Code View sử dụng chung

void LoadTable()

{

flpTable.Controls.Clear();

List<Table> tablelist = TableDAO.Instance.LoadTableList();

foreach (Table item in tablelist)

{

Button btn = new Button() { Width = TableDAO.TableWidth, Height = TableDAO.TableHeight };

btn.Text = item.Name + Environment.NewLine + item.Status;

btn.Click += btn\_Click;

btn.Tag = item;

switch (item.Status)

{

case "Trống":

btn.BackColor = Color.Aqua;

break;

default:

btn.BackColor = Color.LightPink;

break;

}

flpTable.Controls.Add(btn);

}

}

void ShowBill(int id)

{

livBill.Items.Clear();

List<Manu> listBillInfor = MenuDAO.Instance.GetListMenuByTable(id);

float TotalPrice = 0;

foreach (Manu item in listBillInfor)

{

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

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

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

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

TotalPrice += item.TotalPrice;

livBill.Items.Add(lsvItem);

}

CultureInfo culture = new CultureInfo("vi-VN");

txtTotalPrice.Text = TotalPrice.ToString("c", culture);

}

void LoadListFood()

{

foodList.DataSource = FoodDAO.Instance.GetListFood();

}

Code Đăng nhập

**View**

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

{

string userName = txtUserName.Text;

string passWord = txtPassWord.Text;

if (Login(userName, passWord))

{

Account loginAccount = AccountDAO.Instance.GetAccountByUserName(userName);

fTableManager f = new fTableManager(loginAccount);

this.Hide();

f.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("Sai tên tài khoản hoặc mật khẩu!");

}

}

bool Login(string userName, string passWord)

{

return AccountDAO.Instance.Login(userName, passWord);

}

**Controller**

public Account GetAccountByUserName(string userName)

{

DataTable data = DataProvider.Instance.ExecuteQuery("select \* from account where userName = '" + userName + "'");

foreach (DataRow item in data.Rows)

{

return new Account(item);

}

return null;

}

public bool Login(string userName, string passWord)

{

string query = "USP\_Login @userName , @passWord";

DataTable result = DataProvider.Instance.ExecuteQuery(query, new object[] {userName,passWord});

return result.Rows.Count > 0;

}

Code Thanh Toán

**View**

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

{

Table table = livBill.Tag as Table;

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

int discount = (int)nmDisCount.Value;

DateTime today = DateTime.Now;

double totalPrice = Convert.ToDouble(txtTotalPrice.Text.Split(',')[0].Replace(".", ""));

double finalTotalPrice = totalPrice - (totalPrice / 100) \* discount;

if (idBill != -1)

{

if(MessageBox.Show(string.Format("\tNHÀ HÀNG SIU NHÂN\nTHANH TOÁN HÓA ĐƠN BÀN {0} \nTổng tiền : {1} \n Đã giảm giá : {2} % \n Tổng hóa đơn : {3} \n Date: {4} ", table.Name, totalPrice, discount, finalTotalPrice,today), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

BillDAO.Instance.CheckOut(idBill, discount, (float)finalTotalPrice);

ShowBill(table.ID);

LoadTable();

}

}

}

**Controller**

public int GetUncheckBillIDByTableID(int id)

{

DataTable data = DataProvider.Instance.ExecuteQuery("select \* from Bill WHERE idTable= " + id + " AND status = 0 ");

if (data.Rows.Count > 0)

{

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

return bill.ID;

}

return -1;

}

public void CheckOut(int id, int discount, float totalPrice)

{

string query = "UPDATE dbo.Bill SET dateCheckOut = GETDATE(), status = 1, " + "discount = " + discount + ", totalPrice = " + totalPrice + " WHERE id = " + id;

DataProvider.Instance.ExecuteNonQuery(query);

}

Code Thêm món hoặc hủy món

**View**

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

{

Table table = livBill.Tag as Table;

if (table == null)

{

MessageBox.Show("Hãy chọn bàn");

return;

}

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

int foodID = (cbFood.SelectedItem as Food).ID;

int count = (int)nmFoodCount.Value;

if (idBill == -1)

{

BillDAO.Instance.InsertBill(table.ID);

BillInforDAO.Instance.InsertBillInfor(BillDAO.Instance.GetMaxIDBill(), foodID, count);

}

else

{

BillInforDAO.Instance.InsertBillInfor(idBill, foodID, count);

}

ShowBill(table.ID);

LoadTable();

}

**Controller**

public int GetUncheckBillIDByTableID(int id)

{

DataTable data = DataProvider.Instance.ExecuteQuery("select \* from Bill WHERE idTable= " + id + " AND status = 0 ");

if (data.Rows.Count > 0)

{

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

return bill.ID;

}

return -1;

}

public void InsertBillInfor(int idBill, int idFood, int count)

{

DataProvider.Instance.ExecuteNonQuery("exec SP\_InsertBillInfor @idBill , @idFood , @count ", new object[] { idBill , idFood , count });

}

public int GetMaxIDBill()

{

try

{

return (int)DataProvider.Instance.ExecuteScalar("select MAX(id) from Bill");

}

catch

{

return -1;

}

}

Code chức năng chuyển bàn/ gộp bàn

**View**

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

{

int id1 = (livBill.Tag as Table).ID;

int id2 = (cbSwitchTable.SelectedItem as Table).ID;

if (MessageBox.Show(string.Format("Bạn có thật sự muốn chuyển {0} qua {1}", (livBill.Tag as Table).Name, (cbSwitchTable.SelectedItem as Table).Name), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

TableDAO.Instance.SwitchTable(id1, id2);

LoadTable();

}

}

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

{

int id1 = (livBill.Tag as Table).ID;

int id2 = (cbMergeTable.SelectedItem as Table).ID;

if (MessageBox.Show(string.Format("Bạn có thật sự muốn gộp {0} qua {1}", (livBill.Tag as Table).Name, (cbMergeTable.SelectedItem as Table).Name), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

TableDAO.Instance.MergeTable(id1, id2);

LoadTable();

}

}

**Controller**

public void SwitchTable(int id1, int id2)

{

DataProvider.Instance.ExecuteQuery("USP\_SwitchTable @idTable1 , @idTable2 ", new object[] { id1 , id2 });

}

public void MergeTable(int id1, int id2)

{

DataProvider.Instance.ExecuteQuery("USP\_MergeTable @id1 , @id2 ", new object[] { id1, id2 });

}

Code chức năng Quản lý món ăn

**View**

* **Thêm**

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

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn thêm món chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

string name = txtFoodName.Text;

int categoryID = (cbFoodCategory.SelectedItem as Category).ID;

float price = (float)nmFoodPrice.Value;

if (FoodDAO.Instance.InsertFood(name, categoryID, price))

{

MessageBox.Show("Thêm món thành công");

LoadListFood();

if (insertFood != null)

insertFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi thêm món");

}

}

}

* **Xóa**

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

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn xóa món chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel) {

int id = Convert.ToInt32(txtFoodID.Text);

if (FoodDAO.Instance.DeleteFood(id))

{

MessageBox.Show("Xóa món thành công");

LoadListFood();

if (deleteFood != null)

deleteFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi xóa món");

}

}

}

* **Sửa**

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

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn sửa món chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

string name = txtFoodName.Text;

int categoryID = (cbFoodCategory.SelectedItem as Category).ID;

float price = (float)nmFoodPrice.Value;

int id = Convert.ToInt32(txtFoodID.Text);

if (FoodDAO.Instance.UpdateFood(id, name, categoryID, price))

{

MessageBox.Show("Sửa món thành công");

LoadListFood();

if (updateFood != null)

updateFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi sửa món");

}

}

}

* **Xem**

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

{

LoadListFood();

}

**Controller**

public bool InsertFood(string name, int id, float price)

{

string query = string.Format("INSERT Food ( name, idCategory, price)VALUES ( N'{0}' , {1} , {2} )", name, id, price);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateFood(int idFood, string name, int id, float price)

{

string query = string.Format("UPDATE Food SET name = N'{0}', idCategory = {1}, price = {2} where id = {3}", name, id, price,idFood);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool DeleteFood(int idFood)

{

BillInforDAO.Instance.DeleteBillInforByFoodID(idFood);

string query = string.Format("Delete Food where id={0}", idFood);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

Code chức năng quản lý tài khoản

**View**

* **Thêm**

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

{

string userName = txtUserName.Text;

string displayName = txtDisplayName.Text;

int type = (int)nmAccountType.Value;

AddAccount(userName, displayName, type);

}

* **Xóa**

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

{

string userName = txtUserName.Text;

DeleteAccount(userName);

}

* **Sửa**

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

{

string userName = txtUserName.Text;

string displayName = txtDisplayName.Text;

int type = (int)nmAccountType.Value;

EditAccount(userName, displayName, type);

}

* **Cài lại mật khẩu**

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

{

string userName = txtUserName.Text;

ResetPass(userName);

}

void AddAccount(string userName, string displayName, int type)

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn thêm tài khoản mới mới chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

if (AccountDAO.Instance.InsertAccount(userName, displayName, type))

{

MessageBox.Show("Thêm tài khoản thành công");

}

else

{

MessageBox.Show("Có lỗi khi thêm tài khoản");

}

}

LoadAccount();

}

void EditAccount(string userName, string displayName, int type)

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn sửa lại tài khoản mới chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

if (AccountDAO.Instance.UpdateAccount(userName, displayName, type))

{

MessageBox.Show("Cập nhật tài khoản thành công");

}

else

{

MessageBox.Show("Có lỗi khi cập nhật tài khoản");

}

}

LoadAccount();

}

void DeleteAccount(string userName)

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn xóa tài khoản chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

if (loginAccount.UserName.Equals(userName))

{

MessageBox.Show("Không thể xóa tài khoản đang sử dụng");

return;

}

if (AccountDAO.Instance.DeleteAccount(userName))

{

MessageBox.Show("Xóa tài khoản thành công");

}

else

{

MessageBox.Show("Có lỗi khi xóa tài khoản");

}

}

LoadAccount();

}

void ResetPass(string userName)

{

DialogResult r;

r = MessageBox.Show("Bạn thực sự muốn cài lại mật khẩu chứ? ", "Thông báo",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

if (r != DialogResult.Cancel)

{

if (AccountDAO.Instance.ResetPassWord(userName))

{

MessageBox.Show("Đặt lại mật khẩu thành công");

}

else

{

MessageBox.Show("Có lỗi khi đặt lại mật khẩu");

}

}

}

**Controller**

public bool InsertAccount(string name,string displayName, int type)

{

string query = string.Format("INSERT Account ( UserName, DisplayName, Type)VALUES ( N'{0}' , N'{1}' , {2} )", name, displayName, type);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateAccount(string name, string displayName, int type)

{

string query = string.Format("UPDATE Account SET DisplayName = N'{1}', Type = {2} where UserName = N'{0}'", name, displayName, type);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool DeleteAccount(string name)

{

string query = string.Format("Delete Account where UserName = N'{0}'", name);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool ResetPassWord(string name)

{

string query = string.Format("Update Account set password = N'0' where UserName = N'{0}'", name);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

Code Thống kê doanh thu

**View**

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

{

LoadListBillByDate(dtpkFromDate.Value, dtpkToDate.Value);

}

void LoadListBillByDate(DateTime checkIn, DateTime checkOut)

{

dtgvBill.DataSource = BillDAO.Instance.GetBillListByDate(checkIn, checkOut);

}

**Controller**

public DataTable GetBillListByDate(DateTime checkIn, DateTime checkOut)

{

return DataProvider.Instance.ExecuteQuery("Exec USP\_GetListBillByDate @checkIn , @checkOut ", new object[] { checkIn, checkOut });

}