|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  -----**🙦🕮🙤**-----  A red circle with white text and a book and key  Description automatically generated    **ĐỒ ÁN LẬP TRÌNH C#**  ĐỀ TÀI:  **XÂY DỰNG ỨNG DỤNG QUẢN LÝ QUÁN ĂN**   |  |  | | --- | --- | | Giảng viên hướng dẫn | : **Nguyễn Văn Hoàng** | | Học viên thực hiện | : **Hà Thị Thanh Dung**  **Trần Thanh Hải**  **Dương Chí Duy** | | Lớp | : CT06N | | Khóa | : 6 | | Chuyên ngành | : Công nghệ thông tin |   **TP. Hồ Chí Minh, 2024** |

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay,chúng em đã nhân được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè. Với lòng biết ơn sâu sắc nhất,chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thấy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, chúng em xin chân thành cảm ơn Thầy Nguyễn Văn Hoàng - người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn chúng em hoàn thành tốt bài báo cáo này trong thời gian qua

Bản báo cáo đồ án thực hiên trong khoảng thời gian vừa qua. Bước đầu đi vào thực tế của chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiên bổ sung, nâng cao ý thức của mình.

Em xin chân thành cảm ơn!

**LỜI MỞ ĐẦU**

Ngày nay với sự bùng nổ và phát triển mạnh mẽ của lĩnh vực Công nghệ thông tin đặc biệt với sự ra đời của Internet, nó đang dần trở thành một phần không thể thiếu của cuộc sống con người. Công nghệ thông tin đang được phát triển và ứng dụng mạnh mẽ trong các lĩnh vực quản lý sản xuất, quản lý kinh doanh, dịch vụ, quản lý xã hội cũng như tất cả các lĩnh vực khác. Thực tế đã cho thấy hiệu quả của tin học khi áp dụng vào công tác quản lý, nó đã làm giảm bớt công tác bàn giấy đồng thời góp phần đáng kể trong việc thống kẻ tránh những sai sót trong kinh doanh. Có thể nói tin học đã trở thành một công cụ hữu hiệu đem lại hiệu quả trong công tác quản lý và kinh doanh.

Quản lý nhà hàng là một trong những nhu cầu thiết thực đó. Việc duy trì và phát triển nhà hảng không chỉ dừng lại ở những món ăn ngon, mà còn cần giải quyết được những phức tạp, khó khăn trong công việc quản lý về tài chính, nhân sự... và những chiến lược kinh doanh của nhà hàng.

Đáp ứng những nhu cầu đó,chúng em đã chọn đề tài “Xây dựng ứng dụng quản lý nhà hàng để xây dựng và phát triển. Ứng dụng được thiết kế dựa trên thực tế nghiệp vụ của nhà hàng để có khả năng đáp ứng và hỗ trợ tốt trong công việc quản lý và kinh doanh của nhà hàng, tránh nhầm lẫn sai sót, đồng thời hỗ trợ cho việc tìm kiếm thông tin nhanh chóng, chính xác.

**MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc177132174)

[**DANH MỤC BẢNG BIỂU** 6](#_Toc177132175)

[**CHƯƠNG 1: TỔNG QUAN** 7](#_Toc177132176)

[**1.1** **Lý do chọn đề tài:** 7](#_Toc177132177)

[**1.2** **Giới thiệu chung:** 8](#_Toc177132178)

[**1.3** **Phạm vi hệ thống:** 10](#_Toc177132179)

[**1.4** **Giới thiệu ngôn ngữ và công cụ:** 11](#_Toc177132180)

[**1.4.1 Ngôn ngữ C#** 11](#_Toc177132181)

[**1.4.2 SQL Server** 12](#_Toc177132182)

[**1.4.3 Visual Studio** 12](#_Toc177132183)

[**1.4.4 GUNA Framework** 13](#_Toc177132184)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 15](#_Toc177132185)

[**2.1 Kiến trúc hệ thống** 15](#_Toc177132186)

[**2.2 Danh sách tác nhân** 16](#_Toc177132187)

[**2.3 Danh sách Use case** 17](#_Toc177132188)

[**2.4 Mô hình use case:** 20](#_Toc177132189)

[**2.4.1 Sơ đồ use-case chính:** 21](#_Toc177132190)

[**2.4.2 Sơ đồ UseCase Admin** 22](#_Toc177132191)

[**2.5 Đặc tả chi tiết các Usecase(UC)** 24](#_Toc177132192)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU** 37](#_Toc177132193)

[**3.1** **Biểu đồ diagram:** 37](#_Toc177132194)

[**3.2** **Bảng Account:** 37](#_Toc177132195)

[**3.3** **Bảng Food:** 38](#_Toc177132196)

[**3.4** **Bảng Food Category:** 38](#_Toc177132197)

[**3.5** **Bảng Bill:** 39](#_Toc177132198)

[**3.6** **Bảng Bill Info:** 39](#_Toc177132199)

[**3.7** **Bảng Table Food:** 40](#_Toc177132200)

[**3.8** **Các thủ tục (Procedure) và hàm (Function):** 40](#_Toc177132201)

[- **Procedure:** 40](#_Toc177132202)

[- **Function:** 43](#_Toc177132203)

[**CHƯƠNG 4: CÀI ĐẶT VÀ THỰC NGHIỆM** 45](#_Toc177132204)

[**1.** **Giao diện bằng Guna UI:** 45](#_Toc177132205)

[**1.1** **Giao diện đăng nhập:** 45](#_Toc177132206)

[**1.2** **Giao diện quản lý:** 45](#_Toc177132207)

[**1.3** **Giao diện nhân viên:** 46](#_Toc177132208)

[**1.4** **Giao diện đổi mật khẩu:** 47](#_Toc177132209)

[**2.** **Cấu trúc:** 47](#_Toc177132210)

[**KẾT LUẬN** 49](#_Toc177132211)

[**TÀI LIỆU THAM KHẢO** 50](#_Toc177132212)

# **DANH MỤC BẢNG BIỂU**

[Bảng 2.0.1 Danh sách các use case 18](#_Toc176888592)

[Bảng 2.0.2 Usse case thêm tài khoản 23](#_Toc176888593)

[Bảng 2.0.3 Use case xóa tài khoản 24](#_Toc176888594)

[Bảng 0.4 Use case sửa tài khoản 25](#_Toc176888595)

[Bảng 0.5 Use case thêm danh mục món ăn 27](#_Toc176888596)

[Bảng 0.5 Use case xóa danh mục món ăn 28](#_Toc176888597)

[Bảng 0.5 Use case xóa danh mục món ăn 31](#_Toc176888598)

[Bảng 0.5 Use case tạo hóa đơn 32](#_Toc176888599)

[Bảng 0.5 Use case xem lịch sử hóa 34](#_Toc176888600)

[Bảng 0.5 Bảng Account 36](#_Toc176888601)

[Bảng 0.5 Bảng Food 36](#_Toc176888602)

[Bảng 0.5 Bảng Food category 37](#_Toc176888603)

[Bảng 0.5 Bảng Bill 37](#_Toc176888604)

[Bảng 0.5 Bảng Bill Info 37](#_Toc176888605)

