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

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**------ oOo ------**



**BÁO CÁO ĐỒ ÁN 1**

**Quản Lý Quán Coffee**

**GVHD : TS.Lê Văn Vinh**

**SVTH : Nguyễn Đức Nam - 15110086**

**Nguyễn Chí Hiếu - 15110046**

**Tp. Hồ Chí Minh, tháng 12 năm 2017**

Contents

[**1.Khảo sát đề tài:** 1](#_Toc501058237)

[**2.Mục tiêu đề tài:** 6](#_Toc501058238)

[**3.Mô tả đề tài:** 6](#_Toc501058239)

[**3.1.Đặc tả đề tài:** 6](#_Toc501058240)

[**3.2.Phạm vi đề tài:** 6](#_Toc501058241)

[**3.3.Chức năng:** 6](#_Toc501058242)

[**4.Sơ đồ ERD:** 7](#_Toc501058243)

[**5.Lược đồ cơ sở dữ liệu:** 7](#_Toc501058244)

[**6. Công nghệ:** 8](#_Toc501058245)

[**6.1.Giao diện:** 8](#_Toc501058246)

[**6.2.Code:** 9](#_Toc501058247)

[**7.Mô tả đánh giá công việc** 26](#_Toc501058248)

[**Tài liệu tham khảo** 26](#_Toc501058249)

## **1.Khảo sát đề tài:**

* Theo khảo sát của nhóm về vài quán cà phê nhỏ trên địa bàn thành phố Hồ Chí Minh, chắc chắn chủ quán sẽ phải thường xuyên đối mặt với những khó khăn như: Kiểm soát nhân viên, báo cáo doanh số hằng ngày, chất lượng phục vụ khách hàng,…Trong đó kiểm soát hóa đơn là lỗi lo thường trực của các chủ quán. Việc kiểm soát được hóa đơn sẽ giúp việc quản lý quán cà phê trở nên dễ dàng hơn, chủ động về nguồn vốn, tránh để nhân viên gây thất thoát.
* Hai vấn đề lớn trong khẩu quản lý hóa đơn:
  + - Thêm, sửa đồ uống được gọi ở mỗi bàn.
    - Kiểm soát được số tiền của mỗi hóa đơn, tránh được nhân viên gian lận hoặc tính tiền nhưng bỏ túi riêng.
* Vấn đề về nguồn nhập nguyên liệu cho quán:
  + - Thêm nguyên liệu mới và nhà sản cung cấp mới cho quán khi quán hết nguyên liệu.
    - Kiểm soát được tiền nhập nguyên liệu hằng tháng để biết tháng đó chi bao nhiêu về phần nhập hàng.
* Vấn đề quản lý nhân viên của người quản lý:
  + - Kiểm soát thời gian làm việc của nhân viên, số ngày nghỉ, số ca làm thêm,...
    - Dựa vào sự chuyên cần của nhân viên mà hỗ trợ tiền thưởng mỗi tháng,...

## **2.Mục tiêu đề tài:**

Quản lý các hoạt động của quán cafe:

* Nhập nguyên liệu
* Bán hàng
* Quản lý nhân viên
* Thống kê hàng ngày, hàng tuần

## **3.Mô tả đề tài:**

## **3.1.Đặc tả đề tài:**

Ông A có nhu cầu mở 1 quán kinh doanh cafe ở địa bàn Thủ Đức, quán Cafe ông quản lý gồm có 15 bàn, mỗi bàn được phân biệt bởi mã bàn, mỗi bàn thuộc một khu vực nhất định.Bộ phận nhân viên của quán gồm có:Nhân viên phục vụ, nhân viên pha chế, người quản lý và nhân viên giữ xe, mỗi nhân viên phân biệt với nhau bằng mã nhân viên .Có tất cả 3 ca làm (ca 1 : làm vào buổi sáng, ca 2 làm vào buổi chiều, ca 3 là cả ngày) .Tiền lương ,tiền thưởng nhân viên mỗi tháng được người quản lý quy định(Lương mỗi loại nhân viên, loại ca làm, số ngày nghỉ,....).Quán nhập nguyên liệu từ các nhà cung cấp trên địa bàn tp HCM hoặc các địa bàn lân cận để chế biến thành phẩm(mỗi lần nhập hàng được người quản lý kiểm tra số lượng, đơn giá, đạt chuẩn hay không,...). Đồ uống của quán được chia làm 8 loại (trà, coffee, sữa, giải khát, sinh tố, trái cây, nước ép, đồ lạnh), mỗi danh mục có nhiều món khác nhau giá của mỗi món được chủ quán quy định. Hóa đơn của quán sẽ lấy thông tin: ngày lập hóa đơn để tiện cho việc thống kê, trạng thái hóa đơn (chưa thanh toán hay đã thanh toán), các đồ uống đã gọi, số lượng từng đồ uống và tổng tiền của hóa đơn.

