|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |
| --- | --- |

**BÁO CÁO ĐỒ ÁN**

**QUẢN LÝ BÁN XE Ô TÔ**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Tên học phần: **LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS**

Giảng viên hướng dẫn: **Trịnh Đồng Thạch Trúc**

Sinh viên thực hiện:

**Phan Tiến Dũng - 2280600418**

**Lê Việt Hoàng - 2280601037**

**Đậu Mai Xuân Phương - 2280602517**

Lớp: **22DTHD4**

TP. Hồ Chí Minh, 2025

**LỜI CẢM ƠN**

Trước hết, nhóm chúng em xin gửi lời cảm ơn chân thành đến **Thầy Trịnh Đồng Thạch Trúc** là người đã tận tình hướng dẫn và cung cấp những kiến thức quý báu trong suốt quá trình thực hiện đồ án. Với sự nhiệt huyết và góp ý của Cô đã giúp nhóm chúng em hoàn thiện đồ án một cách tốt nhất và đúng thời hạn.

Chúng em cũng xin gửi lời cảm ơn đến các Thầy/Cô trong **Khoa Công nghệ thông tin - Bộ môn Lập Trình Trên Môi Trường Windows** đã thuyết giảng và trang bị cho chúng em những kiến thức cơ bản và chuyên sâu, nền tảng vững chắc giúp nhóm hoàn thành đề tài này.

Bên cạnh đó, nhóm xin chân thành cảm ơn sự ủng hộ từ bạn bè và các thành viên ở các nhóm đã luôn bên cạnh, tạo động lực để chúng mình vượt qua khó khăn trong quá trình nghiên cứu và thực hiện đồ án.

Tuy đã cố gắng hết sức, nhưng chắc chắn đồ án vẫn còn có nhiều thiếu sót. Nhóm rất mong nhận được sự đóng góp ý kiến, lời nhận xét từ Thầy/Cô và các bạn để đồ án này được hoàn thiện hơn.

Một lần nữa, **Nhóm PHD** chúng em xin chân thành cảm ơn!

Phan Tiến Dũng

Lê Việt Hoàng

Đậu Mai Xuân Phương

**MỤC LỤC**

[**Chương 1: TỔNG QUAN 1**](#_biamwe26dbq4)

[1.1 Giới thiệu 1](#_stl6lgnhl5ge)

[1.2 Nhiệm vụ đồ án 1](#_s2el5s660ua2)

[1.2.1 Mục tiêu nghiên cứu: 1](#_8qdoab5ych0k)

[1.2.2 Đối tượng và phạm vi giới hạn: 2](#_ci0loemzjt64)

[1.3 Cấu trúc đồ án 2](#_w1baa328of1k)

[**Chương 2: CƠ SỞ LÝ THUYẾT 3**](#_2et92p0)

[2.1 C# 3](#_tyjcwt)

[2.1.1 C# là gì? 3](#_6mlgjwovzgzz)

[2.1.2 Đặc điểm nổi bật của C# 3](#_de3s0rojd62)

[2.1.3 Ứng dụng của C#: 4](#_de3s0rojd62)

[2.2 .NET Framework: 5](#_xdf8ulaih1rp)

[2.2.1 Giới thiệu 5](#_7tpfhkk90a8o)

[2.2.2 Thành phần chính của .NET Framework 5](#_y76fslwybqz2)

[2.2.3 Ứng dụng thực tế của .NET Framework 7](#_3xdkyvcr2orc)

[2.3 Entity Framework: 8](#_xdf8ulaih1rp)

[2.3.1 Entity Framework (EF): 8](#_3xdkyvcr2orc)

[2.3.2 Các Tính Năng Chính của Entity Framework 8](#_4g6z0thxazcl)

[2.3.3 Phương pháp tiếp cận phát triển với Entity Framework: 9](#_wd4vrt5imgod)

[2.4 LINQ 11](#_4dgj5qfelh59)

[2.5 WINDOWS FORMS 12](#_ge51wml8fqee)

[2.5.1 Đặc điểm chính của Windows Forms: 12](#_szbmvr3m31rl)

[2.5.2 Ưu điểm của Windows Forms: 13](#_szbmvr3m31rl)

[2.5.3 Nhược điểm của Windows Forms: 13](#_szbmvr3m31rl)

[2.5.4 Ứng dụng: 14](#_kywhsjc3akmp)

[2.6 OfficeOpenXml 14](#_hptk3dvw4wy3)

[2.6.1 Đặc điểm chính của OfficeOpenXml: 14](#_rp1n30mklbzy)

[2.6.2 Các tính năng nổi bật: 15](#_9u2bvmqggpwr)

[2.6.3 Ưu điểm của OfficeOpenXml: 15](#_jkrudd4l2znz)

[2.6.3 Nhược điểm của OfficeOpenXml: 15](#_j36n4bcq7a6f)

[**Chương 3: KẾT QUẢ THỰC NGHIỆM 17**](#_2bn6wsx)

[Chương 4: KẾT LUẬN VÀ KIẾN NGHỊ 17](#_bgiaxwk7b5dx)

[Kiến nghị 17](#_nwxwtziv5kpw)

# **Chương 1: TỔNG QUAN**

## **1.1 Giới thiệu**

Trong thời đại công nghệ 4.0, việc ứng dụng công nghệ thông tin vào các lĩnh vực kinh doanh ngày càng trở nên quan trọng. Đồ án "Quản lý gara bán xe ô tô" được thực hiện nhằm xây dựng một hệ thống quản lý hiệu quả cho các gara ô tô, giúp quản lý thông tin xe, khách hàng và quá trình bán hàng một cách tiện lợi và nhanh chóng.

Với nhu cầu ngày càng tăng về số lượng khách hàng và xe, việc phát triển một hệ thống quản lý gara ô tô trên nền tảng WinForms là cần thiết. WinForms, một phần của .NET Framework, cung cấp môi trường phát triển ứng dụng mạnh mẽ, dễ sử dụng và phù hợp với nhiều đối tượng người dùng.

## **1.2 Nhiệm vụ đồ án**

**Tính cấp thiết và lý do hình thành đề tài:** Hiện nay, nhiều gara ô tô vẫn quản lý thông tin bằng phương pháp thủ công hoặc sử dụng các phần mềm lỗi thời, không còn đáp ứng được nhu cầu thực tế. Điều này dẫn đến khó khăn trong việc theo dõi thông tin khách hàng, xe, và dịch vụ, đồng thời làm giảm hiệu suất quản lý. Vì vậy, việc thiết kế một hệ thống quản lý gara ô tô hiện đại là cần thiết để tối ưu hóa quy trình vận hành, tăng độ chính xác và giảm thiểu sai sót.

**Ý nghĩa khoa học và thực tiễn:** Về tài cung cấp một giải pháp áp dụng công nghệ .NET để xây dựng một hệ thống quản lý chuyên nghiệp trong lĩnh vực gara ô tô, góp phần minh họa khả năng của công nghệ trong việc giải quyết các vấn đề thực tiễn. Hệ thống giúp cải thiện hiệu quả quản lý hoạt động bán hàng, dịch vụ và nhân sự trong gara, giảm thiểu công việc thủ công tốn thời gian, đồng thời nâng cao trải nghiệm của khách hàng thông qua quy trình làm việc nhanh chóng và chuyên nghiệp.

### *1.2.1 Mục tiêu nghiên cứu:*

Xây dựng hệ thống quản lý gara ô tô đáp ứng các chức năng cơ bản như quản lý xe, khách hàng, và dịch vụ sửa chữa bảo dưỡng.

Tích hợp giao diện thân thiện, dễ sử dụng cho người quản lý và độc giả.

### *1.2.2 Đối tượng và phạm vi giới hạn:*

Đối tượng nghiên cứu: Các phương pháp và công cụ để phát triển hệ thống quản lý gara ô tô.

Phạm vi: Phần mềm sẽ chỉ tập trung vào các chức năng cơ bản, phù hợp với các gara vừa và nhỏ, không tích hợp các công nghệ phức tạp như AI hoặc Cloud.

## **1.3 Cấu trúc đồ án**

Cấu trúc của đồ án gồm các chương sau:

**Chương 1: TỔNG QUAN**Giới thiệu đề tài, tóm tắt các lý thuyết và nghiên cứu liên quan. Trình bày nhiệm vụ, ý nghĩa, mục tiêu và cấu trúc của đồ án.

