

DOCUMENTATIE

TEMA 3

NUME STUDENT: ZUBAȘCU ILEANA
GRUPA: 30224

CUPRINS

1.Obiectivul temei	3
2.Analiza problemei, modelare, scenarii, cazuri de utilizare	4
3.Proiectare	5
4.Implemenare.....	7
5.Rezultate.....	10
6.Concluzii	10
7.Bibliografie	11

1.Obiectivul temei

Obiectivul temei numarul 3 este de a proiecta si implementa un program care gestioneaza comenzile dintr-o baza de date in care utilizatorul poate adauga, sterge, vizualiza si implementa client, produse si comenzi.

Obiective secundare:

- analiza problemei și identificarea cerințelor (cap.2- Analiza problemei, modelare, scenarii, cazuri de utilizare)
- proiectarea aplicatiei de gestionare a comenzilor(cap.3- Proiectare)
- implementarea aplicatiei de gestionare(cap.4- Implemenare)
- testarea aplicatiei de gestionare(cap.5- Rezultate)

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

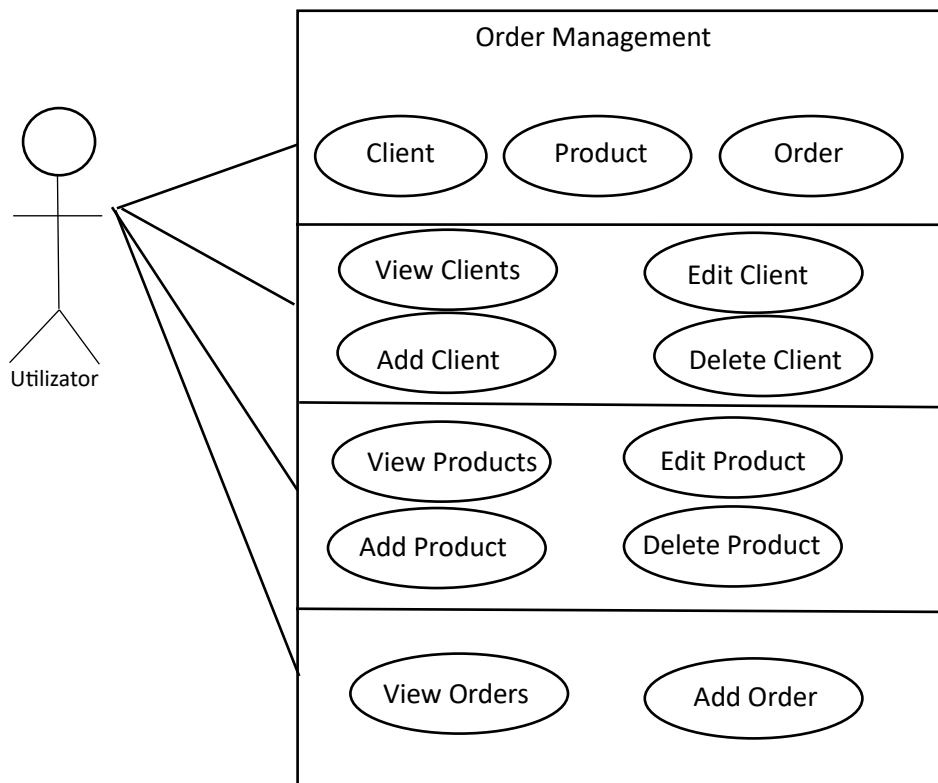
Cerințele funcționale în cadrul acestei teme sunt următoarele:

- Aplicatia de gestionare a comenzilor ar trebuie sa permita utilizatorului sa modifice clientii si produsele
- Aplicatia de gestionare a comenzilor ar trebuie sa permita utilizatorului sa adauge clienti, produse si comenzi.
- Aplicatia de gestionare a comenzilor ar trebuie sa permita utilizatorului sa steraga clientii si produsele.
- Aplicatia de gestionare a comenzilor ar trebuie sa permita utilizatorului sa vizualizeze clientii, produsele si comenzile

Cerințele non funcționale sunt următoarele:

- Aplicatia ar trebui să fie intuitiva și ușor de utilizat de către utilizator.
- Aplicatia ar trebui să aibă un design sugestiv și atractiv.

