****

**TRƯỜNG ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC AN GIANG**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ QUÁN CÀ PHÊ**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC KINH BẮC**

**An Giang, 5/2023**

**SINH VIÊN THỰC HIỆN VÀ MÃ SINH VIÊN:**

**NGUYỄN CHÍ CƯỜNG – DTH205723**

**LỚP: DH21TH3**

**GIẢNG VIÊN HƯỚNG DẪN:**

**TH.S NGUYỄN NGỌC MINH**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH QUẢN LÝ**

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển không ngừng của xã hội, tin học đã trở nên vô cùng phổ biến và đã có những bước phát triển vượt bậc được thể hiện qua nhiều lĩnh vực của đời sống xã hội. Tin học đã dần dần trở thành nhu cầu rất cần thiết và không thể thiếu trong cuộc sống của con người. Ở các quốc gia phát triển hiệu quả do tin học hóa đem lại đã góp phần thiết yếu và chiếm giữ một vị trí quan trọng trong các lĩnh vực kinh tế, chính trị, văn hóa, giáo dục…

Chương trình **phần mềm quản lý quán Cà Phê** của em được lập trình bằng ngôn ngữ lập trình C#, chạy trên nền NetFramwork và kết nối bằng cơ sở SQL, Phần mềm này giúp các quán Cà phê vừa và nhỏ chủ yếu ở các địa phương đang phát triển chưa có nhiều cơ hội tiếp cận các công nghệ, các phần mềm quản lý nhầm bớt việc tìm kiếm mất thời gian mà hiệu quả và tính chính xác vẫn cao. Nhờ tính hiệu quả và nhanh chóng do hệ thống mới đem lại làm cho hiệu suất công việc tăng cao, trao đổi thông tin được diễn ra an toàn, đồng bộ, chính xác do tính năng ưu việt của hệ thống máy tính đem lại. Vì thế mà việc tra cứu không còn phải khó khăn, phức tạp nữa vì hệ thống sẽ thống kê, quản lý và chúng ta có thể tra cứu, tham khảo bất cứ lúc nào khi cần. Do vậy chi phí sẽ giảm thiểu, lợi nhuận tăng cao.

Chương trình **phần mềm quản lý nhân sự** của em được lập trình bằng ngôn ngữ lập trình C#, chạy trên nền NetFramwork và kết nối bằng cơ sở SQL, Phần mềm này giúp công ty quản lý hồ sơ, lương, chế độ của nhân viên giúp giảm tải bớt việc tìm kiếm mất thời gian mà hiệu quả và tính chính xác vẫn cao.

Là sinh viên của trường chúng em luôn luôn mong muốn được học hỏi và áp dụng những kiến thức mà chúng em đã được các thầy cô dạy trong thời gian qua. Đồ án này giúp chúng em tổng kết lại những điều đã học được và phát triển khả năng tư duy, tính sang tạo trong công việc để sau này chúng em có thể vững bước trên con đường mà chúng em đã chọn.

Xin chân thành cảm ơn các thầy cô đã nhiệt tình dạy chúng em trong suốt những năm học qua và để chúng em có đủ các kiến thức làm được một đồ án hoàn chỉnh và là bước khởi đầu cho những dự án trong tương lai.

CHƯƠNG 1: GIẢI QUYẾT BÀI TOÁN QUẢN LÝ NHÂN SỰ

Quy trình bài toán quản lý nhân sự:

1. 1 Khảo sát hệ thống

Các quán Cà Phê với quy mô vừa và nhỏ thường bố trí các bộ phân như sau:

|  |  |
| --- | --- |
| + Bô phận order  + Bộ phận phục vụ | + Bộ phận quản lý |

Công việc quản lý quán Cà Phê với quy mô vừa và nhỏ hiện nay vẫn chưa có nhiều phần mềm quản lý hỗ trợ mà vẫn thực hiện bằng tay với các sổ sách giấy tờ kèm theo. Công việc quản lý phải quản lý những vấn đề sau:

* Quản lý hóa đơn
* Quản lý các bàn trong phạm vi quán
* Quản lý về sản phẩm
* Quản lý loại sản phẩm
* Quản lý về tài khoản người dùng
* Quản lý chi tiết hóa đơn

**Bài toán mô tả nghiệp vụ quản lý quán Cà Phê với quy mô vừa và nhỏ**

Quản lý đặt hàng:

Nhận đơn đặt hàng từ khách hàng và ghi nhận thông tin liên quan như món ăn, đồ uống, số lượng.

Đảm bảo rằng đơn hàng được chuẩn bị và giao đúng thời gian và đáp ứng đúng yêu cầu của khách hàng.

Quản lý thanh toán:

Ghi nhận các giao dịch bán hàng và tính toán tổng giá trị đơn hàng.

 In hóa đơn hoặc cung cấp biên lai cho khách hàng sau khi thanh toán.

Quản lý thực đơn và giá cả:

Xây dựng và duy trì thực đơn với danh sách các món ăn và đồ uống, cùng với thông tin về giá cả và mô tả chi tiết.

Đảm bảo tính nhất quán về giá cả và chính sách giảm giá hoặc khuyến mãi.

Cập nhật thực đơn theo sự thay đổi của loáianr phẩm và sự lựa chọn của khách hàng.

Quản lý tài khoản người dùng: Thêm, xoá sửa, tài khoản người dùng, cấp quyền cho tài khoản người dùng.

**1.2. Thực trạng quản lý.**

**1.2.1. Bộ phận order có nhiệm vụ:**

* Thêm hoá đơn dựa trên yêu cầu của khách hàng,
* Quản lý, lưu trữ thông tin các hoá đơn.
* Thống kê, báo cáo về doanh thu.