**Chương 2: CƠ SỞ LÝ THUYẾT**Ngôn ngữ lập trình và công nghệ sử dụng:

Giới thiệu về ngôn ngữ lập trình (C#, VB.NET, v.v.).

Công nghệ WinForms, cơ sở dữ liệu (SQL Server, MySQL, v.v.).

**Chương 3: KẾT QUẢ THỰC NGHIỆM**Trong quá trình thực hiện đề tài "Quản lý gara bán xe ô tô", nhóm đã tiến hành các công việc thực nghiệm bao gồm:

* **Phân tích yêu cầu:** Xác định các chức năng cốt lõi của hệ thống như quản lý thông tin xe, khách hàng, nhân viên và doanh thu.
* **Thiết kế hệ thống:** Lập hồ sơ thiết kế sử dụng ngôn ngữ mô hình hóa UML, bao gồm sơ đồ lớp, sơ đồ tuần tự và sơ đồ trạng thái để mô tả các đối tượng và tương tác trong hệ thống.
* **Lập trình và cài đặt**: Sử dụng ngôn ngữ C# và công nghệ WinForms để phát triển ứng dụng, kết nối với cơ sở dữ liệu SQL Server để lưu trữ dữ liệu.

**Chương 4: KẾT LUẬN VÀ KIẾN NGHỊ**

Đồ án "Quản lý gara bán xe ô tô" đã đạt được những kết quả đáng ghi nhận trong việc phát triển một hệ thống quản lý hiệu quả. Hệ thống không chỉ giúp cải thiện quy trình làm việc tại gara mà còn tối ưu hóa việc quản lý thông tin về xe, khách hàng và doanh thu. Qua quá trình thực nghiệm, hệ thống đã chứng minh tính khả thi và hiệu quả trong việc đáp ứng nhu cầu quản lý của gara bán xe ô tô.

Nhìn chung, đồ án "Quản lý gara bán xe ô tô" đã thành công trong việc tạo ra một hệ thống quản lý hiệu quả, góp phần nâng cao chất lượng dịch vụ và sự hài lòng của khách hàng. Hy vọng rằng với những đề xuất và kiến nghị trên, hệ thống sẽ tiếp tục phát triển và hoàn thiện hơn trong tương lai.

# **Chương 2: CƠ SỞ LÝ THUYẾT**

## **2.1 C#**

### *2.1.1 C# là gì?*

C# (C-sharp) là một ngôn ngữ lập trình hiện đại, hướng đối tượng, linh hoạt, mạnh mẽ, cho phép phát triển ứng dụng trên nhiều lĩnh vực và nền tảng được phát triển bởi Microsoft như một phần của sáng kiến .NET.

### *2.1.2 Đặc điểm nổi bật của C#*

* **Hướng đối tượng (OOP)**: C# hỗ trợ các nguyên tắc OOP như kế thừa, đóng gói, đa hình và trừu tượng hóa, cho phép xây dựng các mô hình phần mềm phức tạp và có cấu trúc
* **Ngôn ngữ mạnh kiểu (Strongly Typed)**: C# yêu cầu khai báo kiểu dữ liệu rõ ràng như kiểm tra giới hạn mảng, kiểu dữ liệu nullable, xử lý ngoại lệ…, giúp giảm thiểu lỗi liên quan đến kiểu dữ liệu trong quá trình phát triển phần mềm. Ngôn ngữ an toàn: C# được thiết kế với mục đích bảo mật cao và quản lý bộ nhớ tốt, giúp tránh các lỗi phổ biến trong lập trình.
* **Ngôn ngữ tích hợp (Integrated Language Support)**: C# dễ dàng tích hợp với các công nghệ và dịch vụ của Microsoft như ASP.NET, Azure, SQL Server, giúp phát triển ứng dụng web, dịch vụ đám mây và quản lý dữ liệu hiệu quả.
* **Đa nền tảng**: Với .NET Core và .NET 5 trở lên, C# có thể chạy trên các hệ điều hành Windows, macOS, và Linux.
* **Hỗ trợ lập trình bất đồng bộ (Asynchronous Programming)**: C# cung cấp các cấu trúc như async và await, giúp dễ dàng viết mã bất đồng bộ, cải thiện hiệu suất và khả năng phản hồi của ứng dụng.
* **Generic Programming**: C# hỗ trợ generic, cho phép tạo ra các kiểu dữ liệu và phương thức tổng quát, tăng tính linh hoạt và khả năng sử dụng lại mã.
* **Hỗ trợ LINQ (Language Integrated Query)**: LINQ cho phép truy vấn và thao tác dữ liệu một cách mạnh mẽ và linh hoạt ngay trong mã C#.
* **Quản lý bộ nhớ**: C# sử dụng garbage collection tự động thu gom rác, quản lý bộ nhớ tự động.

### *2.1.3 Ứng dụng của C#:*

* **Phát triển Web Backend:** C# được sử dụng rộng rãi trong phát triển các ứng dụng web phía backend, đặc biệt với sự hỗ trợ của các framework và các engine: ASP.NET, MVC, ASP.NET core, Web API,Graph API,gPRC, Blazor server,Uno platform, Mono
* **Phát triển Web Frontend: C#** hỗ trợ xây dựng giao diện người dùng web bằng công nghệ: Blazor WebAssembly,Uno platform
* **Phát triển Ứng dụng Desktop:** C# là một lựa chọn lý tưởng để xây dựng các ứng dụng desktop hiện đại: Winform, WPF, UWP,WinUI, Mono, Uno, MAUI,Blazor desktop…
* **Phát triển Game Đa Nền Tảng (2D và 3D):** C# là sự lựa chọn phổ biến cho việc phát triển game với các engine:Unity , Monogame, Godot, Stride, CryEngine, Flax Engine, Evergine, NeoAxis, XNA ..
* **Thực Tế Ảo (VR), Thực Tế Tăng Cường (AR), và Thực Tế Hỗn Hợp (MR):** C# được sử dụng trong việc phát triển các ứng dụng thực tế ảo và thực tế tăng cường với công nghệ tiên tiến: HoloLens,Unity,CryEngine, Oculus quest..
* **Phát triển Ứng dụng Đồ Họa 2D và 3D Đa Nền Tảng:** C# cung cấp các thư viện mạnh mẽ để phát triển đồ họa: SkiaSharp, OpenTK, Vulkan.NET, Veldrid, Silk.NET…
* **Phát triển Ứng dụng Di động:** C# hỗ trợ phát triển ứng dụng di động đa nền tảng cũng như ứng dụng native cho iOS và Android: Xamarin,MAUI,Uno platform
* **Phát triển Ứng dụng Đám Mây:** C# là công cụ chính để xây dựng các dịch vụ cloud hiện đại: Azure ,AWS,Google Cloud…
* **Machine learning và Artificial Intelligence (AI):** Ngôn ngữ này được ứng dụng trong machine learning và AI với các thư viện: ML.NET, TensorFlow, csiSharp..
* **Blockchain :** C# cũng có những ứng dụng trong công nghệ blockchain: NEO, Stratis
* **Microservices và Containers:** Ngôn ngữ này được sử dụng để xây dựng các kiến trúc ứng dụng microservices trong môi trường container hóa.
* **Internet of Things (IoT):** C# hỗ trợ phát triển các ứng dụng IoT và mạng 5G với khả năng kết nối qua các thiết bị cảm biến hoặc hạ tầng công nghiệp.
* **Hệ Thống Nhúng:** C# giúp phát triển các hệ thống nhúng và ứng dụng trên các thiết bị nhỏ gọn: Raspberry Pi, PLC

## **2.2 .NET Framework:**

### *2.2.1 Giới thiệu*

.NET Framework là một nền tảng phát triển phần mềm được Microsoft giới thiệu lần đầu vào năm 2002. Đây là nền tảng hỗ trợ các nhà phát triển ứng dụng trong việc xây dựng, triển khai và chạy các ứng dụng trên hệ điều hành Windows.

.NET Framework là một khung lập trình (framework) bao gồm một tập hợp lớn các thư viện, một ngôn ngữ trung gian (CIL - Common Intermediate Language) và môi trường thực thi (CLR - Common Language Runtime).

Mục Tiêu: Cung cấp một nền tảng thống nhất cho việc phát triển ứng dụng, giúp đơn giản hóa quá trình xây dựng, triển khai và tích hợp các ứng dụng trên nhiều nền tảng khác nhau, giúp tăng hiệu quả và giảm thiểu rủi ro trong quá trình phát triển phần mềm.