[Bảng 0.5 Bảng Table Food 38](#_Toc176888606)

**CHƯƠNG 1: TỔNG QUAN**

* 1. **Lý do chọn đề tài:**

Ngày nay với sự phát triển mạnh mẽ của lĩnh vực Công nghệ thông tin đặc biệt với sự ra đời của Internet, nó đang dần trở thành một phần thiết yếu của cuộc sống con người. Công nghệ thông tin đang được phát triển và ứng dụng mạnh mẽ trong các lĩnh vực quản lý sản xuất, quản lý kinh doanh, dịch vụ, quản lý xã hội cũng như tất cả các lĩnh vực khác. Quản lý nhà hàng là một trong những nhu cầu thiết thực đó, một ứng dụng quản lý nhà hàng là một giải pháp hoàn toàn hợp lý và đáng được mong đợi. Một ứng dụng mà chủ nhà hàng có thể quản lý được nhà hàng một cách nhanh chóng, chính xác cùng với đó là sự tiện lợi khi khách hàng có thể giao tiếp với nhà hàng một cách nhanh gọn và dễ dàng mà không cần phải đi đâu xa khi chỉ cần ngồi nhà hay văn phòng làm việc vẫn có thể sử dụng một cách dễ dàng.

Đi vào thực tiễn để có thể phát triển đề tài, nhận thấy hiện tại xung quanh khu vực của nhà hàng là các khu công nghiệp và một số công trường đang thi công nên đây phần lớn là nơi sinh sống và làm việc của đa số các công nhân viên. Công việc ngày càng nhiều, công nghệ thông tin ngày một phát triển nên nhu cầu cuộc sống của con người ngày một tăng cao. Mọi người quá bận rộn với công việc hoặc không muốn ra đường nên nhu cầu gọi đồ ăn ship đến là rất cao và lượng người đến với nhà hàng là ngày một ít. Ứng dụng được sinh ra nhằm đáp ứng các đặc điểm riêng dựa trên thực tế nghiệp vụ của nhà hàng, chúng em nhận thấy nhà hàng có rất ít khách ngồi ăn tại quản mà đa phần là mua mang về hoặc ship tận nơi cùng với đó là tác hại của covid-19 dẫn đến chúng ta phải cách ly tập trung hoặc không được tụ tập quá 5 người nên ý tưởng của em là xây dựng một ứng dụng vừa để quản lý nhà hàng vừa giúp nhà hàng kinh doanh một cách thuận lợi và phát triển. Nhà hàng sẽ tập trung bán hàng mang về và nhận ship để có thể duy trì kinh doanh cũng như ngày một phát triển.

Việc duy trì và phát triển nhà hàng không chỉ dừng lại ở những món ăn ngon, mà còn cần giải quyết được những phức tạp, khó khăn trong công việc quản lý về tài chính, nhân sự... và những chiến lược kinh doanh của nhà hàng.

Đáp ứng những nhu cầu nêu trên, đề tài “Xây dựng ứng dụng quản lý nhà hàng” của chúng em được lên kế hoạch và bắt đầu đi vào nghiên cứu cũng như phát triển. Ứng dụng được thiết kế dựa trên thực tế nghiệp vụ của nhà hàng để có khả năng đáp ứng và hỗ trợ tốt nhất trong công việc quản lý và kinh doanh của nhà hàng, tránh nhầm lẫn sai sót, đồng thời hỗ trợ cho việc tìm kiếm thông tin trở nên nhanh chóng và chính xác.

* 1. **Giới thiệu chung:**

Định nghĩa: Quản lý bán hàng có thể được định nghĩa là hoạt động quản trị của cá nhân hoặc một nhóm người thuộc lĩnh vực bán hàng hoặc những người hỗ trợ trực tiếp cho lực lượng bán hàng nhằm triển khai việc cung cấp các sản phẩm hoặc dịch vụ của doanh nghiệp trên cơ sở chiến lược kinh doanh, nguồn lực của doanh nghiệp và môi trường kinh doanh có liên quan.

Quản lý bán hàng có nhiệm vụ phân phối hàng hóa đến những kênh phân phối theo mục tiêu đã đề ra nhằm tối ưu việc hiện diện của hàng hóa, dịch vụ với chi phí hợp lý, đạt hiệu quả tối đa và đạt được mục tiêu marketing của nhà hàng về ngắn hạn cũng như dài hạn.

Chức năng của hệ thống quản lý bán hàng:

Trong thời đại bùng nổ công nghệ 4.0 như hiện nay, một hệ thống giúp quản lý việc bán hàng là không thể thiếu trong hoạt động kinh doanh của doanh nghiệp. Hệ thống quản lý chính là những phần mềm quản lý bán hàng, giúp cho người dùng đặc biệt là các doanh nghiệp kiểm soát quản lý được hàng hóa một cách chặt chẽ hơn. Việc quản lý của hệ thống sẽ được kiểm soát trong tất cả các khâu: Nhập hàng, quản lý đơn hàng, quản lý kho hàng, quản lý nhân viên, chăm sóc khách hàng....

Có một hệ thống giúp quản lý bán hàng sẽ giúp doanh nghiệp tối ưu được hoạt động kinh doanh của mình, cụ thể thông qua những chức năng của hệ thống quản lý bán hàng.

Chiến lược bán hàng:

Đây là kế hoạch tổng thể nhằm đạt được mục tiêu doanh thu và mở rộng thị trường. Chiến lược bao gồm các phương pháp tiếp cận khách hàng, định giá sản phẩm, và các hoạt động khuyến mãi.

Quản lý đội ngũ bán hàng:

Điều này liên quan đến việc tuyển dụng, đào tạo, động viên và đánh giá hiệu suất của nhân viên bán hàng. Một đội ngũ bán hàng hiệu quả có thể tạo ra sự khác biệt lớn trong việc đạt được mục tiêu doanh thu.

Dự báo bán hàng:

Quá trình dự đoán nhu cầu sản phẩm hoặc dịch vụ trong tương lai dựa trên các dữ liệu lịch sử và xu hướng thị trường. Dự báo chính xác giúp lập kế hoạch sản xuất và quản lý hàng tồn kho hiệu quả.

Quản lý quan hệ khách hàng (CRM):

Hệ thống và quy trình giúp quản lý và phân tích tương tác với khách hàng, từ đó cải thiện sự hài lòng và tăng cường mối quan hệ lâu dài với khách hàng.

Kênh phân phối:

Các phương tiện hoặc cách thức mà sản phẩm hoặc dịch vụ được đưa đến tay khách hàng. Quản lý kênh phân phối bao gồm việc lựa chọn và hợp tác với các nhà phân phối, đại lý và bán lẻ.

Phân tích dữ liệu bán hàng:

Sử dụng dữ liệu để phân tích xu hướng bán hàng, hành vi của khách hàng, và hiệu suất của đội ngũ bán hàng. Phân tích này giúp đưa ra quyết định dựa trên dữ liệu thay vì cảm tính.

Chiến lược giá cả:

Xác định giá bán của sản phẩm hoặc dịch vụ dựa trên các yếu tố như chi phí sản xuất, giá thị trường, và giá của đối thủ cạnh tranh. Giá cả là yếu tố quan trọng ảnh hưởng đến doanh thu và lợi nhuận.

