# LỜI CẢM ƠN

Em xin chân thành cảm ơn sâu sắc đến với quý thầy cô đã giúp đỡ em thực hiện đồ án lần này. Đặc biệt là Thầy X đã giúp đỡ, hướng dẫn tận tình, chỉ bảo em trong suốt thời gian học tập và thực hiện đồ án.

Đồng thời em cũng xin trân trọng gửi lời cảm ơn, tình cảm quý báu mà các thầy cô trong khoa CNTT đã truyền đạt kĩ năng, kinh nghiệm cho em, và em xin gửi lời cảm ơn đến Trường ĐH X đã tạo điều kiện để em được học tập trong môi trường tuyệt vời như này.

Và cuối cùng em cũng xin được gửi lời cảm ơn đến với gia đình đã luôn luôn động viên, ủng hộ, những người bạn đã gắn bó, chia sẻ rất nhiều những kinh nghiệm, kiến thức để đồ án lần này được thực hiện một cách hoàn chỉnh .

Tuy nhiên thời gian và lượng kiến thức có hạn nên em không thể phát huy hết khả năng, ý tưởng vào đồ án lần này. Trong quá trình thực hiện, không thể tránh những sai sót, kính mong nhận được sự đóng góp và cảm thông từ quý thầy cô.

Em xin chân thành cảm ơn!

**Sinh viên thực hiện**

***X***

# LỜI MỞ ĐẦU

Ngày nay việc sử dụng máy tính điện tử trong các ngành kinh tế, quản lý sản  
xuất kinh doanh gần như đã trở thành một yêu cầu bắt buộc. Máy tính điện tử với  
việc sử dụng nó một cách hiệu quả đã và đang trở thành một lực lượng sản xuất  
hùng mạnh, tiên tiến nhất của thời đại. Bên cạnh đó áp dụng công nghệ thông tin vào các ngành sản xuất, các lĩnh vực buôn bán hàng hóa, quản lý bán hàng đang là một nhu cầu cấp bách, rất cần thiết, đặc biệt là các ứng dụng tin học trong lĩnh vực quản lý.

Vai trò của thông tin trong đời sống, trong khoa học kỹ thuật, quản lý kinh doanh bán hàng là rất quan trọng. Thông tin đầy đủ, chính xác, kịp thời giúp cho các nhà quản lý đưa ra được những quyết định đúng đắn mang tính khả thi cao. Để làm được điều đó thì cần phải có sự trợ giúp của công nghệ thông tin và sử dụng hệ thống máy tính để lưu trữ, xử lý thông tin một cách khoa học và nhanh nhạy với khối lượng thông tin phức tạp, đồ sộ để đưa ra những thông tin cần thiết, chính xác theo yêu cầu của quản lý.

Với mục tiêu tìm hiểu cấu trúc, cách thức hoạt động và khái quát hóa về phương pháp quản lý bán hàng, em đã xây dựng và đưa vào thử nghiệm thực tế một chương trình quản lý bán thức ăn nhanh. Đây sẽ là bước đi đầu tiên của em trước khi tiến cận với công việc quản lý bán hàng của một cửa hàng thực sự.

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**………………………………………………………………………….…………………………………….……………………………………………………………………………………………………………….………………………………………………………………………….…………………………………………………………………………………………………….………………………………………………………………………….…………………………………….……………………………………………………………………………………………………………….………………………………………………………………………….…………………………………….……………………….……...………………………………………………………………………….………….……………….……………………………………………………………………………………………………………….……………………….……………………………………………………………………………………………………………….………………………………………………………………………………………………….…………….……………………….…………………………………...………………………………………………………………………….….……………………………………………………………………………………………………………….……………………….………...………………………………………………………….….…………………………………………………………………………………………….………………….……………………….………...………………………………………………………….….……………………………………………………………………………………………………………….……………………….………...…………….………………..**

*Tp.HCM, Ngày . . . . tháng . . . . năm . . . .*

**Giảng viên giảng dạy**

(Ký tên và ghi rõ họ tên)

**X**

**MỤC LỤC**

Contents

[LỜI CẢM ƠN 7](#_Toc82157205)

[LỜI MỞ ĐẦU 8](#_Toc82157206)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 9](#_Toc82157207)

[CHƯƠNG I : CƠ SỞ LÝ THUYẾT 11](#_Toc82157208)

[**1/. Giới thiệu.** 11](#_Toc82157209)

[1.1/. Lý do chọn đề tài 11](#_Toc82157210)

[1.2/. Mục tiêu nghiên cứu 11](#_Toc82157211)

[1.3/. Phương pháp nghiên cứu 11](#_Toc82157212)

[**2/. Tìm hiểu lý thuyết.** 12](#_Toc82157213)

[2.1/. Phân tích thiết kế hệ thống thông tin 12](#_Toc82157214)

[2.2/. Lập trình ứng dụng C# Winform 16](#_Toc82157215)

[CHƯƠNG II : PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN CHƯƠNG TRÌNH QUẢN LÝ BÁN THỨC ĂN NHANH 19](#_Toc82157216)

[**1/. Phân tích chức năng:** 19](#_Toc82157217)

[**1.1/. Sơ đồ Usecase:** 19](#_Toc82157218)

[**2/. Sơ đồ class** 22](#_Toc82157219)

[**3/. Cơ sở dữ liệu** 23](#_Toc82157220)

[3.1/. Lược đồ cơ sở dữ liệu quan hệ: 23](#_Toc82157221)

[3.2/. Câu lệnh truy vấn SQL 23](#_Toc82157222)

[CHƯƠNG III : THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ BÁN THỨC ĂN NHANH 26](#_Toc82157223)

[**1/. Chi tiết các chức năng và phương thức:** 26](#_Toc82157224)

[**2/. Chi tiết giao diện các chức năng:** 34](#_Toc82157225)

[CHƯƠNG IV: THỰC NGHIỆM VÀ TRIỂN KHAI 39](#_Toc82157226)

[**1/. Đóng gói và xuất bản chương trình:** 39](#_Toc82157227)

[**2/. Môi trường cài đặt và triển khai, cách vận hành:** 41](#_Toc82157228)

[**3/. Demo các chức năng chính của chương trình:** 42](#_Toc82157229)

[CHƯƠNG V: KẾT LUẬN 47](#_Toc82157230)

[**1/. Kết luận.** 47](#_Toc82157231)

[1.1/. Ưu điểm của đồ án 47](#_Toc82157232)

[1.2/. Hạn chế của đồ án 47](#_Toc82157233)

[**2/. Hướng phát triển** 47](#_Toc82157234)

**DANH MỤC CÁC BẢNG HÌNH**

[Hình 1 Ngôn ngữ C# 13](#_Toc82007430)

[Hình 2 Usecase Tổng quát 21](#_Toc82007431)

[Hình 3 Usecase Quản lý khách hàng 21](#_Toc82007432)

[Hình 4 Usecase Quản lý nhân viên 22](#_Toc82007433)

[Hình 5 Usecase Quản lý món ăn 22](#_Toc82007434)

[Hình 6 Usecase Quản lý hóa đơn 23](#_Toc82007435)

[Hình 7 Sơ đồ class hệ thống quản lý quán pizza 24](#_Toc82007436)

[Hình 8 Lược đồ CSDL quan hệ 25](#_Toc82007437)

[Hình 9 Giao diện form menu 36](#_Toc82007438)

[Hình 10 Giao diện form quản lý loại món 37](#_Toc82007439)

[Hình 11 Giao diện form quản lý món ăn 38](#_Toc82007440)

[Hình 12 Giao diện form quản lý khách hàng 38](#_Toc82007441)

[Hình 13 Giao diện form quản lý nhân viên 39](#_Toc82007442)

[Hình 14 Giao diện form hóa đơn 39](#_Toc82007443)

[Hình 15 Giao diện form tìm kiếm hóa đơn 40](#_Toc82007444)

[Hình 16 Project đóng gói 41](#_Toc82007445)

[Hình 17 Giao diện project đóng gói 41](#_Toc82007446)

[Hình 18 Tạo project output 41](#_Toc82007447)

[Hình 19 Tạo primary output 41](#_Toc82007448)

[Hình 20 Tạo shortcut mới 42](#_Toc82007449)

[Hình 21 Add Fastfood 42](#_Toc82007450)

[Hình 22 Đổi tên tác giả và nhà sản xuất 42](#_Toc82007451)

[Hình 23 Tìm đường dẫn folder Setup 43](#_Toc82007452)

[Hình 24 Cài đặt Fastfood 43](#_Toc82007453)

[Hình 25 App fastfood 43](#_Toc82007454)

[Hình 26 Nhập thông tin để thêm sản phẩm 44](#_Toc82007455)

[Hình 27 Thêm sản phẩm thành công 44](#_Toc82007456)

[Hình 28 Chọn dòng sản phẩm cần sửa và sửa 45](#_Toc82007457)

[Hình 29 Kết quả sau khi sửa sản phẩm 45](#_Toc82007458)

[Hình 30 Dữ liệu đã được lưu vào database 46](#_Toc82007459)

[Hình 31 Xóa sản phẩm thành công 46](#_Toc82007460)

[Hình 32 Thêm hóa đơn 47](#_Toc82007461)

[Hình 33 Hóa đơn được in vào Excel 47](#_Toc82007462)

[Hình 34 Tìm kiếm hóa đơn theo mã nhân viên là 1 48](#_Toc82007463)

[Hình 35 Tìm thấy 2 hóa đơn 48](#_Toc82007464)

# CHƯƠNG I : CƠ SỞ LÝ THUYẾT

## **1/. Giới thiệu.**

### 1.1/. Lý do chọn đề tài

Đời sống ngày càng nâng cao nên nhu cầu trao đổi, buôn bán hàng hóa ngày càng tăng. Việc quản lý hàng hóa, ghi chép hóa đơn một cách thủ công không mấy khả quan, rất dễ xảy ra sai sót trong khâu quản lý, thống kê, chính vì thế việc áp dụng Công nghệ thông tin vào khâu quản lý bán hàng là một phần không thể thiếu trong nhu cầu đời sống. Thế nên em đã chọn đề tài quản lý bán hàng để phục vụ cho nhu cầu lưu trữ, thao tác thông tin với các mặt hàng một cách chính xác, nhanh chóng và tiện lợi.

### 1.2/. Mục tiêu nghiên cứu

- Lập trình tạo ra chương trình quản lý bán hàng kết nối với CSDL và thực hiện chức năng tương tác tham chiếu vào CSDL giúp quản lý đơn hàng, sản phẩm dễ dàng hơn và tiện ích hơn.

- Công việc cụ thể:

* Tạo Button thêm, xóa, sửa, lưu … để tương tác với dữ liệu trong CSDL.
* In dữ liệu ra file Excel.
* Tạo DataGridView để lấy dữ liệu trong CSDL và hiển thị thông tin cho người sử dụng.
* Sử dụng những câu lệnh phức tạp để xét điều kiện cho phần mềm.
* Đóng gói dữ liệu nhập vào để bảo mật thông tin.
* Xây dựng những công cụ hỗ trợ như connection, dataset, … để sử dụng chung rút ngắn thời gian và tối ưu phần mềm.
* Xây dựng giao diện đẹp, thân thiện tạo cảm giác thoải mái cho người sử dụng.

