College of Staten Island

Project 2 Source Code

Christopher Paradiso

Thomas Soutar

Justin Teutonico

CSC 330 46062

Dr. Gueorguieva

9 November, 2016

//Item.cs

class Item

{

private double price;

private string itemName;

public double Price { get { return price; } set { price = value; } }

public string ItemName { get { return itemName; } set { itemName = value; } }

public Item()

{

price = 0;

itemName = "";

}

public Item(double price, string itemName)

{

this.price = price;

this.itemName = itemName;

}

public string formattedPrice()

{

return String.Format("{0:0.00}", Price);

}

}

//Order.cs

class Order

{

private int roomNumber;

public List<Item> itemList = new List<Item>();

public int RoomNumber { get { return roomNumber; } set { roomNumber = value; } }

public Order()

{

roomNumber = 0;

}

public Order(int roomNumber)

{

this.roomNumber = roomNumber;

}

public void add(Item item)

{

itemList.Add(item);

}

public void remove(Item item)

{

itemList.Remove(item);

}

public void remove(int i)

{

itemList.RemoveAt(i);

}

public double totalPrice()

{

double price = 0;

for (int i = 0; i < itemList.Count; i++)

{

price += itemList[i].Price;

}

return price;

}

public string formattedTotalPrice()

{

return String.Format("{0:0.00}", totalPrice());

}

public bool display()

{

string stuff = "";

for (int i = 0; i < itemList.Count; i++)

{

stuff += itemList[i].ItemName + " \t$" + itemList[i].formattedPrice() + "\n";

}

DialogResult dialogResult = MessageBox.Show("You Ordered:\n" + stuff + "\nTotal Price:\t$" + formattedTotalPrice(), "Order", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

return true;

else

return false;

}

}

//Room.cs

class Room

{

private int roomNumber;

private string roomType;

private double roomCost;

public int RoomNumber {get{return roomNumber;}set{roomNumber = value;}}

public string RoomType{get{return roomType;}set{roomType = value;}}

public double RoomCost{get{return roomCost;}set{roomCost = value;}}

}

//BaseForm.cs

public partial class BaseForm : Form

{

public string connectionString = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=| DataDirectory|\Database1.mdf;Integrated Security=True";

}

//Program.cs

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

string connectionString = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=| DataDirectory|\Database1.mdf;Integrated Security=True";

//launches admin creation if first login, otherwise launches login

int adminExists = 0; //0 -> no admin, 1 -> admin, 2 -> exit application

jump:

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand("SELECT type FROM [Table]");

cmd.CommandType = CommandType.Text;

cmd.Connection = connection;

connection.Open();

SqlDataReader myReader = cmd.ExecuteReader();

while (myReader.Read())

{

if ((myReader["type"].ToString()) == "admin")

{

adminExists = 1;

}

}

}

if (adminExists == 1)

{

Application.Run(new Login());

}

else if (adminExists == 0)

{

Application.Run(new CreateAdmin());

adminExists = 2;

goto jump;

}

}

}

 public partial class CreateAdmin : BaseForm

{

public CreateAdmin()

{

InitializeComponent();

}

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

{

MessageBox.Show("An admin account must be created before the system is used.", "Info", MessageBoxButtons.OK, MessageBoxIcon.Question);

}

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

{

if (PasswordBox.Text != VerifyPassword.Text)

{

MessageBox.Show("Password does not match.","Info",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

else

{

//Encrypt password field

byte[] data = System.Text.Encoding.ASCII.GetBytes(PasswordBox.Text);

data = new System.Security.Cryptography.SHA256Managed().ComputeHash(data);

String hash = System.Text.Encoding.ASCII.GetString(data);

using (SqlConnection connection = new SqlConnection(connectionString))

{

//WRITE

SqlCommand cmd = new SqlCommand("INSERT INTO [Table] (username, password, type) VALUES (@username, @password, @type)");