Quản lý hàng tồn kho:

Quá trình theo dõi và quản lý lượng hàng tồn kho để đảm bảo rằng có đủ sản phẩm để đáp ứng nhu cầu của khách hàng mà không gây ra tình trạng thừa hoặc thiếu hàng.

Khuyến mãi và chiết khấu:

Các chương trình khuyến mãi và chiết khấu giúp thúc đẩy doanh số bán hàng và thu hút khách hàng mới. Chúng bao gồm giảm giá, tặng quà, và các chiến lược khuyến mãi khác.

Ứng dụng quản lý quán ăn cần thực hiện các chức năng cơ bản sau:

• Quản lý thực đơn: Thêm, sửa, xóa , tìm kiếm các món ăn và đồ uống.

• Quản lý đơn hàng: Tạo, chỉnh sửa và theo dõi các đơn hàng.

• Quản lý nhân viên: Thêm, sửa, xóa thông tin nhân viên và quản lý phân quyền.

• Quản lý hóa đơn: Xem và quản lý hóa đơn thanh toán của khách hàng.

• Quản lý bàn ăn: Cập nhật tình trạng bàn ăn (trống, đã đặt, đang phục vụ). Chuyển bàn

• Báo cáo doanh thu: Xem các báo cáo về doanh thu hàng ngày, hàng tuần, và hàng tháng.

• Thanh toán bằng MOMO

* 1. **Phạm vi hệ thống:**

Phạm vi nghiên cứu chỉ áp dụng tại địa bàn một cửa hàng nhất định, chưa mở rộng đến chuỗi cửa hàng, do đó các kết quả tìm được có thể chỉ hữu ích trong một phạm vi nhất định, không có ý nghĩa trên chuỗi cửa hàng và không bao quát rộng, bởi những cửa hàng khác nhau sẽ có mức thu nhập và mua hàng khác nhau. Nếu có cơ hội và điều kiện để mở rộng và nghiên cứu dự án thì dự án sẽ mang lại nhiều lợi ích cho doanh nghiệp và đặc biệt trong lĩnh vực thương mại điện tử, mang lại những trải nghiệm mới và tiện ích cho khách hàng.

* 1. **Giới thiệu ngôn ngữ và công cụ:**

### **1.4.1 Ngôn ngữ 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. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

Đặc trưng của ngôn ngữ lập trình C#:

- C# là ngôn ngữ đơn giản

- C# là ngôn ngữ hiện đại

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

- C# là một ngôn ngữ ít từ khóa

### **1.4.2 SQL Server**

SQL Server (hay Microsoft SQL Server) là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft.

SQL Server cung cấp cho người dùng các công cụ và tính năng để quản lý, lưu trữ, xử lý các truy vấn dữ liệu, kiểm soát truy cập, xử lý giao dịch và hỗ trợ tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngoài ra, SQL Server cũng cung cấp các công cụ để tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL.

SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ. SQL Server được liên kết với Transact-SQL hoặc T-SQL, triển khai SQL của Microsoft có bổ sung một tập hợp các cấu trúc lập trình độc quyền.

SQL Server hoạt động độc quyền trên môi trường Windows trong hơn 20 năm. Vào năm 2016, Microsoft đã cung cấp SQL Server trên Linux. SQL Server 2017 ra mắt vào tháng 10 năm 2016 chạy được trên cả Windows và Linux.

SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL,... Cụ thể như sau:

* T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ,...
* SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

### **1.4.3 Visual Studio**

Visual Studio là một môi trường phát triển tích hợp (IDE) do Microsoft phát triển. Nó cung cấp một không gian làm việc toàn diện cho các lập trình viên, giúp họ viết, kiểm tra và triển khai các ứng dụng. Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript,... và được sử dụng rộng rãi để phát triển các ứng dụng Windows, ứng dụng web, ứng dụng di động và các dịch vụ đám mây.

Lợi ích việc sử dụng Visual Studio:

* Giao diện trực quan: Visual Studio có giao diện người dùng thân thiện, dễ sử dụng, giúp các lập trình viên tập trung vào việc viết mã.
* Tính năng phong phú:
* IntelliSense: Tự động hoàn thành mã, cung cấp gợi ý về các từ khóa, hàm, biến,... giúp tăng tốc độ viết mã.
* Debugger: Công cụ gỡ lỗi mạnh mẽ, giúp tìm và sửa lỗi trong mã một cách hiệu quả.
* Refactoring: Tổ chức lại mã một cách tự động, giúp cải thiện chất lượng mã.
* Git Integration: Tích hợp với hệ thống quản lý phiên bản Git, giúp quản lý mã nguồn một cách hiệu quả.
* Hỗ trợ đa nền tảng: Visual Studio có thể được sử dụng để phát triển ứng dụng cho nhiều nền tảng khác nhau như Windows, Android, iOS, web,...
* Cộng đồng lớn: Có một cộng đồng người dùng lớn và sôi động, giúp bạn dễ dàng tìm kiếm tài liệu, giải đáp thắc mắc và chia sẻ kinh nghiệm.

### **1.4.4 GUNA Framework**

**Guna Framework** là một thư viện giao diện người dùng (UI) phổ biến cho C# được xây dựng dựa trên WinForms. Nó cung cấp một bộ sưu tập phong phú các control và component, giúp các nhà phát triển tạo ra các ứng dụng Windows có giao diện hiện đại, chuyên nghiệp và hấp dẫn hơn.

**Tại sao nên sử dụng Guna Framework?**

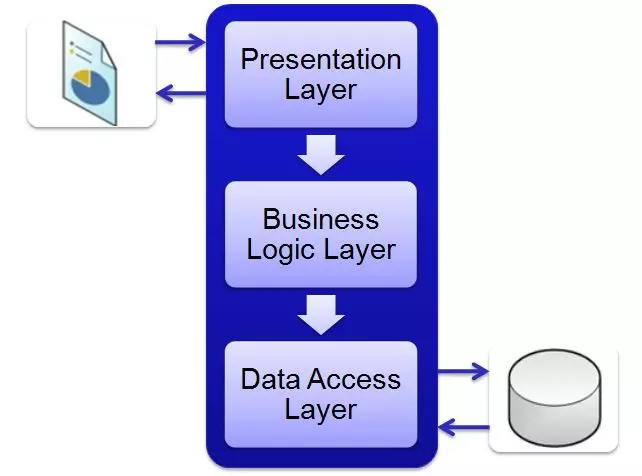
* **Giao diện đẹp mắt:** Guna Framework cung cấp nhiều styles, themes và hiệu ứng chuyển tiếp đẹp mắt, giúp bạn tạo ra các ứng dụng có giao diện bắt mắt, thu hút người dùng.
* **Dễ sử dụng:** Các control của Guna Framework được thiết kế để dễ sử dụng, bạn có thể kéo thả các control vào form và tùy chỉnh chúng một cách nhanh chóng.
* **Hiệu năng cao:** Thư viện được tối ưu hóa để hoạt động hiệu quả, đảm bảo ứng dụng của bạn chạy mượt mà.
* **Cộng đồng lớn:** Guna Framework có một cộng đồng người dùng lớn và sôi động, giúp bạn dễ dàng tìm kiếm tài liệu, giải đáp thắc mắc và chia sẻ kinh nghiệm.

