**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN LẬP TRÌNH**

**Đề tài:**

**ỨNG DỤNG QUẢN LÝ NHÀ HÀNG**

**SINH VIÊN THỰC HIỆN:**

**Trần Ngọc Huy**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Nguyễn Lê Tuấn Cương**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Hoàng Nguyên Vũ**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Đà Nẵng 12-2019**

LỜI MỞ ĐẦU

1. **Lý do chọn đề tài:**

Công việc quản lý hệ thống là một công việc đòi hỏi độ chính xác cao, tối ưa và rõ ràng. Khi mà công nghệ chưa phát triển thì việc quản lý còn phải được thực hiện một cách thủ công và phải làm trên giấy, gây bất tiện mỗi khi cập nhật, thêm hay sửa xóa dữ liệu dẫn đến nhiều vấn đề tồn đọng trong khâu quản lý. Cho đến nay, khi mà công nghệ đã phát triển thì hàng loạt giải pháp tối ưu cho việc quản lý hệ thống đã ra đời, từ việc quản lý dữ liệu trên giấy tờ thì nay đã chuyển qua quản lý trên máy tính, thông qua các phần mềm, các dich vụ lưu trữ. Để tiếp cận với giải pháp trên, trong phạm vi trường học và phạm vi của đồ án, chúng em tiến hành xây dựng một phần mềm Quản lý nhà hàng nhằm tìm hiểu cách thức hoạt động của một hệ thống quản lý thông qua phần mềm trên máy tính, tìm hiểu quá trình kết nói cấu trúc thuật toán với cơ sở dữ liệu, qua đó xây dựng các tính năng mà một hệ thống quản lý cần có.

1. **Mục tiêu:**

* Mục tiêu chính của đồ án:

Hiểu và vận dụng được các kiến thức của các môn Kỹ thuật lập trình, Lập trình hướng đối tượng, cơ sở dữ liệu… Đồng thời nâng cao khả năng tự tìm tòi các kiến thức liên quan đến cấu trúc giải thuật và khả năng làm việc nhóm.

**MỤC LỤC**

[1](#_Toc28790484)

[LỜI MỞ ĐẦU 2](#_Toc28790485)

[DANH MỤC HÌNH VẼ 5](#_Toc28790486)

[1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc28790487)

[2. CƠ SỞ DỮ LIỆU 6](#_Toc28790488)

[2.1. Thiết kế cơ sở dữ liệu 6](#_Toc28790489)

[2.2. Trigger 7](#_Toc28790490)

[2.2.1. BILL – Insert and Update 7](#_Toc28790491)

[2.2.2. ADDFOOD – Insert 9](#_Toc28790492)

[3. CẤU TRÚC DỮ LIỆU 10](#_Toc28790493)

[3.1. Lớp Node 10](#_Toc28790494)

[3.2. Lớp List 11](#_Toc28790495)

[4. NGHIỆP VỤ CHUNG VÀ CÁC CHỨC NĂNG 13](#_Toc28790496)

[4.1. Nghiệp vụ chung 13](#_Toc28790497)

[4.2. Chức năng 13](#_Toc28790498)

[3.2.1. Quản lý Món (Menu) 13](#_Toc28790499)

[3.2.2. Quản lý Bàn (Desk) 13](#_Toc28790500)

[3.2.3. Quản lý Hóa đơn (Bill) 14](#_Toc28790501)

[3.2.4. Quản lý Gọi món (AddFood) 14](#_Toc28790502)

[4. TỔ CHỨC MÃ NGUỒN CỦA DỰ ÁN 15](#_Toc28790503)

[5.1. Khối Tables 15](#_Toc28790504)

[5.2. Khối List 15](#_Toc28790505)

[5.3. Khối Views 15](#_Toc28790506)

[5.4. Khối Models 17](#_Toc28790507)

[5.5. Khối Controllers 17](#_Toc28790508)

[6. THỰC THI CHƯƠNG TRÌNH 17](#_Toc28790509)

[Home Control 17](#_Toc28790510)

[6.1. Menu 17](#_Toc28790511)

[6.1.1. Menu Show 18](#_Toc28790512)

