**<Online Bookstore>**

**Analysis and Design Document**

* **Elaboration – Iteration 1.1**
* **Domain Model**



* **Architectural Design**
* **Conceptual Architecture**

The architectural pattern used is Layers. Components within this pattern are organized into horizontal layers, each layer performing a specific role within the application. Although it does not specify the number and types of layers that must exist in the pattern, most layered architectures consist of three standard layers: presentation, business and database.

MVC Pattern stands for Model-View-Controller Pattern and will be used to create the presentation of the project. This pattern is used to separate application's concerns: the Model will be used to store the data of the aplication; the View represents the GUI, while the Controller has the purpose of controlling the data coming from and to the users.

The main reason I chose to work with layers is because of the flexibility and maintainbility they bring to the application. In order to be able to keep the code well organized, I also chose MVC, which will allow me to independently test and re-use the code written.

* **Package Design**



* **Component and Deployment Diagrams**

Deployment Diagram:



Component Diagram:



* **Elaboration – Iteration 1.2**
* **Design Model**
* **Dynamic Behavior**

**Sequence diagram**. Scenario: order the items placed in the shopping cart



**Communication diagram:**



* **Class Design**



* **Data Model**



* **Test Strategy**

For testing the application I will write small unit tests using JUnit, which is a unit testing framework used in Java. It can be used for both unit testing and UI testing. It works by defining the flow of execution of the code using annotations.

* **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]*