UBND TỈNH BÌNH DƯƠNG

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

****

**BÁO CÁO TIỂU LUẬN**

HỌC PHẦN:

LẬP TRÌNH ỨNG DỤNG TRÊN WINDOWS

**TÊN ĐỀ TÀI**

**XÂY DỰNG ỨNG DỤNG WINDOWS**

**QUẢN LÝ QUÁN ĂN VẶT WVN**

***GIẢNG VIÊN HƯỚNG DẪN :***

*ĐẶNG NHƯ PHÚ*

***SINH VIÊN THỰC HIỆN:***

*TRẦN MINH QUÂN - 1824801030037*

*NGUYỄN HOÀI NAM - 1824801030027*

*ĐỖ THỊ HỒNG NHUNG - 1824801030045*

***Bình Dương, tháng 4 năm 2021***

*Mục lục*

[DANH SÁCH THÀNH VIÊN 1](#_Toc71110471)

[CHƯƠNG 1.CƠ SỞ LÝ THUYẾT 2](#_Toc71110472)

[1.1 Tổng quan 2](#_Toc71110473)

[.NET Framework là gì ? 2](#_Toc71110474)

[1. Thời gian chạy ngôn ngữ chung (Common Language Runtime) 3](#_Toc71110475)

[2. Class Library 4](#_Toc71110476)

[3. Ngôn ngữ 4](#_Toc71110477)

[1.2 Các thành phần đối tượng hỗ trợ xây dựng giao diện trên Windows 6](#_Toc71110478)

[1.3 Cơ sở dữ liệu SQL Server 7](#_Toc71110479)

[CHƯƠNG 2.PHÂN TÍCH HỆ THỐNG 8](#_Toc71110480)

[2.1 Ngôn ngữ lập trình,cơ sở dữ liệu và môi trường ứng dụng 8](#_Toc71110481)

[2.2 UseCase 13](#_Toc71110482)

[2.3 Phân tích và đặc tả các chức năng của chương trình 3](#_Toc71110483)

[2.3.1 Đặc tả use case Đăng nhập 3](#_Toc71110484)

[2.3.2 Đặc tả use case Chọn bàn 3](#_Toc71110485)

[2.3.3 Đặc tả use case Quản lý Đặt món 4](#_Toc71110486)

[2.3.4 Đặc tả use case Quản lý khu vực 6](#_Toc71110487)

[2.3.4 Đặc tả use case Quản lý Bàn 9](#_Toc71110488)

[2.3.4 Đặc tả use case Quản lý Danh mục thực phẩm 11](#_Toc71110489)

[2.3.4 Đặc tả use case Quản lý Thực phẩm 14](#_Toc71110490)

[2.3.4 Đặc tả use case Quản lý Tài khoản 17](#_Toc71110491)

[2.4 Sơ đồ, cấu trúc chương trình 20](#_Toc71110492)

[CHƯƠNG 3.THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG 21](#_Toc71110493)

[3.1 Thiết kế chức năng và Giao diện 21](#_Toc71110494)

[3.1.1 Đăng nhập 21](#_Toc71110495)

[3.1.2 Chọn Bàn 21](#_Toc71110496)

[3.1.3 Quản lý đặt món 23](#_Toc71110497)

[3.1.4 Quản lý Khu vực 23](#_Toc71110498)

[3.1.5 Quản lý Bàn 24](#_Toc71110499)

[3.1.6 Quản lý Danh mục thực phẩm 25](#_Toc71110500)

[3.1.6 Quản lý Thực phẩm 26](#_Toc71110501)

[3.1.6 Quản lý Tài khoản 27](#_Toc71110502)

[CHƯƠNG 4.THỰC NGHIỆM VÀ TRIỂN KHAI 29](#_Toc71110503)

[4.1. Đóng gói và xuất bản chương trình 29](#_Toc71110504)

[4.2. Demo các chức năng của chương trình 29](#_Toc71110505)

[CHƯƠNG 5.KẾT LUẬN 37](#_Toc71110506)

[5.1 Kết quả đạt được 37](#_Toc71110507)

[5.2 Hướng phát triển 37](#_Toc71110508)

[TÀI LIỆU THAM KHẢO 38](#_Toc71110509)

# DANH SÁCH THÀNH VIÊN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **MSSV** | **Nội dung thực hiện** |
| 1 | Trần Minh Quân | 1824801030037 | - Lên ý tưởng.  - Thiết kế giao diện.  - Thiết kế cơ sở dữ liệu.  - Xử lý code hiện thực hóa.  -Viết báo cáo. |
| 2 | Nguyễn Hoài Nam | 1824801030027 | - Lên ý tưởng.  - Cung cấp thông tin tạo cơ sở dữ liệu.  - Thiết kế giao diện. |
| 3 | Đỗ Thị Hồng Nhung | 1824801030045 | - Lên ý tưởng.  - Tạo cơ sở dữ liệu.  - Thiết kế giao diện. |

# CHƯƠNG 1.CƠ SỞ LÝ THUYẾT

## Tổng quan

### ****.NET Framework là gì ?****

[.NET framework](https://bizflycloud.vn/tin-tuc/net-framework-la-gi-cac-khai-niem-co-ban-ve-net-framework-20180912090957839.htm) là một nền tảng phát triển phần mềm do Microsoft phát triển. Framework này tạo ra các ứng dụng chạy trên nền tảng Windows. Phiên bản đầu tiên của .NET framework được phát hành vào năm 2002. Phiên bản đó được gọi là .Net framework 1.0. .NET framework đã đi một chặng đường dài kể từ đó, và phiên bản hiện tại là 4.7.1.

.NET framework có thể được sử dụng để tạo cả những ứng dụng dựa trên biểu mẫu (Form-based) và dựa trên Web (Web-based). Các web service cũng có thể được phát triển bằng cách sử dụng .NET framework. Framework cũng hỗ trợ các ngôn ngữ lập trình khác nhau như Visual Basic và C#. Vì vậy, các nhà phát triển có thể lựa chọn ngôn ngữ để phát triển ứng dụng cần thiết.

Ngày xửa ngày xưa, khi nhắc tới [**C#**](https://toidicodedao.com/tag/c-hay-ho/), người ta nghĩ ngay tới .NET Framework của Microsoft. C# hay bị người đời ném đá bởi vì nó đi kèm với .NET Framework chỉ hoạt động trên Windows. Ngoài ra, để viết ứng dụng C#, lập trình viên còn phải làm việc với 2 thằng vừa nặng vừa mắc tiền là Visual Studio và Microsoft SQL Server. Hai thằng này cũng … chỉ hoạt động trên Windows nốt.

Trước đây, để chạy C# trên các hệ điều hành khác, ta có thể sử dụng [**Mono**](http://www.mono-project.com/docs/about-mono/languages/csharp/)**.** Tuy nhiên, đây không phải là 1 dự án chính thức của Microsoft. Developer vẫn không có IDE C# nào ngon lành, cũng chẳng có SQL Server nào đi kèm.

Sử dụng C#.NET đồng nghĩa với việc phải dùng toàn bộ hệ sinh thái (tốn tiền) của Microsoft!! Vì lẽ đó, C# tiếp tục trùm chăm đắp chiếu trên Windows, chịu sự ghẻ lạnh của cộng đồng developer dùng Linux, MacOS).

**.NET Framework ngày nay**

Microsoft bắt đầu mở cửa dần dần. Thay vì đóng im ỉm như trước, Microsoft bắt đầu open source .NET Framework, ASP.NET MVC, Entity Framework để thu hút cộng đồng lập trình viên Open Source.

.NET Core đã bắt đầu ra những bản beta từ năm ngoái, và bản 1.0 vừa được chính thức ra mắt cách đây vài hôm.

Giờ đây, việc lập trình C# trở nên nhẹ nhàng rất nhiều. Ta chỉ cần cài đặt .NET Core (nhẹ), Visual Studio Code (cũng nhẹ). Tất cả đều free nhé. Trong thời gian sắp tới thì Microsoft sẽ đưa SQL Server 2016 lên Linux luôn.

**.NET Framework bao gồm những gì ?**

### 1. Thời gian chạy ngôn ngữ chung (Common Language Runtime)

"Cơ sở hạ tầng ngôn ngữ chung" hoặc CLI là một nền tảng mà trên đó các chương trình .Net được thực thi.

**CLI có các tính năng chính sau đây:**

- Xử lý ngoại lệ

Ngoại lệ là các lỗi xảy ra khi ứng dụng được thực hiện.

Ví dụ về ngoại lệ là:

Nếu một ứng dụng cố mở một tệp trên local machine, nhưng tệp không có.

Nếu ứng dụng tìm nạp một số bản ghi từ cơ sở dữ liệu, nhưng kết nối đến cơ sở dữ liệu không hợp lệ.

- Garbage Collection

Garbage Collection là quá trình loại bỏ các tài nguyên không mong muốn khi chúng không còn cần thiết nữa.

Ví dụ về Garbage Collection là:

Một xử lý tập tin không còn cần thiết: Nếu ứng dụng đã hoàn thành tất cả các thao tác trên một tệp, thì trình xử lý tệp có thể không còn cần thiết nữa.

Kết nối cơ sở dữ liệu không còn cần thiết nữa: Nếu ứng dụng đã hoàn thành tất cả các hoạt động trên cơ sở dữ liệu, thì kết nối cơ sở dữ liệu có thể không còn cần thiết nữa.

- Làm việc với các ngôn ngữ lập trình khác nhau

Như đã nêu ở phía trên, một developer có thể phát triển một ứng dụng bằng nhiều ngôn ngữ lập trình .Net.

Ngôn ngữ: Cấp độ đầu tiên là ngôn ngữ lập trình, phổ biến nhất là VB.Net và C #.

Trình biên dịch: Có một trình biên dịch sẽ được tách riêng cho từng ngôn ngữ lập trình. Vì vậy, bên dưới ngôn ngữ VB.Net, sẽ có một trình biên dịch VB.Net riêng biệt. Tương tự, đối với C #, bạn sẽ có trình biên dịch khác.

Common Language Interpreter: Đây là lớp cuối cùng trong .Net sẽ được sử dụng để chạy một chương trình .net được phát triển bằng bất kỳ ngôn ngữ lập trình nào. Vì vậy trình biên dịch tiếp theo sẽ gửi chương trình tới lớp CLI để chạy ứng dụng .Net.

### 2. Class Library

.NET Framework bao gồm một bộ các class libraries. Một class library là một tập hợp các phương thức và các hàm có thể được sử dụng cho mục đích cốt lõi.

Ví dụ, có một class library với các phương thức để xử lý tất cả các hoạt động cấp tệp. Vì vậy, có một phương pháp có thể được sử dụng để đọc văn bản từ một tập tin. Tương tự, có một phương pháp để viết văn bản vào một tập tin.

