

DOCUMENTATIE

TEMA *NUMARUL 3*

NUME STUDENT: Sabău Emanuela
GRUPA: 30224

CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3.	Proiectare	4
4.	Implementare	6
5.	Rezultate	Error! Bookmark not defined.
6.	Concluzii.....	7
7.	Bibliografie	8

1. Obiectivul temei

1.1. Obiectivul Principal

Am avut de proiectat și implementat o aplicație pentru gestionarea comenzilor clienților pentru un depozit.

1.2. Obiective secundare

- Analizat problema și identificat cerințele
- Proiectat aplicația de gestionare a comenzilor
- Implementat aplicația de gestionare a comenzilor
- Testat aplicația de gestionare a comenzilor

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

2.1. Cerinte functionale

Aplicația ar trebui să permită unui angajat să adauge un client nou.

Aplicația ar trebui să permită unui angajat să adauge un produs nou.

Aplicatia ar trebui sa permita unui angajat sa realizeze o comanda, selctand clientul si produsul pe care acesta doreste s ail achizitioneze.

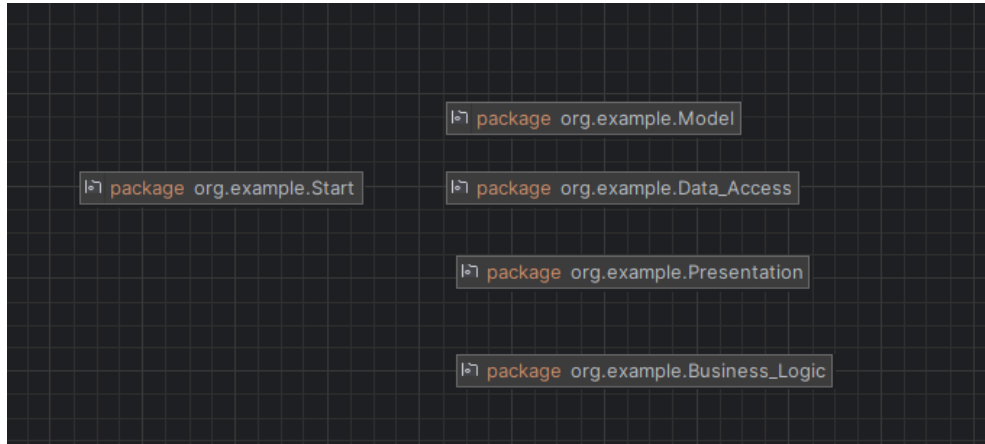
2.2. Cerinte non functionale

Aplicația ar trebui să fie intuitivă și ușor de utilizat de către utilizator.

Aplicatia ar trebui sa ajute la eficientizarea unei baze de date.

3. Proiectare

Am folosit o arhitectura structurata pe nivele, care va fi dezvoltata la punctul 4.



Odata rulat programul se vor deschide 3 ferestre: o fereastră pentru Clients, una pentru products, si alta pentru Orders.

Warehouse Management System				
Clients Products Orders				
Add Client Edit Client Delete Client				
id	name	email	address	phone
1	John Doe	123 Main St, Anytown, USA	john.doe@example.com	123-456-7890
2	Jane Smith	456 Oak St, Anytown, USA	jane.smith@example.com	456-789-0123

In cazul in care vrem sa adaugam un produs nou in baza de date va apare o noua fereastra in care vom putea completa campurile noului produs.

Clients

Products

Orders

Add Product

Edit Product

Delete Product

productId	name	description	price	stockQuantity
1	T-Shirt	Cotton T-Shirt	15.99	100
2	Jeans	Denim Jeans	29.99	50
3	Sneakers	Casual Sneakers	49.99	30

Add New Product

Name:

Description:

Price:

Quantity:

Add

In cazul ferestrei de Orders, se va putea realiza o comanda selectand clientul si produsul pe care acesta doreste sa il achizitioneze. Daca nu mai este pe stoc cantitatea aleasa de angajator, va apare o eroare, altfel se va decrementa numarul produselor pe stoc.

Clients

Products

Orders

Product:

Client:

Quantity:

Create Order

4. Implementare

4.1. BUSSINESS LOGIC LAYER

Clase: ClientBLL, ProductBLL, OrderBLL, validators.

Clasele sunt folosite pentru logica pentru operatiile pe care le realizam in aplicatie: add, edit si delete. Interfata validators este pentru validarea anumitor campuri, ca de exemplu, respectarea formatului unui email sau al unui numar de telefon.

4.2. DATA ACCES

Clase: CientDAO, ProductDAO, OrderDAO, ConnectionFactory

Aceasta ne permite accesarea bazei de date. Clasa ConnectionFactory este cea care face conexiunea cu baza de date, iar restul datelor acceseaza datele din tabelele bazei respective.

4.3. MODEL

Clase: Client, Product, Order

Acestea clase definesc structura pe care o are clientul, produsul si comanda.

4.4. PRESENTATION

Clase: View, Contrroller

Aceste clase sunt folosite pentru realizarea interfetei grafice si pentru functionalitatea ei.

4.5. START

Clase: Start, Reflecion

Clasa Reflection folosește tehnici de reflexie pentru a crea o metodă care primește o listă de obiecte și generează antetul tabelului prin extragerea, prin reflexie, a proprietăților obiectului și apoi populează tabelul cu valorile elementelor din listă.

5. Concluzii

Se vor prezenta concluziile, ce s-a invatat din tema, posibile de dezvoltari ulterioare.

6. Bibliografie

1. Simple layered project - gitlab.com
2. Reflection in Java - jenkov.com/tutorials
3. SQL dump file generation - dev.mysql.com
4. Javadoc - www.baeldung.com/javadoc