**Book Store Management System**



Session: 2021 – 2024

**Submitted by:**

Laiba Arif

2021-CS-173

**Supervised by:**

Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Table of Content**

**Content Page No.**

Domain model 3

>Domain Model with class name 3

>Domain Model with Relation and Constraint 3

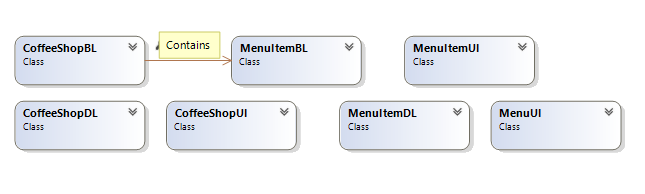
>Domain Model with Multiplicity 3

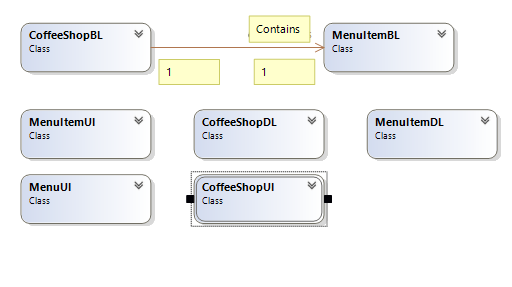
Class Diagram 4

Sequence Diagram 5

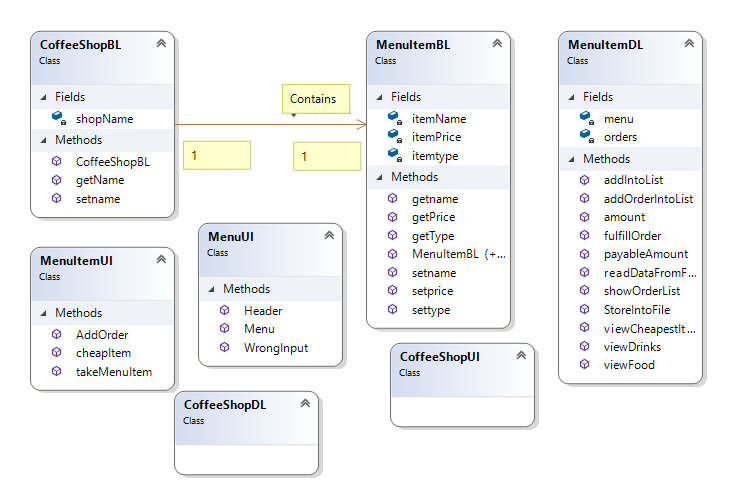
Code for Classes 9

Driver Program 14

* **Domain Model with Relations and Constraints. **
* **Domain Model With Multiplicity**