Hầu hết các phương thức được chia thành các vùng tên System. \*Hoặc Microsoft.\*. (Dấu hoa thị \* chỉ có nghĩa là tham chiếu đến tất cả các phương thức nằm trong vùng tên System hoặc Microsoft). Một namespace là một sự tách biệt logic của các phương thức.

### 3. Ngôn ngữ

Các loại ứng dụng có thể được xây dựng trong .NET framework được phân loại rộng rãi thành các loại sau.

**- WinForms**

Được sử dụng để phát triển các ứng dụng Forms-based, quá trình này chạy trên end user machine. Notepad là một ví dụ về ứng dụng dựa trên ứng dụng khách.

**- ASP.Net**

Được sử dụng để phát triển các ứng dụng dựa trên web, được tạo để chạy trên bất kỳ trình duyệt nào như Internet Explorer, Chrome hoặc Firefox.

Ứng dụng Web sẽ được xử lý trên một máy chủ, sẽ được cài đặt Dịch vụ thông tin Internet.

Dịch vụ thông tin Internet hoặc IIS là một thành phần của Microsoft được sử dụng để thực thi một ứng dụng [Asp.Net](https://bizflycloud.vn/tin-tuc/aspnet-la-gi-20181113153527162.htm).

Kết quả của việc thực hiện sau đó được gửi đến các máy khách và kết quả đầu ra được hiển thị trong trình duyệt.

**- ADO.Net**

Công nghệ này được sử dụng để phát triển các ứng dụng tương tác với Cơ sở dữ liệu như Oracle hoặc Microsoft SQL Server.

Microsoft luôn đảm bảo rằng các .NET framework tuân thủ tất cả các hệ điều hành Windows được hỗ trợ.

**Nói về WinForm**

WinFom là một **công nghệ** của Microsoft có tuổi đời khá cao ( ra đời năm 2003), cho phép lập trình các ứng dụng Windows. Nhờ tính tiện ích, dễ code, giao diện design kéo thả đơn giản, … Win Form đã được sử dụng để phát triển rất nhiều ứng dụng. Vì dễ code, chỉ cần kéo thả, lại có nhiều component có sẵn, WinForm rất phù hợp để làm các phần mềm quản lý, tính tiền, thống kê… . Đây cũng là loại ứng dụng mà các công ty/doanh nghiệp vừa và nhỏ cần. Ngoài ra, chỉ cần **sử dụng component như TelerikUI hoặc DevExpress**(Ở Việt Nam, hầu như chúng ta đều dùng crack nên các component này hoàn toàn miễn phí), WinForm có thể tạo ra các giao diện hiện đại, đẹp như mơ, long lanh hoa mĩ.

**Nói về C#**

**C# là**[**ngôn ngữ**](https://toidicodedao.com/2015/07/30/c-la-ngon-ngu-tuyet-voi-nhat-java-php-c-c-ruby-chi-toan-la-thu-re-tien/)**lập trình tuyệt vời và đáng học nhất**. Lý do thì vô số kể:

* Bản thân ngôn ngữ C# có vô số điều [thú vị](https://toidicodedao.com/2015/01/22/series-c-hay-ho-nhung-dieu-thu-vi-trong-c-phan-1/): static method, partial class, [delegate](https://toidicodedao.com/2015/02/10/series-c-hay-ho-callback-trong-c-delegate-action-predicate-func/), [LINQ](https://toidicodedao.com/2015/03/26/series-c-hay-ho-linq/), [lambda expression](https://toidicodedao.com/2015/02/12/series-c-hay-ho-lambda-expression/), … Cái ngôn ngữ cùi bắp như Java làm gì có partial class, [delegate](https://toidicodedao.com/2015/02/10/series-c-hay-ho-callback-trong-c-delegate-action-predicate-func/), đến Java 8 mới bắt chước được cái [lambda expression](https://toidicodedao.com/2015/02/12/series-c-hay-ho-lambda-expression/) còn gì.
* C# là ngôn ngữ strong-typed: Các tham số, kết quả trả về của hàm đều là một object. Mọi lỗi do đánh nhầm tên trường, tên hàm, nhầm kiểu class đều được báo trong lúc viết code, không phải chờ đến lúc chạy mới báo như mấy cái ngôn ngữ PHP, python.
* C# đi kèm với framework .NET, hỗ trợ nhiều thứ: Tạo ứng dụng Window với WinForm, WPF; Tạo website bằng WebForm, MVC.NET… Mấy cái ngôn ngữ cấp thấp như C, C++ không làm được mấy cái đó.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ. VS ra bản mới đều đặn như FIFA. Reshaper hỗ trợ refactor, tăng tốc độ code … Mấy đứa khác code PHP, Python dùng cái gì để code? Dĩ nhiên là mấy thứ rởm rởm như Notepad++ hoặc Sublime Text rồi, đến cái chức năng “Jump to Definition” còn không có.

## Các thành phần đối tượng hỗ trợ xây dựng giao diện trên Windows

|  |  |
| --- | --- |
| **Tên Component** | **Chức năng** |
| Label | Hiển thị thông tin |
| TextBox | Nhập các giá trị mong muốn |
| Compobox | Hiển thị danh sách dữ liệu dưới dạng Thả chọn cho phép người dùng lựa chọn |
| Button | Các nút bấm |
| DatagridView | Hiển thị dữ liệu dưới dạng danh sách |
| Panel | Nơi chứa các component khác nhằm dễ định hình và sắp xếp |
| GroupBox | Nơi chứa các component khác nhằm dễ định hình và sắp xếp |
| TabControl | Nơi chứa các component khác với dạng phân theo tab nhằm dễ định hình và sắp xếp |

## Cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (**Transact-SQL)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

# CHƯƠNG 2.PHÂN TÍCH HỆ THỐNG

## 2.1 Ngôn ngữ lập trình,cơ sở dữ liệu và môi trường ứng dụng

- Ngôn ngữ lập trình : C#.

- Cơ sở dữ liệu : SQL Server.

- Môi trường ứng dụng : Hệ điều hành Windows ( Phiên bản 7 và 10).

**Danh sách Bảng trong cơ sở dữ liệu.**

**\* Tài khoản**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDTK** | **INT** | Không | Mã tài khoản dùng đề phân biệt các tài khoản với nhau, Mã tăng tự động |
| **TENTK** | **VARCHAR** | Không | Tên tài khoản. |
| **MATKHAU** | **VARCHAR** | Không | Mật khẩu tài khoản. |
| **LOAITK** | **INT** | Không | Phân loại tài khoản 1 là Quản lý, 0 là Nhân viên. |
| **TENNV** | **NVARCHAR** | Không | Tên Nhân viên. |
| **NGAYSINH** | **DATE** | Không | Ngày sinh của nhân viên.. |
| **CMND** | **CHAR** | Có | Chứng minh nhân dân của nhân viên. |
| **SDT** | **CHAR** | Không | Chức vụ nhân viên bao gồm:  Quản lý và Nhân viên. |
| **MUCLUONG** | **INT** | Không | Mức lương nhân viên. |

**\* khu vực**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDKV** | **INTEGER** | Không | Mã Khu vực dùng đề phân biệt các khu vực với nhau, Mã tăng tự động. |
| **TENKV** | **NVARCHAR** | Không | Tên khu vực. |

**\* Bàn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDBAN** | **INTEGER** | Không | Mã Bàn dùng đề phân biệt các Bàn với nhau, Mã tăng tự động. |
| **idkv** | **INTEGER** | Không | Mã khu vực dùng để phân bàn theo khu vực chỉ định. |
| **TENBAN** | **NVARCHAR** | Không | Tên Bàn. |
| **TRANGTHAI** | **BOOLEAN** | không | Biểu thị trạng thái bàn có khách hay không. |

**\* Danh mục thực phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDDMTP** | **INTEGER** | Không | Mã Danh Mục dùng đề phân biệt các Danh Mục với nhau, Mã tăng tự động. |
| **TenDM** | **NVARCHAR** | không | Tên Danh Mục |

**\* Món ăn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDTP** | **INTEGER** | Không | Mã Sản phẩm dùng đề phân biệt các Sản phẩm với nhau, Mã tăng tự động. |
| **IDDM** | **INTEGER** | không | Mã danh mục sản phẩm : 1 là Thức ăn, 0 là Đồ uống. |
| **TENTP** | **NVARCHAR** | Không | Tên Sản phẩm. |
| **GIA** | **INTEGER** | không | Giá sản phẩm. |

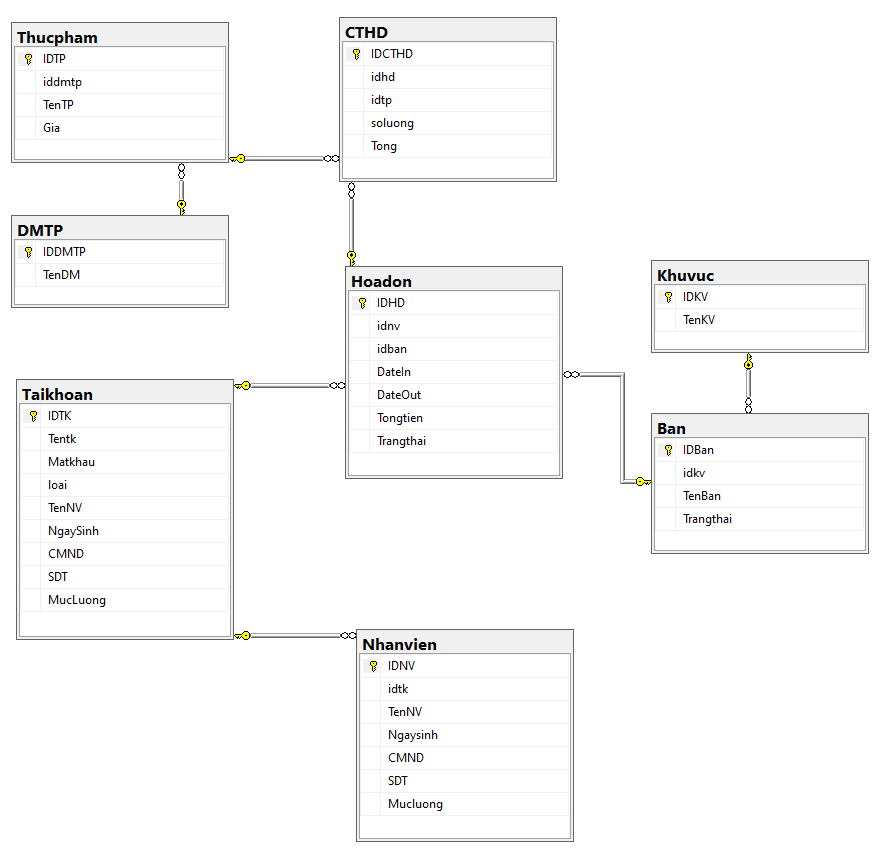
**\* Hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDHD** | **INTEGER** | Không | Mã Hóa đơn dùng đề phân biệt các Hóa đơn với nhau, Mã tăng tự động. |
| **idban** | **INTEGER** | không | Mã bàn (khóa ngoại). |
| **idnv** | **INTEGER** | Không | Mã tài khoản (Khóa ngoại). |
| **TGVAO** | **DATE** | Có | Thời gian lập hóa đơn. |
| **TGRA** | **DATE** | Có | Thời gian thanh toán hóa đơn. |
| **TONGTIEN** | **INTEGER** | Có | Tổng tiền thanh toán. |
| **TRANGTHAI** | **BOOLEAN** | Không | Trạng thái thanh toán : 0 là chưa thanh toán, 1 là đã thanh toán. |

