

---

# Online Social Bookstore Application

## Product Backlog Specification

---

---

## VERSIONS HISTORY

---

Date	Version	Description	Author
3/5/24	<1.0>	1st version of the requirements definition document	A. Zarras

## 1 Introduction

---

The objective of this project is to develop an online social bookstore application that allows individuals to exchange used books for free. The rest of this document is structured as follows. In Section 2, we focus on the development process that shall be followed and other scoring and organizational issues. Sections 3 and 4 provide the Product Backlog, i.e., the "raw" functional and non-functional requirements that should be further analyzed to drive the design, implementation and testing of the application.

## 2 Development process and organization issues

---

To realize the project, you can rely on a Scrum approach, i.e., plan a number of sprints during which the team shall implement **user stories** from the project backlog and their **tests**. The deadline for the project is: 20/5/2024.

### 2.1 Deliverables

---

**Definition of "done" story:** A user story is done if it is **implemented correctly** and validated with one or more appropriate **tests**.

At the end of the project the Scrum team shall **deliver (via turnin)**

- **The project implementation.**
- **A Sprint report**, according to the given **Sprint report template (SprintReport-v0.doc)**, describing the sprints that they performed and the "done" user stories that have been developed during each sprint. The report shall also comprise the specification of **detailed use cases, derived from the given user stories** and the detailed **design** of the application.
- Turn in the **project** and the other deliverables using **turnin deliverables@myy803 <your-project>.zip**, where your-project is a zip file of your Eclipse project.

### 2.2 Scoring

---

1. Working implementations of the **user stories** is **60%** of the total score.
2. **Acceptance, integration and unit tests** is **15%** of the total score.
3. **Design quality**, usage of recommended **patterns** and **best practices** to satisfy the extensibility and maintainability requirements is **15%** of the total score.
4. Quality of **reporting** is **10%** of the total score.

## 3 Functional Requirements / User Stories

---

### 3.1 General user stories

US ID	User story
US1	As a <b>user</b> , I want to <b>create a new account</b> , so that I have access to the functionalities of the social bookstore application.
US2	As a <b>user</b> , I want <b>login to my account</b> , so that I have access to the functionalities of the social bookstore application.
US2	As a <b>user</b> , I want to <b>logout from my account</b> , so as to terminate my interaction with the social bookstore application.
US4	As a <b>user</b> , I want to <b>create a profile</b> that includes my <b>full name</b> , <b>address</b> , <b>age</b> , <b>phone number</b> , the <b>categories</b> of <b>books</b> that I prefer (e.g., Art, Comic, Fantasy, Fiction, Biographies, History, Science, Literature, Adventure, Crime, Other) and my favorite <b>authors</b> , to facilitate the search/recommendation of books that may be interesting for me.

### 3.2 User stories for book offers

US ID	User story
US5	As a user, I want to be able to <b>add a book offer</b> in personal list of book offers with a description that includes the book title, the author(s), the category that the book belongs to (Art, Comic, Fantasy, Fiction, Biographies, History, Science, Literature, Adventure, Crime, Other) and a summary of the book, to facilitate the search/recommendation of books by/to other users.
US6	As a user, I want to <b>browse a list of requests</b> from other users who are interested in a book offer that I have made, so that I can decide to whom I shall give the book.
US7	As a user, I want to <b>select a user</b> who <b>requested a book</b> from my list and <b>notify</b> him that he can take the book. I also want to <b>notify</b> the <b>rest of the users</b> who requested the book that the book has been taken by another user.
US8	As a user, I want to have <b>access</b> to the <b>contact information</b> of a <b>selected user</b> who requested a book that I offer, so that I can contact him to arrange the delivery of the book.
US9	As a user, I want to <b>remove a book</b> that is no longer <b>available</b> from <b>my personal list</b> of book offers and from the <b>book request lists</b> of other <b>users</b> who requested for this book, to enable the accurate book search/recommendation.

### 3.3 User stories for book requests

US ID	User story
US10	As a user, I want to be able to <b>search for book offers</b> with a certain title and authors to find interesting books. Besides specifying the search criteria, I want to <b>choose between</b> an exact or an <b>approximate</b> search <b>strategy</b> . Then, I want to be able to <b>make a request</b> for a book that is <b>included</b> in the <b>search results</b> to the user who offers the book so that he becomes aware that I want to get the book.
US11	As a user, I would like to <b>browse</b> a list of <b>recommended book offers</b> from the social bookstore application, to find interesting books. It would be nice to be able to choose among various recommendation strategies that consider information given in my profile. Then, I want to be

Recommendations base on profile

Filtering

	able to make a request for a recommended book to the user who offers the book so that he becomes aware that I want to get the book.
--	---

## 4 Non-Functional Requirements

---

**[NF1] Maintainability:** In software engineering, maintainability is the degree of effectiveness and efficiency with which a product or system can be modified by the maintainers. In the case of this project, we specifically focus on the following concerns:

- **[NF1.1]** A first concern is to employ an architecture that promotes low coupling and high cohesion. To this end, you can employ **Fowler's enterprise application architecture patterns** [2] that allow to clearly separate the different parts (views, controllers, domain model, database) of the application and facilitate the mapping of the domain model to the underlying database schema.
- **[NF1.2]** In the long term, we want to be able to **easily extend** and **configure** the application with **different book recommendation strategies**. The initial set of **recommendation strategies** that will be supported by the application should include the following strategies.
  - The first strategy recommends a list of books that belong to the categories the user prefers, as specified in the user's profile.
  - The second strategy recommends a list that contains books written by the user's preferred authors, as specified in the user's profile.
  - The third strategy is a composite strategy that combines the first two - it recommends a list of books written by the user's preferred authors and books that belong to the categories the user prefers, as specified in the user's profile.
- **[NF1.3]** In the long term we want to be able to **easily extend** and **configure** the application with **different search strategies, without having to change the code of the application**. We would also like to **avoid duplicate code** that may result from the implementation of similar, but different strategies. The initial set of **search strategies** that will be supported by the application should include the following strategies.
  - The first (exact) strategy should produce a list of books such that the title and the authors of each book match exactly the book title and authors, specified in the query.
  - The second (approximate) strategy should produce a set of books such that the title of each book contains the title specified in the query, and the authors list of each book includes at least the authors, specified in the query.

**HINT** To achieve the **[NF1.2]** and **[NF1.3]** concerns the application should be designed according to well-known **principles** and exploit **best practices** like the **GoF design patterns**. **Specifically, consider using the GoG strategy pattern and the GoF template method pattern for the implementation of alternative/interchangeable strategies.**

**[NF2] Usability:** In software engineering, usability concerns the ease of use and learnability. In the context of this project the application should provide a simple and user-intuitive interface. The application should also provide help, in the form of user guidelines, concerning its main functionalities of the application.

## 5 Technical Requirements/Constraints/Recommendations

---

Following, there is a list of technologies, frameworks and tools to consider for the development of the project.

- Java
- Spring Boot
- MySql
- Junit, Mockito
- Eclipse, IntelliJ
- PlantUML

## 6 References

---

[1] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

[2] Martin Fowler. *Catalog of Patterns of Enterprise Application Architecture*. Addison-Wesley.  
<https://martinfowler.com/eaCatalog/>