**Các tính năng nổi bật của Guna Framework**

* **Hơn 50 control và component:** Bao gồm các control cơ bản như button, textbox, combobox, cho đến các control phức tạp hơn như chart, calendar, và nhiều hơn nữa.
* **16 loại biểu đồ:** Giúp bạn trực quan hóa dữ liệu một cách hiệu quả.
* **Cập nhật thường xuyên:** Thư viện được cập nhật liên tục với các tính năng mới và cải tiến.
* **Hiệu năng cao:** Được tối ưu hóa để hoạt động mượt mà trên các hệ thống khác nhau.
* **Dễ dàng tùy chỉnh:** Bạn có thể tùy chỉnh màu sắc, font chữ, kích thước và các thuộc tính khác của các control để phù hợp với thiết kế của ứng dụng.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **2.1 Kiến trúc hệ thống**

****

Gồm 3 lớp, đó là:

* GUI Layer: Lớp này là lớp hiển thị giao diện và các chức năng để người dùng cuối sử dụng.
* Business (BUS) Layer: Đây là lớp nhận các yêu cầu từ lớp GUI và truy xuất lên lớp Data để lấy thông tin và trả về GUI.
* Data Access Layer: Lớp này là lớp để truy xuất với CSDL, chỉ duy nhất lớp này được làm việc với database.
* DTO Layer: Lớp này chỉ là phụ, đây là lớp định nghĩa các table trong database, định nghĩa cột của nó cũng như để ta gán data khi query lấy dữ liệu.

**Lợi thế của mô hình 3 lớp**

* 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 Webfrom ) 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.

## **2.2 Danh sách tác nhân**

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả** |
| Nhân viên | Nhân viên phục vụ là những người trực tiếp tương tác với khách hàng, nhận đặt món và quản lý các bàn. |
| Admin | Admin (Quản trị viên) đóng vai trò quan trọng trong việc giám sát và quản lý toàn bộ hệ thống |

Nhân viên:

* Quản lí bàn : Quản lý bàn trong nhà hàng để tối ưu hóa việc sắp xếp chỗ ngồi, phục vụ khách hàng hiệu quả, và theo dõi trạng thái bàn trong suốt quá trình hoạt động.
* Quản lí hóa đơn: Quản lý nhà hàng có thể xem tổng doanh thu theo ngày để đánh giá tình hình kinh doanh.

Admin(Quản trị viên)

* Thêm, sửa, xóa tài khoản người dùng: Admin có thể tạo tài khoản cho các nhân viên mới và gán quyền cho từng tài khoản.
* Phân quyền và quản lý quyền truy cập: Xác định quyền truy cập của từng loại nhân viên, đảm bảo mỗi người chỉ có thể truy cập các phần chức năng liên quan đến công việc của họ.
* Quản lý món ăn: Admin có thể quản lý menu của quán, thêm mới hoặc cập nhật thông tin món ăn, thay đổi giá cả
* Quản lý danh mục món ăn: Tạo và quản lý các danh mục (như món khai vị, món chính, đồ uống), sắp xếp và tổ chức menu một cách hợp lý.
* Quản lí bàn : Quản lý bàn trong nhà hàng để tối ưu hóa việc sắp xếp chỗ ngồi, phục vụ khách hàng hiệu quả, và theo dõi trạng thái bàn trong suốt quá trình hoạt động.
* Quản lí hóa đơn: Quản lý nhà hàng có thể xem tổng doanh thu theo ngày để đánh giá tình hình kinh doanh.

## **2.3 Danh sách Use case**

|  |  |  |
| --- | --- | --- |
| **Mã số** | **Tên Usecase** | **Mô tả** |
| UC1 | Đăng nhập | Đăng nhập vào phần mềm quản lý nhà hàng |
| UC2 | Đăng xuất | Đăng xuất tài khoản |
| UC3 | Quản lí tài khoản | Quản lí tài khoản của nhân viên gồm xóa, sửa, thêm tài khoản. |
| UC4 | Quản lí danh mục món ăn | Quản lí các danh mục món ăn gồm xóa, sửa, thêm danh mục |
| UC5 | Quản lí thực đơn | Quản lí các món ăn trong nhà hàng gồm xóa, sửa, thêm món ăn |
| UC5 | Quản lí bàn | Quản lý các bàn ăn tại nhà hàng gồm các thao tác như thêm, sửa, xóa, đặt bàn |
| UC6 | Quản lí doanh thu | Quản lý nhà hàng có thể xem tổng doanh thu theo ngày để đánh giá tình hình kinh doanh. |
| UC7 | Thêm tài khoản | Cho phép Admin thêm tài khoản mới vào hệ thống để người dùng có thể đăng nhập và sử dụng các tính năng của hệ thống. |
| UC8 | Xóa tài khoản | Cho phép Admin xóa tài khoản hiện có trong hệ thống |
| UC9 | Sửa tài khoản | Cho phép Admin chỉnh sửa thông tin tài khoản hiện có trong hệ thống |
| UC10 | Thêm danh mục | Cho phép Admin thêm danh mục món ăn mới vào hệ thống. |
| UC11 | Xóa danh mục | Cho phép Admin xóa danh mục món ăn hiện có trong hệ thống |
| UC12 | Sửa danh mục | Cho phép Admin chỉnh sửa thông tin danh mục món ăn hiện có trong hệ thống |
| UC13 | Thêm món ăn | Cho phép Admin thêm món ăn mới vào hệ thống. |
| UC14 | Xóa món ăn | Cho phép Admin xóa món ăn hiện có trong hệ thống |
| UC15 | Sửa món ăn | Cho phép Admin chỉnh sửa thông tin món ăn hiện có trong hệ thống |
| UC16 | Đặt bàn | Cho phép nhân viên đặt bàn trong quán ăn thông qua ứng dụng, đảm bảo sắp xếp và quản lý chỗ ngồi hiệu quả. |
| UC17 | Chuyển bàn | Cho phép nhân viên hoặc quản lý chuyển khách hàng từ một bàn đã đặt hoặc đang sử dụng sang bàn khác trong cùng một phiên phục vụ. |
| UC18 | Xóa bàn | Cho phép nhân viên hoặc quản lý xóa bàn đã không còn sử dụng trên hệ thống |
| UC19 | Thêm bàn | Cho phép nhân viên hoặc quản lý thêm bàn mới vào hệ thống |
| UC20 | Cập nhật hóa đơn | Cho phép nhân viên hoặc quản lý cập nhật hóa đơn của khách hàng để điều chỉnh món ăn, số lượng, giá cả nhằm đảm bảo hóa đơn chính xác trước khi thanh toán. |
| UC21 | Thanh toán hóa đơn | Cho phép nhân viên hoặc quản lý xử lý việc thanh toán hóa đơn của khách hàng, bao gồm việc xác nhận thanh toán, chọn phương thức thanh toán, và cập nhật trạng thái hóa đơn sau khi hoàn tất. |
| UC22 | Tạo hóa đơn mới | Cho phép nhân viên hoặc quản lý tạo hóa đơn mới cho khách hàng khi bắt đầu phục vụ, ghi nhận các món ăn, đồ uống và dịch vụ được yêu cầu. |
| UC23 | Xem lịch sử hóa đơn | Cho phép nhân viên hoặc quản lý xem và truy xuất các hóa đơn đã được tạo trước đó, nhằm hỗ trợ việc theo dõi doanh thu, phân tích dữ liệu, và xử lý các vấn đề liên quan đến hóa đơn. |

