DOCUMENTATION

ASSIGNMENT 3

STUDENT NAME: Teodora Huluban
GROUP: 30222

CONTENTS

1. Assignment Objective 3

2. Problem Analysis, Modeling, Scenarios, Use Cases 3

3. Design 4

4. Implementation 6

5. Results 9

6. Conclusions 9

7. Bibliography 9

1.Assignment Objective

The main objective is to implement an application for managing the client orders for a warehouse.

The following sub-objectives were acquired to achieve the main objective:

- Analyze the problem and identify requirements

  - an analyze on the problem was done, and requirements were identified (detailed at point "2. Problem Analysis…")

- Design the application

  - a package/class diagram was realized for designing the project (detailed at point "3.Design")

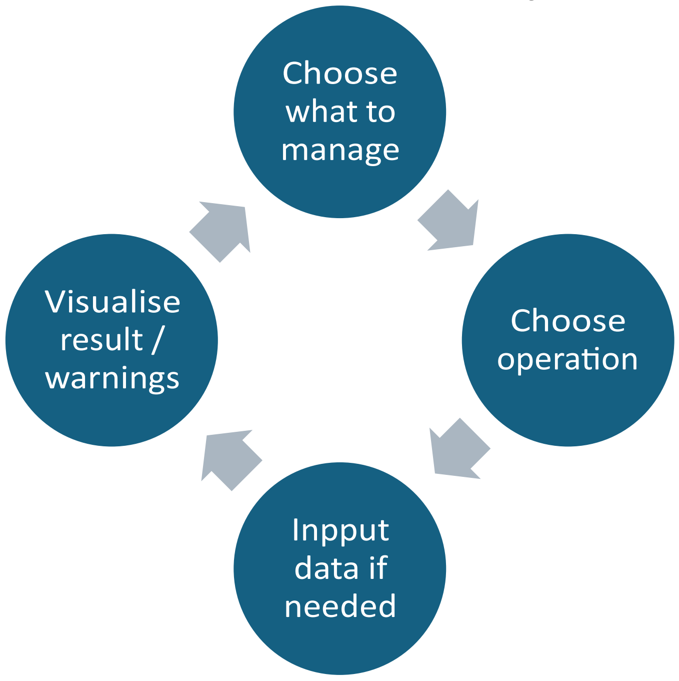
- Implement the application

  - the application was structured in 4 packages and 19 classes (detailed at point "4.Implementation")

- Test the application - several tests were done (detailed at point "5. Results")

2. Problem Analysis, Modeling, Scenarios, Use Cases

The application should manage a database for a warehouse that stores products, keeps record of the clients, and makes orders for them. The interaction with the database will be made trough a graphical interface that lets the user choose between managing the products, the clients, or the orders. For the products and clients, the following operations should be available: add new product/client, edit product/client, delete product/client, view all products/clients in a table. To make an order, the user can introduce a client and a product (by id), and a quantity for the product, than confirm the order trough a button. The order will be saved in the database and a bill will be generated.