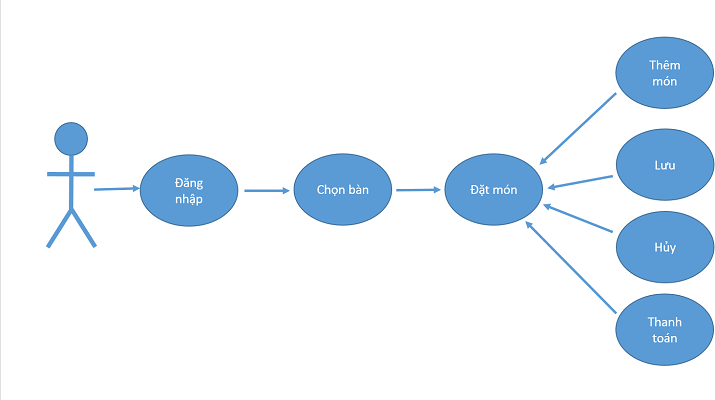
**\* Chi tiết hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép NULL** | **Mô tả** |
| **IDCTHD** | **INTEGER** | Không | Mã Chi tiết Hóa đơn dùng đề phân biệt các Chi tiết Hóa đơn với nhau, Mã tăng tự động. |
| **idhd** | **INTEGER** | không | Mã hóa đơn (Khóa ngoại) |
| **idtp** | **INTEGER** | Không | Mã sản phẩm (Khóa ngoại) |
| **SOLUONG** | **INTEGER** | không | Số lượng sản phẩm |
| **Tong** | **INTEGER** | có | Tổng tiền của sản phẩm |

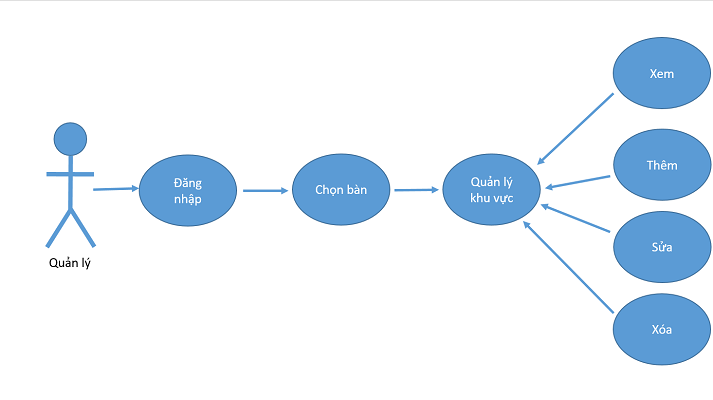
**Database Diagram**



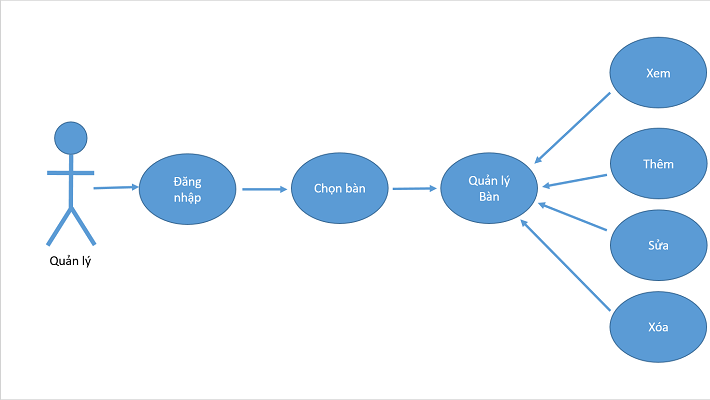
## 2.2 UseCase



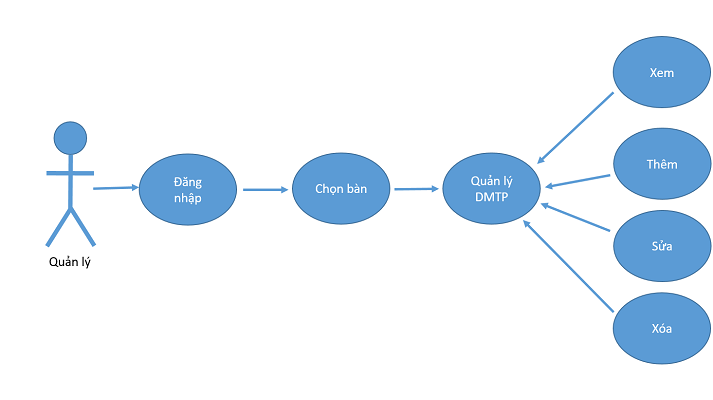
*Use case đặt món*

**

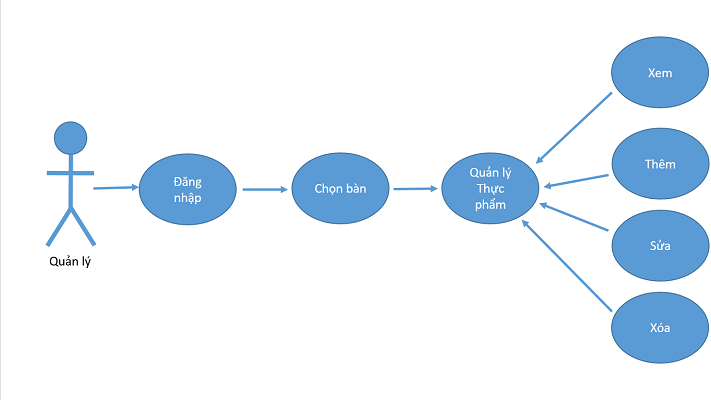
*Use case quản lý khu vực*

**

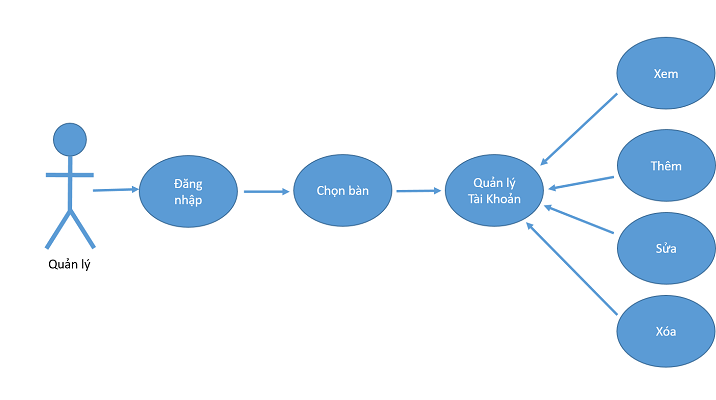
*Use case quản lý bàn*

**

*Use case quản lý danh mục thực phẩm*

**

*Use case quản lý thực phẩm*

**

*Use case quản lý bàn*

## 2.3 Phân tích và đặc tả các chức năng của chương trình

### 2.3.1 Đặc tả use case Đăng nhập

**Tóm tắt:** use case này mô tả cách người dùng đăng nhập vào hệ thống.

**Actor:** Nhân viên, Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu khi một người dùng muốn đăng nhập vào hệ thống.

(1) Người dùng nhập tên đăng nhập và mật khẩu.

(2) Hệ thống kiểm chứng thông tin đăng nhập, nếu trùng khớp với dữ liệu đã lưu thì cho phép người dùng đăng nhập vào hệ thống với giao diện người dùng tương ứng (Giao diện dành cho Admin hoặc khách hàng) .

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính, người dùng nhập sai tên đăng nhập, mật khẩu, hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng nhập, lúc này use case kết thúc.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình đăng nhập, các chức năng không được hiển thị.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công actor có thể sử dụng các chức năng được phân quyền.

**Điểm mở rộng:** không có

### 2.3.2 Đặc tả use case Chọn bàn

**Tóm tắt:** use case này mô tả cách người dùng chọn bàn để đặt món.

**Actor:** Nhân viên, Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu khi một người dùng muốn chọn một bàn bất kỳ để đặt món.

(1) Người dùng chọn Bàn trong khu vực tương ướng.

(2) Hệ thống ghi nhận thông tin bàn đã chọn và chuyển tới giao diện đặt món.

**- Dòng sự kiện khác:** use case kết thúc.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình chính.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công actor có thể sử dụng đặt món cho bàn đã chọn tại giao diện đặt món.

**Điểm mở rộng:** không có

### 2.3.3 Đặc tả use case Quản lý Đặt món

**2.3.3.1 Đặc tả use case Đặt món**

**Tóm tắt:** use case này mô tả cách người dùng đặt món.

**Actor:** Nhân viên, Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn bàn và muốn đăt món.

(1) Người dùng chọn thực phẩm trong ô tương ứng.

(2) Hệ thống ghi nhận thông tin thực phẩm đã chọn và hiển thị dialog xác nhận số lượng thực phẩm cần thêm.

(3) Actor nhập số lượng và xác nhận.

(4) Hệ thống thêm thực phẩm vào danh sách chờ.

(5) Actor nhấn lưu để lưu các thông tin order vừa chọn.

(6) Hệ thống lưu các thông tin đặt món vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** Nếu actor không bấm lưu các thông tin vừa lưu sẽ mất, use case kết thúc.

**Các yêu cầu đăc biệt**: Phải chọn bàn trước khi đặt món.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn chọn bàn

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công actor sẽ quay lại màn hình chọn bàn với bàn được đánh dấu là có khách.

**Điểm mở rộng:** không có.

**2.3.3.2 Đặc tả use case hủy đặt món**

**Tóm tắt:** use case này mô tả cách người dùng hủy đặt món.

**Actor:** Nhân viên, Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn bàn đã có khách và muốn hủy đăt món.

(1) Người dùng chọn hủy bàn đã đăt món bằng cách nhấn vào nút hủy

(2) Hệ thống ghi nhận và hiển thị Thông báo xác nhận hủy bàn.

