

# Management de comenzi

## Tehnici de programare

### Tema 3

Alexandru Nistor Botolan  
An 2 Seria B Grupa 30227  
Facultatea de Automatica si Calculatoare  
Sectia Calculatoare si Tehnologia Informatiei

## 1.Obiectivul temei

Obiectivul principal al temei este de a crea o aplicatie care se comunica cu o baza de date si permite prin intermediul unei interfete grafice manipularea inregistrarilor acesteia si de a simula cat mai bine un management de comenzi intre produse si clienti

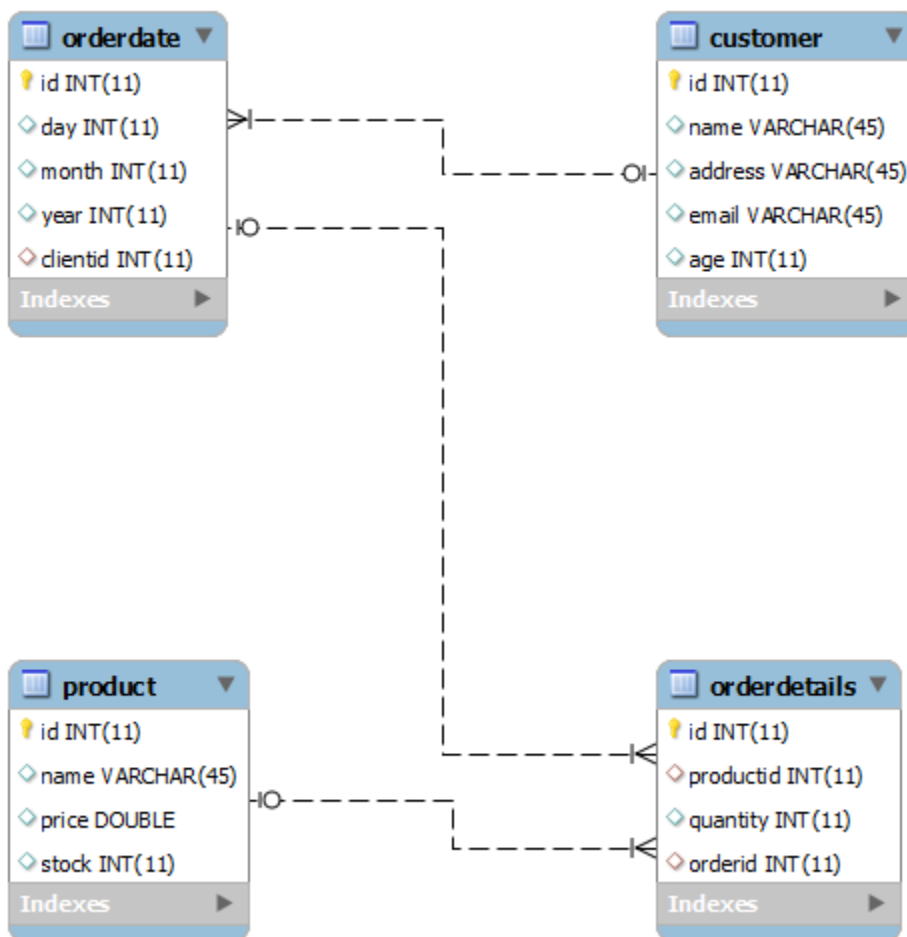
Obiectivele secundare sunt reprezentate de realizarea unei interfete grafice pentru a executa operatiile dorite, afisarea continutului tabelii, de a fi capabili de a afisa inregistrarile, de a putea sterge si a adauga inregistrari.

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

Aceasta aplicatie se ocupa cu managementul de comenzi, astfel ea trebuie sa fie in stare sa adauge noi comenzi in functie de dorintele utilizatorului. Pentru a realiza aceasta, toata informatia trebuie sa fie retinuta intr-o baza de date relationala, iar aplicatia trebuie sa comunice cu ea, fiind capabila sa extraga informatii si sa le prelucreze, dar si sa introduca. Pentru ca aplicatia sa ruleze optim, baza de date trebuie sa fie bine definita, astfel proiectarea acesteia trebuie sa fie buna. De asemenea, aplicatia trebuie sa fie in stare sa primeasca informatii de la utilizator cu privire la operatia dorita, dar si sa primeasca date pe care sa le transmita mai departe bazei de date.