Cazuri de utilizare:



Use-case

- Adaugare produs

Actorul principal: angajat

Scenariul de success:

1. Angajatul selecteaza optiunea de a aduga un produs nou.
2. Aplicatia afiseaza un formular in care sunt detaliile produsului de adugat.
3. Angajatul introduce numele produsului, pretul si stocul.
4. Angajatul face click pe butonul "ADD".
5. Aplicatia stocheaza datele in baza de date si afiseaza mesajul "SUCCESS!".

Secventa alternativa: angajatul introduce date gresite.

-utilizatorul introduce o valoare negativa pentru stocul produsului.

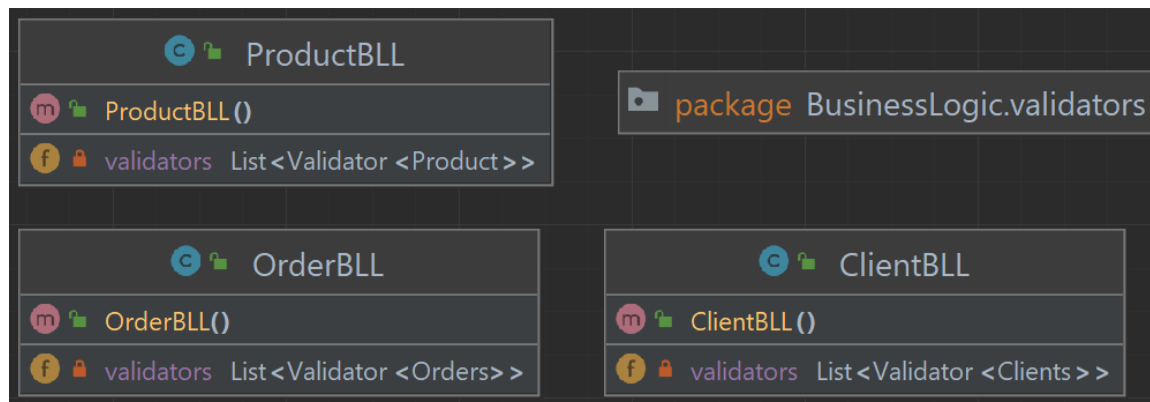
-Aplicatia afiseaza un mesaj de eroare "Wrong Data!".

-scenariul revine la pasul 3.

3.Proiectare

Am folosit o baza de date unde am creat 3 tabele: Clients, Products si Orders.In Java am facut conectarea cu baza de date, dupa care am creat clase identice cu numele tabelor.Claselor din pachetul AccessData implementeaza metode folosite in editare, vizualizare, stergere si adaugarea unor noi obiecte.In pachetul BusinessLogic am creat validatori si clasele BLL care implementeaza clase folosite la interfata.

Diagrame UML de clase și pachete



ReflectionClass
ReflectionClass ()

ConnectionFactory
ConnectionFactory()
PASS String
LOGGER Logger
DBURL String
DRIVER String
singletonInstance ConnectionFactory
USER String

OrderDAO
OrderDAO ()
deleteStatementString String
insertStatementString String
findStatementString String
viewStatementString String
LOGGER Logger
updateStatementString String

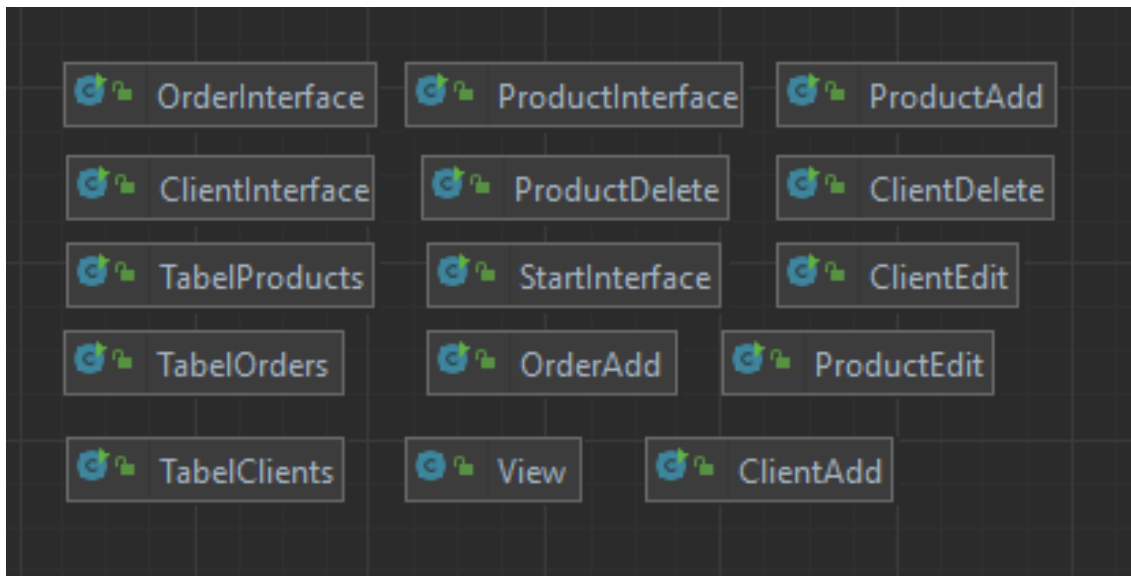
ProductDAO
ProductDAO ()
LOGGER Logger
findStatementString String
updateStatementString String
insertStatementString String
viewStatementString String
deleteStatementString String

ClientDAO
ClientDAO ()
LOGGER Logger
viewStatementString String
insertStatementString String
findStatementString String
updateStatementString String
deleteStatementString String

Orders
Orders(int, int, int)
Orders(int, int, int, int)
cantitate int
CID int
PID int
OID int

Clients
Clients (String, String, String)
Clients (int, String, String, String)
CID int
adresa String
contact String
nume String

Product
Product(String, int, int)
Product(int, String, int, int)
pret int
numeProdus String
PID int
stoc int



4.Implementare

1.Pachetul Model:

In acest pachet sunt implementate clasele Clients, Orders si Product care au attribute identice cu tuplele din baza de date a fiecarui tabel. Clasele implementeaza constructori, settere si gettere.

2.Pachetul Connection:

In acest pachet sunt implementate clasele ConnectionFactory care face legatura dintre baza de date si Java. Clasa ReflectionClass este folosita pentru a parcurge toate attributele clasei unui obiect.

3.Pachetul DataAccess:

In acest pachet sunt implementate clasele ClientsDAO, OrderDAO si ProductDAO care implementeaza metode folosite in editarea, vizualizarea, stergerea si adaugarea unor produse, clienti sau comenzi.

1.Pachetul BusinessLogic

In acest pachet sunt implementati validatori pentru client, produse si comenzi in pachetul validators.Clasele ClientBLL, OrderBLL, ProductBLL implementeaza

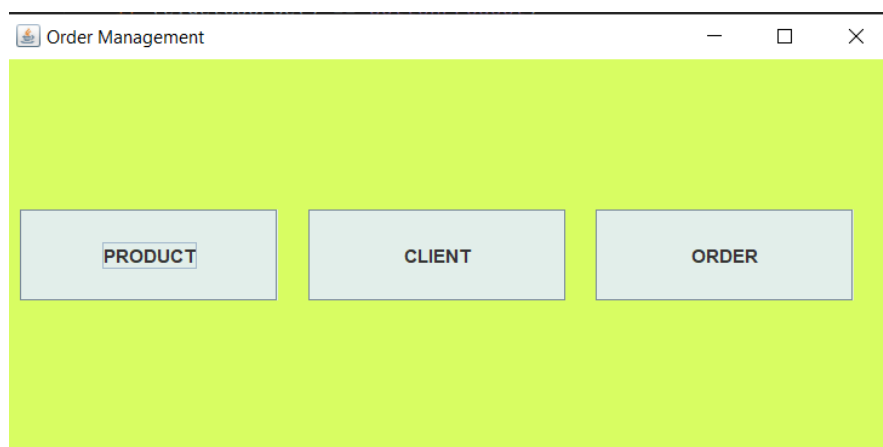
metode care valideaza atunci cand se face o modificare la baza de date, sunt folosite la interfata.

1.Pachetul Presentation:

In acest pachet sunt implementate clasele care definesc interfata grafica a aplicatiei de gestionat comenzi.De asemenea, clasa View contine metode care genereaza tabelele din baza de date.

Interfata

Interfata principala



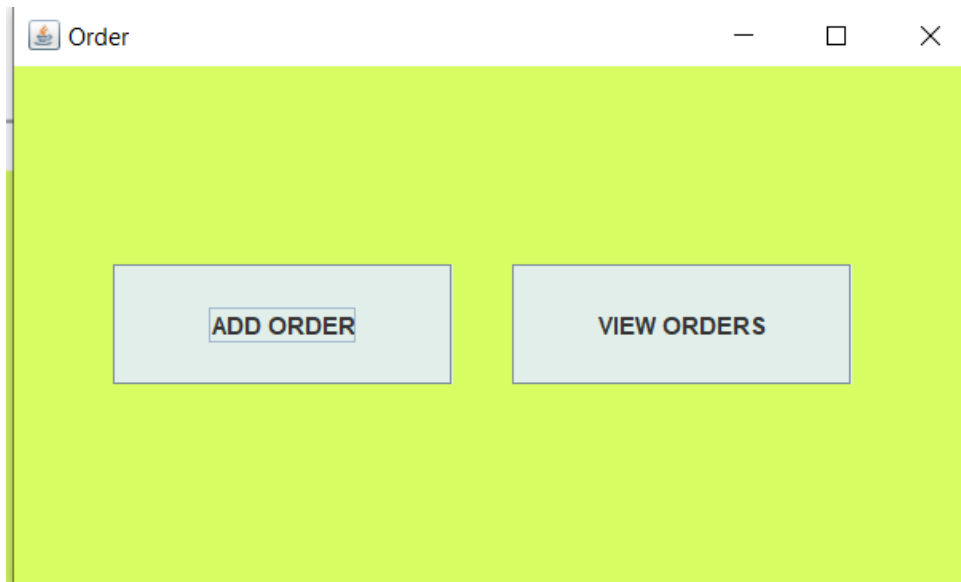
Interfata Products



Interfata Client



Interfata Order



Baza de date SQL

```
7 • CREATE TABLE if not exists Clients(  
8   CID int not null unique auto_increment primary key,  
9   nume varchar(30) not null,  
10  adresa char(30) not null,  
11  contact varchar(10) NOT NULL);  
12  
13 • create table if not exists Product(  
14  PID int not null unique auto_increment primary key,  
15  numeProdus varchar(30) not null,  
16  pret int not null,  
17  stoc int not null);  
18  
19 • create table if not exists Orders(  
20  OID int not null unique auto_increment primary key,  
21  CID int(11) not null,  
22  PID int(11) not null,  
23  cantitate int not null,  
24  FOREIGN KEY (CID) references Clients(CID),  
25  FOREIGN KEY (PID) references Product(PID));  
26
```

5.Rezultate

Testarea aplicatiei a fost facuta pe parcursul implemetarii.

6.Concluzii

În concluzie, tema 3 mi-a dezvoltat tehnica de scriere a codului în Java, mi-a îmbunătățit modul în care îmi organizez codul și modul în care analizez cerințele și problema prezentată astfel încât să implementez un cod cât mai eficient și corect.

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

7.Bibliografie

https://dsrl.eu/courses/pt/materials/PT2023_A3_S1.pdf

<https://www.javatpoint.com/example-to-connect-to-the-mysql-database>