

# Tema 4

Moldovan Radu Octavian  
Grupa 30224

## **CUPRINS**

1. Obiectivul temei
2. Analiza problemei, modelare, scenarii, cazuri de utilizare
3. Proiectare
4. Implementare
5. Concluzii
6. Bibliografie

# Obiectivul temei

## Obiectivul principal:

Obiectivul principal al lucrării îl reprezintă realizarea unei aplicații ce simulează procesul de administrare al unei companii de food delivery. Deși modelul este unul simplificat, el trebuie să fie cât se poate de corect și apropiat de cazul real al prelucrării datelor clienților și ale restaurantului, pentru a putea oferi utilizatorului, adică clientului o înțelegere profundă și o interacțiune facilă cu aplicația.

Aspectele menționate mai sus au fost luate în calcul și la proiectarea interfeței grafice, creându-se astfel un GUI ușor de utilizat de către utilizator. Aplicația permite accesul unui singur administrator, unui singur angajat și mai multor clienți.

## Obiective secundare:

### 1. Utilizarea modelului de design compozit

- Această parte este descrisă în secțiunea de implementare
- Acest model de proiectare constă din 2 clase și o interfață și a fost utilizat pentru clasele `BaseProduct` și `CompositeProduct`, bot implementând interfața `MenuItem`
- Important de menționat că numărul de produse dintr-un produs compozit este 4 (aceasta a fost o decizie de implementare asumată de autor)

### 2. Implementarea interfeței `IDeliveryService` interface

- Care conține toate operațiunile care pot fi efectuate pe datele restaurantului și care sunt implementate în clasa `DeliveryService`

### 3. Definirea clasei `DeliveryService`

- Aceasta conține operațiunile care pot exista în sistemul de gestionare
- În implementarea acestui punct, au fost identificate mai multe obiective

### 4. Folosind `HashMap`

- `HashMap` a fost folosit pentru avantajul accesului rapid la memorie, și anume  $O(1)$
- Pentru a funcționa, este necesar ca unele metode să fie suprascrise

### 5. Crearea unei interfețe grafice pentru utilizator

- A fost practicat înainte, într-o misiune anterioară

### 6. Utilizarea serializării

- Este folosită pentru stocarea și încărcarea datelor despre restaurant, și anume meniul și comenzile plasate într-un fișier între rulările succesive ale aplicației

**Serializare**= transformarea unui obiect într-o secvență de octeți, din care se poate să fie refăcut ulterior obiectul serializat.

Procesul invers, de citire a unui obiect serializat pentru a-i reface starea originală, se numește **deserializare**.

## Analiza problemei, modelare, scenarii, cazuri de utilizare

Aplicatia are rolul de a indeplini sarcini de management specifice oricarui restaurant sau serviciu de livrare. Pentru realizarea aplicatiei au fost luate in calcul scenarii diferite, pentru administrator, client si angajat.

In cazul administratorului: printre capabilitatile administratorului se numara adaugarea, stergerea si actualizarea datelor despre produse, adaugarea de produse compuse, posibilitatea de a cauta produse in baza unui sistem de filtre si generarea de rapoarte referitoare la produse, clienti si comenzi. La prelucrarea datelor introduse de la tastatura se foloseste un sistem de validare bazat pe exceptii pentru a asigura corectitudinea datelor preluate.

In cazul mecanismului de login: pentru administrator si angajat exista cate un cont, insa pentru clienti exista posibilitatea de a crea conturi diferite, cu conditia ca acestea sa aiba nume de utilizator diferite. De asemenea, incercarea de a accesa aplicatia folosind credentiale invalide este semnalata utilizatorului prin aparitia unui mesaj de eroare.

In cazul operatiunilor de creare, stergere, actualizare de produse si creare de produse compuse: introducerea de date eronate sau referentierea unui produs inexistent in cadrul listei de produse disponibile nu este permisa si este o situatie tratata cu ajutorul functiilor si exceptiilor.

Meniul restaurantului contine doua tipuri de produse: produse de baza si produse compuse. In alcatuirea produselor compuse se regasesc unul sau mai multe produse de baza, insa pentru a simplifica lucrul cu interfata grafica, GUI permite introducerea a pana la patru produse de baza pentru a crea un produs compus.

