**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN GIỮA KỲ**

**MÔN HỌC: THIẾT KẾ PHẦN MỀM**

**HƯỚNG ĐỐI TƯỢNG**

**GAME ĐUA XE VƯỢT CHƯỚNG NGẠI VẬT STATE**

**Nhóm thực hiện:**

1. **Hồ Quốc Đạt 17110117**
2. **Lê Kim Đỉnh 17110123**
3. **Trần Gia Hân 17110133**
4. **Nguyễn Thành Tâm 17110219**

**Giảng viên hướng dẫn: Ths. Nguyễn Trần Thi Văn**

**Tp. Hồ Chí Minh, tháng 05 năm 2020**

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc26755634)

[LỜI NÓI ĐẦU 2](#_Toc26755635)

[CHƯƠNG 1: TỔNG QUAN ĐỒ ÁN 3](#_Toc26755636)

[1.1. Giới thiệu Hệ quản trị Cơ sở dữ liệu 3](#_Toc26755637)

[1.2. Đặt vấn đề 3](#_Toc26755638)

[*1.2.1. Lí do, mục đích chọn đề tài 3*](#_Toc26755639)

[*1.2.2. Mục tiêu làm đề tài 4*](#_Toc26755640)

[*1.2.3. Mô tả hoạt động của nhà hàng 4*](#_Toc26755641)

[*1.2.4. Lựa chọn ngôn ngữ lập trình 5*](#_Toc26755642)

[CHƯƠNG 2: NỘI DUNG ĐỒ ÁN 6](#_Toc26755643)

[2.1. Yêu cầu hệ thống 6](#_Toc26755644)

[*2.1.1. Yêu cầu về chức năng 6*](#_Toc26755645)

[*2.1.2. Yêu cầu về Cơ sở dữ liệu 8*](#_Toc26755646)

[2.2. Thiết kế hệ thống 9](#_Toc26755647)

[*2.2.1. Thiết kế kiến trúc 9*](#_Toc26755648)

[*2.2.2. Thiết kế chi tiết 12*](#_Toc26755649)

[*2.2.3. Thiết kế giao diện 15*](#_Toc26755650)

[*2.2.4. Thiết kế cơ sở dữ liệu 16*](#_Toc26755651)

[2.3. Hiện thực hóa hệ thống 20](#_Toc26755652)

[*2.3.1. Sơ lược về đồ án 20*](#_Toc26755653)

[*2.3.2. Tạo Database 22*](#_Toc26755654)

[*2.3.3. Tạo View 25*](#_Toc26755655)

[*2.3.4. Tạo Dynamic SQL 27*](#_Toc26755656)

[*2.3.5. Tạo Trigger 29*](#_Toc26755657)

[*2.3.6. Tạo Transaction 32*](#_Toc26755658)

[*2.3.7. Tạo Index 34*](#_Toc26755659)

[*2.3.8. Tạo Procedure 36*](#_Toc26755660)

[*2.3.9. Mô hình ba tầng 62*](#_Toc26755661)

[*2.3.10. Hiện thực hóa giao diện 117*](#_Toc26755662)

[CHƯƠNG 3: KẾT LUẬN 127](#_Toc26755663)

[3.1. Tổng kết 127](#_Toc26755664)

[3.2. Ưu điểm 127](#_Toc26755665)

[3.3. Nhược điểm 127](#_Toc26755666)

[3.4. Hướng phát triển 128](#_Toc26755667)

[BẢNG PHÂN CÔNG CÔNG VIỆC 129](#_Toc26755668)

# LỜI NÓI ĐẦU

Design Pattern là một kỹ thuật dành cho lập trình hướng đối tượng. Nó cung cấp cho ta cách tư duy trong từng tình huống của việc lập trình hướng đối tượng và phân tích thiết kế hệ thống phần mềm. Nó cần thiết cho cả các nhà lập trình và nhà phân tích thiết kế.

Design pattern được dùng khắp ở mọi nơi, trong các phần mềm hướng đối tượng các hệ thống lớn. Trong các chương trình trò chơi,… và cả trong các hệ thống tính toán song song,…

Design pattern thể hiện tính kinh nghiệm của công việc lập trình, xây dựng và thiết kế phần mềm. Có thể chúng ta đã gặp design pattern ở đâu đó, trong các ứng dụng, cũng có thể chúng ta đã từng sử dụng những mẫu tương tự như design pattern để giải quyết những tình huống của mình, nhưng chúng ta không có một khái niệm gì về nó cả.

# 

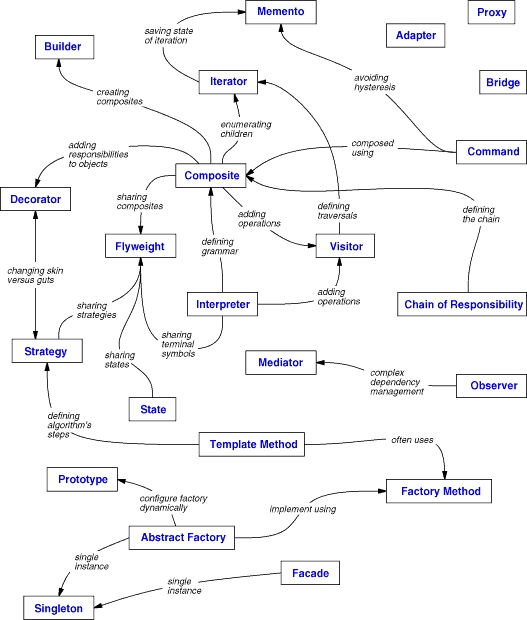
# CHƯƠNG 1: TỔNG QUAN LÝ THUYẾT

## 1.1. Vấn đề trong thiết kế phần mềm hướng đối tượng

Việc thiết kế một phần mềm hướng đối tượng là một công việc khó và việc thiết kế một phần mềm hướng đối tượng phục vụ cho mục đích dùng lại còn khó hơn. CHúng ta phải tìm ra những đối tượng phù hợp, đại diện cho một lớp các đối tượng. Sau đó thiết kế giao diện cho chúng, thiết lập mối quan hệ giữa chúng. Thiết kế của chúng ta phải đảm bảo là giải quyết được các vấn đề hiện tại, có thể tiến hành mở rộng trong tương lai mà tránh phải thiết kế lại phần mềm. Và một tiêu chí quan trọng là phải nhỏ gọn. Việc thiết kế một phần mềm hướng đối tượng phục vụ cho mục đích dùng lại là một công việc khó, phức tạp vì vậy chúng ta không thể mong chờ thiết kế của mình sẽ là đúng, và đảm bảo các tiêu trí trên ngay được. Thực tế là nó cần phải được thử nghiệm sau vài lần và sau đó nó sẽ được sửa chữa lại.

## 1.2. Design Pattern là gì?

Design patterns là tập các giải pháp cho cho vấn đề phổ biến trong thiết kế các hệ thống máy tính. Đây là tập các giải pháp đã được công nhận là tài liệu có giá trị, những người phát triển có thể áp dụng giải pháp này để giải quyết các vấn đề tương tự. Giống như với các yêu cầu của thiết kế và phân tích hướng đối tượng (nhằm đạt được khả năng sử dụng các thành phần và thư viện lớp), việc sử dụng các mẫu cũng cần phải đạt được khả năng tái sử dụng các giải pháp chuẩn đối với vấn đề thường xuyên xảy ra.



Hình : Mối quan hệ giữa các Pattern

Design pattern không phải là một phần của UML cốt lõi,nhưng nó lại đựơc sử dụng rộng rãi trong thiết kế hệ thống hướng đối tượng và UML cung cấp các cơ chế biểu diễn mẫu dưới dạng đồ hoạ.

## 1.3. Hệ thống các mẫu Pattern

Hệ thống các mẫu design pattern hiện có 23 mẫu được định nghĩa trong cuốn “Design patterns Elements of Reusable Object Oriented Software”. Hệ thống các mẫu này có thể nói là đủ và tối ưu cho việc giải quyết hết các vấn đề của bài toán phân tích thiết kế và xây dựng phần mềm trong thời điểm hiện tại.Hệ thống các mẫu design pattern được chia thành 3 nhóm: Creational, Structural, behavioral.

1. Nhóm Creational gồm có 5 pattern: Abstract Factory, Abstract Method, Builder, Prototype và Singleton. Nhóm này liên quan tới việc tạo ra các thể nghiệm (instance) của đối tượng, tách biệt với cách được thực hiện từ ứng dụng. Muốn xem xét thông tin của các mẫu trong nhóm này thì phải dựa vào biểu đồ nào phụ thuộc vào chính mẫu đó, mẫu thiên về hành vi hay cấu trúc.

2. Nhóm Structural gồm có 7 mẫu: Adapter, Bridge, Composite, Decorator, Facade, Proxy và Flyweight. Nhóm này liên quan tới các quan hệ cấu trúc giữa các thể nghiệm, dùng kế thừa, kết tập, tương tác. Để xem thông tin về mẫu này phải dựa vào biểu đồ lớp của mẫu.

3. Nhóm Behavioral gồm có 11 mẫu: Interpreter, Template Method, Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State,Strategy và Visitor. Nhóm này liên quan đến các quan hệ gán trách nhiệm để cung cấp các chức năng giữa các đối tượng trong hệ thống. Đối với các mẫu thuộc nhóm này ta có thể dựa vào biểu đồ cộng tác và biểu đồ diễn tiến. Biểu đồ cộng tác và biểu đồ diễn tiến sẽ giải thích cho ta cách chuyển giao của các chức năng.

## 1.4. State

### 1.4.1. Định nghĩa

State pattern là một trong những mẫu thiết kế thuộc nhóm behaviral cho phép một object có thể biến đổi hành vi của nó khi có những sự thay đổi trạng thái nội bộ.

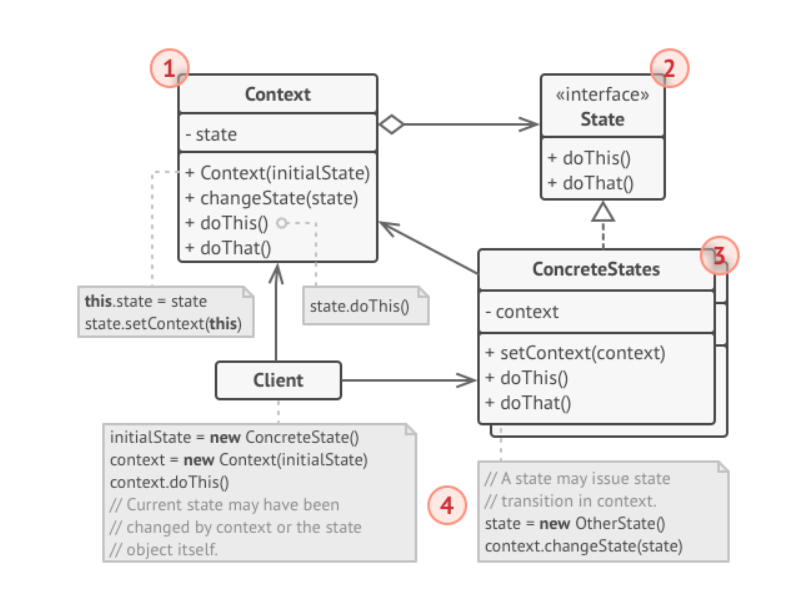
### 1.4.2. Mục đích

- Phân vùng hành vi của đối tượng với những trạng thái khác nhau.

- Đối tượng được thay đổi trạng thái một cách rõ ràng.

- Trạng thái cảu những đối tượng có thể chia sẻ lẫn nhau.

### 1.4.3. Cấu trúc, vai trò và ý nghĩa



Hình : Cấu trúc của State Pattern

Các thành phần tham gia State:

- Context: được sử dụng bởi Client. Client không truy cập trực tiếp đến State của đối tượng. Lớp Context này chứa thông tin của ConcreteState object, cho hành vi nào tương ứng với trạng thái nào hiện đang được thực hiện.

- State: là một interface hoặc abstract class xác định các đặc tính cơ bản của tất cả các đối tượng ConcreteState. Chúng sẽ được sử dụng bởi đối tượng Context để truy cập chức năng có thể thay đổi.

- ConcreteState: cài đặt các phương thức của State. Mỗi ConcreteState có thể thực hiện logic và hành vi của riêng nó tùy thuộc vào Context.

### 1.4.4. Luồng hoạt động

- Context: sẽ định nghĩa những hành vi có thể giao tiếp với clients, vì thế clients sẽ yêu cầu các hành vi thông qua Context

- Context sẽ chứa một thể hiện của Satate, State ban đầu có thể được cài đặt từ client, nhưng khi đã cài đặt rồi thì client không sửa đổi nó nữa.

- Context có thể gửi chính nó như arguments tới State, vì thế Sate có thể truy cập vào context để thay đổi trạng thái nếu cần thiết.

- Khi context thực hiện hành vi, nó sẽ gọi state hiện tại để thực hiện hành vi đó, State có thể sẽ thay đổi trạng thái từ Context nếu cần.

### 1.4.5. Lĩnh vực áp dụng

Trong các trường hợp sau:

- Khi hành vi của đối tượng phụ thuộc vào trạng thái của nó và nó phải có khả năng thay đổi hành vi của nó lúc run-time theo trạng thái mới.

- Khi nhiều điều kiện phức tạp buộc đối tượng phụ thuộc vào trạng thái của nó.

### 1.4.6. Ưu – khuyết điểm

#### 1.4.6.1. Ưu điểm

- Đảm bảo nguyên tắc Single resposibility principle (SRP): tách biệt mỗi State tương ứng với 1 class riêng biệt.

- Đảm bảo nguyên tắc Open/Closed Principle (OCP): có thể thêm một State mới mà không ảnh hưởng đến State khác hay Context hiện có.

- Giữ hành vi cụ thể tương ứng với trạng thái.

- Giúp chuyển trạng thái một cách rõ ràng.

#### 1.4.6.2. Khuyết điểm

Áp dụng mô hình có thể là quá mức nếu một máy trạng thái chỉ có một vài trạng thái hoặc hiếm khi thay đổi.

### 1.4.7. Quan hệ với Pattern khác

Bridge, State, Strategy có cấu trúc rất giống nhau. Thật vậy, tất cả các mẫu này đều dựa trên thành phần đó là úy thác công việc cho các đối tượng khác nhau. Tuy nhiên, tất cả đều giải quyết các vấn đề khác nhau. Một mẫu không chỉ là một công thức để cấu trúc mã của các bạn theo một cách cụ thể. Nó cũng có thể giao tiếp với các nhà phát triển khác vấn đề mà mô hình giải quyết.

State có thể được coi là một phần mở rộng của Strategy. Cả hai mẫu đều dựa trên thành phần: chúng thay đổi hành vi của bối cảnh bằng các h ủy thác một số công việc cho các đối tượng trợ giúp. Chiến lược làm cho các đối tượng này hoàn toàn độc lập và không biết về nhau. Tuy nhiên, State không hạn chế sự phụ thuộc giữa các quốc gia cụ thể, cho phép họ thay đổi trạng thái của bối cảnh theo ý muốn.

# CHƯƠNG 2: TỔNG QUAN ĐỒ ÁN

## 2.1. Lý do, mục đích chọn đề tài

Trong khi học môn Thiết kế phần mềm Hướng đối tượng, nhóm thực hiện được tiếp xúc nhiều hơn với các khối kiến thức cơ sở, điều đó giúp nhóm thực hiện có thể xây dựng các chương trình trên máy tính, áp dụng các kiểu dữ liệu trong thực tế. Thông qua đó, nhóm vận dụng kiến thức về Desgin Pattern (State) của ngôn ngữ lập trình C#.NET để demo Game Đua xe Vượt chướng ngại vật. Với mục đích xây dựng ứng dụng với tính tiện dụng, gần gũi, phù hợp với người dùng, bám sát kiến thức đã học. Và quan trọng là hiểu rõ về mẫu thiết kế State, rèn luyện kỹ năng lập trình, làm tiền đề cho các môn học sau này. Vì vậy nhóm quyết định chọn đề tài Game Đua xe Vượt chướng ngại vật để thực hiện đồ án giữa kỳ môn học Thiết kế phần mềm Hướng đối tượng.

## 2.2. Mục tiêu làm đề tài

Tạo ra một ứng dụng game và trang bị cho một game với đầy đủ các chức năng bằng cách sử dụng mẫu thiết kế State mà môn học Thiết kế phần mềm Hướng đối tượng đã dạy.

## 2.3. Lựa chọn ngôn ngữ lập trình

Nhóm thực hiện đã sử dụng ngôn ngữ Microsoft C# .NET với phiên bản .NET Framework 4.5.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2019.

## 2.4. Giới thiệu Game Đua xe Vượt chướng ngại vật



Hình : Giao diện game

## 2.5. Luật chơi

Người chơi sẽ phải điều khiển xe ô tô bằng các phím điều khiển (qua trái, qua phải, lên, xuống) trên bàn phím. Điều khiển xe ô tô sao cho ô tô không bị va chạm với các xe ô tô khác (được xuất hiện một cách ngẫu nhiên) và chạm vào các đồng xu để được điểm. Nếu ô tô va chạm vào xe ô tô khác thì người chơi thua. Trong một level, nếu người chơi có thể ăn hết các đồng xu thì được đến level tiếp theo.

Để bắt đầu chơi game, người chơi nhấn nút Start. Sau khi đồng hồ đếm ngược đếm hết 3 giây thì trò chơi bắt đầu chạy và xuất hiện các chương ngại vật và các đồng xu (ngẫu nhiên). Người chơi điều khiển xe bằng các phím điều khiển để di chuyển xe tránh chướng ngại vật và ăn đồng xu. Khi di chuyển, trạng thái (state) của xe sẽ hiển thị tương ứng trên màn hình:

- Phím trái: Move left.

- Phím phải: Move right.

- Phím lên: Move up.

- Phím xuống: Move down.

Ăn được đồng xu, màn hình sẽ hiển thị số điểm tương ứng.

Nếu người chơi thua, màn hình sẽ hiển thị thông báo “Game Over” và số điểm người chơi đạt được. Người chơi muốn chơi lại thì nhấn nút “Restart” để bắt đầu ván game mới.

# CHƯƠNG 3: NỘI DUNG ĐỒ ÁN

## 2.1. Yêu cầu hệ thống

### 2.1.1. Yêu cầu về chức năng

#### 2.1.1.1. Tạo sơ đồ chỗ ngồi

Phần mềm Quản lý Nhà hàng cho phép người dùng tạo sơ đồ chỗ ngồi cho nhà hàng. Người dùng sẽ sử dụng và thực hiện các thao tác trên sơ đồ này.

#### 2.1.1.2. Gọi món bằng thiết bị

Phần mềm Quản lý Nhà hàng giúp sắp xếp các yêu cầu đặt chỗ cũng như sơ đồ bố trí phục vụ trong nhà hàng (ghép bàn, tách bàn, chuyển bàn,…). Quản lý quy trình phục vụ nhiều bàn ăn cũng một lúc, thông tin gọi món cũng như thời gian khách gọi món.

+ Hỗ trợ gọi món bằng phần mềm.

+ In phiếu yêu cầu món ăn đến thu ngân để thu ngân nhập liệu vào hóa đơn. Sau đó, thu ngân sẽ chuyển phiếu yêu cầu món ăn đến bộ phận bếp.

+ Quản lý nhà hàng đơn giản và chính xác.

#### 2.1.1.3. Quản lý thực đơn của nhà hàng

+ Phân loại từng loại thực đơn, như: khai vị, lẩu, nướng, gỏi, đồ uống, khai vị…

+ Quản lý thực đơn theo món ăn, bao gồm: tên món ăn, giá bán, đơn vị tính.

#### 2.1.1.4. Quản lý kho nguyên liệu

+ Cho phép nhập kho theo: tên hàng, đơn vị tính, tên nhà cung cấp thực phẩm, số điện thoại, email, địa chỉ của nhà cung cấp.

#### 2.1.1.5. Quản lý việc thu chi của nhà hàng

+ Quản lý chi tiết những lần chuyển tiền lương cho nhân viên.

+ Quản lý chi tiết những lần chi trả cho việc nhập nguyên liệu.

+ Quản lý chi tiết các hóa đơn đã xác nhận thanh toán theo ngày, theo tuần, theo tháng.

#### 2.1.1.6. Quản lý nhân sự

+ Quản lý thông tin nhân viên của nhà hàng, bao gồm: Mã nhân viên, họ tên, giới tính, ngày tháng năm sinh, số chứng minh nhân dân,số điện thoại, địa chỉ, vị trí làm việc, lương cơ bản, tiền thưởng.

+ Tạo lịch làm việc cho nhân viên và theo dõi lịch làm việc của nhân viên theo ca (sáng, chiều, tối).

#### 2.1.1.7. Bán hàng

Phần mềm Quản lý Nhà hàng giúp:

+ Nhập món ăn khách gọi.

+ Phần mềm tự động cộng tiền các món ăn.

+ Hỗ trợ ghép bàn trong trường hợp nhóm khách đi với số lượng lớn.

