

DOCUMENTATIE

TEMA 3

NUME : COJOCARU ANA-CATALINA
GRUPA : 30226

CUPRINS

1. Obiectivul temei.....	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3. Proiectare.....	4
4. Implementare	6
5. Rezultate	8
6. Concluzii	8
7. Bibliografie.....	9

1. Obiectivul temei

Aplicația de Management Comenzi este concepută pentru a procesa comenzile clienților într-un depozit. Pentru a stoca informațiile necesare despre produse, clienți și comenzi, trebuie utilizate baze de date. Pentru a obține un sistem ușor de întreținut, aplicația ar trebui să fie construită conform tiparelor de arhitectură stratificată, care împarte sistemul în straturi separate, fiecare cu o responsabilitate specifică.

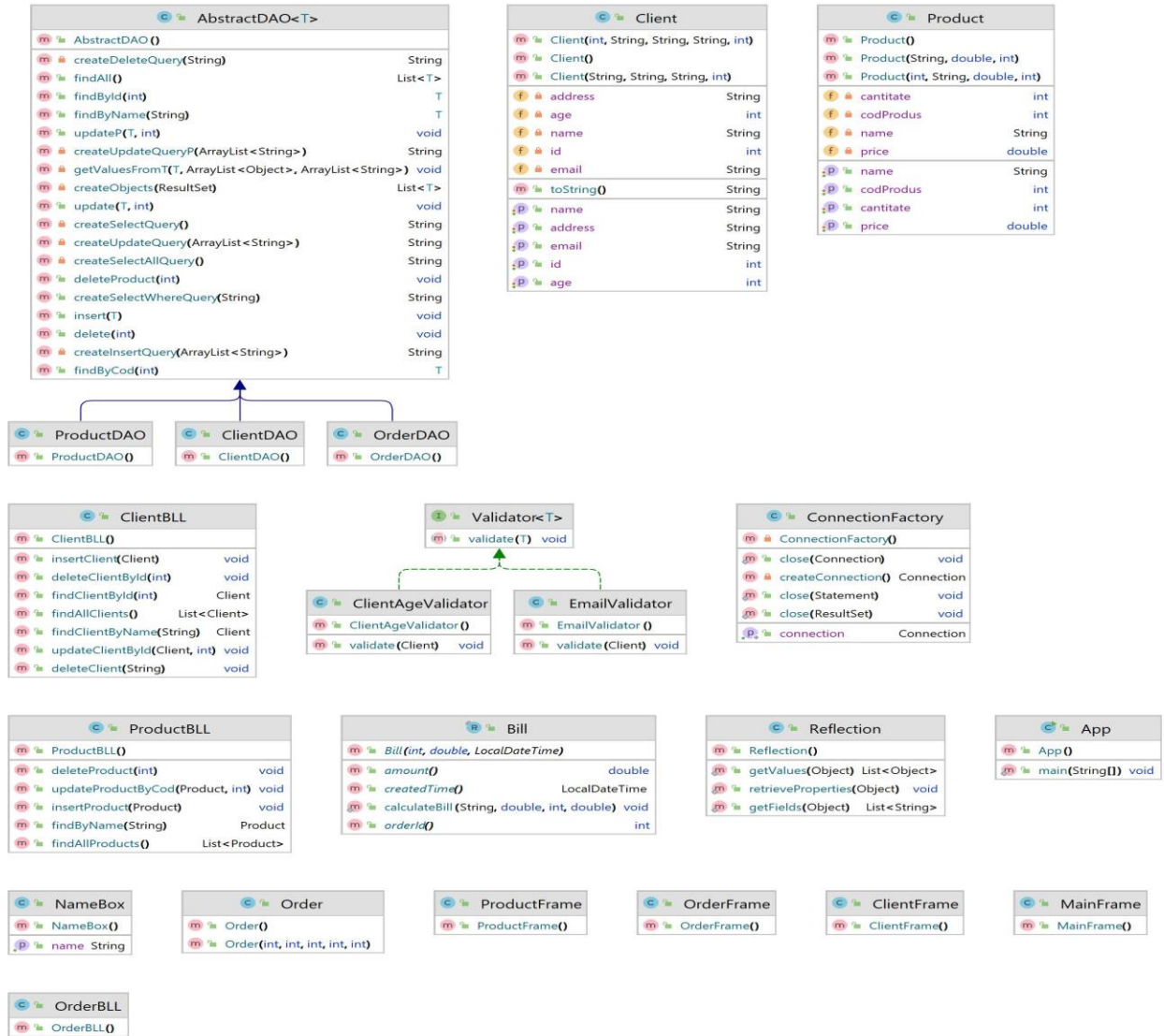
<i>Model</i>	<i>Cap. 3</i>
<i>Business Logic</i>	<i>Cap. 2</i>
<i>GUI</i>	<i>Cap. 4</i>
<i>Data Access</i>	<i>Cap. 4</i>

