**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**BÁO CÁO ĐỒ ÁN 1**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ**

**QUÁN ĂN TỰ CHỌN**

**SVTH 1: Đoàn Văn Long**  **17110174**

**SVTH 2: Trần Nguyên Tài 17110217**

**SVTH 3: Ngô Hoàng Minh Tâm 17110218**

**GVHD: Huỳnh Xuân Phụng**

**Tp. Hồ Chí Minh, tháng 12 năm 2019**

MỤC LỤC

[DANH MỤC CÁC HÌNH 2](#_Toc27238543)

[DANH MỤC CÁC BẢNG 3](#_Toc27238544)

[PHẦN NỘI DUNG 4](#_Toc27238545)

[I. Mô tả đồ án 4](#_Toc27238546)

[1. Lý do chọn đề tài phần mềm quản lý quán ăn tự chọn 4](#_Toc27238547)

[2. Dữ liệu, thông tin đầu vào: 5](#_Toc27238548)

[3. Mục đích và tính năng: 5](#_Toc27238549)

[4. Giao diện dự kiến: 6](#_Toc27238550)

[II. PHÂN CÔNG CÔNG VIỆC: 7](#_Toc27238551)

[III. THIẾT KẾ: 9](#_Toc27238552)

[1. Thiết kế lớp: 9](#_Toc27238553)

[2. Thiết kế cơ sở dữ liệu: 34](#_Toc27238554)

[3. Thiết kế giao diện: 38](#_Toc27238555)

[IV. PHẦN KẾT LUẬN 42](#_Toc27238556)

[1. Ưu điểm, nhược điểm: 42](#_Toc27238557)

[2. Thảo luận 42](#_Toc27238558)

[3. Hướng phát triển 43](#_Toc27238559)

[TÀI LIỆU THAM KHẢO 45](#_Toc27238560)

# DANH MỤC CÁC HÌNH

[Hình 1: Giao diện Đăng nhập 6](#_Toc26289258)

[Hình 2: Giao diện Menu chính 7](#_Toc26289259)

[Hình 3: Cơ sở dữ liệu của hệ thống 34](#_Toc26289279)

[Hình 4.1: Giao diện chính 38](#_Toc26289289)

[Hình 4.2: Giao diện đăng nhập 39](#_Toc26289290)

[Hình 4.3: Giao diện thông tin Khách hàng 39](#_Toc26289291)

[Hình 4.4: Giao diện thông tin Nhân viên 40](#_Toc26289292)

[Hình 4.5: Giao diện Bảng chọn món 40](#_Toc26289293)

[Hình 4.6: Giao diện thông tin Đăng nhập 40](#_Toc26289294)

[Hình 4.7: Giao diện thông tin Món ăn 41](#_Toc26289295)

[Hình 4.8: Giao diện xuất phiếu hóa đơn 41](#_Toc26289296)

[Hình 4.9: Giao diện thống kê doanh thu 42](#_Toc26289297)

# DANH MỤC CÁC BẢNG

[Bảng 1: Bảng phân công công việc và phần trăm đạt được 7](#_Toc27044703)

[Bảng 2: Danh mục các lớp chính được sử dụng trong chương trình 9](#_Toc27044704)

[Bảng 3: Danh mục các phương thức trong một lớp 11](#_Toc27044705)

[Bảng 4: Bảng mô tả các Table trong Cơ sở dữ liệu 35](#_Toc27044707)

[Bảng 5: Bảng mô tả các Field trong Table: 35](#_Toc27044708)

[Bảng 5.1: Table Account (người dùng ở đây có nghĩa là Nhân Viên) 35](#_Toc27044709)

[Bảng 5.2: Table Bill 36](#_Toc27044710)

[Bảng 5.3: Table Category 36](#_Toc27044711)

[Bảng 5.4: Table Customer 36](#_Toc27044712)

[Bảng 5.5: Table Employee 37](#_Toc27044713)

[Bảng 5.6: Table Food 37](#_Toc27044714)

[Bảng 5.7: Table Detail 38](#_Toc27044715)

[Bảng 6: Thiết kế giao diện 38](#_Toc27044716)

# PHẦN NỘI DUNG

## I. Mô tả đồ án

### 1. Lý do chọn đề tài phần mềm quản lý quán ăn tự chọn

Trong thời đại này nay, công nghệ thông tin đang được phát triển mạnh mẽ, nhanh chóng và xâm nhập vào nhiều lĩnh vực khoa học, kỹ thuật cũng như trong cuộc sống. Nó đóng vai trò hết sức quan trọng , có thể tạo ra những bước đột phá lớn. Việc xây dựng các phần mềm quản lý để hỗ trợ cho công việc kinh doanh là một yêu cầu tất yếu. Việc quản lý một quán ăn, nhà hàng sẽ trở nên dễ dàng và trôi chảy hơn nếu có sự hỗ trợ của một phần mềm quản lý. Đó cũng chính là lý do mà nhóm chúng em quyết định thực hiện đề tài “Quản lý quán ăn tự chọn” bằng phương pháp lập trình hướng đối tượng.

Phần mềm ứng dụng công nghệ hiện đại, người quản lý có thể theo dõi mọi hoạt động kinh doanh tại quán ăn, nhà hàng vào bất cứ thời điểm nào mà không cần phải có mặt tại quán thông qua phần mềm. Bên cạnh đó, với giạo diện thân thiện với người dùng, điều này sẽ giúp đỡ cho nhiều khách hàng dù chưa dùng quen vẫn sẽ dễ tiếp cập và sử dụng phần mềm.

### 2. Dữ liệu, thông tin đầu vào:

- Xây dựng giả định một cơ sở dữ liệu gồm 4 accout Nhân Viên của hệ thống, 20 Khách hàng, 26 món ăn được phân ra các dạng thức ăn như: Thức ăn khô, thức ăn nước, đồ uống

### **3. Mục đích và tính năng:**

* Tiết kiệm thời gian cho khách hàng, đóng vai trò như một nhân viên bồi bàn:

+ Phần mềm quản lý quán ăn tự chọn giúp khách hàng có thể thực hiện order điện tử ngay trên máy tính bảng, ứng dụng điện thoại.

+ Thông tin gọi món ngay lập tức được chuyển tới Bếp và Thu ngân mà không cần phải di chuyển.

+ Nhờ thế, có thể hạn chế công việc cho nhân viên trong giờ cao điểm. Khách hàng không cần thiết phải chờ đợi phục vụ bàn để gọi món.

* In hóa đơn tự động nhanh chóng sau khi chọn món thành công:

+ Hỗ trợ in hóa đơn ngay tại quầy bếp, danh sách các món ăn đều được sắp xếp theo thứ tự trước sau giúp bộ phận bếp có thể chủ động điều phối chế biến hợp lý, nhanh chóng.

* Phục vụ khách hàng tức thì thông qua phần mềm:

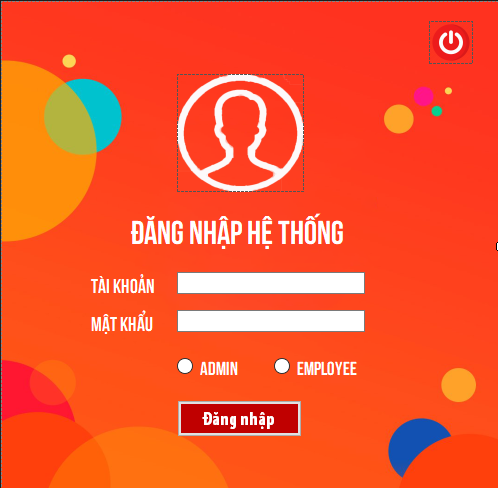
+ Với ưu điểm có thể tích hợp trên thiết bị di động, màn hình có thể hiển thị tất cả món ăn, đồ uống có tại Quán ăn.

