DOCUMENTATION

ASSIGNMENT 3

Orders Management Application

STUDENT NAME: Stroie Anca-Gabriela
GROUP: 30422

CONTENTS

1. Assignment Objective

The main objective of the assignment is to design and implement an application for managing the client orders for a warehouse.
The sub-objectives are to:

- analyze the problem and identify requirements;
- design the orders management application;
- implement the orders management application;
- test the orders management application;

2. Problem Analysis, Modeling, Scenarios, Use Cases

2.1 Problem Analysis

Online ordering is widely used and can have several advantages such as : convenience for customers, increased order accuracy, order tracking and transparency, easy to use for the customer.

Online ordering platforms can offer a user-friendly interface, intuitive navigation, view detailed information, access order history, receive notifications.

Functional requirements:

- The application should allow an employee to add a new client
- The application should allow an employee to add a new product
- The application should allow a client to make an order
- The application should have a user-friendly interface
- The application should allow a client to view information about the product

Non-functional requirements:

- The application should be easy to use by the user
- The application should be able to handle a large volume of database
- The application should implement appropriate security measures
- The application should be stable, minimizing downtime and data loss

2.2 Modeling the problem

The user will be able to select his username and a product he wants to buy and make the command. Also the employees should be able to add new products and new clients in the system.

In the model we use the following classes:
`

1. Client class :this class represents the data model for the application and the fields are the same as the fields from the table in the database.

2. Product class : is the data model for the products and the fields are the same as the fields in the table from database.
3. Order class: represents the data model for the orders.

In view we have classes that implement the graphical user interface such as ClientGUI, OrderGUI, ProductGUI, ClientTableView, ClientTableView.

In connection we have classes that deal with the connection to the database.

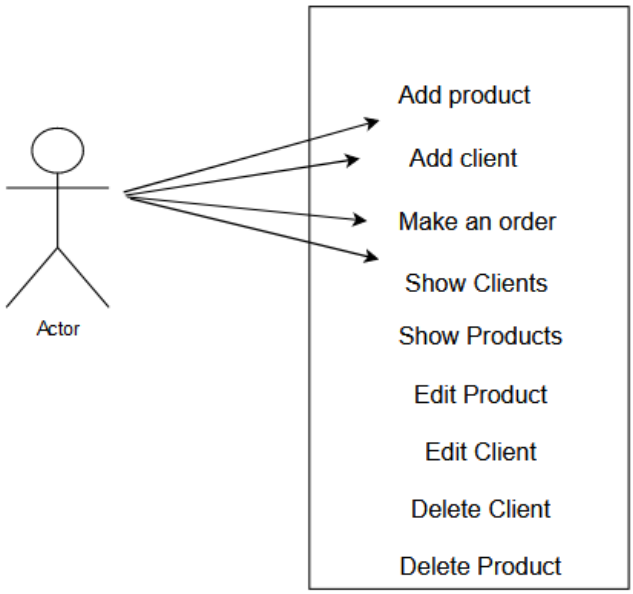
In dao we have classes that implement methods that make operations on tables such as add , delete , edit.

2.3 Scenarios and use cases

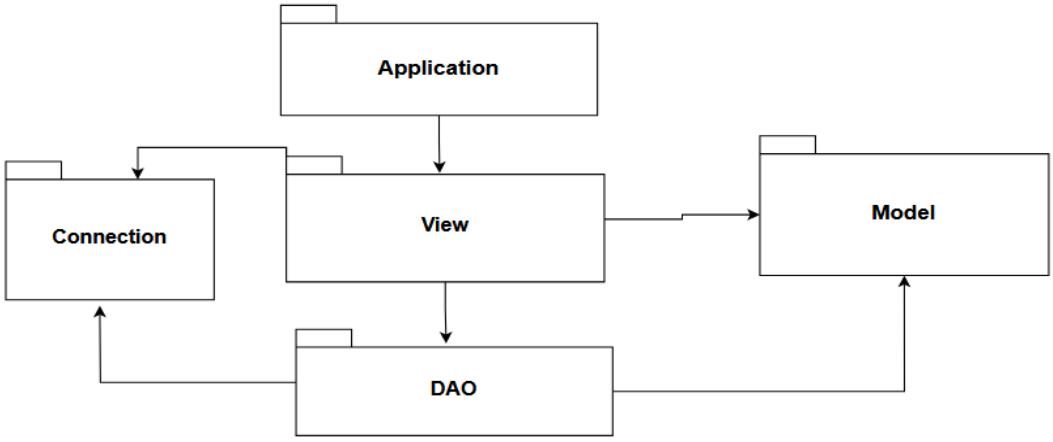
The use case presents the user that interacts with the application and can perform operations such as :make an order , add a client, edit a client, delete a client,edit a product, delete a product.

The user selects the operation, clicks on the coresponding button and the result will be displayed

If the user makes an order and there is not enough stock a message will be displayed. Also a message will be displayed if the information introduce dis not correct.

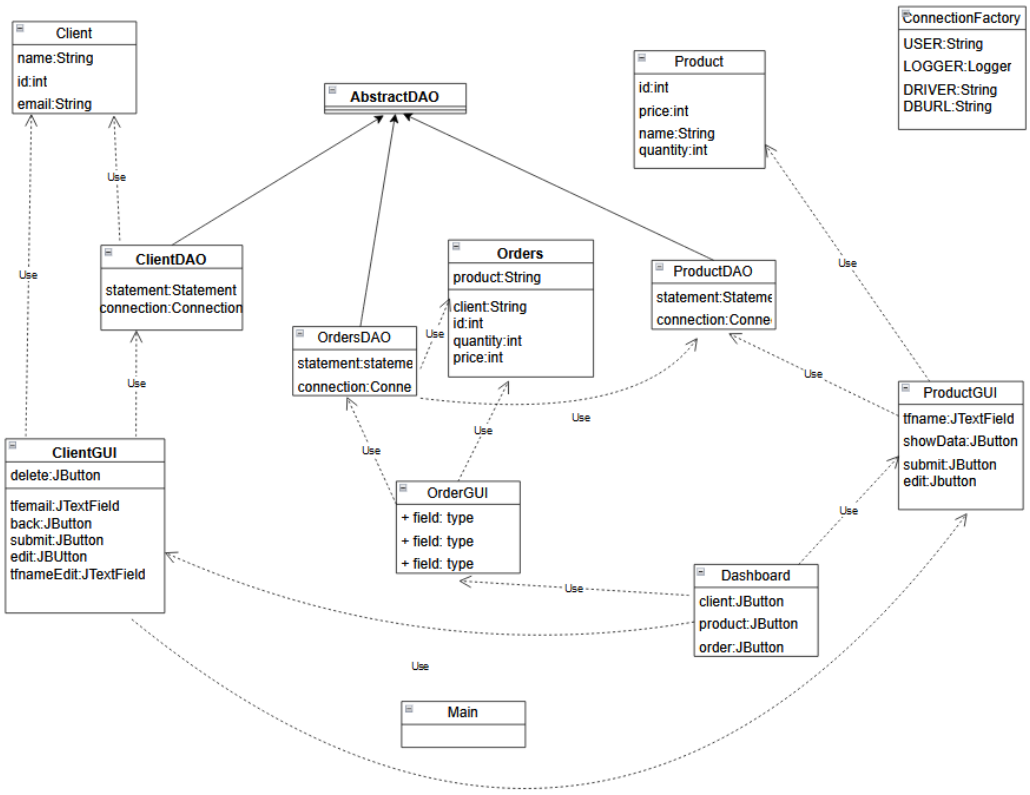


3. Design



Java packages help in organizing multiple modules and group together related classes and interfaces. The projects is splitted in five packages:

- Connection – contains ConnectionFactory class that is used in connecting the application to the database
- Model – contains classes Client,Order,Product
- View – contains classes that implements the user interface
- DAO- contains abstract class AbstractDAO that implements methods for editing, deleting and adding in the database , and also the classes ClientDAO, ProductDAO, OrderDAO
- Application- contains the main class that start the application



4. Implementation

DAO classes
AbstractDAO-contains generic methods to obtain a table from a list of objects, a method to insert in the database a specific object, a method to delete from the database and one to update an existing entry of the database. All the generic methods use reflection techniques.

ClientDAO – extends the AnstractDAO and uses the generic methods to implement CRUD operations on the Client table

OrderDAO– extends the AbstractDAO and uses the generic methods to implement CRUD operations on the Order table

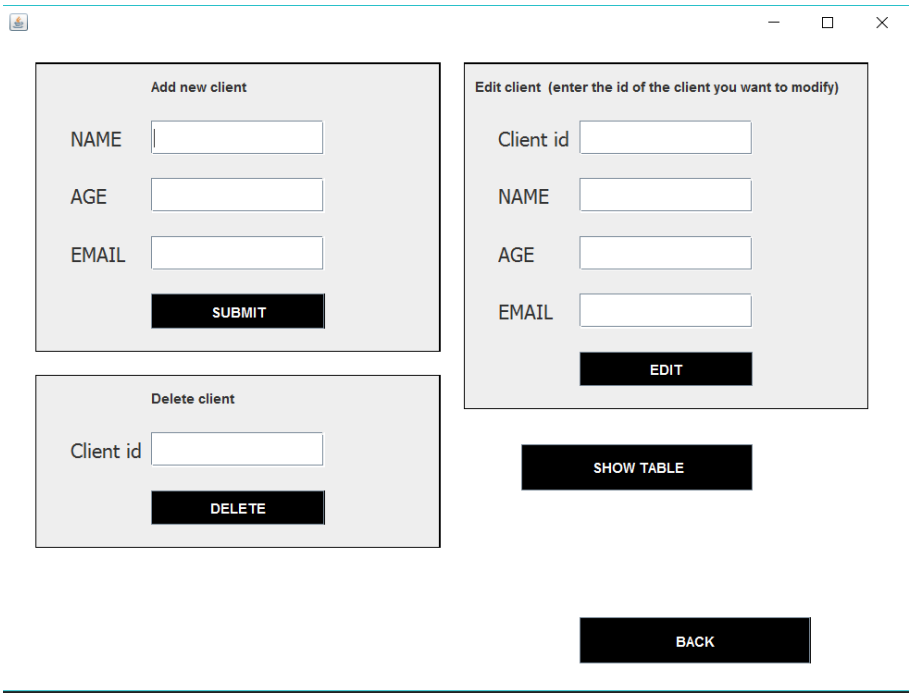
ProductDAO– extends the AbstractDAO and uses the generic methods to implement CRUD operations on the Product table

View package

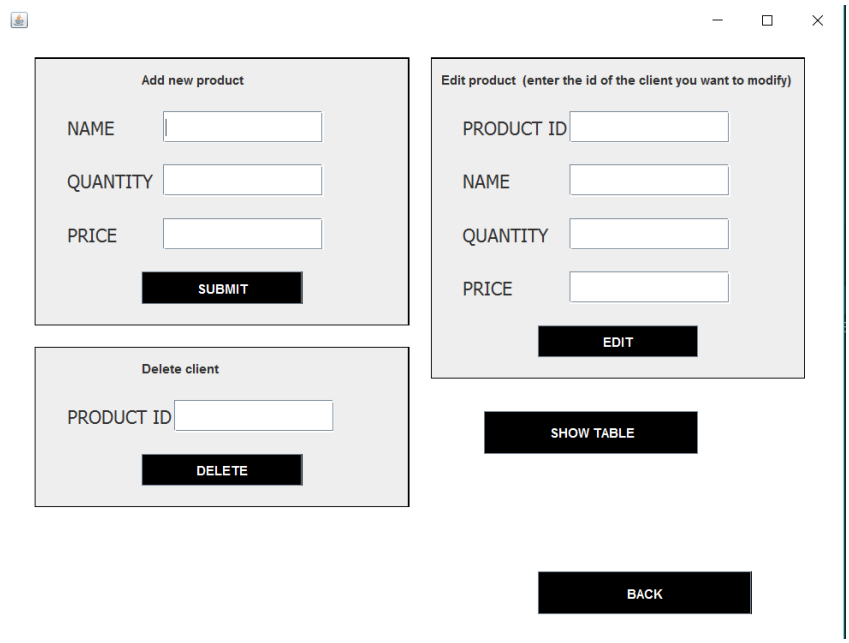
This contains the classes that implements the graphical user interface.

Dashboard class- here the user can select what operation he wants to make

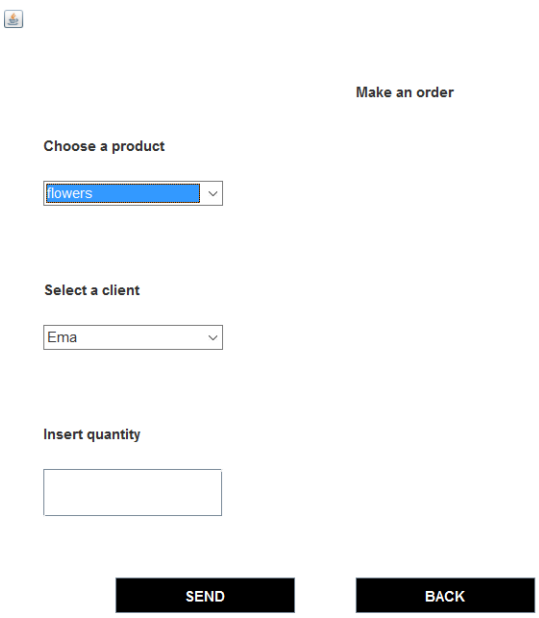
ClientGUI-in this class it is implemented the interface in which the user can add, delete, edit and view the table Client



ProductGUI- the user can add, edit ,delete a product and also view the table



OrderGUI-the user can make an order by selecting the username, a product and the quantity he wants



Connection class

It connects the application to the database, creates a statement and is also responsible with closing the connection or statement.

5. Results

In order to use the application , the user has to select the operation he wants to make:Client operation, Product operation, Make an order. Then the data that needs to be introduces into the table is written into the corresponding fields and the button is pressed.

6. Conclusions

This application is a user friendly application that performs basic operations such as adding, deleting, editing a table form a database. Is was an helpful exercise to remember OOP concepts and GUI.
As future developments, we could add:
-advanced search and filtering to allow users to search products based on multiple criteria
-order notifications
-return and refunds
-ability to order multiple products

7. Bibliography

- https://dsrl.eu/courses/pt/#
- https://gitlab.com/utcn_dsrl/pt-reflection-example
- https://www.baeldung.com/java-reflection