

---

# Online Social Bookstore Application

## Sprint Report

---

Aspiring CSE Diploma Holders

Kondylia Vergou, 4325

Panagiotis Vouzalis, 2653

Panagiotis Paraskevopoulos, 2905

---

## VERSIONS HISTORY

---

Date	Version	Description	Author
11/03/2024	1.0	Definition of User Stories and Use Cases	Lia Vergou
22/03/2024	2.0	CRC cards for Sprint 1	Lia Vergou
04/04/2024	2.1	CRC cards for Sprint 2	Lia Vergou
19/04/2024	2.2	CRC cards for Sprint 3	Lia Vergou
01/05/2024	2.3	CRC cards for Sprint 4	Lia Vergou
20/05/2024	2.4	CRC cards for Sprint 5	Lia Vergou

## 1 Introduction

---

The objective of this project is to develop an online social bookstore application that allows individuals to exchange used books for free.

### 1.1 Purpose

---

This document provides information concerning the last sprint of this project.

### 1.2 Document Structure

---

The rest of this document is structured as follows:

- Section 2 describes our Scrum team and specifies the Sprint's backlog.
- Section 3 specifies the main design concepts for this release of the project.

## 2 Scrum team and Sprint Backlog

---

### 2.1 Scrum team

---

<b>Product Owner</b>	Kondylia Vergou
<b>Scrum Master</b>	Panagiotis Vouzalis
<b>Development Team</b>	Kondylia Vergou, Panagiotis Vouzalis, Panagiotis Paraskevopoulos

## 2.2 Sprints

---

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	11/03/2024	22/03/2024	2	US1, US2, US3
2	25/03/2024	04/04/2024	2	US4
3	08/04/2024	19/04/2024	2	US5, US9
4	22/04/2024	01/05/2024	1.5	US10, US11
5	08/05/2024	20/05/2024	1.5	US6, US7, US8

## 2.3 User Stories

---

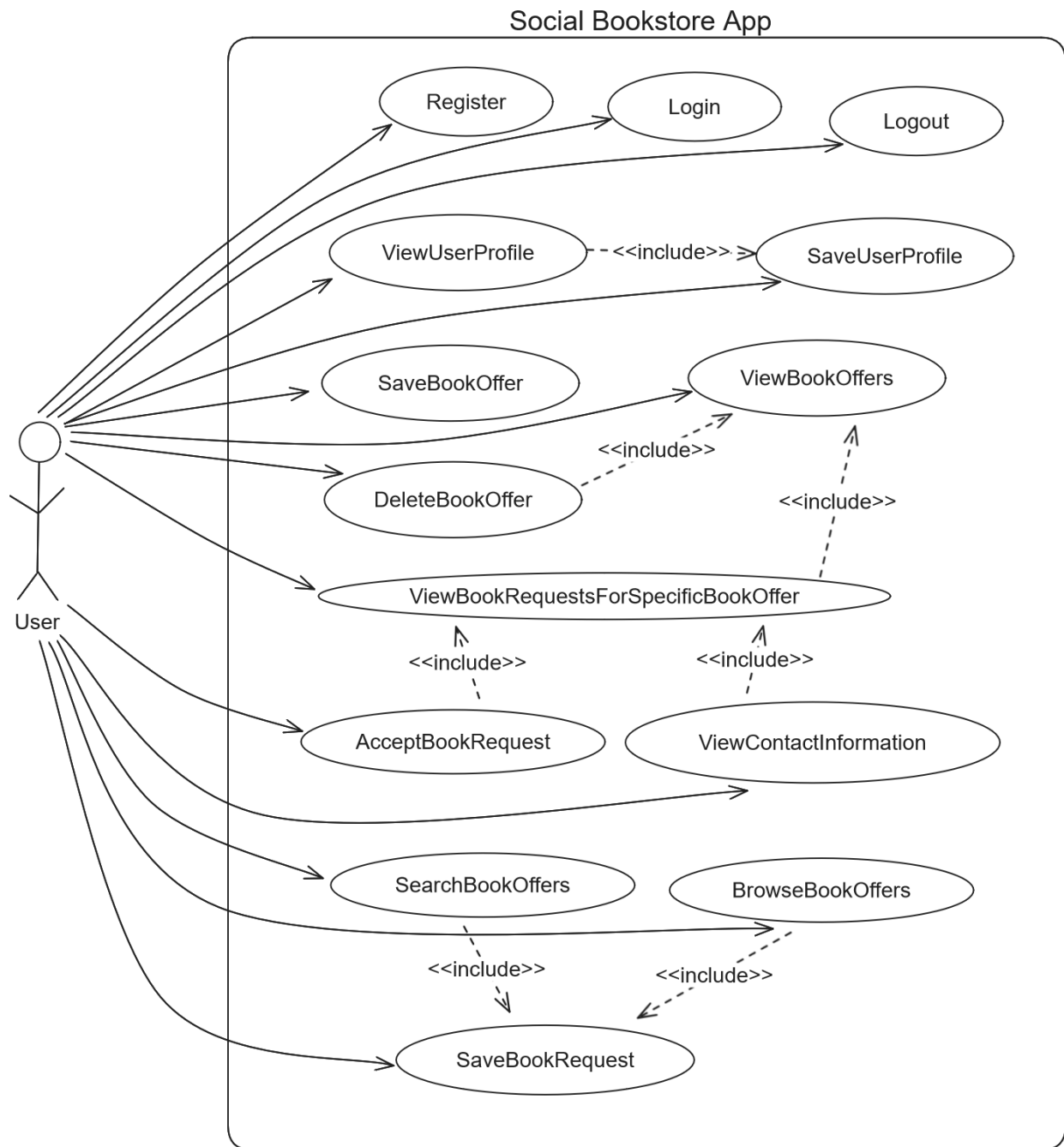
US ID	AS A <USER TYPE>	I WANT TO <AN ACTION>	IN ORDER TO <A BENEFIT/VALUE>
US1	User	Create a new account	Access the functionalities of the social bookstore application
US2	User	Login to my account	Access the functionalities of the social bookstore application
US3	User	Logout from my account	Terminate my interaction with the social bookstore application

US4	User	Create a profile that includes my full name, address, age, phone number, preferred book categories and favorite authors	Facilitate the search/recommendation of books that may be interesting for me
US5	User	Add a book offer in personal list of book offers with a description that includes the book title, the author(s), the category that the book belongs to and a summary of the book	Facilitate the search/recommendation of books by/to other users
US6	User	Browse a list of requests from other users who are interested in a book offer that I have made	Decide to whom I shall give the book
US7	User	Select a user who requested a book from my list and notify him that he can take the book. Notify the rest of the users who requested the book that the book has been taken by another user	Initiate the book exchange with the selected user
US8	User	Access to the contact information of a selected user who requested a book that I offer	Contact the selected user to arrange the delivery of the book
US9	User	Remove a book that is no longer available from my personal list of book offers and from the book request lists of other users who requested for this book	Enable the accurate book search/recommendation

US10	User	<p>Search for book offers with a certain title and authors to find interesting books</p> <p>Choose between an exact or an approximate search strategy</p> <p>Make a request for a book that is included in the search results to the user who offers the book</p>	Make the user that offers the book aware of the request
US11	User	<p>Browse a list of recommended book offers from the social bookstore application, to find interesting books</p> <p>Choose among various recommendation strategies that consider information given in my profile</p> <p>Make a request for a recommended book to the user who offers the book so</p>	Make the user that offers the book aware of the request

### 3 Use Cases

1: UML USE CASE DIAGRAM



### 3.1 UC01: Register

---

<b>Use case name</b>	Register
<b>Use case ID</b>	01
<b>Actors</b>	User
<b>Preconditions</b>	-
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user chooses to register.</li><li>2. The user provides their username and password.</li><li>3. The system validates that the username isn't taken.</li><li>4. The system registers the user.</li><li>5. The system redirects the user to the login page.</li></ol>
<b>Alternative flow</b>	If in step 3 the username is taken, then the system displays an error message and cancels the registration process.
<b>Post conditions</b>	The user is successfully registered.

### 3.2 UC02: Login

---

<b>Use case name</b>	Login
<b>Use case ID</b>	02
<b>Actors</b>	User
<b>Preconditions</b>	-
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user chooses to login.</li><li>2. The user provides their username and password.</li><li>3. The system validates the credentials.</li><li>4. The system logs in the user.</li></ol>
<b>Alternative flow</b>	If in step 3 the credentials are wrong (Username Doesn't Exist or Wrong Password), then the system displays an error message and cancels the login process.
<b>Post conditions</b>	The user is successfully logged in.



### 3.3 UC03: Logout

---

<b>Use case name</b>	Logout
<b>Use case ID</b>	03
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user chooses to logout.</li><li>2. The system logs out the user.</li></ol>
<b>Post conditions</b>	The user is successfully logged out.

### 3.4 UC04: SaveUserProfile

---

<b>Use case name</b>	SaveUserProfile
<b>Use case ID</b>	04
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(ViewUserProfile)</li><li>2. The user changes their full name, address, age, phone number, favorite book categories and favorite authors.</li><li>3. The system saves the profile details provided.</li></ol>
<b>Post conditions</b>	The user profile details are saved.

### 3.5 UC05: ViewUserProfile

---

<b>Use case name</b>	ViewUserProfile
<b>Use case ID</b>	05
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user selects their profile.</li><li>2. The system displays their full name, address, age, phone number, favorite book categories and favorite authors.</li></ol>
<b>Post conditions</b>	The user profile details are displayed.

### 3.6 UC06: SaveBookOffer

---

<b>Use case name</b>	SaveBookOffer
<b>Use case ID</b>	06
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user selects to add a book offer.</li><li>2. The user provides the book title, the author(s), the category that the book belongs to and a summary of the book.</li><li>3. The system records the book offer.</li></ol>
<b>Post conditions</b>	A book offer is saved.

### 3.7 UC07: ViewBookRequestsForSpecificBookOffer

---

<b>Use case name</b>	ViewBookRequestsForSpecificBookOffer
<b>Use case ID</b>	07
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(ViewBookOffers)</li><li>2. The user selects one of their book offers.</li><li>3. The system displays a list of requests from other users for the specific book offer.</li></ol>
<b>Alternative flow</b>	If in step 1 the system displays the message “No Book Offers Found”, then the use case is terminated.
<b>Post conditions</b>	The book requests for the specific book offer are listed.

### 3.8 UC08: ViewBookOffers

---

<b>Use case name</b>	ViewBookOffers
<b>Use case ID</b>	08
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user selects to view their book offers.</li><li>2. The system displays a list of their book offers.</li></ol>
<b>Alternative flow</b>	If in step 2 the system doesn't find any book offers, then it displays the message "No Book Offers Found".
<b>Post conditions</b>	The book offers of the specific user are listed.

### 3.9 UC09: AcceptBookRequest

---

<b>Use case name</b>	AcceptBookRequest
<b>Use case ID</b>	09
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in, has made at least one book offer and there is at least one book request for the specific offer.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(ViewBookRequestsForSpecificBookOffer)</li><li>2. The user selects one of the book requests.</li><li>3. The user approves the request.</li><li>4. The system notifies the user of the selected book request that their request has been approved.</li><li>5. The system notifies the users of the rest book request that their requests have been denied.</li></ol>
<b>Post conditions</b>	The selected book request is approved and the rest are denied.

### 3.10 UC10: ViewContactInformation

---

<b>Use case name</b>	ViewContactInformation
<b>Use case ID</b>	10
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in, has made at least one book offer and there is at least one book request for the specific offer.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(ViewBookRequestsForSpecificBookOffer)</li><li>2. The user selects one of the book requests.</li><li>3. The user chooses to view the contact information of the requestee.</li><li>4. The system displays the contact information.</li></ol>
<b>Post conditions</b>	The contact information of the selected book request is displayed.

### 3.11 UC11: DeleteBookOffer

---

<b>Use case name</b>	DeleteBookOffer
<b>Use case ID</b>	11
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in and has made at least one book offer.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(ViewBookOffers)</li><li>2. The user selects one of their book offers and deletes it.</li><li>3. The system deletes the book offer.</li></ol>
<b>Post conditions</b>	The book offer is deleted.

### 3.12 UC12: SearchBookOffers

---

<b>Use case name</b>	SearchBookOffers
<b>Use case ID</b>	12
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user chooses to search for book offers.</li><li>2. The user provides a certain title and authors.</li><li>3. The user selects an exact or an approximate search strategy.</li><li>4. The system displays the list of book offers that fit the criteria.</li></ol>
<b>Alternative flow</b>	If in step 4 the system doesn't find any book offers, then it display the message "No Book Offers Were Found".
<b>Post conditions</b>	A list of book offers is displayed.

### 3.13 UC13: BrowseBookOffers

---

<b>Use case name</b>	BrowseBookOffers
<b>Use case ID</b>	13
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user chooses to browse the book offers.</li><li>2. The user selects one of the recommendation strategies that utilize their profile details.</li><li>3. The system displays the list of book offers according to the strategy.</li></ol>
<b>Alternative flow</b>	If in step 4 the system doesn't find any book offers, then it displays the message "No Book Offers Were Found".
<b>Post conditions</b>	A list of book offers is displayed.

### 3.14 UC14: SaveBookRequest

---

<b>Use case name</b>	SaveBookRequest
<b>Use case ID</b>	13
<b>Actors</b>	User
<b>Preconditions</b>	The user is logged in.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. Include(SearchBookOffers) or Include(BrowseBookOffers)</li><li>2. The user selects a specific book offer.</li><li>3. The user makes a book request.</li><li>4. The system records the book request.</li></ol>
<b>Alternative flow</b>	If in step 1 the system displays the message “No Book Offers Were Found”, then the use case is terminated.
<b>Post conditions</b>	The book request is saved.

//TODO Some extra use cases are:

- Delete Book Request
- ViewBookRequests (που εχει κανει ο χρηστης)

## 4 Design

---

### 4.1 Architecture

---

<Specify the overall architecture for this release in terms of a **UML package diagram**.>

### 4.2 Design

---

PACKAGE UML DIAGRAM

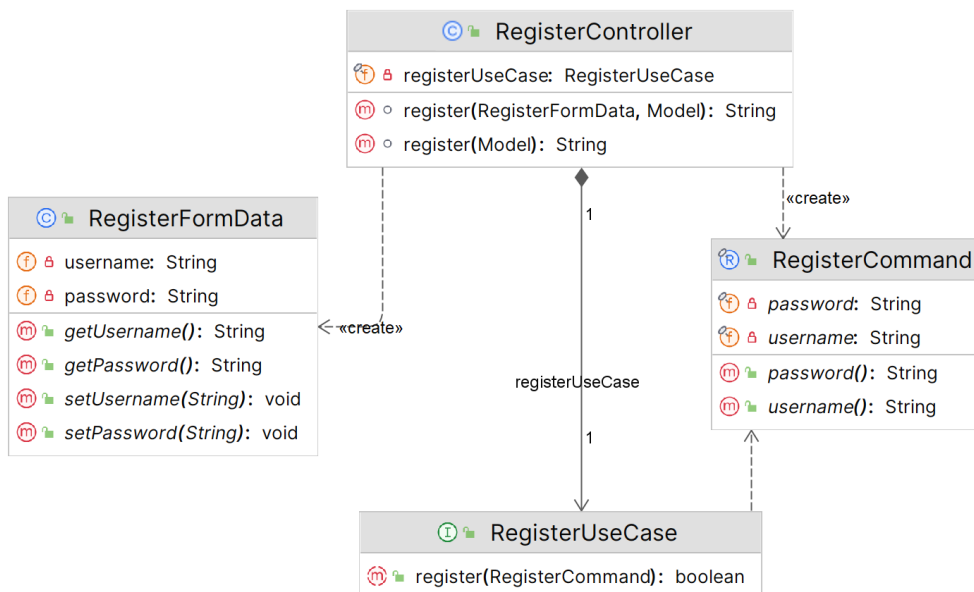
UML DIAGRAM OF PACKAGE CONFIG

UML DIAGRAM OF PACKAGE SERVICES with collaborating classes

UML DIAGRAM OF MAPPERS with collaborating classes

UML DIAGRAM OF DOMAIN

FIGURE 2: UML DIAGRAM OF PACKAGE CONTROLLERS WITH COLLABORATING CLASSES



---

#### 4.2.1 CONFIG

---

package com.myy803.social_bookstore.config	
<b>Class Name:</b> WebSecurityConfig	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Configures web security for the application.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• PasswordEncoder</li><li>• UserDetailsService</li><li>• CustomSecuritySuccessHandler</li><li>• EndpointConfig</li></ul>

package com.myy803.social_bookstore.config	
<b>Class Name:</b> CustomSecuritySuccessHandler	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Handle successful authentication by redirecting users to a specific target URL.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• EndpointConfig</li></ul>

package com.myy803.social_bookstore.config	
<b>Class Name:</b> WebMvcConfig	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Configures MVC (Model-View-Controller) for web application.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• ViewControllerRegistry</li></ul>

package com.myy803.social_bookstore.config	
<b>Class Name:</b> PasswordEncoderConfig	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Configures a bean for password encoding.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.config	
<b>Class Name:</b> EndpointConfig	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define constants for endpoint paths used in the application.</li></ul>	<b>Collaborations:</b> None



package com.myy803.social_bookstore.config	
<b>Class Name:</b> ThymeleafUtil	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Extracts a list of IDs from a list of View objects.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>BookCategoryView</li> <li>AuthorView</li> </ul>

---

#### 4.2.2 CONTROLLERS

---

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> RegisterController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles incoming HTTP requests related to user registration.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> <li>RegisterFormData</li> <li>Model</li> <li>RegisterCommand</li> <li>RegisterUseCase</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> LoginController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles HTTP GET requests for the login page.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> HomepageController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles HTTP GET requests for the homepage.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> ViewUserProfileController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles HTTP GET requests for viewing user profile.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> <li>ViewUserProfileUseCase</li> <li>ViewBookCategoriesUseCase</li> <li>ViewAuthorsUseCase</li> <li>Principal</li> <li>Model</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> SaveUserProfileController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles HTTP requests for saving user profile data.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> <li>SaveUserProfileUseCase</li> <li>UserProfileFormData</li> <li>Principal</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> SaveBookOfferController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Handles HTTP requests for saving book offers</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>EndpointConfig</li> <li>ViewBookCategoriesUseCase</li> <li>ViewAuthorsUseCase</li> <li>BookFormData</li> </ul>

package com.myy803.social_bookstore.controllers	
<b>Class Name:</b> ViewBookOffersController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Handles HTTP GET requests for viewing book offers</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>• EndpointConfig</li> <li>• ViewBookOffersUseCase</li> <li>• Principal</li> <li>• Model</li> </ul>

---

### 4.2.3 SERVICES

---

package com.myy803.social_bookstore.services	
<b>Class Name:</b> RegisterUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Defines the contract for registering a user.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• RegisterCommand</li></ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> RegisterService	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Implements the RegisterUseCase interface to handle user registration.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• RegisterCommand</li><li>• PasswordEncoder</li><li>• UserMapper</li><li>• User</li><li>• Role</li></ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> UserDetailsServiceImpl	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Implements the UserDetailsService interface to load user details by username.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• UserMapper</li></ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewUserProfileUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Defines the contract for viewing a user profile.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewUserProfileService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the ViewUserProfileUseCase interface to map the UserProfile domain model to the UserProfileView view object.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>UserMapper</li> <li>UserProfileMapper</li> <li>UserProfileView</li> <li>BookCategoryView</li> <li>AuthorView</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewBookCategoriesUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Defines the contract for viewing all the book categories.</li> </ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewBookCategoriesService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the ViewBookCategoriesUseCase interface to map the BookCategory domain model to the BookCategoryView view object.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>BookCategoryMapper</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewAuthorsUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Defines the contract for viewing all the authors.</li> </ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewAuthorsService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the ViewAuthorsUseCase interface to map the Author domain model to the AuthorView view object.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>AuthorMapper</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> SaveUserProfileUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Defines the contract for saving user profile data.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>SaveUserProfileCommand</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> SaveUserProfileService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the SaveUserProfileUseCase interface to save the user profile data.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>UserMapper</li> <li>UserProfileMapper</li> <li>BookCategoryMapper</li> <li>AuthorMapper</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> SaveBookOfferUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Defines the contract for saving book offers.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>SaveBookOfferCommand</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> SaveBookOfferService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the SaveBookOfferUseCase interface to save book offers.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>UserMapper</li> <li>UserProfileMapper</li> <li>BookCategoryMapper</li> <li>AuthorMapper</li> <li>BookMapper</li> </ul>

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewBookOffersUseCase	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Defines the contract for viewing book offers.</li> </ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.services	
<b>Class Name:</b> ViewBookOffersService	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Implements the ViewBookOffersUseCase interface to view book offers.</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>• UserMapper</li> <li>• UserProfileMapper</li> <li>• Author</li> <li>• Book</li> </ul>

---

#### 4.2.4 MAPPERS

---

package com.myy803.social_bookstore.mappers	
<b>Class Name:</b> UserMapper	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define methods for accessing and manipulating user data in the database</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• JpaRepository</li><li>• User</li></ul>

package com.myy803.social_bookstore.mappers	
<b>Class Name:</b> UserProfileMapper	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define methods for accessing and manipulating user profile data in the database.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• JpaRepository</li><li>• UserProfile</li></ul>

package com.myy803.social_bookstore.mappers	
<b>Class Name:</b> BookCategoryMapper	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define methods for accessing and manipulating book category data in the database.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• JpaRepository</li><li>• BookCategory</li></ul>

package com.myy803.social_bookstore.mappers	
<b>Class Name:</b> AuthorMapper	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define methods for accessing and manipulating author data in the database.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• JpaRepository</li><li>• Author</li></ul>

package com.myy803.social_bookstore.mappers	
<b>Class Name:</b> BookMapper	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Define methods for accessing and manipulating book data in the database.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>• JpaRepository</li><li>• Book</li></ul>



---

#### 4.2.5 DOMAIN FORMSDATA

---

package com.myy803.social_bookstore.domain.formsdata	
<b>Class Name:</b> RegisterFormData	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers data from the register view page to the RegisterController.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.formsdata	
<b>Class Name:</b> UserProfileFormData	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers data from the user profile view page to the ViewUserProfileController and SaveUserProfileController.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.formsdata	
<b>Class Name:</b> BookFormData	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers data from the save book offer view page to the SaveBookOfferController.</li></ul>	<b>Collaborations:</b> None

---

#### 4.2.6 DOMAIN VIEWS

---

package com.myy803.social_bookstore.domain.views	
<b>Class Name:</b> UserProfileView	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers user profile data from the services layer to the ViewUserProfileController.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>BookCategoryView</li><li>AuthorView</li></ul>

package com.myy803.social_bookstore.domain.views	
<b>Class Name:</b> BookCategoryView	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers book data from the services layer to the ViewUserProfileController.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.views	
<b>Class Name:</b> AuthorView	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers author data from the services layer to the ViewUserProfileController.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.views	
<b>Class Name:</b> BookOfferView	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Transfers book offer data from the services layer to the ViewBookOffersService.</li></ul>	<b>Collaborations:</b> None

---

#### 4.2.7 DOMAIN COMMANDS

---

package com.myy803.social_bookstore.domain.commands	
<b>Class Name:</b> RegisterCommand	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Transfers data from the RegisterController to the services layer.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.commands	
<b>Class Name:</b> SaveUserProfileCommand	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Transfers data from the SaveUserProfileController to the services layer.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.commands	
<b>Class Name:</b> SaveBookOfferCommand	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Transfers data from the SaveBookOfferController to the services layer.</li></ul>	<b>Collaborations:</b> None

---

#### 4.2.8 DOMAIN MODELS

---

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b> User	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Represents a user entity with attributes such as username, password, and role.</li><li>Implements the UserDetails interface, providing user authentication and authorization details.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>Role</li><li>GrantedAuthority</li><li>UserProfile</li></ul>

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b> Role	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Represents the role of a user.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b> UserProfile	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Represents the user profile data.</li></ul>	<b>Collaborations:</b> <ul style="list-style-type: none"><li>User</li><li>BookCategory</li><li>Author</li></ul>

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b> BookCategory	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Represents the book category data.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b> Author	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>Represents the author data.</li></ul>	<b>Collaborations:</b> None

package com.myy803.social_bookstore.domain.models	
<b>Class Name:</b>	Book
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Represents a book entity with its properties.</li> </ul>	<b>Collaborations:</b> None

### 4.3 Tests

---

Tests have been created for the controllers, services, and mappers of the application.

- The controllers are tested with integration tests to verify the interactions between the different components and all three layers, the web, services and database layer.
- The services are tested with unit tests to focus on the functionality and logic.
- The mappers are tested with integration tests since they interact with the database.

The acceptance tests for each use case have been performed through the UI.

## 5 Thank you Note!

---

Thank you for reading our Report!

Lia Vergou  
Notis Vouzalis  
Panagiotis Paraskevopoulos

