

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN MINH HUY

175050002

PHẦN MỀM
QUẢN LÝ QUÁN CÀ PHÊ

BÁO CÁO ĐỒ ÁN
LẬP TRÌNH WINDOWS

TP. HỒ CHÍ MINH, 11.07.2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN MINH HUY

175050002

PHẦN MỀM
QUẢN LÝ QUÁN CÀ PHÊ

BÁO CÁO ĐỒ ÁN
LẬP TRÌNH WINDOWS

GIẢNG VIÊN
ThS. VÕ NGỌC TẤN PHƯỚC

TP. HỒ CHÍ MINH, 11.07.2019

KẾ HOẠCH

Tên đề tài: Phần mềm quản lý quán cà phê
Thời gian: 06.05.2019 - 11.07.2019
Thành viên: Nguyễn Minh Huy - 175050002
Thông tin sản phẩm: <p>Phần mềm quản lý sử dụng cơ sở dữ liệu SQL Server, kết nối bằng công nghệ ADO .NET, Winform và ngôn ngữ C#. Giao diện thiết kế đơn giản, dễ sử dụng cho người dùng, có nhiều chức năng cần thiết từ cơ bản đến nâng cao.</p>
Lịch trình: <ul style="list-style-type: none">• 06.05 – 20.05:<ul style="list-style-type: none">- Ôn tập kiến thức về C#, OOP.- Lên ý tưởng thiết kế phần mềm: giao diện, chức năng.• 11.03 – 17.03:<ul style="list-style-type: none">- Tìm hiểu về Winform và ADO .NET trên nền tảng Visual Studio.• 18.03 – 21.03:<ul style="list-style-type: none">- Thiết kế use case và cơ sở dữ liệu.- Thiết kế giao diện và kết nối cơ sở dữ liệu bằng ADO .NET.- Xây dựng phần mềm bằng mô hình 3 lớp.- Xây dựng các chức năng cơ bản của phần mềm.• 03.06 – 26.06:<ul style="list-style-type: none">- Thêm các chức năng thêm, xóa, sửa thức uống, bàn, danh mục...- Sửa lỗi phần mềm và hoàn thiện phần mềm.- Viết báo cáo.

MỞ ĐẦU

Hiện nay, các quán cà phê xuất hiện rất nhiều trên cả nước với rất nhiều quy mô nhỏ và lớn, chủ yếu là do phục vụ nhu cầu của người dân Việt Nam. Không những phát triển chỉ mỗi trong nước, mà mô hình kinh doanh cà phê thực sự là một loại hình kinh doanh có triển vọng trên khắp cả thế giới.

Nhưng không phải quán cà phê nào cũng có thể áp dụng công nghệ để phục vụ cho việc kinh doanh của mình, vì nhiều lí do như chi phí mua và lắp đặt hệ thống cao, phần mềm quản lý hiện nay còn thiếu những chức năng quan trọng, khó sử dụng, hoặc đòi hỏi quá nhiều các thiết bị tương ứng cho mỗi loại phần mềm.

Để đáp ứng được nhu cầu của các chủ quán cà phê, em muốn thiết kế và xây dựng một phần mềm quản lý cơ bản với những chức năng cần thiết nhất cho một quán cà phê, chi phí sử dụng gần như là tối thiểu, tận dụng những thiết bị cơ bản nhất đã có của quán. Và đặc biệt nhất là giao diện dễ sử dụng với mọi đối tượng, hạn chế thời gian hướng dẫn nhân viên sử dụng phần mềm, qua đó tối đa hóa được thời gian và tăng lợi nhuận quán của mình.

Em thực hiện đề tài này để tìm hiểu rõ hơn về công nghệ .NET cùng với nền tảng Winform, nắm được ngôn ngữ lập trình C# cơ bản và nâng cao. Tất nhiên để hoàn thiện sản phẩm và tối ưu hóa giao diện cần rất nhiều thời gian, phần mềm của em chắc chắn sẽ có những thiếu sót về chức năng, những lỗi gặp phải trong quá trình sử dụng. Mong rằng sẽ nhận được góp ý và đánh giá của thầy để em có thể hoàn thiện phần mềm quản lý quán cà phê của mình.

LỜI CAM ĐOAN

Tôi xin cam đoan những điều sau:

1. Những nội dung trong đồ án này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Thạc sĩ Võ Ngọc Tấn Phước.
2. Tất cả các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Tất cả sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Người cam đoan,

Nguyễn Minh Huy

MỤC LỤC

CHƯƠNG 1 ĐẶC TẢ YÊU CẦU	1
1.1 YÊU CẦU PHẦN MỀM	1
1.2 PHÂN TÍCH YÊU CẦU	1
1.3 NHIỆM VỤ	3
1.3.1 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	3
1.3.2 NHIỆM VỤ CỦA ĐỒ ÁN	3
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	4
2.1 CÔNG NGHỆ .NET	4
2.2 NGÔN NGỮ C#	6
2.2.1 KHÁI NIỆM	6
2.2.2 ĐẶC TRƯNG	7
2.3 WINDOWS FORMS	8
2.4 SQL SERVER	9
2.4.1 KHÁI NIỆM	9
2.4.2 MỤC ĐÍCH	9
2.4.3 CÁC THÀNH PHẦN CỦA SQL SERVER	10
2.4.4 CÁC PHIÊN BẢN SQL SERVER	10
2.5 ADO .NET	11
2.5.1 KHÁI NIỆM	11
2.5.2 CÁC ĐỐI TƯỢNG CỦA ADO.NET	12
2.6 MÔ HÌNH 3 LỚP	14
2.6.1 PRESENTATION LAYER (GUI)	15
2.6.2 BUSSINESS LAYER (BLL)	16

2.6.3 DATA LAYER (DAL)	16
2.6.4 DATA TRANSFER OBJECTS (DTO)	16
2.7 CRYSTAL REPORT	17
2.8 BCrypt.NET	18
<u>CHƯƠNG 3 SẢN PHẨM THỰC TẾ</u>	<u>19</u>
3.1 CƠ SỞ DỮ LIỆU	19
3.1.1 MÔ HÌNH QUAN HỆ THỰC THỂ TỔNG QUÁT (ERD)	19
3.1.2 MÔ HÌNH ERD CHI TIẾT	20
3.1.3 THUỘC TÍNH CỦA CÁC LOẠI THỰC THỂ	21
3.1.4 MÔ HÌNH VẬT LÝ	22
3.2 THIẾT KẾ GIAO DIỆN	24
3.2.1 GIAO DIỆN ĐĂNG NHẬP	24
3.2.2 GIAO DIỆN CÀI ĐẶT	25
3.2.3 GIAO DIỆN CHÍNH	25
3.2.4 GIAO DIỆN HÓA ĐƠN (CRYSTAL REPORT)	27
3.2.5 GIAO DIỆN THÔNG TIN TÀI KHOẢN	28
3.2.6 GIAO DIỆN QUẢN LÝ	28
3.3 THIẾT KẾ MÔ HÌNH 3 LỚP	31
<u>CHƯƠNG 4 KẾT LUẬN</u>	<u>32</u>
4.1 KẾT QUẢ ĐẠT ĐƯỢC	32
4.2 PHƯƠNG HƯỚNG PHÁT TRIỂN	32
<u>CHƯƠNG 5 TÀI LIỆU THAM KHẢO</u>	<u>33</u>
5.1 THAM KHẢO LÝ THUYẾT	33
5.2 THAM KHẢO CODE	33