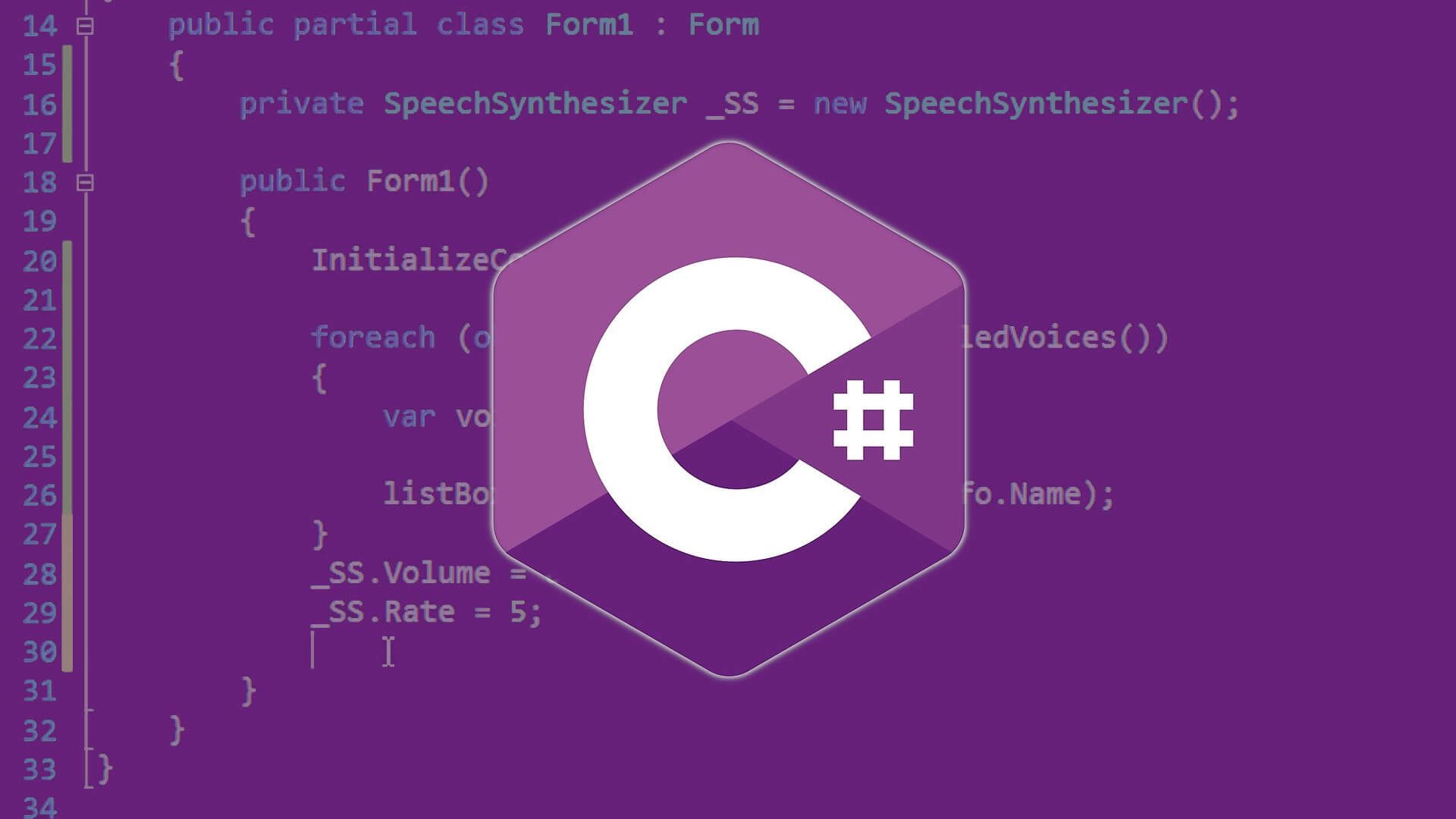
### 1.3/. Phương pháp nghiên cứu

* Tìm hiểu nghiên cứu tài liệu cụ thể để vận dụng vào trong phần mềm.
* Học các cách thức sử dụng công cụ để vận dụng vào đồ án.
* Sử dụng các giao diện có sẵn hay nuget để làm đẹp cho giao diện
* Tham khảo những mẫu phần mềm khác để biến tấu vận dụng vào phần mềm quản lý bán hàng của mình.
* Tìm ra những giải pháp để rút ngắn code dùng chung nhiều form khác vào cùng một class.

## **2/. Tìm hiểu lý thuyết.**

### 2.1/. Phân tích thiết kế hệ thống thông tin

2.1.1/. Ngôn ngữ C# là gì ?



Hình Ngôn ngữ C#

Ngôn ngữ C# là một ngôn ngữ được dẫn xuất từ C và C++, nhưng nó được tạo từ nền tảng phát triển hơn. Microsoft bắt đầu với công việc trong C và C++ và thêm vào những đặc tính mới để làm cho ngôn ngữ này dễ sử dụng hơn. Nhiều trong số những đặc tính này khá giống với những đặc tính có trong ngôn ngữ Java. Không dừng lại ở đó, Microsoft đưa ra một số mục đích khi xây dựng ngôn ngữ này. Những mục đích này được được tóm tắt như sau:

* C# là ngôn ngữ đơn giản
* C# là ngôn ngữ hiện đại
* C# là ngôn ngữ hướng đối tượng
* C# là ngôn ngữ mạnh mẽ và mềm dẻo
* C# là ngôn ngữ có ít từ khóa
* C# là ngôn ngữ hướng module
* C# sẽ trở nên phổ biến

2.1.2/. Các kiểu dữ liệu

Tương tự như C++ hay Java, C# chia thành hai tập hợp kiểu dữ liệu chính: Kiểu xây dựng sẵn (built- in) mà ngôn ngữ cung cấp cho người lập trình và kiểu được người dùng định nghĩa (user-defined) do người lập trình tạo ra.

C# phân tập hợp kiểu dữ liệu này thành hai loại: Kiểu dữ liệu giá trị (value) và kiểu dữ liệu tham chiếu (reference). Việc phân chi này do sự khác nhau khi lưu kiểu dữ liệu giá trị và kiểu dữ liệu tham chiếu trong bộ nhớ. Đối với một kiểu dữ liệu giá trị thì sẽ được lưu giữ kích thước thật trong bộ nhớ đã cấp phát là stack. Trong khi đó kiểu dữ liệu tham chiếu như các đối tượng được cấp phát trên heap. Khi một đối tượng được cấp phát trên heap thì địa chỉ của nó được trả về, và địa chỉ này được gắn đến một tham chiếu.

Bảng 1.2. 1 Kiểu dữ liệu xây dựng sẵn

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu C#** | **Số byte** | **Kiểu .NET** | **Mô tả** |
| byte | 1 | Byte | Số nguyên dương không dấu từ 0-255 |
| char | 2 | Char | Kí tự Unicode |
| bool | 1 | Boolean | Giá trị logic true/ false |
| sbyte | 1 | Sbyte | Số nguyên có dấu ( từ -128 đến 127) |
| short | 2 | Int16 | Số nguyên có dấu giá trị từ -32768 đến 32767 |
| ushort | 2 | Int16 | Số nguyên không dấu 0 – 65.535 |
| int | 4 | Int32 | Số nguyên có dấu –2.147.483.647 và  2.147.483.647 |
| uint | 4 | Uint32 | Số nguyên không dấu 0 – 4.294.967.295 |
| float | 4 | Single | Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-  38 đến 3,4E+38, với 7 chữ số có nghĩa.. |

|  |  |  |  |
| --- | --- | --- | --- |
| float | 4 | Single | Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-  38 đến 3,4E+38, với 7 chữ số có nghĩa.. |
| double | 8 | Double | Kiểu dấu chấm động có độ chính xác gấp  đôi, giá trị xấp xỉ từ 1,7E-308 đến 1,7E+308,  với 15,16 chữ số có nghĩa |
| decimal | 8 | Decimal | Có độ chính xác đến 28 con số và giá trị thập  phân, được dùng trong tính toán tài chính,  kiểu này đòi hỏi phải có hậu tố “m” hay “M”  theo sau giá trị. |
| long | 8 | Int64 | Kiểu số nguyên có dấu có giá trị trong  khoảng :  -9.223.370.036.854.775.808 đến  9.223.372.036.854.775.807 |
| ulong | 8 | Uint64 | Số nguyên không dấu từ 0 đến  0xffffffffffffffff |

**Chọn một kiểu định sẵn**

* Tuỳ vào từng giá trị muốn lưu trữ mà ta chọn kiểu cho phù hợp. Nếu chọn kiểu quá lớn so với các giá trị cần lưu sẽ làm cho chương trình đòi hỏi nhiều bộ nhớ và chạy chậm. Trong khi nếu giá trị cần lưu lớn hơn kiểu thực lưu sẽ làm cho giá trị các biến bị sai và chương trình cho kết quả sai.
* Kiểu char biểu diễn một ký tự Unicode. Ví dụ “\u0041” là ký tự “A” trên bảng Unicode. Một số ký tự đặc biệt được biểu diễn bằng dấu “\” trước một ký tự khác.

**Kiểu dữ liệu do người dùng định nghĩa**

* Tất cả kiểu dữ liệu do người dùng định nghĩa ngoài trừ kiểu cấu trúc điều là kiểu dữ liệu tham chiếu
* Một sô kiểu dữ liệu do người dùng định nghĩa gồm
* **object**: đây là kiểu dữ liệu cơ sở chứa tất cả các kiểu dữ liệu khác trong C#.
* **string**: kiểu dữ liệu chuỗi ký tự.
* **class**: kiểu dữ liệu class.
* **delegate**: kiểu dữ liệu chuyển giao.
* **interface**: kiểu dữ liệu giáo tiếp.
* **array**: kiểu dữ liệu mảng.

2.1.3/. Cấu trúc rẽ nhánh

*a. Câu lệnh i**f… else …*

Cú pháp:

if ( biểu thức logic )

khối lệnh;

hoặc

if ( biểu thức logic )

khối lệnh 1;

else

khối lệnh 2;

*Ghi chú: Khối lệnh là một tập các câu lện trong cặp dấu “{…}”. Bất kỳ**nơi đâu có câu lệnh thì ở đó có thể viết bằng một khối lệnh.*

Biểu thức logic là biểu thức cho giá trị dúng hoặc sai (***true*** hoặc ***false***). Nếu “*biểu thức logic*” cho giá trị đúng thì “*khối lệnh*” hay “*khối lệnh 1*” sẽ được thực thi,ngược lạ**i** “*khối lệnh 2*” sẽ thực thi. Một điểm khác biệt với C++ là biểu thức trongcâu lệnh if phải là biểu thức logic, không thể là biểu thức số.

b. Câu lệnh switch case

Cú pháp:

switch ( biểu\_thức\_lựa\_chọn )

{

case biểu\_thức\_hằng :

khối lệnh;

Biểu thức lựa chọn là biểu thức sinh ra trị nguyên hay chuỗi. Switch sẽ so sánh biểu\_thức\_lựa\_chọn với các biểu\_thức\_hằng để biết phải thực hiện với khối lệnh nào. Lệnh nhảy như break, goto…để thoát khỏi câu switch và bắt buộc phải có.

int nQuyen = 0;

switch ( sQuyenTruyCap )

{

case “Administrator”:

nQuyen = 1;

break;

case “Admin”:

goto case “Administrator”;

default:

nQuyen = 2;

break;

}

c. Câu lệnh for

Cú pháp:

for ( [khởi\_tạo\_biến\_đếm]; [biểu\_thức]; [gia\_tăng\_biến\_đếm] )

khối lệnh;

Ví dụ 3-4 Tính tổng các số nguyên từ a đến b

int a = 10; int b = 100; int nTong = 0;

d. Câu lệnh while

Cú pháp:

while ( biểu\_thức\_logic )

khối\_lệnh;

Khối\_lệnh sẽ được thực hiện cho đến khi nào biểu thức còn đúng. Nếu ngay từ đầu biểu thức sai, khối lệnh sẽ không được thực thi.

e. Câu lệnh do …while

Cú pháp:

do

khối\_lệnh

while ( biếu\_thức\_logic )

Khác với while khối lệnh sẽ được thực hiện trước, sau đó biệu thức được kiểm tra. Nếu biểu thức đúng khối lệnh lại được thực hiện.

### 2.2/. Lập trình ứng dụng C# Winform

2.2.1/. Lớp và đối tượng

a. Đối tượng

- Trong lập trinh hướng đối tượng, tất cả thực thể trong hệ thống đều được coi là các đối tượng cụ thể. Đối tượng là một thực thể hoạt động khi chạy chượng trình.

- Một đối tượng là một thực thể đang tồn tại trong hệ thống và được xác định bằng ba yếu tố:

* **Định danh đối tượng**: xác định duy nhất cho mỗi đối tượng trong hệ thống, nhằm phân biết các đối tượng với nhau.
* **Trạng thái của đối tượng**: là sự tổ hợp của các giá trị của các thuộc tính mà đối tượng đang có.
* **Hoạt động của đối tượng**: là các hành động mà đối tượng có khả năng thực hiện được.

b. Lớp

- Định nghĩa lớp: Trong lập trình hướng đối tượng, đối tượng là một thực thể cụ thể, tồn tại trong hệ thống. Trong khi đó, lớp là một khái niệm trừu tượng, dùng để chỉ một tập hợp các đối tượng cùng loại.

Ví dụ **Khai báo một lớp**

class PackNhomTK

{

public int MaNhom { set; get; }

public string TenNhom { set; get; }

public string GhiChu { set; get; }

}

**Lớp và đối tượng mặc dù có mối tượng quan, nhưng bản chất lại khác nhau:**

* Lớp là sự trừu tượng hóa các đối tượng. Trong khi đó, đối tượng là thể hiện cụ thể của một lớp.
* Đối tượng là một thực thể cụ thể, có thực, tồn tại trong hệ thống. Trong khi đó lớp là một khai niệm trừu tượng, chỉ tồn tại ở dạng khái niệm để mô tả các đặc tính chung của một nhóm đối tượng.
* Tất cả các đối tượng thuộc cùng một lớp giống nhau về thuộc tính và phương thức.

c. Kế thừa lớp

- Một tính năng then chốt của lập trình hướng đối tượng đó là tính kế thừa. Nhờ vào tính kế thừa, nó cho phép một lớp có thể dẫn xuất từ một lớp khác hay nói các khác một lớp có thể thừa có thể sử dụng lại các thuộc tính và phương thức của lớp bị kế thừa, chính vì thế chúng sẽ tự động tiếp nhận các thành viên của bố mẹ và bổ sung thêm các thành viên của riêng chúng. Tính kế thừa cho phép lớp mới có thể nhận được mọi dữ liệu thành viên (private, protected, public) và các hàm thành viên (trừ hàm tạo, hàm hủy, hàm bạn và hàm toán tử gán =)**.**

d. Không gian tên (namespace)

- Namespace trong ngôn ngữ C#, nhằm tránh sự xung đột giữa việc sử dụng các thư viện khác nhau từ các nhà cung cấp. Ngoài ra, namespace được xem như là tập hợp các lớp đối tượng, và cung cấp duy nhất các định dan cho các kiểu dữ liệu và được đặt trong một cấu trúc phân cấp. Việc sử dụng namespace trong khi lập trình là một thói quen tốt, bởi vì công việc này chính là cách lưu các mã nguồn để sử dụng về sau. Ngoài thư viện namespace do MS.NET và các hãng thứ ba cung cấp, ta có thể tạo riêng cho mình các namespace.

