COMP3111H Activity 1 (Group 14)

Task Allocation

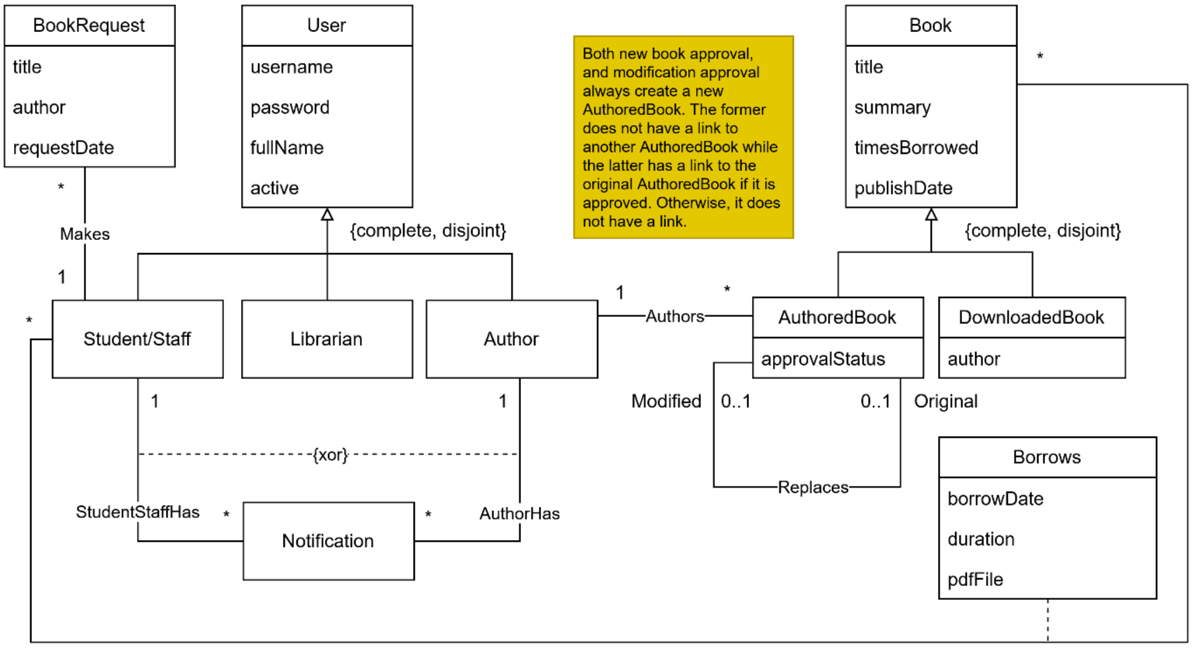
Name	SID	Task #
		1
		2
		3

Repository Link: https://github.com/cx0014/comp3111h-project

Table of Contents

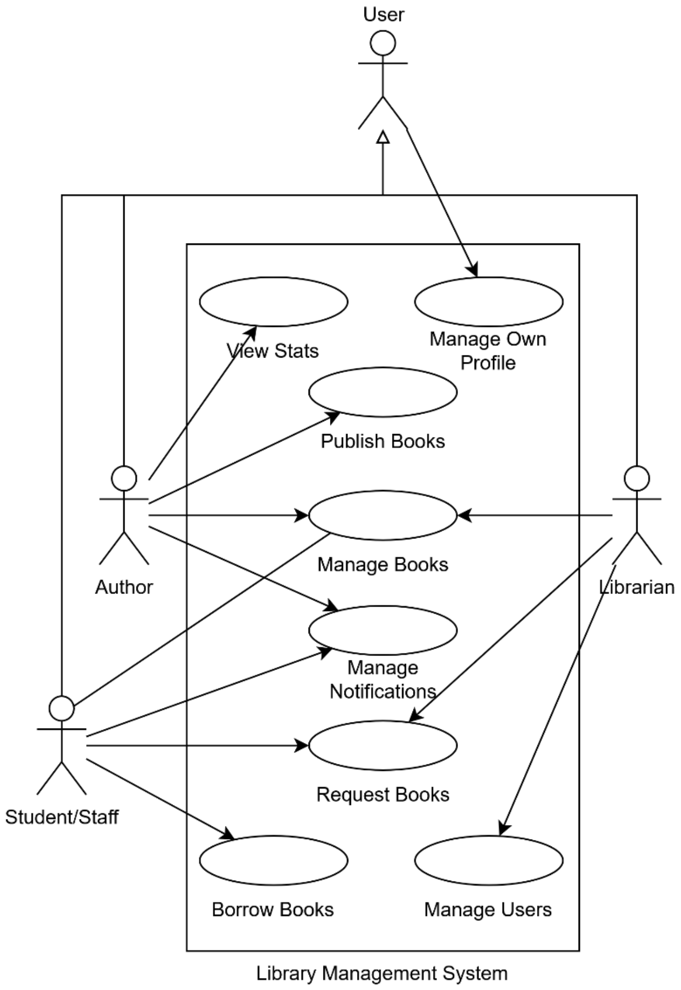
Task Allocation 1
Table of Contents 1
Class Diagram 2
Use-case Diagram 2
Use-case Specifications 3
Task 1 3
Use Case: Borrow Books 3
Use Case: Request Books 6
Use Case: Manage Notifications 7
Use Case: Manage Own Profile 8
Task 2 11
Use Case: Manage Books 11
Use Case: Publish Books 14
Use Case: View Stats 16
Use Case: Manage Notifications 17
Use Case: Manage Own Profile 18
Task 3 21
Use Case: Manage Own Profile 21
Use Case: Manage Users 24
Use Case: Manage Books 25
Use Case: Request Books 28

Class Diagram



Both new book approval, and modification approval always create a new AuthoredBook. The former does not have a link to another AuthoredBook while the latter has a link to the original AuthoredBook if it is approved. Otherwise, it does not have a link.

Use-case Diagram



Use-case Specifications

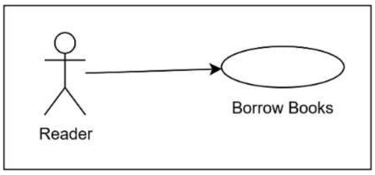
Task 1

Use Case: Borrow Books

Brief Description

This use case describes how a student/staff borrows, reads and returns books.

Use-case Diagram



Preconditions

The Student/Staff actor is logged into the system.

Postconditions

The Student/Staff actor successfully borrows a book with a set return date.

Basic Flow (1)

1. The use case begins when the Student/Staff actor chooses to borrow a book.
2. The Student/Staff selects the “Available Books” tab.
3. The system displays an interface for selecting books to borrow.
   {Select Book to Borrow}
4. The Student/Staff selects an available book from the list of available books.
5. The system retrieves and displays the complete details of the selected book.
6. The Student/Staff confirms his chosen book to borrow.
   {Begin Inputting Borrow Information}
7. The system displays an input dialog with options for the borrower to input the number of minutes and seconds to borrow the chosen book for.
   {Enter Borrow Duration}
8. The Student/Staff enters the duration that he would like to borrow the book for.
9. The Student/Staff confirms his chosen duration.
   {Confirm Borrow Information}
10. The system records the book as borrowed by the Student/Staff and stores the duration.
11. The system notifies the Student/Staff that he has borrowed the book successfully.
12. While the borrowed book has not been returned yet
    - 12.1. If the borrowed book has expired and the system detects the expiry through a routine check every 15 minutes
        - 12.1.1. The system automatically records the book as returned.
    - 12.2. If the borrowed book is deleted by a librarian
        - 12.2.1. The system removes the book from the Student/Staff’s list.
        - 12.2.2. The system sends a notification to the actor informing him of the deletion.
13. The use case ends.

Alternative Flows

A1: No Book Selected to Borrow

At **{Begin Inputting Borrow Information}** if no book has been selected yet,

1. The system informs the Student/Staff that he has not selected a book yet.
2. The flow of events is resumed at **{Select Book to Borrow}**.

A2: Invalid Duration

At **{Confirm Borrow Information}** if the entered duration is invalid,

1. The system informs the Student/Staff that the duration provided is invalid.
2. The flow of events is resumed at **{Enter Borrow Duration}**.

A3: Cancel Borrowing

At any point between **{Begin Inputting Borrow Information}** and **{Confirm Borrow Information}**,

1. The Student/Staff can exit the input dialog and cancel the borrow request.
2. The flow of events is resumed at **{Select Books}**.

* * *

Preconditions

The Student/Staff actor is logged in and has a borrowed book that has not been returned yet.

Postconditions

The Student/Staff actor successfully views the book, and the system stores any saved modifications a user makes for a book.

Basic Flow (2)

1. The use case begins when the Student/Staff wants to read a borrowed book.
2. The Student/Staff selects the “My Borrowed Books” tab.
3. The system displays the interface for reading and returning books.
   {Select Book to Read}
4. The Student/Staff selects a book to read among the list of borrowed books.
5. The Student/Staff selects the “Read Selected Book” option.
   {Display Book}
6. The system creates a window displaying the book in PDF format, along with any modifications the user has saved previously.
7. The reader can modify the book by highlighting text, adding bookmarks etc.
8. If the “Save” option is selected
   8.1. The changes the user has made to the PDF file will be saved by the system.
9. The reader closes the window when he has finished reading the chosen book.
10. The system discards any unsaved modifications by the reader.
11. The use case ends.

Alternative Flows

A1: No Book Selected to Read

At **{Display Book}** if the Student/Staff has not selected a book to read

1. The system informs the Student/Staff that he has not selected a book to read.
2. The flow of events is resumed at **{Select Book to Read}**.

A2: Create PDF on First Read

At **{Display Book}** if the Student/Staff is reading the book for the first time

1. The system converts the stored plaintext content of the book into a new PDF file and assigns it to the user. The new PDF will not have any special modifications like highlighting.
2. The flow of events is resumed at **{Display Book}**.

* * *

Preconditions

The Student/Staff actor is logged in and has a borrowed book that has not been returned yet.

Postconditions

The system records the book as returned by the Student/Staff.

Basic Flow (3)

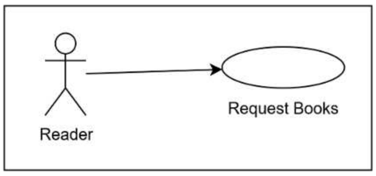
1. The use case begins when the Student/Staff wants to return a borrowed book.
2. The Student/Staff selects the “My Borrowed Books” tab.
3. The system displays the interface for reading and returning books.
4. In the list of borrowed books, the Student/Staff selects “Return” for the book to be returned.
5. The system records the book as returned by the Student/Staff.
6. The list of borrowed books is refreshed to remove the book that has been returned.
7. The system adds the returned book back to the list of available books that can be borrowed by the Student/Staff.
8. The use case ends.

Use Case: Request Books

Brief Description

This use case describes how a Student/Staff requests a new book.

Use-case Diagram



Preconditions

The Student/Staff actor is logged into the system.

Postconditions

The system records a new book request for the Librarian actors to review.

Basic Flow

1. The use case begins when the Student/Staff wants to request a new book.
2. The Student/Staff selects the “Request Book” tab.
3. The system displays the interface for requesting a new book.
   {Enter New Book Information}
4. The Student/Staff enters the title and author of the book he wants to request.
5. The Student/Staff submits the book request.
   {Submit Request}
6. The system records the book request.
7. The system notifies the Student/Staff that the book request is submitted.
8. If the book request is approved/rejected by a Librarian later
   8.1. The system sends a notification to the Student/Staff informing him of the approval or rejection.
9. The use case ends.

Alternative Flows

A1: Invalid or Missing Title or Author Name

At **{Submit Request}** if either the entered title or author name is invalid and/or missing,

1. The system informs the Student/Staff that the title and/or author name is invalid and/or missing.
2. The flow of events is resumed at **{Enter New Book Information}**.

A2: Same Book Requested Again

At **{Submit Request}** if both the title and author name is the same for a previous request,

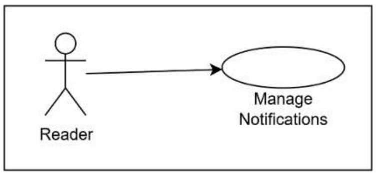
1. The system informs the Student/Staff that the request contains repeated information.
2. The flow of events is resumed at **{Enter New Book Information}**.

Use Case: Manage Notifications

Brief Description

This use case describes how a Student/Staff actor views and clears the system-related notifications he receives.

Use-case Diagram



Preconditions

The Student/Staff actor is logged into the system.

Postconditions

The Student/Staff actor will have seen and possibly removed the notifications he received.

Basic Flow

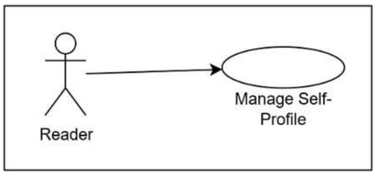
1. The use case begins when the Student/Staff wants to view and/or clear his notifications.
2. The Student/Staff selects the “Inform Board” tab.
3. The system displays a list of the Student/Staff’s notifications and a “Clear All” option.
4. While the Student/Staff has notifications he wants to remove
   - 4.1. If the “Clear All” option is selected
     - 4.1.1. All of the Student/Staff’s notifications are removed from the system, and the displayed notification list is cleared.
   - 4.2. If the “Clear” option of a notification is selected
     - 4.2.1. The system removes the notification and refreshes the notification list.
5. The use case ends.

Use Case: Manage Own Profile

Brief Description

This use case describes how a Student/Staff actor registers an account, logs in/out of the system and modifies their full name and password.

Use-case diagram



Preconditions

The Student/Staff actor does not have an existing account.

Postconditions

The system creates a new account and stores the new Student/Staff information.

Basic Flow (1)

1. The use case begins when the Student/Staff wants to create an account in the system.
   {Display Welcome Screen}
2. The system initially displays a welcome screen for different kinds of users.
3. The Student/Staff selects the login option for Students/Staff.
   {Display Login Screen}
4. The system displays the login screen for Students/Staff.
5. The Student/Staff selects the link to register for a new account.
   {Display Registration Screen}
6. The system displays the registration screen for Students/Staff.
   {Enter Account Details}
7. The Student/Staff enters their username, password and full name.
8. The Student/Staff confirms their details.
   {Create Account}
9. The system opens a new account with the details provided.
10. The system informs the Student/Staff that the account creation is successful.
11. The use case ends.

Alternative Flows

A1: Invalid Account Details

At **{Create Account}** if the username entered is already used by another user of the system, or if the username and/or password is invalid or missing,

1. The system informs the Student/Staff that the account details are invalid.
2. The flow of events resumes at **{Enter Account Details}**.

A2: Cancel Account Creation

At any point between **{Enter Account Details}** and **{Create Account}**,

1. The Student/Staff can return to the login screen by selecting the “Back” option.
2. The flow of events resumes at **{Display Login Screen}**.

A3: Exit Login Screen

At any point between **{Display Login Screen}** and **{Display Registration Screen}**,

1. The Student/Staff can return to the welcome screen by selecting the “Back” option.
2. The flow of events resumes at **{Display Welcome Screen}**.

* * *

Preconditions

The Student/Staff actor is logged out and has an existing account in the system that is active (i.e. not deactivated by a librarian).

Postconditions

The Student/Staff actor is logged into the system.

Basic Flow (2)

1. The use case begins when the Student/Staff wants to log into the system.
   {Display Welcome Screen}
2. The system initially displays a welcome screen for different kinds of users.
3. The Student/Staff selects the login option for Students/Staff.
   {Display Login Screen}
4. The system displays the login screen for Students/Staff.
   {Enter Username and Password}
5. The Student/Staff enters his username and password.
6. The Student/Staff confirms the details and selects “Login”.
   {Login}
7. The Student/Staff logs in, and the system displays the “Available Books” screen.
8. The use case ends.

Alternative Flows

A1: Username, Password Pair Not Found

At **{Login}** if the system does not have a Student/Staff with the provided username, or if the password is incorrect for that username,

1. The system informs the Student/Staff that the login details are incorrect.
2. The flow of events resumes at **{Enter Username and Password}**.

A2: Account Deactivated

At **{Login}** if the system finds that the actor’s account has been deactivated,

1. The system informs the Student/Staff that his account has been deactivated.
2. The flow of events resumes at **{Enter Username and Password}**.

A3: Cancel Login

At any point between **{Display Login Screen}** and **{Login}**,

1. The Student/Staff can return to the welcome screen by selecting the “Back” option.
2. The flow of events resumes at **{Display Welcome Screen}**.

***　　*　　***

Preconditions

The Student/Staff actor is logged into the system.

Postconditions

The Student/Staff actor will have a new name and/or a new password.

Basic Flow (3)

1. The use case begins when the Student/Staff wants to modify their full name or password.
2. The Student/Staff selects the “My Profile” tab.
3. The system displays the interface for updating the actor’s full name and password.
   {Enter Name and Password}
4. The Student/Staff enters the name and/or password they want to change to.
5. The Student/Staff confirms the changes.
   {Update Name and Password}
6. The system stores the new name and password.
7. The system displays a message informing the Student/Staff that their account has been updated successfully.
8. The use case ends.

Alternative Flows

A1: Invalid or Missing Name or Password

At **{Update Name and Password}** if the name or the password is invalid or missing,

1. The system informs the Student/Staff that the entered information is invalid.
2. The flow of events resumes at **{Enter Name and Password}**.

***　　*　　***

Preconditions

The Student/Staff actor is logged into the system, and there are no active dialog windows.

Postconditions

The Student/Staff actor is logged out of the system.

Basic Flow (4)

1. The use case begins when the Student/Staff wants to log out of the system.
2. The Student/Staff selects the “Logout” option present in every tab.
3. The system discards any information that has not been confirmed by the actor (e.g. text entered into input boxes).
4. The system records the Student/Staff as logged out.
5. The system displays the initial welcome screen for different kinds of users.
6. The use case ends.

Task 2

Use Case: Manage Books

Brief Description

This use case describes how the author manages their publications.

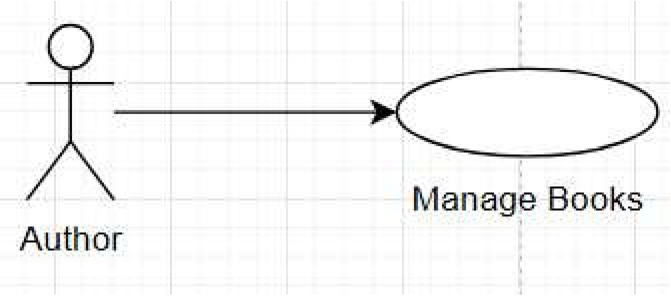
Precondition

1. The author is logged in the system
2. For any author, the titles of all his or her approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Postcondition

1. For any author, the titles of all his or her approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Use-case diagram



Basic Flow

1. The use case begins when the author wants to manage their publications.
2. The author selects the “My Books” tab
3. The system displays an interface for managing the author’s publications, showing all the pending books and approved books.

{Select Book}

4. The author selects the available book in his/her publications (non-rejected publications)

{Select Activity}

5. While the author has any activity to perform on the selected book
    - 5.1. If the author selects the “View” button.
        - *5.1.1.* Perform subflow *View Book.*
    - 5.2. If the author selects the “Modify” button.
        - 5.2.1. Perform subflow *Modify Book.*
    - 5.3. If the author selects the “Delete” button.
        - 5.3.1. Perform subflow *Delete Book.*

{End of Activity}

6. The use case ends.

Subflows

S1: View Book

{Begin View Book}

1. The system retrieves and displays the txt file of the selected book in a new txt reader window
2. While the author is reading the txt file
    2.1. The author can zoom in-out, or scroll up and down
3. The author closes the window, subflow ends.

S2: Modify Book

{Begin Modify Book}

1. If the selected book is approved and borrowed by any readers, subflow ends

{Modify Book}

2. The system retrieves and displays the “Title” and “Summary” of the selected book in a "Modify Book” window, with the corresponding input fields, initializes the original book title as the new title.
3. The author changes the “Title” or “Summary” in the input fields.
4. The author saves changes to the book

{Confirm Modifications to the Book}

5. If the book is an approved book,
    5.1. The system removed the book from the available list of student/staff.
    5.2. The system changes the status of the book to pending status.
    5.3. The changes are uploaded to the system along with the book.
6. If the book is a pending book, the system removes the original upload of the book and creates a new upload with the changed information.
7. The system closes the “Modify Book” window.
8. The system informs the author that modifications to the book are awaiting approval with a popup window.

{End of Modify Book}

9. The author closes the popup window, subflow ends.

S3: Delete Book

{Begin Delete Book}

1. If the selected book is approved and borrowed by any readers, subflow end.
2. The author selects the “Delete” button to delete the book.

{Confirm Delete Book}

3. The system asks author to confirm the author deleting selected book
4. If the author selects “Cancel” to stop deleting the book, subflow end.
5. The author selects the “Ok” button to confirm that he/she is deleting the selected book.
6. The system removes the book.

7. Subflow ends

Alternative Flows

A1: No Book to be managed

At **{Select Book}**, If author have no book having status pending or approved in the system or the author stop to select a book to perform activity,

1. The use case ends.

A2: Invalid Modification

At **{Confirm Modifications to the Book}**, If “New Book Title” is the same as the title of any publications of the author, no matter which status the book is,

1. The system informs the author that it is invalid to have the duplicated book.
2. The flow of events is resumed at **{Modify Book}.**

A3: No Modification

At **{Confirm Modifications to the Book}**, If author did not make any changes,

1. The author closes the “Modify Book” window.
2. The flow of events is resumed at **{Select Activity}.**

A4: Select another book to perform activities

At **{End of Activity}**, If author delete the selected book or the author wants to perform activities for another book,

1. The flow of events is resumed at **{Select Book}.**

A5: Logout

At any point between **{Select Book}** and **{End of Activity}**, if the author wants to log out,

1. If the author has any extra windows or popups exist other than the base window, he/she closes the windows or popups selecting the option cancels the ongoing operation (i.e. cancel in delete book).
2. The author selects the “Log out” button.
3. The system discards all input that is not saved to the system.
4. The system logged the author out.
5. The system displays the interface for the user to choose his/her account role to login.
6. The use case ends.

Use Case: Publish Books

Brief Description

This use case describes how the author publishes a new book

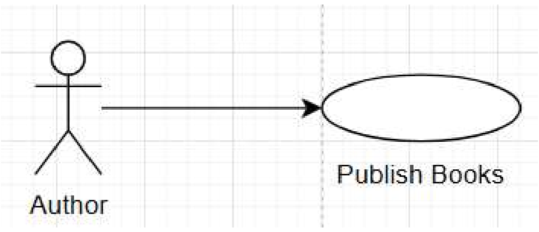
Precondition

1. The author is logged in the system
2. For any author, the titles of all his or her approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Postcondition

1. For any author, the titles approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Use-case diagram



Basic Flow

1. The use case begins when author want to publish a new book to the system
2. The author selects the “Publish New Book” tab,
3. The system displays the interface for publishing new book, showing input field for “Book Title”, “Summary” and “Upload text file” button.

{Enter Book Information}

4. The author enters “Book title”, “Summary”, and upload the “Text file” of the book
5. The author selects the “Submit” button to publish the new book to the system.

{Publish the Book}

6. The system saves the new book information in the back end.
7. The system informs the author that his/her new book is awaiting approval using a popup window.
8. The system sets the new book as pending status initially.
9. The use case ends.

Alternative Flows

A1: Alternative way to fill “Summary” of book

At **{Enter Book Information}**, if “Book Title” is not empty

1. The author request the system to generate the “Summary” based on the “Book Title” using the “Generate” button.
2. The flow of events resumes at **{Enter Book Information}.**

A2: Missing book information

At **{Publish the Book}**, if any of the input fields “Book title”, “Summary”, “Text file” empty,

1. The system informs the author that the information of the new book is/are not completed
2. The flow of events is resumed at **{Enter Book Information}**

A3: Publishing existing book in system (Handling duplicate and rejected publications)

At **{Publish the Book}**, if the author publishes the book that have same “Book Title” in his/her existing publications of all status (Define 2 books having same author and same title are the same),

1. The system informs the author that it is not allowed to upload the same or rejected book again.
2. The flow of events is resumed at **{Enter Book Information}**

A4: Logout

At any point between **{Enter Book Information}** and **{Publish the Book}**, if the author wants to log out,

1. If the author has any extra windows or popups exist other than the base window, he/she closes the windows or popups selecting the option cancels the ongoing operation (i.e. cancel in delete book).
2. The author selects the “Log out” button.
3. The system discards all input that is not saved to the system.
4. The system logged the author out.
5. The system displays the interface for the user to choose his/her account role to login.
6. The use case ends.

Use Case: View Stats

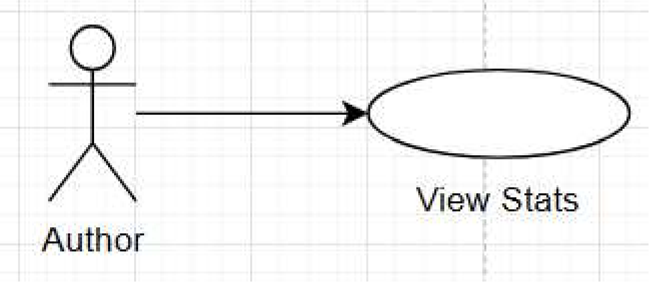
Brief Description

This use case describes how the author views the stats for their publications

Pre-condition

The author is logged in the system

Use-case diagram



Basic Flow

1. The use case begins when author want to view the stats for their publications.
2. The author selects the “Status View” tab
3. The system displays the interface for viewing the stats of the publications of that author.

{View Stats}

4. The system retrieves the data and generates chart(s) for the relevant stats for author’s publication(s).

{End View Stats}

5. The use case ends.

Alternative Flows

A1: Refresh the stats

At **{End View Stats}**, if the author wants to refresh and see the stats again,

1. The author selects the “Refresh” button.
2. The flow of events resumed at **{View Stats**}

A2: Logout

At any point between **{View Stats}** and **{End View Stats}**, if the author wants to log out,

1. The author selects the “Log out” button.
2. The system logged the author out.
3. The system displays the interface for the user to choose his/her account role to login.
4. The use case ends.

Use Case: Manage Notifications

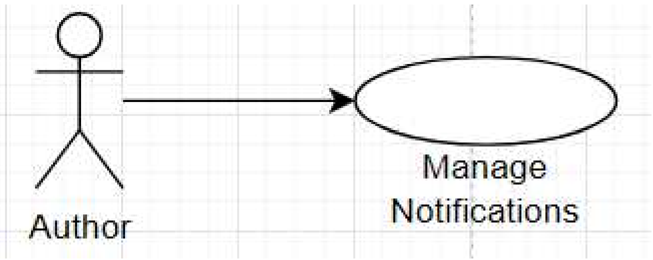
Brief Description

This use case describes how the author manages notifications

Pre-condition

The author is logged in the system

Use-case diagram



Basic Flow

1. The use case begins when the author wants to view his/her notifications or clear his/her notifications.
2. The author selects the “Inform Board” tab,
3. The system displays the interface for managing notifications, showing the list of notifications of that author.

{Manage Notification}

4. While the author has notification(s) he/she wants to remove
    - 4.1. If “Clear All” selected
        - 4.1.1. All notifications of author are removed from the system, system display a cleared list to author.
    - 4.2. Otherwise
        - 4.2.1. The author selects the “Clear” button of his/her target notifications.
        - 4.2.2. The system removes that notification and refreshes the notification list.
5. While the author has notification(s) he/she wants to read, the author stays in the interface.

{End Manage Notification}

6. The use case ends.

Alternative Flows

A1: Logout

At any point between **{Manage Notification}** and **{End Manage Notification}**, if the author wants to log out,

1. The author selects the “Log out” button.
2. The system logged the author out.
3. The system displays the interface for the user to choose his/her account role to login.
4. The use case ends.

Use Case: Manage Own Profile

Brief Description

This use case describes how the author manages their profile, logging in their account.

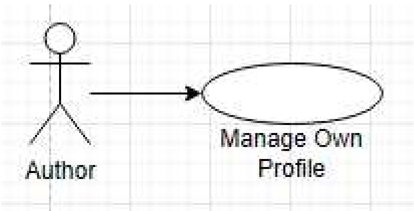
Precondition(1):

1. The author is not logged in the system.

Postcondition(1):

2. The author is logged in the system.

Use-case diagram



Basic Flow (1)

1. The use case begins when the author wants to login his/her account.
2. The system displays the interface for user to choose his/her account role to login.
3. The author selects the “Author Login” button.

{Login}

4. The system displays the interface for the author entering “Username” and “Password” for login.
5. Perform subflow *Login*.
6. The use case ends.

Subflow(1):

S1: Login

{Begin Login}

1. The author enters the “Username”, “Password” of his/her account.
2. The author selects the “Login” button.

{End of Login}

3. The system logged the author in the system.
4. The system displays the “My Books” tab, the interface for the author’s publications, showing his/her appending and approved publications.
5. Subflow ends.

Alternative Flows (1)

A1: Registration for new account

At **{Login}**, if the author did not have an account in the system,

1. The author selects the “Registration” button.

{Begin Account Creation}

2. The system displays the interface for registering an account.

{Fill in Registration Information}

3. The author enters the “Username”, “Password” and “Full name” for his/her account.
4. The author selects the “Register” button to submit the account’s information.

{Confirm Account Creation}

5. The system receives the information
6. The system creates an account with the information in the backend.
7. The system informs the author that the new account created.
8. The flow of events is resumed at **{Login}.**

A2: Invalid Registration Information

At **{Confirm Account Creation}**, if [“Username” is not available (exist same username in the system)] or [“Username”, “Password” or “Full name” are invalid (empty also type of invalid)]

1. The system informs the author that the registration information is/are invalid or incomplete.
2. The flow of events is resumed at **{Fill in Registration Information}.**

A3: Invalid Username or Password

At **{End of Login}**, if the “Username” does not exist in the system or “Username” did not match the “Password”,

1. The system informs the author the either the username or password is invalid.
2. The flow of events is resumed at **{Begin Login}.**

A4: Login a deactivated account

At **{End of Login}**, if the account with the given “Username” and “Password” exist and is deactivated,

1. The system informs the author that his/her account is deactivated.
2. The flow of events is resumed at **{Begin Login}**

Precondition (2):

The author is logged in the system

Basic Flow (2)

1. The use case begins when the author wants to update their “name” or “password”.
2. The author selects the “My Profile” tab
3. The system displays the interface for updating the profile.

{Update Profile}

4. The system displays the interface for the author’s profile, initializing the value of the field “New name” as the original name.
5. The author enters/modifies the new name or new password or both of them.
6. The author confirms the update of the profile with the non-empty new profile information.

{Confirm Profile Update}

7. The system updates the author's profile for each given information. No update for the corresponding field if the empty value provided for that field.
8. The system informs the author that his/her account has been updated with the new information using a popup window.
9. The use case ends.

Alternative Flows (2)

A1: Invalid New Profile Information

At **{Confirm Profile Update}**, if ["New name" and "New Password" are both empty] or ["New Password" or "New name" is/are invalid],

1. The system informs the author that the new profile information is/are invalid or incomplete.
2. The flow of events is resumed at **{Update Profile}.**

A2: Logout

At any point between **{Update Profile}** and **{Confirm Profile Update}** if the author wants to log out,

1. The author selects the "Log out" button.
2. The system logged the author out.
3. The system displays the interface for the user to choose his/her account role to login.
4. The use case ends.

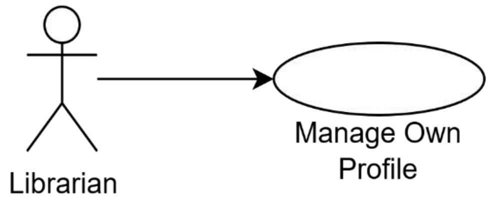
Task 3

Use Case: Manage Own Profile

Brief Description

The use case describes how a librarian registers, logs in, or manages his or her own profile.

Use Case Diagram



Preconditions (1)

- The librarian is not logged in.
- The librarian does not have his or her own librarian profile.

Preconditions (2)

- The librarian is not logged in.
- The librarian has his or her own librarian profile.

Preconditions (3)

- The librarian is logged into his or her own librarian profile.

Basic Flow (1)

1. The use case starts when a librarian wants to register his or her own profile.
2. The system displays interface for choosing login.
3. The librarian clicks the “librarian login” button.
4. The system displays interface for logging into a librarian profile, which prompts for username and password.
5. The librarian clicks the “register” button.
6. The system displays interface for registering a librarian profile, which prompts for username, password, and full name.
7. Perform subflow **Registration**.
8. The use case ends.

Basic Flow (2)

1. The use case starts when a librarian wants to login his or her own profile.
2. The system displays interface for choosing login.
3. The librarian clicks the “librarian login” button.
4. The system displays interface for logging into a librarian profile, which prompts for username and password.
5. Perform subflow **Login**.

6. The use case ends.

Basic Flow (3)

1. The use case starts when a librarian wants to manage his or her own profile.
   {Begin Logged In}
2. The system displays the interface for logged-in librarians, which may be on any tab.
3. The librarian clicks the “my profile” tab button.
4. The system displays interface for editing his or her profile details, which displays the username (fixed) and prompts for new full name (prefilled with existing full name) and new password.
5. Perform subflow **Update Profile**.
   {End Logged In}
6. The use case ends.

Subflows

S1: Registration

{Begin Registration}

1. The librarian enters his or her preferred username, password, and full name.
2. The librarian confirms the new profile details.
   {Confirm Registration}
3. The system creates a new profile with the details provided.
4. The system informs the librarian that a librarian profile has been created.
5. The system displays interface for logging into a librarian profile.
6. The subflow ends.

S2: Login

{Begin Login}

1. The librarian enters his or her username and password.
2. The librarian confirms the login details.
   {Confirm Login}
3. The system logs the librarian into his or her profile.
4. The system displays the interface for pending approvals (the default librarian interface).
5. The subflow ends.

S3: Update Profile

{Begin Updating Profile}

1. The librarian enters his or her new preferred full name or password.
2. The librarian confirms the new details for updating the profile.
   {Confirm Updating Profile}
3. The system updates his or her profile full name, and also its password if the provided password is nonempty.
4. The system informs the librarian that profile has been updated.
5. The subflow ends.

Alternative Flows

A1: Invalid New Profile Details

At **{Confirm Registration}** if the provided username is already used; or username, password, or full name is invalid (nonempty and some other reasonable restrictions)

1. The system informs the librarian that the details provided are invalid and provides the reason.
2. The flow of events resumes at **{Begin Registration}**.

A2. Invalid Login Details

At **{Confirm Login}** if the provided username does not exist, username does not correspond to a librarian profile, or password is wrong

1. The system informs the librarian that the login details provided are invalid and provides the reason.
2. The flow of events resumes at **{Begin Login}**.

A3: Deactivated Profile

At **{Confirm Login}** if the provided combination of username and password is valid, and the corresponding profile is deactivated

1. The system informs the librarian that his or her profile is deactivated.
2. The use case ends.

A4: Logout

At any point between **{Begin Logged In}** and **{End Logged In}** if there is no active dialog and the librarian wants to logout

1. The librarian clicks the “logout” button to logout.
2. The system logs the librarian out.
3. The system informs the librarian that he or she has been logged out.
4. The system displays interface for choosing login.
5. The use case ends.

