**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**

**Đồ án 1: ĐỀ TÀI 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**

**GVPT: HUỲNH XUÂN PHỤNG**

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

MỤC LỤC

[DANH MỤC CÁC HÌNH 3](#_Toc25358052)

[DANH MỤC CÁC BẢNG 3](#_Toc25358053)

[PHẦN NỘI DUNG 4](#_Toc25358054)

[I. Mô tả đồ án 4](#_Toc25358055)

[1. Phần mềm quản lý quán ăn tự chọn 4](#_Toc25358056)

[2. Dữ liệu, thông tin đầu vào: 4](#_Toc25358057)

[4. Giao diện dự kiến: 6](#_Toc25358058)

[II. PHÂN CÔNG CÔNG VIỆC: 7](#_Toc25358059)

[III. THIẾT KẾ: 8](#_Toc25358060)

[1. Thuật toán: 8](#_Toc25358061)

[2. Thiết kế lớp: 18](#_Toc25358062)

[3. Thiết kế cơ sở dữ liệu: 21](#_Toc25358063)

[4. Bảng mô tả các Field trong Table: 22](#_Toc25358064)

[5. Thiết kế giao diện: 24](#_Toc25358065)

[IV. Cài đặt và kiểm thử 27](#_Toc25358066)

[PHẦN KẾT LUẬN 28](#_Toc25358067)

[1. Ưu điểm, nhược điểm: 28](#_Toc25358068)

[a. Ưu điểm 28](#_Toc25358069)

[b. Nhược điểm 28](#_Toc25358070)

[2. Nhìn chung 28](#_Toc25358071)

[3. Bài học rút ra 29](#_Toc25358072)

[4. Hướng phát triển 29](#_Toc25358073)

# DANH MỤC CÁC HÌNH

* Hình 1: Giao diện đăng nhập
* Hình 2: Giao diện Menu chính
* Hình 3: Cơ sở dữ liệu của hệ thống
* Hình 4: Nhóm các hình giao diện

*+ Hình 4.1: Giao diện chính*

*+ Hình 4.2: Giao diện đăng nhập*

*+ Hình 4.3: Giao diện thông tin Khách hàng*

*+ Hình 4.4: Giao diện thông tin Nhân viên*

*+ Hình 4.5: Giao diện Bảng chọn món*

*+ Hình 4.6: Giao diện thông tin Đăng nhập*

*+ Hình 4.7: Giao diện thông tin Món ăn*

# DANH MỤC CÁC BẢNG

* Bảng 1: Bảng phân công công việc và phần trăm đạt được
* Bảng 2: Bảng danh mục các lớp chính được sử dụng trong chương trình
* Bảng 3: Bảng danh mục các phương thức chính được sử dụng trong một lớp
* Bảng 4: Bảng mô tả các Table trong Cơ sở dữ liệu
* Bảng 5: Nhóm các bảng mô tả các Field trong Table
* Bảng 6: Thiết kế giao diện

# 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.

* Thông báo tự động tới quầy bếp / chế biến:

+ Tất cả yêu cầu từ Thu ngân/Bồi bàn sẽ được chuyển xuống bếp tức thì ngay khi khách hàng gọi món.

+ 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.

+ Khi món ăn được hoàn thành, nhân viên bếp có thể thông báo tới Bồi bàn ngay lập tức.

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

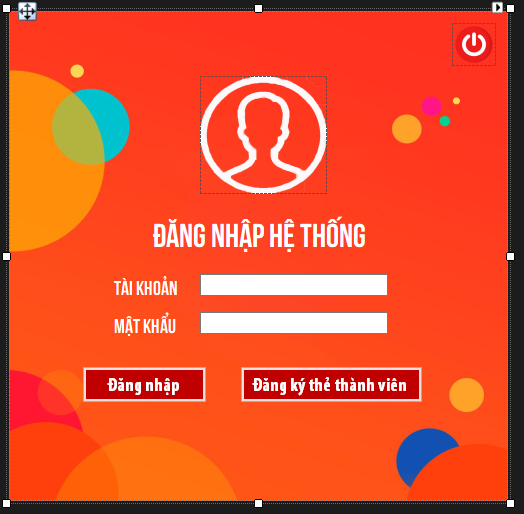
+ Với ưu điểm có thể tích hợp trên thiết bị di động, màn hình có thể hiển thị tất cả món ăn, đồ uống có tại Quán ăn.

