Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

Lanț de magazine de încălțăminte

Profesor îndrumător: Student:

Iordan Anca Andriescu Antonio

Data: 18.04.2022 Grupa 30238

Cuprins

[1. Enunțul problemei 1](#_Toc101172736)

[2. Instrumente utilizate 2](#_Toc101172737)

[2.1. MySQL 2](#_Toc101172738)

[2.2. Singleton 2](#_Toc101172739)

[2.3. Command 2](#_Toc101172740)

[2.4. Java Swing 2](#_Toc101172741)

[3. Justificarea limbajului de programare ales 3](#_Toc101172742)

[4. Descrierea diagramelor UML 4](#_Toc101172743)

[4.1. Diagrama cazurilor de utilizare 4](#_Toc101172744)

[4.2. Diagrama de clase 4](#_Toc101172745)

[4.3. Diagrama de relaționare a entității 6](#_Toc101172746)

[5. Descrierea aplicației 6](#_Toc101172747)

[6. Credentiale pentru conectare 8](#_Toc101172748)

# Enunțul problemei

Dezvoltați (analiză, proiectare, implementare) o aplicație desktop care poate fi utilizată într-un lanț de magazine de încălțăminte. Aplicația va avea două tipuri de utilizatori: angajat al lanțului de magazine de încălțăminte și administrator. Utilizatorii de tip angajat al lanțului de magazine de încălțăminte pot efectua următoarele operații după autentificare:

❖Vizualizarea tuturor produselor de încălțăminte disponibile într-un magazin selectat;

❖Filtrarea produselor de încălțăminte după următoarele criterii: producător, disponibilitate, preț;

❖Căutarea unui produs de încălțăminte după denumire și/sau producător;

❖Operații CRUD în ceea ce privește persistența produselor de încălțăminte din magazinul la care lucrează acel angajat;

❖Salvare rapoarte/liste cu produse de încălțăminte în mai multe formate: csv, json, xml;

❖Vizualizarea unor statistici legate de produsele din magazinul la care este angajat: procente după disponibilitate, preț, producător utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

❖Operații CRUD pentru informațiile legate de utilizatori.

# Instrumente utilizate

## MySQL

MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) dezvoltat de Oracle, care se bazează pe limbajul de interogare structurat SQL. Am folosit MySQL pentru bazele de date in care am avut tabelul cu utilizatori si tabelul cu produsele de încălțăminte.

## Singleton

Singleton este un șablon de proiectare creational care permite asigurarea faptului ca o clasa are o singura instanță, oferind in același timp un punct de acces global la aceasta instanță. Am implementat șablonul de proiectare creational Singleton pentru a putea avea o singura conexiune cu baza de date.

## Command

Command este un șablon de proiectare comportamental care transforma o solicitare intr-un obiect autonom care conține toate informațiile despre cerere. Aceasta transformare sa trecem cererile ca argumente de metoda, sa întârziem sau sa punem in așteptare cererea unei execuții si sa putem avea operații care nu pot fi anulate. Am utilizat Command pentru butoanele interfeței grafice, deci fiecare buton are o clasa in care se face logica, iar in clasa View doar se apelează „execute”.

## Java Swing

Java Swing este un set de instrumente pentru crearea de interfețe grafice care conține un set bogat de widget-uri. Face parte din Java Foundation Classes (JFC) și include mai multe pachete pentru dezvoltarea aplicațiilor desktop. Swing include comenzi incorporate cum ar fi arbori, butoane de imagine, panouri cu file, glisoare, bare de instrumente, selectare de culori, tabele, zone de text etc. Componentele Swing sunt scrise în Java și prin urmare, sunt independente de platformă.

Swing facilitează personalizarea aspectului fiecărei componente dintr-o aplicație fără a aduce modificări semnificative codului aplicației. Implementarea de interfețe grafice se realizează ușor utilizând Java Swing, iar prelucrarea datelor si afișarea acestora in interfață este destul de user-friendly.