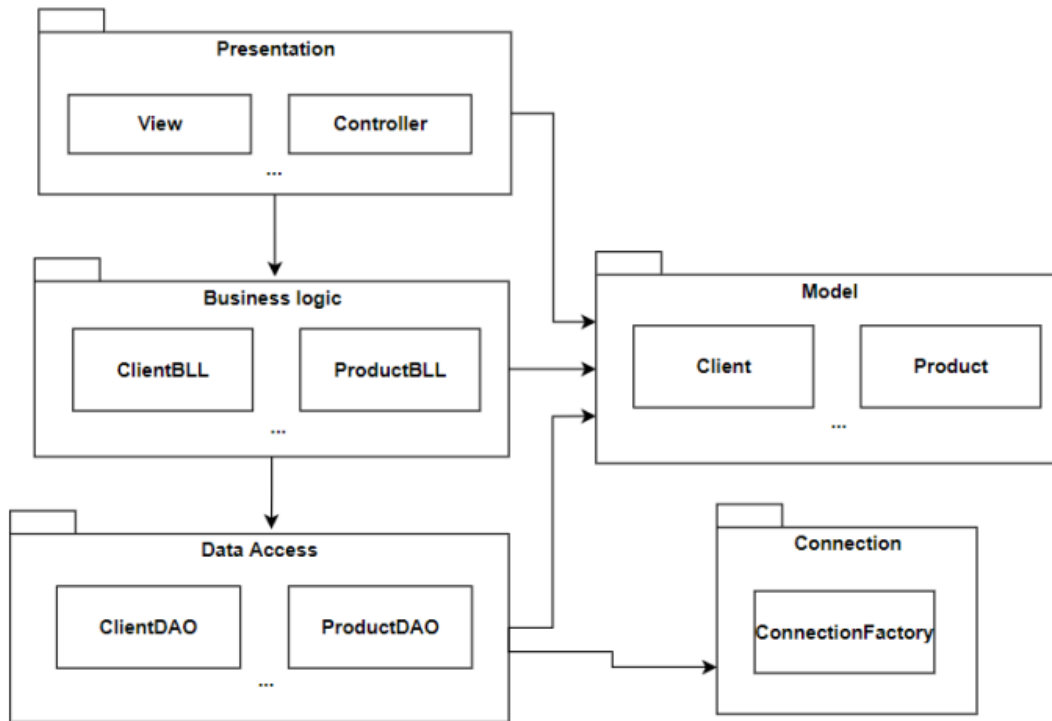
2. Analiza problemei, modelare, scenarii, cazuri de utilizare



Aplicatia e functionala in toate cazurile. In cazul in care la Order se introduce un stoc prea mare pentru cantitatea de produse existenta, se va afisa un mesaj: Under-stock.