**Các thành phần cơ bản:**

* Form: Dùng để chứa tất cả các thành phần của chương trình.
* Panel: Dùng để chứa các thành phần như Button, TexBox, Label,....
* Button: Nút để nhấn làm một công việc nào đó.
* TextBox: Dùng để nhập văn bản, một dòng hoặc nhiều dòng.
* Label: Dùng để hiển thị văn bản hoặc thông tin trên Form.
* ComboBox: các lựa chọn có sẵn để bạn chọn.

# 

# CHƯƠNG II : PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN CHƯƠNG TRÌNH QUẢN LÝ BÁN THỨC ĂN NHANH

## **1/. Phân tích chức năng:**

* Ứng dụng quản lý bán thức ăn nhanh giúp cho công việc bán hàng trở nên dễ dàng, nhanh chóng, độ chính xác cao.

Chức năng ứng dụng:

* Quản lý khách hàng: Thêm, xóa, sửa, lưu trữ, hiển thị thông tin khách hàng.
* Quản lý loại món ăn, món ăn: Thêm, xóa, sửa, lưu trữ, hiển thị danh sách loại, món ăn.
* Quản lý nhân viên: Thêm, xóa, sửa, lưu trữ, hiển thị thông tin nhân viên.
* Hóa đơn đặt món: Thêm, xóa, in hóa đơn ra file Excel, hiển thị chi tiết hóa đơn.
* Tìm kiếm hóa đơn: Tìm kiếm hóa đơn theo số hóa đơn, tháng/năm, theo mã nhân viên, mã khách hàng và hiển thị thông tin chi tiết hóa đơn.

## **1.1/. Sơ đồ Usecase:**

UseCase tổng quát:

Diagram

Description automatically generated

Hình Usecase Tổng quát

UseCase Quản lý khách hàng:

Diagram

Description automatically generated

Hình Usecase Quản lý khách hàng

UseCase Quản lý nhân viên:

Diagram

Description automatically generated

Hình Usecase Quản lý nhân viên

UseCase Quản lý món ăn:

Diagram

Description automatically generated

Hình Usecase Quản lý món ăn

UseCase Quản lý hóa đơn:

Diagram

Description automatically generated

Hình Usecase Quản lý hóa đơn

## **2/. Sơ đồ class**

* Dựa vào các Usecase ở trên, ta xác định được các lớp: KhachHang, NhanVien, LoaiMon, MonAn, HoaDon, ChiTietHD, Gia, KhuyenMai.
* Xác định thuộc tính và phương thức các lớp
* KhachHang: MaKH string, TenKH string, DiaChiKH string, DienThoaiKH string.
* NhanVien: MaNV string, TenNV string, GioiTinh string, DiaChiNV string, DienThoaiNV string.
* LoaiMon: TenLoai string.
* MonAn: MaMon string, TenMon string, GhiChu string.
* HoaDon: MaHD string, MaNV string, NgayBan date, MaKH string, TongTien float.
* ChiTietHD: MaHD string, MaMon string, SoLuong int, DonGia float, MaKhuyenMai string.
* Gia: MaGia string, MaMon string, DonGia float.
* KhuyenMai: MaKhuyenMai string, MaMon string, TongTien float.

Diagram

Description automatically generated

Hình Sơ đồ class hệ thống quản lý quán pizza

## **3/. Cơ sở dữ liệu**

### 3.1/. Lược đồ cơ sở dữ liệu quan hệ:

KhachHang (MaKH, TenKH, DiaChiKH, DienThoaiKH)

NhanVien (MaNV, TenNV, GioiTinh, DiaChiNV, DienThoaiNV, NgaySinhNV)

LoaiMon (MaLoai, LoaiMon)

MonAn (MaMon, TenMon, **MaLoai**, Gia, GhiChu)

HoaDon (MaHD, **MaNV**, NgayBan, **MaKH**, TongTien)

ChiTietHD (**MaHD**, **MaMon**, SoLuong, DonGia, GiamGia, ThanhTien)

Diagram

Description automatically generated

Hình Lược đồ CSDL quan hệ

### 3.2/. Câu lệnh truy vấn SQL

use master

go

create database QLTAN

go

use QLTAN

go

create table LoaiMon (

MaLoai nvarchar (50) not null,

LoaiMon nvarchar (50) null,

primary key clustered (MaLoai ASC)

)

go

create table KhachHang (

MaKH nvarchar (10) not null,

TenKH nvarchar (50) not null,

DiaChiKH nvarchar (50) not null,

DienThoaiKH nvarchar (15) not null,

primary key clustered (MaKH ASC)

)

go

create table MonAn (

MaMon nvarchar (50) not null,

TenMon nvarchar (50) not null,

MaLoai nvarchar (50) not null,

LoaiMon nvarchar (50) null,

Gia float not null,

GhiChu nvarchar (200) null,

primary key clustered (MaMon ASC),

constraint Fk\_LoaiMon foreign key(MaLoai) references LoaiMon(MaLoai)

)

go

create table NhanVien (

MaNV nvarchar (10) not null,

TenNV nvarchar (50) not null,

GioiTinh nvarchar (10) not null,

DiaChiNV nvarchar (50) not null,

DienThoaiNV nvarchar (15) not null,

NgaySinhNV date not null,

primary key clustered (MaNV ASC)

)

go

create table HoaDon (

MaHD nvarchar (30) not null,

MaNV nvarchar (10) not null,

NgayBan date not null,

MaKH nvarchar (10) not null,

TongTien float not null,

primary key clustered (MaHD ASC),

constraint Fk\_NhanVien foreign key(MaNV) references NhanVien(MaNV),

constraint Fk\_KhachHang foreign key(MaKH) references KhachHang(MaKH)

)

go

create table ChiTietHD (

MaHD nvarchar (30) not null,

MaMon nvarchar (50) not null,

SoLuong int not null,

DonGia float not null,

GiamGia float not null,

ThanhTien float not null,

TenMon nvarchar (50) not null,

MaNV nvarchar (10) not null,

MaKH nvarchar (10) not null,

NgayBan date not null,

primary key clustered (MaHD ASC, MaMon ASC),

)

INSERT INTO NhanVien (MaNV, TenNV, GioiTinh, DiaChiNV, DienThoaiNV, NgaySinhNV)

VALUES ('1', N'Khoa Nguyễn', 'Nam',N'TP.Long Xuyên','0989898989', '3-26-1999');

INSERT INTO NhanVien (MaNV, TenNV, GioiTinh, DiaChiNV, DienThoaiNV, NgaySinhNV)

VALUES ('2', N'Mây Mây', 'Nữ',N'TP.Long Xuyên','0389486472', '4-27-1997');

INSERT INTO NhanVien (MaNV, TenNV, GioiTinh, DiaChiNV, DienThoaiNV, NgaySinhNV)

VALUES ('3', N'Duy Huỳnh', 'Nam',N'TP.Long Xuyên','0386886472', '9-28-2000');

INSERT INTO NhanVien (MaNV, TenNV, GioiTinh, DiaChiNV, DienThoaiNV, NgaySinhNV)

VALUES ('4', N'Thư Huỳnh', 'Nữ',N'TP.Long Xuyên','0377657738', '9-13-2005');

INSERT INTO LoaiMon (MaLoai, LoaiMon)

VALUES ('1', N'Gà rán');

INSERT INTO LoaiMon (MaLoai, LoaiMon)

VALUES ('2', N'Nước ngọt');

INSERT INTO LoaiMon (MaLoai, LoaiMon)

VALUES ('3', N'Đồ ăn vặt');

INSERT INTO LoaiMon (MaLoai, LoaiMon)

VALUES ('4', N'Pizza');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('1', N'Đùi gà KFC', '1','36000', N'Đùi gà rất ngon', N'Gà rán');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('2', N'Pepsi','2', '10000', N'Pepsi rất ngon', N'Nước ngọt');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('3', N'Bò húc','2','12000', N'Bò húc rất ngon', N'Nước ngọt');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('4', N'Oishi phồng tôm', '3','5000',N'Snack rất ngon', N'Đồ ăn vặt');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('5', N'Snack khoai tây', '3','20000',N'Snack rất ngon', N'Đồ ăn vặt');

INSERT INTO MonAn (MaMon, TenMon, MaLoai, Gia, GhiChu, LoaiMon)

VALUES ('6', N'Pizza bò và tôm', '4','89000',N'Pizza rất ngon', N'Pizza');

INSERT INTO KhachHang (MaKH, TenKH, DiaChiKH, DienThoaiKH)

VALUES ('1', N'Nguyễn Khánh Luân', N'TP.Long Xuyên','0493972997');

INSERT INTO KhachHang (MaKH, TenKH, DiaChiKH, DienThoaiKH)

VALUES ('2', N'Võ Tấn Lợi', N'TP.Long Xuyên','0850384977');

INSERT INTO KhachHang (MaKH, TenKH, DiaChiKH, DienThoaiKH)

VALUES ('3', N'Nguyễn Hoàng Huy', N'TP.Long Xuyên','0937458432');

INSERT INTO KhachHang (MaKH, TenKH, DiaChiKH, DienThoaiKH)

VALUES ('4', N'Bạch Nhật Linh', N'TP.Long Xuyên','0937347365');

# CHƯƠNG III : THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ BÁN THỨC ĂN NHANH

## **1/. Chi tiết các chức năng và phương thức:**

* Là một chương trình quản lý bán thức ăn nhanh nên các chức năng thêm, xóa, sửa, lưu là không thể thiếu. Ngoài chức năng quản lý, nhân viên có thể ghi hóa đơn cho khách hàng muốn đặt món và tìm kiếm hóa đơn cũ một cách dễ dàng.
* Thiết kế các chức năng chính và phương thức cho việc xây dựng chương trình:
* Kết nối CSDL với project Fastfood: Sau khi hoàn thành và thêm dữ liệu vào các bản, ta dùng Sever Explorer để kết nối với CSDL đó, copy data source trong mục Advance.
* Tạo class kết nối DataConnection: class kết nối này có đường dẫn data source lúc nãy ta copy, được khai báo ở các form để dùng dữ liệu từ database.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace Fastfood

{

class DataConnection

{

string conStr;

public DataConnection()

{

conStr = "Data Source=DESKTOP-TM6ESLE\\SQLEXPRESS;Initial Catalog=QLTAN;Integrated Security=True";

}

public SqlConnection GetConnection()

{

return new SqlConnection(conStr);

}

}

}

* Hiển thị dữ liệu bằng Datagridview: Khi người dùng click vào từng form chức năng, dữ liệu ở các bảng sẽ được lấy từ câu truy vấn các bảng đó đổ lên Datagridview được viết trong hàm DisplayData(), và trong form load bất kì ta gọi hàm DisplayData() để hiển thị dữ liệu mà ta truy vấn tương ứng:

private void frmDMNhanVien\_Load(object sender, EventArgs e)

{

txtMaNV.Enabled = false;

btnLuu.Enabled = false;

btnBoQua.Enabled = false;

DisplayData();

}

private void DisplayData() // Hiển thị dữ liệu

{

SqlConnection conn = dc.GetConnection();

conn.Open();

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select \* from NhanVien", conn);

dgvNhanVien.AllowUserToAddRows = false;

dgvNhanVien.EditMode = DataGridViewEditMode.EditProgrammatically;

adapt.Fill(dt);

dgvNhanVien.DataSource = dt;

conn.Close();

}

* Sự kiện Button Thêm Click: Khi người dùng muốn thêm 1 hóa đơn, 1 nhân viên hay khách hàng nào đó, thì một số textbox chính và button sẽ được mở các để cho người dùng nhập dữ liệu vào, rõ hơn là chuyển từ trạng thái đọc sang trạng thái ghi dữ liệu qua công cụ Enabled = true (hiển thị) hoặc false (ẩn). Cụ thể:

private void btnThem\_Click(object sender, EventArgs e)

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnBoQua.Enabled = true;

btnLuu.Enabled = true;

btnThem.Enabled = false;

ResetValues();

txtMaNV.Enabled = true;

txtMaNV.Focus();

}

* Sự kiện Button Bỏ qua Click: Vẫn sử dụng công cụ Enabled, khi người dùng muốn từ trạng thái ghi dữ liệu sang trạng thái đọc. Cụ thể:

private void btnBoQua\_Click(object sender, EventArgs e)