[6.1.2. Menu Add 18](#_Toc28790513)

[6.1.3. Menu Remove 18](#_Toc28790514)

[6.2. Desk 19](#_Toc28790515)

[6.3. AddFood 19](#_Toc28790516)

[6.4. Bill 20](#_Toc28790517)

[6.4.1. Bill Add 20](#_Toc28790518)

[6.4.2. Bill Payment 21](#_Toc28790519)

[7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22](#_Toc28790520)

[7.1. Kết luận 22](#_Toc28790521)

[7.2. Hướng phát triển 22](#_Toc28790522)

[TÀI LIỆU THAM KHẢO 23](#_Toc28790523)

DANH MỤC HÌNH VẼ

[Hình 1](#_Toc27732586)

[Hình 2](#_Toc27732587)

[Hình 3](#_Toc27732588)

[Hình 4](#_Toc27732589)

[Hình 5](#_Toc27732590)

[Hình 6](#_Toc27732591)

[Hình 7](#_Toc27732592)

# GIỚI THIỆU ĐỀ TÀI

**Đề tài: Ứng dụng quản lý nhà hàng**

- Trong thực tế, khi mở ra một dịch vụ kinh doanh thì việc cấp thiết nhất là phải đề ra được một phương án quản lý dữ liệu hiệu quả, tối ưu và đem lại trực quan cho người dùng. Cụ thể ở đây là ứng dụng quản lý nhà hàng, nhóm chúng em đã vận dụng cá kiến thức có được ở lớp học và tìm hiểu thêm, đã hình thành được ứng dụng. Dù hiện tại chỉ đang được thực hiện thông qua giao diện console nhưng tương lai sẽ có thể phát triển cao hơn.

# CƠ SỞ DỮ LIỆU

## Thiết kế cơ sở dữ liệu

Trong đồ án lần này, nhóm sẽ tập trung vào việc quản lí danh sách các bàn ăn, các món ăn khách gọi và hóa đơn của mỗi bàn. Từ các thuộc tính chung thực tế, chúng ta phân tích được thành 3 table chính đó là: thực đơn (menu), danh sách các bàn ăn (table) và hóa đơn (bill).

Trong thực đơn gồm có các thuộc tính:

* Mã món (Khóa chính)
* Tên món
* Giá thành

Trong bàn ăn có các thuộc tính:

* Mã bàn (Khóa chính)
* Trạng thái (Còn trống/đã có người ngồi)

Trong hóa đơn gồm có:

* Mã hóa đơn (Khóa chính)
* Mã bàn (Khóa ngoại)
* Trạng thái thanh toán
* Tổng giá

Quan hệ giữa các table trong cơ sở dữ liệu như sau:

* Bàn – Hóa đơn: quan hệ 1-1
* Hóa đơn – Menu: quan hệ n-n

Vì trong cơ sở dữ liệu không có quan hệ n-n (nhiều – nhiều), nhận thấy điều đó nhóm đã thiết kế thêm 1 bảng phụ đó là bảng “gọi món” (addfood) để liên kết 2 bảng hóa đơn và thực đơn lại với nhau. Lúc này ta có các quan hệ:

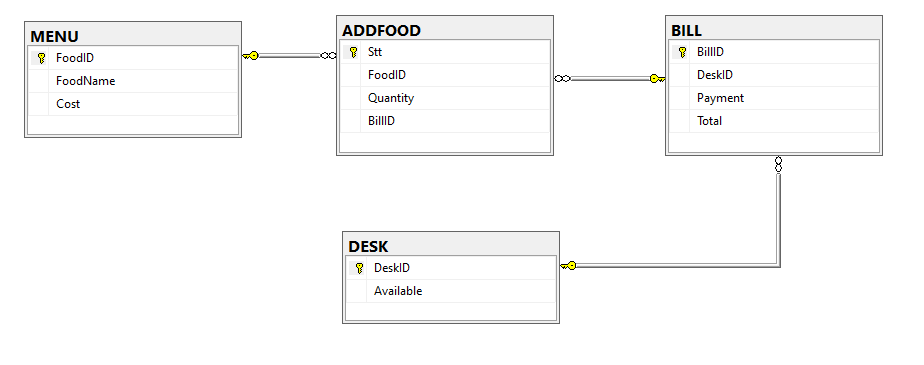
* Hóa đơn – gọi món: 1-n
* Thực đơn – gọi món: 1-n