* 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:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Sinh Viên** | **Khát quát công việc** | **Tỷ lệ đạt** |
| 1 | Đoàn Văn Long | Thiết kế cơ sở dữ liệu | 100% |
| 2 | Ngô Hoàng Minh Tâm | Bổ sung thông tin dữ liệu | 100% |
| 3 | Ngô Hoàng Minh Tâm | Thiết kế giao diện phần mềm | 100% |
| 4 | Đoàn Văn Long | Các thao tác cơ bản trong Quản  lý Nhân Viên (thêm, xóa, sửa,  cập nhật, tìm kiếm) | 100% |
| 5 | Đoàn Văn Long | Các thao tác cơ bản trong Quản  lý Khách Hàng (thêm, xóa, sửa,  cập nhật, tìm kiếm) | 100% |
| 6 | Đoàn Văn Long | Các thao tác cơ bản trong Quản  lý Đăng nhập – phân quyền  (thêm, xóa, sửa, cập nhật, tìm  kiếm) | 100% |
| 7 | Trần Nguyên Tài | Các thao tác trong việc Lựa  chọn món ăn (lựa chọn món,  xuất ra hóa đơn bán lẻ xem  trước) | 100% |
|  |  | Các thao tác cơ bản trong Quản  lý Món Ăn (thêm, xóa, sửa, cập  nhật, tìm kiếm) | 100% |
| 8 |  | Xuất phiếu hóa đơn bán lẻ | 100% |
| 9 |  | Xuất phiếu báo cáo ngày | 100% |
| 10 | Ngô Hoàng Minh Tâm | Viết báo cáo (doc) | 100% |
| 11 |  | Làm file trình chiếu (ppt) | 100% |

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

## III. THIẾT KẾ:

### Thuật toán:

#### Các phương thức ở Tầng dữ liệu

##### Lớp DataContext

public DbSet<Customer> Customers { get; set; }

public DbSet<Employee> Employees { get; set; }

public DbSet<Food> Foods { get; set; }

public DbSet<Category> Categories { get; set; }

public DbSet<Bill> Bills { get; set; }

public DbSet<Detail> Details { get; set; }

public DbSet<Account> Accounts { get; set; }

##### Lớp Account: Tạo đối tượng Đăng nhập

[Table("Account")]

public class Account

{

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.None)]

[Required]

[MaxLength(100)]

public string Username { get; set; }

[Required]

[MinLength(8)]

public string Password { get; set; }

[Required]

public bool IsAdmin { get; set; }

[DisplayName("Họ tên")]

[Required]

[MaxLength(50)]

public string Name { get; set; }

public DateTime DOB { get; set; }

public string Phone { get; set; }

public string Sex { get; set; }

}

##### Lớp Bill: Tạo đối tượng Hóa Đơn

[Table("Bill")]

public class Bill

{

[Key]

public int ID { get; set; }

public DateTime BillDate { get; set; }

public virtual Customer Customer { get; set; }

public virtual Employee Employee { get; set; }

}

##### Lớp Category: Tạo đối tượng Loại Món Ăn

[Table("Category")]

public class Category

{

public int ID { get; set; }

[Required]

[MaxLength(50)]

public string Name { get; set; }

}

##### Lớp Person: Tạo đối tượng Người

public class Person

{

[Key]

public int ID { get; set; }

[DisplayName("Họ tên")]

[Required]

[MaxLength(50)]

public string Name { get; set; }

[DisplayName("Địa chỉ")]

[Required]

public string Address { get; set; }

[DisplayName("Ngày sinh")]

[Required]

public DateTime DOB { get; set; }

[DisplayName("SĐT")]

[Required]

[MaxLength(25)]

public string Phone { get; set; }

[DisplayName("Giới tính")]