HÌNH ẢNH

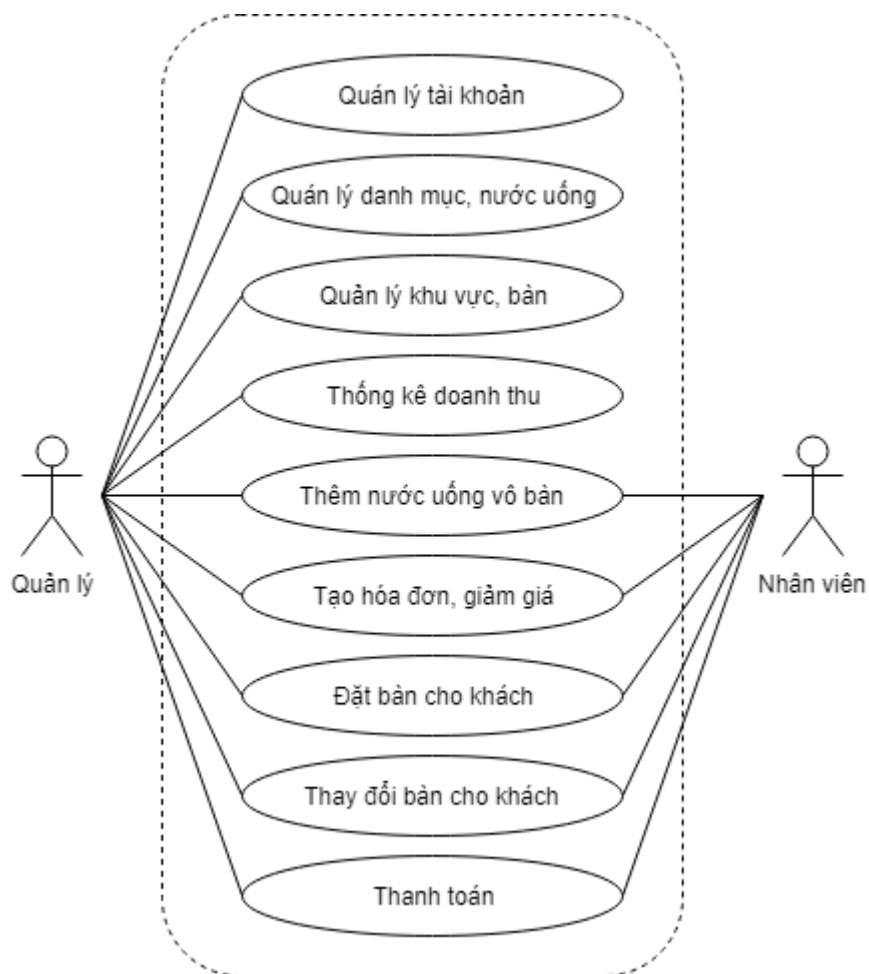
Hình 1.1. Use case diagram.....	1
Hình 2.1 Công nghệ ADO.NET	12
Hình 3.1 Database diagram	19
Hình 3.2 Giao diện đăng nhập.....	24
Hình 3.3 Giao diện cài đặt cấu hình Server	25
Hình 3.4 Giao diện bàn	25
Hình 3.5 Giao diện nước uống	26
Hình 3.6 Giao diện hóa đơn	27
Hình 3.7 Giao diện thông tin tài khoản	28
Hình 3.8 Giao diện quản lý	28
Hình 3.9 Giao diện quản lý nước uống	29
Hình 3.10 Giao diện quản lý bàn	30
Hình 3.11 Giao diện quản lý tài khoản	30
Hình 3.12 Lớp DTO	31
Hình 3.13 Lớp GUI	31
Hình 3.14 Lớp BLL.....	31
Hình 3.15 Lớp DAL	31

CHƯƠNG 1 ĐẶC TẢ YÊU CẦU

1.1 Yêu cầu phần mềm

Xây dựng phần mềm quản lý quán cà phê gồm các chức năng thêm thức uống cho mỗi bàn, quản lý danh mục và thức uống, quản lý bàn theo khu vực, chức năng thanh toán, giảm giá. Giao diện thân thiện, dễ sử dụng.

1.2 Phân tích yêu cầu



Hình 1.1. Use case diagram

Quản lý	Quản lý tài khoản	Thêm xóa sửa tài khoản, thay đổi quyền truy cập vào phần mềm
	Quản lý danh mục và nước uống	Thêm xóa sửa danh mục và nước uống, giá tiền...
	Quản lý khu vực và bàn	Thêm xóa sửa khu vực, bàn
	Thông kê doanh thu	Xem lại doanh thu theo khoảng thời gian được chọn
Nhân viên, Quản lý	Thêm nước uống vô bàn	Lựa chọn bàn và thêm nước uống vào hóa đơn khi được khách gọi
	Tạo hóa đơn, giảm giá	Tạo hóa đơn cho bàn theo thời gian được tạo, thêm giảm giá theo % cho bàn đó
	Đặt bàn cho khách	Khi khách có nhu cầu đặt bàn trước, sẽ lựa chọn bàn đó và đưa về trạng thái đã đặt
	Thay đổi bàn cho khách	Khi khách có yêu cầu chuyển bàn, thay đổi hóa đơn từ bàn này sang bàn khách đổi
	Thanh toán	Thanh toán hóa đơn của bàn được chọn và in hóa đơn

Bảng 1.1 Bảng phân tích chức năng

Quản lý có thể thực hiện tất cả chức năng của nhân viên, với yêu cầu dành cho một quán cà phê quy mô vừa và nhỏ.

Có đầy đủ chức năng của một phần mềm quản lý và thông kê kinh doanh, có khả năng dễ dàng thay đổi thông tin và các loại nước uống, bàn, tài khoản... nhằm giảm thời gian ghi chép và tối thiểu hóa các thao tác thủ công, tăng năng suất hoạt động và giảm chi phí của cà phía nhân viên và quản lý.

1.3 Nhiệm vụ

1.3.1 Đối tượng và phạm vi nghiên cứu

Nghiên cứu công nghệ .NET, ngôn ngữ C#, lập trình hướng đối tượng (OOP), mô hình phát triển phần mềm 3 lớp và các kiến thức liên quan như ADO.NET, kỹ thuật lập trình, xử lý giao diện, sự kiện trong winform...

1.3.2 Nhiệm vụ của đồ án

- Xây dựng phần mềm trên nền tảng ngôn ngữ C# và Winform.
- Cơ sở dữ liệu: SQL Server.
- Kết nối cơ sở dữ liệu bằng công nghệ ADO.NET.
- Xây dựng theo mô hình 3 lớp (3 layers).
- Sử dụng hàm băm mật khẩu.
- Tạo báo cáo bằng Crystal Report.
- Kết nối cơ sở dữ liệu thông qua mô hình Client - Server bằng mạng LAN.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Công nghệ .NET

Công nghệ .NET là một nền tảng (framework) cho phép các ngôn ngữ lập trình khác có thể sử dụng để xây dựng các ứng dụng được phát triển bởi Microsoft. Đây cũng là một nền tảng thực thi ứng dụng và phát triển web chủ yếu trên hệ điều hành Microsoft Windows.

Cốt lõi của nền tảng .NET là một bộ những dòng code được viết sẵn (bởi các lập trình viên Microsoft) mà các lập trình viên có thể sử dụng để tạo các phần mềm một cách nhanh chóng hơn. Do đó, nền tảng .NET được xem là một công cụ bổ sung giúp tiết kiệm thời gian xây dựng, triển khai và chạy các ứng dụng và dịch vụ Web. Nó cung cấp một môi trường đa ngôn ngữ cho phép tích hợp những đầu tư ban đầu dựa trên nền các chuẩn với hiệu năng cao và các ứng dụng, dịch vụ thế hệ kế tiếp và giải quyết những thách thức của việc triển khai và vận hành các ứng dụng trên quy mô Internet.

Thư viện lập trình khổng lồ

Nền tảng .NET bao gồm tập các thư viện lập trình có sẵn rất lớn hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; ứng dụng web; truy cập, kết nối cơ sở dữ liệu; cấu trúc dữ liệu...

Nâng cao năng suất cho các nhà lập trình

Nền tảng .NET giúp tiết kiệm thời gian và đơn giản hóa việc viết ứng dụng cũng như phát triển website bằng cách cung cấp nhiều thành phần thiết kế có sẵn, lập trình viên chỉ cần học cách sử dụng và tùy biến theo mục đích và sáng tạo của riêng mình.

Khả năng biến đổi được thông qua một kiến trúc “ghép nối lỏng”

Đa số các hệ thống lớn, biến đổi được trên thế giới được xây dựng trên những kiến trúc không đồng bộ dựa trên nền thông điệp (Message based). Nhưng công việc xây dựng các ứng dụng trên một kiến trúc như vậy thường rất phức tạp và ít công cụ hơn trong những môi trường phát triển ứng dụng N lớp (N-tier) “ghép nối chặt”. Nền tảng .NET được xây dựng nhằm mang lại những lợi thế về năng suất của kiến trúc “ghép nối chặt” với khả năng biến đổi được và vận hành với nhau của kiến trúc “ghép nối lỏng”.

Hỗ trợ đa ngôn ngữ

Nền tảng .NET cho phép các ứng dụng được viết trong nhiều ngôn ngữ lập trình khác nhau và chúng có khả năng tích hợp chặt chẽ với nhau.

Bảo mật cao

Kiến trúc bảo mật của nền tảng .NET được thiết kế từ dưới lên để đảm bảo các ứng dụng và dữ liệu được bảo vệ thông qua một mô hình bảo mật evidence-based rất tinh vi.

Tận dụng những dịch vụ của hệ điều hành

Windows đa dạng các dịch vụ có sẵn với bất kỳ nền tảng nào như: bảo mật tích hợp, truy cập dữ liệu một cách toàn diện, mô hình đối tượng thành phần đáng tin cậy, các giao diện người dùng tương tác và các giám sát quá trình giao dịch. Nền tảng .NET đã tận dụng lợi thế này để đưa ra cho mọi người theo cách dễ sử dụng nhất.

Các ngôn ngữ được lập trình trên .NET

Có rất nhiều ngôn ngữ lập trình được sử dụng trên nền tảng .NET. Tuy nhiên, trong đó có hai ngôn ngữ lập trình chính mà bạn có thể sử dụng với .NET để tạo ra các ứng dụng desktop hoặc ứng dụng Web là: VB.NET và C#.

Bạn có thể thực hiện một dự án VB.NET và C# trên VS nếu lúc cài đặt bạn chọn hỗ trợ hai ngôn ngữ này. Có thể nói một ứng dụng viết bằng VB.NET thì có thể viết lại bằng C# và ngược lại vì cả C# và VB.NET đều cùng được .NET dịch về chung một ngôn ngữ Assembly. Với các thuật toán ràng buộc khắt khe hơn, lập trình bằng C# thường mất công hơn so với VB.NET nhưng đảm bảo chính xác hơn.