//Table is reserved word -> [Table] instead of Table

cmd.CommandType = CommandType.Text;

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@username", setUserName.Text);

cmd.Parameters.AddWithValue("@password", hash);

cmd.Parameters.AddWithValue("@type", "admin");

connection.Open();

cmd.ExecuteNonQuery();

//READ

SqlCommand cmd2 = new SqlCommand("SELECT userID, username FROM [Table]");

cmd2.CommandType = CommandType.Text;

cmd2.Connection = connection;

SqlDataReader myReader = cmd2.ExecuteReader();

if (myReader.Read())

{

MessageBox.Show("Saved to database...\nUsername: " + myReader["username"].ToString());

}

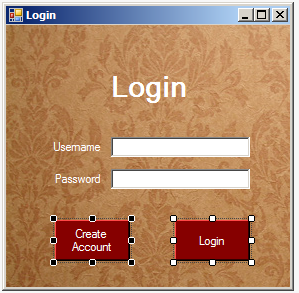
}

}

this.Close();

}

}



public partial class Login : BaseForm

{

public Login()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

//Encrypt password field

byte[] data = System.Text.Encoding.ASCII.GetBytes(password.Text);

data = new System.Security.Cryptography.SHA256Managed().ComputeHash(data);

String hash = System.Text.Encoding.ASCII.GetString(data);

string accessType = "";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand("SELECT username, password, type FROM [Table]");

cmd.CommandType = CommandType.Text;

cmd.Connection = connection;

connection.Open();

SqlDataReader myReader = cmd.ExecuteReader();

while (myReader.Read())

{

if (((myReader["username"].ToString()) == username.Text) &&

((myReader["password"].ToString()) == hash))

{

accessType = myReader["type"].ToString();

}

}

}

if (accessType == "")

{

MessageBox.Show("Account does not exist.");

}

else

{

if (accessType == "admin")

{

AdminCP admincp = new AdminCP();

admincp.Show();

}

if (accessType == "hotelmanager")

{

HotelManCP hotelmancp = new HotelManCP();

hotelmancp.Show();

}

if (accessType == "restmanager")

{

RestManCP restmancp = new RestManCP();

restmancp.Show();

}

if (accessType == "hotelemp")

{

HotelEmpCP hotelempcp = new HotelEmpCP();

hotelempcp.Show();

}

if (accessType == "restemp")

{

RestEmpCP restempcp = new RestEmpCP();

restempcp.Show();

}

}

}

private void button2\_Click(object sender, EventArgs e)

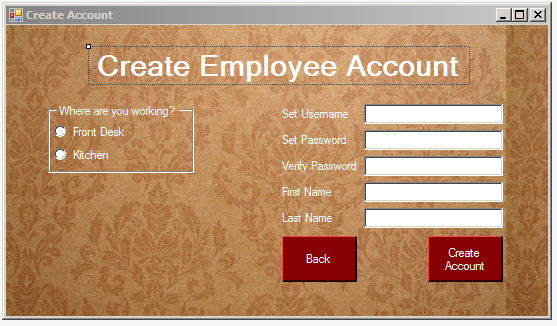
{

CreateAccount createaccount = new CreateAccount();

createaccount.Show();

}

}



public partial class CreateAccount : BaseForm

{

public CreateAccount()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

//Encrypt password field

byte[] data = System.Text.Encoding.ASCII.GetBytes(passwordBox.Text);

data = new System.Security.Cryptography.SHA256Managed().ComputeHash(data);

String hash = System.Text.Encoding.ASCII.GetString(data);

using (SqlConnection connection = new SqlConnection(connectionString))

{

//WRITE

SqlCommand cmd = new SqlCommand("INSERT INTO [Table] (username, password, type, firstname, lastname) VALUES (@username, @password, @type, @firstname, @lastname)");

//Table is reserved word -> [Table] instead of Table

cmd.CommandType = CommandType.Text;