* Thanh toán chính xác hạn chế tỷ lệ thất thoát cho thu ngân

+ Tạo hoá đơn trong tích tắc, ghi nhận tiền khách trả và tiền thừa phải trả khách, tránh sai sót trong tính toán.

+ Hệ thống lưu lại lịch sử thao tác, hóa đơn, giúp quản lý dễ dàng rà soát thông tin khi gặp vấn đề.

### 4. Giao diện dự kiến:



##### Hình 1: Giao diện Đăng nhập



##### Hình 2: Giao diện Menu chính

## II. PHÂN CÔNG CÔNG VIỆC:

##### Bảng 1: Bảng phân công công việc và phần trăm đạt được

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Sinh Viên** | **Khát quát công việc** | **Tỷ lệ đóng góp** |
| 1 | Đoàn Văn Long | Thiết kế cơ sở dữ liệu  Các thao tác cơ bản trong Quản  lý Nhân Viên (thêm, xóa, sửa,  cập nhật, tìm kiếm) | 35% |
| Các thao tác cơ bản trong Quản  lý Món Ăn (thêm, xóa, sửa, cập  nhật, tìm kiếm) |
| Các thao tác cơ bản trong Quản  lý Khách Hàng (thêm, xóa, sửa,  cập nhật, tìm kiếm) |
| Các thao tác cơ bản trong Quản  lý Đăng nhập – phân quyền  (thêm, xóa, sửa, cập nhật, tìm  kiếm) |
| 2 | Trần Nguyên Tài | Các thao tác trong việc Lựa  chọn món ăn (lựa chọn món,  xuất ra hóa đơn bán lẻ xem  trước) | 35% |
| Xuất phiếu hóa đơn bán lẻ |
| Form Doanh Thu theo tháng |
| Thiết kế giao diện tự động cho  Bảng chọn món |
| 3 | Ngô Hoàng Minh Tâm | Làm file trình chiếu (ppt) | 30% |
| Thao tác đăng nhập và  chuyển form |
| Bổ sung thông tin dữ liệu |
| Thiết kế giao diện phần mềm |
| Viết báo cáo (doc) |

## III. THIẾT KẾ:

### Thiết kế lớp:

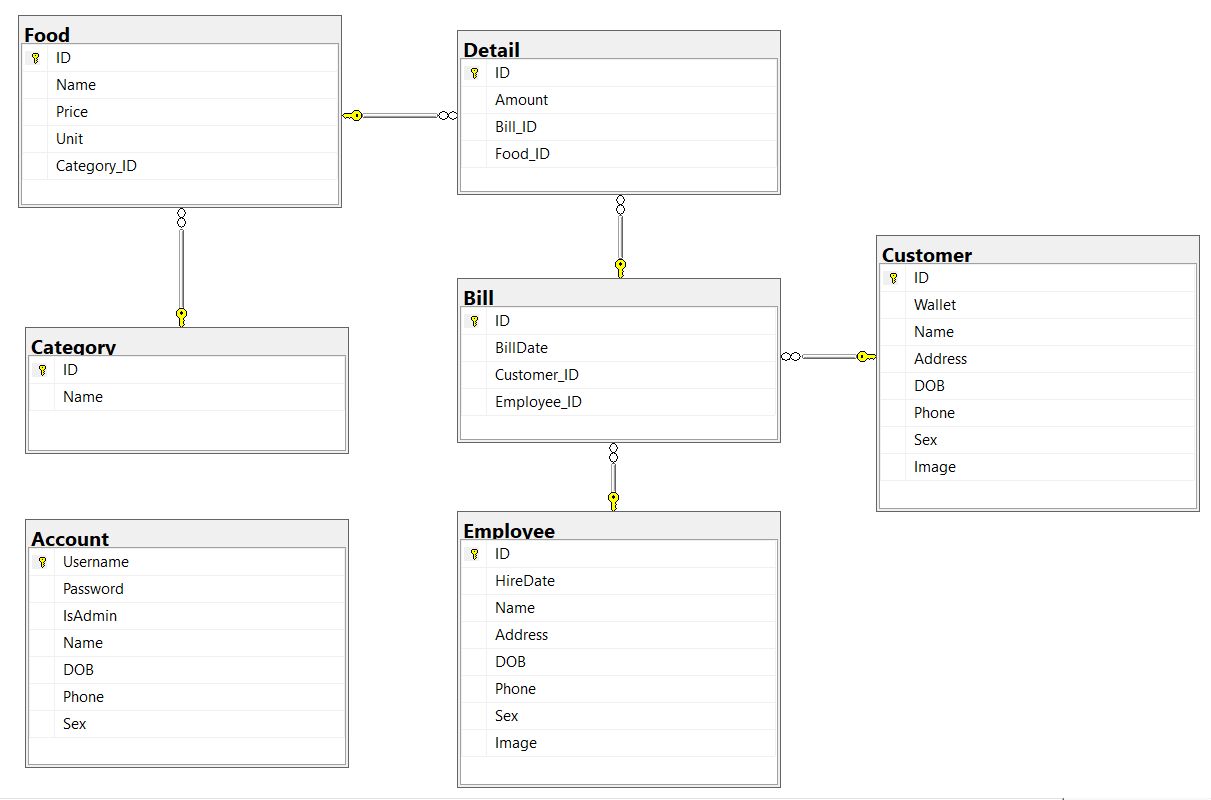
##### Bảng 2: Danh mục các lớp chính được sử dụng trong chương trình