#### 2.1.1.8. Phân quyền người dùng

Có 2 loại người dùng:

+ Admin: tạo tài khoản đăng nhập và phân quyền quản trị chức năng cho nhân viên của từng bộ phận. Ngoài ra, admin có thể:

* Quản lý danh mục và món ăn: xem, sửa, xóa các danh mục và món ăn. Các món ăn trong danh mục bị xóa sẽ được chuyển đến “Danh mục khác”.
* Quản lý voucher giảm giá: thêm, sửa, xóa các voucher giảm giá.
* Quản lý bàn ăn: tăng/giảm số lượng bàn ăn, đặt tên bàn ăn cho từng khu vực.
* Quản lý tài khoản: thêm hoặc xóa và reset mật khẩu cho các tài khoản có trong hệ thống.
* Quản lý nhân viên: thêm, sửa, xóa thông tin nhân viên. Tính lương: lương của mỗi ca làm việc sẽ khác nhau (sáng, chiều, tối). Tổng lương sẽ tính bằng tổng số ca làm việc nhân với mức lương cơ bản của mỗi loại ca và cộng thêm tiền thưởng (nếu có).
* Thống kê doanh thu: xem lại thu nhập trong một khoảng thời gian tùy chỉnh (theo ngày, tuần hoặc tháng).

+ Người được phân quyền chỉ được phép thao tác trên những quyền đã được phân quyền.

#### 2.1.1.9. Thống kê, báo cáo

Phần mềm Quản lý Nhà hàng hỗ trợ quản lý:

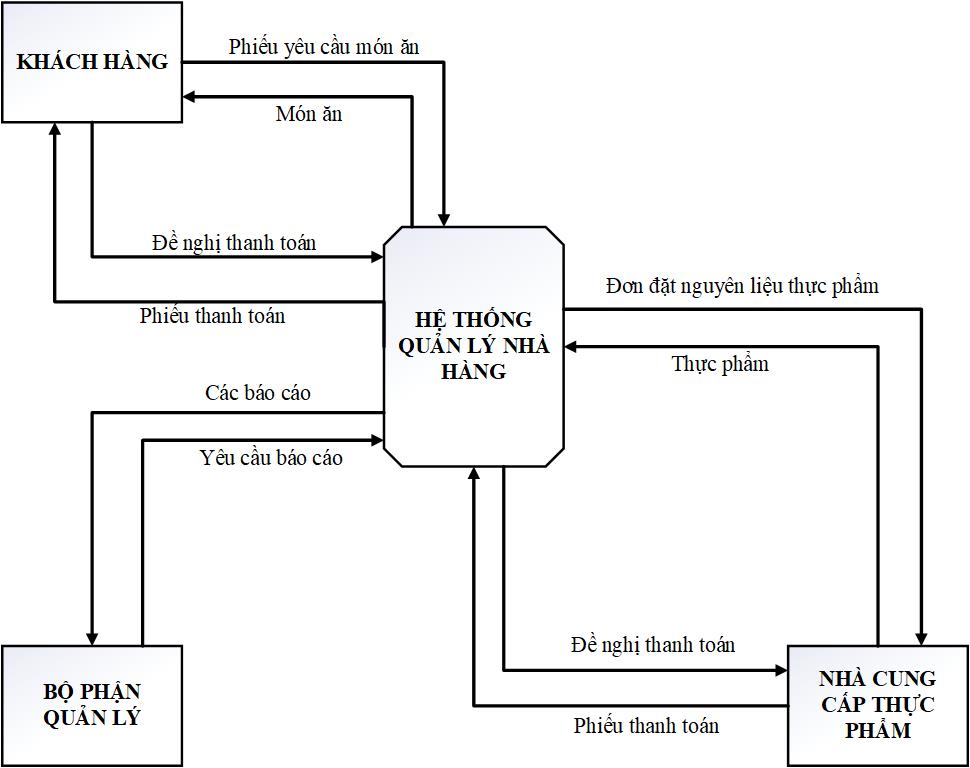
+ Thống kê tổng doanh thu theo ngày, theo tuần, theo tháng.

+ Bộ lọc giúp quản lý tìm kiếm doanh thu theo khoảng thời gian cụ thể.

## 2.2. Thiết kế hệ thống

### 2.2.1. Thiết kế kiến trúc

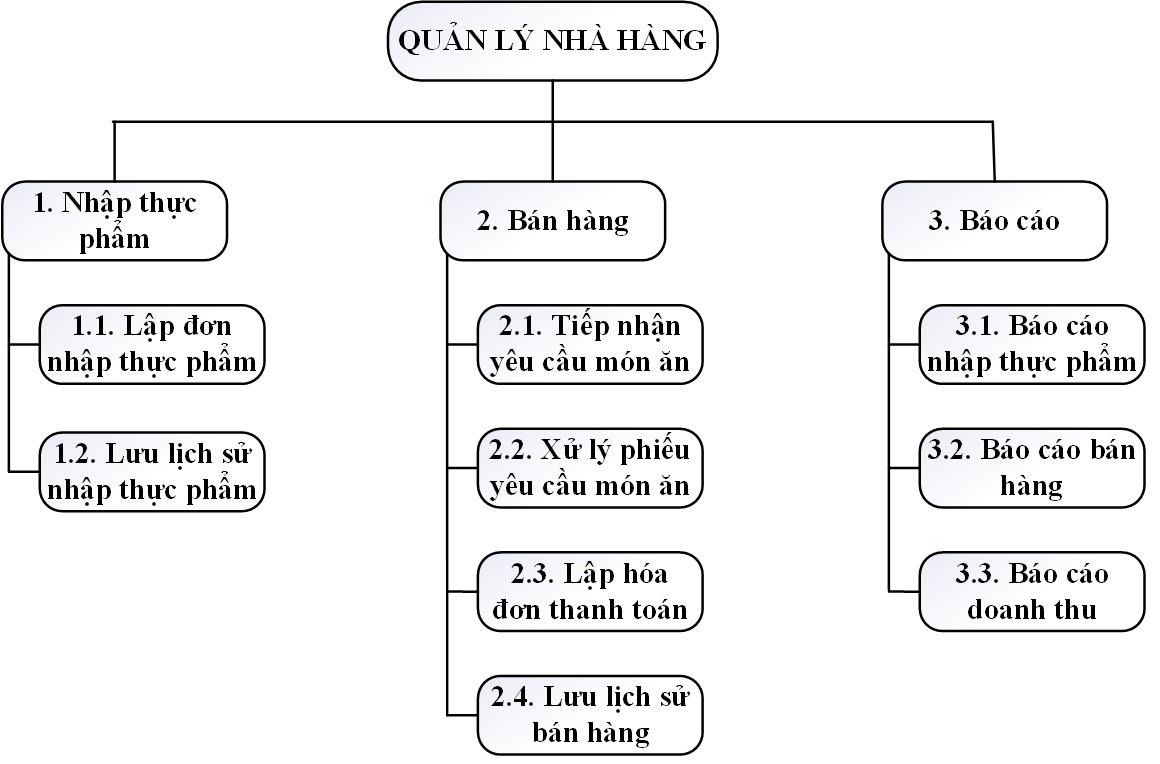
#### 2.2.1.1. Sơ đồ ngữ cảnh



Hình : Sơ đồ ngữ cảnh

#### 2.2.1.2. Xây dựng sơ đồ phân rã chức năng

##### Sơ đồ phân rã chức năng



Hình : Sơ đồ phân rã chức năng

##### Mô tả chi tiết chức năng

###### b.1. Nhập thực phẩm

- Lập đơn nhập thực phẩm: khi nhà hàng có nhu cầu nhập thực phẩm thì nhà hàng sẽ lập và gửi đơn nhập thực phẩm đến nhà cung cấp.

- Lập phiếu nhập thực phẩm: lập phiếu nhập thực phẩm và chuyển tiền cho nhà cung cấp.

- Lưu lại lịch sử nhập thực phẩm.

###### b.2. Bán hàng

- Tiếp nhận yêu cầu món ăn: nhân viên phục vụ sẽ tiếp nhận yêu cầu món ăn của khách hàng.

- Xử lý phiếu yêu cầu món ăn: sau khi bộ phận thu ngân nhập các món ăn vào hóa đơn thì phiếu yêu cầu món ăn được chuyển đến bộ phận bếp để chế biến.

- Lập hóa đơn thanh toán: xuất hóa đơn và thu tiền của khách.

- Lưu lịch sử bán hàng.

###### b.3. Báo cáo

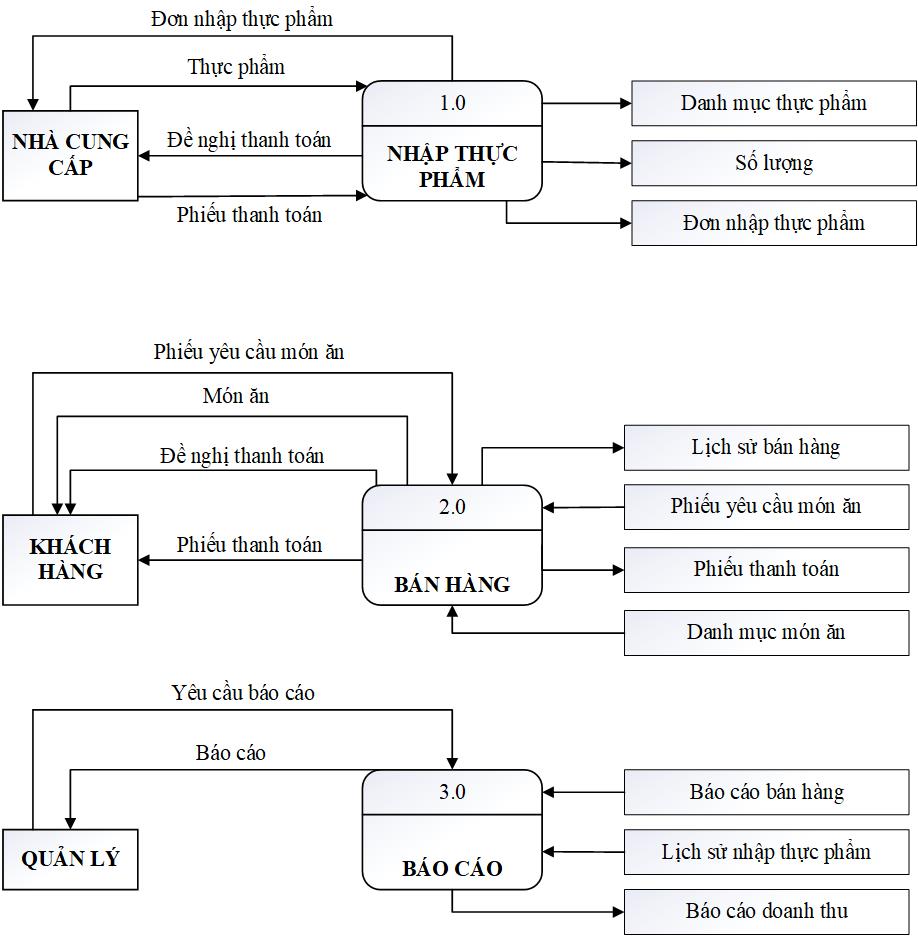
- Báo cáo nhập thực phẩm: quản lý hoặc thu ngân lập báo cáo về tình hình nhập thực phẩm.

- Báo cáo bán hàng: quản lý hoặc thu ngân sẽ lập báo cáo về tình hình bán hàng (theo ngày, theo tuần, theo tháng).

- Báo cáo doanh thu: quản lý hoặc thu ngân sẽ lập báo cáo về doanh thu (theo ngày, theo tuần, theo tháng).

### 2.2.2. Thiết kế chi tiết

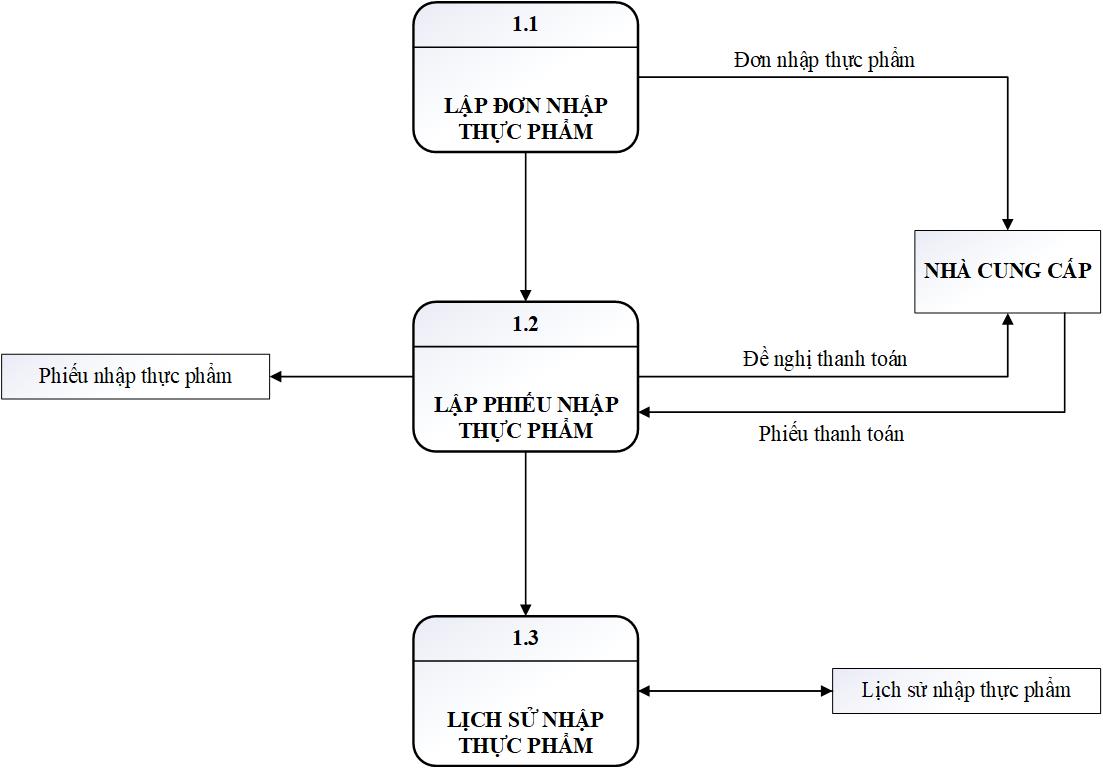
#### 2.2.2.1. Sơ đồ dữ liệu luồng mức 0



Hình : Sơ đồ dữ liệu luồng mức 0

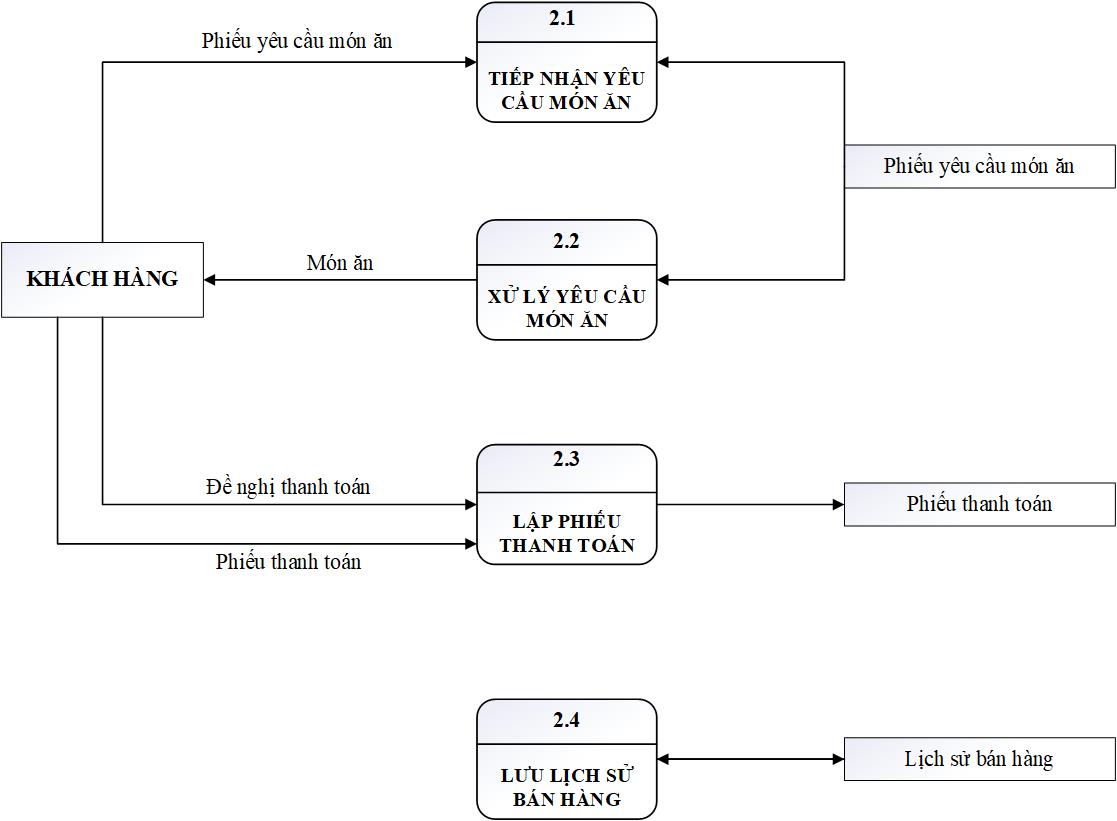
#### 2.2.2.2. Sơ đồ dữ liệu luồng mức 1

##### a. Sơ đồ luồng dữ liệu của tiến trình 1 – Nhập thực phẩm



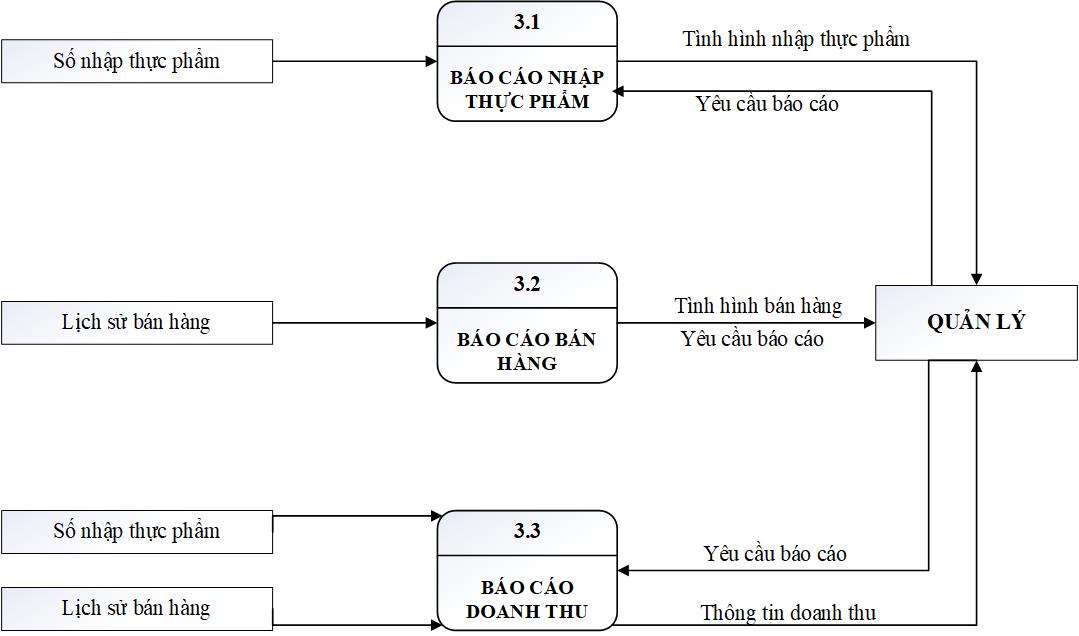
Hình : Sơ đồ luồng dữ liệu của tiến trình “Nhập thực phẩm”

##### b. Sơ đồ luồng dữ liệu của tiến trình 2 – Bán hàng



Hình : Sơ đồ luồng dữ liệu của tiến trình – “Bán hàng”

##### c. Sơ đồ luồng dữ liệu của tiến trình 3 – Báo cáo



Hình : Sơ đồ luồng dữ liệu của tiến trình – “Báo cáo”

### 2.2.3. Thiết kế giao diện

#### 2.2.3.1. Giao diện chính

#### 2.2.3.2. Giao diện đăng nhập hệ thống

#### 2.2.3.3. Giao diện cập nhật dữ liệu

##### a. Cập nhật nhân viên

##### b. Cập nhật danh mục món ăn

##### c. Cập nhật món ăn

##### d. Cập nhật nhà cung cấp

#### 2.2.3.4. Hóa đơn

#### 2.2.3.5. Giao diện báo cáo

## 2.3. Hiện thực hóa hệ thống

### 2.3.1. Sơ lược về đồ án

#### 2.3.1.1. Mô hình ba tầng

Phần mềm Quản lý Nhà hàng được nhóm thực hiện theo mô hình ba tầng, gồm: Presentation layer, Business layer và Data layer.

- Presentation layer: chứa các form để người dùng sử dụng và thao tác trên đó.

- Business layer: chứa các class, các class này sẽ chứa các phương thức gọi đến data layer.

- Data layer: giao tiếp trực tiếp với database.

