

**Use Cases**

**for**

**BookShelf**

**Version 1.0 approved**

**Prepared by**

Ryu Hyunsun

Zhu Yu Hao

**Nanyang Technological University**

**Created on 04/02/2024**

# Revision History

Name	Date	Reason For Changes	Version
Ryu Hyunsun, Zhu Yu Hao	07-02-2024	Initial Draft	1.0

## Use Case List

<i>Primary Actor</i>	<i>Use Cases</i>
User	1 - Login 2 - Registration 3 - Reset Password 4 - Logout 5 - Save Books 6 - Create Collection 7 - View Books on Home Page 8 - Search Books 9 - View Book Collections 10 - View Book Details 11 - Delete a Book from Collection
Database	1- Login 2 - Registration 3 - Reset Password 5 - Save Books 6 - Create Collection 9 - View Book Collections
ClerkJS API	1 - Login 2 - Registration 3 - Reset Password 4 - Logout
Google Books API	8 - Search Books
Server	7 - View Books on Home Page
New York Times API	7 - View Books on Home Page

# Use Case Descriptions

## 1. Login

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun
Date Created:	04/02/2024	Date Last Updated:	04/02/2024

Actors:	Database, User, ClerkJS API
Description:	The user must be able to login to use the app.
Trigger:	The user clicks on the Login button on the website.
Preconditions:	<ol style="list-style-type: none"><li>1. The internet connection is stable.</li><li>2. The website is launched.</li></ol>
Postconditions:	<ol style="list-style-type: none"><li>1. User has logged in to the website.</li><li>2. Authenticated home page is loaded.</li></ol>
Normal Flow:	<ol style="list-style-type: none"><li>1. The user clicks on the Login button on the home page.</li><li>2. Login screen will be launched.</li><li>3. The user enters their username in the 'Username' field.</li><li>4. The user enters their password in the 'Password' field.</li><li>5. The user clicks on the 'Login' button.</li><li>6. ClerkJS API will then verify the username and password with the database.</li><li>7. Upon successful login, the authenticated home page will be loaded.</li></ol>
Alternative Flows:	<p>1.4: If the user inputs an incorrect password.</p> <ol style="list-style-type: none"><li>1. If the user inputs an incorrect password, an error message "Username / Password entered is invalid, Please try again." will be shown.</li><li>2. The password field will be cleared.</li><li>3. The use case returns to step 4.</li></ol> <p>1.3: If the user inputs an incorrect username.</p> <ol style="list-style-type: none"><li>1. If the user inputs an incorrect username, an error message "Username / Password entered is invalid, Please try again." will be shown.</li><li>2. The password field will be cleared.</li><li>3. The use case returns to step 3.</li></ol> <p>1.2: If the user clicks on the "Reset Password" button</p> <ol style="list-style-type: none"><li>1. The user clicks on the "Reset Password" button.</li><li>2. The use case uses the extended Use Case 3 'Reset Password'.</li></ol> <p>1.2: If the user clicks on the "Register" button</p> <ol style="list-style-type: none"><li>1. The user clicks on the "Register" button.</li><li>2. The use case uses the extended Use Case 2 'Register'.</li></ol>
Exceptions:	<p>1.0.E.5: If the password has been found in an online data breach</p> <ol style="list-style-type: none"><li>1. If the password has been found in an online breach, an error message "Password has been found in an online data breach. Please reset your password" will be displayed.</li><li>2. The use case uses the extended Use Case 3 'Reset Password'.</li></ol>
Includes:	N/A
Priority:	High
Frequency of Use:	Once in three months - When the user initially launches the website or when the login session is terminated.
Business Rules:	
Special Requirements:	SR-1: The system should verify the username and password within 3 seconds.
Assumptions:	-
Notes and Issues:	-

## 2. Register

Use Case ID:	2		
Use Case Name:	Register		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun
Date Created:	04/02/2024	Date Last Updated:	04/02/2024

