

DOCUMENTATIE

TEMA 3

NUME STUDENT: POROJAN MĂDĂLIN MARIAN
GRUPA: 30227

CUPRINS

1. Obiectivul temei.....	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	4
3. Proiectare	9
4. Implementare.....	14
5. Rezultate.....	14
6. Concluzii	19
7. Bibliografie	20

1. Obiectivul temei

Obiectivul principal:

Obiectivul principal presupune crearea unei aplicatii usor de utilizat, pentru gestionarea comenzilor clientilor dintr-un depozit.

Obiectivele secundare:

- Verificarea input-ului introdus pentru a asigura o legatură unu la unu, între ceea ce este introdus în casetele text, corespunzătoare parametrilor de intrare (insert, update, delete, order), și ceea ce se află în memoria aplicației;
- Utilizarea unui drop box menu, de unde poate fii selectata fiecare tabela separat, putand fii realizate usor operatii pe acestea.
- Modificarea clientilor doar prin selectarea acestora din tabelele puse la dispozitie.
- Interceptarea tuturor erorilor pentru a evita o situație în care programul se află într-o stare necunoscută;
- Informarea utilizatorului despre erori prin pop-up-uri si coduri de eroare care pot fi depanate mai departe consultând un manual specific pentru erori;
- Informarea utilizatorului cu privință la corectitudinea input-ului introdus de la tastatură prin pop-up-uri;

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

Analiza problemei:

Ideea principală în realizarea aplicației de gestionare este de a pune la dispoziție utilizatorului o interfață ușor de utilizat, prin comenzi și o modalitate de operare sugestivă. Interfața grafică dispune de ferestre specifice pentru fiecare operație de inserare, ștergere, modificare sau crearea unei comenzi, tabelele din baza de date fiind disponibile clientului ce utilizează aplicația.

Scenarii posibile:

1. Introducerea datelor în mod eronat, care ar putea face realizarea producției unei excepții la inserarea/ștergere/modificarea datelor în/din tabel:
 - Introducerea de numere negative în casetele destinate pentru stock/amount;
 - Introducerea de cifre în numele clientului/produsului;
 - Introducerea unui email eronat;
 - Se activează un pop-up ce semnalează că datele nu sunt introduse corect;
2. Neselectarea coloanei care urmează a fi modificată prin butonul de update:
 - Această problemă este verificată și semnalată printr-un pop-up ce indică selectarea unei coloane din tabela respectivă.
3. Introducerea parțială a unor date de input:
 - În acest caz, este activat un pop-up ce semnalează că, casetele de text respective sunt goale.

Modelare:

Aplicația propriu-zisă este formată dintr-o fereastră principală și mai multe ferestre secundare, specifice fîrcarei operații.

Fereastră principală:

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	18	0734856457	Tg. Mures	mircea.mih...

Este formată din mai multe butoane, în funcție de opțiunea aleasă din drop down box:

- Dacă sunt selectate tabelele Client și Product, există 3 butoane:
 - Unul pentru operația de inserare „Insert”;
 - Unul pentru operația de modificare „Update”;
 - Unul pentru operația de ștergere „Delete”;
- Dacă este selectată opțiunea „Order Management”:
 - Unul pentru operația de modificare „Update”;
 - Unul pentru operația de ștergere „Delete”;
 - Unul pentru operația de order „Order”;
 - Unul pentru a tipări factura „Print Bill”.

În cazul în care sunt selectate opțiunile de Client și Product atunci sunt disponibile operațiile doar pe tabelele respective iar în cazul în care este selectată operația de Order Management, sunt disponibile toate cele 3 tabele, dar se pot realiza operații de modificare și ștergere doar pe tabela de Order, în timp ce operația de „Order” este realizată pe toate cele 3 tabele.

Main menu

Select an option:
Product

Product Table

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	100
3	Scaun	5	380
4	Raft	30	56

Insert

Update

Delete

Main menu

Select an option:
Order Management

Client Table

idCii...	name	age	pho...	addr...	email
0	Mad...	21	0746...	Cluj...	mad...
1	Radu	20	0728...	Cluj...	radu...
2	Mircea	18	0734...	Tg. M...	mirc...