[Required]

[MaxLength(3)]

public string Sex { get; set; }

[DisplayName("Hình ảnh")]

public string Image { get; set; }

}

##### Lớp Employee: tạo đối tượng cho Nhân Viên (áp dựng Kế Thừa lớp Person)

[Table("Employee")]

public class Employee : Person

{

public DateTime HireDate { get; set; }

}

##### Lớp Customer: tạo đối tượng cho Khách Hàng (áp dựng Kế Thừa lớp Person)

[Table("Customer")]

public class Customer : Person

{

public int Wallet { get; set; }

}

##### Lớp Detail: Tạo đối tượng Chi tiết hóa đơn

[Table("Detail")]

public class Detail

{

[Key]

public int ID { get; set; }

public int Amount { get; set; }

public virtual Bill Bill { get; set; }

public virtual Food Food { get; set; }

}

##### Lớp Food: Tạo đối tượng cho Món ăn

[Table("Food")]

public class Food

{

[Key]

public int ID { get; set; }

[DisplayName("Họ tên")]

[Required]

[MaxLength(25)]

public string Name { get; set; }

[DisplayName("Đơn giá")]

[Required]

public float Price { get; set; }

[DisplayName("ĐVT")]

[Required]

[MaxLength(25)]

public string Unit { get; set; }

public int Category\_ID { get; set; }

[ForeignKey("Category\_ID")]

public virtual Category Category { get; set; }

#### Các phương thức ở Tầng nghiệp vụ

##### Lớp BL Menu

class BLMenu

{

public static List<dynamic> GetFoodByCategory(int CategoryID)

{

using (var ctx = new StoreContext())

{

var list = (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

}).Distinct();

return list.ToList<dynamic>();

}

}

public static List<Category> GetListCategory()

{

using (var ctx = new StoreContext())

{

return ctx.Categories.ToList();

}

}

}

##### Lớp BL Employee

##### Lớp BL Customer

#### Các phương thức ở Tầng giao diện

##### Lớp FormFoodChoices: Lựa chọn Món ăn và xem Giỏ hàng đã chọn

##### Lớp FormEmployee: Quản lý thông tin khách hàng bằng các phương thức thêm, xóa, sửa, tìm kiếm, cập nhật

public partial class FormEmployee : Form

{

bool Them;

string imagepath;

public FormEmployee()

{

InitializeComponent();

}

void resettext()

{

txtMaNV.ResetText();

txtHoTen.ResetText();

txtDiaChi.ResetText();

txtSoDT.ResetText();

dtpDOB.ResetText();

dtpHireDate.ResetText();

}

void LoadData()

{

using (StoreContext db = new StoreContext())

{

var result = from c in db.Employees

select new

{

ID = c.ID,

Name = c.Name,

Address = c.Address,

DOB = c.DOB,

Phone = c.Phone,

Sex = c.Sex,

HireDate = c.HireDate,

Image = c.Image

};

dgvNV.DataSource = result.ToList();

}

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

// Khai báo biến traloi

DialogResult traloi;

// Hiện hộp thoại hỏi đáp

traloi = MessageBox.Show("Chắc không?", "Trả lời",

MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

// Kiểm tra có nhắp chọn nút Ok không?

if (traloi == DialogResult.OK) this.Close();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

FormManage f = new FormManage();

this.Hide();

f.ShowDialog();

}

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

{

Them = true;

resettext();

picEmployee.Image = null;

groupBox1.Enabled = true;

txtMaNV.Enabled = false;

btnCapNhat.Enabled = true;

btnHuy.Enabled = true;

// Không cho thao tác trên các nút Thêm / Xóa / Thoát

btnThem.Enabled = false;

btnEdit.Enabled = false;

btnDel.Enabled = false;

txtHoTen.Focus();

}

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

{

// Kích hoạt biến Sửa

Them = false;

groupBox1.Enabled = true;

// Cho thao tác trên các nút Lưu / Hủy / Panel

btnCapNhat.Enabled = true;

btnHuy.Enabled = true;