### *2.2.2 Thành phần chính của .NET Framework*

2.2.2.1 Common Language Runtime (CLR)

* Là môi trường thực thi ứng dụng của .NET Framework.
* Cung cấp các dịch vụ như: quản lý memory (bộ nhớ), xử lý ngoại lệ (exception handling), thu gom rác (garbage collection), tối ưu hóa hiệu suất.
* CLR giúp cho ứng dụng viết bằng các ngôn ngữ khác nhau (như C#, VB.NET) có thể tương tác lẫn nhau nhờ cơ chế Ngôn ngữ Trung gian Chung (Common Intermediate Language - CIL).

2.2.2.2. Framework Class Library (FCL)

* Là một tập hợp các thư viện và lớp được tích hợp sẵn trong .NET Framework.
* FCL cung cấp các chức năng cơ bản để phát triển phần mềm, bao gồm:
* Giao tiếp với cơ sở dữ liệu.
* Xử lý tệp tin và luồng dữ liệu.
* Xử lý chuỗi, bộ sưu tập (collections), giao tiếp mạng, bảo mật, v.v.

2.2.2.3. ASP.NET

* Là thành phần chuyên hỗ trợ phát triển ứng dụng Web.
* Cho phép xây dựng ứng dụng Web động, dịch vụ web (Web Services) và các ứng dụng Web-based khác.

2.2.2.4. Windows Forms và WPF

* **Windows Forms**: Hỗ trợ xây dựng giao diện người dùng (UI) cho ứng dụng desktop trên Windows.
* **WPF (Windows Presentation Foundation)**: Cung cấp các công cụ đồ họa tiên tiến để xây dựng các ứng dụng UI phong phú về hình ảnh và hiệu ứng.

2.2.2.5. ADO.NET

Là công nghệ giúp .NET Framework giao tiếp với cơ sở dữ liệu (SQL Server, Oracle, MySQL, v.v.).

Cung cấp khả năng kết nối với nguồn dữ liệu và xử lý dữ liệu theo cách hiệu quả.

**Ưu điểm:**

* **Hỗ trợ đa ngôn ngữ**: Hỗ trợ đa ngôn ngữ lập trình như C#, VB.NET, F# thông qua cơ chế CLR.
* **Tính tương thích đa nền tảng**: .NET Framework có thể chạy trên nhiều hệ điều hành: Windows, Linux, macOS, mang đến sự linh hoạt trong công việc phát triển và triển khai ứng dụng.
* **Cộng đồng lớn**: .NET Framework có rất nhiều cộng đồng hỗ trợ tích cực cho các nhà phát triển, cộng đồng tạo ra nhiều nguồn tài liệu, tài nguyên như thư viện, công cụ.
* **Bảo mật**: .NET Framework có nhiều tính năng bảo mật như bảo mật mã truy cập và chữ ký điện tử, mang đến sự hỗ trợ bảo vệ ứng dụng từ các tác nhân.
* **Hiệu suất cao**: .NET Framework bao gồm một tập hợp các thư viện và công cụ lớn có sẵn, giúp nhà phát triển tiết kiệm thời gian và nâng cao năng suất.

**Nhược điểm:**

* **Chỉ tối ưu trên Windows**: .NET Framework ban đầu được thiết kế chủ yếu cho Windows, khiến nó kém linh hoạt trên các nền tảng khác ngoài hệ sinh thái của Microsoft.
* **Phụ thuộc vào môi trường thực thi**: Yêu cầu máy tính cài đặt .NET Framework để chạy ứng dụng.
* **Dung lượng lớn:**.NET Framework có dung lượng cài đặt lớn, yêu cầu nhiều tài nguyên, có thể gây khó khăn trong việc triển khai ứng dụng trên các hệ thống có dung lượng hoặc băng thông hạn chế.
* **Giấy phép:** một số phiên bản .NET Framework yêu cầu giấy phép dẫn đến tăng chi phí phát triển ứng dụng.
* **Hiệu năng:** Mặc dù .NET Framework cung cấp hiệu năng tốt cho hầu hết các ứng dụng, nhưng nó có thể không phải là lựa chọn tốt nhất cho các ứng dụng yêu cầu hiệu năng cao hoặc tiếp cận tới phần cứng hoặc thuật toán phức tạp.
* **Phiên bản không tương thích**: Trong một số trường hợp, các ứng dụng viết trên các phiên bản .NET Framework khác nhau có thể không tương thích.

### *2.2.3 Ứng dụng thực tế của .NET Framework*

* **Phát triển ứng dụng doanh nghiệp**: Các ứng dụng quản lý, hệ thống ERP và CRM.
* **Xây dựng ứng dụng web và dịch vụ web**: Xây dựng web từ công nghệ ASP.NET.
* **Ứng dụng desktop**: Như phần mềm kế toán, quản lý kho, ứng dụng văn phòng làm việc…
* **Ứng dụng mobile:** Dùng công nghệ Xamarin để phát triển ứng dụng di động cho các nền tảng Android, IOS, Windows Phone
* **Ứng dụng đồ họa và games**: Cung cấp nền tảng cho các ứng dụng yêu cầu độ nhạy cao về khả năng xử lý đồ họa.

## **2.3 Entity Framework:**

### *2.3.1 Entity Framework (EF):*

Entity Framework là framework ánh xạ quan hệ đối tượng (Object-Relational Mapper) mã nguồn mở cho các ứng dụng .NET được phát triển bởi Microsoft. Nó làm tăng năng suất làm việc cho các nhà phát triển, cho phép làm việc với dữ liệu bằng cách sử dụng đối tượng của lớp chuyên biệt (domain-specific classes) mà không cần tập trung thao tác trực tiếp vào bảng, cột và giảm thiểu các câu lệnh truy cập dữ liệu được sử dụng để tương tác với cơ sở dữ liệu. Phiên bản mới nhất thời điểm hiện tại và Entity Framework 6.

### *2.3.2 Các Tính Năng Chính của Entity Framework*

* EF không phụ thuộc vào nền tảng.
* EF sử dụng truy vấn LinQ để thao tác dữ liệu trong cơ sở dữ liệu thay vì truy vấn SQL.
* EF theo dõi các giá trị đã được thay đổi của các thuộc tính của các thực thể.
* EF cũng lưu lại những thay đổi được thực hiện bằng các thao tác chèn, xóa hoặc cập nhật.
* EF cũng xử lý tính đồng thời để dữ liệu được ghi đè bởi người dùng và sẽ phản ánh khi người dùng khác truy xuất dữ liệu đó.
* EF cũng tự động xử lý việc quản lý giao dịch và cung cấp các tùy chọn tùy chỉnh cho việc quản lý giao dịch.
* EF cung cấp tính năng lưu trữ đệm, nghĩa là lưu trữ kết quả của các truy vấn thường dùng.
* EF cũng tuân theo một số quy ước lập trình nhất định nên theo mặc định nó sẽ cấu hình Mô hình EF.
* EF cũng cho phép cấu hình Mô hình EF bằng API lưu loát để ghi đè lên quy ước mặc định.
* Nếu bạn thực hiện bất kỳ thay đổi nào trong lược đồ cơ sở dữ liệu thì bạn có thể phản ánh những thay đổi đó trong mô hình EF bằng cách viết lệnh di chuyển trong CLI (Giao diện dòng lệnh).
* EF cũng hỗ trợ quy trình được lưu trữ.
* EF cũng hỗ trợ các truy vấn có tham số.

### *2.3.3 Phương pháp tiếp cận phát triển với Entity Framework:*

2.3.3.2. Database-First

* Mô tả
  + Bắt đầu với cơ sở dữ liệu hiện có.
  + Sử dụng Entity Framework để tạo mô hình dữ liệu (entities) và DBContext tự động dựa trên cấu trúc của cơ sở dữ liệu.
* Ưu điểm:
  + Nhanh chóng khi đã có cơ sở dữ liệu.
  + Phù hợp với cơ sở dữ liệu phức tạp hoặc đã được thiết kế chi tiết.
* Nhược điểm:
  + Khó khăn trong việc quản lý thay đổi khi cơ sở dữ liệu được cập nhật.
  + Ít linh hoạt trong việc thay đổi mô hình dữ liệu từ phía code.
* Phù hợp với:
  + Dự án có cơ sở dữ liệu đã tồn tại.
  + Khi cần làm việc trực tiếp với cấu trúc cơ sở dữ liệu hiện có.