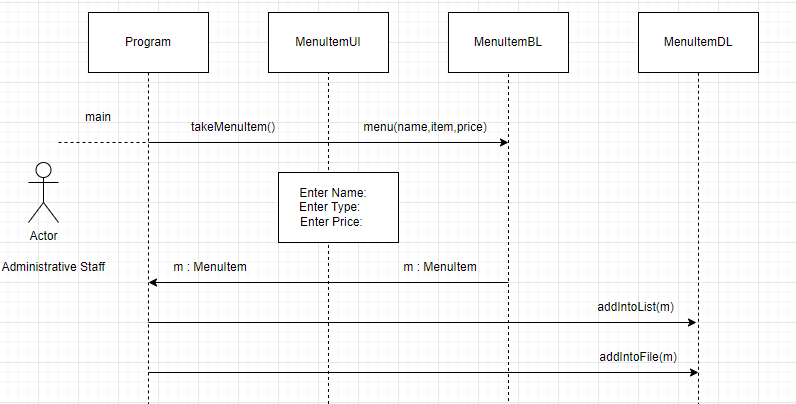
****

**Class Diagram**

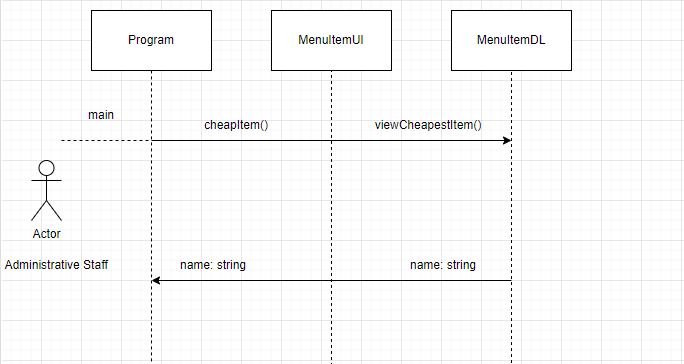
****

**Sequence Diagram**

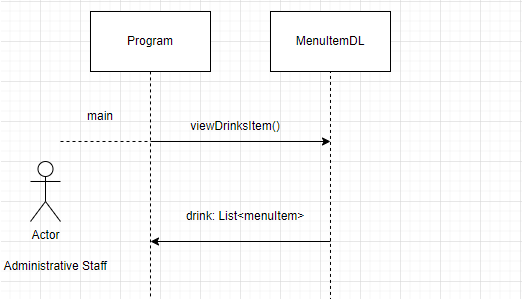
* **Option 1**



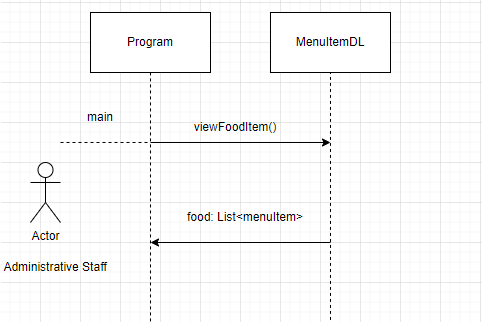
* **Option 2**



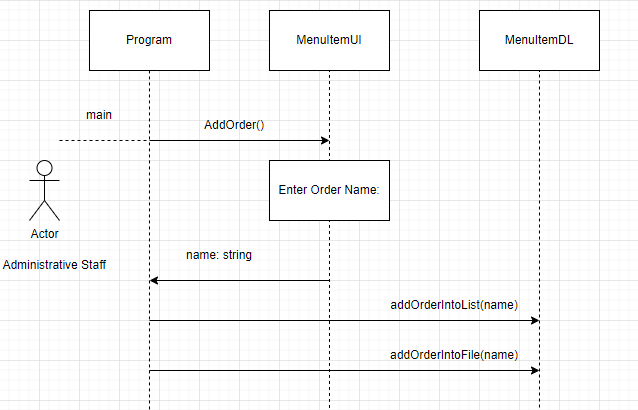
* **Option 3**



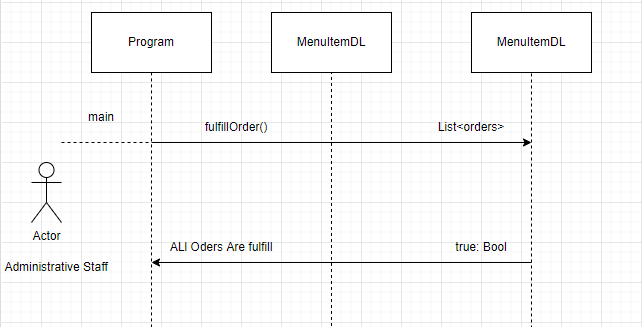
* **Option 4**



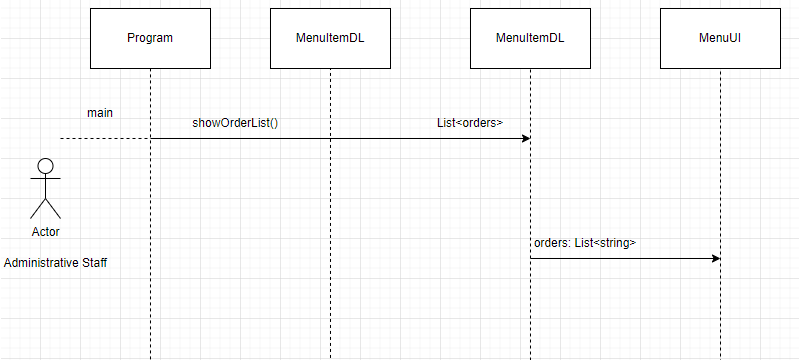
* **Option 5**



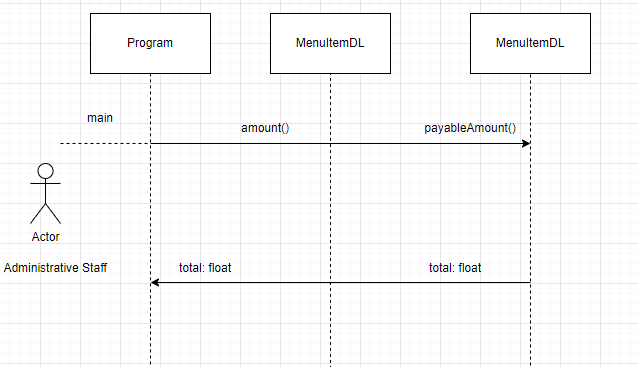
* **Option 6**



* **Option 7**



* **Option 8**



**Code for Classes**

* **Coffee Shop BL**

class CoffeeShopBL

{

private string shopName;

public CoffeeShopBL(string name)

{

this.shopName = name;

}

internal MenuItemBL Contains

{

get => default;

set

{

}

}

public string getName()

{

return shopName;

}

public void setname(string name)

{

shopName = name;

}

}

* **Menu UI**

class MenuUI

{

// Print header

public static void Header(string name)

{

Console.WriteLine("\t\t######################################");

Console.WriteLine("\t\t "+name+ " Coffee Shop ");

Console.WriteLine("\t\t######################################");

}

//Print Menu

public static string Menu()

{

Console.WriteLine("1.Add a Menu item");

Console.WriteLine("2.View the Cheapest Item in the menu");

Console.WriteLine("3.View the Drink’s Menu");

Console.WriteLine("4.View the Food’s Menu");

Console.WriteLine("5.Add Order");

Console.WriteLine("6.Fulfill the Order");

Console.WriteLine("7.View the Orders’s List");

Console.WriteLine("8.Total Payable Amount");

Console.WriteLine("9.Exit");

Console.Write("Your Option: ");

string op = Console.ReadLine();

return op;

}

//For any Other input

public static void WrongInput()

{

Console.WriteLine("Wrong Input");

}

}

* **MenuItem BL**

class MenuItemBL

{

private string itemName;

private string itemtype;

private float itemPrice;

//Parametrized Constructor

public MenuItemBL(string name, string type, float price)

{

this.itemName = name;

this.itemtype = type;

this.itemPrice = price;

}

//Default Constructor

public MenuItemBL()

{

}

// to get name

public string getname()

{

return itemName;

}

// get type

public string getType()

{

return itemtype;

}

// set price

public float getPrice()

{

return itemPrice;

}

// set name

public void setname(string name)

{

itemName = name;

}

// set item type

public void settype(string type)

{

itemtype = type;

}

// set item price

public void setprice(float price)

{

itemPrice = price;

}

}

* **MenuItem DL**

class MenuItemDL

{

static List<MenuItemBL> menu = new List<MenuItemBL>();

static List<string> orders = new List<string>();

public static void addIntoList(MenuItemBL m)

{

menu.Add(m);

}

public static void StoreIntoFile(string path,MenuItemBL m)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(m.getname() + "," + m.getType() + "," + m.getPrice());

file.Flush();

file.Close();

}

public static void readDataFromFile(string path)

{

StreamReader file = new StreamReader(path);

string record;

if(File.Exists(path))

{

while((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string name = splittedRecord[0];

string type = (splittedRecord[1]);

int price = int.Parse(splittedRecord[2]);

MenuItemBL item = new MenuItemBL(name, type, price);

addIntoList(item);

}

}

file.Close();

}

public static string viewCheapestItem()

{

MenuItemBL m = new MenuItemBL();

List<MenuItemBL> sortedList = new List<MenuItemBL>();

sortedList = menu.OrderBy(o => o.getPrice()).ToList();

return sortedList[0].getname();

}

public static void viewDrinks()

{

Console.WriteLine("Name\t\tType\t\tPrice");

foreach (MenuItemBL m in menu)

{

if (m.getType() == "Drink")

{

Console.WriteLine(m.getname() + "\t\t" + m.getType() + "\t\t" + m.getPrice());

}

}

}

public static void viewFood()

{

Console.WriteLine("Name\t\tType\t\tPrice");

foreach (MenuItemBL m in menu)

{

if (m.getType() == "Food")

{

Console.WriteLine(m.getname() + "\t\t" + m.getType() + "\t\t" + m.getPrice());

}

}

}

public static void addOrderIntoList(string name)

{

int flag = 0;

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

{

if (name == menu[i].getname())

{

orders.Add(name);

Console.WriteLine("Order Added");

flag = 1;

break;

}

}

if (flag == 0)

{

Console.WriteLine("This item is currently unavailable!");

}

}

public static void fulfillOrder()

{

int flag = 0;

if (orders != null)

{

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

{

Console.WriteLine("The " + orders[i] + " is Ready!");

flag = 1;

}

if (flag == 1)

{

Console.WriteLine("All orders have been fulfilled!");

orders.Clear();

}

}

else

{

Console.WriteLine("No! Order Placed.");

}

}

public static float payableAmount()

{

float y = 0;

float total = 0;

if (orders != null)

{

for (int j = 0; j < menu.Count; j++)

{

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

{

if (orders[i] == menu[j].getname())

{

total = total + menu[j].getPrice();

}

}

}

}

return total;

}

public static void showOrderList()

{

if (orders != null)

{

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

{

Console.WriteLine(orders[i]);

}

}

}

public static void amount()

{

float amount = 0;

amount = payableAmount();

Console.WriteLine("The Total Payable Amount is: " + amount);

}

}

* **MenuItem UI**

class MenuItemUI

{

public static MenuItemBL takeMenuItem()

{

Console.Write("\nEnter Item Name: ");

string name = Console.ReadLine();

Console.Write("Enter Item Type: ");

string type = Console.ReadLine();

Console.Write("Enter Price: ");

float price = float.Parse(Console.ReadLine());

MenuItemBL m = new MenuItemBL();

m.setname(name);

m.settype(type);

m.setprice(price);

return m;

}

public static void cheapItem()

{

string name = MenuItemDL.viewCheapestItem();

Console.WriteLine("Cheapest Item in Menu is: " + name);

}

public static string AddOrder()

{

Console.WriteLine("Enter Item Name: ");

string name = Console.ReadLine();

return name;

}

}

**Complete Main Code**

using System;

using CoffeShop.UI;

using CoffeShop.DL;

using CoffeShop.BL;

namespace CoffeShop

{

class Program

{

static void Main(string[] args)

{

//Start Of Main

string ShopName;

string path = @"E:\Programming\2nd semester\OOP\SIX PROJECTS\CoffeShop\MenuItem.txt";

MenuItemDL.readDataFromFile(path);

Console.Write("Enter Shop Name: ");

ShopName = Console.ReadLine();

CoffeeShopBL n = new CoffeeShopBL(ShopName);

n.setname(ShopName);

Console.Clear();

string op;

//start of while loop

while(true)

{

Console.Clear();

MenuUI.Header(ShopName);

op = MenuUI.Menu();

Console.Clear();

// if user enter 1

// add item in Menu

if(op == "1")

{

MenuUI.Header(ShopName);

MenuItemBL m = MenuItemUI.takeMenuItem();

MenuItemDL.addIntoList(m);

MenuItemDL.StoreIntoFile(path,m);

}

//if user enter 2

//View Cheap Item

else if(op == "2")

{

MenuUI.Header(ShopName);

MenuItemUI.cheapItem();

}

//if user enter 3

//View Drink Menu

else if (op == "3")

{

MenuUI.Header(ShopName);

MenuItemDL.viewDrinks();

}

//if user enter 4

// View Food Menu

else if (op == "4")

{

MenuUI.Header(ShopName);

MenuItemDL.viewFood();

}

//if user enter 5

//take Order From User

else if (op == "5")

{

MenuUI.Header(ShopName);

string name = MenuItemUI.AddOrder();

MenuItemDL.addOrderIntoList(name);

}

//if user enter 6

//fulfill order

else if (op == "6")

{

MenuUI.Header(ShopName);

MenuItemDL.fulfillOrder();

}

//if user enter 7

//Show Order List

else if (op == "7")

{

MenuUI.Header(ShopName);

MenuItemDL.showOrderList();

}

//if user enter 8

//Total Bill

else if (op == "8")

{

MenuUI.Header(ShopName);

MenuItemDL.amount();

}

//if user enter 9

//Exit

else if (op == "9")

{

break;

}

//for any other option

else

{

MenuUI.Header(ShopName);

MenuUI.WrongInput();

}

Console.ReadKey();

}//End of While Loop

}//End Of Main

}

}