

Systems Requirements Documentation - Library Management System Documentation

Logan Young

- Page 2: Library Management System Proposal
- Page 4: Problem Statements and System Requirements Document
- Page 9: Functional Requirements Specification
- Page 12: Sequence and Activity Diagrams
- Page 20: User Interface Specification
- Page 34: Traceability Matrix and Updates
- Page 37: Development Plan

Library Management System Proposal

Problem Statement

A library management system is used to manage the books in a library through the use of an automated system. Basically, the system tracks the status of books and other media throughout the checkout process (whether a book is checked out or available, in storage, reserved, etc.). It can be difficult to keep track of everything a larger-sized library might carry through typical analog filing systems, so automated management systems are used to keep everything organized.

System Objectives

The library management system (LMS) will contain the entire inventory of the library in a database, converting various properties of books and other media to attributes. It will also keep track of if a book is available or has been loaned out, if it's reserved, if it's damaged, etc. Its primary use is for managing the library's inventory, but there will also be a frontend to allow library members to check out books.

System Requirements

- Let library members check out books.
- Let library members reserve books.
- Let library members request a book from a library in the same network.
- Let library staff mark a book as returned.
- Let library staff create a new entry for a new book the library has received.
- Let library staff delete an entry for a book the library no longer possesses.
- Let library staff mark a user as a new library member.
- Let library staff adjust a book's properties if there's been a mistake in filing (e.g. wrong publish date, wrong author).
- Let library administrator mark a user as new library staff.
- Let library administrator perform various administrative tasks (not too sure what this might entail).

Typical Customers

- Library members (the people who check out the books)
- Library staff (the people who work at the library)
- Library administrator (the person in charge of the library)
- IT Staff (the people who work on the LMS, supposing they're separate from staff)

Project Planning and Development Approach

1. Software
 - a. Frontend: React (or JavaScript if I can't learn React in time)
 - b. Backend: Java
 - c. Database: MySQL
2. Hardware
 - a. PC (or device with an internet browser to access the web app)
 - b. Small server in the library for the database.
3. Network
 - a. [In all honesty, the LMS database and web app shouldn't be so intensive as to need high-speed networking, but it's good to future-proof.]
 - b. Fiber-optic internet with ethernet integration to allow quick uploading and downloading from the server to the web app.

Development Plan

- Weeks 1-2
 - o Conceptualize and complete the system proposal.
- Weeks 3-4
 - o Determine connection between frontend, backend, and database and start construction.
- Weeks 5-7
 - o Build database to organize books and their attributes, build frontend webpage for users to interact with, build backend to connect the two.
- Week 8
 - o Test system with sample entries and actions and record demo for midterm.
- Weeks 9-11
 - o Improve system with feedback from professor. Implement suggested features and bug fixes.
- Weeks 12-14
 - o Write test cases and optionally add some cool last-minute features.
- Week 15
 - o Test final system and record demo for final.

Problem Statements and System Requirements Document

Logan Young

Customer Problem Statement

Problem Statement

A **Library Management System** keeps track of all the goings-on in a library. It catalogs the entire inventory of the library, keeping track of each attribute of each piece of media. It also logs the library's members, its staff, and its administrator as users. If a member checks out a book the LMS will alter that book's checkout status to reflect that change. A member can also view what books are available and submit requests to hold a book. Other users have more privileges related to the system, such as being able to mark a book as unavailable manually or remove a member's records.

Glossary of Terms

LMS – (Library Management System) A tool to electronically manage a library.

User – An entity that uses the LMS to interact with the library.

Member – An LMS user that can only check out and return books.

Staff – An LMS user with elevated privileges related to library upkeep.


Administrator – LMS staff with further elevated privileges.

Checked-out – Status of a book after a user has reserved it. Unavailable for use by other users until returned.

Hold – Status of a book after a user has submitted a hold request. No user other than the one that requested the hold can check it out after it becomes available.

Attribute – Some quality of a book, such as its title or author.

Privilege – Specific groups of users can access more of the LMS than others.

 Welcome, John Library. LMS	
<p style="text-align: center; font-weight: bold; margin-bottom: 10px;">Library Member</p> <div style="display: flex; justify-content: space-around; align-items: center; height: 100px;"> <div style="border: 1px solid black; width: 100px; height: 100px;"></div> <div style="border: 1px solid black; padding: 5px; text-align: center;">Return by 11/20</div> </div> <div style="border: 1px solid black; height: 100px; margin-top: 10px;"> <!-- Hand-drawn scribbles representing a list --> </div>	<p style="text-align: center; font-weight: bold; margin-bottom: 10px;">Books</p> <div style="display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 10px;"> <div style="border: 1px solid black; width: 100px; height: 100px;"></div> <div style="border: 1px solid black; width: 100px; height: 100px;"></div> <div style="border: 1px solid black; padding: 5px; text-align: center;">Unavailable</div> <div style="border: 1px solid black; width: 100px; height: 100px;"></div> <div style="border: 1px solid black; padding: 5px; text-align: center;">On Hold</div> <div style="border: 1px solid black; width: 100px; height: 100px;"></div> </div>
Copyright, Trademarked, Reserved, Monetized - Logan Young	

System Requirements

Functional Requirements

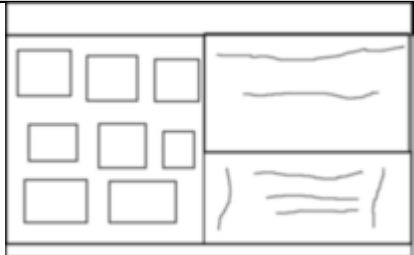
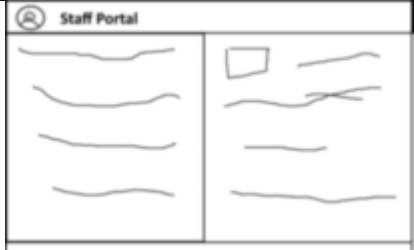
No.	Priority Weight	Description
REQ-1	High	The LMS should be able to store and display books.
REQ-2	High	Library members should be able to use the LMS to check out books.
REQ-3	Medium	Library members should be able to use the LMS to submit a hold request for a maximum of three books.
REQ-4	High	Library staff should be able to use the LMS to mark a book as unavailable.
REQ-5	High	Library members should be able to delete their account from the LMS.
REQ-6	Medium	Library staff should be able to delete a member's account using the LMS.
REQ-7	High	Library staff should be able to convert a new user into a member.
REQ-8	Medium	Library staff should be able to adjust a book's attributes if incorrect.
REQ-9	High	Library staff should be able to create a new entry in the LMS for a newly received book.
REQ-10	Low	Library members should be able to request a book from a different library in the same network.
REQ-11	Medium	Library administrator should be able to convert a new user into staff.
REQ-12	Low	Library administrator should have (nearly) complete access to the LMS.
REQ-13	Medium	Library members should be able to submit a suggestion for new books for the library.





REQ-14	Medium	Library members should be able to create a new donated book entry with approval from staff.
REQ-15	High	LMS users should be able to easily search for a specific book using various filters and sorting techniques.

Nonfunctional Requirements

No.	Priority Weight	Description
N-REQ-1	High	The LMS should be scalable to include as many books as possible.
N-REQ-2	Medium	The LMS should be able to support at least 100 concurrent users.
N-REQ-3	High	The LMS front-end (HTML and CSS) should pass W3C validation with no errors.
N-REQ-4	High	The LMS front-end should pass all test cases for Google Chrome, Microsoft Edge and Mozilla Firefox.
N-REQ-5	Medium	All database processes should be completed within a maximum of 30 seconds.
N-REQ-6	High	LMS users should have password-protected accounts.
N-REQ-7	Medium	Front-end and back-end code should be written with readability and maintainability in mind.
N-REQ-8	Medium	Database should support placeholder entries for testing purposes that won't appear for users with insufficient privileges.
N-REQ-9	Low	Database should support different types of media rather than only books.
N-REQ-10	Medium	Full database backups should be performed weekly.
N-REQ-11	High	Frontend updates should occur only on Saturdays at 2:00 a.m. EST.

