



Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

# Proiectare Software

## Aplicație desktop – Lanț de farmacii

Profesor îndrumător:

Iordan Anca

Data: 28.03.2022

Student:

Mălăescu Bianca

Grupa 30238



## Cuprins

1. Enunțul problemei.....	3
2. Instrumente utilizate.....	3
3. Justificarea limbajului de programare ales .....	4
4. Descrierea diagramelor UML.....	4
5. Descrierea aplicației .....	7
6. Credențiale pentru conectare.....	9



## 1. Enunțul problemei

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

- ❖ Vizualizarea tuturor produselor dintr-o farmacie selectată;
- ❖ Filtrarea produselor după următoarele criterii: disponibilitate, valabilitate, preț, producător;
- ❖ Căutarea unui produs după denumire;
- ❖ Operații CRUD în ceea ce privește persistența produselor;
- ❖ Salvare rapoarte/liste cu produse în mai multe formate: csv, json.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori.

## 2. Instrumente utilizate

Proiectul a fost realizat în IDE-ul IntelliJ de la JetBrains. Limbajul de programare pe care l-am utilizat este Java, iar pentru a modela partea de GUI am folosit Java Swing.

Persistența datelor din aplicație am realizat-o folosind JAXB (Java Architecture for XML Binding). Toate datele necesare sunt salvate în fișiere cu extensia .xml.

Aplicația este contruită utilizând șablonul arhitectural Model-View-Presenter. Principala caracteristică a acestui model este faptul că pachetele Model și View comunică doar prin intermediul Presenter-ului.



Pentru generarea rapoartelor de tip CSV am utilizat PrintWriter, iar pentru a genera rapoarte JSON am utilizat librăria Gson.

De asemenea, pentru a putea face marshal/a genera rapoarte pe baza obiectelor care au attribute de tip LocalDate, am contruit două LocalDate adaptare, deoarece metodele pe care le-am folosit pentru a aceste operații nu permiteau manipularea datelor de acest tip.

### 3. Justificarea limbajului de programare ales

Am ales limbajul de programare Java deoarece este cel mai popular limbaj de programare la momentul actual, cât și cel mai stabil și consider facilă utilizarea lui. Am fost familiarizată cu el din primul an de facultate și consider că exersarea dezvoltării proiectelor în Java îmi va fi foarte utilă pe viitor, în cariera de software developer,

Java este un limbaj foarte versatil, care poate fi utilizat pentru aplicații desktop cu interfață grafică, pentru aplicații de mobil cât și pentru aplicații și servere web.

Momentan, pe piața muncii există o deficiență de programatori în Java, astfel creșterea performanței mele de a programa în acest limbaj poate garanta o mulțime de oportunități în cariera mea.

### 4. Descrierea diagramelor UML

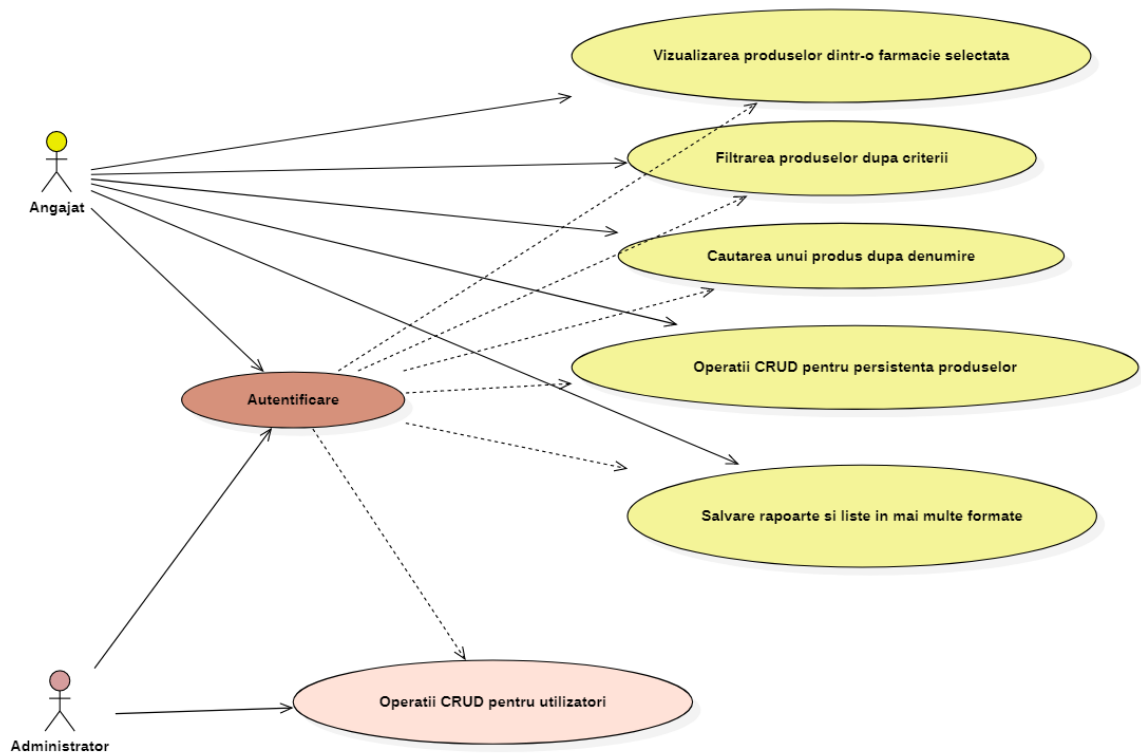
#### Diagrama cazurilor de utilizare

Diagrama cazurilor de utilizare conține 2 tipuri de actori: angajat (al unei farmacii) și administrator. În urma autentificării, fiecărui actor îi sunt permise diferite operații:

-Actorul de tip angajat poate să vizualizeze produsele dintr-o farmacie, să filtreze produsele după diferite criterii, să caute produse după denumire și să realizeze operații de tip CRUD pentru persistența produselor. De asemenea, ei pot să genereze salvarea unor rapoarte în două formate.



-Actorul de tip administrator poate să realizeze operații CRUD pentru persistența utilizatorilor.



## Diagrama de clase

Datorită modelului arhitectural folosit MVP, clasele sunt împărțite în 3 pachete mari:

Model, View și Presenter. Clasele din pachetul Model nu comunică cu clasele din View. Transferul datelor între aceste 2 pachete se face prin intermediul claselor din Presenter.

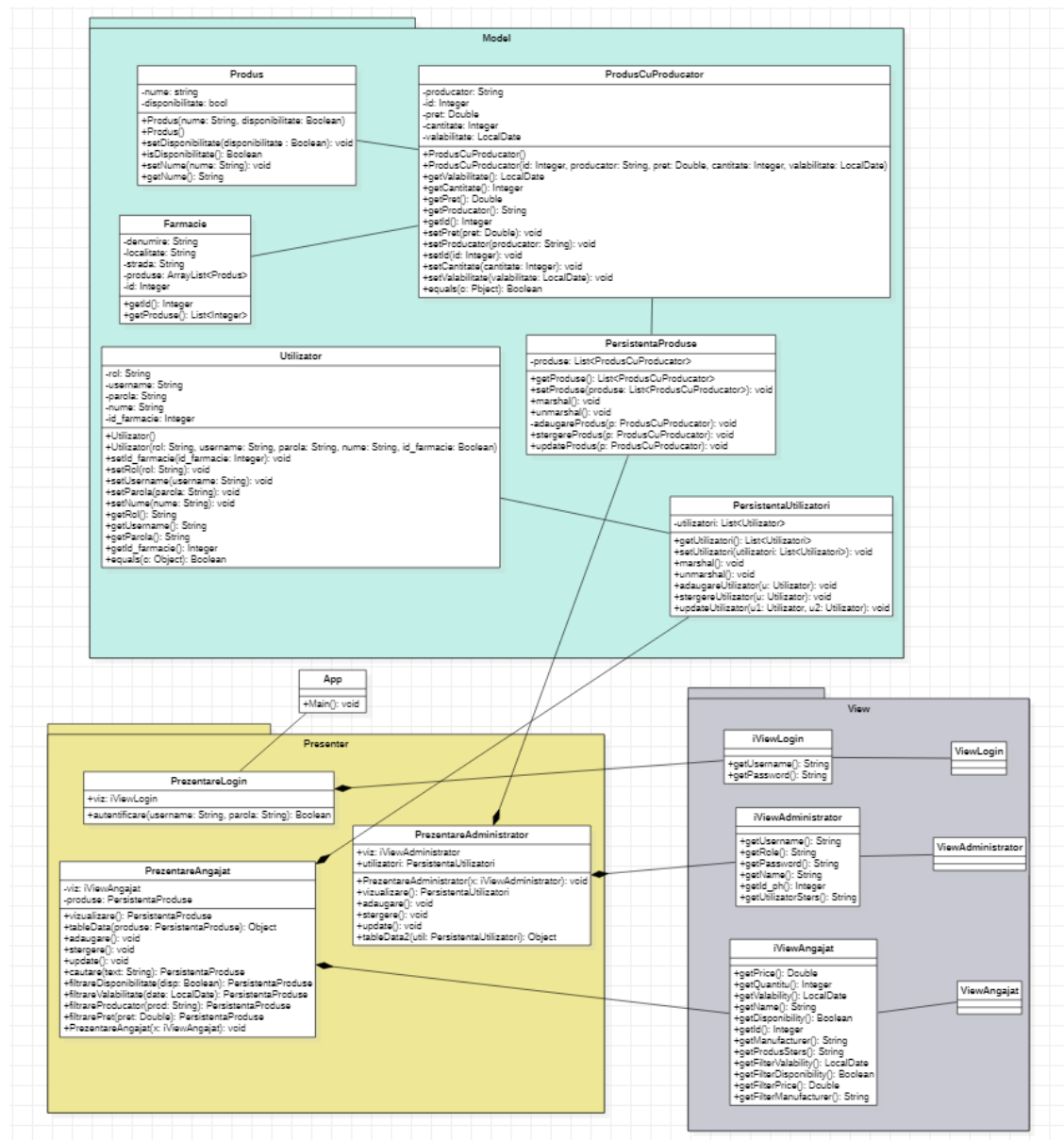
Produsele din farmacie sunt modelate în clasa *ProdusCuProducător*, care moștenește clasa *Produs*. Ca modele mai avem clasele *Farmacie* și *Utilizator*, cât și clasele de persistență *PersistențăProduse* și *PersistențăUtilizatori*, care asigură salvarea datelor.



