

Library Management System  
System Requirements Documentation  
Rachel Watts  
11/11/2025

Table of Contents:

Customer Problem Statement

System Requirements

Functional Requirement Specification

System Sequence Diagram

Activity Diagram

User Interface Specification

Project Plan

References

#### Customer Problem Statement:

Libraries are a place where people of all ages can go to learn, research, study, or just have fun. However, they have not kept up with the increase of technological advances in our society today. Many libraries still use paper or written records and spreadsheets and do not have a specific system to keep track of their large inventory of books. This allows for errors in records and takes a lot of time to manage. My system will make it easier for library members and guests to know if a book is available without having to wait in line or ask for help. It will also help librarians save time helping members and be able to easily keep track of if books are available or borrowed and when they will be returned. My system will allow members to easily search and see if a book is available while also helping librarians easily track if a book is borrowed or not.

#### System Requirements:

No.	Priority Weight	Description
Req1	High	Library members can search for books
Req2	High	System shows which books are borrowed/returned and who borrowed/returned them
Req3	High	Librarians can add/remove books from system
Req4	High	Members can borrow/return books from system
Req5	Medium	Librarians can edit book information
Req6	Medium	Members can see which books they are borrowing right now
Req7	Medium	Librarians can see what books are overdue
Req8	Low	Librarians/members can search for book by title/author

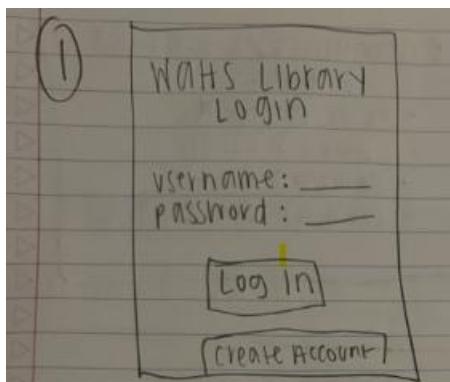
#### Nonfunctional Requirements:

Functionality	High	System must be able to search for books and see
---------------	------	---

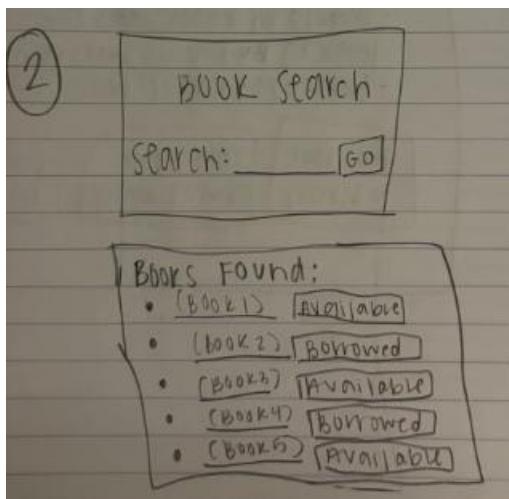
		which are borrowed/returned
Usability	High	System should be simple and understood by librarians and members
Reliability	High	System must have reliable/secure data on books
Performance	Medium	System should show results effectively and not lag
Supportability	Medium	System should work on PC/laptop

#### User Interface Requirements:

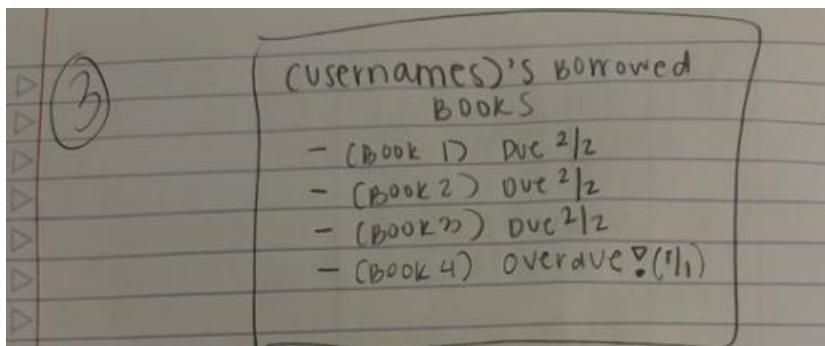
- Login Screen, high priority, members and librarians should be able to log into the system to search for books



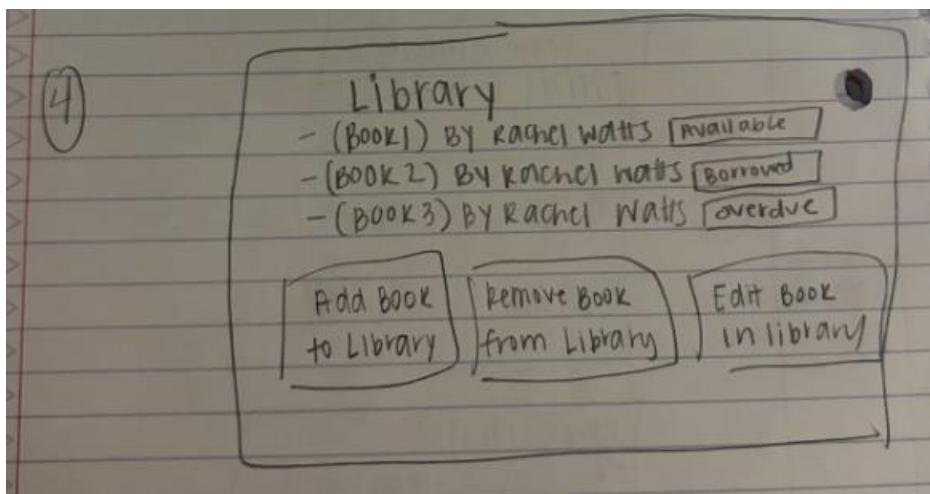
- Search Page, high priority, members/librarians should be able to search for a book by title or author



- Borrow/Return Page, high priority, members should see what books they have borrowed and when they need to return them



- Dashboard, high priority, librarians should be able to edit books available



#### Use Cases:

(2 points is = one engineer's one day of workload)

Library Member (total: 12 points)

- Search for books: Members search for books by title/author (2)
- Borrow a book: Member borrows a book from available books, and the status of the book is updated in the system (2)
- Return a book: Member returns book, status of book is updated in system (2)
- View borrowed books: Member views list of books they are currently borrowing and when they are due (2)
- Login/Logout of system: Member must login to access system (2)
- Check availability: Member can see if a book is available or borrowed (2)

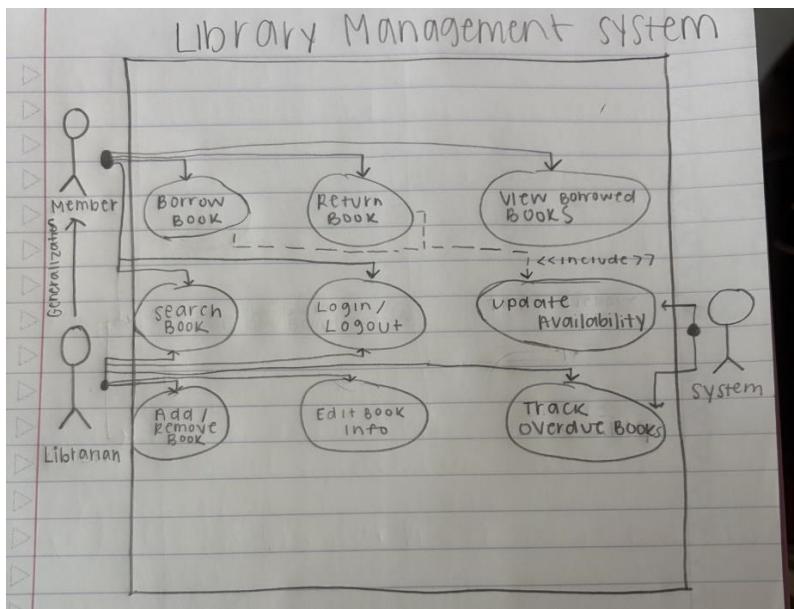
Librarian (total: 12):

- Add book: Add new book to system (2)
- Remove book: Remove book from system (2)
- Edit book information: Update title/author (2)
- Search borrowed/returned books: View which members have borrowed/returned books (2)
- Search overdue books: Look up what books are overdue and who is borrowing them (2)
- Login/Logout: Access for librarians to use system (2)

System (total: 8):

- Update availability: Automatically update the status of a book when it is borrowed or returned (2)
- Authenticate users: Manage login for members and librarians (2)
- Store records: Secure storage of data in MySQL (2)
- Overdue alerts: Notify members and librarians of overdue books (2)

**Use Case Diagram:**



### Class Diagram:

**Book:** Each book must have an ID, title, author, availability status, and due date. Books can be borrowed, returned, or updated.