User Interface Requirements

No.	Priority Weight	Description	Graphic
UI-REQ-1	High	The website should be neat and organized.	
UI-REQ-2	High	Once logged in, each user should see specific options available to them according to their privileges.	

UI-REQ-3	Medium	No backend or database operations should be overtly visible to the user.	
UI-REQ-4	Medium	UI changes should be implemented based on consensus from users and council of IT department.	
UI-REQ-5	High	All essential UI components should be screen-readable (including alt text for images, default text for errors, transcripts for videos).	
UI-REQ-6	Low	UI components should scale based on monitor resolution.	

Plan of Work

- Weeks 1-2
 - o Conceptualize and complete the system proposal.
- Weeks 3-4 (**We are here.**)
 - o Determine connection between frontend, backend, and database and start construction.
 - **Going well so far. My backend has switched from Java to Node.js. I've set up the MySQL schema and started on the HTML for the site. Connecting all three is proving a little confusing, but I'll figure it out.**
- Weeks 5-7

- Build database to organize books and their attributes, build frontend webpage for users to interact with, build backend to connect the two.
- Week 8
 - Test system with sample entries and actions and record demo for midterm.
- Weeks 9-11
 - Improve system with feedback from professor. Implement suggested features and bug fixes.
- Weeks 12-14
 - Write test cases and optionally add some cool last-minute features.
- Week 15
 - Test final system and record demo for final.

Functional Requirements Specification

Logan Young

Use Case Specifications

Stakeholders

1. Library staff – helpful tool to compile information on their library's stock and people who use the library
2. Library members – allows them to see what books are in stock and perform a variety of useful actions
3. Library administrators – allows them to perform administrative duties automatically and/or digitally that would otherwise be time-consuming
4. Library board of trustees – allows them to gauge interest in the library and better understand their goals
5. Local governments – a better functioning library helps to better educate the populace and the populace in turn improves their community and their government

Actors and Goals

Primary Actors:

1. Library member: This actor checks out books, marks books as returned, and places holds.
2. Library staff: This actor manages member accounts and book statuses.
3. Library administrator: This actor manages the entire system, holding the highest privileges.

Secondary Actors:

1. System: This stores information about library users, contents, etc. and performs operations.

Use Cases

Library Member (Total: 14)

1. Check out book: To check out a book from the library. (2)
2. Return book: To return a book to the library. (2)
3. Hold request: To request a hold on a book from the library. (1)
4. Procurement request: To request a book from a different library in local network. (3)
5. Delete self: To delete a user's own account from the LMS. (2)
6. Submit suggestion: To submit a suggestion for a new book to the library. (1)
7. Donate book: To donate a book to the library. (3)

Library Staff (Total: 11)

1. Hide book: To mark a book as unavailable. (1)
2. Delete account: To delete a member's account. (2)
3. Create account: To create a new account for a member. (3)
4. Adjust book: To adjust a book's attributes. (3)
5. Create book: To create a new book. (2)

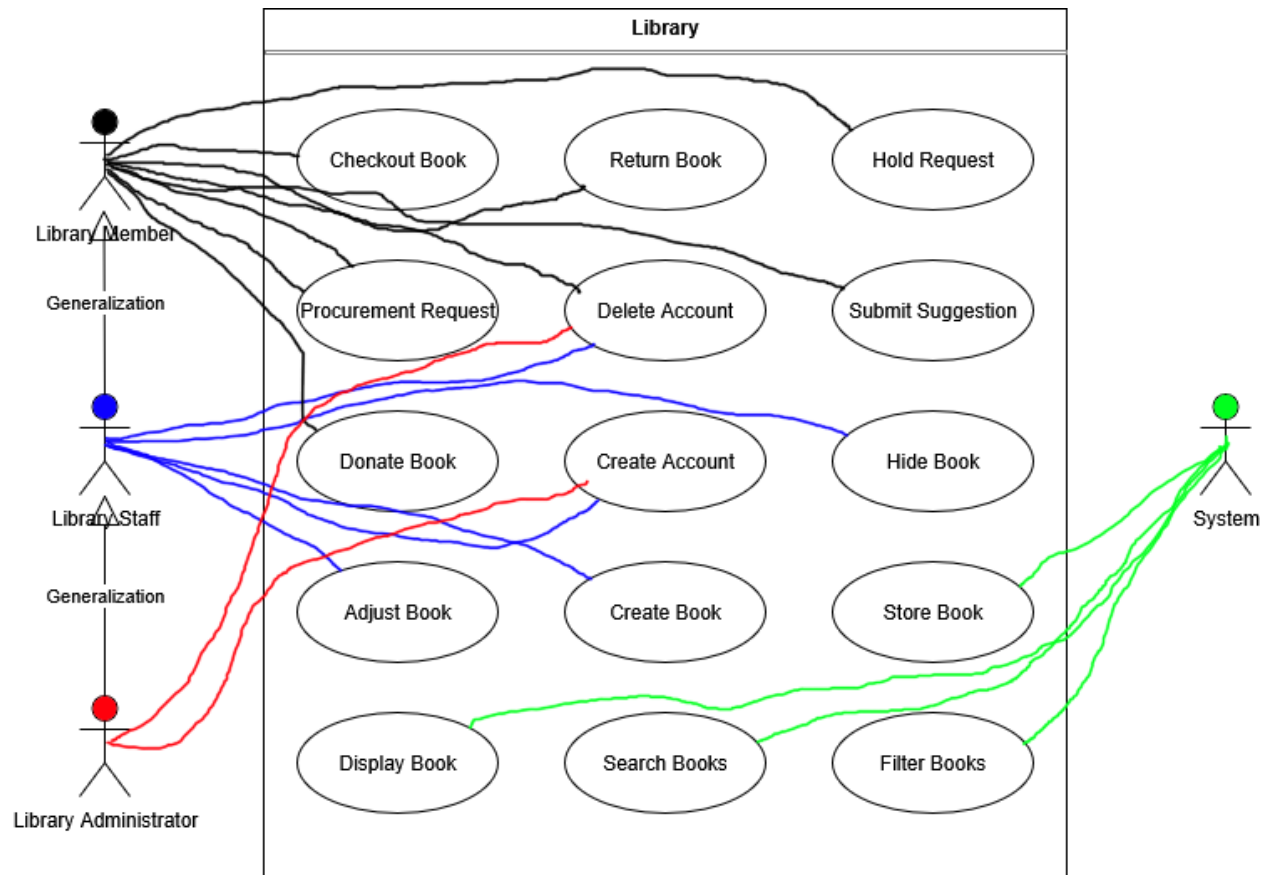
Library Administrator (Total: 4)

1. Create staff account: To convert a member account into a staff account. (2)
2. Delete account: To delete any level account (excepting administrator). (2)

System (Total: 14)

