Magazin online de incaltamine

Proiect realizat de:

Modrea Elena-Alexandra

Fazakas Edina

Nita Iosif-Gabriel

Grupa: 30233

**Cuprins**

1. Introducere
2. Modele use-case
3. Arhitectura
   1. Descrierea arhitecturii
   2. Diagrama arhitecturii
   3. Diagrama bazei de date
4. Specificatii
5. Testare si validare
   1. Validare
   2. Testare
6. Dezvoltari ulterioare
7. Bibliografie
8. **Introducere**

Magazinul de incaltaminte are 3 useri:

* Angajat
* Client
* Administrator

Administratorul are urmatoare atributii:

* Se poate loga: acesta este singurul care nu are nevoie sa se inregistreze, contul lui este deja creat;
* Trimite credentiale noilor angajati
* Plateste salariile angajatilor
* Poate sterge produse
* Poate modifica produse
* Trimite o nootificare anagajatului in momentul in care un produs a revenit in stoc

Angajatul are urmatoarele atributii:

-primeste pe email credentialele pentru a se putea loga; la finalul contractului credentialele nu mai sunt valabile;

-verifica stocul; trimite notificari adminului daca produsele nu mai sunt pe stoc;

-trimite un email clientului in momentul in care comanda e confrmata;

-trimite un email clientului interest de un anumit produs in momentul in care produsul a revenit pe stoc;

-preia feedbackul clientului (stele);

-primeste cereri de retur;

-modifica produse(pret, marime);

Clientul are urmatoarele atributii:

-se poate inregistra, primeste un email cu un cod de confirmare pentru a putea crea contul;

-poate sa adauge si sa stearga produse din cos;

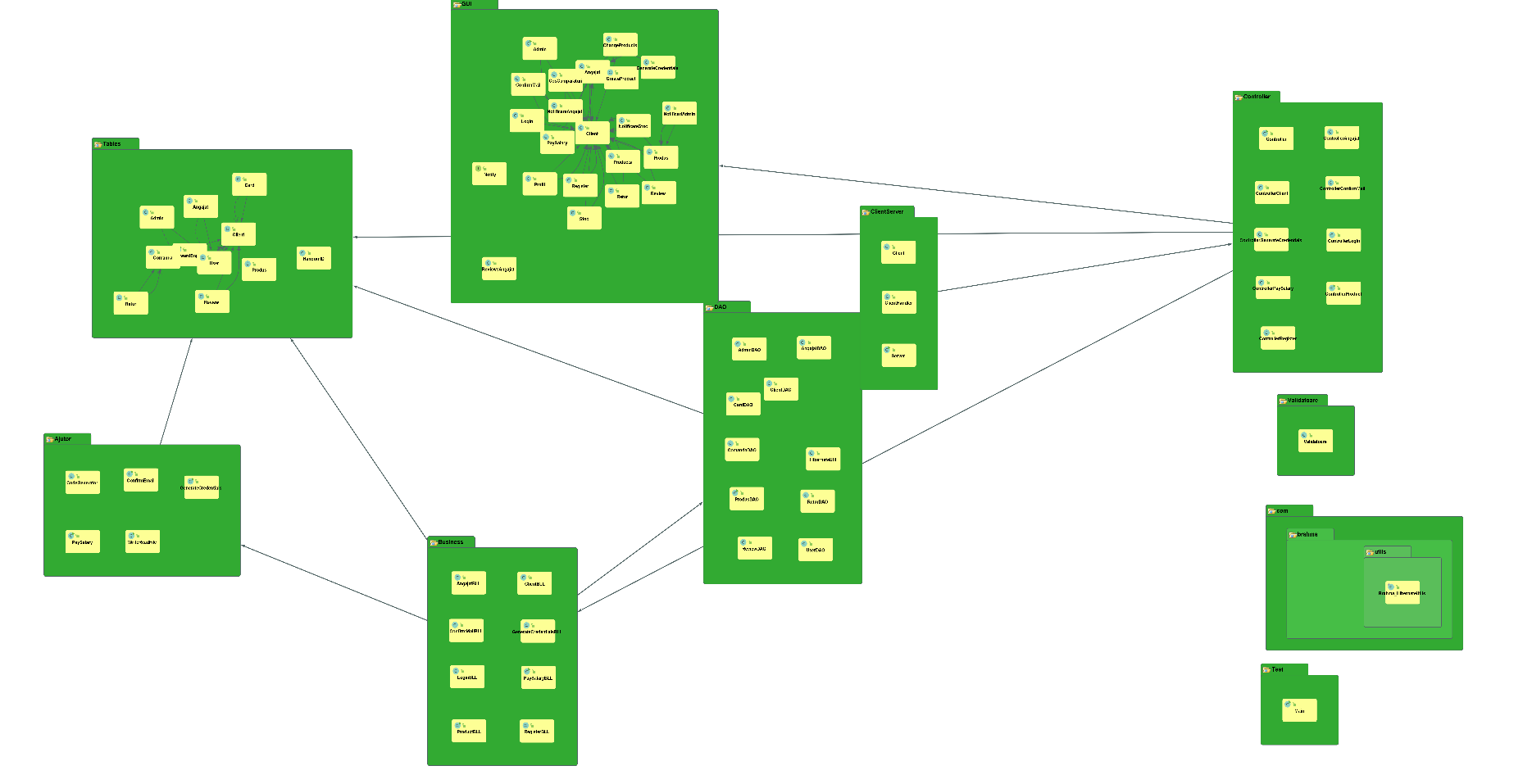
-sa lase feedback;