**1.2.2 Yêu cầu của hệ thống thực**

Nghiên cứu bài toán quản lý quán Cà Phê để thiết kế và lập trình phần mềm nhằm hỗ trợ công tác quản lý bán sản phẩm. Dự kiến lập trình phần mềm quản lý quán Cà Phê đạt được các mục tiêu chính sau*:*

***Tin học hoá những khâu:***

***+ Cập nhật*** thông tin bàn ăn.

***+ Cập nhật*** thông tin món ăn, bao gồm loại món ăn và danh sách các món ăn theo từng loại.

***+ Cập nhật*** thông tin người dùng, hỗ trợ đổi và đặt lại mật khẩu.

***+ Quản lý*** tài khoản người dùng, bao gồm người dùng quản trị và người dùng thông thường.

***+ Quản lý*** các hoá đơn trong bàn, hỗ trợ chuyển và gộp cá hoá đơn trong bàn.

***+ Quản lý*** thanh toán, giá khuyến mãi theo phần trăm trên tổng hoá đơn.

***+ Tìm kiếm, tra cứu*** thông tin sản phẩm.

***+ Kết xuất*** bảng thống kê doanh thu.

***+ Kết xuất ...***

***Quá trình xử lý dữ liệu***:

* Các thông tin về tổng số bàn, số loại sản phẩm, số sản phẩm, thông tin người dùng được lưu vào CSDL.
* Các thông tin phát sinh trong quá trình bán sẽ liên tục được cập nhật nhanh chóng, chính xác và kịp thời.

# CHƯƠNG 2: CÁC KIẾN THỨC CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH C# VÀ NGÔN NGỮ TRUY VẤN CSDL SQL

## 2.1. Các kiến thức cơ bản về ngôn ngữ lập trình C#

### 2.1.1. Giới thiệu về ngôn ngữ lập trình C#

* C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.
* C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
* C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.
* C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.
* C# là một ngôn ngữ khá tốt cho người mới học lập trình bởi vì nó rất mạnh , không nhiều lệnh và học nhanh
* C# có hỗ trợ mạnh mẽ về .NET Framework cho nên việc tạo và lập trình ứng dựng và phần mềm dựa trên Windows Froms sẽ đơn giản hơn những ngôn ngữ khác.
* C# chính là 1 sự lựa chọn tuyệt vời cho sinh viên làm đồ án tốt nghiệp về phần mềm hay ứng dung.

### 2.1.2. Các đặc trưng của ngôn ngữ lập trình C#

* C# là ngôn ngữ đơn giản
* C# là ngôn ngữ hiện đại
* C# là ngôn ngữ hướng đối tượng
* C# là ngôn ngữ mạnh mẽ và mềm dẻo
* C# là ngôn ngữ có ít từ khóa
* C# là ngôn ngữ hướng module
* C# sẽ trở nên phổ biến

**C# là ngôn ngữ lập trình đơn giản**

* Như ta đã biết thì ngôn ngữ C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoậc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

**C# là ngôn ngữ lập trình hiện đại**

* Một vài khái niệm khá mới mẻ khá mơ hồ với người vừa mới học lập trình, như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn..v..v... Đây là những đặc tính được cho là của một ngôn ngữ hiện đại cần có. Và C# chứa tất cả các đặt tính ta vừa nêu trên. Mọi người sẽ dần tìm hiểu được các đặt tính trên qua các bài học trong series này.

**C# là một ngôn ngữ lập trình thuần hướng đối tượng**

* Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (*abstraction*), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên. Và để hiểu rõ hơn thì chúng ta sẽ có một chương trình bày về phần này.

**Ngôn ngữ lập trình C# rất từ khóa**

* C# là ngôn ngữ sử dụng giới hạn những từ khóa (gồm khoảng 80 từ khóa và mười mấy kiểu dữ liệu xây dựng sẵn). Nếu nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, chúng ta có thể lấy ví dụ ngôn ngữ C# làm điển hình. Nếu chúng ta học sâu về C# sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

**C# là ngôn ngữ hướng module**

* Mã nguồn C# có thể được viết trong những phần được gọi là những lớp, những lớp này chứa các phương thức thành viên của nó. Những lớp và những phương thức có thể được sử dụng lại trong ứng dụng hay các chương trình khác. Bằng cách truyền các mẫu thông tin đến những lớp hay phương thức chúng ta có thể tạo ra những mã nguồn dùng lại có hiệu quả.

**C# là ngôn ngữ phổ biến**

* C# đã phát triển nhanh chóng kể từ khi nó được ra mắt lần đầu tiên, với sự hỗ trợ to lớn từ Microsoft giúp nó đạt tạo được tiếng vang lớn. Và giờ đây nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

**Ngoài những đặc điểm trên thì còn một số ưu điểm nổi bật của C#:**

* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

### 2.1.3. Môi trường lập trình C#

*Để lập trình băng ngôn ngữ C# thì đầu tiên chúng ta cần phải cài đặt môi trường lập trình. Chúng ta cần các công cụ cần thiết cho việc lập trình.*

* **.NET framework là gì ?**

.NET Framework là một môi trường thực thi được quản lý cho Windows, cung cấp nhiều dịch vụ cho các ứng dụng đang chạy của nó. Nó bao gồm hai thành phần chính: bộ thực thi ngôn ngữ chung (CLR), là công cụ thực thi xử lý các ứng dụng đang chạy và thư viện lớp .NET Framework, cung cấp một thư viện mã được kiểm tra, có thể sử dụng lại mà các nhà phát triển có thể gọi từ các ứng dụng của riêng họ. Các dịch vụ mà .NET Framework cung cấp cho các ứng dụng đang chạy bao gồm:

* Quản lý bộ nhớ. Trong nhiều ngôn ngữ lập trình, lập trình viên chịu trách nhiệm phân bổ và giải phóng bộ nhớ và xử lý vòng đời của đối tượng. Trong các ứng dụng .NET Framework, CLR cung cấp các dịch vụ này thay mặt cho ứng dụng.
* Một hệ thống loại phổ biến. Trong các ngôn ngữ lập trình truyền thống, các loại cơ bản được xác định bởi trình biên dịch, làm phức tạp khả năng tương tác giữa các ngôn ngữ. Trong .NET Framework, các loại cơ bản được xác định bởi hệ thống loại .NET Framework và phổ biến cho tất cả các ngôn ngữ nhắm vào .NET Framework.
* Một thư viện lớp học rộng lớn. Thay vì phải viết một lượng lớn mã để xử lý các hoạt động lập trình cấp thấp phổ biến, các lập trình viên sử dụng một thư viện các loại có thể truy cập dễ dàng và các thành viên của họ từ Thư viện lớp .NET Framework.
* Khung phát triển và công nghệ .NET Framework bao gồm các thư viện cho các lĩnh vực phát triển ứng dụng cụ thể, chẳng hạn như ASP.NET cho các ứng dụng web, ADO.NET để truy cập dữ liệu, Windows Communication Foundation cho các ứng dụng hướng dịch vụ và Windows Presentation Foundation cho các ứng dụng máy tính để bàn Windows.
* Khả năng tương tác ngôn ngữ. Các trình biên dịch ngôn ngữ nhắm vào .NET Framework phát ra một mã trung gian có tên là Ngôn ngữ trung gian chung (CIL), đến lượt nó, được biên dịch trong thời gian chạy bởi thời gian chạy ngôn ngữ chung. Với tính năng này, các thói quen được viết bằng một ngôn ngữ có thể truy cập được bằng các ngôn ngữ khác và các lập trình viên tập trung vào việc tạo các ứng dụng bằng ngôn ngữ ưa thích của họ.
* Phiên bản tương thích. Với các trường hợp ngoại lệ hiếm hoi, các ứng dụng được phát triển bằng cách sử dụng một phiên bản .NET Framework cụ thể chạy mà không sửa đổi trên phiên bản mới hơn.
* Thi công bên lề. .NET Framework giúp giải quyết xung đột phiên bản bằng cách cho phép nhiều phiên bản thời gian chạy ngôn ngữ chung tồn tại trên cùng một máy tính. Điều này có nghĩa là nhiều phiên bản ứng dụng có thể cùng tồn tại và một ứng dụng có thể chạy trên phiên bản .NET Framework được xây dựng. Việc thực hiện song song áp dụng cho các nhóm phiên bản .NET Framework 1.0 / 1.1, 2.0 / 3.0 / 3.5 và 4 / 4.5.x / 4.6.x / 4.7.x / 4.8.
* Đa mục tiêu. Bằng cách nhắm mục tiêu [.NET Standard](https://docs.microsoft.com/en-us/dotnet/standard/net-standard), các nhà phát triển tạo các thư viện lớp hoạt động trên nhiều nền tảng .NET Framework được hỗ trợ bởi phiên bản tiêu chuẩn đó. Ví dụ: các thư viện nhắm mục tiêu .NET Standard 2.0 có thể được sử dụng bởi các ứng dụng nhắm mục tiêu .NET Framework 4.6.1, .NET Core 2.0 và UWP 10.0.16299.
* **Integrated Development Environment(IDE) cho C#**

IDE viết tắt của cụm từ Integrated Development Environment là phần mềm cung cấp cho các lập trình viên một môi trường tích hợp bao gồm nhiều công cụ khác nhau như chương trình viết mã lệnh hay code editor, chương trình sửa lỗi hay debugger, chương trình mô phỏng ứng dụng khi chạy thực tế hay simulator.... Nói cách khác thì IDE là một phần mềm bao gồm những gói phần mềm khác giúp phát triển ứng dụng phần mềm.

