

# **DOCUMENTATIE**

## **TEMA 4**

NUME STUDENT: POROJAN MĂDĂLIN MARIAN  
GRUPA: 30227

## CUPRINS

1. Obiectivul temei.....	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare .....	4
3. Proiectare .....	10
4. Implementare.....	14
5. Rezultate.....	14
6. Concluzii .....	17
7. Bibliografie .....	18

## **1. Obiectivul temei**

### **Obiectivul principal:**

Obiectivul principal presupune crearea unei aplicatii usor de utilizat, pentru gestionarea comenzilor clientilor dintr-un depozit.

### **Obiectivele secundare:**

- Verificarea input-ului introdus pentru a asigura o legatură unu la unu, între ceea ce este introdus în casetele text, corespunzătoare parametrilor de intrare (add, remove, modify, compose product), și ceea ce se află în memoria aplicației;
- Utilizarea de tabbed pane-uri pentru a separa interfata administratorului în operatii pe tabela de produse si în generarea de rapoarte corespunzătoare produselor.
- Modificarea produselor doar prin selectarea acestora din tabela de produse.
- Interceptarea tuturor erorilor pentru a evita o situație în care programul se află într-o stare necunoscută;
- Informarea utilizatorului despre erori prin pop-up-uri si coduri de eroare care pot fi depanate mai departe consultând un manual specific pentru erori;
- Informarea utilizatorului cu privință la corectitudinea input-ului introdus de la tastatură prin pop-up-uri;

## **2. Analiza problemei, modelare, scenarii, cazuri de utilizare**

### **Analiza problemei:**

Ideea principală în realizarea aplicației de gestionare este de a pune la dispoziție utilizatorului o interfață ușor de utilizat, prin comenzi și o modalitate de operare sugestivă. Interfața grafică dispune de ferestre specifice pentru fiecare operație de inserare, ștergere, modificare sau crearea unei comenzi, tabelele din baza de date fiind disponibile clientului/administratorului/angajatului ce utilizează aplicația.

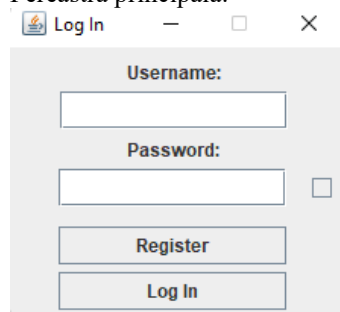
### **Scenarii posibile:**

1. Introducerea datelor în mod eronat, care ar putea face realizarea producerii unei excepții la inserarea/ștergere/modificarea datelor în/din tabel:
  - Introducerea de numere negative în casetele destinate pentru stock/amount;
  - Introducerea de cifre în numele produsului;
  - Introducerea unui titlu de produs eronat;
  - Se activează un pop-up ce semnalează că datele nu sunt introduse corect;
2. Neselectarea coloanei care urmează a fi modificată prin butonul de update:
  - Această problemă este verificată și semnalată printr-un pop-up ce indică selectarea unei coloane din tabela respectivă.
3. Introducerea parțială a unor date de input:
  - În acest caz, este activat un pop-up ce semnalează că, casetele de text respective sunt goale.

## Modelare:

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

Fereastră principală:

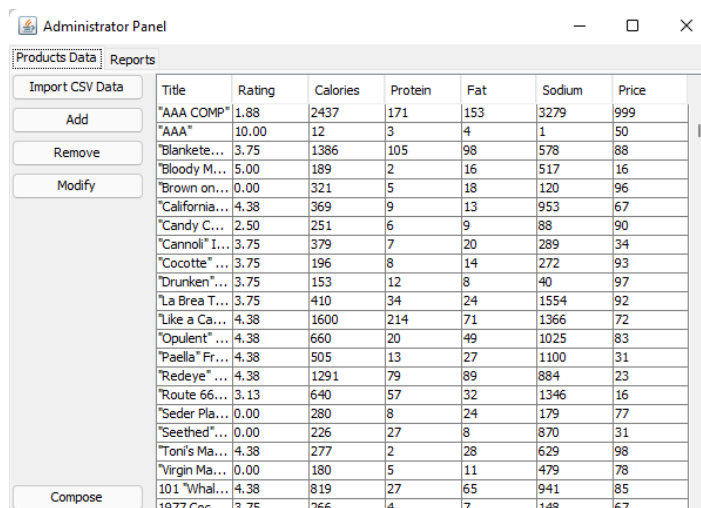


Este formată din două butoane:

- Register:
  - Folosit pentru înregistrarea clientului, 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.