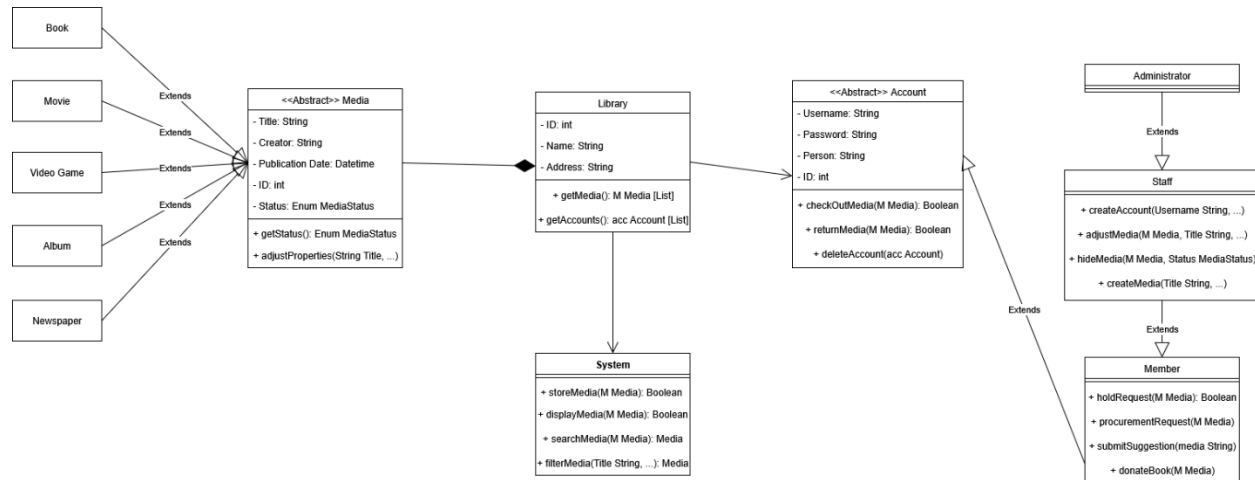
1. Store book: To store a new book in database. (3)
2. Display book: To display a new book from database. (3)
3. Search book: To search through database for a book matching criteria. (4)
4. Filter books: To filter database via matching criteria. (4)

Use Case Diagram



Class Specifications

Class Diagram



(This is a rather bare-bones class diagram. As I get further along with implementation, this is subject to change. I have a tendency to scope-creep all of my projects.)

Class Diagram Explanations

Media

Media is an abstract class since “media” isn’t itself a thing, and more of a category. It’s the template for things like books and movies. If this was a more in-depth project, each implementation of media would have its own attributes. (They still might; I haven’t made up my mind yet.) Media attributes are self-explanatory. Status refers to whether the media is “available”, “checked out”, “reserved”, or otherwise “unavailable”. It has a few basic methods right now, one to check the book’s status for the site and one to adjust properties via library staff.

Library

Library is the base for pretty much everything. It’s got a name, address, and ID since it’s planned to be able to request a piece of media from a networked library. You can get the list of Media and Accounts from the library. The accounts access the library, as well as the system.

System

The system has no attributes right now, but it does have several methods, mainly for use with the website (the user interface). The system can store media, display media, search for specific media, and filter media via provided criterion.

Account

Account is an abstract class like Media. There are a few basic attributes and methods that should be provided to all implementations. Unlike Media, there’s a hierarchy to things with Member at the bottom and Administrator at the top. Administrator can do anything that any account class below it can do.

Sequence and Activity Diagrams

Logan Young

System Sequence Diagrams

Hold Request

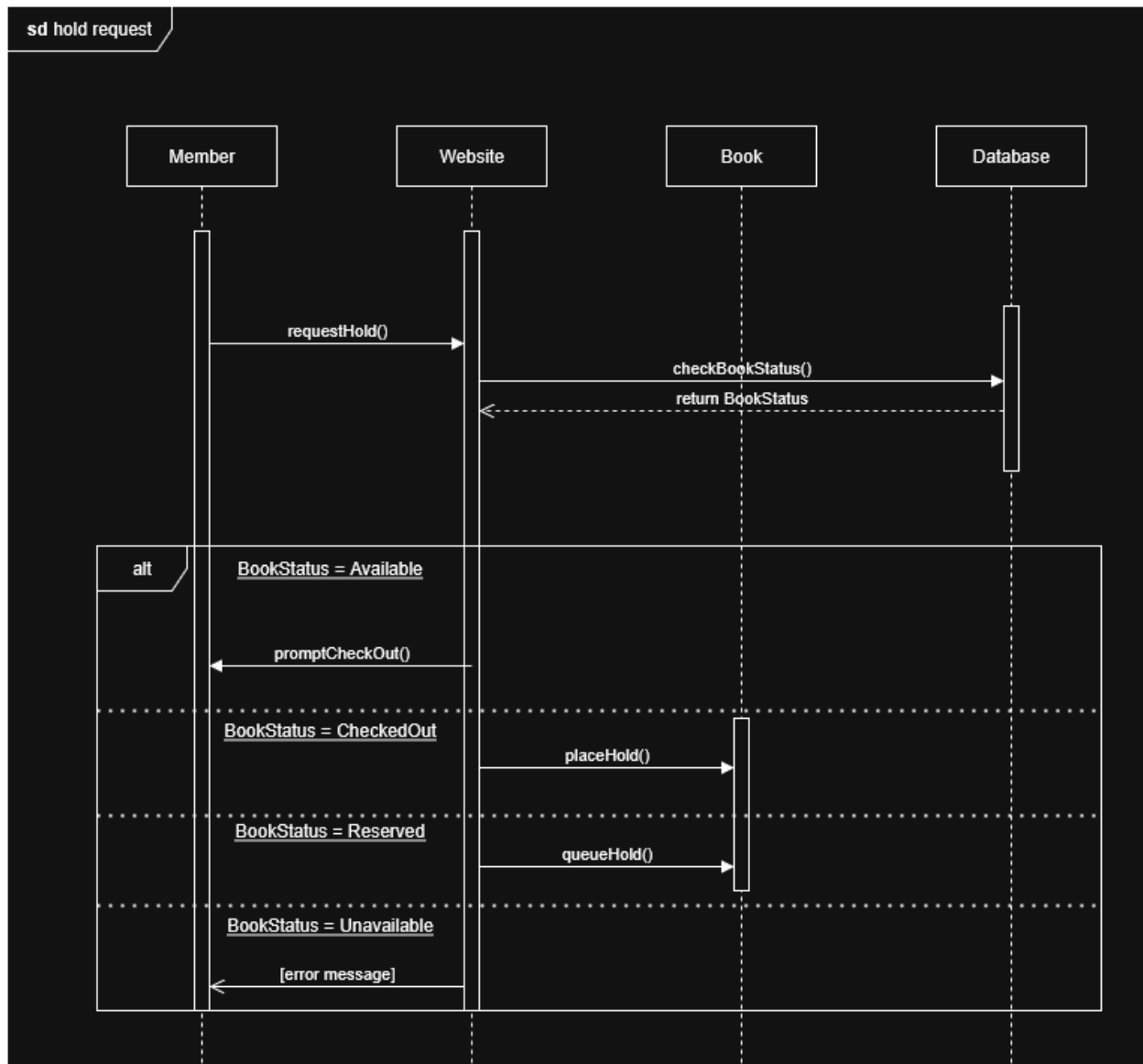
Actor and Objects

- Actor: **Member**
- Objects:
 - o Website
 - o Library Database
 - o Media

Process

1. A library member requests a hold on a book they want to borrow.
2. The system checks the book's current state: if it's available, if it's checked out, if it's reserved by another member, or if it's unavailable.
3. If the book is available: the member receives a prompt to check out the book now.
4. If the book is checked out with no other holds: the hold is placed successfully.
5. If the book is reserved by at least one other member: the member is added to the hold queue.
6. If the book is unavailable: the hold fails and an error message is sent.