Mục đích sử dụng mô hình ba tầng:

* Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
* Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
* Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại  khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.
* Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
* Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

#### 2.3.1.2. Giao diện của phần mềm

Phần mềm Quản lý Nhà hàng của nhóm thực hiện bao gồm các form làm việc như sau:

- Form Hóa đơn: frmHoaDon.cs

- Form Login: frmLogin.cs

- Form Quản lý: frmQuanLy.cs

- Form Quản lý tài khoản: frmQuanLyAccount.cs

- Form Bàn ăn: frmBanAn.cs

- Form Quản lý bàn ăn: frmQuanLyBanAn.cs

- Form Quản lý món ăn: frmQuanLyMonAn.cs

- Form Quản lý nhân viên: frmQuanLyNhanVien.cs

- Form Giảm giá: frmVoucher.cs

- Form Thống kê: frmThongKe.cs

- Form Thông tin tài khoản: frmThonTinTaiKhoan.cs

- Form Công việc: frmWork.cs

- Form Quản lý Công việc: frmQLCongViec.cs

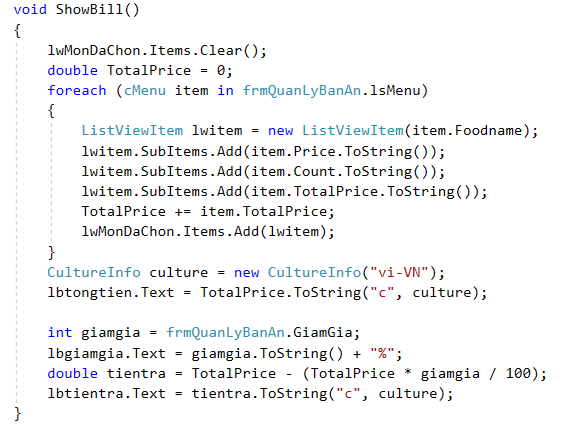
### 2.3.9. Mô hình ba tầng

Trong đó, mô hình ba lớp là phổ biến nhất. Ba lớp này là: Presentation, Business Logic và Data Access. Các lớp này sẽ giao tiếp với nhau thông qua các dịch vụ (service) mà mỗi lớp cung cấp để tạo nên ứng dụng.

#### 2.3.9.1. Presentation Layer

Lớp này làm nhiệm vụ giao tiếp với người dùng để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng.

##### a. Form Hóa đơn



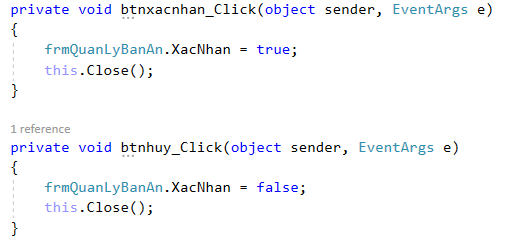
Trong hóa đơn, sẽ tạo một ListView bao gồm tên món, giá, số lượng và tổng tiền. Bên cạnh đó, còn có thông tin về nhân viên phục vụ, số hóa đơn và một số nút chức năng.

- Ban đầu, xóa trống các trường trong ListView và TotalPrice bằng 0.

- Lấy thông tin từ form Quản lý bàn ăn để truyền dữ liệu vào ListView. Các thông tin như sau: Tên món ăn, đơn giá (Price), số lượng (count), tổng tiền (TotalPrice),…

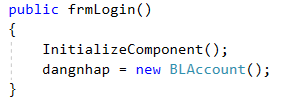
- TotalPrice được ép kiểu string để đưa vào label và xuất ra màn hình.

- Ngoài ra, form còn có đối tượng Giảm giá: nếu nhân viên có nhập phiếu giảm giá trùng khớp với CSDL thì số tiền phải trả của khách hàng được tính theo công thức: tientra = TotalPrice – (TotalPrice \* giamgia / 100).

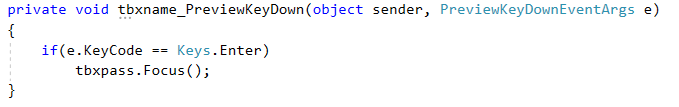


Trên đây là hai nút chức năng để xác nhận lại các thông tin trong hóa đơn. Nếu là true thì hóa đơn được xác nhận, ngược lại thì hóa đơn sẽ bị hủy.

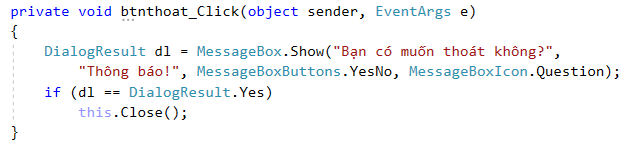
##### b. Form Login



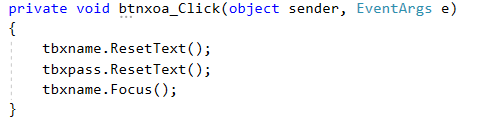
Khai báo và khởi tạo đối tượng dangnhap theo BLAccount của Business Layer.



Tạo event PreviewKeyDown cho textbox username: Khi người dùng gõ xong tên đăng nhập và nhấn Enter thì con trỏ chuột sẽ tự động di chuyển vào textbox password.



Khi nhấn vào nút thoát, sẽ có một hộp thoại (MessageBox) xuất hiện và hỏi để xác nhận người dùng có “Chắc chắn thoát hay không?”. Nếu người dùng chọn Yes thì form Login sẽ tắt.



Khi nhấn vào nút xóa, các thông tin đang nhập ở textbox sẽ bị xóa đi. Đồng thời, con trỏ sẽ chuyển về textbox username.



Kiểm tra username và password để thực hiện đăng nhập. Ở đây, đăng nhập chỉ để phân biệt admin với nhân viên để thực hiện một số chức năng riêng, chứ không phân biệt nhân viên với nhân viên.

- Nếu đối tượng Account (của frmQuanLy) bằng null (trống) thì đối tượng isLogin = false, tức là người dùng không đăng nhập được.

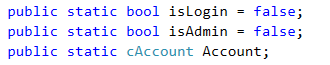
- Ngược lại, thì isLogin = true. Tiếp theo, sẽ kiểm tra tên người dùng và mật khẩu có trùng khớp với admin hay không. Nếu trùng khớp thông tin đăng nhập và đúng loại tài khoản, đối tượng isAdmin = true, tương đương với việc đăng nhập thành công và ngược lại.

Ở đây, có hai loại tài khoản:

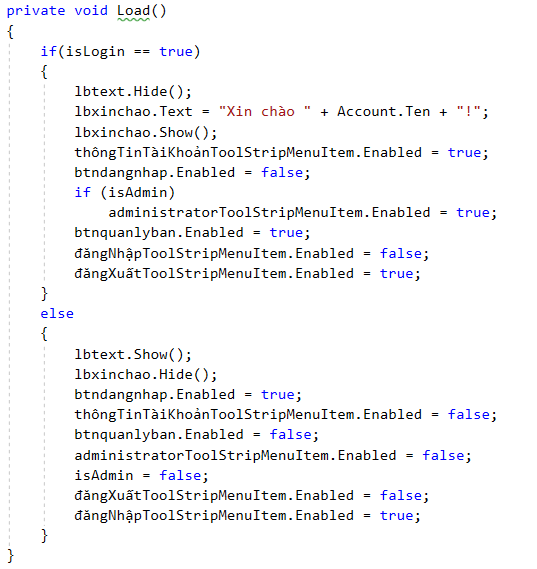
- Admin (Quản lý): Account.Type = 1.

- Nhân viên: Account.Type = 0.

##### c. Form Quản lý



Khởi tạo các đối tượng isLogin, isAdmin, Account.



Phương thức Load()

- Nếu isLogin = true: nghĩa là đăng nhập thành công.

+ Khi đó, người dùng mới có quyền truy cập vào thanh menu Thông tin tài khoản (thôngTinTàiKhoảnToolStripMenuItem.Enabled = true).

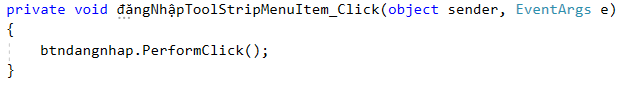
+ Nếu người dùng là admin thì người này có luôn quyền truy cập vào thanh menu administrator (administratorToolStripMenuItem.Enabled = true).

+ Người dùng có thể tương tác với nút Quản lý bàn ăn (btnquanlyban) và đăng xuất (nếu cần).

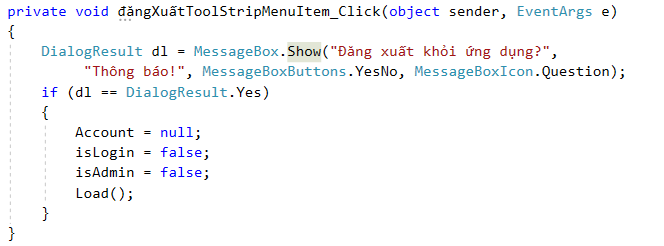
- Ngược lại, isLogin = false:

+ Đăng nhập không thành công. Khi đó, hệ thống cho phép người dùng đăng nhập lại.

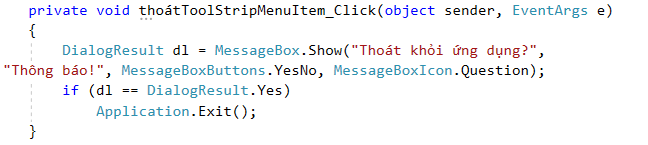
+ Bên cạnh đó, các menu về Thông tin tài khoản, Administrator và đăng xuất đều bị vô hiệu hóa.



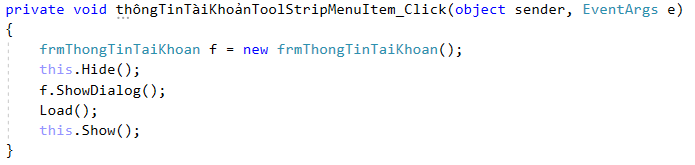
Menu đăng nhập tương tự nút đăng nhập.



Khi chọn menu đăng xuất, hộp thoại xuất hiện và hỏi xác nhận người dùng có chắc chắn “Đăng xuất khỏi ứng dụng?” hay không. Nếu người dùng chọn Yes, các đối tượng sẽ có giá trị null và false. Và tiến hành Load() lại form.

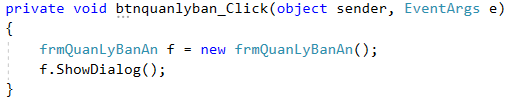


Khi chọn menu thoát, hộp thoại xuất hiện và hỏi xác nhận người dùng có muốn thoát hay không. Nếu người dùng chọn Yes thì ứng dụng sẽ thoát.





Khi nhấp chọn vào các menu thì các form tương ứng sẽ được mở ra và người dùng có thể thao tác trên đó.

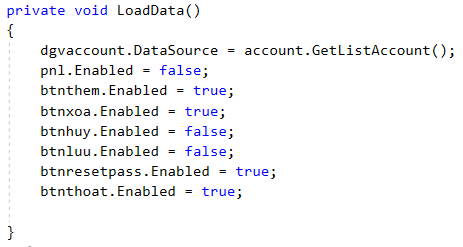


Nếu chọn nút quản lý bàn ăn thì form Quản lý Bàn ăn sẽ xuất hiện.

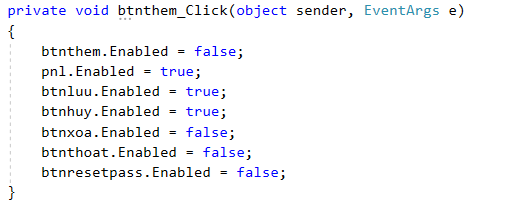
##### d. Form Quản lý Account

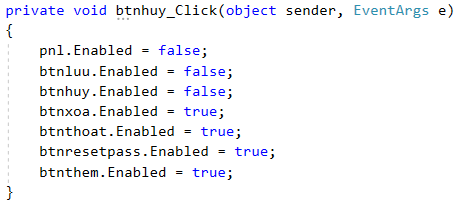


Khởi tạo đối tượng err kiểu string và account (theo class BLAccount).

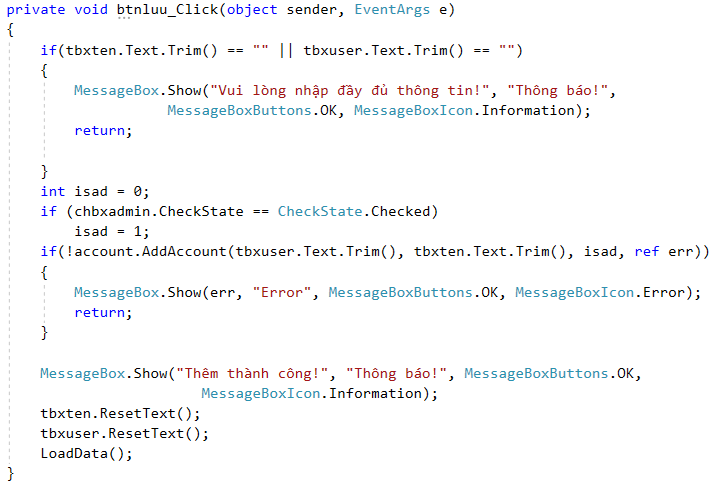


Phương thức Load Dữ liệu.





Ở hai nút này, hệ thống chỉ cho người dùng thao tác ở một số nút cần thiết.



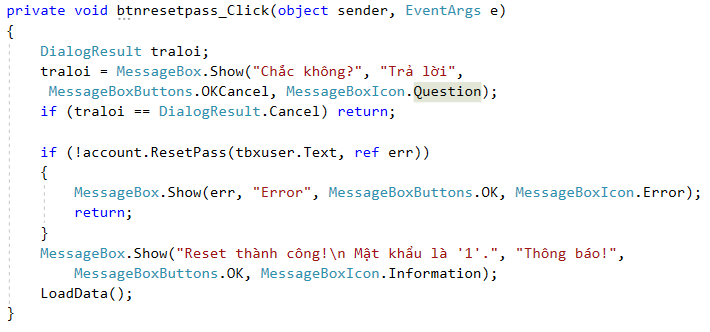
- Nếu tên hiển thị và tên đăng nhập trống thì một hộp thoại sẽ xuất hiện và thông báo cho người dùng để người dùng có thể nhập đầy đủ thông tin.

- Gán đối tượng isad = 0.

- Sử dụng toolbox CheckBox (chbxadmin) để đánh dấu tài khoản có phải là admin hay không. Nếu chbxadmin được check thì trường admin của datagridview account sẽ bằng 1 (isad = 1). Và ngược lại thì bằng 0.

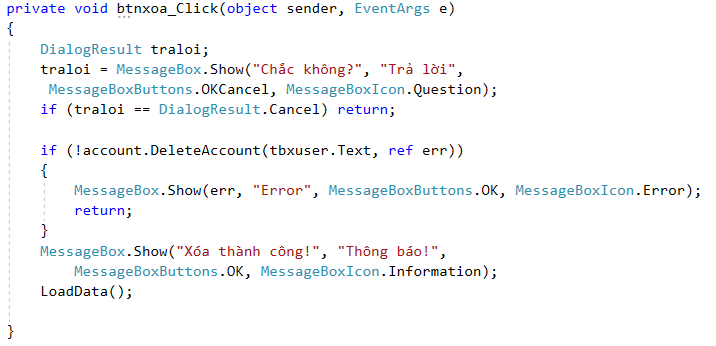
- Khi thêm tài khoản, nếu tài khoản đã tồn tại trong phương thức AddAccount của class BLAccount thì sẽ xuất hiện một hộp thoại báo lỗi. Ngược lại, có hộp thoại thông báo người dùng đã “Thêm thành công!”.

- Sau khi thêm xong, các textbox tên hiển thị và tên đăng nhập sẽ bị xóa trống. Đồng thời, hệ thống sẽ load lại dữ liệu.



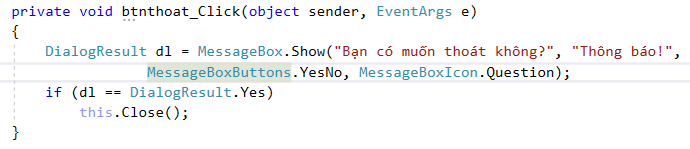
Khi nhấn nút Reset pass, hộp thoại xuất hiện để xác nhận người dùng có chắc chắn đổi mật khẩu hay không.

Nếu có, gọi phương thức ResetPass (class BLAccount) để thao tác. Sau đó, load lại dữ liệu.

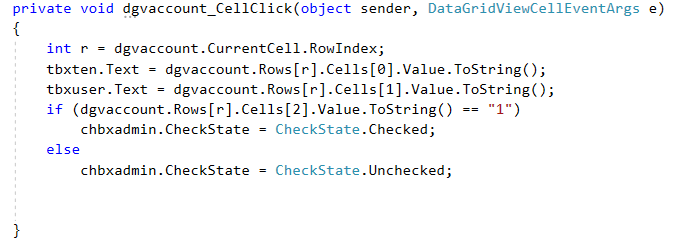


Tương tự, khi nhấn nút Xóa, hộp thoại xuất hiện để xác nhận người dùng có chắc chắn xóa tài khoản hay không.

Nếu có, gọi phương thức DeleteAccount (class BLAccount) để thao tác. Sau đó, load lại dữ liệu.



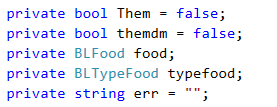
Khi nhấn nút Thoát, hộp thoại xuất hiện để xác nhận. Nếu chắc chắn muốn thoát, form sẽ tắt đi.



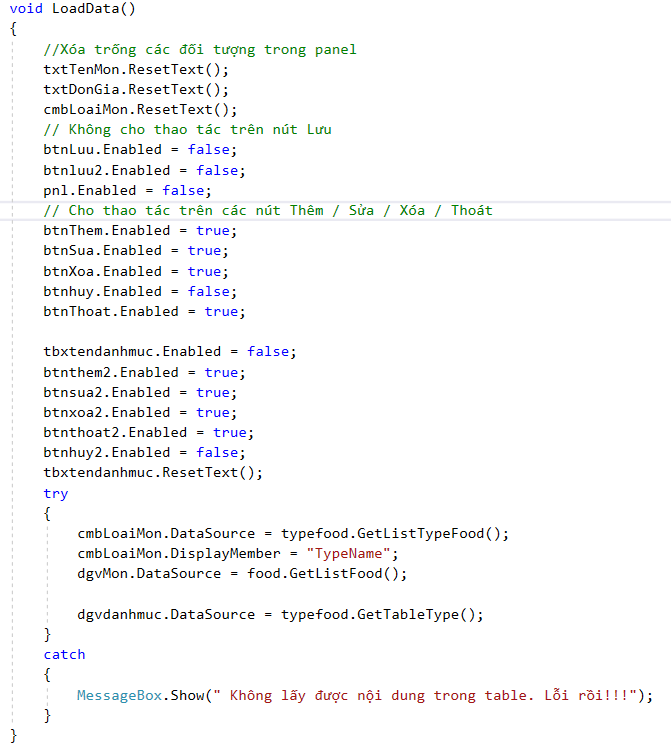
Event CellClick: khi người dùng chọn một ô bất kỳ trên datagridview thì các thông tin của các trường sẽ được đưa lên các textbox tương ứng ở phía trên.

##### e. Form Quản lý Bàn ăn

##### f. Form Quản lý Món ăn



Khai báo các đối tượng: Them, themdm (thêm danh mục món), food (class BLFood), typefood (BLTypeFood), err kiểu dữ liệu string.



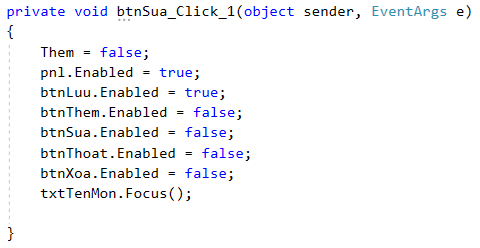
Cho phép thao tác trên một số nút cần thiết.

- Combobox loại món lấy dữ liệu từ GetListTypeFood (BLTypeFood).

- Load dữ liệu lên datagridview Món từ GetListFood (BLFood).

- Datagridview Danh mục món lấy dữ liệu từ GetTableType (BLTypeFood).

*a. Quản lý món ăn:*

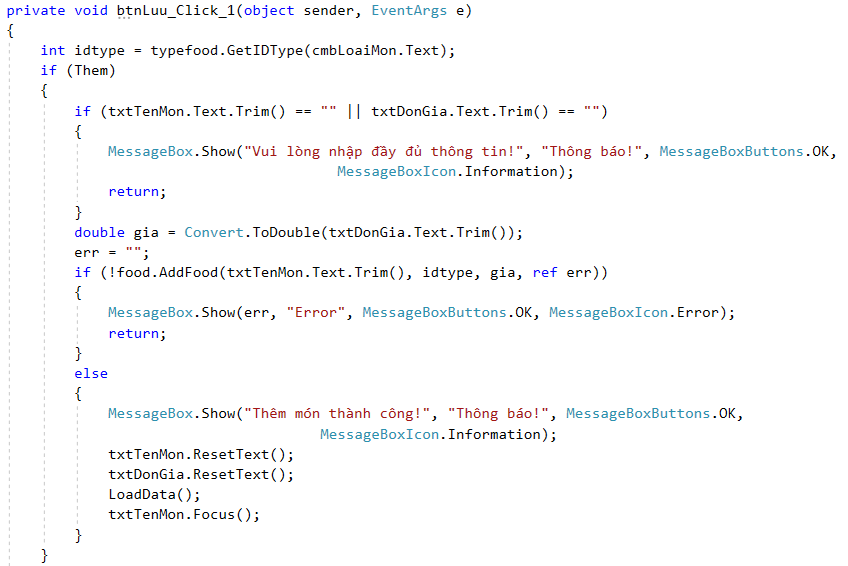




Hai nút này cho phép người dùng thao tác trên một số nút cần thiết. Con trỏ được đưa về textbox tên món nếu hoàn tất thao tác sửa, xóa trống hai textbox đơn giá và tên món nếu nhấn nút hủy.