* Microsoft cung cấp các công cụ phát triển sau cho lập trình C#:
* Visual Studio (VS)
* Visual C# Express (VCE)
* Visual Web Developer
* Một số thông tin về Microsoft Visual Studio:
* **Visual studio** là một trong những công cụ hỗ trợ lập trình rất nổi tiếng nhất hiện nay của Mcrosoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một các dễ dàng và nhanh chóng nhất thông qua Visual Studio.
* Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ [Microsoft](https://www.microsoft.com/vi-vn/). Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.
* Bên cạnh đó, Visual Studio còn cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.
* Tính năng Microsof Visual Studio
* Biên tập mã : Giống như bất kỳ [IDE](https://vi.wikipedia.org/wiki/IDE) khác, nó bao gồm một [trình soạn thảo mã](https://vi.wikipedia.org/w/index.php?title=Tr%C3%ACnh_so%E1%BA%A1n_th%E1%BA%A3o_m%C3%A3&action=edit&redlink=1" \o "Trình soạn thảo mã (trang chưa được viết)) hỗ trợ [tô sáng cú pháp](https://vi.wikipedia.org/w/index.php?title=T%C3%B4_s%C3%A1ng_c%C3%BA_ph%C3%A1p&action=edit&redlink=1" \o "Tô sáng cú pháp (trang chưa được viết)) và [hoàn thiện mã](https://vi.wikipedia.org/w/index.php?title=Ho%C3%A0n_thi%E1%BB%87n_m%C3%A3&action=edit&redlink=1" \o "Hoàn thiện mã (trang chưa được viết)) bằng cách sử dụng [IntelliSense](https://vi.wikipedia.org/w/index.php?title=IntelliSense&action=edit&redlink=1) không chỉ cho các [biến](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_(l%E1%BA%ADp_tr%C3%ACnh)" \o "Biến (lập trình)), [hàm](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_con" \o "Chương trình con) và các[phương pháp](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_th%E1%BB%A9c_l%E1%BA%ADp_tr%C3%ACnh) mà còn các cấu trúc ngôn ngữ như [vòng điều khiển](https://vi.wikipedia.org/w/index.php?title=Ki%E1%BB%83m_so%C3%A1t_l%C6%B0u_l%C6%B0%E1%BB%A3ng&action=edit&redlink=1" \o "Kiểm soát lưu lượng (trang chưa được viết)) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1" \o "Language Integrated Query (trang chưa được viết)).
* Trình gỡ lỗi: Visual Studio có một trình gỡ lỗi hoạt động vừa là một trình gỡ lỗi cấp mã nguồn và là một trình gỡ lỗi cấp máy. Nó hoạt động với cả hai mã quản lý cũng như [ngôn ngữ máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y" \o "Ngôn ngữ máy) và có thể được sử dụng để gỡ lỗi các ứng dụng được viết bằng các ngôn ngữ được hỗ trợ bởi Visual Studio. Ngoài ra, nó cũng có thể đính kèm theo quy trình hoạt động và theo dõi và gỡ lỗi những quy trình.
* Thiết kế :
* Windows Forms Designer: được sử dụng để xây dựng [GUI](https://vi.wikipedia.org/wiki/GUI) sử dụng [Windows Forms](https://vi.wikipedia.org/w/index.php?title=Windows_Forms&action=edit&redlink=1); bố trí có thể được xây dựng bằng các nút điều khiển bên trong hoặc khóa chúng vào bên cạnh mẫu. Điều khiển trình bày dữ liệu (như hộp văn bản, hộp danh sách, ... ) có thể được liên kết với các nguồn dữ liệu như [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Cơ sở dữ liệu) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1" \o "Language Integrated Query (trang chưa được viết)). Các điều khiển dữ liệu ràng buộc có thể được tạo ra bằng cách rê các mục từ cửa sổ nguồn dữ liệu lên bề mặt thiết kế. Các giao diện người dùng được liên kết với mã sử dụng một mô hình [lập trình hướng sự kiện](https://vi.wikipedia.org/w/index.php?title=L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_s%E1%BB%B1_ki%E1%BB%87n&action=edit&redlink=1" \o "Lập trình hướng sự kiện (trang chưa được viết)). Nhà thiết kế tạo ra bằng C# hay [VB.NET](https://vi.wikipedia.org/wiki/VB.NET) cho ứng dụng.
* WPF Designer: có tên mã là Cider, được giới thiệu trong Visual Studio 2008. Giống như Windows Forms Designer, hỗ trợ kéo và thả ẩn dụ. Sử dụng [tương tác người-máy](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%C3%A1c_ng%C6%B0%E1%BB%9Di-m%C3%A1y" \o "Tương tác người-máy) nhắm mục tiêu theo [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation). Nó hỗ trợ các chức năng WPF bao gồm [kết nối dữ liệu](https://vi.wikipedia.org/w/index.php?title=K%E1%BA%BFt_n%E1%BB%91i_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1" \o "Kết nối dữ liệu (trang chưa được viết)) và tự động hóa bố trí quản lý. Nó tạo ra mã [XAML](https://vi.wikipedia.org/w/index.php?title=XAML&action=edit&redlink=1) cho giao diện người dùng. Các tập tin XAML được tạo ra là tương thích với [Microsoft Expression Design](https://vi.wikipedia.org/w/index.php?title=Microsoft_Expression_Design&action=edit&redlink=1), sản phẩm thiết kế theo định hướng. Các mã XAML được liên kết với mã đang sử dụng một mô hình [code-behind](https://vi.wikipedia.org/wiki/Code-behind).
* Web Designer/Development: Visual Studio cũng bao gồm một trình soạn thảo và thiết kế trang web cho phép các trang web được thiết kế bằng cách kéo và thả các đối tượng. Nó được sử dụng để phát triển các ứng dụng [ASP.NET](https://vi.wikipedia.org/wiki/ASP.NET) và hỗ trợ [HTML](https://vi.wikipedia.org/wiki/HTML), [CSS](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Nó sử dụng mô hình [code-behind](https://vi.wikipedia.org/wiki/Code-behind) để liên kết với mã ASP.NET. Từ Visual Studio 2008 trở đi, công cụ bố trí được sử dụng bởi các nhà thiết kế web được chia sẻ với [Microsoft Expression Web](https://vi.wikipedia.org/w/index.php?title=Microsoft_Expression_Web&action=edit&redlink=1). Ngoài ra [ASP.NET MVC Framework](https://vi.wikipedia.org/w/index.php?title=ASP.NET_MVC_Framework&action=edit&redlink=1) hỗ trợ cho công nghệ [MVC](https://vi.wikipedia.org/wiki/MVC) là tải xuống riêng biệt và dự án [ASP.NET Dynamic Data](https://vi.wikipedia.org/w/index.php?title=ASP.NET_Dynamic_Data&action=edit&redlink=1) có sẵn từ Microsoft.
* Class Designer: Các lớp thiết kế được dùng để biên soạn và chỉnh sửa các lớp (bao gồm cả các thành viên và truy cập của chúng) sử dụng mô hình [UML](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%B4_h%C3%ACnh_h%C3%B3a_th%E1%BB%91ng_nh%E1%BA%A5t). Các lớp thiết kế có thể tạo ra mã phác thảo [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng) và [VB.NET](https://vi.wikipedia.org/wiki/VB.NET) cho các lớp và cá phương thức. Nó cũng có thể tạo ra sơ đồ lớp từ các lớp viết tay.
* Data Designer: Thiết kế dữ liệu có thể được sử dụng để chỉnh sửa đồ họa [giản đồ cơ sở dữ liệu](https://vi.wikipedia.org/w/index.php?title=Gi%E1%BA%A3n_%C4%91%E1%BB%93_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1" \o "Giản đồ cơ sở dữ liệu (trang chưa được viết)) bao gồm các bảng, khóa chính, khóa ngoại và các rằng buộc. Nó cũng có thể được sử dụng để thiết kế các truy vấn từ các giao diện đồ họa.
* Mapping Designer: Từ Visual Studio 2008 trở đi, thiết kế ánh xạ được dùng bởi [Language Integrated Query](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1) để thiết kế các ánh xạ giữa các giản đồ cơ sở dữ liệu và các lớp để đóng gói dữ liệu. Các giải pháp mới từ cách tiếp cận [ORM](https://vi.wikipedia.org/w/index.php?title=ORM&action=edit&redlink=1), [ADO.NET Entity Framework](https://vi.wikipedia.org/w/index.php?title=ADO.NET_Entity_Framework&action=edit&redlink=1) sẽ thay thế và cải thiện các công nghệ cũ.