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@username", usernameBox.Text);

cmd.Parameters.AddWithValue("@password", hash);

cmd.Parameters.AddWithValue("@firstname", firstnameBox.Text);

cmd.Parameters.AddWithValue("@lastname", lastnameBox.Text);

if (frontDeskButton.Checked)

cmd.Parameters.AddWithValue("@type", "hotelemp");

if (kitchenButton.Checked)

cmd.Parameters.AddWithValue("@type", "restemp");

connection.Open();

cmd.ExecuteNonQuery();

//READ

SqlCommand cmd2 = new SqlCommand("SELECT userID, username, firstname, lastname FROM [Table]");

cmd2.CommandType = CommandType.Text;

cmd2.Connection = connection;

SqlDataReader myReader = cmd2.ExecuteReader();

while (myReader.Read())

{

if (myReader["username"].ToString() == usernameBox.Text)

MessageBox.Show("Saved to database...\nUsername: " + myReader["username"].ToString() + "\nName:" + myReader["firstname"] + " " + myReader["lastname"]);

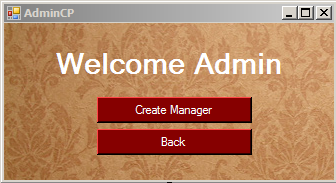
}

}

this.Close();

}

}



public partial class AdminCP : Form

{

public AdminCP()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

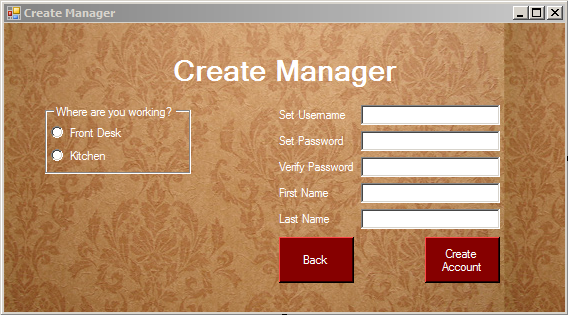
{

CreateManager cm = new CreateManager();

cm.Show();

}

}



public partial class CreateManager : BaseForm

{

public CreateManager()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

//Encrypt password field

byte[] data = System.Text.Encoding.ASCII.GetBytes(passwordBox.Text);

data = new System.Security.Cryptography.SHA256Managed().ComputeHash(data);

String hash = System.Text.Encoding.ASCII.GetString(data);

using (SqlConnection connection = new SqlConnection(connectionString))

{

//WRITE

SqlCommand cmd = new SqlCommand("INSERT INTO [Table] (username, password, type) VALUES (@username, @password, @type)");

//Table is reserved word -> [Table] instead of Table

cmd.CommandType = CommandType.Text;

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@username", usernameBox.Text);

cmd.Parameters.AddWithValue("@password", hash);

if (frontDeskButton.Checked)

cmd.Parameters.AddWithValue("@type", "hotelmanager");

if (kitchenButton.Checked)

cmd.Parameters.AddWithValue("@type", "restmanager");

connection.Open();

cmd.ExecuteNonQuery();

//READ

SqlCommand cmd2 = new SqlCommand("SELECT userID, username FROM [Table]");

cmd2.CommandType = CommandType.Text;

cmd2.Connection = connection;

SqlDataReader myReader = cmd2.ExecuteReader();

while (myReader.Read())

{

if (myReader["username"].ToString() == usernameBox.Text)

MessageBox.Show("Saved to database...\nUsername: " + myReader["username"].ToString());

}

}

this.Close();