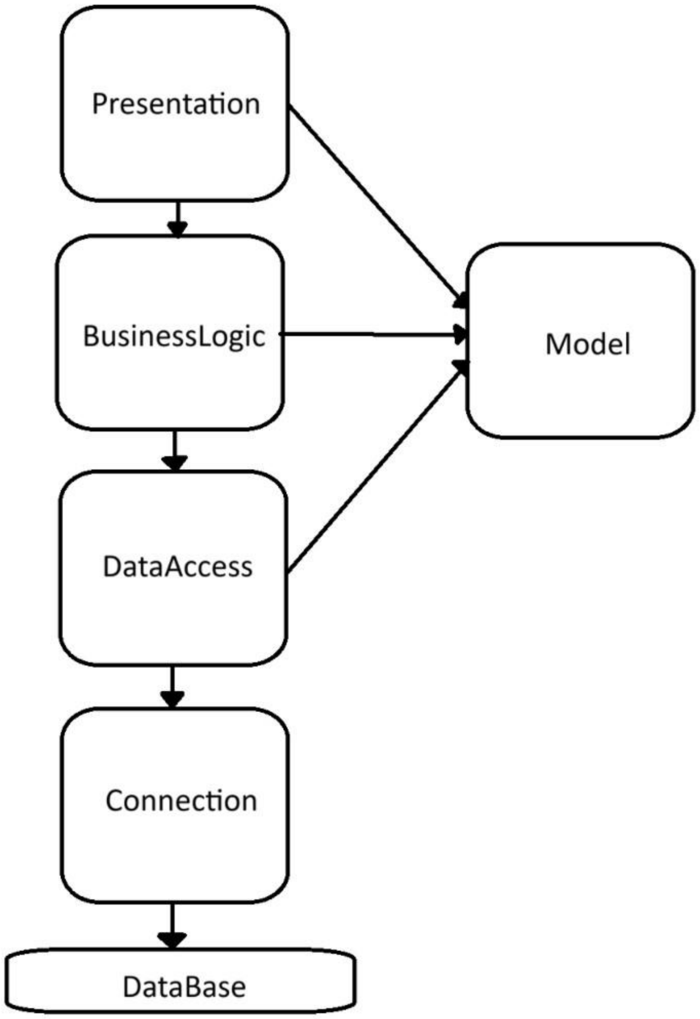


3.Design

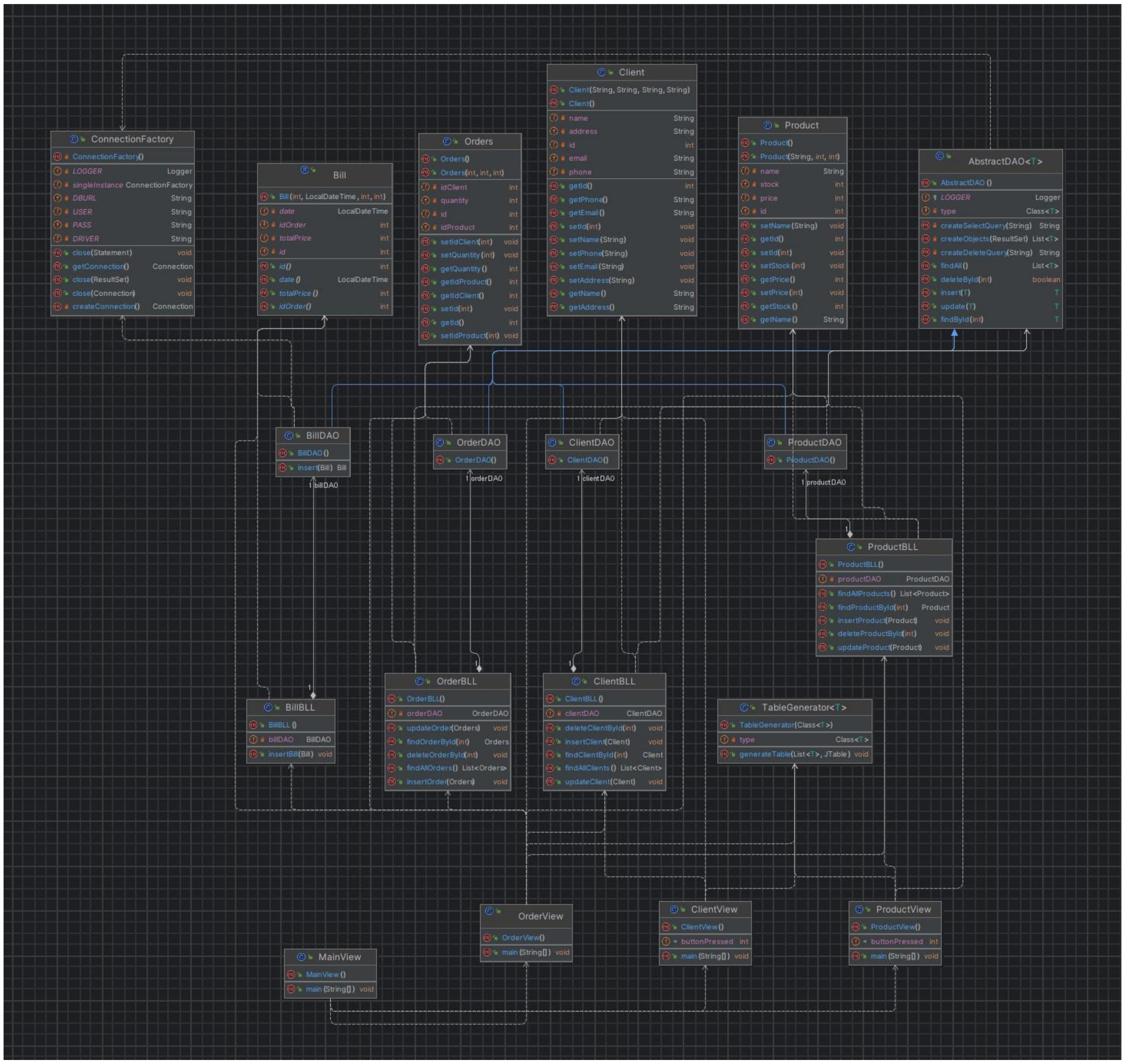
The project was structured in four packages:

- Model package containing the classes that model the client, product, order, and bill
- Presentation package containing the MainView, ClientView, ProductView, OrderView and TableGenerator classes, managing the graphical interface
- BusinessLogic package containing the ClientBLL, ProductBLL, OrderBLL, BillBLL classes that manages the logic of the application
- DataAccess package containing the AbstractDAO, ClientDAO, ProductDAO, OrderDAO, BillDAO classes, which access the data from the database
- Connection package containing the ConnectionFactory class that establishes connection between the java application and the database

Package Diagram:



Class Diagram:



4.Implementation

Client Class:
- This class models the client that has an id, name, email, phone, and address.

Product Class:
- This class models the product which has an id, name, stock, and price.

Orders Class:
- This class models the order which has an id, idClient, idProduct, and quantity.

Bill Class:
- This immutable class models the bill that has an id, date, idOrder, and totalPrice.

AbstractDAO Class:
- This class is a generic class which creates CRUD queries to manipulate the database

ClientDAO Class, ProductDAO Class, OrderDAO Class:
- The classes extend the AbstractDAO class

BillDAO Class:
- The class extends the AbstractDAO class and overrides the insert method, such that the id will not be updated

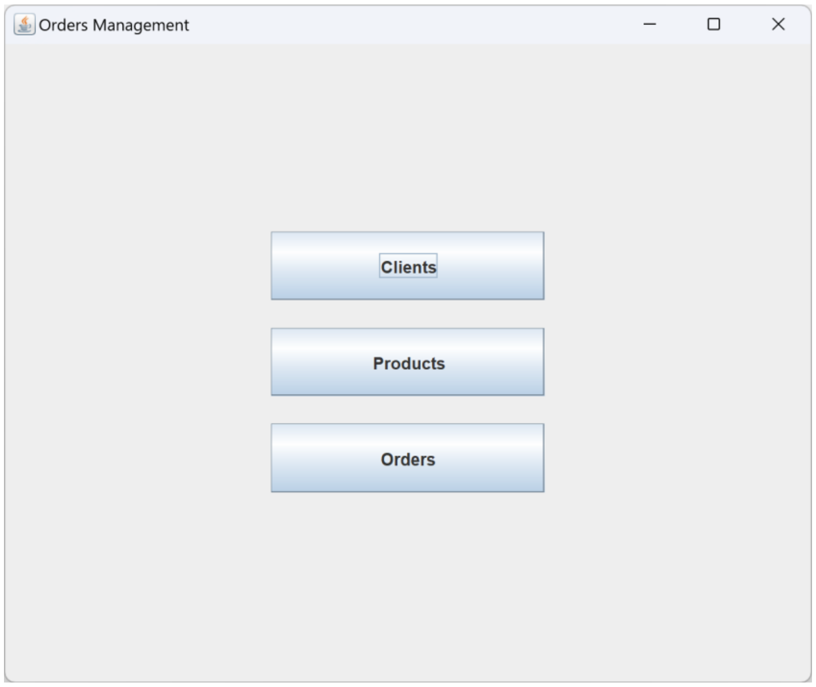
ClientBLL Class, ProductBLL Class, OrderBLL Class, BillBLL Class:
- The classes manage and call the methods from the data access classes

ConnectionFactory Class:
- This class establishes the connection between the java application and the database and provides means of closing the connection, the statement, and the resultSet

TableGenerator Class:
- This class is a generic class that uses reflection to create a model for a table based on the fields of a class specified through the generic parameter