Diagram



Adjust Book

Actors and Objects

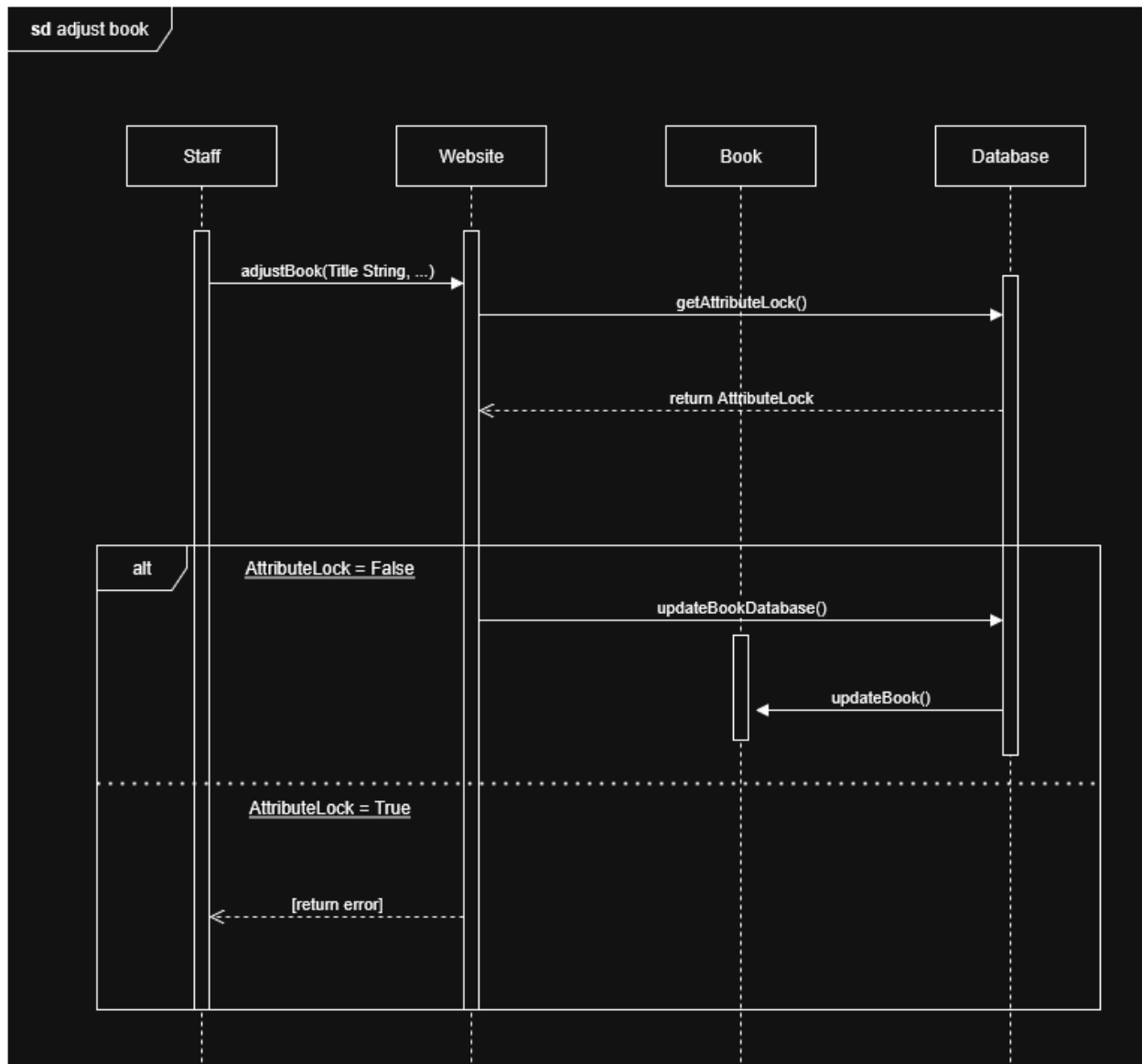
- Actor: **Staff**
- Objects:
 - o Website
 - o Library Database
 - o Media

Process

1. Staff selects a book from website to alter its attributes.

2. Staff inputs changed attributes.
3. Website checks attributeLock status.
4. If attributeLock is false: Website send changed attributes to Database, and Database stores them.
5. If attributeLock is true: Website return error and process is halted.

Diagram



Activity Diagrams

Create Staff Account

States

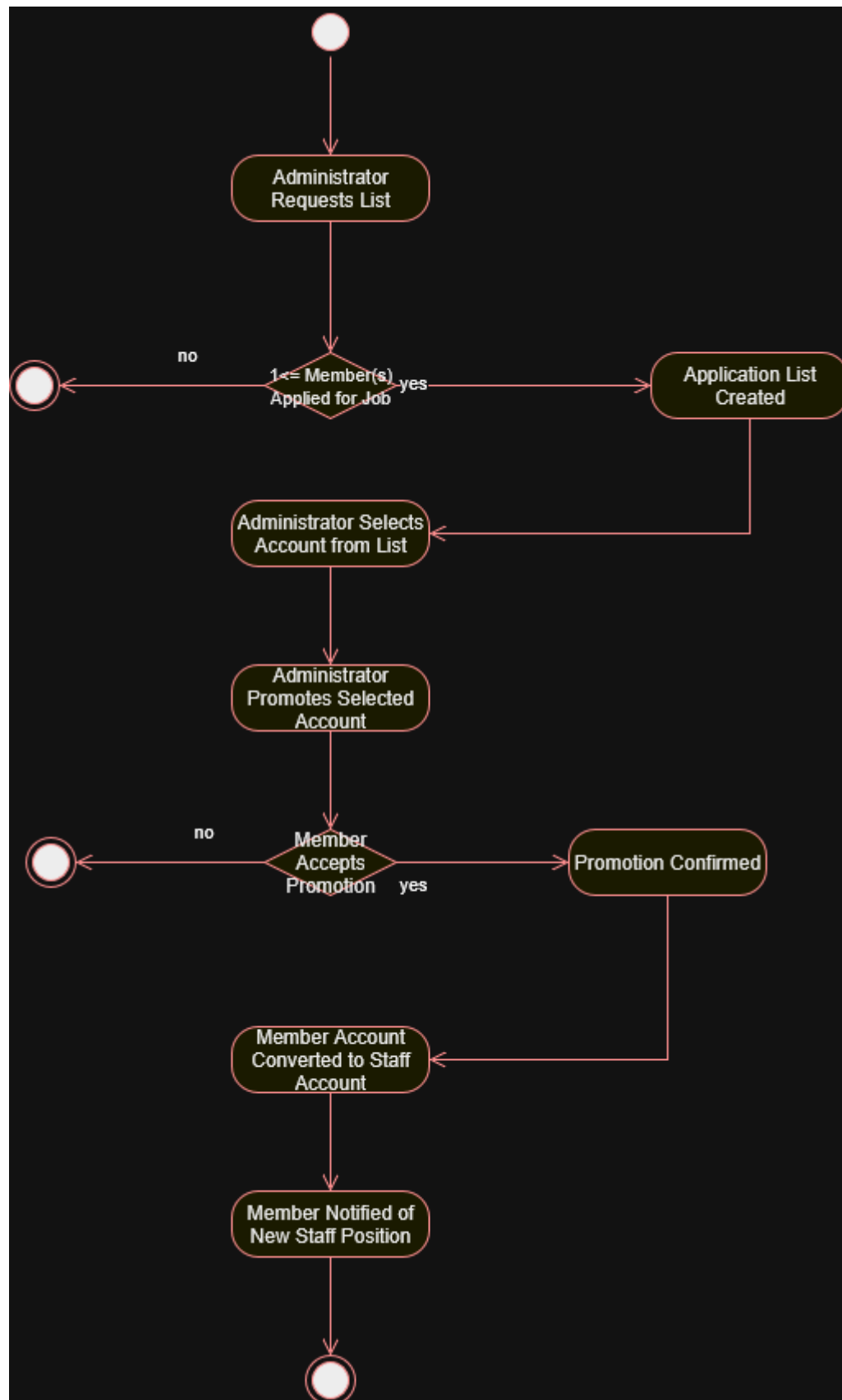
- Initial State: A member account is ready to be converted into a staff account.

- Final State:
 - No member accounts have sent in staff applications. Creation failed.
 - Member does not accept new staff position. Creation failed.
 - A member account becomes a staff account and gains access to its permissions.

Actions

1. The website retrieves a list of member accounts from the database that have applied for a staff position.
2. The Administrator selects a specific member account from the list.
3. The Administrator promotes the member account to a staff account.
4. The database converts the member account into a staff account.
5. The member is notified of their new position as a staff account.