}

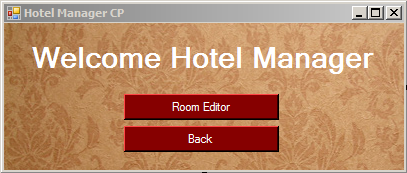
private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}



public partial class HotelManCP : BaseForm

{

public HotelManCP()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

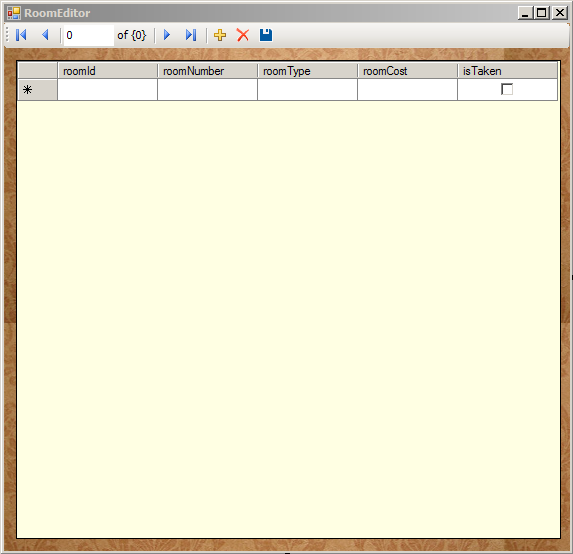
{

RoomEditor re = new RoomEditor();

re.Show();

}

}



public partial class RoomEditor : Form

{

public RoomEditor()

{

InitializeComponent();

}

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

{

this.Validate();

this.roomDBBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

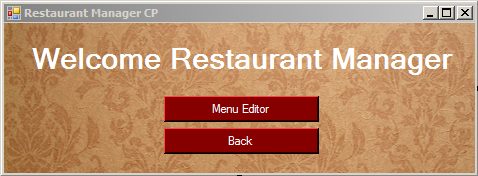
private void RoomEditor\_Load(object sender, EventArgs e)

{

this.roomDBTableAdapter.Fill(this.database1DataSet.roomDB);

}

}



public partial class RestManCP : BaseForm

{

public RestManCP()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

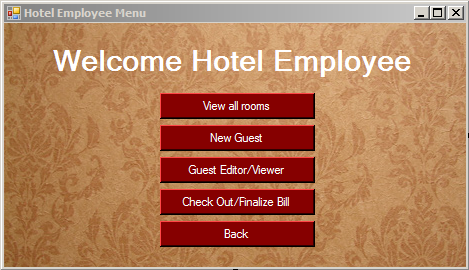
{

MenuEditor me = new MenuEditor();

me.Show();

}

}



public partial class HotelEmpCP : BaseForm

{

public HotelEmpCP()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

GuestEditor ge = new GuestEditor();

ge.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

RoomViewer rv = new RoomViewer();

rv.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

CheckOut co = new CheckOut();

co.Show();

}

private void button4\_Click(object sender, EventArgs e)

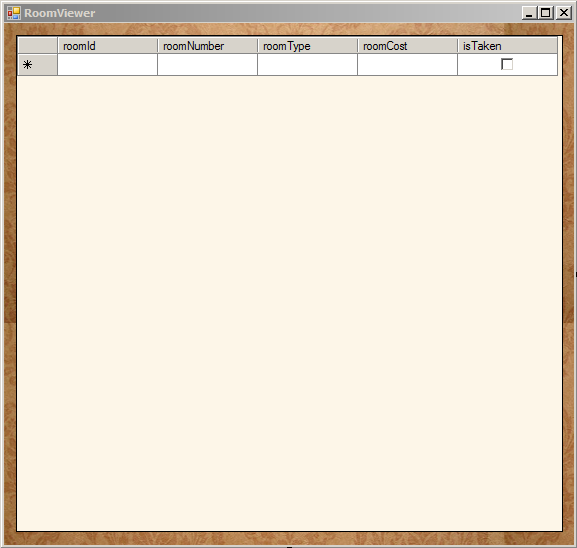
{

NewGuest ng = new NewGuest();

ng.Show();

}

}



public partial class RoomViewer : Form

{

public RoomViewer()

{

InitializeComponent();