## **3.2.Phạm vi đề tài:**

* Phần mềm sẽ giúp người quản lý quán quản lý hết 3 hoạt động lớn của quán:nhập hàng, bán hàng và quản lý nhân viên.
* Phần mềm được xây dựng trên ngôn ngữ lập trình C# và quản lý cơ sở dữ liệu trên Microsoft SQL server.

## **3.3.Chức năng:**

Các chức năng chính :

* Nhập nguyên liệu
* Thêm, xóa (nguyên liệu, ngày nhập, thông tin nhà cung cấp,...)
* Trả lại nguyên liệu không đạt chuẩn
* Bán hàng
  + Khách hàng khi vào bàn nhân viên phục vụ sẽ đến đưa menu để khách chọn món và ghi lại thông tin đồ uống khách đã chọn và nhập thông tin đồ uống, mã số của mình, tên bàn vào phần mềm sau đó nhấn đặt bàn. Sau khi đặt bàn thì bàn sẽ chuyển từ màu trắng (trạng thái trống) sang màu vàng (trạng thái bận).
  + Sau khi khách hàng yêu cầu tính tiền, nhân viên sẽ nhấn vào nút thanh toán để xuất tổng tiền đồ uống của bàn và trả bàn từ màu vàng (trạng thái bận) về lại màu trắng (trạng thái trống).
* Quản lý nhân viên
  + Thêm, xóa, sửa (thông tin nhân viên, tiền lương, ca làm, số ngày nghỉ, ngày vào làm,...)
  + Thống kê lương nhân viên, số ngày nghỉ, ca làm ( thống kê theo tháng và thưởng thêm cho những nhân viên có số ngày nghỉ ít nhất, thưởng cho nhân viên làm full-time,...)
* Thống kê mọi hoạt động thu chi của quán.

## **4.Sơ đồ ERD:**

## **5.Lược đồ cơ sở dữ liệu:**

NHANVIEN(IDNhanVien,HoVaTen,DiaChi,GioiTinh,SDT,NgayLam,TienLuong,CaLam,NgayNghi,Hinh,ChucVu,Tuoi)

HOADON(IDHoaDon,IDBan,IDNhanVien,TrangThai)

BAN(IDBan,Ten,KhuVuc,Status)

CHITIETHOADON(IDChiTiet,IDHoaDon,IDDoUong,SoLuong)

DOUONG(IDDoUong,IDDanhMuc,GiaBan,Ten)

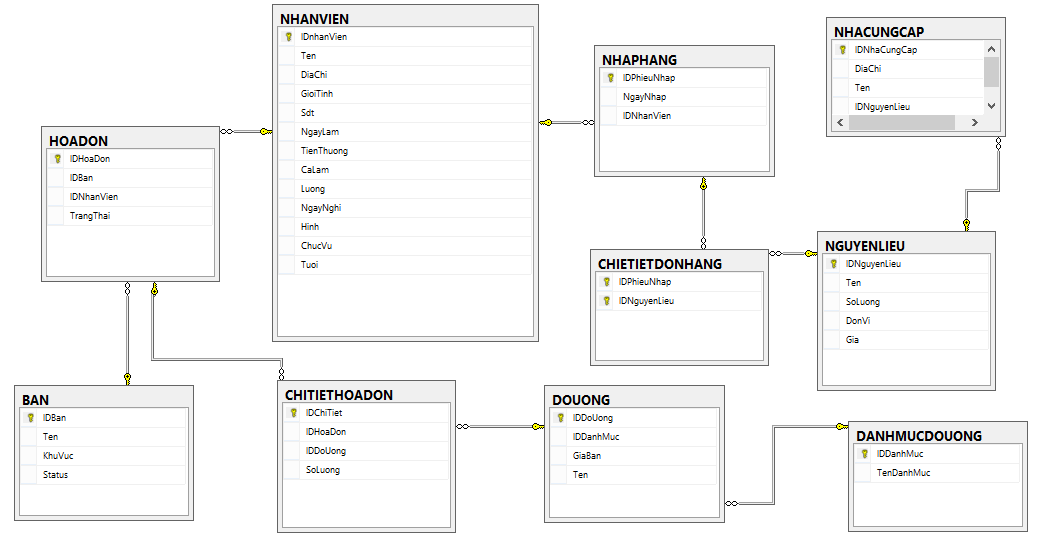
DANHMUCDOUONG(IDDanhMuc,TenDanhMuc)

