**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.
3. **Guest** – person who has only availability to search and view information about books and authors.

**Subsystems**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **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) | TST6 |
| AS2 | Add author | Creates new authors adding general information (title, description etc.), picture. | Create(AuthorViewModel author, IFormFile) | TST1 |
| AS3 | Delete author | Deletes all information available about author. This action cannot be undone. | Delete(string id) | TST2 |
| AS4 | Guest | Search author | Searches all information available about author. | SearchAuthor(string key) | TST31 |

***Author Subsystem***

***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) | TST12 |
| BS2 | Add book | Creates new books adding general information (title, description etc.), picture. | Create(BookViewModel book, IFormFilefile) | TST7 |
| BS3 | Delete book | Deletes all information available about book. This action can not be undone. | Delete(string id) | TST8 |
| BS4 |  | Edit genre | Update genre. |  | TST38 |
| BS5 |  | Add genre | Creates a new genre and assigh it to book. |  | TST34 |
|  |  | Remove genre | Delete genre. |  | TST35 |
| BS4 | Guest | Search book | Searches all information available about book. | SearchBook(string key) | TST30 |

***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) | TST24 |
| WS2 | Delete from wish list | Removes books from the wish list | RemoveFromWishList(string id) | TST25 |
| WS3 | Add note | Adds a note why this book should be in the wish list | AddToWishListConfirmed(  string id, string comment) | TST24 |

***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***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Role** | **Use Case** | **Description** | **Action Method** | **Tests** |
| CS1 | Admin | Add Comment | Leave a comment on book that was read | AddCommentConfirmed(string id, string comment) | TST17 |
| CS2 | Delete Comment | Delete any comment due to reason(bullying, abusive behavior, politics, etc…) | Delete(string id) | TST18 |
| CS3 | Registered user | Add Comment | Leave a comment on book that was read | AddCommentConfirmed(string id, string comment) | TST17 |
| Delete Comment | Delete only own comments | Delete(string id) | TST18 |
| CS4 | See Comments | See all comments than were left | GetByUser(ClaimsPrincipal user) | TST20 |

***Tests***

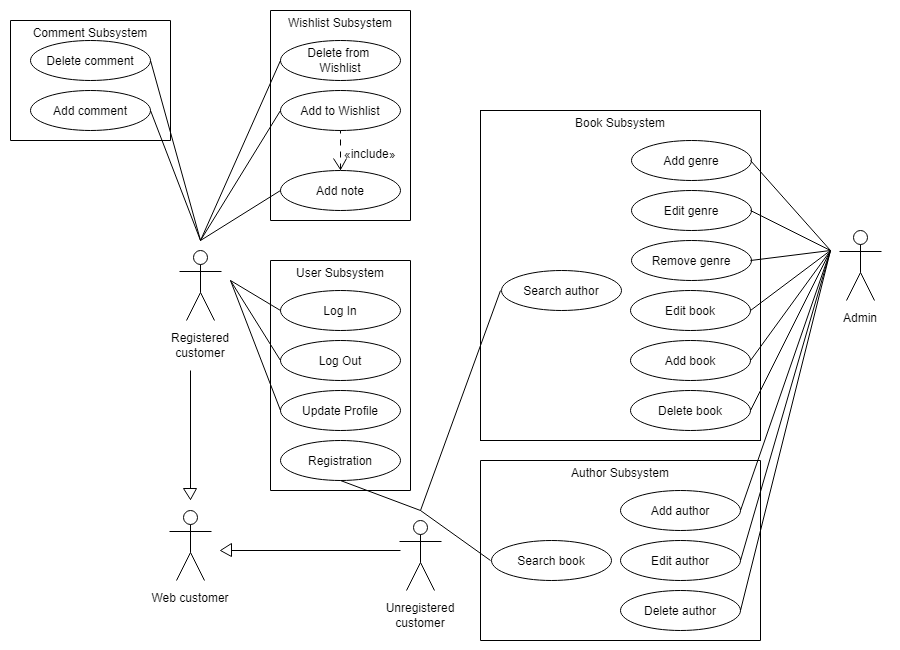
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Test class** | **Test method** | **Code coverage (%)** |
| TST1 | AuthorServiceTests | CreateTest | 100 |
| TST2 | DeleteTest |
| TST3 | GetByIdTest |
| TST4 | GetByIdNotExisting |
| TST5 | GetListTest |
| TST6 | UpdateTest |
| TST7 | BookServicetTests | CreateTest | 100 |
| TST8 | DeleteTest |
| TST9 | GetByIdTest |
| TST10 | GetByIdNotExisting |
| TST11 | GetListTest |
| TST12 | UpdateTest |
| TST13 | AssignExistingAuthorTest |
| TST14 | AssignAuthorTest |
| TST15 | RemoveAuthorsTest |
| TST16 | RemoveNoneAuthorsTest |

|  |  |  |  |
| --- | --- | --- | --- |
| TST17 | CommentServiceTests | AddCommentTest | 100 |
| TST18 | DeleteCommentTest |
| TST19 | GetByIdTest |
| TST20 | GetByUserTest |
| TST21 | UserServiceTests | DeleteTest | 100 |
| TST22 | GetUserProfileByIdTest |
| TST23 | GetUserProfileByIdNotExistingTest |
| TST24 | GetUserProfileTest |
| TST25 | UpdateTest |
| TST26 | WishListServiceTests | AddToWishListTest | 100 |
| TST27 | DeleteFromWishListTest |
| TST28 | GetByIdTest |
| TST29 | GetUserWishListTest |
| TST30 | SearchServiceTests | SearchBookTest | 100 |
| TST31 | SearchAuthorTest |
| TST32 | ImageToolTests | CropMaxSquareTest | 100 |
| TST33 | ResizeTest |

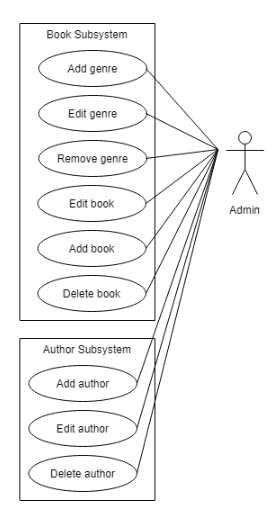
|  |  |  |  |
| --- | --- | --- | --- |
| TST34 | GenreServiceTests | CreateTest | 100 |
| TST35 | DeletetTest |
| TST36 | GetByIdTest |
| TST37 | GetByIdNotExistingTest |
| TST38 | UpdateTest |

**Diagrams**

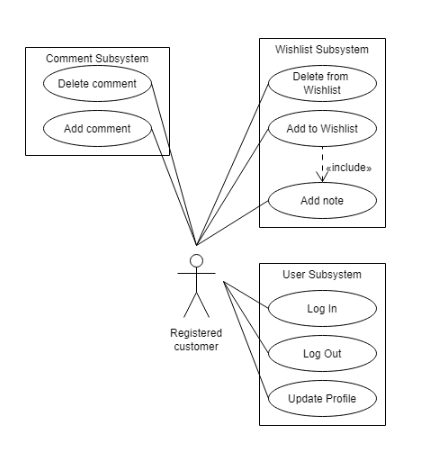
1. Use cases
   1. General use case diagram



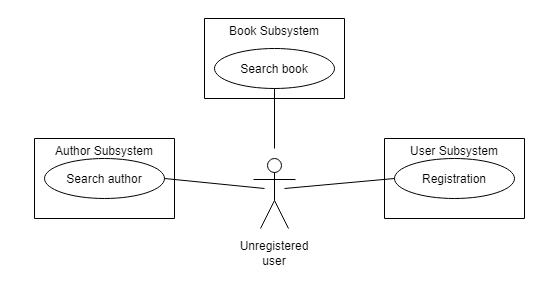
1.2 Use case for Admin



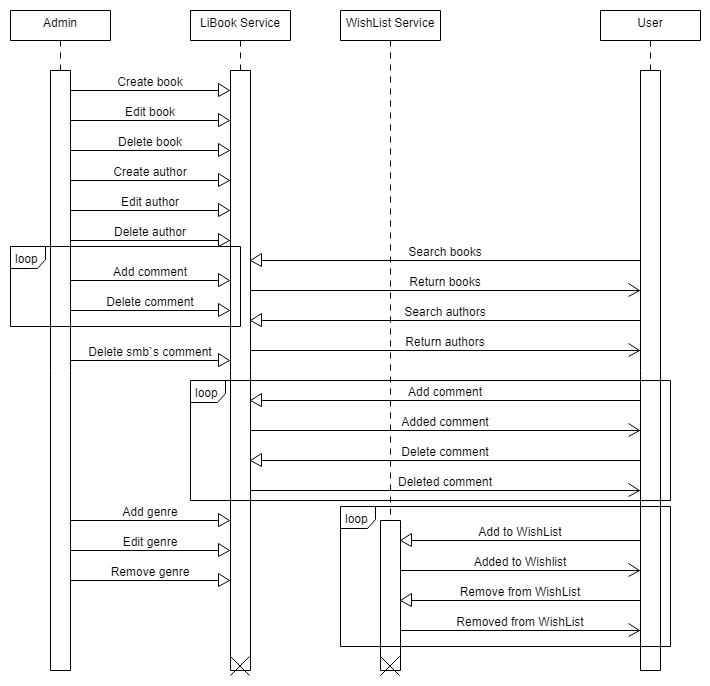
* 1. Use case for registered user



* 1. User case for guest



1. Sequence.



1. Class.