La pornirea aplicatiei, utilizatorului ii este prezentata fereastra de login, care contine casetele de text aferente numelui de utilizator si parolei, doua butoane pentru inregistrarea unui nou cont de client si logare, precum si o lista de butoane radio utilizata pentru a selecta cele trei moduri de utilizare ale aplicatiei: administrator, client sau angajat. Vom oferi o scurta descriere a capabilitatilor pe care le ofera fiecare mod de utilizare:

**Administratorul** dispune de urmatoarele optiuni:

- Import CSV – lista de produse disponibile va fi incarcata dintr-un fisier csv, care contine toate datele necesare pentru realizarea operatiilor cu produse.
- Create product – crearea unui produs de baza
- Delete product – stergerea unui produs din lista
- Update product – actualizarea datelor despre un produs de baza
- Create composite – crearea unui produs compus din mai multe produse de baza
- Filter products – cautarea produselor din lista care indeplinesc conditiile furnizate in fereastra de filtrare (nume, pret, calorii, proteine, grasimi, sodiu, rating etc)
- Generarea de rapoarte : time interval of the orders – a report should be generated with the orders performed
  - o between a given start hour and a given end hour regardless the date.
  - o the products ordered more than a specified number of times so far.
  - o the clients that have ordered more than a specified number of times so far and the value of the order was higher than a specified amount.
  - o the products ordered within a specified day with the number of times they have been ordered

**Clientul** dispune de urmatoarele optiuni:

- Filter products - cautarea produselor din lista care indeplinesc conditiile furnizate in fereastra de filtrare (nume, pret, calorii, proteine, grasimi, sodiu, rating etc)
- Add to order - plasarea unui produs din meniu in lista de produse ce urmeaza sa intre in alcatuirea unei comenzi
- Remove from order – inlaturarea unui produs existent in lista de produse ce urmeaza sa intre in alcatuirea unei comenzi
- Confirm order – plasarea comenzii create si generarea unei chitante ce contine informatii despre comanda plasata; in cazul comandarii de produse compuse este prezentata componenta acestora. De asemenea, se retine momentul de creare al comenzii.

**Angajatul** dispune de urmatoarele opriuni:

- De fiecare data cand se creeaza o noua comanda, angajatul este notificat de realizarea acestui eveniment, iar comanda devine vizibila pe lista comenzilor deja existente, ce ofera informatii precum: ID-ul comenzii, ID-ul clientului care a facut comanda, timpul la care s-a facut comanda, continutul comenzii.

Aplicația propriu-zisă este formată dintr-o fereastră principală de autentificare și mai multe ferestre secundare, specifice fircarei persoane, în funcție de drepturile acestuia.

#### Fereastră principală:

- Register client :
  - Folosit pentru înregistrarea unui client, în cazul în care acesta nu este deja înregistrat.
- Log in:
  - Folosit pentru logarea clientului, în cazul în care datele lui corespund cu cele din memoria aplicației.

#### Administrator:

Title	Rating	Calories	Protein	Fat	Sodium	Price	Orders
***Blanketed** Eg...	3.75	1386	105	98	578	88	0
***Bloody Mary**...	5.0	189	2	16	517	16	0
***Brown on Blon...	0.0	321	5	18	120	96	0
***California Roll**...	4.375	369	9	13	953	67	0
***Candy Corn**...	2.5	251	6	9	88	90	0
***Cannoli** Ice Cr...	3.75	379	7	20	289	34	0
***Cocotte** of Ve...	3.75	196	8	14	272	93	0
***Drunken** Pork...	3.75	153	12	8	40	97	0
***La Brea Tar Pit...	3.75	410	34	24	1554	92	0
***Like a Caesar**...	4.375	1600	214	71	1366	72	0
***Opulent** Chic...	4.375	660	20	49	1025	63	0
***Paella** Fried R...	4.375	505	13	27	1100	31	0
***Redeye** Brais...	4.375	1291	79	89	884	23	0
***Route 66** Chili	3.125	640	57	32	1346	16	0
***Seder Plate** S...	0.0	260	8	24	179	77	0
***Seethed** Mus...	0.0	226	27	8	870	31	0
***Toni's Marinade...	4.375	277	2	28	629	98	0
***Virgin Mary** A...	0.0	180	5	11	479	78	0
**101 **Whaler** Fi...	4.375	819	27	65	941	65	0
*Alaska King Crab...	4.375	69	4	3	187	38	0
*Apple **Flag** Ts...	1.25	304	2	10	174	17	0

Title	Rating	Calor.	Protein	Fat	Sodi...	Price	Ord...
Grilled Zucchini and Yogurt Dip	4.375	152	3	13	694	13	0
Grilled Zucchini with Garlic and Lemon	4.375	129	2	12	13	90	0
Grilled-Fish Tacos	3.125	547	20	32	452	95	0
Grilled-Vegetable Pasta Salad with Pa	3.75	471	10	24	26	64	0
Grits Aligot	0.0	266	12	17	443	83	0
Grits with Goat Cheese and Chives	4.375	152	6	5	637	16	0
Grits with Shrimp and Roasted Red B	4.375	291	14	14	864	78	0
Grog	0.0	173	1	0	3	28	0
Ground Coriander and Cilantro Flatbre	4.375	142	2	13	429	84	0
Ground Sirloin Patties with Red Wine	3.75	510	24	34	651	98	0
Grouper with Tomato and Basil	4.375	336	44	16	413	20	0
Gruyère Cheese Soufflé with Dill	4.375	589	18	17	625	94	0
Gruyère Potato Gratin	3.125	445	24	20	432	25	0
Livêre and Parmesan Cheese Soufflê	4.375	651	29	38	2640	83	0
Guacamole	4.375	172	2	15	9	71	0
Guacamole Tacos with Tomatillo and	4.375	1456	153	79	2248	60	0

Filter products

Add to order

Remove from order

Confirm order

Available product and products to be ordered

Title	Rati...	Cal...	Pro...	Fat	Sod...	Price	Orders
Grilled Skirt Steak	4.375	447	46	29	1579	14	0
Gruyère Cheese Soufflé with Dill	4.375	589	18	17	625	94	0
Gruyère and Parmesan Cheese ...	4.375	651	29	38	2640	83	0

Title contains

Rating between

Calories between

Protein between

Fat between

Sodium between

Price between

Find products

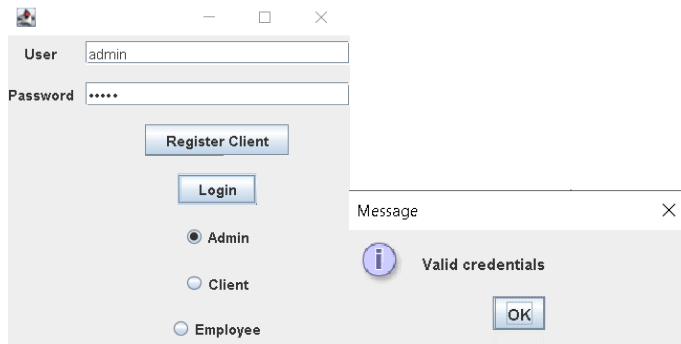
[illegible]

### Cazuri de utilizare:

Inițial clientul/administratorul/angajatul se loghează utilizând datele lui personale iar pe urmă este executată o nouă interfață în funcție de drepturile acestuia: Utilizatorul

Se presupune că se loghează un client. Acesta poate fie să se înregistreze, fie să se logheze cu datele lui corespunzătoare.

Dacă înregistrarea a avut loc cu succes atunci se apasă pe log in și se continuă către fereaștră specifică lui.





Se cauta produsele dorite:

Filter products

Add to order

Remove from order

Confirm order

Available product and products to be ordered

Title	Rating	Calor...	Protein	Fat	Sodi...	Price	Ord...		Title	Rati...	Cal...	Pro...	Fat	Sod...	Price	Orders	
Grilled Zucchini and Yogurt Dip	4.375	152	3	13	694	13	0	▲	Grilled Skirt Steak	4.375	447	46	29	1579	14	0	▲
Grilled Zucchini with Garlic and Lemon	4.375	129	2	12	13	90	0		Gruyère Cheese Soufflé with Dill	4.375	589	18	17	625	94	0	■
Grilled-Fish Tacos	3.125	547	20	32	452	95	0		Gruyère and Parmesan Cheese ...	4.375	651	29	38	2640	63	0	
Grilled-Vegetable Pasta Salad with Pa	3.75	471	10	24	26	64	0										
Grits Aligot	0.0	266	12	17	443	83	0										
Grits with Goat Cheese and Chives	4.375	152	6	5	637	16	0										
Grits with Shrimp and Roasted Red B	4.375	291	14	14	864	78	0	■									
Grog	0.0	173	1	0	3	28	0										
Ground Coriander and Cilantro Flatbre	4.375	142	2	13	428	84	0										
Ground Sirloin Patties with Red Wine	3.75	510	24	34	651	98	0										
Grouper with Tomato and Basil	4.375	336	44	16	413	20	0										
Gruyère Cheese Soufflé with Dill	4.375	589	18	17	625	94	0										
Gruyère Potato Gratin	3.125	445	24	20	432	25	0										
Gruyère and Parmesan Cheese Soufflé	4.375	651	29	38	2640	63	0										
Guacamole	4.375	172	2	15	9	71	0										
Guacamole Tacos with Tomatillo and	4.375	1456	153	79	2248	60	0	▼									▼

Title contains

abc

Rating between

0

99999

Calories between

0

99999

Protein between

0

99999

Fat between

0

99999

Sodium between

0

99999

Price between

0

99999

Find products

Se apasa pe search iar pe urma se plaseaza o comanda, selectand unul sau mai multe produse din tabela:

Filter products

Add to order

Remove from order

Confirm order

Available product and products to be ordered

Title	Rating	Calori...	Protein...	Fat	Sodium...	Price	Orders
"Grilled Scallop "Ceviche" "	5.0	160	14	7	835	66	0
"Herbed Polenta "Fries" "	4.3	283	8	14	206	69	0
"Hershey's "Perfectly Chocolate" Chocolate Cake	4.3	6145	68	213	56	15	0
"Irish "Bacon" and Cabbage "	3.75	1046	69	50	733	29	0
"Israeli Yogurt "Cheese" with Pita and Olives "	5.0	155	6	10	290	65	0
"Jamaican "Jerk" Chicken Wings "	4.3	12	907	919	52	86	0
"Lamb and Polenta "Lasagne" "	5.0	942	69	60	11	13	0
"Lobster "Potpies" "	5.0	305	11	16	28	86	0
"Meyer Lemon "Jam" "	3.1	177	1	14	3	37	0
"Mixed Leafy Green Soup "Caldo Verde" with Chic	5.0	273	17	6	821	45	0
"New England Blackberry "Torte" "	3.75	394	5	18	173	14	0
"Oysters Rockefeller "Deconstructed" "	5.0	887	10	8	13	28	0
"Pasta "Rags" with a Thousand Herbs "	3.75	833	39	1	1	1	0
"Pineapple "Salsa" "	4.3	55	1	1	1	1	0
"Poisson Cru ("Tahitian Salad") "	3.75	154	19	1	1	1	0
"Porcini Mushroom Latkes with Mushroom Salad an	4.3	354	9	1	1	1	0
"th Gorgonzola Butter and Red Pepper Vinaigrette "	4.3	555	16	1	1	1	0
"Roast Beef for "Beef on Weck" "	4.3	295	49	1	1	1	0
"Sage and Onion "Roast" "	4.3	855	37	1	1	1	0
"Salmon "Bulgogi" with Bok Choy and Mushrooms	4.3	459	39	28	994	60	0
"Salmon with Potato "Scales" "	5.0	883	57	41	72	86	0
"San Francisco Crab "Meatball" Chowder "	2.5	1001	52	55	734	13	0
"Sardine "Tonnato" Spread "	4.3	212	13	17	353	75	0
"Sausage and White Bean "Cassoulet" "	4.3	771	38	48	12	77	0
"Scallop "Porcupines" with Basil-Mint Sauce "	3.75	180	13	4	404	57	0
"Scallops with "Melted" Leeks and Tarragon-Cape	4.3	684	21	42	637	86	0
"Singapore "Carrot Cake" "	3.75	435	11	22	14	73	0
"Smoked Fish with Cucumber "Noodles" "	2.5	1234	26	18	835	112	0

Acum comanda a fost plasa cu succes si urmeaza a fii tiparita chitanta.