Actors:	Database, User, ClerkJS API
Description:	The user must be able create a new account.
Trigger:	The user clicks on the “Register” button.
Preconditions:	<ol style="list-style-type: none"> <li>Internet connection is stable.</li> <li>The user’s account must not already exist in the database.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>The user successfully registers an account.</li> <li>The user will be directed to the Login page for login upon successful account creation.</li> <li>3 default book collections (Read, Reading, To be completed) must be created.</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>The user clicks on the ‘Register’ button on the home page./The user clicks on the ‘Register’ button on the Login page.</li> <li>Registration screen will be loaded.</li> <li>The user must input a valid email and password.</li> <li>The user must confirm the password.</li> <li>The user clicks on the ‘Register’ button.</li> <li>The system shall verify that the password and the confirm password match.</li> <li>The system shall check with the database to ensure that the registrant’s username does not exist in the database.</li> <li>The system must ensure that the password fulfils the criteria.</li> <li>ClerkJS API will verify the user’s email address by sending a one-time code with a redirect link to the email address provided.</li> <li>The user will click on the link to enter the verification code.</li> <li>The user clicks on the “Verify Email” button.</li> <li>A dialog box with a message “Account created successfully” will be displayed upon successful account creation.</li> <li>The user will be directed to the Login page.</li> <li>The system returns to the Use Case 1 “Login”.</li> </ol>
Alternative Flows:	-
Exceptions:	<p>2.0.E.5: If the user attempts to use a password that has been found in an online data breach</p> <ol style="list-style-type: none"> <li>An error message “Password has been found in an online data breach. Please use another password” will be displayed.</li> <li>The use case returns to step 3.</li> </ol> <p>2.0.E.5: Email already exists in the database</p> <ol style="list-style-type: none"> <li>System queries and finds a record with the same email address.</li> <li>An error message “The email address is already in use. Please enter a new email address.” will be displayed.</li> <li>The use case returns to step 3</li> </ol> <p>2.0.E.8: Password does not fulfil the criteria</p> <ol style="list-style-type: none"> <li>An error message “The password should contain at least one upper case letter, one lower case letter, one digit, and one special character.” will be displayed.</li> <li>The use case returns to step 3.</li> </ol> <p>2.0.E.8: Password exceeds 12 characters</p> <ol style="list-style-type: none"> <li>An error message “The length of the password must not exceed 12 characters.” will be displayed.</li> <li>The use case returns to step 3.</li> </ol>

	2.0.E.6: The password and confirm password does not match <ol style="list-style-type: none"> <li>1. An error message “The password and confirm password does not match.” will be displayed.</li> <li>2. The use case returns to step 3.</li> </ol>
Includes:	-
Priority:	High
Frequency of Use:	Once per lifetime
Business Rules:	Each email address should be associated with only one user.
Special Requirements:	Registering an account should take less than 1 minute.
Assumptions:	-
Notes and Issues:	-

### 3. Reset Password

Use Case ID:	3		
Use Case Name:	Reset Password		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun
Date Created:	04/02/2024	Date Last Updated:	04/02/2024

Actors:	Database, User, ClerkJS API
Description:	The user must be able to reset their password.
Trigger:	The user clicks on the “Reset Password” button.
Preconditions:	<ol style="list-style-type: none"> <li>1. The internet connection is stable.</li> <li>2. The user has an existing account.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. “Password reset successful” message will be shown and the user will be redirected to the home page.</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Reset Password” button on the Login page./The user clicks on the “Reset Password” button after clicking on the profile picture on the home page.</li> <li>2. “Reset Password” page will be loaded.</li> <li>3. The user will enter their email address.</li> <li>4. The user will click on the “Get Verification Code” button.</li> <li>5. ClerkJS API sends a password reset code to the provided email.</li> <li>6. The user enters the new password.</li> <li>7. The user enters the password reset code.</li> <li>8. The user clicks on the “Reset Password” button.</li> <li>9. The system will check to ensure that the new password meets the criteria.</li> <li>10. ClerkJS API will verify the new password and password reset code.</li> <li>11. Upon successful reset, the user will be redirected to the home page.</li> </ol>
Alternative Flows:	3.8. If the user clicks on the “Cancel” button <ol style="list-style-type: none"> <li>1. The user interface will display a dialog box to confirm the user’s intention to cancel the operation.</li> <li>2. The user will be directed to the home page.</li> </ol>
Exceptions:	3.0.E.10. If the user attempts to use a password that has been found in an online data breach <ol style="list-style-type: none"> <li>1. An error message “Password has been found in an online data breach. Please use another password” will be displayed.</li> <li>2. The use case returns to step 4.</li> </ol> 3.0.E.9.. The new password does not meet the criteria. <ol style="list-style-type: none"> <li>1. An error message “The password should contain at least one upper case letter, one lower case letter, one digit, and one special character.” will be displayed.</li> </ol>

