

DOCUMENTATIE

TEMA 3

NUME STUDENT: Zubascu Maria
GRUPA: 30224

CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3.	Proiectare	5
4.	Implementare	7
5.	Rezultate	9
6.	Concluzii.....	9
7.	Bibliografie	9

1. Obiectivul temei

Obiectivul temei 3 este de a proiecta si implementa o aplicatie de gestionare a comenzilor dintr-o baza de date, cu o interfata grafica in care utilizatorul poate adauga,sterge, vizualiza si edita clienti, produse si comenzi.

Obiective secundare:

- analiza problemei si identificarea cerintelor*
- proiectarea aplicatiei*
- implementarea aplicatiei*
- testarea aplicatiei*

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

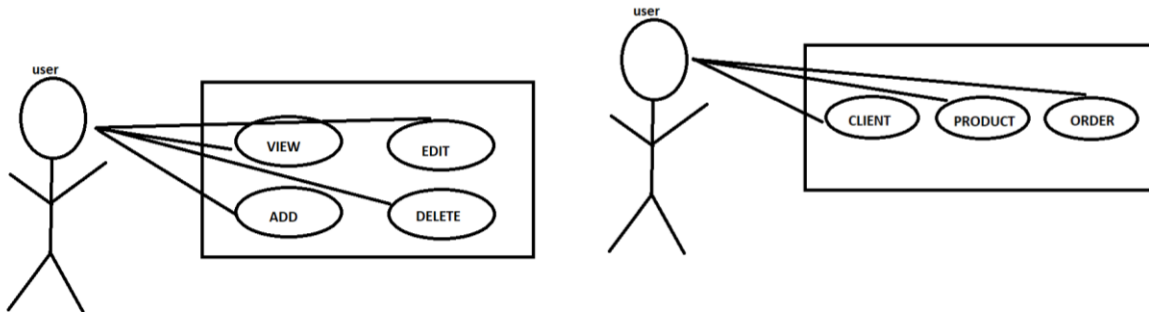
Cerintele functionale in cadrul acestei teme sunt:

- Aplicatie de gestionare a comenzilor ar trebui sa permita utilizatorului sa modifice clientii, produsele si comenzile*
- Aplicatie de gestionare a comenzilor ar trebui sa permita utilizatorului sa adauge clienti, produse si comenzi noi*
- Aplicatie de gestionare a comenzilor ar trebui sa permita utilizatorului sa stearga clientii si produsele*
- Aplicatie de gestionare a comenzilor ar trebui sa permita utilizatorului sa vizualizeze clientii, produsele si comenzile*

Cerintele non functionale sunt urmatoarele:

- Aplicatia ar trebui sa fie intuitive si usor de utilizat de catre utilizator*
- Aplicatia ar trebui sa aiba un design sigestiv si atractiv*

Cazuri de utilizare



Use-case

- **Adaugare**

Actorul principal: utilizatorul

Scenariul de success:

- 1.Utilizatorul alege sa faca adaugare client\podus\comanda*
- 2.Utilizatorul apasa butonul add client/product/order*
- 3.Utilizatorul introduce datele cerute*
- 4.Utilizatorul apasa butonul add*
- 5.Va aparea un mesaj cu “SUCCESS!”*

- **Editare**

Actorul principal: utilizatorul

Scenariu de success:

- 1.Utilizatorul alege sa editeze datele unui client sau produs*
- 2.Utilizatorul apasa butonul edit client/product*
- 3.Utilizatorul introduce datele cerute*
- 4.Utilizatorul apasa butonul edit*
- 5.Va aparea mesajul “SUCCESS!”*

- **Vizualizare**

Actorul principal: utilizatorul

Scenariu de success:

- 1.Utilizatorul alege sa faca vizualizare client\podus\comanda*
- 2.Utilizatorul apasa butonul view client/product/order*

3. Va aparea un tabel cu datele cerute

- **Stergere**
- Actorul principal: utilizatorul
- Scenariul de success:
 1. Utilizatorul alege sa faca stergere client\podus
 2. Utilizatorul apasa butonul delete client/product
 3. Utilizatorul alege clientul/produsul pe care doreste sa-l stearga
 4. Utilizatorul apasa butonul delete
 5. Va aparea un mesaj cu "SUCCESS!"

