DOCUMENTATIE

TEMA *3*

Management al comenzilor

NUME STUDENT: Chiorean Bogdan
GRUPA: 30223

CUPRINS

1. Obiectivul temei

Obiectivul temei este implementarea unei aplicatii de gestionare al unui depozit, folosind baza de date. Tematica aplicatiei mele este un depozit care gestioneaza un magazin online de arte plastice. Baza de date folosita este mySQL, iar structura proiectului foloseste arhitectura grupata pe urmatoarele pachete: Model, Bussiness Logic, Gui, DataAccess and Connection.

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

Aplicatia ofera o interfata usor de utilizat cu care se pot vizualiza si manipula datele legate de clienti, produse si comenzi. Afisarea si modificarile se fac direct in baza de date. Aplicatia este impartita in 3 parti pentru gestionarea fiecaruia dintre: clienti, produse si comenzi, iar la fiecare achizitie se genereaza o tranzactie salvata separat in baza de date. Intr-o comanda putem cumpara mai multe produse, iar pentru rezolvarea problemelor cauzate de relatia many-to-many intre comenzi si produse am creat o clasa Cart (si tabel in baza de date) intermediara care sa realizeze legatura dintre acestea.

3. Proiectare si implementare

Aplicatia este impartita in 5 pachete:

- *Connection:*

Contine clasa ConnectionFactory, care este una de tip Singleton, care realizeaza creare si inchiderea elementelor ce tin de conectarea si utilizarea bazei de date.

- *Data Access*

  Contine o clasa generica: AbstractDAO<T> ce creeaza operatiile CRUD pe un tip generic, folosind reflexia pe clasele mapate la tabelele bazei de date. In pachet se afla inca 4 clase ce mostenesc aceasta clasa generica pentru fiecare clasa ce mapeaza un tabel din baza de date.

- *Model*

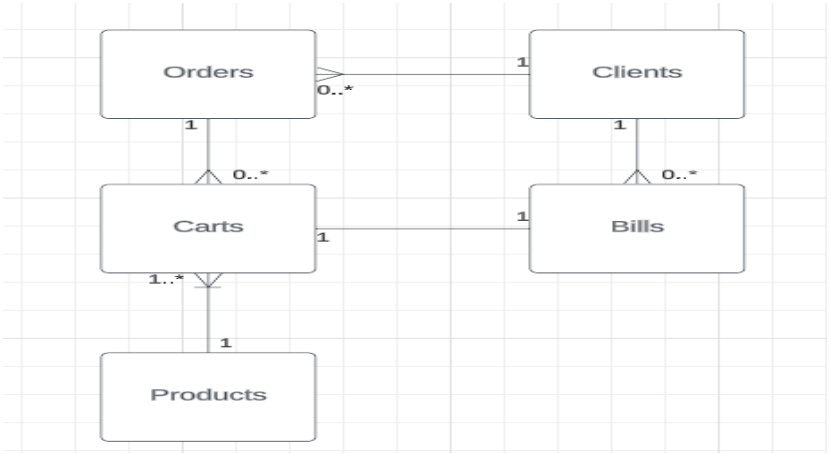
  Contine clasele ce mapeaza fiecare tabel din baza de date. Fiecare dintre acestea au atributele private, gettere,settere,metodele de comparare standard: equals si compareTo si au suprascrise metodele de hash si de transformare in string-uri. In Model, se afla si o clasa creata cu java record care reprezinta tranzactiile facute, in care se salveaza care clientul, numarul comenzi, produsul, cantitea produsului, pretul si data achizitiei. Aceasta clasa este imutabila si reprezinta un backup al comezilor.

