Bahria University,

Karachi Campus



ASSIGNMENT# 1

NAME : Syeda Farwa Hyder SHah

ENRL# :02-131172-035

Submitted On:

24-April-2019

(Date: DD/MM/YY)

**CLASS:**

**CLASS 1**

namespace Authentication

{

public class Hash

{

public static string Encrypt(string text)

{

string key = "";

key += "82s9";

foreach (char c in text)

{

key += Convert.ToInt32(c);

}

key += "!Ysa";

return key;

}

}

}

**CLASS 2**

public class User

{

public static List<User> users = new List<User>();

public string Username { get; set; }

public string Password { get; set; }

public User()

{

}

public User(string u, string p)

{

this.Username = u;

this.Password = p;

}

//Check if username already exists

static bool CheckExistence(string u)

{

if (!users.Any() || users.Count > 0)

{

List<User> temp = users.Where(x => x.Username.Equals(u)).ToList();

if (temp.Count>0)

{

//User already exists

return true;

}

else

{

//User doesnot exist

return false;

}

}

else

{

return false;

}

}

//Signup function

public static bool CreateAccount(string u, string p)

{

if (CheckExistence(u))

{

//User already exists : Signup unsuccessful

return false;

}

else

{

//Password is stored in encrypted form

//Signup successful

users.Add(new User(u, Hash.Encrypt(p)));

return true;

}

}

public static bool SignIn(string u, string p)

{

if (!users.Any() || users.Count > 0)

{

List<User> temp = users.Where(x => x.Username.Equals(u)).ToList();

if (temp.Count > 0)

{

User obj = temp.Where(x => x.Username == u).First();

if ((Hash.Encrypt(p)).Equals(obj.Password))

{

//SignIn Successful

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

else

{

return false;

}

}

}

}

**CLASS 3**

namespace OOSELab5.Classes

{

//A Cart is basically a bill. It has purchased products and relatives calcs

public class Cart

{

//This is to hold all of the generated bills

public static List<Cart> bills = new List<Cart>();

public static int ID = 123;

public int bill\_id;

public List<Product> products\_purchased=new List<Product>();

public DateTime GeneratedOn { get; set; }

public double Discount { get; set; }

public int TotalItems { get; set; }

public double SubTotal { get; set; }

public double Total { get; set; }

public Cart(List<Product> list,double subtotal, double discount)

{

bill\_id = ID;

this.products\_purchased = list;

this.TotalItems = list.Count;

this.SubTotal = subtotal;

this.Discount = discount;

this.Total = subtotal - discount;

GeneratedOn = DateTime.Now;

ID++;

}

}

}

**CLASS 4**

namespace OOSELab5.Classes

{

public class Product

{

public static List<Product> products = new List<Product>();

public string Name { get; set; }

public int Price { get; set; }

public int Quantity { get; set; }

public Product(string name,int price)

{

this.Name = name;

this.Price = price;

}

public Product(string name, int price,int qty)

{

this.Name = name;

this.Price = price;

this.Quantity = qty;

}

}

}

**SIGNIN:**

namespace OOSELab5.Forms

{

public partial class SignIn : Form

{

public SignIn()

{

InitializeComponent();

}

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

{

this.Hide();

SignUp form = new SignUp();

form.Show();

}

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

{

try

{

if (txtUsername.Text == "")

{

MessageBox.Show("Please enter a username", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtUsername.Text.Length < 3)

{

MessageBox.Show("Username must be atleast 3-digits long", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text == "")

{

MessageBox.Show("Please enter a password", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text.Length < 6)

{

MessageBox.Show("Password must be atleast 6-digits long", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

if (User.SignIn(txtUsername.Text,txtPassword.Text))

{

this.Hide();

Dashboard form = new Dashboard();

form.Show();

}

else

{

MessageBox.Show("Invalid Username or Password", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

catch (Exception error)

{

MessageBox.Show(error.Message);

}

}

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

{

}

}

}

**SIGNUP:**

namespace OOSELab5.Forms

{

public partial class SignUp : Form

{

public SignUp()

{

InitializeComponent();

}

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

{

this.Hide();

SignIn form = new SignIn();

form.Show();

}

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

{

try

{

if (txtUsername.Text=="")