A5: Invalid Update Profile

At **{Confirm Updating Profile}** if the provided full name or password is invalid (empty password is valid)

1. The system informs the librarian that the details provided are invalid and provides the reason.
2. The flow of events resumes at **{Begin Updating Profile}**.

Postconditions (1)

- The librarian is not logged in.
- The librarian has his or her own librarian profile.

Postconditions (2)

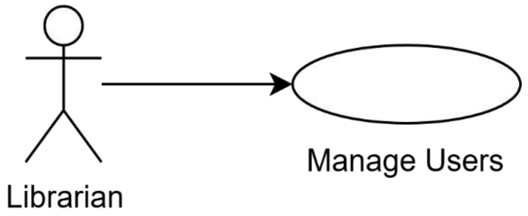
- The librarian is logged into his or her own librarian profile.

Use Case: Manage Users

Brief Description

The use case describes how a librarian manages other users that are not his or her own profile.

Use Case Diagram



Preconditions

- The librarian is logged into his or her own librarian profile.

Basic Flow

1. The use case starts when a librarian wants to manage other users, including all student/staff, all authors, and other librarians.
   {Begin Logged In}
2. The system displays the interface for logged-in librarians, which may be on any tab.
3. The librarian clicks the “manage users” tab button.
4. The system displays a list of all users (including librarian’s own profile), each with username, role, name, active status, and actions to take (activate/deactivate; actions are empty for the librarian’s own profile).
5. For each other user that the librarian wants to activate or deactivate
   5.1. Perform subflow **Activate/Deactivate Other User**.
   {End Logged In}
6. The use case ends.

Subflows

S1: Activate/Deactivate Other User

1. If the user is originally active
   1.1. The librarian clicks the “deactivate” button in the user entry.
   1.2. The system sets the user’s active status to inactive.
   1.3. The system informs the librarian that the user has been deactivated.
2. If the user is originally inactive
   2.1. The librarian clicks the “activate” button in the user entry.
   2.2. The system sets the user’s active status to active.
   2.3. The system informs the librarian that the user has been activated.
3. The system refreshes the “manage users” interface.
4. The subflow ends.

Alternative Flows

A1: Logout

At any point between **{Begin Logged In}** and **{End Logged In}** if there is no active dialog and the librarian wants to logout

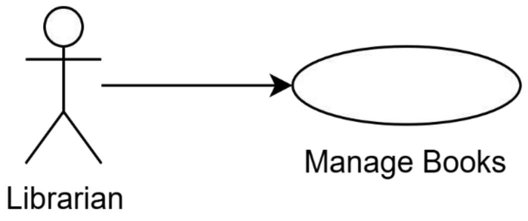
1. The librarian clicks the “logout” button to logout.
2. The system logs the librarian out.
3. The system informs the librarian that he or she has been logged out.
4. The system displays interface for choosing login.
5. The use case ends.

Use Case: Manage Books

Brief Description

The use case describes how a librarian manages existing approved books or processes pending book approvals from authors.

Use Case Diagram



Preconditions (1)

- The librarian is logged into his or her own librarian profile.

Preconditions (2)

- The librarian is logged into his or her own librarian profile.
- For any author, the titles of all his or her approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Basic Flow (1)

1. The use case starts when a librarian wants to manage existing approved books.
   {Begin Logged In}
2. The system displays the interface for logged-in librarians, which may be on any tab.
3. The librarian clicks on the “published books” tab button.
4. The system displays a list of published books, each with title, author (which is the author’s full name), summary, and actions to take (view and delete).
5. For each published book the librarian wants to manage
   5.1. Perform subflow **Manage Book**.
   {End Logged In}
6. The use case ends.

Basic Flow (2)

1. The use case starts when a librarian wants to process pending book approvals from authors.

   {Begin Logged In}
2. The system displays the interface for logged-in librarians, which may be on any tab.
3. The librarian clicks on the "pending approvals" tab button.
4. The system displays a list of pending book approvals, each with title, author (which is the author's username), summary, and actions to take (view, approve, and reject).
5. For each pending book approval in the list
   5.1. Perform subflow **Approve/Reject Pending Book Approval**.

   {End Logged In}
6. The use case ends.

Subflows

S1: Manage Book

{Evaluate Book}

1. If the librarian decides to delete the book
   1.1. The librarian clicks on the "delete" button for the corresponding book entry.
   1.2. The system informs the librarian the book will be deleted and any borrowing privileges are revoked, and asks if he or she really wants to delete the book.
   1.3. If the librarian clicks the button "yes"
      1.3.1. The system notifies the book author and all students and staff borrowing the book of the deletion.
      1.3.2. The system deletes the book and borrowings associated with it.
      1.3.3. The system informs the librarian that the book is deleted.
   1.4. If the librarian clicks the button "no"
      1.4.1. The librarian closes the dialog.

2. The subflow ends.

S2: Approve/Reject Pending Book Approval

{Evaluate Pending Book Approval}

1. If the librarian decides to accept the pending book approval
   1.1. The librarian clicks on the "accept" button for the corresponding pending approval entry.
   1.2. If the pending approval is for a new book
      1.2.1. The system marks the new book as approved and sets its published date to now.
   1.3. If the pending approval is for modifying a book
      1.3.1. The system marks the new book as approved, transfer borrowings of the old book to the new book and deletes the old book.
   1.4. The system notifies the author of the acceptance.
   1.5. The system informs the librarian that the request has been accepted, and a notification has been sent to the author.
2. If the librarian decides to reject the pending approval

   2.1. The librarian clicks on the “reject” button for the corresponding pending approval entry.
   2.2. The system marks the new book as rejected.
   2.3. The system notifies the author of the rejection.
   2.4. The system informs the librarian that the request has been rejected, and a notification has been sent to the author.
3. The system refreshes the “pending approvals” interface.
4. The subflow ends.

Alternative Flows

A1: Logout

At any point between **{Begin Logged In}** and **{End Logged In}** if there is no active dialog and the librarian wants to logout

1. The librarian clicks the “logout” button to logout.
2. The system logs the librarian out.
3. The system informs the librarian that he or she has been logged out.
4. The system displays interface for choosing login.
5. The use case ends.