Update

Delete

Order

Print Bill

Product Table

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	100
3	Scaun	5	380
4	Raft	30	56

Order Table

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4

Cazuri de utilizare:

Înțial se alege opțiunea dorită cu ajutorul drop down box-ului, una din cele 3 (Client, Product, Order Management) iar pe urmă se alege o operație (Insert, Update, Delete, Order, Print Bill), în funcție de opțiunea aleasă:

Se presupune că se alege opțiunea Order Management, cu operația de order. Inițial trebuie să se aleagă câte o coloană din fiecare dintre cele două tabele, Client, Product, deci, se selectează clientul și produsul dorit:

Main menu

Select an option:
Order Management

Client Table

idCii...	name	age	pho...	addr...	email
0	Mad...	21	0746...	Cluj...	mad...
1	Radu	20	0728...	Cluj...	radu...
2	Mircea	18	0734...	Tg. M...	mirc...

Update

Delete

Order

Print Bill

Product Table

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	100
3	Scaun	5	380
4	Raft	30	56

Order Table

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4

Pe urma, se apasa pe butonul de Order, pentru a plasa comanda:

The screenshot shows the 'Main menu' window with three tables: 'Client Table', 'Product Table', and 'Order Table'. The 'Order' button is highlighted. An 'Input' dialog box is open, asking for an 'Amount'.

idCli...	name	age	pho...	addr...	email
0	Mad...	21	0746...	Cluj...	mad...
1	Radu	20	0728...	Cluj...	radu...
2	Mircea	18	0734...	Tg. M...	mirc...

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	100
3	Scaun	5	380
4	Raft	30	56

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4

Acum, se introduce cantitatea dorita de perne, dar aceasta trebuie sa fie cuprinsa intre [1, 100], conform stocului disponibil. Se introduce 20 si se apasa pe OK.

The screenshot shows the 'Main menu' window with the 'Order Table' updated. The new order entry is highlighted.

idCli...	name	age	pho...	addr...	email
0	Mad...	21	0746...	Cluj...	mad...
1	Radu	20	0728...	Cluj...	radu...
2	Mircea	18	0734...	Tg. M...	mirc...

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	80
3	Scaun	5	380
4	Raft	30	56

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4
4	2	2	20

Acum comanda a fost plasa cu succes si urmeaza a fii tiparita chitanta, prin apasarea butonului de Print Bill:

The screenshot shows the 'Main menu' window with the 'Print Bill' button highlighted. A 'SUCCESS' dialog box is open, indicating the bill was printed successfully.

idCli...	name	age	pho...	addr...	email
0	Mad...	21	0746...	Cluj...	mad...
1	Radu	20	0728...	Cluj...	radu...
2	Mircea	18	0734...	Tg. M...	mirc...

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	80
3	Scaun	5	380
4	Raft	30	56

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4
4	2	2	20

Printarea a avut loc cu succes si acum urmeaza a fii vizualizata chitanta din fisierul pdf, cu numele BILL_#4:

Order ID #4:

Client ID #2:

Name: Mircea | Age: 18 | Phone: 0734856457 | Address: Tg. Mures | Email: mircea.mihai@didatec.utcluj.ro

Product ID: 2 | Product Name: Pema | Price: 20 EUR | Amount: 20

Un alt caz de utilizare ar fii modificarea datelor unui client din tabelul de clienti:

Main menu

Select an option:
Client

Client Table

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	18	0734856457	Tg. Mures	mircea.mih...

Insert

Update

Delete

Se incepe prin selectarea optiunii „Client” din drop down menu iar pe urma se selecteaza clientul care se doreste a fii modificat. Acum se apasa butonul de „Update” pentru a modifica clientul:

Update Client

ID
2

Name:
Mircea

Age:
18

Phone number:
0734856457

Address:
Tg. Mures

Email:
i@didatec.utcluj.ro

Execute

În noua fereastră deschisă apar toate datele vechi ale clientului cu numele „Mircea”. Acum se modifică câmpurile „Age” și „Address”:

Acum se apasă pe butonul „Execute” pentru a fi realizată operația de modificare al clientului cu numele „Mircea”:

Astfel, clientul „Mircea” a fost modificat cu succes.

