DOCUMENTATIE

TEMA *3*

NUME STUDENT: MARGINEAN TEODOR IOAN

GRUPA: 6

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

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

[3. Proiectare 4](#_Toc95297887)

[4. Implementare 5](#_Toc95297888)

[5. Rezultate 7](#_Toc95297889)

[6. Concluzii 8](#_Toc95297890)

[7. Bibliografie 8](#_Toc95297891)

# Obiectivul temei

Obiectivul principal al acestei teme este realizarea unui program care sa aiba o interfata pentru client pentru produs si pentru comanda. Fiecare interfata va avea functii complexe care vor trebuii implementate si folosite. Obiectivele secundare sunt:

* + Conectare la baza de date relizata cu ajutorul aplicatie MYSQL Workbanch;
  + Popularea bazei de date;
  + Realizarea interfetie initiale;
  + Conectare butoanelor prin intermediul controlerului;
  + Realizarea interfetei pentru client;
  + Realizarea interfetei pentru produs;
  + Realizarea interfetei pentru comanda;
  + Conectarea butoanelor prin intermediul controlerului;
  + Realizarea claselor din pachetul monome (tabelele din SQL);
  + Realizarea functiilor pentru Client in ClientDAO;
  + Realizarea functiilor pentru Produs in ProdusDAO;
  + Realizarea functiilor pentru Ordere in OrderDAO;
  + Realizare de validatori;
  + Realizare de clasa pentru conectarea la baza de date.

Clasele View ale proiectului sunt o interfata pentru realizarea obiectivelor proiectului. In proiect sunt cinci clase de tip view: View, ViewClient, ViewProduct, ViewOrder si ViewTabel. Clasa ViewTabel este folosita pentru doua functii de afisare atat pentru produse cat si pentru clienti.

Clasele Controller ale proiectului sunt create individual pentru fiecare buton, aceastea asigura functionalitatea proiectului. Sunt create individual pentru a fi mai usor de urmarit. In total sunt doisprezece clase fiecare controland un buton din diferite view-uri.

Exista o clasa de conectare la baza de date denumita ConnectionFactory, prin care se face legatura la baza de date. Aceasta necesita atat adresa, portul si parola pentru conectarea la serverul local hostat pe propriul calculator.

Exista clase validatoare care fiecare in parte reprezinta cate o constrangere care trebuie aplicata pe datele din tabel. Spre exemplu fiecare client trebuie sa aiba o varsta mai mare decat 18 ani. Alt exemplu folosindu-ne de regex verificam daca o adresa de mail este corecta.

Clasele de DAO cuprind implementari pentru query urile necesare realizarii cerintelor date.

Acestea sunt ClientDAO, OrderDAO si ProductDAO. Aceasta clase au in ele implementari pentru functii sql care vor fi necesare pentru realizarea proiectului.

Clasa de refelction este spatial de implementare pentru tehnica de programare de tip reflection. Care ne ajuta la generarea antetului pentru Jtabel in cazul afisarii din ViewTable care apare apelata in cadrul Client Select si Product Select.

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Sunt noua use-case-uri prezente in aceasta tema: Afisare Client, Editare Client, Stergere Client, Adaugare Client. Afisare Produs, Editare Produs, Stergere Produs, Adaugare Produs. Actorii cu care interactioneaza use-case-urile sunt in fiecare caz obiectele de tipul tabelelor definite inainte.

Primul use-case este Afisare Client in urma carauia se adauga intr-un tabel fiecare client cu datele respective acestuia. Se urmareste scrierea prin folosirea functie de reflection.

Al doilea use-case Editare Client care preia datele din textfield si in functie de id-ul preluat schimba in coloanal age varsta unui client cu ce preia din textfield-ul corespunzator valorii de introdus. Acest lucru se poate vedea prin apelarea Client Select sau prin acesarea SQL Workbench si logarea in server.

Al treilea use-case este Stergere Client care sterge clientul cu id-ul care este este preluat din textfield-ul corespunzator label-ului id. Acest lucru se poate vedea prin apelarea Client Select sau prin acesarea SQL Workbench si logarea in server.

Al patrulea use-case este Adaugare Client care adauga un client dummy cu valori dummy la id-ul preluat din text field. Acest lucru se poate vedea prin apelarea Client Select sau prin acesarea SQL Workbench si logarea in server.

Al cincelea use-case este Afisare Produs in urma carauia se adauga intr-un tabel fiecare client cu datele respective acestuia. Se urmareste scrierea prin folosirea functie de reflection.

Al saselea use-case Editare Produs care preia datele din textfield si in functie de id-ul preluat schimba in coloanal age varsta unui produs cu ce preia din textfield-ul corespunzator valorii de introdus. Acest lucru se poate vedea prin apelarea Product Select sau prin acesarea SQL Workbench si logarea in server.

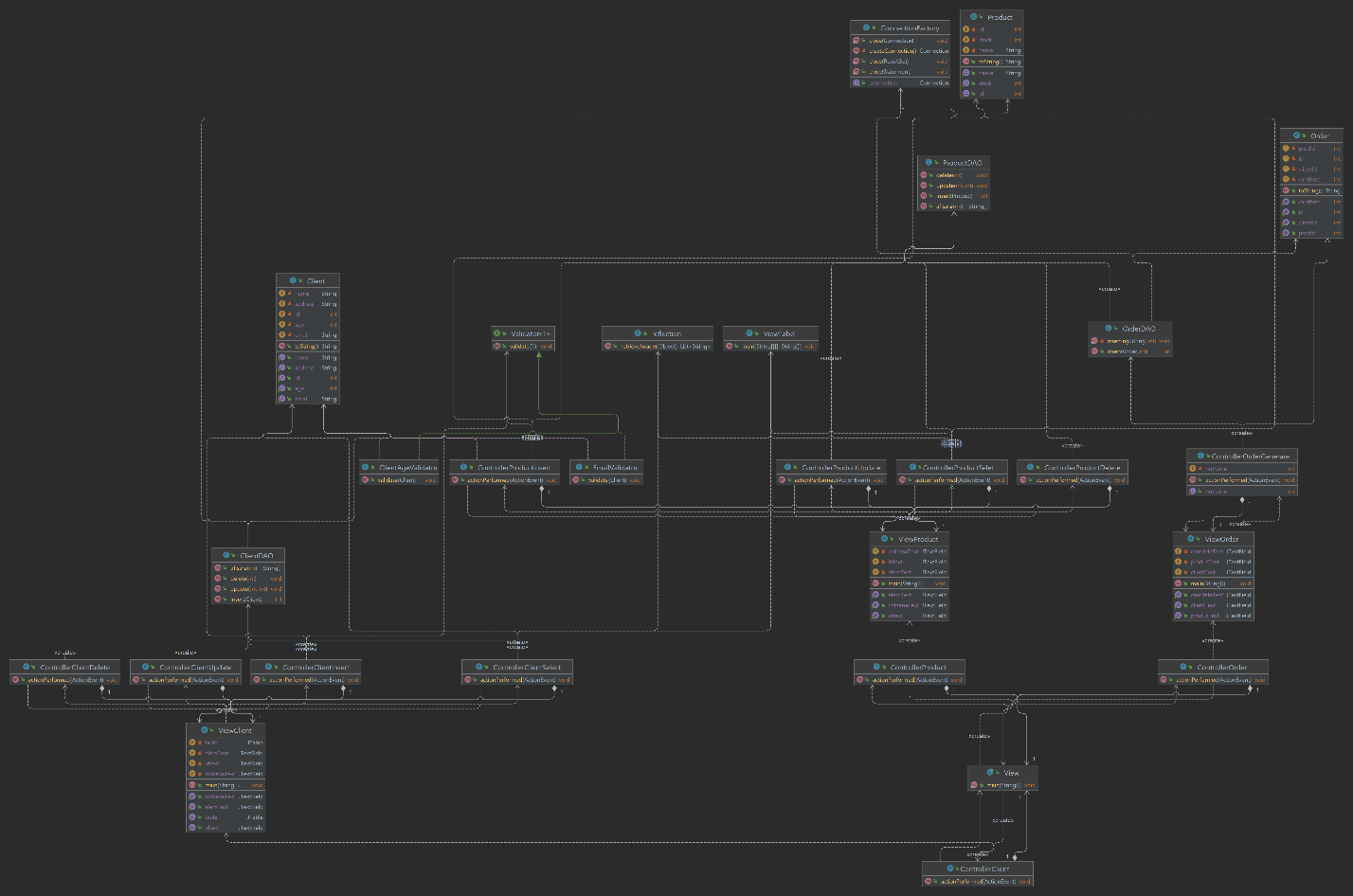
Al saptelea use-case este Stergere Produs care sterge produsl cu id-ul care este este preluat din textfield-ul corespunzator label-ului id. Acest lucru se poate vedea prin apelarea Product Select sau prin acesarea SQL Workbench si logarea in server.

