

View recently added books

Use Case	View recently added books
Primary Actor	Guest, User
Goal in Context	View books that were recently added to the website
Preconditions	N/A
Triggers	Actor loads the homepage
Scenario	1. Actor loads the home page 2. Page sends AJAX request to server 3. Server responds with book
Exceptions	Server is down. Website has no book. What is the definition of "Recent"
Priority	High
When Available	First build
Frequency of Use	High
Channel to Actor	Web browser
Secondary Actors	Moderator, Admin
Channels to Secondary Actors	Web browser
Open Issues	Must store add date along with book and query efficiently

Logout

Use case	Logout

Primary Actor	User, Moderator, Admin
Goal in context:	Remove all access given to user and give them guest status
Preconditions:	Must be already logged in
Trigger:	The actor clicks on the logout button
Scenario:	1. The A clicks on logout on any page
Exceptions:	1. Actor session was timed out/invalid
Priority:	Essential, must be implemented
When available	First increment
Frequency of use	Multiple times a day
Channel to actor	Via web interface
Secondary actors	None
Channels to secondary actors	N/A
Open issues	When the actor is logged out, he should be sent back to his last accessed page. But what if that page is not accessible without having access?

Update Profile

Use Case	Update Profile
Primary Actor	User

Goal in Context	User can edit information stored on their profile
Preconditions	Actor must be logged in and on their profile
Triggers	Actor clicks the edit button
Scenario	<ol style="list-style-type: none"> 1. Actor goes to his profile 2. Actor clicks on the edit button 3. Actor fills in new information in presented modal 4. Actor clicks submit
Exceptions	<ol style="list-style-type: none"> 1. Actor session was timed out/invalid
Priority	High
When Available	First build
Frequency of Use	Moderate
Channel to Actor	Web interface
Secondary Actors	Moderator, Admin
Channels to Secondary Actors	Web interface
Open Issues	How to verify entered information is accurate.

Flag a review

Use Case	Flag a review
Primary Actor	User
Goal in Context	Flag a review that seems inappropriate or irrelevant
Preconditions	Actor must be logged in and on the book profile page
Triggers	Actor clicks on the flag button for the review
Scenario	

	<ol style="list-style-type: none"> 1. Actor navigates to a book profile and starts reading reviews 2. Actor notices a review he doesn't like. 3. Actor clicks on flag button
Exceptions	Review was deleted before actor can flag it
Priority	Moderate
When Available	2nd Build
Frequency of Use	Moderate
Channel to Actor	Web interface
Secondary Actors	N/A
Channels to Secondary Actors	N/A
Open Issues	What happens when a review gets too many flags

Checkout

Use Case	Checkout
Primary Actor	User
Goal in Context	Checkout a book that was in a users shopping cart
Preconditions	The actor must be logged in. Actor must have items in the shopping cart
Triggers	Actor clicks on checkout button
Scenario	<ol style="list-style-type: none"> 1. Actor adds a book to their shopping cart 2. Actor clicks on cart icon to open cart 3. Actor clicks checkout
Exceptions	Selected book currently has no stock. In this case, attempt to reserve

Priority	Essential, must be implemented
When Available	First build
Frequency of Use	Very frequent.
Channel to Actor	Web Interface
Secondary Actors	Moderator, Admin
Channels to Secondary Actors	Web interface
Open Issues	How to handle already checked out book. Should it auto reserve or ask for confirmation

Mark book as not interested

Use Case	Mark book as Not Interested
Primary Actor	User, Moderator, Admin
Goal in Context	User can mark a book as not recommended so they got similar suggestions
Preconditions	The actor must be logged in and be a currently active user. Actor must navigate to the book profile or the suggestions list.
Triggers	The actor clicks the not interested button.
Scenario	<ol style="list-style-type: none"> 1. Actor goes to the book profile. 2. Actor clicks on Not interested button. 3. Button color changes on success <ol style="list-style-type: none"> 1. Actor sees recommendation on home. 2. Actor clicks on Not interested button. 3. Button color changes on success
Exceptions	
Priority	Important, but not necessarily implemented immediately

When Available	Second increment.
Frequency of Use	Occasionally
Channel to Actor	Via web browser
Secondary Actors	
Channels to Secondary Actors	
Open Issues	User clicks the button again while the request is being processed.

View Book recommendations

Use Case	View recently added books
Primary Actor	User
Goal in Context	View books that are recommended to the actor based on their history
Preconditions	N/A
Triggers	Actor loads the homepage
Scenario	<ol style="list-style-type: none"> 1. Actor loads the home page 2. Page sends AJAX request to server 3. Server responds with books
Exceptions	Server is down. Website has no books. User has not activity
Priority	High
When Available	Second build
Frequency of Use	High

Channel to Actor	Web browser
Secondary Actors	Moderator, Admin
Channels to Secondary Actors	Web browser
Open Issues	What algorithm do we use to recommend books

Update Database Structure

Use Case	Update Database structure
Primary Actor	Admin
Goal in Context	The admin can update the structure of the website database to add features
Preconditions	Must be an admin and have SSH access to the database
Triggers	N/A
Scenario	1. Admin connects to the database (Via command line or gui) 2. Admin makes changes to the table DDL
Exceptions	Changes made breaks code in the website
Priority	High
When Available	First build
Frequency of Use	Rare
Channel to Actor	Command line or SQL GUI
Secondary Actors	N/A

Channels to Secondary Actors	N/A
Open Issues	A lot of things can break on such an action so we need a backup database in case something breaks and switch to it

Add/Remove rows from database

Use Case	Update Database structure
Primary Actor	Admin
Goal in Context	The admin can add or remove rows directly into the website database
Preconditions	Must be an admin and have SSH access to the database
Triggers	N/A
Scenario	<ol style="list-style-type: none"> 1. Admin connects to the database (Via command line or gui) 2. Admin writes a CRUD statement or uses the gui to make changes
Exceptions	Changes made breaks code in the website and not follow foreign key and code enforced constraints
Priority	High
When Available	First build
Frequency of Use	Rare
Channel to Actor	Command line or SQL GUI
Secondary Actors	N/A
Channels to Secondary Actors	N/A

Open Issues	A lot of things can break on such an action so we need a backup database in case something breaks and switch to it
-------------	--

Update look of the website

Use Case	Update Database structure
Primary Actor	Admin
Goal in Context	The admin can make changes to the template code and the CSS for the website to update its look and feel
Preconditions	Must be an admin and have SSH access to the server and the code
Triggers	N/A
Scenario	1. Admin makes changes to the code and pushes it
Exceptions	Changes made breaks code in the website
Priority	High
When Available	First build
Frequency of Use	Rare
Channel to Actor	Command line or SQL GUI
Secondary Actors	N/A
Channels to Secondary Actors	N/A
Open Issues	A lot of things can break on such an action so we need a some form a testing system and a backup. If the broken change is pushed we just switch to the old working copy

Get Book Statistics

Use Case	Update Database structure
Primary Actor	Moderator, Admin
Goal in Context	The actor can view book Statistics on his dashboard
Preconditions	The actor must be an admin or a moderator
Triggers	N/A
Scenario	1. The actor navigates to his dashboard. 2. Actor goes to the books tab 3. Actor selects a book to view statistics
Exceptions	The book must exist in the database
Priority	High
When Available	Second build
Frequency of Use	Moderate
Channel to Actor	Web interface
Secondary Actors	N/A
Channels to Secondary Actors	N/A
Open Issues	The book might not have any past use and no stats