}

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

{

this.Validate();

this.roomDBBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

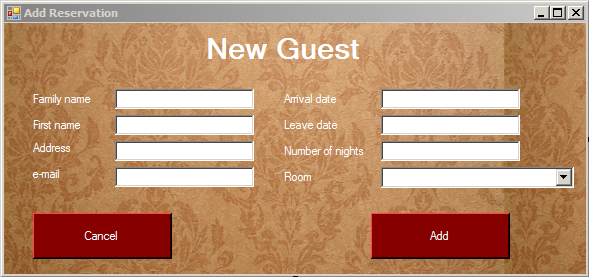
private void RoomViewer\_Load(object sender, EventArgs e)

{

this.roomDBTableAdapter.Fill(this.database1DataSet.roomDB);

}

}



public partial class NewGuest : BaseForm

{

//Make list of rooms (new object)

private List<Room> rooms = new List<Room>();

int selectedRoom;

public NewGuest()

{

InitializeComponent();

FillCombo();

}

private void FillCombo()

{

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand("SELECT roomNumber, roomType, roomCost, isTaken FROM roomDB",sqlConnection);

sqlConnection.Open();

SqlDataReader sqlReader = cmd.ExecuteReader();

while (sqlReader.Read())

{

bool isTaken = false;

string temp = sqlReader["isTaken"].ToString();

if (temp == "" || temp == "False")

{

isTaken = false;

}

else

{

isTaken = true;

}

if (!isTaken)

{

//READ

int roomNumber = Convert.ToInt32(sqlReader["roomNumber"]);

string roomType = sqlReader["roomType"].ToString();

double roomCost = Convert.ToDouble(sqlReader["roomCost"]);

string formattedRoomCost = String.Format("{0:0.00}", roomCost);

string itemStr = "#" + roomNumber + " - " + roomType + " - $" + formattedRoomCost;

comboBox1.Items.Add(itemStr);

Room newRoom = new Room();

newRoom.RoomNumber = roomNumber;

newRoom.RoomType = roomType;

newRoom.RoomCost = roomCost;

rooms.Add(newRoom);

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

if (rooms.Count != 0)

{

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand("INSERT INTO guestDB (lastName, firstName, address, email, arrivalDate, leaveDate, numberOfNights, roomType, roomNumber, totalCost) VALUES (@lastName, @firstName, @address, @email, @arrivalDate, @leaveDate, @numberOfNights, @roomType, @roomNumber, @totalCost)");

cmd.Connection = sqlConnection;

cmd.Parameters.AddWithValue("@lastName", textBox1.Text);

cmd.Parameters.AddWithValue("@firstName", textBox2.Text);

cmd.Parameters.AddWithValue("@address", textBox3.Text);

cmd.Parameters.AddWithValue("@email", textBox4.Text);

cmd.Parameters.AddWithValue("@arrivalDate", textBox5.Text);

cmd.Parameters.AddWithValue("@leaveDate", textBox6.Text);

cmd.Parameters.AddWithValue("@numberOfNights", textBox7.Text);

cmd.Parameters.AddWithValue("@roomType", rooms[selectedRoom].RoomType);

cmd.Parameters.AddWithValue("@roomNumber", rooms[selectedRoom].RoomNumber);

cmd.Parameters.AddWithValue("@totalCost", rooms[selectedRoom].RoomCost \* Convert.ToInt32(textBox7.Text));

sqlConnection.Open();

cmd.ExecuteNonQuery();

}

MessageBox.Show("Guest has been added.");

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

//CHANGE

SqlCommand sqlCmd = new SqlCommand("UPDATE roomDB SET isTaken = 1 Where roomNumber = @rn", sqlConnection);

sqlCmd.Parameters.AddWithValue("@rn", rooms[selectedRoom].RoomNumber);

sqlConnection.Open();

SqlDataReader sqlReader = sqlCmd.ExecuteReader();

}

this.Close();

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex != -1)

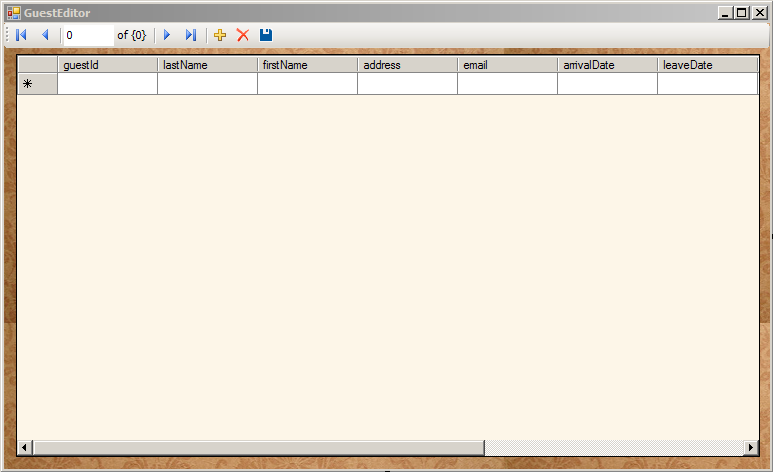
{

selectedRoom = comboBox1.SelectedIndex;

}

}

}



public partial class GuestEditor : Form

{

public GuestEditor()

{

InitializeComponent();

}

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

{

this.Validate();

this.guestDBBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

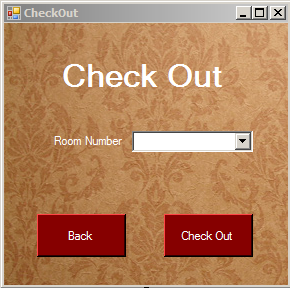
private void GuestEditor\_Load(object sender, EventArgs e)

{

this.guestDBTableAdapter.Fill(this.database1DataSet.guestDB);

}

}



public partial class CheckOut : BaseForm

{

private List<int> rooms = new List<int>();

int selectedIndex;

public CheckOut()

{

InitializeComponent();

FillCombo();

}

private void FillCombo()

{

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand("SELECT lastName, firstName, roomNumber FROM guestDB", sqlConnection);

sqlConnection.Open();

SqlDataReader sqlReader = cmd.ExecuteReader();

while (sqlReader.Read())

{

//READ

int roomNumber = Convert.ToInt32(sqlReader["roomNumber"]);

string name = sqlReader["firstName"].ToString() + " " + sqlReader["lastName"].ToString();

string show = "#" + roomNumber + " - " + name;

roomNum.Items.Add(show);

rooms.Add(roomNumber);

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

// 0->fail 1->cancel 2->yes

int success = 0;

using (SqlConnection connection = new SqlConnection(connectionString))

{

//READ

SqlCommand cmdR = new SqlCommand("SELECT guestId, lastName, firstName, roomNumber, totalCost FROM guestDB");

cmdR.CommandType = CommandType.Text;

cmdR.Connection = connection;

connection.Open();

SqlDataReader myReader = cmdR.ExecuteReader();

while (myReader.Read())

{

if (myReader["roomNumber"].ToString() == rooms[selectedIndex].ToString())

{

DialogResult dialogResult = MessageBox.Show("Name: " + myReader["firstName"].ToString() + " " + myReader["lastName"] + "\nTotal Cost: $" + myReader["totalCost"], "Bill", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

success = 2;

else

success = 1;

}

}

}

if (success == 2)

{

//DELETE

SqlConnection con = new SqlConnection(connectionString);

con.Open();

SqlCommand cmdD = new SqlCommand(@"DELETE FROM guestDB WHERE (roomNumber = '" + rooms[selectedIndex] + "')", con);

cmdD.ExecuteNonQuery();

con.Close();

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

//CHANGE

SqlCommand sqlCmd = new SqlCommand("UPDATE roomDB SET isTaken = 0 Where roomNumber = @rn", sqlConnection);

sqlCmd.Parameters.AddWithValue("@rn", rooms[selectedIndex]);

sqlConnection.Open();