# Justificarea limbajului de programare ales

De la web la mobil la desktop, Java se găsește în aproape toate sistemele de operare. Folosit în mod obișnuit împreună cu aplicațiile de afaceri și sistemele de operare mobile Android, limbajul de programare Java este actualizat constant pentru a ține pasul cu tendințele din industrie și poate accepta multe limbi pe platforma sa.

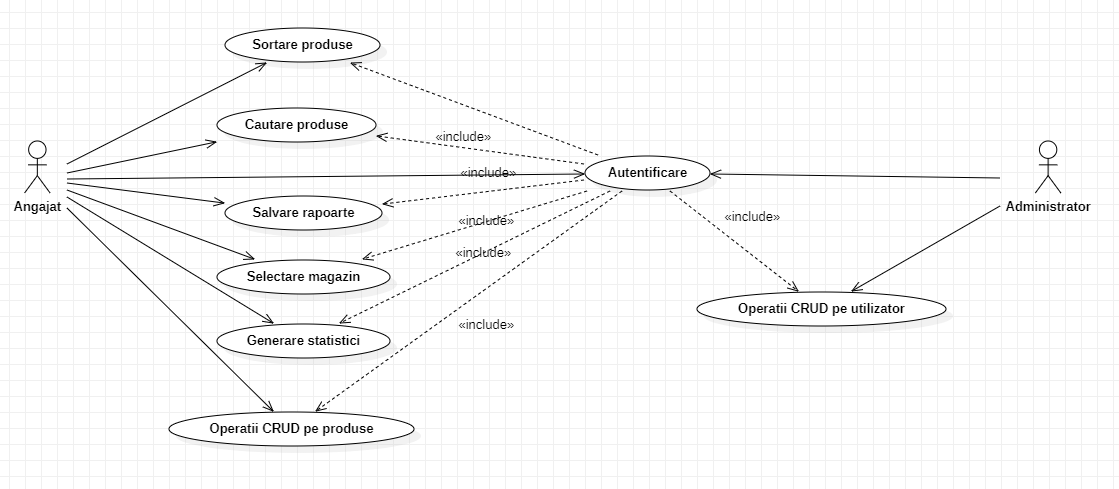
Java a fost conceput pentru a fi prietenos pentru începători, creând un mediu sigur pentru ca dezvoltatorii noi să se joace cu limbajul și să facă greșeli. Deși este simplu de utilizat, Java este un limbaj divers și de nivel înalt, care permite utilizatorilor să se concentreze mai degrabă pe programare decât pe detalii mai complexe ale mașinii. De fapt, instrumentele avansate de gestionare a memoriei Java se ocupă automat de alocarea, realocarea și dealocarea memoriei printr-un instrument software numit Garbage Collector. În plus, Java are o sintaxă asemănătoare englezei cu caractere magice minime, ceea ce face ușor de citit și de învățat programe Java.

Nu numai că Java este utilizat universal în tehnologie, dar este ușor scalabil și, cel mai important, foarte portabil, deoarece trebuie executat printr-o mașină virtuală Java (JVM) compatibilă cu mai multe platforme. Acest JVM trimite instrucțiuni specifice platformei adecvate computerului, adică dacă un dezvoltator scrie cod odată ce acesta poate fi transferat cu ușurință într-un alt sistem de pe aproape orice dispozitiv de calcul. De asemenea, Java le permite dezvoltatorilor să creeze software pe piloni de aplicații moderni, care sunt siguri, robuști și scalabili, făcând Java o alegere versatilă pentru computere, aplicații, console de jocuri și multe altele.

Personal am ales Java deoarece este un limbaj des utilizat și căutat în industrie. Alegerea făcută îmi permite sa învăț mai multe despre acest limbaj și despre ce poate oferi. Alegerea a fost una sigură deoarece suportul oferit utilizatorilor de Java este vast, acest aspect ajutându-mă să găsesc rapid și eficient soluții la problemele pe care le-am întâmpinat pe parcursul dezvoltării aplicației. Pe lângă aceste aspecte, Java este o alegere foarte bună pentru dezvoltarea de aplicații după modelul OOP.