	2. The use case returns to step 6. 3.0.E.10. The password reset code entered is invalid. 1. An error message “The password reset code is incorrect. Please try again.” will be displayed. 2. The use case returns to step 4. 3.0.E.5. The email does not exist. 1. An error message “The email address does not exist.” will be displayed. 2. The use case returns to step 3.
Includes:	-
Priority:	Low
Frequency of Use:	Once per year
Business Rules:	-
Special Requirements:	The email containing the password reset code must be sent within 10 seconds.
Assumptions:	-
Notes and Issues:	-

#### 4. Logout

Use Case ID:	4		
Use Case Name:	Logout		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun
Date Created:	04/02/2024	Date Last Updated:	04/02/2024

Actors:	Database, User, ClerkJS API
Description:	The user must be able to logout from the system.
Trigger:	The user clicks on the “Logout” button on the home page.
Preconditions:	1. The user is already logged in.
Postconditions:	1. The user will be logged out. 2. The user will be directed to the unauthenticated home page.
Normal Flow:	1. The user clicks on the “Logout” button on the home page. 2. A message “Are you sure you want to log out?” will be displayed. 3. User clicks on the “Log me out” button. 4. ClerkJS API will deactivate the current session. 5. Upon successful log out, the user will be directed to the unauthenticated home page.
Alternative Flows:	4.3. The user clicks on the “Cancel” button. 1. The user will be directed to the home page.
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Once per year
Business Rules:	-
Special Requirements:	The user must be logged out within 5 seconds.
Assumptions:	-
Notes and Issues:	-

#### 5. Save Books

Use Case ID:	5		
Use Case Name:	Save Books		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun

Date Created:	04/02/2024	Date Last Updated:	04/02/2024
---------------	------------	--------------------	------------

Actors:	Database, User
Description:	The user must be able to save a book into a collection.
Trigger:	The user clicks on the save button on the View Book Details page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must have an account.</li> <li>2. The user must be already logged in.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The book will be saved into the selected collection.</li> <li>2. The user will be notified in the user interface through a pop-up message upon successful addition.</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on the save button on the View Book Details page.</li> <li>2. The user will have the option to create a new book collection, add to an existing collection, or cancel on a dialog box.</li> <li>3. The user clicks on the "Add to Collection" button.</li> <li>4. A list of book collections created by the user will be displayed.</li> <li>5. The user clicks on one of the book collections.</li> <li>6. When a book is successfully added into a collection, the user will be notified in the user interface through a pop-up message.</li> </ol>
Alternative Flows:	<p>5.2. If the user clicks on the "Create New Collection" button</p> <ol style="list-style-type: none"> <li>1. The use case will use the extended Use Case 6 "Create Collection."</li> <li>2. The use case will return to step 6.</li> </ol> <p>5.2 If a user clicks on the "Cancel" button</p> <ol style="list-style-type: none"> <li>1. The dialog box will be closed.</li> </ol>
Exceptions:	<p>5.0.E.5 If the book already exists in the selected collection</p> <ol style="list-style-type: none"> <li>1. The user will be notified through a pop-up message stating: "Could not be added! Already exists!"</li> <li>2. The operation will be cancelled.</li> </ol> <p>5.0.E.5 If the number of books in the collection already exceeds 500</p> <ol style="list-style-type: none"> <li>1. The user will be notified through a pop-up message stating: "Maximum of 500 books reached!"</li> <li>2. The operation will be cancelled.</li> </ol>
Includes:	-
Priority:	Medium
Frequency of Use:	Once per week
Business Rules:	<ul style="list-style-type: none"> <li>• The user must be able to add a book into one or more collections.</li> <li>• A book cannot be added into the same collection more than once.</li> <li>• The number of books in a collection must not exceed 500.</li> </ul>
Special Requirements:	The user must be able to save a book into a collection within 15 seconds.
Assumptions:	-
Notes and Issues:	-

## 6. Create Collection

Use Case ID:	6		
Use Case Name:	Create Book Collection		
Created By:	Ryu Hyunsun	Last Updated By:	Ryu Hyunsun
Date Created:	04/02/2024	Date Last Updated:	04/02/2024

Actors:	Database, User
Description:	The user must be able to create a new collection.



Trigger:	The user clicks on the “Create New Collection” button after clicking on the “Save Books” button on the View Book Details page / The user clicks on the “Create New Collection” on the Book Collections page.
Preconditions:	1. The user must be already logged in.
Postconditions:	1. The user will be notified in the user interface through a pop-up message confirming the successful creation of a collection.
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user interface will display a dialog box with a textbox for the user to input a name for the new collection, along with “Add” button and “Cancel” buttons.</li> <li>2. The user will enter the name.</li> <li>3. The user will click on the “Add” button.</li> <li>4. The user will be notified in the user interface through a pop-up message confirming the successful creation of a collection.</li> </ol>
Alternative Flows:	6.3.. When a user clicks on the cancel button <ol style="list-style-type: none"> <li>1. The dialog box will be closed.</li> </ol>
Exceptions:	6.0.E.3. When a collection with the same name already exists <ol style="list-style-type: none"> <li>1. The user interface should present an error message via a pop-up.</li> <li>2. The user will be prompted to enter a different name for the new collection.</li> </ol> 6.0.E.3. When the number of collections created by the user exceeds 500 <ol style="list-style-type: none"> <li>1. The user interface should present an error message via a pop-up.</li> <li>2. The operation will be cancelled.</li> </ol>
Includes:	-
Priority:	Medium
Frequency of Use:	Once per month
Business Rules:	The user will be able to create a maximum of 500 collections.
Special Requirements:	The user must be able to create a collection within 15 seconds.
Assumptions:	-
Notes and Issues:	-

## 7. View Books on Home Page

Use Case ID:	7		
Use Case Name:	View Books on Home page		
Created By:	Zhu Yu Hao	Last Updated By:	Zhu Yu Hao
Date Created:	07/02/2024	Date Last Updated:	07/02/2024

Actors:	New York Times API, User, Server
Description:	The user must be able to view the list of books including top bestsellers on the home page.
Trigger:	<ol style="list-style-type: none"> <li>1. The user clicks on the home page</li> <li>2. The user first loads the website</li> </ol>
Preconditions:	1. The user’s computer must be connected to the internet.
Postconditions:	The Home page will be loaded.
Normal Flow:	<ol style="list-style-type: none"> <li>1. The system attempts to retrieve the top 5 bestsellers from the bestsellers cache.</li> <li>2. The system will display the top 5 bestsellers under the “bestsellers” section.</li> <li>3. The system attempts to retrieve the top 5 recommendations from the recommendations cache.</li> <li>4. The system will display the top 5 recommendations under the “Recommendations” section.</li> </ol>

Alternative Flows:	-
Exceptions:	<p>7.0.E.1 The system fails to retrieve from the bestsellers cache</p> <ol style="list-style-type: none"> <li>1. The system fetches the top 5 bestsellers from the New York Times API.</li> <li>2. The top 5 best sellers are cached into the bestsellers cache.</li> <li>3. Return to step 2.</li> </ol> <p>7.0.E.2 The system fails to retrieve from the recommendations cache</p> <ol style="list-style-type: none"> <li>1. The system fetches the top 5 recommendations from the server.</li> <li>2. The top 5 recommendations are cached into the recommendations cache.</li> <li>3. Return to step 4.</li> </ol>
Includes:	-
Priority:	High
Frequency of Use:	Once per day
Business Rules:	-
Special Requirements:	The books must be displayed within 5 seconds.
Assumptions:	-
Notes and Issues:	-

## 8. Search Books

Use Case ID:	8		
Use Case Name:	Search Books		
Created By:	Zhu Yu Hao	Last Updated By:	Zhu Yu Hao
Date Created:	07/02/2024	Date Last Updated:	07/02/2024