NHAPHANG(IDPhieuNhap,NgayNhap,IDNhanVien)

CHITIETDONHANG(IDPhieuNhap,IDNguyenLieu)

NHACUNGCAP(IDNhaCungCap,DiaChi,Ten,IDNguyenLieu)

NGUYENLIEU(IDNguyenLieu,Ten,SoLuong,DonVi,Gia)

****

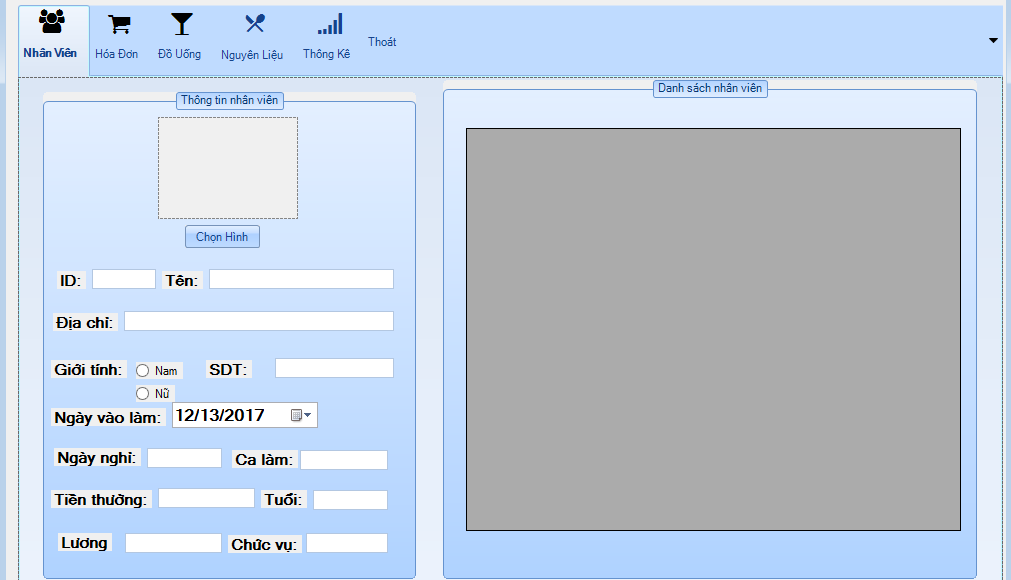
## **6. Công nghệ:**

Làm bằng mô hình 3 Layer

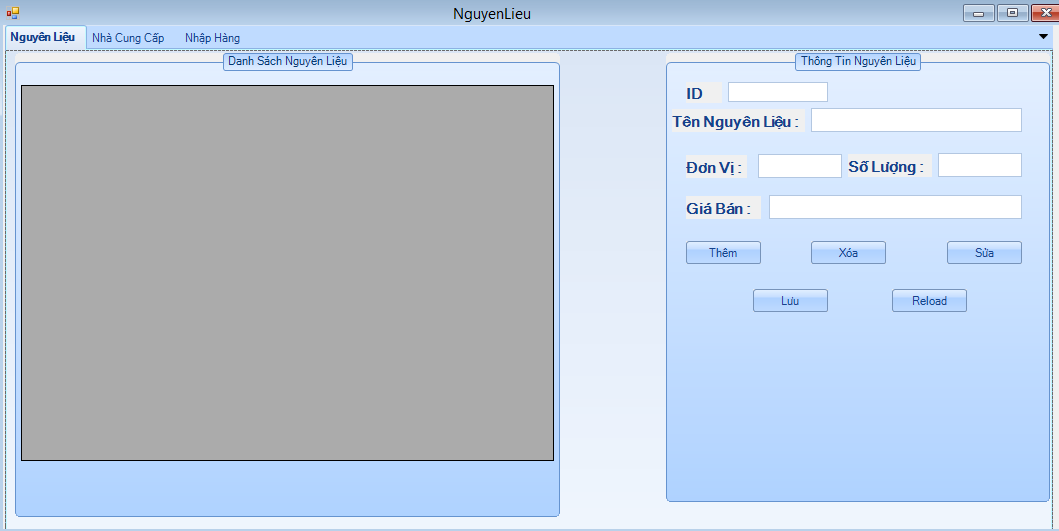
## **6.1.Giao diện:**

Có 2 form chính :

* Form hiển thị (Nhân Viên, Đồ Uống, Hóa Đơn, Nguyên Liệu, Thống Kê)



* Form chi tiết phần Nguyên Liệu



## **6.2.Code:**

* Lớp BALayer dùng để chứa các hàm và thủ tục
* DBBill

namespace BALayer

{

public class DBBill

{

DAL db = null;

public DBBill()

