

DOCUMENTAȚIE

TEMA 3

NUME STUDENT: **Neacșu Ioana-Alina**
GRUPA: 30221

Cuprins

1. Obiectivul temei	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3. Proiectare	4
4. Implementare	5
5. Concluzii	10
6. Bibliografie	10

1. Obiectivul temei

Obiectivul acestei teme este de a dezvolta o aplicație de management al comenzilor pentru procesarea comenzilor clienților unui depozit. Aplicația va folosi o bază de date relațională pentru a stoca informațiile despre produse, clienți și comenzi.

2. Analiza problemei, modelare, scenarii, cazuri de utilizare

Cerințe Funcționale:

- Aplicația trebuie să permită înregistrarea și stocarea datelor despre produse, clienți și comenzi într-o bază de date.
- Utilizatorii trebuie să poată efectua operațiuni CRUD (create, read, update, delete) asupra datelor despre produse, clienți și comenzi.
- Utilizatorii trebuie să poată plasa comenzi pentru produse disponibile în depozit, selectând un client existent dintr-o listă și specificând cantitatea dorită.

Cerințe Non-Funcționale:

- Interfața grafică a utilizatorului trebuie să fie intuitivă și ușor de utilizat.
- Accesul la baza de date trebuie să fie eficient și sigur.

3. Proiectare

Diagrama Pachetelor:

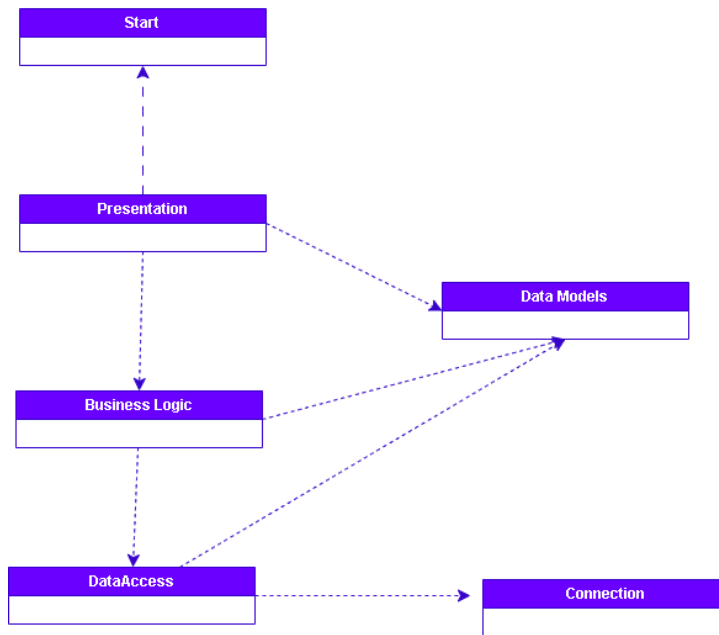
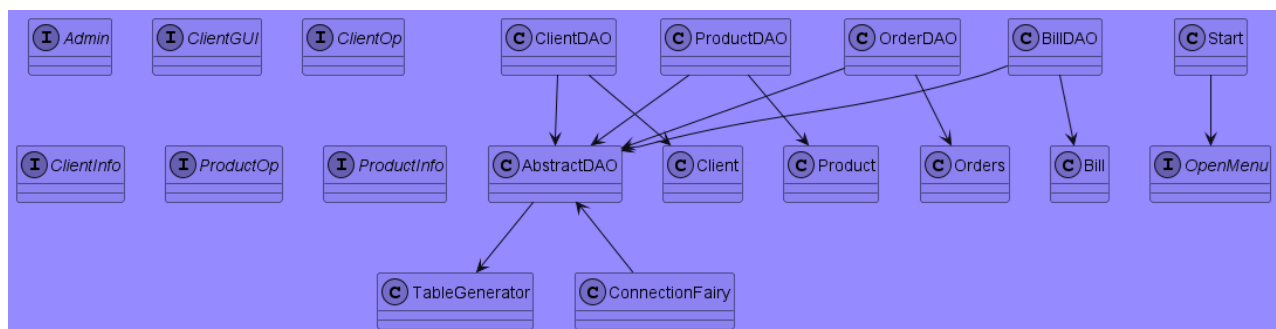


Diagrama Claselor



4. Implementare

1. *Client:*

- **Fields:**
 - id: int - reprezintă identificatorul unic al clientului.
 - name: String - numele clientului.
 - email: String - adresa de email a clientului.
 - password: String - parola clientului.
 - address: String - adresa clientului.
- **Metode importante:**
 - Constructori pentru inițializarea obiectelor cu diferite combinații de date.
 - Metode de acces (getters și setters) pentru a citi și actualiza câmpurile obiectelor.
 - toString(): Returnează o reprezentare text a obiectului, utilă pentru afișarea informațiilor despre client.

2. *Product:*

- **Fields:**
 - id: int - identificatorul unic al produsului.
 - name: String - numele produsului.
 - quantity: int - cantitatea disponibilă în stoc.
 - price: Double - prețul produsului.
- **Metode importante:**
 - Constructori pentru inițializarea obiectelor cu diferite combinații de date.
 - Metode de acces (getters și setters) pentru a citi și actualiza câmpurile obiectelor.
 - decreaseQuantity(int quantity): Scade cantitatea disponibilă în stoc cu o anumită valoare.

3. *Order:*

- **Fields:**
 - id: int - identificatorul unic al produsului.
 - name: String - numele produsului.
 - quantity: int - cantitatea disponibilă în stoc.
 - price: Double - prețul produsului.

- **Metode importante:**

- Constructor pentru inițializarea comenzii cu informațiile necesare.
- Metode de acces (getters și setters) pentru a citi și actualiza câmpurile obiectului.
- getCounter(): Returnează un contor static entru a genera identificatori unici pentru comenzile noi.

4. Bill:

- **Fields:**

- clientId: int - identificatorul unic al clientului pentru care a fost emisă factura.
- product: String - numele produsului cumpărat.
- quantity: int - cantitatea de produse cumpărate.
- totalPrice: Double - prețul total al facturii.
- timestamp: LocalDateTime - marcajul de timp când factura a fost emisă.

- **Metode importante:**

- Constructor pentru inițializarea facturii cu informațiile necesare.
- Metode de acces (getters și setters) pentru a citi și actualiza câmpurile obiectului.

5. Table Generator:

- **Metode importante:**

- generateTable(): Generează un tabel JTable pe baza datelor obținute dintr-un DAO și a clasei de model specificate.

6. AbstractDAO:

Extinsă de ClientDAO, ProductDAO, OrderDAO.

- **Metode importante:**

- findAll(): Returnează o listă cu toate obiectele din baza de date asociate clasei.
- findById(int id): Returnează un obiect din baza de date pe baza unui ID dat.
- insert(T obj): Inserează un nou obiect în baza de date.
- update(T obj): Actualizează datele unui obiect existent în baza de date.
- delete(int id): Șterge un obiect din baza de date pe baza ID-ului său

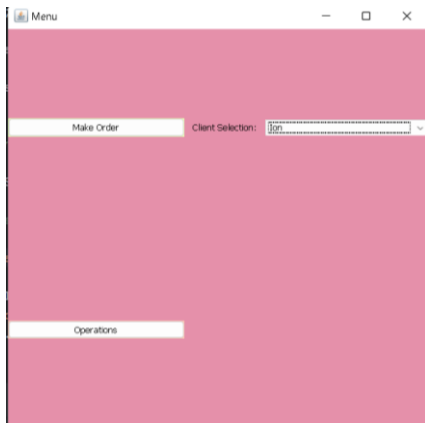
7. *BillDAO:*

- **Metode importante:**

- insertBill(Bill bill): Inserează o nouă factură în baza de date. Această metodă suprascrie metoda insert din clasa AbstractDAO pentru a gestiona obiectele de tip Bill.

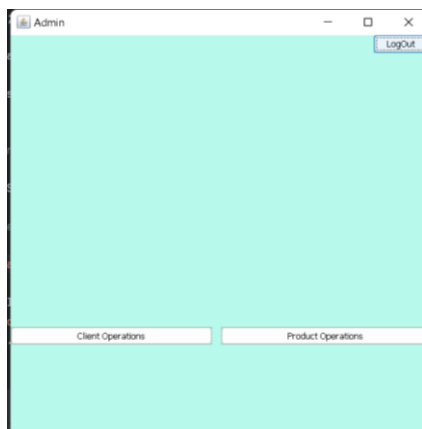
8. *OpenMenu:*

Un meniu de navigare care permite utilizatorului să aleagă între operațiuni precum plasarea unei comenzi sau administrarea datelor (adăugare, ștergere, actualizare) despre clienți și produse.



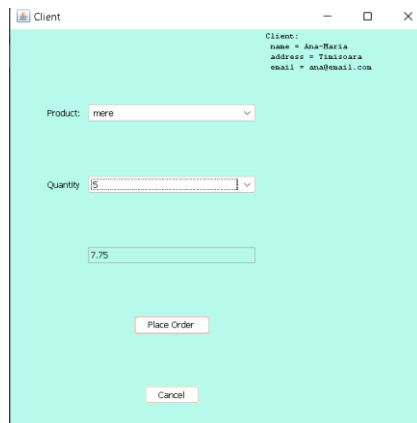
9. *Admin:*

Panou de administrare pentru utilizatorii cu rol de admin, oferind butoane pentru gestionarea clienților și produselor.



