Assignment 1

Analysis and Design Document

Student:Popovici Marius

**Group:30238**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

The task for this application is to build a deal search engine for furniture products using an Object-Oriented Language. An user should be able to create an account ang login to search for various provided deals. Deals are managed by staff and can be filtered by price, name and type. If a deal is available, users can add the associated product to their cart and proceed to checkout.

Payments can be done via a cash only policy and need to be validated by staff. Tihs creates an order in the system that can be tracked by the user from the Order History section. The state of an order is updated by staff. Once an order is delivered, the user can provide feedback in a form, in the specific Order History entry details.

# Functional Requirements

*[Present the functional requirements]*

The functinal requirements of this application are:

* Register – users can create an account that will be used later
* Login – User can access the application data using the chosen username and password registered before
* Search for deals and filter them– They are able to filter the product list by the type of the sale.
* Add products to cart – they have a virtual card where they can add the desired products.
* Order – after products were added in cart, they can proceed to checkout.
* Order history – clients can view the status and details of their orders

# Non-functional Requirements

*[Discuss the non-functional requirements for the system]*

* User interface – system should provide a graphical user interface for offering to clients the possibility of interracting with the app.
* Documentation – clients should be taught how to use the app
* Reliability – the system should keep all the information confidentially and the access to it should be done just through authentification
* Flexibility – the system should be flexible to changes
* Security – the clients should login before having the possibility of using the application
* Data validation – all entered data should be verified
* Error handling – exceptions should be caught and the client should be notified when one happens.

2. Use-Case Model

*[Create the use-case diagrams and provide one use-case description (according to the format below).*