→ Từ đó sẽ tạo ra quan hệ nhiều-nhiều cho hóa đơn và thực đơn

Lúc này, ta sẽ có được bảng ADDFOOD với các thuộc tính:

* Số thứ tự (Khóa chính)
* Mã hóa đơn (Khóa ngoại)
* Mã món (Khóa ngoại)
* Số lượng gọi

→ Từ đó các mối quan hệ giữa các bảng được tối ưu. Ta có sơ đồ quan hệ giữa các bảng với nhau:



*Hình 1: Database diagram*

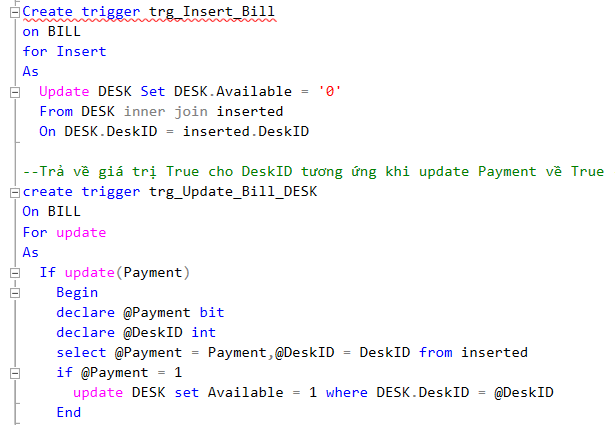
## Trigger

Một trigger là một đối tượng gắn liền với một bảng và được tự động kích hoạt khi xảy ra những giao tác làm thay đỏi dữ liệu trong bảng. Ta sử dụng **trigger** trong cơ sở dữ liệu để đảm bảo tính toàn vẹn dữ liệu trong cơ sở dữ liệu. Khi xảy ra những giao tác làm tha đổi dữ liệu trong các bảng, các **trigger** sẽ được gọi tự động. Ta sẽ áp dụng **trigger** cho 2 bảng đó là BILL và ADDFOOD.

### BILL – Insert and Update

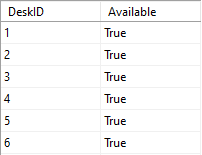
Trigger được tạo ở bảng BILL nhằm quản lí trạng thái của các danh sách bàn. Nghĩa là khi ta thêm 1 mã bàn vào 1 hóa đơn, thì bàn có mã bàn tương ứng ở bảng DESK sẽ chuyển từ trạng thái trống (True) sang trạng thái đã có người ngồi (False). Và ngược lại, khi hóa đơn đã thanh toán thì sẽ trả lại mã bàn tương ứng ở bảng DESK giá trị False về lại giá trị True.

Như vậy, khi có khách hàng, việc điều phối chỗ ngồi sẽ đơn giản hơn và sẽ không bị trùng lặp dữ liệu giữa các record với nhau.



*Hình 2: Mô tả trigger*

Ví dụ, lúc đầu ta có danh sách các bàn đang trống (như hình vẽ):



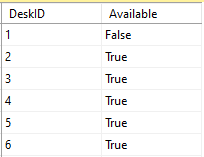
*Hình 3: Danh sách các bàn trống*

Lúc này, có khách vào ngồi tại bàn số một, ta sẽ thêm một record vào bảng BILL với mã bàn tương ứng là 1.



*Hình 4: Sau khi thêm 1 record vào BILL*

Ở bảng BILL lúc này sẽ có 1 hóa đơn với BillID = 1000, DeskID = 1, chưa thanh toán và chưa gọi món ở ADDFOOD nên tổng giá vẫn bằng 0. Và với bàn có DeskID = 1 được khóa ngoại của bảng BILL tham chiếu đến bảng DESK sẽ chuyển về trạng thái False (đã có người ngồi).