# Descrierea diagramelor UML

## Diagrama cazurilor de utilizare



În diagrama cazurilor de utilizare am reprezentat cei doi actori (angajatul și administratorul) și ce acțiuni poate face fiecare. Când un utilizator se autentifică, in funcție de contul pe care îl are va putea face un set de acțiuni.

Administratorul poate face operații CRUD pe utilizatori, acesta putând să vadă toți utilizatorii înregistrați în aplicație.

Angajatul poate sorta/filtra produsele, poate vedea toate produsele disponibile, poate căuta produse in funcție de unele criterii și poate face operații CRUD pe toate produsele.

## Diagrama de clase

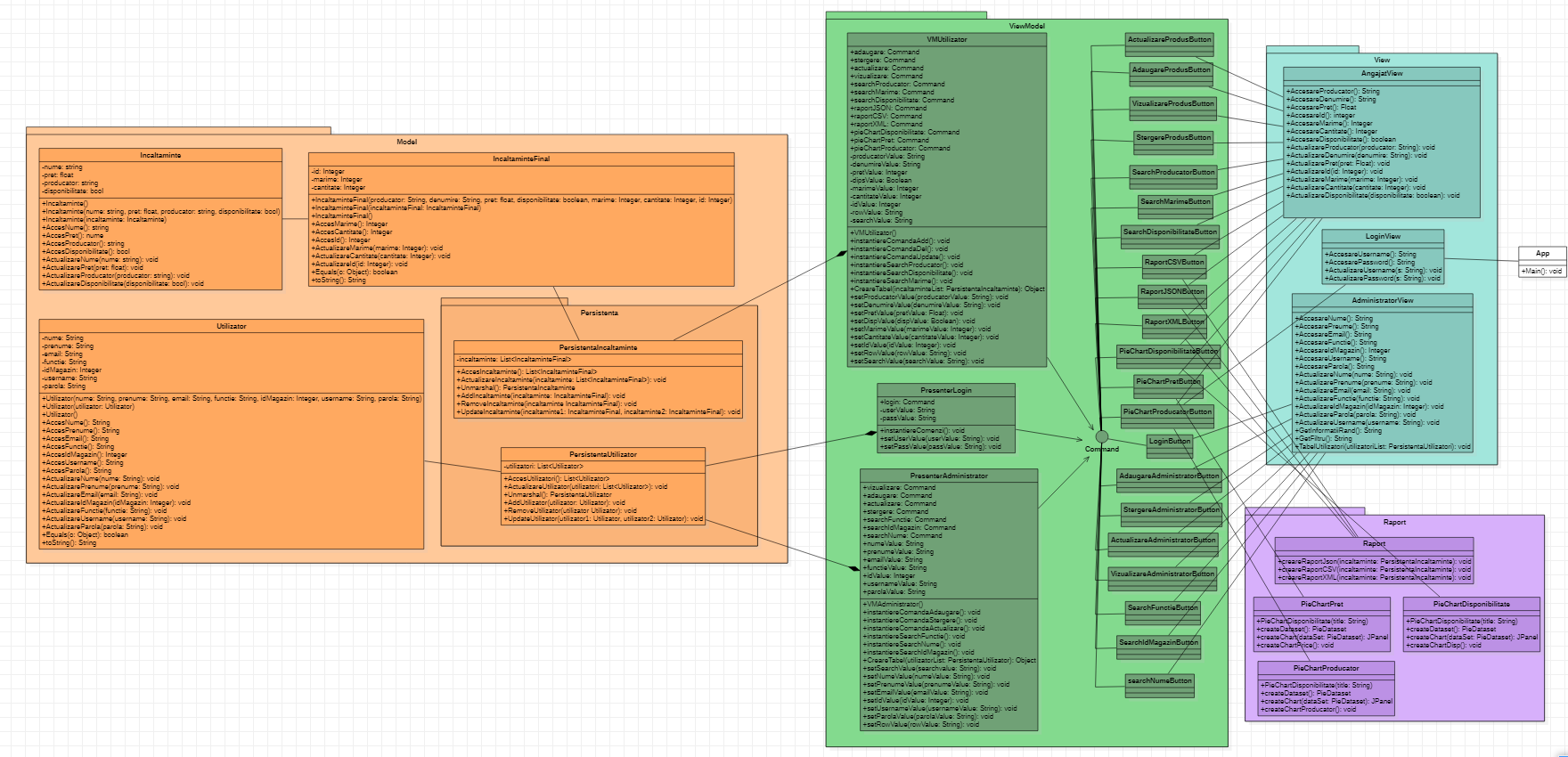
În diagrama de clase am reprezentat toate clasele implementate, pachetele ce le conțin pe acestea și legăturile dintre ele. În fiecare clasă am introdus atributele și metodele pe care le conține.