// Không cho thao tác trên các nút Thêm / Xóa / Thoát

btnThem.Enabled = false;

btnEdit.Enabled = false;

btnDel.Enabled = false;

// Đưa con trỏ đến TextField

txtMaNV.Enabled = false;

txtHoTen.Focus();

}

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

{

// Xóa trống các đối tượng trong Panel

resettext();

groupBox1.Enabled = false;

// Cho thao tác trên các nút Thêm / Sửa / Xóa / Thoát

btnThem.Enabled = true;

btnEdit.Enabled = true;

btnDel.Enabled = true;

// Không cho thao tác trên các nút Lưu / Hủy / Panel

btnCapNhat.Enabled = false;

btnHuy.Enabled = false;

}

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

{

try

{

// Hiện thông báo xác nhận việc xóa mẫu tin

// Khai báo biến traloi

DialogResult traloi;

// Hiện hộp thoại hỏi đáp

traloi = MessageBox.Show("Chắc xóa mẫu tin này không?", "Trả lời", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// Kiểm tra có nhắp chọn nút Ok không?

if (traloi == DialogResult.Yes)

{

try

{

StoreContext db = new StoreContext();

int idEm = Convert.ToInt32(txtMaNV.Text);

var nvQuery = from em in db.Employees

where em.ID == idEm

select em;

db.Employees.RemoveRange(nvQuery);

db.SaveChanges();

// Cập nhật lại DataGridView

LoadData();

// Thông báo

MessageBox.Show("Đã xóa xong!");

}

catch

{

MessageBox.Show("Không xóa được. Lỗi rồi!!!");

}

}

}

catch (SqlException)

{

MessageBox.Show("Không xóa được. Lỗi rồi");

}

}

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

{

dgvNV.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvNV.AllowUserToAddRows = false;

dgvNV.ReadOnly = true;

LoadData();

//// Không cho thao tác trên các nút Lưu / Hủy

btnCapNhat.Enabled = false;

btnHuy.Enabled = false;

//// Cho thao tác trên các nút Thêm / Sửa / Xóa / Thoát

btnThem.Enabled = true;

btnEdit.Enabled = true;

btnDel.Enabled = true;

groupBox1.Enabled = false;

}

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

{

if (Them)

{

using (StoreContext db = new StoreContext())

{

var employee = new Employee();

employee.Name = txtHoTen.Text;

employee.DOB = dtpDOB.Value.Date;

employee.Address = txtDiaChi.Text;

employee.Phone = txtSoDT.Text;

employee.Sex = (rdbNam.Checked) ? "Nam" : "Nữ";

employee.HireDate = dtpHireDate.Value.Date;

employee.Image = imagepath;

db.Employees.Add(employee);

db.SaveChanges();

}

LoadData();

resettext();

//// Không cho thao tác trên các nút Lưu / Hủy

btnCapNhat.Enabled = false;

btnHuy.Enabled = false;

//// Cho thao tác trên các nút Thêm / Sửa / Xóa / Thoát

btnThem.Enabled = true;

btnEdit.Enabled = true;

btnDel.Enabled = true;

// Thông báo

MessageBox.Show("Đã thêm xong!");

}

else

{

try

{

// Thứ tự dòng hiện hành

int r = dgvNV.CurrentCell.RowIndex;

// MaKH hiện hành

int idEm = Convert.ToInt32(dgvNV.Rows[r].Cells[0].Value.ToString());

// Câu lệnh SQL

StoreContext db = new StoreContext();

var khQuery = (from em in db.Employees

where em.ID == idEm

select em).SingleOrDefault();

if (khQuery != null)

{

khQuery.Name = txtHoTen.Text;

khQuery.DOB = dtpDOB.Value.Date;

khQuery.Address = txtDiaChi.Text;

khQuery.Phone = txtSoDT.Text;

khQuery.Sex = (rdbNam.Checked) ? "Nam" : "Nữ";

khQuery.HireDate = dtpHireDate.Value.Date;

khQuery.Image = imagepath;

db.SaveChanges();

LoadData();

}

// Load lại dữ liệu trên DataGridView

LoadData();

resettext();

groupBox1.Enabled = false;

picEmployee.Image = null;