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích chính của lớp trong chương trình** | **SV phụ trách** |
| 1 | FormLogin | Lớp hiển thị giao diện xử lý Đăng nhập | Ngô Hoàng Minh Tâm |
| 2 | Form  FoodChoices | Lớp hiển thị giao diện Chọn món ăn và xuất ra danh mục những món ăn đã chọn, tính toán tổng hóa đơn. | Trần Nguyên Tài |
| 3 | Form  Employee | Lớp hiển thị giao diện Quản lý Nhân viên, gồm tính năng tìm kiếm, thêm, xóa, sửa, cập nhật thông tin. (được liên kết từ Form Manage) | Đoàn Văn Long |
| 4 | FormCustomer | Lớp hiển thị giao diện Quản lý Khách hàng, gồm tính năng tìm kiếm, thêm, xóa, sửa, cập nhật thông tin. (được liên kết từ Form Manage) | Đoàn Văn Long |
| 5 | Form  LoginManage | Lớp hiển thị và xử lí quyền trong bảng DangNhap (được liên kết từ Form Manage) | Đoàn Văn Long |
| 6 | FormManage | Lớp hiển thị giao diện Quản lý dành cho Admin, gồm các menu Quản lý nhân viên, khách hàng, Quản lý doanh thu, Quản lý đăng nhập, món ăn  (chỉ admin mới có thể truy cập) | Ngô Hoàng Minh Tâm |
| 7 | ThongKeDoanhThu | Lớp Thống kê doanh thu của quán ăn (được liên kết từ Form Manage) | Trần Nguyên Tài |
| 8 | ReportBill | Lớp xuất hóa đơn bán lẻ, lưu thông tin hóa đơn tính tổng hóa đơn | Trần Nguyên Tài |
| 9 | FoodManager | Lớp hiển thị giao diện Quản lý Món ăn, gồm tính năng tìm kiếm, thêm, xóa, sửa, cập nhật thông tin. (được liên kết từ Form Manage) | Đoàn Văn Long |
| 10 | Bill | Tạo đối tượng Bill gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Hóa Đơn | Trần Nguyên Tài |
| 11 | Detail | Tạo đối tượng Detail gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Chi tiết Hóa Đơn | Trần Nguyên Tài |
| 12 | Account | Tạo đối tượng Account gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Đăng nhập | Ngô Hoàng Minh Tâm |
| 13 | Employee | Tạo đối tượng Employee gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Nhân viên, là đối tượng con của Person | Đoàn Văn Long |
| 14 | Food | Tạo đối tượng Food gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Món ăn | Đoàn Văn Long |
| 15 | Customer | Tạo đối tượng Customer gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Khách hàng, là đối tượng con của Person | Đoàn Văn Long |
| 16 | Category | Tạo đối tượng Category gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu và các phương thức xử lý thông tin Loại Món ăn | Đoàn Văn Long |
| 17 | Person | Tạo đối tượng Person gồm các thuộc tính truyền giá trị vào cơ sở dữ liệu | Đoàn Văn Long |

