MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Facultatea de Automatica si Calculatoare

-Calculatoare si tehnologia informatiei-

ORDERS MANAGEMENT

Documentatie

TEMA3

Nistor Dalia

Grupa 30224

Cuprins

1.OBIECTIVUL TEMEI

Luați în considerare o aplicație de gestionare a comenzilor pentru procesarea comenzilor clienților pentru un depozit.
Bazele de date relaționale ar trebui să fie utilizate pentru a stoca produsele, clienții și comenzile. Site-ul
aplicația ar trebui să fie proiectată în conformitate cu modelul de arhitectură stratificată și ar trebui să utilizeze
(minim) următoarele clase:
- Clase de model - reprezintă modelele de date ale aplicației.
- Clase de logică de afaceri - conțin logica aplicației
- Clase de prezentare - clase legate de interfața grafică
- Clase de acces la date - clase care conțin accesul la baza de dateAstfel. tema a avut urmatorii pasi pentru indeplinirea obiectivului principal:

- Utilizarea un design de programare orientat pe obiecte, clase cu maximum 300 linii, metode cu maximum 30 de linii, convenții de denumire Java.
- Utilizarea javadoc pentru documentarea claselor și generați fișierele corespunzătoare. JavaDoc corespunzătoare.
- Utilizarea baze de date relaționale pentru stocarea datelor pentru aplicație, minimum trei tabele: Client, Produs și Comandă.
- Crearea unei interfete grafice cu utilizatorul care să includă:
  -fereastră pentru operațiunile cu clienții: adăugarea unui nou client, editarea unui client, ștergerea unui client,vizualizarea tuturor clienților într-un tabel (JTable).
  - fereastră pentru operațiunile cu produse: adăugarea unui nou produs, editarea produsului, ștergerea produsului, vizualizarea tuturor produselor într-un tabel (JTable)
  - O fereastră pentru crearea de comenzi de produse - utilizatorul va putea selecta un produs existent, să selecteze un client existent și să introducă o cantitate dorită pentru un produs existent.pentru a crea o comandă validă. În cazul în care nu există suficiente produse, se va afișa un mesaj de lipsă de stoc. După ce comanda este finalizată, se va afișa stocul de produse este decrementat.
- Utilizarea tehnicilor de reflecție pentru a crea o metodă care primește o listă de obiecte și generează antetul tabelului prin extragerea prin reflexie a elementelor proprietățile obiectelor și apoi populează tabelul cu valorile obiectelor elemente din listă

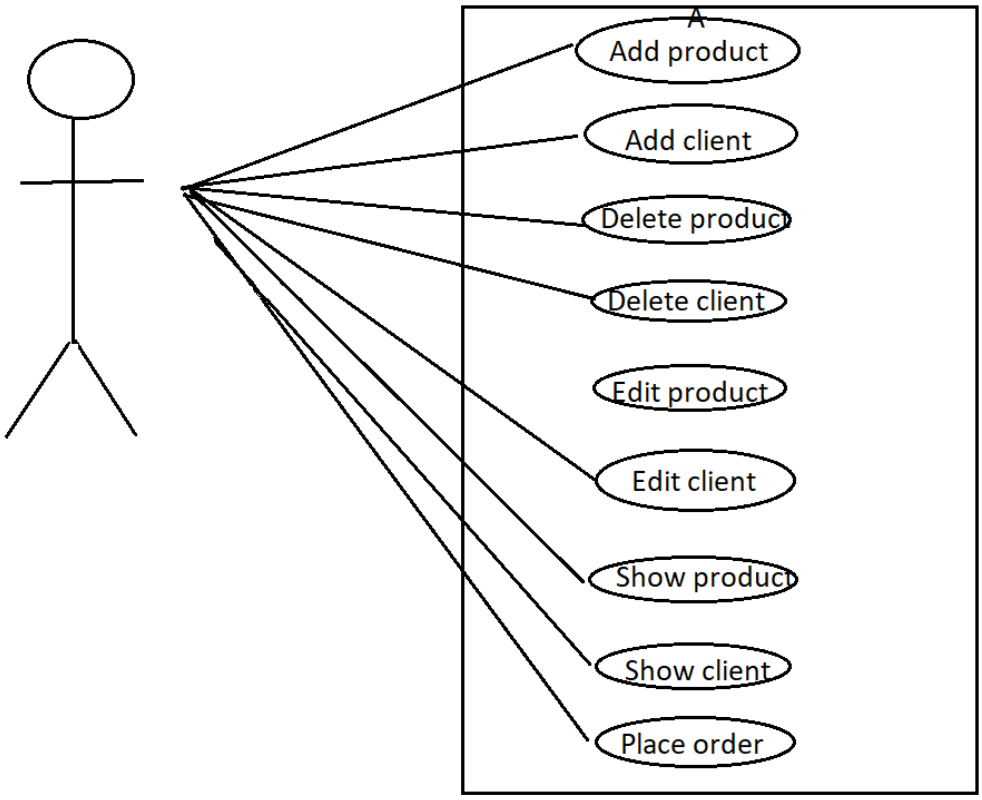
2.ANALIZA PROBLEMEI, MODELARE, SCENARII, CAZURI DE UTILIZARE

Un sistem de gestionare a comenzilor (OMS) sprijină toate etapele procesului de vânzare ale unei companii.

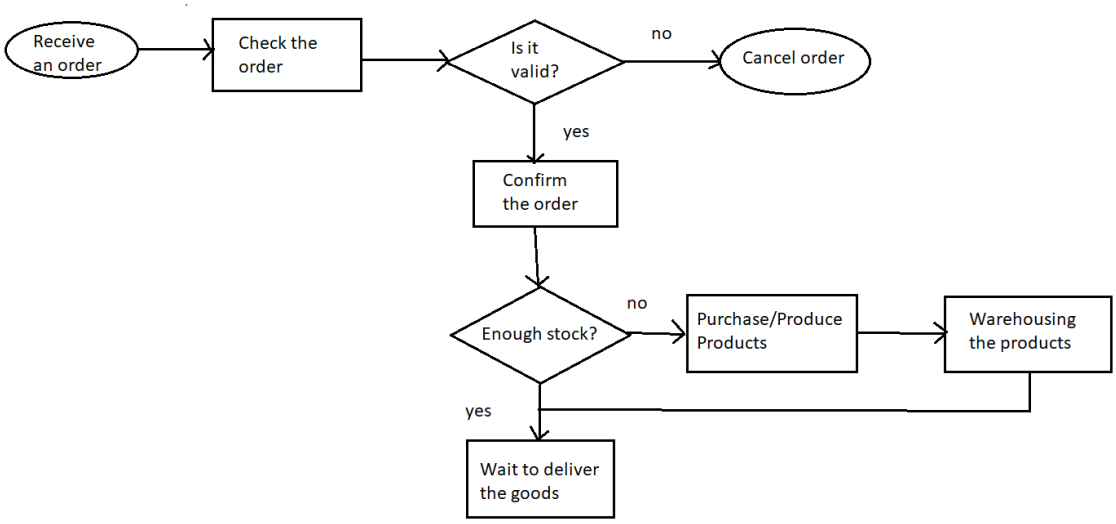
Platformele OMS oferă un sistem unic și centralizat pentru gestionarea comenzilor din mai multe canale de vânzare, inclusiv locații de tip cărămidă și mortar, site-uri web, centre de apel, comenzi mobile, chioșcuri și multe altele. Aceasta simplifică procesul de cumpărare pentru clienți și facilitează gestionarea comenzilor, a inventarului, a îndeplinirii și a retururilor pentru întreprinderi.