În pachetul ”Models” se afla pachetul ”Persistență” și clasele ”Incaltaminte”, ”IncaltaminteFinal” și ”Utilizator”. Clasa ”Incaltaminte” are ca atribute producător, denumire, preț și disponibilitate, iar ca metode are constructori, setter-e și getter-e. Clasa ”IncaltaminteFInal” moștenește clasa ”Incaltaminte” și are ca atribute id, mărime și cantitate, iar ca metode constructori, setter-e, getter-e, metoda ”equals” care verifică daca doua obiecte sunt egale si metoda „toString”. Clasa ”Utilizator” are ca atribute nume, prenume, email, funcție, idMagazin, username și parola, iar ca metode are constructori, setter-e, getter-e și o metodă ”equals”. Clasele ”PersistentaIncaltaminte” și ”PersistentaUtilizator” au ca atribute liste de încălțăminte si utilizatori, iar in acestea sunt implementate metodele pentru operațiile CRUD pe listele respective și preluarea sau scrierea datelor in baza de date.

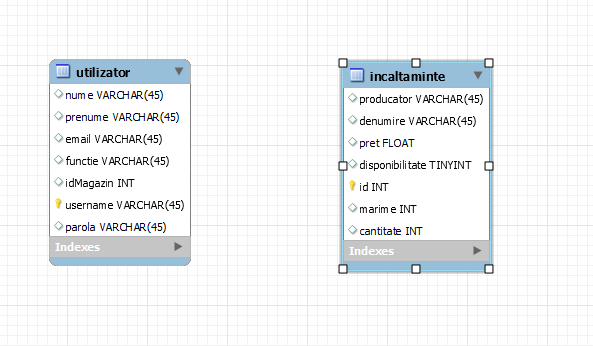
În pachetul ”View-Model” sunt clasele ”VMAdministrator”, ”VMLogin” și ”VMUtilizator” care fac legătura dintre pachetul ”Models” și pachetul ”GUI”. Tot in acest pachet este si interfața „Commands” si clasele in care s-a făcut logica pentru fiecare buton din interfața grafica.

In pachetul ”GUI” am implementat clasele de ”View”, care fac legătura între utilizator și aplicație.

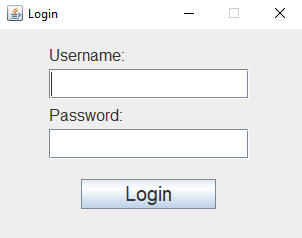
In pachetul „Raport” am implementat clasele necesare pentru generarea rapoartelor si a statisticile.



