|  |  |
| --- | --- |
| logomobile.png | TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM  **KHOA CÔNG NGHỆ THÔNG TIN**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |

**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS**

Đề tài:

**QUẢN LÝ CỦA HÀNG POPEYES**

|  |  |  |
| --- | --- | --- |
| **Giáo viên hướng dẫn:** | ThS. Dương Thành Phết | |
| **Sinh viên thực hiện:** |  | |
| 1. Trịnh Quốc Việt | 18DTHA2 | 1811060885 |
| 1. Phạm Văn Truyền | 15DTH12 | 1511061489 |
| 1. Nguyễn Trịnh Thành Phát | 18DTHA2 | 1811060563 |

**TP. Hồ Chí Minh, 11/2020**

**NHẬN XÉT**

( của giáo viên hướng dẫn)

**MỤC LỤC**

**Đề mục Trang**

MỤC LỤC……..………………………………………….………………………………1

LỜI NÓI ĐẦU…..….……………………………………..………………………..……..3

**CHƯƠNG 1. TỔNG QUAN……..………..………..……………………..…………….4**

* 1. Đặt vấn đề……………………………………………………….……………....4
  2. Cơ sở lý luận…………………………………………………………………….4

1.2.1 Tổng quan ngôn ngữ C#................................................................................4

1.2.2 Đặc trưng của C#...........................................................................................5

1.2.3 Mô hình 3 lớp của C#....................................................................................7

1.2.4 Entity Framework của C#..............................................................................7

1.3.Tổng quan về SQL Server………………………………………………………..8

1.4.Mô tả hệ thống.………………………………………….………………………..9

**CHƯƠNG 2. CÁC LOẠI LƯỢC ĐỒ VÀ MÔ TẢ HÌNH THỨC VẬN HÀNH........12**

2.1. Sơ đồ USECASE.…,,…………………………….……………..……….……...12

2.1.1. Danh sách mô tả tác nhân.………………………………….………..…...12

2.1.2. Danh sách và mô tả usecase trong thực tế……………………..…………12

2.1.3. Sơ đồ usecase tổng quát…………………………………………..………13

2.1.4. Sơ đồ usecase đăng nhập.…………………………………………..…….13

2.1.5. Sơ đồ usecase đặt món.…………………………………………..…..…...14

2.2. Sơ đồ lớp(Class).………………………………………….…..……….………..14

2.3. Sơ đồ tuần tự.……………………………………………………............……...15

2.4. Sơ đồ Activity.…………………………………..…………..………………….17

2.4.1. Sơ đồ xử lý đăng nhập.………………………………………….………17

2.4.2. Sơ đồ xử lý hoá đơn.………………………………….………………....18

2.4.3. Sơ đồ quản lý món ăn.……………………………….………………….19

2.4.4. Sơ đồ quản lý nhân viên.………………………….……………………..20

2.5. Phân tích sơ đồ USECASE.………..……..…………….………...……………....20

2.5.1. Đăng nhập.…………...………………………….………………..…………….20

2.5.2. Đăng xuất.……………………………………………..….…………………….21

2.5.3. Quản lý nhân viên.……………………………………………………..……….22

2.5.4. Đặt món ăn.……………….……………………………………………..……...23

2.5.5. Quản lý nhà cung cấp(Đang phát triển)………………………………..……….24

2.5.6. Lập hoá đơn………….…………………………………………………..……..25

2.5.7. Doanh thu……………….………………………………………………..……..26

**CHƯƠNG 3. THUỘC TÍNH VÀ KIỂU DỮ LIỆU……………...…………………...27**

**CHƯƠNG 4. GIAO DIỆN, MENU CHƯƠNG TRÌNH VÀ MỤC TIÊU.……...…...30**

4.1. Giao diện, menu chương trình:……………………………………....………….30

4.1.1. Giao diện đăng nhập hệ thống:.…………………..….…….……..………30

4.1.2. Giao diện đặt bàn:.………………………………..…………..…………..30

4.1.3. Giao diện đặt món:………….…………………..……….……..…………31

4.1.4. Giao diện hoá đơn demo:……….…………………..…………..………...32

4.1.5. Giao diện thông tin cá nhân:..………………….……..………..….……...32

4.1.6. Giao diện doanh thu:..………………………….…………….….………..33

4.1.7. Giao diện chi tiết hoá đơn:…………………...…………………..……….34

4.1.8. Giao diện báo cáo tổng quát doanh thu…………….………….…...……..35

4.1.9. Giao diện tác vụ món ăn:...……………………………………..…..…….36

4.1.10. Giao diện tác vụ danh mục:...……………………...…...……….…...….36

4.1.11. Giao diện tác vụ bàn ăn:...………………………………….…………...37

4.1.12. Giao diện tác vụ tài khoản………………..………………….….………37

4.1.13. Giao diện báo cáo tài khoản:………………………..…………………..38

4.1.14. Giao diện trợ giúp (File help): ………………………..………………...38

**CHƯƠNG 5. Kết luận, mục tiêu mở rộng của đồ án……………………..……...…...39**

5.1. Mục tiêu mở rộng của đồ án……….……………………………………………39

5.2. Kết luận…………………...……….……………………………………………39

5.3. Các nguồn tài liệu tham khảo..…….……………………………………………39

**LỜI MỞ ĐẦU**

Việc ứng dụng Công Nghệ Thông Tin vào đời sống ngày nay đã không còn xa lạ với chúng ta. Công Nghệ Thông Tin với tốc độ nhanh chóng, chính xác đã giúp con người trong rất nhiều lĩnh vực, nhất là kinh doanh,…Trong đó kinh doanh thức ăn nhanh cũng không ngoại lệ.

Ngày nay, các hệ thống cửa hàng thức ăn nhanh đã trở nên khá phổ biến tại TP HCM. Cửa hàng thức ăn nhanh xuất hiện ồ ạt khiến cho thói quen tiêu dùng của người dân thành phố có sự thay đổi. Đồng thời góp phần làm sạch lòng lề đường ,khiến cho những quán ăn ven đường bị giới hạn khu vực họat động. Sản phẩm thức ăn nhanh POPEYES mang tiêu chí là vệ sinh đảm bảo, gà sạch,thức ăn phong phú đầy đủ chất dinh dưỡng… đã tạo nên chỗ đứng của POPEYES tại Việt Nam.

Một cửa hàng thức ăn nhanh muốn có tên tuổi và doanh thu cao thì phải thu hút được khách hàng , do vậy, nhu cầu về một phần mềm có khả năng quản lý nhanh và chặt chẽ về mọi mặt như quản lý thu chi,quản lý kho,chính sách khuyến mãi … . Để giúp cho hệ thống cửa hàng thức ăn nhanh trở nên gần gũi và tạo nhiều thuận lợi nhất cho người sử dụng , tiết kiệm thới gian giao dịch đồng thời tạo không gian ăn uống thoải mái nhất dành người tiêu dùng khi đến và sử dụng dịch vụ ngay tại cửa hàng. Thấu hiểu được vấn đề nên chúng em đã thực hiện ý tưởng ấp ủ đã lâu, đó chính là một phần mềm quản lý nhà hàng và quán ăn dành riêng cho cửa hàng thức ăn nhanh POPEYES Dưới sự hướng dẫn của thầy **Dương Thành Phết**. Tuy chưa hoàn hảo về mọi mặt nhưng phầm mềm sẽ giúp chúng em có thể tiếp thu những kiến thức tạo ra phần mềm còn khác mới mẻ cũng là nền tảng để có thể phát triển và tạo ra những phần mềm có chất lượng cao và đẹp mắt.

Để có thể tạo ra phần mềm ngoài nững kiến thức cơ bản về ngôn ngữ c# thì chúng em đã áp dụng khá nhiều những kiến thức mới mẻ và tiên tiến hiện nay ví dụ như: ADO.NET Entity models , Devexpress, Crystal reports, File hep, Mô hình 3 lớp,… đồng thời cũng đặt mình vào vai trò người sử dụng để có thể kiểm thử nhiều lần nhằm mang lại chất lượng cao nhất khi mang vào môi trường thực tế. Để hiểu rõ cách hoạt động và các chức năng đã thiết lập thì hôm nay chúng em xin trình bày báo cáo đồ án “ Quản lý cửa hàng gà rán POPEYES”.

**CHƯƠNG 1. TỔNG QUAN**

## Đặt vấn đề:

- Các hoạt động quản lý trong một cửa hàng nói riêng hay công ty nói chung là một chuỗi công việc rất vất vả và tốn nhiều thời gian, công sức. Nếu không có sự cần mẫn, chăm chỉ và sáng suốt thì những sai sót là không tránh khỏi. Phần mềm quản lý cửa hàng sẽ giúp cho quá trình mua bán diễn ra mau lẹ, hợp lý và tránh được những sai sót không đáng có nêu trên.

* 1. ***Cơ sở lý luận****:*

*1.2.1 Ngôn ngữ C#*

**- C#** (**C Sharp**) là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) đa năng vô cùng mạnh mẽ được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft" \o "Microsoft), C# là phần khởi đầu cho kế hoạch [.NET](https://vi.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework) của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo [ECMA](https://vi.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1" \o "ECMA (trang chưa được viết)) là **C#**, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên [C++](https://vi.wikipedia.org/wiki/C%2B%2B" \o "C++) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic" \o "Visual Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

- C# được thiết kế chủ yếu bởi [Anders Hejlsberg](https://vi.wikipedia.org/wiki/Anders_Hejlsberg" \o "Anders Hejlsberg) kiến trúc sư phần mềm nổi tiếng với các sản phẩm [Turbo Pascal](https://vi.wikipedia.org/wiki/Turbo_Pascal" \o "Turbo Pascal), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), J++, WFC. Phiên bản gần đây nhất là 9.0, được phát hành vào năm 2020 cùng với [Visual Studio](https://vi.wikipedia.org/wiki/Visual_Studio" \o "Visual Studio) 2019 phiên bản 16.8.[[14]](https://vi.wikipedia.org/wiki/C_Sharp_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \l "cite_note-new_features_in_8-15)

- Tiêu chuẩn ECMA liệt kê các mục tiêu của việc thiết kế ngôn ngữ C#:

- Ngôn ngữ được dự định là một ngôn ngữ lập trình đơn giản, hiện đại, hướng đến nhiều mục đích sử dụng, và là một ngôn ngữ [lập trình hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng).

- Ngôn ngữ và việc triển khai đáp ứng các nguyên tắc của ngành kỹ thuật phần mềm như kiểm tra chặt chẽ kiểu dữ liệu, kiểm tra giới hạn mảng, phát hiện các trường hợp sử dụng các biến chưa có dữ liệu, và [tự động thu gom rác](https://vi.wikipedia.org/wiki/Thu_gom_r%C3%A1c_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)" \o "Thu gom rác (khoa học máy tính)). Tính mạnh mẽ, sự bền bỉ, và năng suất của việc lập trình là rất quan trọng đối với ngôn ngữ này.

- Ngôn ngữ sẽ được sử dụng để phát triển các thành phần của phần mềm theo hướng thích hợp cho việc triển khai trong các môi trường phân tán.

- Khả năng di chuyển (portability) là rất quan trọng, đặc biệt là đối với những lập trình viên đã quen với C và C++.

- Hỗ trợ [quốc tế hóa](https://vi.wikipedia.org/wiki/Qu%E1%BB%91c_t%E1%BA%BF_h%C3%B3a_v%C3%A0_%C4%91%E1%BB%8Ba_ph%C6%B0%C6%A1ng_h%C3%B3a) (i18n).

- Ngôn ngữ sẽ được thiết kế để phù hợp với việc viết các ứng dụng cho cả hai hệ thống: hosted và [nhúng](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_nh%C3%BAng" \o "Hệ thống nhúng), từ các phần mềm quy mô lớn, đến các phần mềm chỉ có các chức năng đơn giản.

- Mặc dù các ứng dụng C# có tính kinh tế đối với các yêu cầu về bộ nhớ và chế độ xử lý, ngôn ngữ này không cạnh tranh trực tiếp về hiệu năng và kích thước đối với ngôn ngữ [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) hoặc [assembly](https://vi.wikipedia.org/wiki/H%E1%BB%A3p_ng%E1%BB%AF).

Ứng dụng của C# :

* Phát triển web (ASP.NET MVC, ASP.NET core, Blazor...)
* Phát triển desktop app (Winform, WPF, UWP, Mono, Uno..)
* Phát triển game ([Unity](https://en.wikipedia.org/wiki/Unity_(game_engine)" \o "en:Unity (game engine)), Monogame, Godot, Stride..)
* Phát triển mobile app, IOS native, Android native (Xamarin)
* Phát triển đám mây ([Azure](https://azure.microsoft.com/en-us/))
* Học máy và trí tuệ nhân tạo ([ML.Net](https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet), TensorFlow, csiSharp..)
* Phát triển thực tế ảo (VR) và thực tế tăng cường(AR) (HoloLens)
* Blockchain ([NEO, Stratis](https://www.stratisplatform.com/) )
* Microservices and containers

- Internet of thing (IoT)

*1.2.2 Đặc trưng của C#*

* C# là ngôn ngữ đơn giản, mạnh mẽ
* C# được dựng trên nền tảng C++ và Java, ảnh hưởng bởi Delphi, VisualBasic nên ngôn ngữ C# được thừa hưởng các ưu điểm vào loại bỏ các yếu điểm của các ngôn ngữ trên, vì vậy nó khá đơn giản, đồng thời loại bỏ các cú pháp dư thừa và thêm vào đó các cú pháp cải tiến hơn
* C# là [ngôn ngữ lâp trình bậc cao](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh_b%E1%BA%ADc_cao" \l ":~:text=Trong_khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh,chi_ti%E1%BA%BFt_c%E1%BB%A7a_m%C3%A1y_t%C3%ADnh." \o "Ngôn ngữ lập trình bậc cao), [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) vì vậy nó **dễ dàng tiếp cận** và **phù hợp cho người mới bắt đầu học**, ví dụ câu lệnh kinh điển dành cho người mới băt đầu hoc là in ra dòng chữ "Hello world", với C# ta chỉ cẩn 1 câu lệnh: **System.Console.WriteLine("Hello world");**
* C# là ngôn ngữ đa năng và hiện đại
* C# phù hợp cho việc phát triển trong thời đại 4.0, bao gồm việc phát triển web, mobile app, game, học máy và trí tuệ nhân tạo, phát triển đám mây, IoT, blockchain, microservices...
* C# là một ngôn ngữ lập trình hướng đối tượng đồng thời hỗ trợ lâp trình chức năng
* C# hỗ trợ mạnh mẽ cho phương pháp lâp trình hướng đối tượng, ngoài ra C# còn hỗ trợ các phuơng pháp lập trình chức năng thông qua các biểu thức lamba, khớp mẫu, functions, các thuộc tính bất biến.
* C# là ngôn ngữ gõ tĩnh, định kiểu mạnh, hỗ trợ gõ động.
* C# được gõ tĩnh nên nó mang đầy đủ các ưu việt của phuơng pháp gõ tĩnh như bảo đảm an toàn kiểu, tự động phân tích và nhận biết lỗi cú pháp ngay trong quá trình viết mã...
* Ngoài ra khi sử dụng C# kết hợp với IDE Visual Studio, C# được hỗ trợ gợi ý code bởi [Visual Studio IntelliCode](https://visualstudio.microsoft.com/services/intellicode/)sử dụng trí tuệ nhân tạo giúp cho việc viết code trở nên nhanh chóng và dễ dàng hơn
* C# là một ngôn ngữ ít [từ khóa](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/)
* C# có khoảng hơn 80 từ khóa
* C# là một trong các ngôn ngữ lập trình phổ biến và phát triển nhất
* Theo [TIOBE Index](https://www.tiobe.com/tiobe-index/), tính đến tháng 10/2020, C# là ngôn ngữ phổ biến thứ 5 thế giới.
* Theo [PYPL](http://pypl.github.io/PYPL.html), tính đến tháng 10/2020, C# là ngôn ngữ được cộng đồng quan tâm và chia sẻ nhiều thứ 4 thế giới.
* C# kết hợp chặt chẽ với nền tảng [.NET](https://dotnet.microsoft.com/)- một khung nền tảng rất mạnh của Microsoft.
* Ngoài ra C# còn có những ưu điểm:
* [C# là ngôn ngữ lập trình mã nguồn mở](https://github.com/dotnet/csharplang),vì vậy C# là miễn phí với tất cả mọi người, đồng thời mọi người đều có thể cùng tham gia phát triển, đề xuất thiết kế ngôn ngữ C#
* C# là ngôn ngữ đa nền tảng vì vậy có thể biên dịch trên nhiều nền tảng máy tính khác nhau (Windows, Linux, MacOS)
* C# có hiệu suất cao và tốc độ thực thi nhanh do sử dụng trình biên dich trung gian ([CLR](https://docs.microsoft.com/en-us/dotnet/standard/clr)), điểm cộng nữa là tốc độ phát triển phần mềm nhanh chóng so với đa số các ngôn ngữ hiện tại.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.
* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, song cũng được bổ sung các yếu tố mang tính hiện đại nên dễ dàng tiếp cận cho người mới học và học nhanh với C#.
* C# có cộng đồng nhà phát triển vô cùng lớn mạnh.
* C# được phát triển và cải tiến không ngừng với tần suất 1 phiên bản/ 1 năm, đáp ứng các mong muốn cải thiện, cải tiến cho phù hợp với nhu cầu công nghệ của các nhà phát triển.
* C# có [tài liệu tham khảo](https://docs.microsoft.com/en-us/dotnet/csharp/) và hướng dẫn vô cùng phong phú và chất lượng, đồng thời có các buổi hội thảo giới thiệu tính năng mới và định hướng phát triển ngôn ngữ trong tương lai.
* C# và .NET được đánh giá là có design tốt, vì vậy cú pháp và logic rất nhất quán, mã nguồn C# dễ đọc và mở rộng.