##### Bảng 3: Danh mục các phương thức trong một lớp

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Phương thức** | **Mục đích** | **Tên file** |
| 1 | **GetFoodByCategory(int CategoryID)**  Intput: CategoryID  Output: List<dynamic>  Pseudo code:  from t in ctx.Foods join p in ctx.Categories  on t.Category\_ID equals p.ID  where t.Category\_ID == CategoryID  select new {t.ID , t.Name, t.Price, t.Unit, CategoryName=p.Name, t.Image}  Câu lệnh SQL liên kết 2 bảng Food và Category với nhau để lấy được thông tin Tên Loại Món ăn thông qua thuộc tính Mã Loại Món ăn | Hàm lấy thông tin các món ăn dựa trên Mã loại món ăn (CategoryID) | Food.cs  (85) |
| 2 | **GetListCategory():**  Output: List<Category> | Hàm lấy toàn bộ thông tin của Các loại món ăn | Category.cs  (20) |
| 3 | **AddBill(int EmployeeID)**  Input: EmployeeID  Output: BillID  Pseudo code:  Khai báo đối tượng Bill mới bằng new Bill() (trong khi khai báo có lưu lại  BillDate = thời gian hiện tại;  Employee\_ID= đầu vào Mã NV thực hiện xuất Bill;  Total= 0; )  Sử dụng Add(Bill) để thêm dữ liệu vào DatabaseContext  SaveChanges(); | Thêm vào thông tin hóa đơn. Lấy ID của nhân viên đang đăng nhập hệ thống để Tạo hóa đơn, nhân viên này sẽ chịu trách nhiệm về hóa đơn đã tạo. | Bill.cs  (30) |
| 4 | **ThanhToan(intToTal,intCustomer\_ID)**  Input: ToTal, Customer\_ID  Pseudo code:  var dt = ctx.Customers.Where(x => x.ID = Customer\_ID).FirstOrDefault();  if (dt != null)  {  dt.Wallet = dt.Wallet-ToTal;  ctx.SaveChanges();  }  Nếu hành động thanh toán xảy ra, Ví tiền trong tài khoản người mua hàng (Wallet) sẽ bị trừ vào số tiền thanh toán đơn hàng (ToTal). Thông tin Wallet của khách hàng đó được cập nhật | Hàm này sẽ truy xuất về thông tin của Người mua hàng (Customer) để trừ tiền vào Wallet của Customer được chọn. | Bill.cs  (159) |
| 5 | **AddCustomerToBill(int CustomerID, int Bill\_ID)**  Input: ToTal, Customer\_ID  Pseudo code:  var dt = ctx.Bills.Where(x => x.ID == Bill\_ID).FirstOrDefault();  if (dt != null) {  dt.Customer\_ID = CustomerID;  ctx.SaveChanges();  } Hàm nãy sẽ truy xuất vào bảng Bill để lưu lại CustomerID là thông tin đầu vào lưu vào cơ sở dữ liệu của BillID. | Hàm này sẽ kiểm tra trong danh sách khách hàng, những thông tin khách hàng để lưu thông tin vào hóa đơn. | Bill.cs  (147) |
| 6 | **TotalByBill(int Bill\_ID)**  Input: Bill\_ID  Output: Total | Hàm tính ra tổng số tiền phải trả cho một hóa đơn | Bill.cs  (65) |
| 7 | **GetWallet(int CustomerID)**  Input: Customer\_ID  Output: Wallet | Lấy thông tin ví tiền, củ thể là trả về số tiền hiện tại trong Ví thanh toán của Khách hàng | Customer.cs  (65) |
| 8 | **LietKe(DateTime start, DateTime end)**  Input: start (ngày bắt đầu thống kê), end (ngày kết thúc thống kê)  Output: List<dynamic>  Pseudo code:  from b in ctx.Bills  join nv in ctx.Employees on b.Employee\_ID equals nv.ID  where b.BillDate >= start &&  b.BillDate <= end  select new{b.ID, b.BillDate, nv.Name, b.Total}  Câu lệnh SQL liên kết 2 bảng Bill và Employee với nhau để lấy được thông tin Tên Nhân Viên phụ trách hóa đơn. | Liệt kê những thông tin cơ bản của hóa đơn như: Số hóa đơn, ngày xuất hóa đơn, tên nhân viên phụ trách và tổng hóa đơn dựa vào ngày bắt đầu vào ngày kết thúc, xuất thông tin ra datagridview của form ThongKeDoanhThu | Bill.cs  (122) |
| 9 | **ThongKe(DateTime start,DateTime end)**  Input: start (ngày bắt đầu thống kê), end (ngày kết thúc thống kê)  Output: Total  Pseudo code:  Khai báo biến Total (tính tổng) truy xuất vào bảng Bill  Where(t => t.BillDate >= start ).  Where(t => t.BillDate <= end).  Sum(t => t.Total);  return total;  Câu lệnh giới hạn khoảng thời gian đầu vào, tính toán tổng cộng doanh thu từ tổng của từng hóa đơn | Hàm dùng kể thống kê tổng số tiền doanh thu của quán ăn trong một khoảng thời gian nhất định. | BL Bill.cs  (89) |
| 10 | **DeleteBillIfNotExist(int ID\_Bill)**  Input: ID\_Bill | Hàm để xóa những hóa đơn không tồn tại hoặc không được lưu lại | Bill.cs  (74) |
| 11 | **GetBill(int ID\_Bill)**  Input: ID\_Bill  Output: List<dynamic>  Pseudo code:  Truy xuất vào thông tin đối tượng Details, Bills và Foods bằng ID\_Bill.  Xuất ra danh sách Tên món ăn, Đơn vị tính, Gía tiền, Số lượng chọn món ăn đó | Hàm lấy thông tin hóa đơn | Bill.cs  (45) |
| 12 | **AddOrUpdateDetail (int IDFood, int IDBill, int SoLuong)**  Input: IDFood, IDBill, SoLuong  Output: Detail\_ID  Truy xuất vào danh sách của Detail, tìm kiếm những Chi tiết món ăn đã có Food\_ID đầu vào  Nếu có (Cập nhật Chi tiết món ăn)  { Lưu lại số lượng FoodID đó  Return Detail\_ID }  ngược lại (Thêm vào Chi tiết món ăn)  { Khai báo Chi tiết món ăn mới new Detail(), lưu Food\_ID, Bill\_ID và Số lượng món ăn. Add(Detail)  Retu} | Hàm thêm hay cập nhật thông tin cho chi tiết hóa đơn. | Detail.cs  (28) |
| 13 | **DeleteDetail(int ID\_Food,int ID\_Bill)**  Input: IDFood, IDBill  Pseudo code:  Lấy mã món ăn từ Bill\_ID truy xuất vào table Detail để Remove dòng thông tin chi tiết món ăn đó.  SaveChanges(); | Hàm xóa thông tin cho chi tiết hóa đơn | BL Detail.cs  (38) |
| 14 | **IsUser (ref int EmployeeID)**  Input: username, password, quyen  Output: true, false  Pseudo code:  Khai báo biến q là câu lệnh SQL để truy xuất vào table Account dựa vào username, password, quyen  Nếu (q!= null)  { Lưu biến Employee = q.Employee\_ID, ghi nhận Mã nhân viên đang đang nhập hệ thống. Return True }  Ngược lại { Return False } | Lấy thông tin của User vừa nhập đối chiếu thông tin đã lưu trong cơ sở dữ liệu | Account.cs  (34) |
| 15 | **btnDangNhap\_Click(object sender, EventArgs e)**  Input: textUsername.Text, txtPassword.Text, rbAmin.Checked, rbEmployee.Checked | Xử lý đăng nhập theo phân quyền trong giao diện bằng cách chọn vào button Đăng nhập, và cái radio button phân quyền | Form  Login.cs  (21) |
| 16 | **btnThoat\_Click(object sender, EventArgs e)** | Xử lý thoát khỏi giao diện Login bằng cách chọn vào button “X” ở góc phải. | Form  Login.cs  (54) |
| 17 | **Resettext()** | Cập nhật lại trạng thái trống ban đầu cho các ô textbox trong giao diện quản lý thông tin Khách hàng | Form Customer.cs (24) |
| 18 | **LoadData()**  Output: List<dynamic>  Pseudo code:  Truy xuất vào đối tượng Customer  Trả về danh sách chứa các thông tin ID, Name, Address, DateOfBirth, Phone, Sex,  Wallet, Image | Hàm lấy thông tin khách hàng để xuất ra datagridview | Customer.cs (73) |
| 19 | **FormCustomer\_Load(object sender, EventArgs e)** | Hàm xuất thông tin từ datagridview ra màn hình, thay đổi các button để phù hợp cho thao tác | Form Customer.cs (53) |
| 20 | **btnLayAnh\_Click(object sender, EventArgs e)** | Hàm lấy hình ảnh nhân viên từ một đường link trên thư mục ảnh. Hình ảnh sẽ được lưu lại trong cơ sở dữ liệu ở dạng một đường link. | Form Customer.cs (71) |
| 21 | **btnThem\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Thêm | Form Customer.cs (84) |
| 22 | **btnEdit\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Sửa | Form Customer.cs (100) |
| 23 | **btnCapNhat\_Click(object sender, EventArgs e)**  Input: txtHoTen.Text, dtpDOB.Value.Date, txtDiaChi.Text, txtSoDT.Text, rdbNam.Checked, txtWallet.Text  Output: DataSource, datagridview  Pseudo code:  if (Them){  AddData(Customer)}  LoadData();  resettext();  else { Update();  Sửa thông tin theo ID đã xác định trên dữ liệu rồi LoadData(); }  Thông báo kết quả ra messagebox “Đã sửa xong” | Hàm cập nhật dữ liệu sau thao tác Sửa hoặc Thêm | Form Customer.cs (117) |
| 24 | **AddData()**  Pseudo code:  Khai báo đối tượng Customer mới bằng new Customer()  Sử dụng Add(Customer) để thêm dữ liệu vào DatabaseContext  SaveChanges(); | Hàm thêm thông tin Khách hàng mới vào dữ liệu bằng việc truy vấn | Customer.cs  (15) |
| 25 | **Update()**  Pseudo code:  Sử dụng câu lệnh query để truy xuất vào đúng dữ liệu khách hàng mà Admin muốn sửa thông tin (from cus in db.Customers where cus.ID == idcus select cus).  Nếu truy xuất được thì cập nhật.  SaveChanges(); | Hàm thay đổi thông tin Khách hàng trong dữ liệu bằng việc truy vấn | Customer.cs  (32) |
