Online Book System

Analysis and Design Document

Student: Biris Alexandra

**Group: 30432**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/04/18 | 1.0 | First version | Biris Alexandra |
| 25/04/18 | 1.1 | Added iteration 1.2 | Biris Alexandra |
|  |  |  |  |
|  |  |  |  |

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

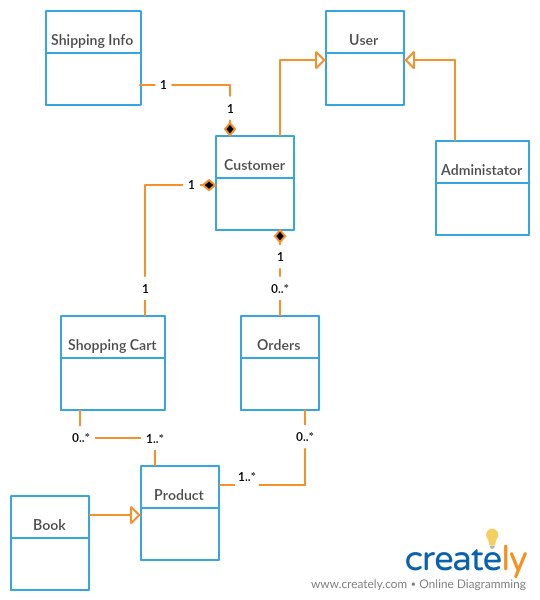
VI. Bibliography 5

# Project Specification

The Online book selling website is a tool that helps customers to buy the books online having a customized recommendation system. The system is one of the strongest tools to increase profit and retaining buyer, due to the fact that the system must recommend books that are of buyer’s interest.

# Elaboration – Iteration 1.1

# Domain Model



# Architectural Design

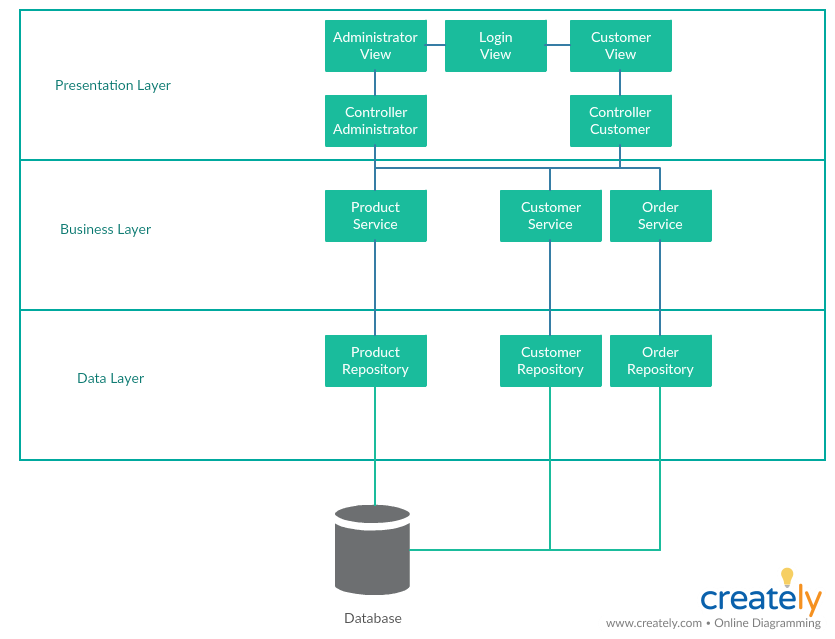
## Conceptual Architecture

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. Data layer will also be divided into entity and repository.

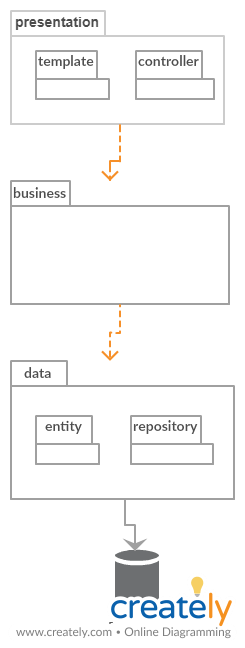
1. Presentation layer: responsible for handling all user interface and browser communication logic
2. Business layer: responsible for executing specific business rules associated with the request
3. Database layer: responsible for executing SQL statements to retrieve the corresponding data and pass it back up in the business layer.

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 as follows:

1. Model - This part of the framework is to store the data of the application, such as databases, text data, files and/or other web resources.
2. View - This is the graphical user interface of the application. That would contain different buttons, text boxes and other controls to let the user interact with the application to complete his projects depending on the sort of the software he is using.
3. Controller - The actual back-end code constitutes the controller of the framework. A controller controls the data coming from the users, or going to the user from a model.



## Package Design



## Component and Deployment Diagrams

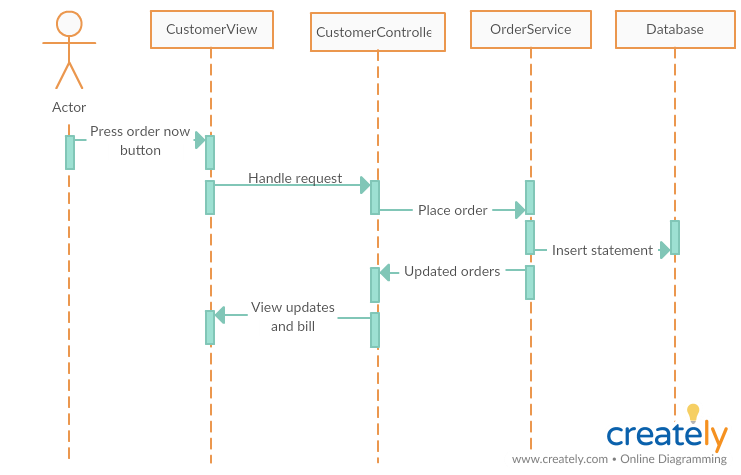
# 

# Elaboration – Iteration 1.2

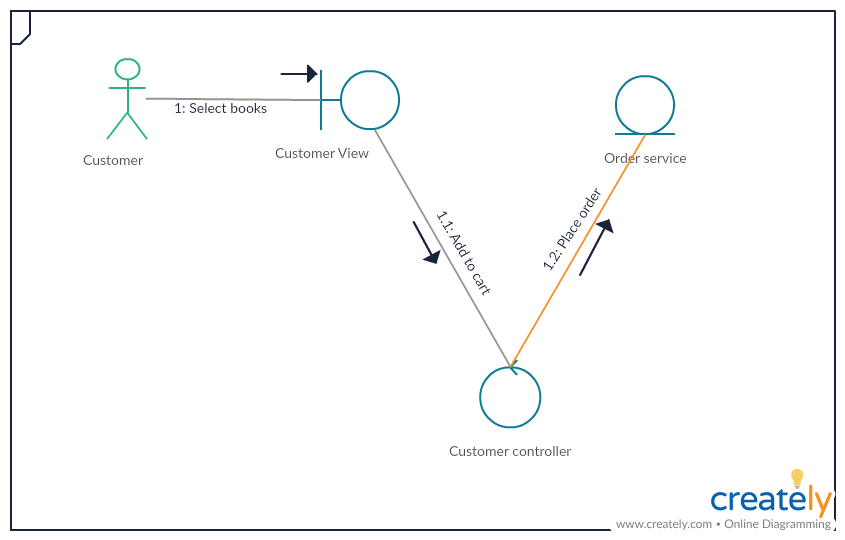
# Design Model

## Dynamic Behavior

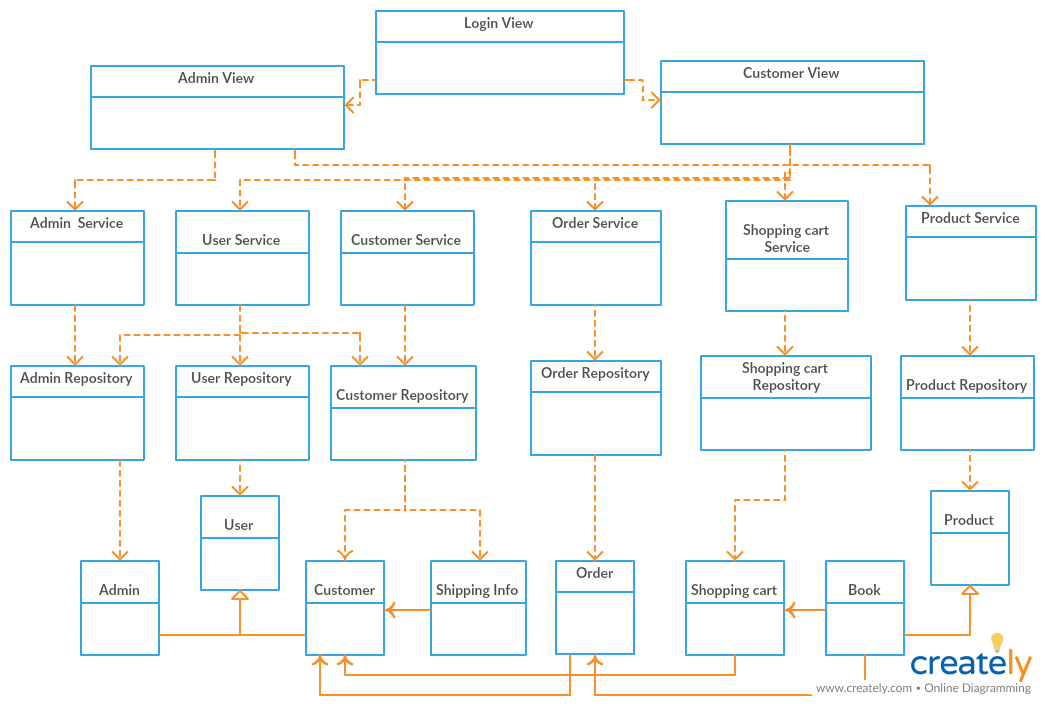
Sequence diagram: order the items placed in the shopping cart



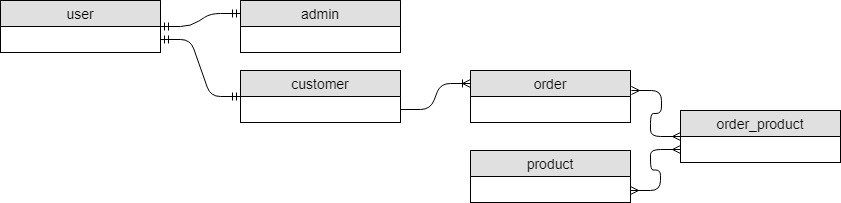
Communication diagram:



## Class Design



# Data Model



# Unit Testing

To test the application I will write some small tests and I will use Mockito. Mockito is a popular mock framework which can be used in conjunction with JUnit. Mockito allows us to create and configure mock objects, by simplifying the development of tests for classes with external dependencies significantly.

# 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