DOCUMENTATIE

TEMA 3

NUME STUDENT: RUS ANA-MARIA CARINA
GRUPA: 30226

CUPRINS

1. OBIECTIVUL TEMEI

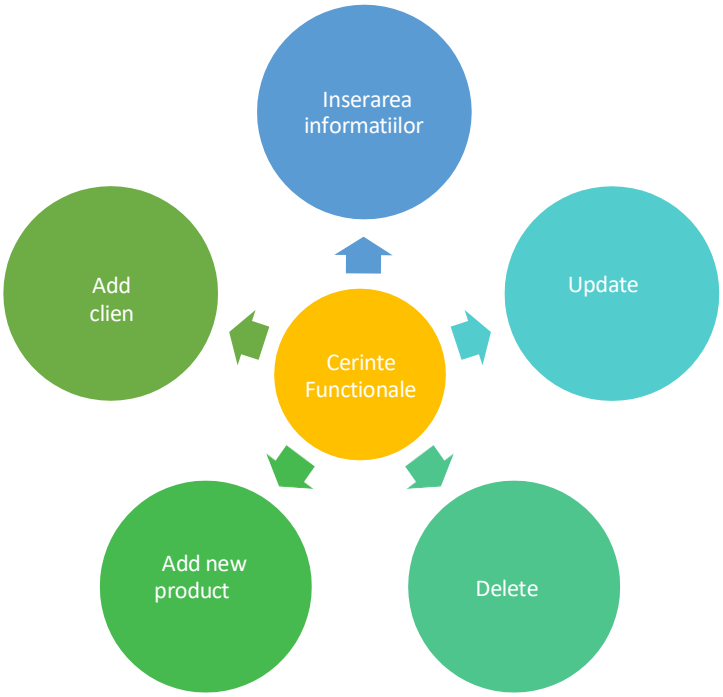
Obiectivul temei este de a crea un proiect Orders Management pentru procesarea comenzilor clientilor.

OBIECTE SECUNDARE:

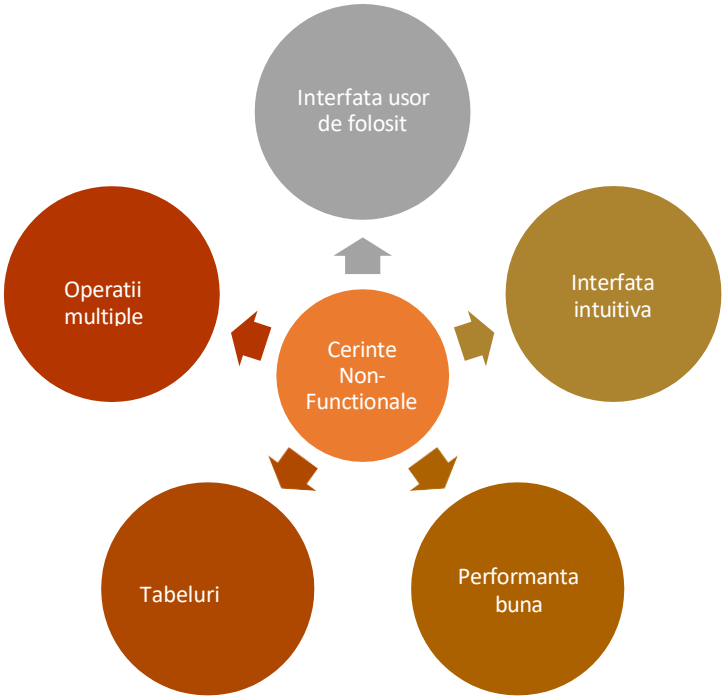
1. Intelegerea problemei
2. Crearea unei interfete potrivite pentru introducerea informatiilor
3. Implementarea aplicatiei
4. Testarea aplicatiei

2. ANALIZA PROBLEMEI, MODELARE, SCENARII, CAZURILE DE UTILIZARE

Cerinte functionale



Cerinte non-functionale



User-Case

User Case: adauga produs
Actor primar: User-ul
Scenariu pentru succes:

1. User-ul selecteaza tabelul Products
2. User-ul selecteaza butonul pentru Add
3. Aplicatia afiseaza care detalii trebuie inserate
4. User-ul insereaza detaliile si apasa pe Add
5. Aplicatia salveaza produsul in baza de date si afiseaza tabelul de produse impreuna cu noua valoare

Secventa alternatica: Valori invalide pentru datele produsului

-User-ul introduce valori invalide

-Aplicatia afiseaza un mesaj care informeaza useu-ul ca valorile sunt invalide

-Scenariul se reintoarce la punctul 1.

