Assignment 2

Analysis and Design Document

Student: Sorin Urcan

**Group: 30442**

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

The objective of this assignment is to allow students to become familiar with the Model View Controller architectural pattern and the Factory Method design pattern.

# Functional Requirements

The regular user can perform the following operations:

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

The administrator can perform the following operations:

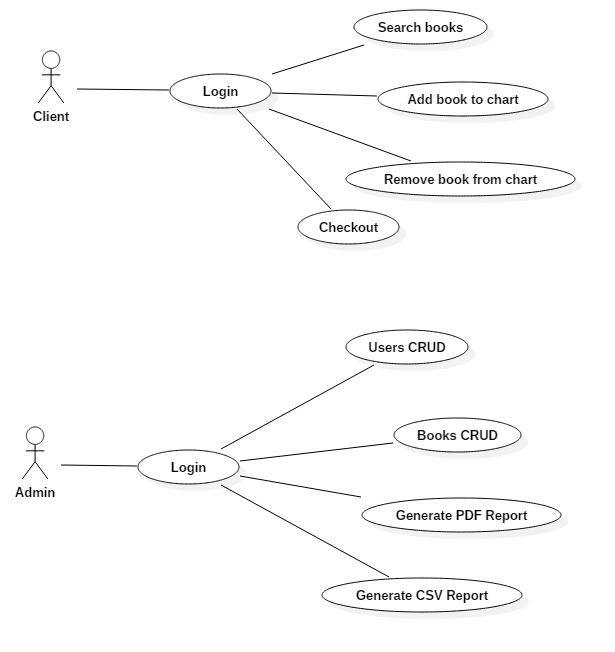
* CRUD on books (book information: title, author, genre, quantity, and price).
* CRUD on regular users’ information.
* Generate two types of reports files, one in pdf format and one in csv format, with the books out of stock.

# Non-functional Requirements

The information about users, books and selling will be stored in multiple XML files. Use the Model View Controller in designing the application. Use the Factory Method design pattern for generating the reports.

All the inputs of the application will be validated against invalid data before submitting the data and saving it.

2. Use-Case Model

**

*Use case: Add User*

*Level: user-goal level*

*Primary actor: Admin*

*Main success scenario:*

- Complete Username and Password fields

- Successfully log in into the Assignment 2 application

- A table containing user fields to complete will appear

- Click the Add User button

- User is created and displayed

*Extensions:*

- Wrong Username or Password will not allow you to enter the application

- Empty user fields will not allow you to add new user

- Used new username will result in a pop-up dialog

*Use case: Remove User*

*Level: user-goal level*

*Primary actor: Admin*

*Main success scenario:*

- Complete Username and Password fields

- Successfully log in into the Assignment 2 application

- A table containing user fields to complete will appear

- Click the Remove User button

- User is deleted and no longer displayed

*Extensions:*

- Wrong Username or Password will not allow you to enter the application

- Empty user fields will not allow you to delete user

*Use case:* *Generate Report*

*Level: user-goal level*

*Primary actor: Admin*

*Main success scenario:*

- Complete Username and Password fields

- Successfully log in into the Assignment 2 application

- A table containing user fields to complete will appear

- Click the *Generate PDF Report* button

- Report is generated and saved into the project folder

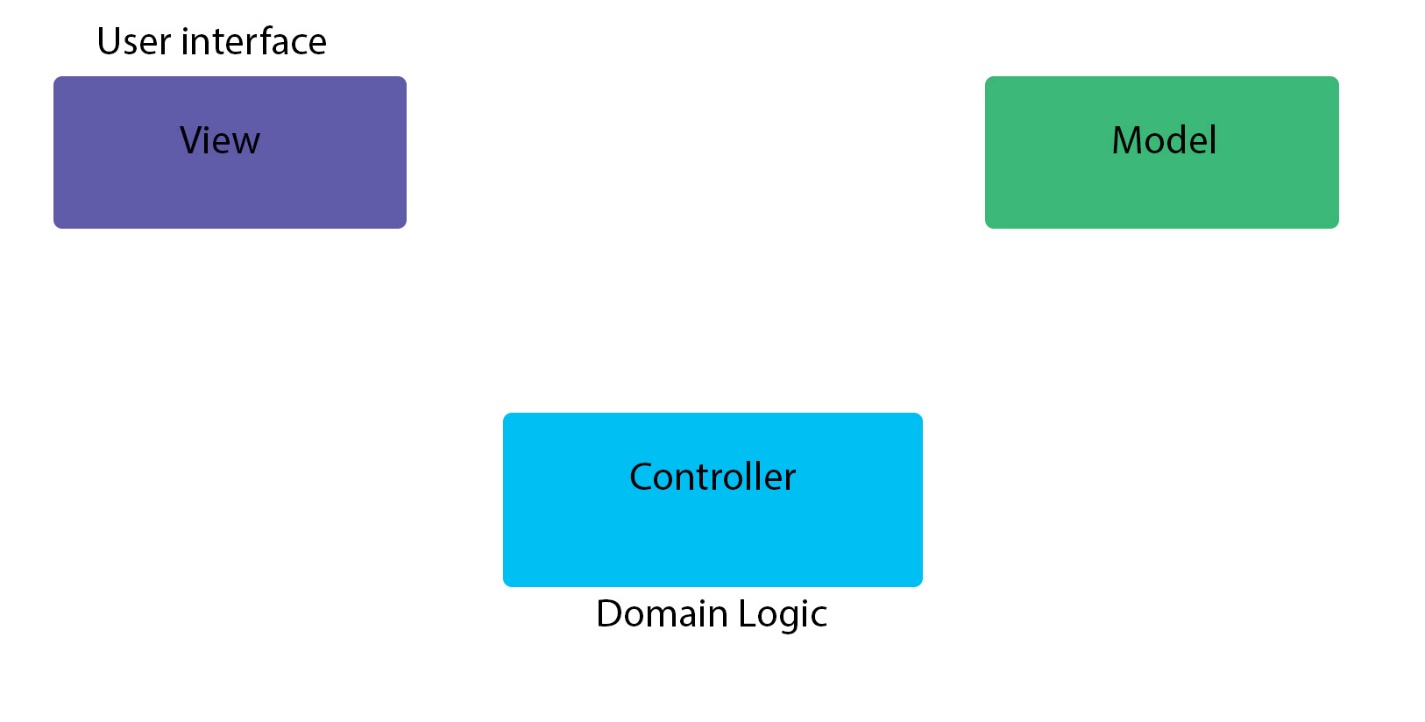
*Extensions:*

- Wrong Username or Password will not allow you to enter the application

3. System Architectural Design

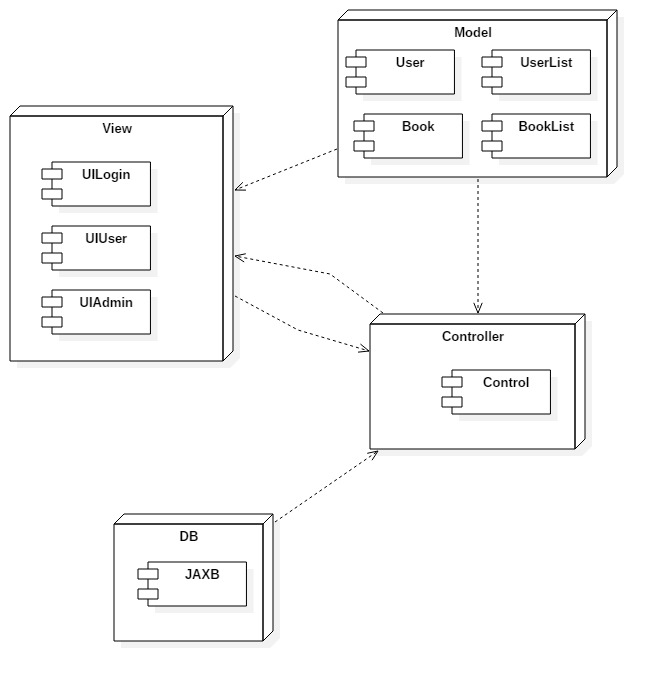
**3.1 Architectural Pattern Description**

The chosen architectural pattern is the model-view-controller. It fits the requirements very well, since it divides the software into three main parts, the model, the view and the controller. For our system, the model is represented by the Book, BookList, User, UserList classes, the view is represented by the user interface and the controller is the domain logic of the system.

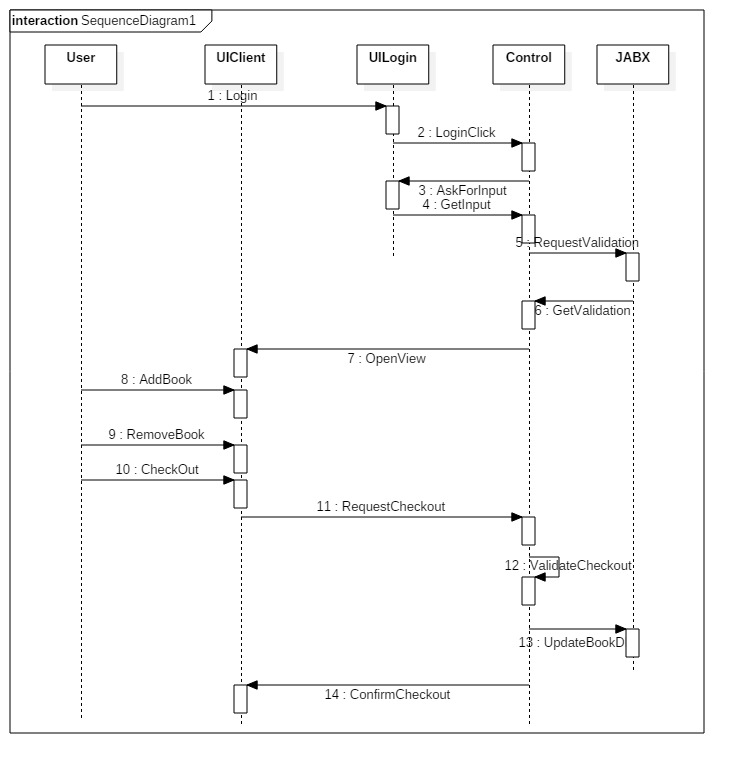
These three components interact with each other. The controller can send commands to the model to change its state, or send commands to the view in order to update its presentation of the model. The view requests information from the model through the controller. By interacting with the view, the user sends commands through the controller, which can modify the model or the view. 

The architectural style is client-server based. This facilitates storage of substantial amounts of data, data which can be accessed remotely. The data will be stored in xml files using JAXB, called “users.xml” and respectively “books.xml”.

**3.2 Diagrams**



4. UML Sequence Diagrams



5. Class Design

**5.1 Design Patterns Description**

*Factory Method is to creating objects as Template Method is to implement an algorithm. A superclass specifies all standard and generic behavior (using pure virtual "placeholders" for creation steps), and then delegates the creation details to subclasses that are supplied by the client.*

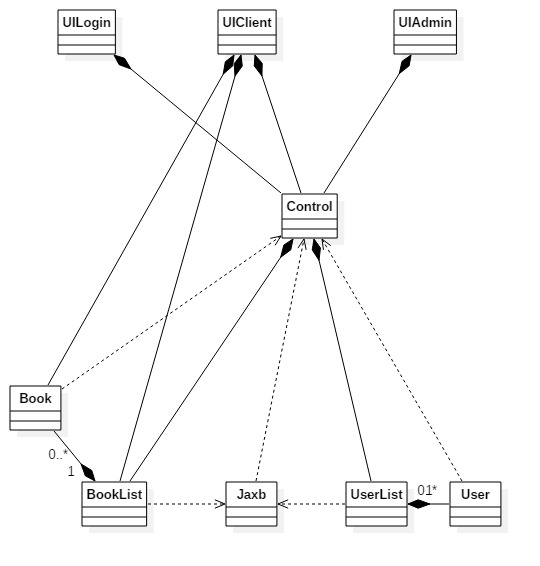
*Factory Method makes a design more customizable and only a little more complicated. Other design patterns require new classes, whereas Factory Method only requires a new operation.*

*People often use Factory Method as the standard way to create objects; but it isn't necessary if: the class that's instantiated never changes, or instantiation takes place in an operation that subclasses can easily override (such as an initialization operation).*

*Factory Method is like Abstract Factory but without the emphasis on families.*

*Factory Methods are routinely specified by an architectural framework, and then implemented by the user of the framework*

**5.2 UML Class Diagram**

**

6. Data Model

*Book* stores a unique ID, title, genre, author, stock and price. User stores ID, username, password, userType and an address. Also, BookList and UserList are an array of Books respectively Users.

//

7. System Testing

Input validation, module testing.

8. Bibliography

<http://docs.oracle.com/javase/tutorial/uiswing/>

<http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>

<http://www.saxproject.org/sax1-roadmap.html>

<http://www.roseindia.net/xml/dom/>

<https://msdn.microsoft.com/en-us/library/54xbah2z(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/e80y5yhx(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlreader.aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlwriter.aspx>

<http://msdn.microsoft.com/en-us/library/ms764730(VS.85).aspx>