Al optulea use-case este Adaugare Produs care adauga un produs dummy cu valori dummy la id-ul preluat din text field. Acest lucru se poate vedea prin apelarea Product Select sau prin acesarea SQL Workbench si logarea in server.

Al noualea use-case este Adaugare Order care adauga un nou order cu produsele curente si clientii curenti doar daca stock ul produslui este mai mare sau egal cu cantitatea din comanda. Se gereza un pdf cu datele comenzii. Acest lucru se poate vedea prin acesarea SQL Workbench si logarea in server.

# Proiectare

In cadrul acestei teme au fost folosite patru sase pachete denumite: model, presentation, dao, reflection, connection, bil.validators. Pachetul model are trei clase denumite Client, Product si Order. Pachetul presentation are 17 clase: View, ViewClient, ViewProduct, ViewOrder, ViewTabel, ControllerClient, ControllerProduct, ControllerOrder, ControllerClientDelete, ControllerClientInsert, ControllerClientSelect, ControllerClientUpdate, ControllerProductDelete, ControllerProductInsert, ControllerProductSelect, ControllerProductUpdate, ControllerOrderGenerare. Pachetul dao are trei clase ClientDAO, OrderDAO, ProductDAO. Pachetul refelction are o clasa: refelction. Pachetul connection are o clasa ConnectionFactory. Pachetul bil.validators are trei clase:EmailValidator, Client AgeValidator, Validator.



# Implementare

Proiectul este un proiect complex care are mai multe tehnici de programare prezente in acesta.

Tot proiectul pleca de la metoda main in clasa View din pachetul presentation. Aceasta instantiaza un gui care prezinta trei butoane. Fiecare buton are un controller instantiate in constructorul view-ului. Prin apasarea butonului Client se instantiaza un nou gui care prezinta mai multe textfield-uri care trebuie comeltate pentru a putea continu executia programului. Exista alte patru butoane care realizeaza operatii aspura tabelului Client din baza de date.

Prin apasarea butonului Add, se intra in metoda insert din clasa ClientDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerClientAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si adauga in tabelul Client o noua instanta de Client.

Prin apasarea butonului Delete, se intra in metoda delete din clasa ClientDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerClientAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si sterge din tabelul Client instanta de Client cu id-ul transmis ca parametru.

Prin apasarea butonului Edit, se intr in metoda update din clasa ClientDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerClientAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si modifica din tabelul Client instanta de Client cu id-ul egal cu id-ul trimis ca parametru si valoare de pe coloanal age este inlocuita cu valoara transmisa ca parametru.

Aceasta este functiomalitate asociata tabelului client. Astfel se satisfac toate cerintele care apar in preznetare pentru tabele Product.

Prin apasarea butonului Product se instantiaza un nou gui care prezinta mai multe textfield-uri care trebuie comeltate pentru a putea continu executia programului. Exista alte patru butoane care realizeaza operatii aspura tabelului Product din baza de date.

Prin apasarea butonului Select, se intr in metoda insert din clasa ClientDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerClientAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si returneaza toate datele referitaore la Client-ul cu id-ul transmis ca parametru. Aceasta metoda din ClientDAO se apeleaza de mai mutle ori pentru a fi preluate toate datele din tabel. Dupa preluarea datelor cu ajutorul metodei refelction se determina antetul tabelului si este instantiate o noau clasa de tip viewTabel in care se introduc intr-un JTable taote datele si afisat pe ecran.