(3) Actor nhập chọn vào nút xác nhận.

(4) Hệ thống xóa thông tin order bàn khỏi hệ thống.

**- Dòng sự kiện khác:** Nếu actor không bấm hủy các thông tin vẫn giữa nguyên.

**Các yêu cầu đăc biệt**: Phải chọn bàn trước khi đặt món.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn chọn bàn

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công actor sẽ quay lại màn hình chọn bàn với bàn được đánh dấu là có khách.

**Điểm mở rộng:** không có.

**2.3.3.3 Đặc tả use case Thanh toán Hóa Đơn**

**Tóm tắt:** use case này mô tả cách người dùng Thanh toán Hóa đơn.

**Actor:** Nhân viên, Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn bàn đã có khách và muốn thanh toán Hóa đơn.

(1) Người dùng chọn thanh toán hóa đơn bằng các nhấn vào nút thanh thanh toán.

(2) Hệ thống ghi nhận và hiển thị thông báo xác nhận.

(3) Actor nhập chọn nút xác nhận.

(4) Hệ thống Cập nhật bàn đã được thanh toán.

**- Dòng sự kiện khác:** Nếu actor không bấm thanh toán các thông tin vẫn giữ nguyên.

**Các yêu cầu đăc biệt**: Phải chọn bàn trước khi thanh toán.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn chọn bàn

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công actor sẽ quay lại màn hình chọn bàn, bàn được chọn trước đó sẽ quay về trạng thái mặc định.

**Điểm mở rộng:** không có.

### 2.3.4 Đặc tả use case Quản lý khu vực

**2.3.4.1 Đặc tả use case xem khu vực**

**Tóm tắt:** use case này mô tả cách người muốn xem thông tin khu vực.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn xem thông tin khu vực.

(1) Người dùng chọn khu vực trong danh sách.

(2) Hệ thống hiển thị thông tin của khu vực tương ứng.

**- Dòng sự kiện khác:** không có.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** không có.

**Điểm mở rộng:** không có.

**2.3.4.2 Đặc tả use case thêm khu vực**

**Tóm tắt:** use case này mô tả cách người dùng thêm khu vực.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực và muốn thêm khu vực.

(1) Người dùng chọn vào nút thêm mới.

(2) Nút thêm mới chuyển thành nút thêm.

(3) Người dùng nhập các thông tin trên giao diện và nhấn thêm.

(4) Hệ thống kiểm tra các thông tin nhập vào.

(5) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách khu vực sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.3 Đặc tả use case sửa khu vực**

**Tóm tắt:** use case này mô tả cách người dùng sửa thông tin khu vực.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn sửa thông tin khu vực.

(1) Người dùng chọn khu vực trong danh sách

(2) Người dùng chọn vào nút sửa.

(3) Nút thêm mới chuyển thành nút cập nhật.

(4) Người dùng nhập các thông tin trên giao diện và nhấn cập nhật.

(5) Hệ thống kiểm tra các thông tin nhập vào.

(6) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn khu vực trước, các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách khu vực sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.4 Đặc tả use case xóa khu vực**

**Tóm tắt:** use case này mô tả cách người dùng xóa khu vực.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn xóa khu vực.

(1) Người dùng chọn khu vực trong danh sách.

(2) Người dùng chọn vào nút xóa.

(3) Hệ thống xuất hiện thông báo yêu cầu xác nhận xóa

(4) Người dùng chọn nút xác nhận.

(5) Hệ thống xóa khu vực khỏi cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn khu vực trước.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách khu vực sẽ được cập nhật.

**Điểm mở rộng:** không có.

### 2.3.4 Đặc tả use case Quản lý Bàn

**2.3.4.1 Đặc tả use case xem Bàn**

**Tóm tắt:** use case này mô tả cách người muốn xem thông tin Bàn.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn xem thông tin Bàn.

(1) Người dùng chọn khu vực trong danh sách.

(2) Hệ thống hiển thị thông tin của khu vực tương ứng.

**- Dòng sự kiện khác:** không có.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** không có.

**Điểm mở rộng:** không có.

**2.3.4.2 Đặc tả use case thêm Bàn**

**Tóm tắt:** use case này mô tả cách người dùng thêm Bàn.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực và muốn thêm Bàn.

(1) Người dùng chọn vào nút thêm mới.

(2) Nút thêm mới chuyển thành nút thêm.

(3) Người dùng nhập các thông tin trên giao diện và nhấn thêm.

(4) Hệ thống kiểm tra các thông tin nhập vào.

(5) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Bàn sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.3 Đặc tả use case sửa khu vực**

**Tóm tắt:** use case này mô tả cách người dùng sửa thông tin Bàn.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn sửa thông tin Bàn.

(1) Người dùng chọn Bàn trong danh sách

(2) Người dùng chọn vào nút sửa.

(3) Nút thêm mới chuyển thành nút cập nhật.

(4) Người dùng nhập các thông tin trên giao diện và nhấn cập nhật.

(5) Hệ thống kiểm tra các thông tin nhập vào.

(6) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Bàn trước, các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Bàn sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.4 Đặc tả use case xóa khu vực**

**Tóm tắt:** use case này mô tả cách người dùng xóa Bàn.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý bàn và khu vực. và muốn xóa Bàn.

(1) Người dùng chọn Bàn trong danh sách.

(2) Người dùng chọn vào nút xóa.

(3) Hệ thống xuất hiện thông báo yêu cầu xác nhận xóa

(4) Người dùng chọn nút xác nhận.

(5) Hệ thống xóa Bàn khỏi cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Bàn trước.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Bàn sẽ được cập nhật.

**Điểm mở rộng:** không có.

### 2.3.4 Đặc tả use case Quản lý Danh mục thực phẩm

**2.3.4.1 Đặc tả use case xem Danh mục thực phẩm**

**Tóm tắt:** use case này mô tả cách người muốn xem thông tin Danh mục thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý thực phẩm. và muốn xem thông tin Danh mục thực phẩm.

(1) Người dùng chọn Danh mục thực phẩm trong danh sách.

(2) Hệ thống hiển thị thông tin của Danh mục thực phẩm tương ứng.

**- Dòng sự kiện khác:** không có.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** không có.

**Điểm mở rộng:** không có.

**2.3.4.2 Đặc tả use case thêm Danh mục thực phẩm**

**Tóm tắt:** use case này mô tả cách người dùng thêm Danh mục thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý thực phẩm và muốn thêm Danh mục thực phẩm.

(1) Người dùng chọn vào nút thêm mới.

(2) Nút thêm mới chuyển thành nút thêm.

(3) Người dùng nhập các thông tin trên giao diện và nhấn thêm.

(4) Hệ thống kiểm tra các thông tin nhập vào.

(5) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Danh mục thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.3 Đặc tả use case sửa Danh mục thực phẩm**

**Tóm tắt:** use case này mô tả cách người dùng sửa thông tin Danh mục thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm. và muốn sửa thông tin Danh mục thực phẩm.

(1) Người dùng chọn Danh mục thực phẩm trong danh sách

(2) Người dùng chọn vào nút sửa.

(3) Nút thêm mới chuyển thành nút cập nhật.

(4) Người dùng nhập các thông tin trên giao diện và nhấn cập nhật.

(5) Hệ thống kiểm tra các thông tin nhập vào.

(6) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Danh mục thực phẩm trước, các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Danh mục thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.4 Đặc tả use case xóa Danh mục thực phẩm**

**Tóm tắt:** use case này mô tả cách người dùng xóa Danh mục thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm. và muốn xóa Danh mục thực phẩm.

(1) Người dùng chọn Danh mục thực phẩm trong danh sách.

(2) Người dùng chọn vào nút xóa.

(3) Hệ thống xuất hiện thông báo yêu cầu xác nhận xóa

(4) Người dùng chọn nút xác nhận.

(5) Hệ thống xóa Danh mục thực phẩm khỏi cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Danh mục thực phẩm trước.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Danh mục thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

### 2.3.4 Đặc tả use case Quản lý Thực phẩm

**2.3.4.1 Đặc tả use case xem Thực phẩm**

**Tóm tắt:** use case này mô tả cách người muốn xem thông tin Thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm. và muốn xem thông tin Thực phẩm.

(1) Người dùng chọn Thực phẩm trong danh sách.

(2) Hệ thống hiển thị thông tin của Thực phẩm tương ứng.

**- Dòng sự kiện khác:** không có.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** không có.

**Điểm mở rộng:** không có.

**2.3.4.2 Đặc tả use case thêm Thực phẩm**

**Tóm tắt:** use case này mô tả cách người dùng thêm Thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm và muốn thêm Thực phẩm.

(1) Người dùng chọn vào nút thêm mới.

(2) Nút thêm mới chuyển thành nút thêm.

(3) Người dùng nhập các thông tin trên giao diện và nhấn thêm.

(4) Hệ thống kiểm tra các thông tin nhập vào.

(5) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.3 Đặc tả use case sửa Thực phẩm**

**Tóm tắt:** use case này mô tả cách người dùng sửa thông tin Thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm. và muốn sửa thông tin Thực phẩm.

(1) Người dùng chọn Thực phẩm trong danh sách

(2) Người dùng chọn vào nút sửa.

(3) Nút thêm mới chuyển thành nút cập nhật.

(4) Người dùng nhập các thông tin trên giao diện và nhấn cập nhật.

(5) Hệ thống kiểm tra các thông tin nhập vào.

(6) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Thực phẩm trước, các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

2.3.4.4 Đặc tả use case xóa Thực phẩm

**Tóm tắt:** use case này mô tả cách người dùng xóa Thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Thực phẩm. và muốn xóa Thực phẩm.

(1) Người dùng chọn Thực phẩm trong danh sách.

(2) Người dùng chọn vào nút xóa.

(3) Hệ thống xuất hiện thông báo yêu cầu xác nhận xóa

(4) Người dùng chọn nút xác nhận.

(5) Hệ thống xóa Thực phẩm khỏi cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Thực phẩm trước.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Thực phẩm sẽ được cập nhật.

**Điểm mở rộng:** không có.

### 2.3.4 Đặc tả use case Quản lý Tài khoản

**2.3.4.1 Đặc tả use case xem Tài khoản**

**Tóm tắt:** use case này mô tả cách người muốn xem thông tin Thực phẩm.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Tài khoản. và muốn xem thông tin Tài khoản.

(1) Người dùng chọn Tài khoản trong danh sách.

(2) Hệ thống hiển thị thông tin của Tài khoản tương ứng.