-se poate abona la newsletter;

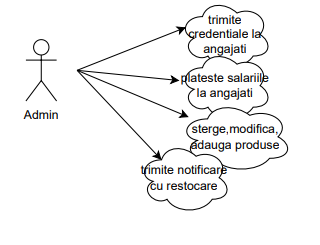
- poate cauta produse dupa criterii(filter), poate sorta(crescator, descrescator, A-Z, Z-A);

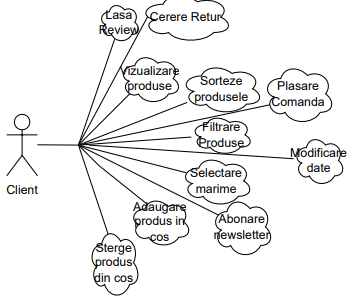
-are un profil; isi poate modifica adresa, cardul, isi poate schimba parola si date;

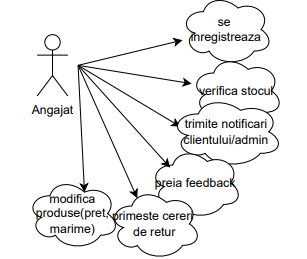
- lasa cereri de retur la comenzi;

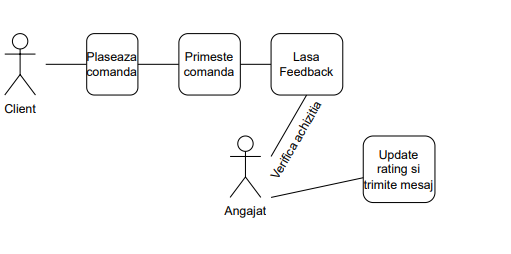
Ca si produse disponibile in magazinul online vom avea incaltaminte pentru femei, barbati, fete, baieti de marimi diferite.

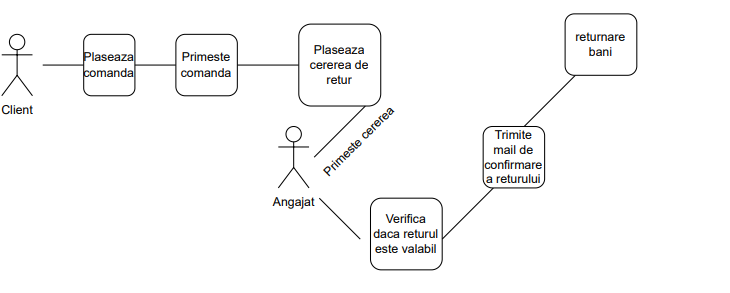
1. **Modele use-case**

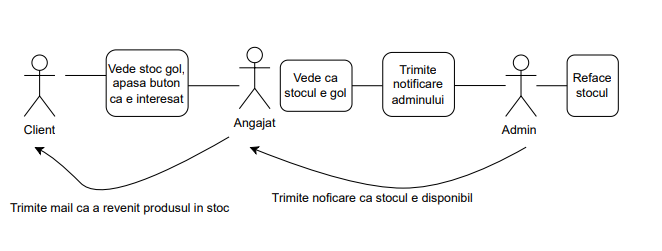
****

****

****

****

****

****

**3.Arhitectura**

**3.1. Descrierea arhitecturii**

Proiectul este alcatuit din 9 pachete si 70 de clase.

