<BOOK STORE>

Analysis and Design Document

Student: Petrusel Diana-Madalina

**Group: 30231**

Table of Content

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 Chira2018

# Assignment Specification

Desktop application for a book store, with two types of users (regular user and admin), each one of them having their responsibilities.

# Functional Requirements

# *The regular user can perform the following operations:*

# *- Search books by genre, title, author, or any combination of multiple filters.*

# *- Sell books.*

# *The administrator can perform the following operations:*

# *- CRUD on books (book information: title, author, genre, quantity, and price).*

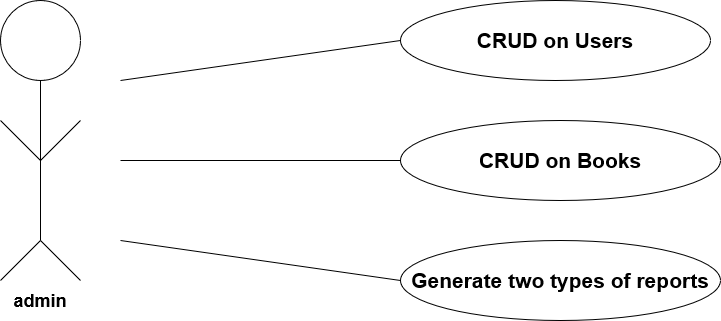
# *- CRUD on regular users’ information.*

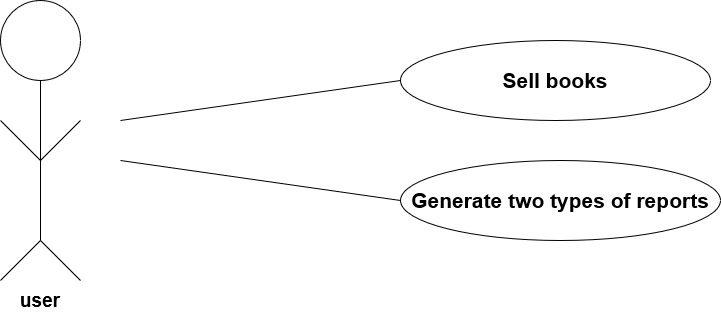
# *- Generate two types of reports files, one in txt format and one in csv format or other, with the books out of stock.*

# Non-functional Requirements

*[Discuss the non-functional requirements for the system]*

2. Use-Case Model



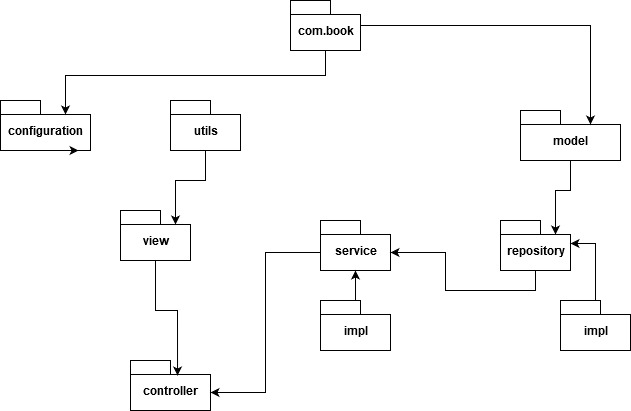


3. System Architectural Design

**3.1 Architectural Pattern Description**

*This application is based on MVC architectural design pattern and Layered Architecture. In order to respect these types of architectures patterns I have structured the classes in multiple packages. For MVC I used packages like model, view and controller and to keep also a layered architecture I used packages like repositories and services.*

**3.2 Diagrams**



4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*

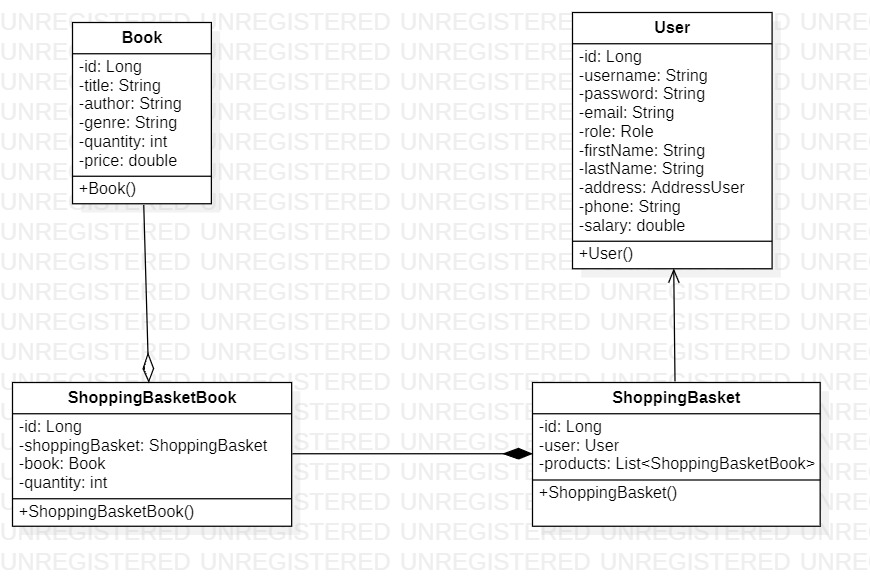
5. Class Design

**5.1 Design Patterns Description**

*I have used the Factory Method design pattern for generating the reports in two types of files: txt and pdf. This method seemed most appropriate for this scenario.*

*I have also used Observer design pattern in order to refresh certain tables when it was necessary. So, I have some observers who are being notified at the right time.*

**5.2 UML Class Diagram**



6. Data Model

*The business logic of the system is carried out by Java objects, while the database provides permanent storage for those objects. Java objects are stored in the database and retrieved later when they are needed.*

*I have used JDBC and MySQL to store all the information needed for later. I have 4 tables: User, Book, Shopping\_Basket and ShoppingBasketBook.*

7. System Testing

*I have run tests during the implementation of the cod, most of them with the help of LOGGER object. Through the user interface I also have introduced multiple messages in order to keep track of possible unanticipated scenarios.*

8. Bibliography

<https://mkyong.com/java8/java-8-streams-filter-examples/>

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

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

<https://www.geeksforgeeks.org/mvc-design-pattern/>

<https://www.codejava.net/java-se/file-io/how-to-read-and-write-text-file-in-java>

<https://www.baeldung.com/java-pdf-creation>

<https://www.codejava.net/java-se/swing/jcheckbox-basic-tutorial-and-examples>