{

db = new DAL();

}

public DataSet LayBill(int id)

{

return db.ExecuteQueryDataSet("SELECT \* FROM HOADON WHERE IDBan = " + id + " AND TrangThai = 'Chua Thanh Toan'", CommandType.Text, null);

}

public bool InsertBill(ref string err, int id,int idNhanVien,DateTime NgayLapHoaDon)

{

return db.MyExecuteNonQuery("SpThemHoaDon",

CommandType.StoredProcedure, ref err,

new SqlParameter("@IDBan ", id),

new SqlParameter("@IDNhanVien", idNhanVien),

new SqlParameter("@NgayLapHoaDon", NgayLapHoaDon.Date)

);

}

public int GetMaxIDBill()

{

try

{

return (int)db.ExecuteScalar("SELECT MAX(IDHoaDon) FROM HOADON");

}

catch

{

return 1;

}

}

public bool updateBill(ref string err, int id/\*, float totalPrice\*/)

{

return db.MyExecuteNonQuery("SpCapNhatHoaDon",

CommandType.StoredProcedure, ref err,

new SqlParameter("@ID", id)

);

}

}

* DBBilllnfo

public class DBBillInfo

{

DAL db = null;

public DBBillInfo()

{

db = new DAL();

}

public DataSet LayBillInfo(int id)

{

return db.ExecuteQueryDataSet("SELECT \* FROM CHITIETHOADON WHERE IDHoaDon = " + id, CommandType.Text, null);

}

public bool thembliiInfo(ref string err, int idBill, int idFood, int count/\*,DateTime NgayLap\*/)

{

return db.MyExecuteNonQuery("SpThemChiTietHoaDon",

CommandType.StoredProcedure, ref err,

new SqlParameter("@idBill ", idBill),

new SqlParameter("@idFood", idFood),

new SqlParameter("@count", count)

);

}

* DBCategory

public class DBCategory

{

DAL db = null;

public DBCategory()

{

db = new DAL();

}

public DataSet LayCategory()

{

return db.ExecuteQueryDataSet("select \* from DANHMUCDOUONG", CommandType.Text, null);

}

}

* DBDoUong

public class DBDoUong

{

DAL db = null;

public DBDoUong()

{

db = new DAL();

}

public DataSet LayDoUong(int IDDanhMuc)

{

return db.ExecuteQueryDataSet("select \* from DOUONG where IDDanhMuc=" + IDDanhMuc, CommandType.Text, null);

}

public bool ThemDoUong(ref string err,

int ID\_DanhMuc,

float GiaBan, String Ten)

{

return db.MyExecuteNonQuery("SpThemDoUong", CommandType.StoredProcedure, ref err,

new SqlParameter("@ID\_DanhMuc", ID\_DanhMuc),

new SqlParameter("@GiaBan", GiaBan),

new SqlParameter("@Ten", Ten));

}

public bool XoaDoUong(ref string err, int ID\_DoUong)

{

return db.MyExecuteNonQuery("spXoaDoUong",

CommandType.StoredProcedure, ref err, new SqlParameter("@ID\_DoUong", ID\_DoUong));

}

public bool CapNhatDoUong(ref string err,

int ID\_DoUong,/\*int ID\_DanhMuc,\*/float GiaBan, string Ten)

{

return db.MyExecuteNonQuery("SpCapNhatDoUong",

CommandType.StoredProcedure, ref err,

new SqlParameter("@IDDoUong", ID\_DoUong),

new SqlParameter("@GiaBan", GiaBan),

new SqlParameter("@Ten", Ten)

);

}

* DBDrink

public class DBDrink

{

DAL db = null;

public DBDrink()

{

db = new DAL();

}

public DataSet LayDrink(int id)

{

return db.ExecuteQueryDataSet("select \* from DOUONG where IDDanhMuc = " + id, CommandType.Text, null);

}

}

* DBMenu

public class DBMenu

{

DAL db = null;

public DBMenu()

{

db = new DAL();

}

public DataSet LayMenu(int id)

{

return db.ExecuteQueryDataSet("select GiaBan,DOUONG.Ten as Ten, SoLuong\*GiaBan as totalPrice,SoLuong, SoLuong\*GiaBan as thanhtien,NgayLapHoaDon from HOADON, CHITIETHOADON, BAN, DOUONG where BAN.IDBan = HOADON.IDBan and HOADON.IDHoaDon = CHITIETHOADON.IDHoaDon and HOADON.IDBan = " + id+ " and CHITIETHOADON.IDDoUong = DOUONG.IDDoUong and TrangThai = 'Chua Thanh Toan'", CommandType.Text, null);

