**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🙠🏵🏵🏵🙢🙠**

****

**TRẦN NGUYỄN HỒNG ÂN**

**Mã số sinh viên: 2051052007**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ KHO HÀNG**

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

Ngành: Công Nghệ Thông Tin

Hướng chuyên ngành: Cơ sở dữ liệu

Lớp: DH20IT03

**Giảng viên hướng dẫn: Ths. NGUYỄN THỊ MAI TRANG**

**TP. HỒ CHÍ MINH, 2024**

**LỜI MỞ ĐẦU**

Theo thời gian, sự phát triển vượt bậc của Công nghệ thông tin ngày nay vào việc ứng dụng trong đời sống con người ngày càng gia tăng đáng kể. Trong thời đại chuyển đổi số hiện nay thì các công ty, doanh nghiệp hay các xí nghiệp đều có nhu cầu phát triển ứng dụng tùy theo mục tiêu mỗi doanh nghiệp, có thể dùng để quảng bá sản phẩm đến các đối tác hay nhu cầu về một hệ thống quản lý hiệu quả, tiện lợi trong doanh nghiệp.

Vì vậy việc thiết kế và xây dựng một ứng dụng về quản lý là rất cần thiết và hiệu quả để doanh nghiệp có thể dễ dàng kiểm soát và quản lý nội bộ doanh nghiệp nhanh chóng và tiện ích hơn. Ngoài ra, việc có một ứng dụng quản lý còn giúp các công ty, doanh nghiệp và xí nghiệp phát triển và cạnh tranh hiệu quả hơn trên thị trường hiện nay.

Bên cạnh đó, việc quản lý kho hàng trong thời đại công nghệ hiện nay đóng vai trò quan trọng trong quá trình vận hành của mọi doanh nghiệp. Việc hiệu quả quản lý kho không chỉ giúp tối ưu hóa chi phí mà còn đảm bảo nguồn cung cấp liên tục và đáp ứng nhanh chóng nhu cầu khách hàng.

Với những lý do trên, tôi đã phát triển đề tài “Ứng dụng quản lý kho hàng”. Đề tài thực tập của tôi sẽ tập trung vào việc phát triển và triển khai một ứng dụng quản lý kho hàng, nhằm cung cấp cho doanh nghiệp một giải pháp hiệu quả cho việc giám sát, theo dõi tiến độ và điều chỉnh các hoạt động kho hàng của họ.

Mục tiêu của đề tài là phát triển một ứng dụng quản lý kho hàng linh hoạt, dễ sử dụng và có khả năng tích hợp với nhiều hệ thống khác. Tôi hy vọng đề tài “Ứng dụng quản lý kho hàng” sẽ mang lại những giải pháp hiệu quả cho doanh nghiệp, giúp họ nâng cao khả năng quản lý, giảm thiểu rủi ro mất mát và tối ưu hóa hiệu suất vận hành.

**LỜI CẢM ƠN**

Tôi xin gửi lời cảm ơn chân thành nhất đến tất cả những người đã đồng hành và hỗ trợ tôi trong quá trình thực hiện báo cáo thực tập này, đặc biệt là những người hướng dẫn, đồng nghiệp, và gia đình.

Đầu tiên và quan trọng nhất, tôi muốn bày tỏ lòng biết ơn sâu sắc đến cô Nguyễn Thị Mai Trang - người đã dành thời gian và công sức để hướng dẫn, hỗ trợ, và giúp đỡ tôi trong suốt quá trình thực tập. Cô không chỉ chia sẻ kiến thức chuyên sâu mà còn tạo điều kiện cho tôi để phát triển kỹ năng và hiểu rõ hơn về lĩnh vực quản lý kho hàng.

Tôi cũng muốn gửi lời cảm ơn đến toàn bộ nhân viên Công Ty Cổ Phần GONSA vì đã tạo điều kiện thuận lợi cho tôi thực hiện dự án này. Sự hỗ trợ và tạo điều kiện làm việc chuyên nghiệp của mọi người đã giúp tôi trải qua một trải nghiệm thực tập tích cực và ý nghĩa.

Cuối cùng, nhưng không kém phần quan trọng, tôi muốn bày tỏ lòng biết ơn đến gia đình và bạn bè đã luôn ở bên cạnh tôi, động viên và hỗ trợ trong suốt thời gian thực tập. Sự ủng hộ của họ đã là nguồn động viên lớn và là động lực mạnh mẽ giúp tôi vượt qua mọi khó khăn.

Một lần nữa, tôi xin chân thành cảm ơn tất cả mọi người. Sự giúp đỡ của mọi người không chỉ giúp tôi hoàn thành dự án mà còn để lại những kỷ niệm đáng nhớ trong quá trình thực tập này.

Trân trọng,

Trần Nguyễn Hồng Ân

TP.HCM, Tháng 1 năm 2024

**ĐÁNH GIÁ KẾT QUẢ THỰC TẬP CỦA KHOA**

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU VỀ TỔ CHỨC GONSA 10](#_Toc156085077)

[1.1 Lịch làm việc tại Công ty GONSA 10](#_Toc156085078)

[1.2 Giới thiệu về Công ty GONSA 10](#_Toc156085079)

[CHƯƠNG 2. NỘI DUNG THỰC TẬP 12](#_Toc156085080)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 13](#_Toc156085081)

[3.1 Giới thiệu về ASP.NET Core Web API 13](#_Toc156085082)

[3.2 Giới thiệu về EF Core Power Tools 14](#_Toc156085083)

[3.3 RestSharp 15](#_Toc156085084)

[CHƯƠNG 4. NỘI DUNG CÔNG VIỆC VÀ KẾT QUẢ ĐẠT ĐƯỢC 17](#_Toc156085085)

[4.1 Đặt vấn đề 17](#_Toc156085086)

[4.1.1 Nhu cầu thị trường 17](#_Toc156085087)

[4.1.2 Những thách thức 17](#_Toc156085088)

[4.2 Phân tích hệ thống 19](#_Toc156085089)

[4.2.1 Phân tích cơ sở dữ liệu 19](#_Toc156085090)

[4.2.2 Phân tích các chức năng hệ thống 21](#_Toc156085091)

[4.2.3 Đối tượng sử dụng 22](#_Toc156085092)

[4.3 Mô tả hệ thống 23](#_Toc156085093)

[4.3.1 Đối tượng các giao diện quản lý: 23](#_Toc156085094)

[4.3.2 Mô tả chức năng “Xem” giao diện quản lý 24](#_Toc156085095)

[4.3.3 Mô tả chức năng “Thêm” 25](#_Toc156085096)

[4.3.4 Mô tả chức năng “Sửa” 26](#_Toc156085097)

[4.3.5 Mô tả chức năng “Xóa” 27](#_Toc156085098)

[4.3.6 Mô tả chức năng “Tìm kiếm” 28](#_Toc156085099)

[4.3.7 Mô tả chức năng “Xem” trên giao diện chi tiết 29](#_Toc156085100)

[4.4 Giao diện ứng dụng 32](#_Toc156085101)

[4.4.1 Giao diện đăng nhập 32](#_Toc156085102)

[4.4.2 Giao diện chính 33](#_Toc156085103)

[4.4.3 Giao diện Loading 34](#_Toc156085104)

[4.4.4 Giao diện quản lý chi nhánh 35](#_Toc156085105)

[4.4.5 Giao diện quản lý kho 36](#_Toc156085106)

[4.4.6 Giao diện quản lý lô 37](#_Toc156085107)

[4.4.7 Giao diện quản lý tài khoản 39](#_Toc156085108)

[4.4.8 Giao diện quản lý nhân viên 40](#_Toc156085109)

[4.4.9 Giao diện quản lý nhóm thuốc 42](#_Toc156085110)

[4.4.10 Giao diện quản lý thuốc 43](#_Toc156085111)

[4.4.11 Giao diện quản lý nhà cung cấp 45](#_Toc156085112)

[4.4.12 Giao diện quản lý đơn nhập 46](#_Toc156085113)

[4.4.13 Giao diện quản lý đơn xuất 48](#_Toc156085114)

[4.4.14 Giao diện chi tiết đơn nhập 49](#_Toc156085115)

[4.4.15 Giao diện chi tiết đơn xuất 50](#_Toc156085116)

[4.4.16 Giao diện chi tiết đơn thuốc 51](#_Toc156085117)

[4.5 Kết quả đạt được 53](#_Toc156085118)

[CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC SAU ĐỢT THỰC TẬP 54](#_Toc156085119)

[5.1 Những kiến thức lý thuyết được củng cố 54](#_Toc156085120)

[5.2 Những kỹ năng thực hành đã học được 54](#_Toc156085121)

[5.2.1 Hoạt động của RESTful API 54](#_Toc156085122)

[5.2.1.1 Client (Người sử dụng API): 54](#_Toc156085123)

[5.2.1.2 REST API (Dịch vụ Web): 55](#_Toc156085124)

[5.2.1.3 Server (Nơi lưu trữ và Xử lý dữ liệu): 55](#_Toc156085125)

[5.2.2 Cách sử dụng EF Core Power Tools 56](#_Toc156085126)

[5.3 Những kinh nghiệm thực tiễn đã đạt được 58](#_Toc156085127)

[TÀI LIỆU THAM KHẢO 60](#_Toc156085128)

**MỤC LỤC HÌNH**

[Hình 1.1 Logo công ty Cổ phần GONSA 10](#_Toc156083968)

[Hình 4.1 Mối quan hệ các bảng 19](#_Toc156083969)

[Hình 4.2 Giao diện đăng nhập 32](#_Toc156083970)

[Hình 4.3 Giao diện chính 33](#_Toc156083971)

[Hình 4.4 Giao diện Loading 34](#_Toc156083972)

[Hình 4.5 Giao diện quản lý chi nhánh 36](#_Toc156083973)

[Hình 4.6 Giao diện quản lý kho 37](#_Toc156083974)

[Hình 4.7 Giao diện quản lý lô 38](#_Toc156083975)

[Hình 4.8 Giao diện quản lý tài khoản 40](#_Toc156083976)

[Hình 4.9 Giao diện quản lý nhân viên 41](#_Toc156083977)

[Hình 4.10 Giao diện quản lý nhóm thuốc 43](#_Toc156083978)

[Hình 4.11 Giao diện quản lý thuốc 44](#_Toc156083979)

[Hình 4.12 Giao diện quản lý nhà cung cấp 46](#_Toc156083980)

[Hình 4.13 Giao diện quản lý đơn nhập 47](#_Toc156083981)

[Hình 4.14 Giao diện quản lý đơn xuất 49](#_Toc156083982)

[Hình 4.15 Giao diện chi tiết đơn nhập 50](#_Toc156083983)

[Hình 4.16 Giao diện chi tiết đơn xuất 51](#_Toc156083984)

[Hình 4.17 Giao diện chi tiết thuốc 52](#_Toc156083985)

[Hình 5.1 Mô hình hoạt động RESTful API 54](#_Toc156083986)

[Hình 5.2 Cài đặt EF Core Power Tools 56](#_Toc156083987)

[Hình 5.3 Giao diện EF Core Power Tools 57](#_Toc156083988)

**MỤC LỤC BẢNG**

[Bảng 2.1 Bảng nội dung thực tập 12](#_Toc156084142)

[Bảng 4.1 Nội dung các bảng 21](#_Toc156084143)

[Bảng 4.2 Bảng phân quyền cho người dùng 22](#_Toc156084144)

**CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Từ hoàn chỉnh** |
| EF | Entity Framework |
| HTTP | Hyper Text Transfer Protocol (Giao thức Truyền tải Siêu Văn Bản) |

CHƯƠNG 1. GIỚI THIỆU VỀ TỔ CHỨC GONSA

1. Lịch làm việc tại Công ty GONSA

Làm việc theo chế độ Fulltime: Công ty làm việc từ thứ 2 tới thứ 6

* Buổi sáng từ 8 giờ – 11 giờ 30
* Buổi chiều từ 13 giờ – 17 giờ 30

1. Giới thiệu về Công ty GONSA

* Địa chỉ: Lô F14-2-2 & F14-2-3, Đường số 24, KCN Hiệp Phước, Xã Hiệp Phước, Huyện Nhà Bè, Tp. Hồ Chí Minh, Việt Nam
* Hotline: 1800 646899
* Email: [dvkh@gonsa.com.vn](mailto:dvkh@gonsa.com.vn)
* Website: <https://gonsa.com.vn/>

Công ty Cổ Phần GONSA là công ty chuyên về Dược phẩm, Vắc-xin, Sinh phẩm, Vật tư y tế, Thực phẩm chức năng và Mỹ phẩm, được thành lập vào ngày 08/03/2010. Và từ năm 2021 thì chuyển đổi sang mô hình “Chuyên dịch vụ dược phẩm” [1].

GONSA mang trong mình sứ mệnh là “Chăm sóc sức khỏe cộng đồng thông qua sản phẩm chất lượng, kết hợp dịch vụ phân phối hiện đại, đạt chuẩn (GSP, GDP, ...)” cùng với tầm nhìn “Trở thành biểu tượng chất lượng dịch vụ dược phẩm”.



Hình 1.1 Logo công ty Cổ phần GONSA

Với tầm nhìn “Trở thành biểu tượng chất lượng dịch vụ dược phẩm”, GONSA đã, đang và sẽ cố gắng trong việc cải tiến, nâng cao hiệu quả quản lý chất lượng theo các tiêu chuẩn ISO 0991-2015 và GSP, GDP. Cơ sở vật chất và hệ thống quy trình được xây dựng hoàn chỉnh nhằm đảm bảo sản phẩm luôn được duy trì chất lượng yêu cầu trong suốt quá trình từ lúc nhận hàng đến lúc giao hàng đến khách hàng đều đảm bảo. GONSA luôn kiểm soát mọi nguy cơ có thể ảnh hưởng đến chất lượng sản phẩm trước khi giao tới tay khách hàng.

GONSA sở hữu đội ngũ vận chuyển xe vận tải với nhiều tải trọng khác nhau, được cài đặt GPS và trang bị đầy đủ các điều kiện cần thiết đáp ứng việc vận chuyển hàng hóa. Ngoài ra, GONSA cũng phát triển đội ngũ giao hàng xe máy để đảm bảo giao những đơn lẻ đến các vùng sâu vùng xa.

Bên cạnh đó, công ty GONSA còn ứng dụng công nghệ phát triển theo hệ thống quản lý nguồn lực toàn diện và nhất quán, từ Hành chính nhân sự, Quản lý tài sản, Kế toán, Mua hàng…, giúp hoạt động GONSA luôn xuyên suốt, nhanh chóng để đạt được kết quả cao. Ứng dụng chuyển đổi số trong nhiều khâu sản xuất và vận hành nhằm hướng tới mục tiêu số hóa toàn công ty.

CHƯƠNG 2. NỘI DUNG THỰC TẬP

|  |  |
| --- | --- |
| **Tuần** | **Nội dung** |
| 1 | Tìm hiểu mô hình 3 lớp trong C#, nghiên cứu .NET, Entity Frame Framework, RESTful API. |
| 2 | Thiết kế Database |
| 3 | Làm Frontend ứng dụng WinForm theo mô hình 3 lớp bằng việc kéo thả các tool |
| 4 | Làm Backend các chức năng thêm, sửa, xóa, export, import |
| 5 | Viết WebAPI thực hiện các chức năng get, post, put, delete |
| 6 | Viết WebAPI, thiết kế cơ sở dữ liệu kết nối vào thực hiện chức năng API (get, put, post, delete), kết nối vào SQL Server |
| 7-8 | Tích hợp Web API vào .Net, Entity Framework |
| 9 | Kết nối các giao diện lại hoàn chỉnh |
| 10 | Tối ưu hóa, kiểm thử ứng dụng, UAT |

Bảng 2.1 Bảng nội dung thực tập

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

1. Giới thiệu về ASP.NET Core Web API

ASP.NET Core Web API là một Framework mạnh mẽ và hiệu quả được phát triển bởi Microsoft, được thiết kế để xây dựng các dịch vụ web RESTful linh hoạt và dễ dàng triển khai. Được giới thiệu lần đầu vào năm 2016 [2], ASP.NET Core Web API nhanh chóng trở thành một lựa chọn phổ biến cho việc xây dựng các ứng dụng web đa nền tảng cũng như đa thiết bị.

ASP.NET Core Web API cung cấp nhiều tính năng tiện ích và mạnh mẽ. Nó hỗ trợ việc tạo ra các ứng dụng web RESTful với các định dạng truyền tải dữ liệu như JSON và XML. Ngoài ra nó còn hỗ trợ Dependency Injection, Middleware, Routing linh hoạt, và cả hệ thống Middleware pipeline cho phép tùy chỉnh và mở rộng dễ dàng.

Với sự tối ưu hóa và hiệu suất cao thì ASP.NET Core Web API giúp việc tối ưu hóa tài nguyên và làm tăng khả năng xử lý của ứng dụng. Và sự đa nền tảng là một trong những điểm mạnh của nó, cho phép ứng dụng chạy trên nhiều hệ điều hành như Windows, Linux và macOS. Khả năng đáp ứng cao và khả năng xử lý đồng thời tốt làm cho nó trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu xử lý lớn và cần tốc độ phản hồi nhanh.

Với ASP.NET Core Web API, việc bảo mật là một ưu tiên hàng đầu. Giúp ta có thể tích hợp các giải pháp bảo mật như xác thực JWT (JSON Web Tokens) để đảm bảo tính toàn vẹn và bảo mật dữ liệu trong quá trình truyền tải dữ liệu. Quản lý phiên bản API là một việc quan trọng, giúp đảm bảo sự ổn định và tương thích trong quá trình phát triển và triển khai.

ASP.NET Core Web API được hỗ trợ bởi một cộng đồng nhà lập trình phát triển đông đảo. Cộng đồng này cung cấp tài liệu đa dạng, phong phú như việc thực hành hay hỗ trợ tận tâm qua các diễn đàn trực tuyến. Điều này giúp việc học và xây dựng ASP.NET Core Web API trở nên dễ dàng hơn.

Tóm lại thì ASP.NET Core Web API không chỉ là một Framework mạnh mẽ để triển khai dịch vụ web mà còn là một phần quan trọng của hệ sinh thái phát triển web của Microsoft. Với tính linh hoạt, hiệu suất cao cùng với việc bảo mật mạnh mẽ thì nó đáp ứng đầy đủ các yêu cầu của ứng dụng ngày nay.

1. Giới thiệu về EF Core Power Tools

Entity Framework (EF) Core Powel Tools là một công cụ mở rộng cho Entity Framework Core, được thiết kế để hỗ trợ lập trình viên trong việc quản lý và tương tác với cơ sở dữ liệu bằng cách sử dụng giao diện người dùng đồ họa. Được tích hợp trực tiếp vào Visual Studio, công cụ này cung cấp nhiều tính năng giúp đơn giản hóa quá trình phát triển ứng dụng trên cơ sở dữ liệu.

EF Core Powel Tools cho phép lập trình viên tạo mô hình dữ liệu từ cơ sở dữ liệu hiện tại và kết xuất cấu trúc cơ sở dữ liệu từ mã nguồn. Điều này làm đẩy tiến độ xây dựng mô hình đối tượng cho ứng dụng mà không cần viết mã một cách thủ công.

Một trong những tính năng đáng chú ý của công cụ này là khả năng xem trực tiếp dữ liệu từ cơ sở dữ liệu thông qua Visual Studio. Lập trình viên có thể xem, thêm, sửa và xóa dữ liệu một cách thuận tiện mà không cần phải chuyển sang môi trường quản lý cơ sở dữ liệu riêng biệt.

Bên cạnh đó khi sử dụng mô hình Code First Approach thì EF Core Powel Tools mang lại nhiều tiện ích mở rộng, giúp tối ưu hóa quá trình phát triển và kiểm soát mã nguồn. Nó cung cấp các tính năng tùy chọn như tạo chuỗi kết nối, tạo Script cơ sở dữ liệu và thậm chí là việc đọc và hiểu mã nguồn để tạo mô hình dữ liệu.

EF Core Powel Tools được tích hợp trực tiếp vào môi trường Visual Studio, giúp tăng cường trải nghiệm người dùng. Việc sử dụng công cụ này không yêu cầu cài đặt riêng lẻ và dễ dàng trong việc tích hợp vào quy trình làm việc hàng ngày.

Với sự hỗ trợ tích cực từ cộng đồng và tích hợp chặt chẽ với Entity Framework Core, EF Core Powel Tools là một công cụ mở rộng mạnh mẽ giúp tối ưu hóa quá trình phát triển dự án sử dụng EF Core.

Tóm lại thì EF Core Powel Tools không chỉ là một công cụ hữu ích để tương tác với cơ sở dữ liệu thông qua nó mà còn giúp cho việc tối ưu hóa quy trình phát triển và kiểm soát mã nguồn dễ dàng hơn. Điều này làm cho việc triển khai và duy trì ứng dụng dựa trên cơ sở dữ liệu trở nên đơn giản và hiệu quả hơn.

1. RestSharp

RestSharp là một thư viện HTTP Client mạnh mẽ được thiết kế để giúp việc tương tác với các dịch vụ web RESTful trở nên dễ dàng hơn trong ngôn ngữ lập trình C#. Với RestSharp, việc thực hiện các yêu cầu HTTP như GET, POST, PUT và DELETE trở nên đơn giản và thuận tiện hơn.

RestSharp cũng hỗ trợ nhiều phương thức HTTP và cung cấp một cách linh hoạt để tương tác với API. Nhờ RestSharp giúp dễ dàng thực hiện các yêu cầu RESTful thông qua các phương thức như Excute, ExcuteAsync, AddParameter và AddHeader.

Thư viện này hỗ trợ nhiều định dạng dữ liệu khác nhau như JSON, XML và các định dạng khác. Điều này giúp cho việc truyền dữ liệu giữa các ứng dụng và API trở nên linh hoạt hơn.

RestSharp cung cấp các tính năng quản lý cookie và session, giúp duy trì trạng thái giao tiếp giữa các yêu cầu liên quan.

RestSharp hoạt động không chỉ trên .NET Framework mà còn trên .NET Core và .NET 5, hỗ trợ nhiều phiên bản khác nhau của Framework.

RestSharp tích hợp dễ dàng nhiều dự án .NET và có thể được cài đặt thông qua NuGet, giúp giảm bớt công đoạn triển khai và quản lý thư viện.

Đối với việc xây dựng ứng dụng giao tiếp với các API RESTful thì RestSharp chính là một công cụ mạnh mẽ và hiệu quả. Với khả năng linh hoạt, đa dạng và dễ sử dụng, RestSharp giúp tối ưu hóa quá trình phát triển và tương tác với các dịch vụ web. Sự tích hợp dễ dàng và hỗ trợ phong phú định dạng dữ liệu làm cho RestSharp trở thành một lựa chọn hàng đầu cho các lập trình viên .NET.

CHƯƠNG 4. NỘI DUNG CÔNG VIỆC VÀ KẾT QUẢ ĐẠT ĐƯỢC

1. Đặt vấn đề
2. Nhu cầu thị trường

Trong bối cảnh kinh doanh ngày nay, có một nhu cầu cấp bách đối với các doanh nghiệp - đó là nhu cầu sử dụng một ứng dụng “Quản lý kho hàng”. Việc thiết kế một ứng dụng không chỉ đáp ứng nhu cầu này mà còn mang lại trải nghiệm người dùng tốt nhất. Tôi đặt ra mục tiêu xây dựng một giao diện thân thiện với người dùng, dễ sử dụng và giúp họ quản lý, kiểm soát kho hàng một cách thuận tiện.

Ứng dụng của tôi không chỉ là công cụ quản lý thông thường mà còn là đối tác đáng tin cậy giúp doanh nghiệp tối ưu hóa quá trình quản lý kho. Giao diện được tối ưu hóa để đảm bảo người dùng có trải nghiệm mượt mà và nhanh chóng, từ đó tăng cường khả năng ra quyết định nhanh chóng dựa trên dữ liệu khoa học và chi tiết từ ứng dụng.

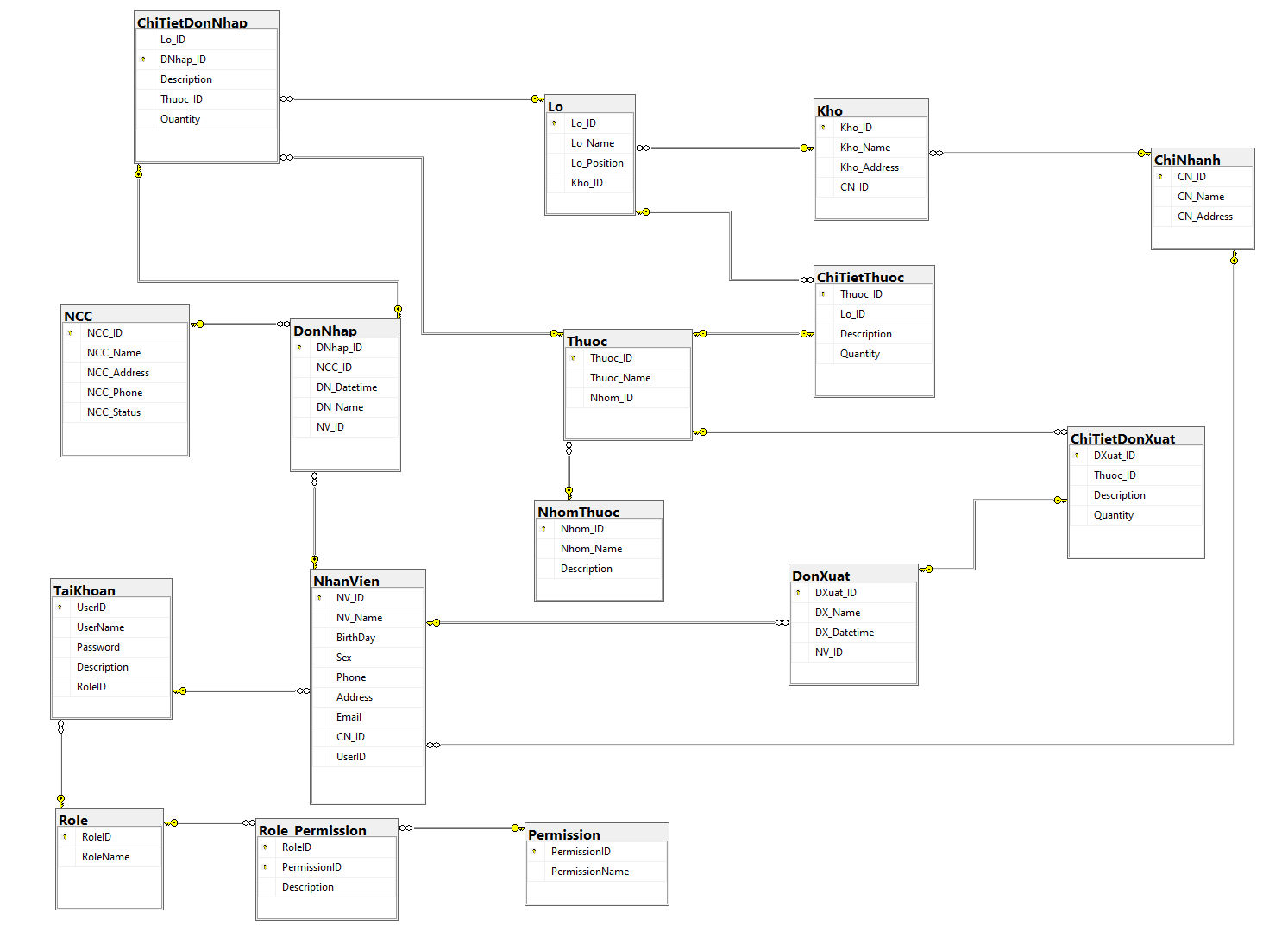
1. Những thách thức

Quản lý kho hàng là một nhiệm vụ phức tạp và đầy thách thức trong môi trường kinh doanh hiện đại. Ứng dụng “Quản lý kho hàng” đối mặt với một loạt các khó khăn mà doanh nghiệp cần vượt qua để duy trì và nâng cao hiệu suất của hệ thống. Dưới đây là những thách thức phổ biến mà ứng dụng này phải đối diện khi tôi phát triển:

* Khả năng tích hợp: Khi doanh nghiệp mở rộng và sử dụng nhiều hệ thống khác nhau, khả năng tích hợp ứng dụng “Quản lý kho hàng” với các hệ thống khác trở thành một thách thức. Sự liên kết và đồng bộ dữ liệu giữa các ứng dụng là quan trọng để tránh sự mất mát thông tin và giảm thiểu sai sót.
* Quản lý dữ liệu lớn: Với lượng dữ liệu ngày càng tăng, quản lý và xử lý dữ liệu lớn trở thành một thách thức. Bảo đảm rằng hệ thống có khả năng xử lý lượng dữ liệu lớn mà không ảnh hưởng đến hiệu suất là một yếu tố quyết định.
* An toàn dữ liệu: Bảo vệ thông tin về kho hàng là một ưu tiên hàng đầu. Thách thức này đặt ra nhu cầu về các biện pháp an toàn thông tin mạnh mẽ để ngăn chặn việc truy cập trái phép và bảo vệ dữ liệu quan trọng của doanh nghiệp.
* Quản lý chuỗi cung ứng: Liên kết với các đối tác và nhà cung cấp khác nhau đôi khi gặp khó khăn, đặc biệt là trong việc theo dõi và quản lý chuỗi cung ứng. Điều này đòi hỏi sự linh hoạt để đảm bảo sự hiệu quả và minh bạch trong mọi khía cạnh của quá trình.
* Thay đổi nhanh chóng trong nhu cầu thị trường: Nhu cầu thị trường thay đổi nhanh chóng, và ứng dụng cần có khả năng linh hoạt để điều chỉnh theo thời gian và đáp ứng đúng nhu cầu thực tế của doanh nghiệp.
* Đào tạo và sử dụng người dùng: Sự thành công của ứng dụng phụ thuộc nhiều vào khả năng sử dụng của người dùng. Đào tạo và hỗ trợ người dùng mới cũng như duy trì sự hài lòng của họ là một thách thức liên tục.

Bằng cách đối mặt và vượt qua những thách thức này, ứng dụng “Quản lý kho hàng” có thể trở thành một công cụ quan trọng và hiệu quả trong việc tối ưu hóa quá trình quản lý kho và đáp ứng nhanh chóng với sự biến động của thị trường.

1. Phân tích hệ thống
2. Phân tích cơ sở dữ liệu



Hình 4.1 Mối quan hệ các bảng

| **Tên bảng** | **Thuộc tính** | **Nội dung** |
| --- | --- | --- |
| ChiNhanh | CN\_ID, CN\_Name, CN\_Address | Lưu dữ liệu các chi nhánh |
| Kho | Kho\_ID, Kho\_Name, Kho\_Address, CN\_ID | Lưu dữ liệu các kho |
| Lo | Lo\_ID, Lo\_Name, Lo\_Position, Kho\_ID | Lưu dữ liệu các lô |
| Role | RoleID, RoleName | Lưu dữ liệu các vai trò của mỗi User |
| Permission | PermissionID, PermissionName | Lưu dữ liệu các quyền của Role |
| Role\_Permission | RoleID, PermissionID, Description | Lưu dữ liệu vai trò nào sẽ được quyền thực hiện chức năng nào |
| TaiKhoan | UserID, UserName, Password, RoleID, Description | Lưu dữ liệu các tài khoản của nhân viên |
| NhanVien | NV\_ID, NV\_Name, BirthDay, Sex, Phone, Email, Address, UserID, CN\_ID | Lưu dữ liệu thông tin mỗi nhân viên |
| NhomThuoc | Nhom\_ID, Nhom\_Name, Description | Lưu dữ liệu các nhóm thuốc |
| Thuoc | Thuoc\_ID, Thuoc\_Name, Nhom\_ID | Lưu dữ liệu thông tin thuốc |
| NCC | NCC\_ID, NCC\_Name, NCC\_Address, NCC\_Phone, Quantity, NCC\_Status | Lưu dữ liệu thông tin các nhà cung cấp |
| DonNhap | DNhap\_ID, DN\_Name, DN\_Datetime, NCC\_ID, NV\_ID | Lưu dữ liệu thông tin các đơn nhập hàng |
| ChiTietDonNhap | DNhap\_ID, Lo\_ID, Thuoc\_ID, Description, Quantity | Lưu dữ liệu thông tin chi tiết đơn nhập |
| DonXuat | DXuat\_ID, DX\_Name, DX\_Datetime, NV\_ID | Lưu dữ liệu thông tin về đơn xuất hàng |
| ChiTietDonXuat | DXuat\_ID, Thuoc\_ID, Description, Quantity | Lưu dữ liệu thông tin về chi tiết đơn xuất |
| ChiTietThuoc | Lo\_ID, Thuoc\_ID, Description, Quantity | Lưu dữ liệu về thông tin về chi tiết thuốc |

Bảng 4.1 Nội dung các bảng

1. Phân tích các chức năng hệ thống

* Xem thông tin dữ liệu: Người dùng có khả năng dễ dàng xem thông tin chi tiết về kho hàng từ mọi giao diện quản lý. Điều này giúp họ nắm bắt tình hình hiện tại và đưa ra quyết định nhanh chóng, hiệu quả.
* Thêm dữ liệu mới: Mỗi giao diện quản lý cung cấp khả năng thêm mới dữ liệu, giúp người dùng nhanh chóng cập nhật thông tin về các mặt hàng, nhập kho mới, hoặc thêm mới thông tin liên quan đến quản lý kho.
* Xóa dữ liệu: Người dùng có quyền xóa dữ liệu không cần thiết hoặc lỗi từ mọi giao diện quản lý. Chức năng này giúp duy trì sạch sẽ và hiệu quả cho cơ sở dữ liệu của hệ thống.
* Sửa dữ liệu: Chức năng sửa dữ liệu cho phép người dùng điều chỉnh thông tin một cách dễ dàng. Điều này rất hữu ích khi cần cập nhật thông tin về số lượng tồn kho, giá cả, hoặc bất kỳ chi tiết nào khác.
* Tìm kiếm linh hoạt: Hệ thống hỗ trợ tính năng tìm kiếm theo nhiều tiêu chí khác nhau, giúp người dùng dễ dàng định dạng và truy xuất thông tin theo nhu cầu cụ thể. Việc này giúp tiết kiệm thời gian và nâng cao hiệu suất công việc.

Những chức năng trên không chỉ mang lại sự thuận tiện cho người dùng mà còn đảm bảo tính linh hoạt và mạnh mẽ trong quá trình quản lý kho hàng, đồng thời hỗ trợ họ trong việc đưa ra các quyết định chiến lược dựa trên thông tin chi tiết và chính xác.

1. Đối tượng sử dụng

Hệ thống có 2 vai trò là “Admin” và “Nhân viên

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Admin** | **Nhân viên** |
| **Thêm danh sách** | Admin được quyền thêm:   * Thành công: “Thêm mã thành công” * Thất bại: “Thêm thất bại” | Nhân viên được quyền thêm:   * Thành công: “Thêm mã thành công” * Thất bại: “Thêm thất bại” |
| **Sửa danh sách** | Admin được quyền cập nhật:   * Thành công: “Cập nhật mã thành công” * Thất bại: “Cập nhật thất bại” | Nhân viên không có quyền cập nhật nên nhận thông báo “Bạn không có quyền cập nhật” |
| **Xóa danh sách** | Admin được quyền xóa:   * Thành công: “Xóa mã thành công” * Thất bại: “Xóa thất bại” | Nhân viên không có quyền cập nhật nên nhận thông báo “Bạn không có quyền xóa” |
| **Tìm kiếm** | Admin được quyền tìm kiếm theo giá trị, từ khóa muốn tìm | Nhân viên được quyền tìm kiếm theo giá trị, từ khóa muốn tìm |

Bảng 4.2 Bảng phân quyền cho người dùng

1. Mô tả hệ thống
2. Đối tượng các giao diện quản lý:

* Chi nhánh: Thông tin về các chi nhánh trong quản lý kho
* Kho: Quản lý thông tin về các kho hàng trong từng chi nhánh
* Lô: Quản lý thông tin về lô hàng, nơi thuốc được sản xuất hoặc nhập về
* Nhà cung cấp: Thông tin về các đối tác cung cấp thuốc
* Thuốc: Thông tin về các loại thuốc, bao gồm tên, mã thuốc, mô tả, và thông tin liên quan
* Nhóm thuốc: Phân loại thuốc thành từng nhóm để dễ dàng quản lý.
* Tài khoản: Quản lý tài khoản người dùng có quyền truy cập vào hệ thống
* Nhân viên: Ghi lại thông tin về nhân viên sử dụng trong hệ thống
* Đơn nhập: Ghi lại thông tin về các đơn hàng nhập kho từ nhà cung cấp
* Đơn xuất: Ghi lại thông tin về các đơn hàng xuất kho
* Chi tiết đơn nhập: Ghi chi tiết về từng mặt hàng trong đơn hàng nhập
* Chi tiết đơn xuất: Ghi chi tiết về từng mặt hàng trong đơn hàng xuất các đơn vị hoặc chi nhánh khác
* Chi tiết thuốc: Ghi chi tiết về thuốc sẽ nằm ở lô nào

1. Mô tả chức năng “Xem” giao diện quản lý

Mô tả quy trình lấy danh sách đối tượng trong hệ thống của tôi có thể được mô tả như sau:

1. Gọi phương thức LoadDataGridView(): Người dùng kích hoạt sự kiện hoặc hệ thống tự động gọi phương thức “LoadDataGridView()” để hiển thị danh sách đối tượng trên DataGridView.
2. Gọi phương thức “LayDS” trong Service: Phương thức “LoadDataGridView()” gọi phương thức “LayDS” trong Service.
3. Tạo yêu cầu GET tới API Endpoint: Phương thức “LayDS” tạo một yêu cầu GET tới API endpoint “/api/Controller” thông qua thư viện RestSharp.
4. Controller nhận yêu cầu GET: Controller nhận yêu cầu GET từ API endpoint và chuyển giao tới dịch vụ xử lý.
5. Dịch vụ xử lý yêu cầu lấy danh sách đối tượng: Dịch vụ xử lý yêu cầu lấy danh sách đối tượng từ cơ sở dữ liệu thông qua phương thức “LayDS” (hoặc một đối tượng đơn).
6. Trả về kết quả: Dịch vụ trả về danh sách đối tượng hoặc “null” nếu có lỗi.
7. Kiểm tra kết quả: Phương thức “LayDS” kiểm tra kết quả từ dịch vụ. Nếu thành công, trả về danh sách đối tượng. Nếu thất bại, in ra lỗi và trả về “null”.
8. Xử lý kết quả tại phương thức LoadDataGridView: Phương thức “LoadDataGridView” nhận danh sách đơn nhập từ Service. Nếu danh sách có dữ liệu, sắp xếp và hiển thị trên DataGridView. Nếu không, hiển thị thông báo lỗi.
9. Hiển thị dữ liệu trên DataGridView: Dữ liệu được hiển thị trên DataGridView sau khi đã được sắp xếp theo một quy tắc nhất định.
10. Xử lý ngoại lệ: Nếu có lỗi trong quá trình lấy dữ liệu, hệ thống hiển thị thông báo lỗi với nội dung chi tiết.

Quy trình này đảm bảo việc lấy dữ liệu đơn nhập từ API endpoint thông qua các lớp Service và Controller, và sau đó hiển thị chúng lên giao diện DataGridView.

1. Mô tả chức năng “Thêm”

Chức năng “Thêm” trong hệ thống của tôi được mô tả như sau:

1. Người dùng nhập thông tin cần thiết khi kích hoạt sự kiện btnThem\_Click: Người dùng nhập thông tin cho các record mới trên giao diện, bao gồm các trường dữ liệu tùy mỗi danh sách giao diện rồi nhấn nút “Thêm” trên giao diện.
2. Kiểm tra quyền truy cập: Hệ thống kiểm tra quyền truy cập của người dùng. Nếu người dùng không có quyền thêm, hệ thống hiển thị thông báo lỗi và kết thúc quy trình.
3. Kiểm tra thông tin nhập liệu: Hệ thống kiểm tra xem các thông tin có hợp lệ không. Nếu không, hiển thị thông báo lỗi và kết thúc quy trình.
4. Tạo mã khóa chính tự động: Hệ thống tạo mã khóa chính tự động thông qua hàm “TaoMaTuDong()”.
5. Tạo đối tượng mới: Hệ thống tạo một danh sách đối tượng với thông tin được nhập và mã được tạo tự động.
6. Gọi Service (Dịch vụ) “Thêm”: Hệ thống gọi phương thức “ThemDoiTuong” trong service, truyền danh sách đối tượng thông qua request.
7. Service gửi yêu cầu POST: Service gửi yêu cầu POST tới API endpoint “/api/Controller” với danh sách đối tượng.
8. Controller xử lý yêu cầu POST: Controller nhận yêu cầu POST từ API endpoint và chuyển giao tới dịch vụ xử lý.
9. Dịch vụ xử lý yêu cầu thêm đối tượng mới: Dịch vụ xử lý yêu cầu thêm bản ghi mới, thêm dữ liệu vào cơ sở dữ liệu và trả về kết quả.
10. Xử lý kết quả: Hệ thống kiểm tra kết quả từ dịch vụ. Nếu thành công, hiển thị thông báo thành công, tải lại danh sách và các ô nhập liệu hoặc ComboBox cần thiết. Nếu thất bại, hiển thị thông báo lỗi.

Quy trình này giúp người dùng thêm mới và đồng thời đảm bảo tính bảo mật và kiểm soát quyền truy cập.

1. Mô tả chức năng “Sửa”

Mô tả quy trình cập nhật đối tượng trong hệ thống của tôi có thể được mô tả như sau:

1. Người dùng kích hoạt sự kiện cập nhật: Người dùng kích hoạt sự kiện cập nhật đối tượng bằng cách nhấn nút “Sửa” trên giao diện.
2. Gọi phương thức btnSua\_Click: Hệ thống gọi phương thức “btnSua\_Click” để xử lý sự kiện cập nhật đối tượng.
3. Kiểm tra quyền cập nhật: Phương thức kiểm tra quyền của người dùng để đảm bảo rằng họ có quyền cập nhật đối tượng.
4. Hiển thị xác nhận: Nếu có quyền cập nhật, hệ thống hiển thị hộp thoại xác nhận để đảm bảo rằng người dùng muốn tiếp tục cập nhật.
5. Gọi phương thức “CapNhat” trong Service: Nếu người dùng xác nhận, hệ thống gọi phương thức “CapNhat” trong Service để thực hiện cập nhật.
6. Tạo yêu cầu PUT tới API Endpoint: Phương thức “CapNhat” tạo một yêu cầu PUT tới API endpoint “/api/Controller/{id}” để cập nhật đối tượng có ID tương ứng.
7. Controller nhận yêu cầu PUT: Controller nhận yêu cầu PUT từ API endpoint và chuyển giao tới Service xử lý.
8. Service xử lý yêu cầu cập nhật đối tượng: Dịch vụ xử lý yêu cầu cập nhật đối tượng từ cơ sở dữ liệu thông qua phương thức “CapNhat”.
9. Trả về kết quả: Service trả về kết quả của quá trình cập nhật đối tượng thông qua response từ API.
10. Xử lý kết quả tại phương thức btnSua\_Click: Phương thức “btnSua\_Click” kiểm tra kết quả từ Service. Nếu thành công, hiển thị thông báo thành công và cập nhật lại dữ liệu trên giao diện. Ngược lại, hiển thị thông báo lỗi.

Quy trình này đảm bảo rằng thông tin đối tượng được cập nhật thông qua giao tiếp giữa giao diện, Service, và Controller của hệ thống.

1. Mô tả chức năng “Xóa”

Mô tả quy trình xóa đối tượng trong hệ thống của tôi có thể được mô tả như sau:

1. Người dùng kích hoạt sự kiện xóa: Người dùng kích hoạt sự kiện xóa bằng cách nhấn nút “Xóa” trên giao diện.
2. Kiểm Tra Quyền Xóa: Hệ thống kiểm tra quyền của người dùng để đảm bảo rằng họ có quyền xóa đối tượng.
3. Kiểm tra chọn dòng cần xóa trên DataGridView: Hệ thống kiểm tra xem người dùng đã chọn ít nhất một đối tượng để xóa hay chưa.
4. Tạo danh sách mã đối tượng cần xóa: Nếu có chọn, hệ thống tạo danh sách mã đối tượng cần xóa từ các dòng được chọn trên giao diện.
5. Hiển thị xác nhận: Hệ thống hiển thị hộp thoại xác nhận để người dùng xác nhận việc xóa các đối tượng đã chọn.
6. Xác nhận xóa: Nếu xác nhận, hệ thống tiếp tục thực hiện quy trình xóa.
7. Gọi phương thức “Xoa” Trong Service: Hệ thống gọi phương thức “Xoa” trong service để thực hiện xóa đối tượng.
8. Tạo yêu cầu DELETE tới API Endpoint: Phương thức “Xoa” tạo một yêu cầu DELETE tới API endpoint “/api/Controller/{id}” để xóa đối tượng có ID tương ứng.
9. Controller nhận yêu cầu DELETE: Controller nhận yêu cầu DELETE từ API endpoint và chuyển giao tới Service xử lý.
10. Service xử lý yêu cầu xóa đối tượng: Service xử lý yêu cầu xóa đối tượng từ cơ sở dữ liệu thông qua phương thức “Xoa”.
11. Trả về kết quả: Service trả về kết quả của quá trình xóa đối tượng thông qua response từ API.
12. Xử lý kết quả tại phương thức btnXoa\_Click: Phương thức “btnXoa\_Click” kiểm tra kết quả từ Service. Nếu thành công, hiển thị thông báo thành công và cập nhật lại dữ liệu trên giao diện. Ngược lại, hiển thị thông báo lỗi.

Quy trình này đảm bảo rằng thông tin đối tượng được xóa thông qua giao tiếp giữa giao diện, service, và controller của hệ thống.

1. Mô tả chức năng “Tìm kiếm”

Mô tả quy trình tìm kiếm đối tượng trong hệ thống của tôi có thể được mô tả như sau:

1. Tìm kiếm đối tượng theo ComboBox:

* Người dùng có thể tìm kiếm đối tượng theo thuộc tính đối tượng cần thiết trên ComboBox.
* Khi người dùng thay đổi giá trị trong các ComboBox và sự kiện “SelectedIndexChanged” được kích hoạt.
* Phương thức “SearchByMa” được gọi để thực hiện tìm kiếm.
* Dữ liệu trả về từ API endpoint “/api/Controller” được lấy thông qua phương thức “LayDS” trong Service.
* Dựa vào giá trị của ComboBox, danh sách đối tượng được lọc và hiển thị trên DataGridView.

1. Tìm kiếm đối tượng theo ComboBox và TextBox:

* Người dùng có thể tìm kiếm đối tượng với các thuộc tính trên ComboBox hoặc TextBox.
* Khi người dùng thay đổi giá trị trong các ComboBox hoặc TextBox, sự kiện “SelectedIndexChanged” của ComboBox hoặc “KeyUp” trên TextBox được kích hoạt.
* Phương thức “SelectedIndexChanged” của ComboBox hoặc “KeyUp” trên TextBox được gọi để thực hiện tìm kiếm.
* Dữ liệu trả về từ API endpoint “/api/Controller” được lấy thông qua phương thức “LayDS” trong Service.
* Dựa vào giá trị của ComboBox và TextBox, danh sách đối tượng được lọc và hiển thị trên DataGridView.

1. Tìm kiếm chung:

* Mỗi khi có sự kiện tìm kiếm, hệ thống kiểm tra giá trị ComboBox và TextBox để xác định yêu cầu tìm kiếm cụ thể.
* Nếu không có giá trị nào được chọn hoặc nhập, hệ thống sẽ load lại dữ liệu đầy đủ thông tin bằng cách gọi “LoadDataGridView()”.

1. Thông báo và xử lý lỗi: Trong quá trình thực hiện tìm kiếm, nếu có lỗi xảy ra (ví dụ: không thể lấy dữ liệu từ API), hệ thống hiển thị thông báo lỗi tương ứng.
2. Kích hoạt sự kiện tìm kiếm: Người dùng có thể kích hoạt sự kiện tìm kiếm bằng cách nhấn nút “btnTim\_Click()” hoặc nhấn phím Enter khi đang focus vào TextBox.

Chú ý:

* Các hàm “SearchData()” được sử dụng để thực hiện quá trình tìm kiếm.
* Dữ liệu được lọc và hiển thị trên DataGridView tùy thuộc vào các điều kiện tìm kiếm được thiết lập.

1. Mô tả chức năng “Xem” trên giao diện chi tiết

Mô tả quá trình “Xem” đối tượng trong hệ thống giao diện chi tiết của tôi được mô tả như sau:

Bên giao diện:

1. Khi Click vào Link Label “Chi Tiết Đơn Nhập”, “Chi Tiết Đơn Xuất” hoặc “Chi Tiết Thuốc”:

* Hệ thống sẽ lấy mã khóa chính từ TextBox tùy theo từng danh sách chi tiết.
* Nếu mã đơn nhập hợp lệ (không rỗng), tiếp tục thực hiện các bước, ngược lại hiển thị thông báo lỗi.

1. Tải dữ liệu chi tiết:

* Gọi hàm “LoadDataByMa” của mỗi UserControl tương ứng và truyền mã khóa chính vừa lấy được.
* Trong hàm “linkLb\_Click”, hệ thống gọi Service “chitietService.LayCT(ma)” để lấy danh sách chi tiết từ API theo mã khóa chính.

1. Hiển thị chi tiết trong mỗi UserControl:

* Nếu danh sách chi tiết có dữ liệu, hệ thống hiển thị thông tin chi tiết lên UserControl.
* Dữ liệu bao gồm các thuộc tính tùy mỗi danh sách chi tiết trong mỗi UserControl.
* Nếu danh sách rỗng, hiển thị thông báo thông báo không có chi tiết cho mã này.

1. Mở giao diện “FrmChiTiet” và thêm UserControl tương ứng:

* Sau khi tải dữ liệu chi tiết vào UserControl tương ứng, hệ thống tạo một đối tượng “FrmChiTiet”.
* Thêm UserControl tương ứng vào trong “FrmChiTiet”.
* Hiển thị “FrmChiTiet” để người dùng xem chi tiết.

1. Xử lý trường hợp mã không hợp lệ: Nếu mã không hợp lệ (rỗng), hệ thống hiển thị thông báo lỗi.

Bên “Service”:

1. Lấy danh sách chi tiết từ API:

* Gọi API endpoint “/api/Controller/{id}” để lấy chi tiết theo mã.
* Dữ liệu trả về từ API được xử lý để đảm bảo có thể đọc được, và kiểm tra lỗi.

1. Phân loại dữ liệu trả về:

* Dựa vào cấu trúc JSON trả về từ API, kiểm tra xem có phải là một mảng hay một đối tượng đơn.
* Parse dữ liệu JSON và trả về danh sách chi tiết nếu là mảng hoặc một đối tượng nếu là đơn.

1. Xử lý lỗi và Log:

* Xử lý các trường hợp lỗi khi gọi API hoặc parse JSON.
* Ghi Log nếu có lỗi để dễ dàng theo dõi và khắc phục sự cố.

Chú ý:

* Dữ liệu chi tiết được trả về từ API và hiển thị trên UserControl tương ứng.
* Xử lý cả trường hợp khi API trả về danh sách rỗng hoặc đối tượng rỗng.

1. Giao diện ứng dụng
2. Giao diện đăng nhập

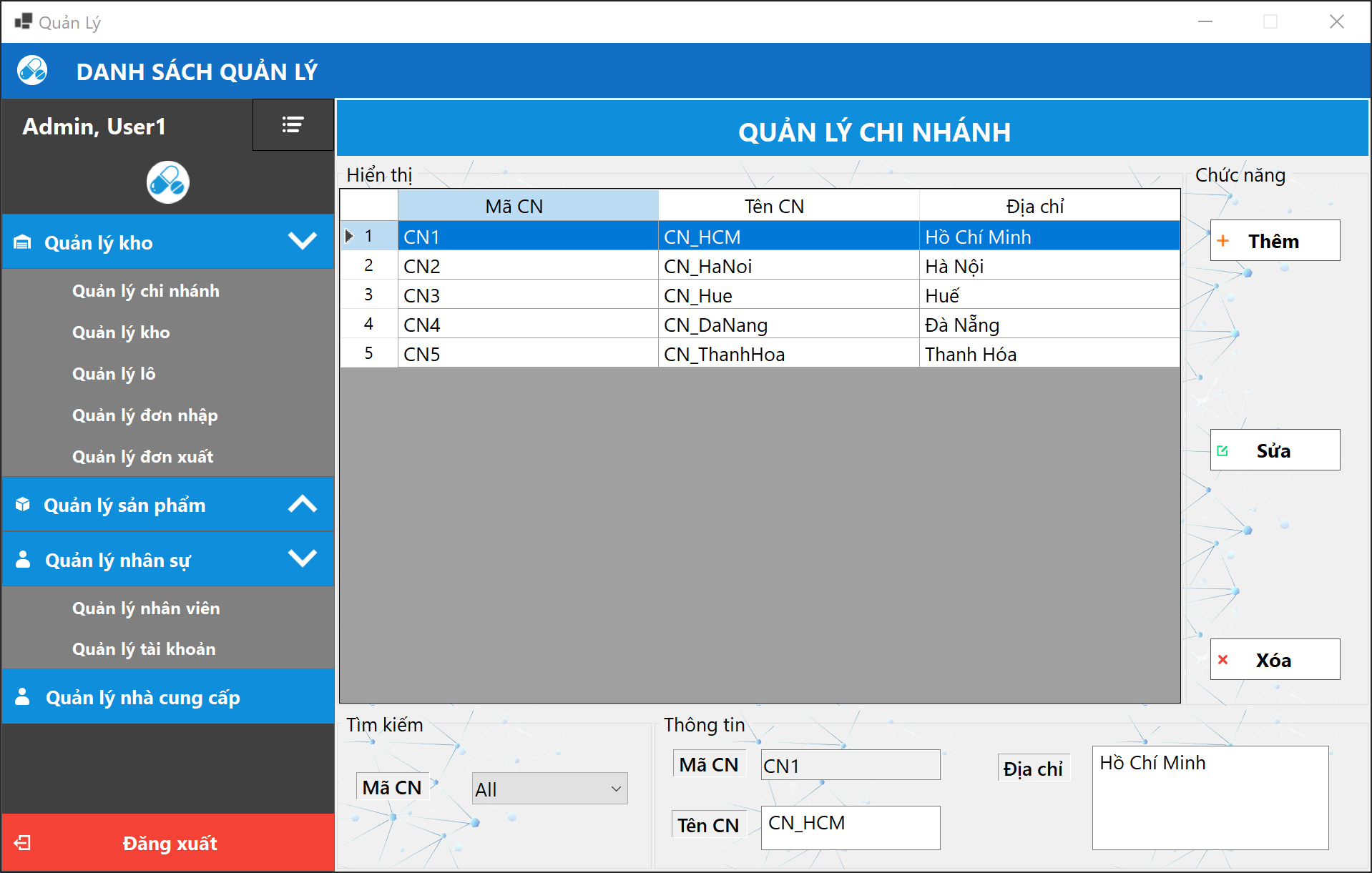
* Giao diện đăng nhập cung cấp các ô nhập liệu cho người dùng nhập thông tin đăng nhập, bao gồm UserName (Tên người dùng) và Password (Mật khẩu)
* Nút “Loign” để người dùng thực hiện quá trình đăng nhập
* Có chức năng ẩn/ hiện mật khẩu nếu người dùng muốn xem
* Khi đăng nhập, hệ thống sẽ kiểm tra thông tin người dùng đăng nhập so với cơ sở dữ liệu tài khoản có trong hệ thống. Nếu thông tin đăng nhập sai sẽ hiện thông báo cho người dùng biết lỗi sai và yêu cầu người dùng kiểm tra lại thông tin.
* Hiển thị thông báo lỗi nếu lỗi về đăng nhập, như tài khoản không tồn tại, mật khẩu không đúng, mật khẩu hoặc tên người dùng chưa nhập,…



Hình 4.2 Giao diện đăng nhập

1. Giao diện chính

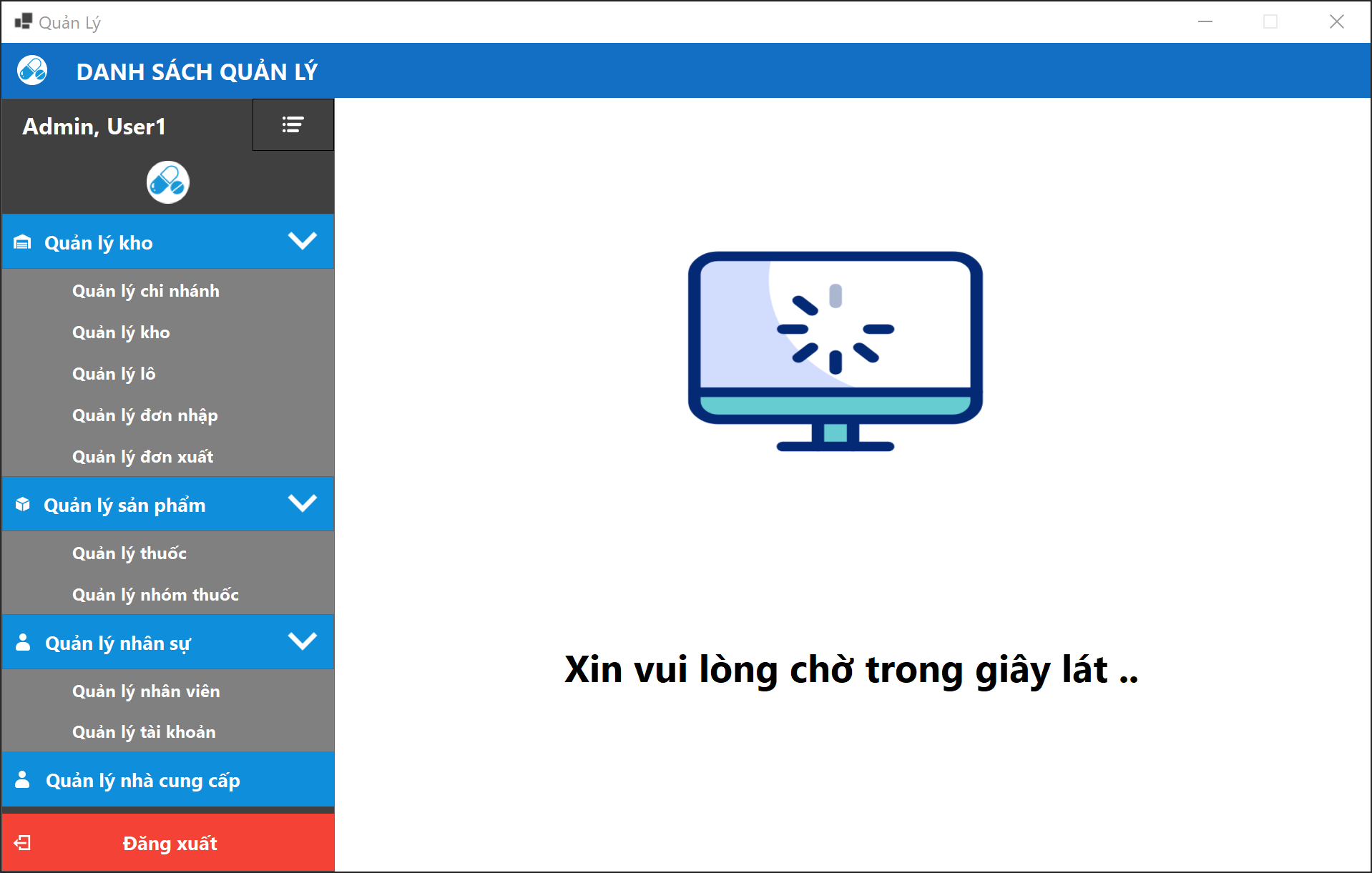
* Giao diện chính có giao diện menu dọc với danh sách menu chính để mở các giao diện quản lý khác nhau
* Các nút menu được nhóm theo 4 nhóm chính là: Quản lý kho, Quản lý sản phẩm, Quản lý nhân sự, Quản lý nhà cung cấp
* Các nút chính trong SubMenu tương ứng với các giao diện, ví dụ nút “Quản lý thuốc” sẽ mở giao diện thuốc, nút “Quản lý kho” sẽ mở giao diện kho,…
* Giao diện menu có thể hiển thị chức năng mở rộng/ ẩn các SubMenu, giúp người dùng linh hoạt, hay các chức năng liên quan giúp dễ dàng chuyển đổi các giao diện quản lý
* Trên đầu giao diện có hiển thị một label để hiển thị thông tin người dùng đăng nhập. Nếu người dùng là nhân viên, hiển thị “NV, Tên người đăng nhập”; nếu là Admin, hiển thị “Admin, Tên người đăng nhập”.
* Có chức năng “Đăng xuất” cho phép người dùng đăng xuất khỏi hệ thống và quay lại giao diện đăng nhập.



Hình 4.3 Giao diện chính

1. Giao diện Loading

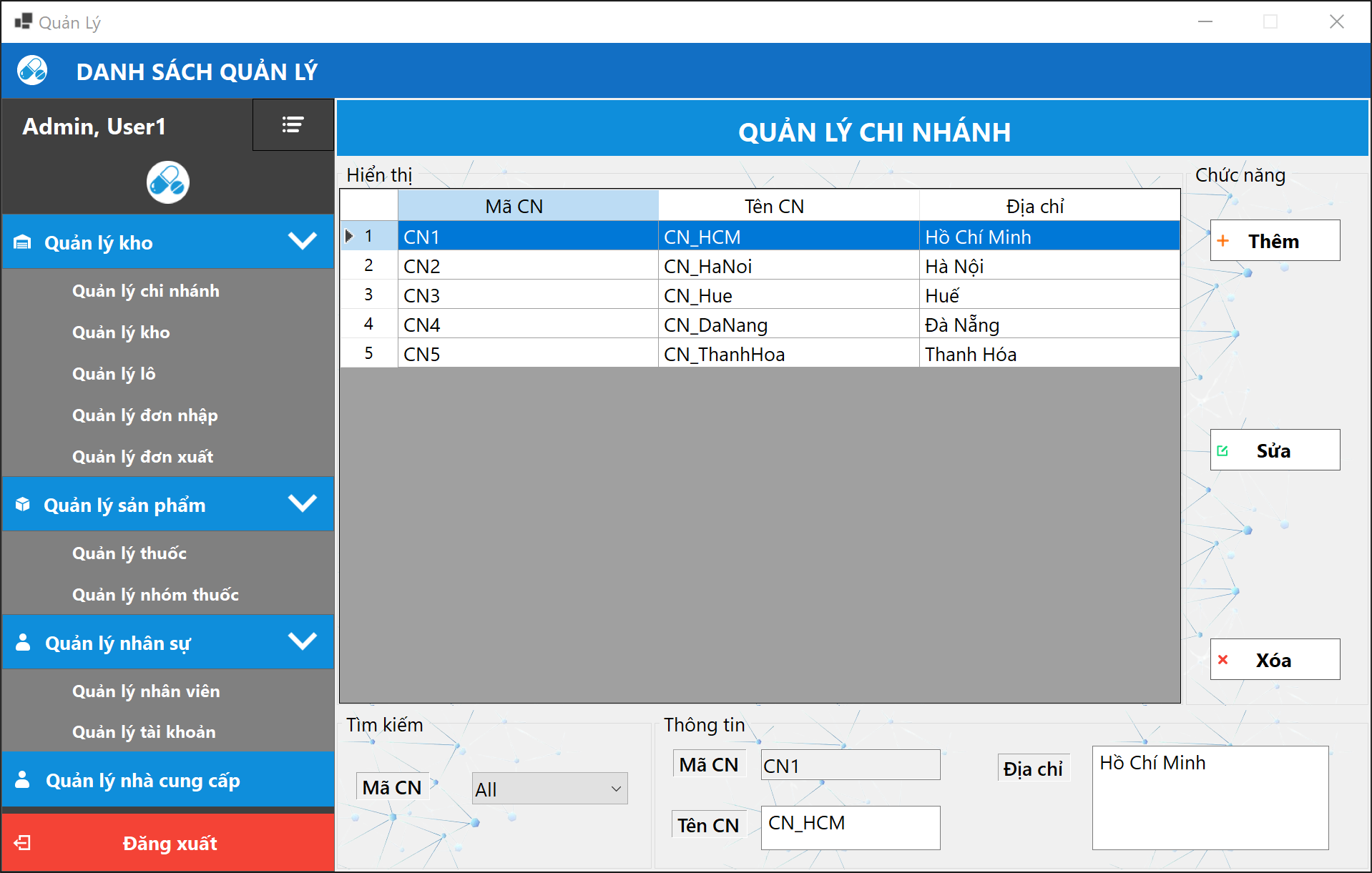
* Giao diện Loading hiển thị hiệu ứng animation chuyển động vòng quay trên máy tính để biết dữ liệu đang được tải.
* Hiệu ứng chuyển động giúp người dùng nhận biết rằng ứng dụng vẫn đang chạy, tránh gây ứng dụng không phản hồi trên giao diện.
* Hiển thị dòng văn bản “Xin vui lòng chờ trong giây lát…” để thông báo người dùng về quá trình tải dữ liệu.
* Sử dụng màu sắc hài hòa với giao diện chính của ứng dụng nhằm tạo cảm giác nhất quán.
* Hiển thị giao diện Loading trong khoảng thời gian ngắn và sẽ tự động ẩn form khi quá trình tải dữ liệu giao diện quản lý hoàn tất.
* Form Loading không chặn khả năng tương tác với giao diện chính. Cho phép người dùng sử dụng các chức năng cơ bản.



Hình 4.4 Giao diện Loading

1. Giao diện quản lý chi nhánh

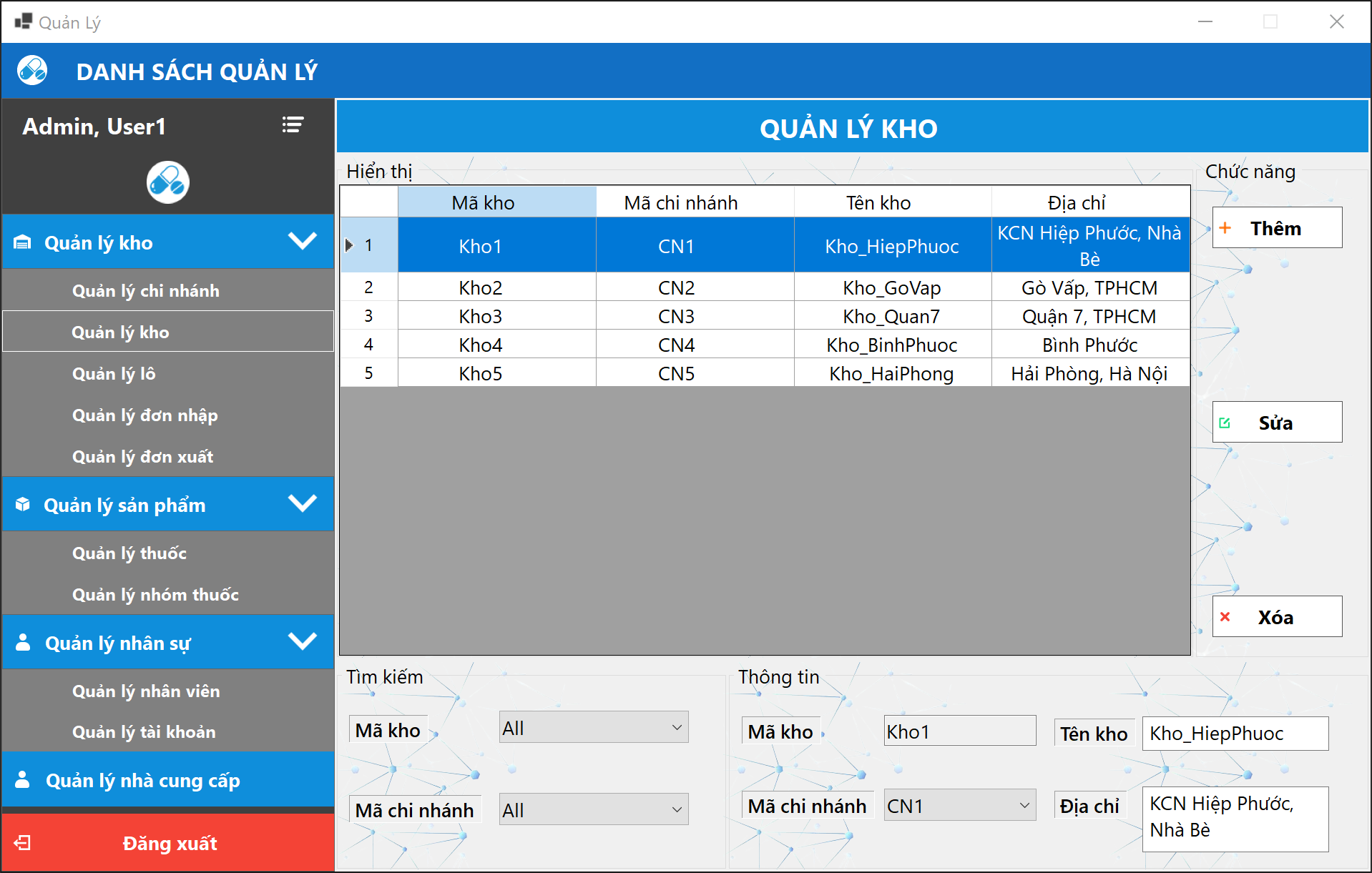
* Tiêu đề trang ghi rõ chức năng “Quản lý chi nhánh”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” – Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các chi nhánh có số thứ tự với các thông tin như sau: Mã chi nhánh, Tên chi nhánh, Địa chỉ
* Các thông tin mỗi dòng trên danh sách khi được chọn đều sẽ hiển thị thông tin tương ứng trên các ô nhập liệu.
* Nút “Thêm” ở góc trên cùng bên phải để thêm chi nhánh mới sau khi nhập đủ các giá trị, mã chi nhánh tự động nên chỉ điền các giá trị khác rồi “Thêm”, sẽ không trùng mã.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan có mã chi nhánh sẽ chuyển sang mã chi nhánh “CN1”
* Các chức năng “Thêm”, “Sửa”, “Xóa” khi người dùng thực hiện đều thể hiện thông báo thành công hoặc thất bại.
* Phần tìm kiếm giúp người dùng có thể tìm kiếm dựa trên mã chi nhánh, khi người dùng chọn giá trị muốn tìm thì sẽ hiện kết quả trên danh sách.
* Giao diện quản lý chi nhánh đã thiết kế trực quan, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách thuận lợi.



Hình 4.5 Giao diện quản lý chi nhánh

1. Giao diện quản lý kho

* Tiêu đề trang ghi rõ chức năng “Quản lý kho”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các kho có số thứ tự với các thông tin như sau: Mã chi nhánh, Mã kho, Tên kho, Địa chỉ
* Các thông tin mỗi dòng trên danh sách khi được chọn đều sẽ hiển thị thông tin tương ứng trên các ô nhập liệu hoặc Combo box.
* Nút “Thêm” ở góc trên cùng bên phải để thêm kho mới sau khi nhập đủ các giá trị, mã kho tự động nên chỉ điền các giá trị khác rồi “Thêm”, sẽ không trùng mã.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan có mã chi nhánh sẽ chuyển sang mã kho “Kho1”
* Các chức năng “Thêm”, “Sửa”, “Xóa” khi người dùng thực hiện đều thể hiện thông báo thành công hoặc thất bại.
* Phần tìm kiếm giúp người dùng có thể tìm kiếm dựa trên mã chi nhánh, mã kho khi người dùng chọn giá trị muốn tìm thì sẽ hiện kết quả trên danh sách.
* Giao diện quản lý kho đã thiết kế đơn giản, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách thuận lợi.

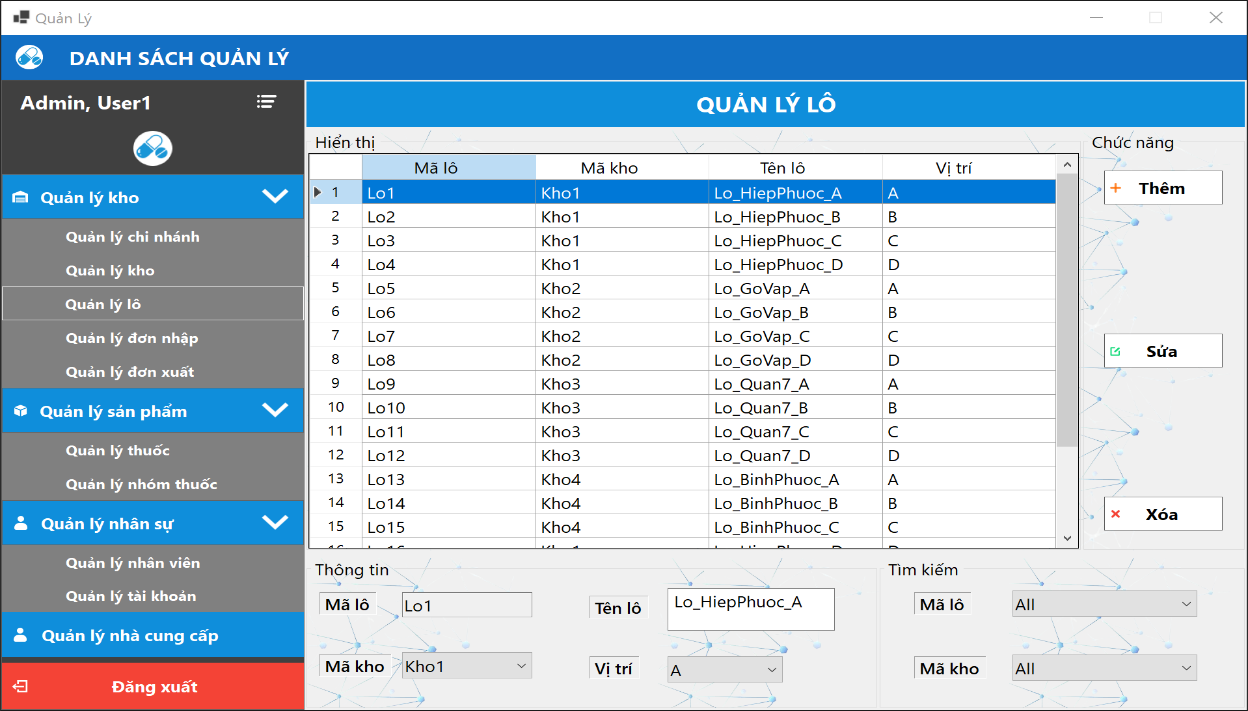


Hình 4.6 Giao diện quản lý kho

1. Giao diện quản lý lô

* Tiêu đề trang ghi rõ chức năng “Quản lý lô”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các lô có số thứ tự với các thông tin như sau: Mã lô, Mã kho, Tên lô, Vị trí.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu hoặc ComboBox dưới danh sách.
* Nút “Thêm” ở góc trên cùng bên phải để thêm lô mới sau khi nhập đủ giá trị. Mã lô tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan cũng sẽ xóa theo.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên Mã kho, Mã lô khi người dùng chọn giá trị muốn tìm kiếm.
* Thiết kế đơn giản, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách thuận lợi.

Giao diện quản lý lô được xây dựng để cung cấp trải nghiệm người dùng thuận lợi và hiệu quả trong quá trình quản lý thông tin các lô.

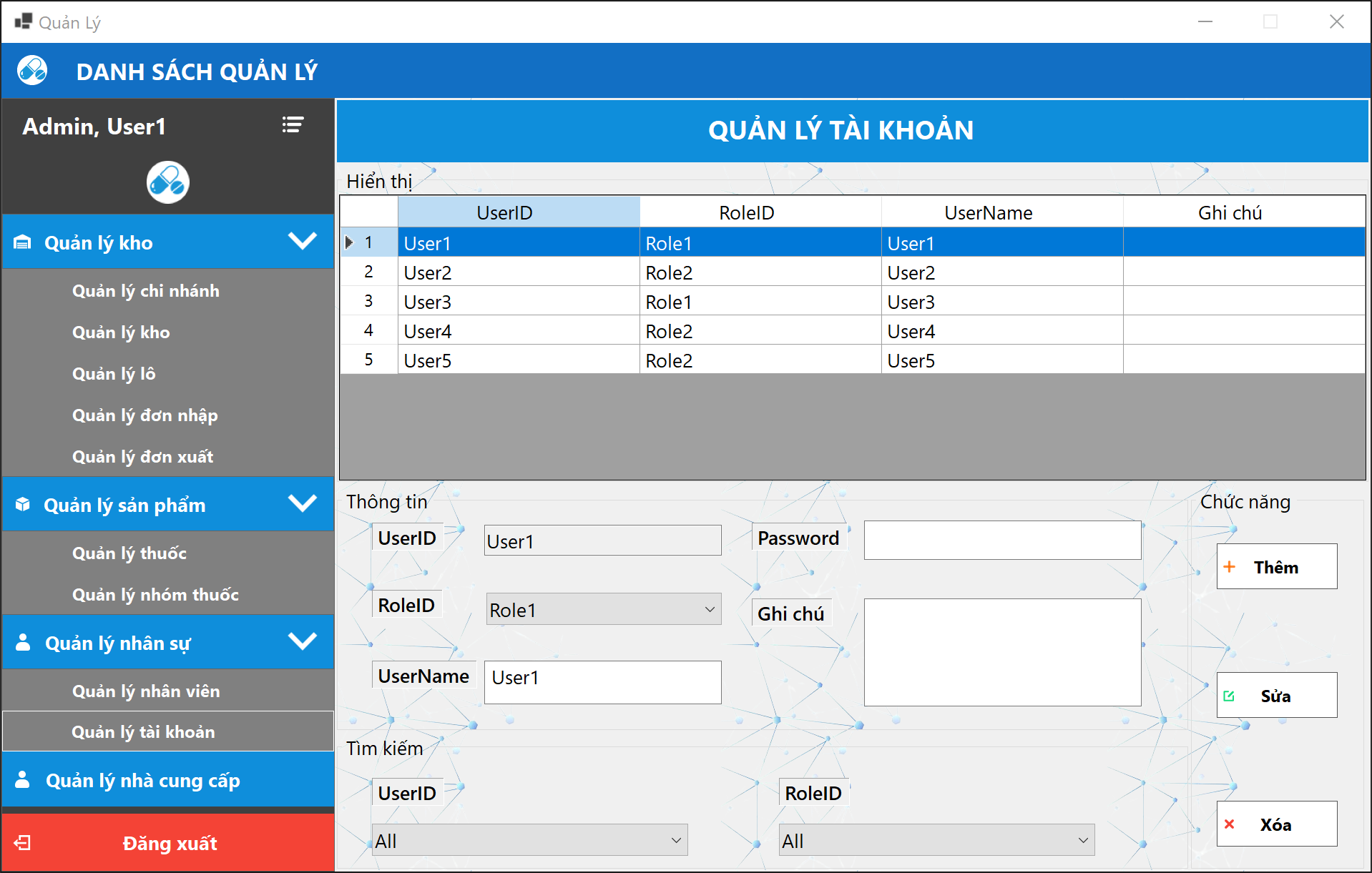


Hình 4.7 Giao diện quản lý lô

1. Giao diện quản lý tài khoản

* Tiêu đề trang ghi rõ chức năng “Quản lý tài khoản”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các tài khoản có số thứ tự với các thông tin như sau: UserID, RoleID, UserName, Ghi chú. Thông tin Password sẽ được ẩn để bảo mật.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu hoặc ComboBox dưới danh sách.
* Nút “Thêm” để thêm một tài khoản mới sau khi nhập đủ giá trị. UserID tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan có UserID sẽ trở thành giá trị NULL.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên UserID và RoleID khi người dùng chọn giá trị muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Thiết kế đơn giản, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách thuận lợi.

Giao diện quản lý tài khoản được thiết kế để cung cấp trải nghiệm sử dụng của người dùng thuận lợi và hiệu quả trong quá trình quản lý thông tin các tài khoản.

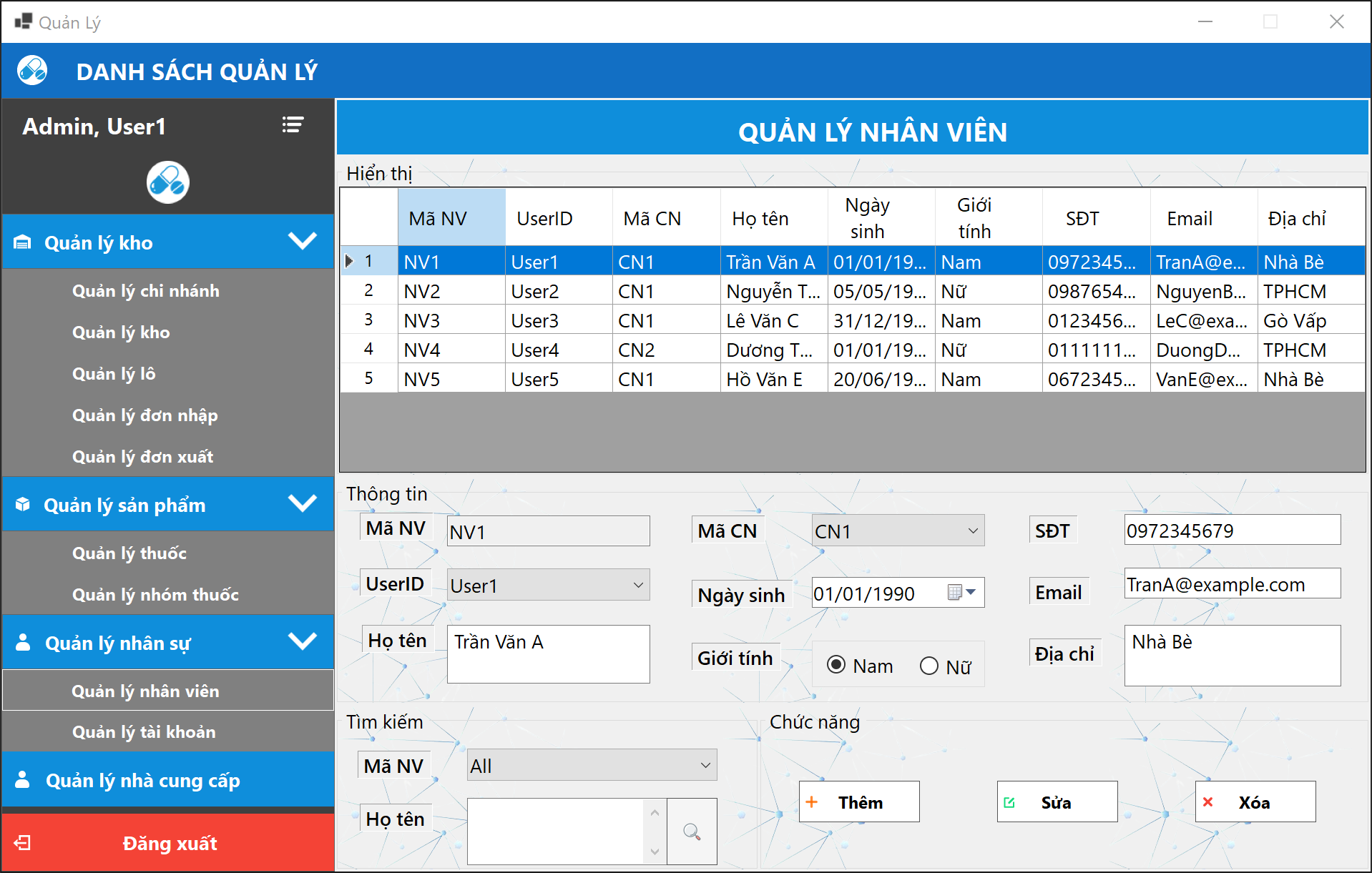


Hình 4.8 Giao diện quản lý tài khoản

1. Giao diện quản lý nhân viên

* Tiêu đề trang ghi rõ chức năng “Quản lý nhân viên”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các nhân viên có số thứ tự với các thông tin như sau: Mã nhân viên, Họ tên, Mã chi nhánh, UserID, Ngày sinh, Giới tính, Số điện thoại, Email, Địa chỉ. Có các điều kiện kiểm tra như: một nhân viên chỉ có một tài khoản, kiểm tra số điện thoại hoặc email hợp lệ,…
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu, ComboBox hoặc Radio Button dưới danh sách.
* Nút “Thêm” để thêm một nhân viên mới sau khi nhập đủ giá trị. Mã nhân viên tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan sẽ được xóa theo.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã nhân viên” và “Họ tên” khi người dùng chọn giá trị hay nhập tên muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Thiết kế đơn giản, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách thuận lợi.