Se retin clientii ,produsele si comenziile intr-o baza de date numita warehouse, unde am creat 3 tabele pentru gestionarea acestora. Fiecare tabela are campurile specifice: clientul are un id, email,nume,adresa, produsul are un id, nume, cantitate iar comenziile au id-ul comenzii, id-ul clientului si id-ul produsului.
Pe baza acestora am creat metode pentru vizualizare, inserare, stergere si editare cu ajutorul sintaxei sql.
In plus, pentru plasarea unei comenzi am folosit de asemenea apeluri sql. Utilizatorul trebuie sa introduca id-ul comenzii, care nu a mai fost introdus anterior, id-ul clientului pentru care doreste plasarea unei comenzi si id-ul produsului care doreste a fi comandat.

USE CASE DIAGRAM



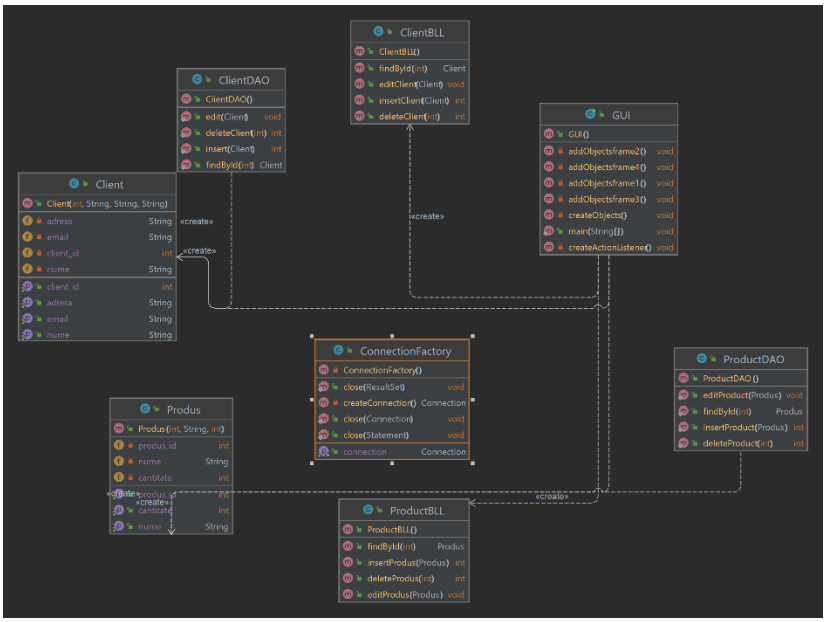
FLOWCHART



3.PROIECTARE

Diagrama UML

Diagrama de clase UML este o notație grafică utilizată pentru a construi și vizualiza sisteme orientate pe obiecte.



Stocarea datelor

Cand vine vorba despre mulțimi de obiecte(elemente) în java, în majoritatea cazurilor apare și nevoia de a le stoca într-un container pentru manipularea lor ulterioară (sortare, afișare, ștergere etc.). In acest proiect datele sunt stocate intr-o baza de date facuta in MySQL.

4.IMPLEMENTARE

Clasa Client

In aceasta clasa am retinut parametrii din tabela ai clientilor: id, nume, email, adresa, unde se afla constructorul client impreuna cu setterele si getterele specifice.

Clasa Produs

In aceasta clasa am retinut parametrii din tabela ai produselor: id, nume, cantitae, unse se afla contructorul produs impreuna cu setterele si getterele specifice.

Clasa ClientDAO

In aceasta clasa am efectuat cate o metoda pentru operatiile de inserare, stergere, editare si de asemenea pentru operatia de gasire a unui client pe baza de id.

- Pentru metoda de inserare am folosit sintaxa din sql impreuna cu conexiunea bazei de date folosindu-ma de clasa ConnectionFactory.
- Pentru metoda de stergere am folosit sintaxa sql specifica impreuna cu conexiunea bazei de date, folosindu-ma de clasa ConnectionFactory

- Pentru metoda de editare am folosit sintaxa sql specifica impreuna cu conexiunea bazei de date, folosindu-ma de clasa ConnectionFactory
- Pentru metoda de de gasire a unui client pe baza id-ului m-am folosit de sintaxa sql specifica si de asemenea de clasa ConnectionFactory pentru a lua legatura cu baza de date

Clasa ProductDAO

In aceasta clasa am efectuat cate o metoda pentru operatiile de inserare, stergere, editare si de asemenea pentru operatia de gasire a unui produs pe baza de id.

