Book Store

Analysis and Design Document

Student:Stefan Olimpiu

**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

The application is designed for the employees of a book store. The application has two types of users, a regular user and an administrator. The regular users sell the books and the administrator manages the books and the users’ info.

# Functional Requirements

The administrator can do the following: CRUD on books info, CRUD users’ information and generate report files, pdf and csv formats, with the books out of stock.

The employee can sell books, and search for the books, ordering them by title, author or genre.

# Non-functional Requirements

Extensions can be done to the project through addition of new functionality or through modification of existing functionality, for example, in the future, the user could modify user info.

Security is an important requirement for a bank application. The passwords should require a high complexity and should not be highlighted when introduced.

Readability of the source code is high enough because of its structure and the way it was written. Testing is supported both in manual and automated ways.

2. Use-Case Model

Use-Case description format:

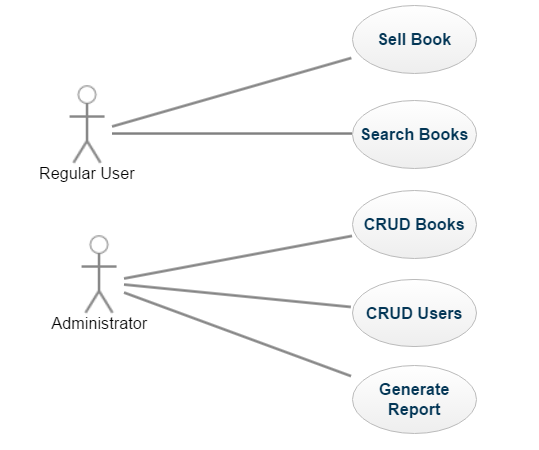
Use case: sell book

Level: user-goal level

Primary actor: regular user (employee)

Main success scenario: the user successfully logins, writes the book’s title in the specific field and clicks the sell button. The quantity is decreased by one.

Extensions: the user wrongly introduce his credentials, after that he finally login, the types in the title of book which doesn’t exist. If this happens, the button does nothing.



3. System Architectural Design

**3.1 Architectural Pattern Description**

MVC is a software architecture - the structure of the system - that separates domain/application/business (whatever you prefer) logic from the rest of the user interface. It does this by separating the application into three parts: the model, the view, and the controller.

The model manages fundamental behaviors and data of the application. It can respond to requests for information, respond to instructions to change the state of its information, and even to notify observers in event-driven systems when information changes. This could be a database, or any number of data structures or storage systems. In short, it is the data and data-management of the application.

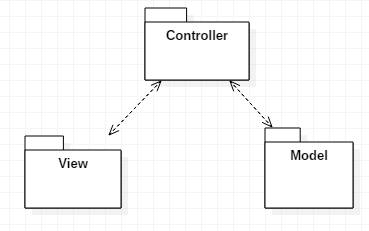
The view effectively provides the user interface element of the application. It'll render data from the model into a form that is suitable for the user interface.

The controller receives user input and makes calls to model objects and the view to perform appropriate actions.

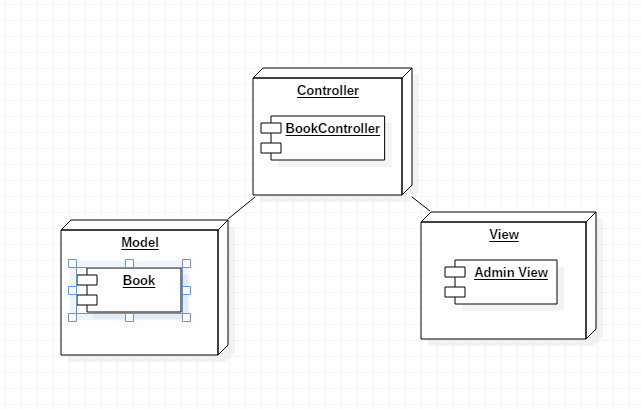
**3.2 Diagrams**

*]*

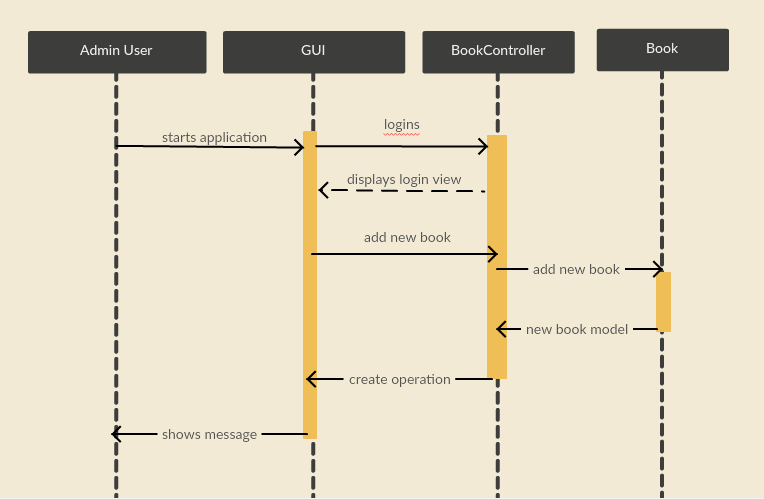
Package Diagram



Component Diagram



4. UML Sequence Diagrams

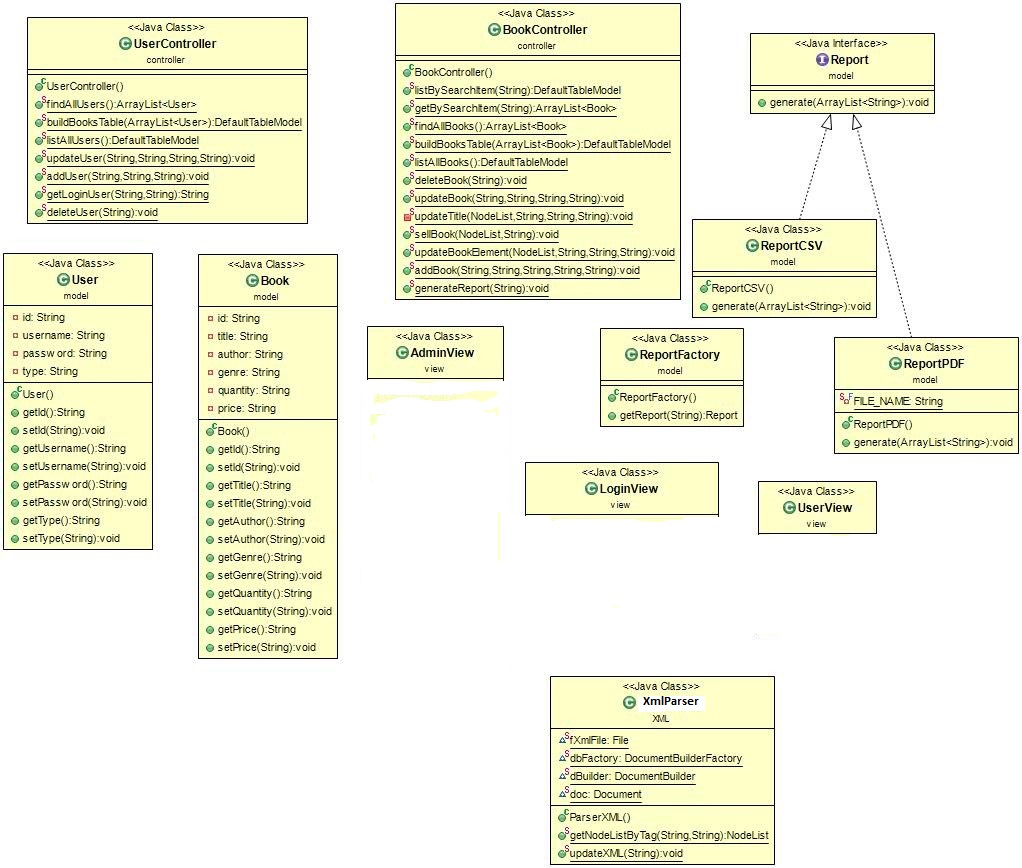


5. Class Design

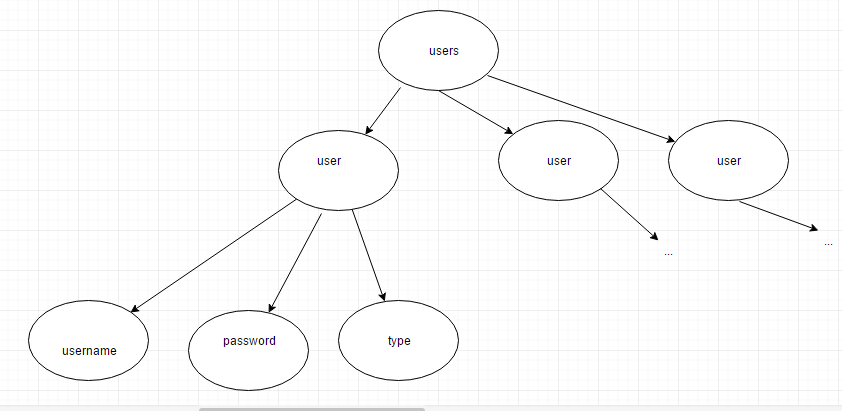
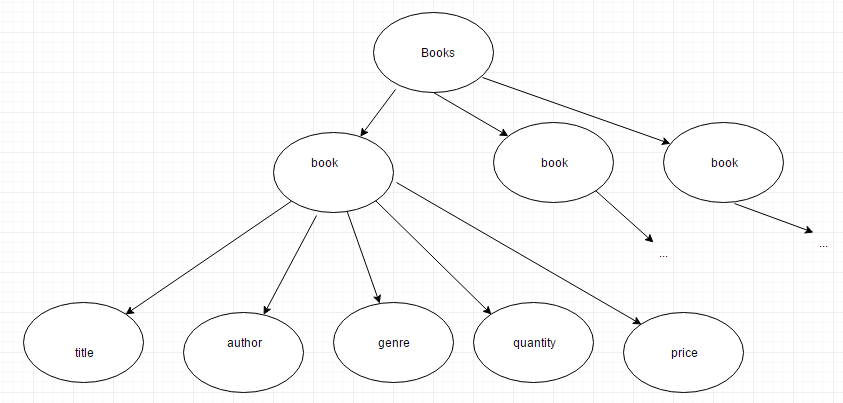
**5.1 Design Patterns Description**

A Factory Pattern or Factory Method Pattern says that just define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate. In other words, subclasses are responsible to create the instance of the class. In our case, I created the Report interface and the ReportCSV and ReportPDF classes which implement this interface. The ReportFactory class has a method, called generate, which returns an ReportCSV or ReportPDF objects, influenced by the value of its parameter.

**5.2 UML Class Diagram**



6. Data Model



7. System Testing

For the main operations the system supports tests: delete, update, create, etc. If something is going wrong the application send an error message to inform the user. In particular, I tested the outcome of the following operations:

1. on login button click, from login view – this takes the user to admin view or user view depending on the introduced credentials
2. on create, list, update, logout, delete books buttons click, CRUD users buttons click, PDF, CSV buttons click (which generate reports with the books out of stock) from admin view
3. on sell, search buttons click, from user view

8. Bibliography

* https://www.martinfowler.com/eaaCatalog/activeRecord.html
* https://www.martinfowler.com/eaaCatalog/tableDataGateway.html
* http://tutorialspoint.com/

https://www.w3schools.com