Giao diện quản lý nhân viên được thiết kế để cung cấp trải nghiệm sử dụng của người dùng thân thiện và hiệu quả trong quá trình quản lý thông tin các nhân viên.

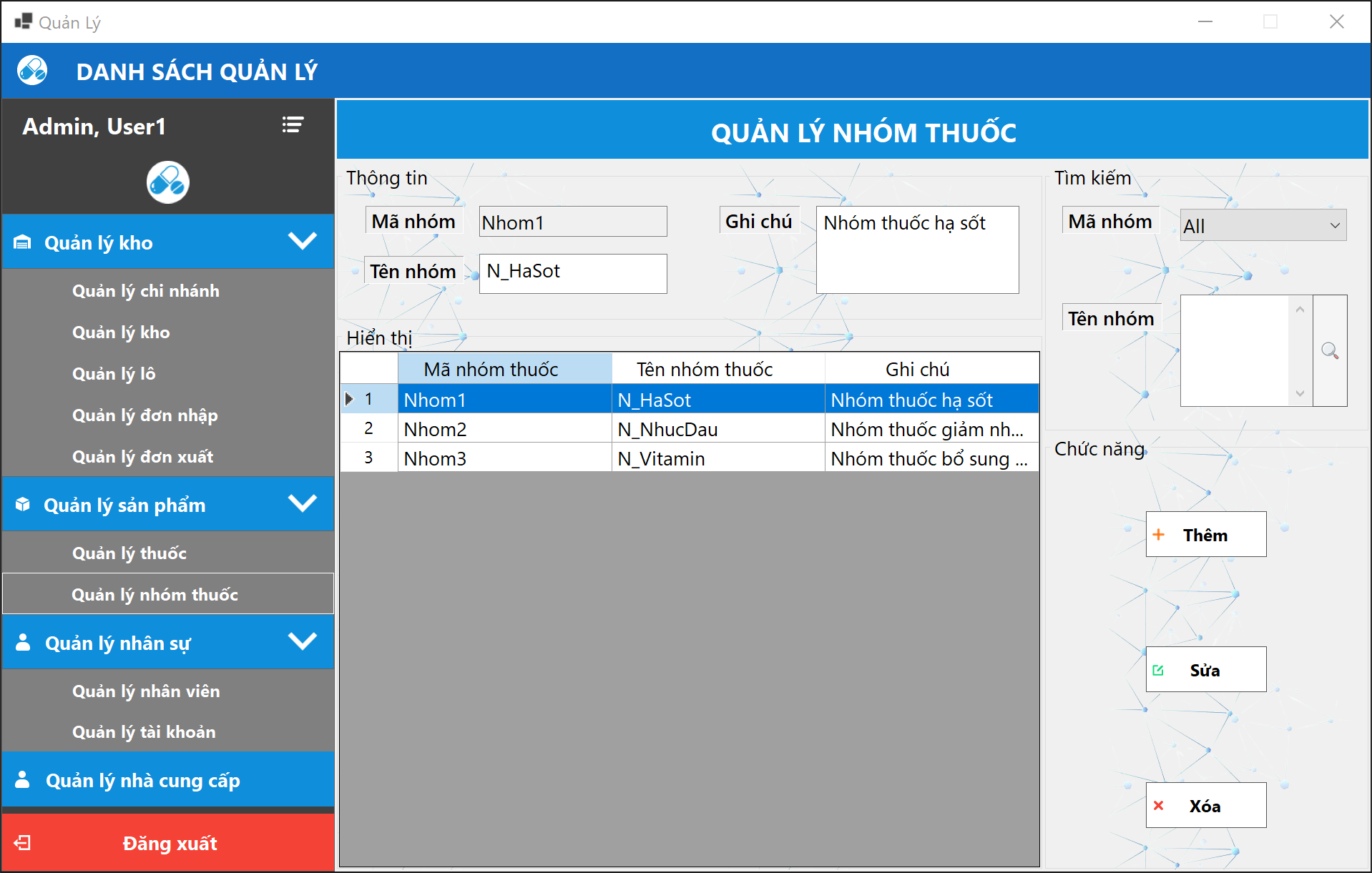


Hình 4.9 Giao diện quản lý nhân viên

1. Giao diện quản lý nhóm thuốc

* Tiêu đề trang ghi rõ chức năng “Quản lý nhóm thuốc”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các nhóm thuốc có số thứ tự với các thông tin như sau: Mã nhóm thuốc, Tên nhóm thuốc, Ghi chú.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu trên danh sách.
* Nút “Thêm” để thêm một nhóm thuốc mới sau khi nhập đủ giá trị. Mã nhóm thuốc tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan sẽ cho “Mã nhóm thuốc” là giá trị NULL.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã nhóm thuốc” và “Tên nhóm” khi người dùng chọn giá trị hay nhập tên nhóm muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Thiết kế đơn giản, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện quản lý nhóm thuốc được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý thông tin các nhóm thuốc.

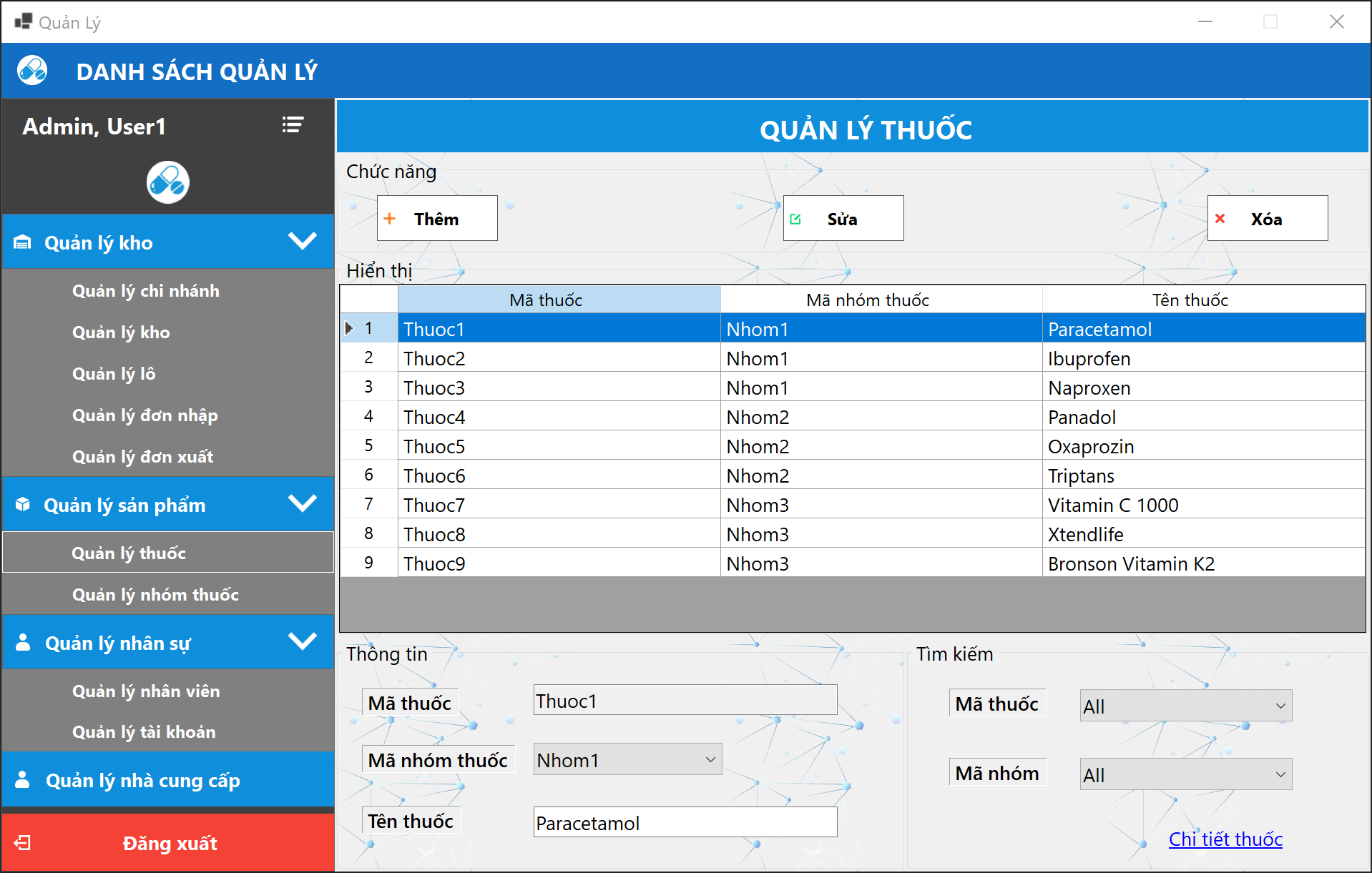


Hình 4.10 Giao diện quản lý nhóm thuốc

1. Giao diện quản lý thuốc

* Tiêu đề trang ghi rõ chức năng “Quản lý thuốc”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các thuốc có số thứ tự với các thông tin như sau: Mã nhóm thuốc, Mã thuốc, Tên thuốc.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu trên danh sách.
* Nút “Thêm” để thêm một thuốc mới sau khi nhập đủ giá trị. Mã thuốc tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan sẽ được xóa theo.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã nhóm thuốc” và “Mã thuốc” khi người dùng chọn giá trị muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Giao diện quản lý thuốc có một Link (siêu liên kết) để mở giao diện chi tiết thuốc cho mỗi dòng trên danh sách. Hiển thị giao diện thành công khi tồn tại mã thuốc trong danh sách chi tiết hoặc thông báo lỗi khi không có.
* Thiết kế thân thiện, dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện quản lý thuốc được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý thông tin các thuốc.

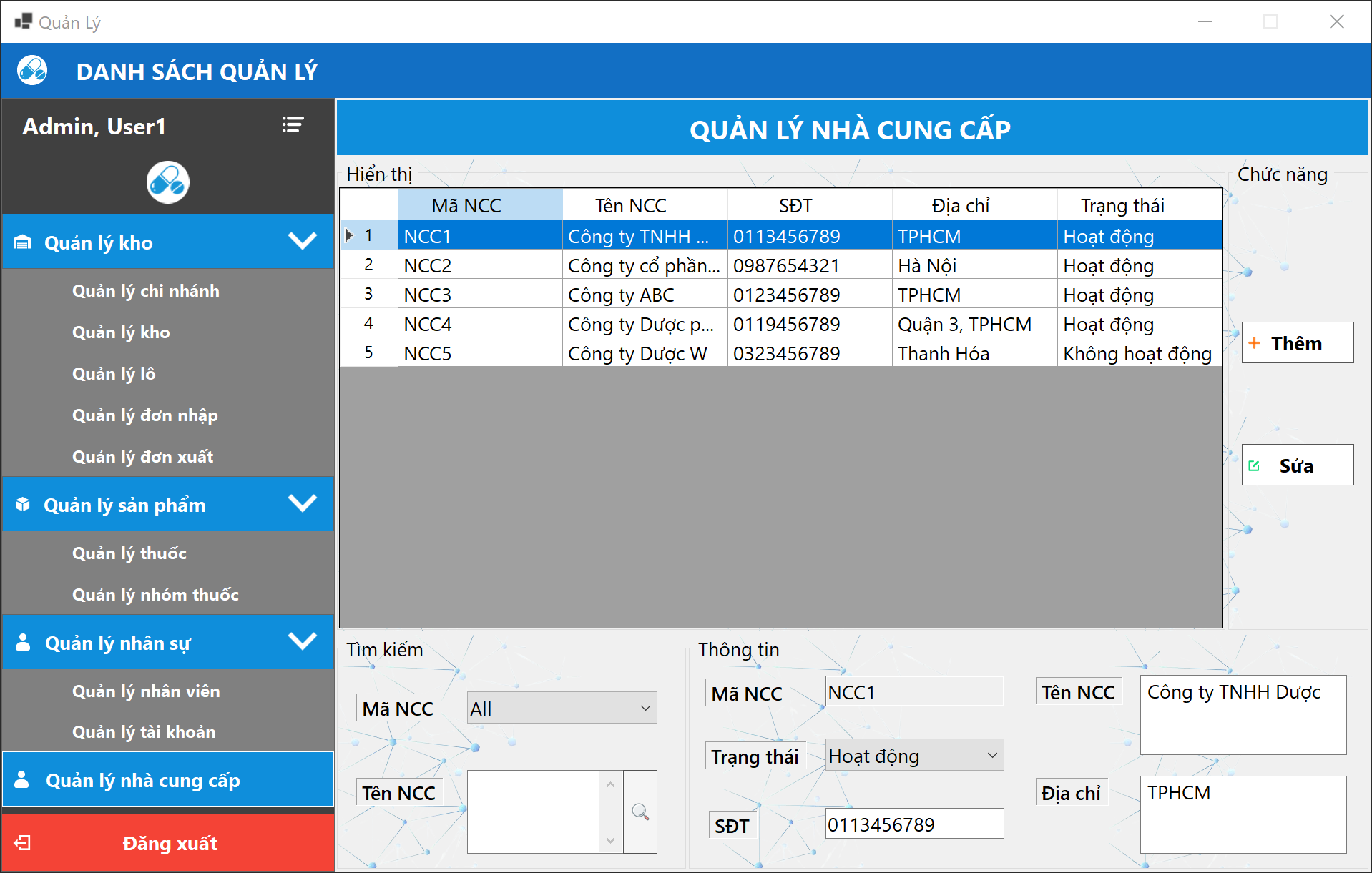


Hình 4.11 Giao diện quản lý thuốc

1. Giao diện quản lý nhà cung cấp

* Tiêu đề trang ghi rõ chức năng “Quản lý nhà cung cấp”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các nhà cung cấp có số thứ tự với các thông tin như sau: Mã nhà cung cấp, Tên nhà cung cấp, Số điện thoại, Địa chỉ, Trạng thái (hiển thị trạng thái nhà cung cấp đó còn “Hoạt động” hay “Không hoạt động”).
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu trên danh sách.
* Nút “Thêm” để thêm một nhà cung cấp mới sau khi nhập đủ giá trị. Mã nhà cung cấp tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” chức năng tương ứng.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã nhà cung cấp” và “Tên nhà cung cấp” khi người dùng chọn giá trị hay nhập giá trị muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Thiết kế giao diện dễ sử dụng và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện quản lý nhà cung cấp được thiết kế để cung cấp trải nghiệm người dùng đơn giản và hiệu quả trong quá trình quản lý thông tin các nhà cung cấp.