2.2 Ngôn ngữ C#

2.2.1 Khái niệm

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... trở nên rất dễ dàng.

2.2.2 Đặc trưng

Các đặc điểm để làm cho C# trở thành một ngôn ngữ lập trình chuyên nghiệp:

C# là ngôn ngữ 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++ hoặc 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 hiện đại

Một vài khái niệm khá mới mẻ khá mơ hồ với các bạn vừa mới học lập trình, như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn... Đây là những đặc tính được cho là của một ngôn ngữ hiện đại cần có. Và C# chứa tất cả các đặc tính ta vừa nêu trên. Các bạn sẽ dần tìm hiểu được các đặc 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 (*Object-oriented programming, 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.

C# là một ngôn ngữ ít từ khóa

C được sử dụng để mô tả ngôn ngữ 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 bạn nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, lấy ví dụ ngôn ngữ C# làm điển hình nhé. Nếu bạn học sâu về C# bạn sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

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 dễ 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.

2.3 Windows Forms

Windows Form thường viết tắt là Winform nó là thuật ngữ chỉ việc phát triển các ứng dụng giao diện người dùng bằng cách sử dụng các thành phần xây dựng sẵn còn được gọi là các điều khiển. Nói một cách dễ hiểu winform cho phép tạo GUI cho các ứng dụng chạy trên desktop.

Winform cho phép người phát triển tạo ra các giao diện người dùng sử dụng các thành phần khác nhau. Các thành phần này được gọi là các điều khiển. Những điều khiển này cho phép chúng ta thu thập thông tin từ người dùng cũng như hiển thị các thông tin để người sử dụng có thể xem.

Một form được chạy trên một máy tính cục bộ và 1 form có thể truy cập đến các tài nguyên khác nhau: các thư mục, bộ nhớ, tệp tin, cơ sở dữ liệu...

Winform phù hợp các ứng dụng trên desktop: ứng dụng quản lý thông tin, ứng dụng tương tác trực tiếp với người dùng.

2.4 SQL Server

2.4.1 Khái niệm

SQL Server hay còn gọi là Microsoft SQL Server, viết tắt là MS SQL Server. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, và nó cũng là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS).

SQL Server cung cấp đầy đủ công cụ để quản lý, từ giao diện GUI cho đến việc sử dụng ngôn ngữ truy vấn SQL. Ngoài ra điểm mạnh của nó là Microsoft có khá nhiều nền tảng kết hợp hoàn hảo với SQL Server như ASP.NET, C# xây dựng Winform, bởi vì nó hoạt động hoàn toàn độc lập.

2.4.2 Mục đích

Chung quy lại chúng ta sử dụng bất kỳ một hệ quản trị CSDL nào cũng để lưu trữ dữ liệu và SQL cũng không ngoại lệ. Tuy nhiên điều mà ta mong đợi ở nó là các tính năng giúp việc sử dụng hiệu quả hơn như:

- Cho phép tạo nhiều cơ sở dữ liệu
- Duy trì lưu trữ bền vững
- Bảo mật cao
- Phân tích dữ liệu bằng SSAS - SQL Server Analysis Services
- Tạo được báo cáo bằng SSRS - SQL Server Reporting Services
- Thực hiện quá trình ETL (Extract-Transform-Load) bằng SSIS - SQL Server Integration Services.

2.4.3 Các thành phần của SQL Server

Do hoạt động theo mô hình Client - Server nên nó được chia làm hai thành phần chính:

- **Workstation:** Được cài trên các thiết bị vận hành, nó là các phần mềm tương tác với máy chủ server. Ví dụ: SSMS, SSCM, Profiler, BIDS, SQLEM
- **Server:** Được cài trên máy chủ chính (máy chủ tập trung), đó là các dịch vụ như: SQL Server, SQL Server Agent, SSIS, SSAS, SSRS, SQL Browser, SQL Full Text Search

Bạn có thể cài nhiều phiên bản của SQL Server trên cùng một máy chủ, điều này giúp tiết kiệm chi phí mua Server nếu hệ thống bạn hoạt động cần nhiều phiên bản khác nhau, bảo mật cũng tách biệt hoàn toàn giúp hệ thống an toàn hơn.

2.4.4 Các phiên bản SQL Server

Hiện nay có rất nhiều nền tảng nên SQL Server đã tạo ra nhiều phiên bản khác nhau, đáp ứng cho một mục đích cụ thể, giúp tiết kiệm tối đa về chi phí và tăng hiệu quả. Các phiên bản đó gồm:

- **Enterprise** - là phiên bản cao cấp, có đầy đủ các tính năng
- **Standard** - gói chuẩn, tính năng tương đối đầy đủ, tuy nhiên vẫn kém hơn so với Enterprise
- **Workgroup** - dành cho các công ty làm việc từ xa với nhiều máy tách biệt
- **Web** - dành cho các ứng dụng website
- **Developer** - dành cho nhà phát triển, chứa đầy đủ tính năng nhưng chỉ phân quyền cho một người duy nhất, dễ dàng nâng cấp lên bản Enterprise mà không cần phải cài đặt lại từ đầu.
- **Express** - là bản cơ bản, sử dụng tối đa 1 CPU và 1GB RAM, dung lượng lưu trữ tối đa là 10GB. Bản này thường được sử dụng khi bạn học SQL Server ở trường.

- **Compact** - nhúng miễn phí vào các môi trường phát triển ứng dụng web. Kích thước tối đa của cơ sở dữ liệu là 4GB.
- **Datacenter** - là bản không giới hạn về bộ nhớ và hỗ trợ 25 bản cài đặt.
- **Business Intelligence** - được giới thiệu ở phiên bản SQL Server 2012, có đầy đủ các tính năng của bản Standard và một số tính năng như: Power View và Power Pivot.
- **Enterprise Evaluation** - bản dành để học tập và thử nghiệm phần mềm, chỉ được dùng trong 6 tháng.

2.5 ADO .NET

2.5.1 Khái niệm

ADO.NET là một bộ các thư viện hướng đối tượng (OOP) cho phép bạn tương tác với dữ liệu nguồn. Thông thường thì dữ liệu nguồn là một cơ sở dữ liệu (database), nhưng nó cũng có thể là file text, excel hoặc XML. Theo những mục tiêu của hướng dẫn này, chúng ta sẽ chỉ xem xét tới cách ADO.NET làm việc với database.

Như bạn có thể biết rằng, có rất nhiều loại database hiện nay như Microsoft SQL Server, Microsoft Access, Oracle, Borland Interbase, và IBM DB2... Để làm rõ hơn phạm vi của loạt bài này, tất cả ví dụ sẽ sử dụng SQL Server.

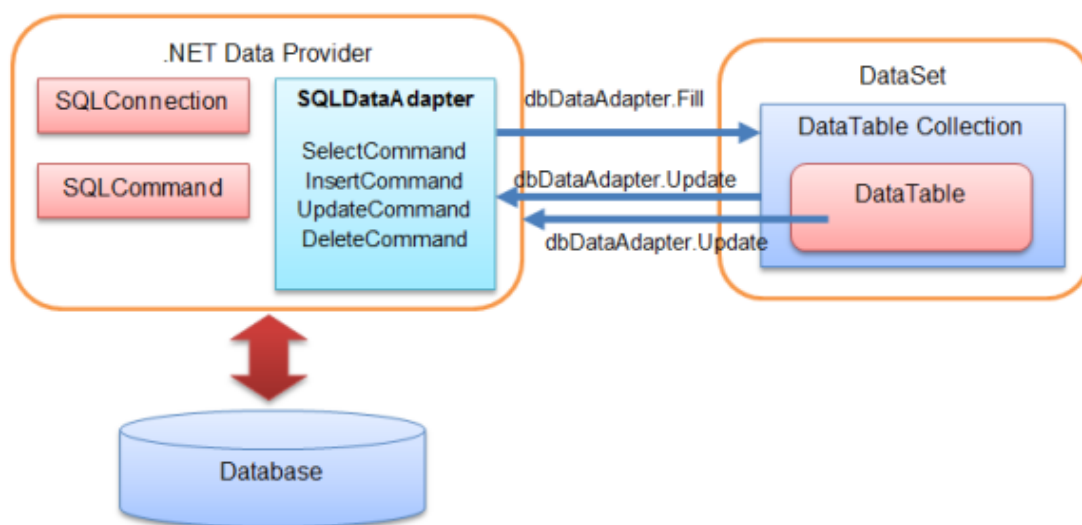
Data Provider

Chúng ta biết rằng ADO.NET cho phép tương tác với các loại dữ liệu và kiểu database. Mỗi loại dữ liệu cần một cách thức khác nhau để có thể truy xuất. Các loại dữ liệu cũ sử dụng giao thức ODBC, các loại dữ liệu mới hơn sử dụng giao thức OleDb. Vì vậy cần có một thư viện thống nhất để làm việc với chúng, đây chính là lý do mà ADO.NET được tạo ra.

ADO.NET cung cấp một cách thức chung để tương tác với nguồn dữ liệu, nhưng với mỗi loại dữ liệu bạn phải sử dụng một thư viện khác nhau. Các thư viện này được

gọi là Data Provider và thường được đặt tên theo giao thức hoặc loại dữ liệu mà chúng cho phép bạn truy xuất. Table 1 liệt kê các data provider phổ biến, tiền tố (prefix) API mà chúng sử dụng và kiểu dữ liệu mà bạn có thể làm việc.