**- Dòng sự kiện khác:** không có.

**Các yêu cầu đăc biệt**: không có.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** không có.

**Điểm mở rộng:** không có.

**2.3.4.2 Đặc tả use case thêm Tài khoản**

**Tóm tắt:** use case này mô tả cách người dùng thêm Tài khoản.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Tài khoản và muốn thêm Tài khoản.

(1) Người dùng chọn vào nút thêm mới.

(2) Nút thêm mới chuyển thành nút thêm.

(3) Người dùng nhập các thông tin trên giao diện và nhấn thêm.

(4) Hệ thống kiểm tra các thông tin nhập vào.

(5) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Tài khoản sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.3 Đặc tả use case sửa Tài khoản**

**Tóm tắt:** use case này mô tả cách người dùng sửa thông tin Tài khoản.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Tài khoản. và muốn sửa thông tin Tài khoản.

(1) Người dùng chọn Tài khoản trong danh sách

(2) Người dùng chọn vào nút sửa.

(3) Nút thêm mới chuyển thành nút cập nhật.

(4) Người dùng nhập các thông tin trên giao diện và nhấn cập nhật.

(5) Hệ thống kiểm tra các thông tin nhập vào.

(6) Hệ thống lưu thông tin vào cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Tài khoản trước, các thông tin nhập vào không được để trống hoặc trùng lặp trong cơ sở dữ liệu.

**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Tài khoản sẽ được cập nhật.

**Điểm mở rộng:** không có.

**2.3.4.4 Đặc tả use case xóa Tài khoản**

**Tóm tắt:** use case này mô tả cách người dùng xóa Tài khoản.

**Actor:** Quản lý.

**Dòng sự kiện:**

**- Dòng sự kiện chính:** use case này bắt đầu sau khi người dùng chọn nút quản lý Tài khoản. và muốn xóa Tài khoản.

(1) Người dùng chọn Tài khoản trong danh sách.

(2) Người dùng chọn vào nút xóa.

(3) Hệ thống xuất hiện thông báo yêu cầu xác nhận xóa

(4) Người dùng chọn nút xác nhận.

(5) Hệ thống xóa Tài khoản khỏi cơ sở dữ liệu.

**- Dòng sự kiện khác:** nếu trong dòng sự kiện chính sẽ ra lỗi bỏ trống hoặc trùng lặp thông tin hệ thống sẽ thông báo lỗi và không lưu thông tin vào cơ sở dữ liệu.

**Các yêu cầu đăc biệt**: Cần chọn Tài khoản trước.

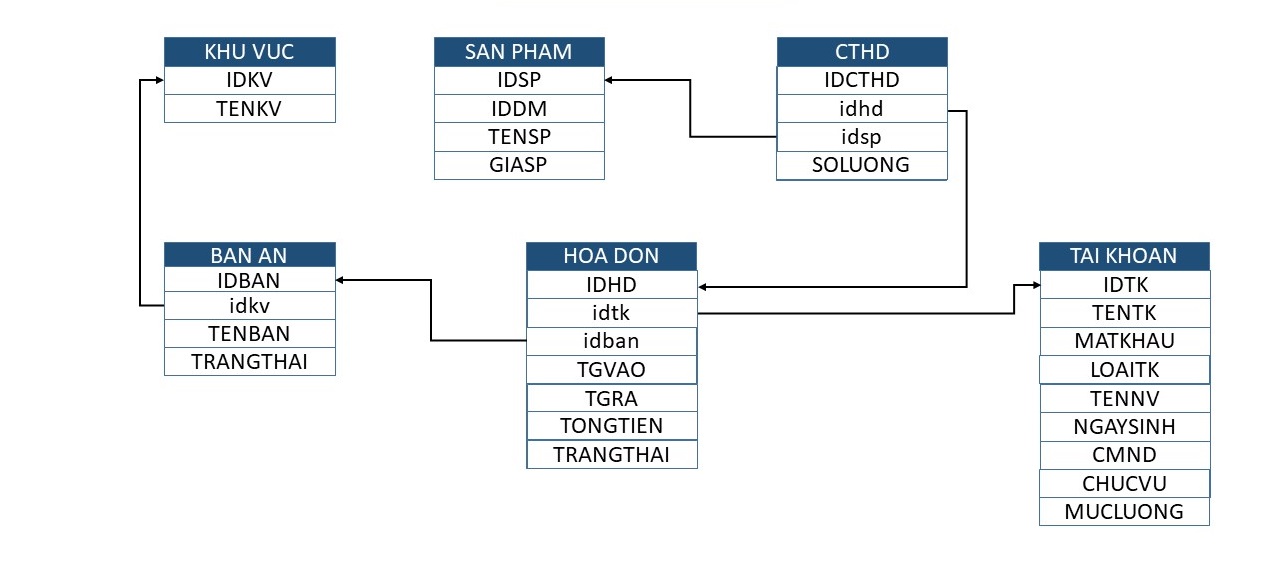
**Trạng thái hệ thống trước khi thực hiện use case:** Đứng ở màn hình hệ thống.

**Trạng thái hệ thống sau khi thực hiện use case:** nếu use case thành công danh sách Tài khoản sẽ được cập nhật.

**Điểm mở rộng:** không có.

## 2.4 Sơ đồ, cấu trúc chương trình

Class diagram



# CHƯƠNG 3.THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG

## 3.1 Thiết kế chức năng và Giao diện

### 3.1.1 Đăng nhập

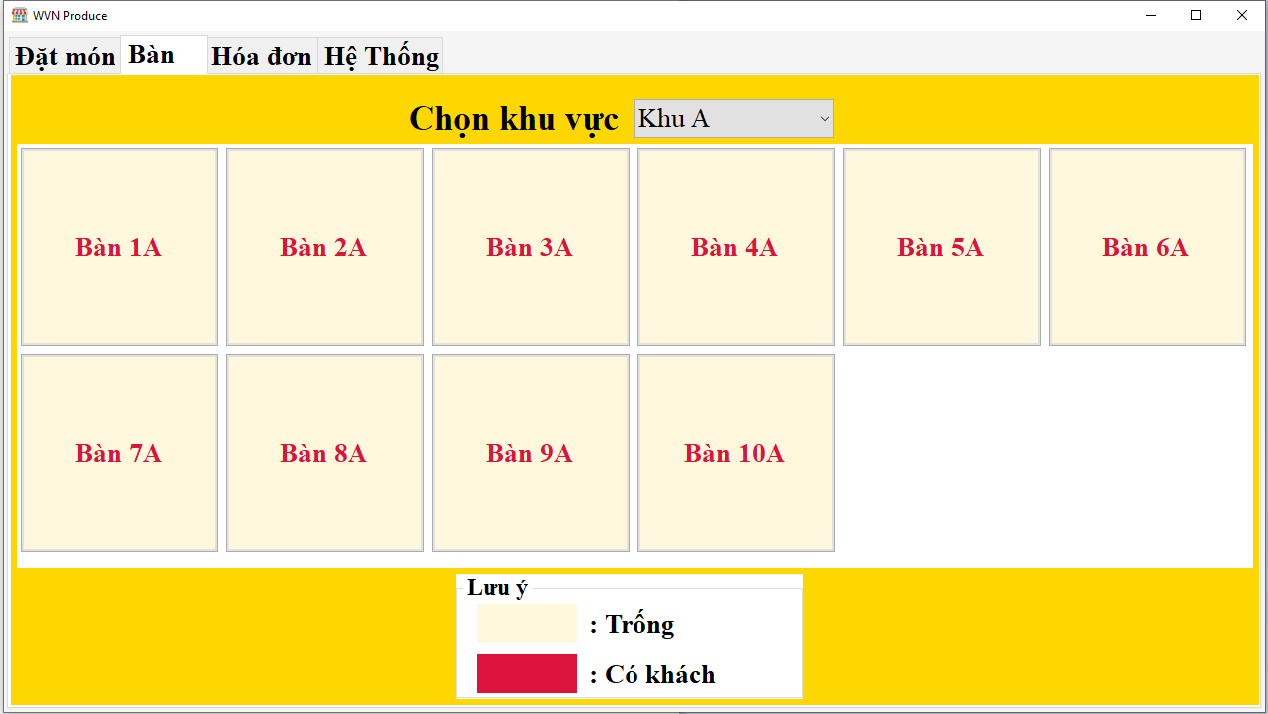
|  |  |
| --- | --- |
| **Name** | Đang nhập |
| **Description** | Người dùng Nhập thông tin tài khoản để đăng nhập vào hệ thống. |
| **Actor** | Nhân viên, quản lý. |
| **Trigger** | Khi người dùng nhấn vào nút đăng nhập. |
| **Pre-condition** |  |
| **Postcondition** | Đăng nhập thành công vào hệ thống. |



*Giao diện đăng nhập*

### 3.1.2 Chọn Bàn

|  |  |
| --- | --- |
| **Name** | Chọn bàn |
| **Description** | Người dùng chọn bàn để đặt món |
| **Actor** | Nhân viên, quản lý |
| **Trigger** | Khi người dùng chọn bàn |
| **Pre-condition** |  |
| **Postcondition** | Chuyển đến giao diện thêm món |