Pachetele proiectului sunt: Ajutor, Business, ClientServer, Controller, DAO, GUI, Tables, Test, Validatoare.

Proiectul functioneaza pe arhitectura Client-Server.

Clientul este reprezentat de pachetul GUI ce contine interfetele proiectului.

Serverul este reprezentat in pachetul ClientServer.

Controller-ul este reprezentat in pachetul Controller.

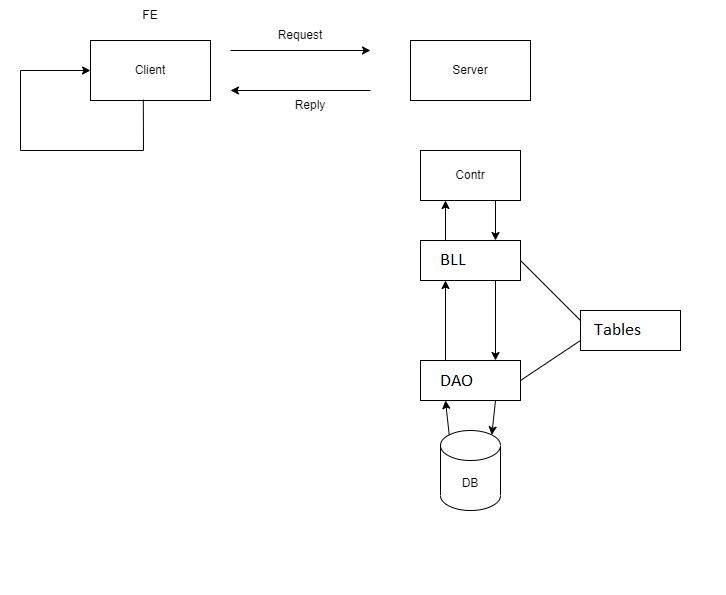
Serverul comunica atat cu view-ul cat si cu controller-ul prin socket-uri astfel: View-ul trimite un request catre server, acesta il interpreteaza si il trimite controllerului care la randului va trimite cerinta catre BLL. In BLL este rezolvata cerinta din socket, apoi este creat un response care va fi trimis de la server la view.

DAO este singurul care comunica cu baza de date.

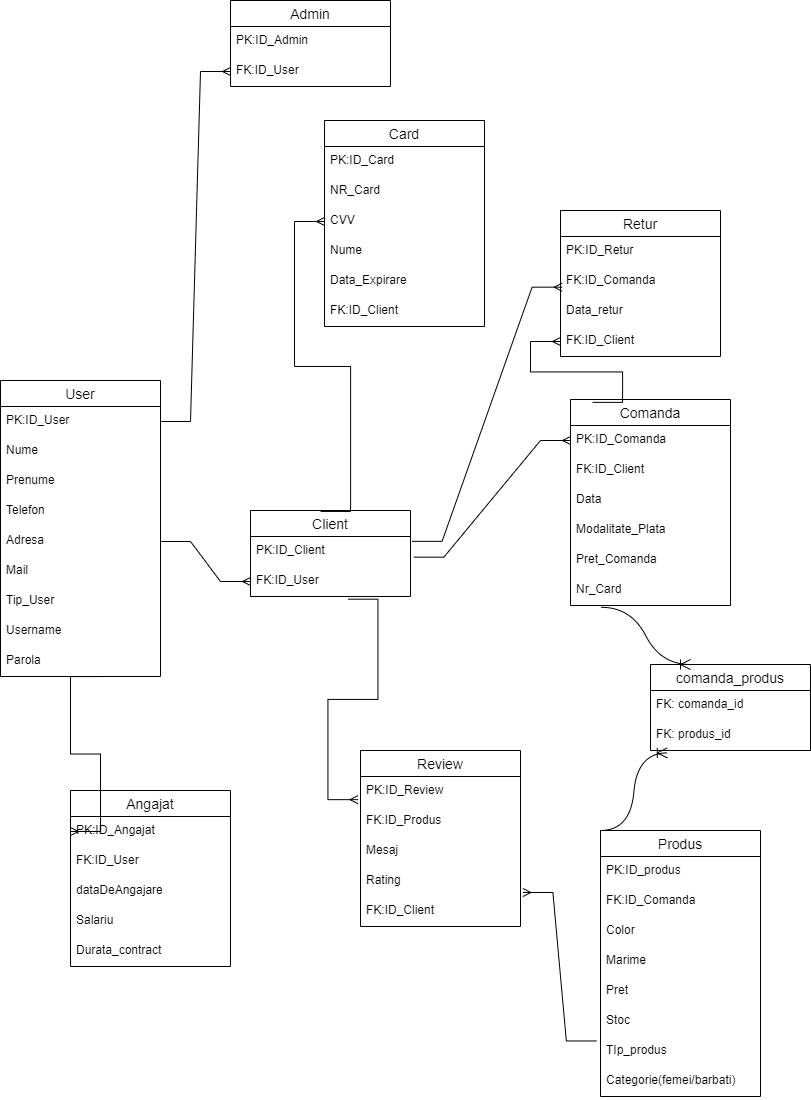
De asemenea, in acest proiect vom folosi hibernate (orm) cu ajutorul caruia cream automat tabele in baza de date. De asemenea cu ajutorul orm-ului hibernate putem realiza operatiile CRUD pe baza de date.