### 3. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator, modul de tratare a erorilor)

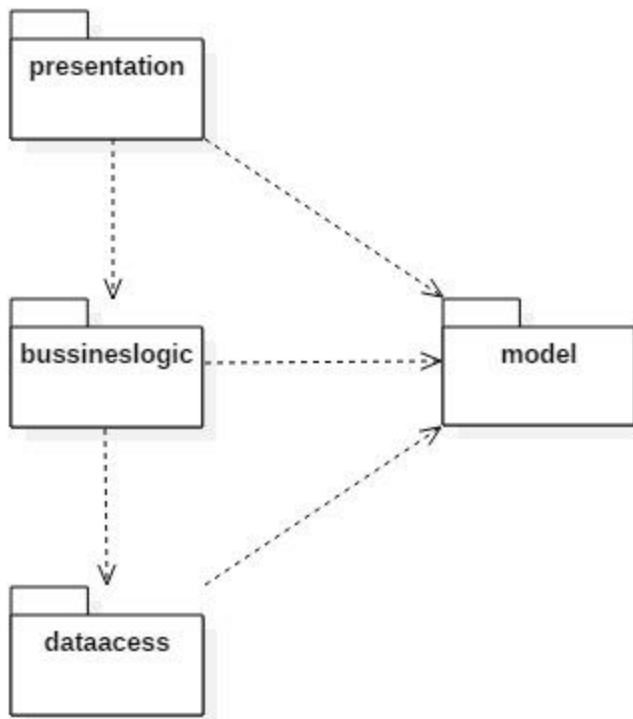
Diagrama bazei de date



In aceasta diagrama se pot observa tabelele: Tabela customer se ocupa cu retinerea informatiei despre un anumit client identificat prin campul id care este, cheia primara a tabelii. De asemenea campurile name, address, email si age vor oferi detalii despre client si va face mai usoara identificarea acestuia de catre utilizator. Tabela product retine informatii despre un anumit produs identificat prin id, care este cheia primara a tabelii. Pe langa aceasta, name, price si stock

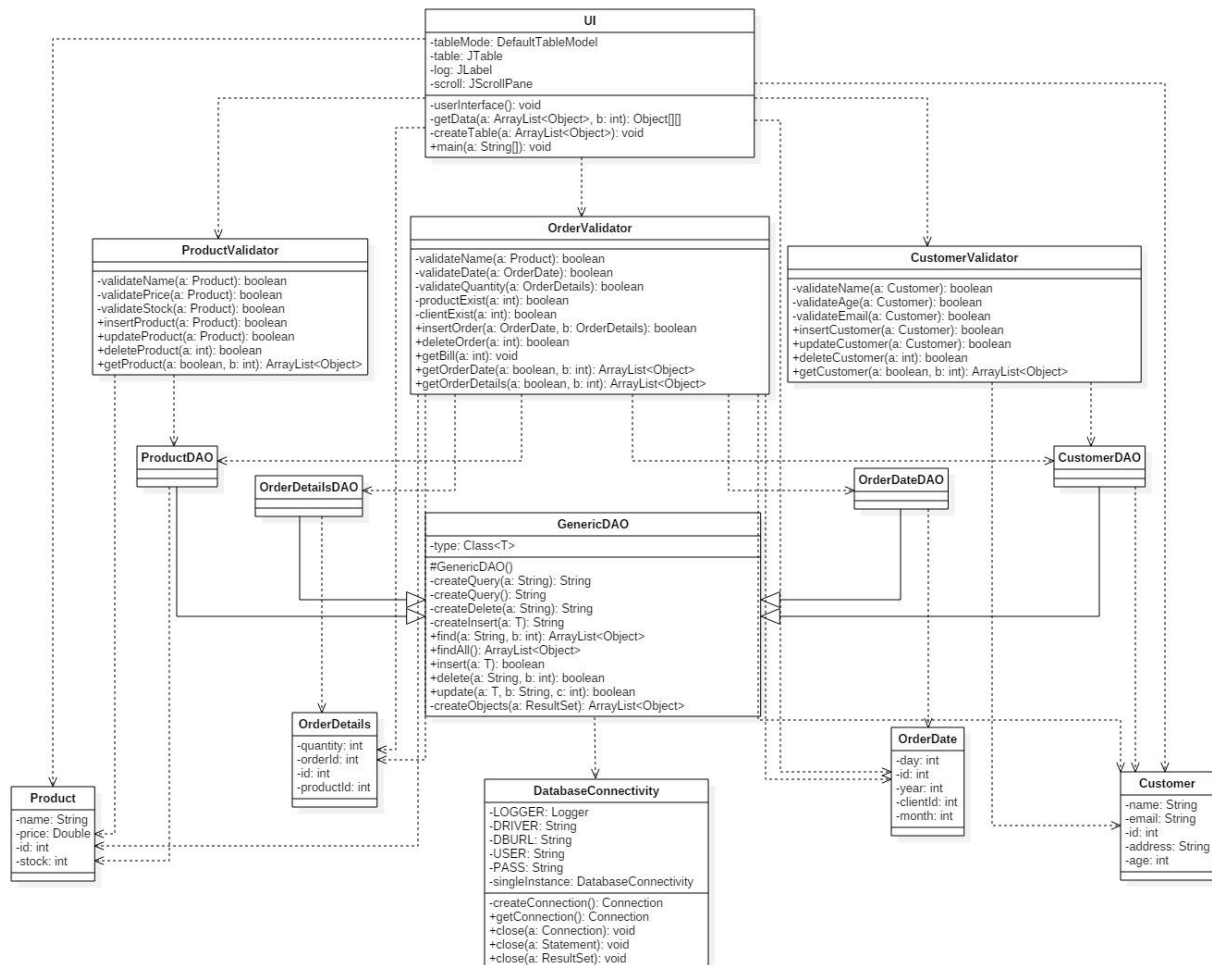
ofera informatii aditionale despre produsul respectiv. Tabela orderdate se ocupa cu retinerea informatiei despre o anumita comanda, comanda identificata prin intermediul campului id, care este cheie primara, iar day, month si year ofera informatii despre comanda. Campul clientid reprezinta cheia straina a tabelei care face legatura cu tabela customer si ne ajuta sa preluam informatii despre clientul care detine acea comanda. Tabela orderdetails are ca si cheie primara campul id, care identifica un camp din tabela. Pe langa aceasta quantity ofera cantitatea unui produs comandat, iar productid ofera informatii despre produsul comandat. Orderid ofera informatii despre comanda creata, cu ajutorul ei putem stoca mai multe produse comandate pentru aceiasi comanda. Baza de date este bine definita datorita faptului ca nu exista relatii ilegale, adica m-n, exista doar 1-n.