3. Proiectare

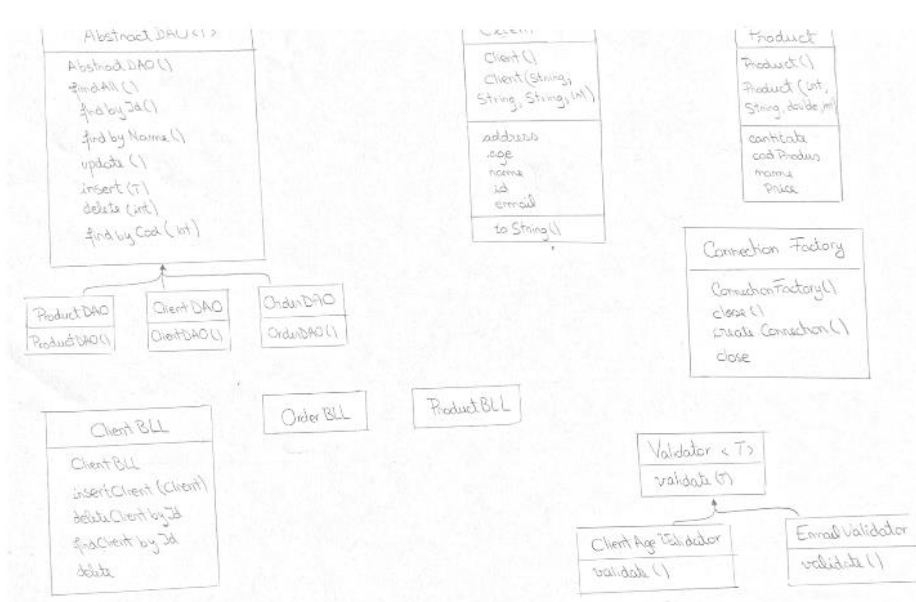




AbstractDAO
<code>createSelectAllQuery()</code>
<code>createSelectWhereQuery(String field)</code>
<code>createInsertQuery(ArrayList<String> field)</code>
<code>createUpdateQuery(ArrayList<String> field)</code>
<code>createUpdateQueryP(ArrayList<String> fields)</code>
<code>createDeleteQuery(String field)</code>
<code>findAll()</code>
<code>findById(int id)</code>
<code>findByCod(int id)</code>
<code>findByName(String name)</code>
<code>createObjects(ResultSet resultSet)</code>
<code>getValuesFromT()</code>
<code>insert(T t)</code>
<code>update(T t, int id)</code>
<code>delete(int id)</code>
<code>deleteProduct(int id)</code>
<code>updateP(T t, int codProdus)</code>

ProductBLL
ProductBLL()
DeleteProduct(int)
findAll()
findByName()
updateProductByCod()
findProductByCod()
insertProduct()

ClientBLL
ClientBLL()
DeleteClient(int)
findAll()
findByName()
updateProductById()
findProductById()
insertClient()