- Pentru metoda de inserare am folosit sintaxa din sql impreuna cu conexiunea bazei de date folosindu-ma de clasa ConnectionFactory.
- Pentru metoda de stergere am folosit sintaxa sql specifica impreuna cu conexiunea bazei de date, folosindu-ma de clasa ConnectionFactory
- Pentru metoda de editare am folosit sintaxa sql specifica impreuna cu conexiunea bazei de date, folosindu-ma de clasa ConnectionFactory
- Pentru metoda de de gasire a unui produs pe baza id-ului m-am folosit de sintaxa sql specifica si de asemenea de clasa ConnectionFactory pentru a lua legatura cu baza de date

Clasa ProductBLL

In aceasta clasa am apelat metodele din clasa ProductDAO

Clasa ClientBLL

In aceasta clasa am apelat metodele din clasa ClientDAO

Clasa ConnectionFactory'

In aceasta clasa am implementat 4 metode: pentru crearea conexiunii intre MySQL si java, pentru inchiderea acesteia, inchiderea statementului si inchiderea resultset-ului.

De asemeena, aici se afla si constructorul pentru preluarea conexiunii.

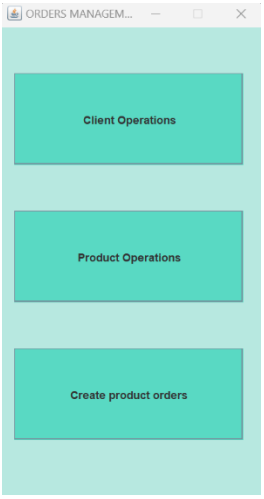
Clasa GUI

In aceasta clasa am implementat interfata cu care interactioneaza utilizatorul. Am folosit pentru aceasta Java Swing. Swing în Java este un set de instrumente de interfață grafică cu utilizatorul (GUI) care include componentele GUI.

Biblioteca Java Swing este construită pe baza Java Abstract Widget Toolkit (AWT), un set de instrumente GUI mai vechi, dependent de platformă. Se pot utiliza componente simple de programare Java GUI, cum ar fi butonul, caseta de text etc., din bibliotecă și nu trebuie să se creeze componente de la zero.

Aplicatia mea contine 16 butoane:

1. 3 butoane pentru alegerea ferestrei pe care utilizatorul isi doreste sa o deschida



2. Iar celelalte butoane pentru alegerea operatiilor specifice

De asemena contine 12 TextField-uri:

TextField= spatii dreptunghiulare in care se pot introduce date de la tastatura. Pot fi folosite si pentru a afisa rezultate.

1. 7 pentru introducerea produselor sau a clientilor
2. unul pentru afisarea rezultatului in frame-ul pentru crearea comenzilor
3. 4 pentru introducerea id-ului clientului, id-ului produsului, id-ul comenzii la momentul plasarii unei comenzi

In plus, mai are si 18 label-uri:

Label = este o etichetă ce poate fi folosita drept titlu sau pentru a ajuta utilizatorul

Pentru fiecare button, s-a făcut o metodă nouă care implică ActionEvent. Astfel, de fiecare dată când are loc o acțiune la un buton, de exemplu a fost apăsat, ActionEvent-ul denumit e, transmite informația la ActionListener care așteaptă astfel informații. Apoi, au loc evenimentele ce se afla în metodă respectiva