- C# được thiết kế và phát triển bởi Microsoft nên rất được Microsoft quan tâm và hỗ trợ.

*1.2.3**Mô hình 3 lớp của C#*

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: Là một lớp 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:

* Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và maintain project tốt hơn.
* Dễ dàng phân loại các hành động tại Business.
* Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,…
* Ứng dụng được cho các project lớn ở bên ngoài.

*1.2.4 Entity Framework của C#*

Entity Framework là một khung ORM mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ. Nó cho phép các nhà phát triển làm việc với dữ liệu bằng cách sử dụng các đối tượng của các lớp cụ thể của miền mà không cần tập trung vào các bảng và cột cơ sở dữ liệu cơ bản nơi dữ liệu này được lưu trữ. Với Entity Framework, các nhà phát triển có thể làm việc ở mức độ trừu tượng cao hơn khi họ xử lý dữ liệu và có thể tạo và duy trì các ứng dụng hướng dữ liệu với ít mã hơn so với các ứng dụng truyền thống.

Các tính năng khung thực thể:  
Đa nền tảng: EF Core là một khung đa nền tảng có thể chạy trên Windows, Linux và Mac.  
Mô hình hóa: EF (Entity Framework) tạo EDM (Mô hình dữ liệu thực thể) dựa trên các thực thể POCO (Plain Old CLR Object) với các thuộc tính get / set của các loại dữ liệu khác nhau. Nó sử dụng mô hình này khi truy vấn hoặc lưu dữ liệu thực thể vào cơ sở dữ liệu cơ bản.  
Truy vấn: EF cho phép chúng tôi sử dụng các truy vấn LINQ (C # / VB.NET) để truy xuất dữ liệu từ cơ sở dữ liệu cơ bản. Nhà cung cấp cơ sở dữ liệu sẽ dịch các truy vấn LINQ này sang ngôn ngữ truy vấn dành riêng cho cơ sở dữ liệu (ví dụ: SQL cho cơ sở dữ liệu quan hệ). EF cũng cho phép chúng tôi thực hiện các truy vấn SQL thô trực tiếp đến cơ sở dữ liệu.  
Theo dõi thay đổi: EF theo dõi các thay đổi xảy ra đối với các phiên bản của các thực thể của bạn (Giá trị thuộc tính) cần được gửi đến cơ sở dữ liệu.  
Lưu: EF thực thi các lệnh INSERT, UPDATE và DELETE vào cơ sở dữ liệu dựa trên những thay đổi xảy ra với các thực thể của bạn khi bạn gọi phương thức SaveChanges (). EF cũng cung cấp phương thức SaveChangesAsync () không đồng bộ.  
Đồng thời: EF sử dụng Đồng thời lạc quan theo mặc định để bảo vệ các thay đổi ghi đè do người dùng khác thực hiện do dữ liệu được lấy từ cơ sở dữ liệu.  
Giao dịch: EF thực hiện quản lý giao dịch tự động trong khi truy vấn hoặc lưu dữ liệu. Nó cũng cung cấp các tùy chọn để tùy chỉnh quản lý giao dịch.  
Bộ nhớ đệm: EF bao gồm cấp bộ nhớ đệm đầu tiên ra khỏi hộp. Vì vậy, truy vấn lặp đi lặp lại sẽ trả về dữ liệu từ bộ đệm thay vì nhấn cơ sở dữ liệu.  
Các quy ước tích hợp: EF tuân theo các quy ước về mẫu lập trình cấu hình và bao gồm một bộ quy tắc mặc định tự động định cấu hình mô hình EF.  
Cấu hình: EF cho phép chúng tôi định cấu hình mô hình EF bằng cách sử dụng các thuộc tính chú thích dữ liệu hoặc API Fluent để ghi đè các quy ước mặc định.  
Di chuyển: EF cung cấp một tập hợp các lệnh di chuyển có thể được thực thi trên Bảng điều khiển Trình quản lý gói NuGet hoặc Giao diện dòng lệnh để tạo hoặc quản lý Lược đồ cơ sở dữ liệu cơ bản.

### *1.3 Tổng quan về hệ quản trị cơ sở dữ liệu SQL Sever*

- Hệ quản trị cơ sở dữ liệu Microsoft SQL Server (MSSQL) là một trong những hệ quản trị cơ sở dữ liệu thông dụng nhất hiện nay. Đây là hệ quản trị cơ sở dữ liệu thường được sử dụng trong các hệ thống trung bình, với ưu điểm có các công cụ quản lý mạnh mẽ giúp cho việc quản lý và bảo trì hệ thống dễ dàng, hỗ trợ nhiều phương pháp lưu trữ, phân vùng và đánh chỉ mục phục vụ cho việc tối ưu hóa hiệu năng.

- SQL Server luôn được Microsoft cải tiến để nâng cao hiệu năng, tính sẵn sàng của hệ thống, khả năng mở rộng và bảo mật, cung cấp nhiều công cụ cho người phát triển ứng dụng được tích hợp với bộ Visual Studio do Microsoft cung cấp. SQL Server có 4 dịch vụ lớn là Database Engine, Intergration Service, Reporting service và Analysis Services.

* DataBase Engine: được phát triển để thực thi tốt hơn với việc hỗ trợ cả dữ liệu có cấu trúc và dữ liệu phi cấu trúc (XML).
* Khả năng sẵn sàng của hệ thống được nâng cao, hỗ trợ các chức năng: Database mirroring (cơ sở dữ liệu gương), failover clustering , snapshots và khôi phục dữ liệu nhanh.
* Việc quản lý chỉ mục được thực hiện song song với việc hoạt động của hệ thống. Người dùng có thể thêm chỉ mục, xây dựng lại chỉ mục hay xóa một chỉ mục đi trong khi hệ thống vẫn được sử dụng.
* Chức năng phân vùng dữ liệu được hỗ trợ: Người dùng có thể phân vùng các bảng và chỉ mục cũng như quản lý phân vùng dữ liệu một cách dễ dàng. Việc hỗ trợ phân vùng dữ liệu giúp nâng cao hiệu năng hoạt động của hệ thống.
* Dịch vụ đồng bộ hóa dữ liệu được mở rộng với việc hỗ trợ mô hình đồng bộ hóa ngang hàng. Đây là dịch vụ giúp đồng bộ hóa dữ liệu giữa các máy chủ dữ liệu, giúp mở rộng khả năng của hệ thống.
* Dịch vụ tích hợp (Integration Service) thiết kế lại cho phép người dùng tích hợp dữ liệu và phân tích dữ liệu từ nhiều nguồn khác nhau. Hỗ trợ việc quản lý chất lượng dữ liệu và làm sạch dữ liệu, một công việc quan trọng trong tiến trình ETL.
* Dịch vụ phân tích dữ liệu (Analysis Service): cung cấp khung nhìn tích hợp và thống nhất về dữ liệu cho người dùng, hỗ trợ việc phân tích dữ liệu.
* Công cụ khai phá dữ liệu (Data mining) được tích hợp hỗ trợ nhiều thuật toán khai phá dữ liệu, hỗ trợ cho việc phân tích, khai phá dữ liệu và xây dựng các hệ thống hỗ trợ ra quyết định cho người quản lý.
* Dịch vụ xây dựng quản lý báo cáo (Reporting Service) dựa trên nền tảng quản trị doanh nghiệp thông minh và được quản lý qua dịch vụ web. Báo cáo có thể được xây dựng dễ dàng với ngôn ngữ truy vấn MDX. Thông qua các công cụ trên Business Intelligent, người dùng dễ dàng truy cập báo cáo và trích xuất ra nhiều định dạng khác nhau thông qua trình duyệt web.

***1.4 Mô tả hệ thống:***

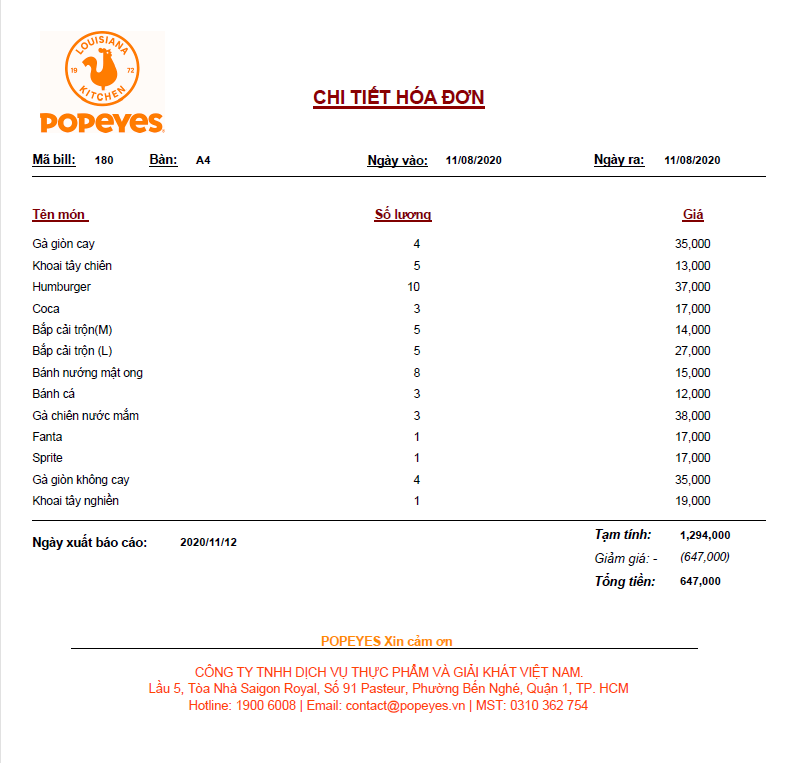
Hệ thống cửa hàng gồm những chức năng chính sau: đăng nhập, đặt bàn, đặt món, tính tiền, xuất hóa đơn, thông tin cá nhân, xem doanh thu, quản lí món ,quản lý danh mục, quản lí bàn ,quản lí tài khoản.

- Phần đăng nhập: có chức năng xác nhận người truy cập , không có chức năng đăng ký tài khoản vì lí do bảo mật (nhân viên có thể tự tạo 1 tài khoản ) thay vào đó , tài khoản sẽ được tạo do admin và cung cấp cho nhân viên hoặc khi quên mật khẩu thì phải liên hệ với quản lí để cấp mật khẩu mới. Nói về bảo mật gồm có 2 phần : User name và password , những thông tin này có xác định chữ hoa và chữ thường vì vậy phải nhập thật chính xác để có thể truy cập vào ứng dụng đây được coi như 1 điểm nổi bật.

- Phần đặt bàn : có chức năng thông báo các bàn đang được sử dụng hoặc đang trống , khi có người đến đặt bàn thì nhân viên sẽ kiểm tra trạng thái của bàn để sắp xếp chỗ ngồi hợp lý. Cách thay đổi trạng thái là bấm vào bàn và sẽ xuất hiện trang đặt món. Sau khi đặt món thì bàn sẽ tự động chuyển về trạng thái “Có người” và sẽ chuyển trạng thái về “Trống” khi không có món được chọn hoặc hóa đơn đã được thanh toán.

- Phần oder(đặt món): có chức năng trình bày menu món ăn và sẽ thống kê số lượng món được thêm khi khách yêu cầu, ngoài ra còn chức năng nhập khuyến mại để giảm giá món ăn, tìm kiếm món ăn thông qua danh mục món ăn (loại thức ăn) và tên món ăn cần tìm sau khi nhập món ăn xong thì sẽ bấm có các tùy chọn: “Chuyển bàn” để chuyển bàn khác khi khách yêu cầu nhưng món ăn vẫn được giữ nguyên, “Hủy” để hủy món ăn mới nhập hoặc đang chọn, “Xong” để lưu danh sách món ăn của bàn và thoát ra ngoài trang đặt bàn để xử lý các giao dịch tiếp theo đồng thời chuyển qua trạng thái chưa thanh toán, “Tính tiền” để tính tổng giá tiền mà khách phải thanh toán và có thể xuất ra 1 hóa đơn để khách hàng có thể kiểm tra.

- Phần thanh toán : khi kích hoạt trạng thái tính tiền trong trang order món ăn thì lập tức hóa đơn sẽ được lưu lại và xuất ra 1 hóa đơn thanh toán ,trong hóa đơn có ghi rõ mã hóa đơn , số bàn , ngày vào , ngày ra món ăn , số lượng, giá , khuyến mãi (giảm giá ) và số tiền phải thanh toán cuối cùng. Không những thế phần mềm cung cấp chế độ in hóa đơn để khách có thể kiểm tra lại bữa ăn của mình , và chế độ xuất hóa đơn dưới nhiều định dạng khác nhau như: PDF, Word, Excel, Richtext, XML… để có thể lưu trữ lại thông tin cần thiết.



*(hình ảnh minh họa)*

- Phần thông tin cá nhân: được tích hợp để nhân viên và cả quản lí có thể thay đổi thông tin hiển thị , mật khẩu. Quy chuẩn để đặt lại thông tin cá nhân là tên hiển thị không được vượt quá 30 kí tự , nhập lại mật khẩu phải trùng với mật khẩu mới ,đặc biệt biệt tên người dùng (username) không thể thay đổi.

- Phần dành cho Quản lí (admin) : chỉ có tài khoản quản lý mới có thể truy cập. Có chức năng xem các báo cáo thống kê doanh thu theo ngày, chức năng thêm, xóa, sửa cho bàn ăn, món ăn, loại món ăn, và tài khoản .

