DOCUMENTATION

ASSIGNMENT *ASSIGNMENT\_03*

STUDENT NAME: Baraian Tudor

GROUP: 30424

# Assignment Objective

*-The main Objective: Create a database supported delivery application with a UI.*

*-sub-objectives: have a layered architecture patter design that uses the following packages:*

*-Model package: is responsible for creating the database objects that have their fields mapped one-to-one*

*to the columns of the database.*

*- Business logic package: is responsible for handling the application logic(an intermediary between the access to the database and the actual application usage.*

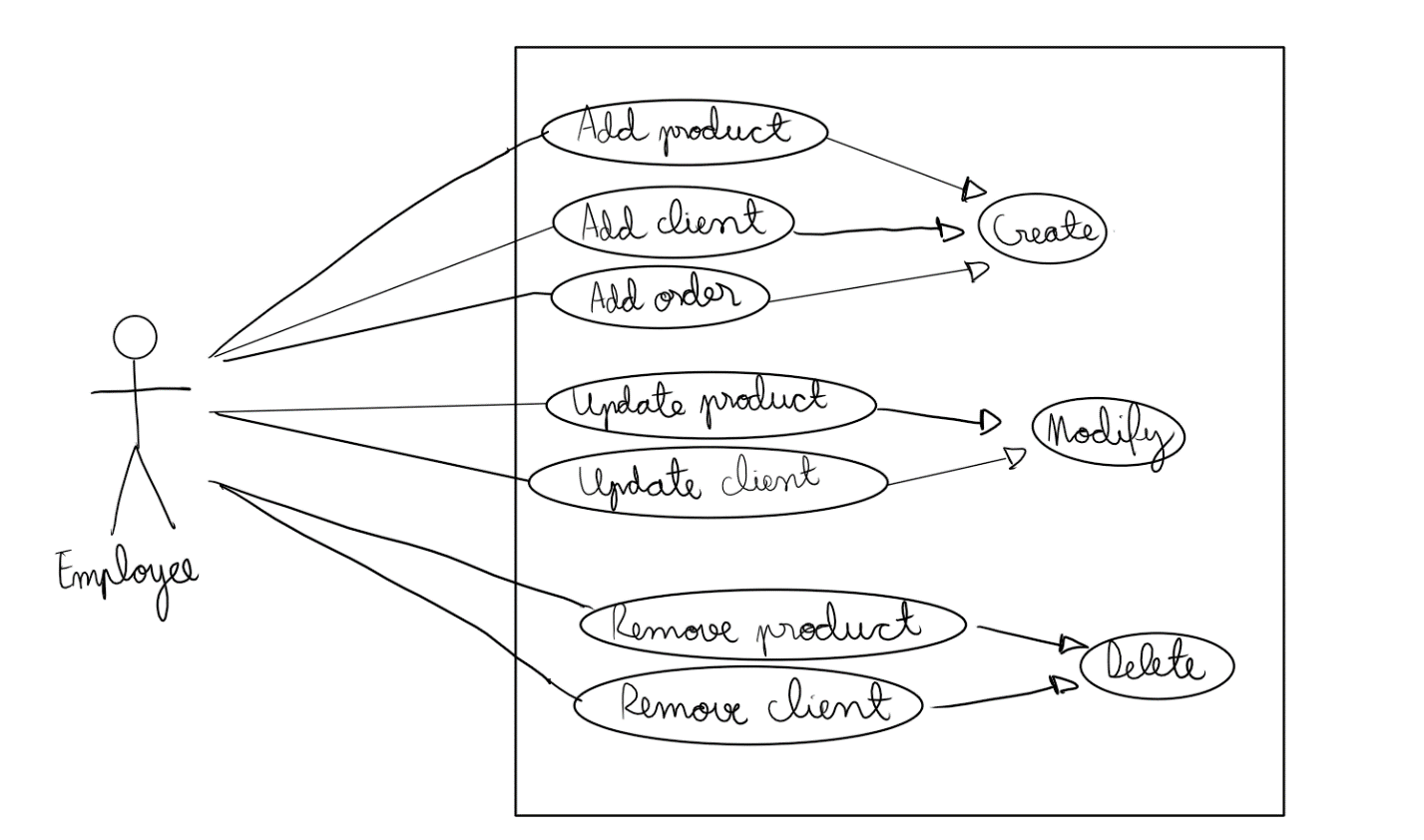
*-Presentation package: responsible for handling the GUI related classes.*

*-Data access package: responsible for sending queries to the database. Receive data from the database and update the database.*

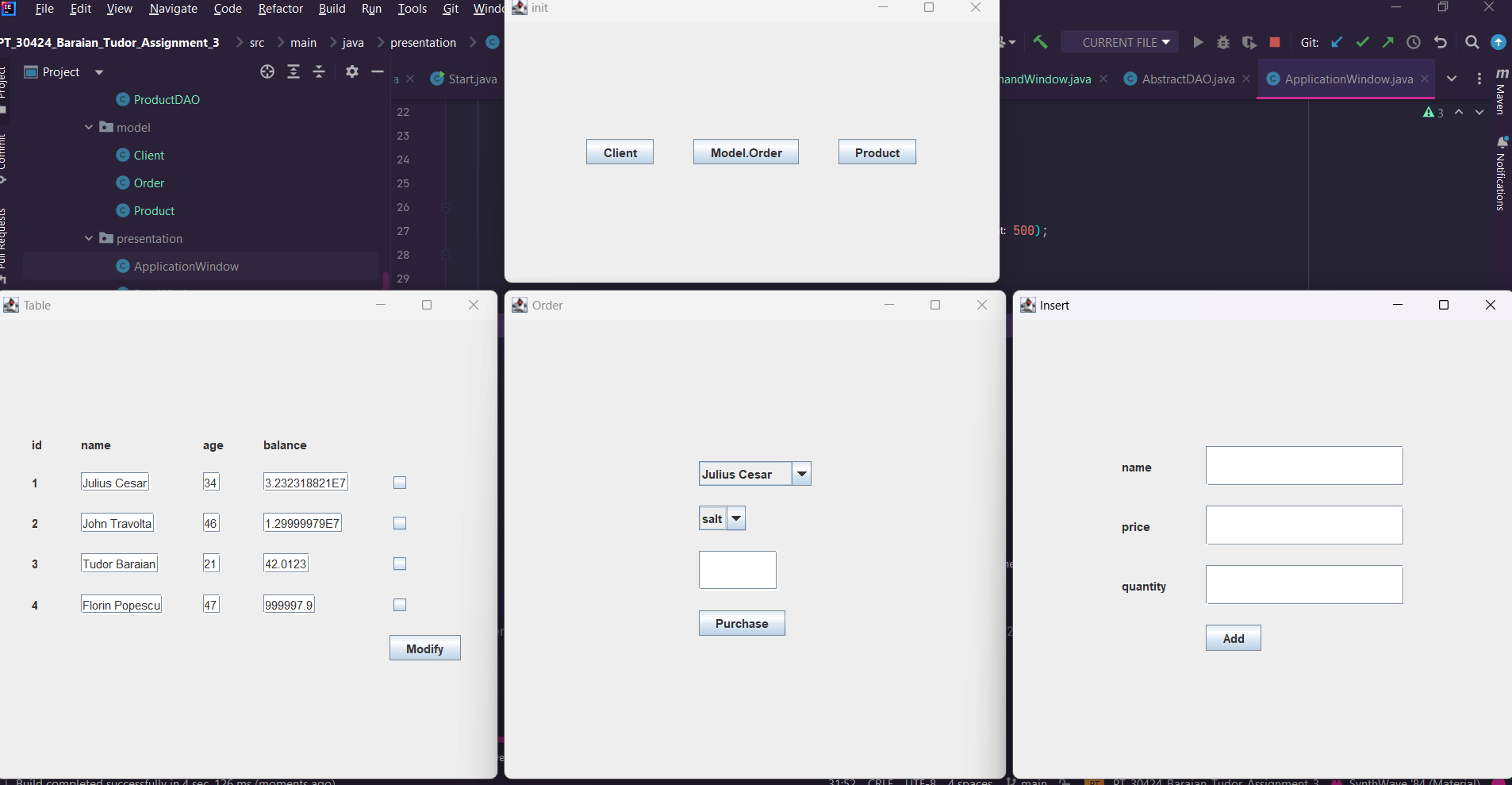
*-connection package: contains the class that makes connections to the database.*

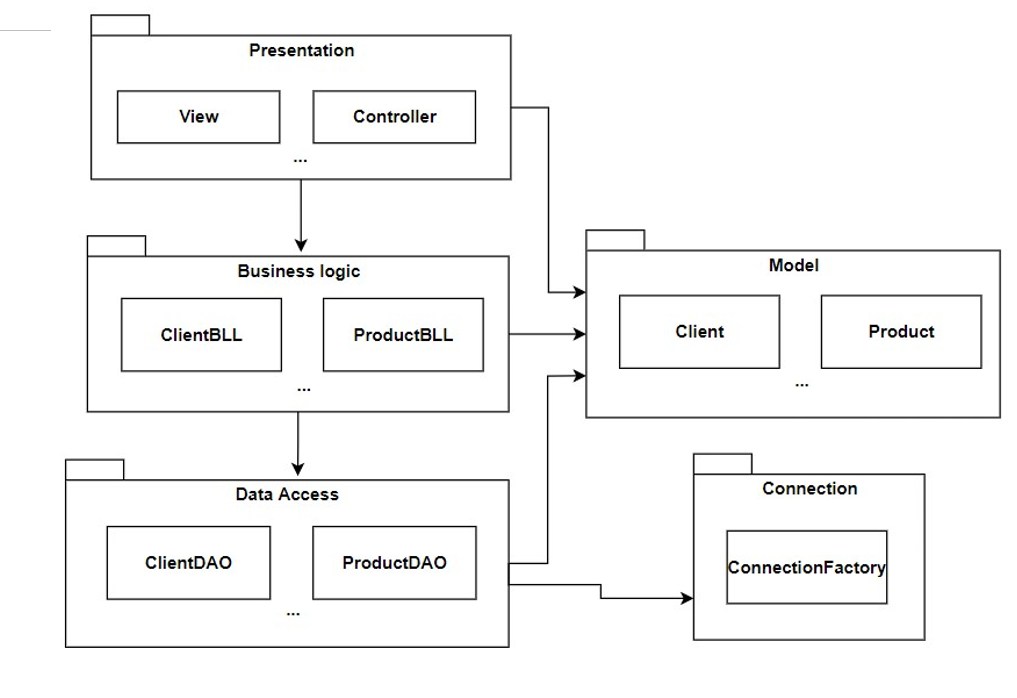
# Problem Analysis, Modeling, Scenarios, Use Cases

*The functional requirements should be presented together with the use cases (use case diagrams and use case description). The use cases’ descriptions can be done as a flow-chart or as a list containing the execution steps of each use case.*



# Design

**



# Implementation

*Each class will be described (fields, important methods). Also, the implementation of the graphical user interface will be described.*

*The* ***Connection Factory*** *class:*

*-this class handles connections between the database and the application*

*-it uses a singleton(a static instance of the class inside itself) that will be retrieved at each connection attempt.*

*The* ***ClientBLL, OrderBLL and ProductBLL*** *classes:*

*-create an instance of the data access specific class, then all the methods are called from that instance.*

*The* ***ClientDAO, OrderDAO and ProductDAO*** *classes:*

*-have methods that handle the queries to the database. Each extends the AbstractDAO class.*

*The* ***AbstractDAO*** *class:*

*-is a generic class that creates the basic queries for each database object that extends it through the DAO classes. It has methods such as findAll, FindById, FindMaxId, insert, update and remove.*

*It uses reflection to retrieve the fields of the specific object at runtime.*

*The* ***Client, Order and Product*** *classes:*

*-those classes represent the database objects, their fields representing the columns one-on-one.*

*-contains methods for setting the fields and getting the fields.*

*The* ***Application Window*** *class:*

*-extends the Jframe class. It is responsible for creating windows that handle client, product, and order queries.*

*The* ***BaseWindow***

*-is used for clients and products. Creates insertWindow and table Window in which a user can remove and update the specific rows of the table.*

*The* ***OrderWindow:***

*-creates a window in which the user can make an order.*

*The* ***CommandWindow:***

*-represents the window in which the user can pick a client, a product and a quantity, then he can request an order. If the client balance is not enough or the product quantity is too low, then an error message will be displayed.*

*The* ***viewWindow:*** *-represents the window in which the user views a table(clients or products) and can update, or remove some of them.*

# Conclusions

In this assignment, I learned how to create an application that works together with a database. Also, I learned how to use reflection techniques in order to extract dynamically the fields of a class(that replaces a generic argument).

# Bibliography

1. [*www.tutorialspoint.com*](http://www.tutorialspoint.com)

* **Connect to MySql from a Java application** o <https://www.baeldung.com/java-jdbc>
  + <http://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/>
* **Layered architectures** o <https://dzone.com/articles/layers-standard-enterprise>
* **Reflection in Java** o <http://tutorials.jenkov.com/java-reflection/index.html>
* **Creating PDF files in Java** o <https://www.baeldung.com/java-pdf-creation>
* **JAVADOC**
  + <https://www.baeldung.com/javadoc>
* **SQL dump file generation** o <https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>