DOCUMENTATIE

TEMA 3

NUME STUDENT: Pescaru Silviu-Mihaita

GRUPA: 30225

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc95297886)

[3. Proiectare 6](#_Toc95297887)

[4. Implementare 8](#_Toc95297888)

[5. Rezultate 15](#_Toc95297889)

[6. Concluzii 15](#_Toc95297890)

[7. Bibliografie 16](#_Toc95297891)

# Obiectivul temei

Obiectivul a fost acela de a implementa o aplicație pentru gestionarea comenzilor la un depozit. Datele au fost stocate cu ajutorul bazei de date PostgreSQL.

Pentru implementarea proiectului am folosit o arhitectură de tip Layered.

Utilizatorul va avea la îndemână mai multe ferestre, după cum urmează:

Fereastra de meniu pentru a selecta pe ce tabel vom efectua operații

Fereastra Client, în care se văd toți clienții, se pot adăuga și șterge clienți

Fereastra Product, în care se văd toate produsele, se pot adăuga și șterge produse

Fereastra Order, în care se văd toate comenzile, se pot plasa și șterge comenzi

Pentru documentație s-au generat fișiere JavaDoc, de asemenea a fost făcut un fișier dump SQL, care conține codul de creare și populare a bazei de date.

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Analiza: Fiecare set de operații are o fereastră proprie (Client, Product, Orders)

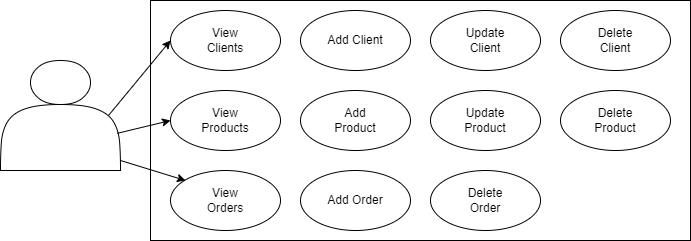
Operații implementate: adăugare, modificare, ștergere

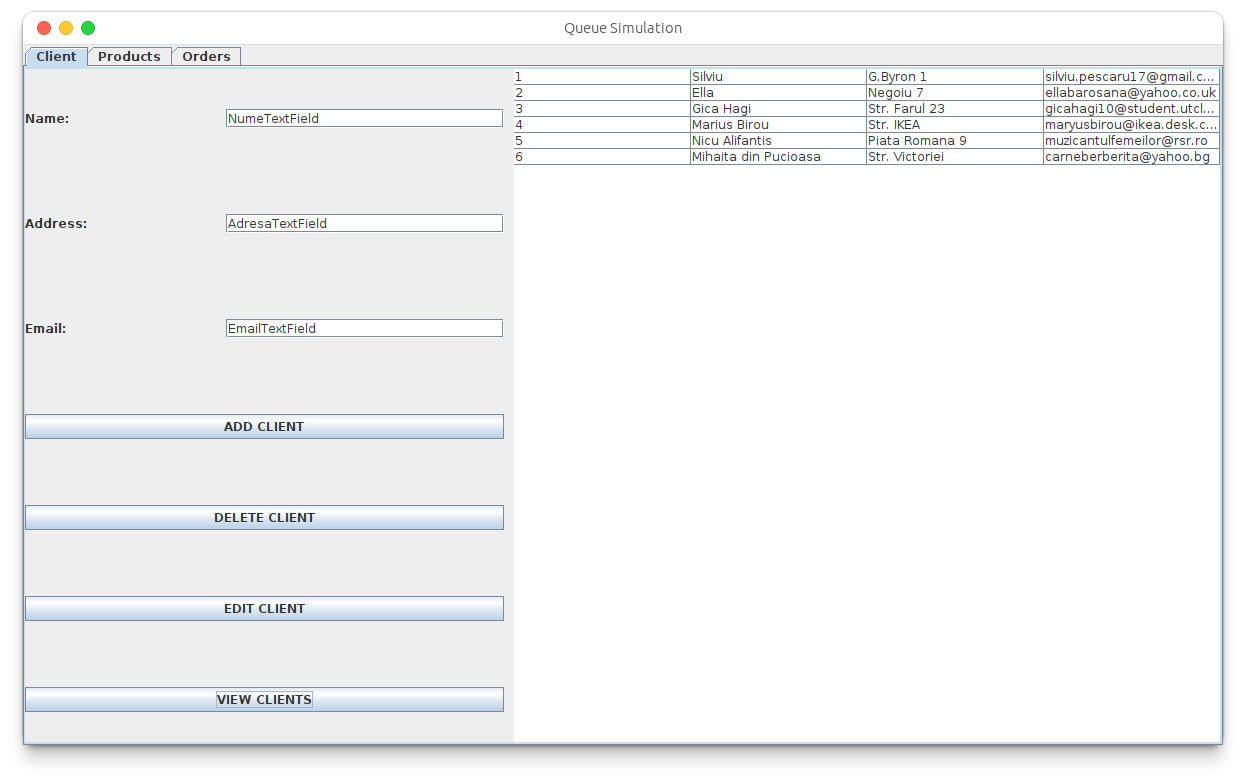
Stocare date: bază e date PostgreSQL, la care programul este conectat prin username și parolă

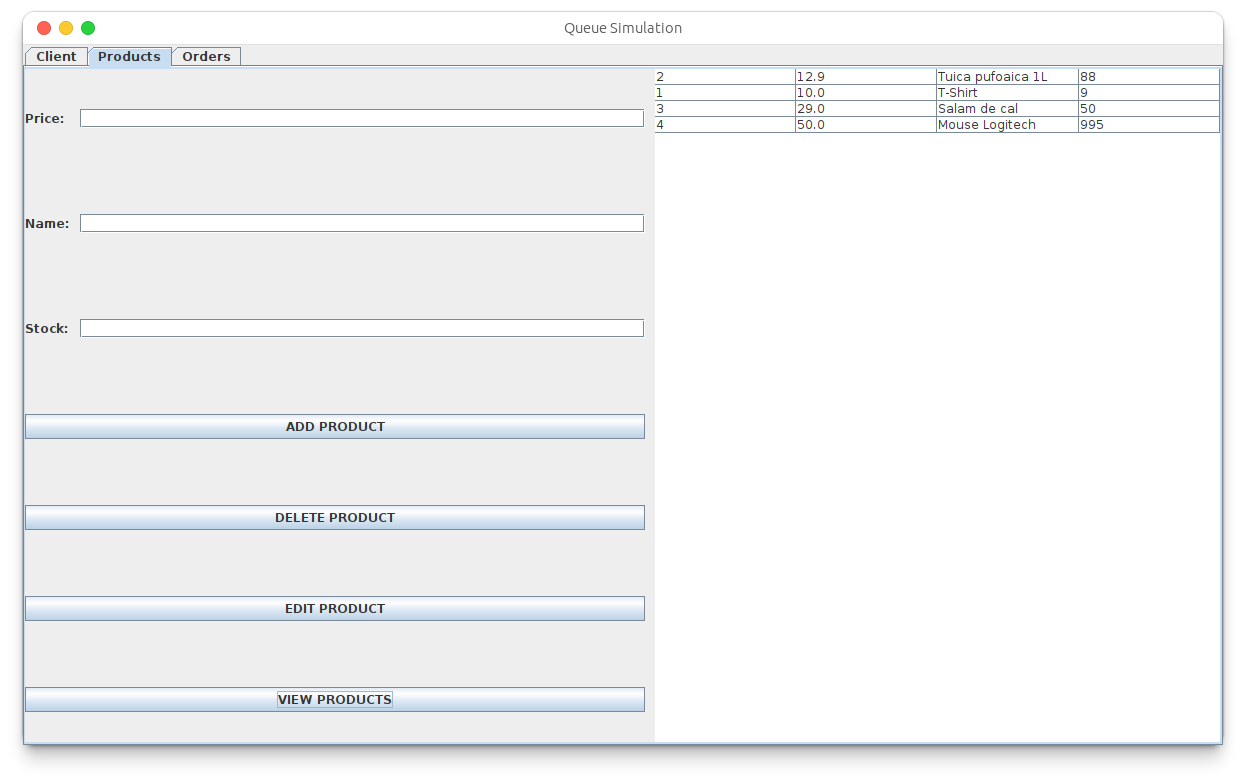
Modelare: Clasele Client, Product și Orders reprezintă câte un obiect cu același nume