Hệ thống xác nhận muốn xóa hay không. Nếu có, gọi phương thức DeleteFood (BLFood) để thực hiện lệnh xóa món ăn theo idfood. Và cập nhật lại dữ liệu.



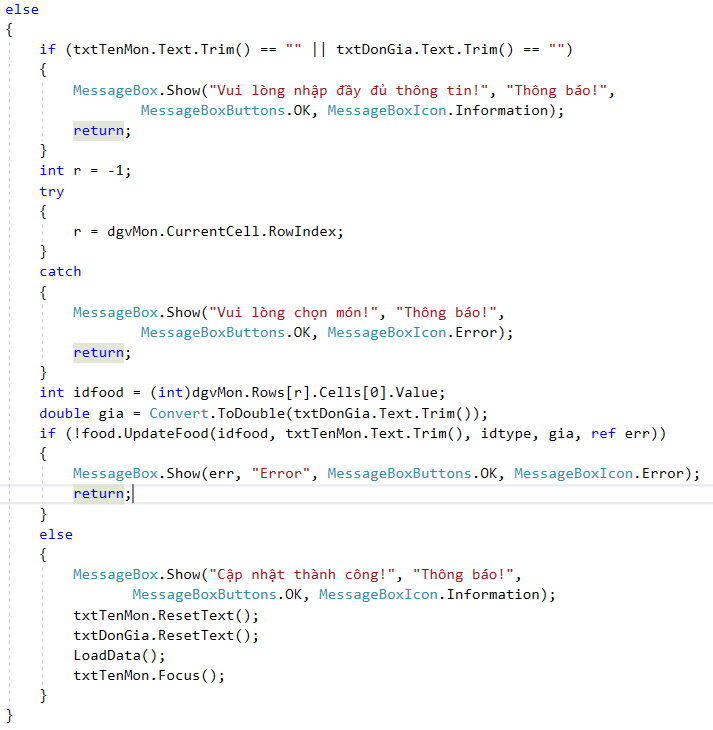
Khai báo đối tượng idtype theo phương thức GetIDType (BLFood), dữ liệu từ cmbLoaiMon.

- Đối tượng Them = true:

+ Để lưu một món mới, người dùng phải nhập đủ thông tin và món đó không tồn tại trong danh sách hiện có.

+ Nếu món cần lưu thỏa điều kiện, hệ thống sẽ gọi phương thức AddFood (BLFood) để thêm món với các tham số như: tên món (txtTenMon), idtype, giá, err.

+ Khi thêm món thành công, load lại dữ liệu, xóa trống các textbox và đưa con trỏ về txtTenMon.



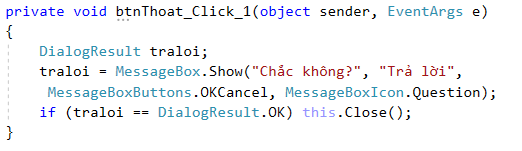
- Đối tượng Them = false:

+ Kiểm tra đã nhập đầy đủ thông tin hay chưa.

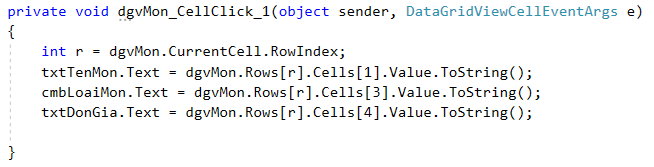
+ Khởi tạo đối tượng r để lấy dữ liệu từ dgvMon lên các textbox, tiện cho việc update thông tin của món ăn.

+ Nếu món cần sửa thỏa điều kiện, hệ thống sẽ gọi phương thức UpdateFood (BLFood) để thêm món với các tham số như: idfood, tên món (txtTenMon), idtype, giá, err.

+ Khi update món thành công, load lại dữ liệu, xóa trống các textbox và đưa con trỏ về txtTenMon.

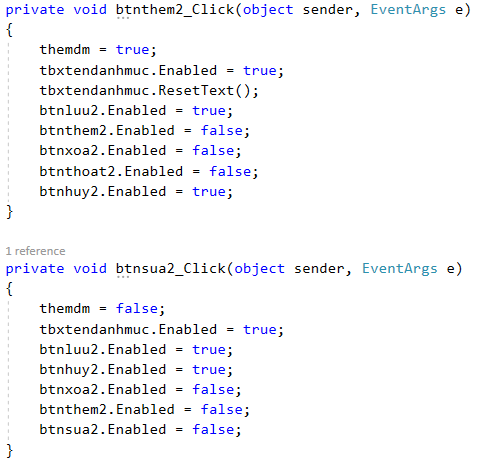


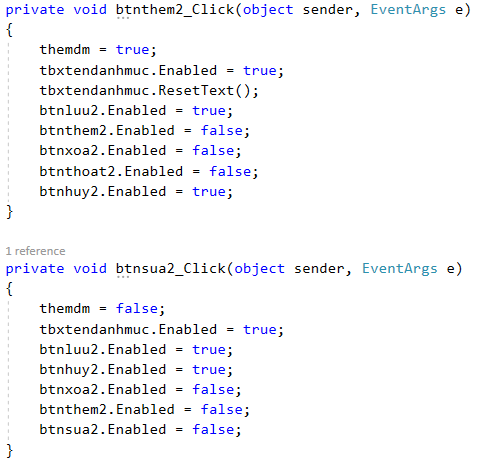
Nút Thoát để xác nhận người dùng muốn thoát hay không.

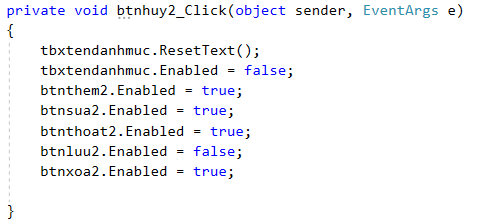


Event CellClick để lấy dữ liệu từ datagridview lên textbox và combobox ở trên.

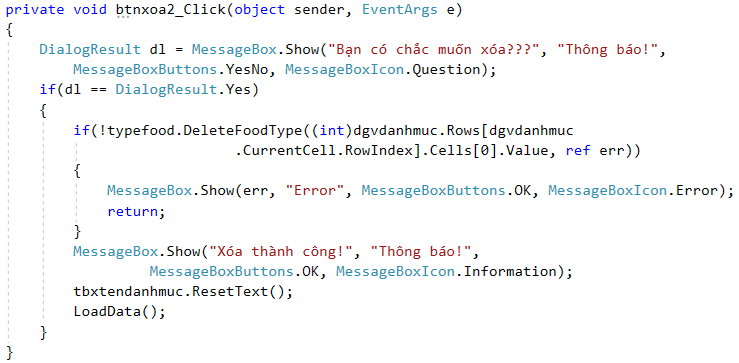
*b. Quản lý danh mục món:*



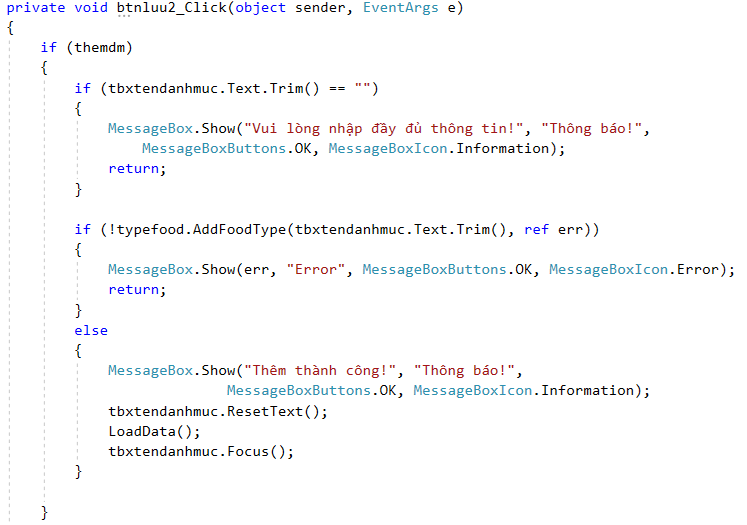




Ba nút này cho phép người dùng thao tác trên một số nút cần thiết.



Gọi phương thức DeleteFoodType (BLTypeFood) để tiến hành xóa danh mục món với tham số: idtype.



- Đối tượng themdm = true:

+ Để lưu một danh mục món mới, người dùng phải nhập đủ thông tin và danh mục món đó không tồn tại trong danh sách hiện có.

+ Nếu danh mục món cần lưu thỏa điều kiện, hệ thống sẽ gọi phương thức AddFoodType (BLTypeFood) để thêm danh mục món với các tham số như: typename, err.

+ Khi thêm món thành công, load lại dữ liệu, xóa trống các textbox và đưa con trỏ về txttendanhmuc.

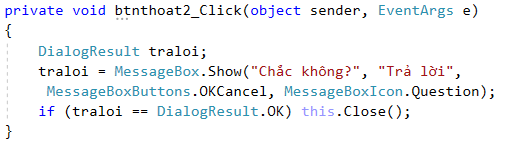


- Đối tượng themdm = false:

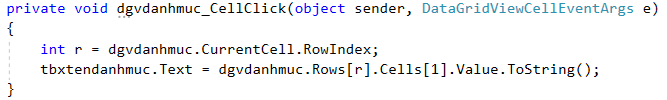
+ Khởi tạo đối tượng idtype để lấy dữ liệu từ dgvdanhmuc lên các textbox, tiện cho việc update thông tin của danh mục món ăn.

+ Nếu món cần sửa thỏa điều kiện, hệ thống sẽ gọi phương thức UpdateFoodType (BLTypeFood) để thêm món với các tham số như: idtype, name, err.

+ Khi update món thành công, load lại dữ liệu, xóa trống các textbox và đưa con trỏ về txttendanhmuc.



Nút Thoát để xác nhận người dùng muốn thoát hay không.

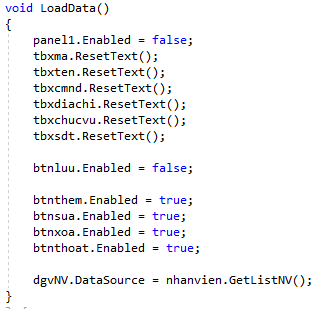


Event CellClick để lấy dữ liệu từ datagridview lên textbox ở trên.

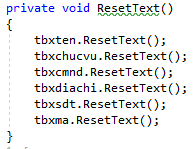
##### g. Form Quản lý nhân viên



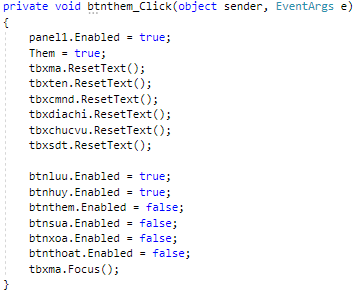
Khởi tạo một số đối tượng cần thiết.



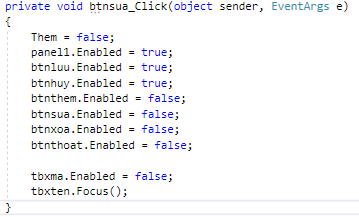
Xóa trống các textbox và gán panel1 = false, cho thao tác trên các nút thêm, sửa, xóa, thoát; không cho thao tác trên nút lưu.



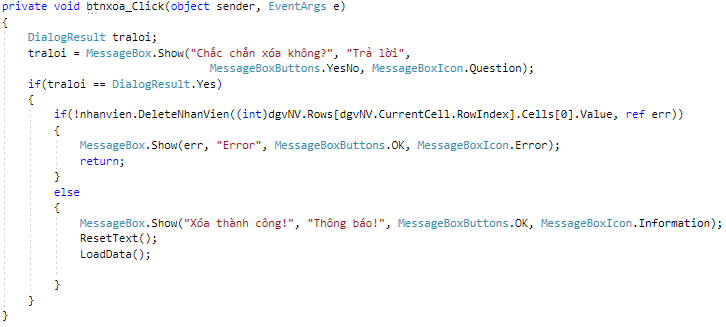
Xóa trống các textbox tbxten, tbxchucvu, tbxcmb, tbxdiachi, tbxsdt, tbxma.



Gán panel1 = true, them = true và xóa trống các textbox. Cho phép thao tác trên các button lưu, hủy và không cho tháo tác trên một số button như thêm, sửa, xóa, thoát. Đưa con trỏ về tbxma.



Gán them = true, cho phép thao tác trên panel1, btnluu, btnhuy. Không cho thao tác trên btnthem, btnsua, btnxoa, btnthoat, tbxma. Đưa con trỏ về tbxten.



Khi click vào btnxoa lập tức hiện ra messagebox nhắc nhở có đồng ý xóa hay không, nếu đồng ý thì xóa trống các dữ liệu và loadData, ngược lại thì sẽ không xóa.



* Them = true:

+ Đầu tiên sẽ kiểm tra xem thông tin ở các textbox đã điền đủ chưa, nếu chưa thì sẽ show messageBox nhắc nhở nhập thông tin.

+ Gán luna = 1, nếu kiểm tra rabNam = true thì luna sẽ bằng 0.

+ Kiểm tra các thông tin nhập có phù hợp với yêu cầu không, nếu có thì thông báo nhập thành công, bắt đầu reset lại các textbox và loadData.

* Them = false:

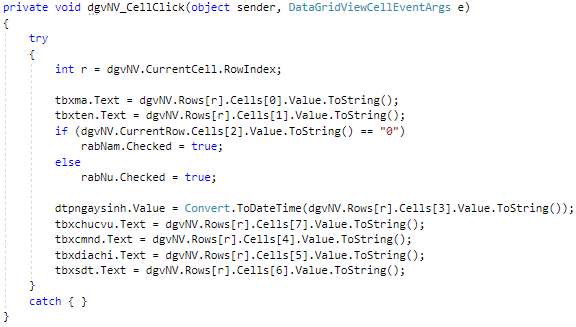
+ Đầu tiên sẽ kiểm tra xem thông tin ở các textbox đã điền đủ chưa, nếu chưa thì sẽ show messageBox nhắc nhở nhập thông tin.

+ Lấy id từ dgvNV, gán luna = 1, nếu kiểm tra rabNam = true thì luna sẽ bằng 0.

+ Kiểm tra các thông tin nhập có phù hợp với yêu cầu không, nếu có thì thông báo nhập thành công, bắt đầu reset lại các textbox và loadData.



Khi click vào btnhuy thì sẽ bất đầu load lại data trong form.

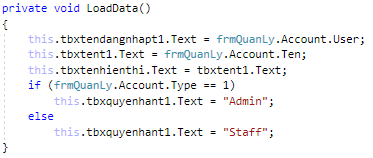


Tạo thêm event cellClick cho dgvNV: khi click vào một ô bất kì của dgvNV thì dữ liệu sẽ được thể hiện ở textbox tương ứng.

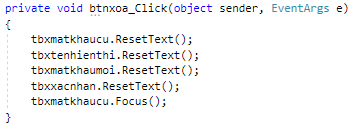
##### h. Form Thông tin tài khoản



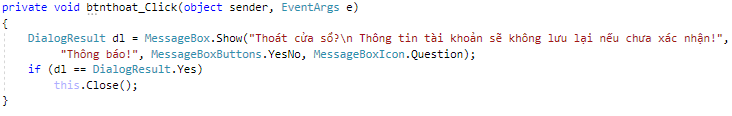
Khởi tạo đối tượng error kiểu string và blaccout (theo class BLAccount).



Gán các dữ liệu vào các textbox, lấy dữ liệu từ frmQuanLy. Nếu Account = 1 thì là admin ngược lại là staff.



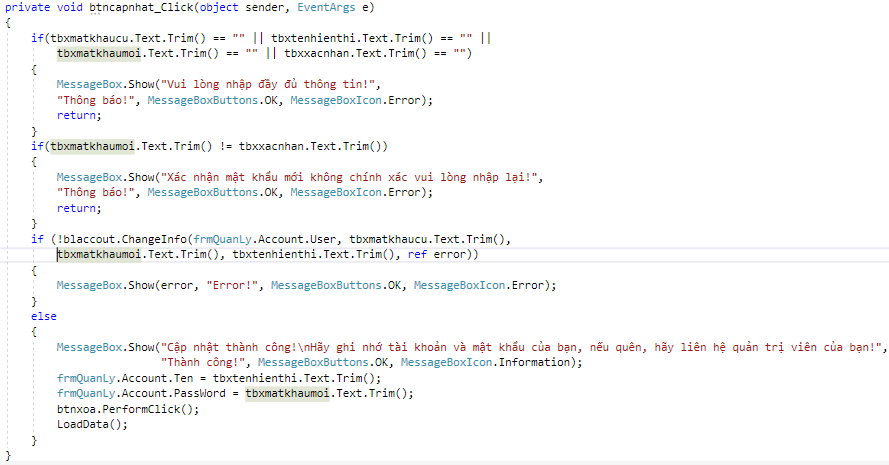
Xóa trống các textbox mật khẩu cũ, tên hiển thị, mật khẩu mới, xác nhận. Và đưa con trỏ về tbxmatkhaucu.



Xuất hiện messagebox nhắc nhở, nếu đồng ý thì close và ngược lại thì giữ nguyên.



Đối với btn trove thì thực hiện đóng form.

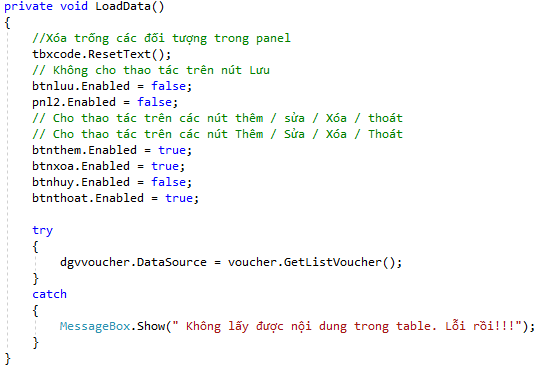


* Khi các dữ liệu chưa được điền đầy đủ thông tin thì xuất hiện mesagebox nhắc nhở, tiếp tục khi thông tin đã đầy đủ.
* Nếu mật khẩu mới không trùng với xác nhận thì thông báo nhập lại chính xác.
* Nếu mật khẩu mới trùng với mật khẩu cũ, tên hiển thị trùng với user trên frmQuanLy thì thông báo error. Ngược lại thì thông báo thành công. Xóa trống các thông tin và loadData.

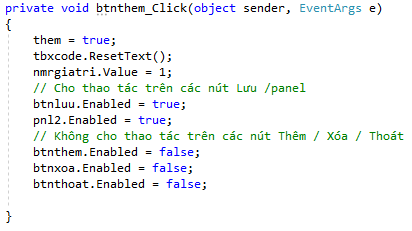
##### i. Form Voucher



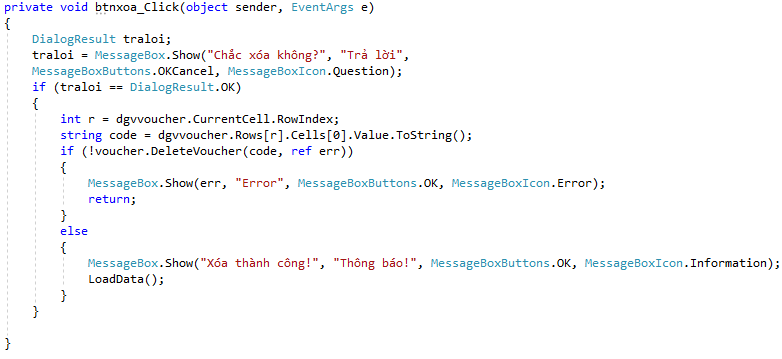
Khởi tạo đối tượng err kiểu string và voucher (theo class BLvoucher).



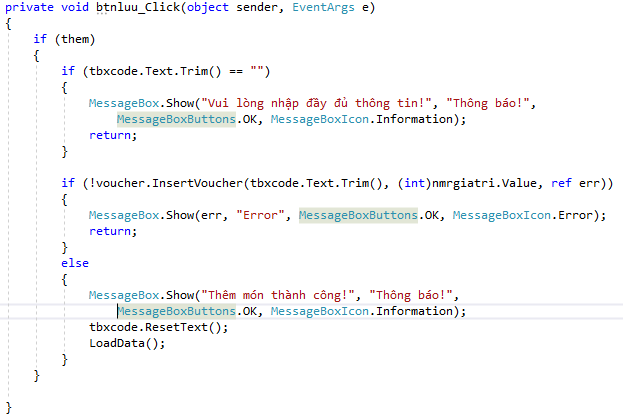
Xóa trống các đối tượng trong panel, không cho thao tác trên nút lưu, Cho thao tác trên nút thêm, sửa, xóa, thoát. Lấy thông tin voucher từ phương thức GetListVoucher().



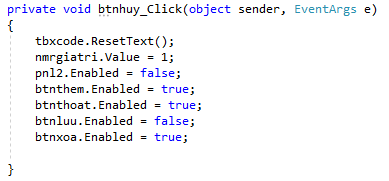
Gán Them = true, xóa trống đối tượng trên tbxcode. Cho thao tác trên nút lưu, panel và không cho thao tác trên các nút thêm, xóa, thoát.



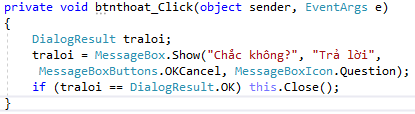
Khi click vào btnXoa sẽ hiện ra mesagebox nhắc nhở trước khi xóa. Nếu đồng ý xóa thì tiếp tục xét xem voucher có tồn tại trong list voucher hay không bằng cách gọi phương thức DeleteVoucher.



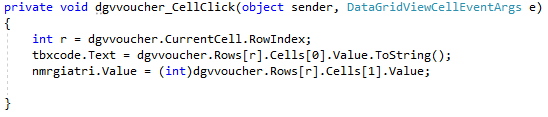
Nếu như textbox code trống thì sẽ hiện messagebox thông báo nhập thông tin rồi mới tiếp tục. Bằng phương thức insertVoucher để báo lỗi và thực hiện thao tác lưu. Xóa trống tbxcode và bắt đầu loadData.



Xóa trống các đối tượng trong panel, không cho thao tác trên panel va nút lưu. Cho phép thao tác trên các nút thêm, thoát và xóa.

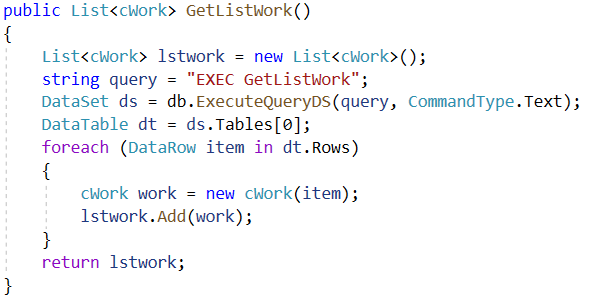


Hiện messagebox hỏi có đồng ý thoát hay không, nếu đồng ý thì thực hiện close, ngược lại vẫn mở form.



Thêm event CellClick cho dgvvoucher: khi click vào một ô trên dgvvoucher thì thông tin voucher đó sẽ được thể hiện trên tbxcode.

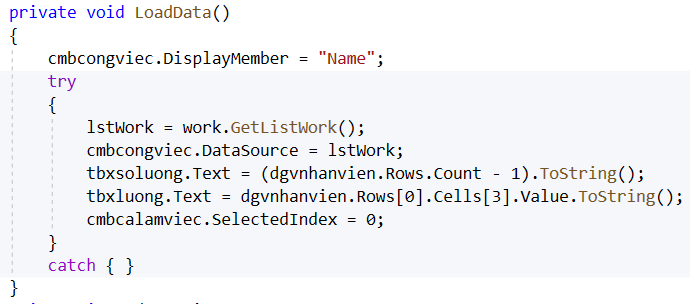
##### k. Form Công việc



- Khởi tạo List công việc.

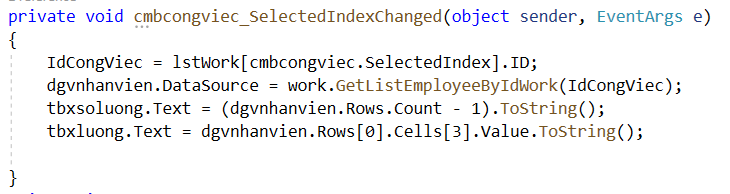
- Thực hiện câu query “EXEC GetListWork” để thực thi procedure ở SQL.

- Kết quả: trả về danh sách các nhân viên thực hiện công việc tương ứng sau khi được thêm vào list.



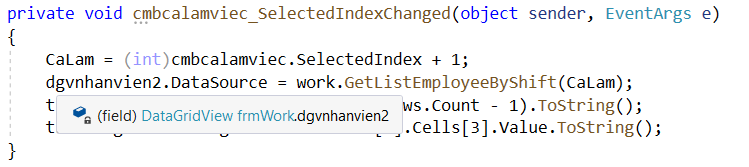
- Load dữ liệu vào combobox công việc theo danh sách công việc.

- Load số lượng nhân viên làm việc và số lương tương ứng với từng loại công việc và ca làm việc ở combobox.



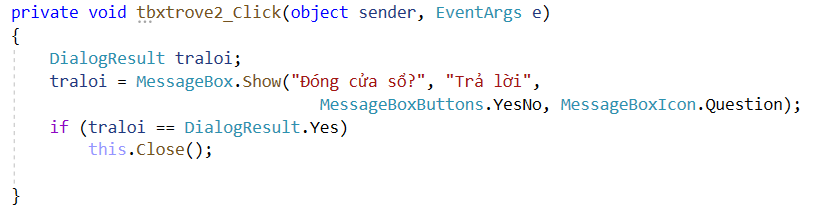
- Thêm loại công việc vào combobox.

- Các thông tin liên quan đến nhân viên thỏa điều kiện (loại công việc) sẽ được hiển thị len datagridview nhân viên.

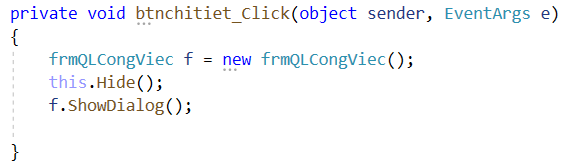


- Thêm ca làm việc vào combobox.

- Các thông tin liên quan đến nhân viên thỏa điều kiện (ca làm việc) sẽ được hiển thị len datagridview nhân viên.



- Nút nhấn Trở về: cửa sổ tin nhắn sẽ xuất hiện và người dùng xác nhận rằng có chắc chắn đóng form hay không. Nếu có thì form sẽ tắt và trở về form chính.

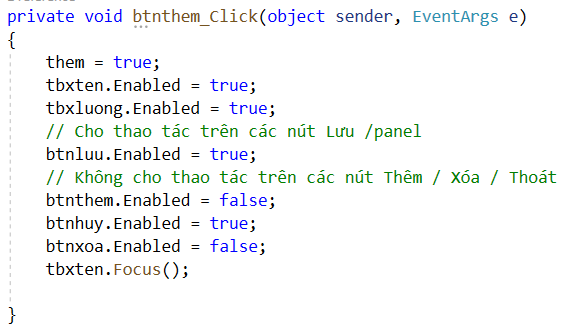


- Nút nhấn Chi tiết: truy cập vào form Quản lý công việc để xem thêm các thông tin.

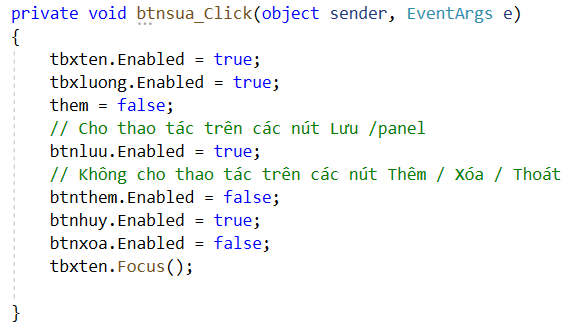
##### l. Form Quản lý công việc



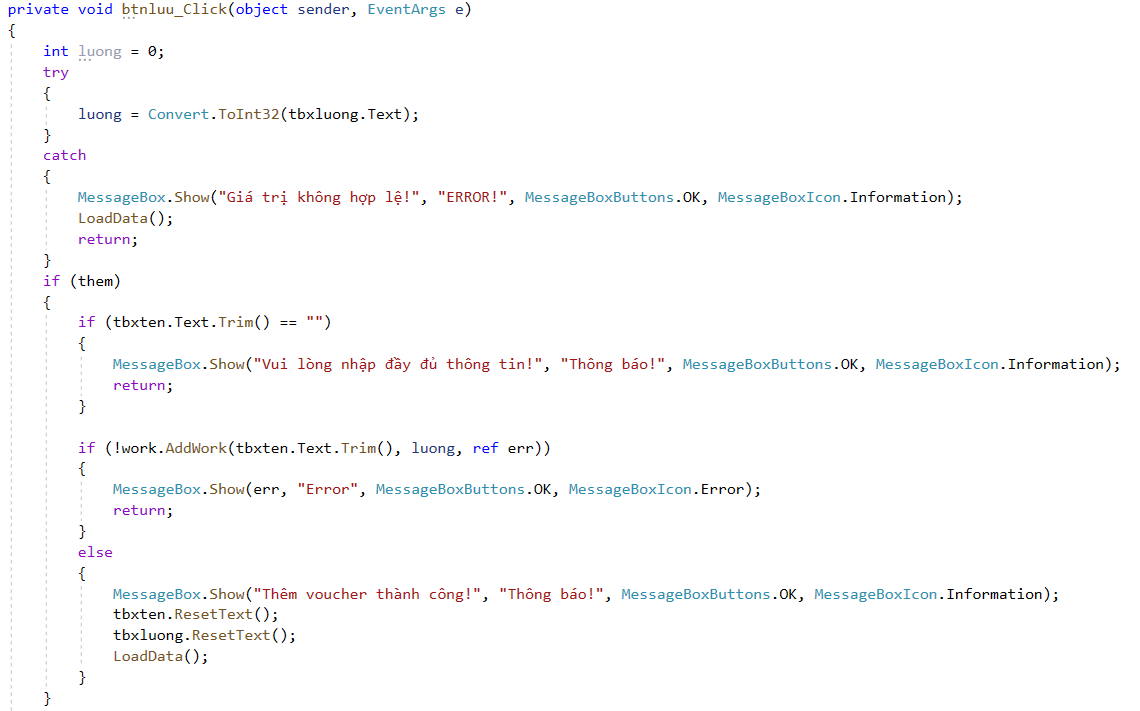
- Load dữ liệu vào các textbox, datagridview và phân quyền cho các button.



- Button Thêm: thêm các loại công việc bằng cách nhập các trường thông tin bắt buộc.

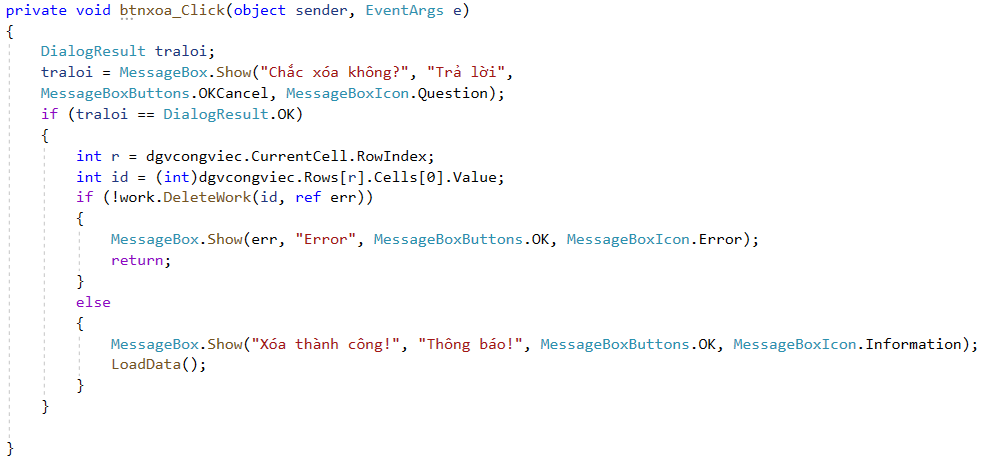


- Button Sửa: sửa thông tin của công việc khi click vào 1 ô bất kỳ của datagridview.

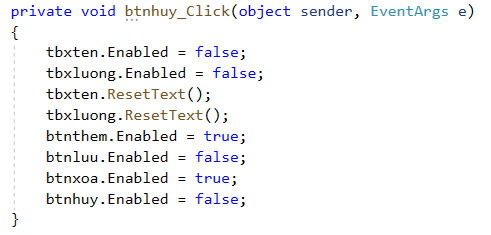




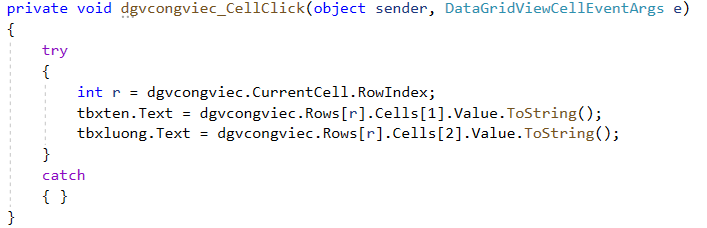
- Lưu thông tin công việc sau khi thêm hoặc sửa.



- Xóa 1 công việc bất kỳ khi click vào 1 ô của datagridview.

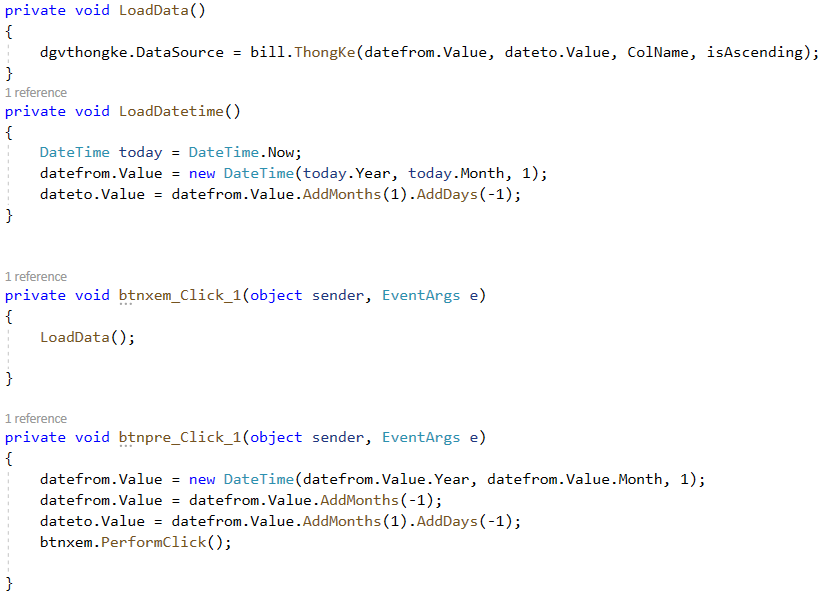


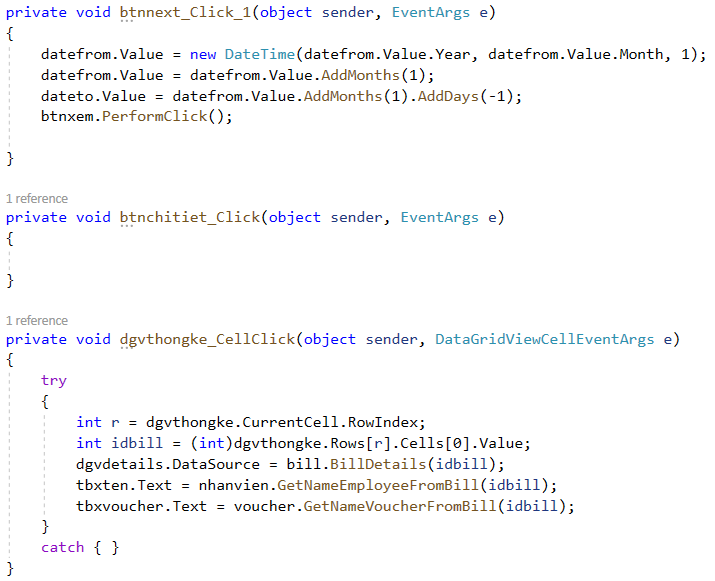
- Button Hủy: xóa trống các textbox tên công việc và lương.

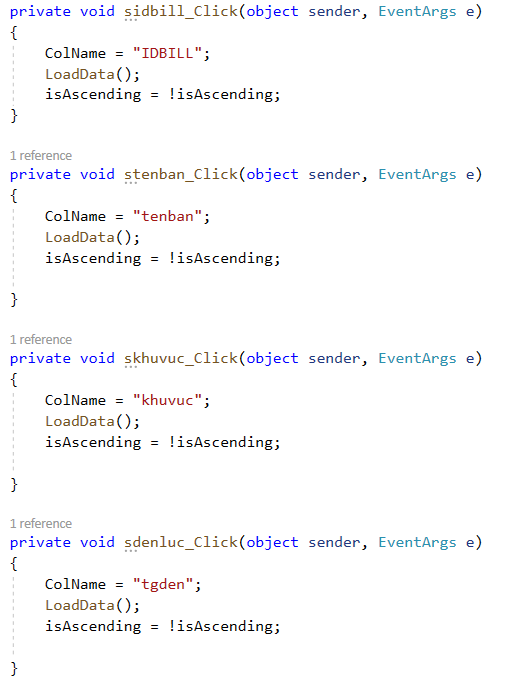


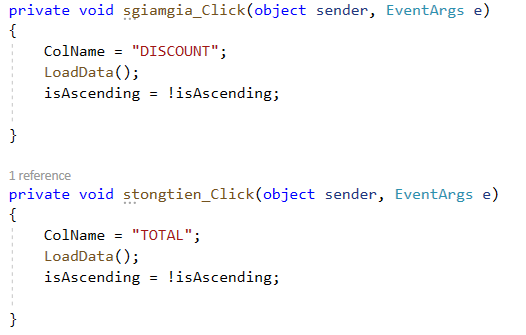
- Event CellClick: lấy thông tin từ datagridview để đưa vào 2 textbox tên công việc và lương tương ứng.

