DOCUMENTATION

ORDERS MANAGEMENT

ASSIGNMENT_3

STUDENT NAME: Maris Radu-Ioan
GROUP: 30421

CONTENTS

1. Assignment Objective

The Main Objective:

The assignment requirements are to design an application Orders Management for processing client orders for a warehouse. Relational databases should be used to store the products, the clients, and the orders. The application should be designed according to the layered architecture pattern.

Sub-Objectives:

- Analyze the problem and identify the requirements.
- Make a design for the Database.
- Create CRUD operations for each table
- Implement the smaller problems mentioned.
- Test the application.

2. Problem Analysis, Modeling, Scenarios, Use Cases

The application is meant for users that want to realize operations on multiple tables. The operations that a user can solve are:

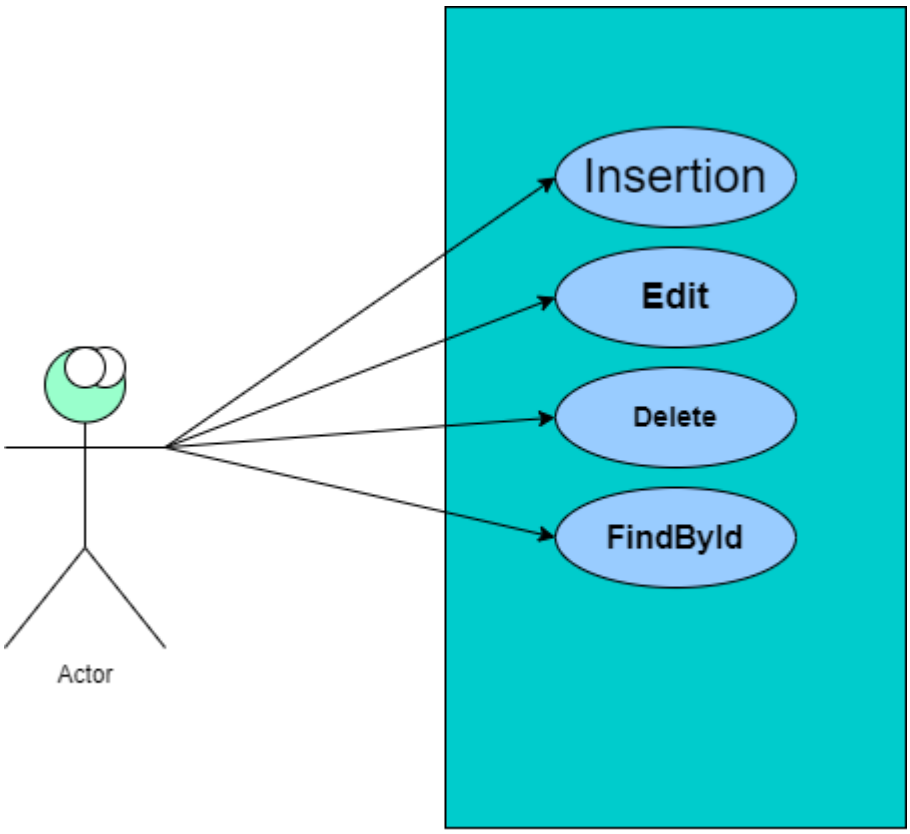
- Insertion of an object in a table
- Deletion of an object from the table based on id
- A way to edit an object in a table
- Find by Id

Successful scenario:

This case is met when the user's input is valid. This means that the user managed to insert an object in a table based on what table he selected. Also, he should be able to execute the delete an object from the table, to edit an object in the table and to find an object based on it's id.

Unsuccessful scenario:

When the input is not valid, considering the operations specified before, the imput is not recognized by the interface. Therefore, if the user does not input valid data, then the operations will not be executed.