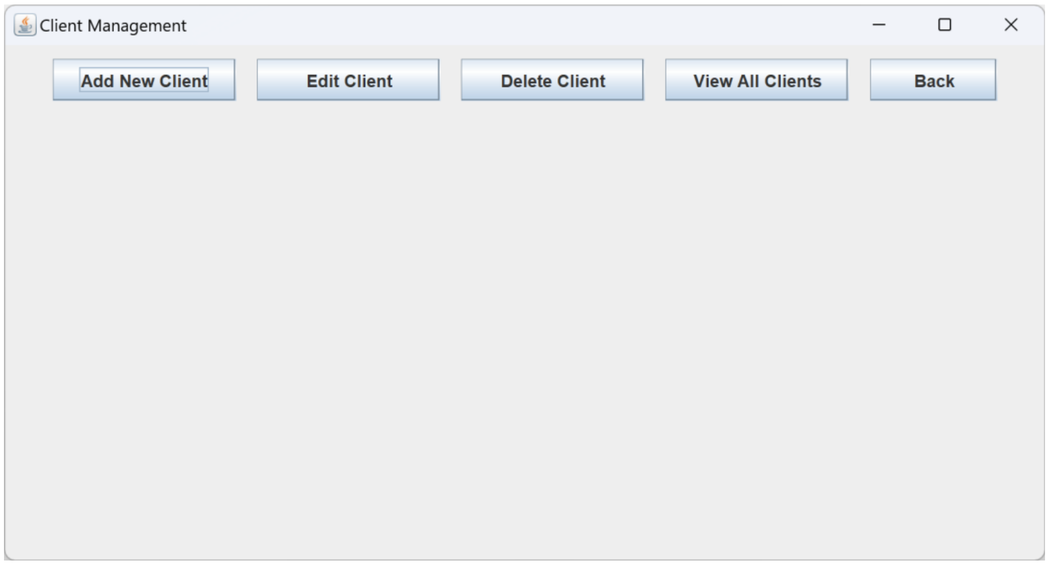
MainView Class:

- This class represents the main panel that has three options for managing clients, products, or orders



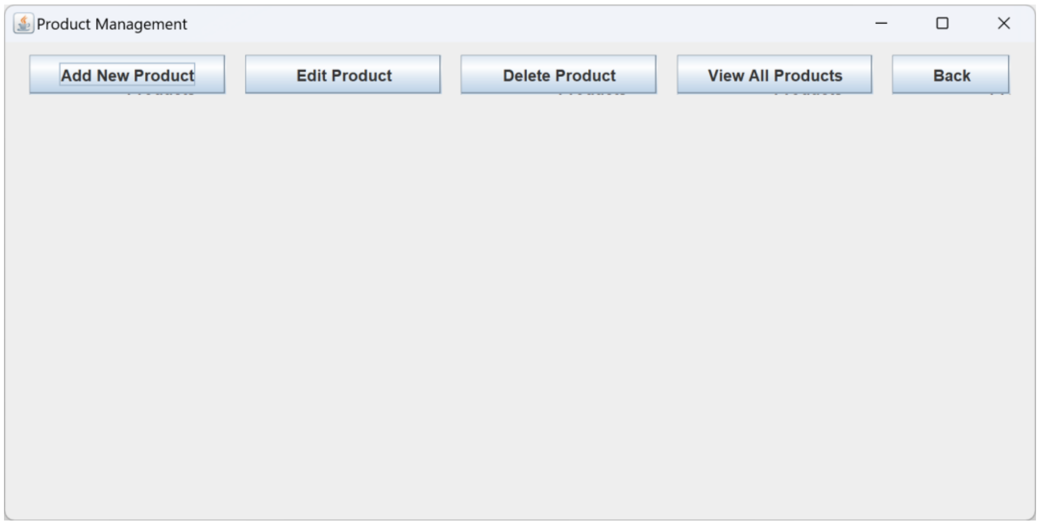
ClientView Class:

- This class represents the client-managing panel that has four options: add new client, edit client, delete client, view all clients in a table