3. PROIECTARE

Diagrama Main

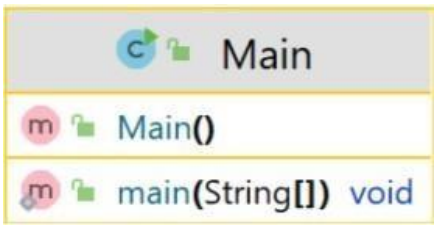


Diagrama ControllerView

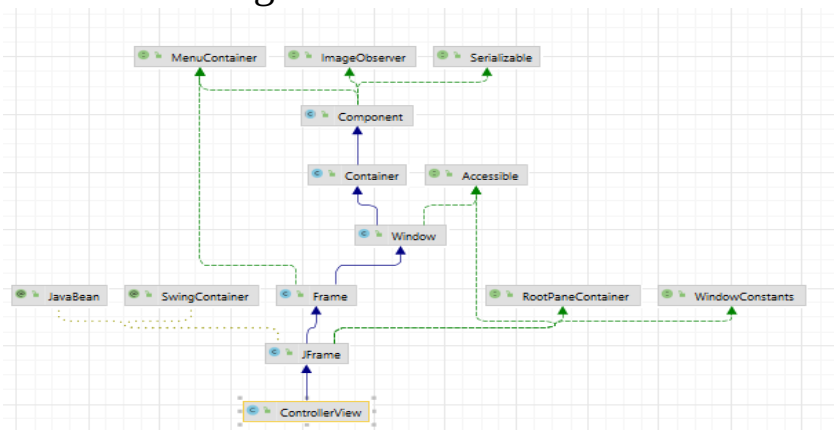


Diagrama Clients, Orders, Products

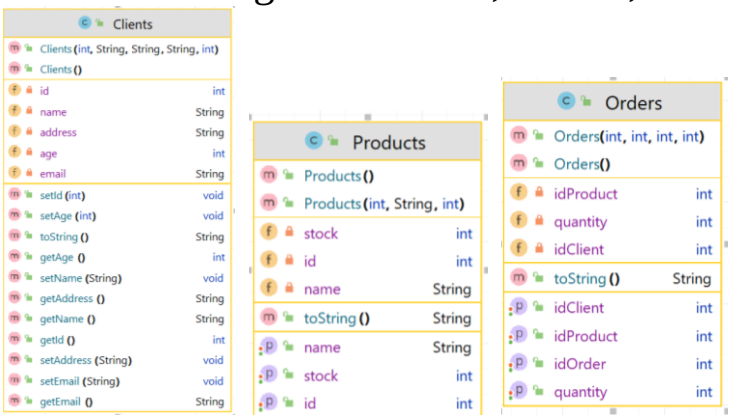


Diagrama AbstractDAO

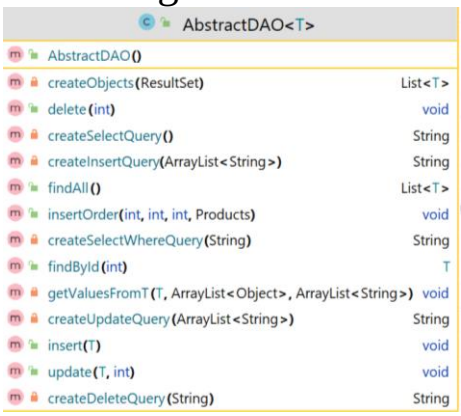
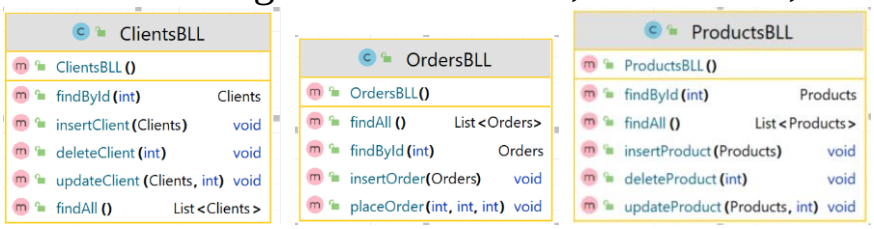


