**TRƯỜNG ĐẠI HỌC AN GIANG - ĐHQG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC**

**LẬP TÌNH QUẢN LÝ**

**QUẢN LÝ QUÁN CÀ PHÊ**

**GIẢNG VIÊN HƯỚNG DẪN: Th.S NGUYỄN NGỌC MINH**

**SINH VIÊN THỰC HIỆN:**

**LƯU MINH TRỌNG – DTH206038**

**AN GIANG, THÁNG 5 NĂM 2023**

**TRƯỜNG ĐẠI HỌC AN GIANG - ĐHQG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC**

**PHÂN TÍCH THIẾT KẾ PHẦN MỀM**

**HƯỚNG DỐI TƯỢNG**

**QUẢN LÝ QUÁN CÀ PHÊ**

**GIẢNG VIÊN HƯỚNG DẪN: Th.S NGUYỄN NGỌC MINH**

**SINH VIÊN THỰC HIỆN:**

**LƯU MINH TRỌNG – DTH206038**

**AN GIANG, THÁNG 5 NĂM 2023**

MỤC LỤC

[LỜI MỞ ĐẦU 3](#_Toc136315901)

[CHƯƠNG 1: GIẢI QUYẾT BÀI TOÁN QUẢN LÝ QUÁN CAFE 4](#_Toc136315902)

[**1.1/Khảo sát hệ thống** 4](#_Toc136315903)

[**1.2/Bài toán mô tả nghiệp vụ quản lý quán cà phê của quán cà phênhư sau:** 4](#_Toc136315904)

[CHƯƠNG 2: CÁC KIẾN THỨC CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH C# VÀ NGÔN NGỮ TRUY VẤN CSDL SQL 6](#_Toc136315905)

[**2.1. Các kiến thức cơ bản về ngôn ngữ lập trình C#** 6](#_Toc136315906)

[**2.1.1. Giới thiệu về ngôn ngữ lập trình C#** 6](#_Toc136315907)

[**2.1.2. Các đặc trưng của ngôn ngữ lập trình C#** 6](#_Toc136315908)

[**2.1.3. Môi trường lập trình C#** 9](#_Toc136315909)

[**2.2. Ngôn ngữ truy vấn CSDL SQL** 14](#_Toc136315910)

[**2.2.1. SQL là gì?** 14](#_Toc136315911)

[**2.2.2. Lịch sử phát triển của SQL** 14](#_Toc136315912)

[***2.2.3. Ưu điểm của ngôn ngữ SQL*** 14](#_Toc136315913)

[***2.2.3. Các bước kết nối C# với SQL*** 15](#_Toc136315914)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](#_Toc136315915)

[**3.1. Sơ đồ phân rã chức năng** 19](#_Toc136315916)

[**3.2. Sơ đồ ngữ cảnh** 21](#_Toc136315917)

[**3.3. Sơ đồ luồng dữ liệu mức đỉnh** 22](#_Toc136315918)

[**3.4. Các sơ đồ luồng dữ liệu mức dưới đỉnh** 22](#_Toc136315919)

[**3.4.1. Sơ đồ mức dưới đỉnh quản lý hệ thống** 23](#_Toc136315920)

[**3.4.2. Sơ đồ mức dưới đỉnh quản lý bán hàng** 23](#_Toc136315921)

[**3.4.3. Sơ đồ mức dưới đỉnh quản lý bàn** 24](#_Toc136315922)

[**3.4.4. Sơ đồ mức dưới đỉnh quản lý nhân viên** 24](#_Toc136315923)

[**3.4.7. Sơ đồ mức dưới đỉnh quản lý hàng hóa** 25](#_Toc136315924)

[**3.4.8. Sơ đồ mức dưới đỉnh báo cáo thống kê** 25](#_Toc136315925)

[**3.5. Xây dựng các thực thể** 25](#_Toc136315926)

[**3.6. Thiết kế vật lý** 26](#_Toc136315927)

[**3.7. Mối quan hệ giữa các thực thể:** 28](#_Toc136315928)

[**3.8. Mô hình thực thể E-R (Entity Relationship):** 30](#_Toc136315929)

[**3.9. Mối quan kết giữa các bảng dữ liệu** 30](#_Toc136315930)

[CHƯƠNG 4: GIAO DIỆN TRONG PHẦN MỀM 31](#_Toc136315931)

[**4.1. Giao diện đăng nhập hệ thống** 31](#_Toc136315932)