Một ví dụ có thể giúp bạn hiểu nghĩa của API prefix. Một trong những đối tượng ADO.NET bạn sẽ học là connection, chúng cho phép thiết lập một kết nối đến nguồn dữ liệu. Nếu bạn dùng OleDb Data Provider để kết nối, bạn sẽ cần dùng để đối tượng connection tên là OleDbConnection. Tương tự, đối tượng connection sẽ dùng prefix là Odbc hoặc Sql cho đối tượng OdbcConnection hoặc SqlConnection để kết nối với nguồn dữ liệu tương ứng. Như đã nói lúc đầu, ta sẽ học cách làm việc với SQL Server nên các đối tượng API của chúng ta sẽ có prefix là Sql.



Hình 2.1 Công nghệ ADO.NET

2.5.2 Các đối tượng của ADO.NET

SqlConnection

Để tương tác với database, bạn phải có một kết nối tới nó. Kết nối giúp xác định database server, database name, user name, password, và các tham số cần thiết để kết

nối tới database. Một đối tượng connection được dùng bởi đối tượng command vì thế chúng sẽ biết database nào để thực thi lệnh.

SqlCommand

Quá trình tương tác với database cần phải biết hành động nào bạn muốn xảy ra. Điều này được thực hiện bởi đối tượng command. Bạn dùng đối tượng command để gửi một câu lệnh SQL tới database. Một đối tượng command dùng một đối tượng connection để xác định database nào sẽ được truy xuất. Bạn có thể dùng một đối tượng command riêng lẻ để thực thi lệnh trực tiếp, hoặc để gán một tham chiếu của đối tượng command cho một SqlDataAdapter – đối tượng giữ các command sẽ làm việc trên một nhóm dữ liệu như sẽ đề cập tới trong phần dưới.

SqlDataReader

Nhiều thao tác dữ liệu đòi hỏi bạn chỉ lấy một luồng dữ liệu để đọc. Đối tượng data reader cho phép bạn lấy được kết quả của một câu lệnh SELECT từ một đối tượng command. Để tăng hiệu suất, dữ liệu trả về từ một data reader là một luồng dữ liệu fast forward-only. Có nghĩa là bạn chỉ có thể lấy dữ liệu từ luồng theo một thứ tự nhất định. Mặc dù điều này có lợi về mặt tốc độ, nhưng nếu bạn cần phải thao tác dữ liệu, thì một DataSet sẽ là một đối tượng tốt hơn để làm việc.

DataSet

Đối tượng DataSet là một thể hiện của dữ liệu trong bộ nhớ. Chúng chứa nhiều đối tượng DataTable, bên trong DataTable lại có nhiều column và row, giống như các database table thông thường. Bạn thậm chí có thể định nghĩa dữ liệu giữa các table để tạo các quan hệ parent-child. DataSet được thiết kế đặc biệt để giúp quản lý dữ liệu trong bộ nhớ và để hỗ trợ các thao tác không cần kết nối (disconnected) trên dữ liệu. DataSet là một đối tượng được dùng bởi tất cả Data Provider, đó là lý do tại sao nó không có một Data Provider prefix trong tên gọi.

SqlDataAdapter

Đôi lúc dữ liệu mà bạn làm việc là read-only và bạn ít khi cần thay đổi dữ liệu nguồn. Vài trường hợp cần lưu trữ tạm dữ liệu trong bộ nhớ để hạn chế truy xuất đến database. Data adapter làm điều này dễ dàng bằng cách giúp bạn quản lý dữ liệu trong chế độ ngắt kết nối. Data adapter sẽ đổ vào DataSet khi đọc dữ liệu và thực hiện thay đổi dữ liệu một lượt vào database.

Data adapter chứa một tham chiếu đến đối tượng connection và mở/đóng kết nối tự động khi đọc và ghi dữ liệu vào database. Hơn nữa, data adapter chứa đối tượng command cho những thao tác SELECT, INSERT, UPDATE và DELETE trên dữ liệu. Bạn sẽ có một data adapter được định nghĩa cho mỗi table trong một DataSet và nó sẽ quản lý các giao tiếp với database cho bạn. Tất cả những gì bạn cần làm là chỉ cho data adapter khi nào nạp hoặc ghi vào database.

2.6 Mô hình 3 lớp

Mô hình 3-layer gồm có 3 phần chính:

Presentation Layer (GUI): Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện (winform, webform...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

Business Logic Layer (BLL): Layer này phân ra 2 thành nhiệm vụ. Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL. Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

Data Access Layer (DAL): Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa...).

Ưu điểm

Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.

Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.

Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại khi có sự thay đổi giữa hai môi trường (winform sang webform) thì chỉ việc thay đổi lại lớp GUI.

Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.

Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

2.6.1 Presentation Layer (GUI)

UI Components: gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng...

UI Process Components: là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI...

2.6.2 Bussiness Layer (BLL)

Lớp này gồm 4 thành phần:

Service Interface: là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.

Bussiness Workflows: chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

Bussiness Components: chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc. Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

Bussiness Entities: thường được sử dụng như Data Transfer Objects (DTO). Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu (DataSets, XML...) hay các lớp đối tượng đã được tùy chỉnh.

2.6.3 Data Layer (DAL)

Data Access Logic Components: chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu (Data Sources) như XML, file system... Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.

Service Agents: giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

2.6.4 Data Transfer Objects (DTO)

Ngoài ra còn có lớp DTO, đây không phải là layer, đây chỉ là 1 gói dữ liệu được trao đổi giữa các lớp. Gói dữ liệu này được xây dựng dưới dạng lớp đối tượng.

2.7 Crystal Report

Crystal Reports là công cụ thiết kế báo cáo cho phép bạn tạo ra những báo cáo bằng cách tìm và định dạng dữ liệu từ một nguồn dữ liệu hay từ những nguồn dữ liệu khác nhau.

Bên cạnh đó, Crystal Reports có một ngôn ngữ riêng để tính toán và một số tính năng khác để biến những dữ liệu thô thành những báo cáo đầy tính chuyên nghiệp. Bạn có thể tạo những báo cáo đi từ những danh sách đơn giản chỉ gồm vài cột cho đến những báo cáo phức tạp có kèm biểu đồ, bảng và chỉ số Key Performance Indicator (KPI).

Ngoài ra, Crystal Reports có một số hàm API cũng như công cụ đặc biệt được thiết kế dành cho các nhà phát triển phần mềm cho phép hội nhập các báo cáo này vào trong ứng dụng riêng của họ.

Đặc điểm của Crystal Reports

- Bạn không cần mở một ứng dụng riêng rẽ để thiết kế báo cáo khi dùng Crystal Reports.
- Đối với Windows Form, Crystal Reports cho phép xem báo cáo và cung cấp tất cả các chức năng cho người sử dụng bao gồm xoay sâu vào chi tiết, truy xuất, xuất khẩu...
- Đối với ASP.NET, cung cấp phần lớn các chức năng trong Windows Form Viewer trong môi trường DHTML “Zero Client”, nghĩa là không client nào được nạp xuống hay cài đặt.
- Crystal Reports giúp việc truy cập dữ liệu trở nên dễ dàng hơn.
- Crystal Reports cho phép truy cập những dữ liệu nguồn khác nhau.

2.8 BCrypt .NET

Thuật toán Bcrypt phát sinh mã được sử dụng rộng rãi cho các chức năng liên quan đến bảo mật khi lưu trữ password của người dùng. Các thuật toán cũ MD5, SHA đã không còn được khuyến dùng.

Mức độ an toàn của bcrypt liên quan đến tốc độ của thuật toán. Bcrypt là rất chậm, thậm chí đến hàng giây để sinh ra một mã hash. Điều này có nghĩa là kiểu tấn công brute force khó mà thi hành vì cần quá nhiều thời gian.

Bcrypt hiện đang là tiêu chuẩn bảo mật không chính thức cho việc băm mật khẩu. Nó xuất phát từ mật mã khối Blowfish, sử dụng bảng tra cứu từ bộ nhớ để tạo ra giá trị băm. Điều này đồng nghĩa với việc phải sử dụng một số dung lượng bộ nhớ nhất định để tạo ra giá trị băm. Việc này có thể được thực hiện trên CPU, nhưng sẽ trở nên cồng kềnh hơn nhiều khi sử dụng GPU do hạn chế về bộ nhớ. Bcrypt đã ra đời được 14 năm, dựa trên một mật mã có tuổi đời hơn 20 năm. Nó đã được xem xét và thử nghiệm kỹ càng và nhờ đó được coi như tiêu chuẩn cho việc băm mật khẩu.

Thực ra vẫn có một điểm yếu, đó là đơn vị xử lý FPGA. Khi bcrypt bắt đầu được phát triển, mối đe dọa chủ yếu của nó là ASICs chuyên biệt được xây dựng nhằm mục đích chuyên tấn công các hàm băm. Ngày nay những ASICs đó sẽ là GPUs (phương thức tấn công bruteforce trên thực tế vẫn có thể chạy trên GPU, nhưng không hoàn toàn song song) có giá bán vừa túi tiền và lý tưởng đối với những quá trình đa luồng nhưng tấn công bruteforce.