Se vizualizeaza chitanta cu id-ul 7 din fisierul text generat:

Order

2022-05-18T23:09:05.823237

Title: Balsamic Strawberries with Whipped Mascarpone Cheese Rating: 4.375 Calories: 183 Protein: 2 Fat:10 Sodium:78 Price: 34

Title: Old-Fashioned Rating: 5.0 Calories: 222 Protein: 0 Fat:0 Sodium:0 Price: 83

Title: Herb Syrup Rating: 0.0 Calories: 198 Protein: 0 Fat:0 Sodium:1 Price: 95


Total: 212

Un alt caz de utilizare ar fii modificarea datelor unui produs din tabelul de produse de catre un administrator:

Filter products								Import CSV
Time interval								Create product
Ordered products								Update product
Clients								Delete product
Products in a day								Create composite
Title	Rating	Calories	Protein	Fat	Sodium	Price	Orders	
***Blanketed** Eg...	3.75	1386	105	98	578	88	0	
***Bloody Mary**...	5.0	189	2	16	517	18	0	
***Brown on Blon...	0.0	321	5	18	120	96	0	
***California Roll**	4.375	369	9	13	953	67	0	
***Candy Corn**...	2.5	251	6	9	88	90	0	
***Cannoli** Ice Cr...	3.75	379	7	20	289	34	0	
***Cocotte** of Ve...	3.75	196	8	14	272	93	0	
***Drunken** Pork...	3.75	153	12	8	40	97	0	
***La Brea Tar Pit...	3.75	410	34	24	1554	92	0	
***Like a Caesar**...	4.375	1600	214	71	1366	72	0	
***Opulent** Chic...	4.375	660	20	49	1025	83	0	
***Paella** Fried R...	4.375	505	13	27	1100	31	0	
***Redeye** Brais...	4.375	1291	79	89	894	23	0	
***Route 66** Chili	3.125	640	57	32	1346	16	0	
***Seder Plate** S...	0.0	280	8	24	179	77	0	
***Seethed** Mus...	0.0	226	27	8	870	31	0	
***Toni's Marinate...	4.375	277	2	28	629	98	0	
***Virgin Mary** A...	0.0	180	5	11	479	78	0	
*101 **Whaler** Fi...	4.375	819	27	65	941	85	0	
*Alaska King Crab	4.375	69	4	3	187	38	0	
*Apple **Flag** Ta...	1.25	304	2	10	174	17	0	

Se incepe prin logarea administratorului si prin selectarea unui produs din tabela de produse. Se apasa pe update si se modifica campurile de Price si Rating si se apasa pe updates:

Title	***Candy Corn*** Frozen Citrus Cream Pops *		
Rating	5		
Calories	251		
Protein	6		
Fat	9		
Sodium	88		
Price	120		
<div> <div>Message</div> <div> <div>Product updated</div> <div>OK</div> </div> </div>			
<div>Confirm</div>			

<div>  <div> <div>Filter products</div> <div>Time interval</div> <div>Ordered products</div> <div>Clients</div> <div>Products in a day</div> </div> </div>								<div> <div>Import CSV</div> <div>Create product</div> <div>Update product</div> <div>Delete product</div> <div>Create composite</div> </div>
Title	Rating	Calories	Protein	Fat	Sodium	Price	Orders	
""Blanketed"" Eg...	3.75	1386	105	98	578	88	0	
""Bloody Mary"" ...	5.0	189	2	16	517	16	0	
""Brown on Blon...	0.0	321	5	18	120	96	0	
""California Roll""	4.375	369	9	13	953	67	0	
""Candy Corn"" ...	5.0	251	6	9	88	120	0	
""Cannoli"" Ice Cr...	3.75	379	7	20	289	34	0	
""Cocotte"" of Ve...	3.75	196	8	14	272	93	0	
""Drunken"" Pork...	3.75	153	12	8	40	97	0	
""La Brea Tar Pit...	3.75	410	34	24	1554	92	0	
""Like a Caesar"" *	4.375	1600	214	71	1366	72	0	
""Opulent"" Chic...	4.375	660	20	49	1025	83	0	
""Paella"" Fried R...	4.375	505	13	27	1100	31	0	
""Redeye"" Brais...	4.375	1291	79	89	884	23	0	
""Route 66"" Chili	3.125	640	57	32	1346	16	0	
""Seder Plate"" S...	0.0	280	8	24	179	77	0	
""Seethed"" Mus...	0.0	226	27	8	670	31	0	
""Toni's Marinade...	4.375	277	2	28	629	98	0	
""Virgin Mary"" A...	0.0	180	5	11	479	78	0	
*101 ""Whaler"" Fi...	4.375	819	27	65	941	85	0	
*Alaska King Crab	4.375	69	4	3	187	38	0	
*Apple ""Flag"" Ta...	1.25	304	2	10	174	17	0	
*Apple ""Pizza"" *	3.75	234	6	11	201	21	0	
*Avocado ""Pesto..."	3.125	354	3	36	37	47	0	
*Barley ""Risotto""	3.75	398	14	18	305	27	0	
*Blackberry ""Cob...	4.375	445	4	16	11	54	0	
*Brioche ""Bread ...	5.0	405	4	21	51	84	0	
*Buttermilk Cabb...	0.0	200	9	11	1195	10	0	