{

MessageBox.Show("Please enter a username","Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtUsername.Text.Length < 3)

{

MessageBox.Show("Username must be atleast 3-digits long", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text=="")

{

MessageBox.Show("Please enter a password", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text.Length < 6)

{

MessageBox.Show("Password must be atleast 6-digits long", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtCPassword.Text == "")

{

MessageBox.Show("Please confirm password", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (!txtPassword.Text.Equals(txtCPassword.Text))

{

MessageBox.Show("The Password doesnot match", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

if (User.CreateAccount(txtUsername.Text, txtCPassword.Text))

{

MessageBox.Show("You've registered successfully!", "Success",

MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Hide();

SignIn form = new SignIn();

form.Show();

}

else

{

MessageBox.Show("Could not register", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

catch (Exception error)

{

MessageBox.Show(error.Message);

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

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

{

}

}

}

**GENERATE BILL:**{

public partial class Dashboard : Form

{

//Temporary list that holds purchased products

List<Product> purchased;

//pid is for serial number of Products in bill

int pid=1;

//Variables for Calculation

double subtotal=0, discount=0, total=0;

public Dashboard()

{

InitializeComponent();

purchased = new List<Product>();

}

//Signout

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

{

if (DialogResult.Yes == MessageBox.Show("Are you sure you want to Signout?","Signout",MessageBoxButtons.YesNo,MessageBoxIcon.Question))

{

this.Hide();

SignIn form = new SignIn();

form.Show();

}

}

//Exit

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

{

if (DialogResult.Yes == MessageBox.Show("Are you sure you want to exit?", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

Application.Exit();

}

}

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

{

//Add Products upon Form Load

addProducts();

//Live Time & Date

Timer timer = new Timer();

lbDateTime.Text = DateTime.Now.ToString("ddd, dd MMMM hh:mm tt");

timer.Tick += new EventHandler(timer\_Tick);

timer.Interval = (60 - DateTime.Now.Second)\*1000;

timer.Enabled = true;

}

private void timer\_Tick(object sender, EventArgs e)

{

lbDateTime.Text = DateTime.Now.ToString("ddd, dd MMMM hh:mm tt");

}

//When any product's quantity is changed

private void quantityEntered(object sender, DataGridViewCellEventArgs e)

{

try

{

//Data of Selected Product

string selectedProduct = dataProducts.Rows[e.RowIndex].Cells[1].Value.ToString();

int selectedProductPrice = Convert.ToInt32(dataProducts.Rows[e.RowIndex].Cells[2].Value.ToString());

string enteredQty = dataProducts.Rows[e.RowIndex].Cells[e.ColumnIndex].Value.ToString();

int qty;

if (!int.TryParse(enteredQty, out qty))

{

dataProducts.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = 0;

MessageBox.Show("Invalid quantity", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if(Convert.ToInt32(enteredQty) <= 0)

{

dataProducts.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = 0;

MessageBox.Show("Quantity must be greater than 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

//Everything's fine. Add it to the list and do Calcs

purchased.Add(new Product(selectedProduct, selectedProductPrice, qty));

int currentProductTotal = selectedProductPrice \* qty;

subtotal += currentProductTotal;

total = subtotal - discount;

//Update Bill ID

pid++;

//Update Labels

lbTotalItems.Text = purchased.Count.ToString();

lbSubtotal.Text = "Rs. " + subtotal;

lbTotal.Text = "Rs. "+total;

//Disable Cell so it can't be edited again

dataProducts.Rows[e.RowIndex].Cells[e.ColumnIndex].ReadOnly = true;

}

}

catch (Exception error)

{

MessageBox.Show(error.Message,"Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

//Apply Fixed Discount

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

{

double amount;

if (!double.TryParse(tbDiscAmount.Text, out amount))

{

MessageBox.Show("Invalid Discount amount", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (amount<0)

{

MessageBox.Show("Discount can't be less than 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

ApplyDiscount("Fixed", amount);

}

}

//Apply Percentage Discount

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

{

double amount;

if (!double.TryParse(tbDiscAmount.Text, out amount))

{

MessageBox.Show("Invalid Discount Percentage", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (amount < 0)

{

MessageBox.Show("Discount can't be less than 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (amount > 100)

{

MessageBox.Show("Discount can't be greater than 100%", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

ApplyDiscount("Percentage", amount);

}

}

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

{

Bills form = new Bills();

form.Show();

}

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

{

//Work only if there is atleast 1 product purchased

if (purchased.Count>0)

{

MessageBox.Show("Bill ID is " + Cart.ID, "Checkout Successful", MessageBoxButtons.OK, MessageBoxIcon.Information);

Cart.bills.Add(new Cart(purchased, subtotal, discount));

//Empty purchase list

this.purchased = new List<Product>();

//Reset Products Table

dataProducts.Rows.Clear();

dataProducts.Refresh();

//Empty Calcs

subtotal = discount = total = 0;

//Display Products again

addProducts();

}

}

//Add some Products to the Table

private void addProducts()

{

Product.products.Add(new Product("Nachos", 450));

Product.products.Add(new Product("Tacos", 500));

Product.products.Add(new Product("Chicken Steak", 900));

Product.products.Add(new Product("Crispy Strip Chicken", 300));

Product.products.Add(new Product("Spicy Chicken Wings", 400));

Product.products.Add(new Product("Smoked BBQ Burger", 600));

Product.products.Add(new Product("Grilled Chicken Burger", 450));

Product.products.Add(new Product("BBQ Chicken Sliders", 250));

Product.products.Add(new Product("Chilli Sauce Pasta", 200));

Product.products.Add(new Product("Cheese Club Sandwich", 200));

Product.products.Add(new Product("Chicken Biryani", 200));

Product.products.Add(new Product("Mutton Biryani", 300));

Product.products.Add(new Product("French Toast", 250));

Product.products.Add(new Product("Late", 300));

Product.products.Add(new Product("Cuppuccino", 300));

Product.products.Add(new Product("Espresso", 500));

Product.products.Add(new Product("Tea", 100));

Product.products.Add(new Product("Ice Tea", 150));

Product.products.Add(new Product("Soft Drink", 80));

Product.products.Add(new Product("Juice", 200));

//Add Products from List to Table

int id = 1;

foreach (Product p in Product.products)

{

dataProducts.Rows.Add(new Object[] { id, p.Name, p.Price, 0 });

id++;

}

}

//Apply Discount

private void ApplyDiscount(string type,double amount)

{

//Count > 0 cause there should be atleast 1 purchased product

if (purchased.Count>0)

{

if (type == "Percentage")

{

amount /= 100;

discount = subtotal \* amount;

total = subtotal - discount;

//Update Labels

lbDiscAmount.Text = "Rs. " + discount;

lbSubtotal.Text = "Rs. " + subtotal;

lbTotal.Text = "Rs. " + total;

}

else if (type == "Fixed")

{

discount = amount;

total = subtotal - discount;

//Update Labels

lbDiscAmount.Text = "Rs. " + discount;

lbSubtotal.Text = "Rs. " + subtotal;

lbTotal.Text = "Rs. " + total;

}

tbDiscAmount.Text = null;

}

}

}

}

BILL:

namespace OOSELab5.Forms

{

public partial class Bills : Form

{

public Bills()

{

InitializeComponent();

HideContent();

}

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

{

try

{

DisplayBill(Convert.ToInt32(tbBillID.Text));

tbBillID.Text = null;

}

catch (Exception)

{

MessageBox.Show("Bill doesnot exist","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

private void HideContent()

{

panelCalc.Visible = false;

panel1.Visible = false;

}

private void ShowContent()

{

panelCalc.Visible = true;

panel1.Visible = true;

lbSubTotal.Visible = true;

}

//Display bill

private void DisplayBill(int id)

{

lbIDs.Text = lbProducts.Text= lbPrices.Text= lbQtys.Text=null;

int pid = 1;

Cart obj = Cart.bills.Where(x => x.bill\_id == id).ToList().FirstOrDefault();

foreach (Product p in obj.products\_purchased)

{

lbIDs.Text +=pid +"."+ Environment.NewLine + Environment.NewLine;

pid++;

lbProducts.Text += p.Name + Environment.NewLine + Environment.NewLine;

lbPrices.Text += p.Price + Environment.NewLine + Environment.NewLine;

lbQtys.Text += p.Quantity + Environment.NewLine + Environment.NewLine;