}

}

* DBNguyenLieu

public class DBNguyenLieu

{

DAL db = null;

public DBNguyenLieu()

{

db = new DAL();

}

public DataSet LayNguyenLieu()

{

return db.ExecuteQueryDataSet("spLayInfoNguyenLieu", CommandType.Text, null);

}

public DataSet LayThongTinNguyenLieu()

{

return db.ExecuteQueryDataSet("spLayNguyenLieu", CommandType.Text, null);

}

public bool ThemNguyenLieu(ref string err,

string Ten, float SoLuong,string DonVi,float Gia )

{

return db.MyExecuteNonQuery("SpThemNguyenLieu",

CommandType.StoredProcedure, ref err,

new SqlParameter("@Ten ", Ten),

new SqlParameter("@SoLuong", SoLuong),

new SqlParameter("@DonVi", DonVi),

new SqlParameter("@Gia ", Gia));

}

public bool CapNhatNguyenLieu(ref string err,

/\*int IDNguyenLieu,\*/string Ten, float SoLuong, string DonVi, float Gia)

{

return db.MyExecuteNonQuery("SpCapNhatNguyenLieu",

CommandType.StoredProcedure, ref err,

//new SqlParameter("@IDNguyenLieu", IDNguyenLieu),

new SqlParameter("@Ten", Ten),

new SqlParameter("@SoLuong", SoLuong),

new SqlParameter("@DonVi", DonVi),

new SqlParameter("@Gia", Gia));

}

public bool XoaNguyenLieu(ref string err,int IDNguyenLieu)

{

return db.MyExecuteNonQuery("spXoaNguyenLieu",

CommandType.StoredProcedure, ref err, new SqlParameter("@IDNguyenLieu", IDNguyenLieu));

}

}

* DBNhaCungCap

public class DBNhaCungCap

{

DAL db = null;

public DBNhaCungCap()

{

db = new DAL();

}

public DataSet LayNhaCugnCap()

{

return db.ExecuteQueryDataSet("spLayInfoNhaCungCap", CommandType.Text, null);

}

public bool ThemNhaCungCap(ref string err,

string Ten, int IDNguyenLieu, string DiaChi)

{

return db.MyExecuteNonQuery("SpThemNhaCungCap",

CommandType.StoredProcedure, ref err,

new SqlParameter("@Ten ", Ten),

new SqlParameter("@DiaChi", DiaChi),

new SqlParameter("@IDNguyenLieu", IDNguyenLieu));

}

public bool CapNhatNhaCungCap(ref string err,

string TenNCC, int IDNguyenLieu, string DiaChi)

{

return db.MyExecuteNonQuery("SpCapNhatNCC",

CommandType.StoredProcedure, ref err,

//new SqlParameter("@IDNguyenLieu", IDNguyenLieu),

new SqlParameter("@TenNCC", TenNCC),

new SqlParameter("@DiaChi", DiaChi),

new SqlParameter("@IDNguyenLieu", IDNguyenLieu));

}

public bool XoaNhaCungCap(ref string err, int IDNhaCungCap)

{

return db.MyExecuteNonQuery("spXoaNhaCungCap",

CommandType.StoredProcedure, ref err, new SqlParameter("@IDNhaCungCap", IDNhaCungCap));

}

}

* DBNhanVien

public class DBNhanVien

{

DAL db = null;

public DBNhanVien()

{

db = new DAL();

}

public DataSet LayNhanVien()

{

return db.ExecuteQueryDataSet("select \* from NHANVIEN", CommandType.Text, null);

}

public bool ThemNhanVien(ref string err,

string Ten, string DiaChi, string GioiTinh, int SDT, DateTime Ngay,float TienThuong,

string CaLam,float Luong,int NgayNghi,byte[] Hinh,string ChucVu,int Tuoi)

{

return db.MyExecuteNonQuery("SpThemNhanVien",

CommandType.StoredProcedure, ref err,

new SqlParameter("@Ten ", Ten),

new SqlParameter("@GioiTinh", GioiTinh),

new SqlParameter("@DiaChi", DiaChi),

new SqlParameter("@Ngay ", Ngay),

new SqlParameter("@SDT", SDT),

new SqlParameter("@TienThuong", TienThuong),

new SqlParameter("@CaLam ", CaLam),

new SqlParameter("@Luong ", Luong),

new SqlParameter("@NgayNghi ", NgayNghi),

new SqlParameter("@ChucVu ", ChucVu),

new SqlParameter("@Tuoi ", Tuoi),

new SqlParameter("@Hinh", Hinh));

}

public bool XoaNhanVien(ref string err, int ID\_NhanVien)

{

return db.MyExecuteNonQuery("spXoaNhanVien",

CommandType.StoredProcedure, ref err, new SqlParameter("@ID\_NhanVien", ID\_NhanVien));

}

public bool CapNhatNhanVien(ref string err, int ID\_NhanVien, string Ten, string DiaChi, string GioiTinh, int SDT, DateTime Ngay, float TienThuong,

string CaLam, float Luong, int NgayNghi, byte[] Hinh, string ChucVu, int Tuoi)

{

return db.MyExecuteNonQuery("SpCapNhatNhanVien",

CommandType.StoredProcedure, ref err,

new SqlParameter("@ID\_NhanVien", ID\_NhanVien),

new SqlParameter("@DiaChi", DiaChi),

new SqlParameter("@Ten", Ten),

new SqlParameter("@GioiTinh", GioiTinh),

new SqlParameter("@Ngay", Ngay),

new SqlParameter("@SDT", SDT),

new SqlParameter("@TienThuong", TienThuong),

new SqlParameter("@CaLam ", CaLam),

new SqlParameter("@Luong ", Luong),

new SqlParameter("@NgayNghi ", NgayNghi),

new SqlParameter("@ChucVu ", ChucVu),

new SqlParameter("@Tuoi ", Tuoi),

new SqlParameter("@Hinh", Hinh));

}

}

* DBNhapHang

public class DBNhapHang

{

DAL db = null;

public DBNhapHang()

{

db = new DAL();

}

public DataSet LayThongTinNhapHang()

{

return db.ExecuteQueryDataSet("select \* from NHAPHANG", CommandType.Text, null);

}

public DataSet LayThongTinChiTietNhapHang()

{

return db.ExecuteQueryDataSet("select \* from CHIETIETDONHANG", CommandType.Text, null);

}

public bool ThemNhapHang(ref string err,

int IDPhieuNhap, DateTime NgayNhap, int IDNhanVien)

{

return db.MyExecuteNonQuery("SpThemNhapHang",

CommandType.StoredProcedure, ref err,

new SqlParameter("@IDPhieuNhap ", IDPhieuNhap),

new SqlParameter("@NgayNhap", NgayNhap),

new SqlParameter("@IDNhanVien", IDNhanVien));

}

public bool XoaNhanVien(ref string err, int IDPhieuNhap)

{

return db.MyExecuteNonQuery("spXoaNhapHang",

CommandType.StoredProcedure, ref err, new SqlParameter("@IDPhieuNhap", IDPhieuNhap));

}

public bool CapNhatNhapHang(ref string err,

int IDPhieuNhap, DateTime NgayNhap, int IDNhanVien)

{

return db.MyExecuteNonQuery("SpCapNhatNhanVien",

CommandType.StoredProcedure, ref err,

new SqlParameter("@IDPhieuNhap ", IDPhieuNhap),

new SqlParameter("@NgayNhap", NgayNhap),

new SqlParameter("@IDNhanVien", IDNhanVien));

}

}

* DBTable

public class DBTable

{

DAL db = null;

public DBTable()

{

db = new DAL();

}

public DataSet LayNhanVien()

{

return db.ExecuteQueryDataSet("select \* from BAN", CommandType.Text, null);

}

public void CapNhatSauThanhToan (int id)

{

db.ExecuteQueryDataSet("EXEC dbo.SpCapNhatBan "+id

, CommandType.Text, null);

}

public void CapNhatTruocThanhToan(int id)

{

db.ExecuteQueryDataSet("EXEC dbo.SpCapNhatBanTruocTT " + id

, CommandType.Text, null);

}

}

* Lớp BDLayer dùng để kết nối tới cơ sở dữ liệu
  + DAL

public class DAL

{

string ConnStr = @"Data Source=.\SQLEXPRESS;" +

"Initial Catalog=coffee;" +

"Integrated Security=True";

SqlConnection conn = null;

SqlCommand comm = null;

SqlDataAdapter da = null;

public DAL()

{

conn = new SqlConnection(ConnStr);

comm = conn.CreateCommand();

}

public DataSet ExecuteQueryDataSet(string strSQL, CommandType ct, params SqlParameter[] p)

{

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

comm.CommandText = strSQL;

comm.CommandType = ct;

da = new SqlDataAdapter(comm);

DataSet ds = new DataSet();

da.Fill(ds);

return ds;

}

public bool MyExecuteNonQuery(string strSQL, CommandType ct,

ref string error, params SqlParameter[] param)

{

bool f = false;