*Use-Case description format:*

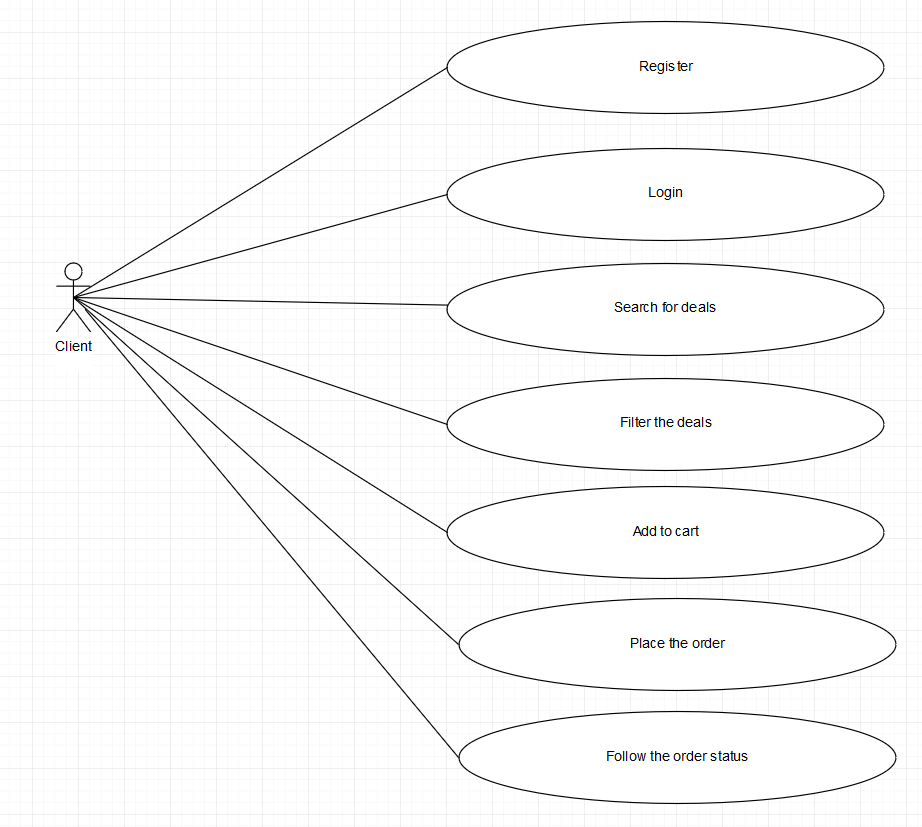
*Use case: <use case goal>*

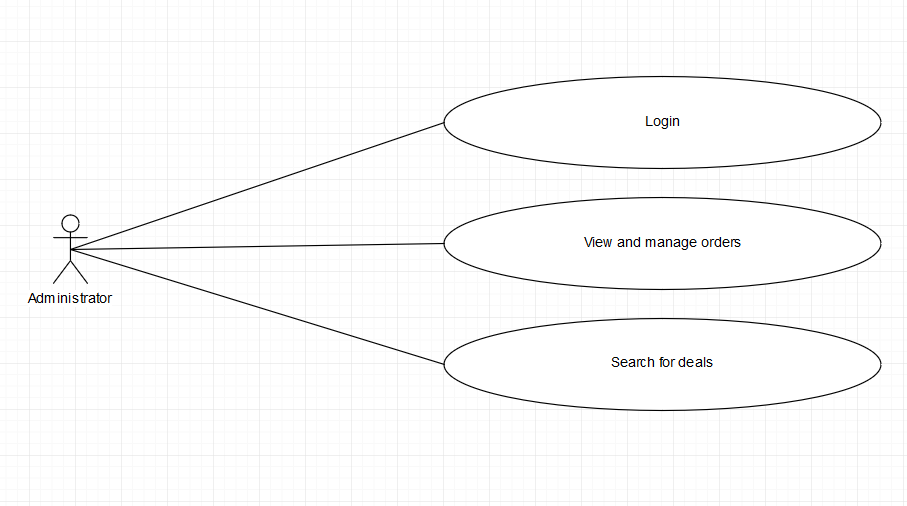
*Level: <one of: summary level, user-goal level, sub-function>*

*Primary actor: <a role name for the actor who initiates the use case>*

*Main success scenario: <the steps of the main success scenario from trigger to goal delivery>*

*Extensions: <alternate scenarios of success or failure>*





**Use case description:**

**Login:**

Use case goal: description of how a client can log into the system of Furniture Deals management.

Main success scenario:

1. The system shows a form where a client should enter his/her username and password.
2. The client enters his/her credentials.
3. The system verify and validates the entered data searching for them in the database
4. If the username and password are found in the database, the client will be logged into the system and another form will be opened.

Alternate scenarios:

If a client enters incorrect data, he/she is notified and asked to enter valid data.

3. System Architectural Design

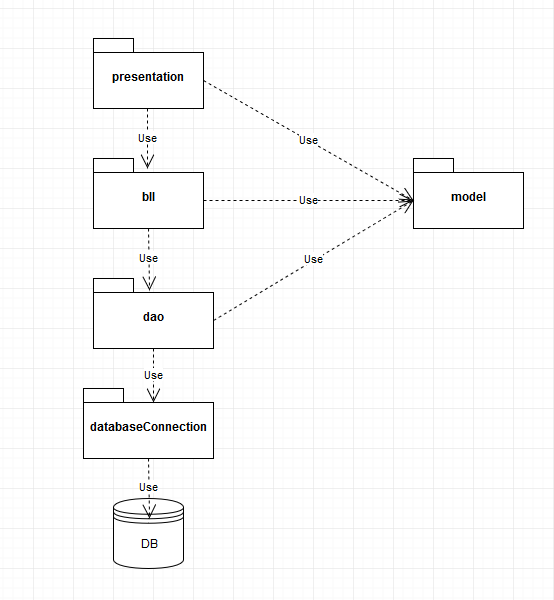
**3.1 Architectural Pattern Description**

*[Describe briefly the used architectural patterns.]*

For this application, I used Layered Architecture Pattern. The system is organized on five layers: model, dao, bll, presentation and database acces.

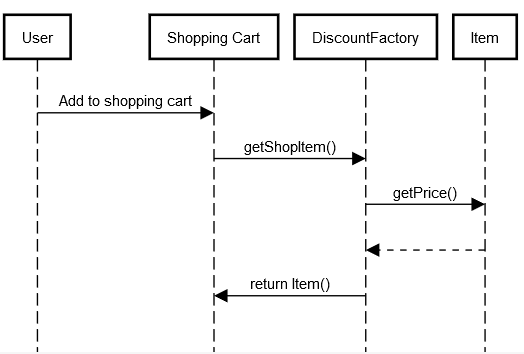
Model contains the main classes, dao (data access object) is the bridge between model and database tabels. Bll (bussiness logic level) is a “wrapper” on dao, whereas presentation is the highest level (it uses bll objects). The database access is responsible for creating the link to the database tabels.

**3.2 Diagrams**

**

4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*



5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

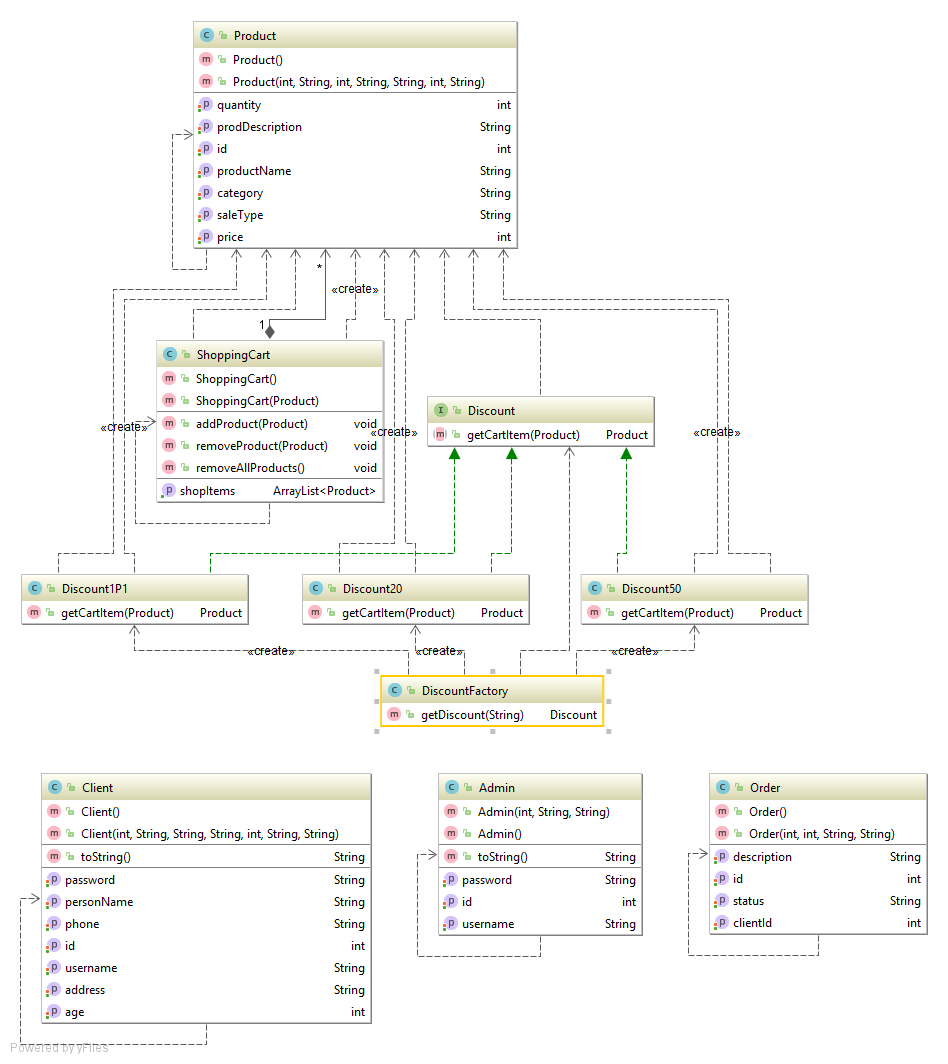
The used design pattern is Factory method. It is a creational pattern that uses factory methods to deal with the problem of creating objects. You don’t have to explicitely specify the class of the object that will be created. It is enough to pass a string like “20%” and it will do on itself the logic for creating the Discout20 object type.

The idea is creating an interface with a method that will be inherited by the implementing classes. The constructors of the classes will be called by another class (Factory class) that will instantiate, for every case, the specific needed object.

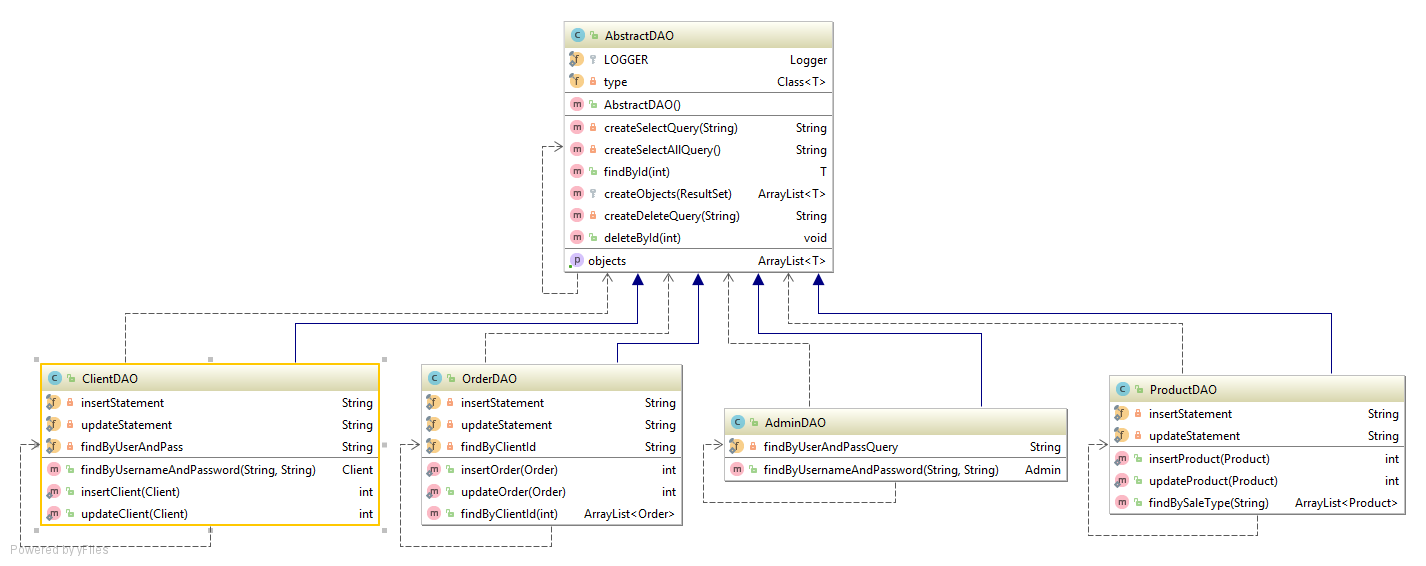
**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

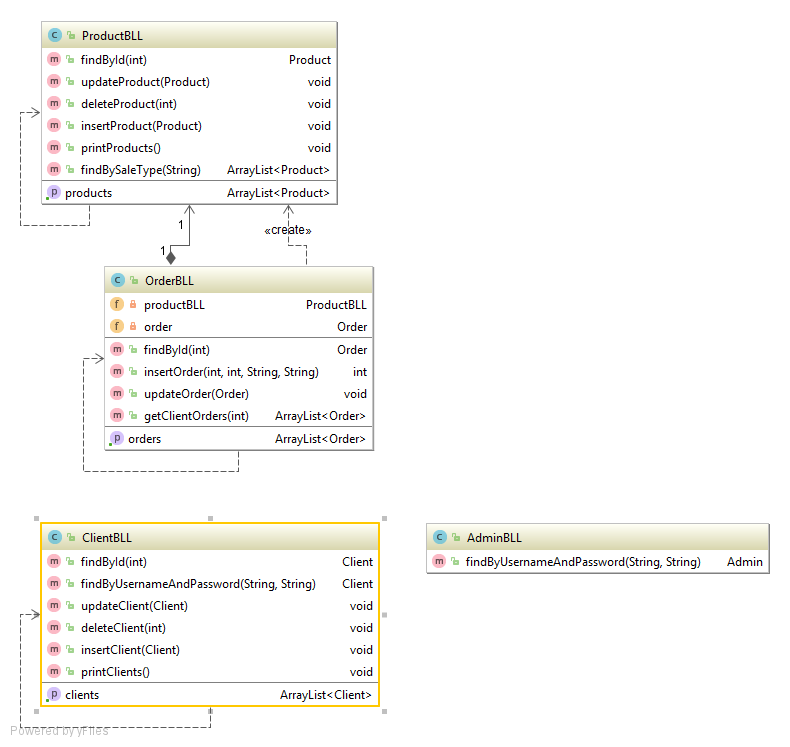
Main package:



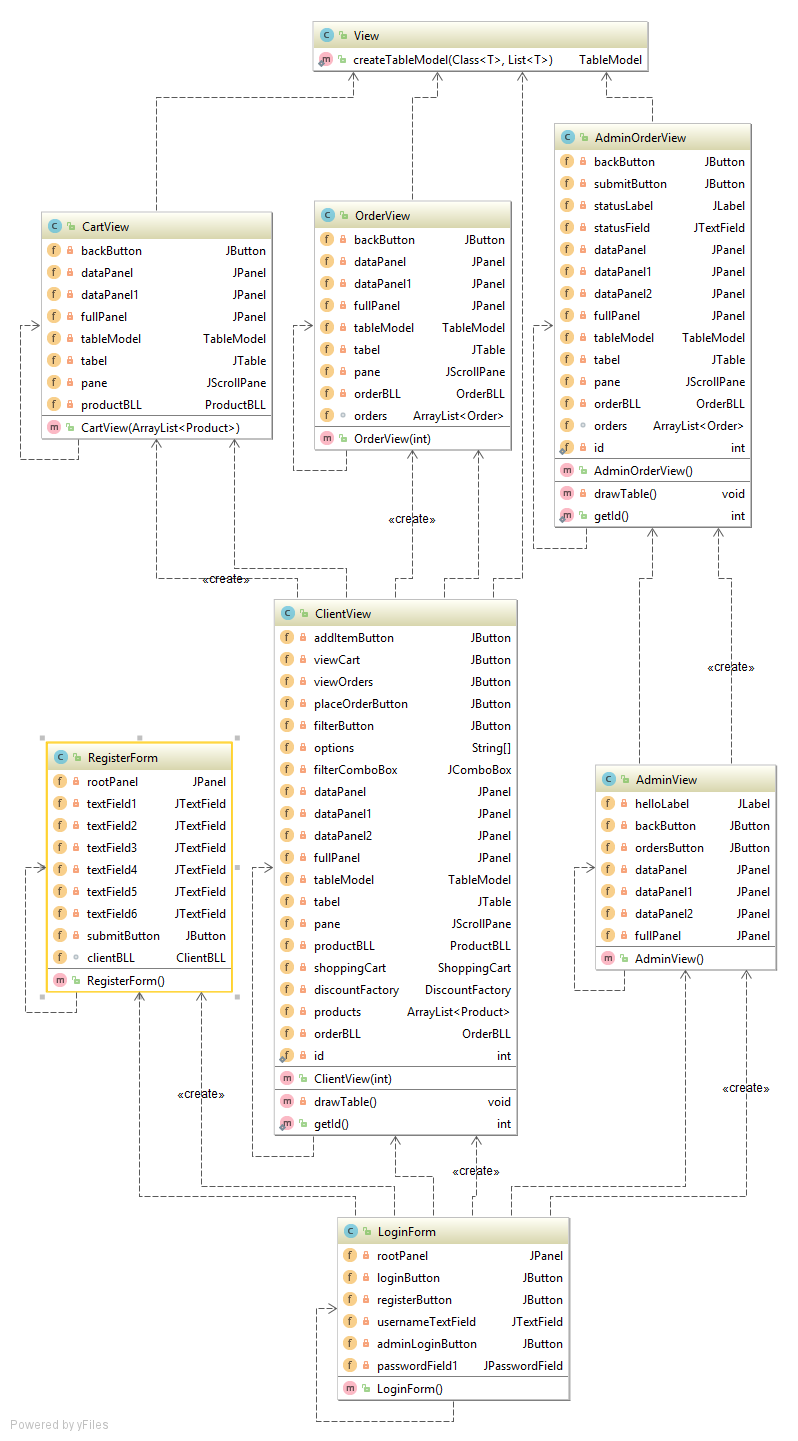
*DAO package*



BLL Package

**

View Package



6. Data Model

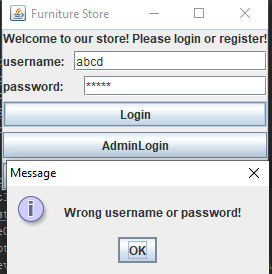
*[Present the data models used in the system’s implementation.]*

Data model is represented through Java classes and database tables. Each table in database has its equivalent in an object (class) in our application.

7. System Testing

*[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]*

Example: Wrong username or password:



8. Bibliography

<https://www.tutorialspoint.com/design_pattern/factory_pattern.htm>

<https://www.draw.io/>