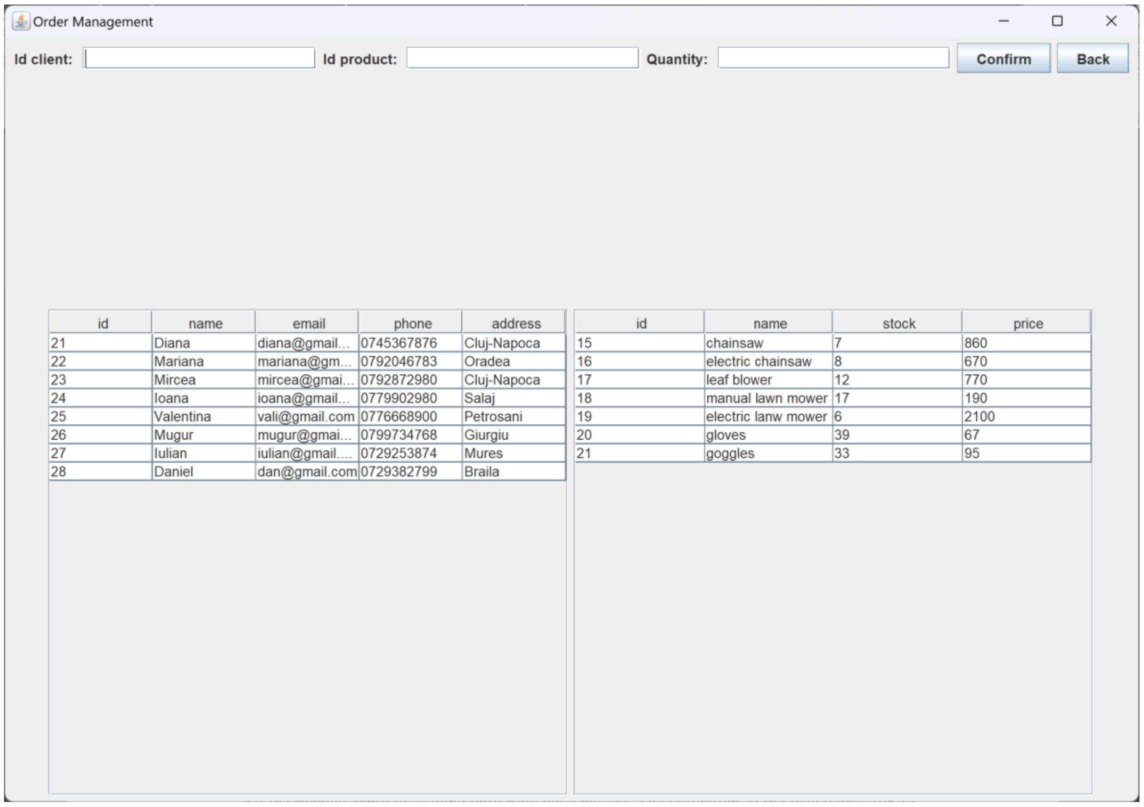
ProductView Class:

- This class represents the product-managing panel that has four options: add new product, edit product, delete product, view all products in a table



OrderView Class:

- This class represents the order-managing panel where the user can choose from the tables an existing client and product, and a quantity for that product to make an order



id	name	email	phone	address
21	Diana	diana@gmail...	0745367876	Cluj-Napoca
22	Mariana	mariana@gm...	0792046783	Oradea
23	Mircea	mircea@gmai...	0792872980	Cluj-Napoca
24	Ioana	ioana@gmail...	0779902980	Salaj
25	Valentina	vali@gmail.com	0776668900	Petrosani
26	Mugur	mugur@gmai...	0799734768	Giurgiu
27	Iulian	iulian@gmail....	0729253874	Mures
28	Daniel	dan@gmail.com	0729382799	Braila

id	name	stock	price
15	chainsaw	7	860
16	electric chainsaw	8	670
17	leaf blower	12	770
18	manual lawn mower	17	190
19	electric lanw mower	6	2100
20	gloves	39	67
21	goggles	33	95

5. Results

All the manual-input tests done until now have passed. The validation of the data works for all tested cases of inputting the wrong data. All the use cases have been tested and application works as expected.

6. Conclusions

In conclusion, the application for managing client orders in a warehouse has successfully achieved its main objective. The application effectively manages client orders, product inventory, and client information through a user-friendly graphical interface, contributing to smoother warehouse operations. Potential improvements could involve enhancing the application's scalability to handle larger datasets and perform a wider range of operations on the database.

7. Bibliography

-https://www.baeldung.com/java-jdbc
-http://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/
-https://dzone.com/articles/layers-standard-enterprise
-http://tutorials.jenkov.com/java-reflection/index.html
-https://www.baeldung.com/javadoc
-https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html