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

comm.Parameters.Clear();

comm.CommandText = strSQL;

comm.CommandType = ct;

foreach (SqlParameter p in param)

comm.Parameters.Add(p);

try

{

comm.ExecuteNonQuery();

f = true;

}

catch (SqlException ex)

{

error = ex.Message;

}

finally

{

conn.Close();

}

return f;

}

public object ExecuteScalar(string query, object[] parameter = null)

{

object data = 0;

using (SqlConnection connection = new SqlConnection(ConnStr))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

data = command.ExecuteScalar();

connection.Close();

}

return data;

}

}

* Lớp Form dùng để gọi và thực thi các hàm và thủ tục
  + Lớp DAO
    - BillDAO

public class BillDAO

{

DBBill dbBill = new DBBill();

public int GetUncheckBillIDByTableID(int id)

{

DataTable data = dbBill.LayBill(id).Tables[0];

if (data.Rows.Count > 0)

{

Bill bill = new Bill(data.Rows[0]);

return bill.IDHoaDon;

}

return -1;

}

}

* + - Billlnfo

public class BillInfoDAO

{

public List<BillInfo> GetListBillInfo(int id)

{

DBBillInfo db = new DBBillInfo();

List<BillInfo> listBillInfo = new List<BillInfo>();

DataTable data = db.LayBillInfo(id).Tables[0];

foreach (DataRow item in data.Rows)

{

BillInfo info = new BillInfo(item);

listBillInfo.Add(info);

}

return listBillInfo;

}

}

* + - CategoryDAO

public class CategoryDAO

{

public List<Category> GetListCategory()

{

List<Category> list = new List<Category>();

DBCategory db = new DBCategory();

DataTable data = db.LayCategory().Tables[0];

foreach (DataRow item in data.Rows)

{

Category category = new Category(item);

list.Add(category);

}

return list;

}

}

* + - DrinkDAO

public class DrinkDAO

{

public List<Drink> GetFoodByCategoryID(int id)

{

DBDrink db = new DBDrink();

List<Drink> list = new List<Drink>();

DataTable data = db.LayDrink(id).Tables[0];

foreach (DataRow item in data.Rows)

{

Drink food = new Drink(item);

list.Add(food);

}

return list;

}

}

* + - MenuDAO

class MenuDAO

{

public List<Menu> GetListMenuByTable(int id)

{

DBMenu db = new DBMenu();

List<Menu> listMenu = new List<Menu>();

DataTable data = db.LayMenu(id).Tables[0];

foreach (DataRow item in data.Rows)

{

Menu menu = new Menu(item);

listMenu.Add(menu);

}

return listMenu;

}

}

* + - TableDAO

public class TableDAO

{

DBTable dbTable = new DBTable();

public static int TableWidth = 100;

public static int TableHeight = 100;

public List<Table> LoadTableList()

{

List<Table> tableList = new List<Table>();

DataTable data = dbTable.LayNhanVien().Tables[0];

foreach (DataRow item in data.Rows)

{

Table table = new Table(item);

tableList.Add(table);

}

return tableList;

}

}

* + Lớp DTO
    - Bill

public class Bill

{

public Bill(int idhoadon,int idban,int idnhanvien, string trangthai,DateTime ngaylaphoadon){

this.IDHoaDon = idhoadon;

this.IDBan = idban;

this.IDNhanVien = idnhanvien;

this.IDTrangThai = trangthai;

this.Ngaylaphoadon = ngaylaphoadon;

}

public Bill(DataRow row)

{

this.IDBan = (int)row["IDBan"];

this.IDHoaDon = (int)row["IDHoaDon"];

this.IDNhanVien= (int)row["IDNhanVien"];

this.IDTrangThai = row["TrangThai"].ToString();

this.Ngaylaphoadon = DateTime.Parse( row["NgayLapHoaDon"].ToString());

}

private int iDBan;

private int iDHoaDon;

private DateTime ngaylaphoadon;

public DateTime Ngaylaphoadon

{

get

{

return ngaylaphoadon;

}

set

{

ngaylaphoadon = value;

}

}

public int IDBan

{

get

{

return iDBan;

}

set

{

iDBan = value;

}

}

public int IDHoaDon

{

get

{

return iDHoaDon;

}

set

{

iDHoaDon = value;

}

}

public string IDTrangThai

{

get

{

return iDTrangThai;

}

set

{

iDTrangThai = value;

}

}

public int IDNhanVien

{

get

{

return iDNhanVien;

}

set

{

iDNhanVien = value;

}

}

private int iDNhanVien;

private string iDTrangThai;

}

* + - Billlnfo

public class BillInfo

{

public BillInfo(int idchitiet, int idHoadon, int idDouong, int soLuong,DateTime NgayLap)