## Diagrama de relaționare a entității



# Descrierea aplicației

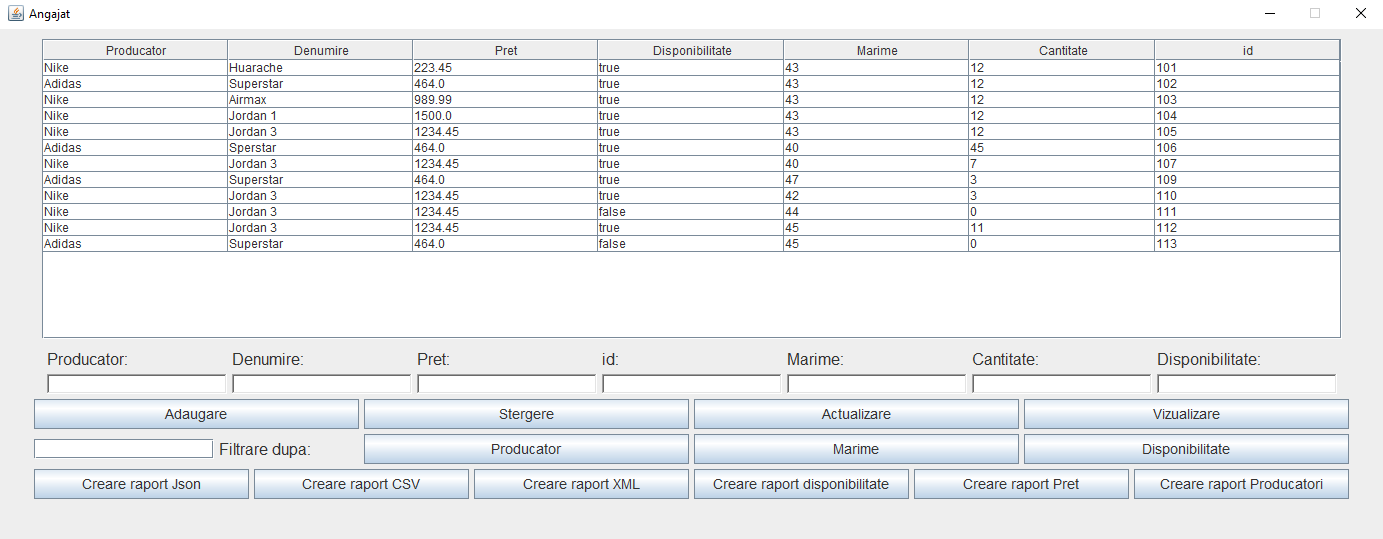


La pornirea programului suntem întâmpinați de pagina de ”Login” unde ne putem autentifica, iar în funcție de contul pe care îl introducem programul va deschide fie pagina de angajat, fie pagina de administrator.

O imagine care conține masă

Descriere generată automat

În pagina de ”Administrator” vedem tabelul cu toți utilizatorii aplicației, fie ca sunt administratori sau angajați. Aici putem adăuga utilizatori noi completând câmpurile cu toate datele acestora și apăsând butonul ”Adaugare”. Pentru ștergerea unui utilizator trebuie să selectăm, prin a da click, utilizatorul pe care dorim să îl ștergem și să apăsăm butonul ”Stergere”. Pentru a modifica datele despre un utilizator trebuie să selectam utilizatorul dorit, să completam câmpurile cu datele pe care dorim sa le modificăm și să apăsăm butonul ”Actualizare”. Pentru a vedea modificările aduse trebuie să apăsăm butonul ”Vizualizare” pentru a se reseta datele din tabel. In aceasta pagină mai putem filtra utilizatorii in funcție de nume, funcție si id-ul magazinul la care lucrează completând câmpul respectiv și apăsând unul din butoanele ”Nume”, ”Functie” sau ”IdMagazin”.



În pagina de ”Angajat” vedem tabelul cu toate produsele magazinelor din lanțul de magazine. Putem adauga, șterge, actualiza și vizualiza produsele prin același mod ca și la pagina de ”Administrator”. Filtrarea se face la fel doar ca se poate face după ”Producator”, ”Mărime” și ”Disponibilitate”. În plus față de pagina ”Administrator” sunt butoanele de creare rapoarte. Prin apăsarea butonul ”Creare raport Json” se va crea un raport Json, la apăsarea butonului ”Creare raport CSV” se va crea un raport CSV, care poate fi deschis și ca un fișier Excel, iar la apăsarea butonului „Creare raport XML” se va crea un raport XML cu produsele din tabel. La apăsarea butoanelor „Creare raport Disponibilitate”, „Creare raport Pret” si „Creare raport Producatori” se vor crea rapoartele cu statisticile respective.

# Credentiale pentru conectare

Administrator -> Username: „Antonio01”, Password: „parola1234”

Angajat -> Username: „Vlad05”, Password: „parola1237”