## 2.2. Ngôn ngữ truy vấn CSDL SQL

### 2.2.1. SQL là gì?

SQL là viết tắt của Structured Query Language có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ.

SQL là ngôn ngữ chuẩn cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

### 2.2.2. Lịch sử phát triển của SQL

* **1970** – Tiến sĩ Edgar F. “Ted” Codd của IBM được biết đến như là cha đẻ của các cơ sở dữ liệu quan hệ. Ông mô tả một mô hình quan hệ cho cơ sở dữ liệu.
* **1974** – Đã xuất hiện Structured Query Language.
* **1978** – IBM đã làm việc để phát triển các ý tưởng của Codd và đưa ra một sản phẩm có tên System / R.
* **1986** – IBM đã phát triển nguyên mẫu đầu tiên của cơ sở dữ liệu quan hệ và được tiêu chuẩn hóa bởi ANSI. Cơ sở dữ liệu quan hệ đầu tiên được phát hành bởi Relational Software sau này được gọi là Oracle.

### **2.2.3. Ưu điểm của ngôn ngữ SQL**

Gần gũi với ngôn ngữ tự nhiên (tiếng Anh), do vậy dễ tiếp cận, dễ hiểu. Nhờ SQL chương trình trở nên ngắn gọn. Do ưu điểm của SQL, ngày nay phần lớn các ngôn ngữ lập trình đều có nhúng ngôn ngữ SQL. Có thể dùng các lệnh của SQL thực hiện các công việc sau:

* Cho phép người dùng truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.
* Cho phép người dùng mô tả dữ liệu.
* Cho phép người dùng xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.
* Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện và trình biên dịch trước.
* Cho phép người dùng tạo và thả các cơ sở dữ liệu và bảng.
* Cho phép người dùng tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.
* Cho phép người dùng thiết lập quyền trên các bảng, thủ tục và view.

### **2.2.3. Các bước kết nối C# với SQL**

 Để mở kết nối cần các thứ và bước sau:

1. B1. Chuẩn bị một chuỗi kết nối connection string (là đường dẫn đến database, thông tin username password để vào database)
2. B2. Đối tượng hỗ trợ kết nối đến database (SqlConnection)
3. B3. Sử dụng SqlConnection để mở kết nối bằng SqlConnection.Open
4. B4. Xử lý các ngoại lệ nếu có lỗi trong quá trình mở kết nối
5. B5. Đóng kết nối sau khi sử dụng

* Code minh họa

 //xac dinh duong dan den database

            String connString = @"Data Source=localhost\SQLEXPRESS;Initial Catalog=master;Integrated Security=True";

            //ket noi csdl bang Sqlconnection

            SqlConnection connection = new SqlConnection(connString);

            try

            {

                //Mo ket noi

                connection.Open();

            }

            catch (InvalidOperationException ex)

            {

                //xu ly khi ket noi co van de

                MessageBox.Show("Khong the mo ket noi hoac ket noi da mo truoc do");

            }

            catch (Exception ex)

            {

                //xu ly khi ket noi co van de

                MessageBox.Show("Ket noi xay ra loi hoac doc du lieu bi loi");

            }

            finally

            {

                //Dong ket noi sau khi thao tac ket thuc

                connection.Close();

            }

 Để thực hiện các thao tác truy vấn đến dữ liệu thì cần thực hiện các bước sau:

B1. Chuẩn bị một chuỗi query viết bằng ngôn ngữ SQL

B2. Chuẩn bị một đối tượng SqlCommand, đối tượng này sẽ chứa câu query chuẩn bị ở bước 1, và kết nối đến SqlConnect đã tạo kết nối đến database.

B3. Sử dụng SqlCommand để thực hiện câu truy vấn (sử dụng SqlCommand.ExcuteReader nếu là thao tác select, SqlCommand.ExcuteNonQuery nếu là các thao tác thêm, sửa, xóa).

B4. Xử lý kết quả( Đối với thêm, sửa, xóa kết quả nhận về là số dòng bị ảnh hưởng bởi câu query (khi thêm sẽ trả về số dòng được thành công, khi sửa sẽ trả về số dòng bị sửa, xóa sẽ trả về số dòng bị xóa, còn đối với select thì kết quả sẽ là một đối tượng SqlDataReader sẽ hỗ trợ ta đọc các dòng dữ liệu hợp lệ trong câu select mà ta chuẩn bị bước 1 – và tiếp tục làm bước 5,6,7)

B5. Chỉ áp dụng cho trường hợp Select: Kiểm tra có còn dòng nào chưa đọc từ kết quả khi thực hiện truy vấn hay không bằng SqlDataReader.HasRows (true là còn dòng chưa đọc, false là đã hết dữ liệu)

B6. Đọc một dòng dữ liệu bằng SqlDataReader.Read() (trả về true nếu đọc ok, false nếu đọc có vấn đề hoặc hết dòng để đọc).

B6. Lấy các cột dữ liệu mong muốn từ SqlDataReader nếu bước Read thành công. Muốn lấy dữ liệu cột i (chạy từ 0) của  dòng hiện tại đang đọc thì sử dụng các lệnh: SqlDataReader.GetString(i), SqlDataReader.GetInt32(i), SqlDataReader.GetBoolean(i),….

B7. Chuyển qua bước đóng kết nối.

# CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1. Sơ đồ phân rã chức năng

**A chart with blue rectangles and white text

Description automatically generated with low confidence**

*Hình 3.1. Mô hình phân rã chức năng*

### Mô tả hệ thống

### Chức năng quản lý hệ thống:

- Đăng nhập, đăng xuất hệ thống

- Đổi mật khẩu người dùng

- Phân quyền cho các nhóm người dung: các nhóm người dùng được quy định quyền hạn sử dụng các chức năng của phần mềm khác nhau.

### Chức năng quản lý danh mục:

- Cập nhật danh sách danh mục (thêm, xoá, sửa) bao gồm ID danh mục và tên danh mục.

