PROIECT - INGINERIA SISTEMELOR SOFTWARE

Universitatea Babes-Bolyai,Facultatea de matematica si informatica

AN UNIVERSITAR 2022-2023

SEMSTRUL 2

Aplicatie

Agenti de vanzari

**Nume si Prenume Student Coordonator:**

Nedelcu Gabriel-Alexandru Prof. Sima Ioan

Grupa 225/1

Sectia Informatica

# Prezentarea cerintei

Aplicatia dezvoltata , aduce in prim-plan un sistem de iteractiune intre agentii de vanzari si clienti.

O firma are mai multi agenti de vanzari, care fac comenzi pentru clientii firmei si le distribuie acestora produsele. Pentru desfasurarea activitatilor, firma ofera o aplicatie prin care agentii pot comanda produse.

Aplicatia pune la dispozitia fiecarui agent al firmei un terminal prin care:

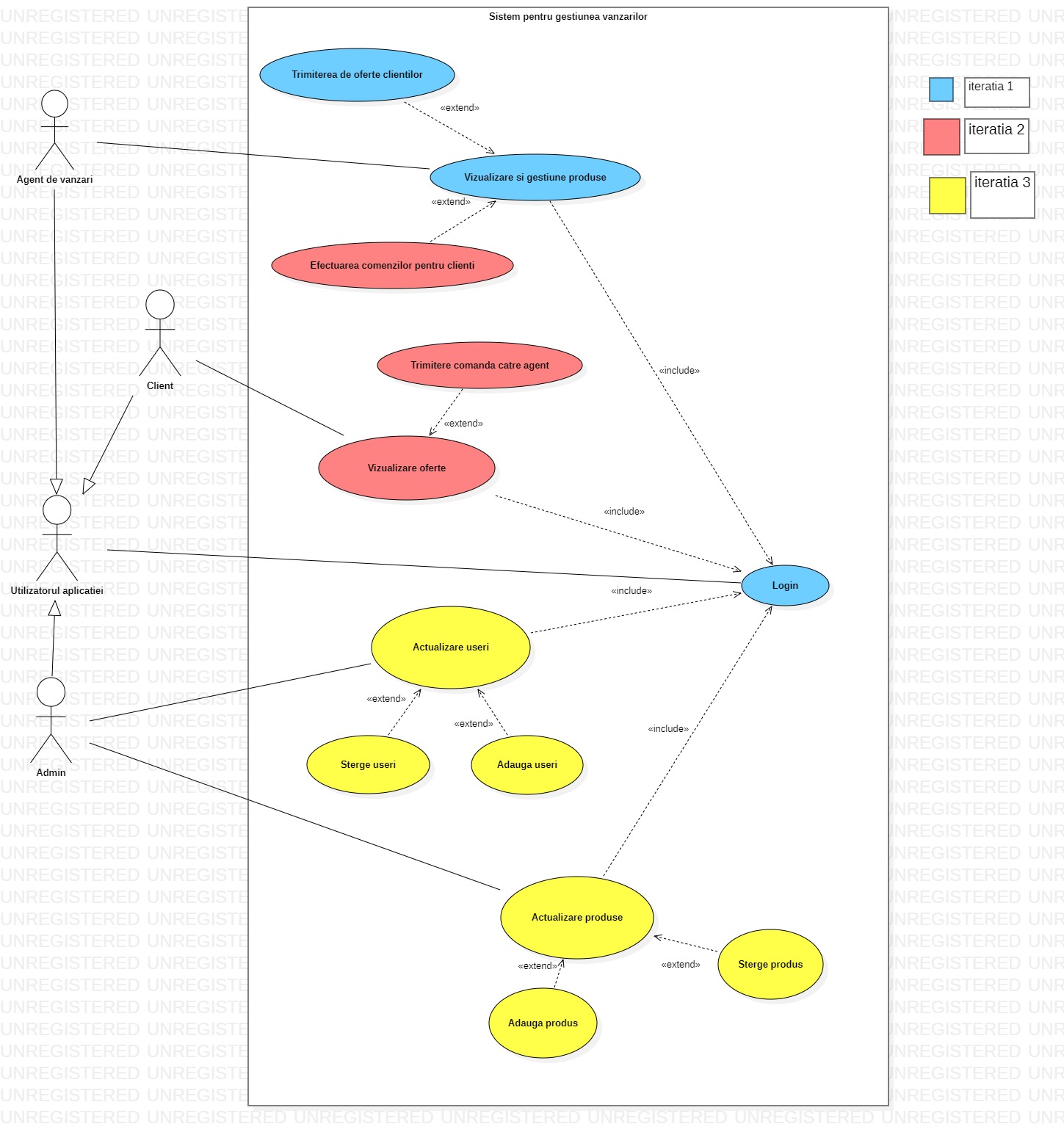
• agentul vizualizeaza lista tuturor produselor vândute de firma, împreuna cu preturile aferente și cantitatile existente pe stoc

• agentul poate face o oferta pentru clienti pentru a ii instiinta de produse

• clientul poate sa aleaga o oferta si sa scrie un mesaj pentru agent , in care trebuie sa treaca si cantitatile dorite , si sa trimita o oferta pentru client

• agentul poate comanda o cantitate din produsele din comanda clientului. După orice comanda valida, toti agentii logati în aplicație vor vedea lista actualizata a stocurilor (este posibil ca declansarea înregistrarii unei comenzi din partea unui agent sa conduca la un mesaj informativ "cantitate insuficienta în stoc").

