

## 6. Documentatia tehnica

### Cerinte non-functionale:

- Aplicatia trebuie sa fie una desktop
- Aplicatia trebuie sa aiba un timp de raspuns mic pentru a permite utilizatorului sa interactioneze rapid cu aceasta
- Aplicatia trebuie sa utilizeze un ORM pentru interactiunea cu baza de date indiferent de SGBD-ul utilizat
- Interfata grafica intuitiva si prietenoasa pentru a facilita utilizarea aplicatiei de catre utilizatori

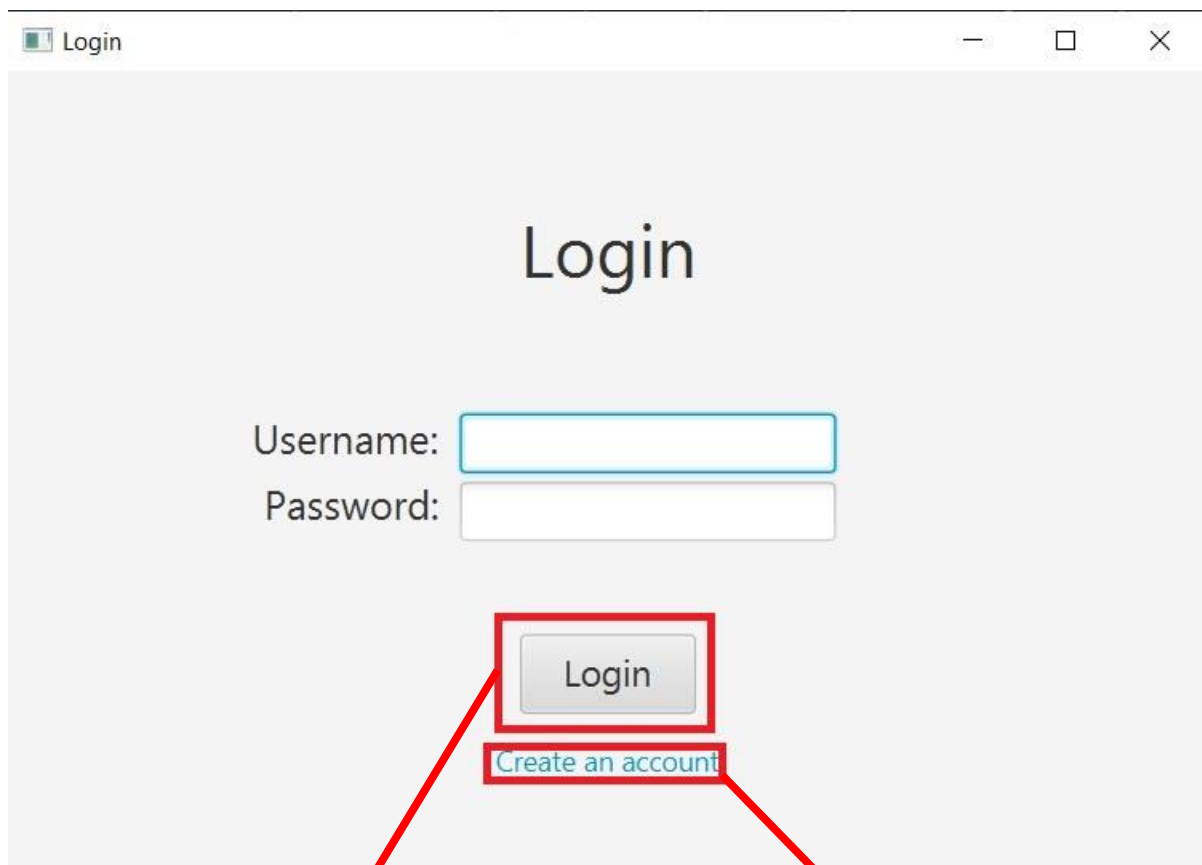
### Framework-uri si tool-uri:

- Ca limbaj de programare am ales **Java**, deoarece este un limbaj popular cu care sunt familiarizat, independent de platforma de operare si cu o comunitate de utilizatori existinsa si care, de asemenea, ofera o multitudine de framework-uri.
- Pentru gestiunea dependentelor proiectului si utilizarea librariilor si framework-urilor externe am utilizat **Gradle**.
- Am utilizat **Java Spring** framework pentru configurarea aplicatiei, deoarece ofera un model de programare simplu si coerent in gestiunea dependentelor, ceea ce duce la un cod mai clar si mai usor de intretinut.
- Persistenta datelor am asigurat-o folosind **PostgreSQL**, un sistem de gestiune a bazelor de date robust si performant ce ofera un acces rapid si sigur la datele stocate.

- Ca ORM am ales sa utilizez **Hibernate**, deoarece este foarte popular si usor de configurat, putand fi folosit cu o multitudine de sisteme de gestiune a bazelor de date si independent de cel ales.
- Pentru GUI am folosti **JavaFX** pentru ca este un framework cu care am mai lucrat anterior si pentru ca ofera o usurinta in dezvoltarea aplicatiilor desktop. De asemenea, in combinatie cu JavaFX am folosit si **Scene Builder** pentru posibilitatea crearii de interfete grafice drag and drop.
- Pentru controlul versiunilor si salvarea progresului am folosit **Git** impreuna cu **GitHub**.
- Am folosit **StarUML** la proiectarea diagramelor pentru multitudine de avantaje ale programului si usurinta organizarii.
- Ca mediu de dezvoltare (IDE) am ales sa folosesc **IntelliJ Ultimate**, deoarece este foarte versatil si ofera o multitudine de tool-uri si functionalitati precum gestionarea directa a bazei de date, debugging avansat, generare de diagrame, VSC etc., ce sporesc productivitatea.

## 7. Help

La deschiderea aplicatiei se afiseaza fereastra de login:



The image shows a standard login window with a title bar containing the text 'Login' and standard window control buttons (minimize, maximize, close). The main content area has a large 'Login' title in the center. Below the title, there are two input fields: 'Username:' followed by a text box, and 'Password:' followed by a text box. At the bottom of the form, there are two buttons: a 'Login' button and a 'Create an account' button. Red boxes highlight the 'Login' button and the 'Create an account' button. Red lines connect these buttons to explanatory text boxes below the window.

Dupa introducerea datelor pentru autentificare, la apasarea butonului se va deschide fereastra principala a aplicatiei in functie de tipul utilizatorului

In cazul in care utilizatorul nu are un cont existent acesta isi poate crea unul din fereastra de register ce se deschide la apasarea butonului

Fereastra de register:

Register

## Sign in

First name:

Last name:

Username:

Password:

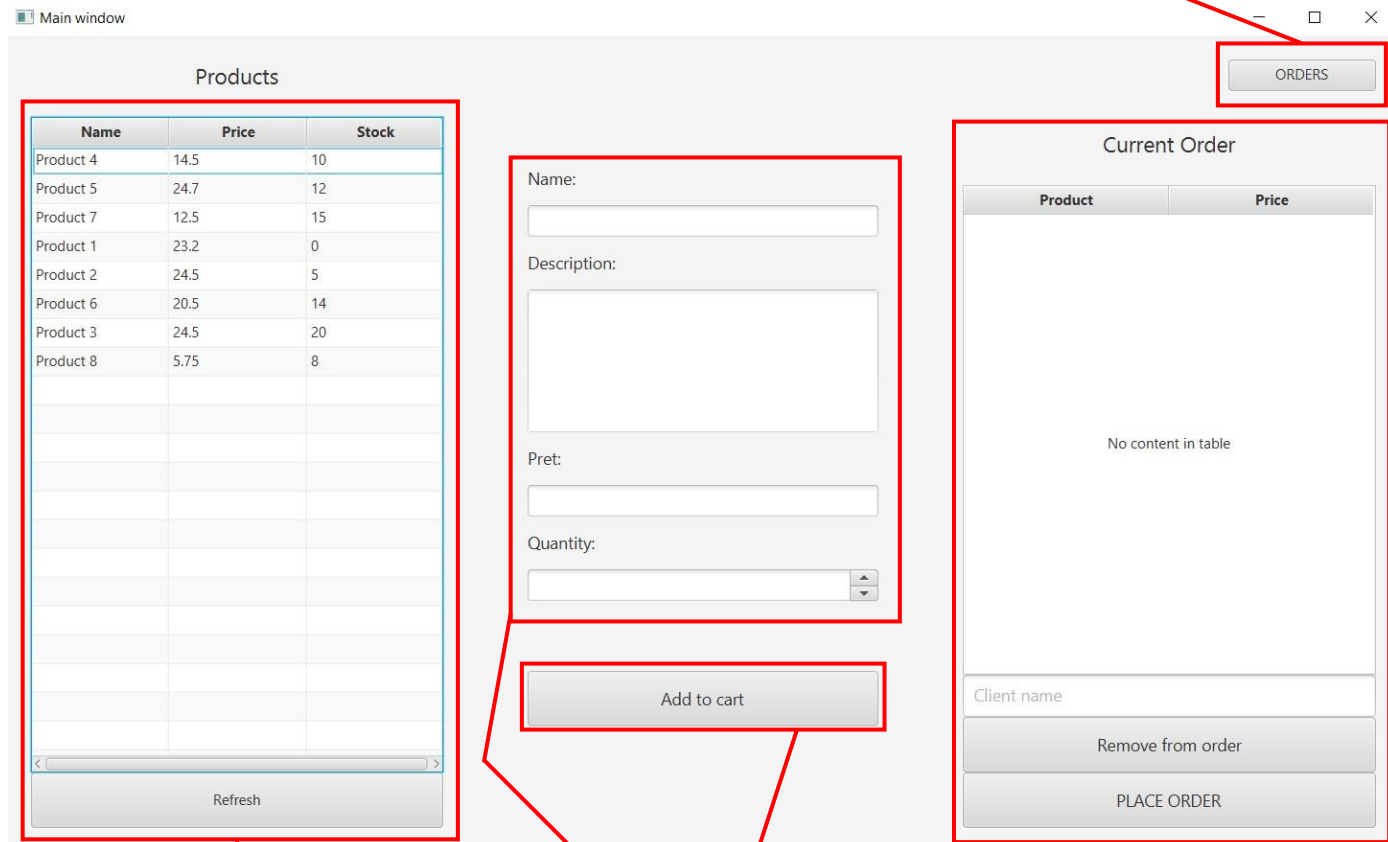
Repeat password:

Dupa completarea cu date a textfield-urilor, la apasarea butonului se va crea un nou cont pentru utilizator iar acesta va fi autentificat in aplicatie

In cazul in care utilizatorul are deja un cont existent la apasarea acestui buton se revine la fereastra de login

## Fereastra principala pentru agentul de vanzari:

La apasarea butonului se deschide fereastra ce contine toate comenzile plasate de catre agentul autentificat in aplicatie



In partea stanga se afiseaza o lista cu toate produsele si detalii despre ele precum si un buton care reincarca lista

La apasarea pe un produs din lista, in chenarul din mijloc se incarca datele despre produs, iar dupa selectarea cantitatii si apasarea butonului acest produs se va adauga la comanda curenta

In partea dreapta se afiseaza o lista cu produsele adaugate la comanda curenta.

Dupa selectarea unui produs din lista, se poate actiona butonul de remove pentru a scoate acel produs din comanda.

Dupa completarea numelui  
clientului prin apasarea  
butonului de jos se trimite o  
noua comanda

Fereastra cu comenzile agentului:

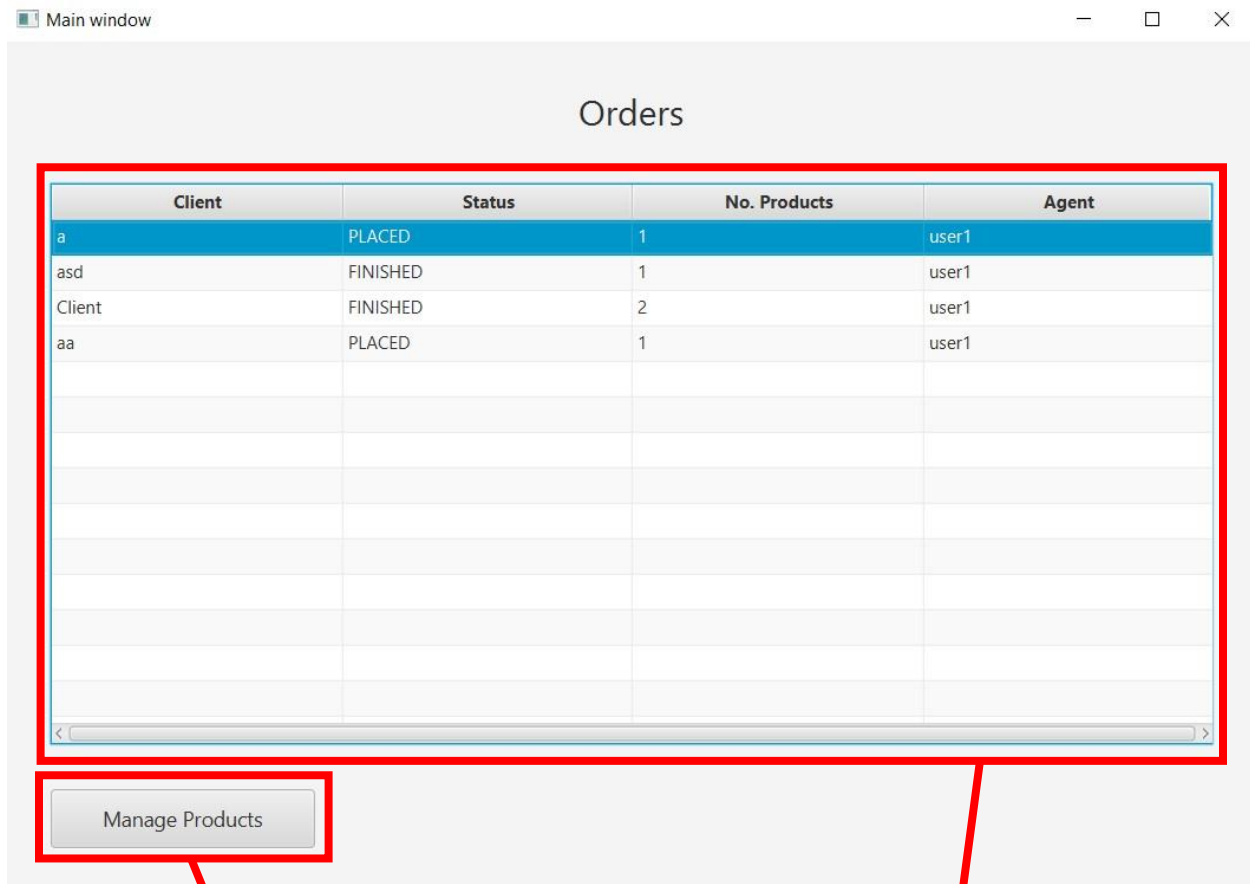
The screenshot shows a window titled "Orders" with a table of orders and two buttons below it. The table has three columns: "Client", "Status", and "No. Products". The data in the table is as follows:

Client	Status	No. Products
a	PLACED	1
asd	FINISHED	1
Client	FINISHED	2
aa	PLACED	1

Below the table are two buttons: "Mark as finished" and "Cancel order". Red boxes and lines highlight these buttons and their corresponding descriptions:

- Mark as finished:** Dupa selectarea unei comenzi, la apasarea butonului, comanda respectiva va fi marcata ca indeplinita
- Cancel order:** Dupa selectarea unei comenzi, la apasarea butonului, comanda respectiva va fi anulata si stearsa definitiv

Fereastra principala pentru admin:



La apasarea butonului se deschide fereastra din care se pot aduce modificari produselor

In lista se afiseaza toate comenzile plasate de toti agentii de vanzari

Fereastra de modificare a listei de produse:

Name	Price	Stock
No content in table		

Name

Description

Price

Stock

Add

Update

Delete

Lista cu toate produsele si detalii despre fiecare

Butoane ce permit adaugarea, actualizarea si stergere produselor in functie de datele completate in textfield-urile de mai sus



## 8. Referinte

- Dr. Lect. Petrascu Vladila – Ingineria Sistemelor Soft, Note de curs
- Dr. Conf. Cojocar Grigoreta – Medii de proiectare si programare, Note de curs
- <https://simaioan.wordpress.com/2023/03/13/seminarii-iss-2021/>
- <https://www.javadevjournal.com/spring-boot/spring-boot-with-hibernate/>
- <https://www.javatpoint.com/hibernate-one-to-many-mapping-using-annotation-example>
- <https://www.codejava.net/frameworks/hibernate/hibernate-parent-child-mapping-example>
- <https://www.baeldung.com/spring-boot-hibernate>
- <https://chat.openai.com/>
- <https://www.baeldung.com/spring-autowire>
- <https://stackoverflow.com/questions/3964059/jpa-default-column-name-mapping-for-manytoone-relations>
- <https://vladmihalcea.com/manytoone-jpa-hibernate/>
- <https://vladmihalcea.com/the-best-way-to-map-a-onetomany-association-with-jpa-and-hibernate/>