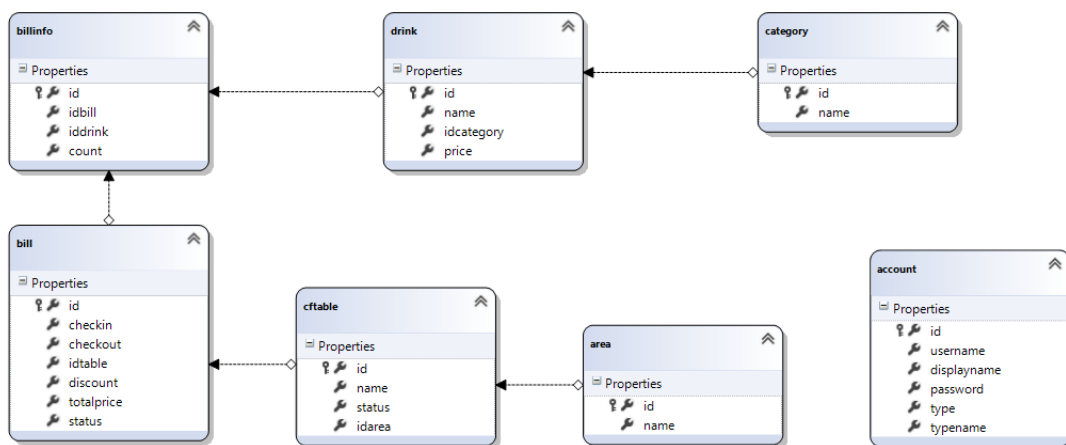
FPGAs (Field Programmable Gate Arrays) cũng tương tự như GPUs nhưng việc quản lý bộ nhớ lại rất khác. Bruteforcing bcrypt có thể được thực hiện hiệu quả hơn trên những chip này so với trên GPUs, nhưng nếu mật khẩu đủ dài thì vẫn không thể bị phá bằng phương thức này.

CHƯƠNG 3 SẢN PHẨM THỰC TẾ

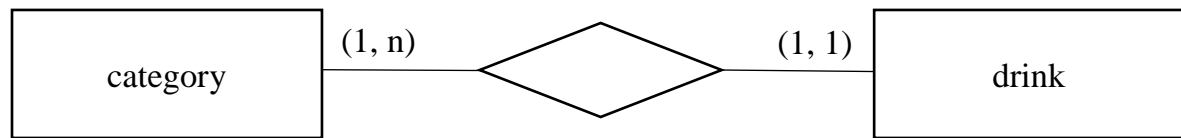
3.1 Cơ sở dữ liệu

3.1.1 Mô hình quan hệ thực thể tổng quát (ERD)

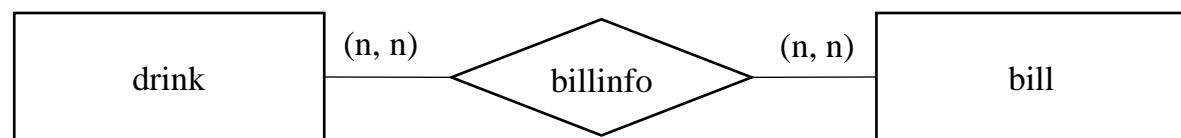
- **area** (id, name),
- **cftable** (id, name, idarea, status),
- **category** (id, name),
- **drink** (id, name, idcategory, price),
- **account** (id, username, displayname, password, type, typename),
- **bill** (id, checkin, checkout, idtable, discount, totalprice, status),
- **billinfo** (id, idbill, iddrink, count)



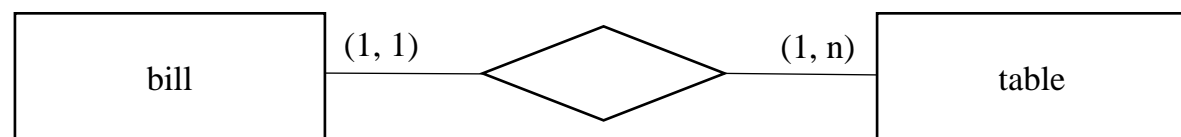
Hình 3.1 Database diagram

3.1.2 Mô hình ERD chi tiết

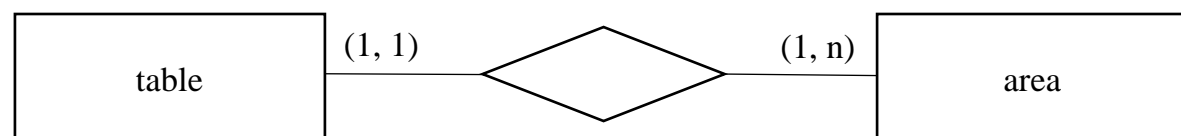
Mỗi nước uống có một danh mục, mỗi danh mục có nhiều loại nước uống. Mỗi quan hệ giữa nước uống và danh mục là quan hệ 1 - nhiều.



Mỗi hóa đơn có một hoặc nhiều nước uống, mỗi nước uống có thể có trong một hoặc nhiều hóa đơn. Mỗi quan hệ giữa nước uống và hóa đơn là quan hệ nhiều - nhiều.



Mỗi hóa đơn của một bàn, mỗi bàn có thể có một hoặc nhiều hóa đơn. Mỗi quan hệ giữa hóa đơn và bàn là quan hệ 1 - nhiều.



Mỗi bàn trong một khu vực, mỗi khu vực có thể có một hoặc nhiều bàn. Mỗi quan hệ giữa bàn và khu vực là quan hệ 1 - nhiều.

3.1.3 Thuộc tính của các loại thực thể

area (khu vực)

- id (mã khu vực)
- name (tên khu vực)

cftable (bàn)

- id (mã bàn)
- name (tên bàn)
- status (trạng thái)
- idarea (mã khu vực)

category (danh mục)

- id (mã danh mục)
- name (tên danh mục)

drink (nước uống)

- id (mã nước uống)
- name (tên nước uống)
- idcategory (mã danh mục)
- price (giá tiền)

bill (hóa đơn)

- id (mã hóa đơn)
- checkin (ngày vào)
- checkout (ngày thanh toán)
- idtable (mã bàn)
- discount (giảm giá)
- totalprice (tổng tiền hóa đơn)
- status (trạng thái)

billinfo (chi tiết hóa đơn)

- id (mã chi tiết hóa đơn)
- idbill (mã hóa đơn)
- iddrink (mã nước uống)
- count (số lượng)

account (tài khoản)

- id (mã tài khoản)
- username (tên đăng nhập)
- displayname (tên hiển thị)
- password (mật khẩu)
- type (loại tài khoản)
- typename (tên loại tài khoản)

3.1.4 Mô hình vật lý

- **area**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã khu vực	khóa chính
name	nvarchar	100	tên khu vực	not null

- **cftable**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã khu vực	khóa chính
name	nvarchar	100	tên khu vực	not null
status	nvarchar	100	trạng thái	not null
idarea	int		mã khu vực	khóa ngoại

- **category**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã danh mục	khóa chính
name	nvarchar	100	tên danh mục	not null

- **drink**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã nước uống	khóa chính
name	nvarchar	100	tên nước uống	not null
price	float		giá tiền	default 0
idcategory	int		mã danh mục	khóa ngoại

- **bill**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã hóa đơn	khóa chính
checkin	date		ngày lập hóa đơn	getdate()
checkout	date		ngày thanh toán	not null
idtable	int		mã khu vực	khóa ngoại
discount	int	1 - 100	giảm giá	default 0
totalprice	float		tổng tiền	
status	int		trạng thái	default 0

- **bill**

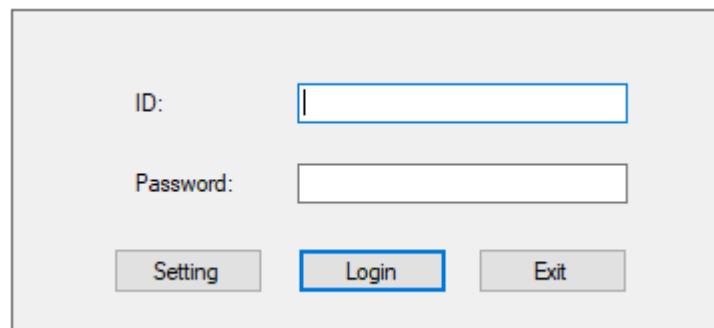
Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã chi tiết hóa đơn	khóa chính
idbill	int		mã hóa đơn	khóa ngoại
iddrink	int		mã nước uống	khóa ngoại
count	int		số lượng	default 0

- **account**

Tên cột	Kiểu	Chiều dài	Diễn giải	Ghi chú
id	int	identity	mã tài khoản	khóa chính
username	varchar	100	tên đăng nhập	not null
displayname	nvarchar	100	tên hiển thị	
password	varchar	100	mật khẩu	not null
type	int		loại tài khoản	default 0
typename	nvarchar	100	tên loại tài khoản	

3.2 Thiết kế giao diện

3.2.1 Giao diện đăng nhập



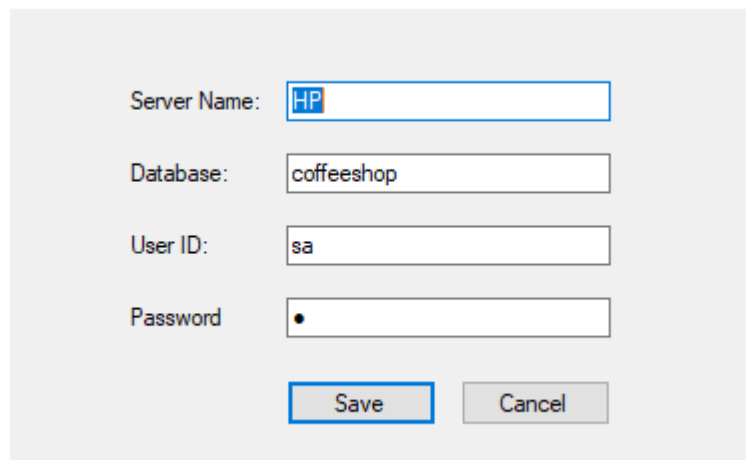
The image shows a login form with a light gray background. It contains two input fields: 'ID:' and 'Password:'. Below the input fields are three buttons: 'Setting', 'Login', and 'Exit'. The 'Login' button is highlighted with a blue border.