- Hiển thị danh sách danh mục món ăn, thức uống, …

* **Chức năng quản lý sản phẩm**

**-** Thực hiện thêm sửa xóa, với tên sản phẩm, loại sản phẩm, giá bán.

- Tìm kiếm theo tên gần đúng của sản phẩm.

* **Chức năng quản lý bán hàng:**

- Giải quyết các vấn đề liên quan đến bán hàng như: lập danh sách hoá đơn, thêm, xoá sửa món trên hoá đơn, thanh toán, giảm giá.

- Câp nhật hoá đơn: chuyển hoá đơn, gộp hoá đơn

- Cập nhật bàn ăn: thêm xoá sửa bàn ăn.

* **Chức năng báo cáo:**

- Báo cáo theo yêu cầu của người kinh doanh: báo cáo danh thu theo mốc thời gian.

## 3.2. Sơ đồ ngữ cảnh

Sơ đồ DFD mức 0.

A picture containing text, receipt, line, diagram

Description automatically generated

Hình 3.1. Sơ đồ ngữ cảnh

**3.3. Sơ đồ luồng dữ liệu mức đỉnh**

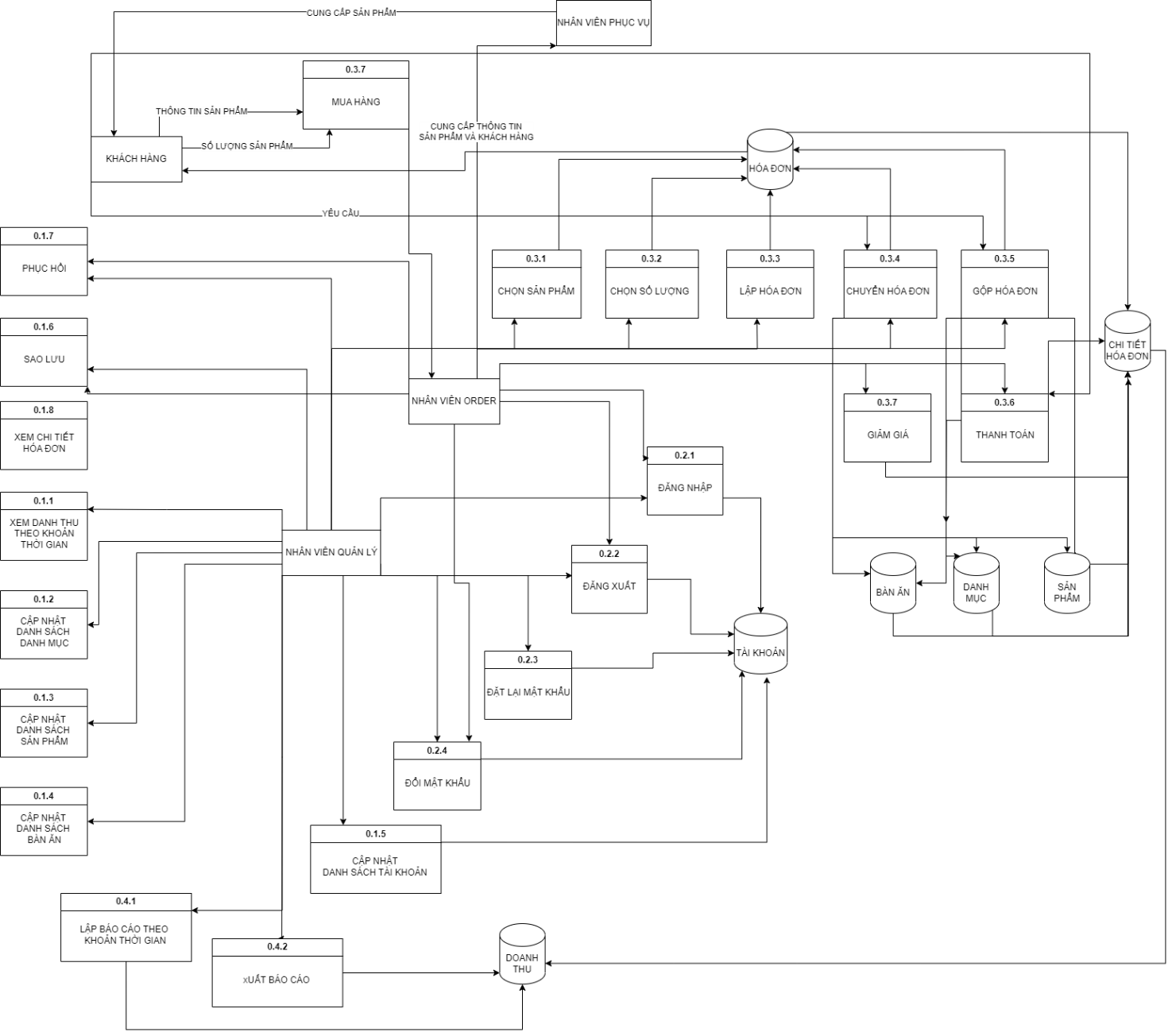
**3.3.1 Mức 1**

A picture containing diagram, text, plan, technical drawing

Description automatically generated

Hình 3.2. Sơ đồ luồng dữ liệu mức đỉnh

3.4. Các sơ đồ luồng dữ liệu mức dưới đỉnh



## 3.5. Xây dựng các thực thể:

***TABLEFOOD*** (**id**, name, status)

***ACCOUNT*** (**username**, displayname, password, type)

***FOODCATEGORY*** (**id**, name)

***FOOD*** (**id**, name, price, *idcategory*)

***BILL*** (**id**, datecheckin, datecheckout, discount, status, totalprice, *idtable*)

***BILLINFO*** (**id**, count, *idbill*, *idfood*)

## 3.6. Thiết kế vật lý

Bảng **Account** (người dùng)