*Giao diện chọn bàn*

**

*Giao diện đặt món*

### 3.1.3 Quản lý đặt món

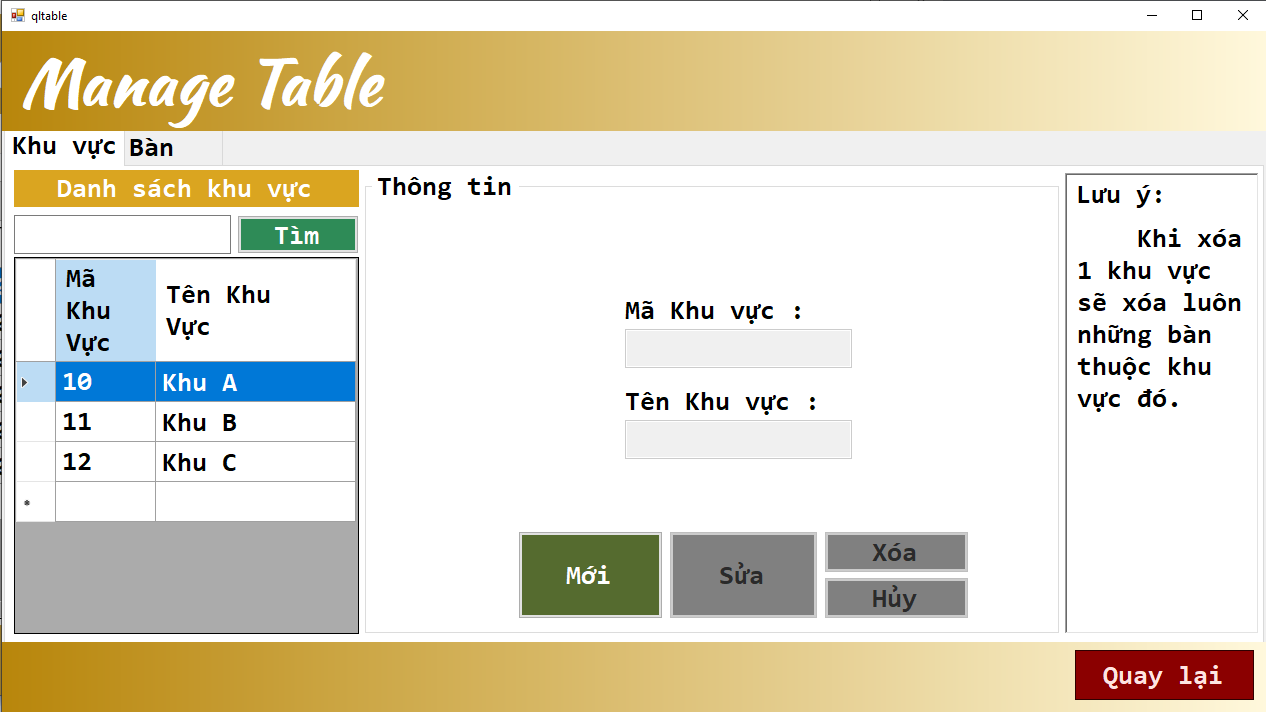
|  |  |
| --- | --- |
| **Name** | Đặt món |
| **Description** | Người dủng nhấp chọn thực phẩm muốn thêm |
| **Actor** | Nhân viên, quản lý |
| **Trigger** | Khi người dủng nhấp chọn thực phẩm |
| **Pre-condition** | Giao diện chọn bàn |
| **Postcondition** | Đặt món thành công |



*Giao diện xác nhận số lượng*

### 3.1.4 Quản lý Khu vực

|  |  |
| --- | --- |
| **Name** | Xem, Thêm, Sửa, Xóa khu vực |
| **Description** | Người dùng quản lý thông tin khu vực |
| **Actor** | Quản lý |
| **Trigger** | Chọn các nút tương ứng |
| **Pre-condition** | Giao diện hệ thống. |
| **Postcondition** | Danh sách khu vực cập nhật |



*Giao diện quản lý khu vực*

### 3.1.5 Quản lý Bàn

|  |  |
| --- | --- |
| **Name** | Xem, Thêm, Sửa, Xóa Bàn |
| **Description** | Người dùng quản lý thông tin Bàn |
| **Actor** | Quản lý |
| **Trigger** | Chọn các nút tương ứng |
| **Pre-condition** | Giao diện hệ thống. |
| **Postcondition** | Danh sách Bản cập nhật |

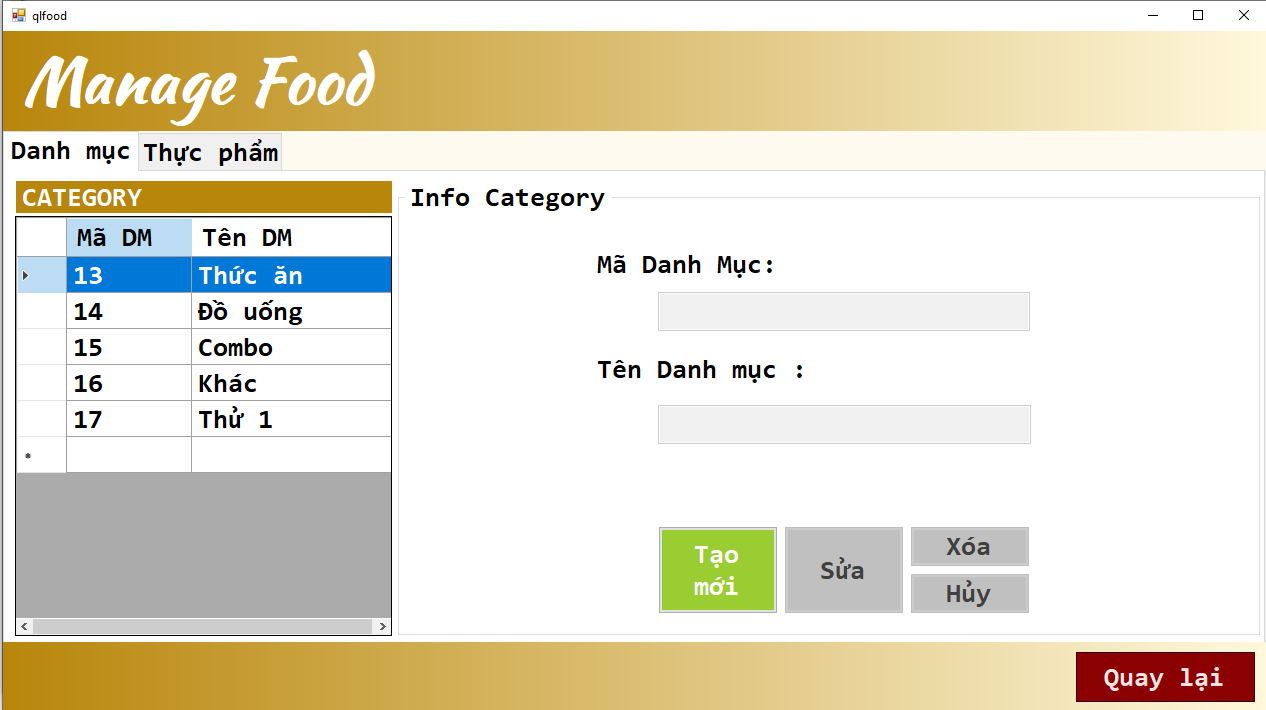


*Giao diện quản lý bàn*

### 3.1.6 Quản lý Danh mục thực phẩm

|  |  |
| --- | --- |
| **Name** | Xem, Thêm, Sửa, Xóa Danh mục thực phẩm |
| **Description** | Người dùng quản lý thông tin Danh mục thực phẩm |
| **Actor** | Quản lý |
| **Trigger** | Chọn các nút tương ứng |
| **Pre-condition** | Giao diện hệ thống. |
| **Postcondition** | Danh sách Danh mục thực phẩm cập nhật |