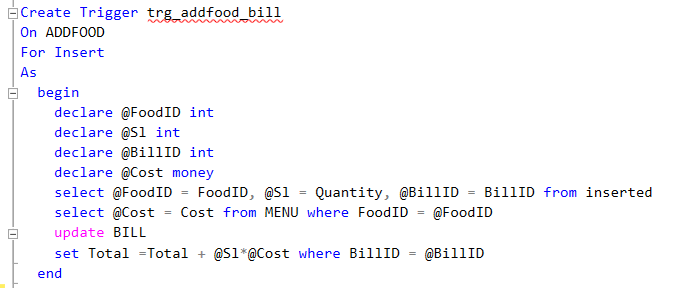


*Hình 5: DESK sau khi được trigger kích hoạt*

Và khi hóa đơn đã thanh toán xong, thì bàn 1 sẽ lại được trả về giá trị True (còn trống).

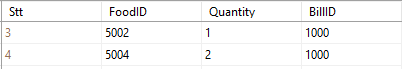
### ADDFOOD – Insert

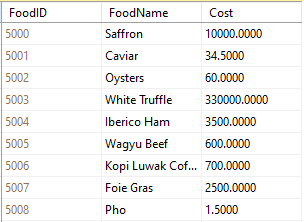
Một bảng nữa mà khi thay đổi dữ liệu trên bảng thì ảnh hưởng tới dữ liệu của bảng khác đó là bảng ADDFOOD. Trigger được áp dụng cho bảng này nhằm tính tổng giá tiền (Total). Khi order của khách hàng được nhập vào ADDFOOD thì bên table BILL sẽ tự động cộng thêm số tiền vào Total dựa trên số lượng món đã gọi và giá tiền của từng món đã có ở bảng Menu.



*Hình 6: Trigger ở bảng ADDFOOD*

Lấy ví dụ với hóa đơn 1000 ở mục trước, lúc này khách hàng có yêu cầu các món như sau:

  
*Hình 7: Dữ liệu khi thêm vào bảng ADDFOOD*

  
*Hinh 8: Các món có ở Menu*

Ta đã thêm các món ăn mà khách gọi vào bảng ADDFOOD với mã hóa đơn tương ứng của khách. Vã hay xem sự thay đổi ở bảng BILL với hóa đơn có BillID = 1000:

  
*Hình 9: Tổng số tiền ứng với các món và số lượng đã gọi*

Với mã món đã gọi là 5002, số lượng 1, giá là 60$ và mã món 5004, số lượng 2, giá là 3500$ thì ta có tổng giá tiền của hóa đơn 1000 là 7060$.

# CẤU TRÚC DỮ LIỆU

## Lớp Node

Là một lớp template cho kiểu **Data**. Đóng vai trò là một Nút của Danh sách liên kết.

Thuộc tính

**Dữ liệu**

Kiểu Data.

Dùng để lưu trữ dữ liệu.

**Node tiếp theo**

Con trỏ kiểu Node.

Chứa địa chỉ của Node tiếp theo.

Thao tác

**Khởi tạo** với tham số truyền vào kiểu **Data.**

Gán **Dữ liệu** bằng tham số truyền vào.

Gán **Node tiếp theo** bằng rỗng.

**Lấy dữ liệu**

Trả về tham trị kiểu **Data** lấy từ **Dữ liệu**.

## Lớp List

Là một lớp template cho kiểu **Data**. Đóng vai trò liên kết các Nút để tạo thành Danh sách liên kết.

Thuộc tính

**Số phần tử**

Kiểu int. Chỉ số phần tử của danh sách liên kết.

**Nút bắt đầu**

Con trỏ kiểu Node. Là địa chỉ của Node đầu danh sách.

Thao tác

**Hàm dựng**

Gán **Nút bắt đầu** bằng rỗng.

**Hàm hủy**

Nếu thao tác **Kiểm tra rỗng** trả vể sai. Thì thực hiện tiếp.

Khai báo **Nút hiện tại** bằng **Nút bắt đầu**.