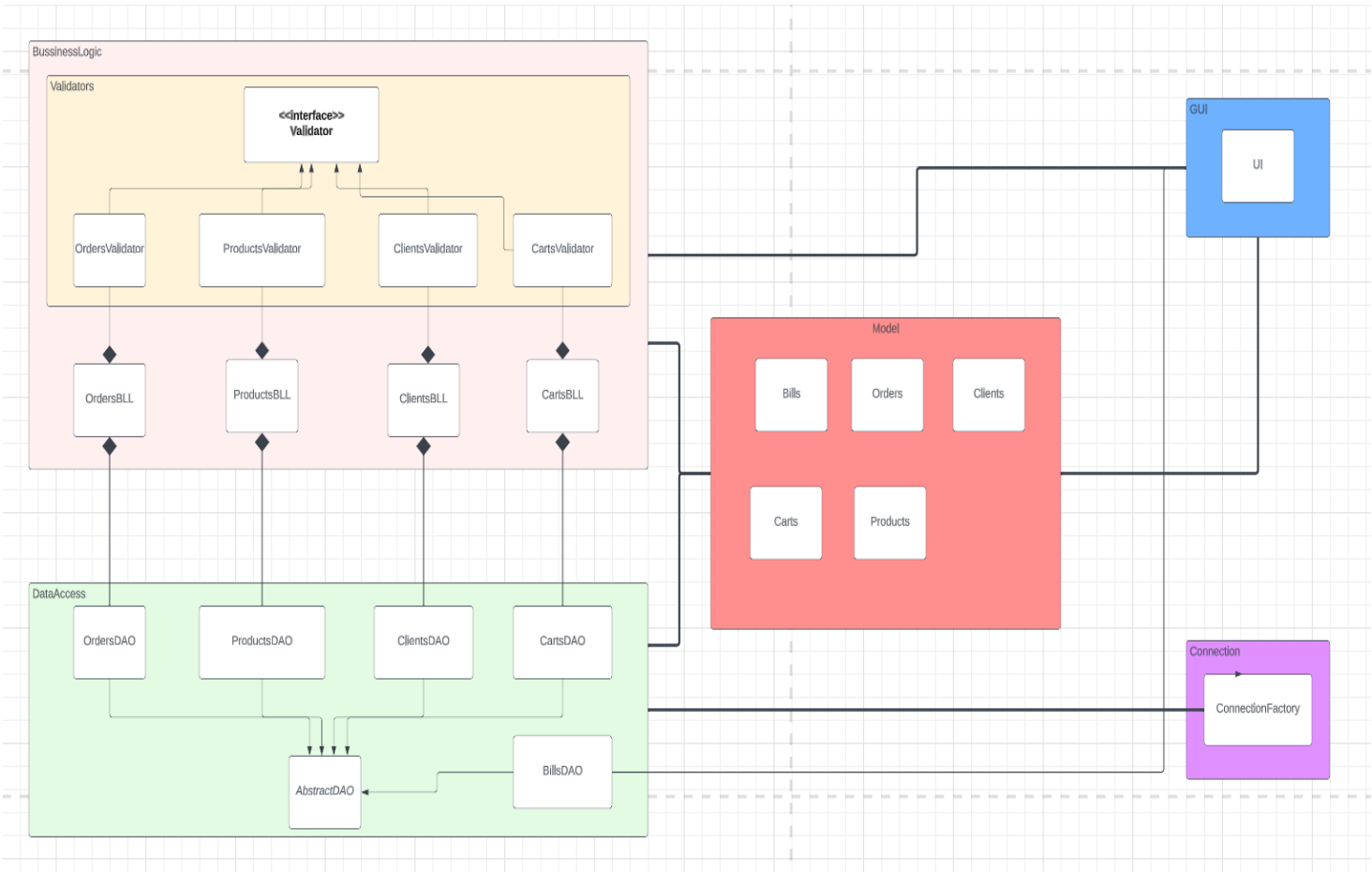
- *Business Logic*

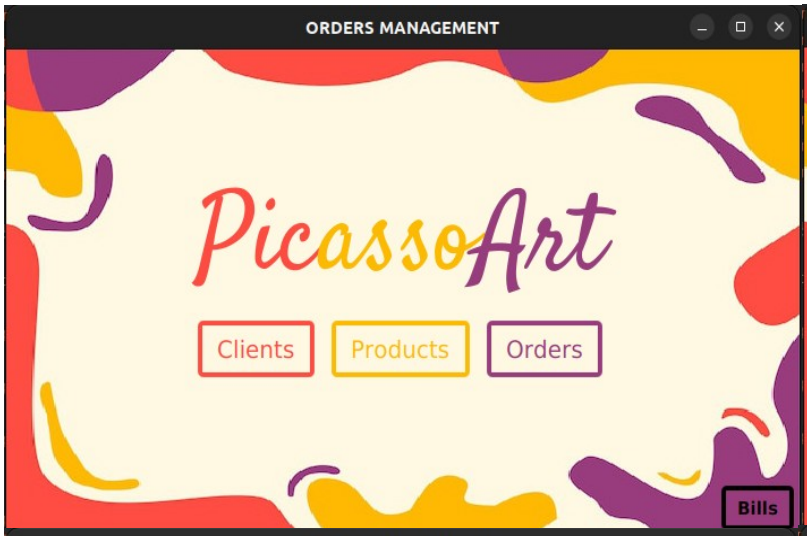
  Contine clasele care iau informatiile din interfata, le valideaza si in cazul in care sunt valide, apeleaza metodele din clasele de acces pentru a manipula datele. Acest pachet are un pachet propriu care contine o interfata de validator, si clase care implementeaza aceasta interfata pentru fiecare tip de obiect.