3. Proiectare

Implementarea OOP a aplicației propriu-zise constă în utilizarea modelului Layered Architecture (Business layer, Data Access Layer, Presentation Layer, model), pentru o proiectare concisă și o structurare a tuturor datelor în mod uniform.

1. Pachetul **bll** cuprinde patru clase și un pachet **validators**. Clasele sunt **ClientBLL**, **DatabaseBLL**, **OrderBLL**, **ProductBLL**.

- Aceste clase realizează defapt legătura dintre pachetul **controller** și operațiile pe baza de date propriu zisă. Concret, fiecare clasă de tipul BLL sunt formate din următoarele:
 - Metode:
 - Public Client **find**”NumeClasaModel”ById(int id);
 - Public List<Client> **findAll**”NumeClasaModel”());
 - Public void **insert**”NumeClasaModel”(„NumeClasaModel” „obiectClasaModel”);

- Public void **update**”NumeClasaModel” („NumeClasaModel”
„obiectClasaModel”);
 - Public void **delete**”NumeClasaModel”ById(int id);
2. Pachetul **connection** conține strict realizarea conexiunii dintre limbajul de dezvoltare Java și baza de date mySql.
3. Pachetul **dao** conține toate operațiile propriu-zise realizate pe baza de date. Acest pachet conține o clasă generică, implementată prin metoda reflectiei, pentru a nu avea secțiuni de cod duplicat care fac o acțiune similară pentru toate tabelele în parte. Acest pachet conține 5 clase.
- Clasa **AbstractDAO**, utilizată pentru realizarea tuturor operațiilor comune pe fiecare tabelă din baza de date.
 - Aceasta conține următoarele metode:
 - Public **AbstractDAO**();
 - Public void **setNextIdInTable**();
 - Public int **getNextIdInTable**();
 - Public String **createSelectQuery**(String field);
 - Private String **createInsertQuery**();
 - Private String **createUpdateQuery**();
 - Private String **createDeleteQuery**();
 - Public List<T> **findAll**();
 - Public T **findById**(int id);
 - Protected List<T> **createObjects**(ResultSet resultSet);
 - Public T **insert**(T t);
 - Public T **update**(T t);
 - Public boolean **deleteById**(int id);
 - Următoarele câmpuri private:
 - Private final Class<T> **type** – utilizat pentru a determina clasa fiecărei tabele;
 - Private static int **nextIdInTable** – utilizat pentru a găsi următoare poziție disponibilă din tabelă respectivă;
 - Clasele **ClientDAO**, **DatabaseDAO**, **OrderDAO** și **ProductDAO**, care implementează clasa **AbstractDAO**, fiecare având ca generic modelul cu același nume. Acestea sunt folosite în principiu pentru a suprascrie metode din **AbstractDAO**, în cazul în care se dorește o funcționalitate în plus pe lângă ceea ce este deja realizat în clasa superclasă.
 - Dintre acestea ies în evidență:
 - **OrderDAO**, utilizată pentru suprascrierea metodelor de deleteById, insert și update, pentru a modifica în același timp și stocul din tabelă de Product.
 - Aici se adaugă totodată și metodele: public void **updateStock**(int id, Operation operation, Order updatedOrder), și public String **createUpdateQuery**();
 - **DatabaseDAO**, utilizată pentru obținerea tuturor datelor din toate tabelele, folosind un join între toate tabelele. Aici este suprascrisă metoda findAll din superclasă pentru a se realiza findAll doar pentru tabelă rezultată din acel join.
4. Pachetul **model**, conține clasele **BillData**, **Client**, **Order**, **Product**, care reprezintă o interpretare a tabelor din baza de date, cu aceleași nume de câmpuri ca în tabelele din schema **warehousebd**. Toate clasele conțin accesoare și mutatoare.
5. Pachetul **presentation**, conține clasele **Controller**, **GUIOperations**, utilizată pentru a lega operațiile din **MainGUI** de operațiile din baza de date propriu-zisă, cât și clasele **InsertIntoClientGUI**, **InsertIntoProductGUI**, **MainGUI**, **UpdateClientGUI**, **UpdateOrderGUI** și **UpdateProductGUI**.
- Din clasa **GUIOperations** se remarcă următoarele metode:
 - public static void **updateObject**(Class classBLL, Class classModel, List<JTextField> textFields) throws Exception;
 - public static void **makeOrder**(List<JTable> tables);
 - public static void **deleteObject**(Class classBLL, JTable table);
 - public static Object **setObjectProperties**(Class classModel, List<JTextField> textFields);