##### m. Form Thống kê

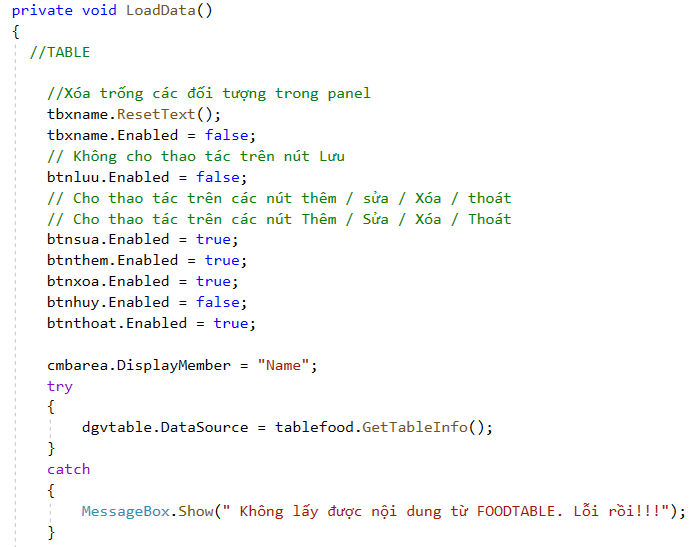


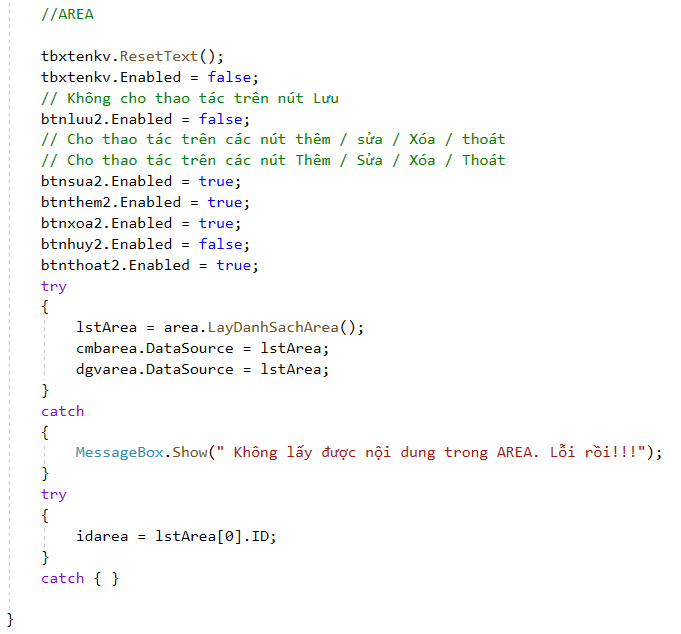


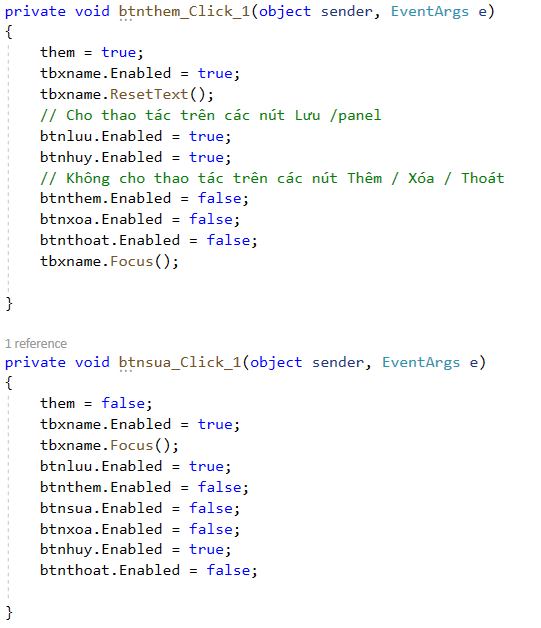


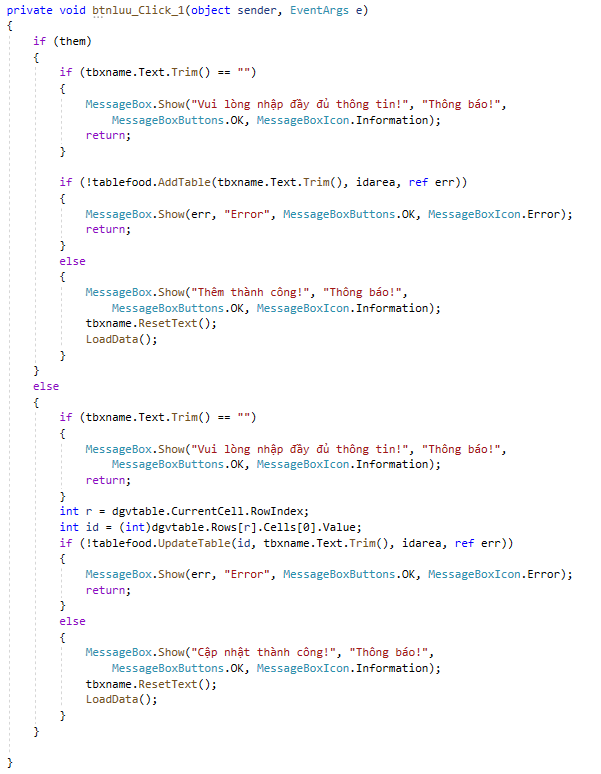


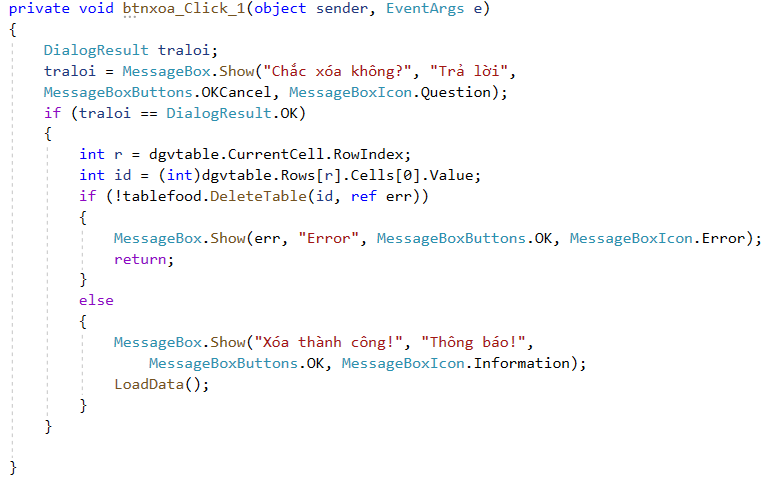
##### n. Form Bàn ăn

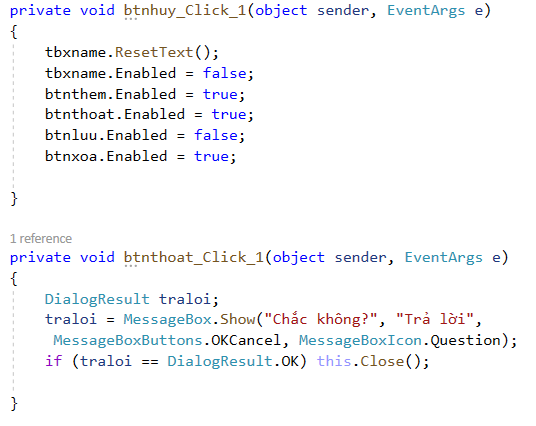












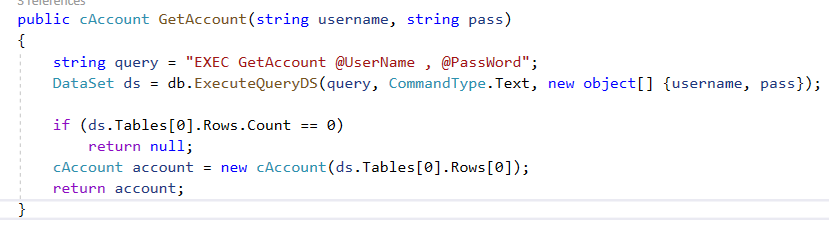
#### 2.3.9.2. Business Layer

Lớp này thực hiện các nghiệp vụ chính của he, sử dụng các dịch vụ do lớp Data Access cung cấp và vung cấp các dịch vụ cho lớp Presentation.

* **Class BLAccount**

Class BLAccount bao gồm các phương thức liên quan đến việc quản lý tài khoản cá nhân trong nhà hàng.

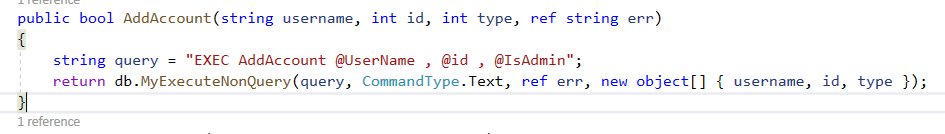
##### a. Phương thức Lấy tài khoản từ Cơ sở dữ liệu



Phương thức này giúp hệ thống lấy tài khoản từ CSDL với tham số truyền vào là username và password. Phương thức này trả về tài khoản có sẵn trong CSDL hoặc trả về null nếu tài khoản không tồn tại.

Sử dụng cú pháp: SELECT…FROM ACCOUNT WHERE…

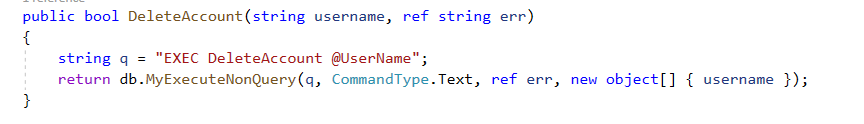
##### b. Phương thức Thêm tài khoản



Phương thức này giúp cho người dùng có thể thêm bất kỳ tài khoản nào mà chưa tồn tại trong CSDL. Các tham số: username, name, type.

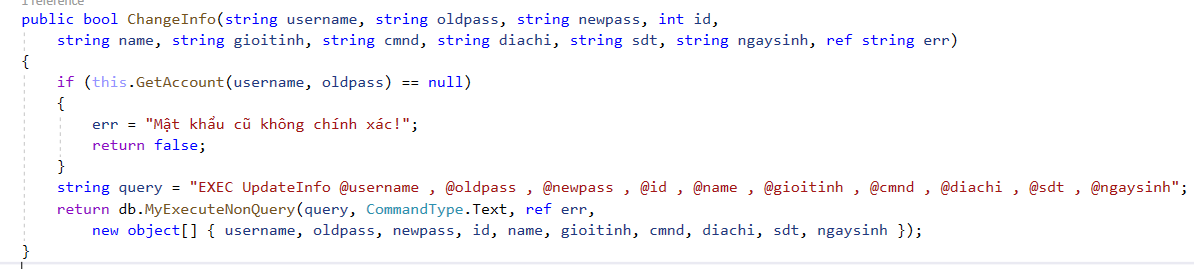
Cú pháp: INSERT INTO ACCOUNT VALUES…

##### c. Phương thức Xóa tài khoản



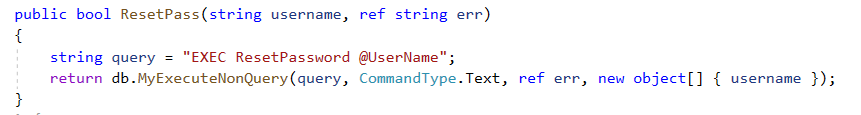
Xóa tài khoản theo cú pháp câu truy vấn: DELETE FROM ACCOUNT WHERE… Ở đây, người dùng có thể xóa tài khoản theo username.

##### Phương thức Thay đổi thông tin tài khoản



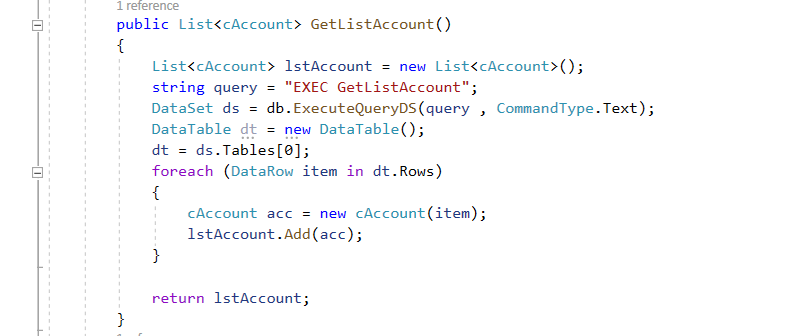
Thay đổi thông tin tài khoản với cú pháp: UPDATE…SET NAME… Với các tham số: tên user cũ, mật khẩu cũ, mật khẩu mới, tên user mới. Nếu người dùng nhập sai mật khẩu hiện dùng thì hệ thống sẽ không cho phép đổi thông tin và buộc phải nhập lại mật khẩu.

##### d. Phương thức Reset mật khẩu tài khoản



Sử dụng câu truy vấn: UPDATE ACCOUNT SET PASSWORD… Với tham số truyền vào: username.

##### e. Phương thức Lấy một danh sách tài khoản

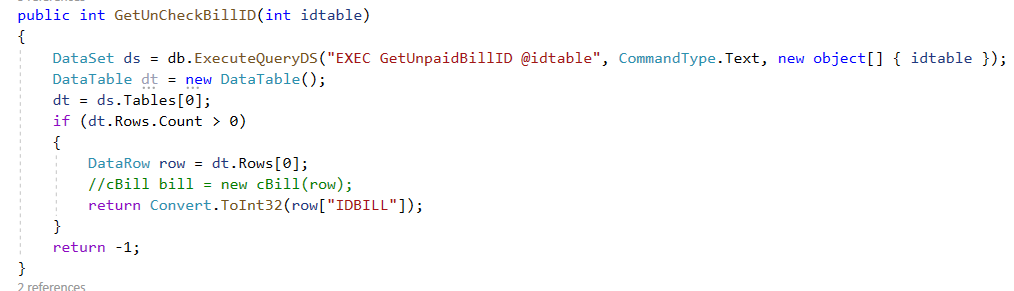


Phương thức sẽ trả về kết quả là một list tên tài khoản có trong CSDL.

* **Class BLBill**

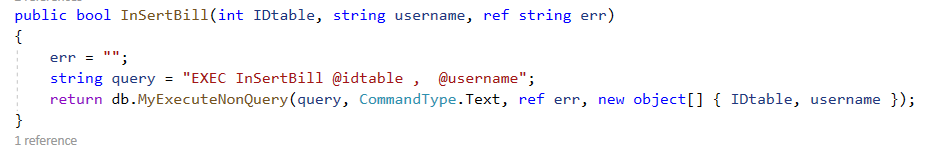
Class BLBill bao gồm các phương thức liên quan đến việc quản lý hóa đơn bán hàng trong nhà hàng.

##### a. Phương thức Lấy ID hóa đơn



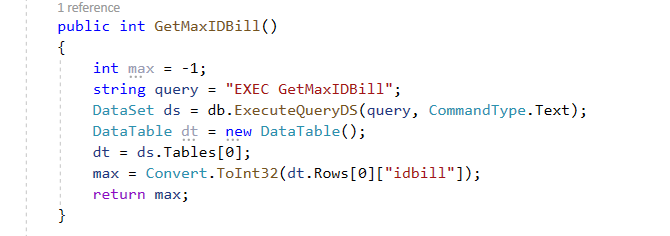
Cú pháp: SELECT \* FROM BILL WHERE… Tham số truyền vào là số bàn ăn (idtable) cần xuất hóa đơn.

##### b. Phương thức Thêm hóa đơn



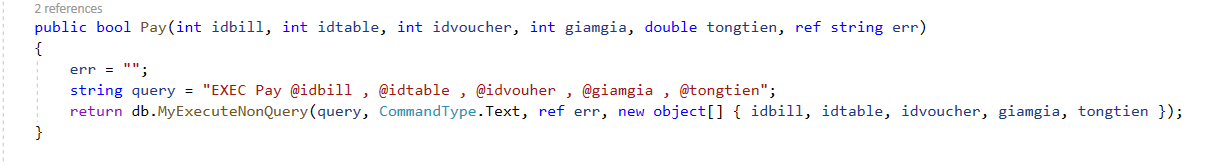
Để thêm hóa đơn, ta sử dụng cú pháp: INSERT INTO BILL VALUES… Tham số là idtable.

##### c. Phương thức Lấy số hóa đơn mới tạo

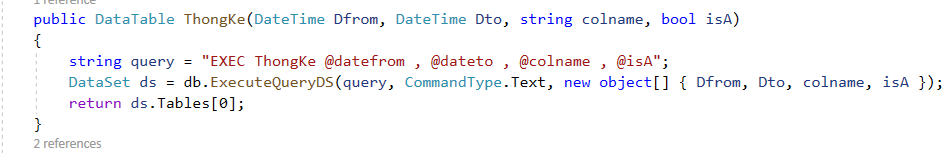


Phương thức này trả về kết quả là số hóa đơn lớn nhất, tức là số hóa đơn cuối ngày. Cho ta biết trong một ngày, nhà hàng đã xuất ra bao nhiêu hóa đơn.

##### d. Phương thức Thanh toán.

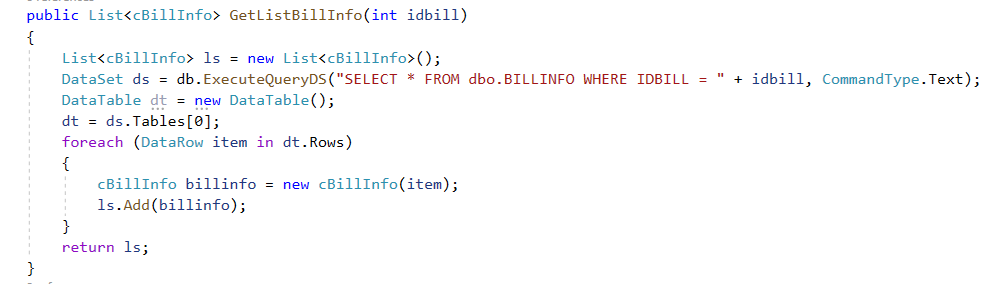


##### k. Phương thức Thống kê hóa đơn



* **Class BLBillInfo**

##### a. Phương thức Lấy danh sách thông tin hóa đơn

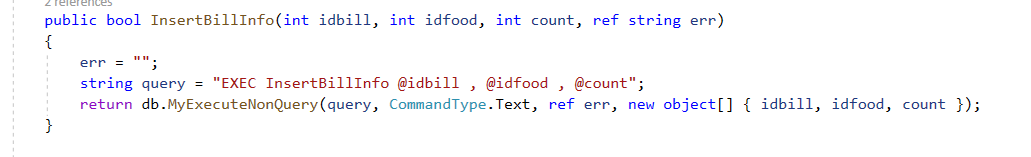


- Ban đầu, khởi tạo một danh sách theo class cBillInfo.

- Lọc lấy thông tin hóa đơn bằng cú pháp: SELECT \* FROM dbo.BILLINFO WHERE IDBILL = …

- Thông tin của hóa đơn sẽ lần lượt được thêm vào list.

##### b. Phương thức Thêm thông tin hóa đơn



Để thêm thông tin hóa đơn, ta sử dụng câu truy vấn: INSERT INTO BILLINFO VALUES… Với tham số: số hóa đơn, mã món ăn, số lượng.

##### c. Phương thức Lấy số lượng hóa đơn

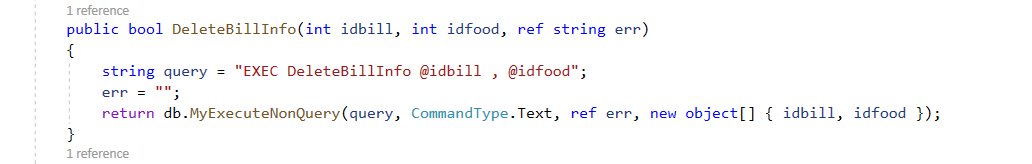


- Khởi tạo đối tượng count = 0.

- Sử dụng cú pháp: SELECT COUNT FROM BILLINFO WHERE IDFOOD = … AND IDBILL = … Tham số lần lượt là idfood và idbill.

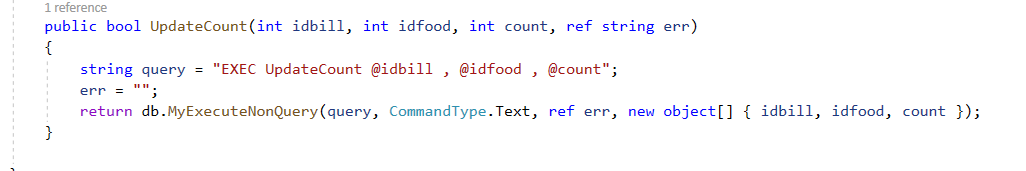
- Sau đó, gán cho đối tượng count và kết quả được trả về.

##### d. Phương thức Xóa thông tin của hóa đơn



Xóa thông tin hóa đơn với các tham số: idbill và idfood. Cú pháp như sau: DELETE BILLINFO WHERE IDBILL = … AND IDFOOD = …

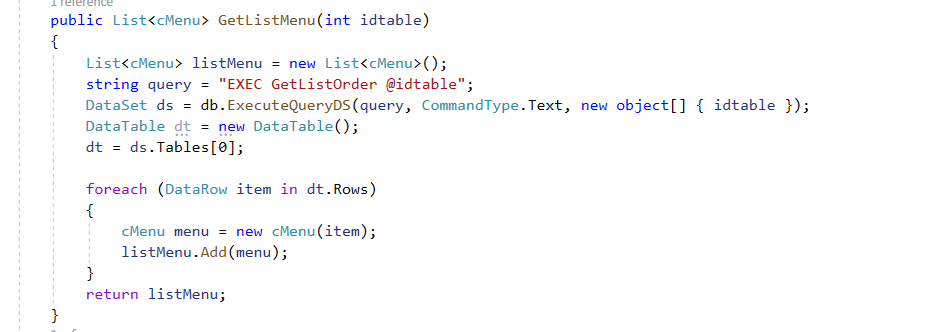
##### e. Phương thức Cập nhật thông tin hóa đơn



Cập nhật thông tin: idbill, idfood và số lượng món ăn.

* **Class BLFood**

##### a. Phương thức Lấy danh sách Menu

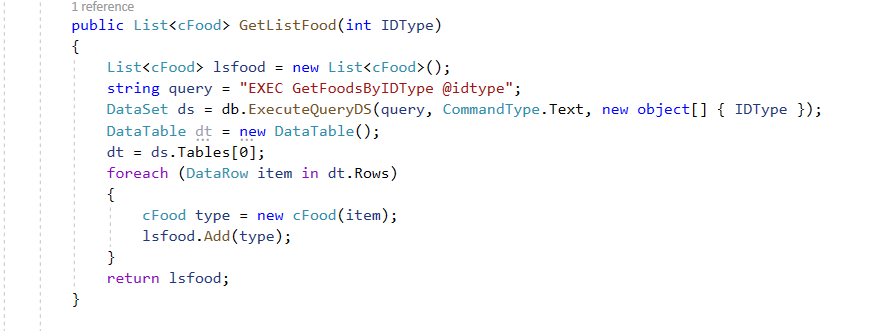


- Khởi tạo một list Menu thức ăn (cMenu).

- Sử dụng câu truy vấn: SELECT để chọn ra các menu theo các tham số như: tên món ăn, số lượng, đơn giá, tổng tiền từ quan hệ BILLINFO ở CSDL.

- Sau đó thêm từng món ăn vào trong list Menu.

##### b. Phương thức Lấy danh sách món ăn

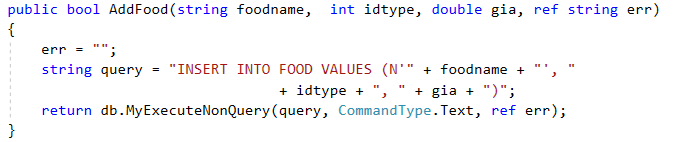


- Khởi tạo một danh sách món ăn trống (cFood).

- Sử dụng câu truy vấn: SELECT để chọn ra các món ăn với tham số là IDType (mã danh mục món ăn).

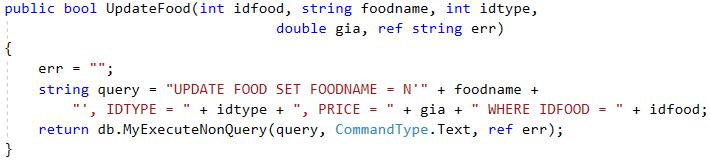
- Sau đó thêm từng danh mục món vào list.