{

ResetValues();

btnBoQua.Enabled = false;

btnThem.Enabled = true;

btnXoa.Enabled = true;

btnSua.Enabled = true;

btnLuu.Enabled = false;

txtMaNV.Enabled = false;

}

* Sự kiện Datagridview Click: Khi người dùng click vào dữ liệu ở Datagridview, dữ liệu sẽ được đổ lên các textbox, combobox trên form tương ứng, tiện cho việc xóa, sửa dữ liệu. Cụ thể:

private void dgvNhanVien\_Click(object sender, EventArgs e)

{

if (btnThem.Enabled == false)

{

MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaNV.Focus();

return;

}

txtMaNV.Text = dgvNhanVien.CurrentRow.Cells["MaNV"].Value.ToString();

txtTenNV.Text = dgvNhanVien.CurrentRow.Cells["TenNV"].Value.ToString();

if (dgvNhanVien.CurrentRow.Cells["GioiTinh"].Value.ToString() == "Nam") chkGioiTinh.Checked = true;

else chkGioiTinh.Checked = false;

txtDiaChiNV.Text = dgvNhanVien.CurrentRow.Cells["DiaChiNV"].Value.ToString();

txtDienThoaiNV.Text = dgvNhanVien.CurrentRow.Cells["DienThoaiNV"].Value.ToString();

dateTimePicker1.Text = dgvNhanVien.CurrentRow.Cells["NgaySinhNV"].Value.ToString();

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnBoQua.Enabled = true;

}

* Sự kiện Button Lưu: Khi người dùng chọn Thêm, các ô textbox sẽ được hiển thị cho việc nhập dữ liêu, sau khi hoàn tất quá trình nhập dữ liệu ta chọn Lưu, dữ liệu sẽ được execute vào database theo câu truy vấn Insert, gọi hàm DisplayData() để dữ liệu hiển thị lên Datagridview ngay lập tức. Cụ thể, ta xét từng điều kiện trong quá trình nhập dữ liệu, rồi sau đó ghi dữ liệu, hàm ResetValues() để trả dữ liệu về trạng thái ban đầu:

private void btnLuu\_Click(object sender, EventArgs e)

{

SqlConnection conn = dc.GetConnection();

if (chkGioiTinh.Checked == true)

gt = "Nam";

else

gt = "Nữ";

if (txtMaNV.Text != "" && txtTenNV.Text != "" && txtDiaChiNV.Text != "" && txtDienThoaiNV.Text != "")

{

cmd = new SqlCommand("insert into NhanVien(MaNV, TenNV, DiaChiNV, DienThoaiNV, NgaySinhNV, GioiTinh) values(@manv,@tennv,@diachinv,@dienthoainv,@ngaysinhnv,@gioitinh)", conn);

conn.Open();

cmd.Parameters.AddWithValue("@manv", txtMaNV.Text);

cmd.Parameters.AddWithValue("@tenNV", txtTenNV.Text);

cmd.Parameters.AddWithValue("@diachinv", txtDiaChiNV.Text);

cmd.Parameters.AddWithValue("@dienthoainv", txtDienThoaiNV.Text);

cmd.Parameters.AddWithValue("@ngaysinhnv", dateTimePicker1.Text);

cmd.Parameters.AddWithValue("@gioitinh", gt);

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Thêm vào thành công");

DisplayData();

ResetValues();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnSua.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaNV.Enabled = false;

}

}

private void ResetValues()

{

txtMaNV.Text = "";

txtTenNV.Text = "";

chkGioiTinh.Checked = false;

txtDiaChiNV.Text = "";

dateTimePicker1.Text = "";

txtDienThoaiNV.Text = "";

}

* Sự kiện Button Xóa Click: Khi người dùng muốn xóa thông tin của loại, món ăn, khách hàng, nhân viên hay hóa đơn nào đó chỉ cần ta click chuột vào hàng dữ liệu muốn xóa trên Datagridview và chọn Xóa, dữ liệu hàng đó sẽ biến mất ngay lập tức. Cụ thể phương thức được viết dựa trên câu lệnh Delete trong Sql:

private void btnXoa\_Click(object sender, EventArgs e)

{

SqlConnection conn = dc.GetConnection();

if (txtMaNV.Text != "")

{

cmd = new SqlCommand("delete NhanVien where MaNV=@manv", conn);

conn.Open();

cmd.Parameters.AddWithValue("@manv", txtMaNV.Text);

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Xóa nhân viên thành công");

DisplayData();

ResetValues();

}

else

{

MessageBox.Show("Vui lòng nhập mã nhân viên để xóa");

}

}

* Sự kiện Button Sửa Click: Khi người dùng muốn sửa thông tin của khách hàng, nhân viên hoặc loại, món ăn nào đó (không thể sửa thông tin hóa đơn) thì chỉ cần click vào hàng dữ liệu muốn sửa trên Datagridview, dữ liệu đó sẽ được đổ lên các textbox, combobox để cho người dùng sửa, sau khi hoàn tất quá trình sửa dữ liệu, ta chọn Sửa. Cụ thể phương thức trên được viết dựa trên câu truy vấn Update trong Sql:

private void btnSua\_Click(object sender, EventArgs e)

{

SqlConnection conn = dc.GetConnection();

if (txtMaNV.Text != "")

{

if (chkGioiTinh.Checked == true)

gt = "Nam";

else

gt = "Nữ";

cmd = new SqlCommand("update NhanVien set MaNV=@manv, TenNV=@TenNV, DiaChiNV=@diachinv, DienThoaiNV=@dienthoainv, NgaySinhNV=@ngaysinhnv, GioiTinh=@gioitinh where MaNV=@manv", conn);

conn.Open();

cmd.Parameters.AddWithValue("@manv", txtMaNV.Text);

cmd.Parameters.AddWithValue("@tenNV", txtTenNV.Text);

cmd.Parameters.AddWithValue("@diachinv", txtDiaChiNV.Text);

cmd.Parameters.AddWithValue("@dienthoainv", txtDienThoaiNV.Text);

cmd.Parameters.AddWithValue("@ngaysinhnv", dateTimePicker1.Text);

cmd.Parameters.AddWithValue("@gioitinh", gt);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công");

conn.Close();

DisplayData();

ResetValues();

btnBoQua.Enabled = false;

}

else

{

MessageBox.Show("Vui lòng nhập mã nhân viên để cập nhật");

}

}

* Sự kiện Button Thoát Click: Khi click vào nút này, form hiện tại sẽ đóng và trở về giao diện trang chủ.

private void btnThoat\_Click(object sender, EventArgs e)

{

this.Close();

}

* Sự kiện Button In Click: Hiển thị ở form Hóa đơn, sau quá trình thêm hóa đơn thành công, người dùng chọn In thì ứng dụng Excel trong máy sẽ hiện lên và ghi toàn bộ dữ liệu hóa đơn đó và tự động lưu vào folder ta đã định sẵn. Cụ thể:

private void btnInHoaDon\_Click(object sender, EventArgs e)

{

//Nút Export to Excel

// creating Excel Application \_ Tạo ứng dụng Excel

Microsoft.Office.Interop.Excel.\_Application app = new Microsoft.Office.Interop.Excel.Application();

// Tạo WorkBook mới

Microsoft.Office.Interop.Excel.\_Workbook workbook = app.Workbooks.Add(Type.Missing);

// Tạo Sheet mới trong WorkBook

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

// Xem Sheet Excel

app.Visible = true;

worksheet = workbook.Sheets["Sheet1"];

worksheet = workbook.ActiveSheet;

//Đổi tên Sheet thành Exported from Data gridview

worksheet.Name = "Exported from Data gridview";

//Tạo tiêu đề cột

for (int i = 1; i < dgvHoaDon.Columns.Count + 1; i++)

{

worksheet.Cells[1, i] = dgvHoaDon.Columns[i - 1].HeaderText;

}

//lấy dữ liệu từng dòng

for (int i = 0; i < dgvHoaDon.Rows.Count - 1; i++)

{

for (int j = 0; j < dgvHoaDon.Columns.Count; j++)

{

worksheet.Cells[i + 2, j + 1] = dgvHoaDon.Rows[i].Cells[j].Value.ToString();

}

}

//Lưu file .Xlsx

workbook.SaveAs("D:\\Fastfood\\output.xlsx", Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

//Thoát khỏi Excel

app.Quit();

}

* Sự kiện Button Tìm kiếm Click: Được hiển thị ở form Tìm kiếm hóa đơn, hóa đơn được tìm kiếm dựa trên 1 trong nhiều bộ lọc như tìm kiếm theo số hóa đơn, tìm kiếm theo mã nhân viên, tìm kiếm theo tháng, năm, v.v…

private void btnThem\_Click(object sender, EventArgs e)

{

string sql;

if ((txtMaHD.Text == "") && (txtThang.Text == "") && (txtNam.Text == "") &&

(txtMaNV.Text == "") && (txtMaKH.Text == "") &&

(txtThanhTien.Text == ""))

{

MessageBox.Show("Hãy nhập một điều kiện tìm kiếm!!!", "Yêu cầu ...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

//MaHD, MaNV, NgayBan, MaKH, ThanhTien

sql = "SELECT \* FROM ChiTietHD WHERE 1=1";

if (txtMaHD.Text != "")

sql = sql + " AND MaHD Like N'%" + txtMaHD.Text + "%'";

if (txtThang.Text != "")

sql = sql + " AND MONTH(NgayBan) =" + txtThang.Text;

if (txtNam.Text != "")

sql = sql + " AND YEAR(NgayBan) =" + txtNam.Text;

if (txtMaNV.Text != "")

sql = sql + " AND MaNV Like N'%" + txtMaNV.Text + "%'";

if (txtMaKH.Text != "")

sql = sql + " AND MaKH Like N'%" + txtMaKH.Text + "%'";

if (txtThanhTien.Text != "")

sql = sql + " AND ThanhTien <=" + txtThanhTien.Text;

tblHDB = Functions.GetDataToTable(sql);

if (tblHDB.Rows.Count == 0)

{

MessageBox.Show("Không có bản ghi thỏa mãn điều kiện!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

MessageBox.Show("Có " + tblHDB.Rows.Count + " bản ghi thỏa mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvTimKiemHD.DataSource = tblHDB;

LoadDataGridView();

}

* Class Functions: Class dùng chung, gồm các hàm phương thức CreatKey và ConvertTimeTo24 để sinh số hóa đơn tự động và GetDataToTable để đưa dữ liệu từ bảng.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

using System.Data;

namespace Fastfood

{

class Functions

{

DataConnection dc = new DataConnection();

public static string CreateKey(string tiento)

{

string key = tiento;

string[] partsDay;

partsDay = DateTime.Now.ToShortDateString().Split('/');

//Ví dụ 07/08/2009

string d = String.Format("{0}{1}{2}", partsDay[0], partsDay[1], partsDay[2]);

key = key + d;

string[] partsTime;

partsTime = DateTime.Now.ToLongTimeString().Split(':');

//Ví dụ 7:08:03 PM hoặc 7:08:03 AM

if (partsTime[2].Substring(3, 2) == "PM")

partsTime[0] = ConvertTimeTo24(partsTime[0]);

if (partsTime[2].Substring(3, 2) == "AM")

if (partsTime[0].Length == 1)

partsTime[0] = "0" + partsTime[0];

//Xóa ký tự trắng và PM hoặc AM

partsTime[2] = partsTime[2].Remove(2, 3);

string t;

t = String.Format("\_{0}{1}{2}", partsTime[0], partsTime[1], partsTime[2]);

key = key + t;

return key;

}

public static string ConvertTimeTo24(string hour)

{

string h = "";

switch (hour)

{

case "1":

h = "13";

break;

case "2":

h = "14";

break;

case "3":

h = "15";

break;

case "4":

h = "16";

break;

case "5":

h = "17";

break;

case "6":

h = "18";

break;

case "7":

h = "19";

break;

case "8":

h = "20";

break;

case "9":

h = "21";

break;

case "10":

h = "22";

break;

case "11":

h = "23";

break;

case "12":

h = "0";

break;

}

return h;

}

public static DataTable GetDataToTable(string sql)

{

DataConnection dc = new DataConnection();

SqlConnection conn = dc.GetConnection();

SqlDataAdapter dap = new SqlDataAdapter(); //Định nghĩa đối tượng thuộc lớp SqlDataAdapter

//Tạo đối tượng thuộc lớp SqlCommand

dap.SelectCommand = new SqlCommand();

dap.SelectCommand.Connection = conn; //Kết nối cơ sở dữ liệu

dap.SelectCommand.CommandText = sql; //Lệnh SQL

//Khai báo đối tượng table thuộc lớp DataTable

DataTable table = new DataTable();

dap.Fill(table);

return table;