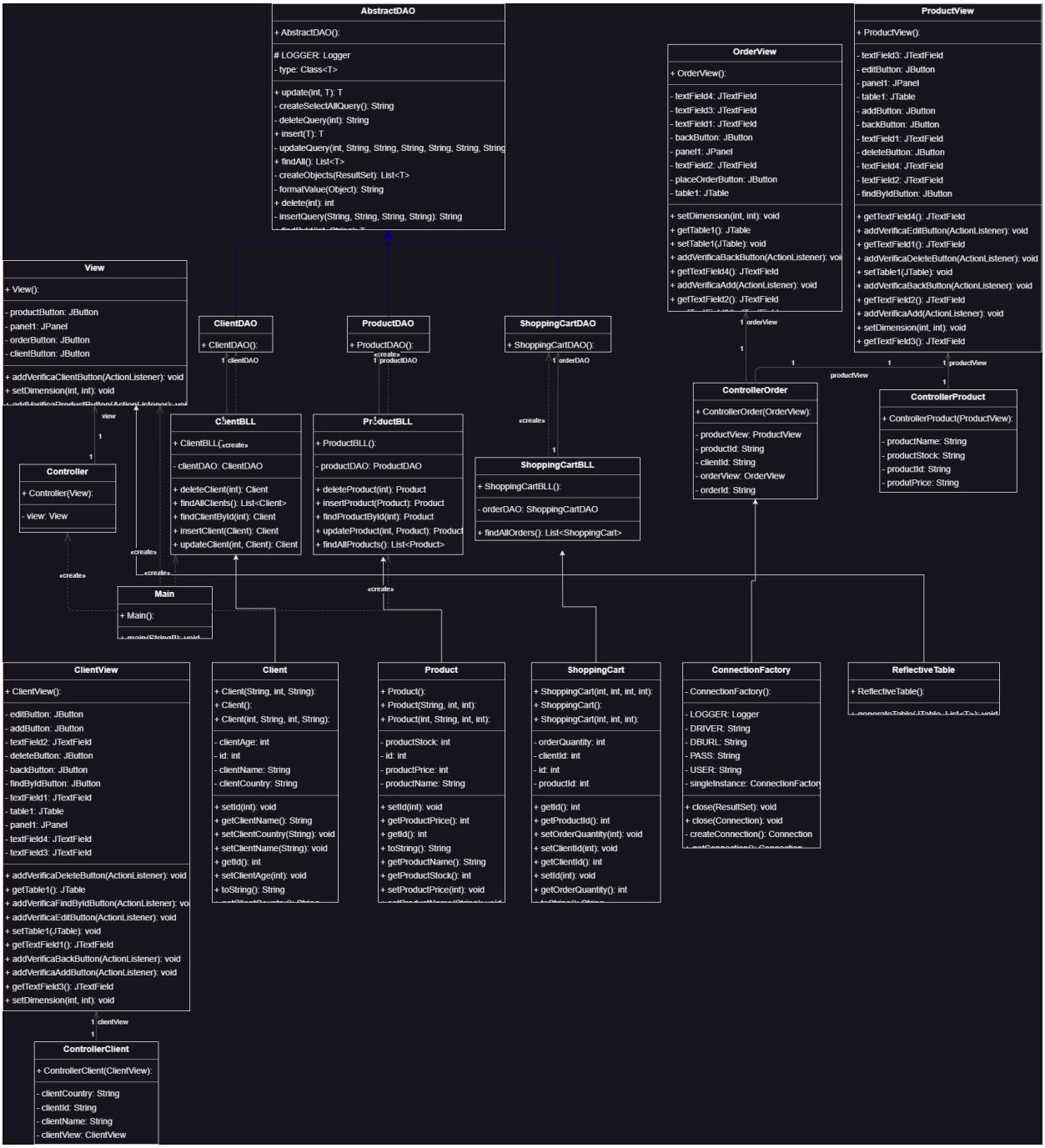


Use case diagram

3. Design

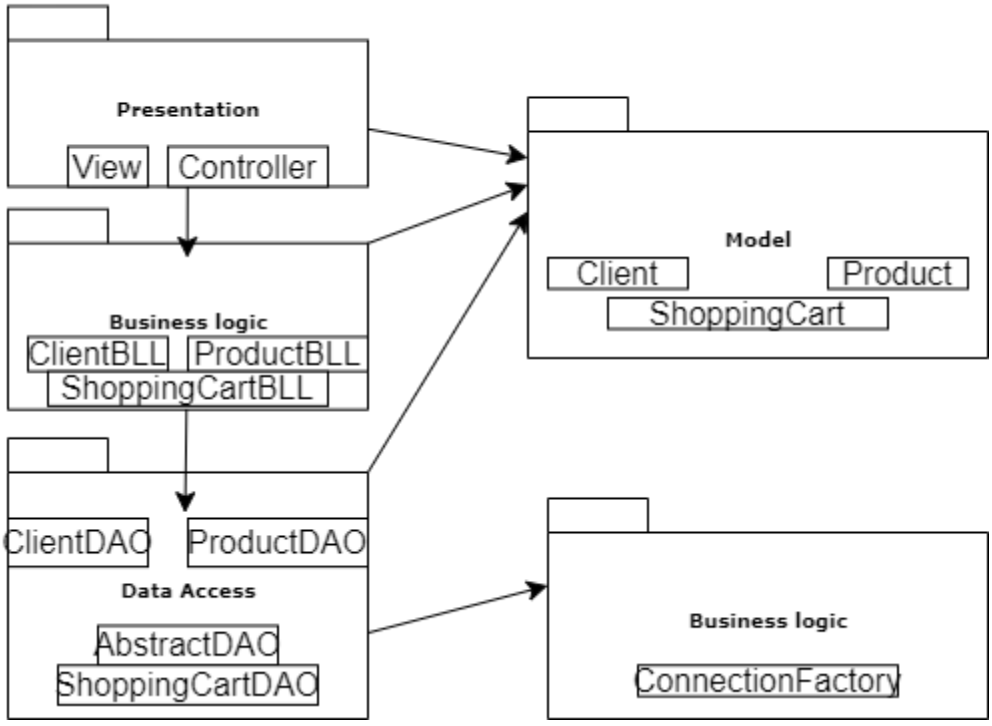
I used an OOP design in implementing the solution, by distributing them in classes and methods. For the implementation I used 21 main classes that I distributed in 6 packages.

The Interface is made using Java Swing.
The following are the UML Diagram and Package Diagram.
UML Diagram:



Package Diagram:

<<model>>



4. Implementation

The Presentation package contains 4 Java classes for the Interface. The View has 3 buttons and they are meant to transport the user to it's wanted table that wants to edit. The other 3 classes are ClientView, OrderView and ProductView. They each have 4 text fields and are used for the imputing the entry data. Then, there are 4 buttons: an Add button that is used to add the entry data in the table, a Delete button that is used to delete an entry from the table, an Edit button that when pressed the data written at the given id is rewritten to the ones that the user wrote.

Also, in the same package, I implemented the controllers for the interfaces. I did class Controller for each interface. Given this, there are 4 classes: Controller, ControllerClient, ControllerOrder and ControllerProduct, each implementing the functionality for the buttons.

In the Model package there are 4 Java classes: Client, Product, ShoppingCart and ReflectiveTable. The first 3 are meant to implement the models of the objects.

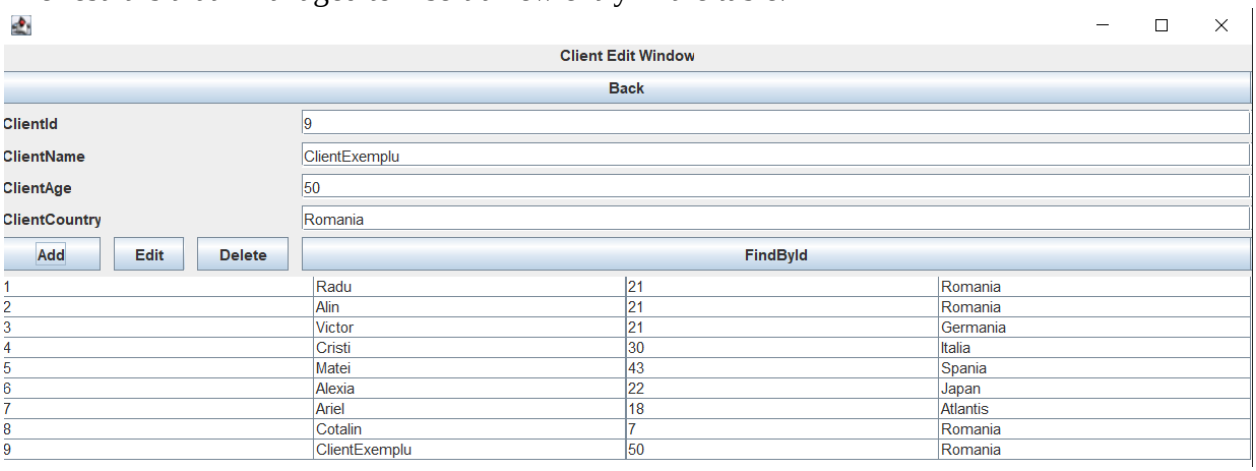
In the Connection package there is a Java class named Connection Factory. It manages the connection between the database and the Java application.

DataAccess package has 4 classes. The AbstractDAO is actually implementing the functionality of the classes and then the rest 3 classes: ClientDAO, ProductDAO, ShoppingCartDAO only extend the first class.

BusinessLogic package also has 3 classes: ClientBLL, ProductBLL and ShoppingCartBLL. They call their specific DAO and Model and call the implementation of their specific functionalities.

5. Results

The result is that I managed to insert a new entry in the table:



The same happens for the other 3 tables.

6. Conclusions

I conclusion, I learned how to use reflexion in my creation of functions in Java and how to access a database using MySQL. Also, I found out how to connect the database to the Java application.

As a future improvement, the application could have multiple tables in order to introduce multiple details about the products or the Client. Also, it can have a more friendly user interface.

7. Bibliography

The references that I consulted during the implementation of the homework are the following:

1. https://www.oracle.com/technical-resources/articles/java/javareflection.html
2. https://www.mysql.com
3. https://www.javatpoint.com/steps-to-connect-to-the-database-in-java