SqlDataReader sqlReader = sqlCmd.ExecuteReader();

}

MessageBox.Show("Guest checked out.");

this.Close();

}

else if (success == 0)

MessageBox.Show("The record does not exist.");

}

private void roomNum\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

if (roomNum.SelectedIndex != -1)

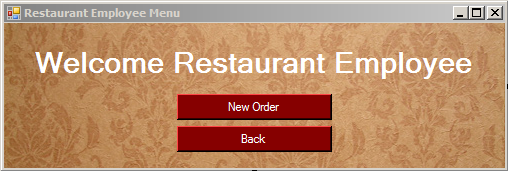
{

selectedIndex = roomNum.SelectedIndex;

}

}

}



public partial class RestEmpCP : BaseForm

{

public RestEmpCP()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

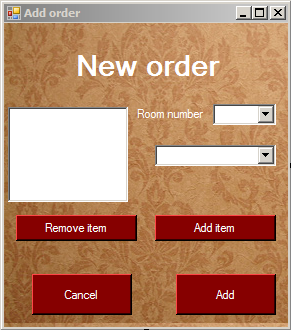
{

NewOrder no = new NewOrder();

no.Show();

}

}



public partial class NewOrder : BaseForm

{

private Order order = new Order();

private List<Item> menu = new List<Item>();

//When clicking on button, the listbox selection becomes -1: fix

private int saveSelectionValue = 0;

public NewOrder()

{

InitializeComponent();

Fillcombo();

}

void Fillcombo()

{

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand sqlCmd = new SqlCommand("SELECT foodName, foodCost FROM menuDB", sqlConnection);

sqlConnection.Open();

SqlDataReader sqlReader = sqlCmd.ExecuteReader();

while (sqlReader.Read())

{

double itemCost = Convert.ToDouble(sqlReader["foodCost"]);

string itemName = sqlReader["foodName"].ToString();

string iC = String.Format("{0:0.00}", itemCost);

string itemStr = itemName + " - " + iC;

comboBox1.Items.Add(itemStr);

//Each item is aded to the menu, accessible by it's index.

menu.Add(new Item(itemCost, itemName));

}

sqlReader.Close();

}

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand sqlCmd = new SqlCommand("SELECT roomNumber FROM guestDB", sqlConnection);

sqlConnection.Open();

SqlDataReader sqlReader = sqlCmd.ExecuteReader();

while (sqlReader.Read())

{

roomNum.Items.Add(sqlReader["roomNumber"]);

}

sqlReader.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

//Ignore unselection

if (listBox1.SelectedIndex != -1)

saveSelectionValue = listBox1.SelectedIndex;

}

private void button3\_Click(object sender, EventArgs e)

{

if (roomNum.Text == "")

{

MessageBox.Show("Please enter a room number.");

}

else

{

order.RoomNumber = Convert.ToInt32(roomNum.Text);

if (order.display())

{

using (SqlConnection sqlConnection = new SqlConnection(connectionString))

{

SqlCommand sqlCmd = new SqlCommand("UPDATE guestDB SET totalCost += @tc Where roomNumber = @rn", sqlConnection);

sqlCmd.Parameters.AddWithValue("@tc", order.totalPrice());

sqlCmd.Parameters.AddWithValue("@rn", order.RoomNumber);

sqlConnection.Open();

SqlDataReader sqlReader = sqlCmd.ExecuteReader();

}

MessageBox.Show("Your order has been processed.");

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex != -1)

{

listBox1.Items.Add(comboBox1.Text);

order.add(menu[comboBox1.SelectedIndex]);

}

}

private void button4\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedIndex != -1)

{

order.remove(saveSelectionValue);

for (int i = listBox1.SelectedIndices.Count - 1; i >= 0; i--)

{

listBox1.Items.RemoveAt(listBox1.SelectedIndices[i]);

}

}

}

}