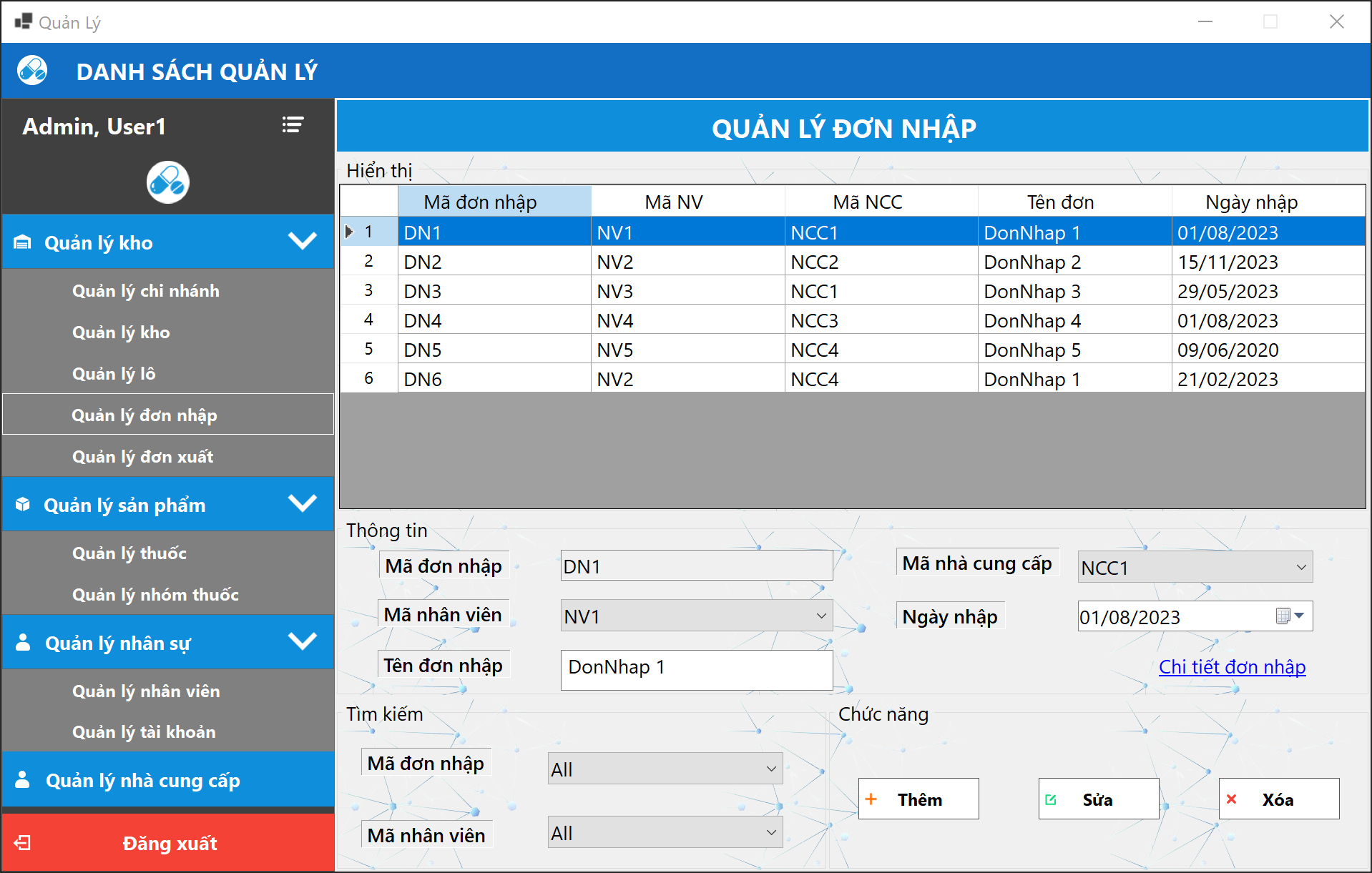


Hình 4.12 Giao diện quản lý nhà cung cấp

1. Giao diện quản lý đơn nhập

* Tiêu đề trang ghi rõ chức năng “Quản lý đơn nhập”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các đơn nhập có số thứ tự với các thông tin như sau: Mã nhà cung cấp, Mã đơn nhập, Mã nhân viên, Tên đơn nhập, Ngày nhập đơn.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu trên danh sách.
* Nút “Thêm” để thêm một đơn nhập mới sau khi nhập đủ giá trị. Mã đơn nhập tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan sẽ được xóa theo.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã đơn nhập” và “Mã nhân viên” khi người dùng chọn giá trị muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Giao diện quản lý đơn nhập có một Link (siêu liên kết) để mở giao diện chi tiết đơn nhập cho mỗi dòng trên danh sách. Hiển thị giao diện thành công khi tồn tại mã đơn nhập trong danh sách chi tiết hoặc thông báo lỗi khi không có.
* Thiết kế giao diện đơn giản, dễ nhìn và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện quản lý đơn nhập được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý thông tin các đơn nhập.

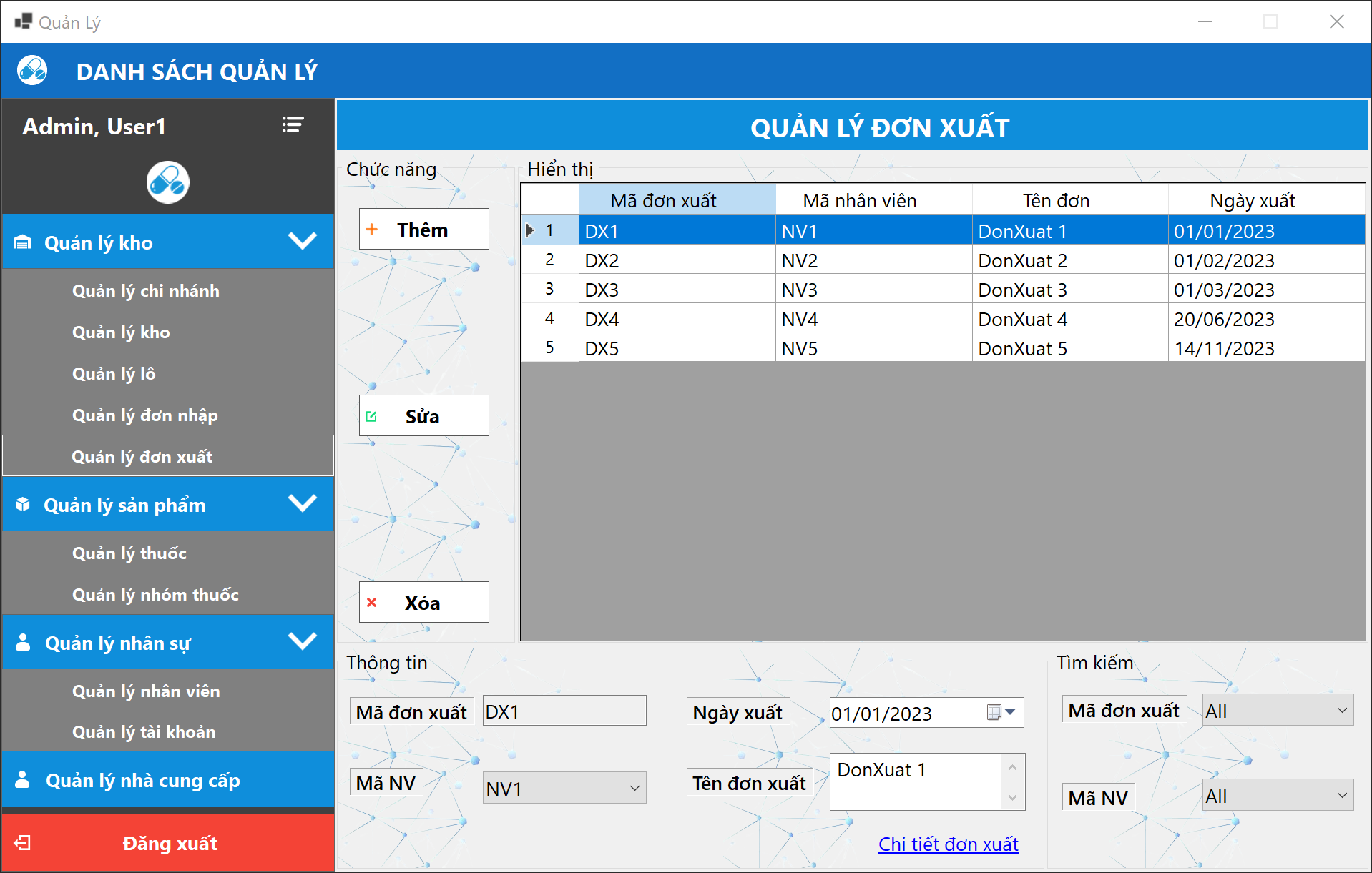


Hình 4.13 Giao diện quản lý đơn nhập

1. Giao diện quản lý đơn xuất

* Tiêu đề trang ghi rõ chức năng “Quản lý đơn xuất”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem, Thêm và Tìm kiếm.
* Hiển thị danh sách các đơn xuất có số thứ tự với các thông tin như sau: Mã đơn xuất, Mã nhân viên, Tên đơn xuất, Ngày xuất đơn.
* Khi chọn một dòng trong danh sách, hiển thị thông tin tương ứng trên các ô nhập liệu trên danh sách.
* Nút “Thêm” để thêm một đơn xuất mới sau khi nhập đủ giá trị. Mã đơn xuất tự động, người dùng chỉ cần điền các giá trị khác và nhấn “Thêm”.
* Mỗi dòng trong danh sách có nút “Sửa” và “Xóa” để thực hiện các chức năng tương ứng. Chức năng “Xóa” cho phép người dùng xóa nhiều dòng trên danh sách và khi xóa các danh sách liên quan sẽ được xóa theo.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Thêm”, “Sửa”, “Xóa”.
* Chức năng tìm kiếm giúp người dùng tìm kiếm dựa trên “Mã đơn xuất” và “Mã nhân viên” khi người dùng chọn giá trị muốn tìm kiếm thì sẽ hiển thị kết quả trên danh sách.
* Giao diện quản lý đơn xuất có một Link (siêu liên kết) để mở giao diện chi tiết đơn xuất cho mỗi dòng trên danh sách. Hiển thị giao diện thành công khi tồn tại mã đơn nhập trong danh sách chi tiết hoặc thông báo lỗi khi không có.
* Thiết kế giao diện hài hòa, dễ nhìn và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện quản lý đơn xuất được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý thông tin các đơn xuất.



Hình 4.14 Giao diện quản lý đơn xuất

1. Giao diện chi tiết đơn nhập

* Tiêu đề trang ghi rõ chức năng “Chi tiết đơn nhập”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem.
* Hiển thị danh sách các chi tiết đơn nhập với các thông tin như sau: Mã đơn nhập, Mã lô, Mã thuốc, Số lượng thuốc, Ghi chú.
* Khi chọn một dòng trong giao diện danh sách đơn nhập, nếu có tồn tại mã đơn nhập trong dữ liệu chi tiết đơn nhập thì hiển thị thông tin tương ứng trên các ô nhập liệu hoặc ComboBox. Nếu không tồn tại thì thông báo lỗi và kết thúc quá trình.
* Mỗi dòng trong danh sách có nút “Sửa” chức năng tương ứng.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Sửa”.
* Thiết kế giao diện chi tiết nhỏ gọn, dễ nhìn và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện chi tiết đơn nhập được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý chi tiết đơn nhập.

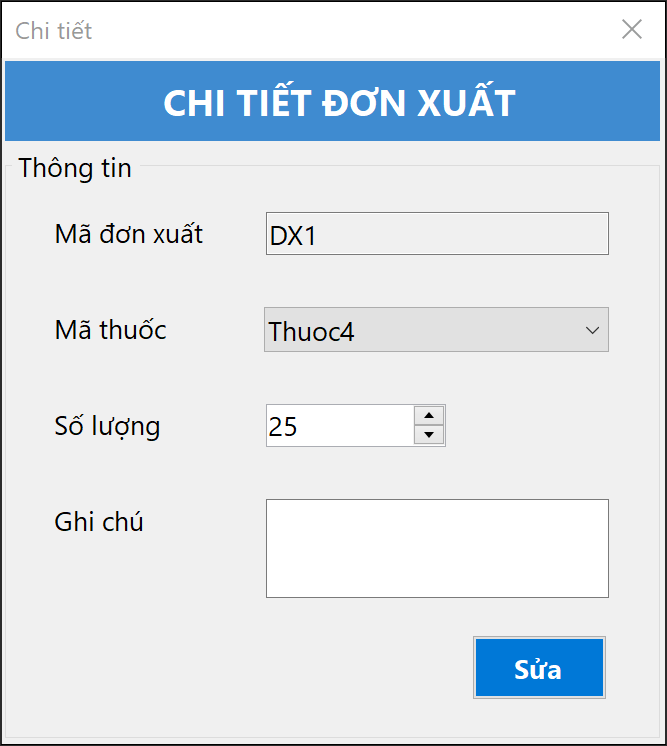


Hình 4.15 Giao diện chi tiết đơn nhập

1. Giao diện chi tiết đơn xuất

* Tiêu đề trang ghi rõ chức năng “Chi tiết đơn xuất”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem.
* Hiển thị danh sách các chi tiết đơn xuất với các thông tin như sau: Mã đơn xuất, Mã thuốc, Số lượng, Ghi chú.
* Khi chọn một dòng trong giao diện danh sách đơn xuất, nếu có tồn tại mã đơn xuất trong dữ liệu chi tiết đơn xuất thì hiển thị thông tin tương ứng trên các ô nhập liệu hoặc ComboBox. Nếu không tồn tại thì thông báo lỗi và kết thúc quá trình.
* Mỗi dòng trong danh sách có nút “Sửa” chức năng tương ứng.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Sửa”.
* Thiết kế giao diện chi tiết nhỏ gọn, dễ nhìn và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện chi tiết đơn xuất được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý chi tiết đơn xuất.

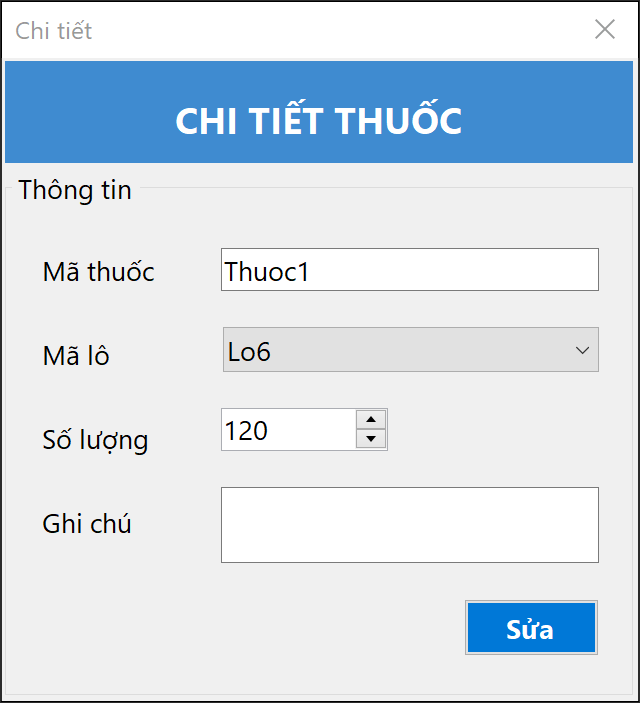


Hình 4.16 Giao diện chi tiết đơn xuất

1. Giao diện chi tiết đơn thuốc

* Tiêu đề trang ghi rõ chức năng “Chi tiết thuốc”
* Sẽ hiển thị thông báo các chức năng được cấp quyền hoặc không tùy vai trò người dùng, “Admin” - Tất cả chức năng, “Nhân viên” - Xem.
* Hiển thị danh sách các chi tiết thuốc với các thông tin như sau: Mã thuốc, Mã lô, Số lượng, Ghi chú.
* Khi chọn một dòng trong giao diện danh sách quản lý thuốc, nếu có tồn tại mã thuốc trong dữ liệu chi tiết thuốc thì hiển thị thông tin tương ứng trên các ô nhập liệu hoặc ComboBox. Nếu không tồn tại thì thông báo lỗi và kết thúc quá trình.
* Mỗi dòng trong danh sách có nút “Sửa” chức năng tương ứng.
* Hiển thị thông báo thành công hoặc thất bại sau khi thực hiện các chức năng “Sửa”.
* Thiết kế giao diện chi tiết nhỏ gọn, đơn giản và hiệu quả để người dùng có thể thao tác một cách dễ dàng.

Giao diện chi tiết thuốc được thiết kế để cung cấp trải nghiệm người dùng thuận tiện và hiệu quả trong quá trình quản lý chi tiết thuốc.



Hình 4.17 Giao diện chi tiết thuốc

1. Kết quả đạt được

Sử dụng công nghệ C#, ASP.NET và .NET Core Framework, Entity Framework, RESTful service/Web API, SQL Server với các công việc:

* Chức năng CRUD: Xây dựng chức năng thêm, sửa, xóa và tìm kiếm dữ liệu sử dụng C#, ASP.NET và Entity Framework.
* Giao diện Windows Form Apps: Thiết kế giao diện ứng dụng Windows Form Apps để cung cấp trải nghiệm người dùng thân thiện khi thao tác với các chức năng CRUD.
* Thiết kế Database: Xây dựng cơ sở dữ liệu sử dụng SQL Server và Entity Framework Code First để đảm bảo quản lý hiệu quả và tương tác dữ liệu linh hoạt.
* RESTful API: Viết các RESTful API sử dụng ASP.NET Core để cung cấp các phương thức GET, POST, PUT và DELETE để thao tác dữ liệu.
* Tích hợp Web API vào .NET và Entity Framework: Kết hợp Web API vào ứng dụng .NET bằng cách sử dụng HTTP requests để gọi và xử lý dữ liệu từ các endpoint đã thiết kế.

Sử dụng Entity Framework để tương tác với cơ sở dữ liệu thông qua Web API, đảm bảo tính nhất quán và bảo mật dữ liệu.

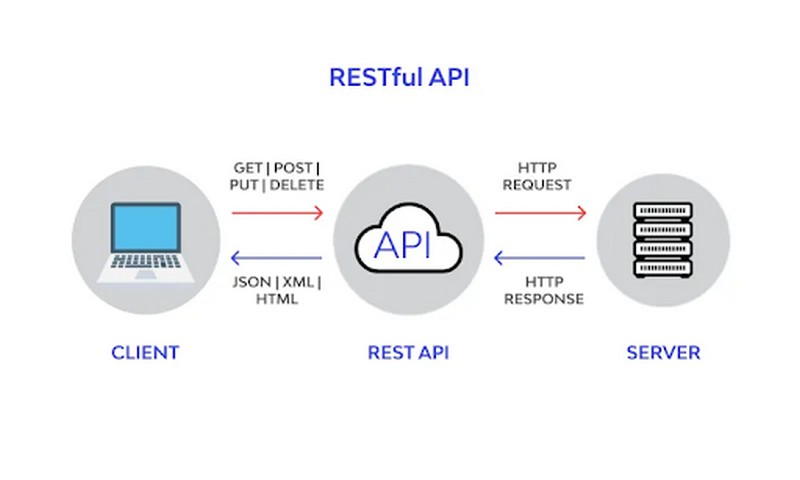
Với những điều chỉnh trên, dự án sẽ trở nên rõ ràng và toàn diện hơn các cách tích hợp các công nghệ và thực hiện các chức năng cần thiết trong một ứng dụng sử dụng C#, ASP.NET, .NET Core Framework, Entity Framework, RESTful service/Web API và SQL Server.

CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC SAU ĐỢT THỰC TẬP

1. Những kiến thức lý thuyết được củng cố

* Kiến thức về .NET, Entity Framework
* Kiến thức về RESTful API

1. Những kỹ năng thực hành đã học được
2. Hoạt động của RESTful API



Hình 5.1 Mô hình hoạt động RESTful API

RESTful API (Application Programming Interface) là một kiểu kiến trúc thiết kế API cho các ứng dụng web. Các thành phần chính của RESTful API bao gồm Client (người sử dụng API), REST API (dịch vụ web), và Server (nơi lưu trữ và xử lý dữ liệu). Dưới đây là cách mà các thành phần này tương tác với nhau:

1. Client (Người sử dụng API):

* Gửi HTTP Request: Người sử dụng gửi các HTTP request (GET, POST, PUT, DELETE) đến REST API. Request này chứa thông tin cần thiết như endpoint (địa chỉ của tài nguyên), headers, và dữ liệu (nếu có).
* Định dạng dữ liệu: Các dữ liệu thường được truyền dưới dạng JSON hoặc XML, đôi khi có thể là các loại dữ liệu khác như form-urlencoded.
* Xử lý kết quả: Người sử dụng xử lý kết quả trả về từ REST API. Thông thường, kết quả là một phản hồi chứa dữ liệu được yêu cầu hoặc thông báo về trạng thái của yêu cầu.

1. REST API (Dịch vụ Web):

* Routing và Endpoint: REST API định tuyến các request đến các endpoint tương ứng. Mỗi endpoint thường đại diện cho một tài nguyên hoặc một hành động cụ thể trên tài nguyên.
* Xử lý yêu cầu: API nhận và xử lý các yêu cầu từ Client. Quá trình này bao gồm xác thực (nếu cần), xử lý logic kinh doanh, và truy xuất dữ liệu từ Server.
* Tạo phản hồi: Sau khi xử lý yêu cầu, REST API tạo ra một phản hồi HTTP chứa thông tin cần thiết, bao gồm cả dữ liệu yêu cầu hoặc trạng thái của yêu cầu.

1. Server (Nơi lưu trữ và Xử lý dữ liệu):

* Lưu trữ dữ liệu: Server chịu trách nhiệm lưu trữ dữ liệu và thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu theo yêu cầu của REST API.
* Xử lý Logic kinh doanh: Server thực hiện logic kinh doanh cần thiết, có thể bao gồm xử lý nghiệp vụ phức tạp như xử lý thanh toán, quản lý tài khoản người dùng, và nhiều chức năng khác.
* Phản hồi tới REST API: Server tạo ra phản hồi cho REST API, chứa kết quả của thao tác (thành công, lỗi) và dữ liệu liên quan (nếu có).

Tóm lại, quá trình này tạo ra một chuỗi liên kết giữa Client, REST API, và Server, cho phép họ tương tác và trao đổi thông tin một cách hiệu quả qua giao thức HTTP.

1. Cách sử dụng EF Core Power Tools

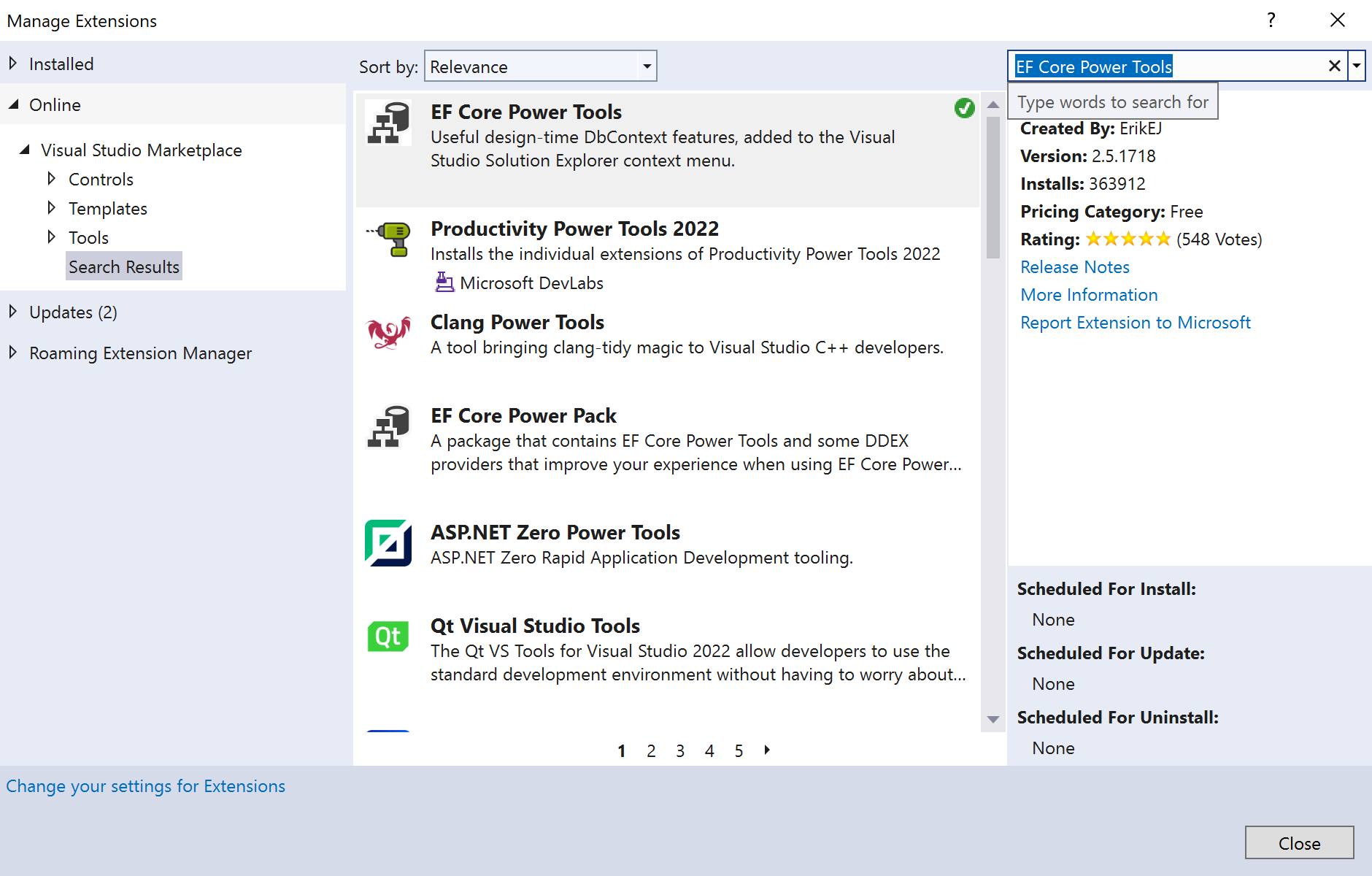
Entity Framework Core Power Tools là một bộ công cụ mở rộng cho Visual Studio giúp làm việc với Entity Framework Core trở nên dễ dàng hơn. Dưới đây là hướng dẫn cơ bản về cách sử dụng EF Core Power Tools khi tôi sử dụng trong dự án:

**Bước 1**. **Cài đặt EF Core Power Tools:**

Mở Visual Studio.

Chọn “Extensions” -> “Manage Extensions.”

Tìm “EF Core Power Tools” trong trình quản lý Extension và cài đặt nó.



Hình 5.2 Cài đặt EF Core Power Tools

**Bước 2.** **Tạo đối tượng DbContext:**

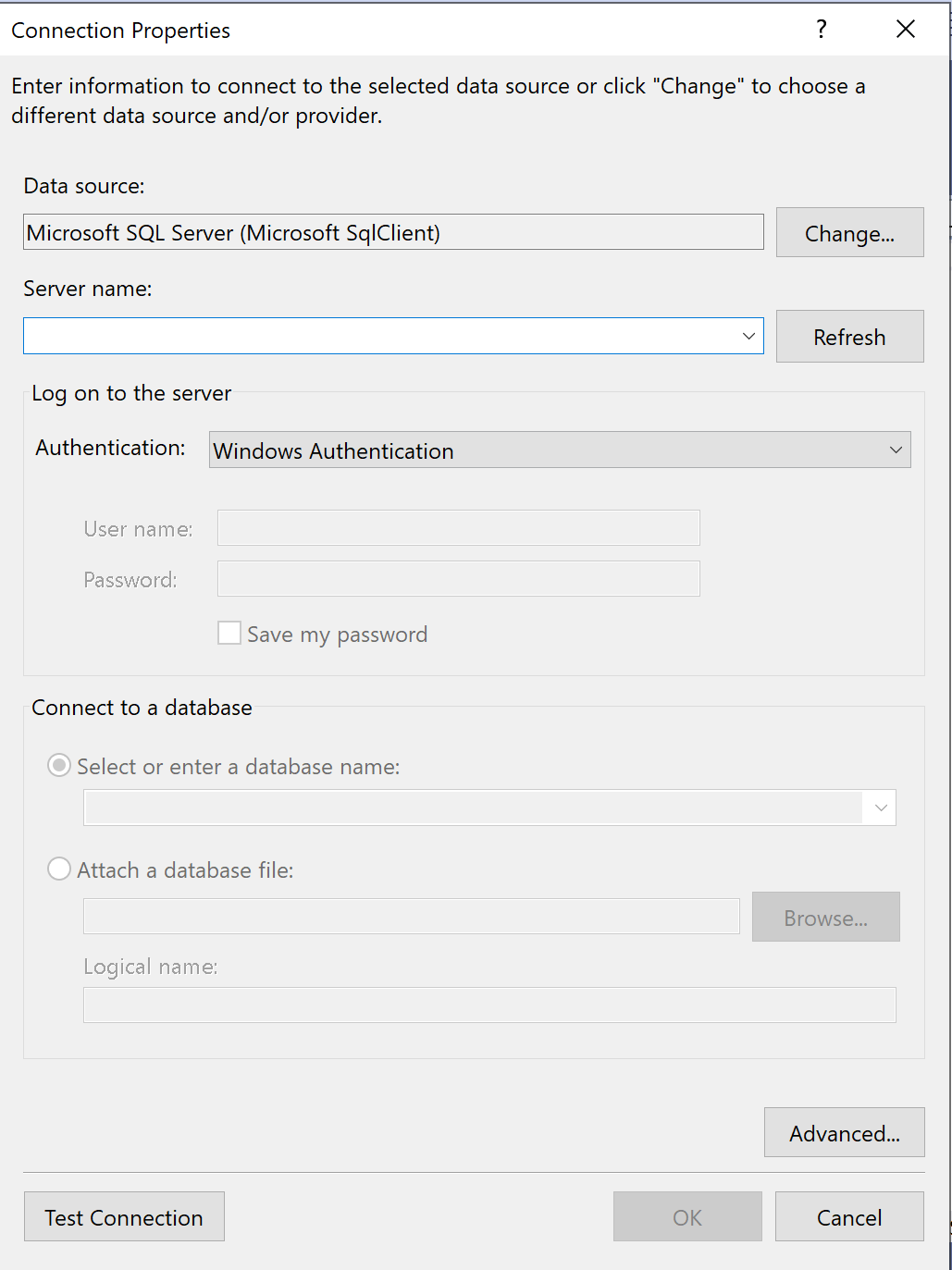
Mở dự án của bạn trong Visual Studio.

Tạo một đối tượng DbContext hoặc mở một DbContext đã có.

**Bước 3. Sử Dụng EF Core Power Tools:**

Sau khi cài đặt xong, bạn sẽ thấy một menu mới trong Visual Studio: “Entity Framework Core” -> “EF Core Power Tools.”

**Bước 4. Reverse Engineer:**



Hình 5.3 Giao diện EF Core Power Tools

Chọn “EF Core Power Tools” 🠒 “Reverse Engineer.”

Nhập chuỗi kết nối của cơ sở dữ liệu.

Chọn các bảng, chế độ xem, hoặc thậm chí stored procedures mà bạn muốn thêm vào DbContext.

1. Những kinh nghiệm thực tiễn đã đạt được

Qua hai tháng thực tập, tôi hiểu biết rõ hơn về mục đích quan trọng của việc thực tập đối với sinh viên. Không chỉ là cơ hội để học hỏi và áp dụng kiến thức chuyên môn, mà còn là bước ngoặt quan trọng trong quá trình hình thành và phát triển kỹ năng nghiệp vụ. Trải qua những ngày làm việc tại công ty, tôi đã gặp phải và vượt qua nhiều thách thức, từ đó rút ra được nhiều kinh nghiệm quý báu.

Trong dự án “Quản lý kho hàng”, tôi không chỉ áp dụng những kiến thức trên sách giáo trình, mà còn học được cách tương tác và làm việc hiệu quả với đồng nghiệp. Quá trình làm việc đòi hỏi sự chủ động, linh hoạt và khả năng giải quyết vấn đề nhanh chóng. Tôi đã hiểu rõ hơn về quy trình quản lý kho hàng, từ việc nhập xuất hàng hóa đến xử lý các vấn đề liên quan đến logistics.

Một trong những điều quan trọng nhất tôi học được là kỹ năng làm việc nhóm. Trong môi trường doanh nghiệp, không có công việc nào có thể đạt được mục tiêu một mình. Sự hợp tác và giao tiếp chính là chìa khóa để giải quyết công việc một cách hiệu quả. Tôi đã có cơ hội làm việc với những người có kinh nghiệm, học hỏi từ họ không chỉ về kiến thức chuyên sâu mà còn về tư duy và cách tiếp cận vấn đề.

Ngoài ra, tôi còn rèn luyện khả năng quản lý thời gian và ưu tiên công việc một cách hợp lý. Đôi khi, áp lực từ công việc đòi hỏi sự tự quản lý tốt để có thể đảm bảo tiến độ và chất lượng công việc. Điều này giúp tôi phát triển kỹ năng tự lập và trách nhiệm cá nhân.

Tóm lại, qua đợt thực tập, tôi không chỉ tích lũy thêm kiến thức nghề nghiệp mà còn phát triển những kỹ năng quan trọng, từ kỹ năng mềm đến kỹ năng chuyên sâu, góp phần làm cho hành trang của mình trở nên phong phú và đầy đủ hơn cho những thử thách sắp tới trong sự nghiệp của mình.

TÀI LIỆU THAM KHẢO

[1] Công ty Cổ phần GONSA, “Lịch sử hình thành và phát triển”. [Trực tuyến]. Địa chỉ <https://gonsa.com.vn/chung-toi-la-gonsa-2#:~:text=GONSA%20%C4%91%C6%B0%E1%BB%A3c%20th%C3%A0nh%20l%E1%BA%ADp%20ng%C3%A0y,Chuy%C3%AAn%20d%E1%BB%8Bch%20v%E1%BB%A5%20d%C6%B0%E1%BB%A3c%20ph%E1%BA%A9m%E2%80%9D.> [Truy cập ngày 01/12/2023]

[2] Hưng Nguyễn, “ASP.NET là gì? So sánh giữa lập trình ASP.NET và PHP” 24/11/2023. [Trực tuyến]. Địa chỉ <https://vietnix.vn/asp-net-la-gi/> . [Truy cập ngày 11/12/2023].