Secventa alternativa: utilizatorul introduce datele gresit (contact la nume, etc)

-utilizatorul introduce date gresite

-utilizatorul apasa butonul (done)

-va aparea un mesaj de eroare

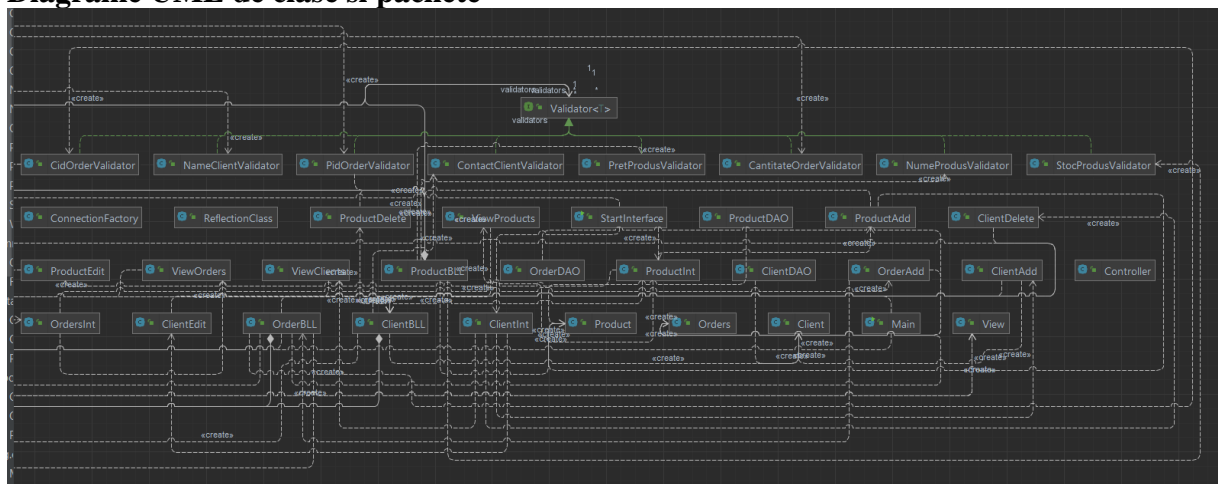
-utilizatorul poate corecta datele

-scenariul revine la pasul 1

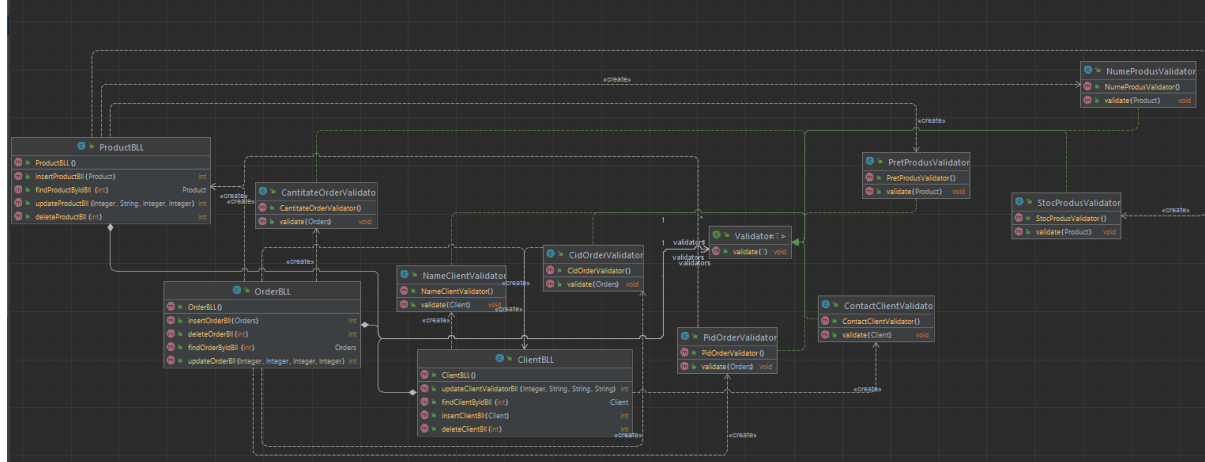
3. Proiectare

Am folosit o baza de date in care am creat 3 tabele (client, product, order), in Java am facut conectarea cu baza de date si am creat clase cu attribute si nume identice cu tabelele. In pachetul DataAccess am implementat metode de stergere, adaugare, editare si vizualizare pentru aceste tabele.