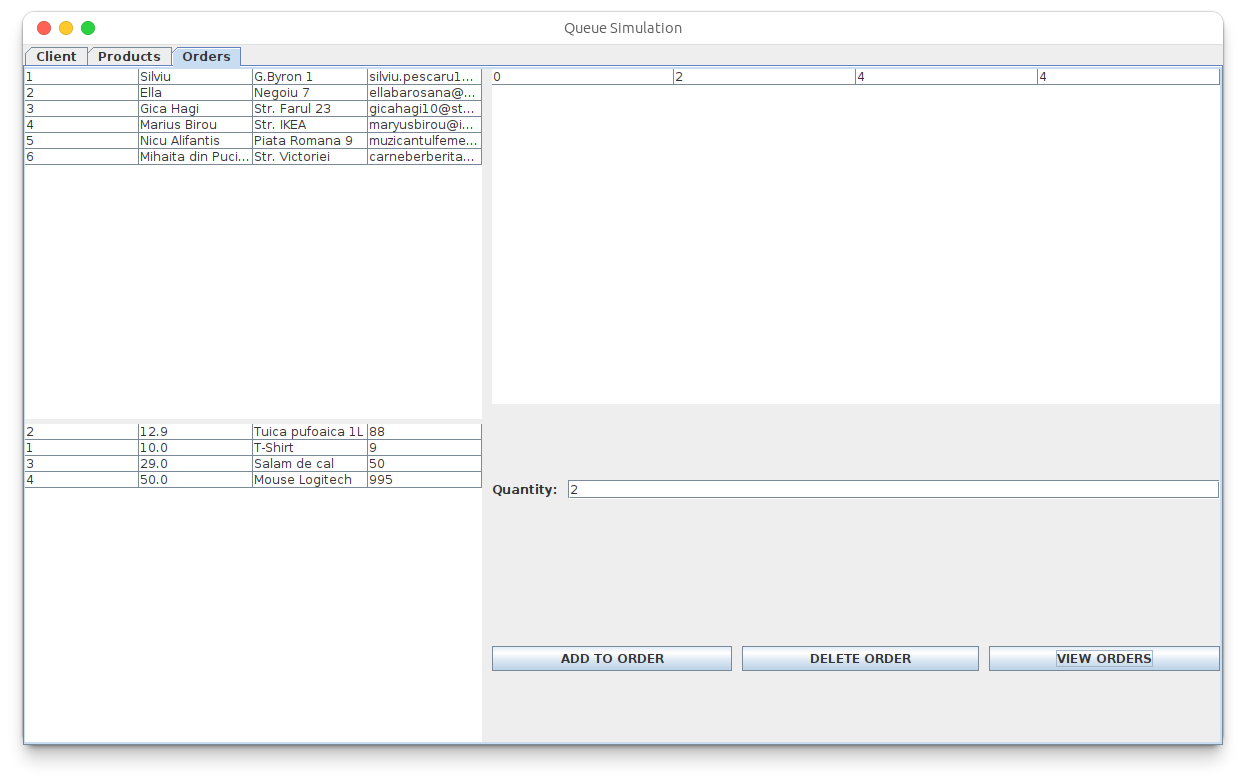
Scenarii: Nu se vor putea șterge produse sau clienți implicați într-o comandă plasată din cauza cheilor străine din tabelul Orders