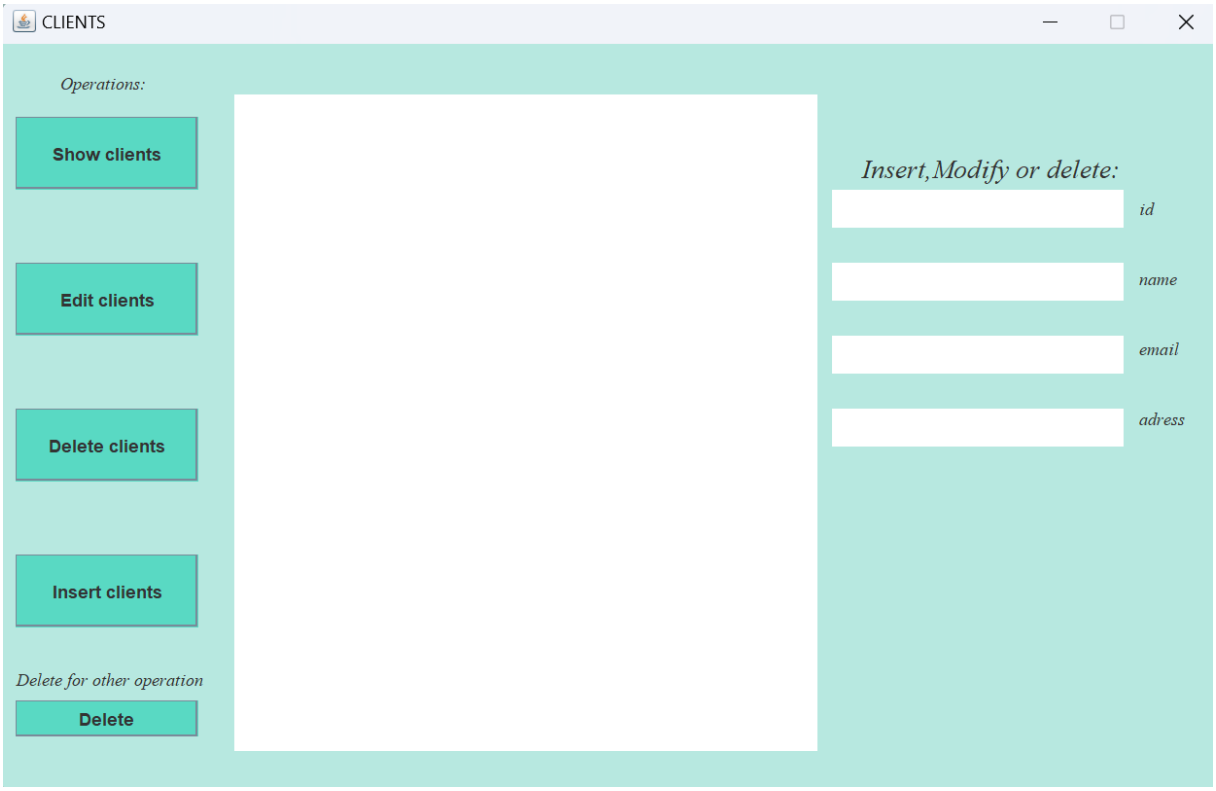
Interfata grafica propriu zisa este formata in constructor unde se initializeaza un frame cu proprietatile dorite, urmat de celelalte 3 frame-uri stimulate la apasarea butoanelor .

Pentru ca la inchiderea unuia dintre frame-uri aplicatia sa nu se opreasca am folosit functia dispose pe celelalte frame-uri decat cel principal.