Diagrama ClientsBLL, OrdersBLL, ProductsBLL



Am folosit 6 pachete:

1. Org-example
2. Presentation
3. Model
4. Dao
5. Connection
6. Bll

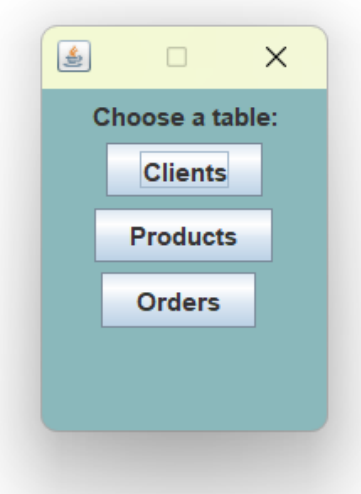
7. IMPLEMENTARE

1. Clasa Main

In clasa Main am initializat interfata ControllerView.

2. Clasa ControllerView

In clasa CalcView am creat interfata folosind butoane, text fields, panel-lui si frame-uri pentru fiecare tabel si operatie.
Am folosit ActionListener pentru a implementa operatiile facute de fiecare buton.



3. Clasele Clients, Orders, Products

Aceste clase definesc obiectele de tipul clienti, produse si comenzi fiecare cu id-ul respectiv si detaliile cerute. Aici sunt definite gettere si settere.

4. Clasele AbstractDAO, ClientsDAO, OrdersDAO, ProductsDAO

   Clasele ClientsDAO, OrdersDAO, ProductsDAO extind clasa AbstractDAO care contine query-urile pentru select, insert, delete si operatiile pentru find all si find by id. La clasele DAO se foloseste reflexia.

5. Clasa ConnectionFactory

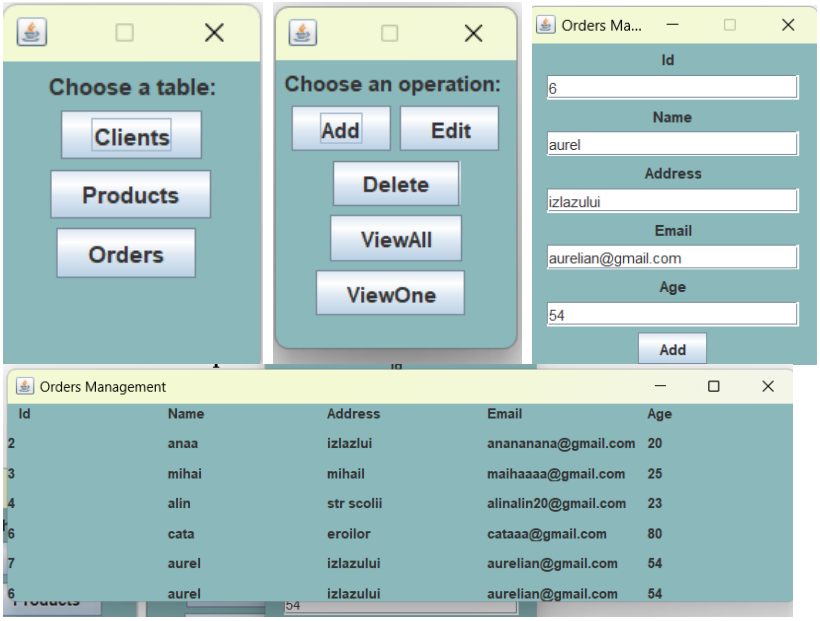
   Clasa ConnectionFactory face legatura cu baza de date.

6. Clasele ClientsBLL, OrdersBLL, ProductsBLL

   Clasele acestea fac legatura dintre interfata si clasele DAO.

7. REZULTATE

User-ul poate alege tabelul pe crae v rea sa faca operatii si operatiile dorite dupa care este afisat rezultatul in urma operatiilor.



8. CONCLUZII

Am invatat sa creez interfata mai sofisticata, sa folosesc bazele de date si sa organizez codul si clasele in mai multe pachete si sa folosesc reflection.

9. BIBLIOGRAFIE

1. https://gitlab.com/utcn_dsrl/pt-reflection-example/-/blob/master/src/main/java/bll/validators/Validator.java
2. https://www.w3schools.com/sql/sql_insert.asp