[**4.7. Giao diện quản lý hàng hóa** 34](#_Toc136315933)

[**4.8. Giao diện quản lý báo cáo thống kê** 35](#_Toc136315934)

[TỔNG KẾT VÀ ĐÁNH GIÁ 36](#_Toc136315935)

[**\*Ưu điểm** 36](#_Toc136315936)

[**\*Khuyết điểm** 36](#_Toc136315937)

[**\*Hướng phát triển** 37](#_Toc136315938)

[TÀI LIỆU THAM KHẢO 38](#_Toc136315939)

# LỜI MỞ ĐẦU

Trong thời đại hiện đại, ngành công nghiệp dịch vụ đang trở thành một phần không thể thiếu của cuộc sống hàng ngày. Trong số đó, quán cà phê đã trở thành điểm đến phổ biến và yêu thích của nhiều người. Với sự gia tăng về số lượng quán cà phê, nhu cầu quản lý hiệu quả và tối ưu hóa hoạt động của quán trở nên vô cùng quan trọng.

Quán cà phê mong muốn có một phần mềm quản lý quán của họ để có thể quản lý quán một cách hiệu quả và tăng hiệu suất của quán, vì vậy em đã thực hiện một phần mềm quản lý quán cà phê.

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, áp dụng mô hình 3 lớp. Phần mềm giúp quản lý bán hàng, nhân viên, khách hàng, thức uống và báo cáo, thống kê doanh thu và lương cho nhân viên.

Bài báo cáo này giống như một bài tổng kết lại những kiến thực mà em đã học được trong 3 năm qua để giúp cho em có nhìu bài học cũng như thêm kinh nghiệp để có thể áp dụng sau này. Cảm ơn thầy đã nhiệt tình dạy và hỗ trợ em đễ có thể làm bài báp cáo này một cách tốt nhất mà em có thể làm.

# CHƯƠNG 1: GIẢI QUYẾT BÀI TOÁN QUẢN LÝ QUÁN CAFE

## **1.1/Khảo sát hệ thống**

Quản lý quán cà phê của quán vẫn chưa thiết thực vì chưa thể áp dụng khoa học công nghệ vào, vẫn xử lý bằng tay và giấy tờ kèm theo vì vậy quán mong muốn một chương trình có các chức năng sau đây:

* Quản lý bán hàng
* Quản lý Nhân viên
* Quản lý hàng hóa (các món trong quán)
* Báo cáo thống kê (Doanh thu của quán)
* Chức năng gộp bàn và chuyển bàn

## **1.2/Bài toán mô tả nghiệp vụ quản lý quán cà phê của quán cà phênhư sau:**

Khi khách hàng lựa chọn chỗ ngồi, thì nhân viên phục vụ sẽ đưa cho khách menu của quán.Trong thời gian khách hàng chọn món thì nhân viên ghi lại số bàn và ngày vào giấy oder, đến khi khách gọi món thì ghi thêm vào giấy order tên món khách gọi và số lượng. Sau đó nhân viên phục vụ sẽ chuyển thông tin vừa ghi cho nhân viên pha chế và nhân viên thu ngân. Tại đây nhân viên thu ngân có nhiệm vụ nhập đúng thông tin trên giấy order vào hệ thống.

Nếu khách hàn muốn chuyển sang một bàn khác hoặc muốn gộp bàn lại thì nhân viên phục vụ sẽ ghi nhận lại số bàn khách hàng muốn chuyển (hoặc gộp), đồng thời báo cho nhân viên thu ngân để thao tác trên hệ thống.

Khách hàng có nhu cầu tính tiền nhân viên thu ngân sẽ thanh toán trên hệ thống và báo giá tiền, in hóa đơn nếu khách hàng muốn và sau khi nhận được tiền thanh toán nhân viên thu ngân sẽ thanh toán cho khách hàng và trả lại tiền thừa. Sau khi thanh toán hệ thông sẽ chuyển trạng thái bàn trên hệ thống từ “Có người” sang “Trống”. Hóa đơn hiển thị tên bàn, mã hóa đơn, ngày lập, thông tin đồ uống, đơn giá, số lượng, thành tiền và tổng hóa đơn.

Nhân viên thu ngân có nhiệm vụ mở bàn, nhập đúng thông tin ghi trên oder tên hàng hóa (tên món), số lượng mà nhân viên phục vụ đã ghi. Nếu mở bàn sai có thể mở bàn lại.