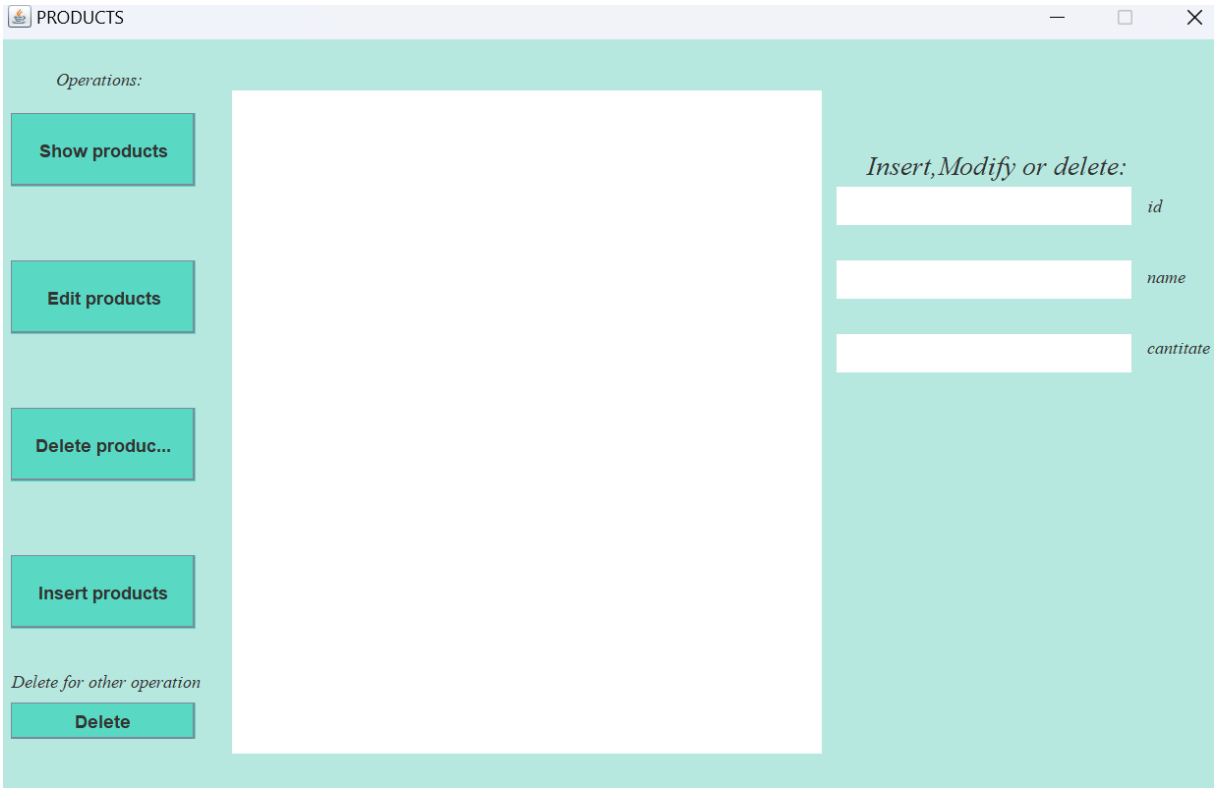
In main-ul programului este instantiata clasa GUI, astfel de fiecare data cand se initialieaza clasa main se va instantia o noua interfata grafica.

Interfata arata in felul urmator:

1. Frame-ul pentru operatiile pe clienti



2. Frame-ul pentru operatiile pe produse:



3. Frame-ul pentru crearea unei comenzi:

5.CONCLUZII

In concluzie, consider ca aceasta tema mi-a imbunatatit remarcabil abilitatile de a scrie cod Java, mi-a aprofundat cunostiintele in ce semnifica implementarea paradigmelor OOP, mi-a imbunatatit modul in care incep implementarea unei aplicatii. De asemenea mi-a imbunatatit cunostintele legate de java in conexiune cu o baza de date, in acest caz formata in MySQL.

Posibile dezvoltari ulterioare:

La crearea comenzilor sa nu mai fie utilizatorul nevoit sa scrie id-ul comenzii deoarece acesta putea fi generat automat in functie de cele deja existente, de asemenea imbunatatirea interfetei pentru a fi user friendly.

6.BIBLIOGRAFIE

- https://mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/
- https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html
- https://dzone.com/articles/layers-standard-enterprise
- https://dsrl.eu/courses/pt/materials/PT2023.pdf
- https://www.baeldung.com/javadoc