Khai báo **Tạm**

Trong khi **Nút hiện tại** khác rỗng thì gán **Tạm** bằng **Nút hiện tại**. Gán **Nút hiện tại** bằng **Node tiếp theo** của nó. Xóa **Tạm**.

**Chèn** với dữ liệu truyền vào kiểu Data

Kiểm tra danh sách có rỗng không.

Nếu rỗng **Khởi tạo** một Node từ dữ liệu truyền vào.

Gán **Nút bắt đầu** bằng Nút vừa tạo.

Tăng **Số phần tử** lên một đơn vị.

Nếu không rỗng thì **Khởi tạo** một Node từ dữ liệu truyền vào.

Gán **Node tiếp theo** của Node vừa tạo bằng **Nút bắt đầu**.

Gán **Nút bắt đầu** bằng Node vừa tạo.

Tăng **Số phần tử** lên một đơn vị.

**Cập nhật** với dữ liệu truyền vào kiểu Data

Khai báo con trỏ **Tạm** bằng **Nút bắt đầu**.

Trong khi dữ liệu của con trỏ **Tạm** khác với dữ liệu của **Nút bắt đầu**. Lặp lệnh Gán Tạm bằng Node tiếp theo của nó.

Sau khi lặp nếu **Tạm** khác rỗng thì cập nhật giá trị của **Tạm** bằng dữ liệu truyền vào.

**Xóa**

Khai báo con trỏ **Tạm** bằng **Nút bắt đầu**.

Nếu dữ liệu truyền vào bằng với **Nút bắt đầu**.

Gán **Nút bắt đầu** bằng Node tiếp theo của **Nút bắt đầu**.

Xóa nút vừa bị hủy thông qua con trỏ **Tạm**.

Con không thì lặp cho đến khi **Node tiếp theo** của con trỏ **Tạm** là rỗng, hoặc **Node tiếp theo** của con trỏ **Tạm** bằng với dữ liệu truyền vào.

Nếu **Node tiếp theo** của con trỏ **Tạm** khác rỗng. Thì Node cần gỡ là **Node tiếp theo** của con trỏ **Tạm**.

Gán con trỏ **Old** bằng **Node tiếp theo** của con trỏ **Tạm**.

Gán **Node tiếp theo** của con trỏ **Tạm** bằng **Node tiếp theo** của **Node tiếp theo**. Xóa Node thông qua một biến Old.

**Xuất**

Nếu thao tác **Kiểm tra rỗng** trả vể sai. Thì thực hiện tiếp.

Khai báo **Tạm** bằng **Nút bắt đầu**.

Trong khi **Tạm** khác rỗng thì xuất giá trị của **Tạm** và gán **Tạm** bằng **Node tiếp theo** của nó.

**Làm rỗng**

Gọi **Hàm hủy**.

Gán **Nút bắt đầu** bằng rỗng.

Gán **số phần tử** bằng 0.

**Truy xuất bằng toán tử []** với giá trị truyền vào là i

Khai báo **Tạm** bằng **Nút bắt đầu**.

Trong khi i khác 0 và **Tạm** khác rỗng.

Gán **Tạm** bằng **Node tiếp theo** của nó và giảm i một đơn vị.

Nếu **Tạm** khác rỗng thì trả về giá trị của **Tạm**.

Còn không thì trả về giá trị của **Node bắt đầu**.

**Kiểm tra rỗng**

Trả về đúng nếu **Nút bắt đầu** có giá trị rỗng.

# NGHIỆP VỤ CHUNG VÀ CÁC CHỨC NĂNG

## Nghiệp vụ chung

- Quản lý các đối tượng cơ bản của nhà hàng để thiết lập hóa đơn và thanh toán trên cửa sổ console.

- Lưu dữ liệu vào hệ thống cơ sở dữ liệu của SQL Server.

## Chức năng

### 3.2.1. Quản lý Món (Menu)

- Thuộc tính:

**IdMón**(FoodId)

Kiểu số nguyên.

**Tên** (Name)

Chỉ gồm các kí tự có trong bản chữ cái và các chữ số.

**Giá** (Cost)

Kiểu số thực. Đơn vị tính bằng USD.

