Bai 6: Tạo chức năng đăng nhập

-ExcuteNonquery chỉ trả về số dòng của các chức năng insert, delete, update

-Nếu muốn kiểu tra mật khẩu đăng nhập có đúng không thì phải kiểm tra bằng cách trả về 1 Datatable rồi đếm xem trong Datatable có bao nhiêu dòng, nêu > 0 thì đăng nhập thành công9vì tài khoản và mật khẩu đã nhập trùng với 1 dòng trong cơ sở dữ liệu!)

Bài 7: hạn chế lỗi SQL injection

-lỗi đăng nhập mà không cần tài khoản mật khẩu:

C1: Sử dụng stored procedure trong sql

Cach tao procedure:

create proc usp\_login --user’s procedure

@username nvarchar(100),

@password nvarchar(100)

as

begin

Select \* from dbo.account where @username = username and @password = password

End

Bai 8: Hien thi danh sach ban an

TableDAO: Xử lý lấy dữ liệu từ CSDL lên, tạo 1 đối tượng static instance để lớp view gọi đến

Có 1 hàm loadTableList để load danh sach các table từ csdl lên

Table(trong thư mục DTO): lưu trữ các thuộc tính của đối tượng table

Các bước của phần này:

+Tạo lớp Table với các thuộc tính bình thường và có thêm hàm khởi tạo, lưu ý óc thêm

public Table(DataRow row)

{

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

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

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

}

+tạọ TableDAO và singleton cho nó, sau đó viết thêm hàm loadTableList, mỗi row trong cơ sở dữ liệu sẽ tương ứng với 1 đối tượng table được tạo mới cho lớp Table

public List<Table> loadTableList()

{

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

DataTable data = DataProvider.Instance.ExcecuteQuery("exec usp\_GetTableList");

foreach(DataRow row in data.Rows)

{

Table table = new Table(row);

tablelst.Add(table);

}

return tablelst;

}

+Ở tầng view, Viết hàm loadTableList để gọi thằng loadTableList của thằng TableDAO lên và tạo ra các button tương ứng với mỗi bàn trong DataTable, set kích thước cho button sao cho hợp ní

Bài 9: Hiển thị hóa đơn theo bàn:”

+Tạo các lớp Bill, BillInfo một cách bình thường và có thêm cái constructor trả về kiểu row như ở bài 8.

+Lớp BillDAO có hàm getUncheckedBillIdByTableId() để trả về id bill qua id của cái bàn nào chưa checkout

+Lớp BillInfoDAO có hàm getListBillInfo() để trả về danh sách BillInfo qua Id của những thằng Bill chưa checkout

+Lớp Menu tự tạo có các thuộc tính foodName, cout, price, totalPrice

+MenuDAO trả về menu qua câu lệnh:

select f.Name, bi.count, f.price, f.price \* bi.count as totalPrice from billinfo as bi, Bill as b, Food as f where bi.idBill = b.id and bi.idFood = f.id and b.idTable = "+id

với thằng id kia là id của Bill

+mỗi khi người dung click vào 1 bàn, thì hàm click đó gọi ra hàm ShowBill()

+hàm ShowBill() sẽ clear cái bill cũ và hiện ra 1 Bill mới

void ShowBill(int id)

{

lsvBill.Items.Clear();

List<DTO.Menu> lstMenu = MenuDAO.Instance.getListMenuByTable(id);

foreach(Menu item in lstMenu)

{

ListViewItem lstvItem = new ListViewItem(item.FoodName.ToString());

lstvItem.SubItems.Add(item.Count.ToString());

lstvItem.SubItems.Add(item.Price.ToString());

lstvItem.SubItems.Add(item.TotalPrice.ToString());

lsvBill.Items.Add(lstvItem);

}

}

Bài 10: Hiển thị tổng tiền

Cộng tất cả các item.totalPrice trong hàm ShowBill() là xong

Bài 11: Thêm bới món ăn vào hóa đơn

Trước hết, tạo các lớp Category, CategoryDAO, Food, FoodDAO

Trong CategoryDAO có hàm getListCategory() siêu dễ

Trong FoodDAO có hàm getListFoodByCategoryID(int id) cũng dễ, câu truy vấn là:

String query = "select \* from food where idCategory = " + id;

Tạo thêm ở BillDAO hàm void insertBill()

public void InsertBill(int id)

{

DataProvider.Instance.ExcecuteQuery("exec usp\_InsertBill @idTable", new object[] { id });

}

Trong khi đó phải tạo 1 procedure trong SQL là create proc usp\_insertBill dễ vl

\*Để hiển thị danh sách FoodCategory lên cboCategory khá dễ, khi click vào bàn nào thì sẽ có load 1 hàm loadListCategory()

List<Category> lstCategory = CategoryDAO.Instance.getListCategory();

cboCategory.DataSource = lstCategory;

cboCategory.DisplayMember = "name";

Chúng ta sẽ cần gọi hàm getListCategory từ CategoryDAO lên để trả về 1 danh sách các category sau đó cho datasorce của cái combobox thành cái list đó (ez), chú ý muốn hiển thị tên của category lên combobox thì phải chọn DisplayMember là trường “Name”//

\*Để hiển thị danh sách thức ăn theo trường của chúng là Category đã chọn ở trên thì chúng ta phải code 1 sự kiện mỗi khi thay đổi Category = cách bấm vào Events của Category và chọn cái CboCategory\_SelectedIndexChanged

Khi đó chúng ta mới gọi hàm loadFoodListByCategoryID(int ID);

Sau đó chúng ta sẽ xử lý ở nút Thêm Món

Bài 12: Thanh toán Hóa Đơn

Khi thêm món ăn thì bàn đó trở thành có người

Khi đã thanh toán thì quay trở về là bàn trống 🡪 Đổi status của Bill

+Tạo một hàm CheckOut trong BillDAO

public void CheckOut(int id) //id này là id của Bill

{

String query = "update dbo.bill set status = 1 where id = "+id;

DataProvider.Instance.ExcecuteNonQuery(query);

}

+khi bấm nút thanh toán sẽ xử lý như sau:

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

{

//Lấy table đang chọn

Table table = lsvBill.Tag as Table;

//Lấy ra idBill theo table đó

int idBill = BillDAO.Instance.getUnCheckedBillIdByTableId(table.ID);

//nếu không có idBill ở bàn đó thì không thanh toán và ngược lại

if(idBill != -1)

{

if (MessageBox.Show("Bạn có chắc thanh toán hóa đơn cho bàn + " + table.Name , "thông báo!", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

BillDAO.Instance.CheckOut(idBill);

ShowBill(table.ID); //Load lại cái hàm showBill()

}

}

}

+Trong cơ sở dữ liệu tạo 2 trigger update cho Bill và BillInfo

Mỗi khi chúng ta thực hiện Update BillInfo thì thì status sẽ là Có người

Mỗi khi thực hiện update Bill mà bàn không có Bill nào thì status bàn đó sẽ là Trống

* Thực hiện load lại các bàn trên form sao cho nó load lại status và màu sắc mỗi khi thực hiện thêm món hoặc thanh toán cho bàn đó!