Quản lý nhân viên đối với nhân viên sẽ được ghi thông tin vào danh sách thông tin nhân viên và sẽ có một tài khoản riêng để đăng nhập nhân viên có thể tùy chỉnh thay đổi. Cùng với đó quản lý có thể truy cập chỉnh sửa quẻn lý nhân viên, bàn, xem báo cáo thống kê hay chỉnh sửa (thêm xóa sửa) tài khoản của nhân viên.

Quản lý bàn hệ thống sẽ hiển thị giao diện để xem và chỉnh sửa thông tin bàn và nhân viên với quyền admin sẽ có quyền chỉnh sửa xóa bàn và các thông tin liên quan tới nó.

Thống kê và báo cáo doanh thu và lương nhân viên sẽ thêm phần báo cáo cùng với đó phải có chức năng sao lưu và khôi phục dữ liệu từ đó có thể dễ dàng sửa chữa cũng như bảo trì hệ thống.

# 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) 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) 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) 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)), [hàm](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_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) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1).
* 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) 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) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1). 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). 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) 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) 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) 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**

Hình 1:Sơ đồ phân ra chức năng

**\*Mô tả hệ thống:**

* Chức năng quản lý bán hàng

Tạo hóa đơn: tạo hóa đơn thanh toán theo bàn.

Thanh toán hóa đơn: cập nhật trạng thái hóa đơn đã thanh toán.

* Chức năng quán lý bàn

Cập nhật danh sách bàn: thêm, xóa, cập nhật trạng thái của bàn.

* Chức năng quản lý nhân viên

Cập nhật nhân viên: thêm, xóa, sửa nhân viên, thông tin nhân viên.

Tra cứu ca làm việc theo mã nhân viên : tra cứu để thuận tiện cho nhân viên xem ca làm việc của mình.

* Chức năng quản lý hàng hóa

Cập nhật loại hàng hóa: thêm, xóa, sửa loại hàng hóa.

Cập nhật hàng hóa: thêm, xóa, sửa hàng hóa.

Tra cứu hàng hóa theo loại hàng hóa: cho phép ta có thể tra cứu hàng hóa theo loại hàng hóa tương ứng một cách hiệu quả.

* Chức năng báo cáo thống kê

Báo cáo doanh thu: hằng tháng nhân viên với quyền admin sẽ thống kê báo cáo lại doanh thu cho quán.

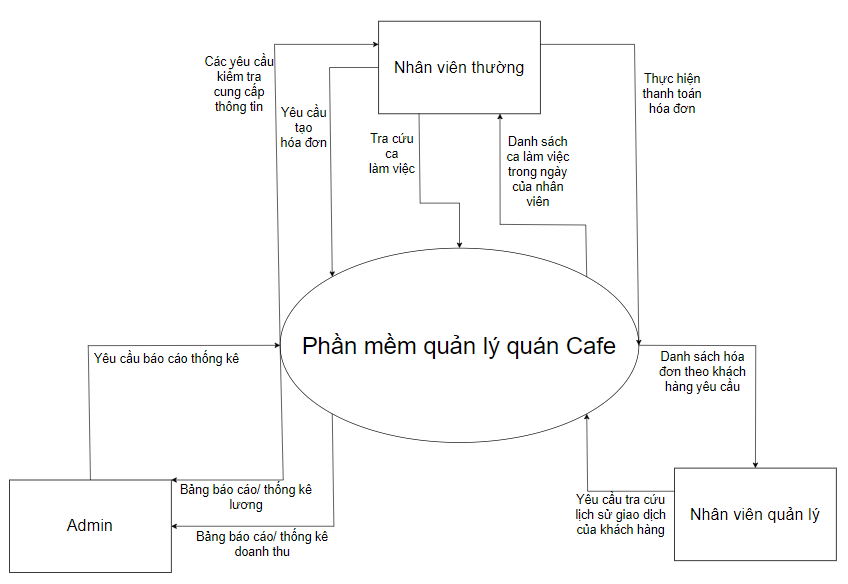
Chức năng quản lý hệ thống

Đăng nhập nhập 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.