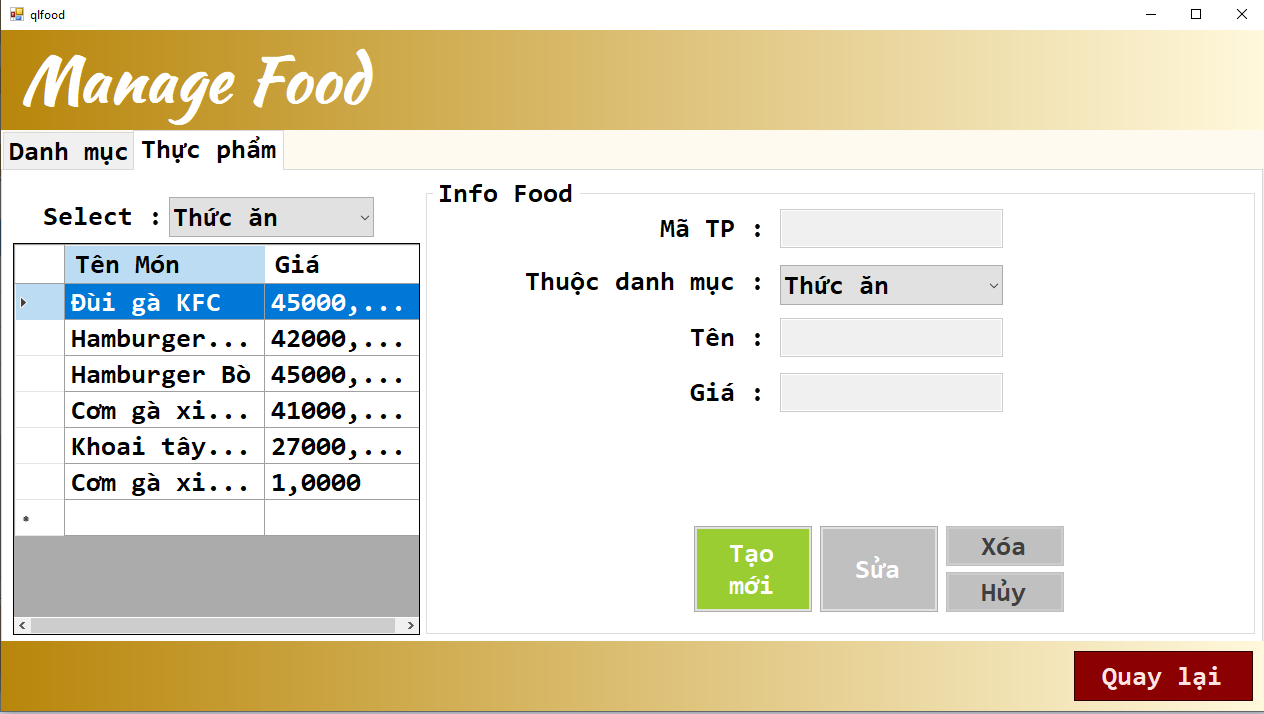
a



*Giao diện Quản lý DMTP*

### 3.1.6 Quản lý Thực phẩm

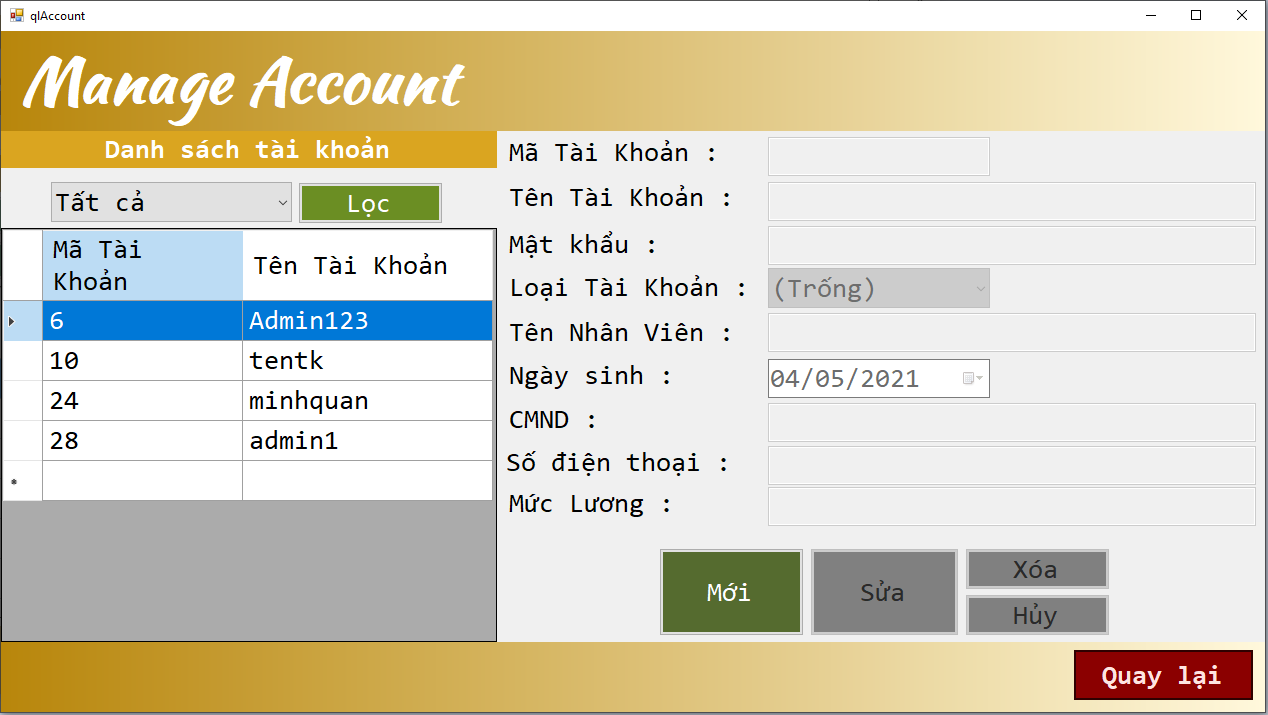
|  |  |
| --- | --- |
| **Name** | Xem, Thêm, Sửa, Xóa Thực phẩm |
| **Description** | Người dùng quản lý thông tin Thực phẩm |
| **Actor** | Quản lý |
| **Trigger** | Chọn các nút tương ứng |
| **Pre-condition** | Giao diện hệ thống. |
| **Postcondition** | Danh sách Thực phẩm cập nhật |



*Giao diện quản lý Thực phẩm*

### 3.1.6 Quản lý Tài khoản

|  |  |
| --- | --- |
| **Name** | Xem, Thêm, Sửa, Xóa Tài khoản |
| **Description** | Người dùng quản lý thông tin Tài khoản |
| **Actor** | Quản lý |
| **Trigger** | Chọn các nút tương ứng |
| **Pre-condition** | Giao diện hệ thống. |
| **Postcondition** | Danh sách Tài khoản cập nhật |



*Giao diện quản lý Tài khoản*

# CHƯƠNG 4.THỰC NGHIỆM VÀ TRIỂN KHAI

## 4.1. Đóng gói và xuất bản chương trình

Đóng gói chương trình bằng **Microsoft Visual Studio Installer Projects.**



## 4.2. Demo các chức năng của chương trình

**4.2.1 Đăng nhập**

**-** Người dùng nhập thông tin đăng nhập.

- Hệ thống kiểm tra nếu đúng sẽ đăng nhập vào hệ thống.

- Nếu sai sẽ thông báo ( sai quá 3 lần sẽ thoát ứng dụng).



**4.2.2 Đặt món**

**-** Người dùng chọn bàn muốn đặt món.

- Chọn thực phẩm muốn đặt.

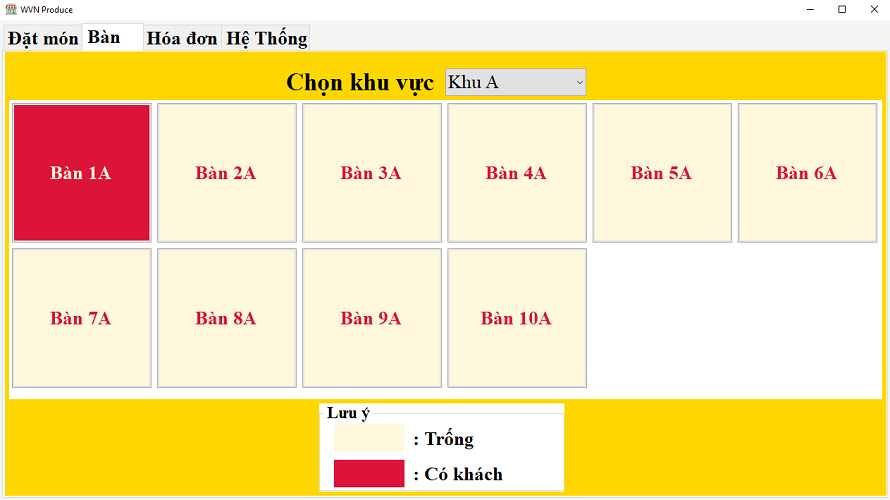
- nhập số lượng và xác nhận.

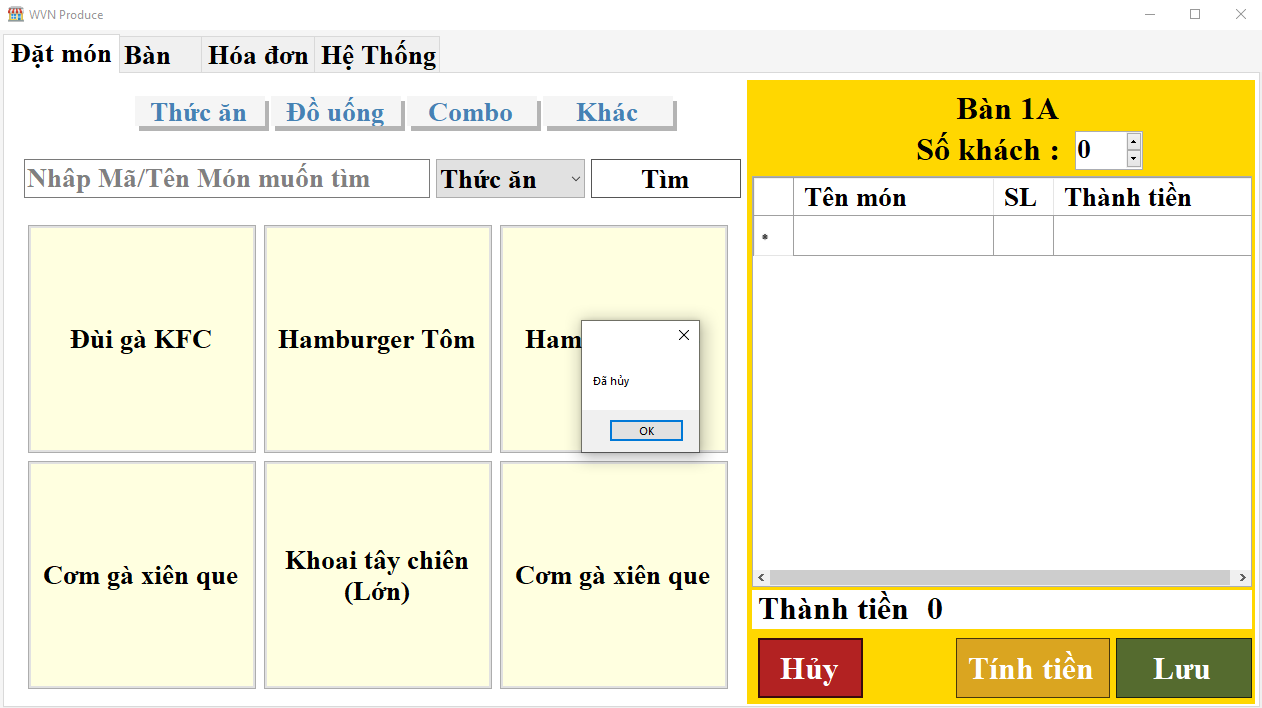
- Bấm hủy để hủy hóa đơn.

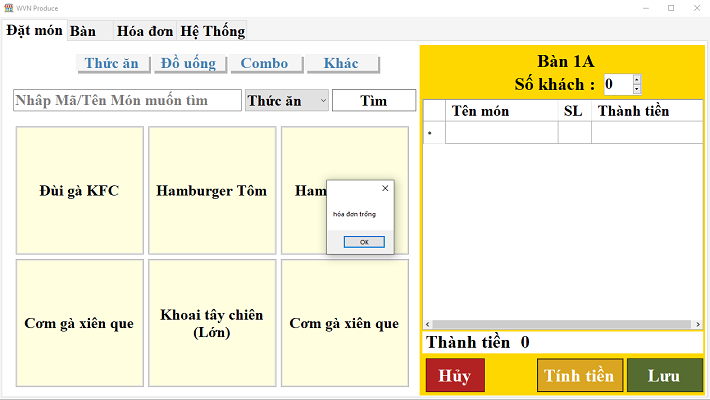
- Bấm thanh toán để thanh toán.

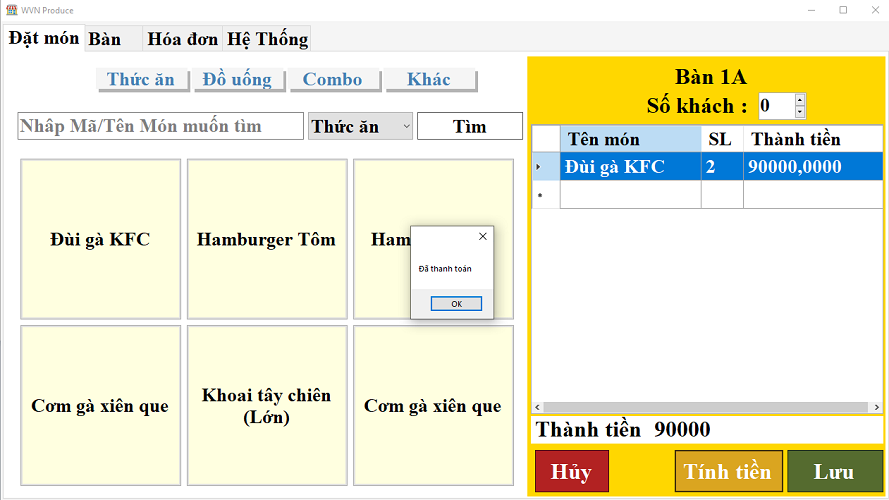
- Bấm lưu để lưu lại các món đã order.











