<Online winter shop>

Analysis and Design Document

Student:Stefan Olimpiu

**Group:30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | Stefan Olimpiu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

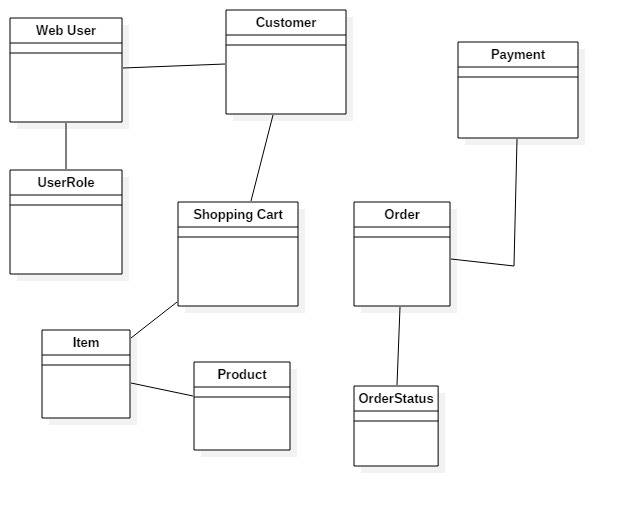
VI. Bibliography 5

# Project Specification

# Elaboration – Iteration 1.1

# Domain Model

The model will contain the following classes: Customer, WebUser, UserState, Account, Payment, ShoppingCart, Order, WinterItem, WinterProduct, OrderStatus



# Architectural Design

## Conceptual Architecture

Three-tier architecture will be used as my client–server software architectural pattern in which the [user interface](https://en.wikipedia.org/wiki/User_interface) (presentation), [functional process logic](https://en.wikipedia.org/wiki/Business_logic_layer) ("business rules"), [computer data storage](https://en.wikipedia.org/wiki/Computer_data_storage) and [data access](https://en.wikipedia.org/wiki/Data_access) are developed and maintained as independent [modules](https://en.wikipedia.org/wiki/Modular_programming)

Presentation logic is about how to handle the interaction between the user and the software. In my case, there will be more pages that will allow user to navigate through the application

Data source logic is about communicating with other systems that carry out tasks on behalf of the application. For our application, the biggest piece of data source logic is a database that is primarily responsible for storing persistent data.

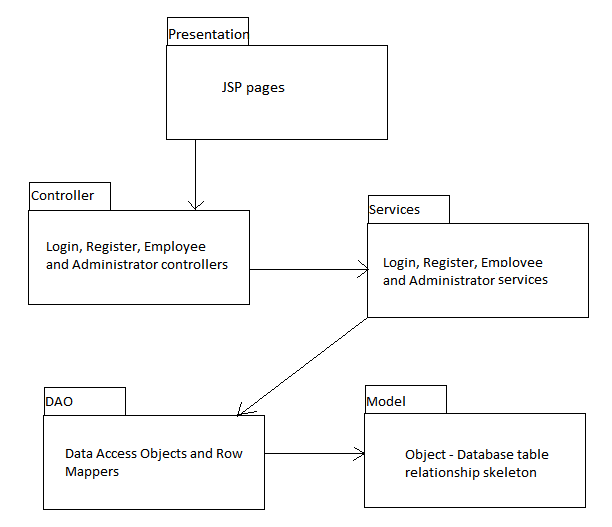
The remaining piece is the domain logic, also referred to as business logic. This is the work that this application needs to do for the domain you’re working with. It involves calculations based on inputs and stored data, validation of any data that comes in from the presentation, and figuring out exactly what data source logic to dispatch, depending on commands received from the presentation.

By using this pattern, my application will be easy to mantain and to extend. Components will be tested independently.