DIAGRAMA CAZURILOR DE UTILIZARE



|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-1: Log in | | |
| Primary actor | Agent de vanzari/Client/Admin | Secondary actors |  |
| Description | Un utilizator se logheaza in aplicatie. | | |
| Trigger | Un utilizator vrea sa foloseasca aplicatia. | | |
| Preconditions | PRE-1. Utilizatorul este agent de vanzari, client sau admin | | |
| Postconditions | POST-1. Utilizatorul foloseste aplicatia | | |
| Normal flow | 1.Utilizatorul deschide aplicatia  2.Sistemul afiseaza 3 butoane dintre care clientul sa aleaga daca doreste sa se logheze drept client, agent de vanzari sau admin  3.Acesta selecteaza agent de vanzari, client sau admin  4.Sistemul ii cere sa isi completeze datele pentru logare  5.Utilizatorul isi completeaza datele si alege submit(E1)  6.Sistemul inchide fereastra si deschide alta | | |
| Alternative flows | 5.1 Utilizatorul este agent de vanzari.  1.Sistemul il afiseaza optiunile trimitere oferte si efectuare comezi.  2.Agentul alege una dintre ele.  3.Sistemul deschide o fereastra noua.  5.2 Utilizatorul este admin.  1.Sistemul ii afiseaza optiunile actualizare clienti si actualizare produse.  2.Adminul alege o optiune.  3.Sistemul deschide o fereastra noua. | | |
| Exceptions | E1. Datele introduse de utilizator sunt gresite  1.Sistemul il anunta ca a introdus datele gresite.  2.Utilizatorul reintroduce corect datele.  3.Inoarcere la punctul 5 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-2: Efectuarea comenzilor pentru clienti | | |
| Primary actor | Agent de vanzari | Secondary actors |  |
| Description | Un agent de vanzari utilizeaza aplicatia pusa la dispozitie de firma pentru a face o comanda pentru un client. Sistemul afiseaza o lista cu clienti, iar pentru fiecare client lista cu produsele de care este interesat si cantitatea pentru fiecare plus o lista cu comenzile deja trimise si cu statusul acestora. | | |
| Trigger | Un agent de vanzari doreste sa faca o comanda pentru client | | |
| Preconditions | PRE-1. Agentul este logat in aplicatie | | |
| Postconditions | POST-1. Comanda este plasata | | |
| Normal flow | 1.Agentul selecteaza o comanda pe care doreste sa o efectueze si cantitatea pentru fiecare produs (E1)  2. Sistemul plaseaza(trimite datele catre baza de date si afiseaza un mesaj corespunzator) comanda. | | |
| Exceptions | E1. Nu mai exista atat de multe exemplare dintr-un produs  1.Sistemul anunta agentul ca nu mai exista exemplare suficiente din produsul respectiv  2a.Agentul trimite un mesaj clientului pentru a reface comanda si nu o mai plaseaza  3.Sistemul trimite mesajul clientului  4.Intoarcere la punctul 1 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-3: Trimiterea de oferte clientilor | | |
| Primary actor | Agent de vanzari | Secondary actors |  |
| Description | Un agent de vanzari utilizeaza aplicatia pusa la dispozitie de firma pentru a identifica potentialii clienti si a le trimite o liste cu produse de care e posibil sa fie interesati. Sistemul afiseaza o lista cu clienti, iar pentru fiecare client diverse informatii si o lista cu produsele disponibile | | |
| Trigger | Un agent de vanzari vrea sa trimita oferte clientilor | | |
| Preconditions | PRE-1. Agentul este logat in aplicatie | | |
| Postconditions | POST-1. Lista cu produse este trimisa clientului | | |
| Normal flow | 1.Agentul selecteaza unul sau mai multi clienti si unul sau mai multe produse si apasa butonul trimite pentru a trimite clientilor lista de produse.(E1)  2.Sistemul efectueaza trimiterea datelor si afiseaza un mesaj corespunzator. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-4: Trimitere oferta catre agent | | |
| Primary actor | Client | Secondary actors |  |
| Description | Clientul utilizeaza aplicatie pentru a trimite agentului comanda sa si a-i plasa comanda. Sistemul afiseaza o lista cu oferte, una cu comenzile trimise si statusul lor (daca o comanda este respinsa la status este trecut un mesaj de la agentul de vanzari cu motivul respingerii) | | |
| Trigger | Un client doreste sa trimita o comanda unui agent de vanzari | | |
| Preconditions | PRE-1. Clientul este logat in aplicatie | | |
| Postconditions | POST-1. Comanda este trimisa agentului | | |
| Normal flow | 6.Sistemul afiseaza o lista cu produsele din fiecare oferta.  7.Clientul selecteaza o oferta si scrie un mesaj cu cantitatile dorite si apasa trimite oferta.  8.Sistemul trimite comanda agentului. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-5: Sterge produse | | |
| Primary actor | Admin | Secondary actors |  |
| Description | Adminul utilizeaza aplicatia pentru a modifica produsele. Sistemul ii afiseaza o lista cu produse | | |
| Trigger | Un admin vrea sa stearga produse. | | |
| Preconditions | PRE-1. Utilizatorul este admin | | |
| Postconditions | POST-1. Lista de produse este modificata. | | |
| Normal flow | 1.Adminul selecteaza produsele si apasa delete.  2.Sistemul afiseaza lista actualizata. | | |
| Alternative flows | 1.1 Adminul trece datele noului produs si cantitatea in niste textbox-uri si apasa add. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-6: Adaugare produse | | |
| Primary actor | Admin | Secondary actors |  |
| Description | Adminul utilizeaza aplicatia pentru a modifica produsele. Sistemul ii afiseaza o lista cu produse | | |
| Trigger | Un admin vrea sa adauge produse. | | |
| Preconditions | PRE-1. Utilizatorul este admin | | |
| Postconditions | POST-1. Lista de produse este modificata. | | |
| Normal flow | 1. Adminul trece datele noului produs si cantitatea in niste textbox-uri si apasa add.  2.Sistemul afiseaza lista actualizata. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-7: Stergere useri | | |
| Primary actor | Admin | Secondary actors |  |
| Description | Adminul utilizeaza aplicatia pentru a modifica userii. Sistemul ii afiseaza o lista cu useri | | |
| Trigger | Un admin vrea sa stearga useri. | | |
| Preconditions | PRE-1. Utilizatorul este admin | | |
| Postconditions | POST-1. Lista de useri este modificata. | | |
| Normal flow | 1.Adminul selecteaza userii si apasa delete.  2.Sistemul afiseaza lista actualizata. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | CU-8: Adaugare useri | | |
| Primary actor | Admin | Secondary actors |  |
| Description | Adminul utilizeaza aplicatia pentru a modifica userii. Sistemul ii afiseaza o lista cu useri | | |
| Trigger | Un admin vrea sa adauge useri. | | |
| Preconditions | PRE-1. Utilizatorul este admin | | |
| Postconditions | POST-1. Lista de useri este modificata. | | |
| Normal flow | 1.Adminul trece datele noului user in niste textbox-uri si apasa add.  2.Sistemul afiseaza lista actualizata. | | |

DIAGRAMA DE CLASE A MODELULUI

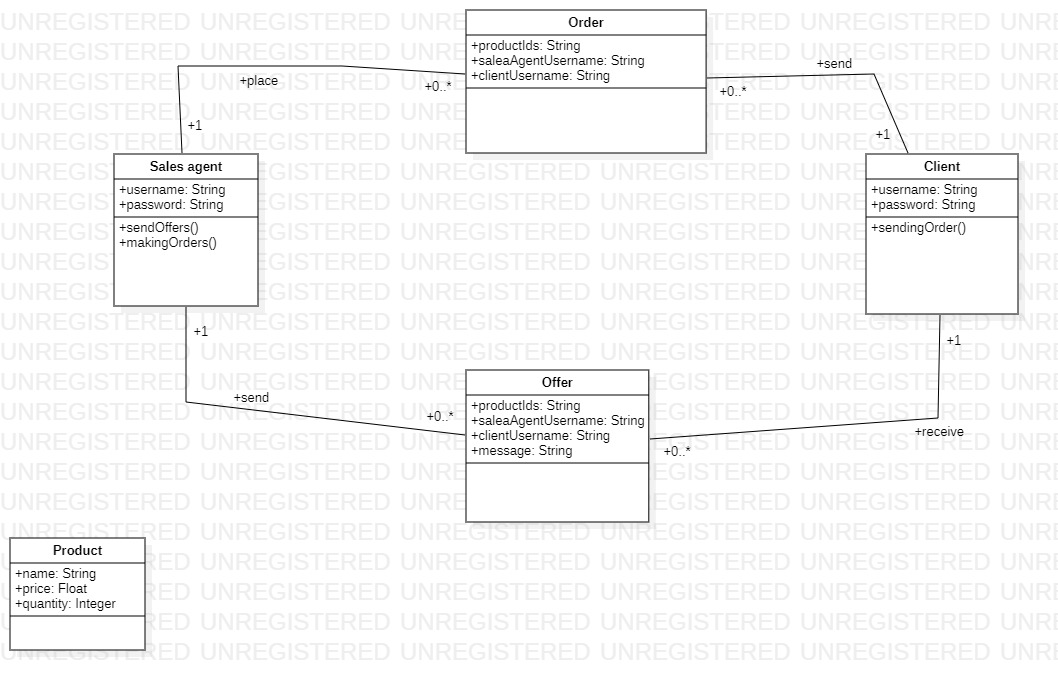
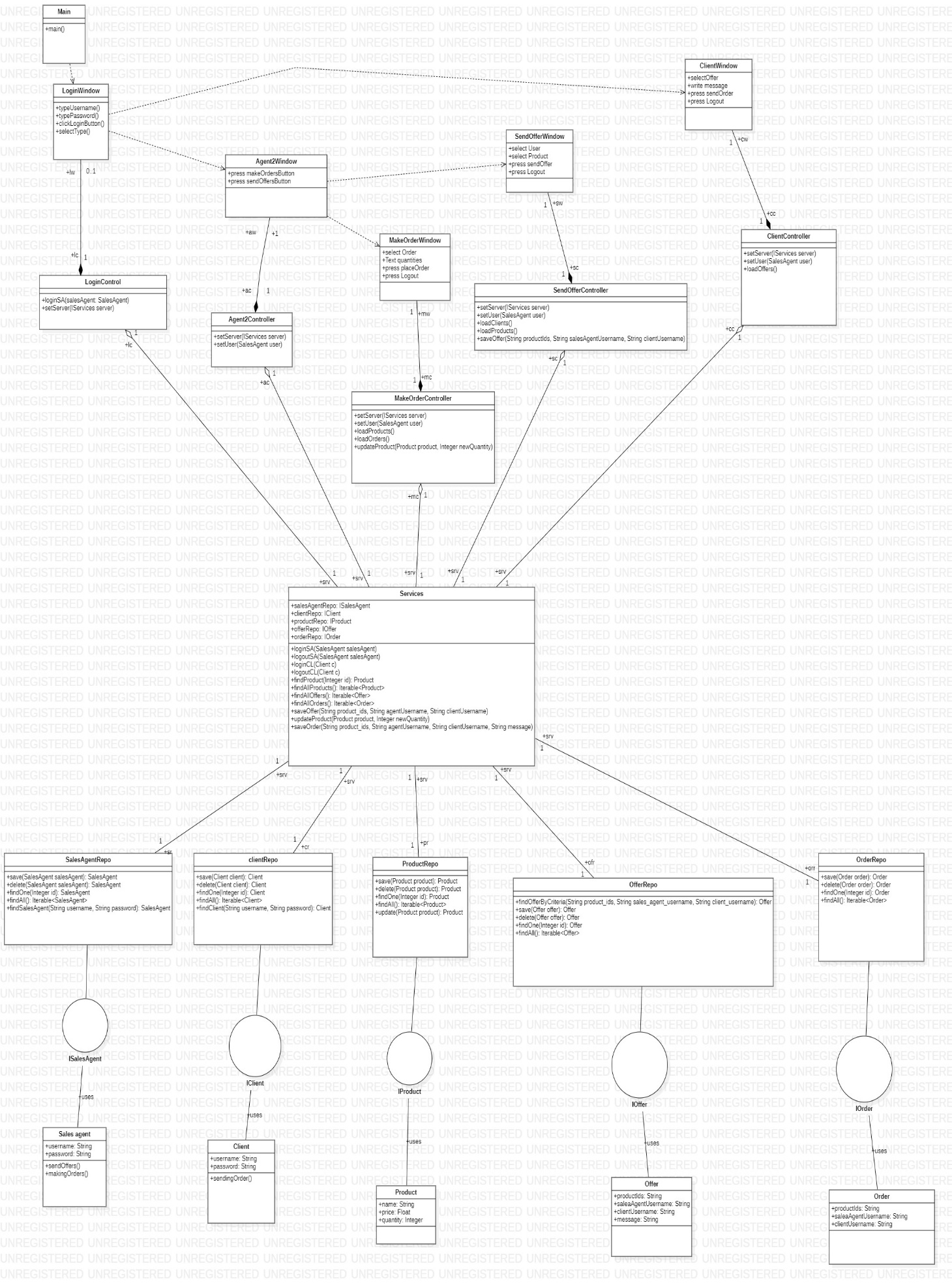


DIAGRAMA DE CLASE FINISATA



# Diagrama bazei de date

# 

# Diagrame de secventa

# 

# 

# 

# Diagrame de comunicare

# 

# 

# Documentatie tehnica:

## Baza de date:

Am ales SQLite ca baza de date din mai multe motive. SQLite este un sistem de gestionare a bazelor de date relaționale care este încorporat în sistemul de operare. Acesta nu necesită un proces server separat, ceea ce simplifică dezvoltarea și testarea. În plus, SQLite este foarte ușor de utilizat și de configurat, nu necesită instalare sau administrare separată.

## Mediul de dezvoltare:

Am optat pentru IntelliJ IDEA datorită eficienței sale. Acest IDE suportă o multitudine de limbaje de programare, oferind funcționalități precum completare automată a codului, analiza codului în timp real și o multitudine de plugin-uri. Tocmai datorită acestor funcționalități, dezvoltarea aplicației a fost mai rapidă și mai eficientă.

## Limbaj de programare:

Am ales Java ca limbaj de programare datorită popularității sale și a comunității vaste de dezvoltatori. Java oferă un ecosistem extins de biblioteci și framework-uri, facilitând dezvoltarea aplicației. În plus, Java este un limbaj de programare orientat-obiect, ceea ce a făcut ca structurarea și organizarea codului să fie mai ușoară.

## ORM:

Am folosit Hibernate ca soluție ORM. Hibernate este un framework robust și matur, care ne-a permis să manipulăm baza de date într-un mod orientat-obiect, eliminând necesitatea de a scrie mult SQL manual. Acesta a facilitat crearea, citirea, actualizarea și ștergerea înregistrărilor din baza de date.

## Proiectarea diagramelor:

## Am folosit StarUML pentru a proiecta diagramele UML. StarUML este un instrument puternic și flexibil care suportă o varietate de diagrame UML, precum clase, secvențe, cazuri de utilizare etc. Acesta ne-a permis să proiectăm și să vizualizăm structura aplicației noastre într-un mod vizual și intuitiv.

## GUI:

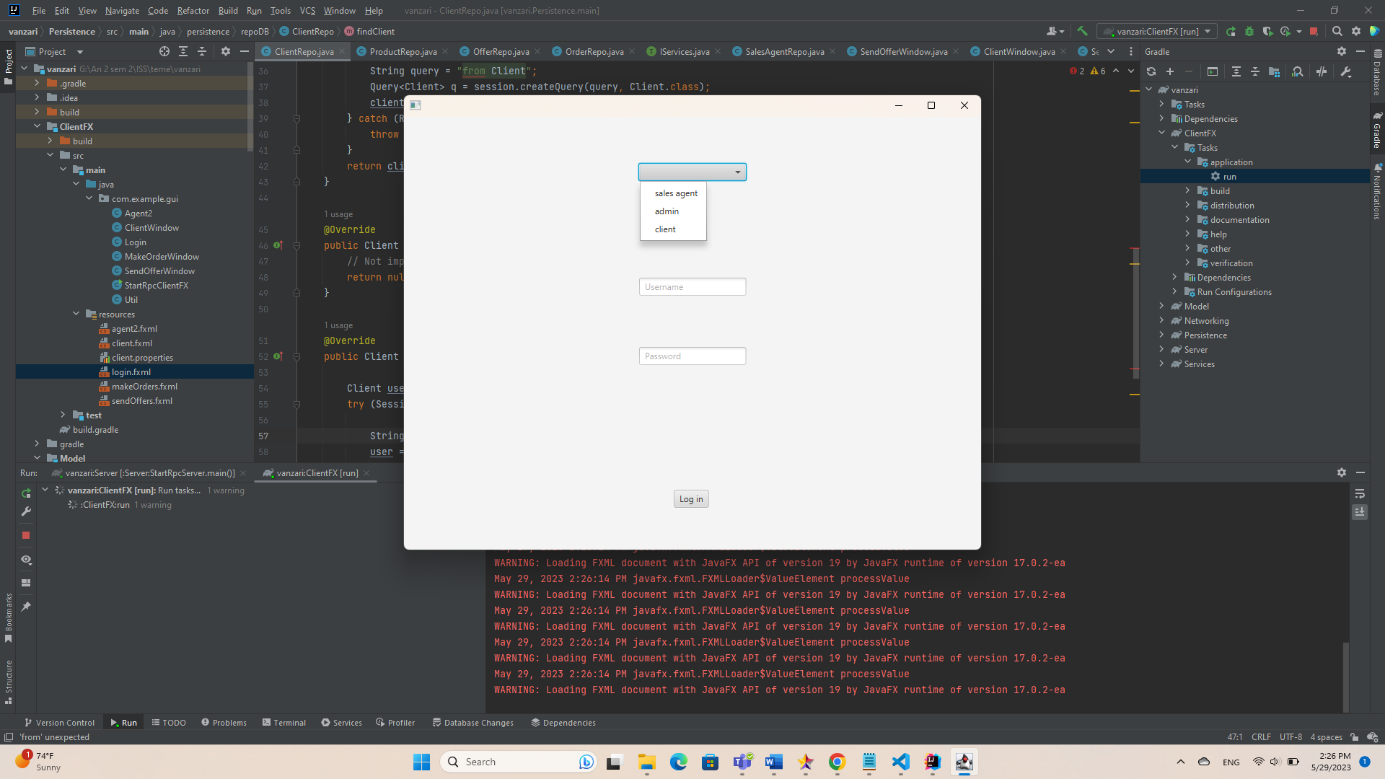
Am utilizat JavaFX pentru construcția interfeței grafice. JavaFX este un set de biblioteci grafice și media care oferă o platformă unificată pentru dezvoltarea aplicațiilor client. Am ales JavaFX datorită faptului că acesta permite o integrare ușoară cu limbaje de stilizare precum CSS, făcându-ne mai ușor să oferim o experiență de utilizare plăcută.

## Comunicare Client-Server:

Am adoptat o arhitectură client-server pentru proiectul nostru. Acest model de comunicare permite distribuirea sarcinilor între furnizorii de resurse sau servicii, numiți servere, și cei care solicită aceste resurse sau servicii, numiți clienți. Această arhitectură a făcut ca aplicația noastră să fie mai ușor de gestionat, de întreținut și de scalat.

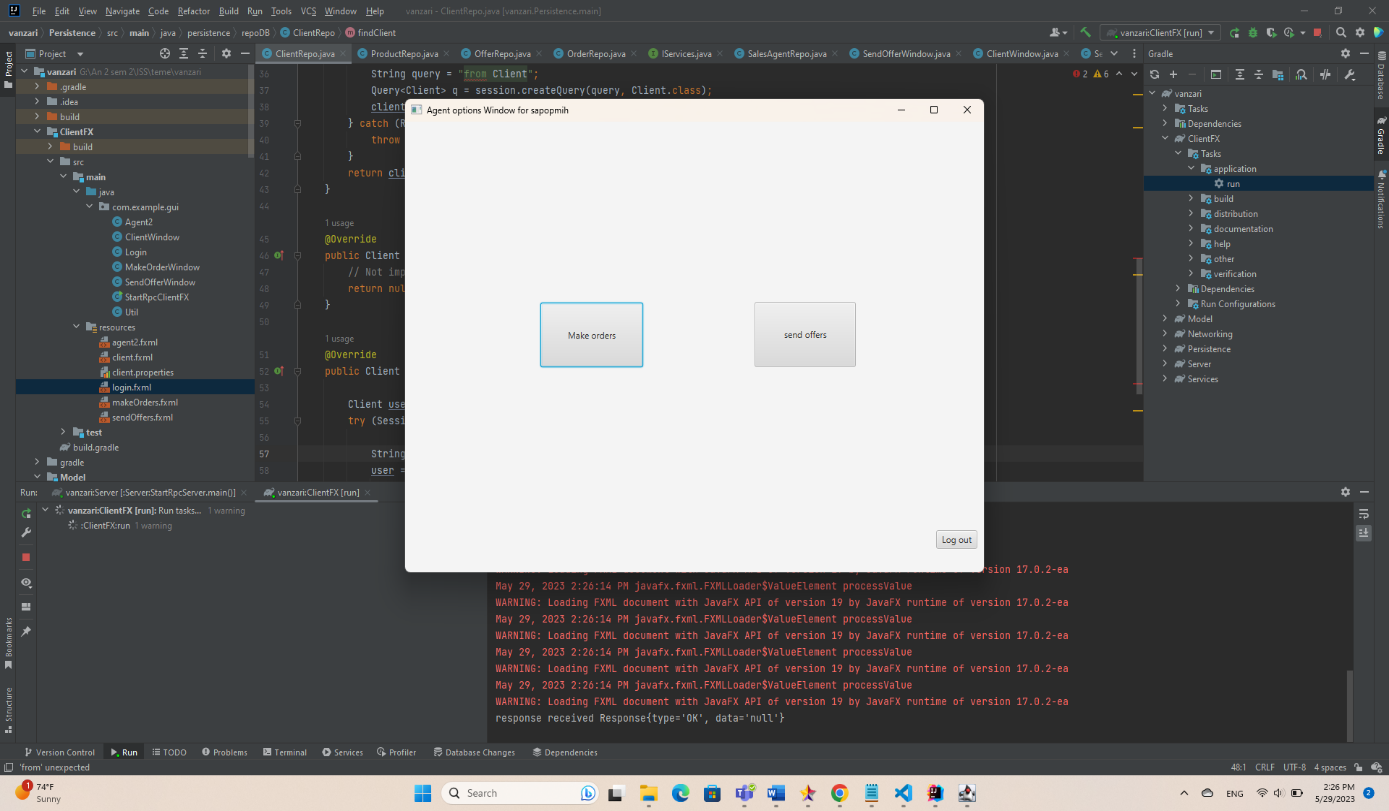
Tutorial

Interfata pentru log in



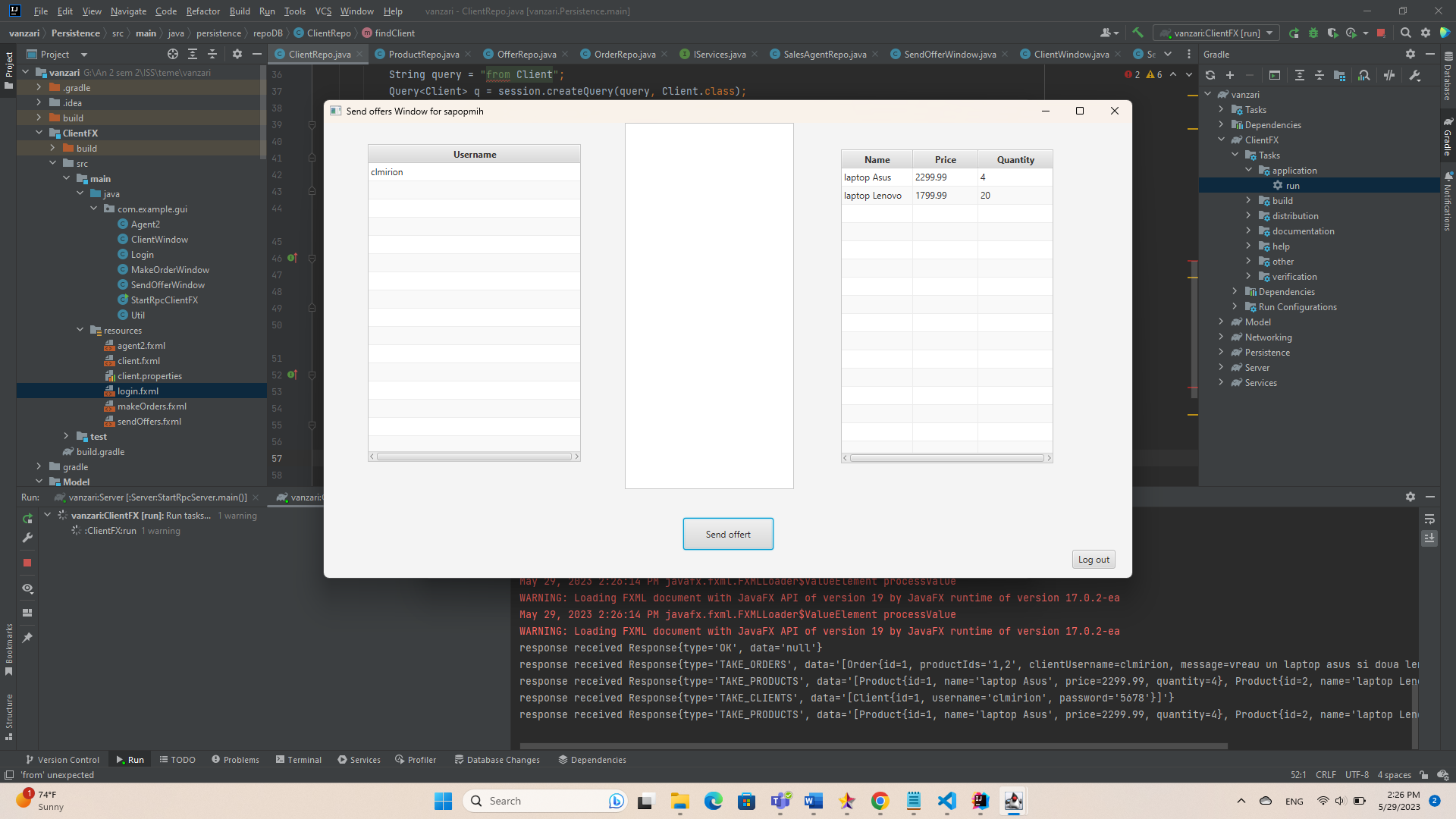
Alegem tipul utilizatorului pentru a ne conecta. Introducem usernameul si parola , apoi apasam butonul Log in.

