DOCUMENTATIE

*TEMA 3*

Contents

[DOCUMENTATIE 1](#_Toc100742359)

[*TEMA 3* 1](#_Toc100742360)

[1. Obiectivul temei 2](#_Toc100742361)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 2](#_Toc100742362)

[3. Proiectare 2](#_Toc100742363)

[4. Implementare 3](#_Toc100742364)

[5. Rezultate 5](#_Toc100742365)

[6. Concluzii 5](#_Toc100742366)

# Obiectivul temei

Obiectivul temei este de a proiecta o aplicatie de managemnt al unui depozit cu clienti si produse . Obiectivele principale sunt folosirea tehnicilor de reflexie in vederea dezvoltarii unui cod reutilizabil . Obiective secundare in cadrul proiectui presupun lucrul cu o baza de date si crearea unei arhitecturi layered .

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Dupa rularea aplicatiei utilizatorul va trebui sa aleaga din cele trei optiuni prezentate : clienti , produse , comenzi . Daca utilizatorul va alege clienti acesta va putea vedea un tabel cu toti clienti curent inregistrati , si posibilitatea de a alege din trei optiuni : adaugare client , stergere client , si editare client . Daca se alege adaugare client va aparea o noua fereastra cu campuri pentru a insera numele , adresa , data nasterii si email necesare pentru a crea un nou client in baza de date . Pentru a edita sa sterge un client utilizatorul trebuie sa selecteze un rand din tabelul alaturat.Daca nu se efectueaza selectia unui client va aparea o eroare.Atunci cand un client este editat va aparea o fereastra similara cu cea de inserarea clientiilor insa cu diferenta ca datake sunt completate cu camnpurile respectiv informatiile stiute de catre client . Utilizatorul poate sa modifice oricare dintre randuri la alegerea iar dupa apasarea butonului de edit acestea vor fi modificate iar tabelul va afisa noile date modificate ale clientului. In cazul butonului de produse utilizatorulu are aceleasi optiuni ca si in cadrul ferestrei pentru clienti . Utilizatorul poate de asemenea sa introduca , stearga , si sa modifice produse . In cadrul feresteri de produse utilizatorul poate sa aleaga clientii pentru care sa se faca comanda . Alegerea clientiilor va afisa un tabel specializat pentru clientul respectiv fiind astfel un tabel interactiv . Dupa alegera clientului utilizatorul are optiunea de a alege un produs de a introduce cantitatea in cadrul fieldtextului . Dupa apasarea butonului sa va introduce produsul in cadrul tabelei de comenzi . Se poate alege si sterge o comanda deja exitstenta in cadrul tabelei de comenzi . Dupa apasarea butonului de comanda se va genera un fisier text continand Id – ul clientului Si produsele pe care acesta lea comandat si totalul de plata generat . Intefata ofera use – caseurile necesare pentru clienti produse si comenzi utilizand jtable pentru afisarea datelor stocate in baza de date . Daca un utilizator alege o cantitate prea mare fata de ceea ce este in stoc se va afisa un mesaj de eroare oferind informatia ca comanda este prea mare . Interafata ofera o simpla solutie pentru manipularea datelor in cadrul bazaei de date avand o mare importanta asupra partei de front end.

# Proiectare

Proiectarea temei sa bazeaza pe lucrul cu baza de date respectiv reflectie pentru generarea datelor , inserarea obiectelor , stergerea acestora precum si update - ul acestora in baza de date . Deoarece obiectele sunt generate din cadrul bazei de date la inserare respectiv stergere si update se creaza dummy object pentru conectiunea la baza de date . Din moment ce baza de date are chei primare obiectele vor fi identificate ca cu ajutorul campurilor de Id care se genereaza automat la fiecare inserarea in baza de date . Pentru conexiunea la baza de date se va folosi obiectul de Connection factory generat prin reflection astfel acesta avand o unica instantiere in cadrul proiectului .