##### c. Phương thức Thêm món ăn



Sử dụng INSERT để thêm món ăn. Ta có thể thêm món ăn bằng cách nhập đầy đủ các thông tin: tên món ăn, mã danh mục món ăn, đơn giá.

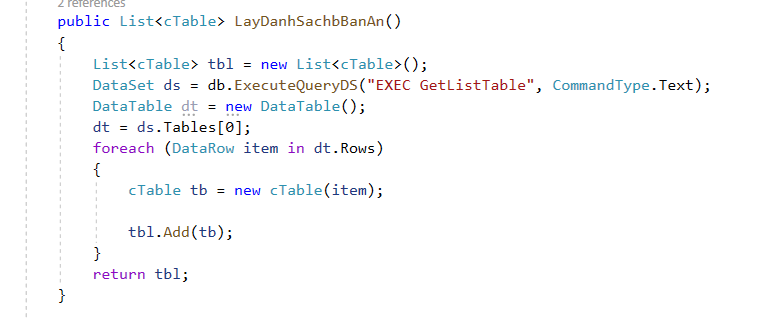
##### d. Phương thức Cập nhật món ăn



Phương thức này cho phép cập nhật lại món ăn với cú pháp UPDATE…SET với các thông tin như foodname, idtype, gia, idfood.

* **Class BLTableFood**

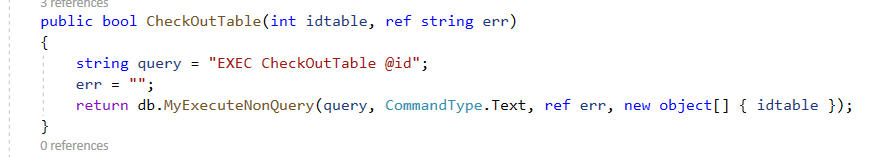
##### a. Phương thức LayDanhSachBanAn()



- Khởi tạo một list danh sách bàn ăn (cTable)

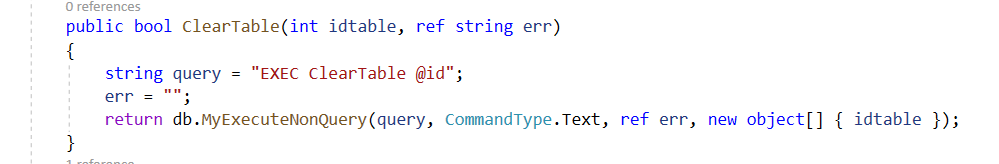
- Sử dụng câu truy vấn SELECT để chọn ra bàn ăn, sau đó thêm từng bàn vào list bàn ăn.

##### b. Phương thức Settable



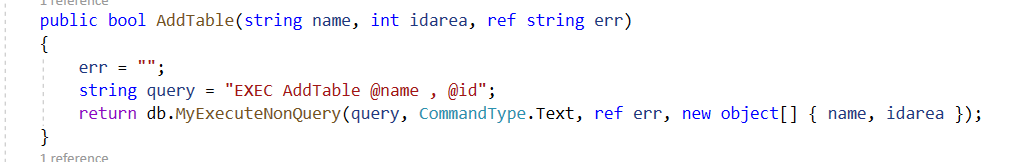
Phương thức SetTable để đặt bàn theo cú pháp UPDATE…SET STATUS đưa dữ liệu bàn có người vào.

##### c. Phương thức ClearTable



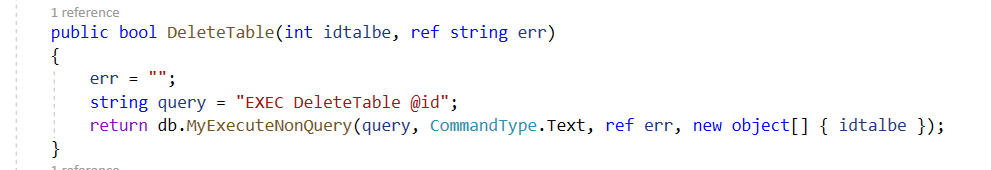
Với phương thức này sử dụng cú pháp UPDATE…SET STATUS thể hiện các bàn còn trống chưa có người.

##### d. Phương thức AddTable



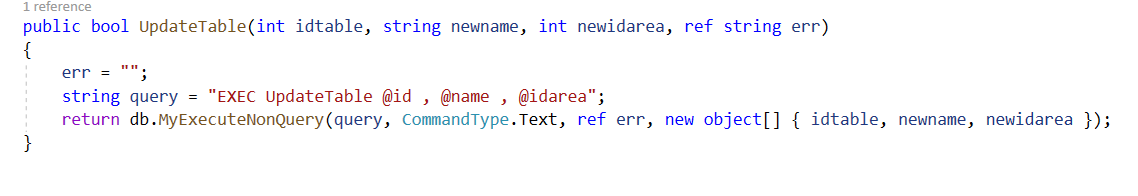
Cho phép thêm bàn trống với cú pháp INSERT INTO…VALUES.

##### e. Phương thức DeleteTable



Phương thức cho phép xóa bở các bàn khỏi list với tham số idtable theo cú pháp DELETE…WHERE.

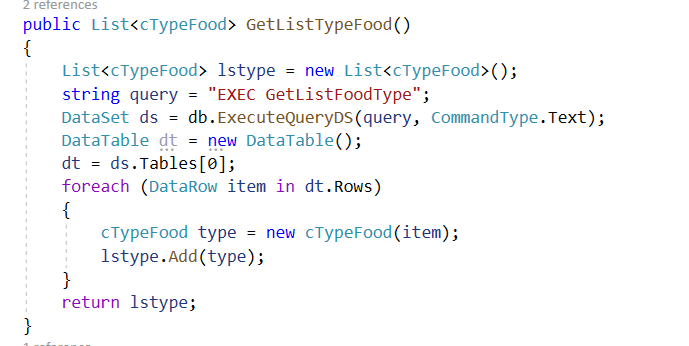
##### f. Phương thức UpdateTable



Giúp cập nhật lại các bàn vào list bàn ăn bằng cách truyền vào newname và idtable. Thực hiện theo cú pháp UPDATE TABLEFOOD SET NAME…WHERE…

* **Class BLTypeFood**

##### a. Phương thức GetListTypeFood()

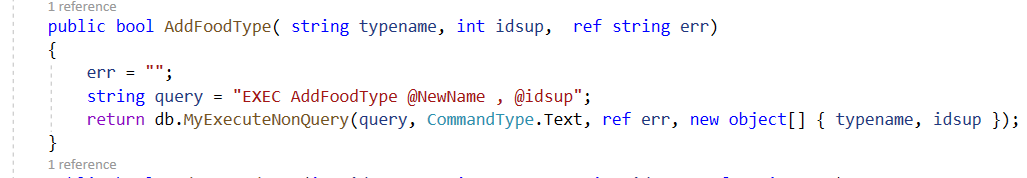


- Khởi tạo một list danh mục món ăn (cTypeFood).

- Sử dụng SELECT để chọn các loại món vào danh mục món từ FOODTYPE.

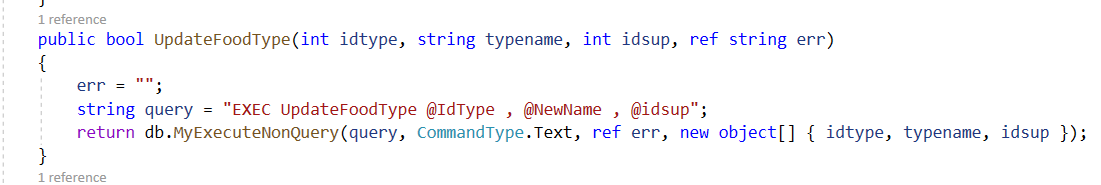
- Sau đó thêm từng loại món vào danh mục món.

##### b. Phương thức AddFoodType



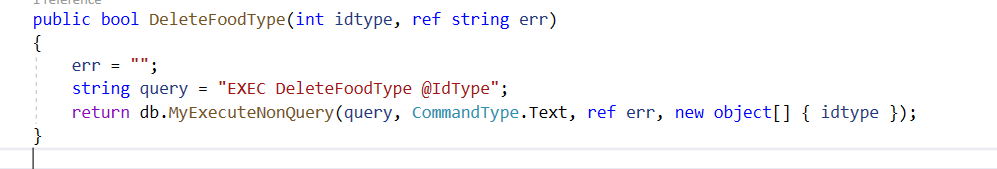
Cho phép thêm loại món vào list danh mục món theo cú pháp INSERT INTO…VALUES gồm typename.

##### c. Phương thức UpdateFoodType



Giúp cập nhật lại các loại món ăn vào list danh mục món ăn bằng cách truyền vào typename và idtype. Thực hiện theo cú pháp UPDATE…SET TYPENAME…WHERE…

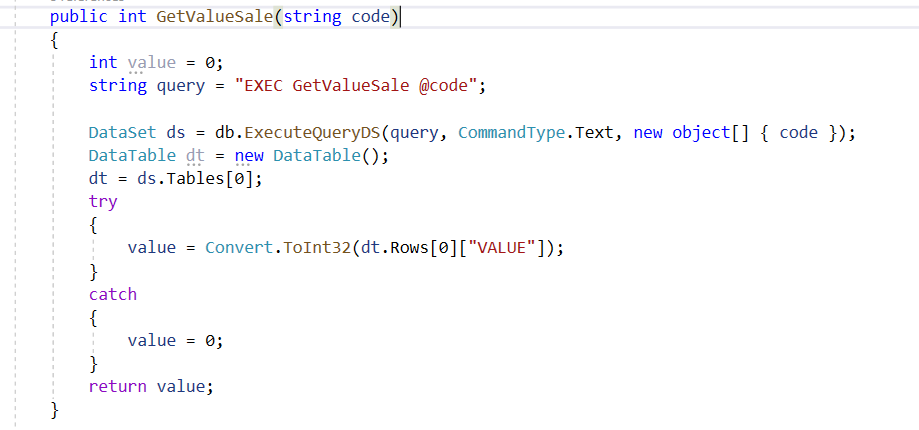
##### d. Phương thức DeleteFoodType



Phương thức cho phép xóa các loại món không sử dụng ra khỏi danh mục món với idtype bằng phương thức DELETE FROM…WHERE…

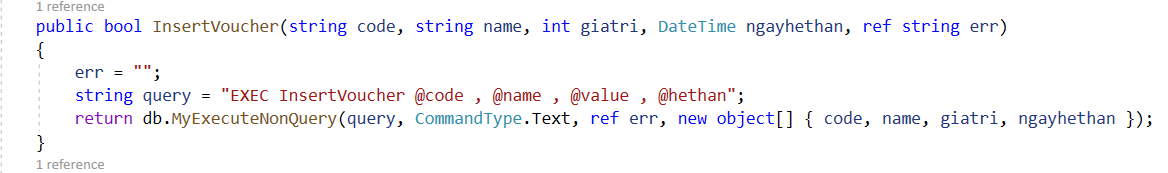
* **Class BLVoucher**

##### a. Phương thức GetValueSale



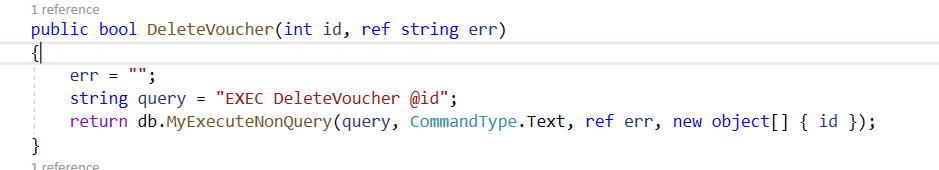
- Lấy cơ sỡ dữ liệu từ VOUCHER. Cho phép nhập mã code voucher từ list voucher.

##### b. Phương thức InsertVoucher



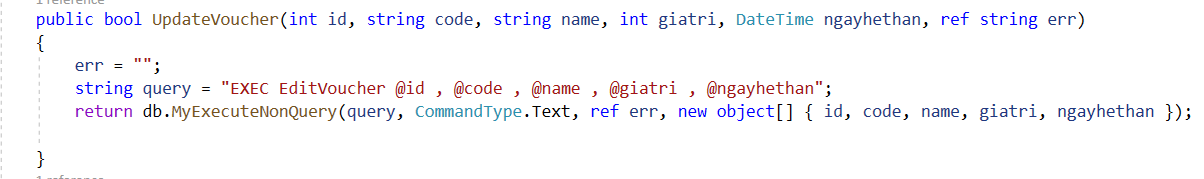
Cho phép nhận giá trị của voucher từ VOUCHER với code và giatri.

##### c. Phương thức DeleteVoucher



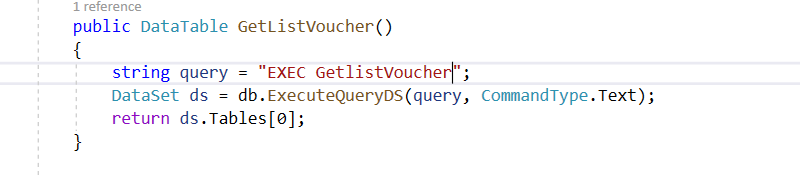
Giúp xóa voucher bằng cú pháp DELETE…WHERE… với đối tượng code.

##### d. Phương thức UpdateVoucher



Phương thức này cho phép cập nhật giá trị voucher với các đối tượng voucher, code bằng cú pháp UPDATE…SET…WHERE…

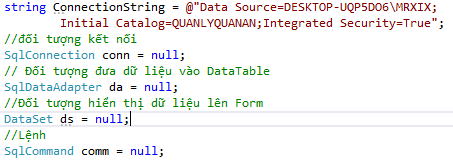
##### e. Phương thức GetListVoucher



- Khởi tạo một list Voucher.

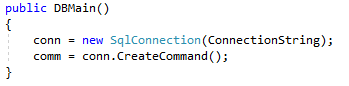
- Sử dụng câu truy vấn SELECT để chọn các voucher từ VOUCHER ở cơ sỡ dữ liệu.

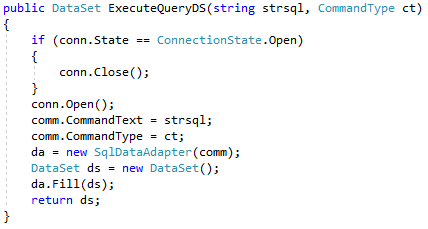
#### 2.3.9.3. DataLayers – DBMain.cs

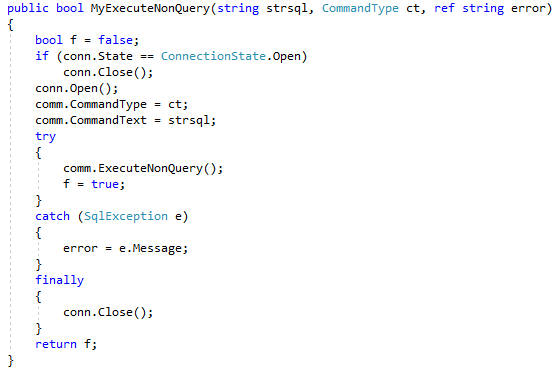


Kết nối cơ sở dữ liệu và truy vấn dữ liệu từ các nguồn dữ liệu DataSouse, khởi tao đối tượng kết nối, đưa đối tượng kết nối vào DataTable, đối tượng sẽ hiển thị dữ liệu trên Form DataSet ds = null.

Lệnh gán đối tượng comm= null.





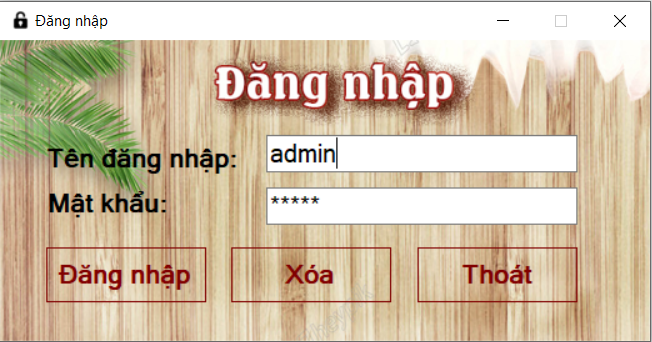


### 2.3.10. Hiện thực hóa giao diện

#### 2.3.10.1. Giao diện chính



#### 2.3.10.2. Giao diện đăng nhập hệ thống

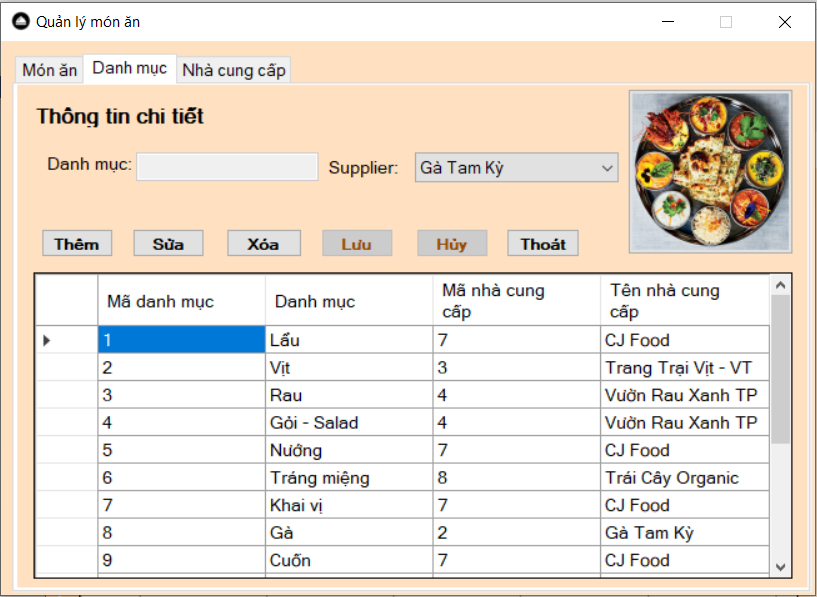


#### 2.3.10.3. Giao diện cập nhật dữ liệu

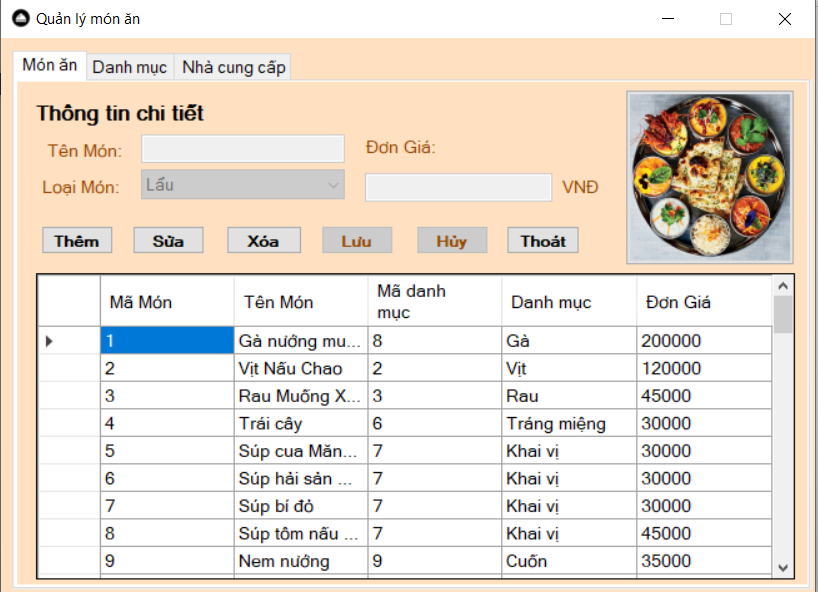
##### a. Cập nhật nhân viên



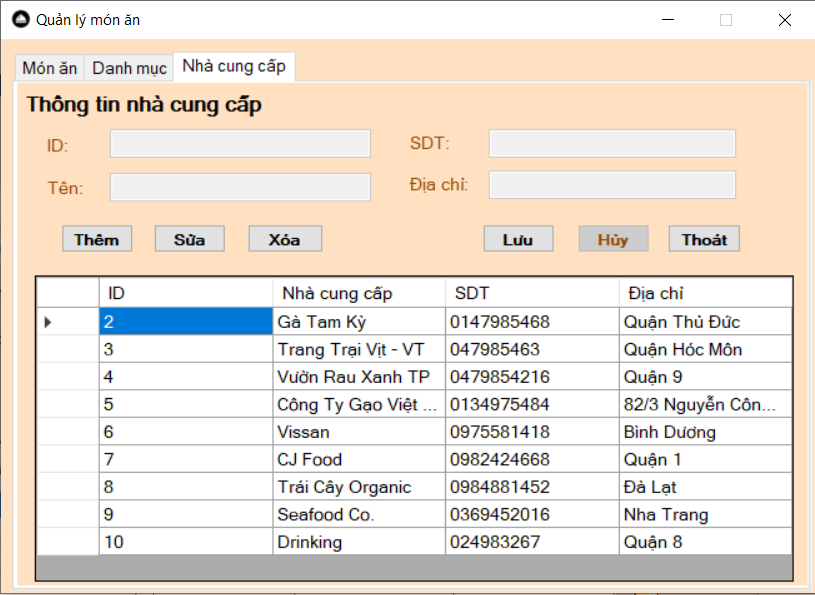
##### b. Cập nhật danh mục món ăn



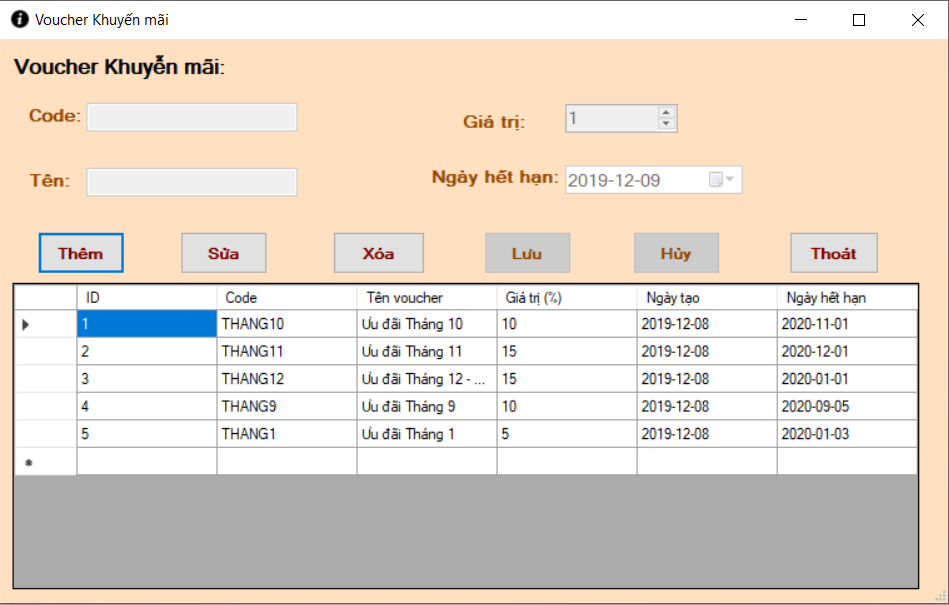
##### c. Cập nhật món ăn



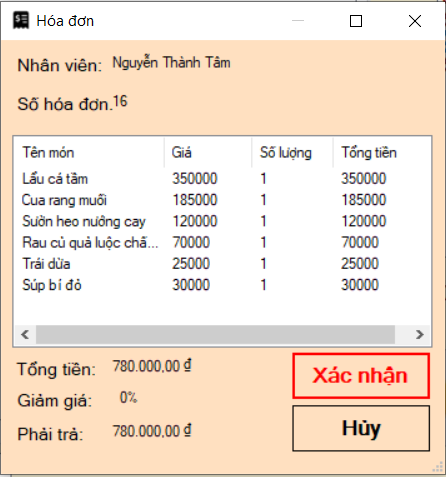
##### d. Cập nhật nhà cung cấp thực phẩm