- Phần xem doanh thu : dùng để lưu trữ thông tin các cuộc giao dịch các hóa đơn đã thanh toán , những dữ liệu này cung cung cấp dữ liệu cụ thể từ mã hóa đơn đến thiền thực tế đã nhận, khi có quá nhiều trường dữ liệu hãy dùng chức năng tìm kiếm có sẵn cho phép người dùng tìm hóa đơn theo mã hoặc thống kê theo ngày một cách nhanh chóng. Ngoài ra khi chọn 1 dòng dữ liệu và chọn chức năng xem chi tiết thì sẽ cung cấp chi tiết các món ăn và số lượng cụ thể...Điều này tạo thuận lợi cho việc truy xuất thông tin từ những hóa đơn đã lâu . Không những thế khi chọn mục xem báo cáo thì các dữ liệu sẽ bắt đầu được sắp xếp ngăn nắp và hiển thị qua privew với đầy đủ các thuộc tính cần thiết như: in , tìm kiếm, tính tổng số tiền đã nhận được thực tế , đếm các hóa đơn và giao dịch đã hoàn thành.

- Quản lý món ăn: phần này được tích hợp trong form chính admin đồng nghĩa với việc những tác vụ thêm xóa sửa sẽ chỉ được thực hiện bởi quản lý. Trong phần này quản lý sẽ có thể dễ dàng quản lý những món ăn đang có , có thể thêm món mới, sửa món ăn , xóa món ăn (lưu ý khi sử dụng sửa và xóa thì dữ liệ sẽ bị ảnh hưởng gây nên sự sai lệch về số tiền tổng hợp và chi tiết hóa đơn ở trang thống kê doanh thu). Khi muốn thêm một món mới Quản lí chỉ cần nhập các thuộc tính khác của món như tên món , loại, giá, hình ảnh (với định dạng png, jpeg) và sau đó mã ID của món sẽ được cập nhật tự động vô cùng tiện lợi , khi chọn “Lưu” mọi thứ sẽ hoàn tất và sẽ được thêm ngay trong menu món ăn để sẵn sàng phục vụ,khi chọn “Không lưu” thì tất cả các dữ liệu chờ sẽ bị hủy và chọn “Thoát” để rời khỏi .

- Quản lý danh mục món ăn: cũng tương tự như quản lý món ăn , quản lý loại đồ ăn(danh mục) cho phép thêm danh mục mới , sửa danh mục đang chọn, xóa danh mục (Lưu ý: khi chọn xóa danh mục hãy để ý những món ăn đang để theo danh mục này vì khi xóa thì mọi món ăn của danh mục nãy sẽ biến mất và khi thống kê sẽ xảy ra sai sót về món ăn và giá tiền, tốt nhất nên chọn những thao tác này vào lúc khách vắng để đảm bảo chất lượng),chọn “Lưu” để lưu danh mục mới hoặc mới sửa, chọn “Không lưu” để hủy bỏ thao tác , chọn “Thoát để rời khỏi”.

- Quản lý bàn ăn : chức năng thêm xóa sửa cũng được áp dụng nguyên xi những tính năng cơ bản phải có (Thêm, xóa, sửa). Tiện lợi hơn khi bạn có thể không cần làm gì tới mã ID vì khi xử lý hệ thống sẽ cung cấp mã ID tự động.

- Quản lý tài khoản: là phần khá quan trọng vì để tạo ra và quản lý nó với nhiều chi nhánh khác nhau thì chuyện này cần phải có 1 báo cáo tài khoản và phần mềm đã tích hợp tạo báo cáo danh sách tài khoản đồng thới đếm số lượng tài khoản đang tồn tại . Về phần quản lý thì vẫn có những tác vụ cơ bản (Thêm, xóa, sửa) và lưu ý phần tên người dùng là bắt buộc phải có và phải nhập bằng tay, điều kiện kèm theo là không được trùng với bất kì những tên đăng nhập nào đã và đang tồn tại trước đó bên cạnh đó tài tên đăng nhập cũng khoong được vượt quá 20 kí tự.

**CHƯƠNG 2. CÁC LOẠI LƯỢC ĐỒ VÀ MÔ TẢ HÌNH THỨC VẬN HÀNH**

## 2.1. Sơ đồ USECASE:

\*Biểu đồ Use case là một mô hình đồ hoạ về các chức năng của hệ thống từ khung nhìn của người sử dụng.

*2.1.1. Danh sách mô tả tác nhân (actor):*

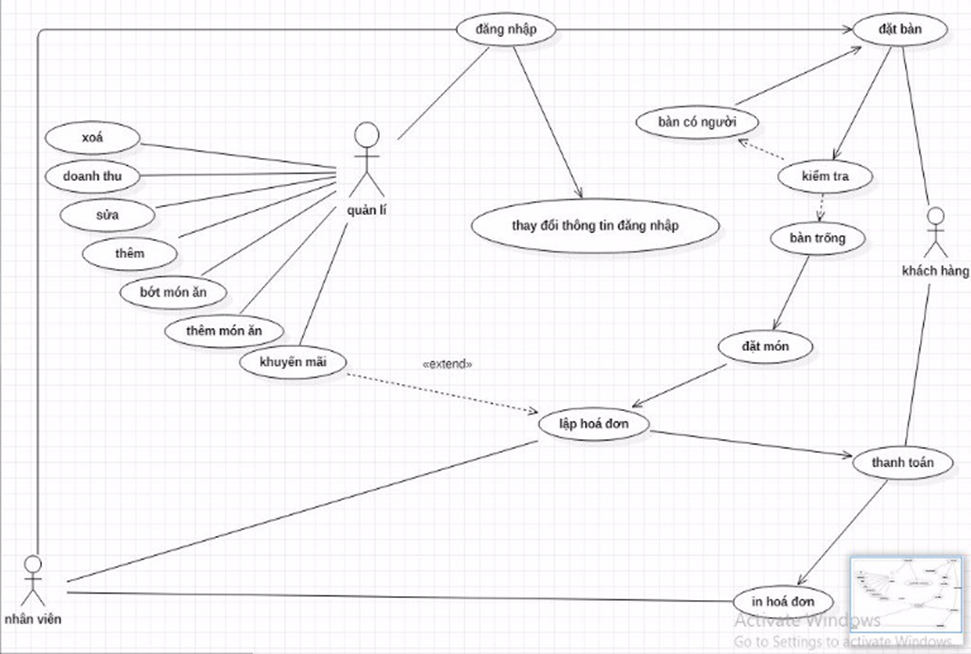
- Admin(nhân viên quản lý): Có thể sao lưu và phục hồi CSDL, quản lý nhân viên, quản lý người dùng,…

- Nhân viên cửa hàng: Nhập khách hàng mới, lập phiếu nhận hàng, phiếu bán hàng, lập báo cáo.

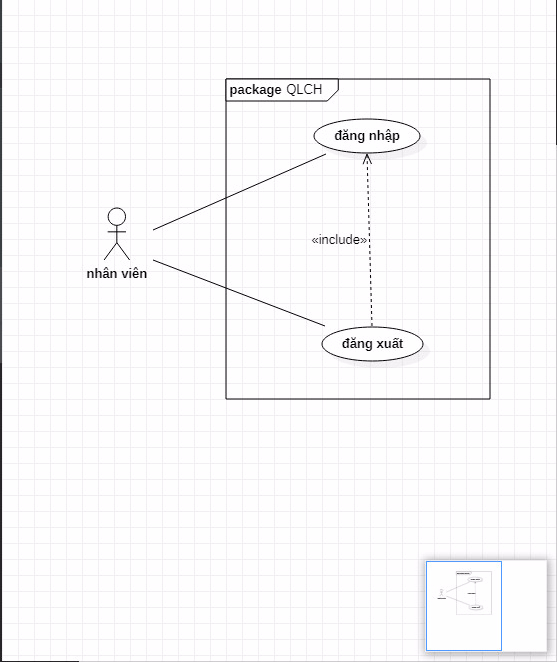
*2.1.2. Danh sách và mô tả usecase trong thực tế:*

|  |  |
| --- | --- |
| **USECASE** | **Mô tả** |
| Đăng nhập | Nhân viên cửa hàng khi muốn sử dụng hệ thống điều phải đăng nhập vì đây là một hệ thống nội bộ. |
| Đăng xuất | Khi nhân viên sử dụng xong thì tiến hành đăng xuất để thoát khỏi hệ thống. |
| Quản lí món ăn | Nhân viên cửa hàng quản lý các món ăn có trong kho để biết rõ về thông tin các món ăn nào hết món hay tồn lại và lập báo cáo tồn kho trong mỗi tháng. |
| Quản lí nhà cung cấp | Quản lý các nhà cung cấp cung ứng món ăn cho cửa hàng, bao gồm: thêm, xóa, sửa nhà cung cấp. |
| Quản lí nhân viên - người dùng | Quản lý các thông tin nhân viên, bao gồm thêm, xóa, sửa và tính lương nhân viên. |
| Quản lí khách hàng | Quản lý các khách hàng đã mua món ăn bao gồm: thêm, xóa, sửa nhà cung cấp. |
| Lập phiếu nhập hàng | Khi phát hiện số lượng hàng tồn nhỏ hơn 5 đơn vị thì nhân viên cửa hàng sẽ lập một phiếu đề nghị nhập hàng. |

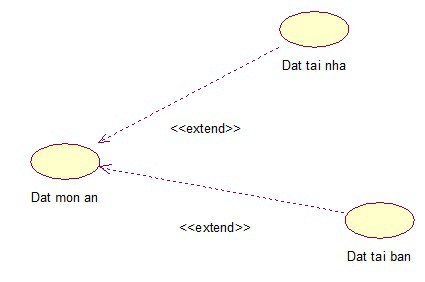
*2.1.3. Sơ đồ usecase tổng quát*

**

*2.1.4. Sơ đồ usecase đăng nhập:*



*2.1.5. Sơ đồ usecase đặt món:*



\* Chức năng đặt món ăn:

- Khi khách hàng đến nhà hàng yêu cầu nhân viên nhà hàng cho xem thực đơn gồm những món gì. Tùy theo khách hàng có thể đến trực tiếp nhà hàng đặt hoặc đặt trước qua điện thoại.

- Sau khi khách hàng chọn được món, đọc để nhân viên ghi vào hóa đơn. Nhân viên đưa cho nhân viên nhà bếp liên 1 của hóa đơn.

- Khi nhận được yêu cầu của khách nhân viên bếp dựa vào yêu cầu của khách để thực hiện chế biến món ăn mà khách hàng đặt.

\* Trường hợp ngoại lệ:

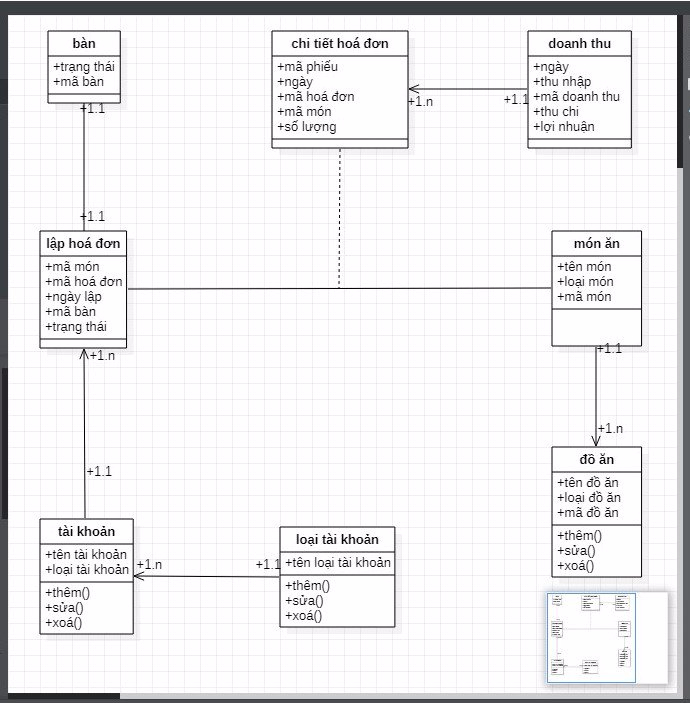
- Món ăn khách hàng đặt đã hết.

- Không đủ nguyên liều để làm món ăn khách hàng đặt.

- Làm không đúng yêu cầu của khách hàng, và khách hàng yêu cầu đổi món.

*2.2. Sơ đồ lớp(Class)*

-Biểu đồ lớp biểu diễn các khái niệm lớp và mối quan hệ giữa chúng với nhau. Các lớp phân tích chỉ có tên lớp, tên thuộc tính, tên dịch vụ.

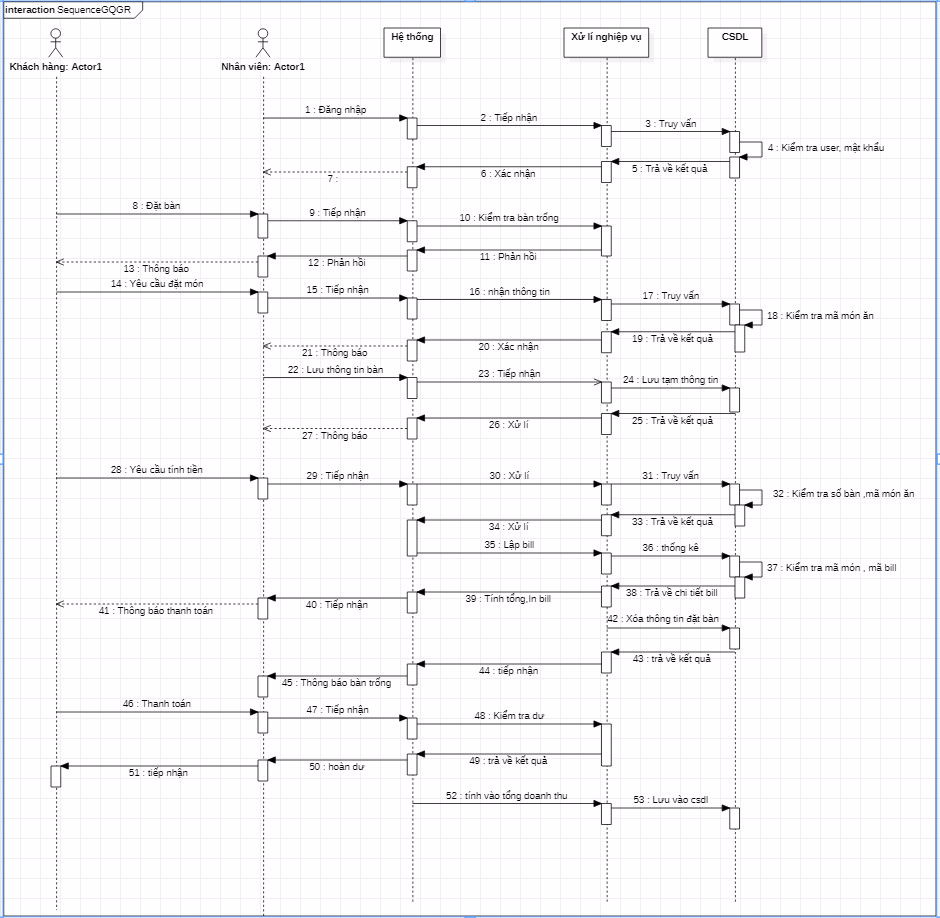


**Hình 4: Sơ đồ lớp**

- Các lớp thiết kế thì các tên được chuẩn hoá, thay đổi và thêm những nội dung khác liên quan đến ngôn ngữ cài đặt như: Tên biến, kiểu biến, tên hàm, kiểu hàm, danh sách các đối của hàm....