Prin apasarea butonului Add, se intra in metoda insert din clasa ProductDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerProductAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si adauga in tabelul Product o noua instanta de Product.

Prin apasarea butonului Delete, se intra in metoda delete din clasa ProductDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerProductAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si sterge din tabelul Product instanta de Product cu id-ul transmis ca parametru.

Prin apasarea butonului Edit, se intr in metoda update din clasa ProductDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerProductAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si modifica din tabelul Product instanta de Product cu id-ul egal cu id-ul trimis ca parametru si valoare de pe coloanal age este inlocuita cu valoara transmisa ca parametru.

Prin apasarea butonului Select, se intr in metoda insert din clasa ProductDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerProductAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si returneaza toate datele referitaore la Product-ul cu id-ul transmis ca parametru. Aceasta metoda din ProductDAO se apeleaza de mai mutle ori pentru a fi preluate toate datele din tabel. Dupa preluarea datelor cu ajutorul metodei refelction se determina antetul tabelului si este instantiate o noau clasa de tip viewTabel in care se introduc intr-un JTable taote datele si afisat pe ecran.

Aceasta este functionalitatea programului asociata tabelului Product. Astfel se satisafac taote criterile din cerinta legate de tabelul Client.

Prin apasarea butonului Client se instantiaza un nou gui care prezinta mai multe textfield-uri care trebuie completate pentru a putea continu executia programului. Exista un butoan care realizeaza crearea unei noi comenzi.

Prin apasarea butonului Generare, se inta in metoda insert din clasa ProductDAO, in care se iau textul SQL se inlocuiesc elementele notate cu ? cu elementele transmise ca parametrii, elemente preluate in controllerProductAdaugare, functia continua executia si face legatura la baza de date prin intermediul clasei ConnectionFactory si adauga in tabelul ordered doar daca ceea ce introduce ca parametru este mai mica decat stocul din produs, pentru a putea compara cele doua se foloseste o apelare a metodei select din ProductDAO, care retureneaza sub forma de string toate datele necesare pentru produsul cu id-ul timis ca parametru. In urma verificarii daca rezultatul este pozitiv se peleaza o metoda de afisare care va genera cu ajutorul PDFWriter-ului un PDF care va contine clientul, prdusul si cantitatea care va fi solicitata.

Astfel se satisfac toate cerintele necesare pentru tabelul Order.

# Rezultate

Dupa fiecare operatie pe care am realizato cu ajutorul controlerlui si a serveruil SQL, se pot verifica rezultatele in aplicatia SQL Workbacnh folosindu-ne de tabelel create si un query de afisare total sau doar a cee ace ne intereseaza. O alta modalitate de afisare o reprezinta folosirea Afisarii din cadrul programului, care afiseaza intr-un JTable toate datele din tabele. Astfel se poate face o corectare rapida a fiecarei instructiuni in vederea sesizarii unei greseli( Nu se gaseste 😊).

# Concluzii

In concluzie in urma realizarii temei, am invatat sa folosim conversie de string prin intermediul regex, am revenit asupra conceptelor invatate si le-am reimprospatat, am refacut un model architectural de tip model controller view. Am invatat sa avem un proiect structurat. Am invatat sa facem ferestre embricate in view. Am invatat sa facem o legaura intre o baza de date si programul nostru prin intermediul programului SQL Workbench. In concluzie acest proiect a fost putin chelenging deoarece am avut un timp restrains dar totusi am reusit sa il rezolvam. A fost o experienta interesanta care m-a facut sa ma simt fericit la sfarsit.

# Bibliografie

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

<https://cadredidactice.ub.ro//sorinpopa/files/2018/10/L1_diagrame_use_case.pdf>

https://www.google.com/search?q=exemple+de+impartire+cu+polinoame&sxsrf=APq-WBuzcU7CenE1BgjH5DJjtDs7Ia6m9w:1646581625198&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiX9\_PG6rH2AhUghf0HHaJfATUQ\_AUoAXoECAEQAw&biw=870&bih=876&dpr=1.1#imgrc=YWvtVLDSkx6u8M