2.3.3.2. Model-First

* Mô tả
  + Bắt đầu bằng cách thiết kế mô hình dữ liệu thông qua Entity Framework Designer trong Visual Studio.
  + Từ mô hình này, Framework sẽ tạo cơ sở dữ liệu tương ứng.
* Ưu điểm:
  + Trực quan, dễ dàng thiết kế và hình dung mô hình dữ liệu.
  + Kiểm soát tốt cấu trúc cơ sở dữ liệu từ mô hình.
* Nhược điểm:
  + Quy trình phức tạp hơn khi cập nhật mô hình và cơ sở dữ liệu.
  + Ít linh hoạt khi làm việc với code so với Code-First.
* Phù hợp với:
  + Dự án bắt đầu từ đầu và muốn thiết kế mô hình dữ liệu một cách trực quan.
  + Khi ưu tiên thiết kế mô hình trước khi triển khai code.

2.3.3.3. Code-First

* Mô tả
  + Bắt đầu bằng việc viết code để định nghĩa các lớp entities và DBContext.
  + Entity Framework sẽ tạo cơ sở dữ liệu dựa trên mô hình code này.
* Ưu điểm:
  + Linh hoạt cao, dễ dàng thay đổi và mở rộng mô hình dữ liệu thông qua code.
  + Hỗ trợ Migrations, giúp quản lý và cập nhật cơ sở dữ liệu khi mô hình thay đổi.
* Nhược điểm:
  + Yêu cầu kiến thức tốt về lập trình và mẫu thiết kế.
  + Ban ầu có thể tốn thời gian để thiết lập và cấu hình.
* Phù hợp với:
  + Dự án cần linh hoạt và thường xuyên thay đổi yêu cầu.
  + Khi muốn kiểm soát hoàn toàn mô hình dữ liệu từ code.

2.3.3.4. Code-First From Database

* Mô tả
  + Kết hợp giữa Code-First và Database-First.
  + Bắt đầu với cơ sở dữ liệu hiện có, sau đó sử dụng công cụ để sinh ra code (entities và DBContext) từ cơ sở dữ liệu.
  + Tiếp tục phát triển dựa trên phương pháp Code-First,sử dụng migrations cho quản lý thay đổi.
* Ưu điểm:
  + Tận dụng cơ sở dữ liệu hiện có mà không cần thiết kế lại.
  + Hưởng lợi từ tính linh hoạt và khả năng quản lý thay đổi của Code-First.
* Nhược điểm:
  + Code sinh ra có thể cần tùy chỉnh để phù hợp với tiêu chuẩn dự án.
  + Phụ thuộc vào công cụ hỗ trợ sinh code.
* Phù hợp với:
  + Dự án có cơ sở dữ liệu hiện có nhưng muốn chuyển sang quản lý mô hình dữ liệu bằng code.
  + Khi cần kết hợp lợi ích của cả Database-First và Code-First.

2.3.3.4. Lý do chọn Code-First From Database cho dự án

* Chúng tôi có sẵn một cơ sở dữ liệu, đã được thiết kế và triển khai. Chúng tôi muốn tận dụng nó để không cần làm lại một cái khác, tiết kiệm thời gian và công sức.
* Chúng tôi muốn tận dụng tính linh hoạt và khả năng quản lý thay đổi của phương pháp Code-First, đặc biệt là thông qua Migrations.
* Phương pháp này cho phép bắt đầu nhanh chóng với cơ sở dữ liệu hiện có đồng thời có khả năng kiểm soát và điều chỉnh mô hình dữ liệu thông qua code, giảm thiểu lỗi do nhập liệu thủ công.

## **2.4 LINQ**

LINQ (Language Integrated Query. Tên tiếng việt: Ngôn ngữ truy vấn tích hợp) là một tập hợp các công nghệ được xây dựng dựa trên việc tích hợp khả năng truy vấn trực tiếp vào ngôn ngữ C#.

Trước đây, các truy vấn dữ liệu thường được biểu diễn dưới dạng chuỗi ký tự đơn giản, không có kiểm tra kiểu dữ liệu tại thời điểm biên dịch cũng như không được hỗ trợ bởi các công cụ như IntelliSense. Hơn nữa, bạn phải học nhiều ngôn ngữ truy vấn khác nhau cho từng loại nguồn dữ liệu, chẳng hạn như SQL cho cơ sở dữ liệu, XPath cho tài liệu XML, hay các API riêng biệt cho các dịch vụ web. Với LINQ, truy vấn trở thành một cấu trúc ngôn ngữ chính thức, ngang hàng với các khái niệm như lớp, phương thức và sự kiện.

Khi viết truy vấn, phần "tích hợp ngôn ngữ" nổi bật nhất của LINQ chính là biểu thức truy vấn. Các biểu thức này được viết bằng cú pháp truy vấn khai báo, cho phép bạn thực hiện các thao tác như lọc, sắp xếp và nhóm dữ liệu từ các nguồn dữ liệu khác nhau chỉ với một lượng mã tối thiểu. Đặc biệt, bạn có thể sử dụng cùng một mẫu biểu thức truy vấn để truy vấn và chuyển đổi dữ liệu từ bất kỳ loại nguồn dữ liệu nào, từ cơ sở dữ liệu quan hệ đến tài liệu XML hay thậm chí là các dịch vụ web.

## **2.5 WINDOWS FORMS**

Windows Forms, thường được gọi là WinForms, là một framework đồ họa người dùng (GUI) được phát triển bởi Microsoft, là một phần của nền tảng .NET Framework. Nó cho phép các nhà phát triển tạo các ứng dụng desktop và hỗ trợ tính năng với giao diện người dùng một cách trực quan trên hệ điều hành Windows.

### *2.5.1 Đặc điểm chính của Windows Forms:*

* Giao diện đồ họa trực quan:
  + WinForms cung cấp các chức năng như: các điều khiển (controls), nút bấm (Button), hộp văn bản (TextBox), danh sách (ListBox), bảng (DataGridView), …., để tạo giao diện người dùng.
  + Các điều khiển này có thể được kéo và thả vào form thông qua công cụ Windows Forms Designer trong Visual Studio, giúp quá trình thiết kế giao diện trở nên trực quan và dễ dàng.
* Hỗ trợ đa ngôn ngữ:
  + WinForms được tích hợp với các ngôn ngữ lập trình .NET như C#, VB.NET, và F#, cho phép các nhà phát triển sử dụng ngôn ngữ họ quen thuộc để xây dựng ứng dụng.
* Mô hình sự kiện (Event-driven):
  + WinForms hoạt động dựa trên mô hình lập trình hướng sự kiện. Các sự kiện như nhấp chuột, nhập liệu, hoặc thay đổi giá trị có thể được xử lý thông qua các phương thức xử lý sự kiện (event handlers), cho phép ứng dụng phản hồi tương tác của người dùng một cách hiệu quả.
* Tích hợp với .NET Framework:
  + WinForms tận dụng các thư viện từ .NET Framework, bao gồm ADO.NET để tương tác với cơ sở dữ liệu, LINQ để truy vấn dữ liệu, và nhiều tính năng khác, giúp tăng tốc độ phát triển và chất lượng ứng dụng.
* Dễ học và sử dụng:
  + Với các công cụ trực quan và tài liệu phong phú, WinForms là một lựa chọn phổ biến cho các nhà phát triển mới bắt đầu với lập trình desktop trên Windows.
* Tương thích với các phiên bản Windows:
  + WinForms hoạt động trên nhiều phiên bản Windows, từ Windows XP đến Windows 10/11, đảm bảo phạm vi sử dụng rộng rãi.

### *2.5.2 Ưu điểm của Windows Forms:*

* **Phát triển nhanh chóng**: Các công cụ kéo thả và thiết kế trực quan giúp tạo ứng dụng nhanh chóng và hiệu quả.
* **Tài liệu và cộng đồng hỗ trợ lớn**: WinForms có một cộng đồng lớn và nhiều tài liệu hướng dẫn, giúp giải quyết vấn đề và học hỏi trở nên dễ dàng.
* **Ổn định và đáng tin cậy**: Được sử dụng rộng rãi trong nhiều năm, WinForms là một công nghệ ổn định và đã được kiểm chứng qua thời gian.

### *2.5.3 Nhược điểm của Windows Forms:*

* Giới hạn về giao diện hiện đại: WinForms không hỗ trợ các hiệu ứng đồ họa phức tạp và giao diện người dùng hiện đại như các framework mới hơn như WPF (Windows Presentation Foundation) hoặc UWP (Universal Windows Platform).
* Chỉ hỗ trợ Windows: WinForms chỉ chạy trên hệ điều hành Windows, không hỗ trợ đa nền tảng, điều này có thể hạn chế đối với các dự án cần triển khai trên nhiều nền tảng khác nhau.