*2.3. Sơ đồ tuần tự:*

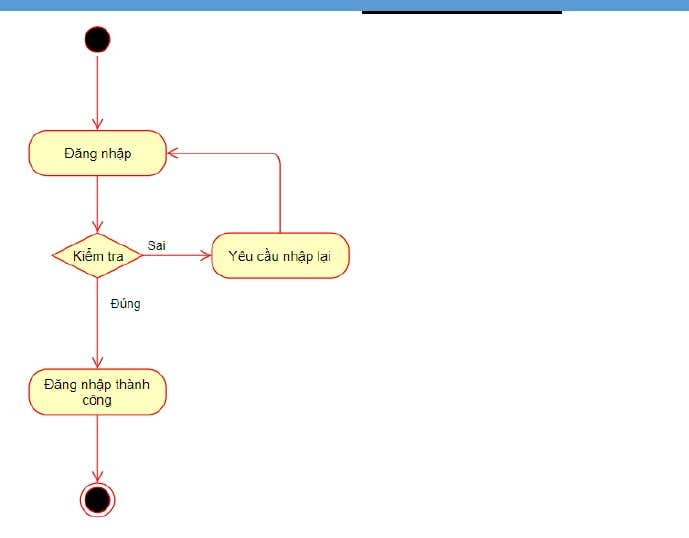
- Sơ đồ trình tự mô tả sự tương tác và trao đổi thông tin giữa các đối tượng trong hệ thống theo trình tự thời gian.



\* Các lớp thiết kế thì các tên lớp, đối tượng được chuẩn hoá, Thông điệp được đổi thành hàm. Có thể thêm các đối tượng giao diện, vào ra cơ sở dữ liệu.

*2.4. Sơ đồ Activity:*

*2.4.1. Sơ đồ xử lý đăng nhập*



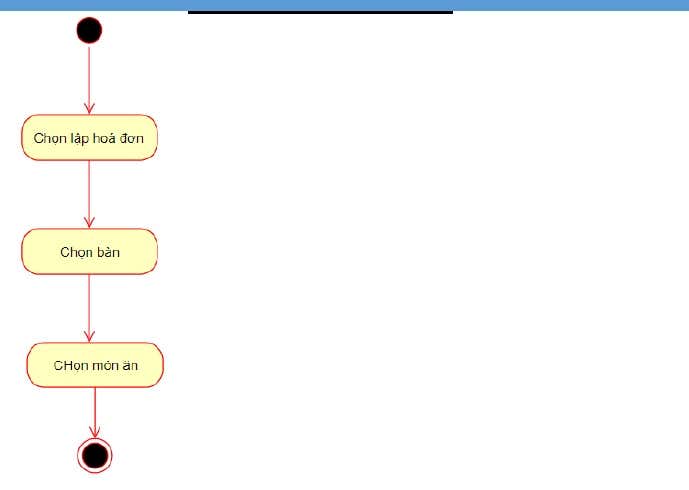
Bước 1:

* Chức năng đăng nhập do tất cả người dùng thực hiện
* Khi đăng nhập người dùng thực hiện các công việc như sau
* Chọn đăng nhập
* Kiểm tra thông tin
* Thông báo khi đăng nhập thành công

Bước 2:

* Chọn đăng nhập
* Kiểm tra thông tin
* Đúng: đăng nhập thành công
* Sai: yêu cầu nhập lại
* Thông báo khi đăng nhập thành công

*2.4.2. Sơ đồ xử lý hoá đơn:*



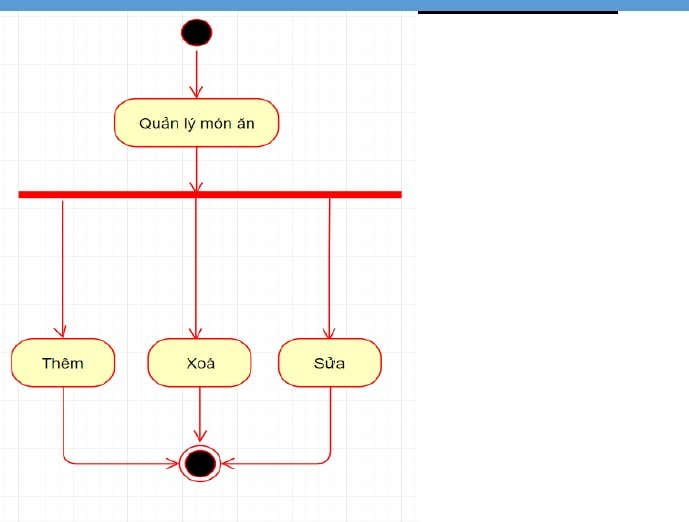
Bước 1:

* Chức năng lập hóa đơn do nhân viên thu ngân thực hiện
* Khi nhân viên thu ngân muốn lập hóa đơn cần thực hiện các công việc như sau
* Chọn lập hóa đơn
* Chọn bàn
* Chọn món ăn

Bước 2:

* Chọn lập hóa đơn
* Chọn bàn
* Chọn món ăn

*2.4.3. Sơ đồ quản lý món ăn:*



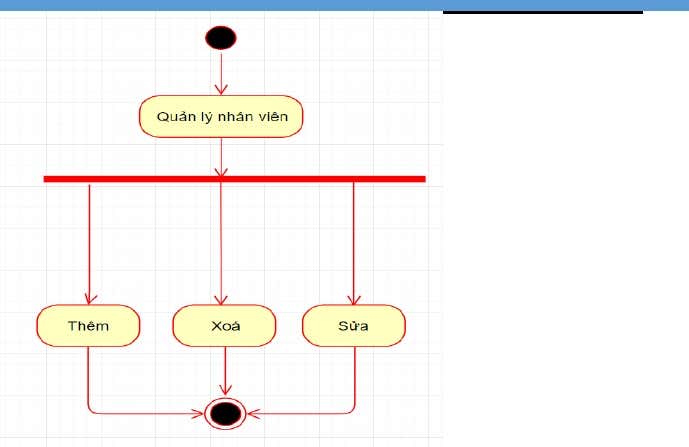
Bước 1:

* Chức năng quản lí món ăn do quản lý thực hiện
* Khi quản lí muốn thay đổi dữ liệu món ăn cần thực hiện các công việc sau
* Thêm món ăn
* Xóa món ăn
* Sửa món ăn

Bước 2

* Thêm món ăn:
* Nhập thông tin món ăn
* Xóa món ăn:
* Chọn món ăn cần xóa
* Sửa món ăn:
* Chọn món ăn cần sửa
* Thay đổi thông tin món ăn

*2.4.4. Sơ đồ quản lý nhân viên:*



Bước 1:

* Chức năng quản lí nhân viên do quản lý thực hiện
* Khi quản lí muốn thay đổi dữ liệu nhân viên cần thực hiện các công việc sau
* Thêm nhân viên
* Xóa nhân viên
* Sửa nhân viên

Bước 2

* Thêm nhân viên:
* Nhập thông tin nhân viên
* Xóa nhân viên:
* Chọn nhân viên cần xóa
* Sửa nhân viên:
* Chọn nhân viên cần sửa
* Thay đổi thông tin nhân viên

*2.5. Phân tích chi tiết sơ đồ USECASE:*

*2.5.1. Đăng nhập:*

|  |  |
| --- | --- |
| Tên usecase: Đăng nhập | ID:1 |
| Tác nhân chính: Nhân viên | Kiểu: Chi tiết |
| Người tham gia/ quan tâm:  *Nhân viên:* đăng nhập để làm việc với hệ thống | |
| Mô tả:  Đăng nhập là bước đầu tiên để nhân viên có thể sử dụng được các chức năng mà hệ thống cung cấp (dựa trên loại tài khoản được sử dụng). Hệ thống được sử dụng nội bộ cho nên việc đăng nhập là rất cần thiết cho việc quản lý. | |
| Sự kiện kích hoạt:  Khi nhân viên muốn làm việc với hệ thống. | |
| Các quan hệ:  +Association (kết hợp):  +Include(bao gồm): Có kết nối mạng và kết nối với form đăng nhập tương ứng. +Extend(mở rộng):  +Generalization(tổng quát hóa): | |
| Luồng sự kiện chính:   1. Người dùng chọn chức năng “Đăng nhập”. 2. Hiển thị form đăng nhập và yêu cầu người dùng nhập thông tin. 3. Người dùng nhập thông tin đăng nhập vào hệ thống. và nhấn nút “Xác nhận” để xác nhận đăng nhập. nếu bấm “Hủy” thì thực hiện luồng phụ A1. 4. Hệ thống tiến hành đăng nhập. Nếu sai thông tin tài khoản hoặc mật khẩu, … thì thực hiện luồng phụ A2. 5. Hệ thống tiến hành thao tác đăng nhập vào hệ thống. 6. Hệ thống trả về những chức năng mà người dùng có thể sử dụng trong hệ thống (dựa vào loại tài khoản) trên form chính.   7. Kết thúc sự kiện. | |
| Luồng phụ:  Luồng phụ A1: Người dùng thoát đăng nhập.   1. Hệ thống thông báo đăng nhập thất bại. 2. Kết thúc use case.   Luồng phụ A2: Kiểm tra bị lỗi.   1. Hệ thống báo lỗi cho người đặng nhập(không tồn tại người dùng, sai mật khẩu,…). 2. Hệ thống đưa ra câu hỏi đăng nhập tiếp hay kết thúc. Nếu đăng nhập tiếp thì quay lại bước 2 trong luồng sự kiện chính.   3. Kết thúc use case. | |

*2.5.2. Đăng xuất:*

|  |  |
| --- | --- |
| Usecase name: Đăng xuất | ID:2 |
| Tác nhân chính: Nhân viên | Kiểu: Chi tiết |
| Người tham gia/ quan tâm:  *Nhân viên:* đăng xuất để thoát khỏi hệ thống | |
| Mô tả:  Đăng xuất là khi sử dụng xong, tiến hành đăng xuất để thoát khỏi hệ thống. | |
| Sự kiện kích hoạt:  Nhân viên đăng nhập vào hệ thống.  Có yêu cầu thoát khỏi hệ thống. | |
| Các quan hệ:  +Association (kết hợp): Nhân viên  +Include(bao gồm): Đăng nhập  +Extend(mở rộng):  +Generalization(tổng quát hóa): | |
| Luồng sự kiện chính:   1. Người dùng chọn chức năng đăng xuất. 2. Hệ thống yêu cầu xác nhận đăng xuất. 3. Người dùng chọn xác nhận. nếu hủy bỏ thì thực hiện luồng phụ A1. 4. Hiển thị form đăng nhập. 5. Hệ thống tiến hành đăng xuất. 6. Hiển thị form đăng nhập.   7. Kết thúc use case. | |
| Luồng phụ:  Không có | |

*2.5.3. Quản lý nhân viên:*

|  |  |
| --- | --- |
| Tên usecase: Quản lý nhân viên | ID:3 |
| Tác nhân chính: Nhân viên quản lý(admin) | Kiểu: Chi tiết |
| Người tham gia/ quan tâm:  *Nhân viên:* muốn làm việc với chức năng quản lý nhân viên của hệ thống | |
| Mô tả:  Khi cần xem hoặc tùy chỉnh thông tin nhân viên thì nhân viên quản lý sẽ thực hiện chức năng này. | |
| Sự kiện kích hoạt:  Nhân viên mở chức năng trên giao diện | |
| Các quan hệ:  +Association (kết hợp):  +Include(bao gồm): Đăng nhập  +Extend(mở rộng): In danh sách nhân viên  +Generalization(tổng quát hóa): | |
| Luồng sự kiện chính:  1. Nhân viên chọn các chức năng trên giao diện  Luồng phụ 1: Thêm nhân viên  Luồng phụ 2: Xóa nhân viên  Luồng phụ 3: Sửa nhân viên  2. Cập nhật xuống cơ sở dữ liệu  3. Kết thúc sự kiện. | |
| Luồng phụ:  Luồng phụ 1: Thêm nhân viên  1. Nhập thông tin nhân viên  2. Kiểm tra thông tin nhập vào  3. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực hiện bước 2 trong luồng sự kiện chính  4. Kết thúc use case  Luồng phụ 2: Xóa nhân viên  1. Chọn nhân viên cần xóa  2. Chọn chức năng xóa  3. Xác nhận yêu cầu, nếu “OK” thì thực hiện bước 2 trong luồng sự kiện  chính, ngược lại thì đến bước 4 – luồng phụ  4. Kết thúc use case  Luồng phụ 3: Sửa nhân viên  1. Chọn nhân viên cần sửa  2. Nhập thông tin tùy chỉnh  3. Kiểm tra thông tin nhập vào  4. Nhấn nút cập nhật  5. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực  hiện bước 2 trong luồng sự kiện chính  6. Kết thúc usecase | |
| Trường hợp ngoại lệ: | |

*2.5.4. Đặt món ăn:*

|  |  |
| --- | --- |
| Tên usecase: Đặt món ăn | ID:4 |
| Tác nhân chính: Nhân viên | Kiểu: Chi tiết |
| Người tham gia/ quan tâm:  *Nhân viên:* muốn làm việc với chức năng quản lý nhân viên của hệ thống | |
| Mô tả:  Khi cần xem hoặc tùy chỉnh thông tin nhân viên thì nhân viên quản lý sẽ thực hiện chức năng này. | |
| Sự kiện kích hoạt:  Nhân viên mở chức năng trên giao diện | |
| Các quan hệ:  +Association (kết hợp):  +Include(bao gồm): Đăng nhập  +Extend(mở rộng): In danh sách món ăn  +Generalization(tổng quát hóa): | |
| Luồng sự kiện chính:  1. Nhân viên chọn các chức năng trên giao diện  Luồng phụ 1: Thêm món ăn  Luồng phụ 2: Xóa món ăn  Luồng phụ 3: Sửa món ăn  2. Cập nhật xuống cơ sở dữ liệu  3. Kết thúc sự kiện. | |
| Luồng phụ:  Luồng phụ 1: Thêm món ăn  1. Nhập thông tin món ăn  2. Kiểm tra thông tin nhập vào  3. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực hiện bước 2 trong luồng sự kiện chính  4. Kết thúc use case  Luồng phụ 2: Xóa món ăn  1. Chọn món ăn cần xóa  2. Chọn chức năng xóa  3. Xác nhận yêu cầu, nếu “OK” thì thực hiện bước 2 trong luồng sự kiện chính, ngược lại thì đến bước 4 – luồng phụ  4. Kết thúc use case  Luồng phụ 3: Sửa món ăn  1. Chọn món ăn cần sửa  2. Nhập thông tin tùy chỉnh  3. Kiểm tra thông tin nhập vào  4. Nhấn nút cập nhật  5. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực hiện bước 2 trong luồng sự kiện chính  6. Kết thúc usecase | |
| Trường hợp ngoại lệ: | |

*2.5.5. Quản lí nhà cung cấp(đang trong thời gian phát triển):*

|  |  |
| --- | --- |
| Tên usecase: Quản lý nhà cung cấp | ID:5 |
| Tác nhân chính: Nhân viên | Kiểu: Chi tiết |
| Người tham gia/ quan tâm:  *Nhà cung cấp:* muốn làm việc với chức năng quản lý nhân viên của hệ thống | |
| Mô tả:  Khi cần xem hoặc tùy chỉnh thông tin nhập nguyên liệu thì phải thông qua quản lý để thực hiện chức năng này. | |
| Sự kiện kích hoạt:  Quản lý mở chức năng trên giao diện | |
| Các quan hệ:  +Association (kết hợp):  +Include(bao gồm): Đăng nhập  +Extend(mở rộng): In danh sách nhà cung cấp  +Generalization(tổng quát hóa): | |
| Luồng sự kiện chính:  1. Nhân viên chọn các chức năng trên giao diện  Luồng phụ 1: Thêm cung cấp  Luồng phụ 2: Xóa cung cấp  Luồng phụ 3: Sửa cung cấp  2. Cập nhật xuống cơ sở dữ liệu  3. Kết thúc sự kiện. | |
| Luồng phụ:  Luồng phụ 1: Thêm cung cấp  1. Nhập thông tin cung cấp  2. Kiểm tra thông tin nhập vào  3. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực hiện bước 2 trong luồng sự kiện chính  4. Kết thúc use case  Luồng phụ 2: Xóa cung cấp  1. Chọn cung cấp cần xóa  2. Chọn chức năng xóa  3. Xác nhận yêu cầu, nếu “OK” thì thực hiện bước 2 trong luồng sự kiện chính, ngược lại thì đến bước 4 – luồng phụ  4. Kết thúc use case  Luồng phụ 3: Sửa cung cấp  1. Chọn cung cấp cần sửa  2. Nhập thông tin tùy chỉnh  3. Kiểm tra thông tin nhập vào  4. Nhấn nút cập nhật  5. Thông báo lỗi khi thông tin sai quy định và quay lại bước 1, ngược lại thực hiện bước 2 trong luồng sự kiện chính  6. Kết thúc usecase | |
| Trường hợp ngoại lệ: | |

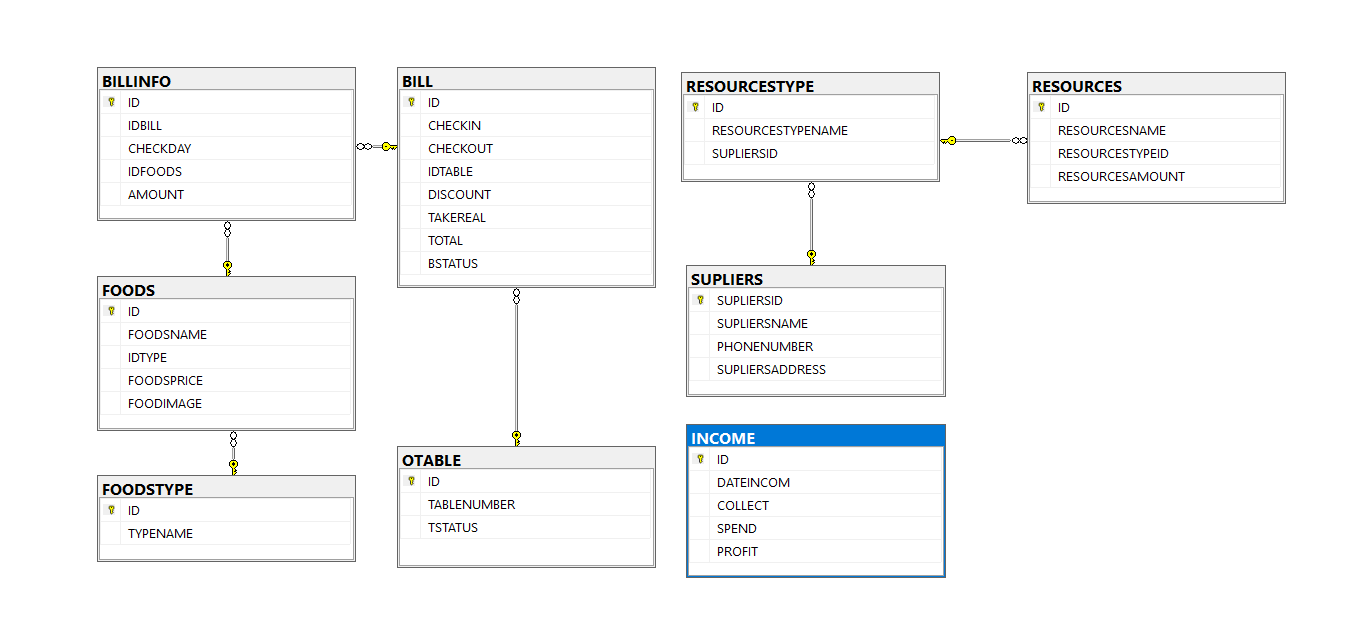
*2.5.6. Lập hoá đơn:*

|  |  |
| --- | --- |
| Tên usecase: Lập hoá đơn | ID:7 |
| Tác nhân chính: Nhân viên, quản lý | Kiểu: Chi tiết |
| Những người tham gia/quan tâm: Nhân viên, quản lý | |
| Mô tả:  Khi có yêu cầu bán sản phẩm nào đó cho khách hàng thì nhân viên sẽ lập phiếu bán cho các sản phẩm đó. | |
| Phạm vi: Khi một khách hàng đến mua món ăn của cửa hàng | |
| Điều kiện tiên quyết: Món ăn khách hàng yêu cầu phải có trong danh mục món ăn của cửa hàng | |
| Điều kiện thực hiện: - Khách hàng có yêu cầu mua món ăn  - Nhân viên phải có quyền lập phiếu bán | |
| Sự kiện kích hoạt:  Nhân viên mở chức năng trên giao diện | |
| Các quan hệ:  +Include(bao gồm): Không có  +Extend(mở rộng): Không có  +Generalization(tổng quát hóa): Không có | |
| Luồng sự kiện chính:   1. Yêu cầu khách hàng nhập thông tin món ăn 2. Khách hàng nhập thông tin món ăn 3. Hệ thống kiểm tra thông tin món ăn của khách hàng đã yêu cầu. Nếu món ăn không có trong cửa hàng th ìthực hiện luồng phụ A1 4. Hệ thống yêu cầu khách hàng nhập thông tin gồm: tên, địa chỉ, số điện thoại,… 5. Khách hàng nhập thông tin cho hệ thống 6. Tiến hành lập phiếu bán với các thông tin: tên, món ăn, ngày lập phiếu, đơn giá, số lượng, tổng số tiền 7. Lưu và in phiếu vừa lập 8. Thông tin hoàn thành 9. Kết thúc   Luồng phụ:  Luồng phụ A1: Món ăn đã hết   1. Hệ thống thông báo không có món ăn khách hàng cần   2. Kết thúc usecase | |

*2.5.7. Doanh thu:*

|  |  |
| --- | --- |
| Tên usecase: Doanh thu | ID:8 |
| Tác nhân chính: Nhân viên | Kiểu: Chi tiết |
| Những người tham gia/quan tâm: Nhân viên | |
| Mô tả:  Bộ phận tài chính sẽ dựa vào hoá đơn xuất, nhập hàng, các khoản thu chi | |
| Sự kiện kích hoạt:  Khi nhân viên nhập món ăn được yêu cầu | |
| Các quan hệ:  +Include(bao gồm): Không có  +Extend(mở rộng): Không có  +Generalization(tổng quát hóa): Không có | |
| Luồng chính:   1. Yêu cầu nhân viên nhập doanh thu kiểm tra tổng doanh thu trong 1 tháng. 2. Nhân viên sẽ tiến hành nhập doanh thu gồm các thông tin sau: tiền từng món ăn, tổng tiền thu được,…   Luồng phụ: Không có | |

**CHƯƠNG 3. THUỘC TÍNH VÀ KIỂU DỮ LIỆU:**

 OTABLE( **TABLENUMBER**, STATUS )

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **TABLENUMBER(PK)** | VARCHAR | 5 |
| STATUS | NVARCHAR | 10 |
|  |  |  |

ACCOUNTTYPE(**POSITION,** ACCOUNTTYPE)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **POSITION(PK)** | INT |  |
| ACCOUNTTYPE | NVARCHAR | 20 |
|  |  |  |

ACCOUNT(**USERNAME ,** DISPLAYNAME, UPASSWORD, POSITION)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **USERNAME (PK)** | VARCHAR | 30 |
| DISPLAYNAME | NVARCHAR | 30 |
| UPASSWORD | VARCHAR | 20 |
| POSITION(FK) | INT |  |

FOODSTYPE(**ID**, TYPENAME)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID (PK)** | INT |  |
| TYPENAME | NVARCHAR | 50 |

FOODS(**ID**, FOODSNAME, IDTYPE, FOODSPRICE, FOODIMAGE)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| FOODSNAME | NVARCHAR | 50 |
| IDTYPE(FK) | INT |  |
| FOODSPRICE | MONEY |  |
| FOODIMAGE | VARBINARY | MAX |

BILL(**ID**, CHECKIN, CHECKOUT, TABLENUMBER, BSTATUS)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| CHECKIN | DATE |  |
| CHECKOUT | DATE |  |
| TABLENUMBER | VARCHAR | 5 |
| BSTATUS | BIT |  |

BILLINFO(**ID**, IDBILL, CHECKDAY, IDFOODS, AMOUNT)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| IDBILL(FK) | INT |  |
| CHECKDAY | DATE |  |
| IDFOODS(FK) | INT |  |
| AMOUNT | INT |  |

SUPLIER (**SUPLIERID,** SUPLIERNAME, SUPLERADDRESS, PHONENUMBER)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **SUPLIERID(PK)** | VARCHAR | 10 |
| SUPLIERNAME | NVARCHAR | 50 |
| PHONENUMBER | VACHAR | 15 |
| SUPLIERADDRESS | NVARCHAR | 50 |

RESOURCETYPE (**ID,** RESOURCETYPENAME, SUPLERID)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| RESOURCETYPENAME | NVARCHAR | 30 |
| SUPLIERID(FK) | VACHAR | 10 |

RESOURCE (**ID,** RESOURCESNAME, RESOURCETYPEID , RESOURCEAMOUNT )

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| RESOURCENAME | NVARCHAR | 50 |
| RESOURCETYPEID (FK) | INT |  |
| RESOURCEAMOUNT | INT |  |

INCOME (**ID,** DATEINCOME, COLLECT, PROFIT)

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Độ dài |
| **ID(PK)** | INT |  |
| DATEINTCOME | DATE |  |
| COLLECT | FLOAT |  |
| PROFIT | FLOAT |  |

**CHƯƠNG 4. GIAO DIỆN, MENU CHƯƠNG TRÌNH**

*4.1. Giao diện và menu chương trình:*

*4.1.1. Giao diện đăng nhập hệ thống:*

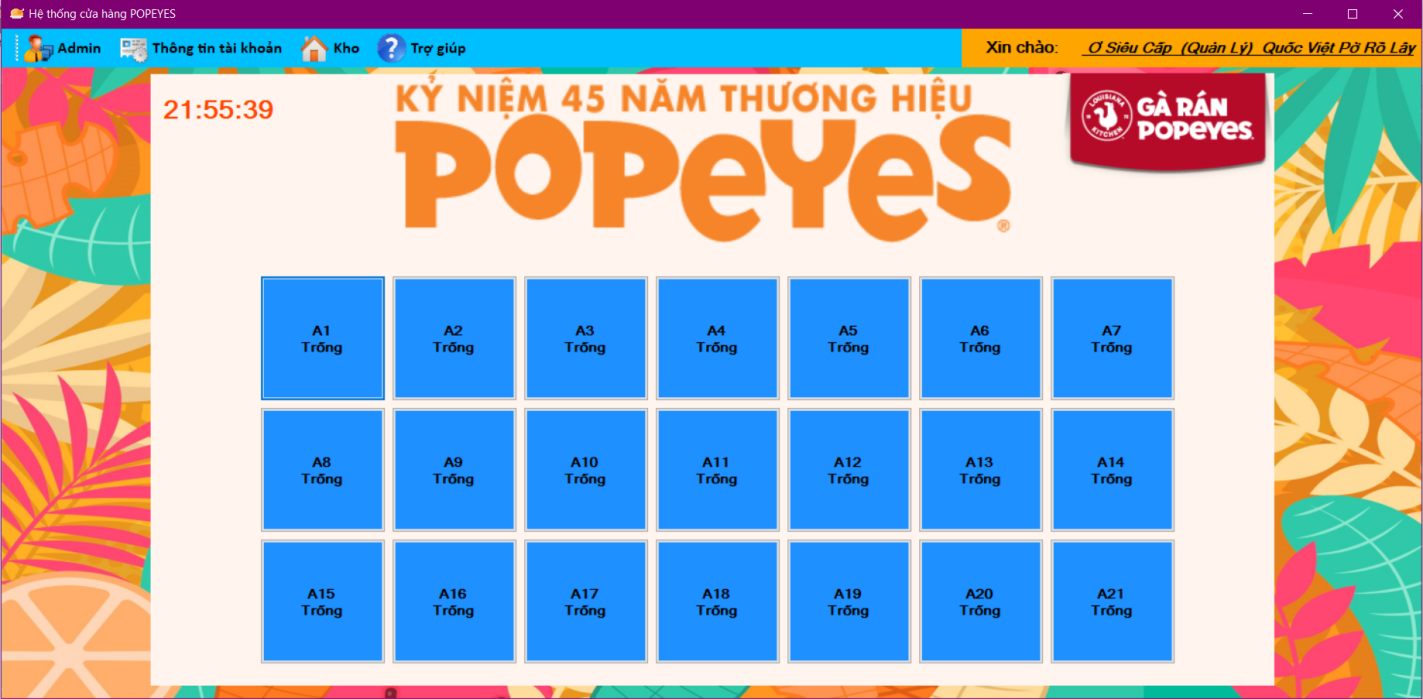


- Giao diện dùng để người dùng đăng nhập tài khoản của mình để kết nối tới hệ thống , ô mật khẩu và tên đăng nhập có phân biệt kí tự hoa thường.

- Khi nhập đúng tài khoản và mật khẩu chọn đăng nhập để hệ thống xác minh, xác thực chính xác bạn sẽ được chuyển đến trang giao diện làm việc.

- Khi hệ thống xác thực nhập sai sẽ hiện lên thông báo “sai tên đăng nhập hoặc mật khẩu , vui long thử lại!”

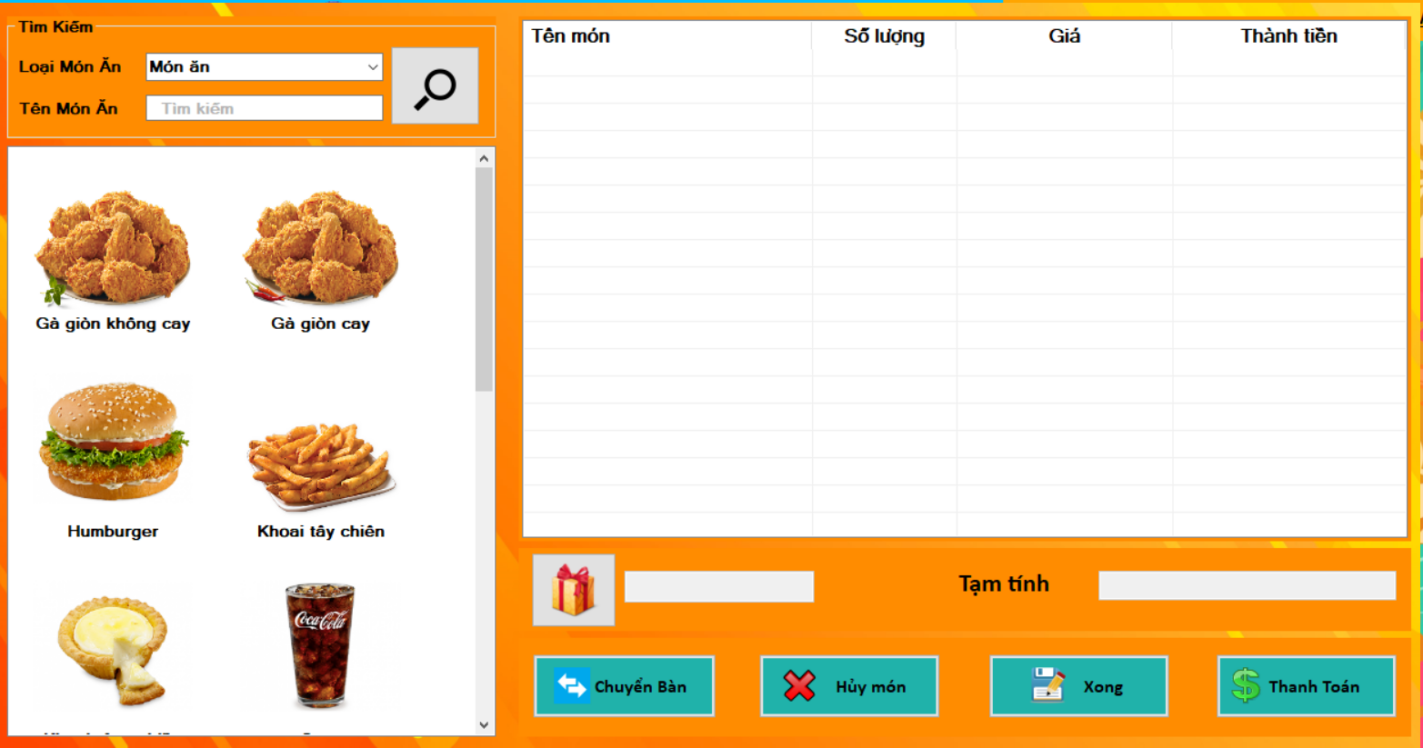
*4.1.2. Giao diện đặt bàn:*



-Giao diện này dùng để nhân viên có thể quản lý các tình trạng bàn còn trống hoặc có người để sắp xếp khác ngồi hợp lý nhất .

- Những bàn hiên đang có người sẽ được đánh dấu mày xám và tình trạng hóa đơn là chưa thanh toán. Bàn không có hóa đơn hoặc món sẽ đánh dấu màu xanh.

*4.1.3. Giao diện đặt món:*



Giao diện này gồm chức năng:

• Thêm (thêm món ăn)

• Xoá ( xoá món ăn)

• Nhập khuyến mãi giảm giá

• Thanh toán .

- Giao diện đặt món giúp nhân viên dễ dàng xác định món ăn khách yêu cầu thông qua một menu tích hợp bên trái , khi click chọn , món ăn sẽ được tự động cập nhật vào trong hóa đơn , nếu món đó đã có trong hóa đơn thì số lượng sẽ được cập nhật.

- Chức năng nhằm giúp cho nhân viên có thể nhập món ăn của khách hàng yêu cầu vào trong 1 danh sách và lưu giữ nó , hệ thống sẽ tự động tính tiền hóa đơn đó trong mục tạm tính.

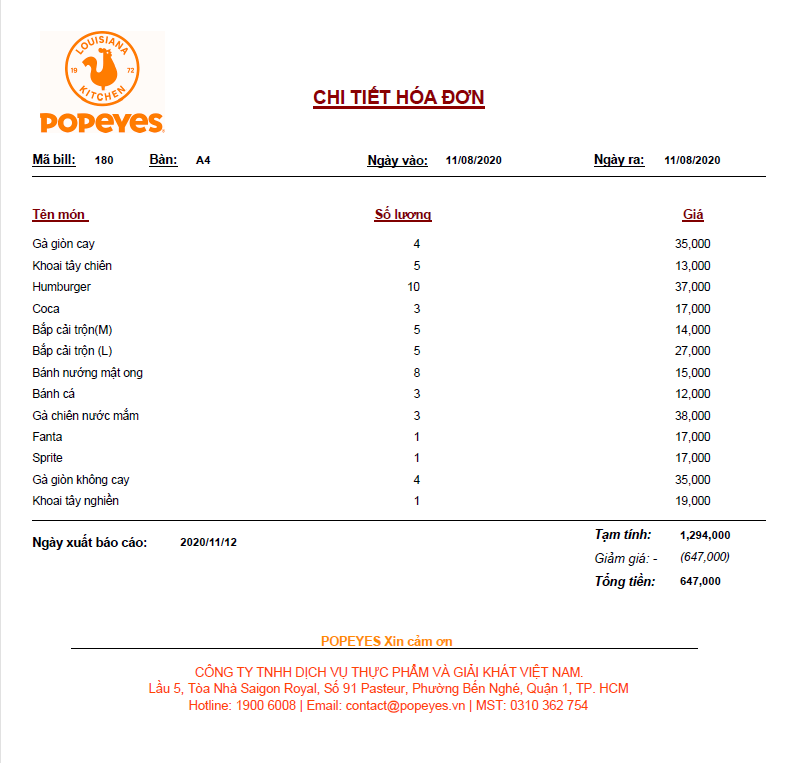
- Các món hủy sẽ bị xóa khỏi hóa đơn và cập nhật lại số tiền phải thanh toán.

- Khuyến mãi giúp giảm giá hóa đơn theo phần trăm, khi chọn hộp quà nhân viên phụ trách sẽ được chuyển đến trang nhập khuyến mãi , khuyến mãi sẽ được tính tối thiểu là 0% và tối đa là 100%. Khi khi điền giá trị và bấm “Xong” thì sẽ quay lại trang đặt món với số tiền khuyến mãi đã áp dụng.

- Để tránh trường hợp áp dụng sẵn và sau đó lại tiếp tục áp dụng nữa thì phần mềm đã khắc phục bằng cách chỉ khi nào chọn và thanh toán luôn thì giá trị mới cập nhật.

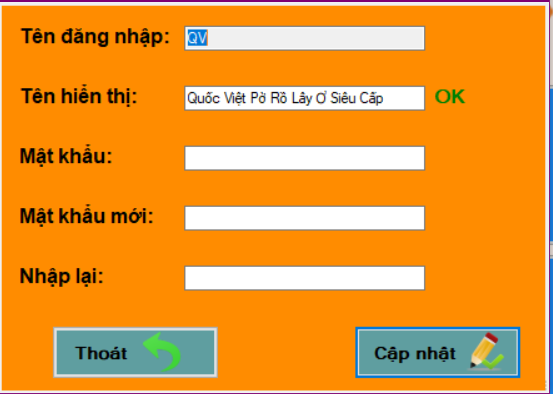
- Chọn thanh toán để thanh toán và in hóa đơn cho khách hàng.

*4.1.4. Giao diện hóa đơn demo:*



- Giao diện hóa đơn cho phép khách hàng xem các thuộc tính cơ bản như: mã hóa đơn, số bàn, ngày vào , ngày rời đi, món ăn, số lượng, số tiền giảm giá . Sau khi xuất hóa đơn khách hàng có thể cầm để kiểm tra và xác nhận thanh toán.

*4.1.5. Giao diện thông tin cá nhân:*



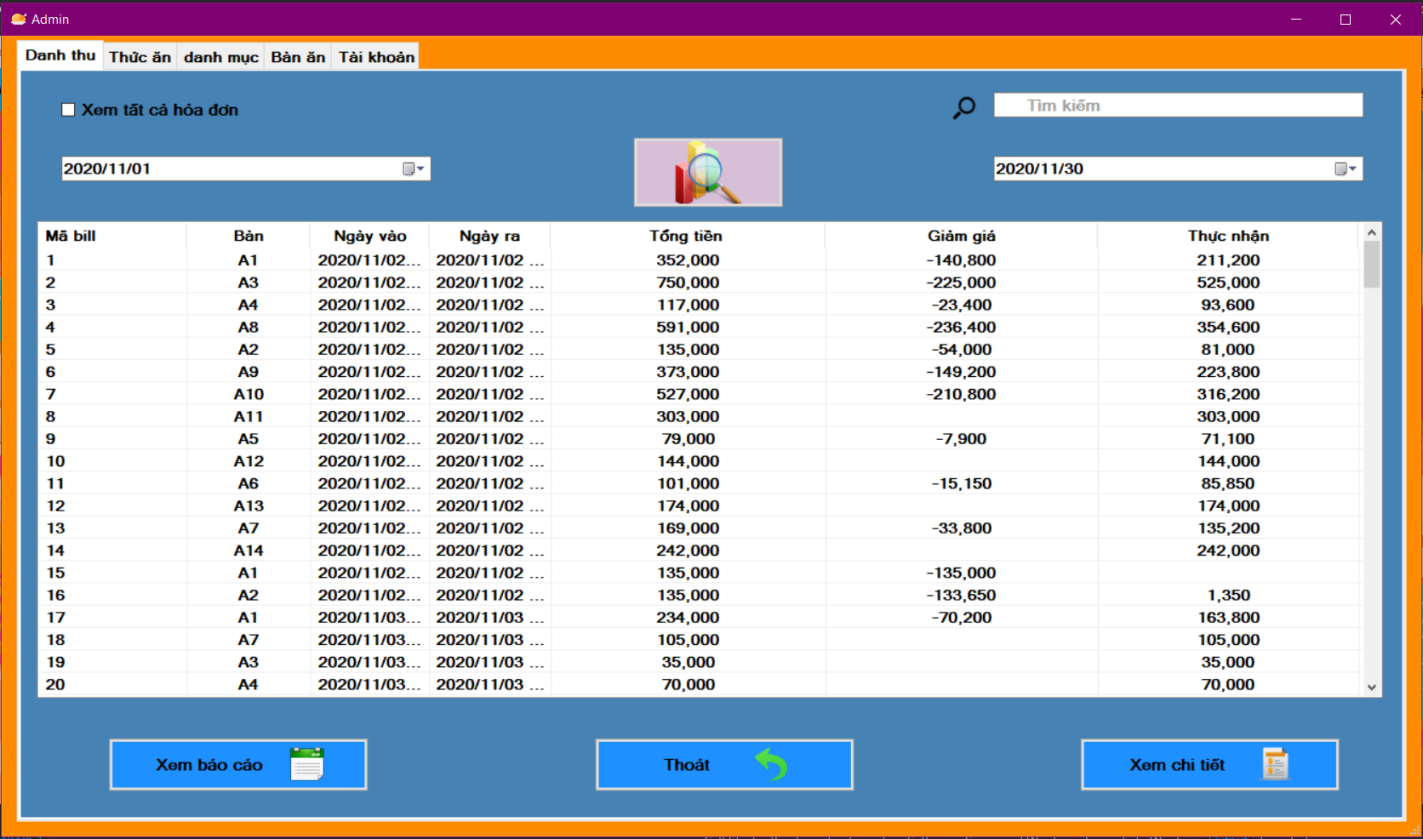
- Giao diên này cho phép người dùng thay đổi nhanh thông tin cá nhân của mình từ tên hiển thị , mật khẩu nhưng không thể thay đổi tên đăng nhập của mình , nếu lỡ quên mật khẩu nhân viên phải liên hệ với quản lý để được cấp mật khẩu mới.

- Với phần tên hiển thị chỉ quy định tối đa 30 kí tự , khi đủ sẽ không thể nhập thêm nữa và sẽ xuất hiện dòng chữ “quá dài bên cạnh” ngược lại sẽ hiển thị “ok” tức là tên hợp lệ .

- Mật khẩu mới sẽ quy định 30 kí tự và khi nhập quá cũng sẽ không thể nhập thêm được nữa.

- Nhập lại được tích hợp để xác nhận mật khẩu mới vì vậy phải nhập chính xác mật khẩu mới ở ô bên trên , khi nhập đúng thì sẽ thông báo “ok”, ngược lại sẽ tick “X”

*4.1.6. Giao diện doanh thu:*



- Với giao diện này, quản lý sẽ sễ dàng nắm thông tin của tất cả các hóa đơn đã xuất từ mã hóa đơn , bàn đặt , ngày vào , ngày ra … với mức độ chính xác tương đối.

- Nói tương đối là bởi khi cập nhật giá món ăn , xóa món ăn, xóa danh mục thì tất cả các thành phần bên trong sẽ bị xóa theo nhưng ở 3 cột :tổng tiền, giảm giá, thực nhân giá trị không thay đổi.

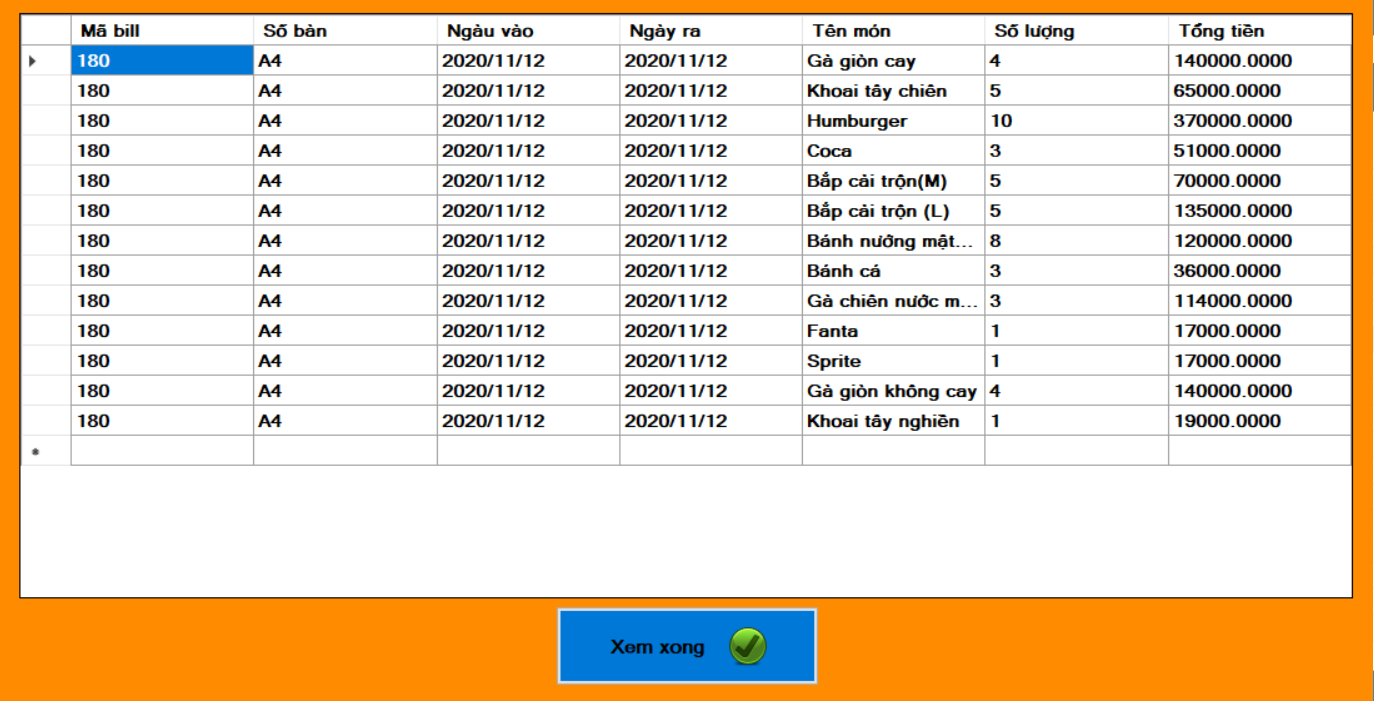
- Khi đang chọn 1 đối tượng mà muốn xem thông tin chi tiết thì chọn xem chi tiết.

- Cần xem báo cáo tổng quát chọn xem báo cáo .

- Chức năng xóa Doanh thu các tháng khác sẽ tự động xóa hết các doanh thu của các tháng trước đó và chỉ giữ lại doanh thu trong tháng

- Chọn thoát để rời khỏi.

*4.1.7. Giao diện chi tiết hóa đơn:*

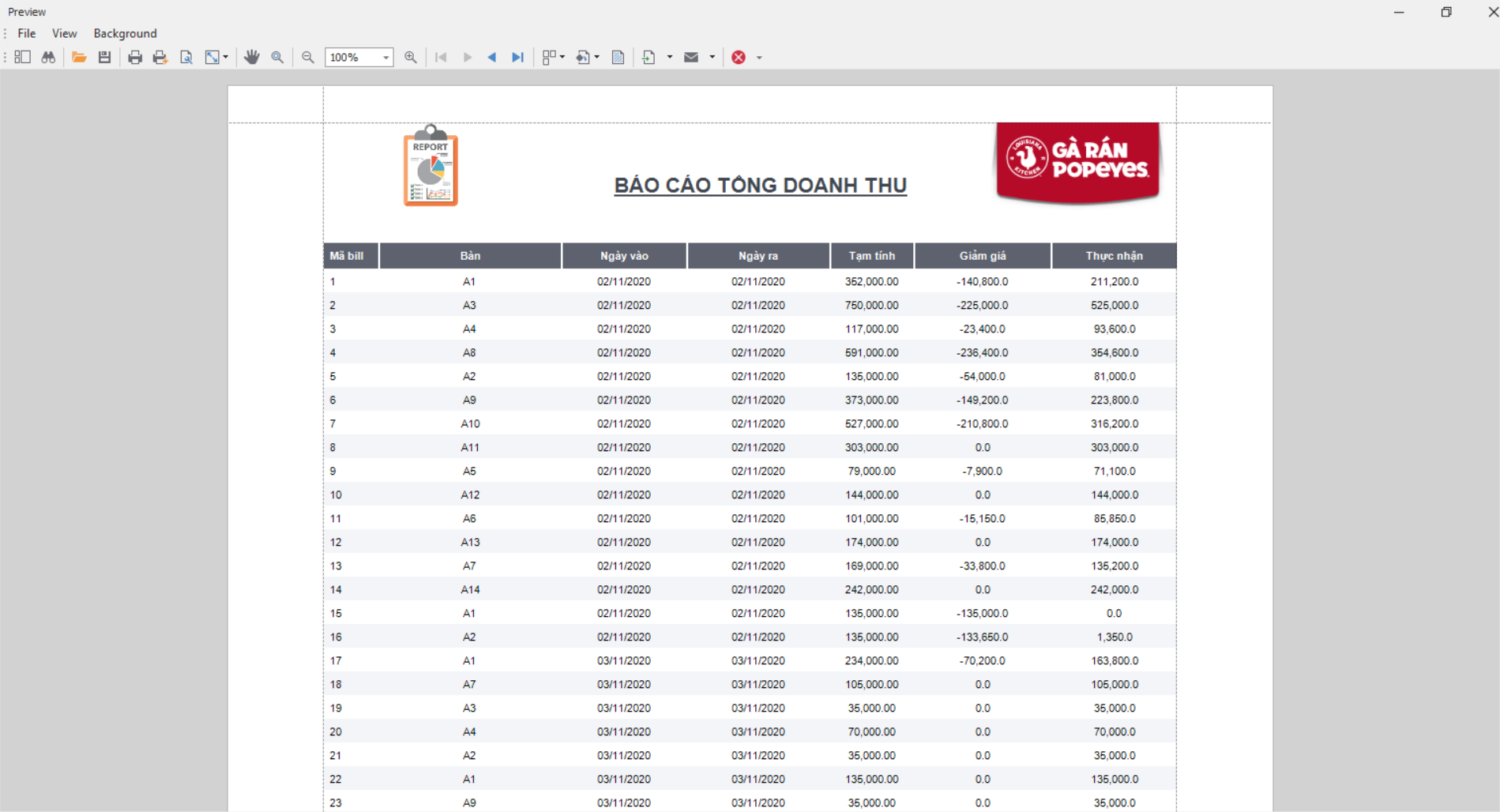


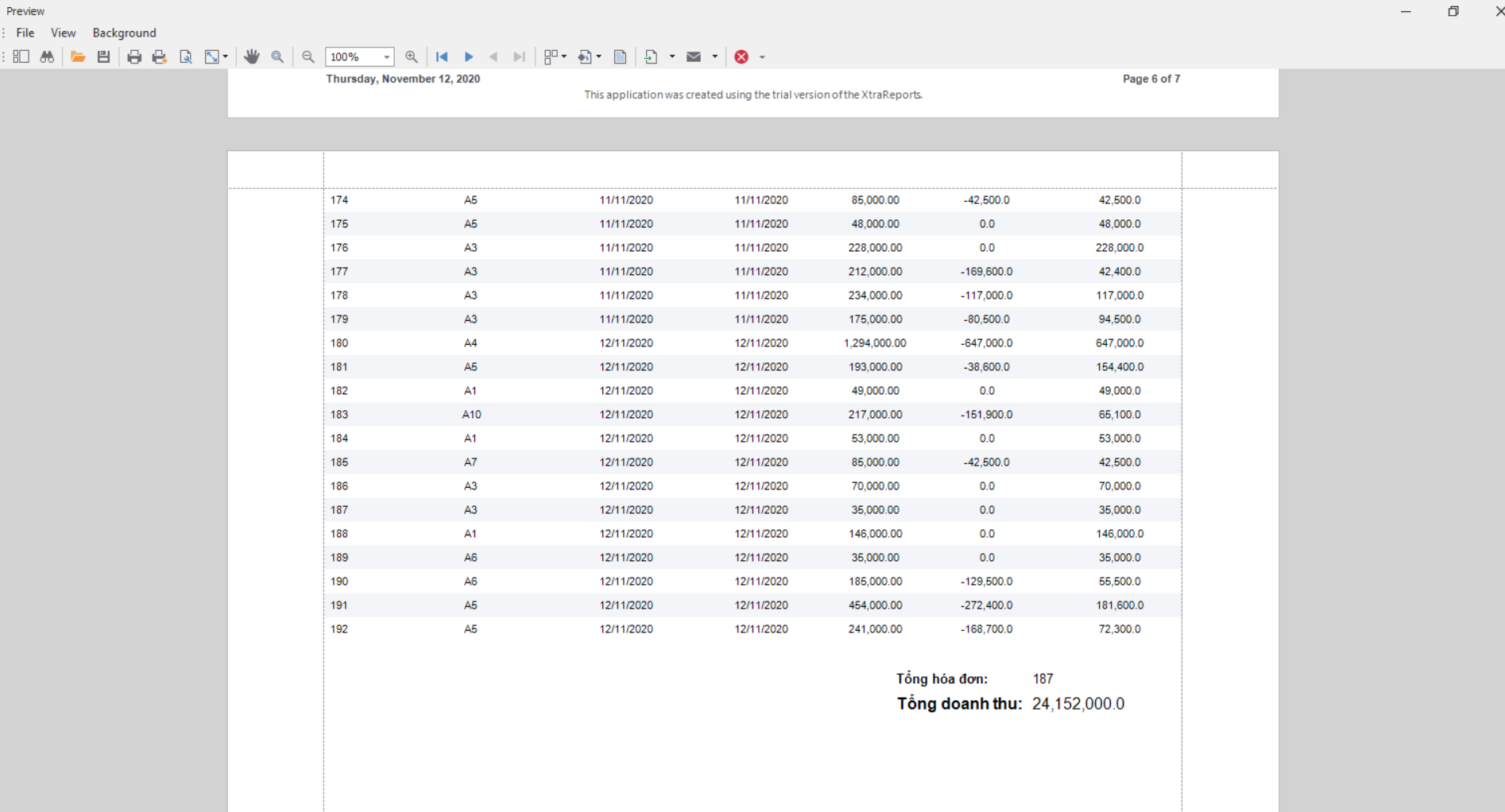
- Giao diện này cung cấp số liệu chi tiết về các món ăn đã đặt , số lượng và tổng g iá tiền các món ăn đó.

- Cung cấp tính năng xuất lại hóa đơn thanh toán khi muốn xuất lại một hóa đơn khi khách yêu cầu .

- Chọn xem xong để quay lại trang thống kê.

*4.1.8. Giao diện báo cáo tổng quát doanh thu:*

**

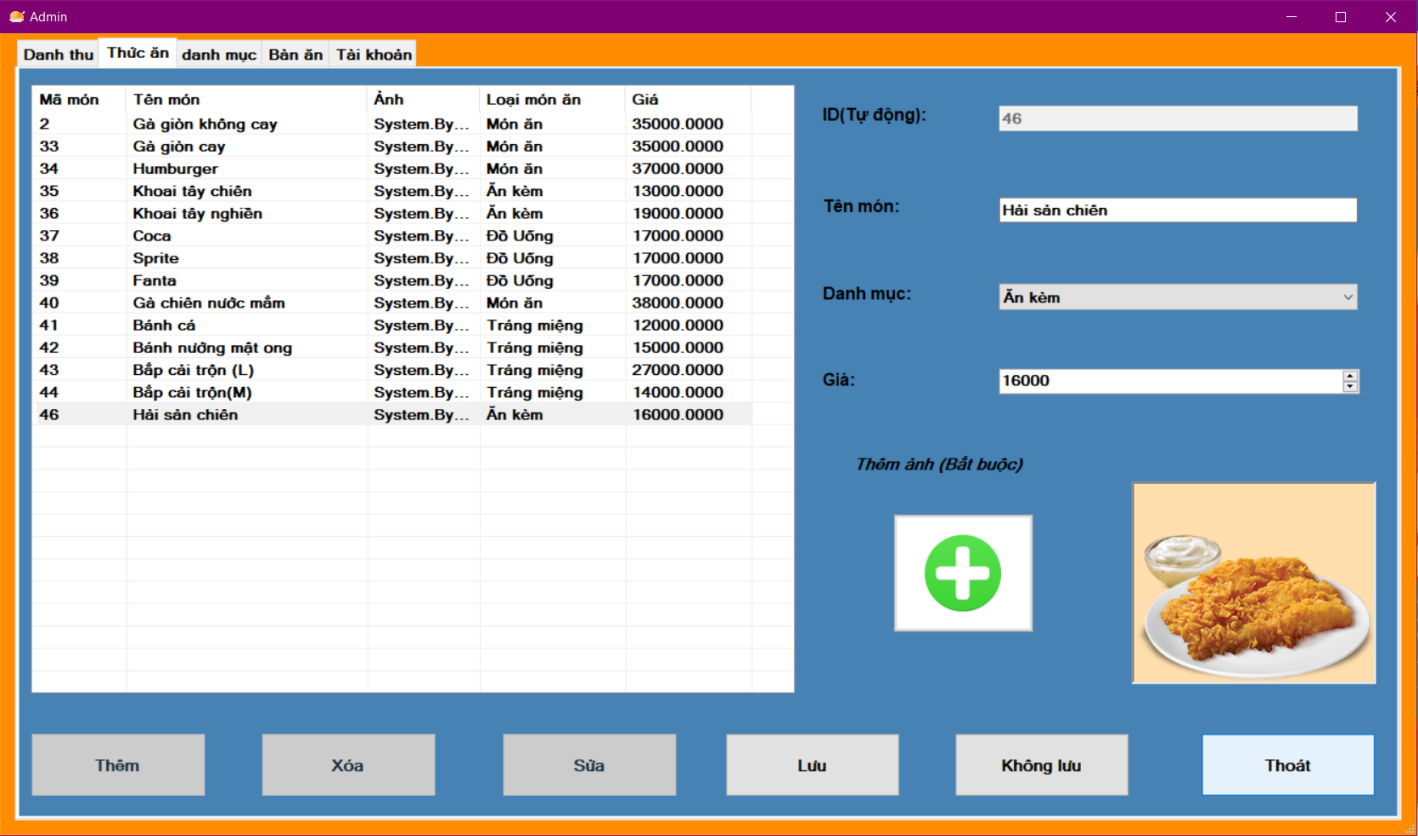
**

- Giao diện này cho phép xuất các hóa đơn dưới dạng báo cáo. Có các parameter giúp việc thống kê ra các báo cáo có cùng ngày mà người dùng muốn truy xuất.Không những thế , báo cáo Có thể in , xuất dưới dạng excel , word , pdf…để lưu trữ.

- Đếm tổng hóa đơn đã xuất trong suốt thời gian kinh doanh.

- Tính tổng doanh thu có được trong thời gian kinh doanh.

*4.1.9. Giao diện tác vụ món ăn:*

**

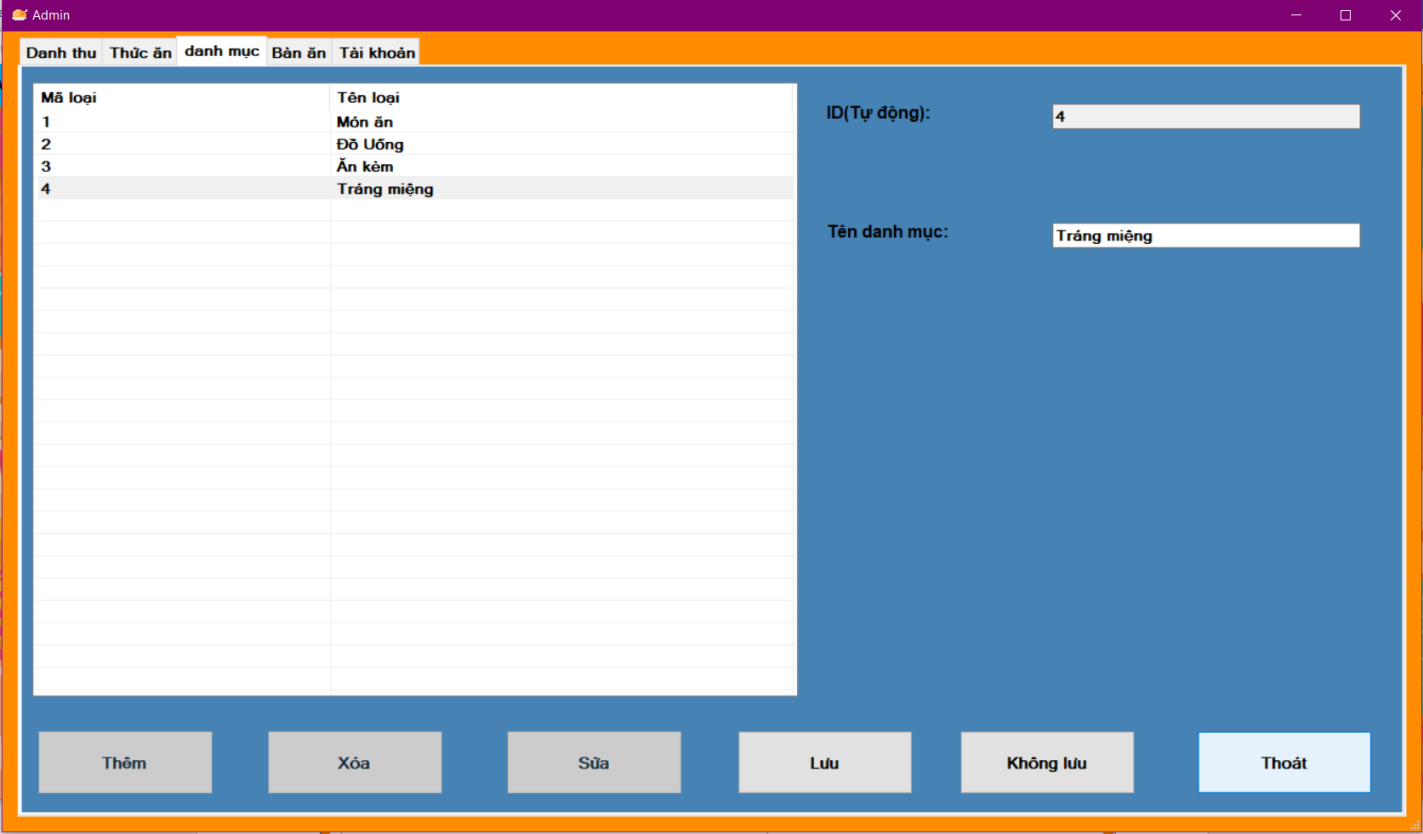
- Giao diện cho phép quản lý thêm, xóa, sửa món ăn .

- Ở mục này không được để trống bất cứ thứ gì (Trừ mã id tăng tự động) và mở file chọn hình ảnh minh họa cho món ăn được định dạng với png , jpeg.

- Chọn lưu để hoàn tất việc thêm hoặc sửa.

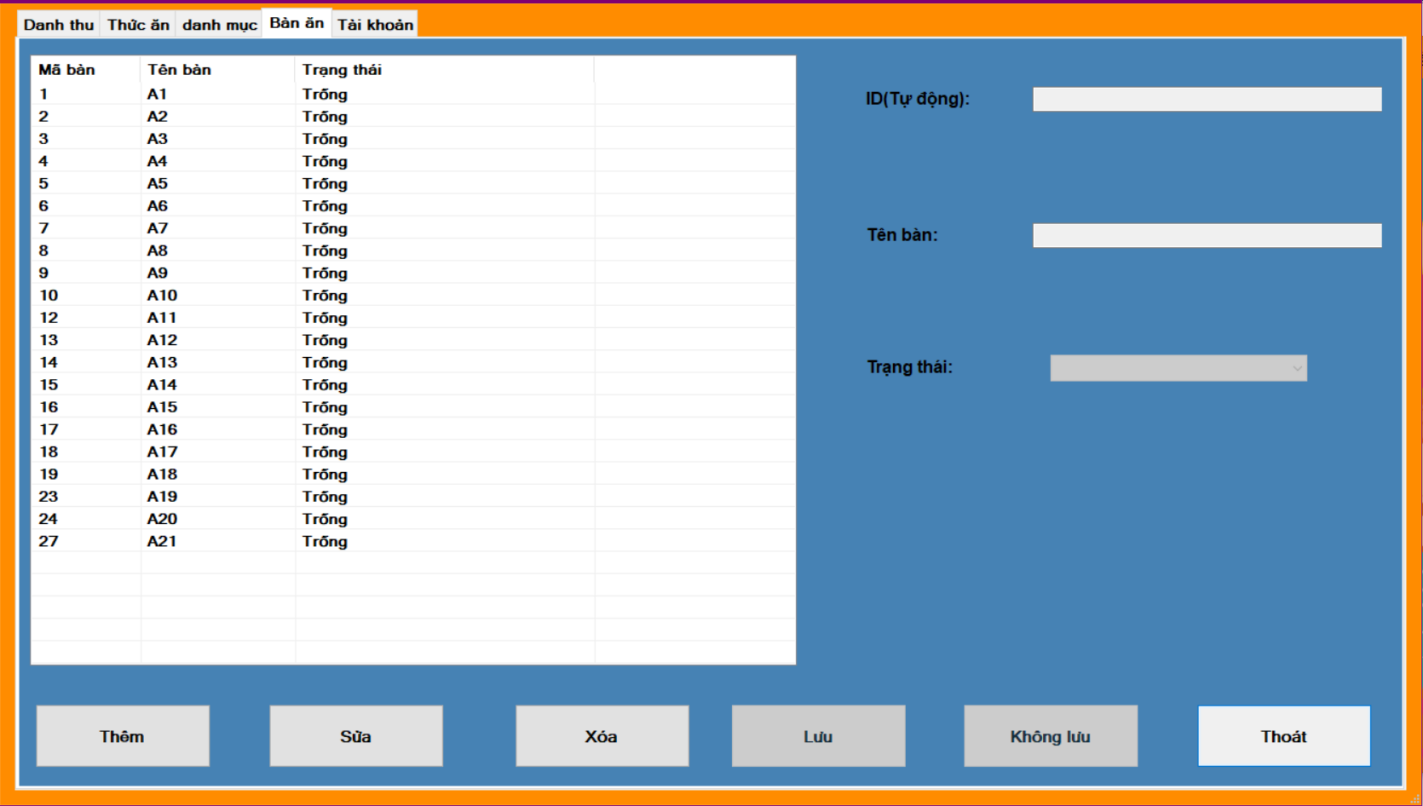
- Không lưu để hủy tác vụ và thoát để rời khỏi.

*4.1.10. Giao diện tác vụ danh mục:*



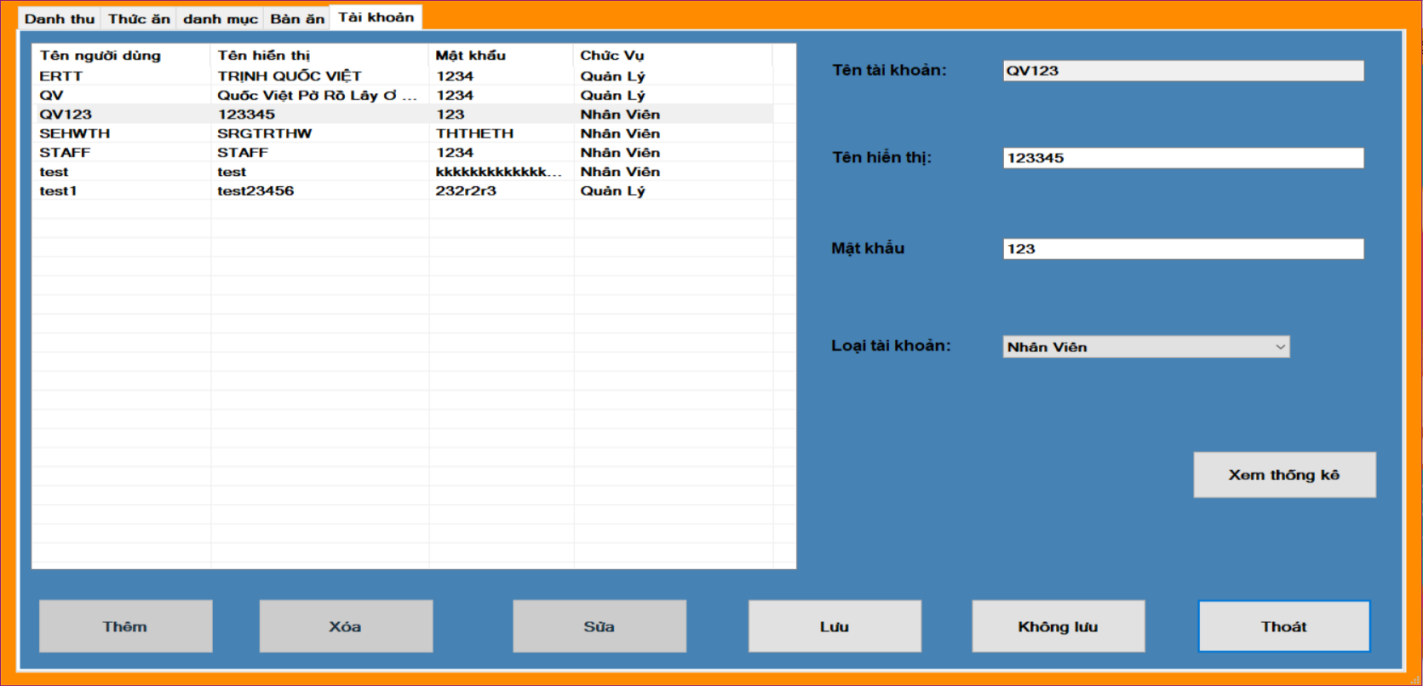
- Giao diện này cung cấp các tác vụ cơ bản: thêm danh mục, xóa danh mục, sửa danh mục. Tất cả mã được tăng tự động khiến công cuộc thêm xóa sửa vô cùng đơn giản và thân thiện .

*4.1.11. Giao diện tác vụ bàn ăn:*

**

- Giao diện này cung cấp các tác vụ cơ bản: thêm bàn, xóa bàn, sửa bàn. Tất cả mã được tăng tự động khiến công cuộc thêm xóa sửa vô cùng đơn giản và thân thiện .

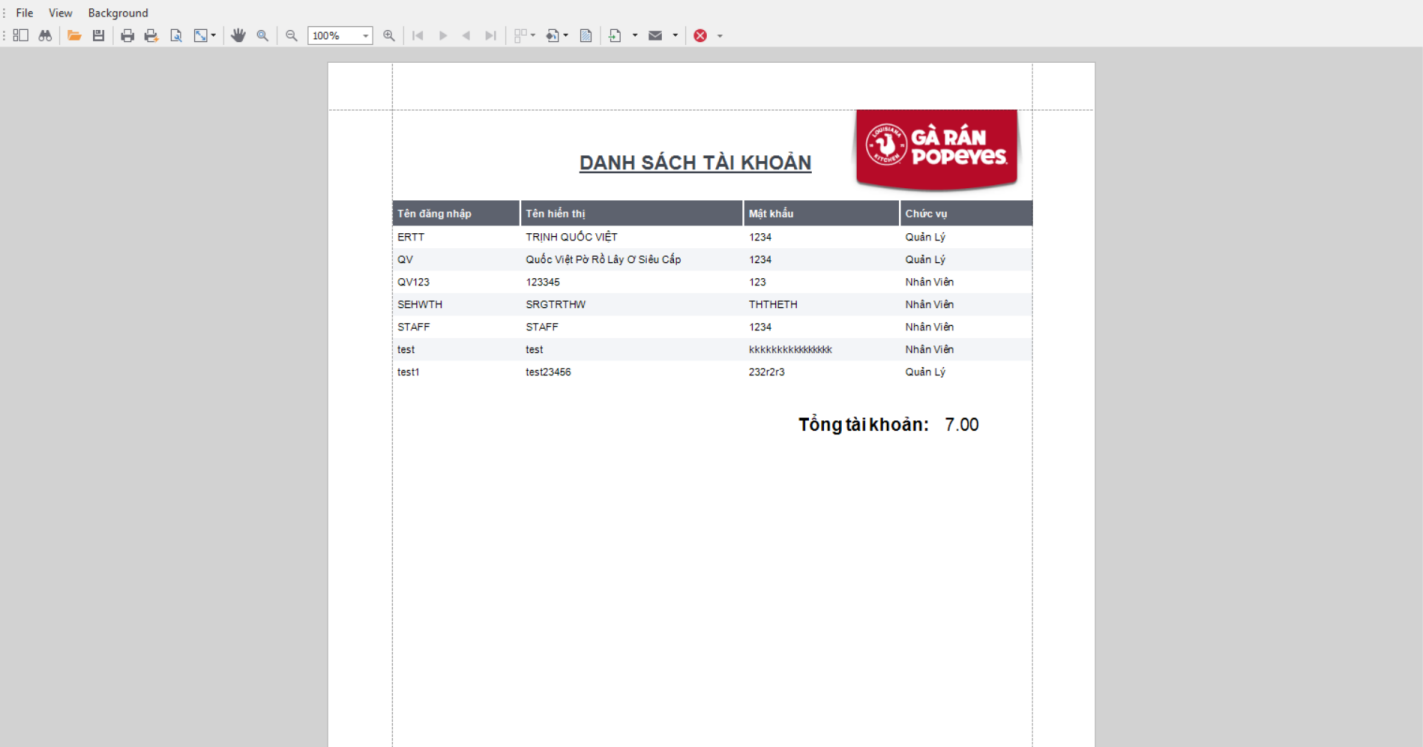
*4.1.12. Giao diện tác vụ tài khoản:*

**

- Giao diện tác vụ quản lý tài khoản cho phép người dùng chức vụ quản lý thêm tài khoản, xóa tài khoản, sửa tài khoản, với những tác vụ nư vậy dễ thấy nhất đó là quản lý có thể tạo ra tài khoản cho 1 nhân cố định, thay đổi và cấp lại mật khẩu nếu nhân viên vô tình quên mất, thăng tiến chức vụ trong mục loại tài khoản…Cũng giống như những tác vụ khác, chọn nút Lưu để lưu lại tất cả thay đổi khi sửa hoặc thêm 1 tài khoản mới (điều kiện ràng buộc: không được để trống bất kì thuộc tính nào trừ ID ), chọn Không lưu để hủy tác vụ, chọn Thoát để rời khỏi.

- Là mục quan trọng vì vậy phần mềm đã tích hợp báo cáo tài khoản ở mục xem thống kê.

*4.1.13. Giao diện báo cáo tài khoản:*

**

- Giao diện được thiết kế để thống kê toàn bộ tài khoản bao gồm: Tên đăng nhập, tên hiển thị, mật khẩu , chức vụ .

* Cuối danh sách là tổng các tài khoản đã và đang có đến thời điểm hiện tại.

*4.1.14. Giao diện trợ giúp (File help):*

**

- Giao diện này là công cụ trợ giúp khi mới làm quen phần mềm hoặc có thắc mắc trong quá trình sử dụng và khám phá các chức năng mới mẻ.

- Tìm hiểu về các chức năng và làm sao để có chất lượng chạy chương trình nhanh nhất mà không có lỗi xuất hiện khi chạy.

**CHƯƠNG 5. KẾT LUẬN VÀ MỤC TIÊU MỞ RỘNG CỦA ĐỒ ÁN:**

*5.1 Mục tiêu mở rộng của đồ án:*

- Hoàn thành chức năng nhập nguyên liệu có quản lý theo nhà cung cấp, khối lượng nhập , loại.

- Hoàn thành chức năng tính lợi nhuận hằng năm.

- Liên kết chức năng định danh , quản lý giờ làm của nhân viên .

- Quản lý kho đông lạnh và nguyên liệu.

- Hoàn thiện hơn những chức năng đang còn thiếu trong quá trình phát triển

- Sửa lỗi , tìm ra những phương pháp mới để cải thiện hiệu xuất , tốc độ xử lý.

*5.2 Kết luận :*

- Là đồ án tâm huyết vì vậy nhóm của chúng em đã tham khảo rất nhiều bài viết, các kĩ thuật và phương pháp được sàng lọc kĩ trước khi đưa vào đồ án . Cũng vì chưa tiếp xúc thường xuyên với lập trình giao diện và sự kiện cho nên đồ án có sai sót là điều không thể tránh khỏi ,chỉ mong Thầy và các bạn sẽ là người sử dụng thực tế để có thể chỉ ra yếu điểm và những cách xử lý chưa hợp lý bên trong đồ án để chúng em có thể khắc phục nhược điểm của mình.

*5.3 Các nguồn tài liệu tham khảo:*

<https://stackoverflow.com>

<https://www.it-swarm-vi.tech/>

<https://quantrimang.com/>

<https://icon-icons.com/>

<https://icons.iconarchive.com>

<https://www.ddth.com/>

<https://thichchiase.com/programing/c/tong-hop-cac-ham-xu-ly-chuoi-trong-lap-trinh-c/>

<https://www.howkteam.vn/course/tao-report-trong-phan-mem-quan-ly-quan-cafe-voi-c-winform/>

<https://daynhauhoc.com/>

<https://comdy.vn/c-sharp/tinh-dong-goi-trong-c-sharp/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HẾT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_