- Tính năng

Thêm

Nhập **Tên** và **Giá**để thêm.

Hiển thị

Hiển thị tất cả các món có trong Menu.

Xóa

Nhập **IdMón** để xóa.

### 3.2.2. Quản lý Bàn (Desk)

- Thuộc tính:

**IdBàn**(DeskId)

Kiểu số nguyên.

**Khả dụng** (Available)

Kiểu đúng/sai. Chỉ trạng thái khả dụng của bàn.

- Tính năng

Thêm

Nhập **Id Bàn**để thêm.

Bàn sau khi thêm thì thuộc tính Khả dụng mặc định là đúng.

Hiển thị

Hiển thị tất cả các bàn.

### 3.2.3. Quản lý Hóa đơn (Bill)

- Thuộc tính:

**IdHóa đơn**(BillId)

Kiểu số nguyên.

**Id Bàn** (DeskId)

Kiểu số nguyên và là giá trị của một Bàn tồn tại.

**Đã thanh toán** (Payment)

Kiểu đúng/sai. Hiển thị trạng thái thanh toán của bàn.

**Tổng** (Total)

Kiểu số thực. Là tổng thanh toán của Hóa đơn.

- Tính năng

**Thêm**

* + - Hiển thị các Bàn còn trống.
    - Yêu cầu người dùng nhập Id Bàn cần tạo hóa đơn.
    - Hóa đơn sau khi tạo thì thuộc tính Đã thanh toán mặc định là sai.

**Cập nhật Trạng thái thanh toán**

* + - Hiển thị thông tin các Hóa đơn chưa thanh toán.
    - Yêu cầu người dùng nhập Id Hóa đơn để xem thông tin chi tiết.
    - Hiển thị chi tiết hóa đơn.
    - Nhập Trạng thái thanh toán để cập nhật.
    - Sau khi thanh toán. Thuộc tính Khả dụng của Bàn tương ứng sẽ đặt về đúng.

### 3.2.4. Quản lý Gọi món (AddFood)

- Thuộc tính:

**Id Gọi Món** (AddFoodId)

Kiểu số nguyên

**Id Món** (FoodId)

Kiểu số nguyên.

**Id Hóa đơn** (BillId)

Kiểu số nguyên.

- Tính năng

**Thêm**

Hiển thị danh sách Hóa đơn chưa thanh toán.

* + - * Yêu cầu người dùng nhập Id Hóa đơn.
      * Hiển thị danh sách Món.
      * Yêu cầu người dùng nhập Id Món.
      * Yêu cầu người dùng nhập Số lượng.
      * Sau khi thêm Món. Tổng của Hóa đơn được tính lại.

# 4. TỔ CHỨC MÃ NGUỒN CỦA DỰ ÁN

## Khối Tables

Chứa các Class trong đó:

* Có thuộc tính giống với các bảng của Co sở dữ liệu.
* Hàm dựng để khởi tạo các giá trị mặc định.
* Hàm nhập để chỉ nhập những thuộc tính cần thiết.
* Hàm xuất để chuẩn hóa khoảng cách các thuộc tính được in ra.
* Hàm head để hiển thị Tiêu đề của bảng.

## Khối List

Chứa lớp List là một lớp template. Lớp List là danh sách liên kết đơn dùng để chứa các Node là các Class trong khói Tables, lớp List gồm những thao tác:

* Hàm dựng để khởi tạo danh sách trống.
* Nhận một Node truyền vào, tạo một Node mới đầu danh sách và sao chép giá trị vào.
* Nhận một Node truyền vào, tìm kiếm và cập nhật giá trị cho Node đó.
* Nhận một Node truyền vào, tìm kiếm và xóa Node đó.
* Xuất tất cả các Node của danh sách.
* Làm rỗng danh sách.
* Xem danh sách có rỗng hay không.

## Khối Views

Chịu trách nhiệm xuất ra Console và tương tác với người dùng. Nhận giá trị người dùng nhập vào và trả về cho tên hàm. Gồm các hàm với chức năng như sau:

Trang chủ

Chờ người dùng nhập một số để điều hướng tới phần quản lí các đối tượng: **Món, Bàn, Hóa Đơn, Gọi Món.**

**Món**

Chờ người dùng nhập số để điều hướng tới chức năng Thêm, Hiển Thị hoặc Xóa

Thêm Món

Nhận giá trị người dùng nhập vào và trả cho tên hàm.