## **3.2. Sơ đồ ngữ cảnh**



Hình 2:Sơ đồ ngữ cảnh

## **3.3. Sơ đồ luồng dữ liệu mức đỉnh**

Hình 3:Sơ đồ luồng dữ liệu mức đỉnh

## **3.4. Các sơ đồ luồng dữ liệu mức dưới đỉnh**

### **3.4.1. Sơ đồ mức dưới đỉnh quản lý hệ thống**

Hình 4:Sơ đồ mức dưới đỉnh quản lý hệ thống

### 

### **3.4.2. Sơ đồ mức dưới đỉnh quản lý bán hàng**

Hình 5:Sơ đồ mức dưới đỉnh quản lý bán hàng

### **3.4.3. Sơ đồ mức dưới đỉnh quản lý bàn**

Hình 6:Sơ đồ mức dưới đỉnh quản lý bàn

### **3.4.4. Sơ đồ mức dưới đỉnh quản lý nhân viên**

Hình 7:Sơ đồ mức dưới đỉnh quản lý nhân viên

### **3.4.7. Sơ đồ mức dưới đỉnh quản lý hàng hóa**

Hình 10:Sơ đồ mức dưới đỉnh quản lý hàng hóa

### **3.4.8. Sơ đồ mức dưới đỉnh báo cáo thống kê**

Hình 11:Sơ đồ mức dưới đỉnh báo cáo thống kê

## **3.5. Xây dựng các thực thể**

**Ban** (id, tinhtrang)

**DoUong** (id, ten, gia)

**HoaDon** (id, idBan, thoigianlap, tinhtrang)

**ThongTinHoaDon** (id, idHoaDon, idDoUong, soLuong)

**TaiKhoan** (tendangnhap, matkhau, tenhienthi, loai)

## **3.6. Thiết kế vật lý**

* Bảng bàn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Ý nghĩa** |
| id | int identity |  | PK, not null | Mã bàn |
| Trangthai | bit |  | Not null,  Default: 0 | Trạng thái của bàn |

* Bảng loại đồ uống:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Ý nghĩa** |
| Id | varchar | 50 | PK | Mã đồ uống |
| Ten | nvarchar | 150 | Not null | Tên đồ uống |
| Gia | Int |  |  | Giá đồ uống |

* Bảng tài khoản:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Ý nghĩa** |
| tendangnhap | varchar | 20 | PK, not null | Tên đăng nhập TK |
| matkhau | nvarchar | 1000 | Not null | Mật khẩu |
| tenhienthi | nvarchar | 100 | Not null | Tên hiển thị |
| loai | int |  | FK,Not null | Loại tài khoản |

* Bảng hóa đơn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Ý nghĩa** |
| Id | int identity |  | PK, not null | Mã hóa đơn tự tăng |
| idBan | Int |  | FK,Not null | ID bàn |
| Thoigianlap | Datetime |  | FK,Not null | Thời gian thanh toán |
| Tinhtrang | Bit |  | not null,Default: 0 | Tình trạng đã thanh toán hay chưa66 |
| Tongtien | Int |  | not null,Default: 0 | Tổng tiền thanh toán |

* + Bảng thông tin hóa đơn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Ý nghĩa** |
| Id | int identity |  | PK, not null | Mã TTHD |
| idHoaDon | int |  | FK,Not null | Mã hóa đơn |
| idDoUong | varchar | 50 | FK,Not null | Mã đồ uống |
| Soluong | int |  | not null | Số lượng đồ uống |

## **3.7. Mối quan hệ giữa các thực thể:**

* Mối kết hợp HoaDon-Ban: Một bàn có nhiều hóa đơn bán hàng, một hóa đơn bán hàng chỉ thuộc một bàn.



Hình 12:Mối quan hệ Bàn - Hóa đơn

* Mối kết hợp HoaDon-KhachHang: Mỗi khách hàng có nhiều hóa đơn bán hàng, một hóa đơn bán hàng chỉ thuộc 1 hoặc không thuộc khách hàng nào.



Hình 13:Mối quan hệ Hóa đơn - khách hàng

* Mối kết hợp HoaDon-NhanVien: Mỗi hóa đơn do một nhân viên phụ trách, một nhân viên có thể phụ trách nhiều hóa đơn



Hình 16:Mối quan hệ Hóa đơn - Nhân viên

* Mối kết hợp HoaDon-ChiTietHoaDon: Mỗi hóa đơn có nhiều chi tiết hóa đơn, mỗi chi tiết hóa đơn chỉ thuộc một hóa đơn.