Diagrama de pachete



Conform acestei diagrame, aplicatia este realizata cu ajutorul Layered Architecture, aplicatia fiind impartita pe straturi care comunica una cu alta doar intr-o singura directie.

## Diagrama UML



In primul rand avem clasele care definesc tabelele din baza de date: Product, OrderDetails, OrderDate si Customer care se ocupa cu stocarea informatiilor din tabele sub forma de obiecte si implementeaza operatiile CRUD, operatii de baza pentru orice clasa. Toate aceste clase fac parte din pachetul model.

Avem clasa UI care face parte din pachetul presentation si se ocupa cu crearea interfetei grafice, preloarea datelor si a optiunilor dorite de utilizator si de a prezenta rezultatele operatiei.

ProductValidator, OrderValidator si CustomerValidator fac parte din pachetul bussineslogic si se ocupa cu prelucrarea datelor introduse de catre utilizator, verificarea daca acestea sunt corecte si de a le transmite mai departe claselor de accesarea a bazei de date sub forma de obiecte.

GenericDAO impreuna cu celelalte clase din pachetul dataaccess se ocupa cu accesul la baza de date. GenericDAO acceseaza baza de date cu ajutorul clasei DatabaseConnectivity si extrage datele din tabela sub forma necunoscuta si cu ajutorul metodei reflection, aceasta creaza obiectele dorite si la trimite inapoi claselor din pachetul bussineslogic.

#### 4. Implementare

Clasa UI va construi o fereastră compusă din 3 câmpuri mari: câmpul de selecție care se ocupa cu selecția panoului dorit: operații pe clienți, operații pe produse sau operații pe comenzi. Al doilea câmp se ocupa cu preluarea datelor din câmpul textfield si transmiterea lor mai departe. Pe lângă acestea mai există si opțiunile pe care utilizatorul le poate selecta în funcție de ce își dorește, selecția acestora vor genera un rezultat în funcție de opțiunea selectată.

Clasele de validare se ocupă toate cu același lucru: validează datele de intrare transmise de către utilizator print intermediul interfeței grafice. CustomerValidator se ocupa cu validarea datelor pentru clienți, verifică dacă numele si adresa de email sunt valide cu ajutorul unor regex-uri, în cazul în care sunt valide sunt permise operații adiționale cum ar fi adăugare, ștergere, actualizare sau căutare după un criteriu. Aceste operații sunt realizate cu apelurile către clasele DAO, cu ajutorul cărora se extrag informații. Clasa OrderValidator se ocupa si cu generarea unei facturi pentru o anumită comandă, factura care conține momentul în care s-a generat factura, numele clientului, produsele si suma totală de plată.