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| Username | Nvarchar (100) | Tài khoản |
| PassWord | Nvarchar (1000) | Mật khẩu |
| DisplayName | Nvarchar (100) | Tên hiển thị |
| Type | Int | Loại tài khoản |

Bảng **TableFood** (thông tin bàn ăn cơ bản)

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Int | Mã bàn |
| Name | Nvarchar (100) | Tên bàn |
| Status | Nvarchar (100) | Trạng thái |

Bảng **Food** (thông tin sản phẩm)

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Int | Mã sản phẩm |
| Name | Nvarchar (100) | Tên sản phẩm |
| IdCategory | Int | Mã danh mục |
| Price | Float | Giá bán |

Bảng **FoodCategory** (bảng loại thức ăn, đồ uống (loại sản phẩm))

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Int | Mã danh mục |
| Name | Nvarchar (100) | Tên danh mục |

Bảng **Bill** (Hoá đơn)

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Int | Mã hoá đơn |
| DateCheckIn | Date | Thời gian vào quán |
| DateCheckOut | Date | Thời gian rời khỏi quán |
| idTable | Int | Mã bàn |
| Discount | Int | Phần trăm giảm giá |
| Status | Int | Trạng thái thanh toán hoặc chưa thanh toán |
| TotalPrice | Float | Thành tiền |

Bảng **BillInfo** (Bảng chi tiết hoá đơn)

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Int | Mã chi tiết hoá đơn |
| IdBill | Int | Mã hoá đơn |
| IdFood | Int | Mã món ăn (sản phẩm) |
| Count | Int | Số lượng |

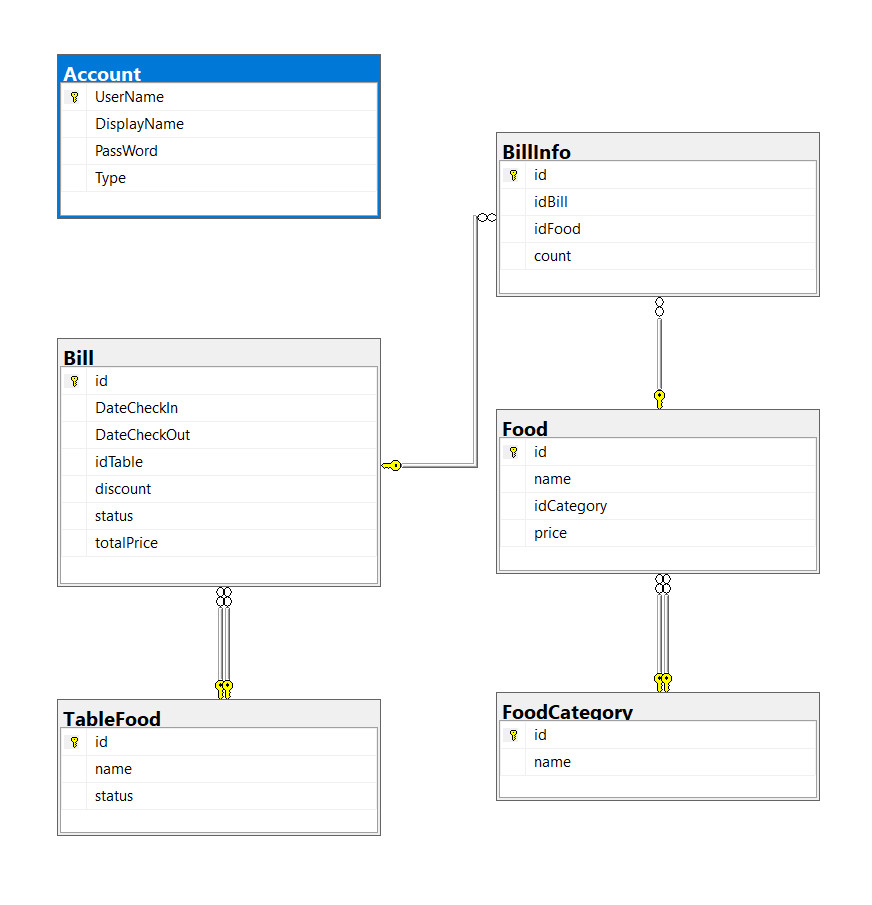
## 3.6. Mô hình thực thể E-R (Entity Relationship):

**A picture containing diagram, line, plan, technical drawing

Description automatically generated**

*Hình 3.3. Mô hình thực thể E-R*

## 3.7. Mối quan kết giữa các bảng dữ liệu



*Hình 3.4: Mối quan kết giữa các bảng dữ liệu*

# CHƯƠNG 4: GIAO DIỆN TRONG PHẦN MỀM

## 4.1. Giao diện đăng nhập hệ thống

## A screenshot of a computer Description automatically generated with medium confidence

*Hình 4.1. Giao diện đăng nhập hệ thống*

- Đây là giao diện khởi động của hệ thống. Muốn vào trong hệ thống chúng ta cần thông qua chức năng đăng nhập của giao diện này.

- Cần sử dụng tài khoản và mật khẩu đã tạo trước để đăng nhập vào hệ thống.

## 4.2. Giao diện Quản lý tài khoản

*A screenshot of a computer

Description automatically generated*

*Hình 4.2. Giao diện quản lý tài khoản*

- Chỉ có tài khoản admin được sử dụng chức năng này.

- Tại đây admin có thể cập nhật thông tin, phân quyền cho các tài khoản, thêm và xóa các tài khoản đăng nhập.

## 4.3. Giao diện Đổi mật khẩu

**A screenshot of a computer

Description automatically generated with medium confidence**

*Hình 4.3. Giao diện đổi mật khẩu*

- Tại đây người dùng có thể đổi lại mật khẩu tài khoản của mình.