- public static void **insertObject**(Class classBLL, Class classModel, List<JTextField> textFields);
- public static void **setTableProprieties**(DefaultTableModel tableModel, Object object);
- public static void **updateTableEntries**(DefaultTableModel tableModel, Class classBLL);

Diagrama UML a pachetelor:

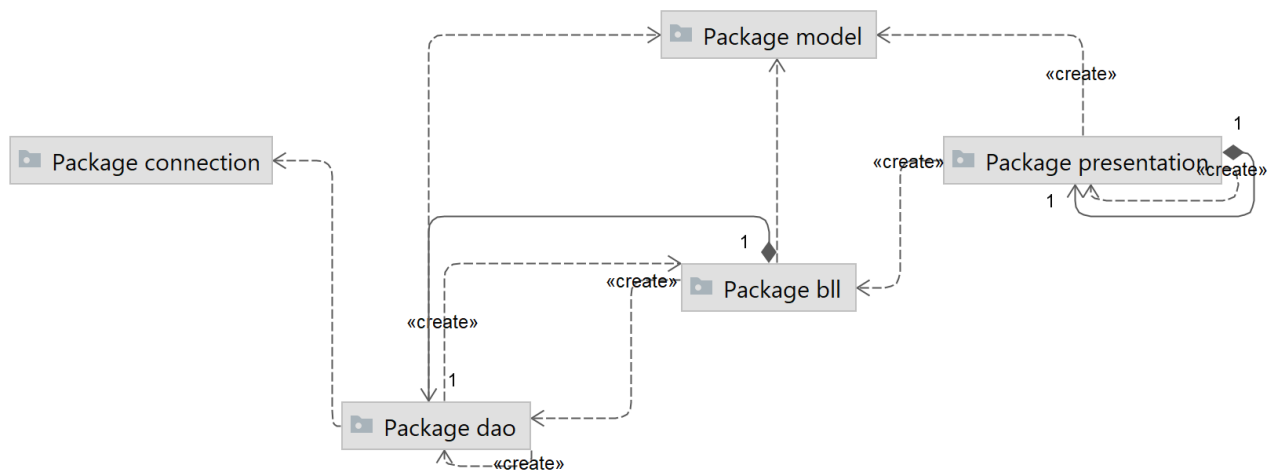
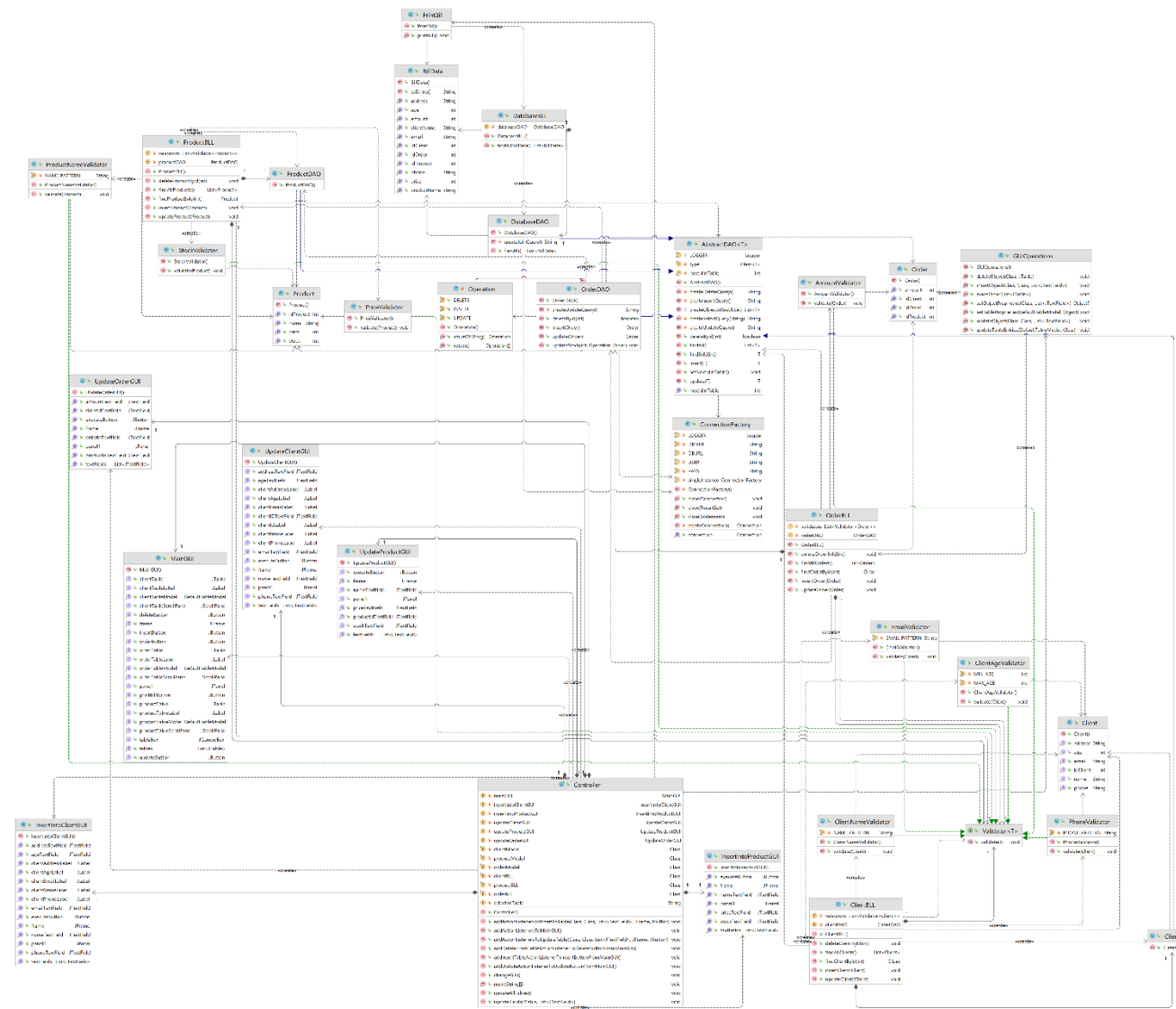