Interfata pentru alegerea meniului agentului



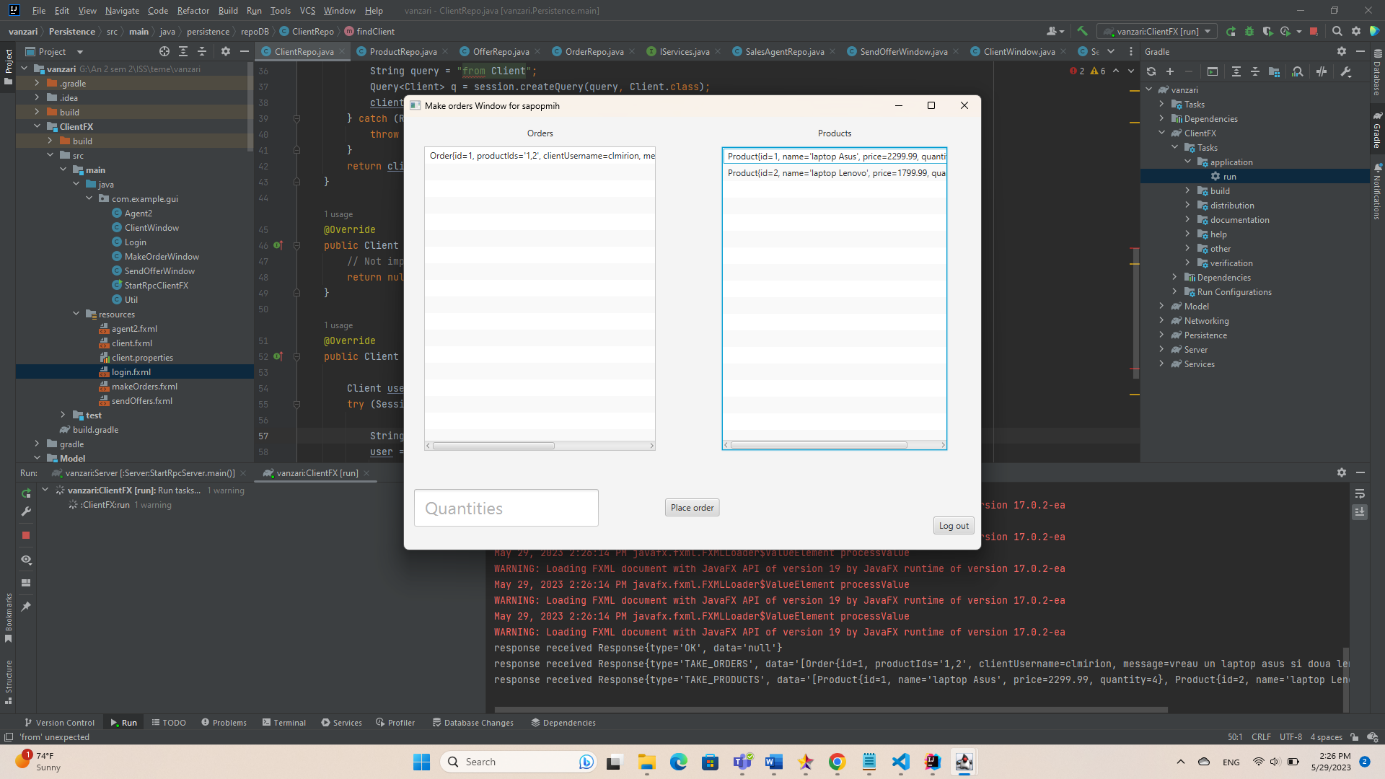
Daca ne-am logat ca agent de vanzari: Alegem in ce meniu dorim sa intram si apasam butonul Make orders sau Send Offers.

Interfata pentru trimiterea ofertelor catre agenti:



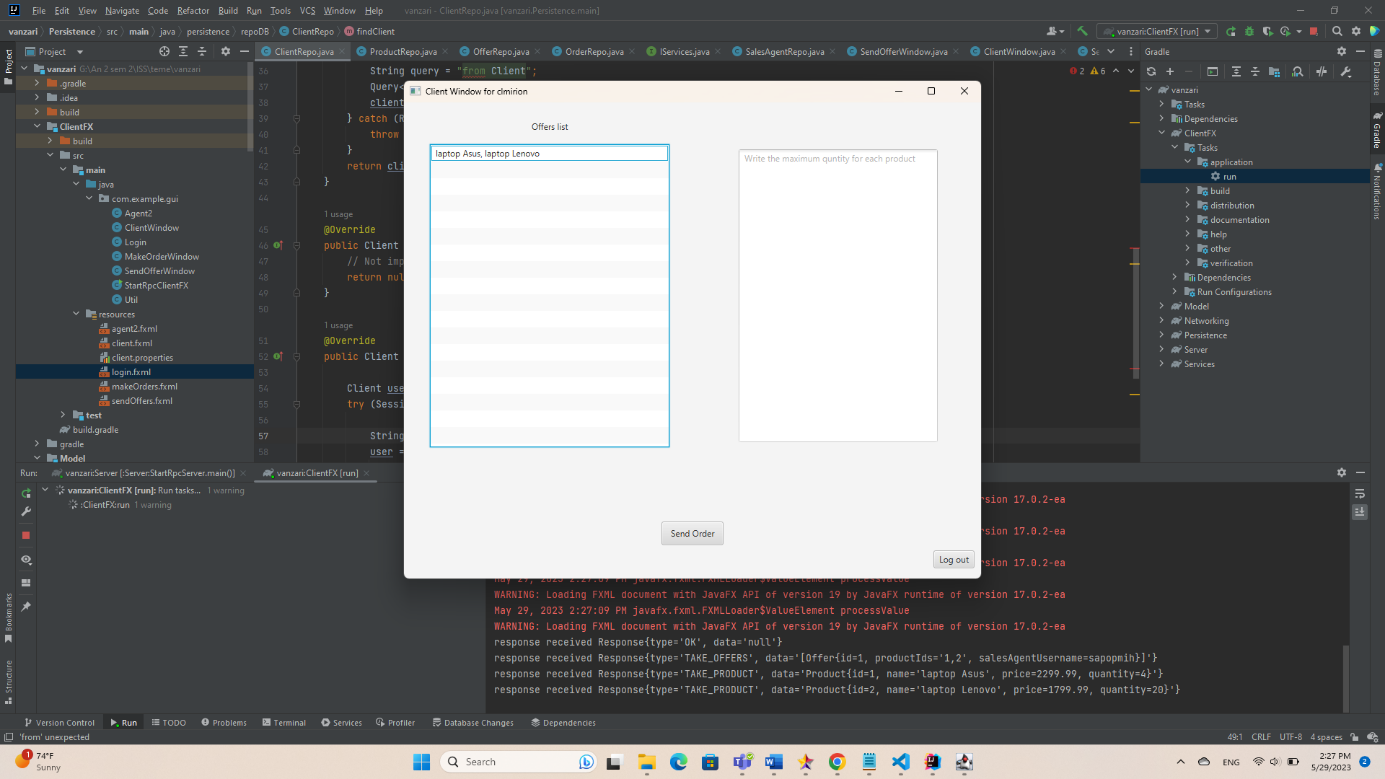
Daca am selectat Send offers: Putem selecta un utilizator din tabelul din stanga, si mai multe produse din tabelul din dreapta care vor fi adaugate in lista din mijloc(apasand pe un produs din lista acesta o sa fie deselectat). Apasand Send offert , oferta o sa fie trimisa , iar daca apasam Log out inchidem aceasta fereastra.

Interfata pentru efectuarea comenzilor pentru clienti:



Daca am ales Make orders : Alegem o oferta din lista din stanga si introducem in textfieldul quantities cantitatile dorite in formatul 1,2,3,4,5,.... in ordinea in care sunt trecute produsele in oferta selectata. Apasand butonul Place offer , oferta este plasata, iar apasand butonul Log out iesim din aceasta fereastra. In lista din dreapta putem vedea produsele si cantitatile lor din stoc, care se modifica odata cu trimiterea comenzilor.

Interfata pentru trimiterea comenzilor catre clienti:



Daca ne-am logat drept client: Alegem o oferta din lista din stanga si introducem un mesaj catre agent in dreapta care trebuie sa contina obligatoriu cantitatile dorite din fiecare produs. Apasand butonul Send Order trimitem comanda, iar stocurile scad.

Referinte:

<https://hibernate.org/orm/contribute/>

<https://docs.oracle.com/javafx/2/>

<https://www.youtube.com/watch?v=9XJicRt_FaI&t=9526s&ab_channel=BroCode> - javafx

Curs 2,3 mpp 2023 – Gradle

Curs 5,6 mpp 2023 - Aplicații client-server;Șablonul Proxy;Networking și threading în Java

Curs 9 mpp 2023 – Hibernate

<https://www.youtube.com/watch?v=tUVakYv7PJ8>

<https://www.youtube.com/watch?v=ccBbwNR5lwc>

<https://www.youtube.com/watch?v=yjzDOAzUecw>

<https://www.youtube.com/watch?v=SNmW68mNSbw>

<https://www.youtube.com/watch?v=AsZRRzRbnCg>