BookStore

Analysis and Design Document

Student: Toader Daiana

**Group: 30233**

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

Use Java/C# API to design and implement an application for the employees of a book store. The application should have two types of users (a regular user represented by the book store employee and an administrator user) which have to provide a username and a password in order to use the application.

# Functional Requirements

The regular user can perform the following operations:

* Search books by genre, title, author.
* Sell books.

The administrator can perform the following operations:

* CRUD on books (book information: title, author, genre, quantity, and price).
* CRUD on regular users’ information.
* Generate two types of reports files, one in pdf format and one in csv format, with the books out of stock.

# Non-functional Requirements

* The information is stored in multiple XML files
* The application will use MVC for the design
* Factory Pattern for generating reports

2. Use-Case Model

*Use-Case description format:*

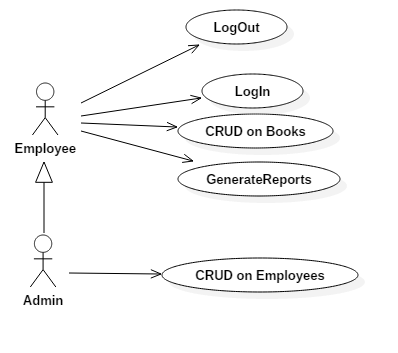
*Use case: Add Book*

*Level: user-goal level*

*Primary actor: Employee*

*Main success scenario: The book is successfully added to the XML file.*

*Extensions: The book already exists.*



3. System Architectural Design

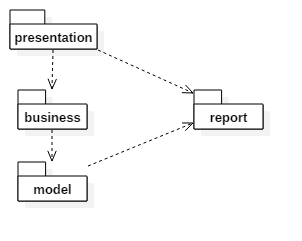
**3.1 Architectural Pattern Description**

Model**–**view**–**controller (MVC) is a [software architectural pattern](https://en.wikipedia.org/wiki/Architectural_pattern) for implementing [user interfaces](https://en.wikipedia.org/wiki/User_interface) on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user.The MVC design pattern decouples these major components allowing for efficient [code reuse](https://en.wikipedia.org/wiki/Code_reuse) and parallel development.

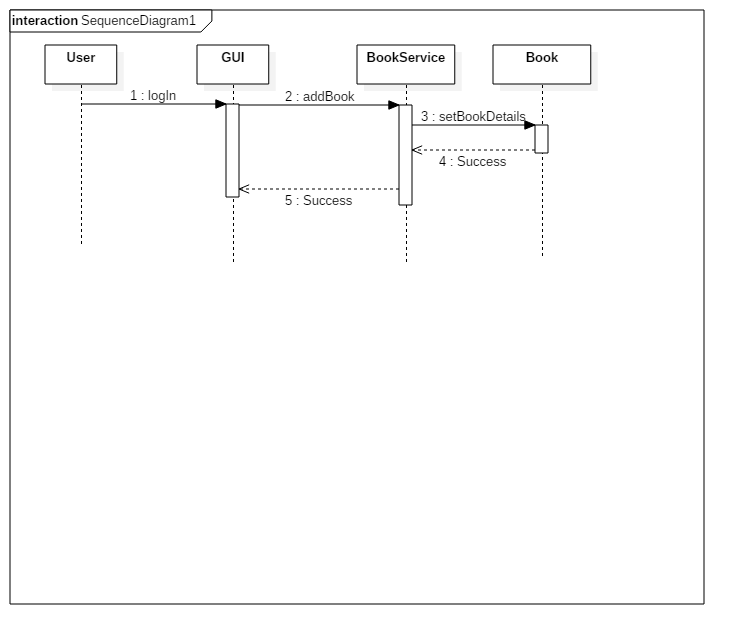
**3.2 Diagrams**

The application contains three packages that define the chosen architecture:

* model
* presentation
* business
* report



4. UML Sequence Diagrams

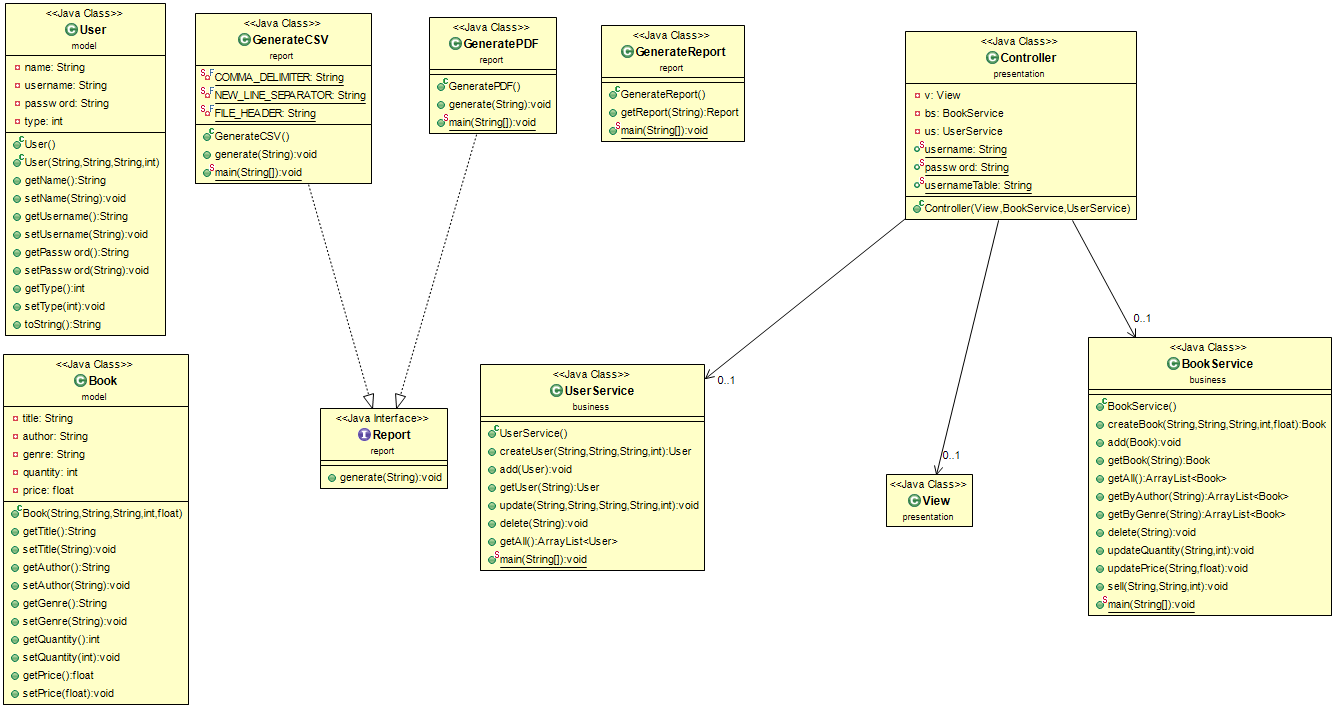


5. Class Design

**5.1 Design Patterns Description**

The factory method pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created. This is done by creating objects by calling a factory method—either specified in an interface and implemented by child classes, or implemented in a base class and optionally overridden by derived classes—rather than by calling a constructor.

**5.2 UML Class Diagram**



6. Data Model

The information is stored in 3 different XML files: books.xml, users.xml, sellings.xml.

7. System Testing

All the operations that a user and an admin wants to make are tested before they are send to the xml files. The users need to introduce a valid username and password so they can log inIf all of these requirements are not fulfilled than a warning message pops up and the user need to do the operation again.

8. Bibliography

<https://www.tutorialspoint.com/design_pattern/factory_pattern.htm>

<https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm>

<http://stackoverflow.com/>