#### *2.5.4 Ứng dụng:*

* Phát triển ứng dụng desktop đơn giản, công cụ quản lý nội bộ.
* Phù hợp khi cần tạo ứng dụng nhanh với giao diện cơ bản.

## **2.6 OfficeOpenXml**

OfficeOpenXml (thường được viết tắt là OpenXML) là một thư viện mã nguồn mở được sử dụng để tạo, đọc, chỉnh sửa và thao tác các tệp tin Microsoft Office (như Excel, Word, PowerPoint) ở định dạng Open XML (định dạng mặc định của các tệp Office từ phiên bản 2007 trở đi). Thư viện này được phát triển dựa trên tiêu chuẩn Open XML SDK của Microsoft và hỗ trợ làm việc với các tệp .xlsx, .docx, và .pptx.

### *2.6.1 Đặc điểm chính của OfficeOpenXml:*

* Hỗ trợ định dạng Open XML:
  + Làm việc với các tệp Excel, Word, và PowerPoint.
  + Đọc, ghi, và chỉnh sửa nội dung của các tệp này mà không cần cài đặt Microsoft Office.
* Dễ sử dụng:
  + Cung cấp API đơn giản và trực quan để thao tác với các tệp Office.
  + Hỗ trợ các tính năng phổ biến như tạo bảng, thêm biểu đồ, định dạng ô, và nhiều hơn nữa.
* Hiệu suất cao:
  + Xử lý các tệp lớn một cách hiệu quả nhờ cơ chế đọc/ghi tuần tự (streaming).
* Tích hợp với .NET:
  + Là một thư viện .NET, OfficeOpenXml có thể được sử dụng trong các ứng dụng C#, VB.NET, và các ngôn ngữ .NET khác.
* Mã nguồn mở:
  + OfficeOpenXml là một dự án mã nguồn mở, được phát triển và duy trì bởi cộng đồng.

### *2.6.2 Các tính năng nổi bật:*

* Tạo và chỉnh sửa tệp Excel:
  + Tạo bảng tính (worksheets), thêm dữ liệu, định dạng ô, và tạo biểu đồ.
  + Hỗ trợ các công thức Excel, định dạng điều kiện, và bảo vệ trang tính.
* Làm việc với tệp Word:
  + Tạo và chỉnh sửa tài liệu Word, thêm văn bản, hình ảnh, bảng biểu, và các phần tử khác.
* Xử lý tệp PowerPoint:
  + Tạo và chỉnh sửa các slide, thêm văn bản, hình ảnh, và hiệu ứng.
* Đọc và xuất dữ liệu:
  + Đọc dữ liệu từ các tệp Office và xuất dữ liệu sang các định dạng khác.

### *2.6.3 Ưu điểm của OfficeOpenXml:*

* Không cần cài đặt Microsoft Office: Bạn có thể làm việc với các tệp Office mà không cần cài đặt phần mềm Office trên máy.
* Linh hoạt: Hỗ trợ nhiều tính năng phức tạp như biểu đồ, công thức, và định dạng các điều kiện.
* Hiệu suất tốt: Xử lý các tệp lớn một cách hiệu quả.
* Mã nguồn mở: Miễn phí và có thể tùy chỉnh theo nhu cầu.

### *2.6.3 Nhược điểm của OfficeOpenXml:*

* Độ phức tạp: Một số tính năng nâng cao có thể yêu cầu kiến thức sâu về cấu trúc tệp Open XML.
* Hỗ trợ hạn chế: Một số tính năng của Microsoft Office có thể không được hỗ trợ đầy đủ.