## 4.4. Giao diện Người dùng quản lý

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

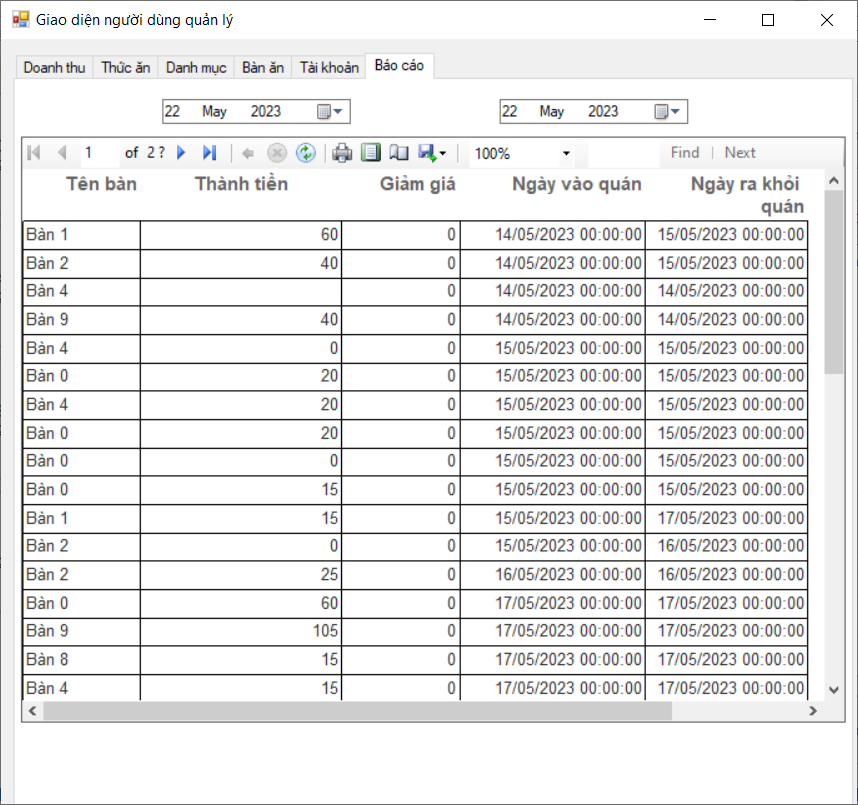
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Đây là giao diện có thể thêm, sửa, xóa các sản phẩm, danh mục, bàn ăn, tài khoản, báo cáo.



## 4.5. Giao diện Thanh toán

A screenshot of a computer

Description automatically generated

*Hình 4.7. Giao diện thanh toán*

TỔNG KẾT VÀ ĐÁNH GIÁ

* Ưu điểm:
* Chương trình có giao diện thân thiện, thuận tiện cho người sử dụng.
* Hỗ trợ nhiều chức năng phù hợp với thực tế của việc tìm kiếm thông tin, trao đổi và quản lý nhân sự.
* Có nhiều chức năng xử lý linh hoạt các yêu cầu thực tế. Ví dụ như:

+ Cung cấp khả năng thống kê, báo cáo linh hoạt có tính chất tuỳ chọn theo những tiêu chí cụ thể.

+ Chức năng quản lý đăng nhập theo nhóm người dùng hỗ trợ tương đối tốt cho việc phân phối công việc theo chức năng và nhiệm vụ của các bộ phận chuyên trách trong hệ thống quản lý nhân sự.

+ Chức năng nhập và quản lý nhân viên được thực hiện logic đảm bảo cho dữ liệu được hợp lệ hoá và cập nhật chính xác.

* Dễ dàng sử dụng và quản lý.
* Cơ sở dữ liệu tương đối lớn và tính bảo mật cao.
* Khuyết điểm:
* Do thời gian nghiên cứu có hạn và khả năng còn hạn chế nên một số vẫn chỉ còn trên ý tưởng, chưa được thực thi.
* Một số chức năng còn chưa hoạt động ổn định.
* Chưa giải quyết chọn vẹn các vấn đề nảy sinh trong quá trình quản lý.
* Chương trình chưa có tính thẩm mỹ chưa cao
* **Hướng phát triển của phần mềm :**
* Tìm hiểu và hoàn thiện một số chức năng chưa thực hiện trong ứng dụng.
* Hỗ trợ cơ sở dữ liệu Oracle để khả năng bảo mật, tốc độ truy vấn, khả năng quản lý dữ liệu tốt hơn.

# Lời cảm ơn

Đề tài “Quản lý nhân sự bằng ngôn nhữ C#” là nội dung tôi chọn để nghiên cứu và làm luận án tốt nghiệp sau bốn năm theo học chương trình Đại học chuyên ngành Công nghệ thông tin tại trường Đại học Kinh Bắc.

Để hoàn thành quá trình nghiên cứu và hoàn thiện luận văn này, lời đầu tiên tôi xin chân thành cảm ơn sâu sắc đến Th.s Nguyễn Thị Mười Phương. Cô đã trực tiếp chỉ bảo  và hướng dẫn tôi trong suốt quá trình nghiên cứu để tôi hoàn thiện luận văn này.

Nhân dịp này, tôi cũng xin cảm ơn  Khoa Công nghệ thông tin Trường Đại học Kinh Bắc, lãnh đạo và các anh chị đang công tác tại TNHH Công Nghệ Hoàng Trường đã tạo điều kiện và thời gian cho tôi trong suốt quá trình nghiên cứu.

Cuối cùng, tôi xin cảm ơn những người thân, bạn bè đã luôn bên tôi, động viên tôi hoàn thành khóa học và bài luận văn này.

Trân trọng cảm ơn!

**Em xin chân thành cảm ơn!**

# Tài Liệu Tham Khảo

https://stackoverflow.com/

https://github.com/

https://www.slideshare.net/trongthuy3/luan-van-xay-dung-phan-mem-quan-li-quan-ca-phe-hot-9d/

How Kteam

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng nhóm chúng tôi. Những tài liệu trong đồ án tốt nghiệp là hoàn toàn trung thực. Các kết quả nghiên cứu do chính chúng tôi thực hiện dưới sự chỉ đạo của giáo viên hướng dẫn.

**Sinh Viên Thực Hiện**

Cường

Nguyễn Chí Cường