

# LiBook

*Represents modern online library with simple and user-friendly UI to interact with information about books and authors.*

The goal of the system is to simplify the process of searching books and authors for user. The service provides quick access to the main information about books and their authors available across the system. The system allows to manage items (books and authors) in a really easy way using comfortable modern interface. In addition, amongst options available, an end user can add books to the wish list and read them if it is available. Users also can leave comments to book which they like.

## **Roles**

1. **Admin** – user of the system who can manage all information about books and authors (fulfil CRUD operations with authors and books)
2. **Registered user** – target wearer of the system who has such opportunities as searching and retrieving information about authors and books; can add books to his/her wish list; leave comments about book and like own or other comments.
3. **Guest** – person who has only availability to search and view information about books and authors.

## Subsystems

### *Author Subsystem*

№	Role	Use Case	Description	Action Method	Tests
AS1	Admin	Edit author	Updates information about author such as general information (first name, last name etc.), picture.	Edit(string id, AuthorViewModel author, IFormFile file)	
AS2		Add author	Creates new authors adding general information (title, description etc.), picture.	Create(AuthorViewModel author, IFormFile)	
AS3		Delete author	Deletes all information available about author. This action cannot be undone.	Delete(string id)	
AS4	Guest	Search author	Searches all information available about author.	SearchAuthor(string key)	

### *Book Subsystem*

№	Role	Use Case	Description	Action Method	Tests
BS1	Admin	Edit book	Updates information about book such as general information (title, description etc.), picture and data about authors.	Update(string id, BookViewModel book, IFormFile file)	
BS2		Add book	Creates new books adding general information (title, description etc.), picture.	Create(BookViewModel book, IFormFilefile)	
BS3		Delete book	Deletes all information available about book. This action can not be undone.	Delete(string id)	
BS4		Edit genre	Updates genre name and genre color.	Edit(GenreViewModel model)	
BS5		Add genre	Creates a new genre and assign it to book.	Create(GenreViewModel model)	
		Delete genre	Deletes selected genre. This action can not be undone	Delete(string id)	
BS4	Guest	Search book	Searches all information available about book.	SearchBook(string key)	

### *Wishlist Subsystem*

№	Role	Use Case	Description	Action Method	Tests
WS1	Registered user	Add to wish list	Adds books to the wish list	AddToWishList(string id)	
WS2		Delete from wish list	Removes books from the wish list	RemoveFromWishList(string id)	
WS3		Add note	Adds a note why this book should be in the wish list	AddToWishListConfirmed(string id, string comment)	

### *User Subsystem*

№	Role	Use Case	Description	Action Method	Tests
US1	Registered user	Log in	Authorizes in system	Login(LoginViewModel model, string returnUrl)	
US2		Log out	Signs out from system	Logout()	
US3		Update profile	Change first and last name	EditConfirmed(UserProfileViewModel profile)	
US	Guest	Registration	Registration in system	Register(RegisterViewModel model, string returnUrl)	

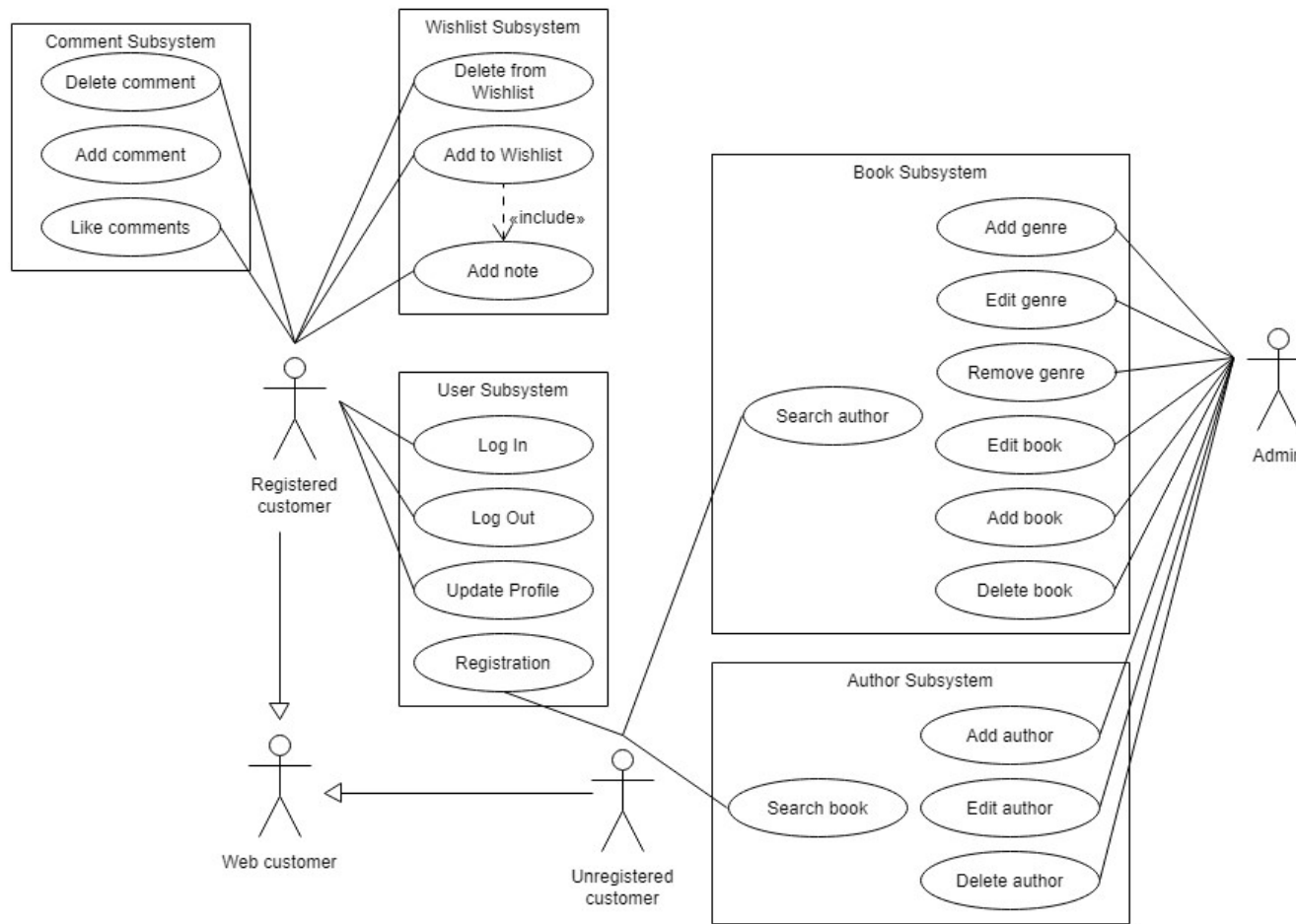
### *Comment Subsystem*

<b>№</b>	<b>Role</b>	<b>Use Case</b>	<b>Description</b>	<b>Action Method</b>	<b>Tests</b>
CS1	Admin	Add Comment	Leaves a comment on book that was read	AddCommentConfirmed(string id, string comment)	
CS2		Delete Comment	Deletes any comment due to reason(bullying, abusive behavior, politics, etc...)	Delete(string id)	
CS3	Registered user	Add Comment	Leaves a comment on book that was read	AddCommentConfirmed(string id, string comment)	
CS4		Delete Comment	Deletes only own comments	Delete(string id)	
CS5		See Comments	Sees all own comments that were left	GetByUser(ClaimsPrincipal user)	
CS6		Like Comment	Likes all comments that were left	Like(string id)	

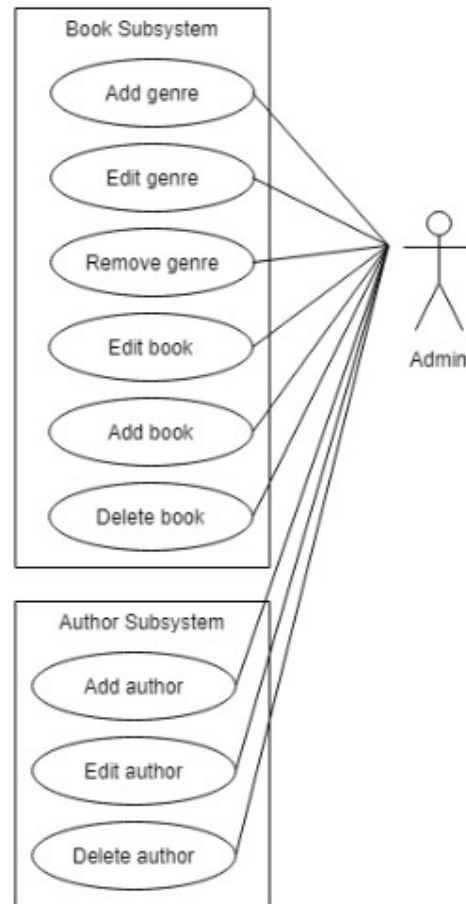
# Diagrams

## 1. Use cases

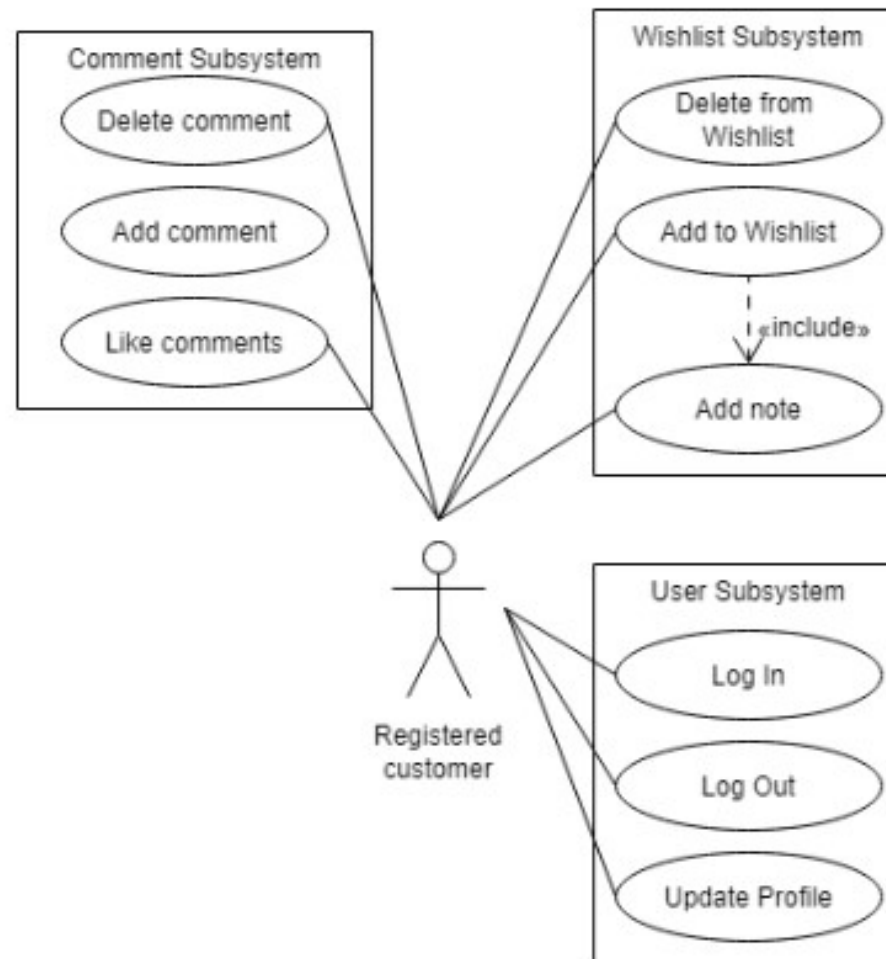
### 1.1 General use case diagram



## 1.2 Use case for Admin

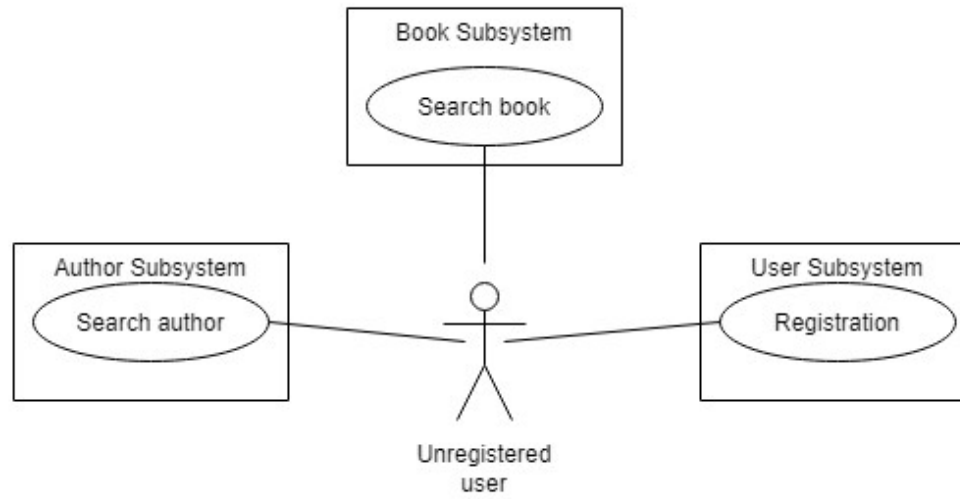


### 1.3 Use case for registered user





## 1.4 User case for guest



## 2. Sequence.