//// Không cho thao tác trên các nút Lưu / Hủy

btnCapNhat.Enabled = false;

btnHuy.Enabled = false;

//// Cho thao tác trên các nút Thêm / Sửa / Xóa / Thoát

btnThem.Enabled = true;

btnEdit.Enabled = true;

btnDel.Enabled = true;

// Thông báo

MessageBox.Show("Đã sửa xong!");

}

catch

{

MessageBox.Show("Không sửa được. Lỗi rồi!");

}

}

}

private void dgvNV\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

int r = dgvNV.CurrentCell.RowIndex;

this.txtMaNV.Text = dgvNV.Rows[r].Cells["ID"].Value.ToString();

this.txtHoTen.Text = dgvNV.Rows[r].Cells["Name"].Value.ToString();

this.dtpDOB.Text = dgvNV.Rows[r].Cells["DOB"].Value.ToString();

this.txtDiaChi.Text = dgvNV.Rows[r].Cells["Address"].Value.ToString();

this.txtSoDT.Text = dgvNV.Rows[r].Cells["Phone"].Value.ToString();

this.dtpHireDate.Text = dgvNV.Rows[r].Cells["HireDate"].Value.ToString();

if (dgvNV.Rows[r].Cells["Sex"].Value.ToString() == "Nam") rdbNam.Checked = true; else rdbNam.Checked = false;

if (dgvNV.Rows[r].Cells["Sex"].Value.ToString() == "Nữ") rdbNu.Checked = true; else rdbNu.Checked = false;

if (dgvNV.Rows[r].Cells["Image"].Value != "" /\*|| dgvNV.Rows[r].Cells["Image"].Value != null\*/)

{

this.picEmployee.Image = System.Drawing.Image.FromFile(dgvNV.Rows[r].Cells["Image"].Value.ToString());

}

else this.picEmployee.Image = null;

}

catch

{ }

}

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

{

using (OpenFileDialog ofd = new OpenFileDialog() { Filter = "JPEG|\*.jpg" })

{

if (ofd.ShowDialog() == DialogResult.OK)

{

picEmployee.Image = System.Drawing.Image.FromFile(ofd.FileName);

imagepath = ofd.FileName;

}

}

}

### Thiết kế lớp:

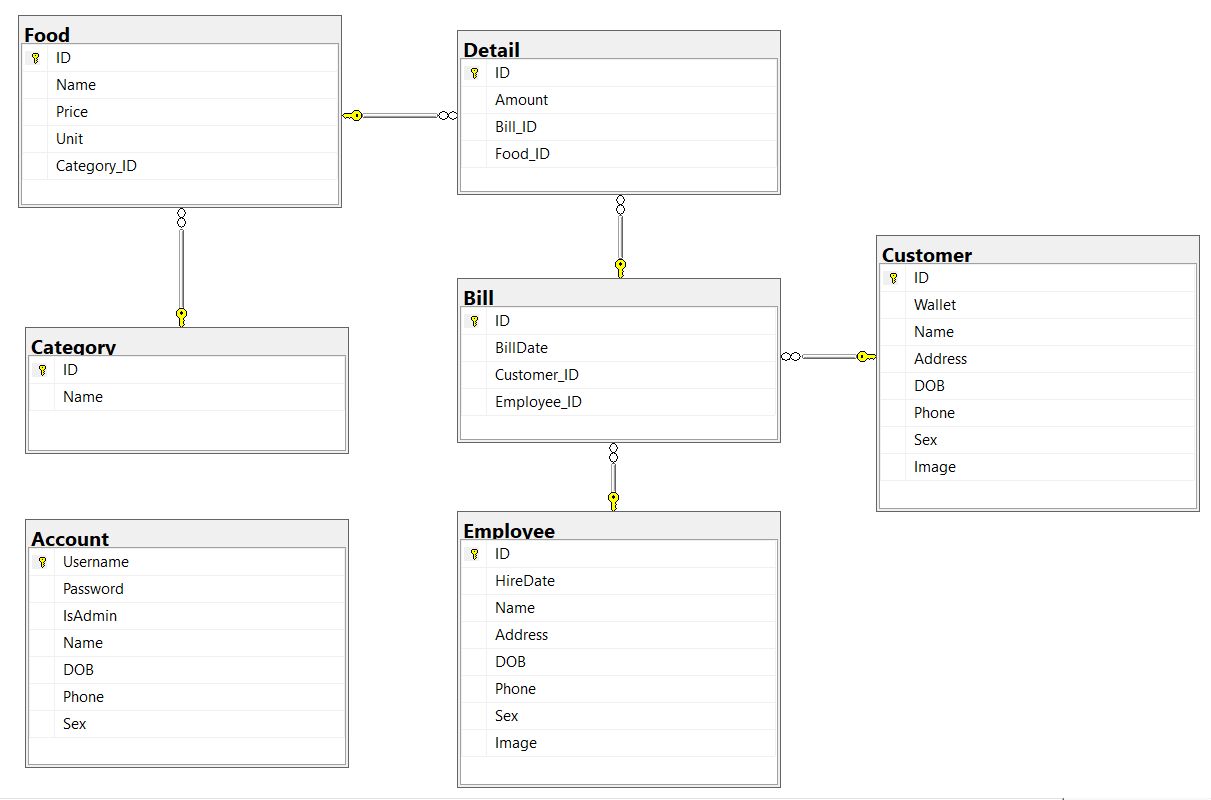
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích chính của lớp trong chương trình** |
| 1 | FormLogin | Lớp hiển thị giao diện xử lý Đăng nhập |
| 2 | Form  FoodChoices | Lớp hiển thị giao diện Chọn món ăn và xuất ra danh mục những món ăn đã chọn, tính toán tổng hóa đơn. |
| 3 | Form  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) |
| 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) |
| 5 | Form  LoginManage | Lớp hiển thị và xử lí quyền trong bảng DangNhap (được liên kết từ Form Manage) |
| 6 | FormManage | Lớp hiển thị giao diện Quản lý dành cho Admin, gồm các menu Quản lý nhân viên, khách hàng, báo cáo ngày, Quản lý đăng nhập  (chỉ admin mới có thể truy cập) |
| 7 | BLMenu | Tầng nghiệp vụ của bảng Food chứa các truy vấn đến Database, thiết lập giao diện tự động khi chạy đến FormFoodChoices. |
| 8 | BLEmployee | Tầng nghiệp vụ của bảng Employee chứa các truy vấn đến Database. |
| 9 | BLCustomer | Tầng nghiệp vụ của bảng Customer chứa các truy vấn đến Database. |
| 10 | BLAccout | Tầng nghiệp vụ của bảng Account chứa các truy vấn đến Database. |
|  |  |  |
|  |  |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Phương thức** | **Mục đích** | **Tên file** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

### Thiết kế cơ sở dữ liệu:



*Hình 3: Cơ sở dữ liệu của hệ thống*

|  |  |  |
| --- | --- | --- |
| **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 |
| 8 | REPORTDATE | Thuộc tính thống kê số Hóa Đơn theo ngày |

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

### Bảng mô tả các Field trong Table:

##### 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 |

##### 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 |

##### 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 |

##### 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 |

##### 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 |

##### 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 |

##### 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:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Màn hình/Cửa sổ/Dialog** | **Giải thích ngắn gọn** | **Mục đích chính của màn hình** |
| 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 |

*Bảng 6: Thiết kế giao diện*

## IV. Cài đặt và kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tình huống** | **Mục đích** | **Giải thích** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

# PHẦN KẾT LUẬN

## 1. Ưu điểm, nhược điểm:

### a. Ưu điểm

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

- Đã ứng dụng được kiến thức Kế thừa vào thiết kế Database (lớp Employee và lớp Customer là lớp con của lớp Peson)

### b. Nhược điểm

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

## 2. 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:

+ Chúng em chưa rõ cách áp dụng triệt để Kế thừa vào bài quản lý

+ 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.

## 3. 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.

## 4. 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

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