

MyDrinkShop

Se cere realizarea unei aplicații software pentru gestionarea activității unui drinks shop, care să permită administrarea produselor vândute (băuturi), a ingredientelor utilizate, a rețetelor de preparare și a comenzilor plasate de clienți.

Aplicația trebuie să ofere funcționalități complete de gestiune (adăugare, modificare, ștergere, vizualizare), să asigure persistența datelor în fișiere, precum și o interfață grafică intuitivă pentru utilizator.

Soluția trebuie implementată respectând principiile programării orientate pe obiect și o arhitectură stratificată, care separă clar:

- modelul de date,
- logica de business,
- accesul la date,
- interfața cu utilizatorul.

Cerințe funcționale

1. Aplicația trebuie să permită gestionarea produselor (băuturi):
 - adăugare produs nou;
 - modificare produs existent;
 - ștergere produs;
 - afișare listă produse.
2. Fiecare produs trebuie să aparțină unui tip și unei categorii.
3. Utilizatorul poate adăuga, modifica și șterge tipuri și categorii de băuturi.
4. Aplicația trebuie să permită definirea rețetelor pentru produse:
 - fiecare rețetă este compusă din mai multe ingrediente;
 - pentru fiecare ingredient se specifică o cantitate.
5. Aplicația trebuie să gestioneze stocurile de ingrediente:
 - afișarea cantității disponibile;
 - actualizarea stocului în urma comenzilor.
6. Aplicația trebuie să permită crearea comenzilor:
 - o comandă conține unul sau mai multe produse;
 - fiecare produs are asociată o cantitate.
7. Aplicația trebuie să calculeze automat conținutul unei comenzi pe baza rețetelor produselor.
 - la finalizarea comenzii se generează bonul de casă care se salvează în format .csv.
8. La finalul fiecărei zile se salvează informațiile despre comenzile înregistrate și se calculează totalul zilei. Datele pot fi exportate în format CSV.

Cerințe non-funcționale

1. Aplicația trebuie să fie o **aplicație desktop Java**.
2. Interfața grafică trebuie realizată folosind **JavaFX**.
3. Persistența datelor trebuie realizată în **fișiere text**.
4. Accesul la date trebuie implementat folosind **Repository Pattern**.
5. Logica aplicației trebuie separată de interfața grafică (layer Service).
6. Codul trebuie să respecte principiile OOP (încapsulare, separarea responsabilităților).