Use cases



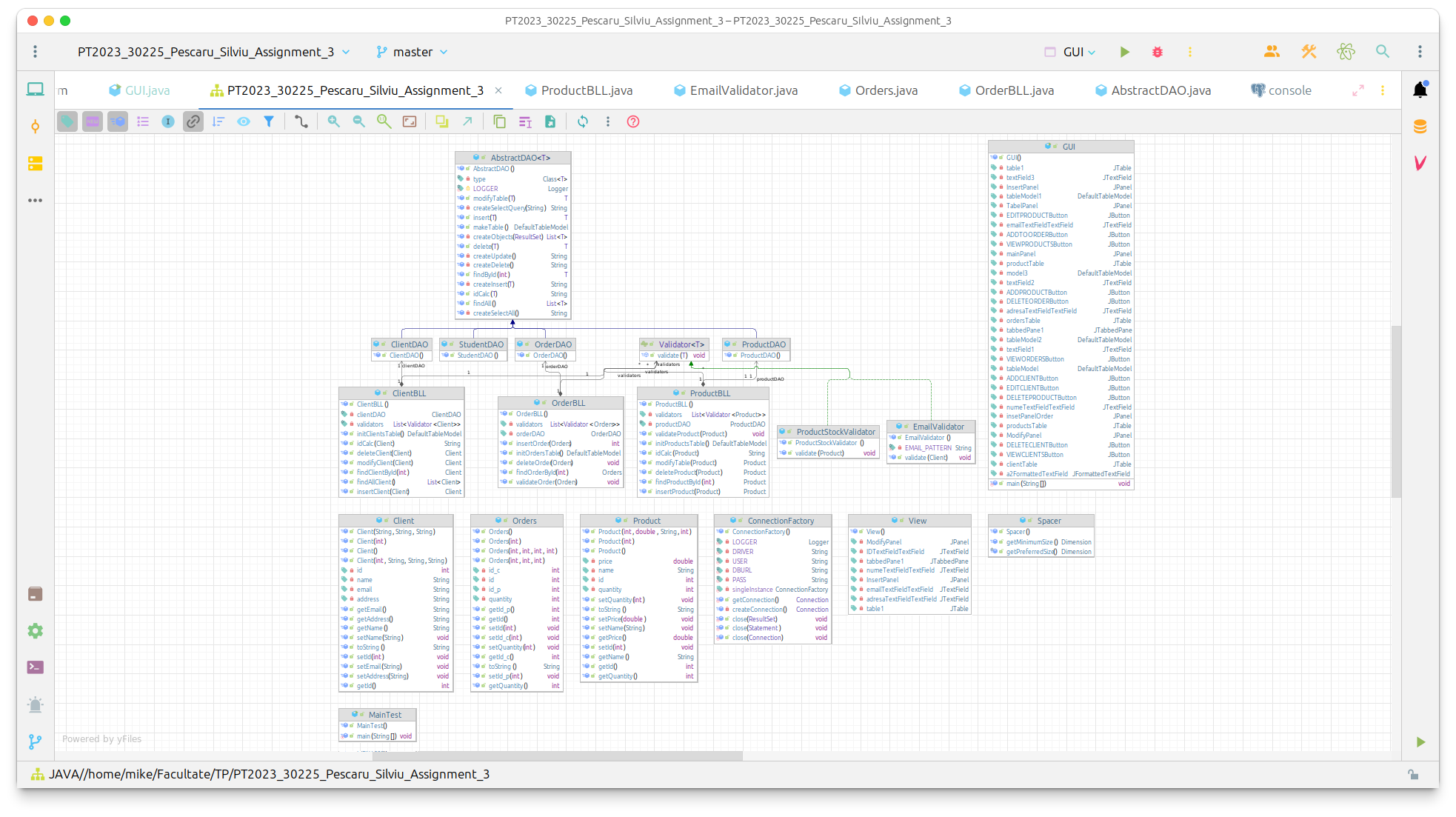




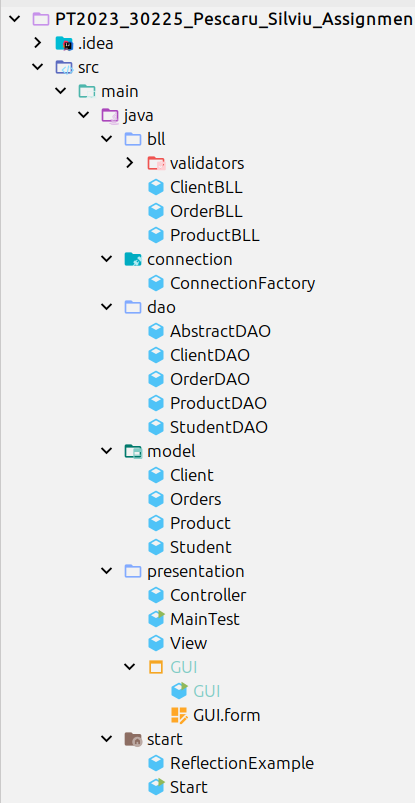


# Proiectare

Diagrama UML



Ierarhia de fișiere a proiectului (Arhitectura Layered)



# Implementare

Clasa GUI: clasa în care se instanțiază interfața grafică dar si actionListenerele

Clasele ClientBLL, ProductBLL, Order BLL: clasele de business logic pentru accesul de date al clienților, produselor, respectiv comenzilor

# Rezultate

Programul funcționează conform cerințelor, mai putin ca nu se poate insera mai mult de o comanda deoarece id-ul nu se incrementeaza.

# Concluzii

Cu acest proiect mi-am aprofundat cunostințele de a lucra cu JDBC.

Ca dezvoltare ulterioară se poate genera o factură pentru fiecare comandă plasată.

# Bibliografie

<https://www.baeldung.com/java-jdbc>

[http://www.mkyong.com/jdbc/how-to-connect-to-PostgreSQL-with-jdbc-driver-java/](http://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/)

<https://dzone.com/articles/layers-standard-enterprise>

<http://tutorials.jenkov.com/java-reflection/index.html>

<https://www.baeldung.com/java-pdf-creation>