Computer Parts Online Shop

Analysis and Design Document

Student: Cotet Eusebio Calin

**Group:30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <05/04/2017> | <1.0> |  | Cotet Eusebio Calin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

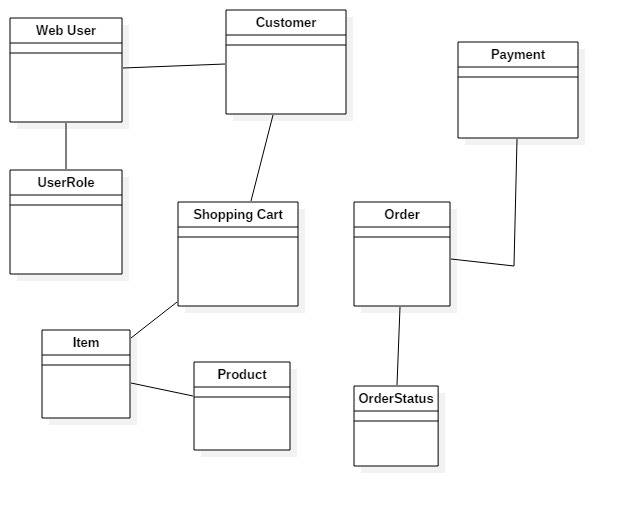
# Project Specification

# Elaboration – Iteration 1.1

The application represents an online computer parts shop. The application is a way for the guests to visualize, create an account, buy computer parts, review products and see reviews from other clients and also a way for the administrator to manage the online shop by modifying stocks and block users if needed.

# Domain Model

The domain model will be structured as shown in the conceptual class diagram given below:



# Architectural Design

## Conceptual Architecture

The application will be structured using the Layers architectural pattern and will also include MVC pattern since I will use Spring web MVC in my application.

The Layers pattern will structure the application in four major modules:

* Presentation Layer – this layer will include everything that can be seen in a browser by the user
* Business Layer – it will include all the functionality the application needs (i.e. add to cart and place order actions)
* DAO Layer – this layer will be responsible with the database access and CRUD operations requested by the business layer
* DB Layer – this layer will consist of the database itself

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 tht 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

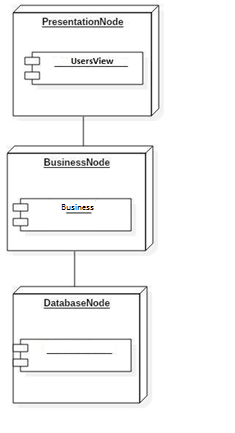
*[Create a package diagram]*

## Component and Deployment Diagrams

Deployment diagram

# C:\Users\Calin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\17807145_1442162715855971_1839308671_n.png

Component Diagram



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# 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