Pachetul View conține 3 clase, câte una pentru fiecare fereastră a interfeței (login, angajat și administrator). Fiecare din cele 3 view-uri are o interfață corespunzătoare iView.

În Presenter am implementat 3 clase (PrezentareLogin, PrezentareAngajat și PrezentareAdministrator), câte una pentru fiecare view.

Clasa App conține doar metoda main care rulează aplicația, instanțiând un PresenterLogin.





## 5. Descrierea aplicației

Aplicația are 2 tipuri de utilizatori: angajat al unei farmacii și administrator. În urma logării, interfața și funcționalitățile permise de aplicație vor diferi.

La rularea aplicației, se va deschide fereastra de Login. Utilizatorul trebuie să-și introducă credențialele în textfield-urile corespunzătoare pentru a se conecta. În cazul în care acestea nu sunt corecte, se va afișa un mesaj corespunzător și utilizatorul poate reintroduce datele.

Pagina de login

Login

Username : administrator01

Parola : 123456

Logare

În cazul utilizatorilor de tip administrator, se va deschide o fereastră care afișează datele tuturor utilizatorilor aplicației ilustrate într-un tabel (în urma apăsării butonului “Afișare”) și butoane care asigură realizarea operațiilor de tip CRUD.

La selectarea unui utilizator din tabel și apăsarea butonului “Stergere”, utilizatorul selectat va fi șters. Pentru a adăuga un not utilizator, este necesară completarea datelor în câmpurile rezervate și apăsarea butonului “Adaugare”. Operația de update folosește aceleași câmpuri din



interfață, iar pentru a face update este necesară selectarea unei linii din table (corespunzătoare utilizatorului care va fi updatat) , completarea noilor date și apăsarea butonului “Update”.

Pentru a vizualiza rezultatele în urma operațiilor este necesară apăsarea butonului “Afișare” pentru a face refresh la date.

-- Tabel de produse -- **Afișare**

Rol	Username	Parola	Nume	Id_farmacie
administrator	Bianca15	12345	Malaescu Bianca	66
administrator	administrator_01	123456	Muresan Anca	11
angajat	Ang1726	1290	Pop Simona	17
angajat	angajat_01	1234	Malaescu Bianca	33
angajat	angajat1	1234	Angajat Ang	44

Stergere utilizator selectat din tabel: **Sterge**

Adaugare/update utilizator:

Rol:  Username:  Parola:  Nume:

Id farmacie:  **Adauga** **Update**

În cazul unui utilizator de tip angajat, fereastra deschisă va permite vizualizarea tabelului cu produse și efectuarea mai multor operații prin butoanele corespunzătoare. Acestea cuprind: căutarea unui produs după denumire, filtrarea produselor după diferite câmpuri: disponibilitate, valabilitate, preț și producător, generarea unor rapoarte de produse în format csv și în format json cât și operații de tip CRUD.

Operațiile de tip crud se realizează efectuând aceiași pași ca cele din fereastra pentru administrator. Pentru a căuta produse după denumire se folosește textfield-ul din stânga butonului “Cautare” și se apasă butonul. Filtrările au textfield comun, iar criteriul se alege prin apăsarea unuia dintre butoanele “Disponibilitate”, “Producător”, “Valabilitate” sau “Pret”.





Pentru a vizualiza rezultatele în urma operațiilor este necesară apăsarea butonului “Afișare” pentru a face refresh la date.

Butonul din partea de jos a interfeței generează rapoarte ale produselor în format csv și în format json.

Lant de farmacii

-- Tabel de produse -- Afișare

Nume	Disponibilitate	Id	Producator	Pret	Cantitate	Valabilitate
Paracetamol	false	10	MedPharma	19.28	10	2023-05-06
Algocalmin	true	51	PharmaRO	12.99	50	2025-01-01
Paracetamol	true	12	MedSan	17.0	15	2024-03-04
Tusin	true	167	ProdPh	18.0	12	2023-10-09
Test Covid	true	1	MedPharma	11.99	100	2025-03-04
Test Covid	false	16	ProdPh	15.99	0	2022-05-05

Cautare dupa denumire:  Cautare Stergere

Filtrarea in functie de: Disponibilitate Valabilitate Pret Producator

Adaugare produs/update:

Nume:  Disponibilitate:  Id:  Producator:

Pret:  Cantitate:  Valabilitate:  Adauga Update

Generare rapoarte

## 6. Credențiale pentru conectare

-Administrator: “administrator\_01” , “123456”

-Angajat: “angajat\_01” , “1234”