

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
7. Link-uri pentru commit-uri

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.

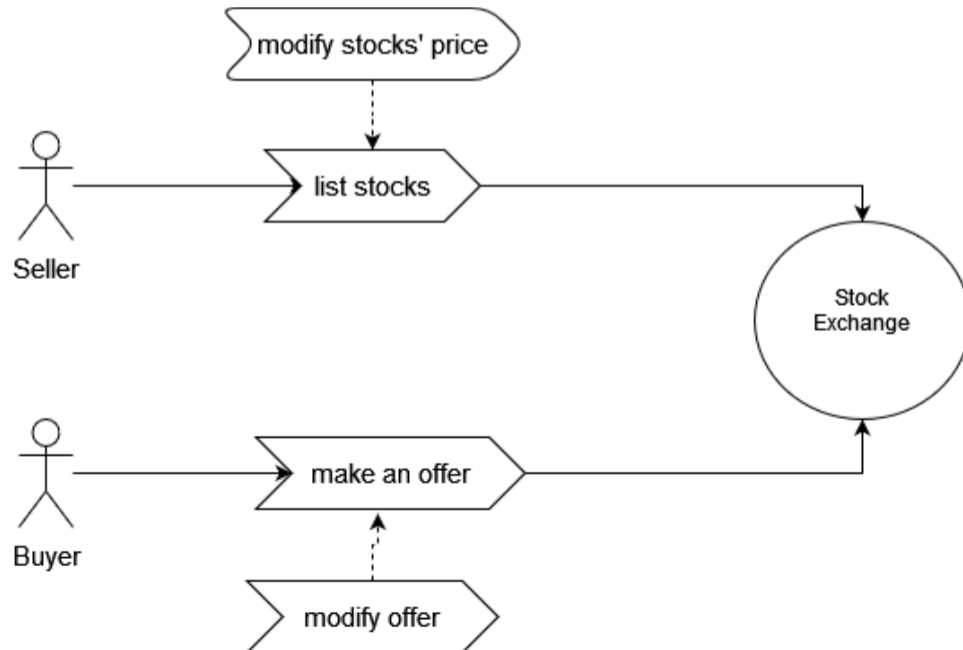
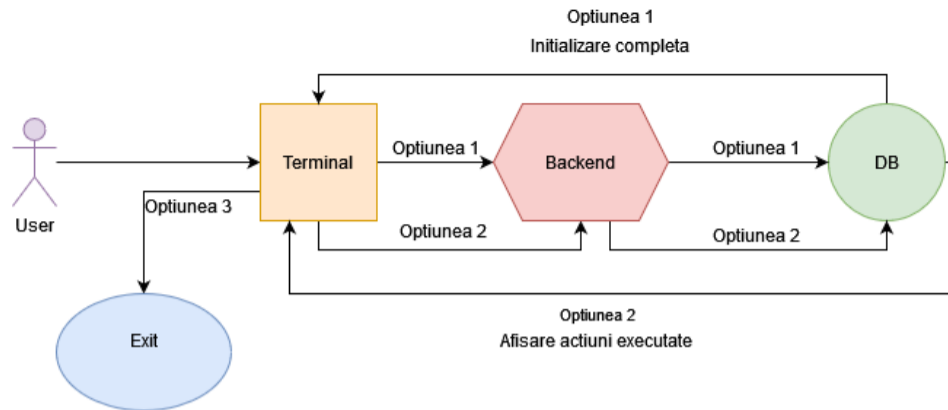
La executia programului se vor afisa mai multe optiunii intr-un meniu din terminal.

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

5. Probleme concurenta

In momentul in care se realizeaza o tranzactie, istoricul trebuie pastrat, si pana ce se finalizeaza toate celelalte actiunii o sa fie blocate (sa nu se poata face alte oferte).

Tehnica prin care un thread asteapta executia altor thread-uri inainte de a-si continua propria executie se numeste sincronizare. In java avem urmatoarele mecanisme de sincronizare ale thread-urilor:

- Mecanismul Synchronize
- Metodele Wait, Notify si Notify All

O alta problema de sincronizare apare atunci cand mai multi cumparatori liciteaza pentru aceasi actiune la preturi diferite. Acest mecanism blocheaza executia thread-ului respectiv pana in momentul in care se ajunge la pretul dorit de vanzator.

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

7. Link-uri pentru commit-uri

Autor	Descriere & Link
DumitrescuBogdanDoru (Dumitrescu Bogdan-Doru)	M-1: Milestone1 documentation added
DumitrescuBogdanDoru (Dumitrescu Bogdan-Doru)	M-2: empty project created
EvoShady (Danci Ionut-Cosmin)	Moved .gitignore to root
cristiandrpg (Dragoescu Cristian-Florin)	M-2: added Bid
cristiandrpg (Dragoescu Cristian-Florin)	M-2: added Seller
cristiandrpg (Dragoescu Cristian-Florin)	M-2: added Stock
DumitrescuBogdanDoru (Dumitrescu Bogdan-Doru)	M-2: added SellerProcess
DumitrescuBogdanDoru (Dumitrescu Bogdan-Doru)	M-2: added SellerThread
EvoShady (Danci Ionut-Cosmin)	M-2: Added Buyer Model
EvoShady (Danci Ionut-Cosmin)	M-2: Added Buyer Thread
DumitrescuBogdanDoru (Dumitrescu Bogdan-Doru)	M-2: modified Server