

# **Proiect PCBE**

## **Bursa**

Echipa:

Danci Ionut-Cosmin

Dragoescu Cristian-Florin

Dumitrescu Bogdan-Doru

Cuprins :

1. Descriere
2. UML
3. Tools
4. Link repository
5. Probleme concurenta
6. Functionalitati nou-incluse

## 1. Descriere aplicatie

Am stabilit de comun acord arhitectura proiectului : - un sistem distribuit (server si clienti): in partea de server ne ocupam de thread-uri si logica aplicatiei - iar in partea de client exista 2 clase (buyer si seller) unde se realizeaza comunicarea pe socket cu serverul.

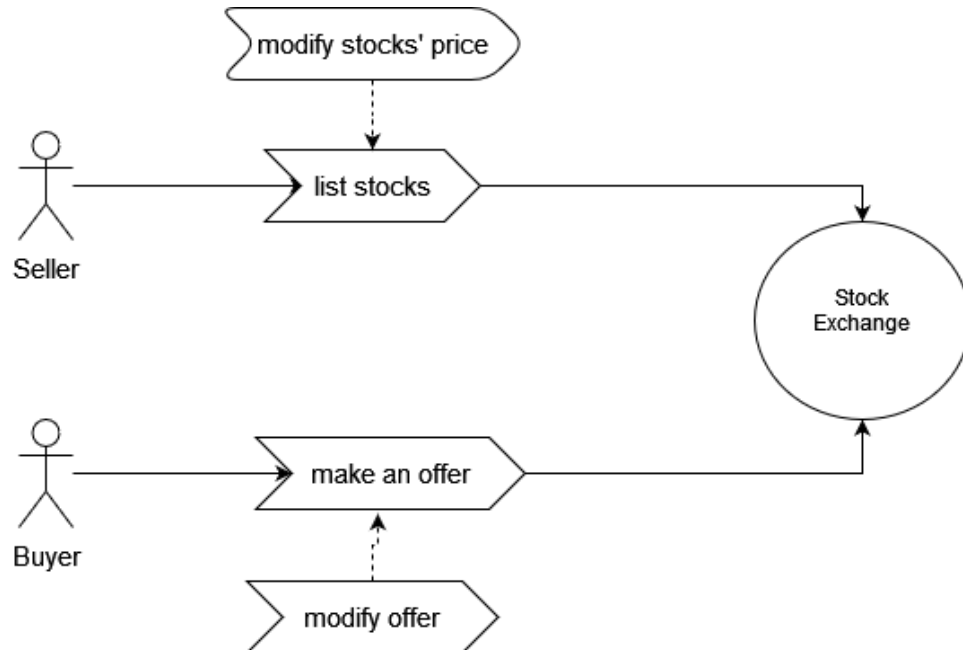
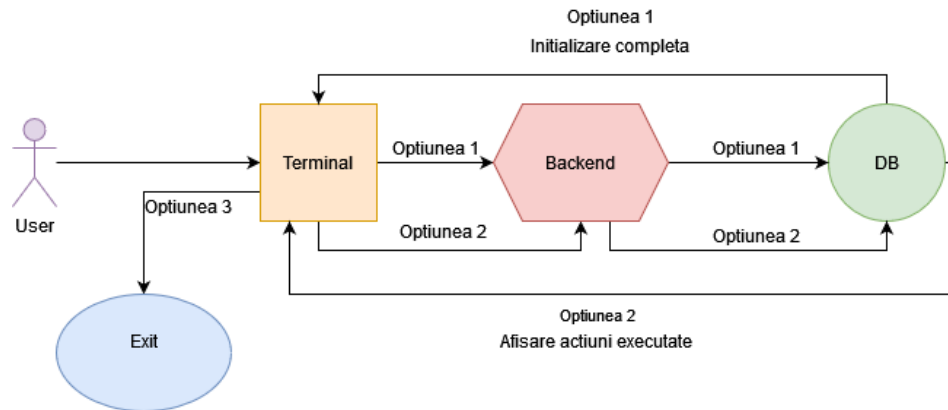
La executia programului se vor afisa mai multe optiunii intr-un meniu din terminal.

Prima optiune din meniu este cea de initializare a campurilor unor obiecte prin randomizare. In baza de date vor exista anumite actiuni, iar la initializare se vor aloca un anumit numar de actiuni la fiecare seller(un numar random, cu o valoare random).

A doua optiune este cea de simulare. Aceasta va genera evenimentele si thread-urile. Dupa fiecare tranzactie efectuata, baza de date va fi initializata. Aici se vor putea observa situatiile de concurenta in terminal in forma de log-uri(reprezentand diverse informatii despre buyer, seller si tranzactii). Daca obiectele nu sunt initializate folosind prima optiune, aceasta optiune nu poate fi executata.

A treia optiune este cea de oprire a programului.

## 2. UML



### 3. Tools

Tehnologii: Java, JMS, Maven

Baza de date aleasa: SQLite (sau MySQL)

Specific algorithms used: citim in buffered reader

Tools used for development: Intelij, Git, GitHub

### 4. Link repository

[https://github.com/cristiandrgg/Proiect\\_PCBE.git](https://github.com/cristiandrgg/Proiect_PCBE.git)

## 5. Probleme concurenta

În momentul în care se realizează o tranzacție, istoricul trebuie păstrat, și până ce se finalizează toate celelalte acțiuni o să fie blocate (să nu se poată face alte oferte).

Tehnica prin care un thread așteaptă execuția altor thread-uri înainte de a-și continua propria execuție se numește sincronizare. În Java avem următoarele mecanisme de sincronizare ale thread-urilor:

- Mecanismul Synchronize
- Metodele Wait, Notify și Notify All

O altă problemă de sincronizare apare atunci când mai mulți cumpărători licitează pentru aceeași acțiune la prețuri diferite. Acest mecanism blochează execuția thread-ului respectiv până în momentul în care se ajunge la prețul dorit de vânzător.

## 6. Functionalitati nou-incluse

- BuyerThread
  - Metoda run()
- SellerThread
  - Metoda run()
- BuyerProcess
- SellerProcess
- Server
- Seller
  - Constructor
  - Setters and Getters
- Buyer
  - Constructor
  - Setters and Getters
- Bid
  - Constructor
  - Setters and Getters
- Stock
  - Constructor
  - Setters and Getters