Se foloseste layered arhitecture clasele fiind impartite in pachete respectiv create pentru functionalitatea fiecarei clase . Pachetul connection contine connectionfactory mentionat anterior . Pentru pachetul DAO contine abstractdao o clasa utilizata in a manipula date din cadrul bazai de date cu ajutorul metodelor de reflexie . Modelul este alcatuit din din clasele de Client ,Produs si Comanda ce sunt create in conformitate cu tabelele din cadrul bazei de date , respectiv avand aceleasi atribute precum coloanele tabelelor din baze de date . Pachetul view este format din 6 clase utilizate pentru interfata . Clasa selectview este clasa si interfata principala deoarece celelalte clase de Clienti\_frame , Produse\_frame si Comenzi\_frame se regasesc in cadrul acesteia ca campuri atribut . Celelalte 2 clase sunt clase utilizate pentru inserare si update fiind astfel clase mai utlizabile din moment ce pentru inserare putem folosi dummy object astfel campurile fiind din cadrul textfields fiind nule . Clasa controller de asemenea utilizeaza reflexie pentru a putea utiliza obiectul ca o unica instanta. Proiectul este alcatuit din 13 clase ce sunt starns legate unele de altele .

# Implementare

Clasa de client este utilzata pentru a modela obiecte specifice tabelului de clienti din cadrul bazei de date . Acesta clasa doar implementeaza settere si gettere precum si 3 constructori cu campuri de initializare diferite.

Clasa de produs este utilzata pentru a modela obiecte specifice tabelului de produs din cadrul bazei de date . Acesta clasa doar implementeaza settere si gettere precum si 2 constructori cu campuri de initializare diferite.

Clasa de produs este utilzata pentru a modela obiecte specifice tabelului de comanda din cadrul bazei de date . Acesta clasa doar implementeaza settere si gettere precum si 2 constructori cu campuri de initializare diferite.

Clasa AbstractDao este utilizata pentru a formata date din cadrul bazei de date si ale returna in cadrul interferi GUI prentru a putea fi vizualizate si manipulate

Metoda table\_get\_all este utilizata pentru a retuna toate randurile din cadrul tabelei ce este data ca paramentru string . Acesta utilizeaza conectionfactory pentru facilizarea operatiei de select asupra bazei de date.

Metoda insertQuery construieste string de inserarea o unui obiect pe baza paramentrului table.

Metoda deleteQuery construieste string de stergere o unui obiect pe baza paramentrului table si al unui Id care se va alege ulterior.

Metoda updateQuery construieste string de update o unui obiect pe baza paramentrului table si al unui Id care se va alege ulterior.

Metoda CosClient construieste string pentru a selecta toate comenzile unui client pe baza unui id.

Metoda deleteObject foloseste reflexie pentru a vizualiza toate campul de id al unui obiect si in folosete pentru a sterge instanta respectiva din tabel folosind stringul din deleteQuery.

Metoda insert foloseste reflexie pentru a insera obiecte in baza de date . Aceaste metoda se uita la campurile din cadrul obiectului si in functie de tipul acestora le pune in statement pentru indexul respectiv in stringul query din insertQuery .

Metoda update foloseste reflexie fiind asemanatoarea cu metoda de insert insa folosind campul id pentru a identifica instanta de update in baza de date.

Clasa controller utilizeaza reflexie prentru a construi o unica instanta . Acesta clasa face legatura dintre intefaata si model . Aceata are atribure fereasta principala de select . Un obiect abstractDao pentru a manipula date si atribut controller ca unica instrantiere.

Metoda bon genereaza un fisier text pe baza clientului selecta in cadrul interfetei . Aceasta metoda preia din tabela de comendi informatiile relevante si le scrie in fisier . La final se calculeaza totalul comenziilor si se scrie in fisier.

Metoda DeleteObject primeste tabela din care se sterge si un obiect dummy utilizat pentru stergere . Pe baza clasei obiectului se crea un dummy object utilizand date din cadrul tabelei si apeleaza functia de delete a lui abstractDao . Dupa stergere se updateaza tabela .

Metoda EditClient precum si EditProdus creaza un dummyobject respectiva clasei sale cu campurile introduse in cadrul interfetei la care utilizatorul se afla momentan si apeleaza metoda update din AbstractDao . Apoi updateaza tabela de clienti .

Metoda InsertComanda creaza un obiect comanda pe baza clientului selectat si produsului selectat din tabela acestea avand un id generat automat pentru fiecare client si apeleaza insert din AbstractDao . Tabela de comenzi se va actualiza dupa inserare .

Metoda insertClient creaza un obiect Client cu datele din cadrul interfetei si apeleaza insert din AbstractDao . Se updateaza tabela de clienti si disparea fereastra de inserare .

Metoda insertProdus creaza un obiect produs cu datele din cadrul interfetei si apeleaza insert din AbstractDao . Se updateaza tabela de produse si disparea fereastra de inserare .

Metoda getSingeController returneaza singura instanta a obiectului Controller.

Metoda buildCos construieste un model de tabela pe baza datelor oferite din result se in cadrul metodei respective din AbstractDao .Pentru fiecare coloana din result set acesta este adaugata in tabela .Pentru fiecare rand din resultset se creaza o lista de obiecte de tipuri generice ce sunt introduse in tabela. Paramentrul ClientId este clientul pentru care se cauta comenziile existente

Metoda buildList construieste un model pentru combolistul folosit pentru a selecta clienti la care se adauga comenzi . Fiecare instanta din lista este construita din Id , Nume , Prenume pentru fiecare client din cadrul unui result set ce selecteaza toate datele despre clienti din baza de date .

Metoda buildTable este metoda principala in cadrul contruirii tabelor din cadrul interfetelor . Se creaza un result set prin apelarea metodei build\_all din abstractDao ce returnezaa toate datele din tabela . Pe baza coloanelor returnate se adauga coloanele respectiva in tabela . Pentru fiecare rand din result set se creaza o lista de obiecte generice nesesare pentru a crea un rand in tabela din interfata . Metoda returneaza modelul de date al tablei ce a fost selectata .

Clasa Select\_Frame reprezinta clasa principala a intefetei din moment ce contine atribute din cadrul celorlalte 3 interferte de clienti ,produse si comenzi . Acesta clasa implementeaza doar settere si gettere pentru atribute .

Clasa Connerction factory creaza o instanta unica a clasei respective utilizand reflexie.

Metodele de close inchide conexiunile result setul si statemntul dupa ce acestea au fost uilizate.

Metoda create connection geneaza o conexiune noua pe baza parolei si userului din baza de date .

Clasele de edit\_add contin un dummy object in vederea de o oferi modulariare si flexibilitae in cadrul utilizari acestora . Pentru inserare se primeste un obiect null astfel datele din campurile de inserat vor fi initial nule asfel incat sa se poate insera complet de la inceput client . Daca obiectul dummy nu este null atunci se va edita un obiect in cadrul tabelei si a bazei de date .Campurile din cadrul interfetei nu vor mai fi nule si vor contine datele actuale ale clientului selectat astfel incat utilizatorul poate sa supra scrie aceste date pentru a updata astfel instanta in baza de date precum si tabela din interfata . Dupa apasarea butonului se va insera sau actualiza clientul sau produsul in cadrul bazei de date. Clasele de interfata contin gettere pentru obtine datele oferite de utilizator in cadrul interfetei astfel aceste fielduri si butoane sunt atribute in cadrul fiecarei clase de interfata.

# Rezultate

Se pot observa rezultatele operatiilor efectuate in cadrul intefetei in bazei de date la unde se vor crea updata si sterge noi instante de date prin dummy objects din cadrul aplicatiei java . O alta metoda de testare a efectuarii datelor poate fi obsevata prin fisierul text bon ce contine date destpre comneziile din cadrul interfetei si a bazei de date prin selectia in functie de clientii pentru care sa generat bonul respectiv .Bonul va contine la inceput idClientului apoi Produsele comandate cu cantitate si suma de plata a comnezi . La final se va regasi suma de plata totala pe baza comenziilor din tablea de comenzi a clientului respetiv .

# Concluzii

In cadrul acestei teme se impune a se lucra cu metode de reflection pentru a crea metode ce pot fi foate reutilizabile . Metodele de reflexie ofera o puternica metoda de a testa si modifica diferite diferite aspecte ale codului . De asmenenea se impune lucrul cu o baza de date folosind java database conection o unealta puternica in ca crea o stransa legatura intre partea de backend severe si pastrare de date si front end manipularea datelor si afisarea acestora preum si a oferi o unealta in a efectua operatiile intr – o maniera mai simpla .