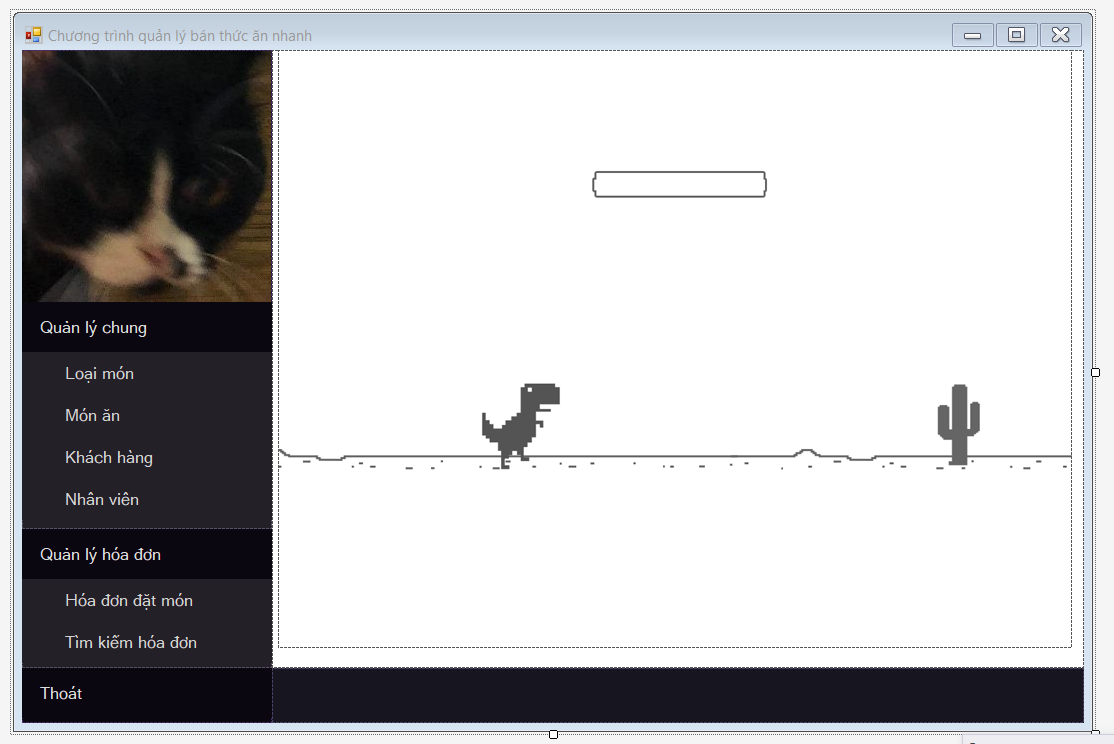
}

}

}

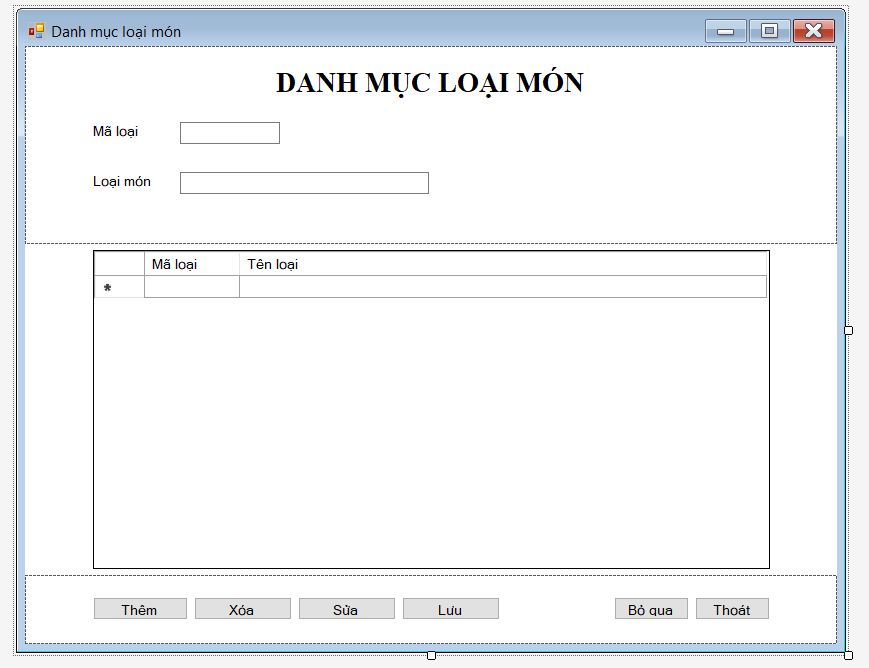
## **2/. Chi tiết giao diện các chức năng:**

* Các công cụ chính tạo nên giao diện được sử dụng:
* Label: dùng để trình bày một chuỗi văn bản thông thường nhằm mục đích mô tả thêm thông tin cho đối tượng khác.
* Textbox: dùng để nhập dữ liệu đầu vào, ngoài ra còn có thể dùng để xuất dữ liệu.
* Combobox: Hiển thị như một Textbox kết hợp với một Listbox, cho phép người dùng lựa chọn các mục từ danh sách hoặc nhập giá trị mới.
* Button: là thành phần tương tác cho phép người dùng giao tiếp với một ứng dụng.
* Datagridview: dùng để hiển thị và chỉnh sửa dữ liệu dạng bảng.
* Panel: là một container hay cơ chế nhóm control phù hợp cho giao diện chương trình.
* Thiết kế chi tiết các giao diện:
* Giao diện form menu:



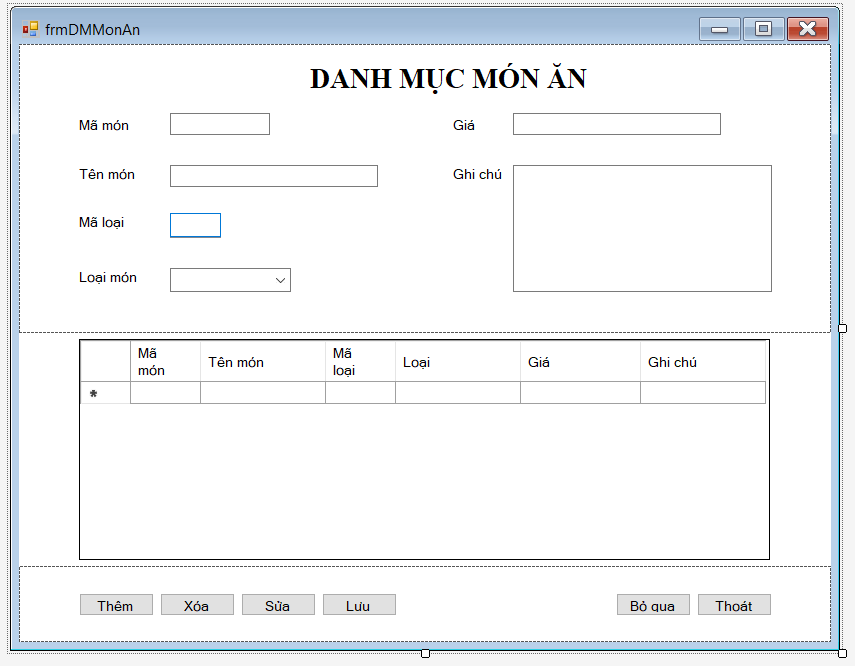
Hình Giao diện form menu

* Giao diện form quản lý loại món:



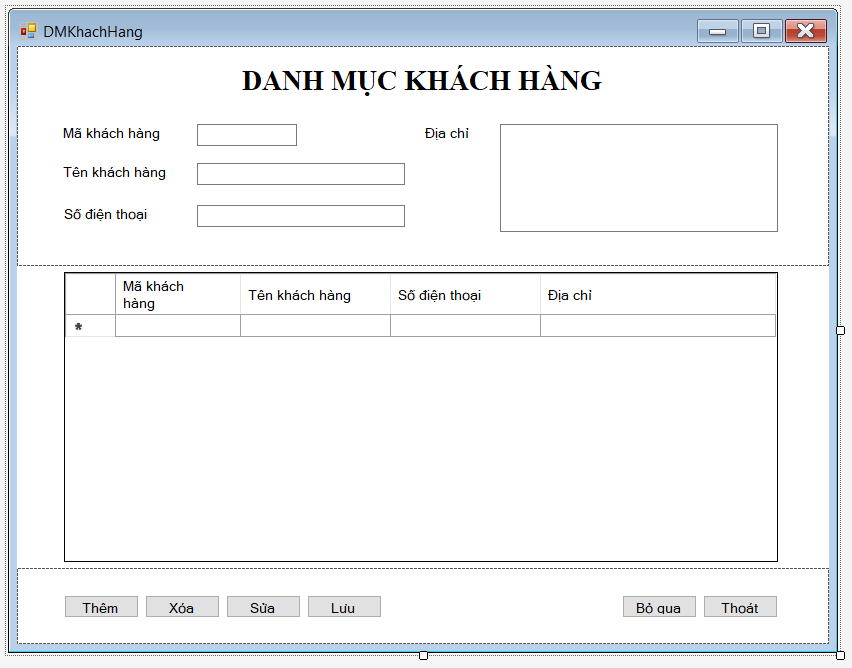
Hình Giao diện form quản lý loại món

* Giao diện form quản lý món ăn:



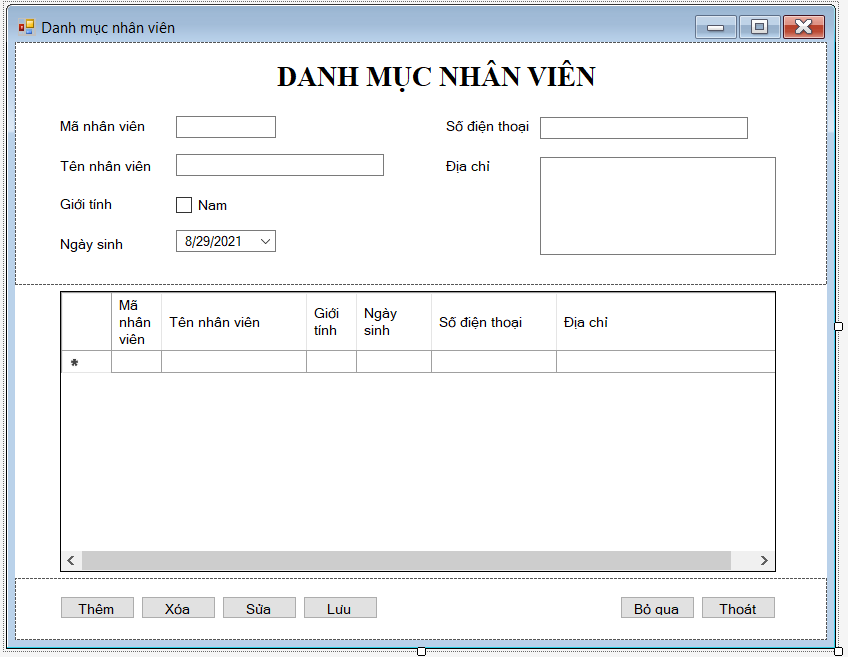
Hình Giao diện form quản lý món ăn

* Giao diện form quản lý khách hàng:



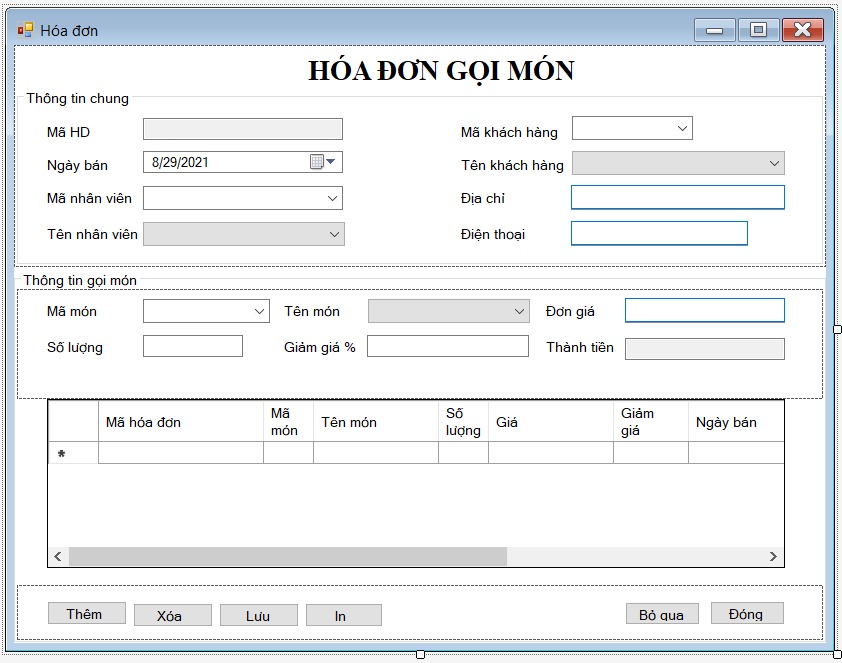
Hình Giao diện form quản lý khách hàng

* Giao diện form quản lý nhân viên:



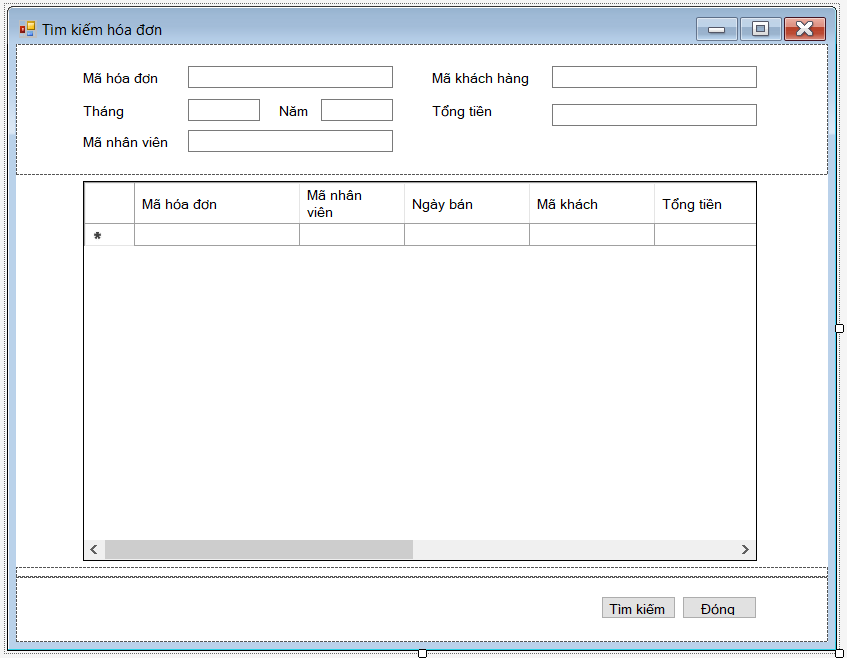
Hình Giao diện form quản lý nhân viên

* Giao diện form hóa đơn:



Hình Giao diện form hóa đơn

* Giao diện form tìm kiếm hóa đơn:



Hình Giao diện form tìm kiếm hóa đơn

# CHƯƠNG IV: THỰC NGHIỆM VÀ TRIỂN KHAI

## **1/. Đóng gói và xuất bản chương trình:**

* Để đóng gói và xuất bản một chương trình, trước hết ta phải cài đặt gói Setup Projects của Visual Studio về máy tính. Đường dẫn tải xuống: <https://marketplace.visualstudio.com/items?itemName=VisualStudioClient.MicrosoftVisualStudio2017InstallerProjects>
* Sau khi cài đặt xong, ta tiến hành tạo project Setup lúc nãy vừa cài trong solution chứa project cần đóng gói (Fastfood).



Hình 16 Project đóng gói

* Sau khi tạo xong, sẽ xuất hiện giao diện như thế này

Graphical user interface, text, application

Description automatically generated

Hình 17 Giao diện project đóng gói

* Chuột phải vào Application Folder -> Add -> Project Output

Graphical user interface, text, application, Word

Description automatically generated

Hình 18 Tạo project output

* Chọn tên project là Fastfood, nhấn vào Primary output và nhấn Ok

Graphical user interface, text, application

Description automatically generated

Hình 19 Tạo primary output

* Nhấn chuột phải vào Primary output vừa tạo và chọn Create Shortcut to Primary output và đặt tên là Fastfood
* Di chuyển Shortcut vừa tạo ở mục Application Folder vào thư mục User’s Desktop và Properties nó để thêm Icon vào
* Nhấp chuột phải vào thư mục User’s Programs Menu chọn Create New Shortcut

Graphical user interface, application, Word

Description automatically generated

Hình 20 Tạo shortcut mới

* Thêm Fastfood vào và nhấn Ok, ta thay đổi Icon tương tự như trên

Graphical user interface

Description automatically generated

Hình 21 Add Fastfood

* Tiến hành Rebuild lại project Setup.

## **2/. Môi trường cài đặt và triển khai, cách vận hành:**

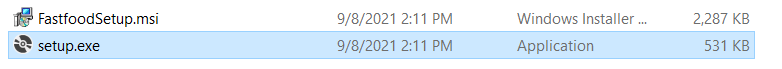
* Sau khi đã Rebuild lại project Setup, ta tiến hành truy cập vào đường dẫn nơi chứa folder Setup nãy tạo.

Text

Description automatically generated

Hình 23 Tìm đường dẫn folder Setup

* Chọn Setup -> FastfoodSetup -> Debug -> setup.exe nhấn để cài đặt và làm theo yêu cầu



Hình 24 Cài đặt Fastfood

* Quá trình cài đặt hoàn tất, sẽ xuất hiện một ứng dụng với ảnh Pizza trên màn hình máy tính

A picture containing text

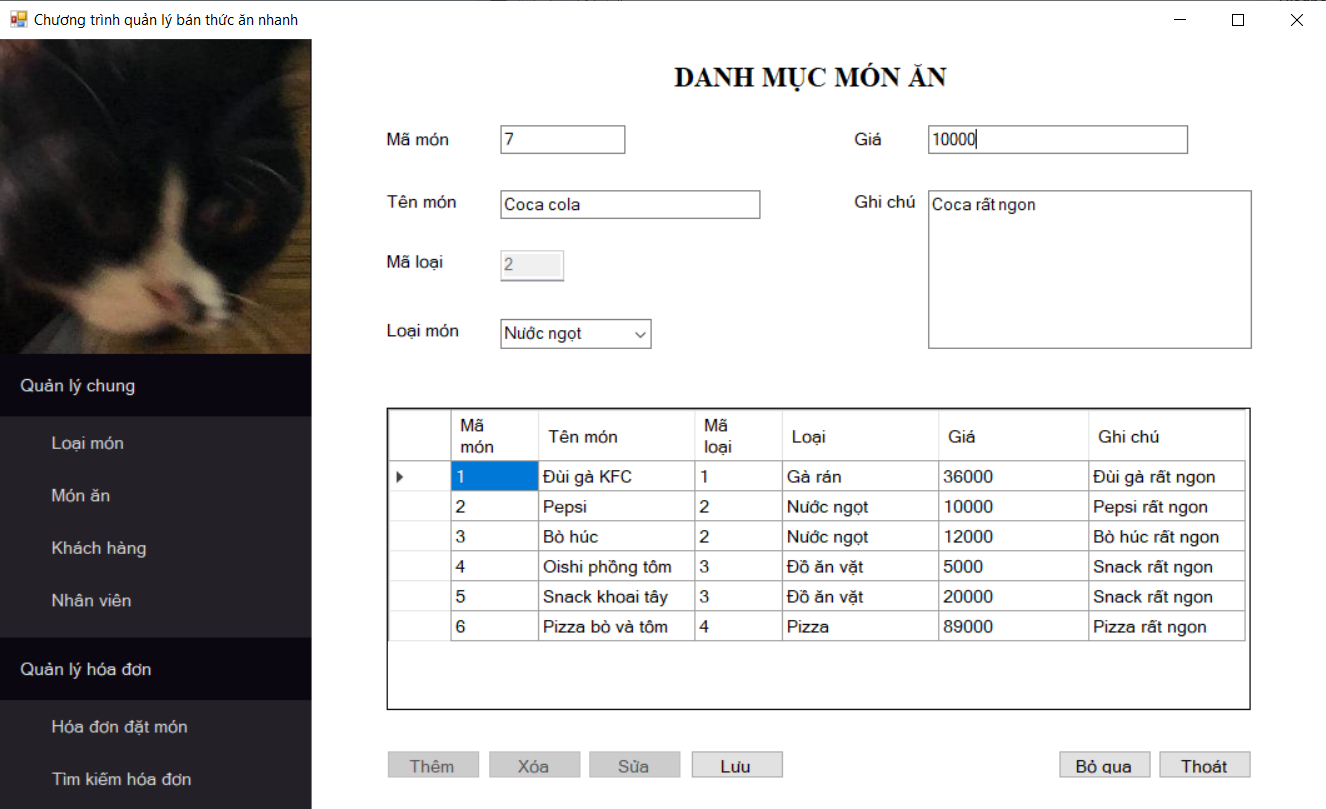
Description automatically generated

Hình 25 App fastfood

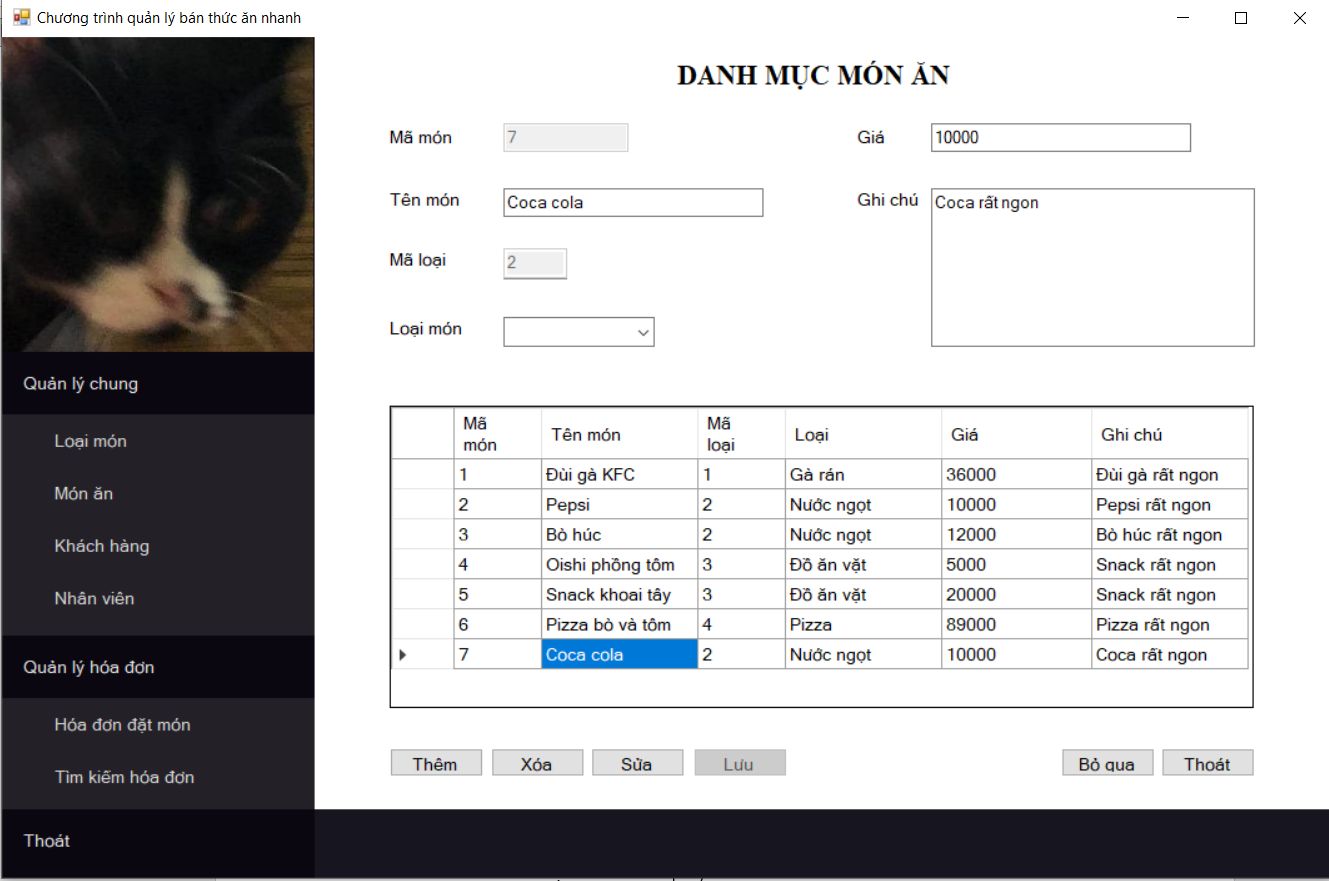
* Mở ứng dụng lên để thực hiện công việc quản lý bán thức ăn nhanh