Hình 17: Mối quan hệ Hóa đơn -Chi tiết hóa đơn

* Mối kết hợp ChiTietHoaDon-HangHoa: Mỗi chi tiết hóa đơn có một hàng hóa, mỗi hàng hóa thuộc nhiều chi tiết hóa đơn.

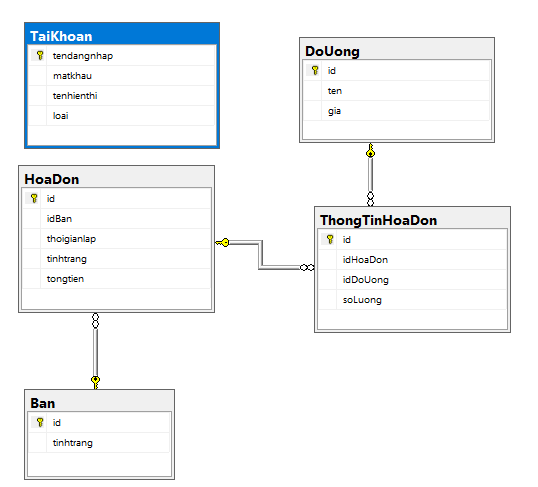


Hình 19:Mối quan hệ Chi tiết hóa đơn - Hàng hóa

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

Hình 22:Mô hình thực thể E-R

## **3.9. Mối quan kết giữa các bảng dữ liệu**



# CHƯƠNG 4: GIAO DIỆN TRONG PHẦN MỀM

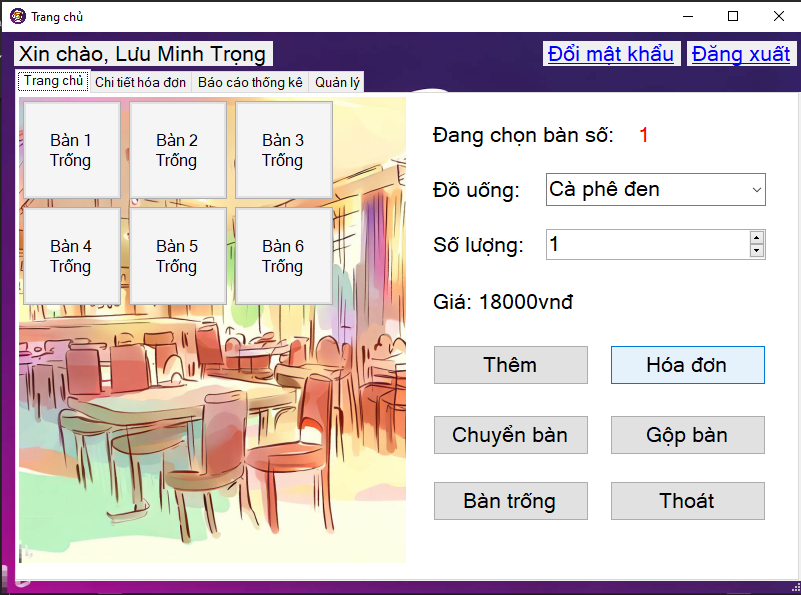
## **4.1. Giao diện đăng nhập hệ thống**

- 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 tên đăng nhập và mật khẩu đã tạo trước để đăng nhập vào hệ thống, và có thể đổi mật khẩu sau khi đăng nhập.

**4.2.Giao diện chính**

- Đây là giao diện chính với tất cả chức năng khi bạn đăng nhập với quyền admin



Hình 29:Giao diện bán hàng

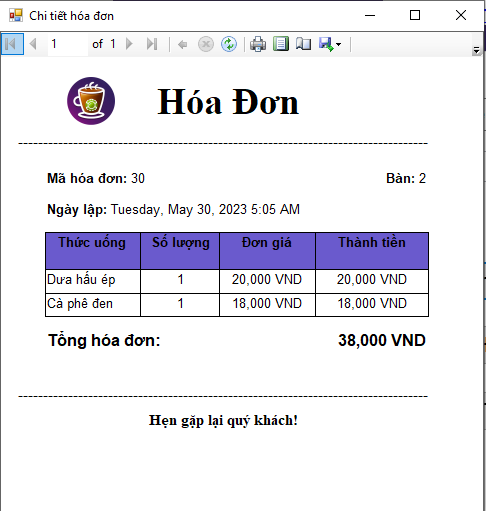
- Ở đây bạn có thể thêm hóa đơn theo bàn với danh sách hiện ở góc dưới phải với một số thông tin và bạn có thể tìm khách hàng và có thể thêm mới khách hàng.