10. *ClientGUI:*

Interfață pentru utilizatorii obișnuiți, permite plasarea de comenzi și vizualizarea informațiilor despre clienți.



The screenshot shows a window titled "Client" with a light blue background. In the top right corner, client information is displayed: "Client:", "name = Ana-Maria", "address = Timisoara", and "email = anamail.com". The main area contains a form with the following elements: a "Product:" label followed by a dropdown menu showing "mere"; a "Quantity:" label followed by a dropdown menu showing "5"; a text input field containing "7.75"; a "Place Order" button; and a "Cancel" button.

11. *ClientInfo:*

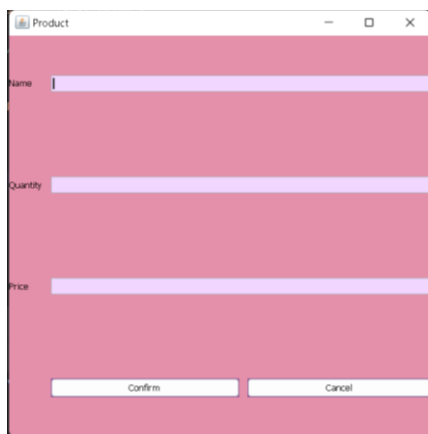
Formular pentru adăugarea sau editarea informațiilor despre un client.



The screenshot shows a window titled "Client" with a light blue background. It contains a form with four labels and corresponding text input fields: "username", "mail", "password", and "address". At the bottom of the form are two buttons: "Confirm" and "Cancel".

12. *ProductInfo:*

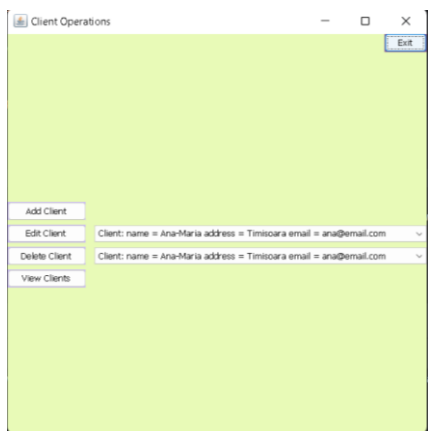
Formular pentru adăugarea sau editarea informațiilor despre un produs.



A screenshot of a Windows application window titled "Product". The window has a pink background. It contains three text input fields labeled "Name", "Quantity", and "Price" on the left side. At the bottom of the window, there are two buttons: "Confirm" and "Cancel".

13. *ClientOP:*

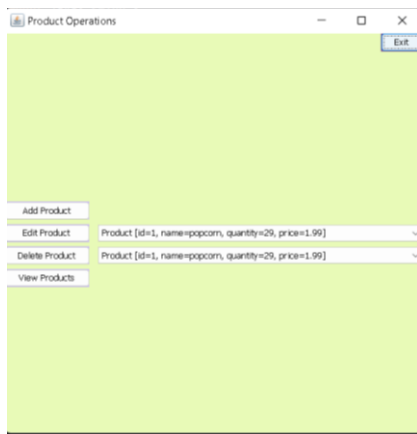
Panou de operații pentru admin, permitând adăugarea, ștergerea, editarea și vizualizarea clienților.



A screenshot of a Windows application window titled "Client Operations". The window has a light green background. In the top right corner, there is an "Exit" button. On the left side, there are four buttons: "Add Client", "Edit Client", "Delete Client", and "View Clients". To the right of the "Edit Client" and "Delete Client" buttons, there are dropdown menus showing client information: "Client: name = Ana-Maria address = Timisoara email = ana@email.com".

14. ProductOP:

Panou de operații pentru admin, permitând adăugarea, ștergerea, editarea și vizualizarea produselor.



5. Concluzii

Tema implementată a oferit oportunitatea de a învăța și aplica concepte esențiale în dezvoltarea aplicațiilor, cum ar fi organizarea modulară a codului. Prin implementarea claselor de acces la date și utilizarea unei baze de date relaționale, am înțeles importanța gestionării datelor într-un mod eficient și sigur. Dezvoltarea interfeței utilizator a consolidat înțelegerea despre cum să proiectăm o experiență intuitivă pentru utilizatori.

6. Bibliografie

1. https://gitlab.com/utc_n_dsrl/pt-reflection-example
2. <https://blog.jetbrains.com/idea/2021/03/java-16-and-intellij-idea/>
3. <https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>
4. <https://www.baeldung.com/javadoc>