**4.2.3 Tính năng quản lý.**

**-** Người dùng chọn thông tin trên khung danh sách ( Khu vực,Bàn,…)

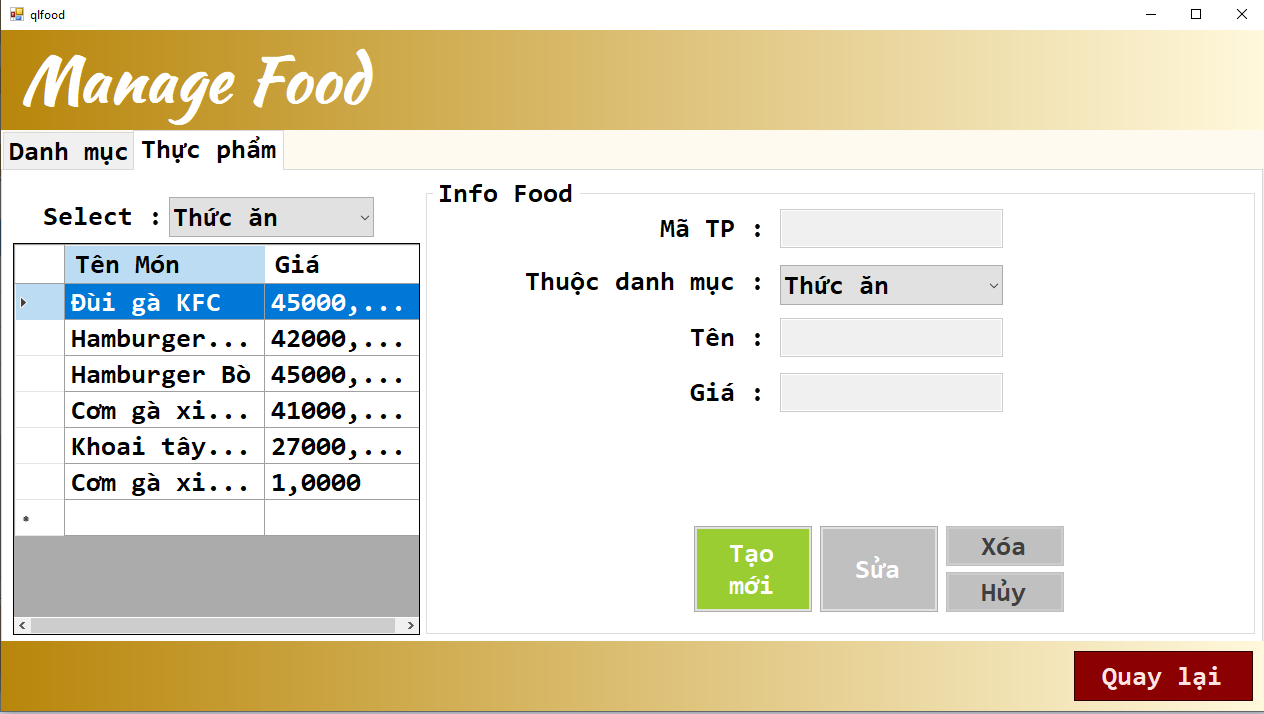
- Thông tin hiển thị.

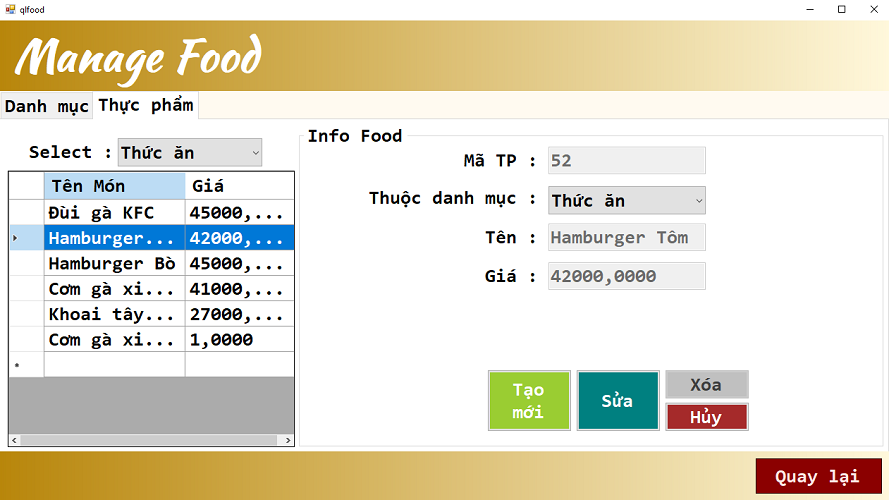
- Người dùng Bấm thêm để thêm thông tin.

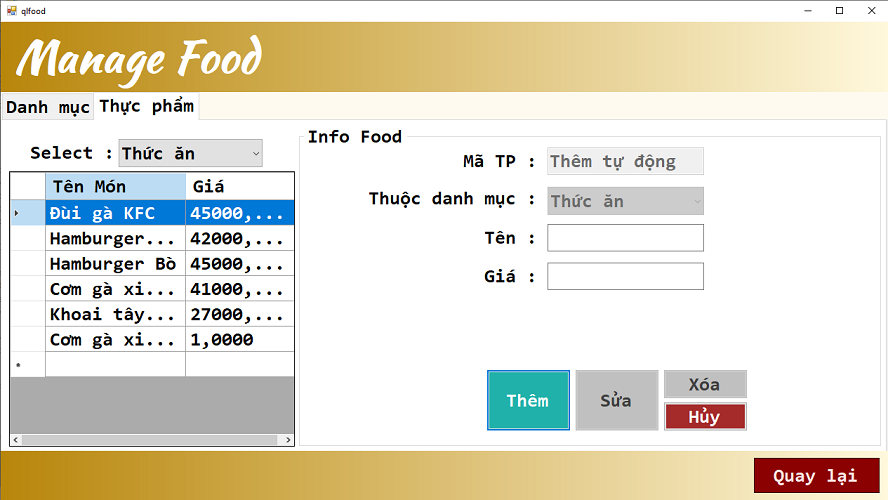
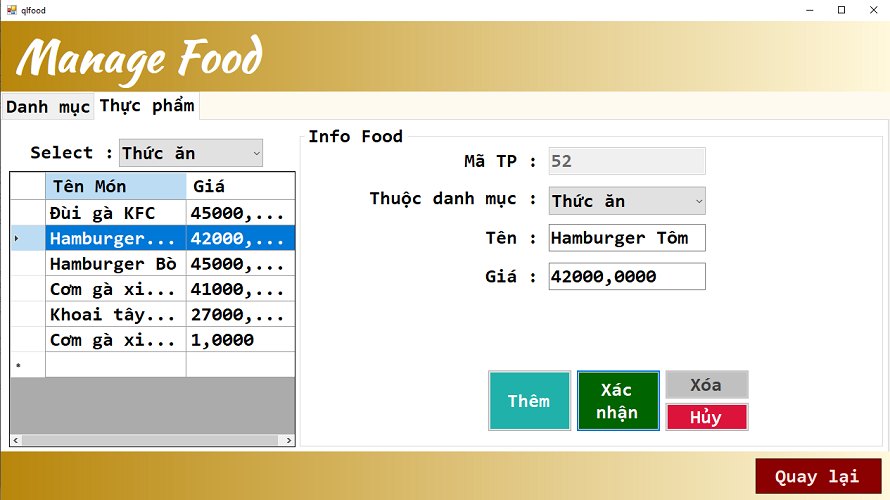
- Người dùng bấm Sửa để cập nhật thông tin.

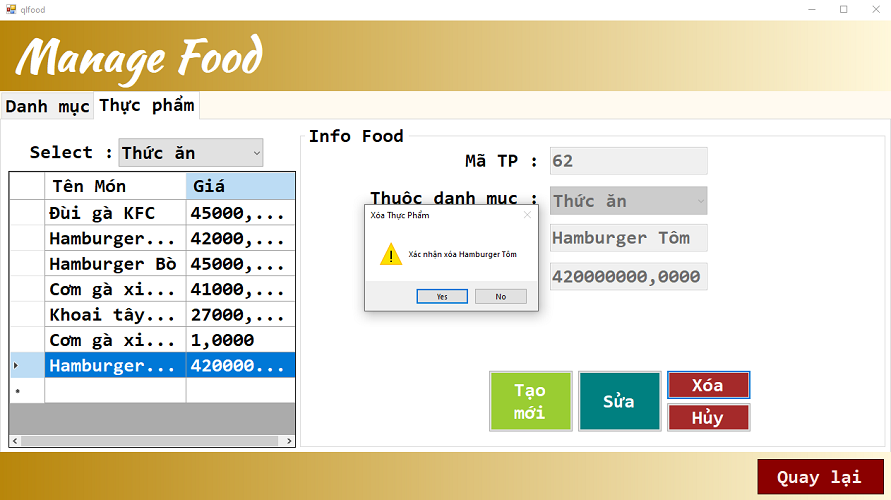
- Người dùng Bấm Xóa để xóa thông tin.

- Người dùng Bấm hủy để Hủy thao tác đang làm.









Các danh mục khác : Khu vực, Bàn, DMTP, Tài khoản tương tự.

# CHƯƠNG 5.KẾT LUẬN

## 5.1 Kết quả đạt được

**\* Những tính năng đã làm được :**

- Đăng nhập.

+ Quản lý:

- Quản lý khu vực (Thêm, sửa).

- Quản lý bàn (Thêm, sửa, Lọc).

- Quản lý Thực phẩm ( Lọc, Thêm, Sửa, Xóa) trong 2 danh mục là Thức ăn và đồ uống.

- Quản lý Hóa đơn.

- Quản lý tài khoản ( Cấp tài khoản cho nhân viên , sửa thông tin, xóa, Lọc)

+ Phục vụ:

- Quản lý việc đặt món cho khách ( thêm, xóa, sửa món trong hóa đơn đặt món).

- Xem lại các hóa đơn đã thanh toán.

- Xem và cập nhập thông tin cá nhân ( bao gồm : Tên, CMND, Ngày sinh).

**\* Những hạn chế :**

- Thời gian gặp mặt trực tiếp hạn chế.

- Kinh nghiệm của các thành viên không đều.

## 5.2 Hướng phát triển

- Cải thiện giao diện.

- Tối ưu hóa code mang tính thống nhất và phân theo thư mục hợp lý nâng cao khả năng chỉnh sửa về sau.

- Nâng cấp tốc độ truy xuất với SQL Server.

- Cập nhật các tính năng mới cần thiết.

# TÀI LIỆU THAM KHẢO

***Lập trình trên môi trường Windows,*** *Đại học Công nghệ Tp.HCM.*

***C# Series****, https://www.toidicodedao.com*

***Giáo trình SQL Server toàn tập****,* *https://www.howkteam.vn*

***Tài liệu và trao đổi về C#*** *,* [*https://stackoverflow.com*](https://stackoverflow.com)

***Tài liệu UML****, http://hoc-uml.blogspot.com*