A2: Read Book to Evaluate Book

At **{Evaluate Book}** if the librarian decides to read the book to decide what actions should be taken for the book, if any

1. The librarian clicks on the “view” button for the corresponding book entry.
2. The system opens a new dialog window, which displays a text viewer with scroll and zoom functionality containing the book content.
3. The librarian reads the book to help decide what actions to take for the book, if any.
4. The librarian closes the dialog window after reading the book.
5. The flow of events resumes at **{Evaluate Book}**.

A3: Check Book Borrowings

At **{Evaluate Book}** if the librarian decides to check if a book is borrowed to decide what actions should be taken for the book, if any

1. The librarian clicks on the “borrowed books” tab button.
2. The system displays a list of borrowings, each with title, borrower (use username), borrow date, and time left.
3. The librarian checks if the book is borrowed or not to help decide what actions to take for the book, if any.
4. The librarian clicks on the “published books” tab button.
5. The system displays a list of published books, each with title, author (which is the author’s full name), summary, and actions to take (view and delete).
6. The flow of events resumes at **{Evaluate Book}**.

A4: Read Book to Evaluate Pending Book Approval

At **{Evaluate Pending Book Approval}** if the librarian decides to read the book to evaluate the pending book approval

1. The librarian clicks on the “view” button for the corresponding approval entry.
2. The system opens a new dialog window, which displays a text viewer with scroll and zoom functionality containing the book content.
3. The librarian reads the book to help decide if the pending book approval should be accepted.
4. The librarian closes the dialog window after reading the book.
5. The flow of events resumes at **{Evaluate Pending Book Approval}**.

Postconditions (2)

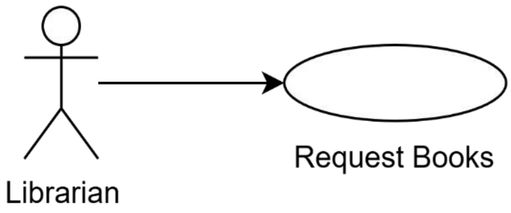
- For any author, the titles of all his or her approved, pending, or rejected books are unique, except for book modification approval requests, which can have the same title as the original book.

Use Case: Request Books

Brief Description

The use case describes how a librarian processes book requests from students and staff.

Use Case Diagram



Preconditions

- The librarian is logged into his or her own profile.

Basic Flow

1. The use case starts when a librarian wants to process book requests from students and staff.
   {Begin Logged In}
2. The system displays the interface for logged-in librarians, which may be on any tab.
3. The librarian clicks on the “request books” tab button.
4. The system displays a list of book requests from students and staff, each entry with title, author, requester (use username), request date, and actions to take (confirm and reject).
5. For each book request in the list
   5.1. Perform subflow **Accept/Reject Book Request**.
   {End Logged In}
6. The use case ends.

Subflows

S1: Accept/Reject Book Request

1. If the librarian decides to accept the book request
   1.1. The librarian clicks on the “book download” tab button.

1.2. The system displays the interface for downloading books online, with a search bar, empty list (with at least two headers, title and author), and empty book details.

{Search Books}

1.3. The librarian enters the appropriate search terms to search.

1.4. The system informs the librarian that it is searching for books.

1.5. The system searches for books online using the search terms provided.

{System Searching Books}

1.6. The system populates the empty list with book results.

1.7. The system informs the librarian the number of book results found and time taken.

1.8. While the librarian is deciding on the appropriate book to download and there are unevaluated book results

1.8.1. The librarian clicks on a book result he or she is interested in

1.8.2. The system populates the book details with details of the book result.

1.8.3. The librarian evaluates if the book result is appropriate for the book request.

{Download Book}

1.9. The librarian clicks on the “download selected” button.

{Confirm Download Book}

1.10. The system informs the librarian that it is downloading the book.

1.11. The system downloads the book online.

{System Downloading Book}

1.12. The system adds and publishes the book with the downloaded metadata (title, author, summary) and its publish date set to now.

1.13. The system informs the librarian that the book has been downloaded and published.

{Accept Book Request}

1.14. The librarian clicks on the “request books” tab button.

1.15. The librarian clicks the “confirm” button for the corresponding entry.

1.16. The system notifies the student or staff of the acceptance.

2. If the librarian decides to reject the book request

{Reject Book Request}

2.1. The librarian clicks the “reject” button for the corresponding entry.

2.2. The system notifies the student or staff of the rejection.

3. The system removes the book request.
4. The system refreshes the “request books” interface.
5. The subflow ends.

Alternative Flows

A1: Logout

At any point between **{Begin Logged In}** and **{End Logged In}** if there is no active dialog and the librarian wants to logout

1. The librarian clicks the “logout” button to logout.
2. The system logs the librarian out.

3. The system informs the librarian that he or she has been logged out.
4. The system displays interface for choosing login.
5. The use case ends.

A2: Error Searching or Downloading Books

At **{System Searching Books}** or **{System Downloading Book}** if an error from outside the system, e.g. internet problems, occurs

1. The system informs the librarian of the error.
2. The use case ends.

A3: No Appropriate Book Results

At **{Download Book}** if the librarian finds all book results are not appropriate for the book request (including no book results)

1. If the librarian decides that the previously used search terms are not exhaustive
   - 1.1. The librarian thinks of better search terms.
   - 1.2. The flow of events resumes at **{Search Books}**.
2. If the librarian decides that the previously used search terms are exhaustive
   - 2.1. The flow of events resumes at **{Reject Book Request}**.

A4: Replacing an Existing Book

At **{Confirm Download Book}** if a book of the same title and author (where author is compared against downloaded books only) is already in the system

1. The system informs the librarian that a book of the same title and author is in the system, shows the existing book's summary, and provides the buttons "yes" and "no".
2. If the librarian decides to replace the existing book
   - 2.1. The librarian clicks the "yes" button.
   - 2.2. The flow of events resumes at **{Confirm Download Book}**.
3. If the librarian decides to not replace the existing book
   - 3.1. The librarian clicks the "no" button and closes the dialog.
   - 3.2. The flow of events resumes at **{Accept Book Request}**.