# Proiectare

**Implementarea OOP** a aplicației propriu-zise constă în utilizarea modelului Layered Architecture (Business layer, Data Access Layer, Presentation Layer), pentru o proiectare concisă și o structurare a tuturor datelor în mod uniform.

Pachetul **business** cuprinde mai multe clase printre care se numara:

**DeliveryServiceProcessing, Order, Client, BaseProduct, CompositeProduct.**

- Aceste clase realizeaza defapt legatura dintre pachetul **controller** si operatiile pe obiectele de date serializate, propriu zise. Concret, cea mai importanta clasa este **DeliveryServiceProcessing**, ce contine operatiile pe datele serializate cum ar fii, orders, reports si lista de produse. Aceasta este formata din metodele:
  - public ArrayList<MenuItem> importMenu(String file)
  - public void addProduct(MenuItem product)
  - public void updateProduct(String title, MenuItem product)
  - public void deleteProduct(String title)
  - public void createComposite(String title, ArrayList<MenuItem> items)
  - public void notifyObserver(ObserverInterface obs)

Pachetul **datalayer** contine toate operatiile de serializare propriu-zise realizate pe datele din tabela de produse.

- Clasa **Serializator**, utilizata pentru realizarea tuturor operatiilor comune pe fiecare tabela din baza de date.
  - Aceasta contine urmatoarele metode:
    - public static void saveDsp(DeliveryServiceProcessing dsp, String file)
    - public static DeliveryServiceProcessing retrieveDsp(String file)

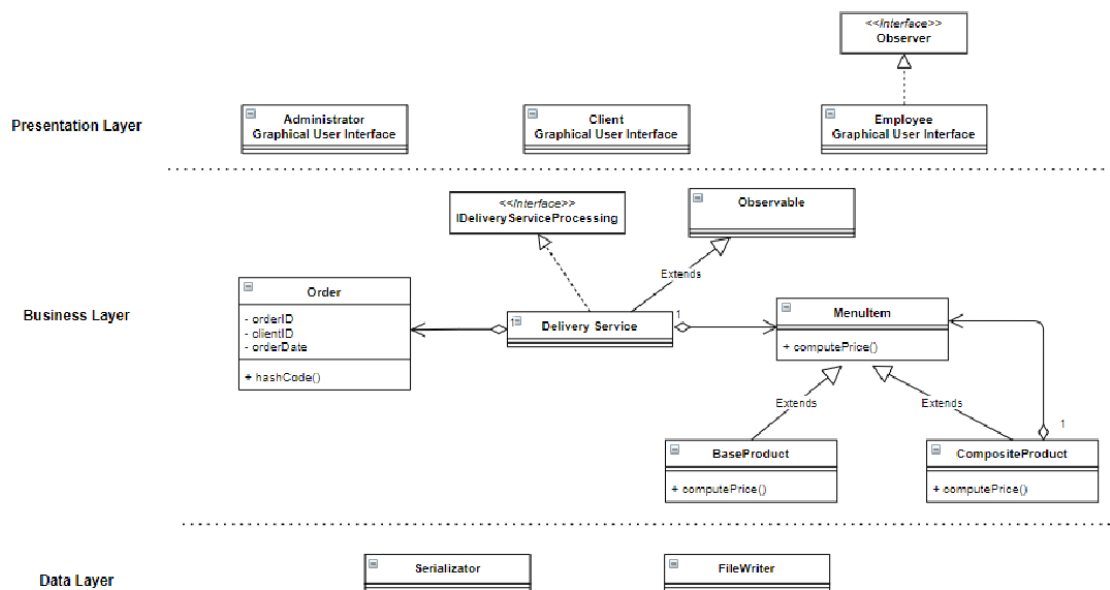
Pachetul **business**, contine de asemenea clasele **MenuItem, BaseProduct, CompositeProduct**. **MenuItem** reprezinta defapt o interfata care contine metode, ce vor fii implementate ulterior in clasele BaseProduct si CompositeProduct. In BaseProduct acest compute actioneaza ca un getter dar, in clasa de CompositeProduct, pretul, ca de exemplu, este calculat ca suma pretului tuturor produselor continute in composite product.

Pachetul **presentation**, contine clasele responsabile pentru functionarea interfetelor grafice pentru **administrator, client, employee**. Aceste clase contin la randul lor cate un controller care fac o oarecare legatura cu clasa **MainController** din pachetul controller.

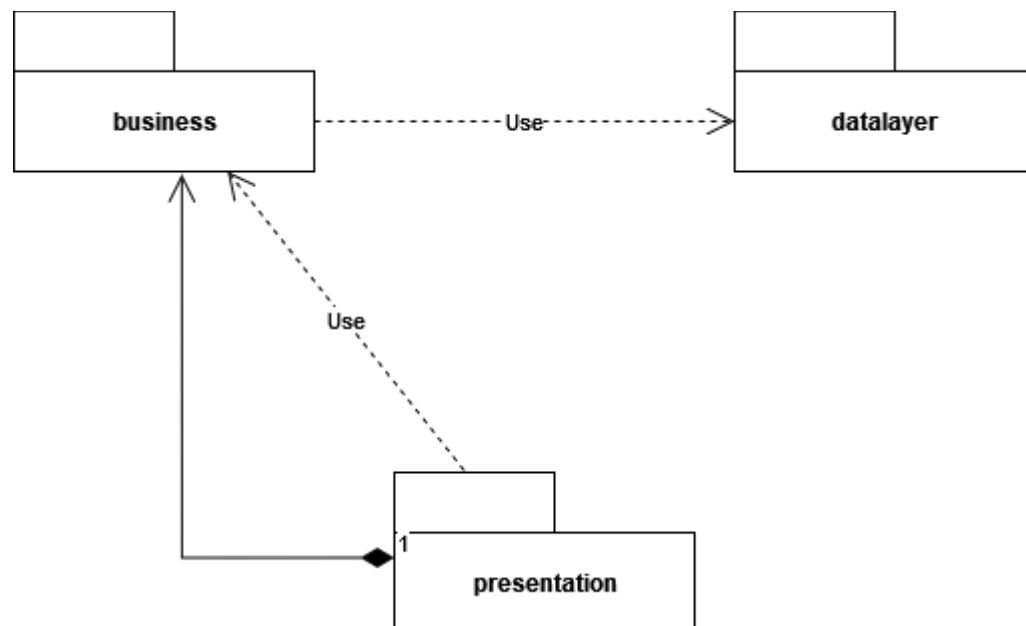
- Din clasa **Controller** se remarca urmatoarele metode:
  - public Controller(DeliveryServiceProcessing dsp)
- Din clasa **ClientController** se remarca urmatoarele metode:
  - public ClientController(DeliveryServiceProcessing dsp, ClientGui clientGui, EmployeeController employeeController)
  - public void showMenuClient(ArrayList<MenuItem> items)
  - public void showItemsClient(ArrayList<MenuItem> items)
  - public void filterButtonListener()
  - public void addItemClientListener()
  - public void removeItemClientListener()
  - public void confirmOrderClientListener()
- Din clasa **AdminController** se remarca urmatoarele metode:
  - public **AdminController**(MainController mainController, DefaultTableModel productsTableModel, ProductsDataSerializator productsDataSerializator, DeliveryService deliveryService);
  - public AdminController(DeliveryServiceProcessing dsp, AdminGui adminGui)
  - public void showMenuAdmin(ArrayList<MenuItem> items)
  - public void showClientsAdmin(ArrayList<Client> clients)
  - public void showOrdersAdmin(ArrayList<Order> orders)
  - public void createButtonListener()
  - public void updateButtonListener()
  - public void deleteButtonListener()

- public void importButtonListener()
- public void filterButtonListener()
- public void createCompositeButtonListener()
- public void productNumberReportListener()
- public void clientFilterReportListener()
- public void hourReportListener()
- public void productDayReport()
- public void addAdminListeners()
- Din clasa **LoginController** se remarca urmatoarele metode:
  - public LoginController(DeliveryServiceProcessing dsp, LoginGui loginGui, AdminGui adminGui, ClientGui clientGui, EmployeeGui employeeGui)
  - public void loginButtonListener()
  - public void registerButtonListener()
  - public void addLoginListeners()
- Din clasa **EmployeeController** se remarca urmatoarele metode:
  - public EmployeeController(DeliveryServiceProcessing dsp, EmployeeGui employeeGui)
  - public void update()
  - public String itemsToString(ArrayList<MenuItem> items)

**Diagrama UML a claselor:** pentru realizarea aplicatiei s-a folosit diagrama de clase oferita in cerinta temei



**Diagrama de pachete:**



# Implementare

Descrierea metodelor importante din unele clase:

- În clasa **DeliveryService**:
  - Metoda **importMenu** (...) – este utilizata pentru a importa datele despre produse, dintr-un fisier CSV selectat de catre administrator. Datele din fisier sunt citite cu ajutorul unei stream-urilor, linie cu linie, fiind adaugate toate produsele in obiectul de tip HashMap al produselor, format dintr-o cheie de tip String, ce reprezinta numele produsului, si valoarea acestuia ce reprezinta obiectul propriu-zis.
  - Metoda **addProduct** (...) – este utilizata pentru a adauga un produs nou in colectia de produse. Initial se incearca sa se verifice daca exista deja o cheie ce are ca titlu, titlul produsului nou adaugat. Daca acesta exista deja, adaugarea nu este realizata. Daca nu exista, se adauga produsul in colectie si dupa acesta este serializat.
  - Metoda **updateProduct** (...) – este utilizata pentru a modifica un produs din colectia de produse. Aici este apelata functia de addProduct pe obiectul modificat si se realizeaza serializarea.
  - Metoda **deleteProduct** (...) – este utilizata pentru a sterge un produs din colectia de produse. Initial se incearca sa se verifice daca exista deja o cheie ce are ca titlu, titlul produsului ce urmeaza a fii sters. Daca produsul exista deja atunci stergerea poate fii realizata si colectia este serializata. Daca nu exista atunci stergerea nu are loc.
  - Metoda **createComposite** (...) – este utilizata pentru a adauga un produs compus din mai multe produse in colectia de produse. Daca produsul nou adaugat nu are acelasi titlu cu o alta cheie din colectia de produse, atunci se apeleaza addProduct si se realizeaza adaugarea acestuia in colectia de produse.
  - Metoda **createOrder** (...) – este utilizata pentru a adauga o noua comanda in colectia de comenzi. Initial se creeaza un nou obiect de tip order, cu id-ul lui specific, clientID si data curenta la care se plaseaza comanda.
- În clasele **Controller (AdminController, ClientController, EmployeeController)**:
  - **Metode de tip show**: acestea au rolul de a seta continutul tabelelor utilizate pentru a afisa informatia despre produse, clienti si comenzi
  - **Metode de tip listener**: au rolul de a apela metoda corespunzatoare functionalitatii dorite la apasarea unui buton, si de a apela functiile care sunt responsabile pentru preluarea datelor de intrare din interfata grafica



## **Concluzii**

In concluzie, consider ca cele mai importante aspecte pe care le-am invatat realizand acest proiect le reprezinta lucrul cu functii lambda si utilizarea sistemului de serializare oferit de Java.

## ***Bibliografie***

- [1] <http://javahungry.blogspot.com/2013/08/hashing-how-hash-map-works-in-java-or.html>
- [2] <http://javahungry.blogspot.com/2014/03/hashmap-vs-hashtable-difference-with-example-java-interview-questions.html>
- [3] <http://javahungry.blogspot.com/2013/08/how-sets-are-implemented-internally-in.html> [4] [https://en.wikipedia.org/wiki/Red%E2%80%93black\\_tree](https://en.wikipedia.org/wiki/Red%E2%80%93black_tree)
- [5] [https://en.wikipedia.org/wiki/Hash\\_table](https://en.wikipedia.org/wiki/Hash_table)
- [6] <https://www.baeldung.com/java-serialization>
- [7] <https://www.geeksforgeeks.org/serialization-in-java/>
- [8] <https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>