Diagram



Filter Books

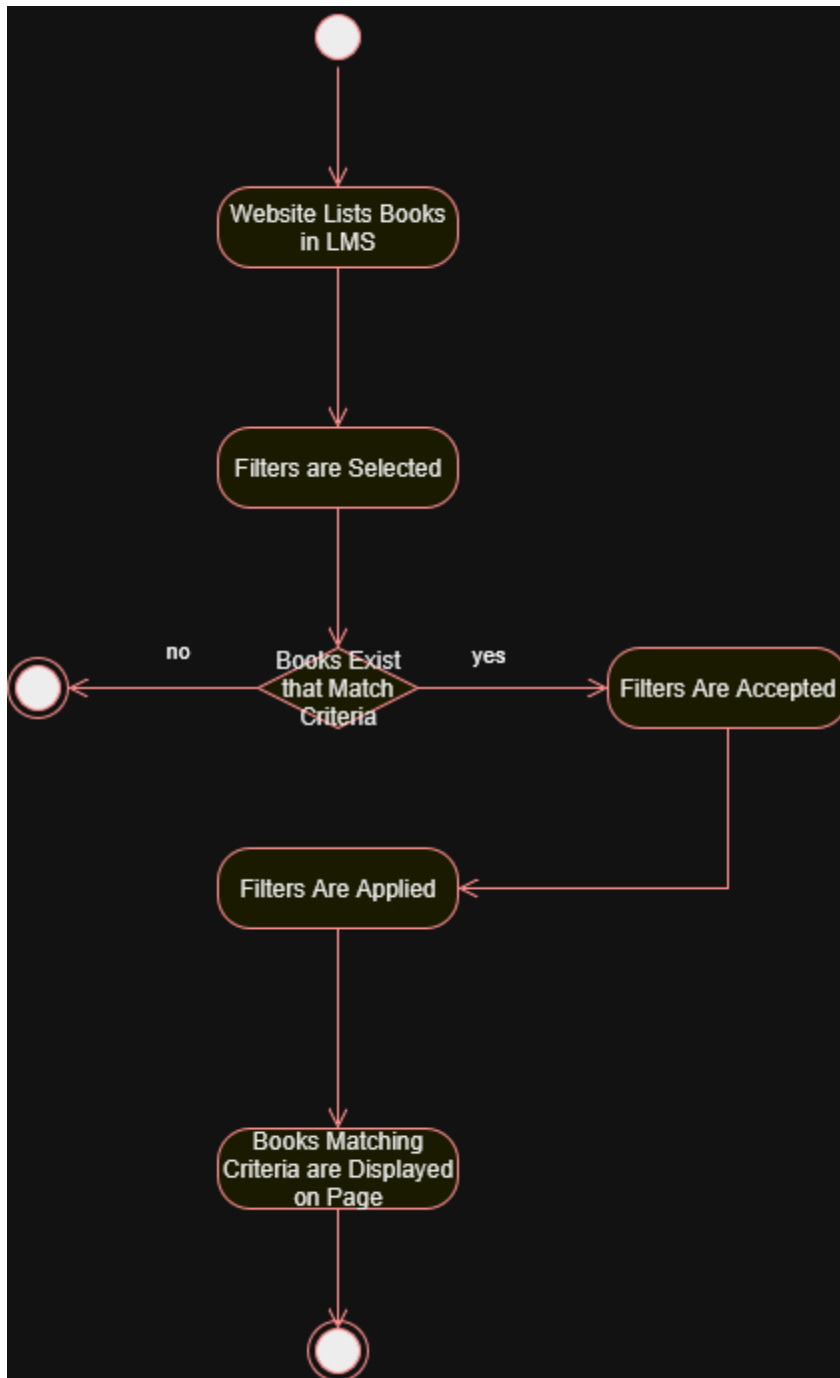
States

- Initial State: An account selects filters to apply to the book list on the website.
- Final State:
 - Books matching the filters selected are shown.
 - The page is empty to show that there are no books that match the criteria.

Actions

1. The website shows a list of books available at the library.
2. An account selects filters to apply to the list of books.
3. Filters are applied, selecting only the books that match the input criteria.
4. The books are shown on the website.

Diagram



User Interface Specification

Logan Young

Preliminary Design

Use Case 1

- A library member checks out a book.

1. Home page.

Library Management System	
Home	
<div>User View</div> <p>Welcome, User!</p> <p>Account Level: Member</p> <p>Books Currently Checked Out: 4, 8, 15</p> <div>Check Out</div> <p>Select books to check out in Book View.</p> <div>Check Out</div> <div>Return</div> <p>Select books to return in Book View.</p> <div>Return</div> <div>Suggest Book</div> <p>Author Name: <input type="text" value="Author Name"/></p> <p>Book Title: <input type="text" value="Book Title"/></p> <div>Suggest</div>	<div>Book View</div> <div>All BooksYour Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div>
Designed by Logan Young	

2. Select book.

Library Management System	
Home	
<div>User View</div> <p>Welcome, User!</p> <p>Account Level: Member</p> <p>Books Currently Checked Out: 4, 8, 15</p> <div>Check Out</div> <p>Select books to check out in Book View.</p> <div>Check Out</div> <div>Return</div> <p>Select books to return in Book View.</p> <div>Return</div> <div>Suggest Book</div> <p>Author Name: <input type="text" value="Author Name"/></p> <p>Book Title: <input type="text" value="Book Title"/></p> <div>Suggest</div>	<div>Book View</div> <div>All BooksYour Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div>
Designed by Logan Young	

3. Select check out button.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Check out selected book?

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All BooksYour Books

Book 1Book 2Book 3

Book 4Book 5Book 6

Book 7

Designed by Logan Young

4. Confirmation.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

Book 2Book 3

Book 5Book 6

Checkout Successful!

Confirm

Designed by Logan Young

Use Case 2

- A library member returns a book and then views their current checked-out books.

1. Home page.

Library Management System	
Home	
<div>User View</div> <div>Welcome, User!</div> <div>Account Level: Member</div> <div>Books Currently Checked Out: 4, 8, 15</div> <div>Check Out</div> <div>Select books to check out in Book View.</div> <div>Check Out</div> <div>Return</div> <div>Select books to return in Book View.</div> <div>Return</div> <div>Suggest Book</div> <div>Author Name: <input type="text" value="Author Name"/></div> <div>Book Title: <input type="text" value="Book Title"/></div> <div>Suggest</div>	<div>Book View</div> <div>All Books Your Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div>
Designed by Logan Young	

2. Select Your Books section.

Library Management System	
Home	
<div>User View</div> <div>Welcome, User!</div> <div>Account Level: Member</div> <div>Books Currently Checked Out: 4, 8, 15</div> <div>Check Out</div> <div>Select books to check out in Book View.</div> <div>Check Out</div> <div>Return</div> <div>Select books to return in Book View.</div> <div>Return</div> <div>Suggest Book</div> <div>Author Name: <input type="text" value="Author Name"/></div> <div>Book Title: <input type="text" value="Book Title"/></div> <div>Suggest</div>	<div>Book View</div> <div>All Books Your Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div>
Designed by Logan Young	

3. Select a book to return.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books Your Books

Book 1 Book 2 Book 3

Designed by Logan Young

4. Click return button.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Return book?

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books Your Books

Book 1 Book 2 Book 3

Designed by Logan Young

5. Confirmation.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

Return Successful!

Book 2

Book 3

Confirm

Designed by Logan Young

6. Return to book view.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books Your Books

Book 1

Book 2

Designed by Logan Young

Use Case 3

- A user filters books by selected criteria.