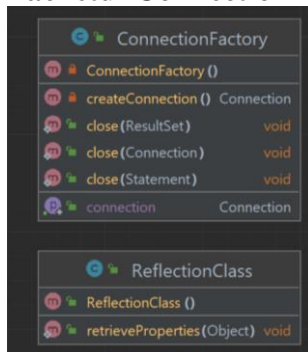
Diagrame UML de clase si pachete



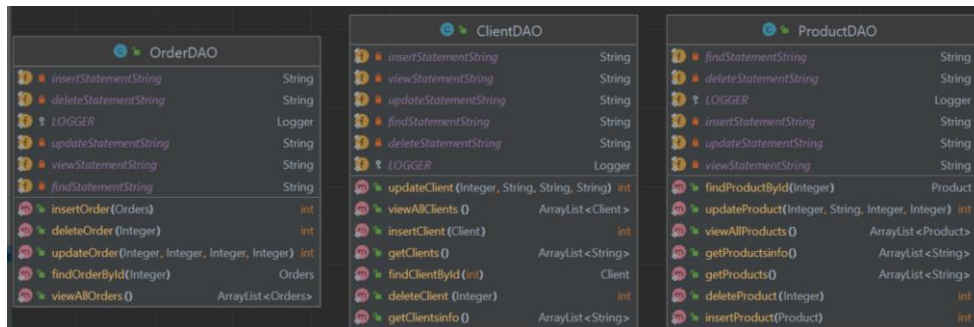
Pachetul BusinessLogic



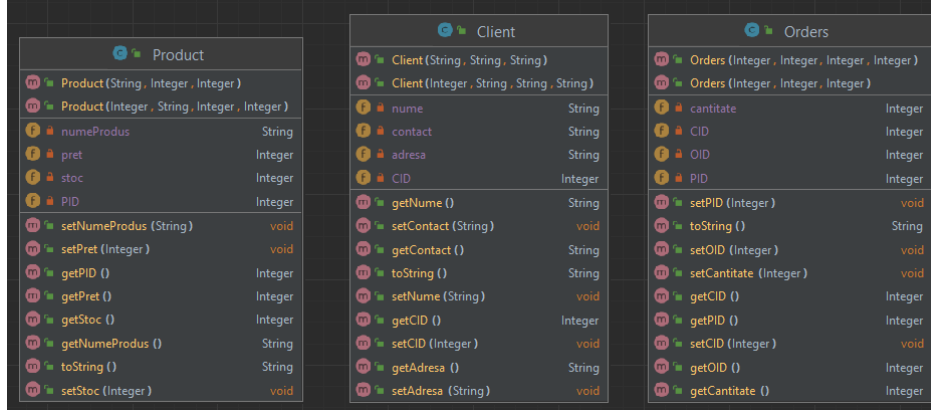
Pachetul Connection



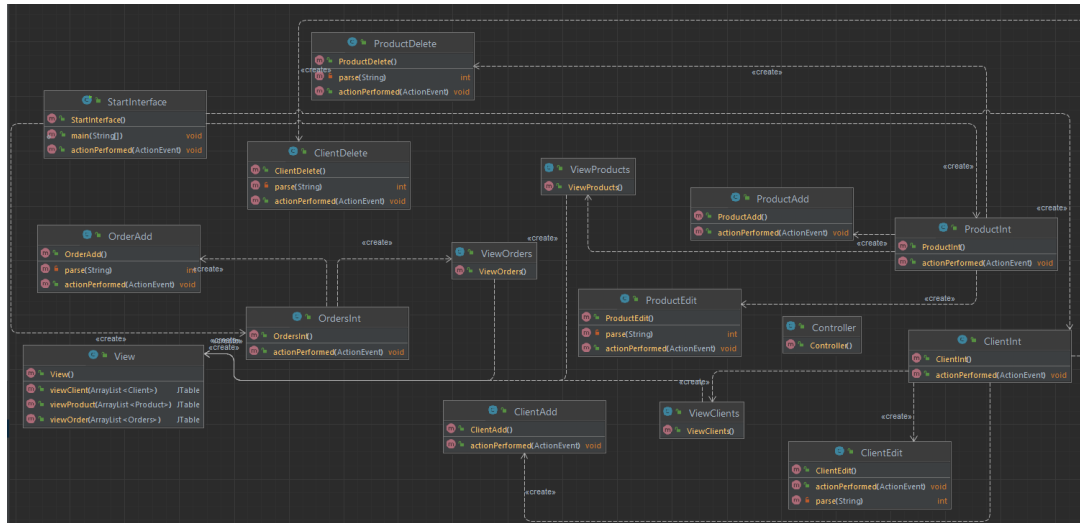
Pachetul DataAccess



Pachetul Model



Pachetul Presentation



4. Implementare

Pachetul BusinessLogic

In acest pachet sunt implementati validatori pentru client, produce si comenzi

In clasele BLL se afla toate metodele din DAO, dar care verifica toti parametrii inainte de a le apela

Pachetul Connection

Contine clasele si metodele care fac legatura intre java si baza de date

Pachetul DataAccess

Contine clase in care se afla metode de stergere, adaugare, editare si vizualizare client/produse/comenzi.