##### e. Cập nhật voucher

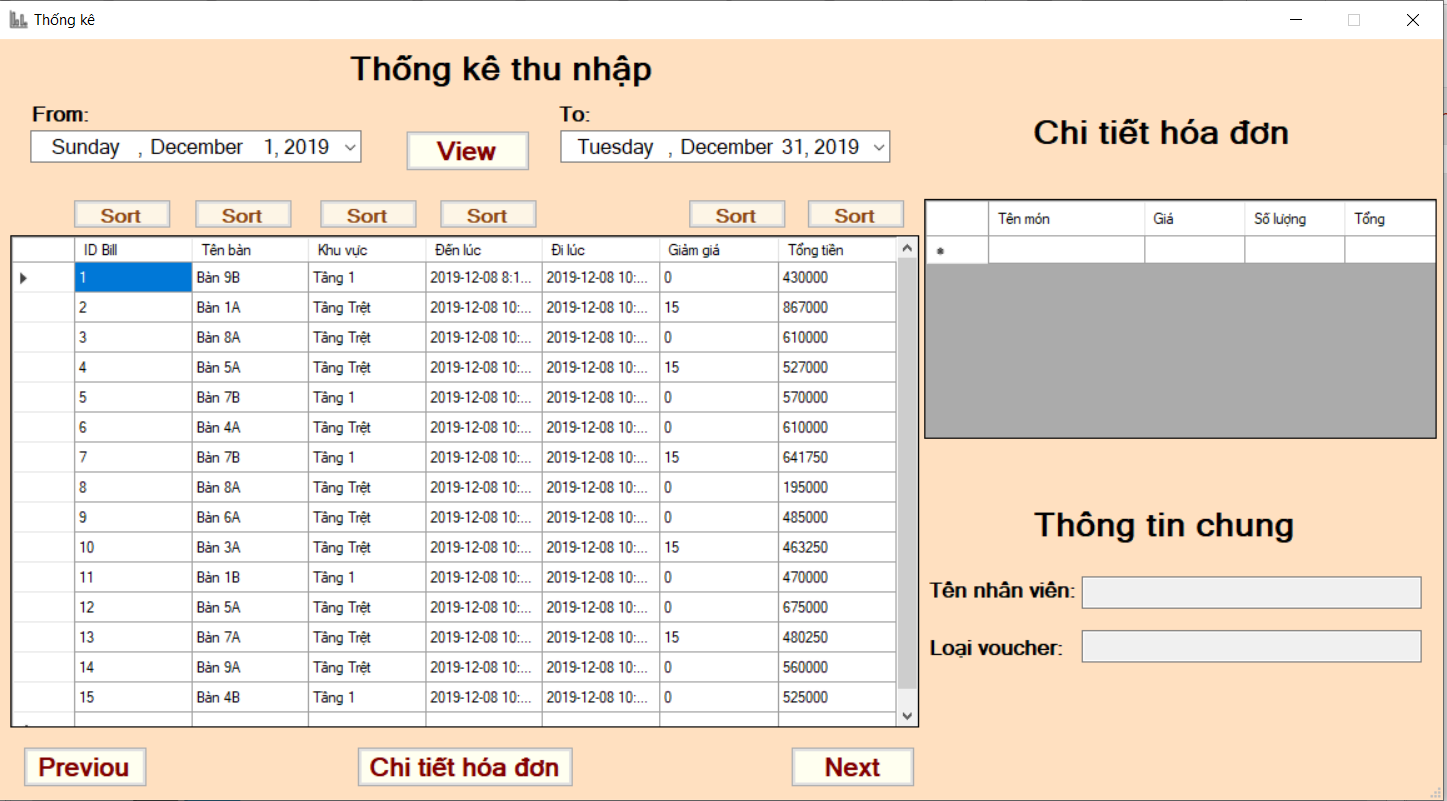


#### 2.3.10.4. Hóa đơn



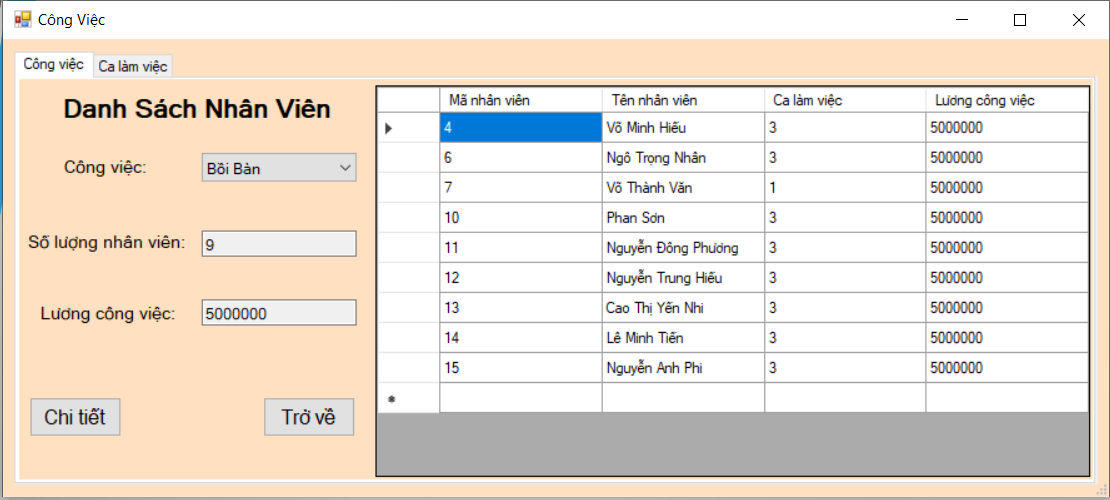
#### 2.3.10.5. Giao diện thống kê

##### a. Thống kê thu nhập

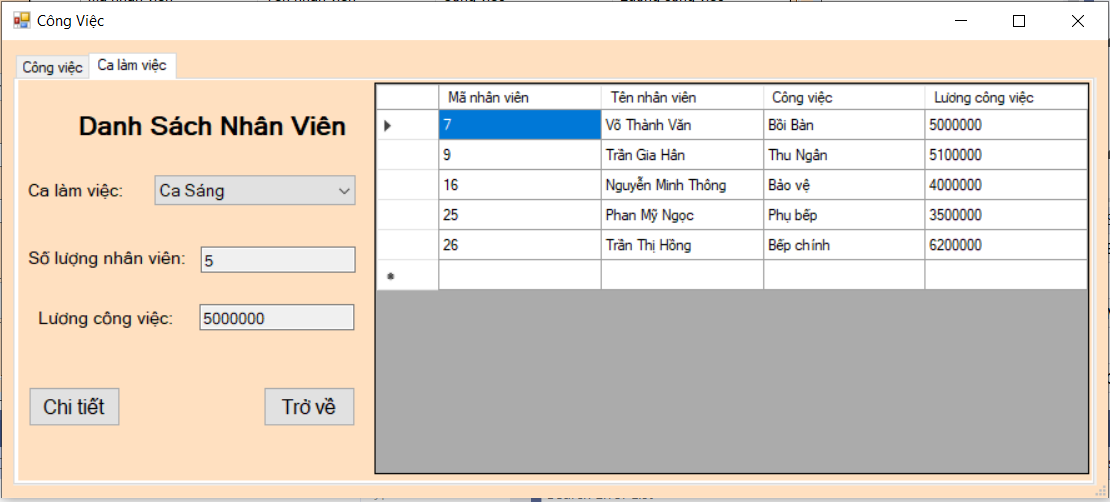


##### b. Thống kê công việc

**Thống kê theo công việc:**

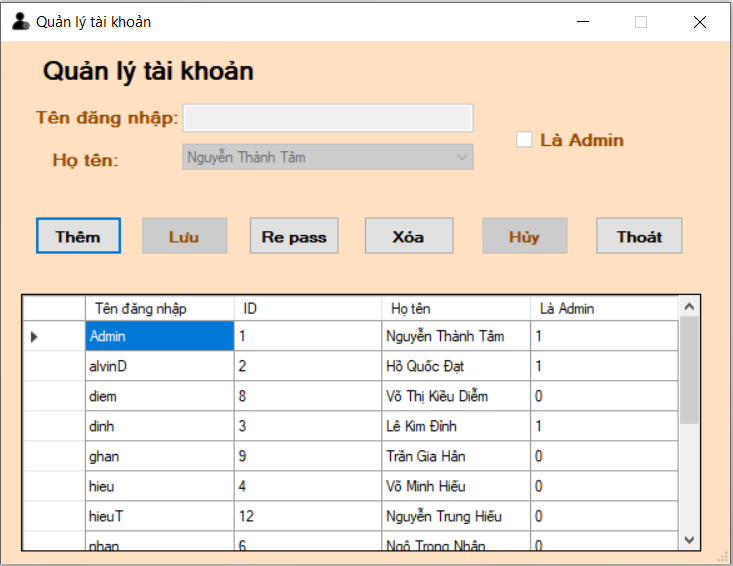
****

**Thống kê theo ca làm việc:**

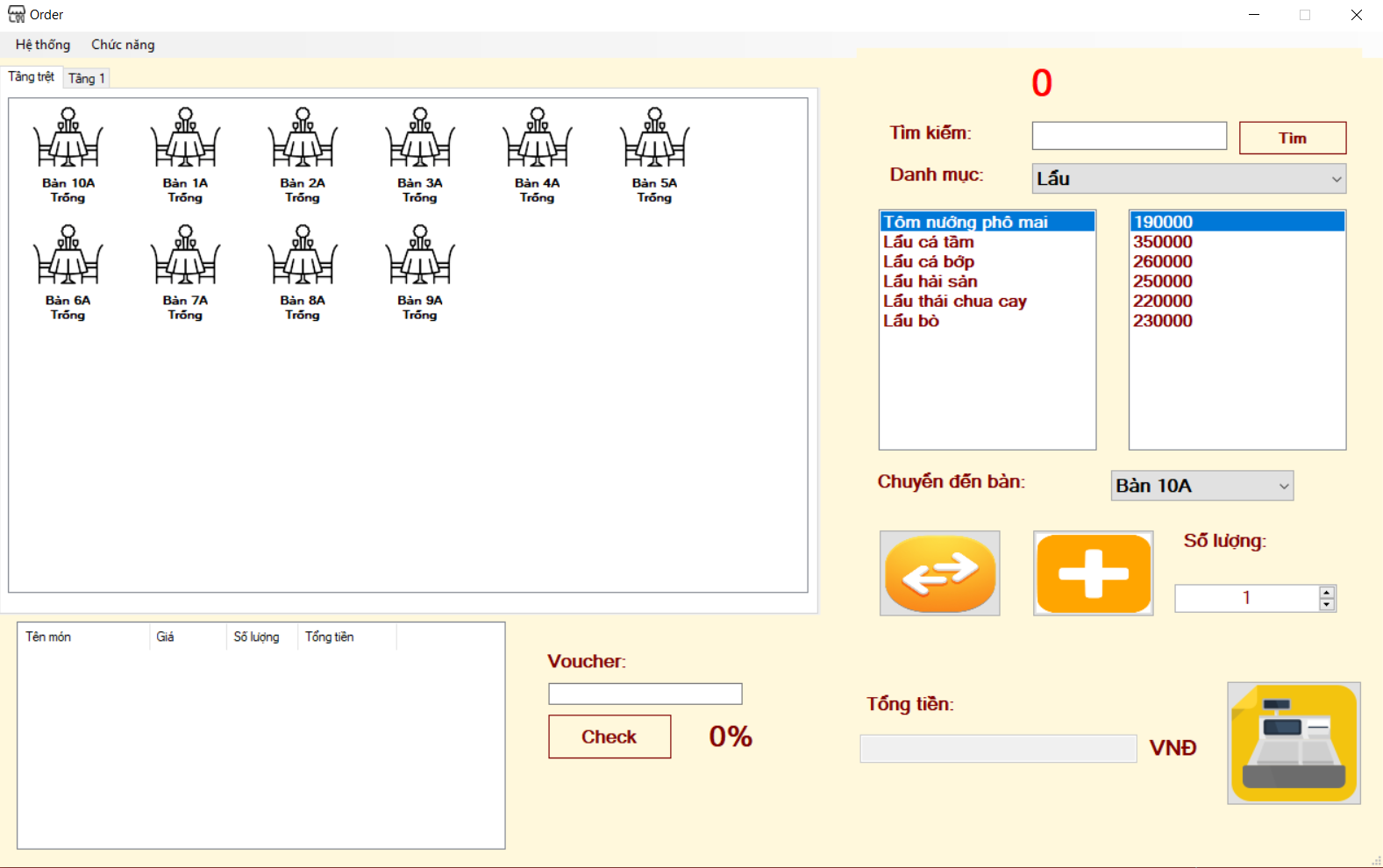
****

#### 2.3.10.6. Giao diện quản lý

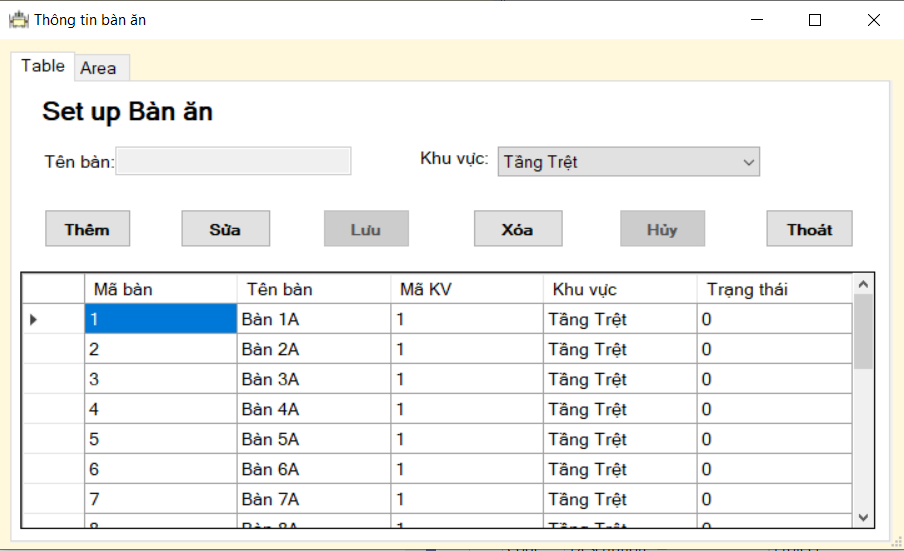
##### a. Quản lý Thông tin tài khoản



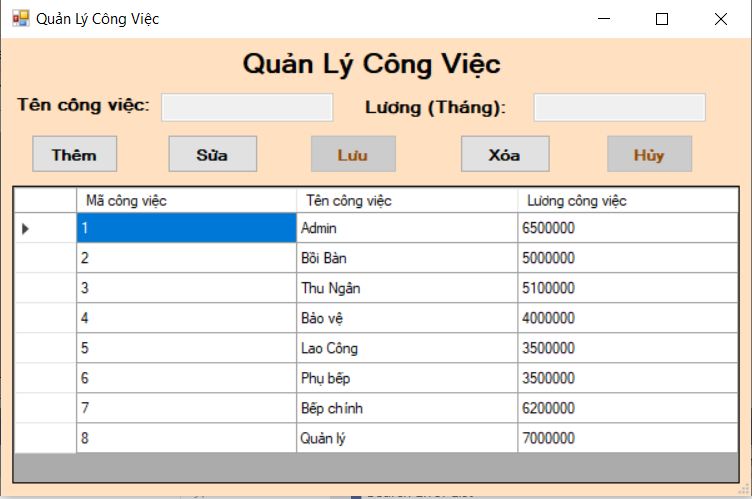
##### b. Quản lý Bàn ăn



##### c. Quản lý Thông tin bàn ăn



##### d. Quản lý công việc



# CHƯƠNG 3: KẾT LUẬN

## 3.1. Tổng kết

Việc ứng dụng công nghệ thông tin trong công tác quản lý nhà hàng là một việc làm cần thiết, nhằm nâng cao chất lượng của công tác quản lý, góp phần thúc đẩy nhà hàng phát triển toàn diện, mang lại lợi nhuận kinh tế cao. Dưới sự góp ý và hướng dẫn tận tình của thầy **Hoàng Long**, nhóm thực hiện đã hoàn thành đồ án khoảng 90% mục tiêu nhóm đã đề ra.

## 3.2. Ưu điểm

Với đồ án này, nhóm thực hiện đã hoàn thành những chức năng sau:

- Chương trình được xây dựng trên một hệ thống giao diện đơn giản, dễ sử dụng, thích hợp cho từng đối tượng người sử dụng.

- Tất cả các form nhập liệu đều được bẫy lỗi cho các trường.

- Lập trình để thực hiện các thao tác lưu trữ và quản lý về nhân viên, thực đơn, các báo cáo, hóa đơn…

- Mã hóa dữ liệu mật khẩu đăng ký của nhân viên.

- Lập trình trợ giúp tra cứu và lấy thông tin cần thiết một cách dễ dàng.

- Việc tính toán trong các hóa đơn nhập hàng và hóa đơn thanh toán giúp nhân viên thao tác nhanh hơn, chính xác và hiệu quả hơn.

## 3.3. Nhược điểm

- Dữ liệu dễ bị chỉnh sửa, tính bảo mật kém.

- Do thời gian chuẩn bị còn hạn chế nên chưa tối ưu hết các chức năng của chương trình.

- Giao diện chưa bắt mắt.

## 3.4. Hướng phát triển

- Bổ sung và hoàn thiện thêm giao diện người dùng.

- Thiết kế và lập trình thêm một số chức năng để hệ thống hoàn thiện hơn.

- Tăng cường chế độ bảo mật hệ thống:

+ Bảo mật Server.

+ Bảo mật cơ sở dữ liệu trên SQL Server.

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Nguyễn Thành Tâm** | **Trần Gia Hân** |
| 1 | Form Hóa đơnBLBillInfo BLBill  cBill, cBillInfo |  | X |
| 2 | Form Login  BLAccount  cAccount | X |  |
| 3 | Form Quản lý |  | X |
| 4 | Form Quản lý Account |  | X |
| 5 | Form Quản lý bàn ăn  BLTableFood  cTable | X |  |
| 6 | Form Quản lý món ăn  BLTypeFood  cTypefood | X |  |
| 7 | BLFood  cFood |  | X |
| 8 | Form Report | X |  |
| 9 | Form Voucher  BLVoucher |  | X |
| 10 | DBMain | X |  |
| 11 | Kiểm thử | X | X |
| 12 | Báo cáo file word | X | X |