Hình 3.2 Giao diện đăng nhập

Khi đăng nhập vào tài khoản sẽ kiểm tra quyền đăng nhập (type) là quản lý hoặc nhân viên để cấp quyền sử dụng phần mềm tương ứng.

Mật khẩu đăng nhập được băm bằng thư viện BCrypt .NET và so sánh với mật khẩu đã lưu trong cơ sở dữ liệu.

3.2.2 Giao diện cài đặt



Server Name:

Database:

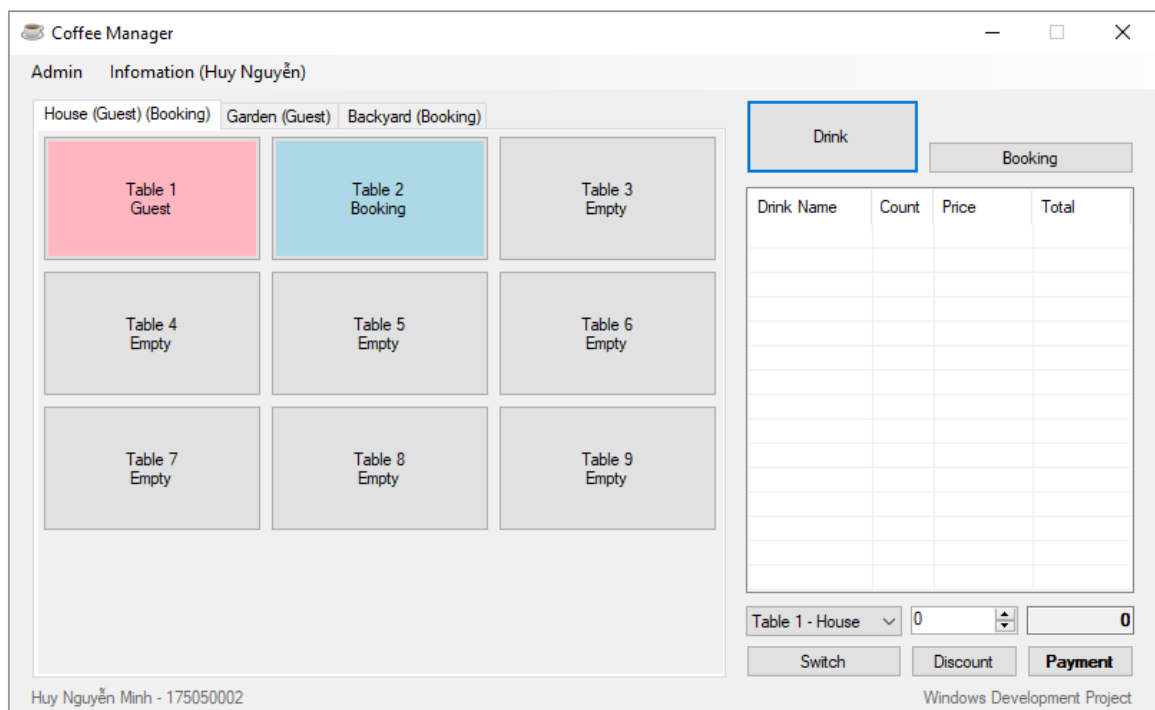
User ID:

Password:

Hình 3.3 Giao diện cài đặt cấu hình Server

Cài đặt cấu hình của Server, phần mềm sẽ lưu cài đặt sau mỗi lần thay đổi.

3.2.3 Giao diện chính



Coffee Manager

Admin Information (Huy Nguyễn)

House (Guest) (Booking) Garden (Guest) Backyard (Booking)