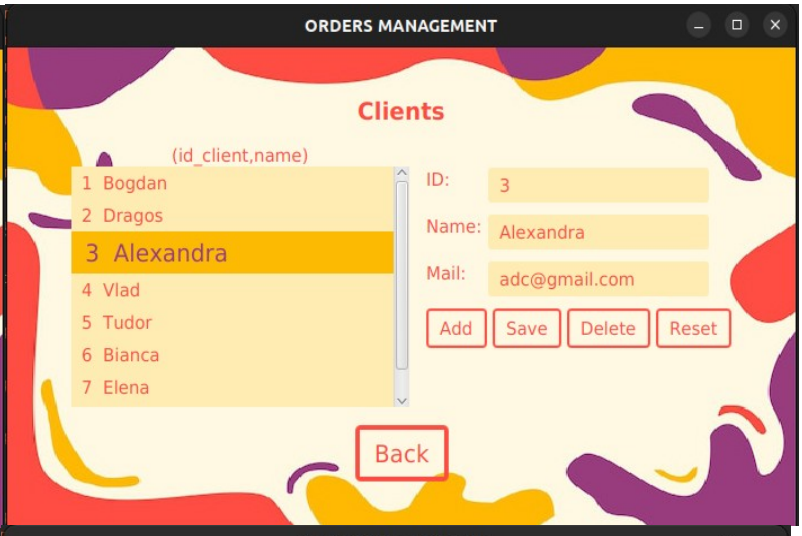
- *GUI*

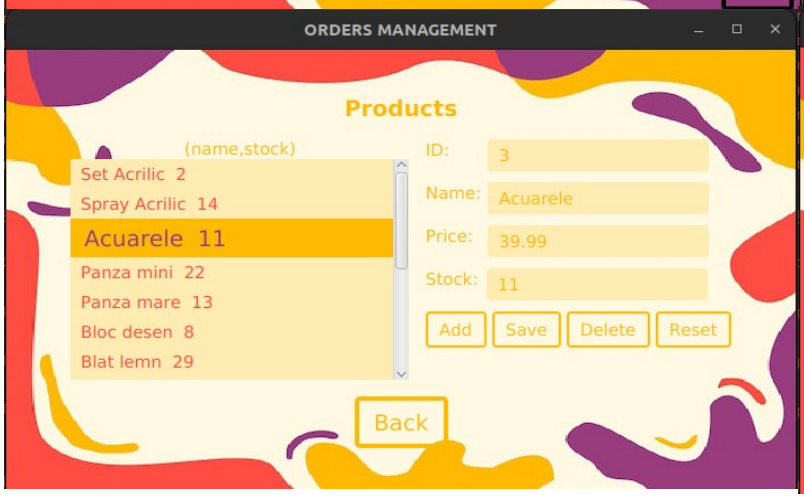
  Contine o clasa principala de user interface, o clasa care creeaza un combo-box cu search (luata de pe net) si un fisier de tip css. Interfata are 4 pagini, una principala din care se acceseaza celalalte 3 pagini pentru clienti,produse si comenzi si din care se poate descarca un fisier care sa contina toate tranzactiile facute.

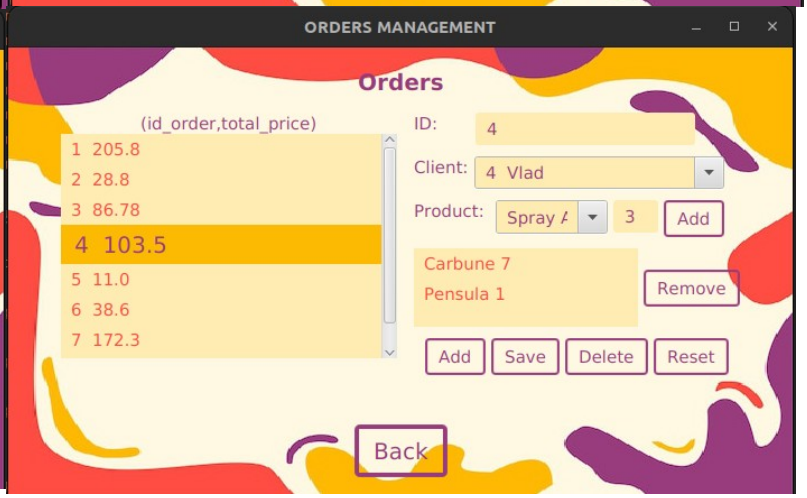












4. Rezultate

Testarea aplicatie se face din interfata, toate functionalitatile aplicatiei functioneaza.

5. Concluzii

In reazlizarea temei am invatat cum functioneaza tipul generic, cum se foloseste eficient, cum se foloseste reflexia pentru a afla detaliile unei clase, si construirea unor obiecte noi cu acestea. In plus, am folosit java records pentru a creea o clasa imutabila cu constructor si gettere generate automat si am creat conexiunea cu baza de date folosind un Singleton.

6. Bibliografie

Stackoverflow
Techgalery
Resursele din laborator