Clasele DAO extind clasa GenericDAO care se ocupa cu accesul la baza de date. Aceasta clasă se ocupa cu manipularea parametrilor generici. În funcție de tipul dorit, adică tipul obiectului cu care se accesează metodele, aceasta va prelucra si transmite către clasele apelante obiectele de tipul dorit. Clasa conține

metode care construiesc, in functie de obiect, operatii SQL de inserare, stergere, actualizare si interogare sub forma de sir de caractere.

Clasa DatabaseConnectivity se ocupa cu conectarea la baza de date. Aceasta foloseste strategia de Singleton, ceea ce inseamna ca va exista doar o singura instanta a clasei pe toata durata executiei programului.

## 5. Testare/Rezultate

[illegible]

Pentru testare vom rula programul propriu-zis. Din screenshot-ul de mai sus putem observa interfata grafica care este prezentata utilizatorului. In partea de sus se observa 3 butoane: Panou Clienti, Panou Comenzi si Panou Produse. Aceste butoane au rolul de a selecta panoul si odata cu el si operatiile care se vor face pe tabela din baza de date care corespunde fiecarui panou. In partea din mijloc a ferestrei se observa butoanele care semnifica operatiile care se pot face pe tabela care este prezentata pe interfata. View all se ocupa cu extragerea tuturor inregistrarilor dintr-o tabela si prezentarea lor mai departe pe baza unui tabel care este desenat mai jos. Interogarea se realizeaza cu ajutorul unei valori introduse in campul id din panoul pentru introducerea datelor. Butonul Insereaza se ocupa cu introducerea unei noi inregistrari in tabelul care corespunde panoului deschis. In functie de datele introduse se realizeaza o inserare in tabela respectiva, sau, in cazul unei erori se afiseaza un mesaj care va anunta utilizatorul de aceasta. Butonul cauta dupa id se ocupa cu selectarea inregistrarii care are ca si id valoarea introduse in campul care ii corespunde. Sterge dupa id se ocupa cu stergerea unei inregistrari din tabela care are ca si id valoarea introdusa in campul care ii corespunde. Actualizeaza client se ocupa cu executarea unei instructiuni de update in tabela respectiva pe baza datelor introduse in campurile respective. Campurile care se afla dedesuptul butoanelor de operatii se ocupa cu colectarea informatiei introduse de catre utilizator si transmiterea ei mai departe claselor care se ocupa cu prelucraturile datelor si realizarea operatiilor pe tabelele corespunzatoare bazei de date. Tabelul care se afla in partea de jos a interfetei se ocupa cu prezentarea datelor rezultate in urma executarii unei instructiuni SQL. Tabelul va fi intotdeauna format dintr-un numar variabil de coloane, numarul de coloane fiind determinat de numarul de campuri care exista in tabela, in functie de tabela selectata acesta poate fii mai mare sau mai mic. In cazul in care se executa o instructiune care nu are ca scop extragerea datelor dintr-o tabela din baza de date, tabelul nu va fi afisat, insa va fi afisat un mesaj care va prezenta situatia de succes sau esuare a operatiei care a fost executata in urma selectiei utilizatorului.

## 6. Concluzii

Aceasta aplicatie este potrivita pentru utilizarea in orice domeniu care se ocupa cu vanzari de produse, care doreste sa memoreze un istoric al vanzarilor



dar si a clientilor care au efectuat achizitii de la vanzatorul care utilizeaza aplicatia. Aplicatia mai este folositoare in cazul in care se doreste o metoda usor de folosit pentru a manageria o astfel de distributie de produse. Din aceasta tema am invatat despre folosirea claselor care implementeaza doar operatii CRUD, despre utilizarea functiilor care prelucreaza parametrii generici si unde sunt aceste functii folositoare, despre utilizarea tehnicii de reflexie care ne ajuta prelucrarea oricarui tip de obiecte, dar si despre metodele de dezvoltare Singleton care presupune utilizarea unei singure instante a unei anumite clase in tot programul, dar si cand nu este bine sa o folosim. Despre metoda de dezvoltare Layered Architecture care este des folosita in practica si care presupune foarte multe avantaje, cum ar fi independenta cat mai mare a claselor si modificarea foarte usoara a acestora cand este nevoie de imbunatatiri. De asemenea am invatat cum se face accesul la o baza de date cu ajutorul JDBC, cum se pot executa interogari direct dintr-un limbaj de programare si cum se pot prelucra datele primite. Ca si imbunatatiri aduse proiectului se pot adauga mai multe optiuni pentru executarea instructiunilor in baza de date, o interfata imbunatatita si mai usor de folosit, si permiterea selectiei dinamice si fara restrangeri a operatiilor dorite de catre utilizator.