Table 1 Guest

Table 2 Booking

Table 3 Empty

Table 4 Empty

Table 5 Empty

Table 6 Empty

Table 7 Empty

Table 8 Empty

Table 9 Empty

Drink

Booking

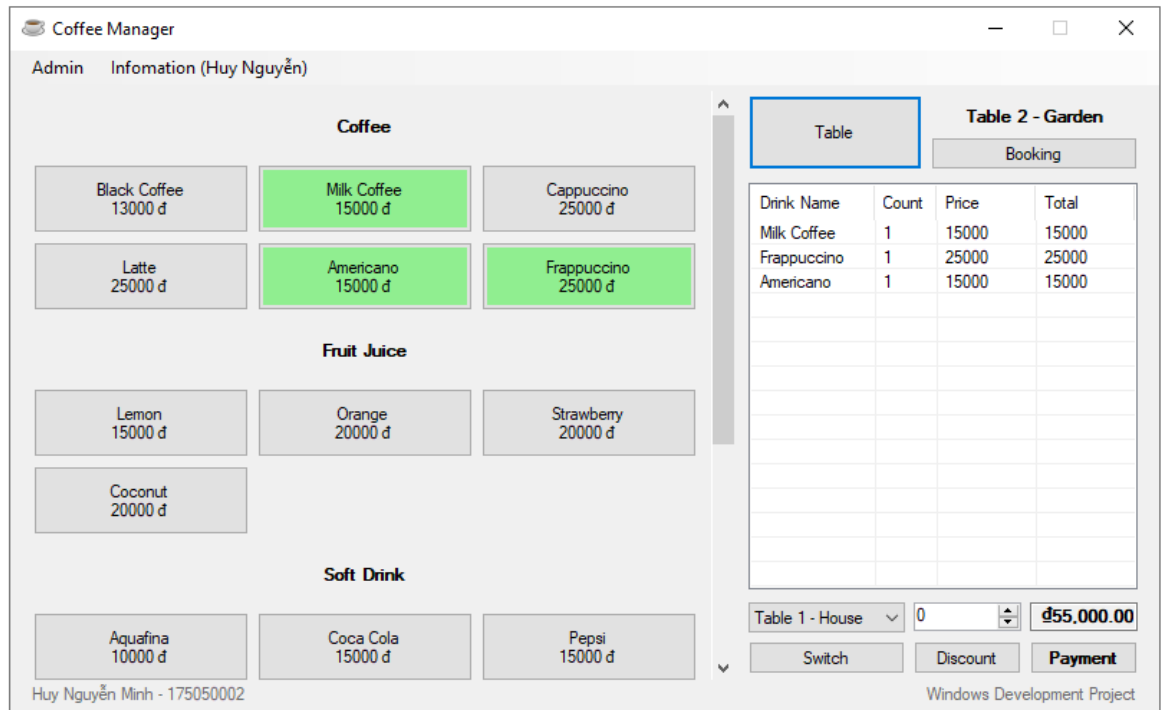
Drink Name	Count	Price	Total

Table 1 - House 0 0

Switch Discount Payment

Huy Nguyễn Minh - 175050002 Windows Development Project

Hình 3.4 Giao diện bàn

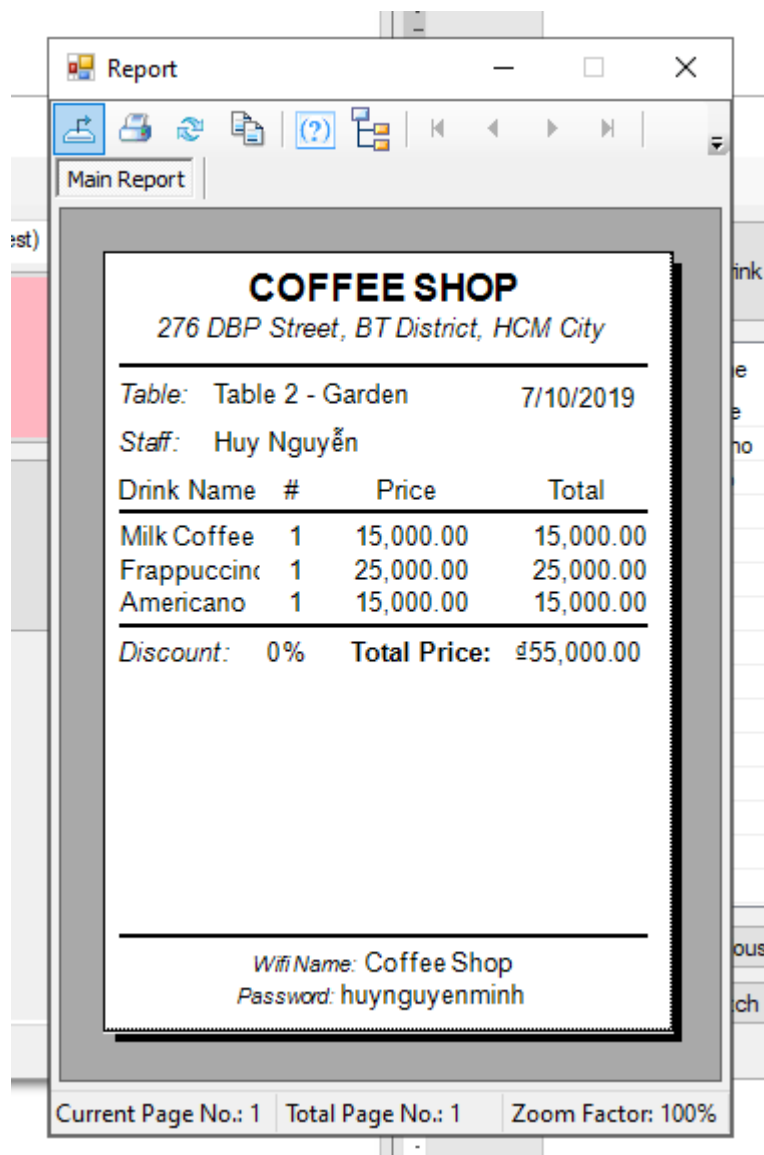


Hình 3.5 Giao diện nước uống

Chức năng:

- Hiện thị bàn theo khu vực, dựa theo cơ sở dữ liệu.
- Hiện thị nước uống theo danh mục.
- Chức năng đặt bàn.
- Chức năng đổi bàn.
- Chức năng thêm giảm giá tổng đơn hàng.
- Thanh toán.

3.2.4 Giao diện hóa đơn (Crystal Report)



Hình 3.6 Giao diện hóa đơn

Khi lựa chọn thanh toán bàn đang có hóa đơn sẽ hiện ra bảng hóa đơn theo kích thước chuẩn (74mm), đây là hình ảnh minh họa cho hóa đơn được in bằng máy in nhiệt.

3.2.5 Giao diện thông tin tài khoản

The 'Profile' dialog box contains the following fields and buttons:

- ID:
- Password:
- New Password:
- Confirm:
- Buttons: Save, Cancel

Hình 3.7 Giao diện thông tin tài khoản

3.2.6 Giao diện quản lý

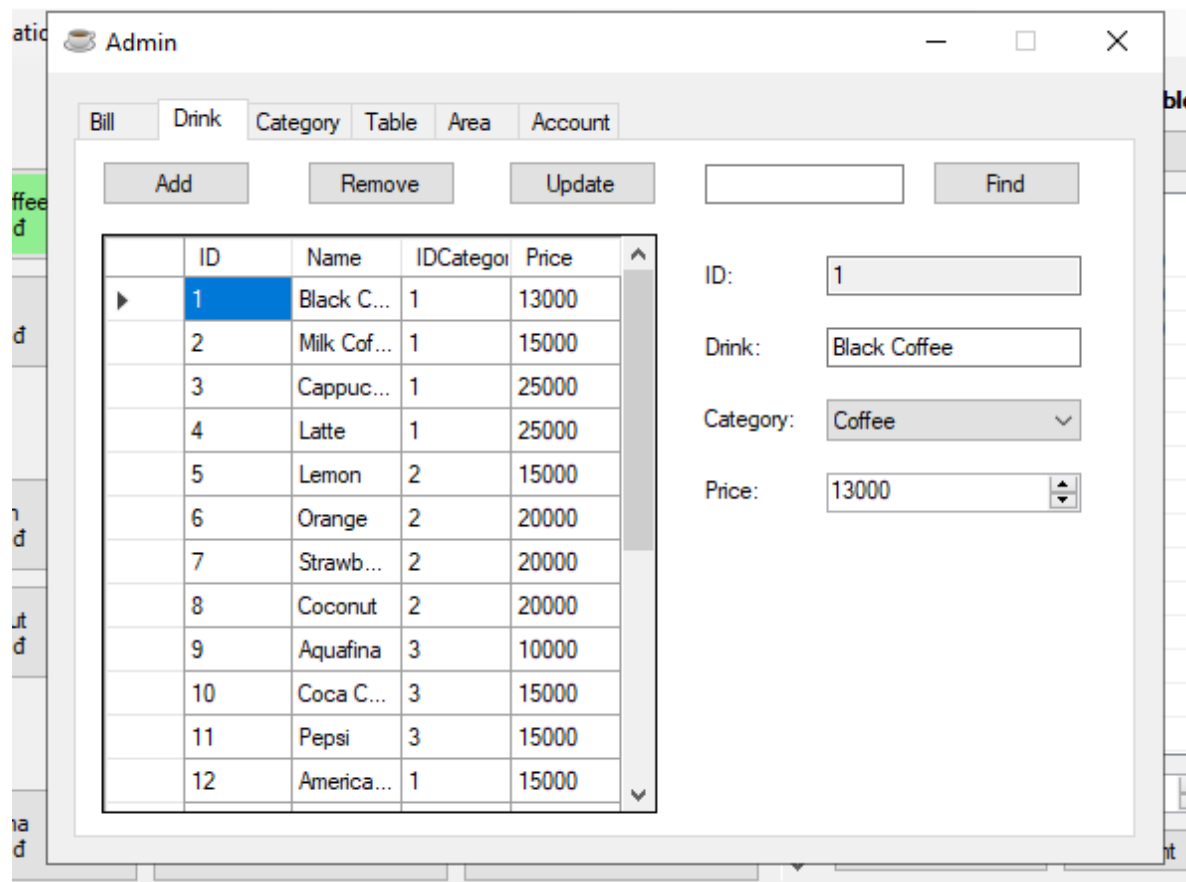
The 'Admin' window displays a table with the following data:

	Table Name	Checkin	Checkout	Discount	Total Price
▶	Table 5	7/1/2019	7/1/2019	0	0
	Table 2	7/1/2019	7/1/2019	0	30000
	Table 2	7/1/2019	7/1/2019	0	35000
	Table 6	7/1/2019	7/1/2019	0	70000
	Table 6	7/1/2019	7/2/2019	0	35000
	Table 5	7/1/2019	7/1/2019	0	15000
	Table 5	7/1/2019	7/1/2019	0	0
	Table 5	7/1/2019	7/1/2019	0	0
	Table 5	7/1/2019	7/2/2019	20	32000
	Table 1	7/1/2019	7/2/2019	0	15000
	Table 3	7/1/2019	7/2/2019	0	45000
	Table 2	7/1/2019	7/2/2019	0	15000
	Table 4	7/2/2019	7/2/2019	5	15000