4. Implementare

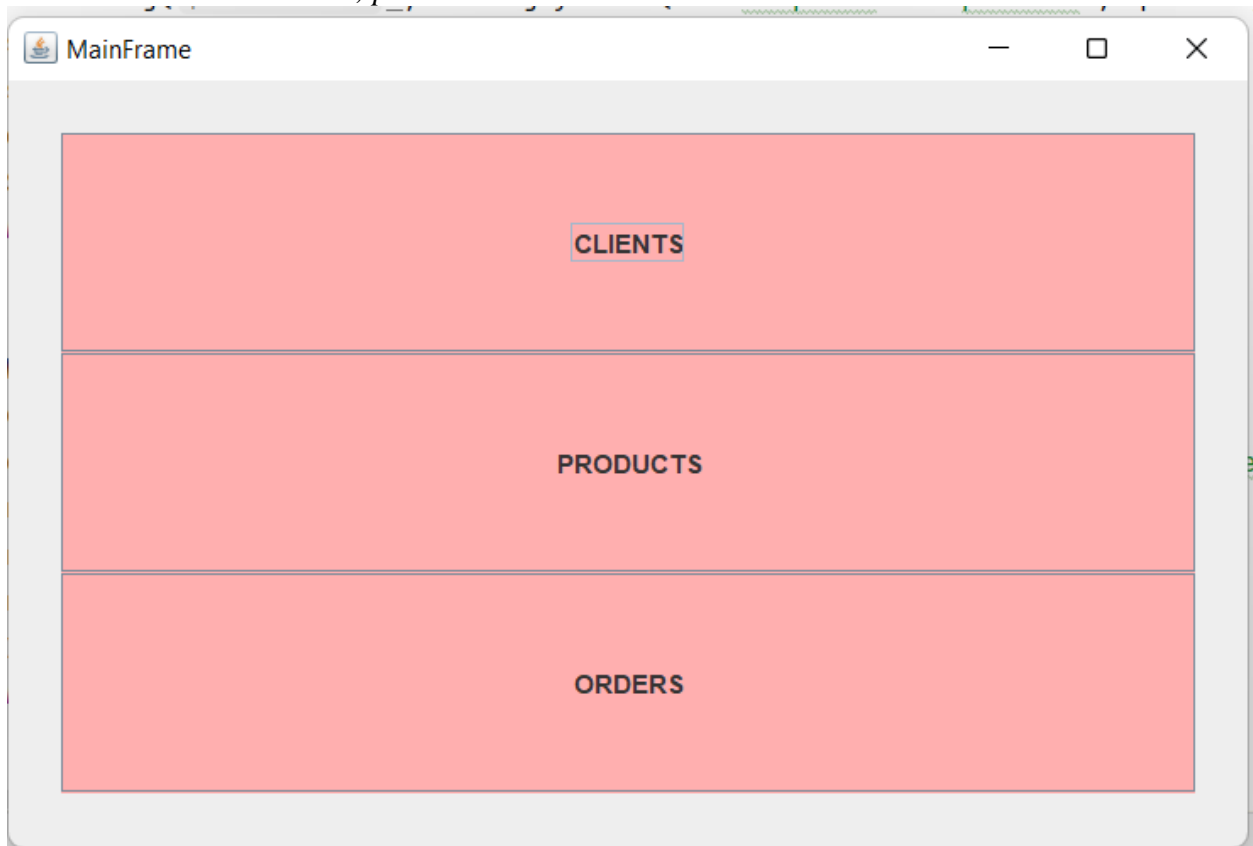
Primul strat, stratul Model, include clase care reprezintă modelele de date ale aplicației, cum ar fi Produs, Client și Order.

Cel de-al doilea strat, Business Logic, include clase care conțin logica aplicației, precum crearea și procesarea comenzilor.

Al treilea strat, Presentation, include clase care se ocupă de interfața cu utilizatorul, inclusiv crearea de interfețe grafice pentru utilizator (GUI).

Al patrulea strat, Data Access, include clase care conțin accesul la baza de date, cum ar fi clasa de conectare la baza de date și clasele care gestionează interogările și actualizările datelor în baza de date.

In interfata cu utilizatorul la inceput gasim 3 butoane: Clients, Products si Orders. Pentru Clients si Orders se deschide o noua fereastra unde putem insera, sterge, edita clienti, respective produse sau sa vedem intreaga tabela cu respectivele. De asemenea, putem plasa si comezi selectand numele clientului, produsul



ORDER

SELECT CLIENT

Popa Marcel

SELECT PRODUCT

cutite de bucatarie

INSERT QUANTITY

1

STATUS

VALID

VALIDARE

CONFIRMARE

View products

codProds	name	price	cantitate
1	cutite de bucatarie	35.0	49
2	linguri	20.0	15
3	furculite	20.0	4
4	boluri	15.0	63
5	esarfe	3.0	19
6	telefoane	999.0	20

5. Rezultate

Aplicatia e functionala pe toate datele, insertiile, stegerile, editarile si plasarile de produse functioneaza atat in cazul clientilor cat si in cel al produselor.

6. Concluzii

Am invatat sa lucrez cu o baza de date, sa fac conexiunile aferente, sa dezvolt o aplicatie apropiata de cele utilizate in toata lumea.

7. Bibliografie

*Se vor adauga referintele care au fost consultate de student pe parcursul implementarii temei .
Exemplu:*

1. *Bruce Eckel, Thinking in Java (4th Edition), Publisher: Prentice Hall PTR Upper Saddle River, NJ United States, ISBN: 978-0-13-187248-6 Published: 01 December 2005.*
2. *What are Java classes? - www.tutorialspoint.com*
3. *<https://dsrl.eu/courses/pt/>*
4. *https://users.utcluj.ro/~igiosan/teaching_poo.html*