1. Home page.

Library Management System	
Home	
<div>User View</div> <div>Welcome, User!</div> <div>Account Level: Member</div> <div>Books Currently Checked Out: 4, 8, 15</div> <div>Check Out</div> <div>Select books to check out in Book View.</div> <div>Check Out</div> <div>Return</div> <div>Select books to return in Book View.</div> <div>Return</div> <div>Suggest Book</div> <div>Author Name: <input type="text" value="Author Name"/></div> <div>Book Title: <input type="text" value="Book Title"/></div> <div>Suggest</div>	<div>Book View</div> <div>All Books Your Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div>
Designed by Logan Young	

2. Select filter menu (will appear as expandable menu below Books sections).

Library Management System	
Home	
<div>User View</div> <div>Welcome, User!</div> <div>Account Level: Member</div> <div>Books Currently Checked Out: 4, 8, 15</div> <div>Check Out</div> <div>Select books to check out in Book View.</div> <div>Check Out</div> <div>Return</div> <div>Select books to return in Book View.</div> <div>Return</div> <div>Suggest Book</div> <div>Author Name: <input type="text" value="Author Name"/></div> <div>Book Title: <input type="text" value="Book Title"/></div> <div>Suggest</div>	<div>Book View</div> <div>All Books Your Books</div> <div>Book 1Book 2Book 3</div> <div>Book 4Book 5Book 6</div> <div>Book 7</div> <div>Filter List</div> <div>Title contains: <input type="text"/></div> <div>Author contains: <input type="text"/></div> <div>Publication date: XX-XX-XXXX</div> <div><input type="checkbox"/> Has digital copy<input type="checkbox"/> Has audiobook</div> <div><input type="checkbox"/> Has physical copy<input type="checkbox"/> Is hardcover</div>
Designed by Logan Young	

3. Choose filters.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books

Your Books

Book 1

Book 2

Book 3

Book 4

Book 5

Book 6

Book 7

Filter List

Title contains:

Author contains:

Publication date: XX-XX-XXXX

☐ Has digital copy

☐ Has audiobook

☒ Has physical copy

☐ Is hardcover

Apply

Designed by Logan Young

4. Click apply.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books

Your Books

Book 1

Book 2

Book 3

Book 4

Book 5

Book 6

Book 7

Filter List

Title contains:

Author contains:

Publication date: XX-XX-XXXX

☐ Has digital copy

☐ Has audiobook

☒ Has physical copy

☐ Is hardcover

Apply

Designed by Logan Young

5. Filtered books.

Library Management System

Home

User View

Welcome, User!

Account Level: Member

Books Currently Checked Out: 4, 8, 15

Check Out

Select books to check out in Book View.

Check Out

Return

Select books to return in Book View.

Return

Suggest Book

Author Name:

Book Title:

Suggest

Book View

All Books Your Books

Book 2

Book 5

Filter List

Title contains:

Author contains:

Publication date: XX-XX-XXXX

☐ Has digital copy

☒ Has physical copy

☐ Has audiobook

☐ Is hardcover

Apply

Designed by Logan Young

Use Case 4

- A staff member adds a book.

1. Home page.

Library Management System

Home

User View

Welcome, User!

Account Level: Staff

Books Currently Checked Out: 4, 8, 15

Member Staff

Create Book

Open Form

...

...

Book View

All Books Your Books

Book 1

Book 2

Book 3

Book 4

Book 5

Book 6

Book 7

Designed by Logan Young

2. Select Open Form.

Library Management System

Home

User View

Welcome, User!

Account Level: Staff

Books Currently Checked Out: 4, 8, 15

Member **Staff**

Create Book

Open Form

...

...

Book View

All Books Your Books

Book 1Book 2Book 3

Book 4Book 5Book 6

Book 7

Designed by Logan Young

3. Fill out form for book creation.

Library Management System

Home

User View

Welcome, User!

Account Level: Staff

Books Currently Checked Out: 4, 8, 15

Member **Staff**

Create Book

Open Form

...

...

Book Creation Form

Title: Stars and Space 3: More Space

Author: Jessica Spacey

Publication Date: 01-23-1997

Format: Physical, Softcover

Create

Book View

Book 2Book 3

Book 5Book 6

Designed by Logan Young

4. Select Create.

The screenshot displays the 'Library Management System' Home page. The layout is divided into three main sections: 'User View' on the left, a central content area, and 'Book View' on the right. The 'User View' section includes a welcome message, account level ('Staff'), checked-out books, and a 'Create Book' button labeled 'Open Form'. The central content area features a 'Book Creation Form' with fields for Title ('Stars and Space 3: More Space'), Author ('Jessica Spacey'), Publication Date ('01-23-1997'), and Format ('Physical, Softcover'). A 'Create' button is located at the bottom of this form. The 'Book View' section shows a list of books, including 'Book 2', 'Book 3', 'Book 5', and 'Book 6'. The footer indicates the system was 'Designed by Logan Young'.

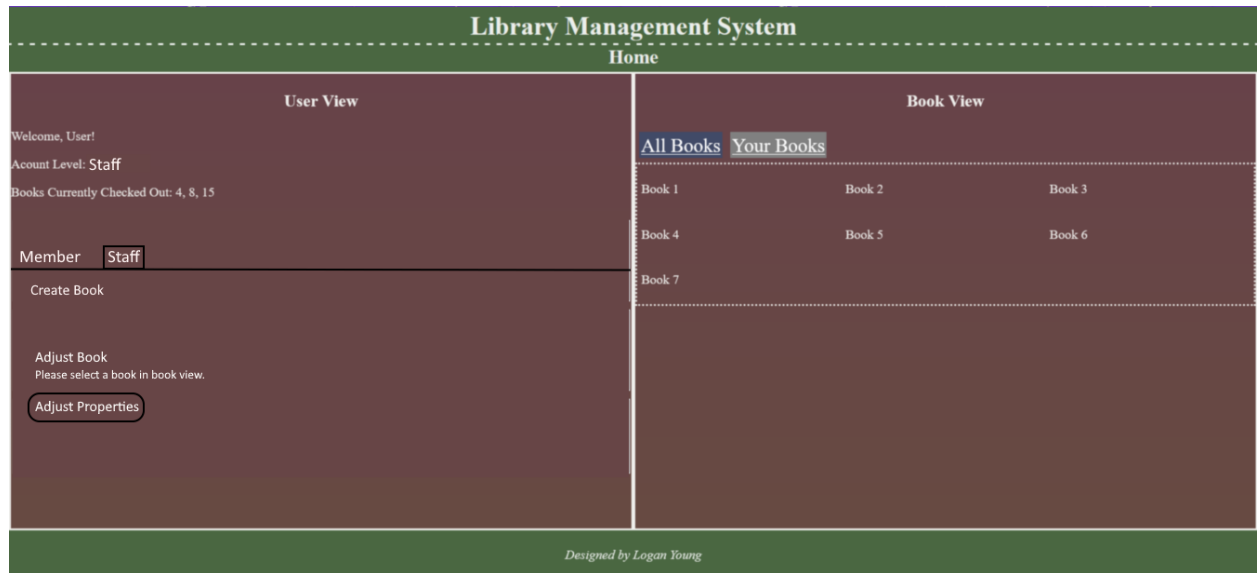
5. Confirmation.

The screenshot displays the 'Library Management System' Home page after a successful book creation. The layout remains the same as the previous screenshot, but the central content area now shows a 'Book Creation Successfull' confirmation message with a 'Confirm' button at the bottom. The 'User View' and 'Book View' sections are unchanged, showing the same user information and book list. The footer still indicates the system was 'Designed by Logan Young'.

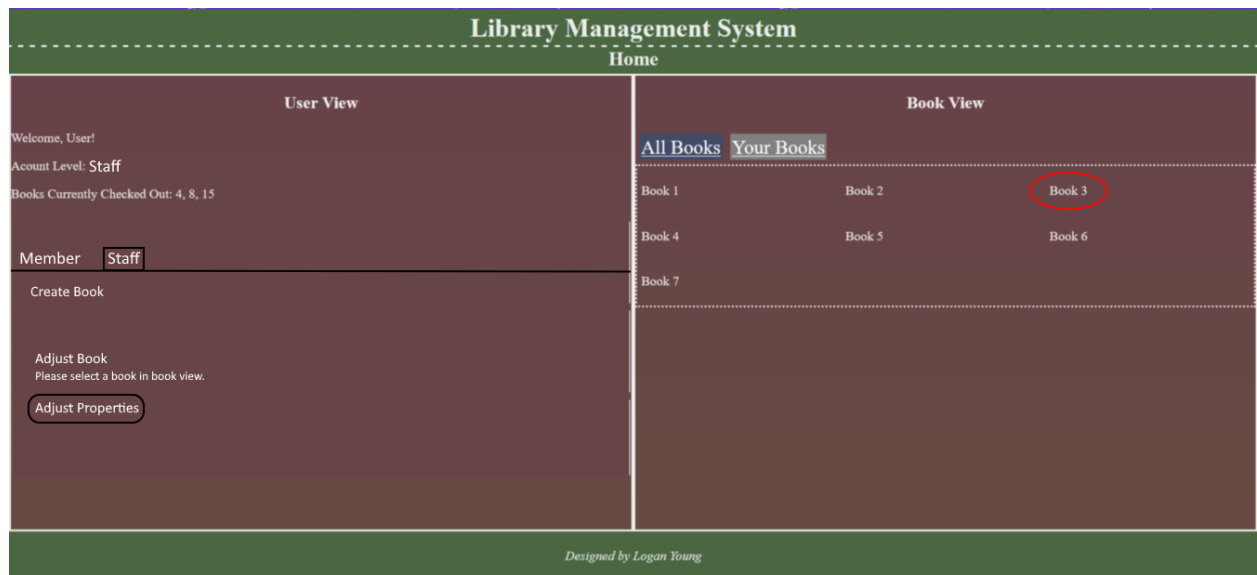
Use Case 5

- A staff member adjusts a book's properties.

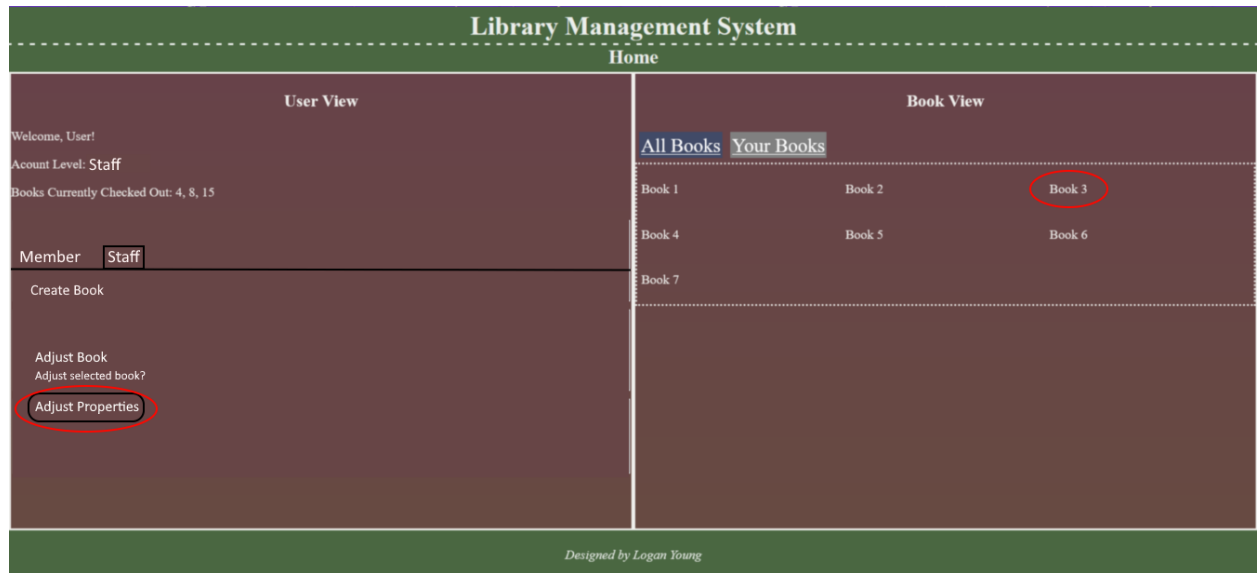
1. Home page.



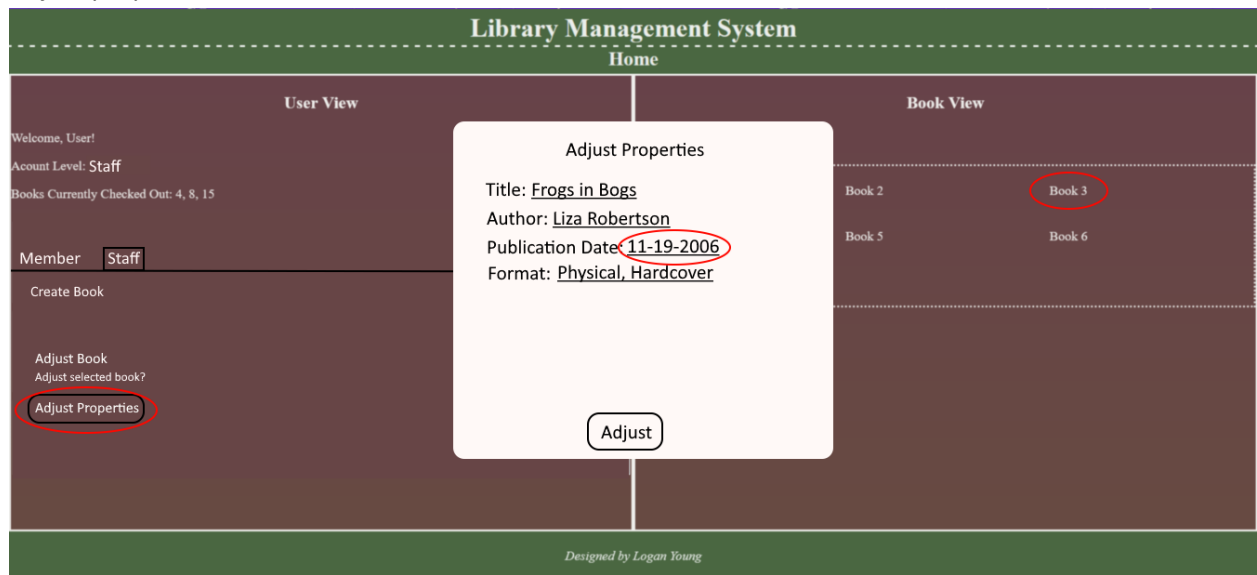
2. Select book.