{

this.IDChiTiet = idchitiet;

this.IDHoaDon = idHoadon;

this.IDDoUong = idDouong;

this.SoLuong = soLuong;

this.NgayLap = NgayLap;

}

public BillInfo(DataRow row)

{

this.IDChiTiet = (int)row["IDChiTiet"];

this.IDHoaDon = (int)row["IDHoaDon"];

this.IDDoUong = (int)row["IDDoUong"];

this.SoLuong = (int)row["SoLuong"];

this.NgayLap = (DateTime)row["NgayLapHoaDon"];

}

private int iDChiTiet;

public int IDChiTiet

{

get

{

return iDChiTiet;

}

set

{

iDChiTiet = value;

}

}

public int IDHoaDon

{

get

{

return iDHoaDon;

}

set

{

iDHoaDon = value;

}

}

public int IDDoUong

{

get

{

return iDDoUong;

}

set

{

iDDoUong = value;

}

}

public int SoLuong

{

get

{

return soLuong;

}

set

{

soLuong = value;

}

}

public DateTime Ngaylap

{

get

{

return NgayLap;

}

set

{

NgayLap = value;

}

}

private int iDHoaDon;

private int iDDoUong;

private int soLuong;

private DateTime NgayLap;

}

* + - Category

public class Category

{

public Category(int id, string name)

{

this.ID = id;

this.Name = name;

}

public Category(DataRow row)

{

this.ID = (int)row["IDDanhMuc"];

this.Name = row["TenDanhMuc"].ToString();

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private int iD;

public int ID

{

get { return iD; }

set { iD = value; }

}

}

* + - Drink

public class Drink

{

public Drink(int id, int categoryID, float price, string name )

{

this.ID = id;

this.Name = name;

this.CategoryID = categoryID;

this.Price = price;

}

public Drink(System.Data.DataRow row)

{

this.ID = (int)row["IDDoUong"];

this.Name = row["Ten"].ToString();

this.CategoryID = (int)row["IDDanhMuc"];

this.Price = (float)Convert.ToDouble(row["GiaBan"].ToString());

}

private float price;

public float Price

{

get { return price; }

set { price = value; }

}

private int categoryID;

public int CategoryID

{

get { return categoryID; }

set { categoryID = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private int iD;

public int ID

{

get { return iD; }

set { iD = value; }

}

}

* + - Menu

class Menu

{

public Menu(string foodName, int count, float price, DateTime ngaylap, float totalPrice=0)

{

this.FoodName = foodName;

this.Count = count;

this.Price = price;

this.TotalPrice = totalPrice;

this.NgayLap = ngaylap;

}

public Menu(DataRow row)

{

this.FoodName = row["Ten"].ToString();

// this.FoodName = (row["totalPrice"].ToString());

this.Count = (int)row["SoLuong"];

this.Price = (float)Convert.ToDouble(row["GiaBan"].ToString());

this.TotalPrice = (float)Convert.ToDouble(row["totalPrice"].ToString());

this.NgayLap = DateTime.Parse(row["NgayLapHoaDon"].ToString());

}

private float totalPrice;

public float TotalPrice

{

get { return totalPrice; }

set { totalPrice = value; }

}

private float price;

public float Price

{

get { return price; }

set { price = value; }

}

private int count;

public int Count

{

get { return count; }

set { count = value; }

}

private string foodName;

public string FoodName

{

get { return foodName; }

set { foodName = value; }

}

private DateTime ngaylap;

public DateTime NgayLap

{

get { return ngaylap; }

set { ngaylap = value; }

}

}

* + - Table

public class Table

{

public Table(int id, string name, string status,string khuvuc)

{

this.ID = id;

this.Name = name;

this.Status = status;

this.KhuVuc = khuVuc;

}

public Table(DataRow row)

{

this.ID = (int)row["IDBan"];

this.Name = row["Ten"].ToString();

this.KhuVuc = row["KhuVuc"].ToString();

this.Status = row["status"].ToString();

}

private string khuVuc;

public string KhuVuc

{

get { return khuVuc; }

set { khuVuc = value; }

}

private string status;

public string Status

{

get { return status; }

set { status = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private int iD;

public int ID

{

get { return iD; }

set { iD = value; }

}

}

## **7.Mô tả đánh giá công việc**

Nguyễn Chí Hiếu – 15110046 :Đóng góp 60%(Thiết kế CSDL, Code phần DoUong,NguyenLieu,tìm tài liệu cho Nam)

Nguyễn Đức Nam – 15110... :Đóng góp 100%(Thiết kế CSDL, Code phần NhanVien,HoaDon,ThongKe)

# **Tài liệu tham khảo**