În cazul în care este selectată casuta din dreapta casetei text de parola, parola devine vizibilă, nemaifiind ascunsă de alte caractere.

Administrator:



Title	Rating	Calories	Protein	Fat	Sodium	Price
"AAA COMP"	1.88	2437	171	153	3279	999
"AAA"	10.00	12	3	4	1	50
"Blankete..."	3.75	1386	105	98	578	88
"Bloody M..."	5.00	189	2	16	517	16
"Brown on..."	0.00	321	5	18	120	96
"California..."	4.38	369	9	13	953	67
"Candy C..."	2.50	251	6	9	88	90
"Cannoli" I...	3.75	379	7	20	289	34
"Cocotte" ...	3.75	196	8	14	272	93
"Drunken"...	3.75	153	12	8	40	97
"La Brea T..."	3.75	410	34	24	1554	92
"Like a Ca..."	4.38	1600	214	71	1366	72
"Opulent" ...	4.38	660	20	49	1025	83
"Paella" Fr...	4.38	505	13	27	1100	31
"Redeye" ...	4.38	1291	79	89	884	23
"Route 66..."	3.13	640	57	32	1346	16
"Seder Pla..."	0.00	280	8	24	179	77
"Seethed"...	0.00	226	27	8	870	31
"Toni's Ma..."	4.38	277	2	28	629	98
"Virgin Ma..."	0.00	180	5	11	479	78
101 "Whal..."	4.38	819	27	65	941	85
1977 Coc...	3.75	766	4	7	148	67

Administrator Panel

Products Data
Reports

Generate report based on:

☒ Time interval of orders:

Start hour:
End hour:

☐ The products:

Ordered more than:

☐ The clients:

Ordered more than:
Value higher than:

☐ The products ordered:

Monday

View Report

Client:

Client Manager

Title	Rating	Calories	Protein	Fat	Sodium	Price
"AAA CO...	1.88	2437	171	153	3279	999
"AAA"	10.00	12	3	4	1	50
"Blankete...	3.75	1386	105	98	578	88
"Bloody M...	5.00	189	2	16	517	16
"Brown o...	0.00	321	5	18	120	96
"Californi...	4.38	369	9	13	953	67
"Candy C...	2.50	251	6	9	88	90
"Cannoli" ...	3.75	379	7	20	289	34
"Cocotte" ...	3.75	196	8	14	272	93
"Drunken...	3.75	153	12	8	40	97
"La Brea ...	3.75	410	34	24	1554	92
"Like a Ca...	4.38	1600	214	71	1366	72
"Opulent"...	4.38	660	20	49	1025	83
"Paella" F...	4.38	505	13	27	1100	31
"Redeye"...	4.38	1291	79	89	884	23
"Route 6...	3.13	640	57	32	1346	16
"Seder Pl...	0.00	280	8	24	179	77
"Seethed...	0.00	226	27	8	870	31
"Toni's M...	4.38	277	2	28	629	98
"Virgin Ma...	0.00	180	5	11	479	78
101 "Wha...	4.38	819	27	65	941	85
1977 Coc...	3.75	266	4	7	148	67
1985 Chic...	5.00	1260	67	104	355	56

Search

Place Order

Search Manager

Title

Rating

Calories

Protein

Fat

Sodium

Price

0

0

0

0

0

0

0

5

1000

1000

1000

1000

1000

1000

Search

Angajat:

Employee notifications

The order with ID #1 from the client with ID #1 costing 316 EUR was placed!  
The order with ID #2 from the client with ID #2 costing 641 EUR was placed!  
The order with ID #3 from the client with ID #3 costing 641 EUR was placed!  
The order with ID #4 from the client with ID #4 costing 39 EUR was placed!  
The order with ID #5 from the client with ID #5 costing 240 EUR was placed!