Hiển thị Món

Hiển thị tất cả các món

Xóa Món

Nhận giá trị người dùng nhậpv à trả cho tên hàm.

**Bàn**

Chờ người dùng nhập số để điều hướng tới các chức năng Thêm hoặc Hiển thị.

Thêm Bàn

Nhận giá trị người dùng nhập và trả cho tên hàm.

Hiển thị bàn

Hiển thị tất cả các bàn.

**Hóa đơn**

Chờ người dùng nhấp số để điều hướng tới các chức năng Thêm hoặc Cập nhật trạng thái thanh toán.

Thêm Hóa đơn

Nhận giá trị người dùng nhập và trả cho tên hàm.

Cập nhật trạng thái hóa đơn

Hiển thị tất cả danh sách hóa đơn. Nhận giá trị người dùng nhập và trả cho tên hàm.

Chi tiết hóa đơn

Hiển thị tất cả danh sách món của bàn đó.Nhận giá trị người dùng nhập và trả cho tên hàm.

**Gọi món**

Chờ người dùng nhập số để điều hướng tới chức năng Thêm Gọi Món

Thêm Gọi món

Nhận giá trị người dùng nhập và trả cho tên hàm.

## Khối Models

Xử lý dữ liệu, đồng bộ giữa Cơ sở dữ liệu của SQK Server và Danh sách liên kết của khối List. Gồm các chức năng:

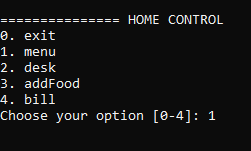
* Đồng bộ dữ liệu từ Cơ sở dữ liệu của SQL Server về Danh sách liên kết.
* Thêm, sửa, xóa dữ liệu trong Danh sách liên kết và sau đó đồng bộ Danh sách liên kết về Cơ sở dữ liệu của SQL Server.

## Khối Controllers

Tạo mạch liên kết qua lại giữa các khối Views. Đóng gói dữ liệu được trả về từ khối Views, sau đó gọi khối Models để xử lý. Chứa các hàm có tên giống với các hàm của khối Views. Những xử lý nhỏ có thể xử lý trực tiếp tại Hàm mà không cần gọi khối Models.

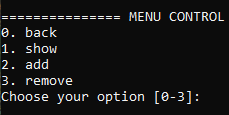
# THỰC THI CHƯƠNG TRÌNH

## Home Control

Có 4 lựa chọn cho Home Control: menu, desk, addFood và bill.

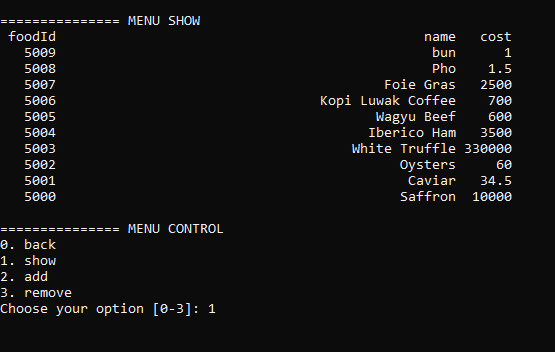
*Hình 10: Home Controller*

## Menu

- Giao diện của Menu Control gòm 3 lựa chọn thao tác: hiện ra danh sách món ăn, thêm món ăn và xóa món ăn khỏi danh sách.

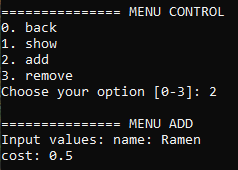
*Hình 11: Menu Control*

### Menu Show

- Hiện ra danh sách các món ăn có trong dữ liệu.

*Hình 12: Menu Show*

### Menu Add

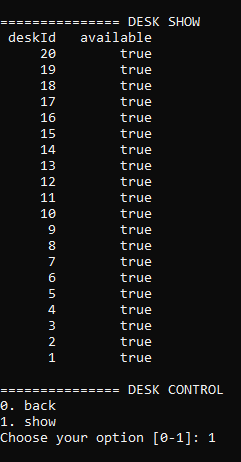
- Thêm một món ăn vào danh sách dữ liệu. Tại đây, người dùng sẽ nhập tên món và giá tiền để tiến hành thêm vào dữ liệu.

*Hình 13: Home Controller*

### Menu Remove

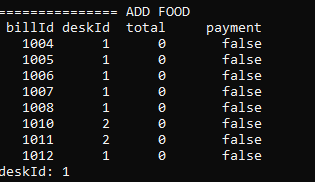
- Muốn xóa món ăn nào trong menu thì người dùng chỉ việc nhập vào foodId cua món đó, hệ thống sẽ tiến hành xóa dữ liệu tương ứng.

## Desk

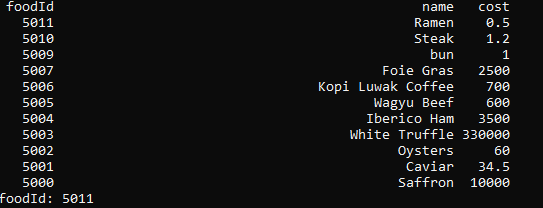
- Hiển thị danh sách bàn còn trống.

*Hình 14: Desk Show*

## AddFood

- Đây là chức năng khi có khách gọi món, người dùng sẽ tiến hành add món ăn vào bàn cần đặt.

*Hình 15: Add Food*

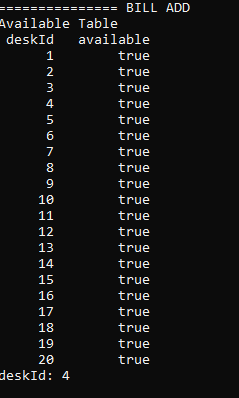


*Hình 16: Menu Food*

## Bill

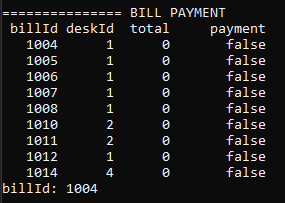
### Bill Add

- Hiển thị danh sách bàn chưa đặt, nhập deskId để add vào bàn cần đặt chỗ.



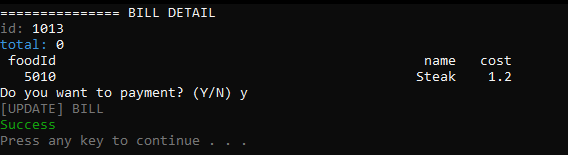
*Hình 17: Bill Add*

### Bill Payment

- Hiển thị danh sách bill cần thanh toán.

*Hình 18: Bill Payment*

- Sau khi nhập billId sẽ xuất hiện bảng billDetail hiện thông tin chi tiết của hóa đơn và hỏi xem người dùng có muốn tiến hành thanh toán hay không.



*Hình 19: Bill Detail*

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

* Nắm được phương thức quản lý một hệ thống cơ sở dữ liệu.
* Phát triển được các chức năng của một hệ thống quản lý nhà hàng: thêm, sửa, xóa, cập nhật…
* Tìm hiểu được cách kết nối code với cơ sở dữ liệu để đồng bộ.
* Nắm được lý thuyết và cách cài đặt danh sách liên kết đơn sử dụng ngôn ngữ lập trình C++.

## Hướng phát triển

* Có thể phát triển chạy ứng dụng trên màn hình có đồ họa thay vì chỉ chạy trên màn hình console.
* Tối ưu hóa hiển thị hơn bằng việc bỏ bớt một số thủ tục không cần thiết.
* Sử dụng thêm các cấu trúc dữ liệu để tối ưu chương trình.

TÀI LIỆU THAM KHẢO

[1] GV. Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, Đà Nẵng, 09/2002

[2] GV. Trần Nguyên Phong, Giáo trình SQL, Huế, 2004

[4] List (abstract data type), https://en.wikipedia.org/wiki/List\_(abstract\_data\_type)