| 26 | **btnHuy\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Hủy | Form Customer.cs (194) |
| 27 | **btnDel\_Click(object sender, EventArgs e)**  Input: txtMaKH.Text  Output: datagridview  Pseudo code:  Khai báo biến traloi  Hiện hộp thoại hỏi đáp “Có chắc xóa tin này hay không?”  Nếu (traloi = = Yes) { DeleteData();  SaveChange()  LoadData() để cập nhật lại datagridview  Thông báo “Đã xóa xong”}  Sử dụng try {} catch {} báo lỗi “Không xóa được” | Hàm xóa dữ liệu sau thao tác Xóa | Form Customer.cs (208) |
| 28 | **DeleteData()**  Pseudo code:  Lấy mã khách hàng từ txtMaKH.Text truy xuất vào table Customer để Remove dòng thông tin của khách hàng đó.  SaveChanges(); | Hàm xóa thông tin Khách hàng trong dữ liệu bằng việc truy vấn | Customer.cs  (40) |
| 29 | **btnTim\_Click(object sender, EventArgs e)**  Input: rdbSDT, rdbName, txtSearch.Text  Output: datagridview  Pseudo code:  Nếu (rdbSDT.Checked){  FindSDT(txtSearch.Text) }  Ngược lại (rdbName.Checked) {  FindName(txtSearch.Text) } | Hàm thực thi việc tìm kiếm để xuất ra danh sách Khách hàng trên màn hình dự vào lựa chọn | Form Customer.cs (270) |
| 30 | **FindName()**  Input: Name  Output: List<dynamic>  Pseudo code:  Lấy đoạn Name trên txtSearch.Text truy xuất theo Customer\_Name trong bảng Customer. Xuất ra các thông tin của khách hàng đó | Hàm tìm kiếm thông tin khách hàng theo Tên khách hàng | Customer.cs (93) |
| 31 | **FindSDT()**  Input: Phone  Output: List<dynamic>  Pseudo code:  Lấy đoạn Phone trên txtSearch.Text truy xuất theo Customer\_Phone trong bảng Customer. Xuất ra các thông tin của khách hàng đó | Hàm tìm kiếm thông tin khách hàng theo SDT khách hàng | Customer.cs (113) |
| 32 | **dgvKH\_CellClick(object sender, DataGridViewCellEventArgs e)** | Hàm xuất thông tin Khách hàng từ textbox lên datagridview | Form Customer.cs (316) |
| 33 | **Resettext()** | Cập nhật lại trạng thái trống ban đầu cho các ô textbox trong giao diện quản lý thông tin Nhân viên | Form Employee.cs (24) |
| 34 | **LoadData()**  Output: List<dynamic> | Hàm lấy thông tin nhân viên lưu vào datagridview | Employee.cs (17) |
| 35 | **btnThem\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Thêm | Form Employee.cs (53) |
| 36 | **btnEdit\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Sửa | Form Employee.cs (69) |
| 37 | **btnCapNhat\_Click(object sender, EventArgs e)**  Input: txtHoTen.Text, dtpDOB.Value.Date, txtDiaChi.Text, txtSoDT.Text, rdbNam.Checked, dtpHireDate.Value.Date  Output: DataSource, datagridview  Pseudo code:  if (Them){ AddData(); }  LoadData();  resettext();  else { Update();  Sửa thông tin theo ID đã xác định trên dữ liệu rồi LoadData(); }  Thông báo kết quả ra messagebox “Đã sửa xong” | Hàm cập nhật dữ liệu sau thao tác Sửa hoặc Thêm | Form Employee.cs (86) |
| 38 | **AddData()**  Pseudo code:  Khai báo đối tượng Employee mới bằng new Employee()  Sử dụng Add(Employee) để thêm dữ liệu vào DatabaseContext  SaveChanges(); | Hàm thêm thông tin Nhân Viên mới vào dữ liệu bằng việc truy vấn | Employee.cs  (15) |
| 39 | **Update()**  Pseudo code:  Sử dụng câu lệnh query để truy xuất vào đúng dữ liệu khách hàng mà Admin muốn sửa thông tin (from em in db.Employees where em.ID == idem select em).  Nếu truy xuất được thì cập nhật.  SaveChanges(); | Hàm thay đổi thông tin Nhân viên trong dữ liệu bằng việc truy vấn | Employee.cs  (32) |
| 40 | **btnHuy\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Hủy | Form Employee.cs (163) |
| 41 | **btnDel\_Click(object sender, EventArgs e)**  Input: txtMaNV.Text  Output: datagridview  Pseudo code:  Khai báo biến traloi  Hiện hộp thoại hỏi đáp “Có chắc xóa tin này hay không?”  Nếu (traloi = = Yes) {  DeleteData();  SaveChange()  LoadData() để cập nhật lại datagridview  Thông báo “Đã xóa xong”}  Sử dụng try {} catch {} báo lỗi “Không xóa được” | Hàm xóa dữ liệu sau thao tác Xóa | Form Employee.cs (178) |
| 42 | **DeleteData()**  Pseudo code:  Lấy mã nhân viên từ txtMaNV.Text truy xuất vào table Employee để Remove dòng thông tin của nhân viên đó.  SaveChanges(); | Hàm xóa thông tin Nhân viên trong dữ liệu bằng việc truy vấn | Employee.cs  (40) |
| 43 | **btnTim\_Click(object sender, EventArgs e)**  Input: rdbSDT, rdbName, txtSearch.Text  Output: datagridview  Pseudo code:  Nếu (rdbSDT.Checked){  FindSDT(txtSearch.Text) }  Ngược lại (rdbName.Checked) {  FindTen(txtSearch.Text) } | Hàm thực thi việc tìm kiếm để xuất ra danh sách Nhân viên trên màn hình dự vào lựa chọn | Form Employee.cs (302) |
| 44 | **FindName()**  Input: Name  Output: List<dynamic>  Pseudo code:  Lấy đoạn Name trên txtSearch.Text truy xuất theo Employee\_Name trong bảng Employee. Xuất ra các thông tin của nhân viên đó | Hàm tìm kiếm thông tin nhân viên theo Tên nhân viên | Employee.cs (96) |
| 45 | **FindSDT()**  Input: Phone  Output: List<dynamic>  Pseudo code:  Lấy đoạn Phone trên txtSearch.Text truy xuất theo Employee\_Phone trong bảng Employee. Xuất ra các thông tin của nhân viên đó | Hàm tìm kiếm thông tin khách hàng theo SDT nhân viên | Employee.cs (76) |
| 46 | **dgvNV\_CellClick(object sender, DataGridViewCellEventArgs e)** | Hàm xuất thông tin Nhân viên từ textbox lên datagridview | Form Employee.cs (229) |
| 47 | **FormEmployee\_Load(object sender, EventArgs e)** | Hàm xuất thông tin từ datagridview ra màn hình, thay đổi các button để phù hợp cho thao tác | Form Employee.cs (252) |
| 48 | **Resettext()** | Cập nhật lại trạng thái trống ban đầu cho các ô textbox trong giao diện quản lý thông tin Món ăn | Food Manager.cs (42) |
| 49 | **LoadData()**  Output: List<dynamic>  Pseudo code:  Truy xuất vào đối tượng Food  Trả về danh sách chứa các thông tin ID, Name, Unit, Price, Category\_ID, Image | Hàm lấy thông tin món ăn lưu vào datagridview | Food.cs (105) |
| 50 | **btnThem\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Thêm | Food Manager.cs (155) |
| 51 | **btnEdit\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Sửa | Food Manager.cs (185) |
| 52 | **btnCapNhat\_Click(object sender, EventArgs e)**  Input: txtTen.Text, txtDonVi.Text, txtDonGia.Text, cmbLoai.SeletedValue  Output: DataSource, datagridview  Pseudo code:  if (Them){ AddData(); }  LoadData();  resettext();  else { Update();  Sửa thông tin theo ID đã xác định trên dữ liệu rồi LoadData(); }  Thông báo kết quả ra messagebox “Đã sửa xong” | Hàm cập nhật dữ liệu sau thao tác Sửa hoặc Thêm | Food Manager.cs (81) |
| 53 | **AddData()**  Pseudo code:  Khai báo đối tượng Food mới bằng new Food()  Sử dụng Add(Food) để thêm dữ liệu vào DatabaseContext  SaveChanges(); | Hàm thêm thông tin Món ăn mới vào dữ liệu bằng việc truy vấn | Food.cs  (15) |
| 54 | **Update()**  Pseudo code:  Sử dụng câu lệnh query để truy xuất vào đúng dữ liệu món ăn mà Admin muốn sửa thông tin (from fd in db.Foods where fd.ID == idfd select fd).  Nếu truy xuất được thì cập nhật.  SaveChanges(); | Hàm thay đổi thông tin Món ăn trong dữ liệu bằng việc truy vấn | Food.cs  (32) |
| 55 | **btnHuy\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Hủy | Food Manager.cs (171) |
| 56 | **btnDel\_Click(object sender, EventArgs e)**  Input: txtMaMonAn.Text  Output: datagridview  Pseudo code:  Khai báo biến traloi  Hiện hộp thoại hỏi đáp “Có chắc xóa tin này hay không?”  Nếu (traloi = = Yes) {  DeleteData();  LoadData() để cập nhật lại datagridview  Thông báo “Đã xóa xong”}  Sử dụng try {} catch {} báo lỗi “Không xóa được” | Hàm xóa dữ liệu sau thao tác Xóa | Food Manager.cs (202) |
| 57 | **DeleteData()**  Pseudo code:  Lấy mã món ăn từ txtMaMonAn.Text truy xuất vào table Food để Remove dòng thông tin của món ăn đó.  SaveChanges(); | Hàm xóa thông tin Món ăn trong dữ liệu bằng việc truy vấn | Food.cs  (40) |
| 58 | **btnTim\_Click(object sender, EventArgs e )**  Input: rdbLoai, rdbName, txtSearch.Text  Output: datagridview  Pseudo code:  Nếu (rdbLoai.Checked){  FindTypeName(txtSearch.Text) }  Ngược lại (rdbName.Checked) {  FindFoodName(txtSearch.Text) } | Hàm thực thi việc tìm kiếm để xuất ra danh sách Món ăn trên màn hình dự vào lựa chọn | Food Manager.cs (298) |
