

Magazin de Parfumuri

Proiect realizat de: Dragotoniu Corina- Madalina





**CUPRINS**

* Introducere
* Specificațiile și analiza sistemului
* Diagrame de secventa + Diagrame Use Case
* Design Pattern
* Instrucțiuni și utilizare
* Concluzii și dezvoltări ulterioare

**Spring & java**

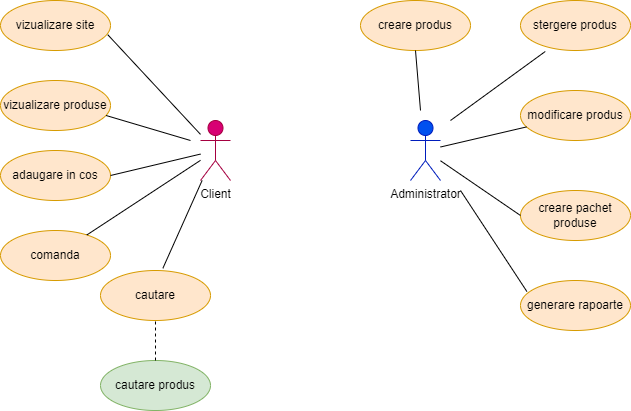
**Java** este un limbaj de programare orientat-obiect, puternic tipizat, conceput la începutul anilor ʼ90. Cele mai multe aplicații distribuite sunt scrise în Java, iar noile evoluții tehnologice permit utilizarea sa și pe dispozitive mobile, spre exemplu telefon, agenda electronică, etc. Acesta este utilizat în prezent cu succes și pentru programarea aplicațiilor destinate intranet-urilor. Limbajul împrumută o mare parte din sintaxă de la C și C++, dar are un model al obiectelor mai simplu și prezintă mai puține facilități de nivel jos.

**** **Spring Framework** este o platformă cu sursă deschisă pentru simplificarea scrierii aplicațiilor în limbajul Java. Deși este folosit în principal pentru platforma Java EE, Spring poate fi utilizat pe orice aplicație Java. Framework-ul Spring furnizează acces la baze de date și posibilitatea de a crea aplicații web dinamice.

**Specificațiile sistemului**

Acest proiect incearca sa simuleze un magazin de parfumuri, utilizatorul putand sa vizualizeze produsele, sa caute parfumuri in functie de diverse criterii, sa adauge produse la comanda si la final sa plaseze o comanda. Pentru a putea vizualiza produsele si a realiza comenzi, un utilizator trebuie sa isi faca un cont si sa se logheze in aplicatie.Atunci cand un client plaseaza o comanda, se creeaza o chitanta cu toate detaliile tranzactiei. Avem, de asemenea, si un administrator care are mai multe posibilitati de actiune: sa creeze un produs nou, sa stearga un produs deja existent, sa modifice un produs si sa creeze pachete de produse (poate grupa mai multe parfumuri, formand un nou produs, precum sunt pachetele promotionale). De asemenea, poate genera diverse rapoarte, pentru a gestiona mai bine toate achizitiile facute de clienti.

**Diagrama Use Case**



Cele 2 roluri posibile sunt: client și administrator.

**Un client** are posibilitatea de a-și crea cont sau de a se loga în contul său deja existent. Odată logat, acesta poate să vizualizeze produsele, să caute anumite produse după nume sau după categorie, să adauge produse în coș și ulterior să plaseze comenzi.

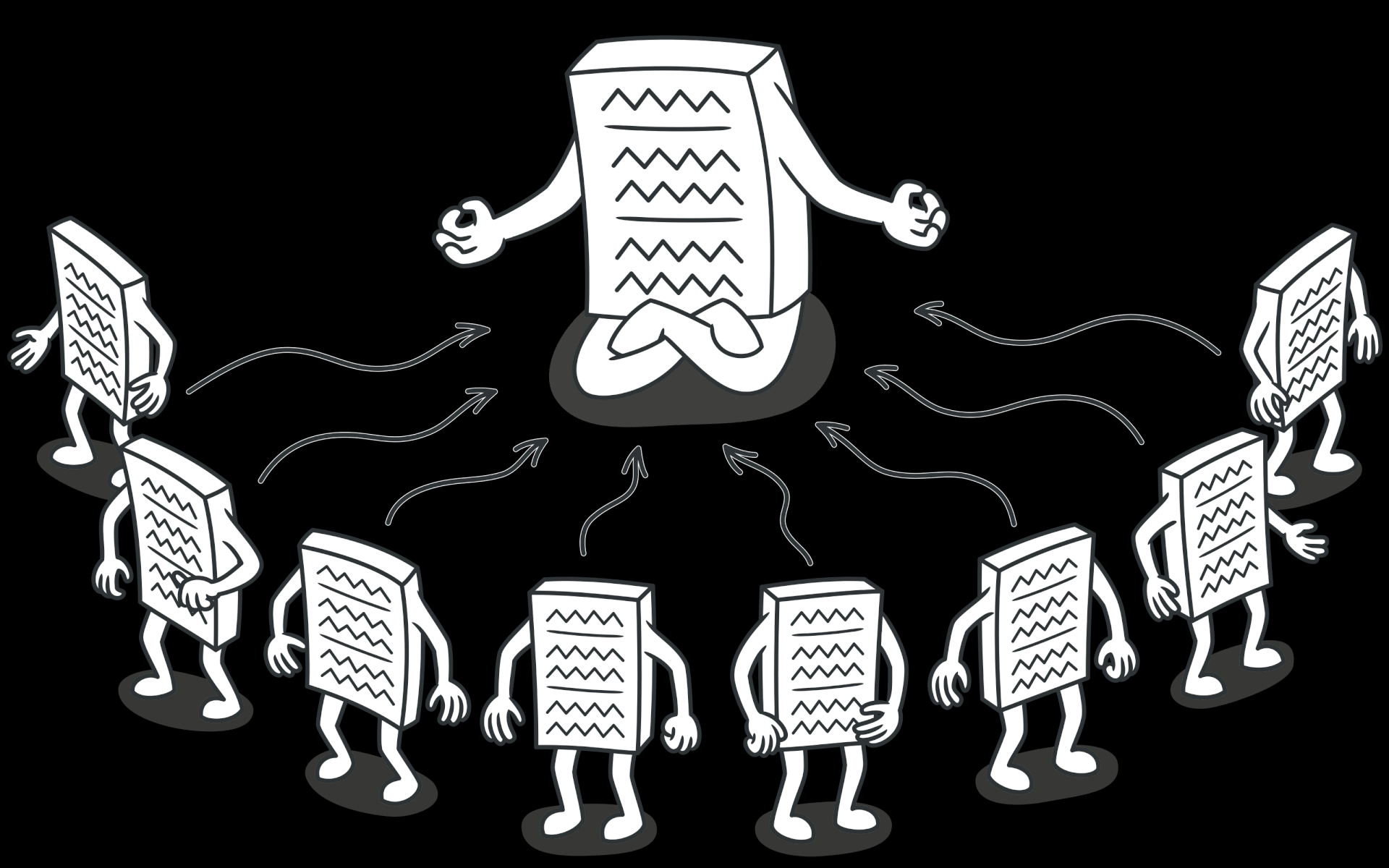
**Administratorul** are posibilitatea de a adauga produse , de a sterge produse existente, de a modifica anumite produse si de a crea pachete de mai multe parfumuri. De asemenea, poate genera rapoarte cu privire la comenzile plasate de clienti.



Mai sus este diagrama de secventa pentru un client, in care sunt reprezentate toate etapele prin care poate si trebuie sa treaca un client pentru a plasa o comanda.

**Design Pattern**

Uneori ne dorim sa avem un obiect care sa apara doar o singura data intr-o aplicatie, in cazul nostru un administrator care se ocupa de gestionarea magazinului (deoarece nu are sens sa existe mai multi in acelasi context). De aceea folosim Singleton, un mod prin care restrictionam numarul de instantieri a unei clase, ea avand o singura referinta ce va fi folosita in intreg proiectul. Astfel, pentru a asigura restrictionarea, clasa de tip Singleton va avea un constructor privat (il facem privat prin a bloca instantierea multipla a clasei), care va fi apelat in interiorul clasei. Pentru a putea crea o instanta unica a clasei (prin care se asigura o economisire a memoriei), vom avea nevoie de un membru static si privat, care va avea acelasi tip cu clasa pe care vrem o sa instantiem, si o metoda statica si publica prin care este returnata instanta clasei.

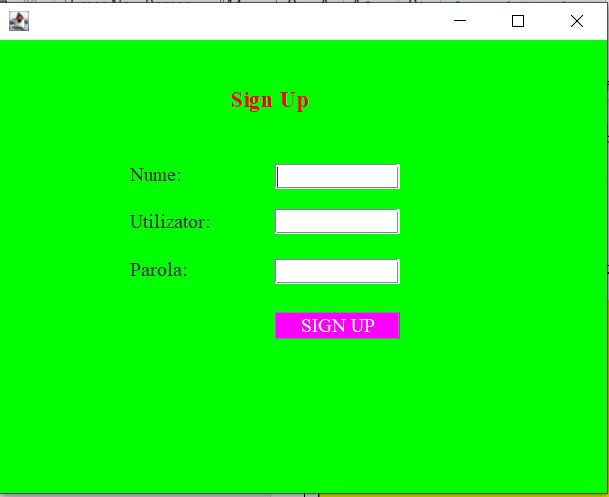


Astfel, in cadrul aplicatiei mele, am folosit acest design pattern pentru a crea o singura instanta a clasei “Admin”, ce poate nu poate fi modificata ulterior, ci doar accesata.

**Instructiuni de utilizare**

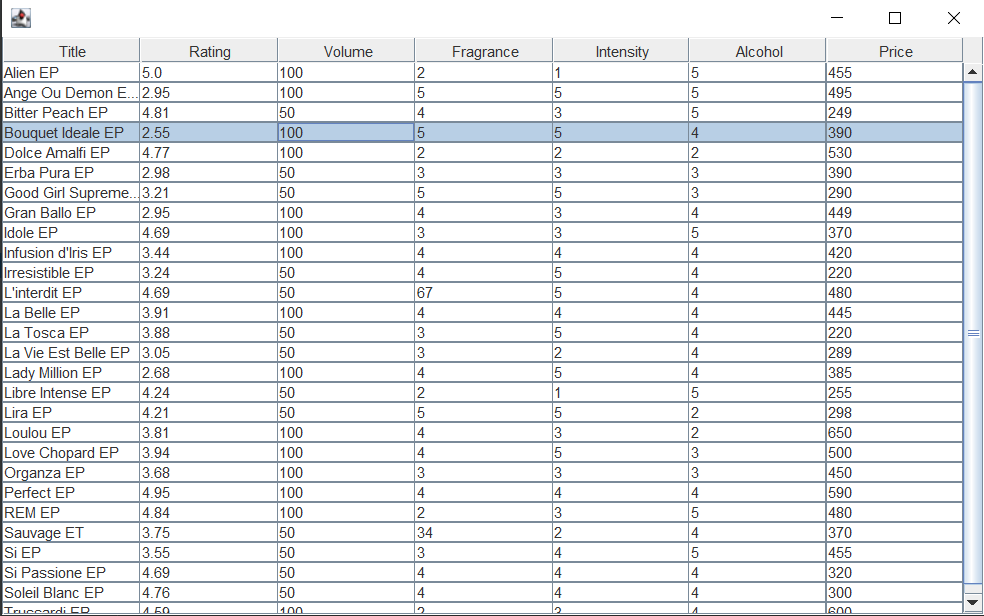
In momentul in care rulam aplicatia, ne apare fereastra de login, unde ne putem conecta daca avem un cont deja existent sau unde ne putem inregistra.



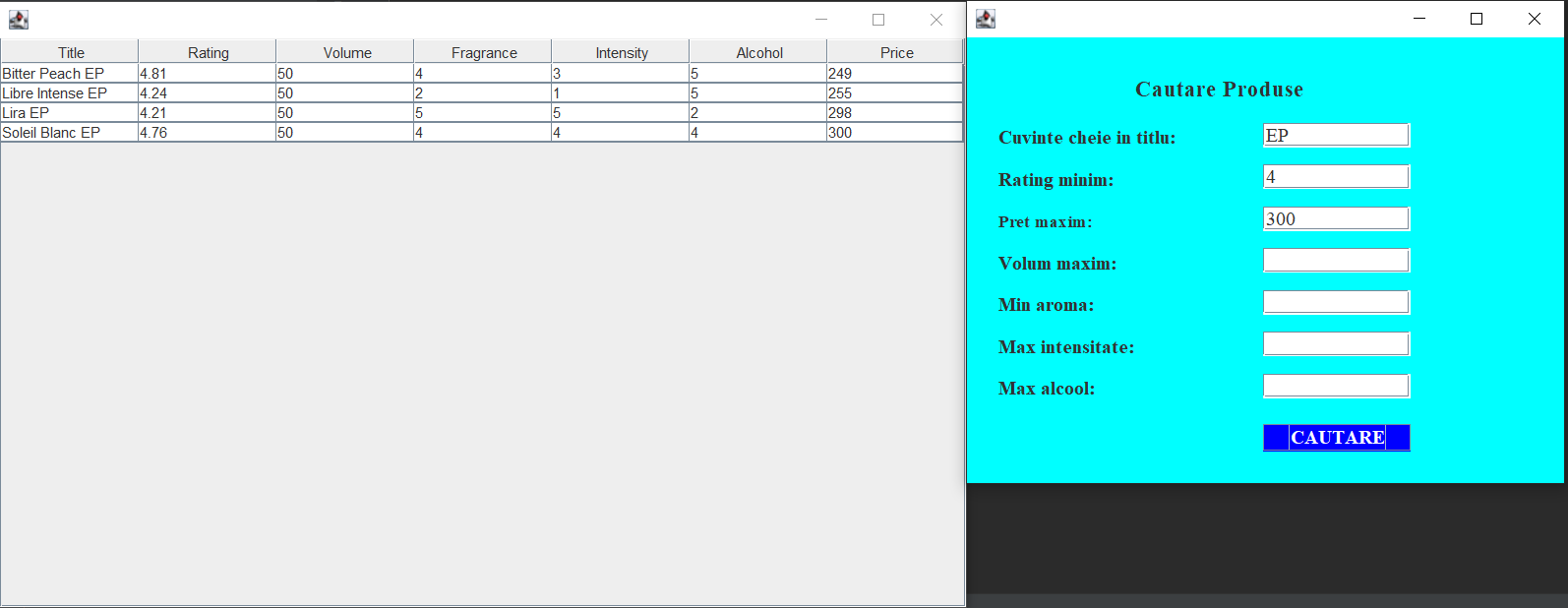


Daca apasam butonul de inregistrare, ni se deschide o noua fereastra de Sign Up, in care ne putem crea un cont completand cu un nume, un username si o parola. Daca totul este in regula, ne apare un mesaj in care ne anunta ca noul cont a fost creat cu succes.

Odata ce am intrat in contul de client, ne apare un tabel cu toate produsele disponibile in stoc, cu detalii despre ele, inclusiv volum, intensitate si pret.



Dand click dreapta pe un produs din tabel, avem urmatoarele variante: sa adaugam produsul curent la comanda, sa finalizam comanda sau sa cautam un produs dupa anumite criterii. Daca alegem varianta de cautare produs, ne apare urmatoarea pagina, iar rezultatele cautarii ne actualizeaza in timp real tabelul:



Dupa ce am plasat o comanda, ni se genereaza un fisier .txt care contine toate detaliile acelei comenzi, reprezentand un fel de chitanta.

Logandu-ne ca si administrator, ni se deschide urmatoarea fereastra, din care putem alege mai multe operatii: sa importam produsele din stoc pentru a fi vizibile de clienti, sa gestionam produsele sau sa generam rapoarte despre comenzi in functie de anumite criterii dorite de noi.

O imagine care conține text

Descriere generată automat

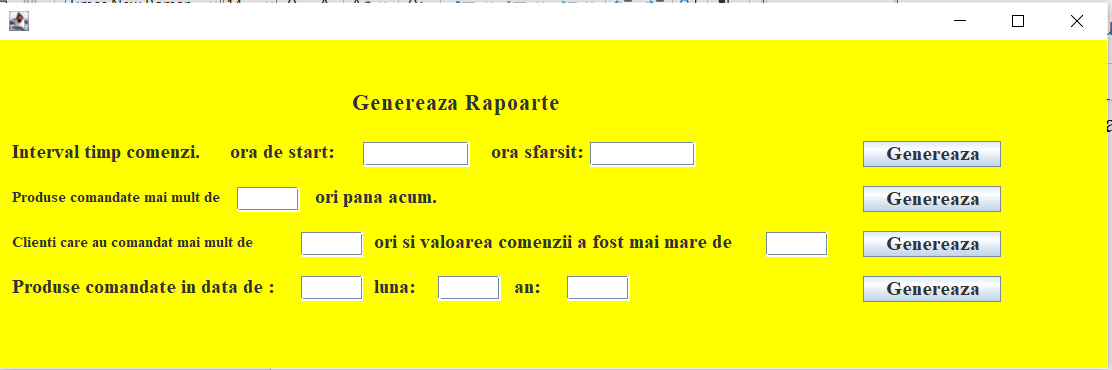
Daca intram la gestionarea produselor, avem urmatoarele variante:

O imagine care conține masă

Descriere generată automat

Putem sa adaugam un produs nou, sa stergem unul deja existent, sa salvam un produs pe care il modificam si sa cream pachete de produse.

Daca alegem sa generam rapoarte, avem posibilitatea de a alege intre 4 tipuri: sa vedem comenzile intr-un anumit interval orar, sa vedem produsele comandate de cel putin n ori pana acum, clientii care au comandat mai mult de n ori pana acum si de o anumita valoare sau doar produsele comandate intr-o anume data.



La final, nu este nevoie sa ne deconectam din cont, ci doar sa inchidem pagina.

**Concluzii si dezvoltari ulterioare**

In concluzie, acest proiect a fost un bun exercitiu pentru a invata sa gestionez mai bine lucrul cu diverse clase din Java, sa folosesc expresii lamda si sa realizez utilitati ale unei aplicatii in viata reala. O dezvoltare ulterioara ar fi sa ma folosesc de partea de backend creata pentru a realiza un site web, accesibil tuturor.