- Có thể thực hiện chuyển bàn từ bàn có người sang bàn trống.

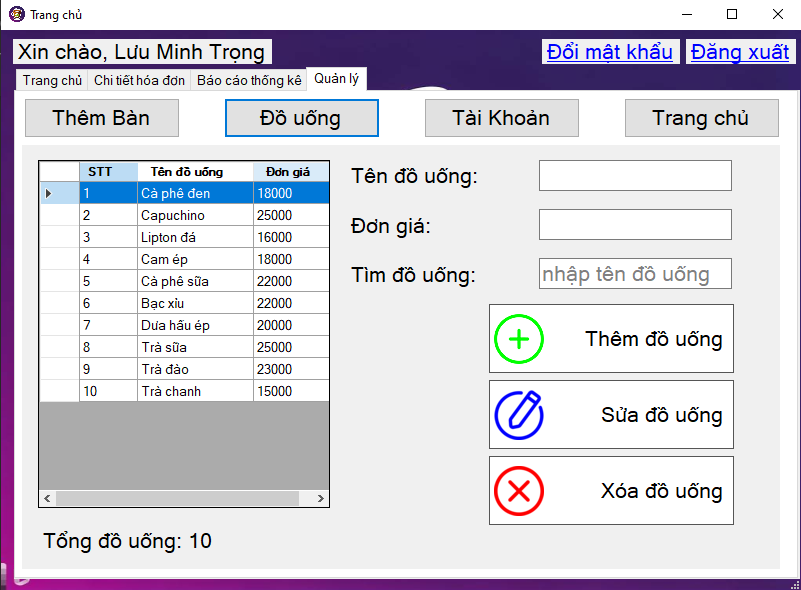
- Khi thanh toán các thông tin cần thiết sẽ hiện ra.

- Hóa đơn nãy được tạo và với trạng thái chưa thanh toán.

- Khi bấm thanh toán các thông tin sẽ hiện lên cùng với dnh sách món của hóa đơn. Nếu bạn muốn in hóa đơn chúng ta sẽ nhấn biểu lượng in hóa đơn ở bên tay phải.

- Khi thanh toán xong hóa đơn sẽ chuyển trạng thái bàn sang trống và hóa đơn chuyển sang đã thanh toán 

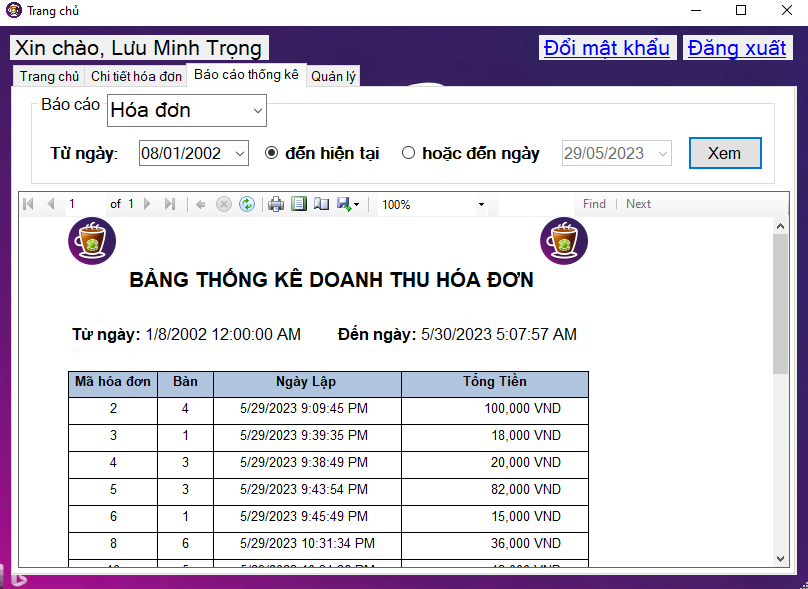
## **4.7. Giao diện quản lý hàng hóa**



Hình 35:Giao diện quản lý hàng hóa

-Thêm loại món ăn và món ăn mới ở chức năng này.

## **4.8. Giao diện quản lý báo cáo thống kê**



# 

# 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 quán cà phê.
* 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 trong phần mềm

+ 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.

## **\*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.
* Chức năng gộp bàn chỉ hoạt động được đối với 2 bàn
* 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**

* 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.

# 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