## **3/. Demo các chức năng chính của chương trình:**

* Chức năng thêm sản phẩm

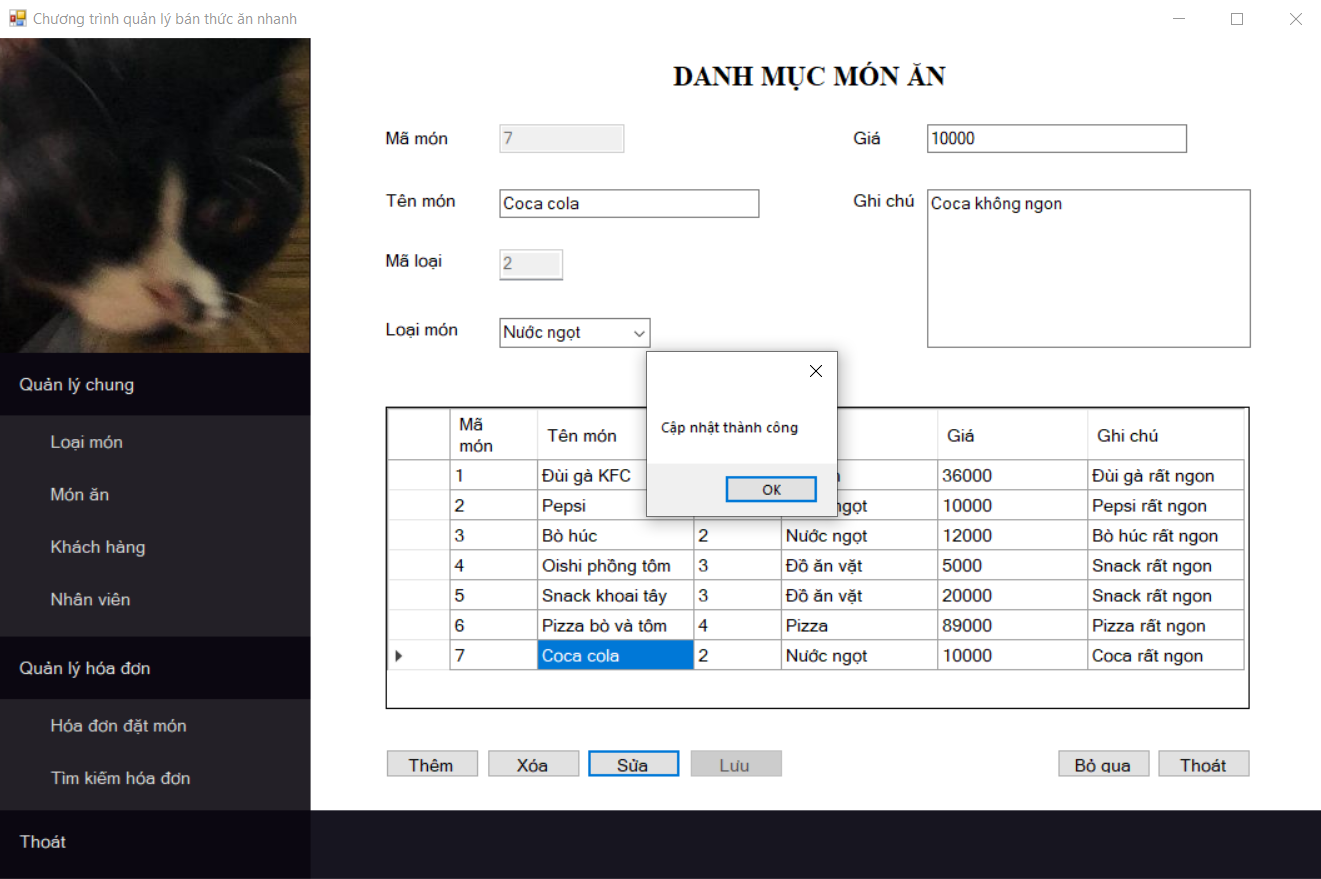


Hình Nhập thông tin để thêm sản phẩm

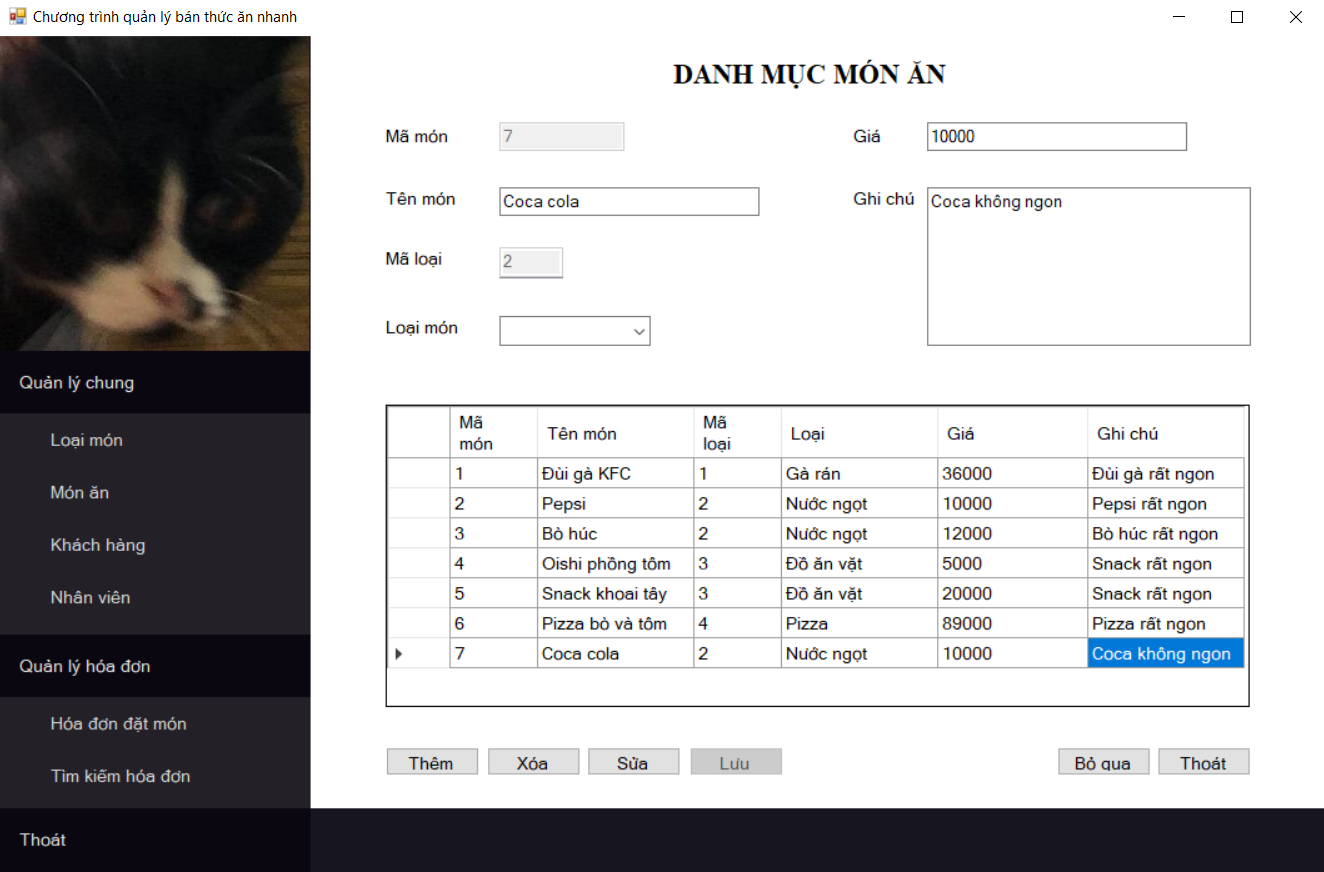


Hình Thêm sản phẩm thành công

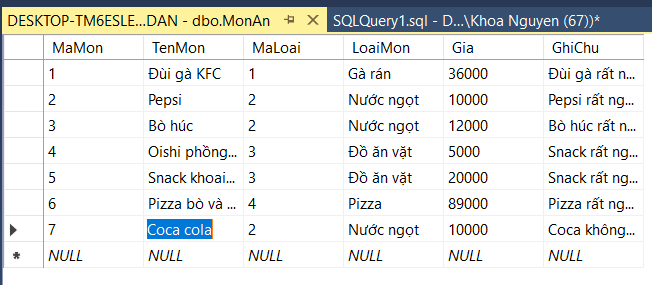
* Chức năng sửa sản phẩm



Hình Chọn dòng sản phẩm cần sửa và sửa

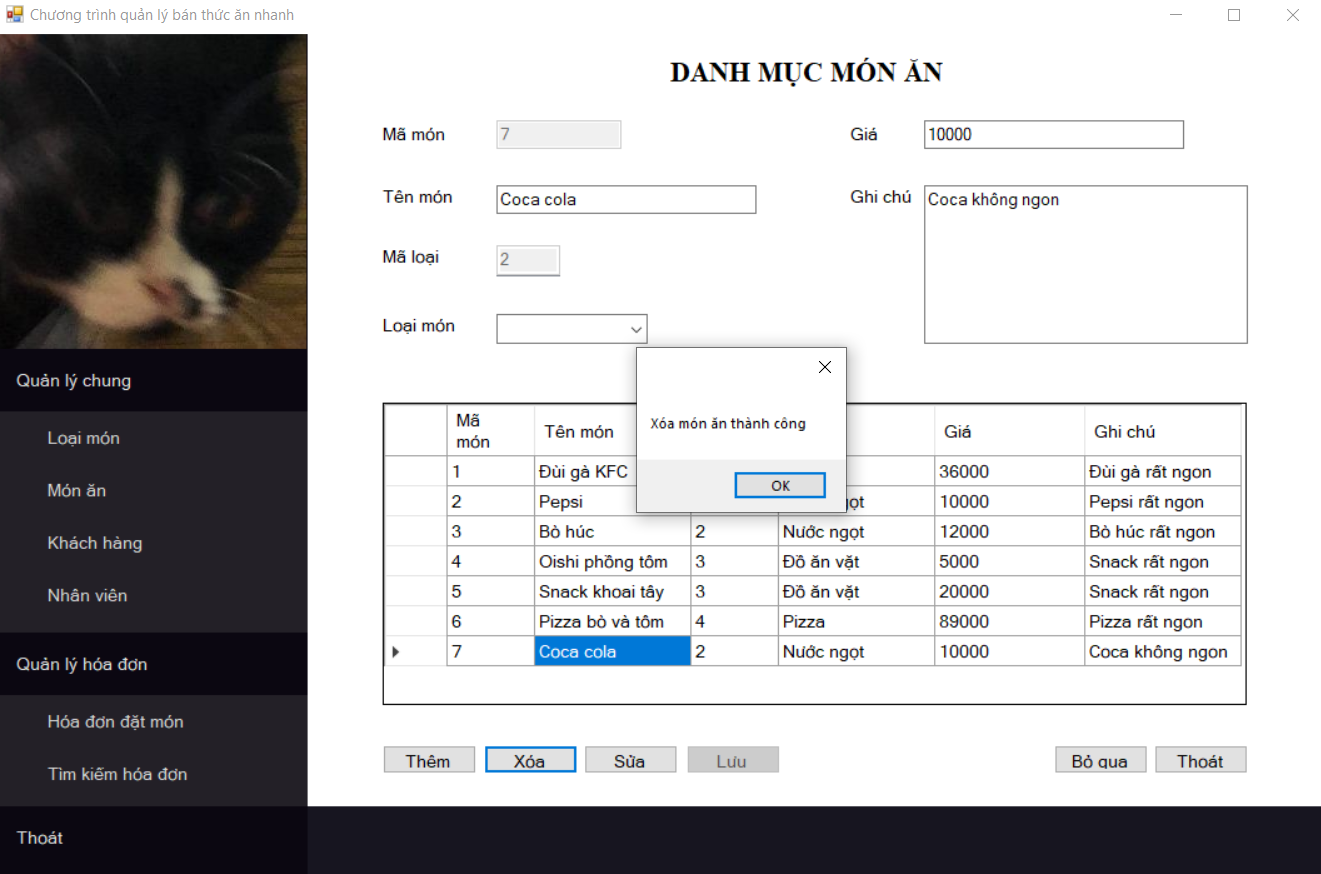


Hình Kết quả sau khi sửa sản phẩm



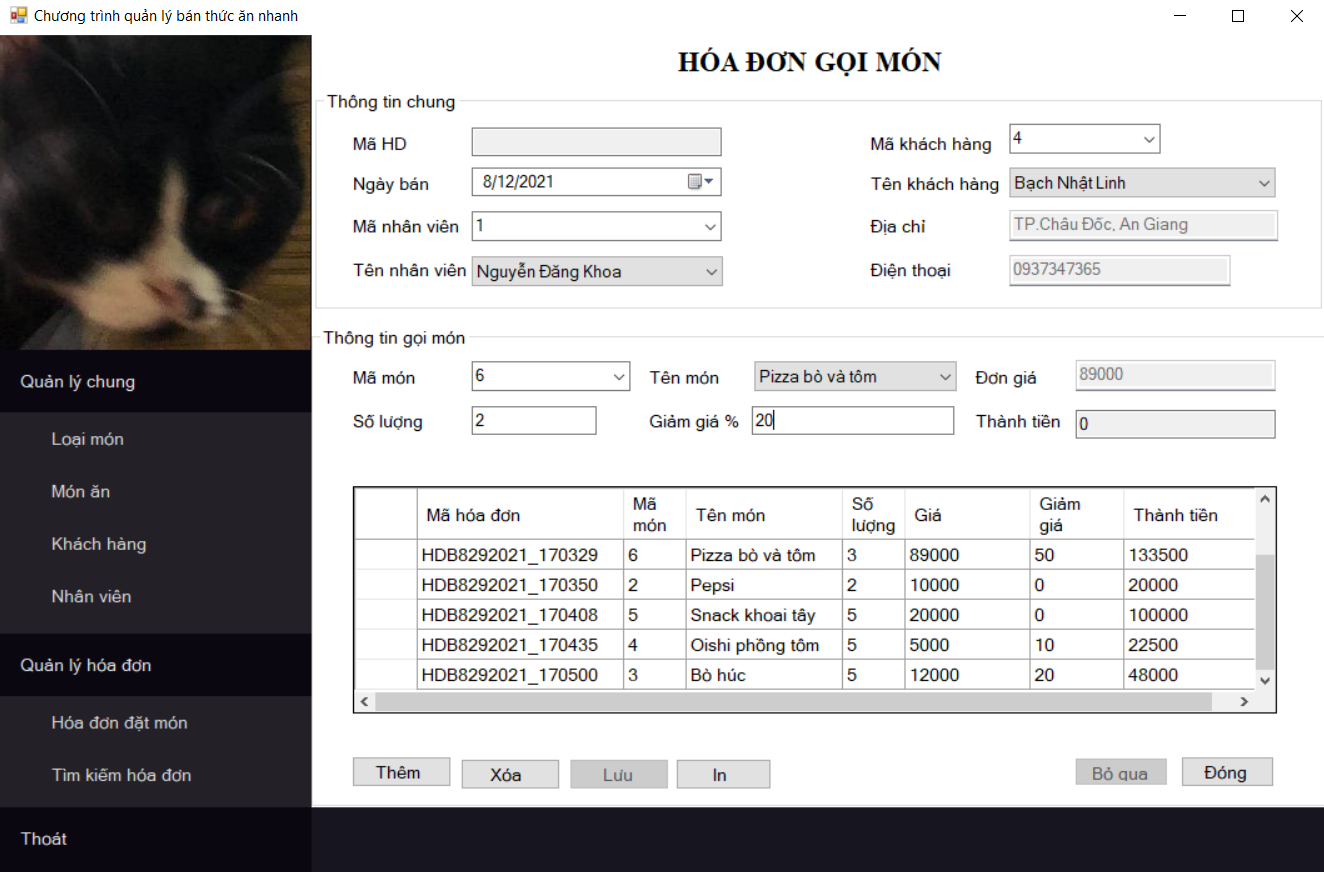
Hình Dữ liệu đã được lưu vào database

* Chức năng xóa sản phẩm:

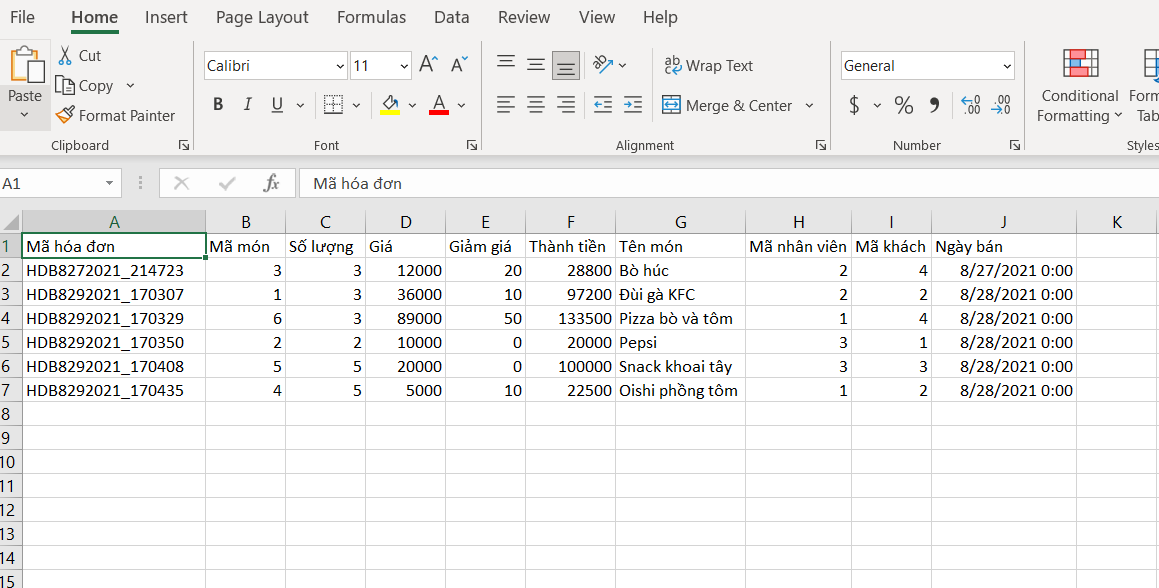


Hình Xóa sản phẩm thành công

* Chức năng in hóa đơn:

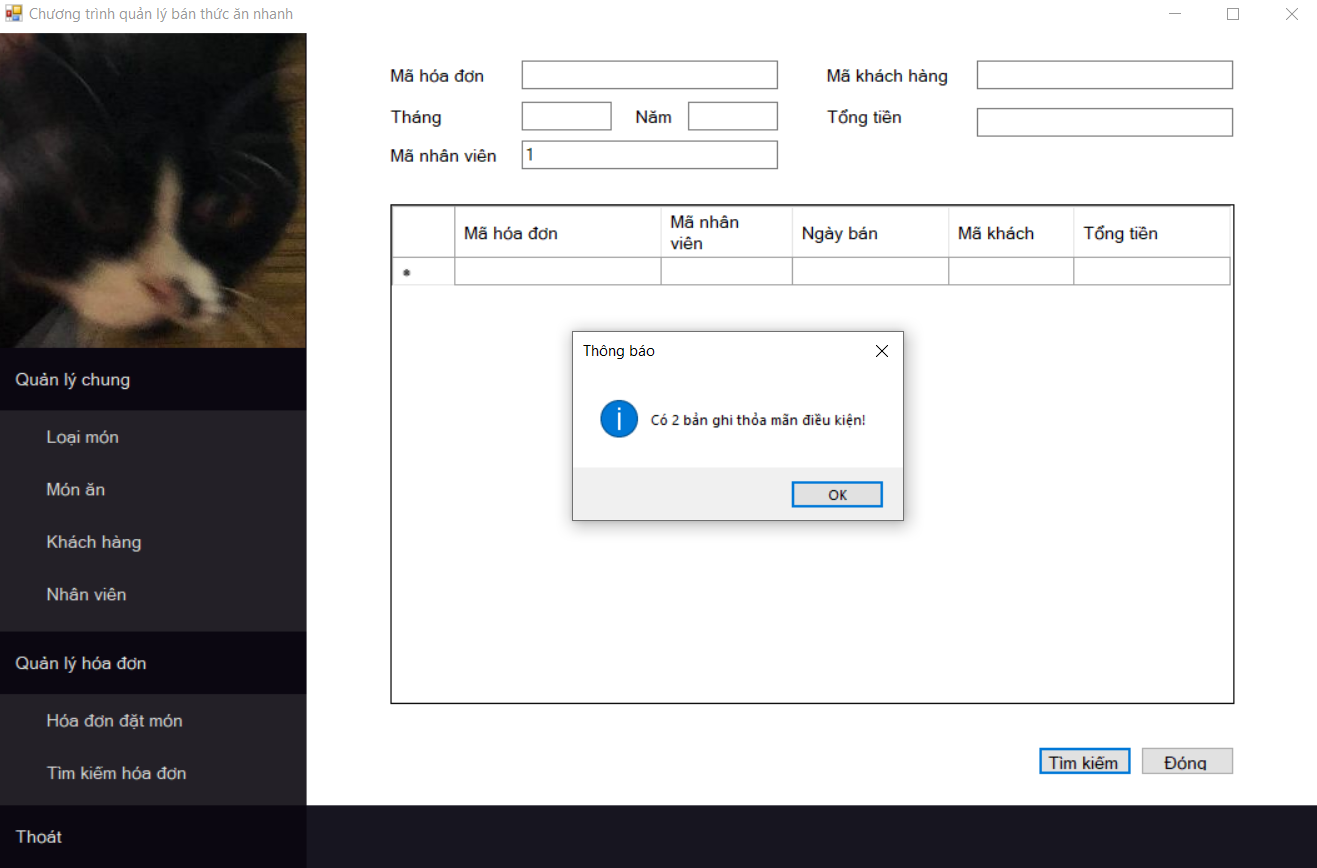


Hình Thêm hóa đơn

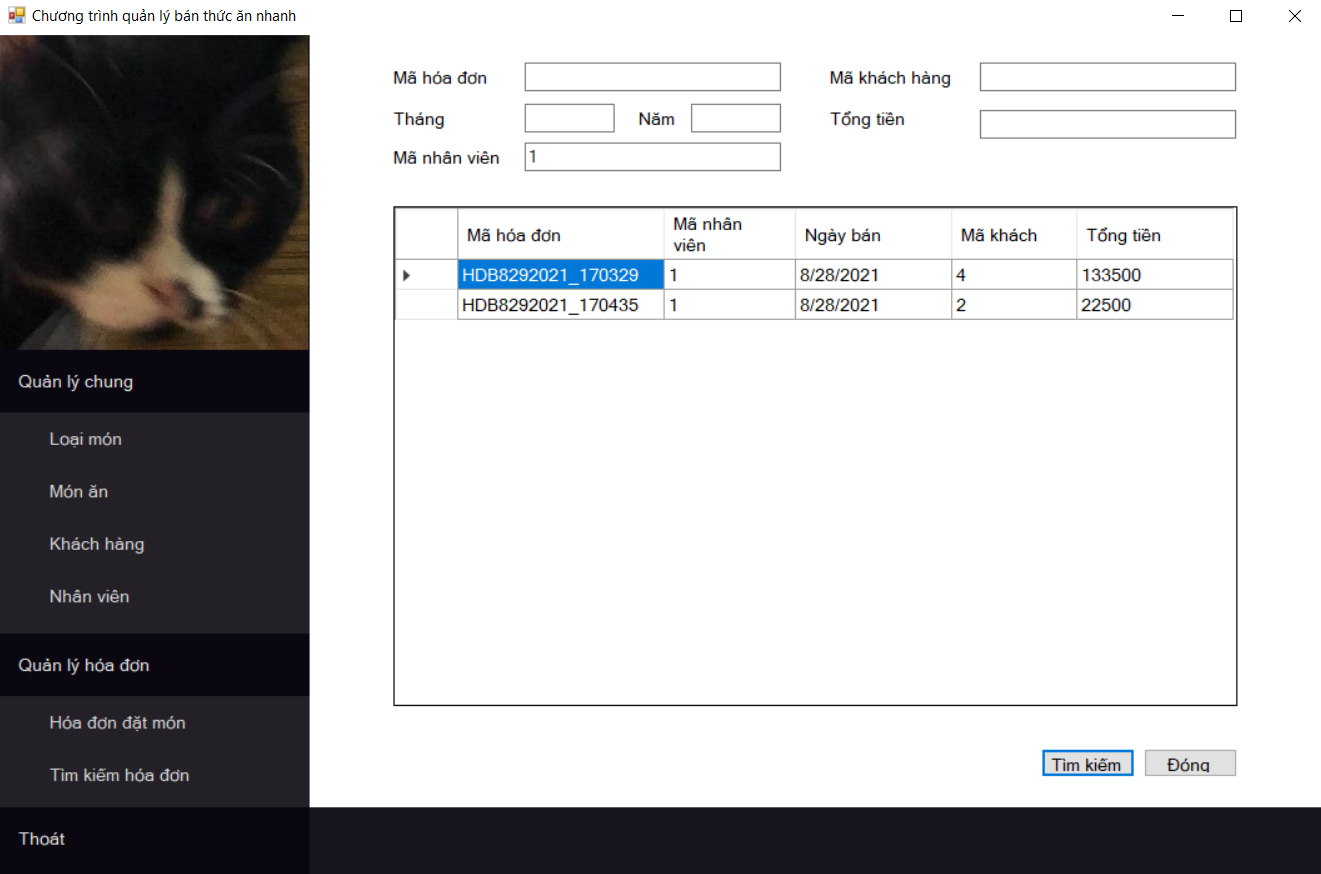


Hình Hóa đơn được in vào Excel

* Chức năng tìm kiếm hóa đơn:



Hình Tìm kiếm hóa đơn theo mã nhân viên là 1



Hình Tìm thấy 2 hóa đơn

# CHƯƠNG V: KẾT LUẬN

## **1/. Kết luận.**

### 1.1/. Ưu điểm của đồ án

* Hoàn thiện yêu cầu căn bản của đồ án thêm, xóa, sửa, ….
* Đóng gói dữ liệu để bảo mật thông tin.
* Hiển thị dữ liệu từ database lên DataGridView để người dùng tương tác.
* Sử dụng những lớp tham chiếu dùng chung cho các Form giúp tối ưu ứng dụng.

### 1.2/. Hạn chế của đồ án

* Chưa làm được mô hình 3 lớp.
* Ứng dụng còn nhiều thiếu sót.
* Chưa thể hiện phần báo cáo doanh thu.

## **2/. Hướng phát triển**

Nếu có cơ hội em sẽ phát triển hệ thống quản lý tốt và chặt chẽ hơn, hoàn thiện tất cả các công cụ cơ bản và thêm những công cụ quan trọng, nâng cấp giao diện, logic của ứng dụng và bảo mật cho người sử dụng giúp cho ứng dụng hoản chỉnh và có tính ứng dụng cao.