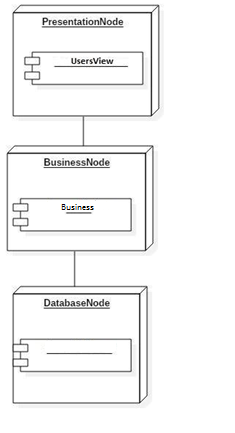
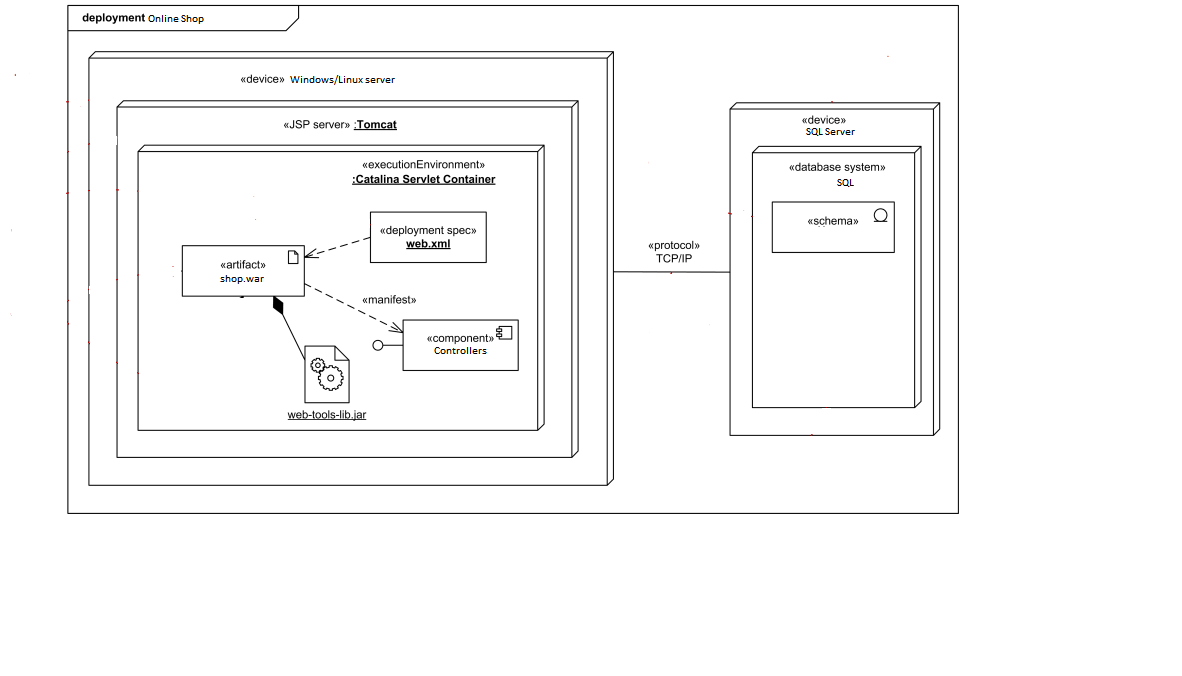
Also, the application will use the MVC pattern since I will be using Spring web MVC.

The MVC pattern will have a similar structure. The View and Controller components will be similar to the Presentation and Controller layers described above. The Model layer will represent a skeleton for the structure of the application. A more precise example of the concept of the model layer would be a package that contains the classes the define the structure of a table from a database (a class that contains class attributes representing the table columns from the database, constructors and getter and setter methods for those attributes).

## Package Design



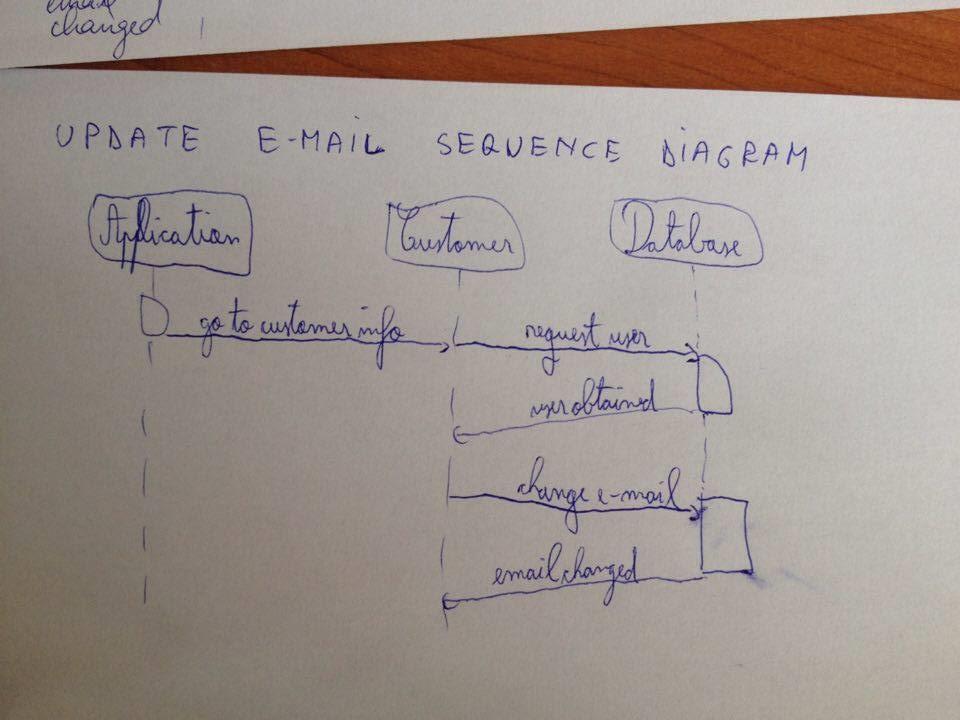
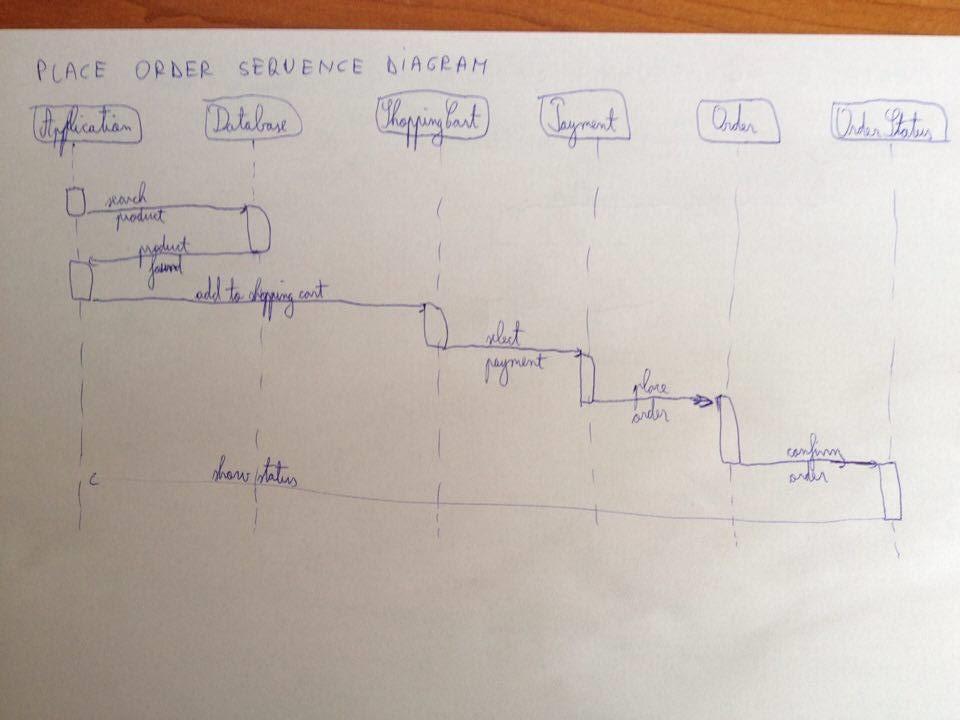
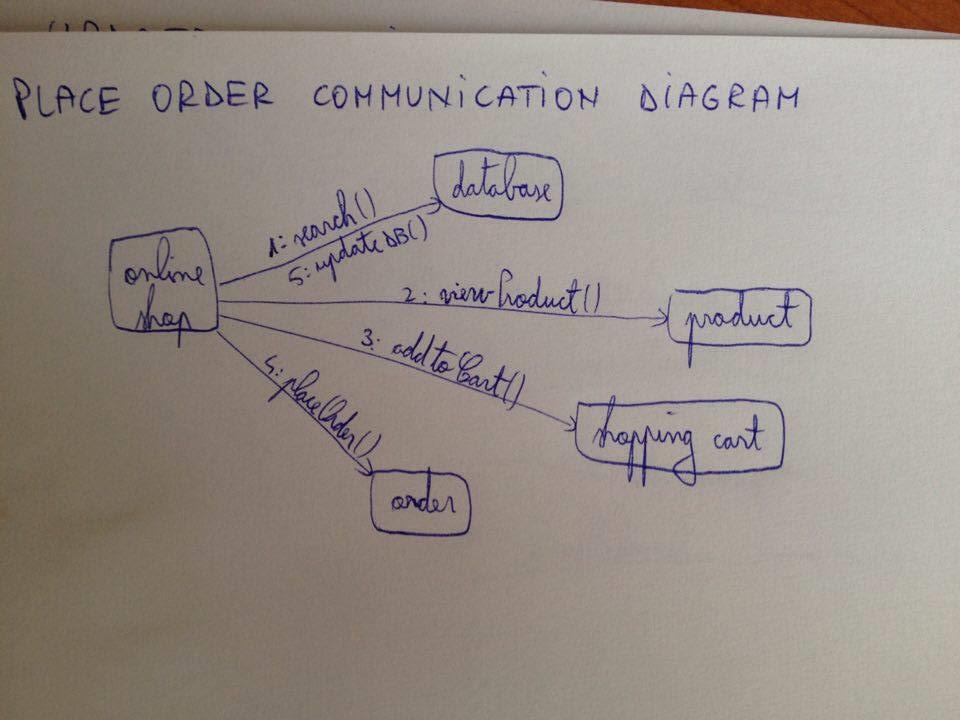
## Component and Deployment Diagrams



# Elaboration – Iteration 1.2

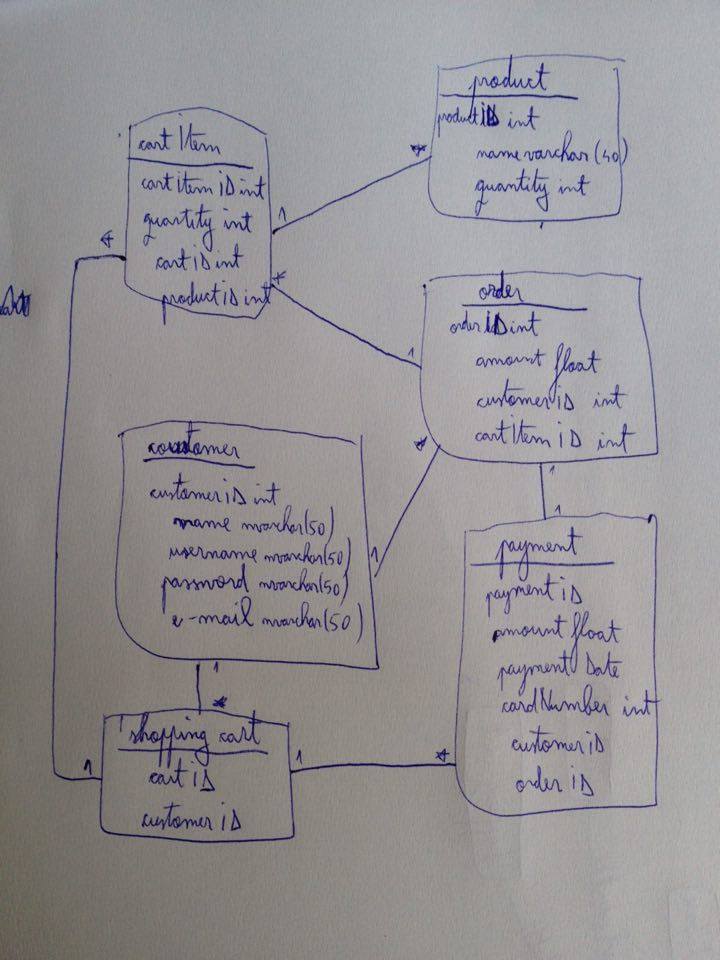
# Design Model

## Dynamic Behavior



## Class Design

# Data Model



# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography