

The Online Bookstore System

List of possible use cases and use case description examples.

1. Use Cases for the Online Bookstore System

ID#	Primary Actor	Use Cases
1	Web user	Create Account
2	Registered User	Login
3	Logged-in User	Edit Profile
5	Web user, logged-in user	Search and Filter
6	Web user, logged-in user	View Book details
7	Web user, logged-in user	Add Book to cart
9	Logged-in user	View order history
10	Logged-in user	Checkout
11	Logged-in user	Apply Promotion
12	Logged-in User	Reset password
13		Return Book(s)
14	Admin	Manage Books
15		Add New Book
16		Delete Book
17		Manage Promotions
18		Add New Promotion
19		Edit Existing Promotion
20		Delete Existing Promotion
21		Manage Users
22		Edit Prices
23		Edit Online Fees
24		Display Order Total
25		Enter Payment Info
26		Calculate order total payment
27	System	Send order confirmation email
28	System	Send account verification email

The above list is a suggested list. Complete the actors list in your document. You may add other small use cases to **decompose** a major user case. For Example, decompose **manage books** to invoke the use cases add new book, edit book info, and delete book.

The use case diagram should be consistent with the use description:

- If you invoke a use case in the main flow, with no condition, this is an **includes** relationship between the base case and the invoked use case (e.g. decomposition or a shared use case).
- If you invoke a use case under a condition, this is an **extends** relationship, the base use case is extended by the invoked use case. For example, Manage Books use case has three options; the admin can choose

to add, edit or delete a book (e.g. If the admin selects to edit existing book the system invokes Edit Book use case).

- If you invoke a use case in the exceptions row, this is an **extends** relationship. The bases use case is extended by the invoked use case. (e.g. In the exception of checkout: User password is not logged in, system invokes login use case.

Example 1: Use case description of edit profile using decomposition

Use Case ID:	3		
Use Case Name:	Edit Profile		
Created By:	Eman	Last Updated By:	Eman
Date Created:		Date Last Updated:	6/22/2020

Brief description	Users can edit their profile to change their personal information, credit card information, or address information. Users can also change their subscription to receive promotions.	
Related Use cases	This use case uses: Edit personal info, change password, and edit credit card use cases.	
Actors	Registered user	
Precondition	User is logged into the system	
PostCondition	User profile changes are saved into the database.	
Normal Flow	User	System
	<ol style="list-style-type: none"> 1. User selects edit profile 2. User selects to change password 3. User selects to change address 4. User selects to change address 5. User selects to subscribe/ unsubscribe to receive promotional email 	<ol style="list-style-type: none"> 1.1 System invokes Edit Personal Info use case. 2.1 System invokes change password use case. 3.1 System invokes Edit Address use case 4.1 System invokes Edit Credit Card use case. 5. System updates subscription information.
Exceptions		
Assumptions	validity checks and exceptions are handled by the invoked use cases.	
Alternative flows	Users can skip any of the steps 2-5.	

Example 2: An alternative user case description No decomposition

Use Case ID:	3		
Use Case Name:	Edit Profile		
Created By:	Eman	Last Updated By:	Eman
Date Created:		Date Last Updated:	1/29/2020

Brief description	Users can edit their profile to change their personal information, credit card information, or address information. Users can also change their subscription to receive promotions.	
Related Use cases	This use case uses: Edit personal info, change password, and edit credit card use cases.	
Actors	Registered user	
Precondition	User is logged into the system	
PostCondition	User profile changes are saved into the database.	
Normal Flow	User	System
	<ol style="list-style-type: none"> 1. User selects edit profile 2. User inserts and submits personal information. 3. User inserts and submits address changes. 4. User inserts and submits credit card changes. 5. users select to subscribe/unsubscribe to receive promotional emails. 6. If users indicate the desire to change the password <ol style="list-style-type: none"> 6.2 users enter current password, new password and password confirmation. 	<ol style="list-style-type: none"> 1.1 The system displays existing Personal information: First name, last name, email address, and the password providing the user with a facility to change the displayed information, except the email address. 2.1 The system checks the validity of personal information 2.2 System saves personal information in the database. 2.3 System displays existing address providing the user with a facility to change the address, add new address or delete the existing address 3.1 System validates address information and store it in the database. 3.2 System displays existing credit card info, providing the user with a facility to change or delete the credit card information. 4.1 System validates credit card information and stores it in the database. 5.1 System stores subscription changes. 6.1 System displays change password form 6.3 System validates the password fields and stores the new password. 6.4 System sends a password change alert email to the user.
Exceptions	<ol style="list-style-type: none"> 2.1 Personal information are invalid or incomplete 3.1 Address information are invalid or incomplete 	System prompts the user to change the invalid fields.

	<p>4.1 Card is invalid</p> <p>6.3 current password is not correct or the confirmation field value is not the same as the new password field value.</p>	
Special requirements	<ul style="list-style-type: none"> - Password value should be hidden. - Show only the last 4 digits of the credit card number. <p>Users can exit at any time choosing to save/ not save changes.</p>	<p>Security requirement: (1) Passwords and credit card number should be encrypted in the system database.</p> <p>(2) CVV is not stored in the database.</p>
Alternative flows	<ul style="list-style-type: none"> - Users can skip any of the steps 2-6. 	