Assignment 1

Analysis and Design Document

Student: Vadean Andrada-Anastasia

**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

[Application description]

I was tasked to build a deal search engine for furniture products. The user should be able to create an account and login to search for various provided deals. The deals must be managed by staff and can be filtered by price, name and type. If a deal is available users can add the associated product to his cart and proceed to checkout.

Payments can be done via a cash only policy and need to be validated by staff. This 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 on the specific Order History entry details.

# Functional Requirements

*[Present the functional requirements]*

I have two kinds of users: normal user and staff.

A user can: login, create an account, search for furniture and filtered them, see their orders history and add products to cart and checkout.

A staff should be able to: login, create account, see all the products, change the price and the quantity of a product, delete a product and add a product. Also, he can validate orders.

# Non-functional Requirements

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

We have some non-functional requirements like using a OOP language (Java), implement and test the application or commit my work in my Git repository. We have to use a layered architecture and store all the data in a database. All the inputs of the application have to be validated. Another requirement is to use a factory method to build and apply discounts on existing products witch change the order quantity and overall price accordingly.

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>*

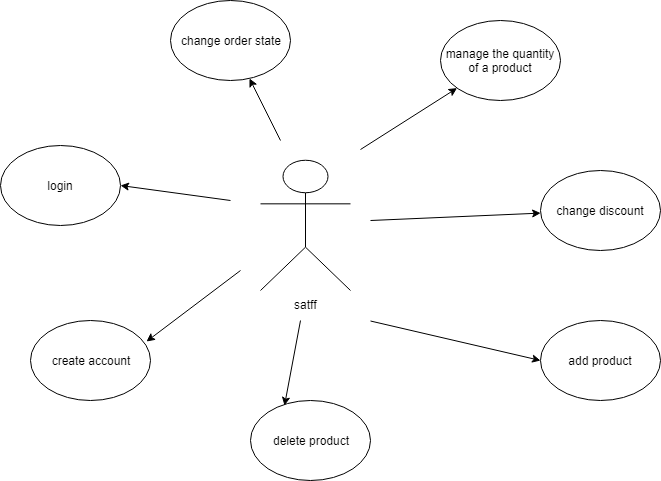
*Level: <one of: summary level, user-goal level, sub-function>*

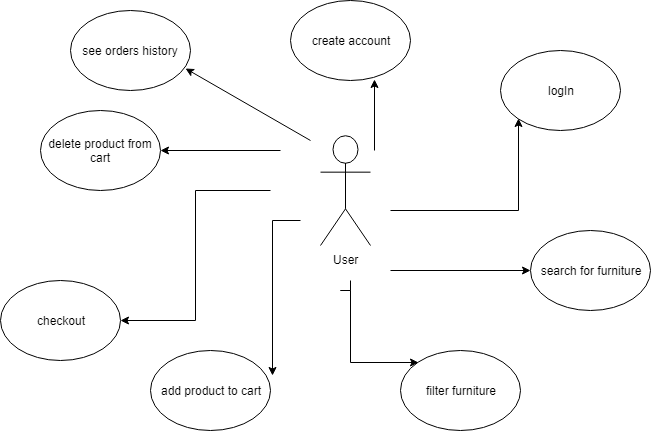
*Primary actor: <a role name for the actor who initiates the use case>*

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

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

*]*

****



Use case: Login user

Level: user-goal level

Primary actor: user

Main success scenario: the user enters his data and he can log in because he has an account

Extensions: the user doesn’t have an account and he can’t login

3. System Architectural Design

**3.1 Architectural Pattern Description**

*[Describe briefly the used architectural patterns.]*

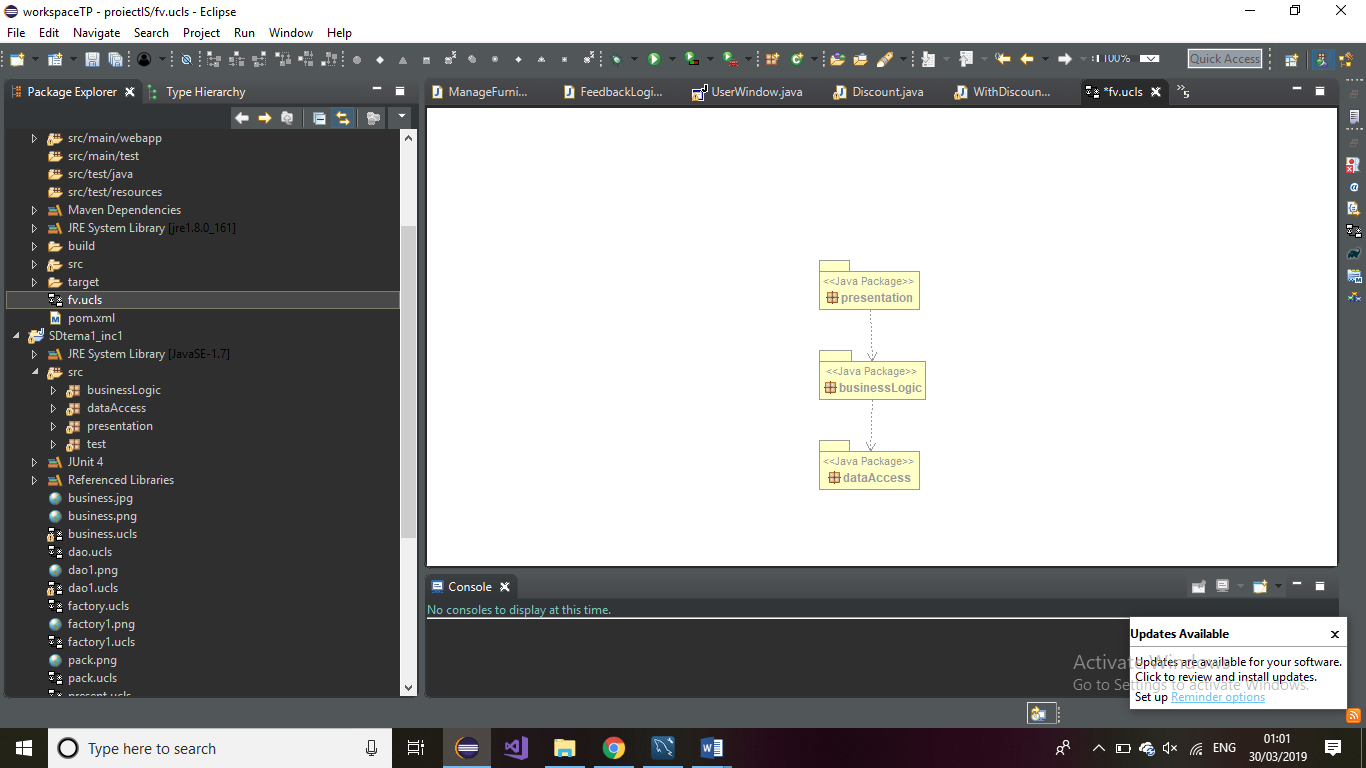
An architectural pattern expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.

I organized the system using Layers. So, I have a database, which was created using MySQL Workbench. The application has four packages: dataAccess (here I have the connection to the database), businessLogic (in this package are all the functionality of the application), factory (here I implemented a design factory pattern for choose the witch discount has to be applied for every product/deal), presentation (package used for the graphic interface for the user).

**3.2 Diagrams**

*[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]*

Package diagram:



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.]*

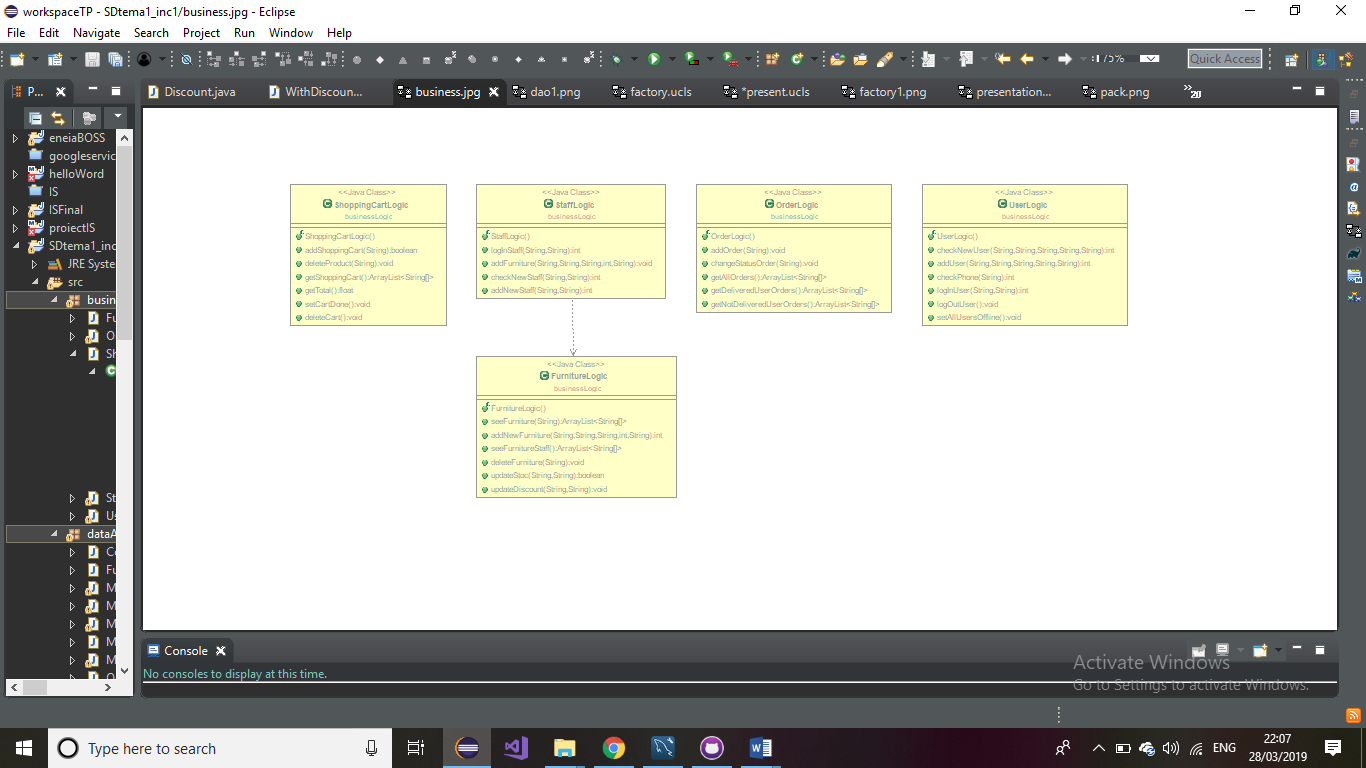
Like I said before, I use the layer architecture. Each layer of the layered architecture pattern has a specific role and responsibility within the application. For example, the presentation layer is responsible for handling all user interface logic, whereas the business layer is responsible for executing specific business rules associated with the request.

For decide how the products will be affected by their discounts I use a Factory pattern. The factory method pattern is a [creational pattern](https://en.wikipedia.org/wiki/Creational_pattern) that uses factory methods to deal with the problem of [creating objects](https://en.wikipedia.org/wiki/Object_creation) without having to specify the exact [class](https://en.wikipedia.org/wiki/Class_(computer_programming)) of the object that will be created.

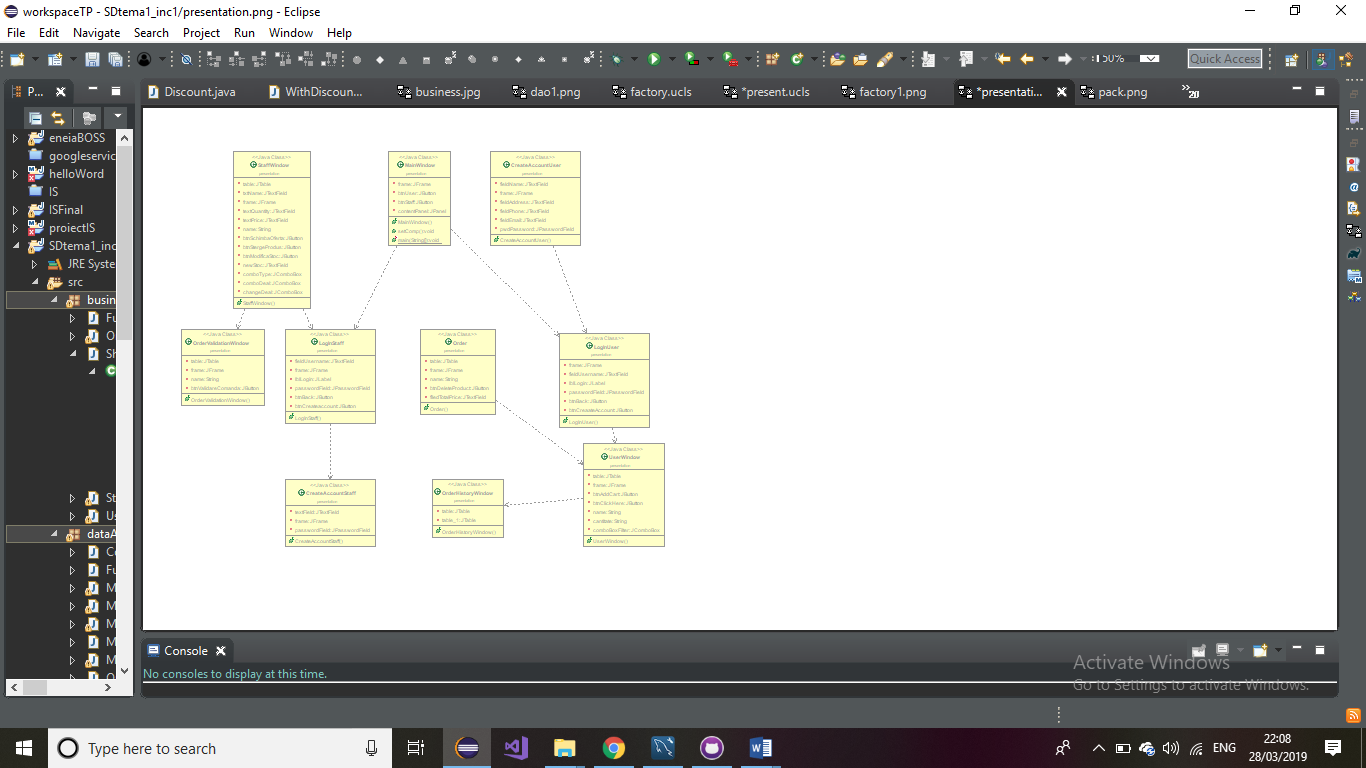
**5.2 UML Class Diagram**

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

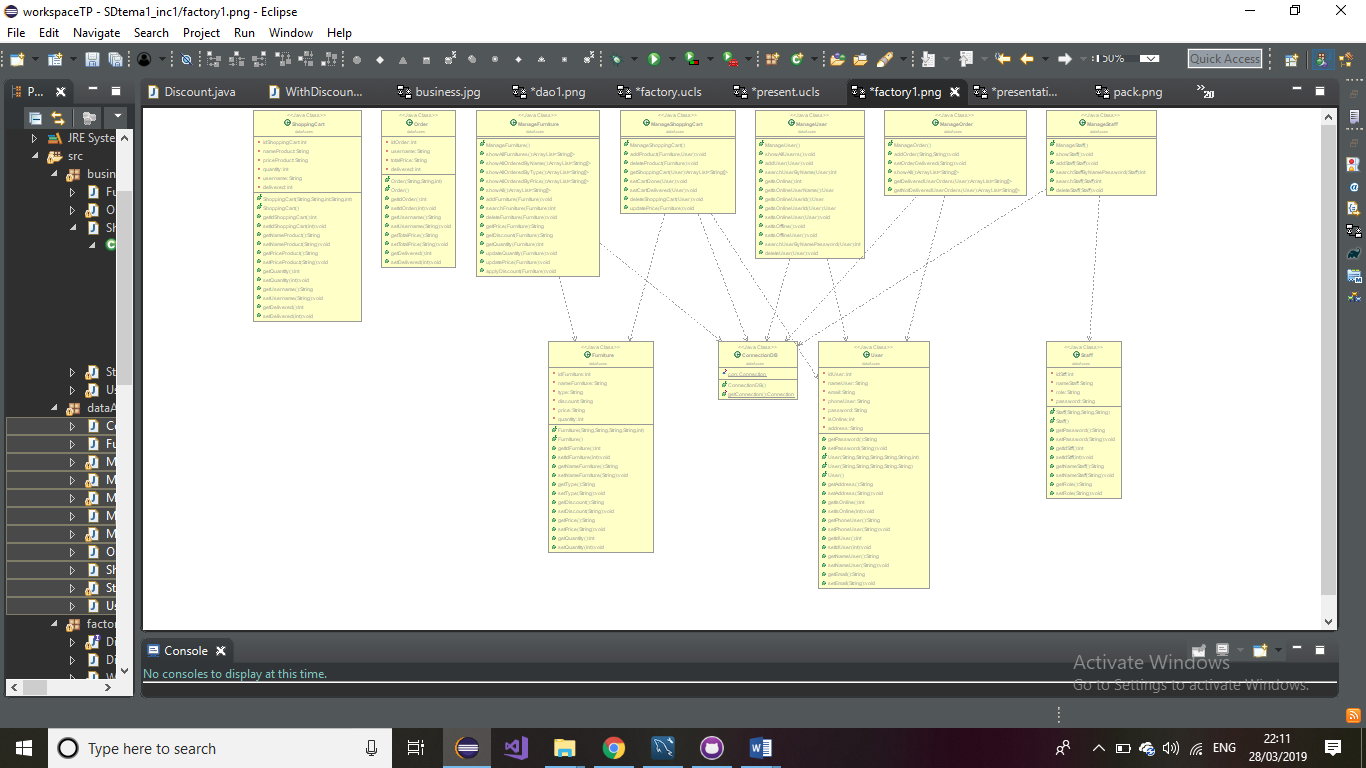
BusinessLogic package:



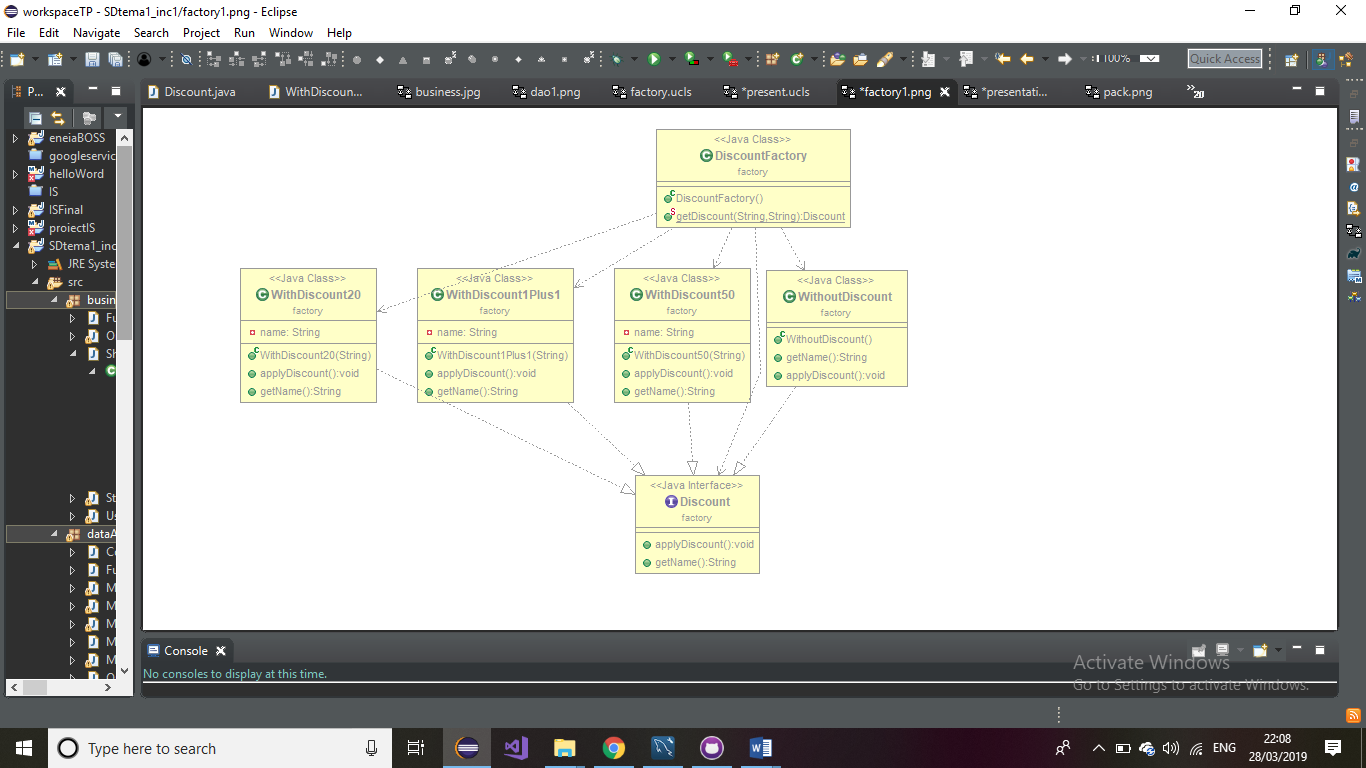
Presentation package:



DataAccess package:



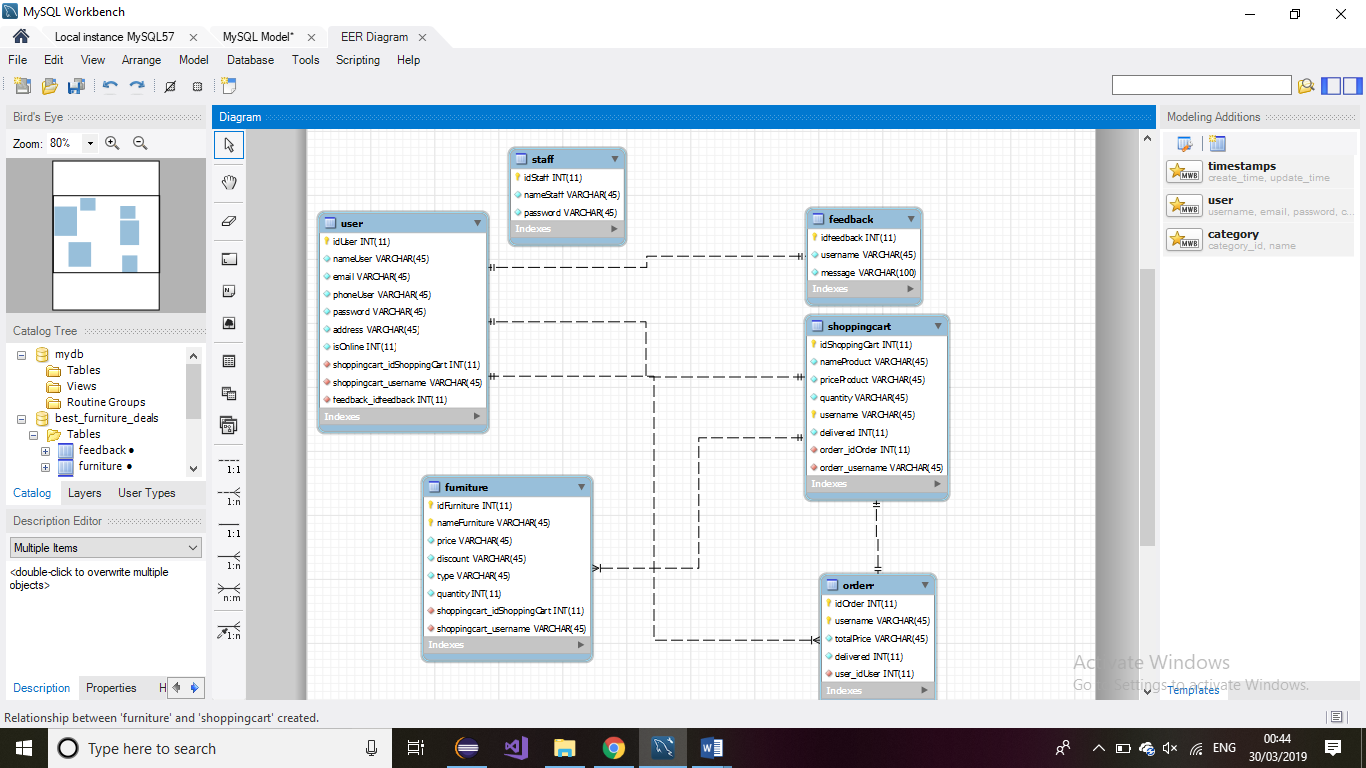
Factory package:



6. Data Model

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

Data Model is representing in the package dataAccess, by the classes User, Staff, Furniture, Order and ShoppingCart, which will be the objects used for implementing the system. Each class contain the principal attributes, which correspond with the column of the tables in database. For example, the class User has idUser, nameUser, address, password, email, isOnline and phoneUser. For each attribute I wrote methods for set and get attributes.



7. System Testing

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

I tested my application using the graphical interface. I have a Main class where I test the main functionalities.

8. Bibliography

<https://stackoverflow.com/>

<https://en.wikipedia.org/wiki/Factory_method_pattern>

<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>