Ca baza de date am folosit baza de date PostgreSQL.

**3.2. Diagrama arhitecturii**

****

**3.3. Diagrama bazei de date**

****

**4.Specificatii**bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

Ca specificatii non-functionale avem urmatoarele:

1. Confirmarea contului printr-un cod trimis pe email
2. Trimiterea credentialelor pentru angajati pe email
3. Sesiune de Securitate
4. Pentru realizarea acestei specificatii ne folosim de clasa CodeGenerator care creaza un cod de 6 cifre random, dar si de clasa ConfirmEmail cu ajutorul caruia trimitem email-ul user-ului.
5. Similar cu primul punct am procedat si pentru aceasta specifcatie, doar ca aici am generat random si username format din prenumele userului si 4 cifre random.
6. Sesiunea de Securitate este folosita in momentul in care cream un cont nou, atunci cand trebuie sa adaugam contul pentru ca se putea crea contul si in momentul in care ne aflam la profilul clientului.

Pentru a realiza aceasta specificatie am folosit un timer, iar in momentul in care timerul s-a oprit se apeleaza metoda: System.exit(0).

**5. Validare si testare**

5.1. Validare

Pentru validare am create un pachet Validatoare si o clasa cu acelasi nume.

Am create validatoare pentru:

* Email:

O imagine care conține text

Descriere generată automat

* Telefon:

O imagine care conține text

Descriere generată automat

* Rating

O imagine care conține text

Descriere generată automat

* O imagine care conține text

  Descriere generată automatCard Number
* CVV
* O imagine care conține text

  Descriere generată automatExpiration date

O imagine care conține text

Descriere generată automat

* Data
* O imagine care conține text

  Descriere generată automatSQL Injection

O imagine care conține text

Descriere generată automat

5.2.Testare

Pentru aceasta aplicatie am folosit Unit Test.

Am testat metodele din pachetul BLL.

Cateva exemple de testare:

* LoginBLL:

O imagine care conține text, monitor, captură de ecran, ecran

Descriere generată automat

* RegisterBLL:

O imagine care conține text, captură de ecran, monitor

Descriere generată automat

* ClientBLL:

O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

* AngajatBLL:

O imagine care conține text, monitor, negru, captură de ecran

Descriere generată automat

O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

* GenerateCredentialsBLL:

Text

Description automatically generated

* PaySalaryBLL:

Text

Description automatically generated

* ProductBLL:

Text

Description automatically generated

**6. Dezvoltari ulterioare**

Pentru viitor ne-am propus sa implementam urmatoarele functionalitati:

* Dezvoltarea pe web a aplicatiei;
* Adaugarea unui chat in care angajatul sa poata ajuta clientul;
* Dezvoltarea aplicatiei astfel incat sa putem adauga alte tipuri de produse pe langa incaltaminte

**7.Bibliografie**

* <https://www.geeksforgeeks.org/automatic-table-creation-using-hibernate/>
* <https://www.geeksforgeeks.org/socket-programming-in-java/>
* <https://www.dineshonjava.com/hibernate/crud-operations-using-hibernate-3/>
* <https://stackoverflow.com/questions/23107944/spring-hibernate-illegal-attempt-to-associate-a-collection-with-two-open-sessi>