Assignment 2

Analysis and Design Document

Student: Nimigean Mihnea

**Group: 30237**

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.

The regular user can perform the following operations:

1. Search books by genre, title, author.
2. Sell books.

The administrator can perform the following operations:

1. CRUD on books (book information: title, author, genre, quantity, and price).
2. 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.

# 1.2 Functional Requirements

The system should permit regular users to search for books and sell books. The administrator of the application is allowed to execute CRUD statements over regular users and generate reports about a specific employee.

# 1.3 Non-functional Requirements

The non-functional requirements of the system include the usage of a layered architectural pattern, an MVC pattern, a Factory Method and ensuring all the data is being verified before being introduced in the database.

2. Use-Case Model

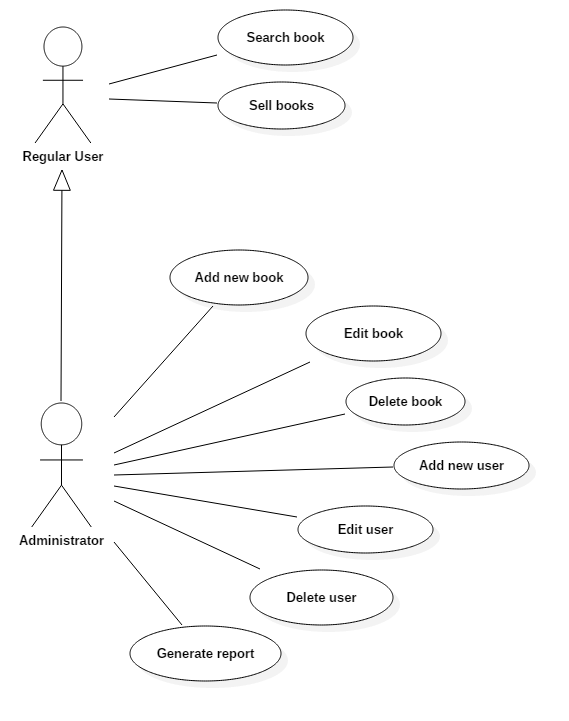
Use case: ***Search users***

Level: user-goal level

Primary actor: Administrator employee (Admin)

Main success scenario:

|  |  |
| --- | --- |
| Action | System Response |
| 1. Access the web application | The login page is displayed |
| 1. Log in | The user is redirected to the home page |
| 1. Click on the ADMIN button | Show admin form |
| 1. Click on the SEARCH button | The search bar would expand |
| 1. Enter a search text and select User from the combo box |  |
| 1. Click on the search button | The user is redirected to the search result page |

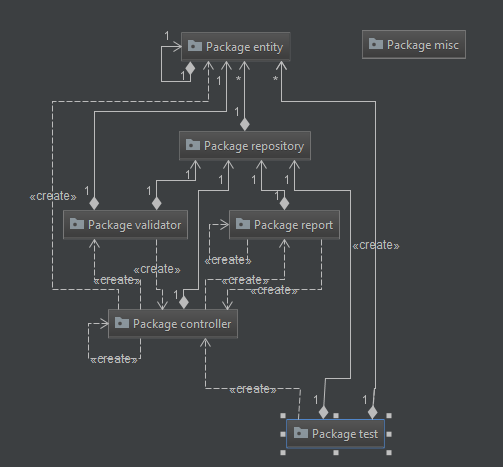


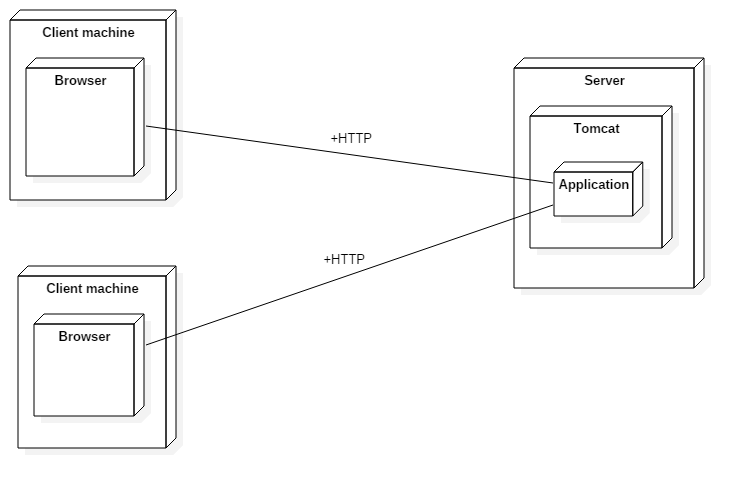
3. System Architectural Design

**3.1 Architectural Pattern Description**

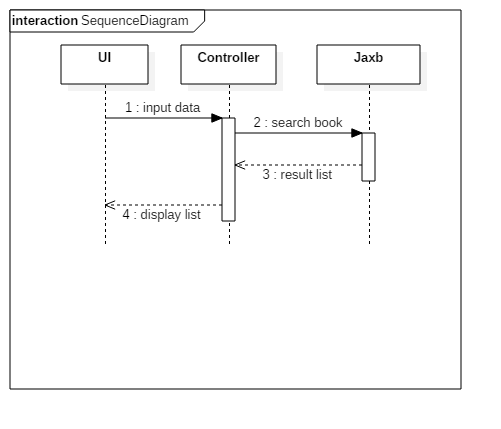
The system uses the layers architectural design pattern. It divides the application in three layers: the presentation layer, which is used to handle the front end of the application, the business layer which handles all the logic of the system (input handling, validations) and the data layer which handles the connection and access to the database.

**3.2 Diagrams**





4. UML Sequence Diagrams

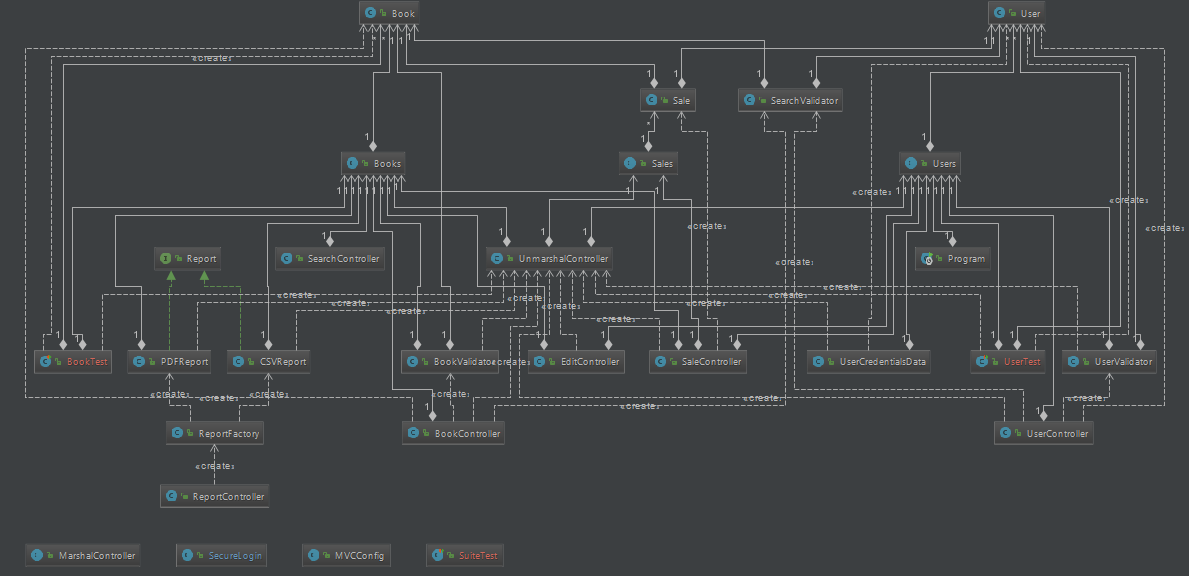


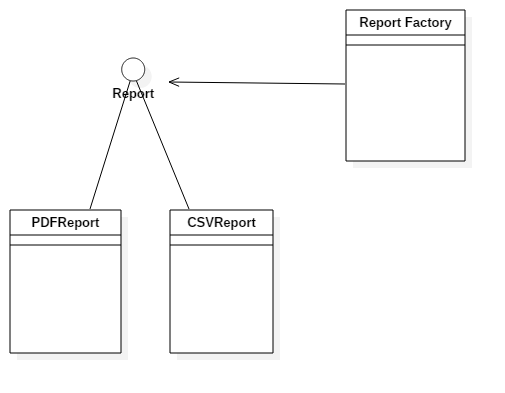
5. Class Design

**5.1 Design Patterns Description**

The Factory Method pattern used in the application is a creational pattern. It creates objects without exposing the creation logic to the client and refers to the newly created object using a common interface.

**5.2 UML Class Diagram**

****



6. Data Model

The data model used for the application was derived from the problem specification. The classes created are the following: Book, User and Sale. For each class there is a “wrapper” class called Books, Users and Sales which represent the classes which will be written in the XML file.

7. System Testing

The application was tested using Junit Test and all the functionality was tested from the web interface in order to ensure the desired functionality.

8. Bibliography

1. Lectures M. Dansoreanu 2017 UTCN
2. Patterns of enterprise application architecture – Martin Fowler
3. Various web resources about Spring and Thymeleaf
4. https://www.tutorialspoint.com/design\_pattern/factory\_pattern.htm