Tính linh hoạt và dễ dàng mở rộng: Khi cần thay đổi cơ sở dữ liệu, ta chỉ cần tạo ra một class mới tương ứng với cơ sở dữ liệu mới, và thay đổi dependency injection để sử dụng class mới này. Việc thay đổi cơ sở dữ liệu trở nên dễ dàng hơn mà không làm ảnh hưởng đến business logic layer.
* Dễ dàng kiểm thử: Với mô hình Generic Repository Pattern, ta có thể dễ dàng kiểm thử các phương thức truy vấn cơ sở dữ liệu mà không cần phải kết nối đến cơ sở dữ liệu thực sự.
* Giảm thiểu lỗi: Việc sử dụng mô hình Generic Repository Pattern giúp cho code trở nên trừu tượng hóa hơn, giảm thiểu các lỗi do việc truy vấn cơ sở dữ liệu sai hoặc không phù hợp.
* Cài đặt Generic Repository Class:

public class GenericRepository<TEntity> where TEntity : class

{

internal SchoolContext context;

internal DbSet<TEntity> dbSet;

public GenericRepository(SchoolContext context)

{

this.context = context;

this.dbSet = context.Set<TEntity>();

}

public virtual IEnumerable<TEntity> Get(

Expression<Func<TEntity, bool>> filter = null,

Func<IQueryable<TEntity>, IOrderedQueryable<TEntity>> orderBy = null,

string includeProperties = "")

{

IQueryable<TEntity> query = dbSet;

if (filter != null)

{

query = query.Where(filter);

}

foreach (var includeProperty in includeProperties.Split

(new char[] { ',' }, StringSplitOptions.RemoveEmptyEntries))

{

query = query.Include(includeProperty);

}

if (orderBy != null)

{

return orderBy(query).ToList();

}

else

{

return query.ToList();

}

}

public virtual TEntity GetByID(object id)

{

return dbSet.Find(id);

}

public virtual void Insert(TEntity entity)

{

dbSet.Add(entity);

}

public virtual void Delete(object id)

{

TEntity entityToDelete = dbSet.Find(id);

Delete(entityToDelete);

}

public virtual void Delete(TEntity entityToDelete)

{

if (context.Entry(entityToDelete).State == EntityState.Detached)

{

dbSet.Attach(entityToDelete);

}

dbSet.Remove(entityToDelete);

}

public virtual void Update(TEntity entityToUpdate)

{

dbSet.Attach(entityToUpdate);

context.Entry(entityToUpdate).State = EntityState.Modified;

}

}

* Cài đặt UnitOfWork Class

public class UnitOfWork : IUnitOfWork

{

private readonly AppDbContext \_dbContext;

public UnitOfWork(AppDbContext dbContext)

{

\_dbContext = dbContext;

}

public IGenericRepository<T> GenericRepository<T>() where T : class

{

return new GenericRepository<T>(\_dbContext);

}

public void Save()

{

\_dbContext.SaveChanges();

}

}

* Các service được viết trong tầng business logic đều Inject UnitOfWork để sử dụng

### Cài đặt EntityFramework Core (Mô hình Code First)

* Cài đặt AppDbContext Class

public class AppDbContext : IdentityDbContext<AppUser>

{

public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)

{

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

base.OnConfiguring(optionsBuilder);

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

modelBuilder.ApplyConfiguration(new MediasConfiguration());

modelBuilder.ApplyConfiguration(new ProductConfiguration());

modelBuilder.ApplyConfiguration(new ReviewsConfiguration());

modelBuilder.ApplyConfiguration(new ShopConfiguration());

modelBuilder.ApplyConfiguration(new UserConfiguration());

modelBuilder.ApplyConfiguration(new PhieuNhapConfiguration());

modelBuilder.ApplyConfiguration(new ChiTietPhieuNhapConfiguration());

modelBuilder.ApplyConfiguration(new DatLichConfiguration());

modelBuilder.ApplyConfiguration(new CartConfiguration());

modelBuilder.ApplyConfiguration(new OrderConfiguration());

modelBuilder.ApplyConfiguration(new OrderDetailConfiguration());

modelBuilder.ApplyConfiguration(new GiaXeConfiguration());

}

public DbSet<AppUser> Users { get; set; }

public DbSet<Shop> Shops { get; set; }

public DbSet<Product> Products { get; set; }

public DbSet<Medias> Medias { get; set; }

public DbSet<Reviews> Reviews { get; set; }

public DbSet<ChiTietPhieuNhap> ChiTietPhieuNhaps { get; set; }

public DbSet<PhieuNhap> PhieuNhaps { get; set; }

public DbSet<DatLich> DatLiches{ get; set; }

public DbSet<Cart> Carts { get; set; }

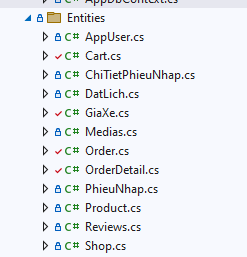
public DbSet<Order> Orders { get; set; }

public DbSet<OrderDetail> OrderDetails { get; set; }

public DbSet<GiaXe> GiaXes { get; set; }

}

* Cài đặt các entity



Hình 4‑1: Entity sử dụng trong project

* Tạo ra cơ sở dữ liệu trong MS SQL Server từ code
  + Dùng lệnh: Add-Migration <tên migration>
  + Sau đó dùng lệnh: Update-Database

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO