**Requirements Documentation**

The following use cases represents a story that describes the interaction between the actor and the system.

**A**.  
**Use case:** View book-list   
**Primary actor:** User, Admin  
**Goal in context:** To show the available books’ list when user or admin wants to see it  
**Precondition:** Admin or any user has to be authorized  
**Trigger:** User or admin logs in to the website and clicks on the booklist button or enters the correct url.  
**Scenario:** 1. Enter user id and password  
 2. Click on ‘booklist’ menu to view the list  
**Exceptions:** 1. User id is incorrect  
 2. Password is incorrect  
 3. User is not registered yet  
 4. Entered url is wrong  
**Priority:** Essential, must be implemented  
**Channel to actor:** Via url or webpage  
**Secondary actor:** Server  
**Channel to secondary actor**: via API

**B**.  
**Use case:** Add book   
**Primary actor:** Admin  
**Goal in context:** To add a new book and details of that book  
**Precondition:** Admin has to be authorized  
**Trigger:** Admin logs in to the website and clicks on the booklist button and then enters the ‘add’ button or enters the correct url.  
**Scenario:** 1. Enter user id and password  
 2. Click on ‘booklist’ menu to view the list  
 3. Click ‘add’ button  
 4. Add the details and then save  
**Exceptions:** 1. Admin’s user id is incorrect  
 2. Password is incorrect  
 3. Admin is not registered yet  
 4. Entered url is wrong  
**Priority:** Essential, must be implemented  
**Channel to actor:** Via url or webpage

**C**.  
**Use case:** Edit book  
**Primary actor:** Admin  
**Goal in context:** To edit any detail of a book from the available books’ list when necessary  
**Precondition:** Admin has to be authorized  
**Trigger:** User or admin logs in to the website and clicks on the booklist button and then enters the edit button of a certain book or enters the correct url with the book-id  
**Scenario:** 1. Enter user id and password  
 2. Click on ‘booklist’ menu to view the list  
 3. Click ‘edit’ button of the book to be edited  
 4. Edit as necessary and then save  
**Exceptions:** 1. Admin’s user id is incorrect  
 2. Password is incorrect  
 3. Admin is not registered yet  
 4. Entered url is wrong  
**Priority:** Essential, must be implemented  
**Channel to actor:** Via url or webpage  
**Secondary actor:** Server  
**Channel to secondary actor**: via API

**D**.  
**Use case:** Delete book  
**Primary actor:** Admin  
**Goal in context:** To delete a book from the available books’ list when necessary  
**Precondition:** Admin has to be authorized  
**Trigger:** User or admin logs in to the website and clicks on the booklist button and then enters the ‘delete’ button of a certain book or enters the correct url with the book-id  
**Scenario:** 1. Enter user id and password  
 2. Click on ‘booklist’ menu to view the list  
 3. Click ‘edit’ button of the book to be edited  
 4. Edit as necessary and then save  
**Exceptions:** 1. Admin’s user id is incorrect  
 2. Password is incorrect  
 3. Admin is not registered yet  
 4. Entered url is wrong  
**Priority:** Essential, must be implemented  
**Channel to actor:** Via url or webpage

**E**.  
**Use case:** Search books by name  
**Primary actor:** Admin and User  
**Goal in context:** To search a book from the available books’ list  
**Precondition:** Admin or user has to be authorized  
**Trigger:** User or admin logs in to the website and clicks on the search button and then enters the name of a certain book or enters the correct url with the book-id  
**Scenario:** 1. Enter user id and password  
 2. Click on ‘search’ menu  
**Exceptions:** 1. Admin’s or user’s user id is incorrect  
 2. Password is incorrect  
 3. Not registered yet  
 4. Entered url is wrong  
 5. Searched book is not available  
**Priority:** Essential, must be implemented  
**Channel to actor:** Via url or webpage  
**Supporting issues:** Search functionality should support partial name matches and simple mistakes in spelling.

Fig: Use case diagram for library management system

Server

Admin

User