}

lbBillID.Text = obj.bill\_id.ToString();

lbGenDate.Text = obj.GeneratedOn.ToString("ddd, dd MMMM hh:mm tt");

lbItems.Text = obj.TotalItems.ToString();

lbDiscount.Text = "Rs. "+obj.Discount;

lbSubTotal.Text = "Rs. " + obj.SubTotal;

lbTotal.Text = "Rs. " + obj.Total;

this.panel1.Height = this.lbIDs.Size.Height + 50;

this.panelCalc.Location = new Point(1, lbIDs.Location.Y + lbIDs.Size.Height + 40);

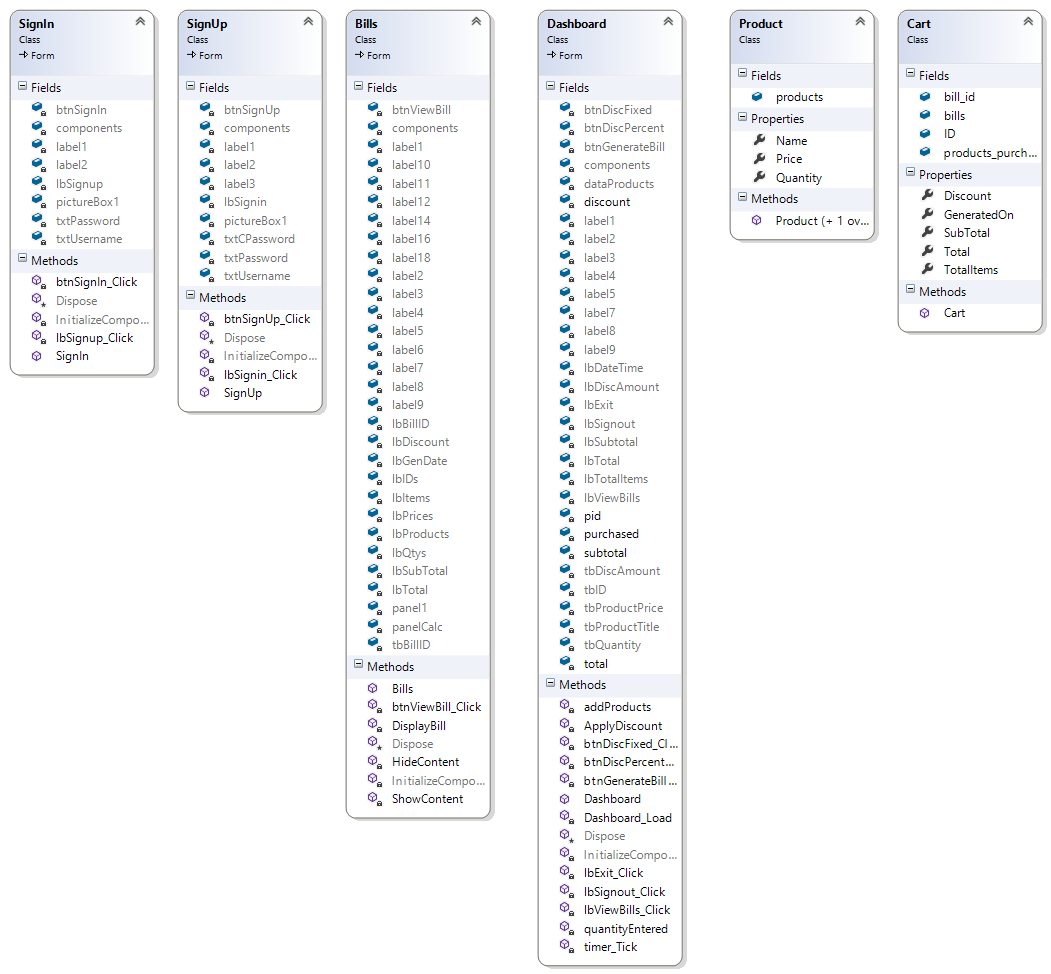
ShowContent();

}

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

{ }}}

**CLASS DIAGRAM:**



**COMPONENT DIAGRAM:**



**OBJECT DIAGRAM:**

