DOCUMENTATIE

TEMA *3*

Order management

NUME STUDENT: Niță Alexandru-Gabriel

GRUPA: 30223

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc95297886)

[3. Proiectare 3](#_Toc95297887)

[4. Implementare 3](#_Toc95297888)

[5. Rezultate 3](#_Toc95297889)

[6. Bibliografie 3](#_Toc95297891)

# Obiectivul temei

Obiectivul principal al temei date este proiectarea si implementarea unei aplicații de gestionare pentru procesarea comenzilor clienților pentru un depozit.Bazele de date trebui să fie utilizate pentru a stoca produsele, clienții și comenzile.

Aplicația trebuie să fie proiectată în conformitate cu modelul prezentat in cerinta si să utilizeze (minim) următoarele clase:

Clasa Model – reprezinta modelele de date ale aplicatiei

Clasa Business Logic – reprezinta clasa care se ocupa cu logica aplicatiei

Clasa Presentation – reprezinta clasa referitoare la interfata grafica

Clasa Data Access – clasa care se va ocupa cu accesul la baza de date.

Programul trebuie să conțină minim 3 tabele Client, Produs si Comandă în care se vor stoca datele clientului (id, nume, adresă), datele produsului (id, numele produsului, cantitatea, prețul), datele comenzii (id-ul comenzii, id-ul clientului, id-ul produsului, cantitatea produsului). Programul va trebui să afișeze clienții, produsele și comenzile din baza de date. Se vor implementa operații de inserare în tabel a unui client, a unui produs sau a unei comenzi, ștergere a acestora sau modificarea lor.

# 

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

Modelul utilizat in rezolvarea problemei a fost cel primit in materialul suport. Astfel, Programul va trebui să afișeze clienții, produsele și comenzile din baza de date. Se vor implementa operații de inserare în tabel a unui client, a unui produs sau a unei comenzi, ștergere a acestora sau modificarea lor.

**A picture containing text, diagram, sketch, drawing

Description automatically generated**

**2.1Diagrama USE CASE:**

A picture containing diagram, text, line, technical drawing

Description automatically generated

**3.Proiectare**

**4.Implementare.Clase**

Deoarece tema propusă este mai complicată și se conectează la o bază de date pentru a păstra datele despre clienți, produse și comenzi, a fost necesară utilizarea mai multor clase pentru a o gestiona. Urmând modelul demonstrat în laborator, au fost construite clasele.

**Clasa Client**: Această clasă furnizează informații despre client, inclusiv numele prenumele varsta și ID-ul acestuia. Id, este un număr discret pentru client, prenumele este de tip șir de caractere, împreună cu numele. În această clasă sunt incluse, de asemenea, metode de setare și de obținere a atributelor clientului.

**Clasa Order**: Id-ul, cantitatea, clientul și produsul comenzii sunt toate incluse în această clasă. Id este un număr special pentru o comandă de tip int, , iar produs este tot de tipul int care arată id-ul produsului care se dorește comandat.

**Clasa Produs**: Această clasă include detalii despre articolul care este achiziționat, inclusiv numele, cantitatea, prețul și ID-ul. Id-ul fiecărui produs este un număr întreg distinct, iar cantitatea este un număr întreg care indică numărul de produse aflate în stoc. Numele este o variabilă de tip String care conține numele fiecărui produs, iar prețul este un număr de tip double care ne informează cu privire la prețul unui anumit produs.

**Clasa AbstractDAO**: Funcțiile create pentru a construi șirul de caractere pentru a accesa tabelul, funcția findall, findById, createObjects și insert, care adaugă un client, un produs sau o comandă în tabelul corespunzător fiecăruia, sunt toate conținute în această clasă.

**Clasa ConnectionFactory**: Această clasă se conectează la baza de date care conține informații despre clienți, produse și comenzi. Această clasă are metode care ajută la conectarea la baza de date, precum și LOGGER, DRIVER, DBURL, USER și PASS.

**Clasa ClientBLL**: Această clasă include toate funcțiile ClientDAO, cum ar fi findClientById, findALL, delete, update și insert pentru a accesa datele din baza de date.

**Class ProductBLL**: Această clasă include toate acțiunile ProductDAO, inclusiv findProdustById, findALL, delete, update și insert pentru accesarea bazei de date.

**Class OrderBLL**: pentru a accesa datele din baza de date, această clasă conține toate acțiunile din OrderDAO, inclusiv findCommandById, findALL, delete, update și insert.

**Clasa View**: În această clasă se implementează interfața pentru fereastra clientului. Această clasă va conține crearea unui tabel specific pentru client unde se vor afișa datele corespunzătoare clientului din baza de date, din acest tabel se vor șterge dacă se dorește datele despre un client, se va insera un nou client sau se vor modifica datele unui client.

**Clasa ProductGUI**: În această clasă se implementează interfața pentru fereastra produsului. Această clasă va conține crearea unui tabel specific pentru produs unde se vor afișa datele corespunzătoare produsului din baza de date, din acest tabel se vor șterge dacă se dorește datele despre un produs, se va insera un nou produs sau se vor modifica datele unui produs

**Clasa OrderGUI**: În această clasă se implementează interfața pentru fereastra comenzii. Această clasă va conține crearea unui tabel specific pentru comandă unde se vor afișa datele corespunzătoare comenzii din baza de date, din acest tabel se vor șterge dacă se dorește datele despre o comandă, se va insera o nouă comandă sau se vor modifica datele unei comenzi.

Bibliografie:

<https://dsrl.eu/courses/pt/>

<https://dsrl.eu/courses/pt/materials/PT2023_A3_S2.pdf>

<https://dsrl.eu/courses/pt/materials/PT2023_A3_S1.pdf>

<https://www.youtube.com/>

+materiale de la laborator