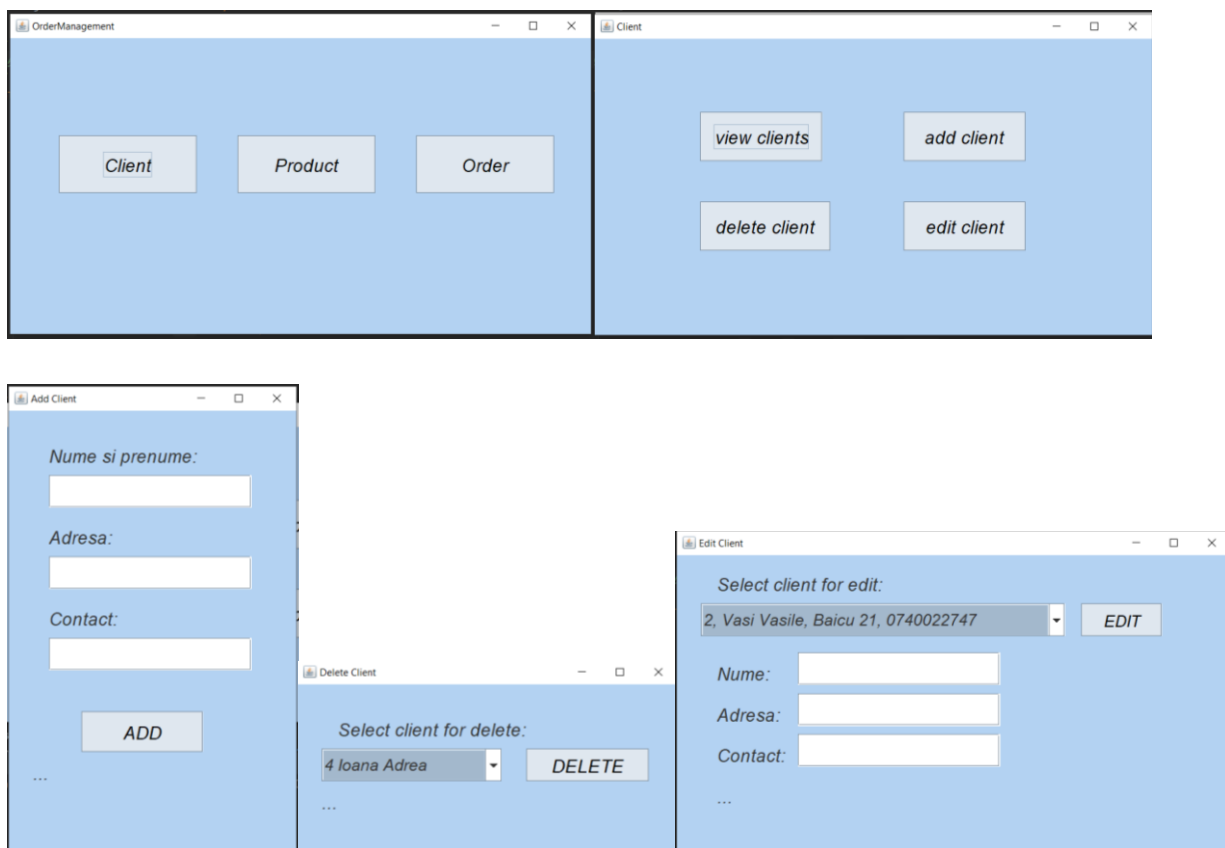
In aceste metode se face conexiunea cu baza de date si se efectueaza operatii pe tabelele din cadrul acesteia.

Pachetul Model

Acest pachet contine clase cu attribute identice din baza de date pentru client/produs/comanda

Pachetul Presentation

In acest pachet am implementat interfata grafica



5. Rezultate

Testarea aplicatiei a fost facuta pe parcursul implementarii.

6. Concluzii

In concluzie, tema3 mi-a dezvoltat tehnica de scriere a codului in java, mi-a imbunatatit modul de organizare al codului si m-a invatat cum sa analizez cerintele si problema prezentata.

Posibile dezvoltari ulterioare ar putea fi: crearea unui bon fiscal la adaugarea unei comenzi

7. Bibliografie

1. *What are Java classes?* - www.tutorialspoint.com
2. <https://regexr.com/>
3. https://www.w3schools.com/java/java_try_catch.asp
4. <https://www.tutorialspoint.com/display-mysql-table-values-using-java>
5. <https://www.geeksforgeeks.org/java-swing-jtable/>
6. <https://stackoverflow.com/questions/1291704/how-do-i-populate-a-jcombobox-with-an-arraylist>
7. https://www.rapidtables.com/web/color/RGB_Color.html