Diagrama UML a claselor:



4. Implementare

Descrierea metodelor importante din unele clase:

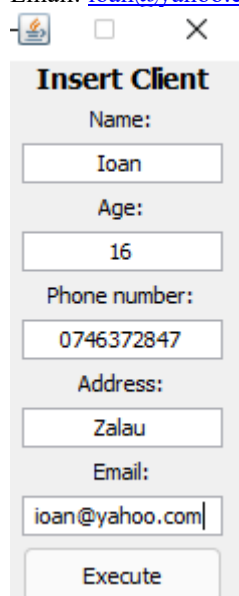
- În clasa **GUIOperations**:
 - Metoda **updateObject(...)** - este utilizata pentru a modifica un obiect generic creat prin intermediul interfeței grafice astfel: Utilizatorul apasa pe o coloana din tabelul respectiv iar datele din coloana selectata sunt transpuse in cod. Modificarea obiectului se face implicit dupa id-ul clientului/produsului/comenzii, prin transmiterea, prin intermediul reflectiei, a id-ului obiectului deja existent in baza de date.
 - Metoda **makeOrder(...)** – este utilizata pentru a insera o comanda in tabela Order astfel: Utilizatorul selecteaza o coloana din tabela Client si o coloana din tabela Product. Datele din aceste coloane sunt transmise mai departe la obiectul order din aceeasi metoda. Se creeaza obiectul conform cantitatii introduse de utilizator („amount”), celelalte date ale obiectului fiind deja cunoscute datorita selectiei anterioare. Se apeleaza insertOrder din Business Layer-ul corespunzator si se realizeaza inserarea in tabela.
 - Metoda **deleteObject(...)** – este utilizata pentru a sterge un obiect generic din tabela corespunzatoare a acestuia, data in urma alegerii optiunii de catre utilizator, din drop box („Client”, „Product” si „Order Management”). Aceasta metoda este scrisa cu ajutorul reflectiei. Se primeste o clasa la nivel de Business Layer si se cauta metoda de deleteById din clasa respectiva. Cand este gasita se apeleaza cu id-ul obiectului selectat de utilizator din interfata grafica.
 - Metoda **setObjectProprieties(...)** – este utilizata pentru a seta proprietatile unui obiect generic, in functie de clasa model data ca parametru cat si de datele introduse de catre utilizator in casetele text. Conform datelor introduse, se scrie in cadrul obiectului, instantiat prin metoda **newInstance()** din java.util.reflection, cu ajutorul metodei **writeMethod**.
 - Metoda **insertObject(...)** – este utilizata pentru a insera un obiect intr-o tabela respectiva, din baza de date, metoda este creata tot prin reflectie. Se incepe prin setarea proprietatilor un obiect de tipul clasei model iar pe urma se cauta metoda de insert din clasa de Business Layer corespunzatoare si se apeleaza cu obiectul creat.
 - Metoda **setTableProprieties(...)** – este utilizata pentru a popula datele din tabelele din GUI, cu datele din tabelele din baza de date. Se primeste un obiect corespunzator tablei si se adauga in tabela din GUI folosind reflectia.
 - Metoda **updateTableEntries(...)** – este utilizata pentru a actualiza datele tabelor din GUI. Aceasta actualizare se foloseste de metoda findAll, corespunzatoare tablei, care se invoca si returneaza toata datele actualizate necesare popularii tabelor.
- În clasa **AbstractDAO**:
 - Metoda **update(...)** – este o metoda generica utilizata pentru a transmite o instructiune de update catre baza de date pe coloana creata cu ajutorul obiectului t transmis ca parametru.
 - Metoda **deleteById(...)** – este o metoda generica utilizata pentru a transmite o instructiune de delete catre baza de date, asupra coloanei cu id-ul egal cu parametrul dat, din tabela respectiva. Initial se cauta daca obiectul exista in baza de date. Daca exista, se creeaza query-ul de stergere si este executat.

5. Rezultate

Pentru a testa aplicatia am folosit urmatoarele scenarii:

Primul test cuprinde:

- Inserarea in tabela Client;
 - Name: Ioan;
 - Age: 16;
 - Phone Number: 0746372847;
 - Address: Zalau;
 - Email: ioan@yahoo.com;



Insert Client

Name:

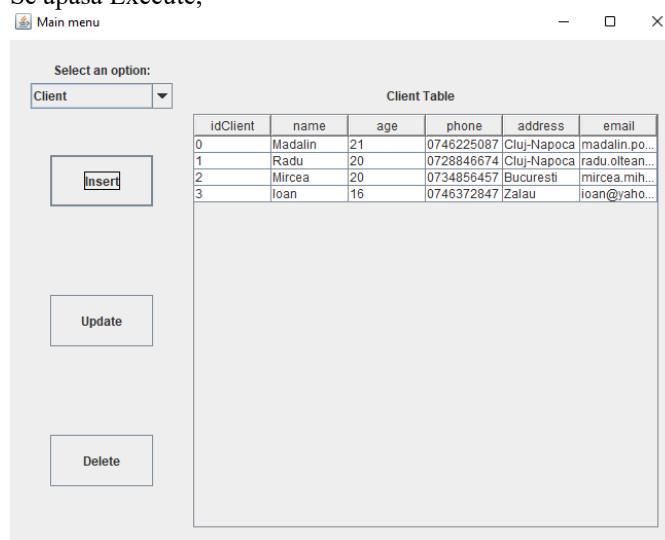
Age:

Phone number:

Address:

Email:

-
- Se apasa Execute;



Main menu

Select an option:

Client Table

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	20	0734856457	Bucuresti	mircea.mih...
3	Ioan	16	0746372847	Zalau	ioan@yahoo...