| 59 | **FindFoodName()**  Input: Food\_Name  Output: List<dynamic>  Pseudo code:  Lấy đoạn Tên món ăn trên txtSearch.Text truy xuất theo Food\_Name trong bảng Food. Xuất ra các thông tin của món hàng đó | Hàm tìm kiếm thông tin món hàng theo Tên món hàng | Food.cs (141) |
| 60 | **FindTypeName()**  Input: Category\_Name  Output: List<dynamic>  Pseudo code:  Lấy đoạn Phone trên txtSearch.Text truy xuất theo Category\_Name trong bảng Food. Xuất ra các thông tin của món hàng đó | Hàm tìm kiếm thông tin món hàng theo Tên loại món hàng | Food.cs (123) |
| 61 | **dgvMA\_CellClick(object sender, DataGridViewCellEventArgs e)** | Hàm xuất thông tin Món ăn từ textbox lên datagridview | Food Manager.cs (254) |
| 62 | **FoodManager\_Load(object sender, EventArgs e)** | Hàm xuất thông tin từ datagridview ra màn hình, thay đổi các button để phù hợp cho thao tác | Food Manager.cs (24) |
| 63 | **btnLayAnh\_Click(object sender, EventArgs e)** | Hàm lấy hình ảnh nhân viên từ một đường link trên thư mục ảnh. Hình ảnh sẽ được lưu lại trong cơ sở dữ liệu ở dạng một đường link. | Food Manager.cs (241) |
| 64 | **btnChon\_Click(object sender, EventArgs e)**  Input: CustomerID, BillID  Output: Report (BillID) | Hàm chọn Khách hàng từ dữ liệu, và thông báo trạng thái ví tiền nếu không đủ tiền, và trạng thái thanh toán nếu thanh toán thành con | Chose Customer.cs (35) |
| 65 | **ChoseCustomer\_Load(object sender, EventArgs e)** | Hàm xuất thông tin từ datagridview ra màn hình | Chose Customer (78) |
| 66 | **btnThem\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Thêm | Chose Customer.cs (101) |
| 67 | **btnLayAnh\_Click(object sender, EventArgs e)** | Hàm lấy hình ảnh nhân viên từ một đường link trên thư mục ảnh. Hình ảnh sẽ được lưu lại trong cơ sở dữ liệu ở dạng một đường link. | Chose Customer.cs (156) |
| 68 | **btnCapNhat\_Click(object sender, EventArgs e)**  Input: txtHoTen.Text, dtpDOB.Value.Date, txtDiaChi.Text, txtSoDT.Text, rdbNam.Checked, txtWallet.Text  Output: DataSource, datagridview  Pseudo code:  if (Them){ AddData(); }  LoadData();  resettext();  Thông báo kết quả ra messagebox “Đã thêm xong” | Hàm cập nhật dữ liệu sau thao tác Sửa hoặc Thêm | Chose Customer.cs (127) |
| 69 | **btnHuy\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Hủy | Chose Customer.cs (115) |
| 70 | **btnTim\_Click(object sender, EventArgs e)**  Input: rdbSDT, rdbName, txtSearch.Text  Output: datagridview  Pseudo code:  Nếu (rdbSDT.Checked){  FindSDT(txtSearch.Text) }  Ngược lại (rdbName.Checked) {  FindName(txtSearch.Text) } | Hàm thực thi việc tìm kiếm để xuất ra danh sách Khách hàng trên màn hình dự vào lựa chọn | Chose Customer.cs (169) |
| 71 | **dgvKH\_CellClick(object sender, DataGridViewCellEventArgs e)** | Hàm xuất thông tin Khách hàng từ textbox lên datagridview | Chose Customer.cs (78) |
| 72 | **Resettext()** | Cập nhật lại trạng thái trống ban đầu cho các ô textbox trong giao diện quản lý thông tin Đăng nhập | Form  Login  Manage.cs (22) |
| 73 | **LoadData()**  Output: List<dynamic>  Pseudo code:  Truy xuất vào đối tượng Account  Trả về danh sách chứa các thông tin Username, Password, IsAdmin, Price, Employee\_ID | Hàm lấy thông tin đăng nhập lưu vào datagridview | Account.cs (98) |
| 74 | **btnThem\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Thêm | FormLoginManage.cs (124) |
| 75 | **btnEdit\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Sửa | FormLoginManage.cs (154) |
| 76 | **btnCapNhat\_Click(object sender, EventArgs e)**  Input: txtUserName.Text, txtNVID.Text, txtPassword.Text, rdbAdmin.Checked  Output: DataSource, datagridview  Pseudo code:  if (Them){ AddData(); }  LoadData();  resettext();  else { Update(); }  Thông báo kết quả ra messagebox “Đã sửa xong” | Hàm cập nhật dữ liệu sau thao tác Sửa hoặc Thêm | FormLoginManage.cs (56) |
| 77 | **AddData()**  Pseudo code:  Khai báo đối tượng Account mới bằng new Account()  Sử dụng Add(Account) để thêm dữ liệu vào DatabaseContext  SaveChanges(); | Hàm thêm thông tin Đăng nhập mới vào dữ liệu bằng việc truy vấn | Account.cs  (52) |
| 78 | **Update()**  Pseudo code:  Sử dụng câu lệnh query để truy xuất vào đúng dữ liệu món ăn mà Admin muốn sửa thông tin (from ac in db.Accounts where ac.Username == Username select ac).  Nếu truy xuất được thì cập nhật.  SaveChanges(); | Hàm thay đổi thông tin Đăng nhập trong dữ liệu bằng việc truy vấn | Accountcs  (66) |
| 79 | **btnHuy\_Click(object sender, EventArgs e)** | Hàm xử lý giao diện để phù hợp cho thao tác Hủy | FormLoginManage.cs (139) |
| 80 | **btnDel\_Click(object sender, EventArgs e)**  Input: txtUsername.Text  Output: datagridview  Pseudo code:  Khai báo biến traloi  Hiện hộp thoại hỏi đáp “Có chắc xóa tin này hay không?”  Nếu (traloi = = Yes) {  DeleteData();  LoadData() để cập nhật lại datagridview  Thông báo “Đã xóa xong”}  Sử dụng try {} catch {} báo lỗi “Không xóa được” | Hàm xóa dữ liệu sau thao tác Xóa | FormLoginManage.cs (202) |
| 81 | **DeleteData()**  Pseudo code:  Lấy mã món ăn từ txtUsername.Text truy xuất vào table Account để Remove dòng thông tin đăng nhập đó.  SaveChanges(); | Hàm xóa thông tin Đăng nhập trong dữ liệu bằng việc truy vấn | Account.cs (85) |
| 82 | **btnTim\_Click(object sender, EventArgs e)**  Input: rdbID, rdbUsername, txtSearch.Text  Output: datagridview  Pseudo code:  Nếu (rdbID.Checked){  FindID(txtSearch.Text) }  Ngược lại (rdbUsername.Checked) {  FindUsername(txtSearch.Text) } | Hàm thực thi việc tìm kiếm để xuất ra danh sách Đăng nhập trên màn hình dự vào lựa chọn | FormLoginManage.cs (261) |
| 83 | **FindID()**  Input: Employee\_ID  Output: List<dynamic>  Pseudo code:  Lấy đoạn Mã nhân viên trên txtSearch.Text truy xuất theo Employee\_ID trong bảng Account. Xuất ra các thông tin của món hàng đó | Hàm tìm kiếm thông tin đăng nhập theo Mã nhân viên | Account.cs (131) |
| 84 | **FindUsername()**  Input: Username  Output: List<dynamic>  Pseudo code:  Lấy đoạn Username trên txtSearch.Text truy xuất theo Username trong bảng Account. Xuất ra các thông tin đăng nhập đó | Hàm tìm kiếm thông tin đăng nhập theo Username | Account.cs (114) |
| 85 | **dgvAcc\_CellClick(object sender, DataGridViewCellEventArgs e)** | Hàm xuất thông tin Đăng nhập từ textbox lên datagridview | FormLoginManage.cs (209) |
| 86 | **FormLoginManage\_Load(object sender, EventArgs e)** | Hàm xuất thông tin từ datagridview ra màn hình, thay đổi các button để phù hợp cho thao tác | FormLoginManage.cs (235) |
| 87 | **btnTK\_Click(object sender, EventArgs e)**  Input: Ngày đầu, ngày cuối  Output: Total | Hàm xuất ra danh sách hóa đơn được lọc theo ngày được chọn trên giao diện | ThongKeDoanhThu.cs (29) |
| 88 | **chon\_Click(object sender, EventArgs e)**  Output: datagridview Hóa đơn, textbox Tổng tiền hóa đơn  Pseudo code:  Khai báo biến button chọn b và numberupdown Số lượng món num.  Nếu người dùng đã chọn số lượng (num.Value > 0) {  AddOrUpdateDetail(ID\_Food, ID\_Bill, SoLuong).  Xuất ra màn hình đơn hàng những món đã chọn bằng hàm GetBill.  textBox2.Text sử dụng hàm TotalByBill từ Bill để tính tổng hóa đơn đã chọn } | Hàm chọn món ăn trên giao diện chọn món để xuất ra màn hình Hóa đơn những món ăn đã chọn. Tính tổng hóa đơn và xuất ra Số tiền khách phải trả | FormFoodChoices.cs (143) |
| 89 | **bochon\_Click(object sender, EventArgs e)**  Output: datagridview Hóa đơn, textbox Tổng tiền hóa đơn  Pseudo code:  Khai báo biến button chọn b và numberupdown Số lượng món num.  Gán số lượng bằng 0 (num.Value = 0) {  DeleteDetail(ID\_Food, ID\_Bill)  Xuất ra màn hình đơn hàng những món đã chọn bằng hàm GetBill.  textBox2.Text sử dụng hàm TotalByBill từ Bill để tính tổng hóa đơn đã chọn } | Hàm bỏ chọn món ăn trên giao diện chọn món để xóa đi những món ăn không mong muốn đã chọn hiển thị trên Hóa đơn. Tính tổng hóa đơn và xuất ra Số tiền khách phải trả. | FormFoodChoices.cs (159) |

### Thiết kế cơ sở dữ liệu:



##### Hình 3: Cơ sở dữ liệu của hệ thống

##### Bảng 4: Bảng mô tả các Table trong Cơ sở dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Mục đích** |
| 1 | ACCOUNT | Thuộc tính quản lý đăng nhập |
| 2 | BILL | Thuộc tính quản lý Hóa Đơn |
| 3 | CATEGORY | Thuộc tính quản lý Loại Món Ăn (Nước, Khô,  Đồ Uống) |
| 4 | CUSTOEMR | Thuộc tính quản lý Khách Hàng |
| 5 | DETAIL | Thuộc tính quản lý chi tiết Hóa Đơn |
| 6 | EMPLOYEE | Thuộc tính quản lý Nhân Viên |
| 7 | FOOD | Thuộc tính quản lý Món Ăn |

##### Bảng 5: Bảng mô tả các Field trong Table:

##### Bảng 5.1: Table Account (người dùng ở đây có nghĩa là Nhân Viên)

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Username | string[MaxLen(100)] | Tên đăng nhập người dùng |
| 2 | Password | string[MinLen(8)] | Mật khẩu người dùng |
| 3 | IsAdmin | bool | Kiểm tra xem có phải Admin không |
| 4 | Name | string[MaxLen(50)] | Tên người dùng |
| 5 | DOB | datetime | Ngày sinh người dùng |
| 6 | Phone | string | Số điện thoại người dùng |
| 7 | Sex | string | Giới tính người dùng |

##### Bảng 5.2: Table Bill

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Bill\_ID | int | Mã Hóa đơn |
| 2 | BillDate | Datetime | Ngày lập hóa đơn |
| 3 | Customer\_ID | int | Mã khách hàng |
| 4 | Employee\_ID | Int | Mã nhân viên |

##### Bảng 5.3: Table Category

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Category\_ID | int | Mã Loại Món Ăn |
| 2 | Name | string | Tên Loại Món ăn |

##### Bảng 5.4: Table Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Customer\_ID | int | Mã Khách hàng |
| 2 | Name | string[MaxLen(50)] | Tên Khách hàng |
| 3 | Address | String | Địa chỉ của Khách hàng |
| 4 | DOB | datetime | Ngày sinh Khách hàng |
| 5 | Phone | String | Số điện thoại Khách hàng |
| 6 | Sex | String | Giới tính Khách hàng |
| 7 | Image | String | Ảnh Khách hàng |
| 8 | Wallet | int | Ví tiền trong tài khoản của KH |

##### Bảng 5.5: Table Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Employee\_ID | int | Mã Nhân viên |
| 2 | Name | string[MaxLen(50)] | Tên Nhân viên |
| 3 | Address | String | Địa chỉ của Nhân viên |
| 4 | DOB | datetime | Ngày sinh Nhân viên |
| 5 | Phone | String | Số điện thoại Nhân viên |
| 6 | Sex | String | Giới tính Nhân viên |
| 7 | Image | String | Ảnh Nhân viên |
| 8 | HireDate | datetime | Ngày vào làm của Nhân viên |

##### Bảng 5.6: Table Food

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Food\_ID | int | Mã Nhân viên |
| 2 | Name | string[MaxLen(25)] | Tên Nhân viên |
| 3 | Price | Float | Địa chỉ của Nhân viên |
| 4 | Unit | String | Ngày sinh Nhân viên |
| 5 | Category\_ID | int | Số điện thoại Nhân viên |

##### Bảng 5.7: Table Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mục đích** |
| 1 | Datail\_ID | int | Mã Chi tiết hóa đơn |
| 2 | Amount | Int | Số lượng |
| 3 | Bill\_ID | Int | Mã Hóa đơn |
| 4 | Food\_ID | int | Mã Món Ăn |

### Thiết kế giao diện:

##### Bảng 6: Thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Màn hình/Cửa sổ/Dialog** | **Giải thích ngắn gọn** | **Mục đích** |
| 1 | Màn hình giao diện chính:   Hình 4.1: Giao diện chính | Hiện ra những button với các chức năng tương ứng. (nhóm quyền Admin được phép truy cập) | Hiển thị ra giao diện chính của phần mềm. |
| 2 | Màn hình đăng nhập :   Hình 4.2: Giao diện đăng nhập | Lấy thông tin và quyền người dùng tương ứng khi đăng nhập vào. | Hiển thị giao diện đăng nhập vào phần mềm. |
| 3 | Màn hình bảng khách hàng :   Hình 4.3: Giao diện thông tin Khách hàng | Cho phép thêm hoặc chỉnh sửa, xóa nhân viên (nhóm quyền Admin, Staff được phép truy cập). | Hiển thị giao diện xem thông tin cá nhân của Khách Hàng |
| 4 | Màn hình bảng nhân viên:   Hình 4.4: Giao diện thông tin Nhân viên | Cho phép thêm hoặc chỉnh sửa, xóa nhân viên (nhóm quyền Admin được phép truy cập). | Hiển thị giao diện nhân viên. |
| 5 | Màn hình bảng chọn món:   Hình 4.5: Giao diện Bảng chọn món | Cho phép khách hàng chọn Món với các Tab: Món ăn Khô, món ăn Nước, Nước giải khát | Hiển thị giao diện bảng chọn món |
| 6 | Màn hình bảng chọn món:   Hình 4.6: Giao diện thông tin Đăng nhập | Cho phép thêm hoặc chỉnh sửa, xóa thông tin đăng nhập của các nhân viên (nhóm quyền Admin được phép truy cập). | Hiện thị giao diện quản lý đăng nhập |
| 7 | Màn hình bảng món ăn:   Hình 4.7: Giao diện thông tin Món ăn | Cho phép thêm hoặc chỉnh sửa, xóa món ăn (nhóm quyền Admin được phép truy cập). | Hiển thị giao diện quản lý món ăn |
| 8 | Màn hình Phiếu hóa đơn   Hình 4.8: Giao diện xuất phiếu hóa đơn | Xuất ra Phiếu hóa đơn để khách hàng xem lại Thông tin hóa đơn vừa thanh toán. Trong phiếu thanh toán có bao gồm: tên nhân viên xuất phiếu, tên khách hàng, ngày xuất phiếu, tổng giá trị đơn mua | Hiển thị giao diện Report Hóa đơn |
| 9 | Màn hình Thống kê doanh thu:   Hình 4.9: Giao diện thống kê doanh thu | Cho phép Amin xem lại và thống kê tổng doanh thu trong từng khoảng thời gian | Hiển thị giao diện thống kê doanh thu |

## IV. PHẦN KẾT LUẬN

### 1. Ưu điểm, nhược điểm:

#### 1.1. Ưu điểm

- Giao diện đẹp, ưa nhìn, đơn giản. Giúp cho người dùng mới dễ tiếp cận và sử dụng

- Đã thực hiện được những chức năng cơ bản của một Phần mềm Món ăn Tự chọn

- Đã ứng dụng được kiến thức về Lập trình hướng đối tượng vào Phần mềm

#### 1.2. Nhược điểm

- Vì chưa chuyển hình ảnh sang dạng tối ưu nên khi kéo scroll sẽ bị chậm đối với 1 số máy tính

- Việc thống kê doanh thu chưa thực sự hiệu quả.

### 2. Thảo luận

#### 2.1. Nhìn chung

Trong quá trình thực hiện đề tài, mặc dù đã ra sức nghiên cứu và cố gắng nhưng chắc chắn khó tránh khỏi những thiếu sót, chúng em rất mong nhận được sự chỉ dẫn, ý kiến đóng góp của thầy để chúng em ngày càng hoàn thiện hơn.

Sau một thời học tập và nghiên cứu, em nhận thấy một số khó khăn:

+ Việc tham khảo và viết ra báo cáo đồ án hoàn chỉnh cũng là một khó khăn lớn. Cụ thể là cách điễn đạt cho các đoạn code được sử dụng trong đồ án cũng như trình tự các phần mục trong đồ án.

+ Bên cạnh đó, chúng em cũng gặp phải khó khăn trong việc thực hiện theo kế hoạch. Cụ thể là có một số thay đổi về người thực hiện công việc và một số trì hoãn về thời gian hoàn thành so với thời gian dự kiến.

#### 2.2. Bài học rút ra

Sau quá trình thực hiện đề tài, chúng em đã đúc kết thêm được một số kinh nghiệm có thể giúp cho các đồ án tiếp theo có thể tốt hơn. Cụ thể là:

+ Cần phải chủ động hơn về thời gian, dự trù trước thời hạn hoàn thành để đi đúng theo thời gian hoàn thành dự kiến đã đề ra.

+ Cần phải dành nhiều thời gian hơn trong việc tìm hiểu về đồ án để có thể hiểu rõ mình đang làm gì.

+ Cần phải họp nhóm nhiều hơn là mỗi người tự làm. Vì đôi khi ban đầu chia theo kế hoạch, nhưng người này có khả năng làm được công việc của người kia, và ngược lại. Điều này còn giúp cho cả 3 hiểu được mình đang làm gì và sẽ làm gì cùng lúc, đỡ phải gặp trường hợp 3 người không hiểu ý nhau, hoặc 1 người làm xong code nhưng không phù hợp với người kia, tránh làm chậm tiến trình công việc.

### 3. Hướng phát triển

Vì không có đủ kiến thức, cũng như thời gian nên chúng em chưa thể làm tốt nhất trong đồ án lần này. Vì vậy, nếu có thêm nhiều cơ hội chúng em tự hứa phát triển đồ án này tốt hơn bây giờ. Sau đây là một số hướng đi khác để phát triển đồ án trở nên tốt hơn:

+ Tìm hiểu thêm các cách ứng dụng Kế thừa để có thể áp dụng triệt để hơn khi Kế thừa các phương thức Thêm, Xóa, Sửa, Tìm kiếm, Cập Nhật, Xuất dữ liệu

+ Thiết kế phần mềm tương thích với nhiều công cụ: điện thoại di động, máy tính bảng,…

+ Hệ thống hiện tại chỉ có 2 phân quyền cơ bản: Admin, Employee. Hệ thống cần được bổ sung thêm phân quyền đầu bếp để nhận được thông tin đặt hàng từ khách hàng ngay lập tức và theo trình tự. Từ đó, khách hàng chỉ việc chọn món thì đầu bếp đã nhận được món ăn và thông báo cho tiếp tân khi hoàn thành món ăn.

# TÀI LIỆU THAM KHẢO

1. Tài liệu Entity Framework – Code First: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx> (12/11/2019)
2. Tài liệu Inheritance with Entity Framework Code First: <https://weblogs.asp.net/manavi/inheritance-mapping-strategies-with-entity-framework-code-first-ctp5-part-2-table-per-type-tpt> (12/11/2019)
3. Video C# & MySQL Create a CRUD Operations using class OOP: <https://www.youtube.com/watch?v=oZQESfwrOgo&t=297s> (8/12/2019)
4. Tài liệu hướng dẫn dùng OpenFileDialog: <https://csharpcanban.com/c-huong-dan-su-dung-openfiledialog.html#.XfSK6R83vIU> (30/11/2019)
5. Tài liệu dùng Linq-to-Entities Query: <https://www.entityframeworktutorial.net/querying-entity-graph-in-entity-framework.aspx> (1/12/2019)