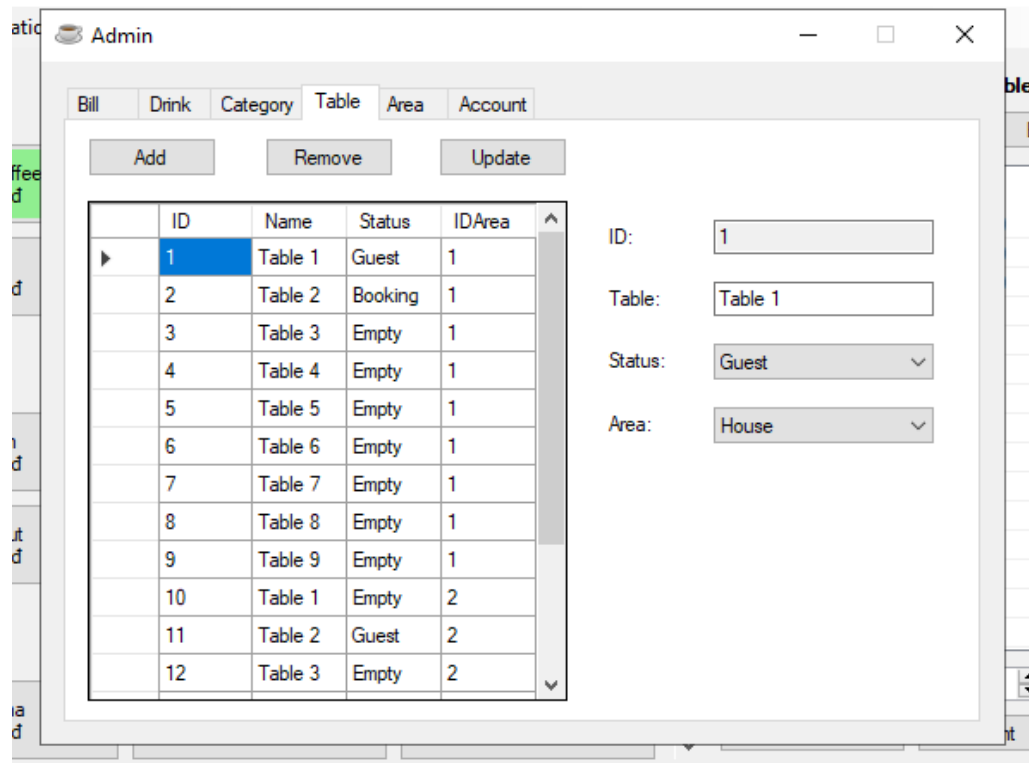
Hình 3.8 Giao diện quản lý

Chức năng

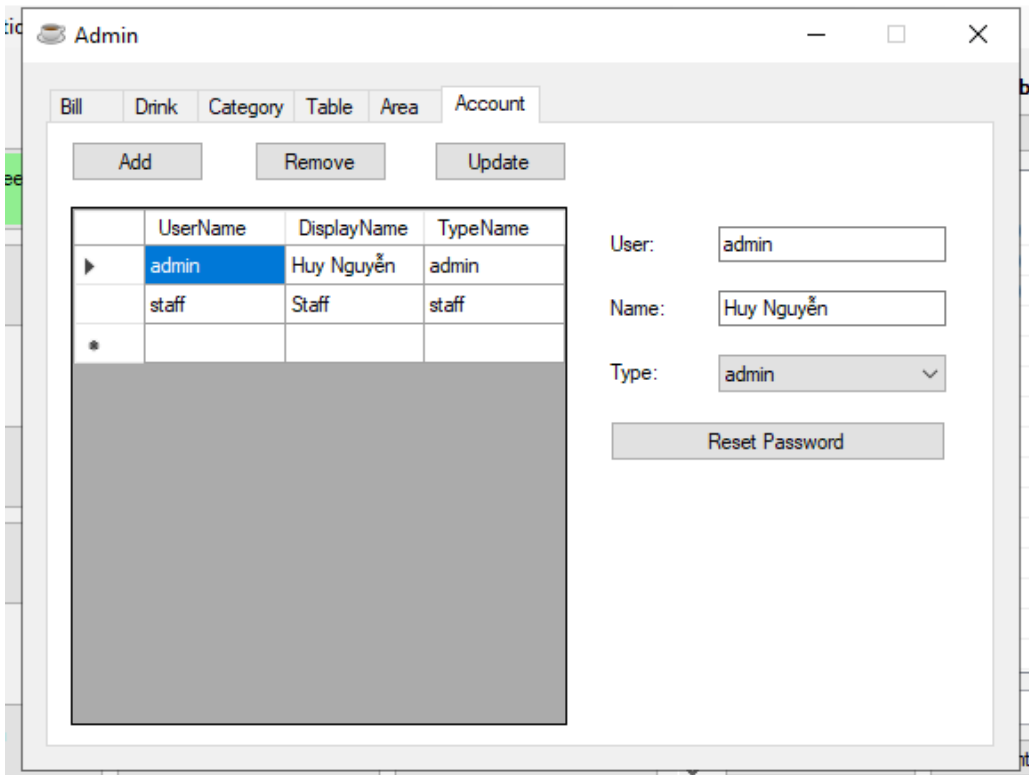
- Xem doanh thu theo ngày những hóa đơn đã thanh toán.
- Tìm kiếm nước uống.
- Thêm, xóa, sửa nước uống.
- Thêm, xóa, sửa danh mục.
- Thêm, xóa, sửa bàn
- Thêm, xóa, sửa khu vực.
- Thêm, xóa, sửa, phục hồi mật khẩu tài khoản.



Hình 3.9 Giao diện quản lý nước uống

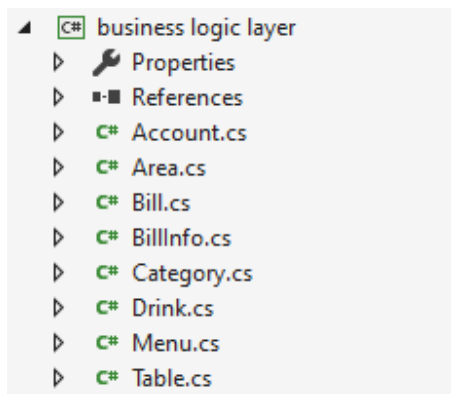


Hình 3.10 Giao diện quản lý bàn

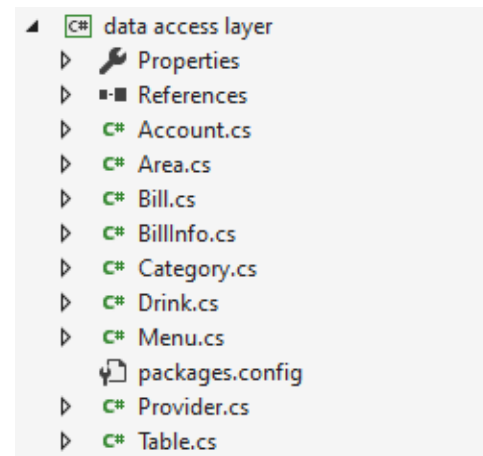


Hình 3.11 Giao diện quản lý tài khoản

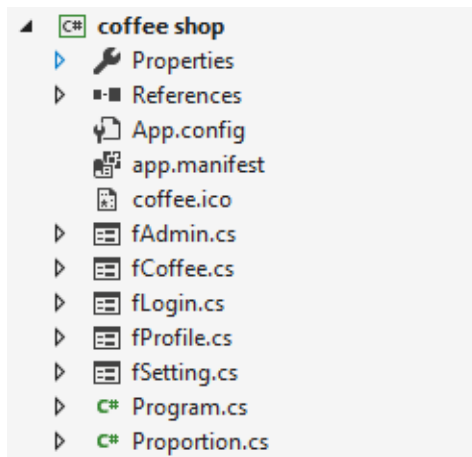
3.3 Thiết kế mô hình 3 lớp



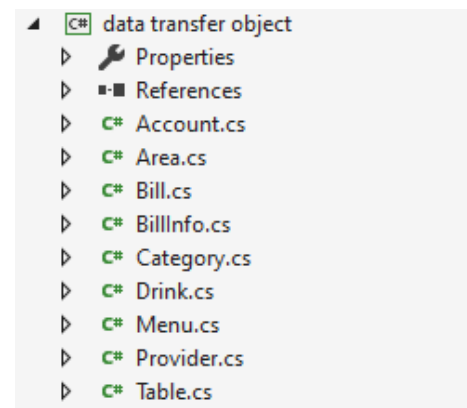
Hình 3.12 Lớp BLL



Hình 3.13 Lớp DAL



Hình 3.14 Lớp GUI



Hình 3.15 Lớp DTO

Phần mềm được xây dựng theo mô hình 3 lớp GUI - BLL - DAL

Lớp DTO có các lớp con tương ứng trong cơ sở dữ liệu để giao vận dữ liệu giữa các lớp trên.

CHƯƠNG 4 KẾT LUẬN

Phần mềm quản lý quán cà phê là một phần mềm về quản lý sử dụng trên máy tính hệ điều hành Windows thông qua mô hình Client – Server. Ngày nay các phần mềm quản lý đã có rất nhiều, tuy nhiên phần mềm của em chỉ đạt được một số yêu cầu cơ bản và còn một số khuyết điểm vẫn chưa thực hiện được.

4.1 Kết quả đạt được

Đối với cá nhân người thực hiện

- Hoàn thành nghiên cứu các công nghệ và kiến thức liên quan khi thực hiện.
- Phát triển hơn về kỹ năng lập trình với ngôn ngữ C#.

Phần mềm

- Tối ưu hóa được thời gian và lợi nhuận cho quán cà phê.
- Phần mềm cho phép quản lý (thêm, xóa, sửa) các thông tin cần thiết.

4.2 Phương hướng phát triển

- Thiết kế giao diện thân thiện với người dùng hơn.
- Phát triển thêm những chức năng cần thiết.
- Tối ưu hóa thời gian xử lý.
- Nâng cao khả năng bảo mật.
- Sửa lỗi còn tồn đọng trong phần mềm.
- Tích hợp các công nghệ mới nhất cho phần mềm.

CHƯƠNG 5 TÀI LIỆU THAM KHẢO

5.1 Tham khảo lý thuyết

- <http://bugnetproject.com/khac-nhau-giua-net-va-cac-ngon-ngu-lap-trinh-khac/>
- <https://freetuts.net/c-sharp-la-gi-tong-quan-ve-c-sharp-1045.html>
- <https://freetuts.net/ms-sql-server-la-gi-dac-diem-cua-no-1574.html>
- <https://yinyangit.wordpress.com/2011/07/31/ado-net-tutorial-lesson-01-introduction-to-ado-net/>
- <https://techtalk.vn/mo-hinh-3-lop-co-gi-hay.html>
- https://www.slideshare.net/nghia_790a/crystal-repor
- <http://trankyphat.com/programing/su-dung-ham-bam-de-bao-ve-mat-khau-hash-password-218/>

5.2 Tham khảo code

- <https://stackoverflow.com/questions/tagged/winforms>
- <https://www.codeproject.com>
- <https://www.howkteam.vn/course/>
- <http://diendan.congdongcviet.com/forum.php>
- <https://codervietnam.com/>
- <https://www.howkteam.vn/learn/lap-trinh/lap-trinh-c-net-7-5>
- <https://www.howkteam.vn/course/lap-trinh-winform-co-ban-27>