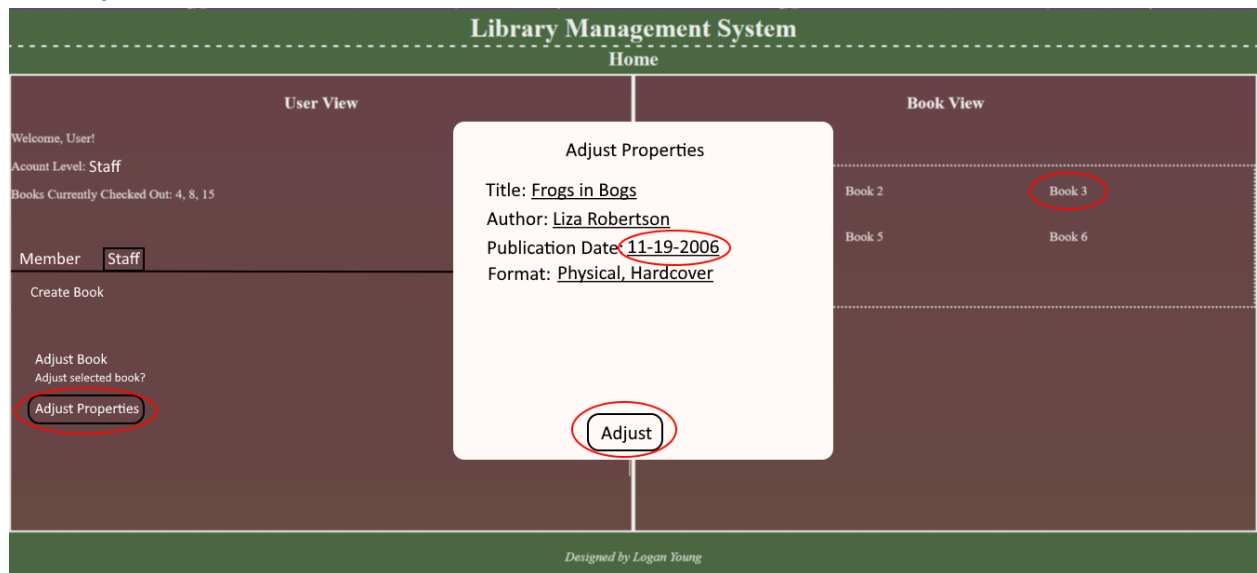
3. Click Adjust Properties.



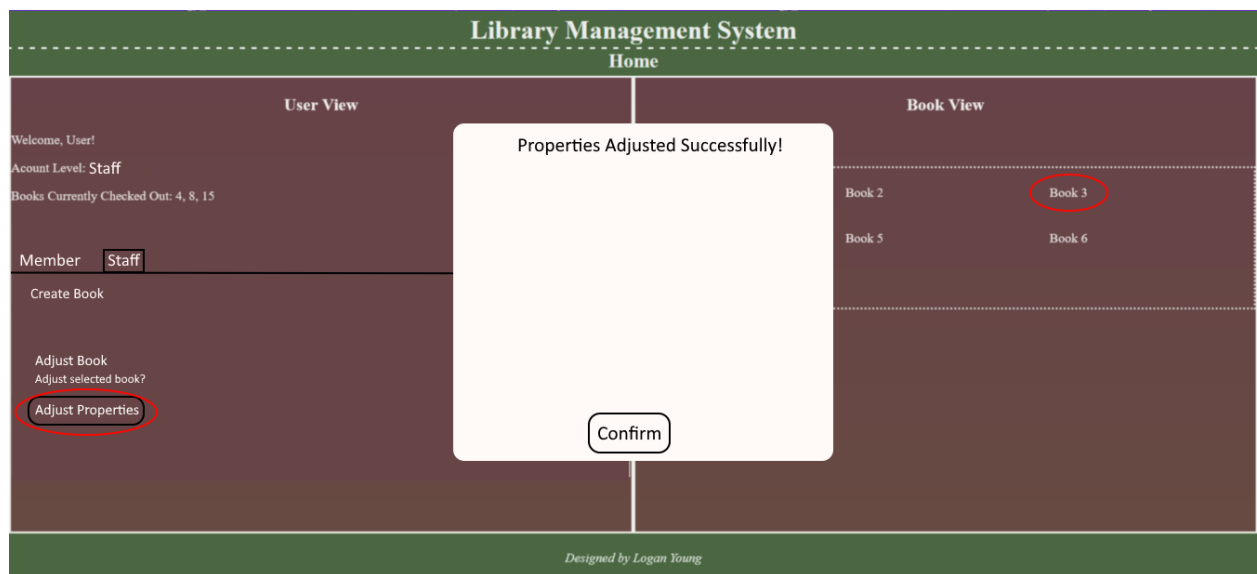
4. Adjust properties in form.



- Click Adjust.



- Confirmation.



User Effort Estimation

Usage Scenario	Navigation	Clicks	Keystrokes
Member checks out a book	Home -> Book View -> User View -> Confirmation	3	0
Member returns a book and views current checked out books	Home -> Book View-> Confirmation -> Home -> Book View	5	0
Staff adjusts a book's properties	Home -> Book View -> User View -> Popup Form -> Confirmation	>=5	100>=x>=1

Staff adds a book	Home -> User View -> Popup Form -> Confirmation	>=4	100>=x>=10
User filters books	Home -> Book View	>=3	>=0

Additional Notes

- Please excuse the poor photoshopping done on top of my website screenshots.
- Like the sample example, each use case assumes that the user has already signed in. Below is an early work-in-progress example of the sign in form.

Library Management System

Sign In

Sign In Form

Account Type: Member ▼

Email: YourName@email.com

Password:

Submit

Designed by Logan Young

- The base home page and sign in page are currently being worked on, but they are fully coded in HTML and CSS. JavaScript and node.JS functionality are coming soon. Might support JavaScript frontend with jQuery as I progress.
- The appearance of the site and UI placement are subject to change, but the general functionality will remain the same.

Traceability Matrix and Updates

Logan Young

Updated Functional Requirements

No.	Priority Weight (1-5; 1: lowest)	Description
REQ-1	5	The LMS should be able to store and display books on the website.
REQ-2	5	Library members should be able to use the LMS to check out and return books.
REQ-3	3	Library members should be able to use the LMS to submit a hold request for a maximum of three books.
REQ-4	4	Library staff should be able to use the LMS to mark a book as unavailable.
REQ-5	3	Library members should be able to delete their account from the LMS.
REQ-6	4	Library staff should be able to delete a member's account using the LMS.
REQ-7	5	Library staff should be able to create a new member account for a user.
REQ-8	5	Library staff should be able to adjust a book's attributes if incorrect.
REQ-9	5	Library staff should be able to create a new entry in the LMS for a newly received book.
REQ-10	2	Library members should be able to request a book from a different library in the same network.
REQ-11	4	Library administrator should be able to convert a new user into staff.
REQ-12	4	Library administrator should be able to convert staff into members or delete accounts entirely.
REQ-13	2	Library members should be able to submit a suggestion for new books for the library.
REQ-14	1	Library members should be able to create a new donated book entry to be approved by staff.
REQ-15	5	LMS users should be able to easily search for a specific book using various filters and sorting techniques.
REQ-16	5	LMS users should be able to log in to personal accounts using email and passwords from the website and log out.
REQ-17	2	LMS users should be able to activate 2-factor authentication for their accounts from user settings.
REQ-18	3	Library administrators should be able to completely delete a book from the database.

Updated Use Cases

No.	Name	Description
UC-1	Log In	To let a user log in to their account.

RE Q3	3	X					X												
RE Q4	4	X														X			
RE Q5	3	X							X										
RE Q6	4	X							X										
RE Q7	5	X								X									
RE Q8	5	X													X				
RE Q9	5	X												X					
RE Q1 0	2	X						X											
RE Q1 1	4	X									X								
RE Q1 2	4	X							X	X	X								
RE Q1 3	2	X										X							
RE Q1 4	1	X											X						
RE Q1 5	5	X																X	X
RE Q1 6	5	X	X																
RE Q1 7	2	X		X															
RE Q 18	3																X		
Ma x PW		5	5	2	5	5	3	2	4	5	4	2	1	5	5	4	3	5	5
Tot al PW		5 9	5	2	5	5	3	2	1 1	9	8	2	1	5	5	4	3	5	5

Development Plan

- Weeks 1-2
 - Conceptualize and complete the system proposal.
- Weeks 3-4
 - Determine connection between frontend, backend, and database and start construction.
- Weeks 5-7
 - Build database to organize books and their attributes, build frontend webpage for users to interact with, build backend to connect the two.
- Week 8 (**We are here.**)
 - Test system with sample entries and actions and record demo for midterm.
- Weeks 9-11
 - Improve system with feedback from professor. Implement suggested features and bug fixes. Implement rest of features left out from demo.
- Weeks 12-14
 - Write test cases and optionally add some cool last-minute features.
- Week 15
 - Test final system and record demo for final.