Bảng 2.1 Danh sách các use case

## **2.4 Mô hình use case:**

### **2.4.1 Sơ đồ use-case chính:**

A diagram of a diagram

Description automatically generated

### **2.4.2 Sơ đồ UseCase Admin**

A diagram of a diagram

Description automatically generated

**2.4.3 Sơ đồ use case Nhân viên**A diagram of a person with text

Description automatically generated

## **2.5 Đặc tả chi tiết các Usecase(UC)**

2.5.1 Use Case Đăng nhập hệ thống

|  |  |
| --- | --- |
| Use Case | Đăng nhập hệ thống |
| Actor | Nhân viên, Admin |
| Brief Description | Use case mô tả các bước đăng nhập của người dùng vào hệ thống |
| Pre-Condition | Không có |
| Basic Flows | 1. Hệ thống yêu cầu người dùng nhập tên đăng nhập và mật khẩu 2. Người dùng nhập tên đăng nhập và mật khẩu của mình và nhấn nút đăng nhập 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống thông báo đăng nhập thành công và cho người dùng vào hệ thống, đồng thời phân quyền tùy theo vai trò |
| Alternative Flows | 3.1 Hệ thống kiểm tra thông tin đăng nhập sai và hiển thị thông báo ra màn hình |
| Post-conditions | Cho phép người dùng đăng nhập vào hệ thống |

Bảng 2.0 Use case đăng nhập hệ thống

2.5.2 Use Case Đăng xuất hệ thống

|  |  |
| --- | --- |
| Use Case | Đăng xuất hệ thống |
| Actor | Nhân viên, Admin |
| Brief Description | Cho phép người dùng đăng xuất khỏi hệ thống |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống |
| Basic Flows | Người dùng nhấn nút đóng form |
| Alternative Flows | Không có |
| Post-conditions | Không có |

Bảng 2.1 Use case đăng xuất hệ thống

2.5.3 Use Case Thêm tài khoản

|  |  |
| --- | --- |
| Use Case | Thêm tài khoản |
| Actor | Admin |
| Brief Description | Cho phép Admin thêm tài khoản mới vào hệ thống để người dùng có thể đăng nhập và sử dụng các tính năng của hệ thống |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase |
| Basic Flows | 1. Actor (Quản trị viên) chọn chức năng "Thêm tài khoản" 2. Hệ thống hiển thị form nhập thông tin tài khoản, bao gồm các trường cần thiết như: Tên tài khoản, Tên hiển thị, Loại tài khoản. 3. Actor điền đầy đủ thông tin và nhấn nút "Thêm" 4. Hệ thống kiểm tra tính hợp lệ của thông tin: 5. Nếu thông tin hợp lệ, hệ thống lưu thông tin tài khoản mới vào cơ sở dữ liệu. 6. Hệ thống thông báo thành công 7. Actor nhận thông báo tài khoản đã được thêm thành công. |
| Alternative Flows | 3.1 Actor điền thiếu thông tin |
| Post-conditions | Tài khoản mới được tạo và sẵn sàng cho người dùng đăng nhập. |

Bảng 2.2 Usse case thêm tài khoản

2.5.4 Use Case Xóa tài khoản

|  |  |
| --- | --- |
| Use Case | Xóa tài khoản |
| Actor | Admin |
| Brief Description | Cho phép Admin xóa tài khoản hiện có trong hệ thống |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase  Tài khoản đã tồn tại trong hệ thống. |
| Basic Flows | 1. Actor chọn chức năng "Xóa tài khoản" từ danh sách tài khoản 2. Actor chọn tài khoản cần xóa 3. Actor nhấn nút xóa |
| Alternative Flows | Không có |
| Post-conditions | Tài khoản đã được xóa và cập nhật trong hệ thống |

Bảng 2.3 Use case xóa tài khoản

2.5.5 Use Case Sửa tài khoản

|  |  |
| --- | --- |
| Use Case | Sửa tài khoản |
| Actor | Admin |
| Brief Description | Cho phép Admin chỉnh sửa thông tin tài khoản hiện có trong hệ thống  Tài khoản đã tồn tại trong hệ thống. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase |
| Basic Flows | 1. Actor chọn chức năng "Sửa tài khoản" từ danh sách tài khoản 2. Hệ thống hiển thị thông tin tài khoản hiện tại, bao gồm các trường như: Tên tài khoản, Tên hiển thị, Loại tài khoản.... 3. Actor cập nhật các thông tin cần thay đổi. 4. Actor nhấn nút "Lưu" hoặc "Cập nhật". 5. Hệ thống kiểm tra tính hợp lệ của thông tin cập nhật:  * Kiểm tra các trường bắt buộc có được điền đầy đủ không. * Kiểm tra định dạng các thông tin (ví dụ: email hợp lệ, số điện thoại hợp lệ). * Kiểm tra nếu có thay đổi mật khẩu, xác nhận mật khẩu phải khớp và đủ mạnh.  1. Nếu thông tin hợp lệ, hệ thống cập nhật thông tin tài khoản vào cơ sở dữ liệu. 2. Hệ thống thông báo thành công và cập nhật giao diện với thông tin mới. 3. Actor nhận được thông báo tài khoản đã được cập nhật thành công. |
| Alternative Flows | 5.1 Nếu thông tin nhập vào không hợp lệ hệ thống hiển thị thông báo lỗi tương ứng và yêu cầu actor chỉnh sửa lại thông tin. |
| Post-conditions | Thông tin tài khoản được cập nhật thành công trong hệ thống.  Actor nhận được xác nhận về việc cập nhật thành công. |

Bảng 2.4 Use case sửa tài khoản

2.5.6 Use Case Thêm danh mục món ăn

|  |  |
| --- | --- |
| Use Case | Thêm danh mục món ăn |
| Actor | Admin, Nhân viên |
| Brief Description | Cho phép nhân viên quản lý hoặc quản lý quán ăn thêm danh mục mới (như món khai vị, món chính, đồ uống, tráng miệng, v.v.) vào hệ thống, giúp tổ chức thực đơn một cách logic và dễ dàng tìm kiếm. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase |
| Basic Flows | 1. Nhân viên quản lý/Quản lý chọn chức năng "Thêm danh mục" từ giao diện quản lý danh mục trong ứng dụng. 2. Hệ thống hiển thị form để nhập thông tin danh mục mới, bao gồm:  * Tên danh mục (bắt buộc, ví dụ: Món khai vị, Món chính, Đồ uống). * Hình ảnh đại diện cho danh mục (nếu có).      1. Nhân viên quản lý/Quản lý nhập thông tin cần thiết vào form. 2. Nhân viên quản lý/Quản lý nhấn nút "Lưu" hoặc "Thêm danh mục". 3. Hệ thống kiểm tra các thông tin nhập vào: 4. Đảm bảo tên danh mục là duy nhất (không trùng lặp với các danh mục đã có). 5. Kiểm tra định dạng của các thông tin (ví dụ: tên danh mục không để trống, hình ảnh đúng định dạng). 6. Nếu thông tin hợp lệ, hệ thống thêm danh mục mới vào cơ sở dữ liệu. 7. Hệ thống hiển thị thông báo thêm danh mục thành công và cập nhật danh sách các danh mục hiện có. 8. Nhân viên quản lý/Quản lý có thể xem danh mục mới trên giao diện hoặc quay lại để tiếp tục thêm các danh mục khác. |
| Alternative Flows | 5.1 Nếu thông tin nhập không hợp lệ (ví dụ: tên danh mục bị trùng, thông tin bắt buộc còn thiếu), hệ thống hiển thị thông báo lỗi và yêu cầu chỉnh sửa lại thông tin. |
| Post-conditions | * Danh mục mới được thêm vào hệ thống và sẵn sàng để sử dụng trong việc quản lý thực đơn. * Hệ thống cập nhật danh sách danh mục để nhân viên hoặc quản lý có thể sử dụng ngay lập tức. |

Bảng 2.5 Use case thêm danh mục món ăn

2.5.7 Use Case Xóa danh mục món ăn

|  |  |
| --- | --- |
| Use Case | Xóa danh mục món ăn |
| Actor | Admin, Nhân viên |
| Brief Description | Cho phép nhân viên quản lý hoặc quản lý quán ăn xóa các danh mục không còn cần thiết khỏi hệ thống để đảm bảo thực đơn được tổ chức gọn gàng và chỉ hiển thị các mục hiện hành. |
| Pre-Condition | Nhân viên quản lý hoặc quản lý đã đăng nhập vào hệ thống và có quyền truy cập để xóa danh mục.  Hệ thống đã có sẵn các danh mục cần quản lý và cho phép thao tác xóa. |
| Basic Flows | 1. Nhân viên quản lý/Quản lý chọn chức năng "Quản lý danh mục" từ giao diện chính của ứng dụng. 2. Hệ thống hiển thị danh sách các danh mục hiện có trong hệ thống, bao gồm tên danh mục, mô tả 3. Nhân viên quản lý/Quản lý chọn danh mục cần xóa. 4. Nhân viên quản lý/Quản lý nhấn nút "Xóa" bên cạnh danh mục đã chọn. 5. Hệ thống cập nhật danh sách các danh mục và hiển thị thông báo "Xóa danh mục thành công". 6. Nhân viên quản lý/Quản lý tiếp tục với các thao tác khác hoặc thoát khỏi chức năng. |
| Alternative Flows |  |
| Post-conditions | Danh mục được xóa khỏi hệ thống và không còn hiển thị trong danh sách danh mục. |

Bảng 2.6 Use case xóa danh mục món ăn

2.5.8 Use Case Sửa danh mục món ăn

|  |  |
| --- | --- |
| Use Case | Sửa danh mục món ăn |
| Actor | Admin, Nhân viên |
| Brief Description | Cho phép nhân viên quản lý hoặc quản lý quán ăn chỉnh sửa các danh mục (như tên, mô tả) để đảm bảo các danh mục trong thực đơn luôn chính xác và cập nhật theo nhu cầu kinh doanh |
| Pre-Condition | Nhân viên quản lý hoặc quản lý đã đăng nhập vào hệ thống và có quyền truy cập để xóa danh mục.  Hệ thống đã có sẵn các danh mục cần quản lý và cho phép thao tác sửa. |
| Basic Flows | 1. Nhân viên quản lý/Quản lý chọn chức năng "Quản lý danh mục" từ giao diện chính của ứng dụng. 2. Hệ thống hiển thị danh sách các danh mục hiện có, bao gồm các thông tin như tên danh mục, mô tả 3. Nhân viên quản lý/Quản lý chọn danh mục cần sửa và nhấn vào nút "Sửa" bên cạnh danh mục đó. 4. Hệ thống hiển thị form chỉnh sửa với các thông tin hiện tại của danh mục được điền sẵn:  * Tên danh mục. * Mô tả danh mục.      1. Nhân viên quản lý/Quản lý chỉnh sửa các thông tin cần thay đổi, ví dụ:  * Thay đổi tên danh mục. * Cập nhật hoặc thay đổi mô tả.      1. Nhân viên quản lý/Quản lý nhấn nút "Lưu" hoặc "Cập nhật" để lưu các thay đổi. 2. Hệ thống kiểm tra tính hợp lệ của các thông tin mới nhập: 3. Đảm bảo tên danh mục là duy nhất và không để trống. 4. Kiểm tra định dạng và kích thước của hình ảnh (nếu có). 5. Nếu thông tin hợp lệ, hệ thống cập nhật danh mục trong cơ sở dữ liệu. 6. Hệ thống hiển thị thông báo sửa danh mục thành công và cập nhật danh sách các danh mục. 7. Nhân viên quản lý/Quản lý tiếp tục với các thao tác khác hoặc thoát khỏi chức năng. |
| Alternative Flows | 7.1 Thông tin không hợp lệ  Nếu thông tin mới nhập không hợp lệ (ví dụ: tên danh mục bị trùng, thông tin bắt buộc còn thiếu), hệ thống hiển thị thông báo lỗi và yêu cầu nhân viên quản lý/Quản lý chỉnh sửa lại thông tin.  6.1 Hủy bỏ thao tác sửa  Nếu Nhân viên quản lý/Quản lý chọn "Hủy" thay vì "Lưu", hệ thống hủy bỏ thao tác sửa và quay lại màn hình quản lý danh mục mà không lưu các thay đổi. |
| Post-conditions | Danh mục được cập nhật với các thông tin mới trong hệ thống.  Các thay đổi được phản ánh ngay lập tức trên giao diện danh sách danh mục và có hiệu lực trong việc quản lý thực đơn. |

Bảng 2.7 Use case xóa danh mục món ăn

2.5.9 Use Case Tạo hóa đơn mới

|  |  |
| --- | --- |
| Use Case | Tạo hóa đơn mới |
| Actor | Nhân viên, Admin |
| Brief Description | Cho phép nhân viên hoặc quản lý tạo hóa đơn mới cho khách hàng khi bắt đầu phục vụ, ghi nhận các món ăn, đồ uống và dịch vụ được yêu cầu. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase |
| Basic Flows | 1. Nhân viên/Quản lý chọn chức năng "Tạo hóa đơn mới" trên ứng dụng quản lý quán ăn. 2. Nhân viên/Quản lý chọn bàn nhập thông tin về các món ăn/đồ uống mà khách hàng đã gọi:  * Chọn món từ thực đơn. * Nhập số lượng.  1. Hệ thống tự động tính tổng tiền tạm thời dựa trên các món đã chọn và hiển thị thông tin này trong hóa đơn. 2. Nhân viên/Quản lý có thể tiếp tục thêm, sửa, hoặc xóa các món ăn trong hóa đơn khi khách hàng gọi thêm món hoặc thay đổi. 3. Sau khi nhập xong, Nhân viên/Quản lý xác nhận việc tạo hóa đơn. 4. Hệ thống lưu hóa đơn vào cơ sở dữ liệu với trạng thái "Đang xử lý" hoặc "Đang phục vụ". |
| Alternative Flows | 4.1 Không có món ăn nào được chọn   * Nếu không có món nào được chọn, hệ thống sẽ nhắc nhở nhân viên/ quản lý thêm món trước khi lưu hóa đơn. |
| Post-conditions | Hóa đơn mới được tạo và lưu trong hệ thống với trạng thái ban đầu.  Hóa đơn có thể được tiếp tục cập nhật hoặc thanh toán tùy theo tiến trình phục vụ. |

Bảng 2.8 Use case tạo hóa đơn

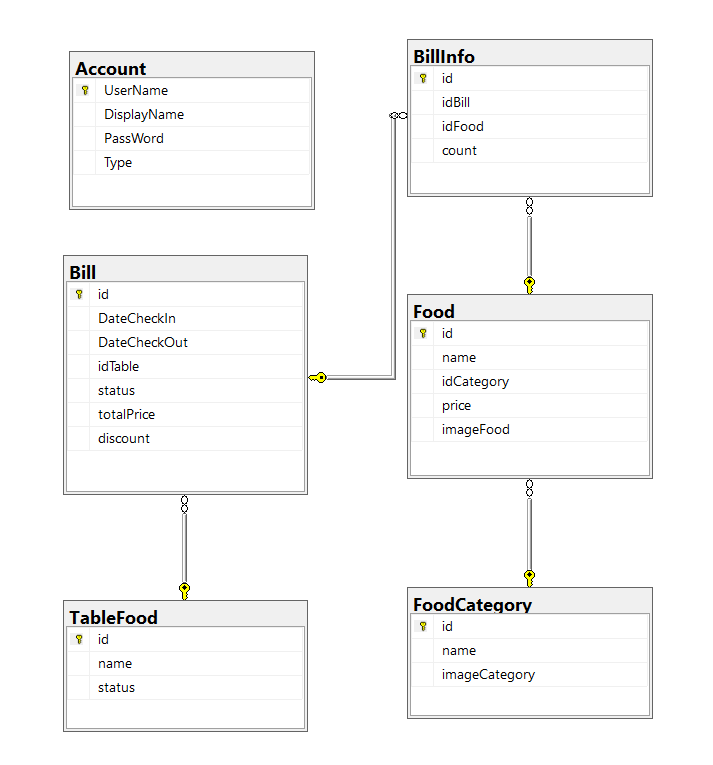
2.5.10 Use Case Xem lịch sử hóa đơn

|  |  |
| --- | --- |
| Use Case | Xem lịch sử hóa đơn |
| Actor | Nhân viên, Admin |
| Brief Description | Cho phép nhân viên hoặc quản lý xem và truy xuất các hóa đơn đã được tạo trước đó, nhằm hỗ trợ việc theo dõi doanh thu, phân tích dữ liệu, và xử lý các vấn đề liên quan đến hóa đơn. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập UseCase |
| Basic Flows | 1. Nhân viên/Quản lý chọn chức năng "Xem lịch sử hóa đơn" từ giao diện quản lý trong ứng dụng. 2. Hệ thống yêu cầu nhân viên/Quản lý nhập tiêu chí ngày bắt đầu và ngày kết thúc để tìm kiếm để lọc lịch sử hóa đơn        1. Nhân viên/Quản lý nhập các tiêu chí tìm kiếm và nhấn nút "Tìm kiếm" 2. Hệ thống hiển thị danh sách các hóa đơn phù hợp với tiêu chí tìm kiếm, bao gồm thông tin chi tiết như:  * Tên bàn. * Tổng tiền. * Giảm giá  1. Nhân viên/Quản lý có thể xuất báo cáo hoặc in hóa đơn chi tiết nếu cần. |
| Alternative Flows | 3.1 Không tìm thấy hóa đơn :   * Nếu không có hóa đơn nào phù hợp với tiêu chí tìm kiếm, hệ thống hiển thị thông báo không có kết quả |
| Post-conditions | Nhân viên/Quản lý có thể xem và phân tích lịch sử hóa đơn để theo dõi các giao dịch và thực hiện các hoạt động liên quan đến quản lý hóa đơn. |

Bảng 2.9 Use case xem lịch sử hóa

# **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU**

* 1. **Biểu đồ diagram:**

****

* 1. **Bảng Account:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường thông tin |  | Cụ thể | Ghi chú |
| UserName |  | Tên đăng nhập | Khóa |
| DisplayName |  | Tên hiển thị |  |
| Password |  | Mật khẩu |  |
| Type |  | Loại tài khoản |  |

Bảng 3.1 Bảng Account

* 1. **Bảng Food:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thức ăn, thức uống | Khóa |
| name | Tên thức ăn, thức uống |  |
| idCategory | Mã loại thức ăn, thức uống |  |
| price | Giá |  |
| imageFood | Hình ảnh của thức ăn | Tạm thời để trống |

Bảng 3.2 Bảng Food

* 1. **Bảng Food Category:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã loại thức ăn, thức uống | Khóa |
| name | Tên loại thức ăn, thức uống |  |
| imageCategory | Hình ảnh danh mục thức ăn | Tạm thời để trống |

Bảng 3.3 Bảng Food category

* 1. **Bảng Bill:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã hóa đơn | Khóa |
| DateCheckIn | Thời gian vào quán |  |
| DateCheckOut | Thời gian ra khỏi quán |  |
| idTable | Mã bàn |  |
| status | Trạng thái |  |
| discount | Giảm giá |  |
| totalPrice | Tổng tiền |  |

Bảng 3.4 Bảng Bill

* 1. **Bảng Bill Info:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thông tin hóa đơn | Khóa |
| idBill | Mã hóa đơn |  |
| idFood | Mã thức ăn, thức uống |  |
| count | Số lượng |  |

Bảng 3.5 Bảng Bill Info

* 1. **Bảng Table Food:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã bàn | Khóa |
| name | Tên bàn |  |
| status | Trạng thái  (Trống hoặc Có người) |  |

Bảng 3.6 Bảng Table Food

* 1. **Các thủ tục (Procedure) và hàm (Function):**
* **Procedure:**

1. StoredProcedure [dbo].[HoaDonTTBan]

Thủ tục lấy hóa đơn theo bàn

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[HoaDonTTBan]

(

    @name nvarchar(100)

)

AS

    SET NOCOUNT ON;

SELECT f.name, bi.count, f.price, f.price \* bi.count AS totalPrice

FROM BillInfo AS bi INNER JOIN

     Bill AS b ON bi.idBill = b.id INNER JOIN

     Food AS f ON bi.idFood = f.id INNER JOIN

     TableFood AS t ON b.idTable = t.id

WHERE (b.status = 0) AND (t.name = @name)

GO

1. StoredProcedure [dbo].[USP\_GetAccountByUserName]

Thủ tục lấy tài khoản theo username

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--Lấy tài khoản từ user name

CREATE PROC [dbo].[USP\_GetAccountByUserName]

@userName nvarchar(100)

AS

BEGIN

    SELECT \* FROM dbo.Account WHERE UserName = @userName

END

GO

1. StoredProcedure [dbo].[USP\_GetFoodList]

Thủ tục lấy danh sách thức ăn

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--Lấy danh sách món ăn

CREATE PROC [dbo].[USP\_GetFoodList]

AS SELECT \* FROM dbo.Food

GO

1. StoredProcedure [dbo].[USP\_GetListBillByDate]

Thủ tục lấy danh sách Bill theo ngày

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--Lấy danh sách hóa đơn theo thời gian

CREATE PROC [dbo].[USP\_GetListBillByDate]

@checkIn date, @checkOut date

AS

BEGIN

    SELECT t.name AS [Tenban], b.totalPrice AS [Tongtien], DateCheckIn AS [Ngayvao], DateCheckOut AS [Ngayra], discount AS [Giamgia]

    FROM dbo.Bill AS b,dbo.TableFood AS t

    WHERE DateCheckIn >= @checkIn AND DateCheckOut <= @checkOut AND b.status = 1

    AND t.id = b.idTable

END

GO

1. StoredProcedure [dbo].[USP\_GetListBillByTableName]

Lấy danh sách hóa đơn từ tên bàn

1. StoredProcedure [dbo].[USP\_GetTableList]

Lấy danh sách bàn

1. StoredProcedure [dbo].[USP\_InsertBill]

Thêm hóa đơn

1. StoredProcedure [dbo].[USP\_InsertBillInfo]

Thêm chi tiết hóa đơn

1. StoredProcedure [dbo].[USP\_Login]

Thủ tục đăng nhập

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROC [dbo].[USP\_Login]

@userName nvarchar(100), @passWord nvarchar(100)

AS

BEGIN

    SELECT \* FROM dbo.Account WHERE UserName = @userName AND PassWord = @passWord

END

GO

1. StoredProcedure [dbo].[USP\_SwitchTabel]

Thủ tục chuyển bàn

1. StoredProcedure [dbo].[USP\_UpdateAccount]

Thu tục cập nhật thông tin tài khoản

* **Function:**

Tìm món ăn theo tên

CREATE FUNCTION [dbo].[fuConvertToUnsign1] ( @strInput NVARCHAR(4000) ) RETURNS NVARCHAR(4000)

AS

BEGIN

IF @strInput IS NULL

RETURN @strInput

IF @strInput = ''

RETURN @strInput

DECLARE @RT NVARCHAR(4000)

DECLARE @SIGN\_CHARS NCHAR(136)

DECLARE @UNSIGN\_CHARS NCHAR (136)

SET @SIGN\_CHARS = N'ăâđêôơưàảãạáằẳẵặắầẩẫậấèẻẽẹéềểễệế ìỉĩịíòỏõọóồổỗộốờởỡợớùủũụúừửữựứỳỷỹỵý ĂÂĐÊÔƠƯÀẢÃẠÁẰẲẴẶẮẦẨẪẬẤÈẺẼẸÉỀỂỄỆẾÌỈĨỊÍ ÒỎÕỌÓỒỔỖỘỐỜỞỠỢỚÙỦŨỤÚỪỬỮỰỨỲỶỸỴÝ' +NCHAR(272)+ NCHAR(208)

SET @UNSIGN\_CHARS = N'aadeoouaaaaaaaaaaaaaaaeeeeeeeeee iiiiiooooooooooooooouuuuuuuuuuyyyyy AADEOOUAAAAAAAAAAAAAAAEEEEEEEEEEIIIII OOOOOOOOOOOOOOOUUUUUUUUUUYYYYYDD' DECLARE @COUNTER int

DECLARE @COUNTER1 int

SET @COUNTER = 1

WHILE (@COUNTER <=LEN(@strInput))

BEGIN

SET @COUNTER1 = 1 WHILE (@COUNTER1 <=LEN(@SIGN\_CHARS)+1)

BEGIN

IF UNICODE(SUBSTRING(@SIGN\_CHARS, @COUNTER1,1)) = UNICODE(SUBSTRING(@strInput,@COUNTER ,1) )

BEGIN

IF @COUNTER=1 SET @strInput = SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)-1)

ELSE

SET @strInput = SUBSTRING(@strInput, 1, @COUNTER-1) +SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)- @COUNTER)

BREAK END

SET @COUNTER1 = @COUNTER1 +1

END

SET @COUNTER = @COUNTER +1

END

SET @strInput = replace(@strInput,' ','-') RETURN @strInput

END

GO

# **CHƯƠNG 4: CÀI ĐẶT VÀ THỰC NGHIỆM**

* 1. **Giao diện bằng Guna UI:**
  2. **Giao diện đăng nhập:**

**A screenshot of a computer screen

Description automatically generated**

* 1. **Giao diện quản lý:**

**A screenshot of a computer

Description automatically generated**

* 1. **Giao diện nhân viên:**

**A screenshot of a computer

Description automatically generated**

* 1. **Giao diện đổi mật khẩu:**

**A screenshot of a computer

Description automatically generated**

* 1. **Cấu trúc:**

Đầu tiên là FORM GUI gồm các FORM người dùng sẽ giao tiếp với màn hình giao diện này.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

Các nghiệp vụ xử lý chính sẽ được đặt ở lớp DAO gồm các nghiệp vụ insert, update, delete, …., và truy vấn đến csdl.

A screenshot of a computer

Description automatically generated

Cúi cùng là thư mục Report chứ các Report Vierwer:

A screenshot of a computer

Description automatically generated

# **KẾT LUẬN**

Ứng dụng quản lý nhà hàng được phát triển bằng C# đã đáp ứng được các yêu cầu cơ bản của một hệ thống quản lý nhà hàng. Với các tính năng quản lý thực đơn, đơn hàng, thanh toán và báo cáo doanh thu, ứng dụng giúp cải thiện hiệu quả quản lý và điều hành nhà hàng.

Đề xuất:

Mở rộng tính năng: Có thể mở rộng thêm các tính năng như quản lý kho và lịch đặt bàn để đáp ứng nhu cầu của các nhà hàng lớn hơn.

Cải thiện giao diện: Tiếp tục cải thiện giao diện người dùng để tăng cường trải nghiệm người dùng và làm cho ứng dụng trở nên thân thiện hơn.

Tích hợp các công nghệ mới: Xem xét tích hợp với các công nghệ mới như mobile app hoặc cloud-based solutions để mở rộng khả năng và tiện ích của ứng dụng.

Nhìn chung, dự án đã hoàn thành thành công và đáp ứng được các mục tiêu đề ra, mang lại giải pháp hiệu quả cho việc quản lý nhà hàng.

# **TÀI LIỆU THAM KHẢO**

* Youtube
* <https://viblo.asia/p/gioi-thieu-mo-hinh-3-lop-trong-c-gDVK2Q9w5Lj>
* <https://codegym.vn/blog/c-la-gi-tim-hieu-ve-ngon-ngu-lap-trinh-c/>
* <https://topdev.vn/blog/sql-server-la-gi>