Proiect T.P.

Procesarea comenzilor pentru un depozit

de Miulescu Alexandru

**Analiza problemei**

Problema de soluționat este reprezentată de faptul că în lumea reală, sistemele bazate pe politici de marketing dovedesc adesea ineficiență în ceea ce privește înregistrarea comenzilor sau păstrarea consistentă a datelor referitoare la clienți și produse. Prin urmare, a fost identificat, în urma analize problemei menționate, următorul cadru de cerințe funcționale:

• Aplicația trebuie să poată permite introducerea numărului de noi clienți, caracterizați prin nume, adresă, adresă de e-mail și vârstă, și de produse disponibile, identificate prin nume, preț și cantitate disponibilă în stoc;

• Aplicația trebuie să informeze utilizatorul dacă inputul introdus este invalid, i. e. e-mailul nu respectă șablonul universal pentru o adresă de e-mail, clienții înregistrați nu sunt majori sau cantitatea disponibilă pentru un produs nu este un număr natural;

• Aplicația trebuie să poată permite selectarea actualizarea și ștergerea informațiilor referitoare la clienți și produse și menținerea consistentă a acestora;

• Aplicația trebuie să permită selectarea unui client existent în baza de date, selectarea unui tip de produs din stoc, introducerea unei cantități dorite de către client și plasarea comenzii;

• Aplicația trebuie să permită vizualizarea tuturor clienților înregistrați și a tuturor produselor disponibile spre a fi achiziționate;

• Aplicația trebuie să informeze utilizatorul dacă în stoc nu se află suficiente produse pentru a putea satisface cererea acestuia.

Cadrul cerințelor nefuncționale conturate în jurul problemei de rezolvat este:

• Aplicația trebuie să dispună de o interfață grafică intuitivă.

• Aplicația trebuie să fie ușor de utilizat pentru orice tip de utilizator, indiferent de domeniul de activitate al acestuia.

Aplicația a fost proiectată folosind un model de sistem cu trei intrări compuse și o ieșire. Intrările sunt reprezentate de trei entități: client, produs și comandă. Fiecare entitate are atributele sale specifice:

1. Client: Atributele clientului includ numele, adresa, adresa de e-mail și vârsta.

2. Produs: Produsul este caracterizat prin nume, preț și cantitatea disponibilă în stoc.

3. Comandă: Comanda reprezintă o înregistrare a achiziției și conține informații despre clientul care a plasat comanda și produsul comandat.

Ieșirea aplicației este reprezentată de o factură generată în urma plasării și înregistrării comenzii în baza de date. Factura conține un identificator unic pentru comandă și prețul total al comenzii.

În modelul orientat pe obiecte al problemei, sunt definite următoarele clase:

1. Clasa "Client" reprezintă entitatea client și conține atributele specifice ale unui client.

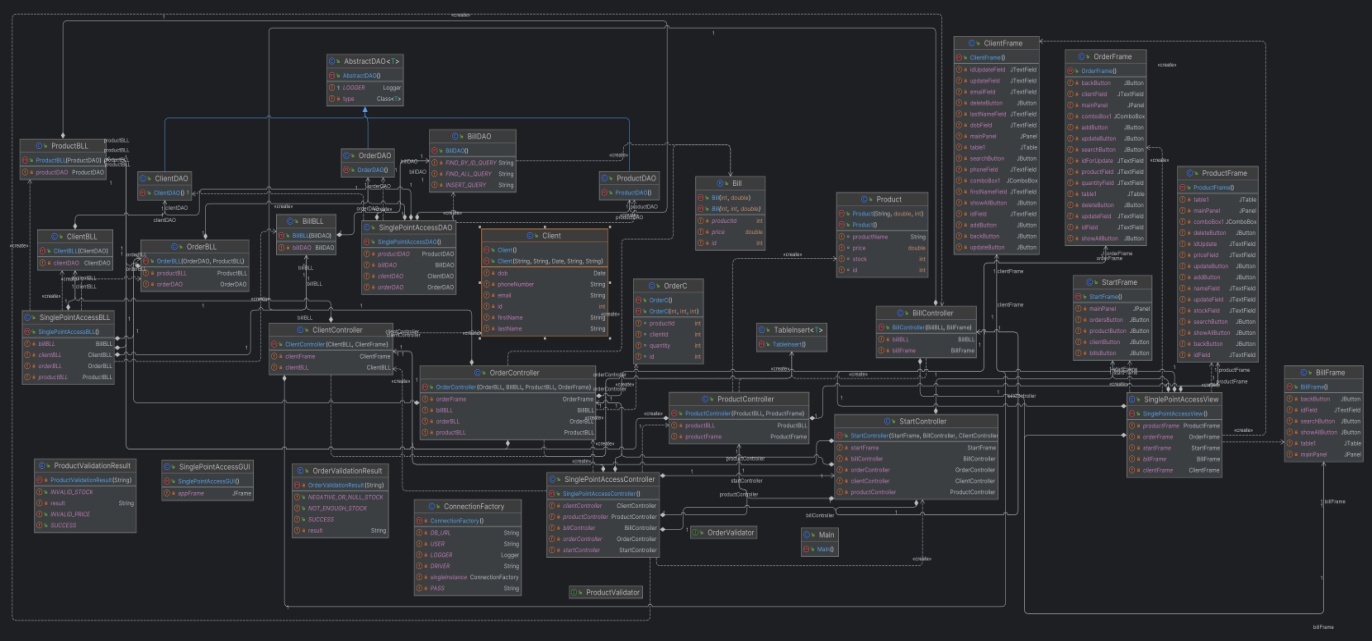
2. Clasa "Product" reprezintă entitatea produs și conține atributele specifice ale unui produs.

3. Clasa "Order" reprezintă entitatea comandă și conține informații despre clientul care a plasat comanda și produsul comandat.

4. Record "Bill" reprezintă entitatea factură și conține identificatorul unic al comenzii și prețul total al comenzii.

Astfel, modelul problemei este tradus într-un model orientat pe obiecte, unde fiecare clasă reprezintă o entitate specifică și conține atributele corespunzătoare.

**Proiectare**



Proiectarea constă în împărțirea aplicației în subaplicații specializate în funcționalități. Componentele subsistemelor aplicației sunt organizate conform unui pattern arhitectural de tip Layered Architecture, care include următoarele niveluri de divizare