Book Store

Analysis and Design Document

Student: Mocan Ioana

**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

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:

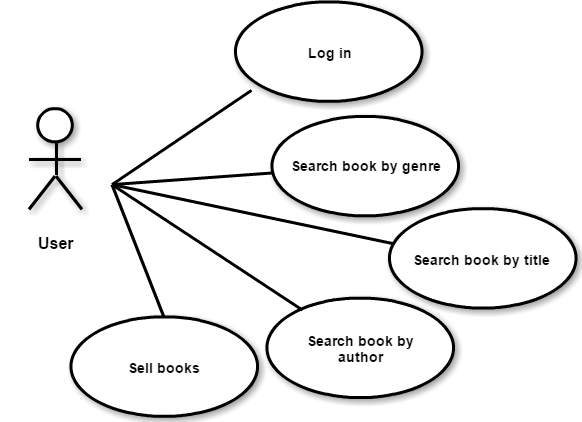
* CRUD on books (book information: title, author, genre, quantity, and price).
* CRUD on regular users’ information.
* Generate two types of reports files, one in pdf format and one in csv format, with the books out of stock.

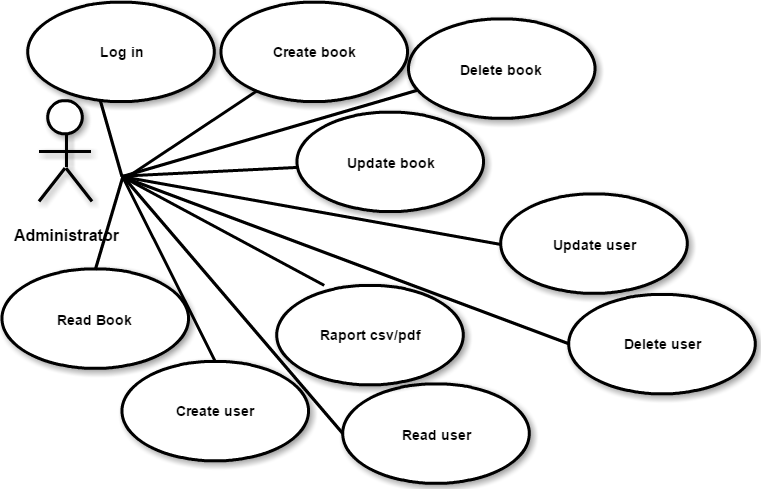
# Non-functional Requirements

The application will have two types of users that can login on the application. The regular user will be asked what option he chooses meaning : to search book by genre , title , author or is selling a book.

The administrator will be sent to another set of options like user , book , report. The first 2 option will send the administrator to another set of option meaning to read , update , create or delete a book or a employee. The last option will be for creating a report of all the books sold in the book store. The information presented are the user that sold the book the name of the book , the author , the genre , the quantity , how much money and in which date.

2. Use-Case Model





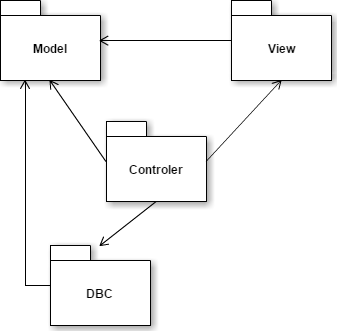
3. System Architectural Design

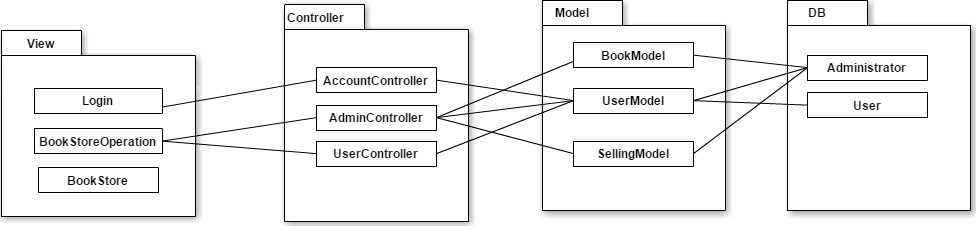
* 1. **Architectural Pattern Description**

**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.

Components

* The model is the central component of the pattern. It expresses the application's behavior in terms of the [problem domain](https://en.wikipedia.org/wiki/Problem_domain), independent of the user interface. It directly manages the data, logic and rules of the application.
* A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
* The third part, the controller, accepts input and converts it to commands for the model or view.
  1. **Diagrams**





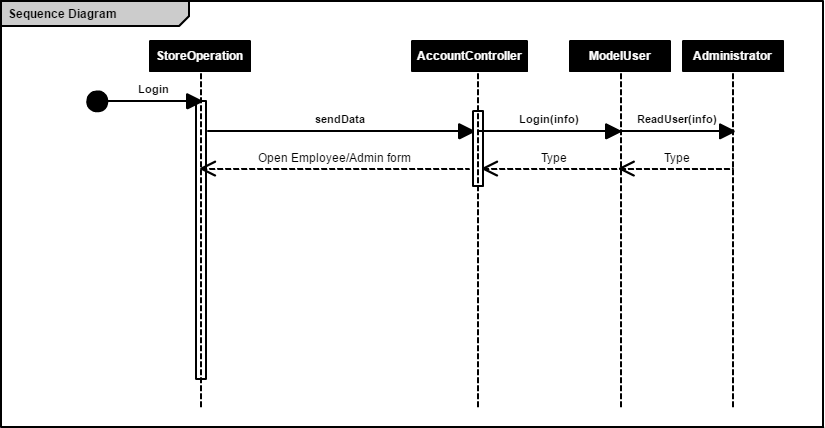
Xml files

BD SERVER

Computer

Book Store System

4. UML Sequence Diagrams

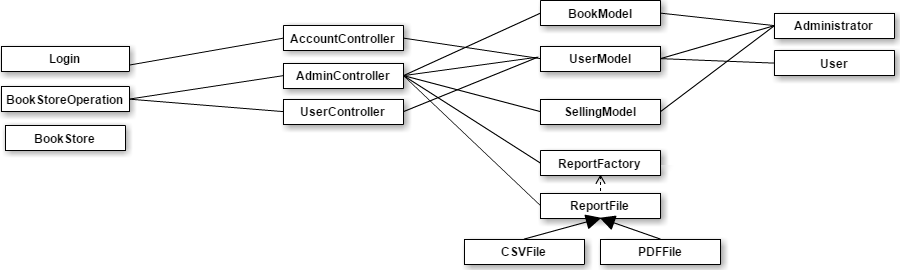


5. Class Design

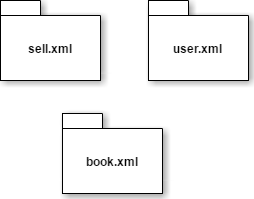
**5.1 Design Patterns Description**

Factory pattern is one of the most used design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object. In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

**5.2 UML Class Diagram**



6. Data Model

**

7. System Testing

I designed a couple of tests such that all the inputs of the application (JTextFields) for the classes form model part which are processing data that will be next insert , update , delete , read , sell book, search books. I also verify in the login part if the username is ok and the password if they match with the ones in the data base and I next send the employer type( employee or admin) to the next window so I can make the difference between the action an employee can do and the admin.

8. Bibliography

* <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
* <https://www.tutorialspoint.com/design_pattern/factory_pattern.htm>
* <https://www.w3schools.com/xml/xml_xpath.asp>
* <http://stackoverflow.com/>
* <https://www.daniweb.com/programming/software-development/threads/255757/inserting-a-new-node-using-xpath-in-an-xml-document>
* <http://io.typepad.com/eben_hewitt_on_java/2007/08/reading-updatin.html>
* <https://examples.javacodegeeks.com/core-java/writeread-csv-files-in-java-example/>
* <https://pdfbox.apache.org/1.8/cookbook/documentcreation.html>