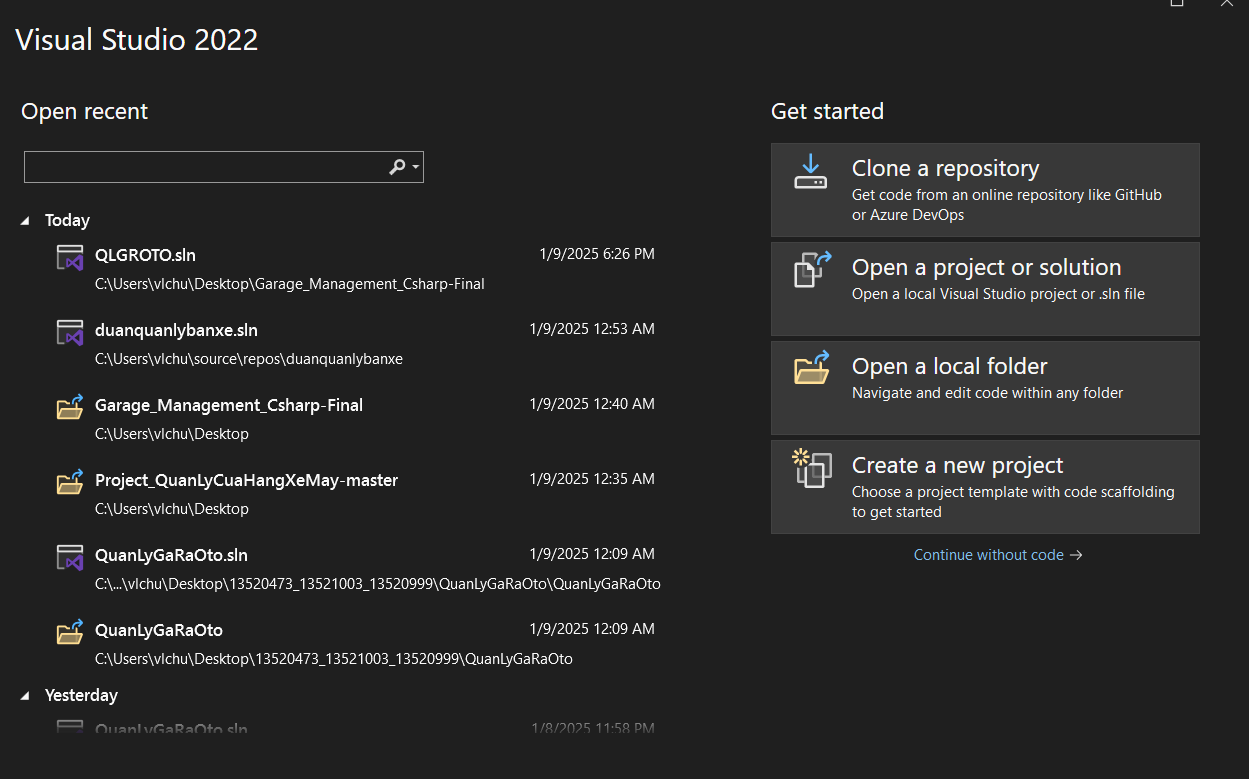
### 

### 

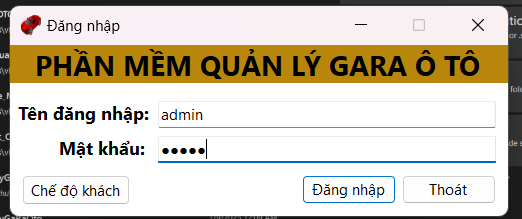
# **Chương 3: KẾT QUẢ THỰC NGHIỆM**

### 

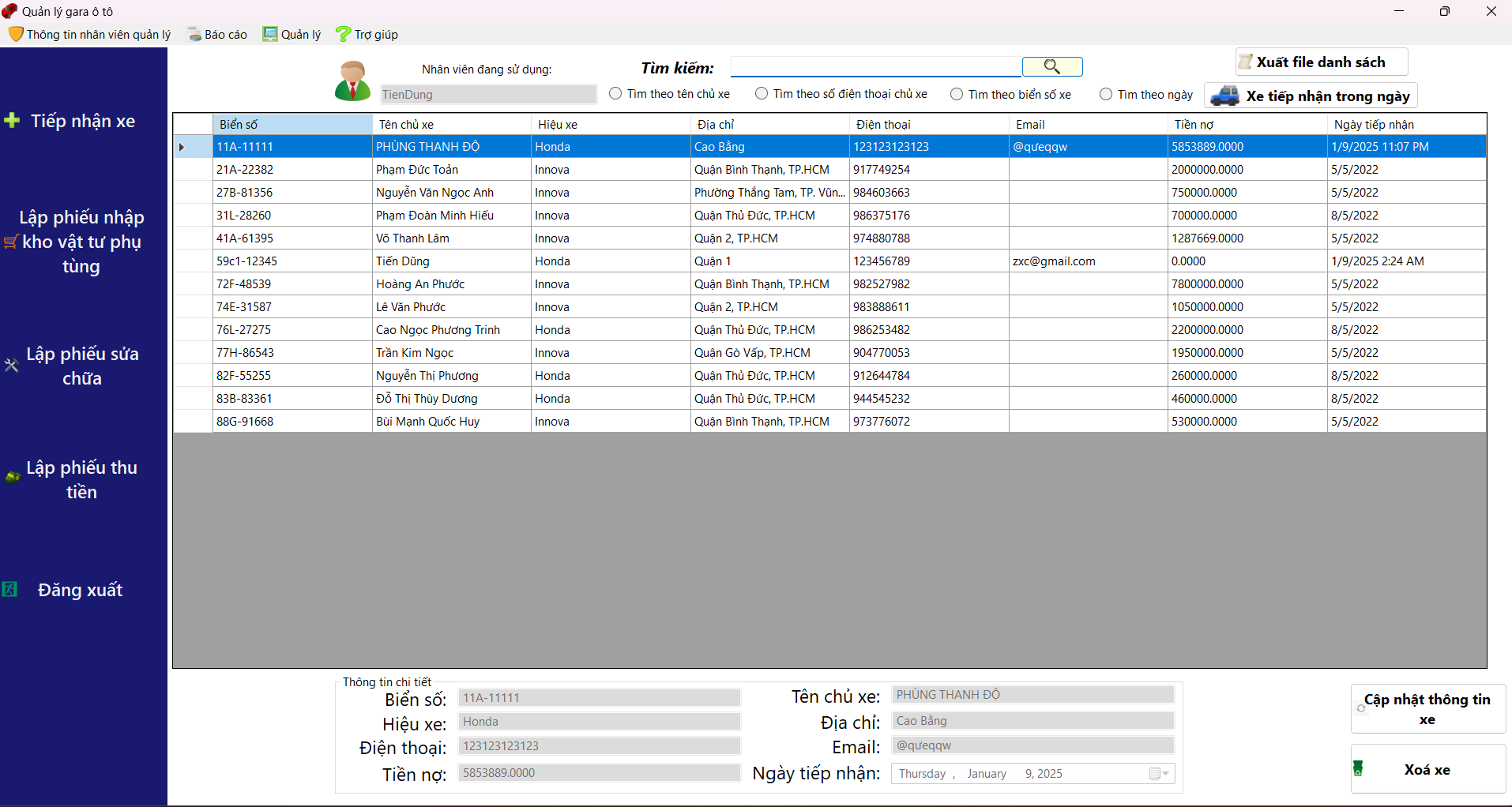
*Hình 3.1: Cơ sở dữ liệu của đồ án*

**

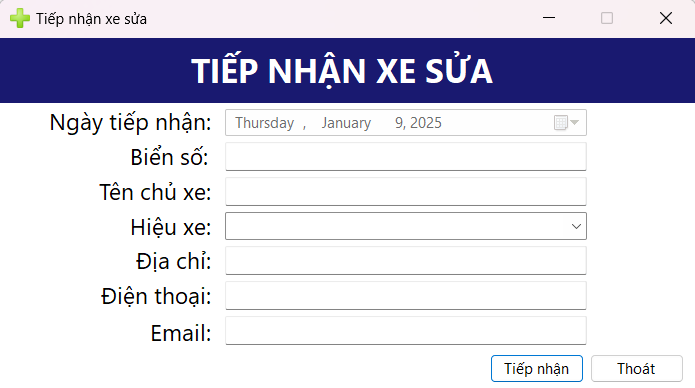
*Hình 3.2: Sử dụng Visual Studio để xây dựng đồ án*

**

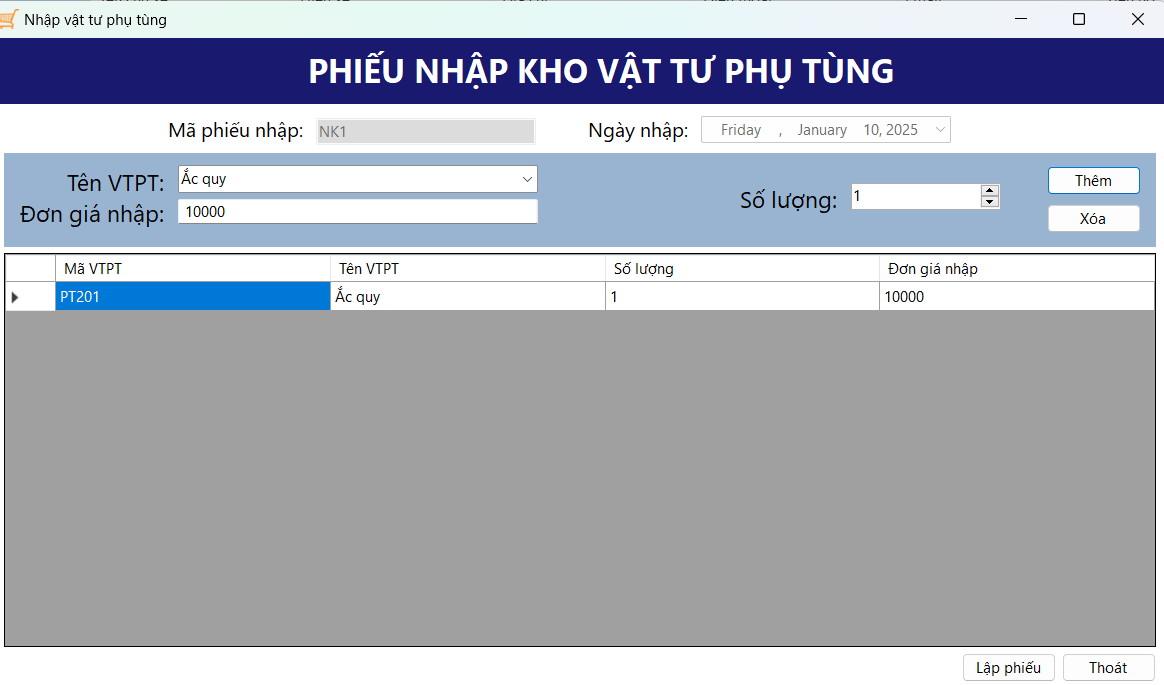
*hình 3.3 : Giao diện đăng nhập*

**

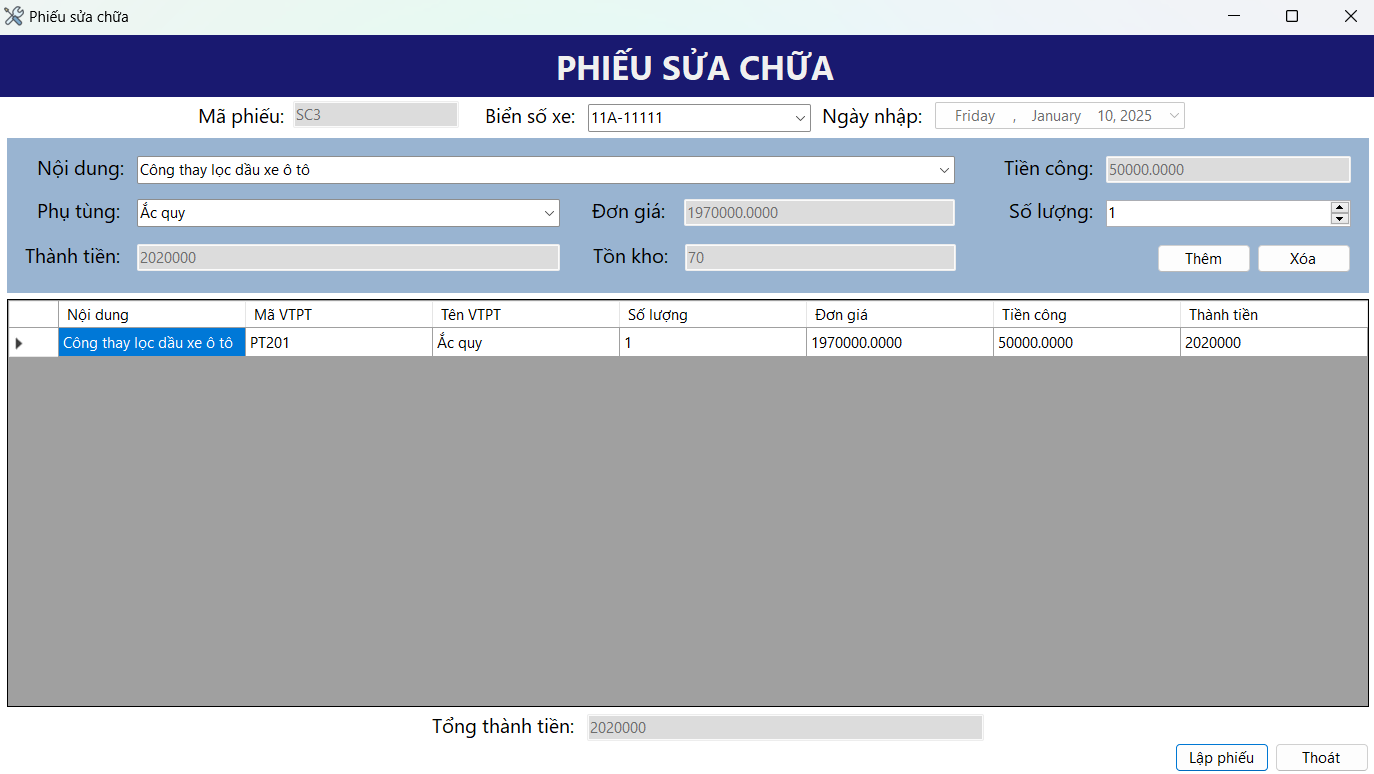
*hình 3.4 : Giao diện màn hình chính*

**

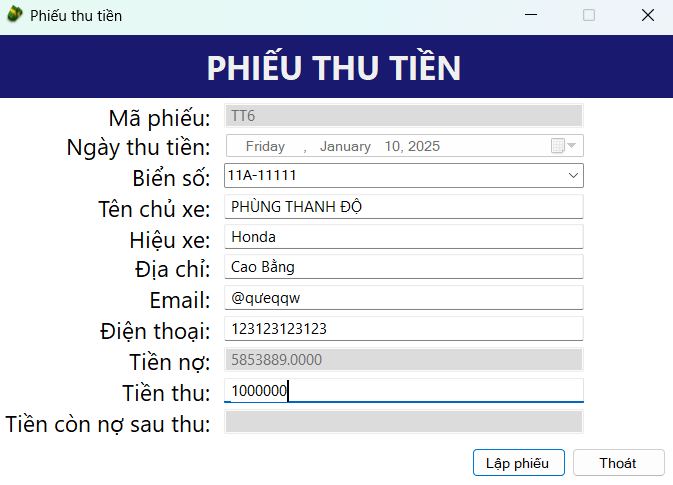
*hình 3.5 :Giao diện tiếp nhận xe sửa*

**

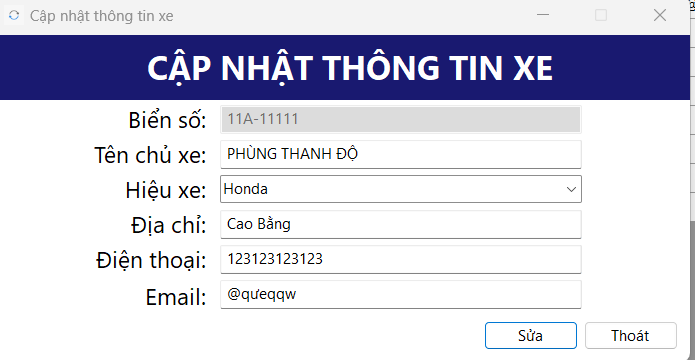
*hình 3.6: Giao diện phiếu nhập kho vật tư phụ tùng*

**

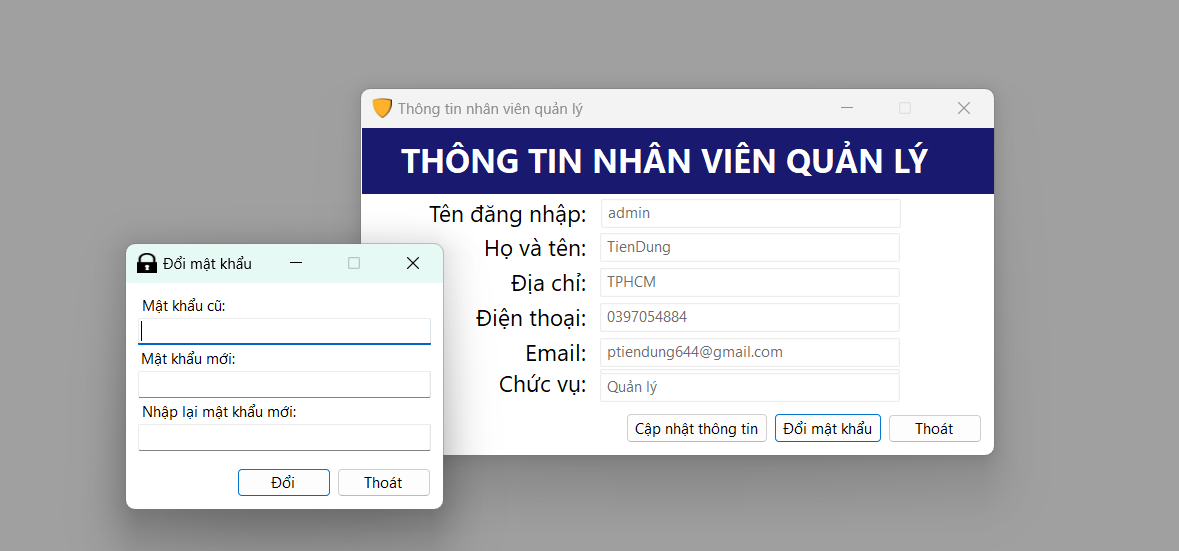
*hình 3.7 : Giao diện phiếu sửa chữa*

**

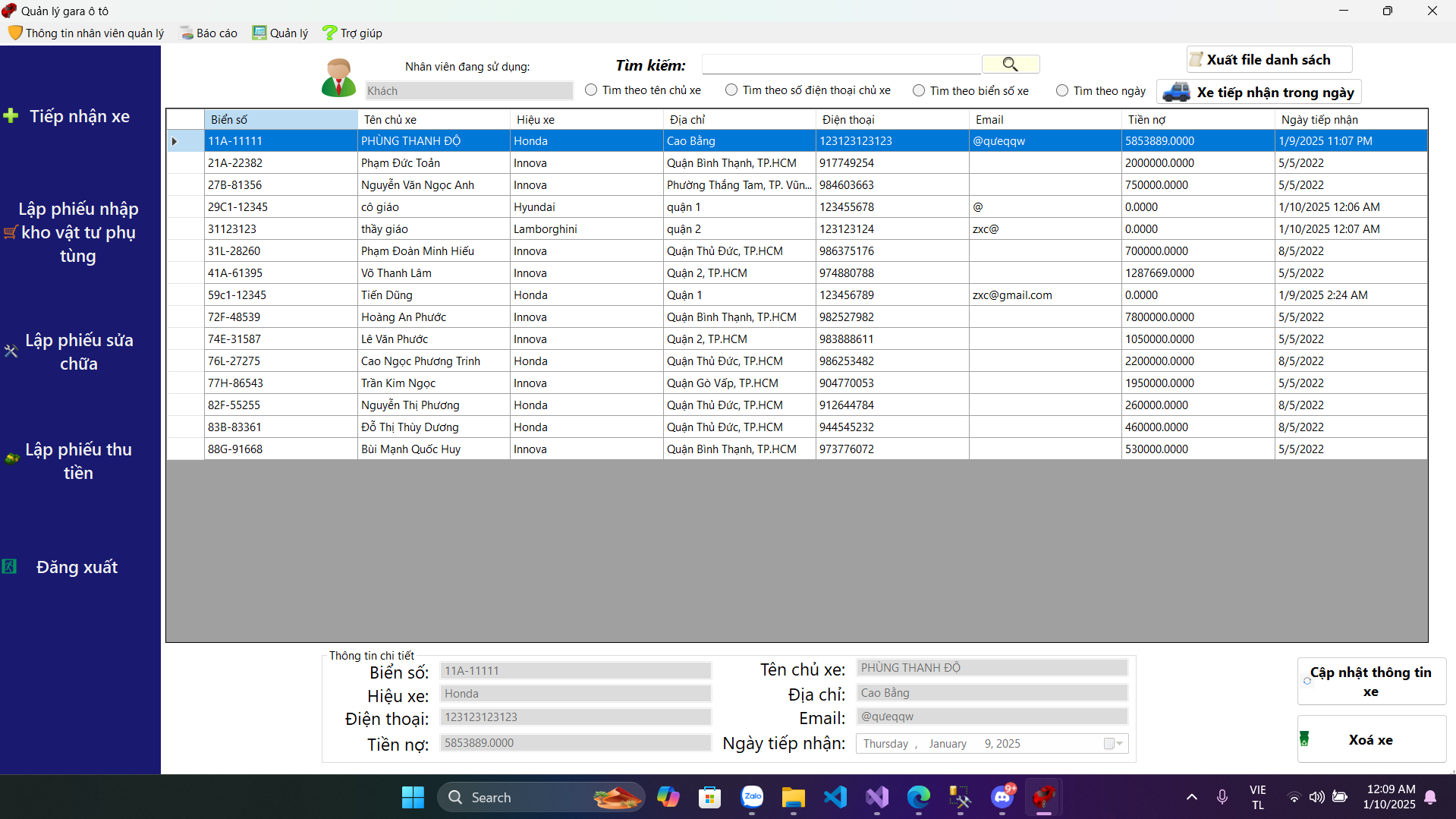
*hình 3.8: Giao diện phiếu thu tiền*

**

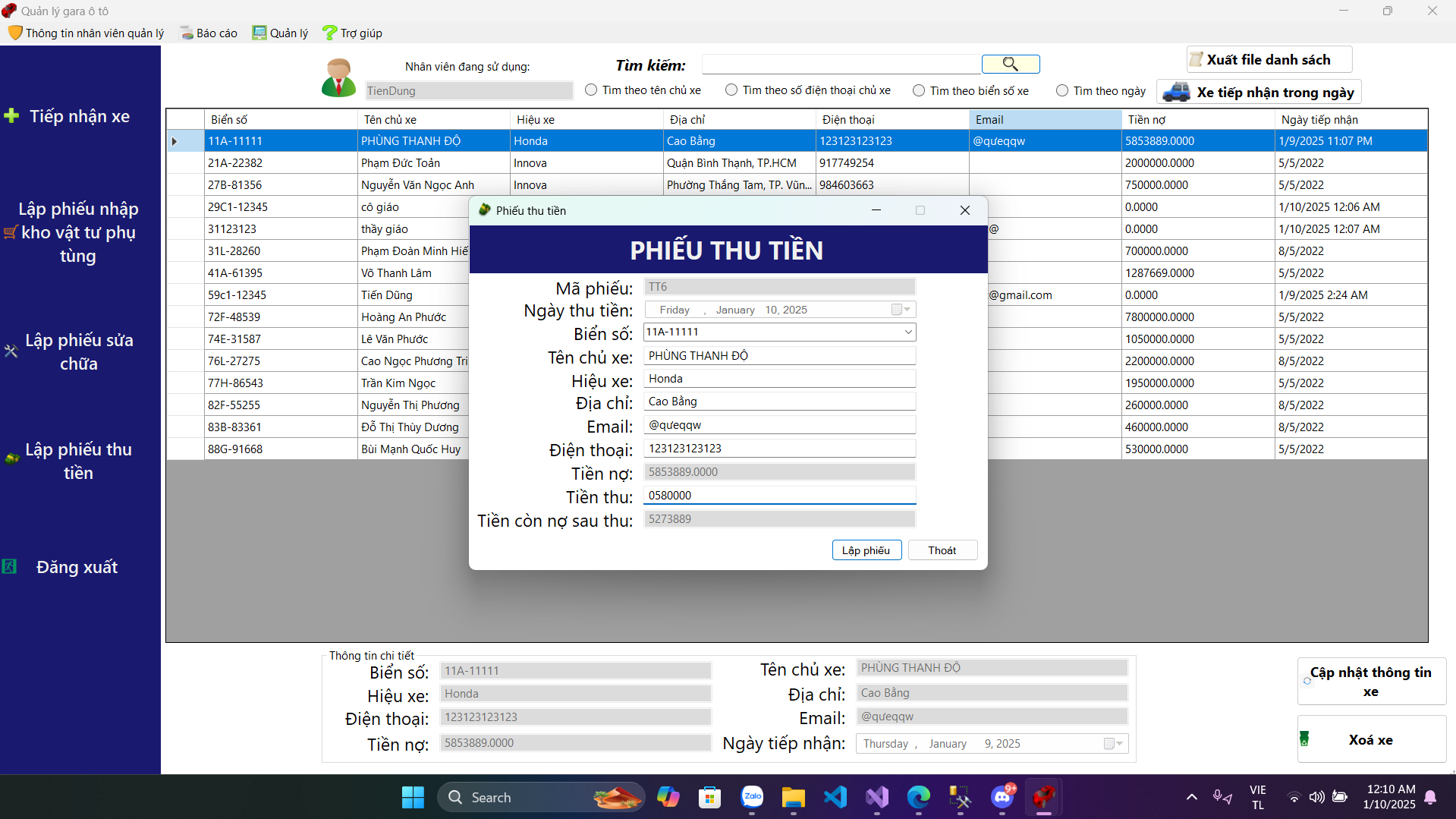
*hình 3.9: Giao diện cập nhật thông tin xe*

**

*hình 3.10: Giao diện thông tin nhân viên quản lý*

**

*hình 3.11: Giao diện chế độ khách*

**

*hình 3.11: Giao diện thu tiền chế độ khách*

# **Chương 4: KẾT LUẬN VÀ KIẾN NGHỊ**

Trong quá trình thực hiện đề tài "Quản lý thư viện ", hệ thống đã được xây dựng và hoàn thiện với các chức năng đáp ứng cơ bản nhu cầu quản lý thư viện hiện nay. Cụ thể:

1. **Hệ thống phân quyền người dùng:**

* Xây dựng thành công hai loại tài khoản đăng nhập: **admin** và **chế độ khách**
* Mỗi loại tài khoản có các chức năng và quyền hạn phù hợp với vai trò của người dùng.

1. **Chức năng quản lý dành cho Admin**:

* Admin được cung cấp quyền truy cập đầy đủ vào các danh mục quản lý, bao gồm: **tiếp nhận xe** , **danh sách xe, lập phiếu nhập kho vật tư phụ tùng , nhân viên , lập phiếu sửa chữa , lập phiếu thu tiền và quản lý**
* Admin có thể thực hiện các thao tác **thêm, sửa, xóa** dữ liệu trong từng chức năng, đáp ứng nhu cầu quản lý toàn diện.

1. **Chức năng quản lý dành cho Chế Độ Khách**

* Chế Độ Khách có quyền truy cập các danh mục quản lý như : **danh sách xe , tìm kiếm xe và phiếu thu tiền**

1. **Giao diện thân thiện và dễ sử dụng**:

* Hệ thống được thiết kế bằng WinForms với giao diện đơn giản, dễ thao tác và thuận tiện cho người dùng.

1. **Hiệu quả quản lý và tiết kiệm thời gian**:

* Hệ thống hỗ trợ việc lưu trữ và truy xuất dữ liệu nhanh chóng, giúp nâng cao hiệu quả làm việc, đồng thời giảm thiểu sai sót trong quá trình quản lý.

#### **Kiến nghị**

1. **Cải thiện và mở rộng hệ thống**:

* Nâng cấp giao diện và tối ưu hóa trải nghiệm người dùng.
* Phát triển thêm các chức năng như gửi email nhắc sách bảo hành và chăm sóc khách hàng online , thống kê chi tiết theo yêu cầu cụ thể.

1. **Ứng dụng công nghệ mới**:

* Nghiên cứu tích hợp hệ thống với công nghệ **đám mây** để lưu trữ và quản lý dữ liệu một cách linh hoạt hơn.
* Xây dựng phiên bản **ứng dụng di động** để hỗ trợ quản lý gara ô tô từ xa.

1. **Đào tạo và hướng dẫn người dùng**:

* Tổ chức các buổi tập huấn hoặc cung cấp tài liệu hướng dẫn để người dùng nắm rõ cách sử dụng hệ thống hiệu quả nhất.

Với những kết quả đạt được, hệ thống quản lý gara ô tô bằng WinForms đã bước đầu đáp ứng tốt các yêu cầu quản lý, góp phần hỗ trợ công tác quản lý gara ô tô trở nên thuận tiện và chuyên nghiệp hơn. Tuy nhiên, vẫn cần tiếp tục cải tiến và phát triển thêm các tính năng để đáp ứng những nhu cầu quản lý hiện đại hơn trong tương lai.