Actors:	User, Google Books API
Description:	The user must be able to search for books using a search bar.
Trigger:	The user clicks on the search bar.
Preconditions:	The user must be on the home page.
Postconditions:	The Search Books page will be loaded based on the search query.
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user types the title/genre/author/ISBN of the book in the search bar.</li> <li>2. The system will send requests to the cache database to retrieve recent books that match the search.</li> <li>3. The books are displayed in a pagination view.</li> <li>4. The search bar will show a greyed out text beside the actual query for autocompletion.</li> <li>5. The user can press Tab to autocomplete the query to the exact book.</li> <li>6. The user can select a book to view book details using extended Use Case 10 "View Book Details".</li> </ol>
Alternative Flows:	-
Exceptions:	<p>8.0.E.2 The system fails to retrieve the books from the cache.</p> <ol style="list-style-type: none"> <li>1. The system will send a request to Google Books API to retrieve the list of books</li> <li>2. Return to step 3.</li> </ol> <p>8.0.E.3 Google Books API replies with an error response.</p> <ol style="list-style-type: none"> <li>1. The system will display the error message "Unable to retrieve books" in the centre of the page.</li> </ol>
Includes:	-
Priority:	High

Frequency of Use:	Once per day
Business Rules:	<ol style="list-style-type: none"> <li>1. The search text must be of at least 1 character and less than 128 characters.</li> <li>2. There must be less than 5 books in each row displayed in the pagination.</li> <li>3. Each cell in the table must contain the title, author and cover image of the book in the pagination.</li> <li>4. If the number of books on a page exceeds 30, the system must provide pagination for the user to navigate to a subsequent page.</li> </ol>
Assumptions:	-
Notes and Issues:	-

## 9. View Collections

Use Case ID:	9		
Use Case Name:	View Collections		
Created By:	Zhu Yu Hao	Last Updated By:	Zhu Yu Hao
Date Created:	07/02/2024	Date Last Updated:	07/02/2024

Actors:	User, Database
Description:	The user can display the list of collections created by the user.
Trigger:	The user selects the View Collections navigation tab.
Preconditions:	<ol style="list-style-type: none"> <li>1. User is logged in</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The Collections Page will be loaded with a list of collections created by the user.</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the View Collections navigation tab.</li> <li>2. The system queries the database for the collections owned by this user.</li> <li>3. The system displays the collections owned by this user.</li> <li>4. The user can select one of the collections.</li> </ol>
Alternative Flows:	-
Exceptions:	9.0.E.1 The database does not contain any collections owned by this user <ol style="list-style-type: none"> <li>1. The system displays “no collections owned by this user” in the centre of the page.</li> </ol>
Includes:	-
Priority:	Medium
Frequency of Use:	High
Business Rules:	-
Special Requirements:	The collections created by the user including the default collections should be loaded within 10 seconds.
Assumptions:	-
Notes and Issues:	-

## 10. View Book Details

Use Case ID:	10		
Use Case Name:	View Book Details		
Created By:	Zhu Yu Hao	Last Updated By:	Zhu Yu Hao
Date Created:	07/02/2024	Date Last Updated:	07/02/2024

Actors:	User
Description:	The user can see a list of details of the book.

Trigger:	The user selects a book in the list of books displayed in the pagination view.
Preconditions:	The user selects one of the categories in the home page, or selects one of the books from the search query.
Postconditions:	
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user clicks on a link of a book cell in the pagination.</li> <li>2. The system will retrieve the specific book information stored temporarily during the search query.</li> <li>3. The system will display the book description, book pre-requisites and book preview link.</li> <li>4. The user can click on the book preview link to read the preview.</li> </ol>
Alternative Flows:	
Exceptions:	
Includes:	
Priority:	High
Frequency of Use:	High
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

## 11. Delete a Book from Collection

Use Case ID:	11		
Use Case Name:	Delete a book from Collection		
Created By:	Zhu Yu Hao	Last Updated By:	Zhu Yu Hao
Date Created:	07/02/2024	Date Last Updated:	07/02/2024

Actors:	User, Database
Description:	The user can delete a book from the selected collection.
Trigger:	The user clicks on the “Delete” button on the Collections page.
Preconditions:	The user is in the specified collection page.
Postconditions:	The selected book will be deleted from the collection .
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the delete icon beside the book cell in the collection.</li> <li>2. The system sends a confirmation pop-up message on deletion of the book.</li> <li>3. The system requests a delete operation to the database for the user collection for that book.</li> <li>4. The system displays a “successfully deleted book” pop-up message to the user.</li> </ol>
Alternative Flows:	11.2 Cancel confirmation of pop-up <ol style="list-style-type: none"> <li>1. Returns to the specified collection page.</li> </ol> 11.3 Delete operation failed <ol style="list-style-type: none"> <li>1. System displays a “Failed to delete book” message to the user.</li> </ol>
Exceptions:	
Includes:	
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