Close

## Cazuri de utilizare:

Inițial clientul/administratorul/angajatul se logheaza utilizand datele lui personale iar pe urma este executata o noua interfata in functie de drepturile acestuia:

Se presupune ca se logheaza un client. Acesta poate fie sa se inregistreze, fie sa se logheze cu datele lui corespunzatoare.

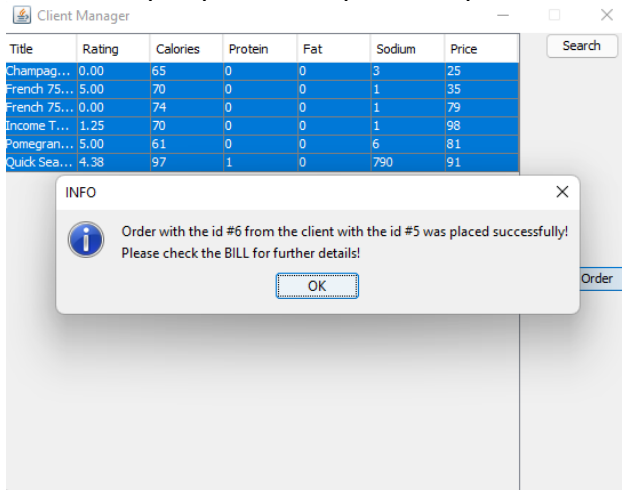
Daca inregistrarea a avut loc cu succes atunci se apasa pe log in si se continua catre fereastra specifica lui.

The image shows a 'Log In' window with a title bar containing a small icon, the text 'Log In', and standard window controls (minimize, maximize, close). Inside the window, there are two input fields: 'Username:' with the text 'test' and 'Password:' with the text '123'. Below these fields are two buttons: 'Register' and 'Log In'. To the right of the input fields, there is a checked checkbox and an 'INFO' icon. A small information dialog box is open over the 'Log In' button, displaying the message 'Registration was successful. Please Log In!' with an 'OK' button.

Se cauta produsele dorite:

The image displays two windows side-by-side. The 'Client Manager' window on the left features a table with columns: Title, Rating, Calories, Protein, Fat, Sodium, and Price. It lists 25 items, including 'AAA CO...', 'AAA\*', 'Blankete...', 'Bloody M...', 'Brown o...', 'Californi...', 'Candy C...', 'Cannoli\*', 'Cocotte\*', 'Drunken...', 'La Brea ...', 'Like a Ca...', 'Opulent\*', 'Paella\* F...', 'Redeye\*', 'Route 6...', 'Seder Pl...', 'Seethed...', 'Toni's M...', 'Virgin Ma...', '101 \*Wha...', '1977 Coc...', and '1985 Chic...'. The 'Search Manager' window on the right has a title bar with a search icon, the text 'Search Manager', and window controls. It includes a 'Search' button at the top. Below it, there is a 'Title' input field containing 'Cocktail'. A series of sliders are shown for 'Rating' (range 0 to 5), 'Calories' (range 50 to 100), 'Protein' (range 0 to 1000), 'Fat' (range 0 to 1000), 'Sodium' (range 0 to 1000), and 'Price' (range 0 to 1000). Each slider has a green bar indicating the selected range. A 'Place Order' button is located to the left of the sliders, and a 'Search' button is at the bottom.

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



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 ID.....#7
Client ID.....#5
Products Ordered.....
    Income Tax Cocktail
    French 75 Cocktail
    Champagne Pineapple Cocktail
    French 75 Cocktail II
    Quick Seafood Cocktail Sauce
    Pomegranate-Ginger Champagne Cocktail

Total Price.....409 EUR
```



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

Administrator Panel

Products Data Reports

Import CSV Data

Add

Remove

Modify

Title	Rating	Calories	Protein	Fat	Sodium	Price
"AAA COMP"	1.88	2437	171	153	3279	999
"AAA"	10.00	12	3	4	1	50
"Blankete..."	3.75	1386	105	98	578	88
"Bloody M..."	5.00	189	2	16	517	16
"Brown on..."	0.00	321	5	18	120	96
"California..."	4.38	369	9	13	953	67
"Candy C..."	2.50	251	6	9	88	90
"Cannoli" I...	3.75	379	7	20	289	34
"Cocotte" ...	3.75	196	8	14	272	93
"Drunken" ...	3.75	153	12	8	40	97
"La Brea T..."	3.75	410	34	24	1554	92
"Like a Ca..."	4.38	1600	214	71	1366	72
"Opulent" ...	4.38	660	20	49	1025	83
"Paella" Fr...	4.38	505	13	27	1100	31
"Redeye" ...	4.38	1291	79	89	884	23
"Route 66..."	3.13	640	57	32	1346	16
"Seder Pla..."	0.00	280	8	24	179	77
"Seethed" ...	0.00	226	27	8	870	31
"Toni's Ma..."	4.38	277	2	28	629	98
"Virgin Ma..."	0.00	180	5	11	479	78
101 "Whal..."	4.38	819	27	65	941	85
1977 Coc	3.75	266	4	7	148	67

Compose

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

Administrator Panel

Title

on Blonde" Blondies

Rating

5

Calories

321

Protein

5

Fat

18

Sodium

120

Price

300

Modify

INFO

Product modified successfully!

OK

Administrator Panel

Products Data Reports

Import CSV Data

Add

Remove

Modify

Compose

Title	Rating	Calories	Protein	Fat	Sodium	Price
"AAA COMP"	1.88	2437	171	153	3279	999
"AAA"	10.00	12	3	4	1	50
"Blankete..."	3.75	1386	105	98	578	88
"Bloody M..."	5.00	189	2	16	517	16
"Brown on..."	5.00	321	5	18	120	300
"California..."	4.38	369	9	13	953	67
"Candy C..."	2.50	251	6	9	88	90
"Cannoli" I...	3.75	379	7	20	289	34
"Cocotte"...	3.75	196	8	14	272	93
"Drunken"...	3.75	153	12	8	40	97
"La Brea T..."	3.75	410	34	24	1554	92
"Like a Ca..."	4.38	1600	214	71	1366	72
"Opulent"...	4.38	660	20	49	1025	83
"Paella" Fr...	4.38	505	13	27	1100	31
"Redeye"...	4.38	1291	79	89	884	23
"Route 66..."	3.13	640	57	32	1346	16
"Seder Pla..."	0.00	280	8	24	179	77
"Seethed"...	0.00	226	27	8	870	31
"Toni's Ma..."	4.38	277	2	28	629	98
"Virgin Ma..."	0.00	180	5	11	479	78
101 "Whal..."	4.38	819	27	65	941	85
1077 Co...	3.75	266	4	7	148	67

Acum produsul „Brown...” a fost modificat cu succes.

### 3. Proiectare

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

#### 1. Pachetul **bll** cuprinde mai multe clase și un pachet **models**. Clasele sunt **DeliveryService**, **Order**, **OrderBLL**, **ProductBLL**.

- Aceste clase realizează defapt legătura dintre pachetul **controller** și operațiile pe obiectele de date serializate, propriu-zise. Concret, cea mai importantă clasă este **DeliveryService**, ce conține operațiile pe datele serializate cum ar fi, orders, reports și lista de produse. Aceasta este formată din metodele:
  - public void **placeOrder**(Map<String, MenuItem> products);
  - public boolean **addCompositeProduct**(CompositeProduct compositeProduct, int customPrice);
  - public void **importBaseProductsDataFromCSV**(File importFile);
  - public boolean **modifyProduct**(MenuItem baseProduct);
  - public boolean **removeProduct**(String baseProduct);
  - public boolean **addProduct**(MenuItem baseProduct);
  - public void **notifyObservers**(int price, int orderID, int clientID);
  - public void **placeOrder**(Map<String, MenuItem> products);

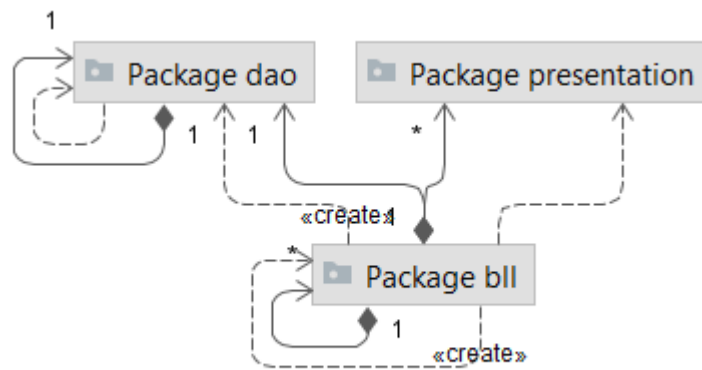
#### 2. Pachetul **dao** conține toate operațiile de serializare propriu-zise realizate pe datele din tabela de produse.

- Clasa **Serializer**, utilizată pentru realizarea tuturor operațiilor comune pe fiecare tabelă din baza de date.
  - Aceasta conține următoarele metode:
    - public void **serialize**(Object object, String serFile)
    - public Object **deserialize**(String serFile)
- Clasele **LoginSerializer**, **LoginInfo**, **ProductsDataSerializer**, folosite pentru implementarea individuală a serializării pe obiectele de autentificare respectiv pe produse:
  - Dintre acestea iese în evidență:
    - LoginInfo**, utilizată pentru a realiza legătura cu ce este introdus în interfața de autentificare și datele din obiectul serializat.
      - Aici se adaugă totodată și metodele: public boolean **isAlreadyRegistered**(String username) și public boolean

**isPasswordGood**(String username, String password), public void **registerUser**(User user), care verifica daca username-ul introdus exista in baza de date, iar daca exista, se verifica daca parola este introdusa corect. Daca este introdusa corect, se realizeaza logarea utilizatorului, altfel se poate alege inregistrarea acestuia;

4. Pachetul **model**, contine clasele **MenuItem**, **BaseProduct**, **CompositeProduct**, **User**. **MenuItem** reprezinta defapt o clasa abstracta care contine metodele abstracte **computePrice**, **computeFat**, **computeProtein**..., 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.
5. Pachetul **presentation**, contine pachetele **administrator**, **client**, **controller**, **employee**. Aceste pachete contin la randul lor cate un controller care fac o oarecare legatura cu clasa **MainController** din pachetul controller.
  - Din clasa **MainController** se remarca urmatoarele metode:
    - public void **setToolTipText**(JTable table);
    - public void **importData**(JTable productsTable);
    - public void **sortTableDataModel**();
    - private void **setAuthenticatorLogInVerifiers**();
    - private void **setAuthenticatorActionListeners**();
    - private void **setAuthenticatorProprieties**();
    - private **MainController**();
  - Din clasa **ClientController** se remarca urmatoarele metode:
    - public **ClientController**(MainController mainController, DefaultTableModel productsTableModel, ProductsDataSerializator productsDataSerializator, DeliveryService deliveryService);
    - private void **setClientUIProprieties**();
    - private void **setSearchProductsUIProprieties**();
    - private void **setSearchProductsVerifiers**();
    - private void **setClientUIActionListeners**();
    - private void **setRangeSlidersActionListeners**();
    - private void **setSearchProductsActionListeners**();
    - private boolean **checkRanges**(Vector row);
    - private void **importFoundData**();
  - Din clasa **AdministratorController** se remarca urmatoarele metode:
    - public **AdministratorController**(MainController mainController, DefaultTableModel productsTableModel, ProductsDataSerializator productsDataSerializator, DeliveryService deliveryService);
    - private void **setAdministratorProprieties**();
    - private void **setReportData**();
    - private void **adminImportData**();
    - private void **setComposeProductProprieties**();
    - private void **setModifyProductProprieties**();
    - private void **setAddBaseProductProprieties**();
    - private void **setReportVerifiers**();
    - private void **setComposeProductVerifiers**();
    - private void **setModifyProductVerifiers**();
    - private void **setAddBaseProductVerifiers**();
    - private void **setComposeProductActionListeners**();
    - private void **setReportActionListener**();
    - private void **setAdministratorActionListeners**();

**Diagrama UML a pachetelor:**





### 3. Implementare

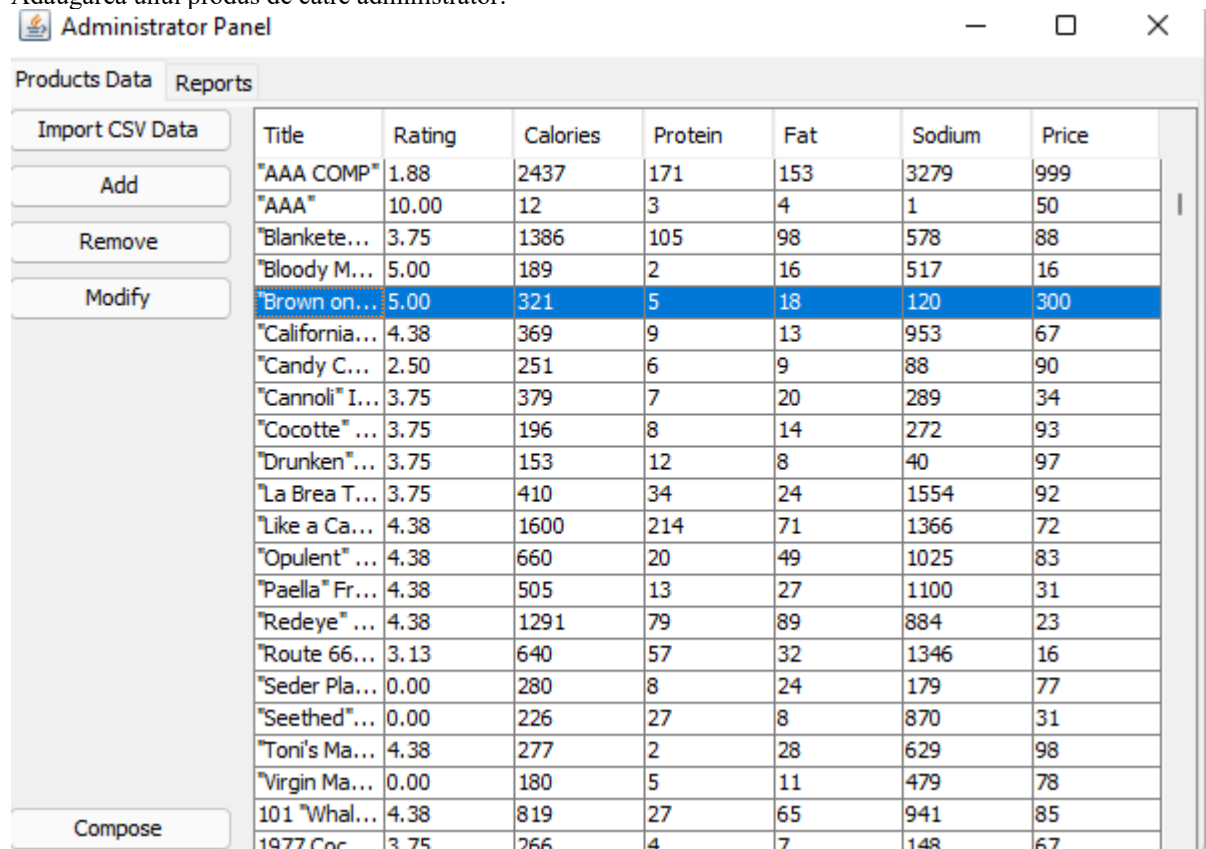
Descrierea metodelor importante din unele clase:

- În clasa **DeliveryService**:
  - Metoda **importBaseProductsDataFromCSV (...)** – este utilizata pentru a importa datele despre produse, dintr-un fisier CSV selectat de catre administrator. Datele din fisier sunt citite cu ajutorul unei librarii open source, 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 **modifyProduct (...)** – 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 **removeProduct (...)** – 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 **addCompositeProduct (...)** – 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 **placeOrder (...)** – 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. La final se realizeaza chitanta formata din produsele comandate, intr-un fisier text ce are ca nume „billOrder#NR.txt”, ce contine pretul total al comenzii si produsele comandate.
- În clasa **MainController**:
  - Metoda **public void setToolTipTex(...)** – este o metoda utilizata pentru a actualiza tool-tip-ul fiecărei coloane din tabela de produse. Aceasta functioneaza prin obtinerea coloanei selectate de catre administrator, iar pe urma, se cauta produsul selectat si daca este gasit si acesta este composite product, se parcurg toate produsele din care este format si se adauga titlurile lor in caseta de tool tip.
  - Metoda **importData (...)** – este o metoda utilizata pentru a importa datele din colectia de produse in modelul tabeli de produse, astfel: se parcurg toate produsele din colectie, iar pentru fiecare dintre acestea, se creeaza o noua linie in tabel, cu datele corespunzatoare (Title, Calories, Protein, Fat, Sodium, Price, Rating).
  - Metoda **setAuthenticatorActionListeners(...)** – este o metoda utilizata pentru a crea toate ActionListener-urile corespunzatoare ferestrei de log in. De aici se controleaza ce fereastră este deschisa, in functie de datele introduse de utilizatorului aplicatiei. Daca datele acestuia corespund unui utilizator cu drepturi de administrator, se deschide o fereastră pentru administrator. Daca corespund unui utilizator cu drepturi de client, se deschide o fereastră pentru client. Daca corespund unui utilizator cu drepturi de employee, se deschide o fereastră pentru angajat. De fiecare data cand este deschisa o noua fereastră, se populeaza tabela de produse cu ajutorul colectiei de obiecte deja serializate, ce contine datele obtinute din accesarea anterioara a aplicatiei.

## 4. Rezultate

Testarea aplicatiei a fost realizata folosind diverse scenarii, precum:

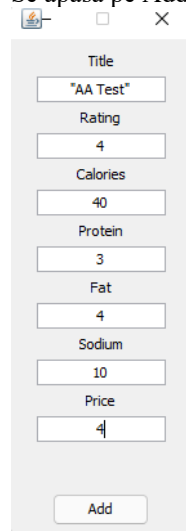
Adaugarea unui produs de catre administrator:



The screenshot shows the 'Administrator Panel' window. It has two tabs: 'Products Data' (selected) and 'Reports'. On the left, there are buttons for 'Import CSV Data', 'Add', 'Remove', 'Modify', and 'Compose'. The main area displays a table with 8 columns: Title, Rating, Calories, Protein, Fat, Sodium, and Price. The table contains 20 rows of product data. The row with Title 'Brown on...' is highlighted in blue.

Title	Rating	Calories	Protein	Fat	Sodium	Price
"AAA COMP"	1.88	2437	171	153	3279	999
"AAA"	10.00	12	3	4	1	50
"Blankete..."	3.75	1386	105	98	578	88
"Bloody M..."	5.00	189	2	16	517	16
"Brown on..."	5.00	321	5	18	120	300
"California..."	4.38	369	9	13	953	67
"Candy C..."	2.50	251	6	9	88	90
"Cannoli" I...	3.75	379	7	20	289	34
"Cocotte" ...	3.75	196	8	14	272	93
"Drunken" ...	3.75	153	12	8	40	97
"La Brea T..."	3.75	410	34	24	1554	92
"Like a Ca..."	4.38	1600	214	71	1366	72
"Opulent" ...	4.38	660	20	49	1025	83
"Paella" Fr...	4.38	505	13	27	1100	31
"Redeye" ...	4.38	1291	79	89	884	23
"Route 66..."	3.13	640	57	32	1346	16
"Seder Pla..."	0.00	280	8	24	179	77
"Seethed" ...	0.00	226	27	8	870	31
"Toni's Ma..."	4.38	277	2	28	629	98
"Virgin Ma..."	0.00	180	5	11	479	78
101 "Whal..."	4.38	819	27	65	941	85
1977 Coc	3.75	266	4	7	148	67

Se apasa pe Add:

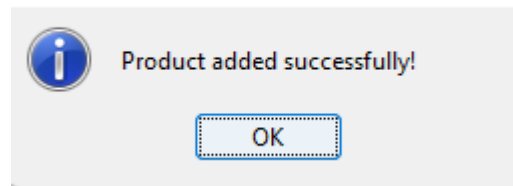


The screenshot shows the 'Add' form in the Administrator Panel. It has a title bar with a close button. The form contains input fields for 'Title', 'Rating', 'Calories', 'Protein', 'Fat', 'Sodium', and 'Price'. The 'Title' field is filled with 'AA Test'. The 'Rating' field is filled with '4'. The 'Calories' field is filled with '40'. The 'Protein' field is filled with '3'. The 'Fat' field is filled with '4'. The 'Sodium' field is filled with '10'. The 'Price' field is filled with '4'. There is an 'Add' button at the bottom.

Title	"AA Test"
Rating	4
Calories	40
Protein	3
Fat	4
Sodium	10
Price	4
Add	

Si se adauga produsul cu numele „AA Test”, cu Rating 4, calorii 40, proteine 3, grasimi 4, sodiu 10 si pret 4.

INFO



Produsul a fost adaugat cu succes.

Administrator Panel							
Products Data Reports							
Import CSV Data	Title	Rating	Calories	Protein	Fat	Sodium	Price
Add	"AA Test"	4.00	40	3	4	10	4
Remove	"AAA COMP"	1.88	2437	171	153	3279	999
Modify	"AAA"	10.00	12	3	4	1	50
	"Blankete..."	3.75	1386	105	98	578	88
	"Bloody M..."	5.00	189	2	16	517	16
	"Brown on..."	5.00	321	5	18	120	300
	"California..."	4.38	369	9	13	953	67
	"Candy C..."	2.50	251	6	9	88	90
	"Cannoli" I...	3.75	379	7	20	289	34
	"Cocotte" ...	3.75	196	8	14	272	93
	"Drunken" ...	3.75	153	12	8	40	97
	"La Brea T..."	3.75	410	34	24	1554	92
	"Like a Ca..."	4.38	1600	214	71	1366	72
	"Opulent" ...	4.38	660	20	49	1025	83
	"Paella" Fr...	4.38	505	13	27	1100	31
	"Redeye" ...	4.38	1291	79	89	884	23
	"Route 66..."	3.13	640	57	32	1346	16
	"Seder Pla..."	0.00	280	8	24	179	77
	"Seethed" ...	0.00	226	27	8	870	31
	"Toni's Ma..."	4.38	277	2	28	629	98
	"Virgin Ma..."	0.00	180	5	11	479	78
Compose	"Virgin Ma..."	0.00	180	5	11	479	78



## 5. Concluzii

În concluzie, această temă mi-a dezvoltat cunoștințele de OOP, învățând să lucrez cu tehnica observer și serializare pe care le voi utiliza în viitoarele proiecte, dovedindu-se un aspect foarte important pentru gestionarea unei baze de date prin intermediul limbajului de dezvoltare Java. În același timp, mi-am dezvoltat abilitățile de structurare și depanare a codului, utilizând platforma Gitlab ce s-a dovedit extrem de folositoare în acest scop.

Ca dezvoltări ulterioare, se poate opta pentru o nouă interfață grafică, mai actuală și mai accesibilă, cu noi funcții, cum ar fi selectarea de produse multiple de către un client și realizarea unei chitanțe comune pentru toate produsele alese de acesta, etc.

## 6. Bibliografie

1. [https://dsrl.eu/courses/pt/materials/A3\\_Support\\_Presentation.pdf](https://dsrl.eu/courses/pt/materials/A3_Support_Presentation.pdf)
2. <https://www.thoughtco.com/using-java-naming-conventions-2034199>
3. <https://docs.oracle.com/javase/tutorial/uiswing/>
4. <https://itextpdf.com/en>
5. <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>
6. <https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>
7. <https://jenkov.com/tutorials/java-reflection/index.html>
8. <https://dzone.com/articles/layers-standard-enterprise>