-
- S-a introdus cu succes;

Al doilea test cuprinde:

- Stergerea din tabela Client:

- Se selecteaza clientul cu numele Ioan:

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	20	0734856457	Bucuresti	mircea.mih...
3	Ioan	16	0746372847	Zalau	ioan@yahoo...

-
- Se apasa pe butonul de Delete:

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	20	0734856457	Bucuresti	mircea.mih...

-
- Stergerea s-a realizat cu succes;

Al treilea test cuprinde:

- Modificarea unui client:

- Se selecteaza clientul cu numele Radu:

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	20	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	20	0734856457	Bucuresti	mircea.mih...

-
- Se apasa pe butonul de update;

Update Client

ID

1

Name:

Radu

Age:

20

Phone number:

0728846674

Address:

Cluj-Napoca

Email:

oltean@yahoo.com

Execute

-
- Se modifica parametrul Age de la 20 la 22:

Update Client

ID

1

Name:

Radu

Age:

22

Phone number:

0728846674

Address:

Cluj-Napoca

Email:

oltean@yahoo.com

Execute

-
- Se apasa Execute:

Main menu

Select an option:
Client

Insert

Update

Delete

idClient	name	age	phone	address	email
0	Madalin	21	0746225087	Cluj-Napoca	madalin.po...
1	Radu	22	0728846674	Cluj-Napoca	radu.oltean...
2	Mircea	20	0734856457	Bucuresti	mircea.mih...

-
- Modificarea a avut loc cu succes.

Al patrulea test cuprinde:

- Crearea unei comenzi:
 - Se selecteaza clientul cu numele Radu si produsul cu numele Birou:

Main menu

Select an option:
Order Management

Update

Delete

Order

Print Bill

idCil.	name	age	pho.	addr.	email
0	Mad.	21	0746	Cluj-	mad.
1	Radu	22	0728	Cluj-	radu.
2	Mircea	20	0734	Bucu.	mirc.

idProduct	name	price	stock
0	Birou	100	48
1	Lampa	10	100
2	Perna	20	80
3	Scaun	5	380
4	Raft	30	56

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4
4	2	2	20

-
- Se apasa pe Order:

Input

Amount:

OK Cancel

-
- Se introduce 40 si se apasa pe OK:

Main menu

Select an option:
Order Management

Update

Delete

Order

Print Bill

idCil.	name	age	pho.	addr.	email
0	Mad.	21	0746	Cluj-	mad.
1	Radu	22	0728	Cluj-	radu.
2	Mircea	20	0734	Bucu.	mirc.

idProduct	name	price	stock
0	Birou	100	9
1	Lampa	10	100
2	Perna	20	80
3	Scaun	5	380
4	Raft	30	56

idOrder	idClient	idProduct	amount
0	0	0	2
1	1	2	100
2	2	3	20
3	2	4	4
4	2	2	20
5	1	0	40

-
- Crearea comenzii a avut loc cu succes.

6. Concluzii

În concluzie, această temă mi-a dezvoltat cunoștințele de OOP, învățând să lucrez cu tehnica reflectiei pe care o voi utiliza în viitoarele proiecte, dovedindu-se un aspect foarte important pentru gestionarea unei baze de date prin intermediul limbajului de dezvoltare Java. În același timp, mi-am dezvoltat abilitățile de structurare și depanare a codului, utilizând platforma Gitlab ce s-a dovedit extrem de folositoare în acest scop.

Ca dezvoltări ulterioare, se poate opta pentru o nouă interfață grafică, mai actuală și mai accesibilă, cu noi funcții, cum ar fi selectarea de produse multiple de către un client și realizarea unei chitanțe comune pentru toate produsele alese de acesta, etc.

7. Bibliografie

1. https://dsrl.eu/courses/pt/materials/A3_Support_Presentation.pdf
2. <https://www.thoughtco.com/using-java-naming-conventions-2034199>
3. <https://docs.oracle.com/javase/tutorial/uiswing/>
4. <https://itextpdf.com/en>
5. <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>
6. <https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>
7. <https://jenkov.com/tutorials/java-reflection/index.html>
8. <https://dzone.com/articles/layers-standard-enterprise>