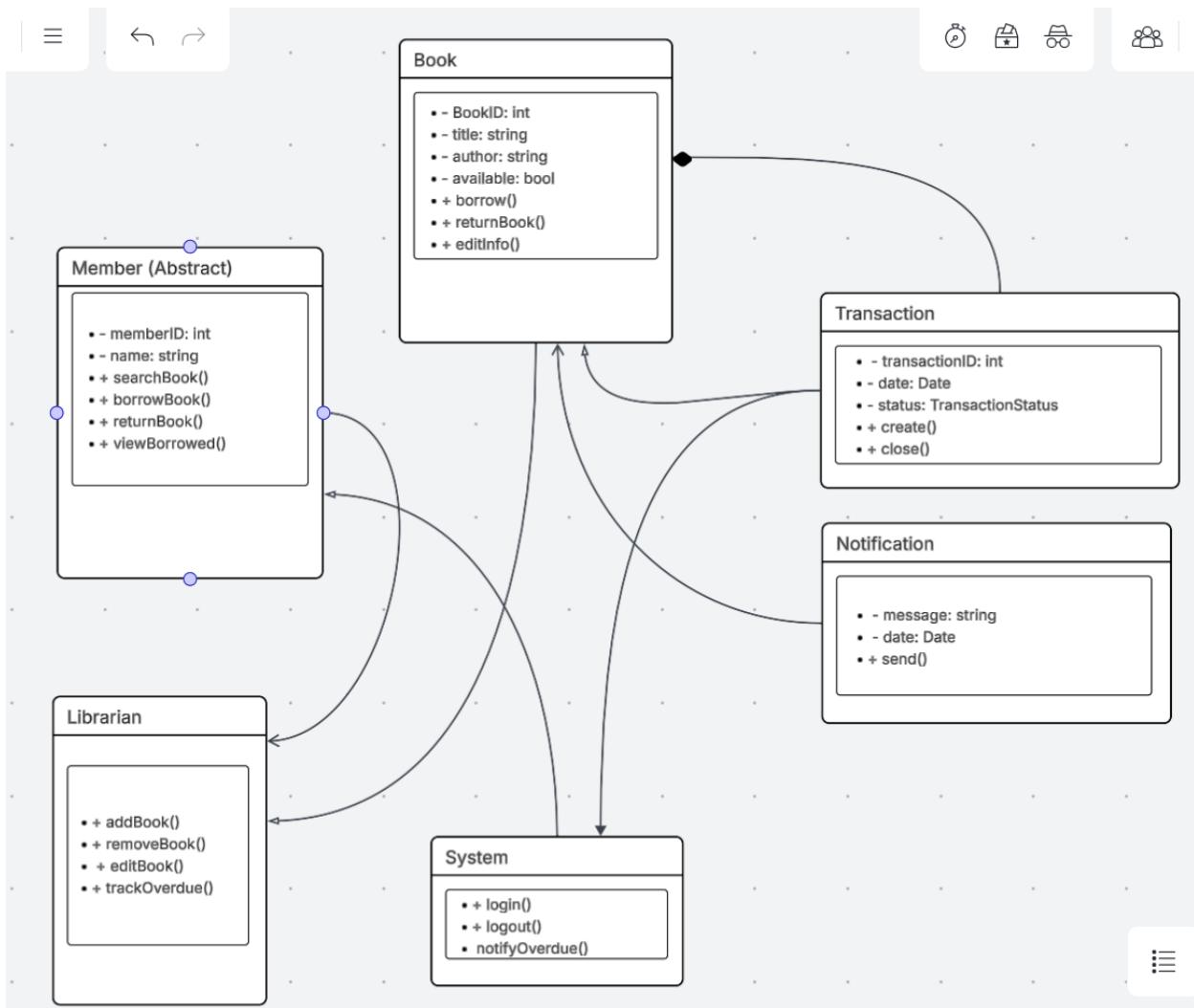
**Member:** Member is abstract class because there are different types of members with different roles. Members can search for books, borrow/return, and view what they currently borrow.

**Librarian (inherited from member):** Librarian is a specialized member. They can do everything members can, plus add, remove, and edit book information and track overdue books.

**Transaction:** Action of borrowing or returning a book. Links book with members and tracks transactions.

**Notification:** System alerts members when a book is overdue.

**System:** Controls application. Manages members, books, notifications.



## System Sequence Diagrams:

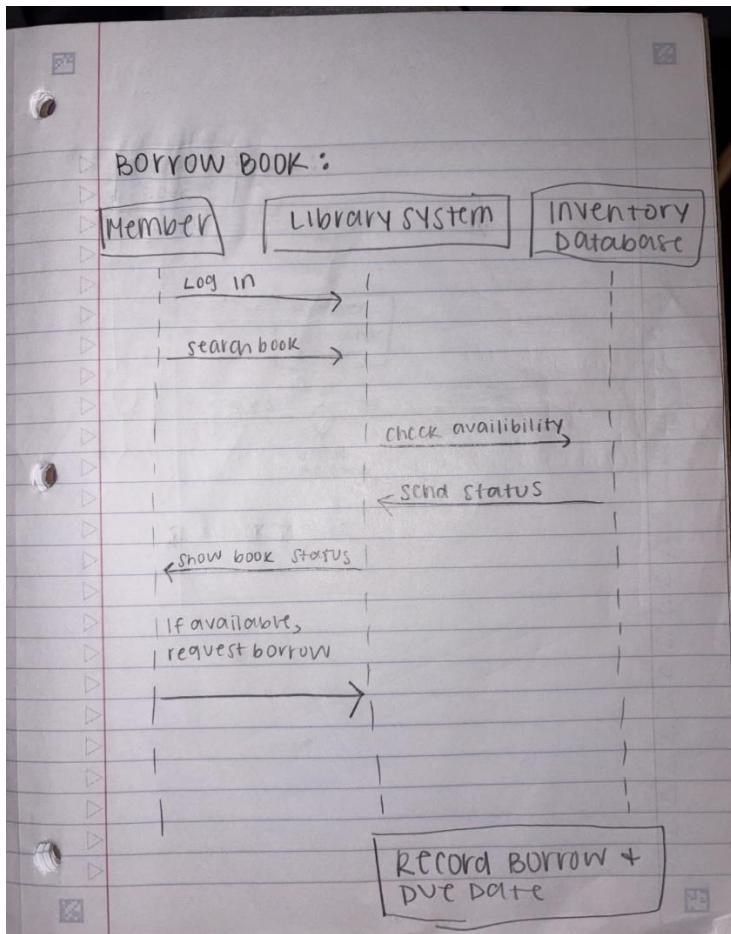
### Borrow a Book

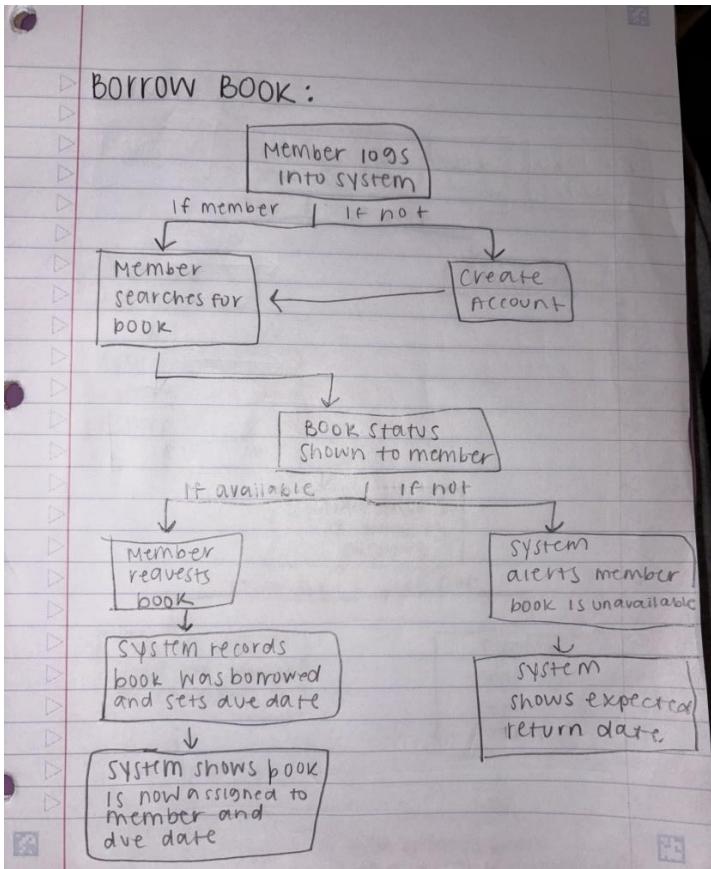
- Actor: Member
- Objects: Library System, Inventory Database

Steps:

1. Member logs into system
2. Member searches for book
3. System checks availability from database
4. System shows status of book to member

5. If status is available 1. Member requests to borrow book 2. System records that book was borrowed and sets due date 3. System shows success and due date to member
6. If status is unavailable 1. Alerts member book is unavailable and says when it is expected to be returned



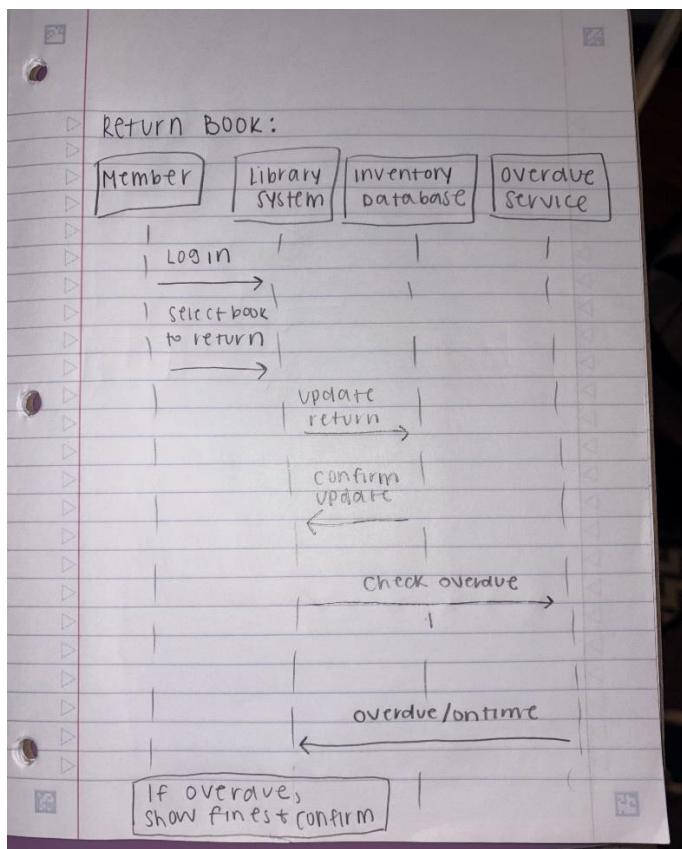


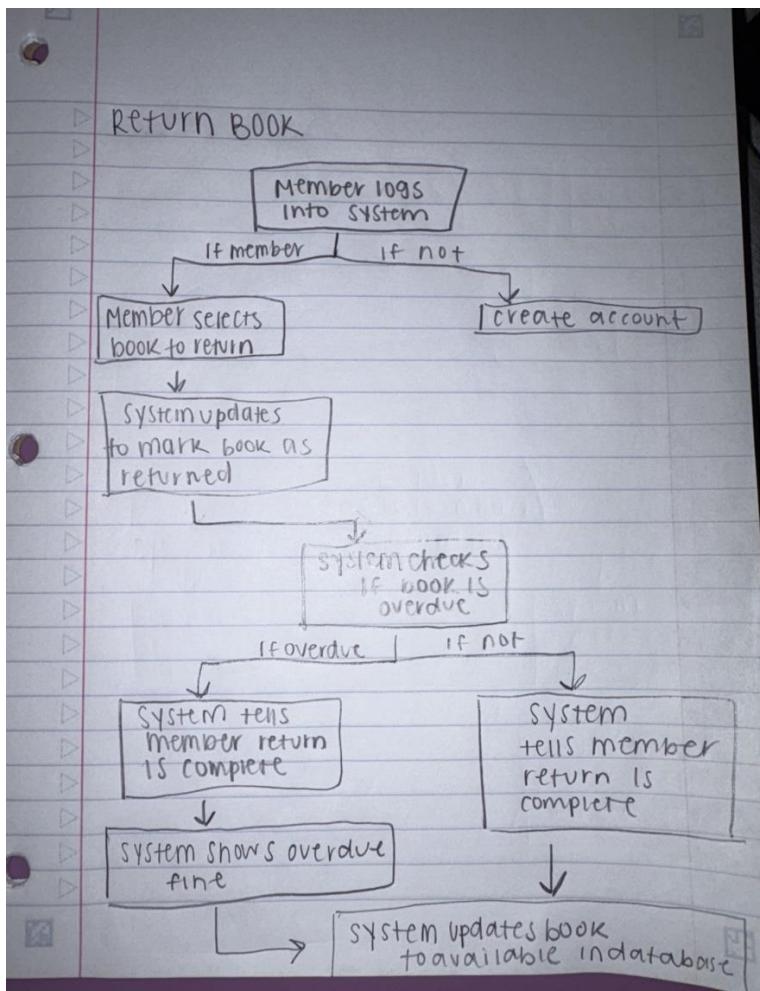
## Return a Book

- Actor: Member
- Objects: System Inventory Database, Overdue Service

Steps:

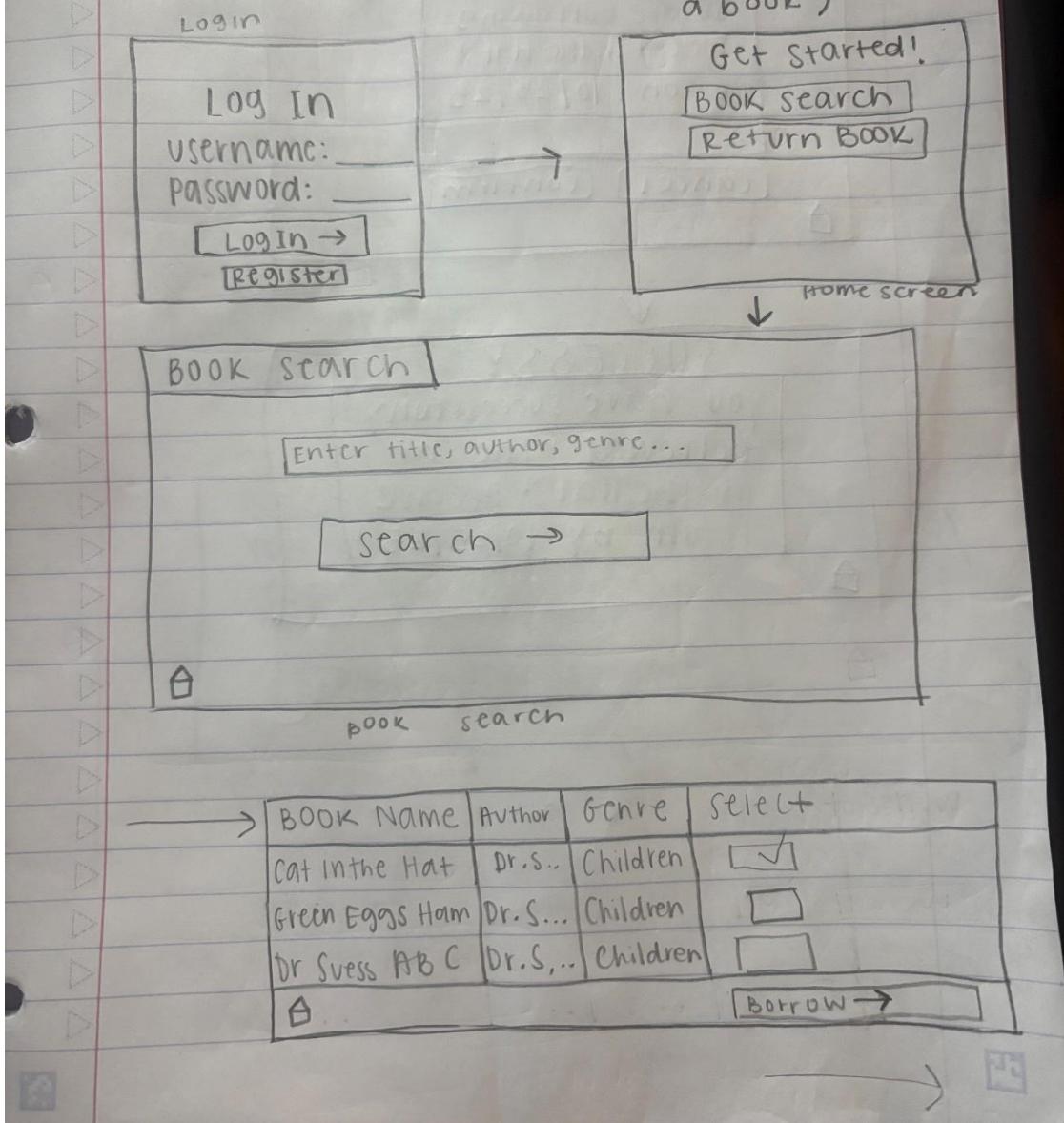
1. Member logs in
2. Member selects book to return
3. System updates database to mark book as returned
4. Database confirms
5. System checks if is overdue
6. If overdue: 1. System tells member return is complete and shows overdue fines
7. If not overdue: 1. System tells member return is complete
8. System updates book to available in database

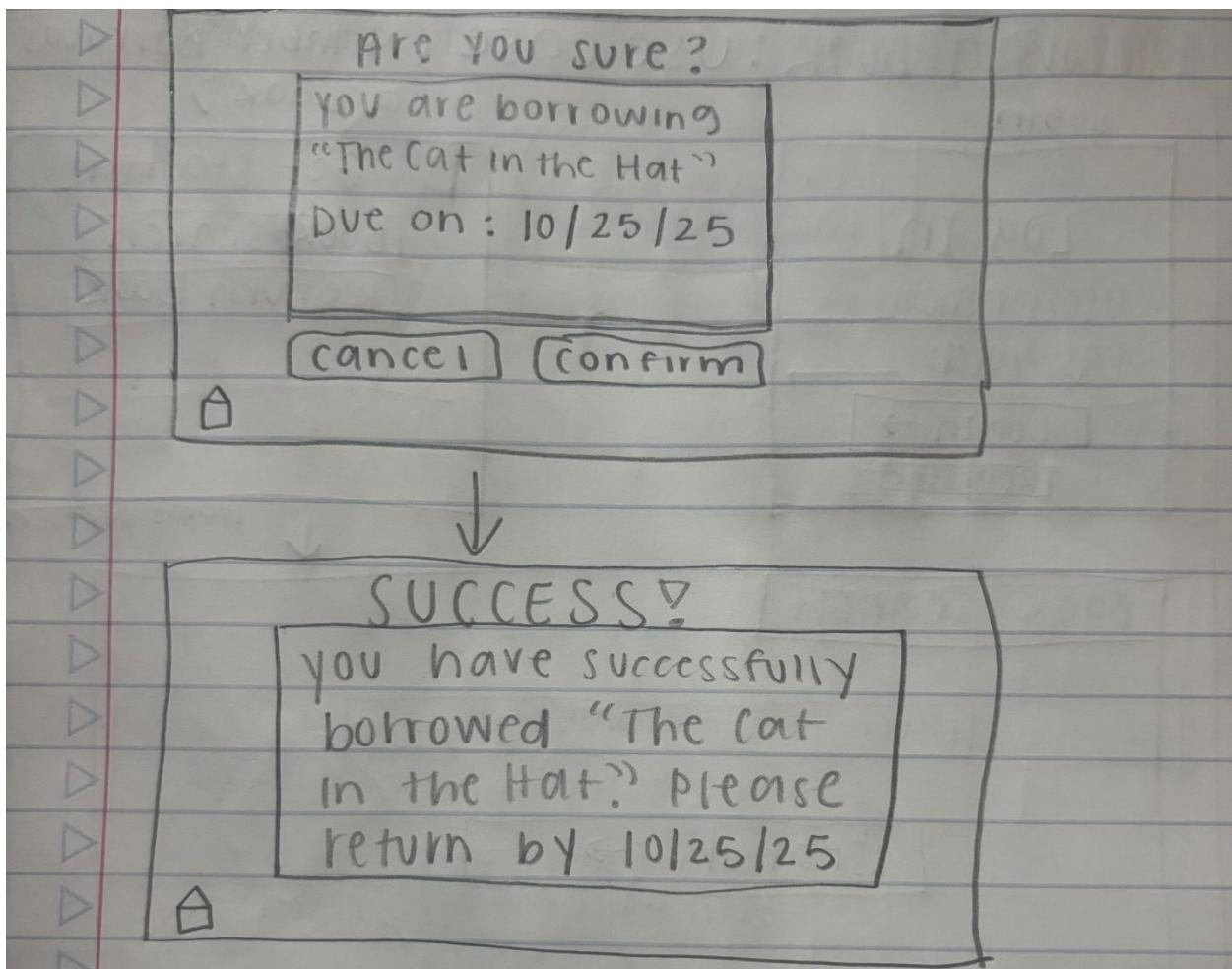




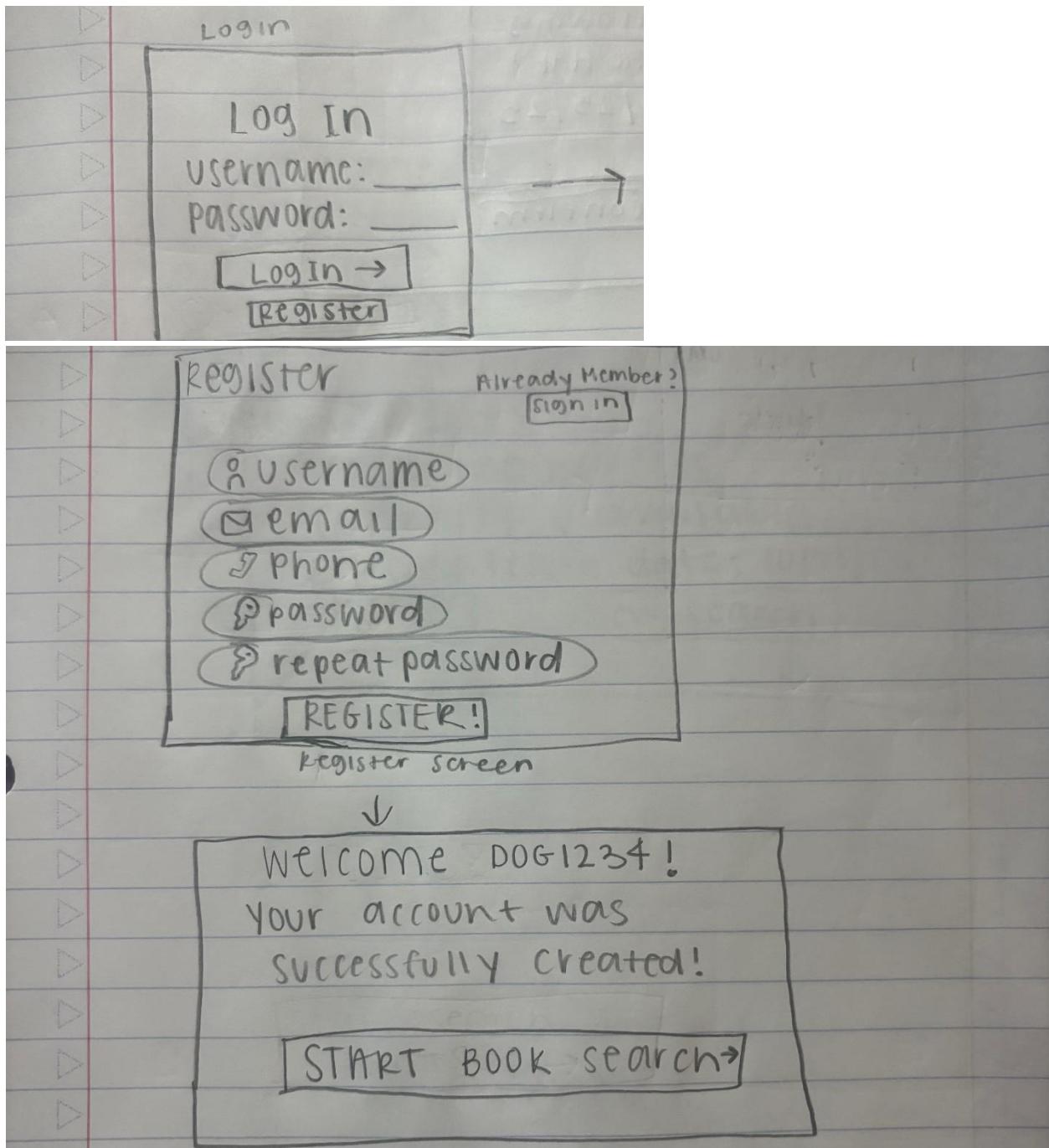
Use Case 1: Member borrows a book

BORROW A BOOK : USC CASE (Member borrows a book)

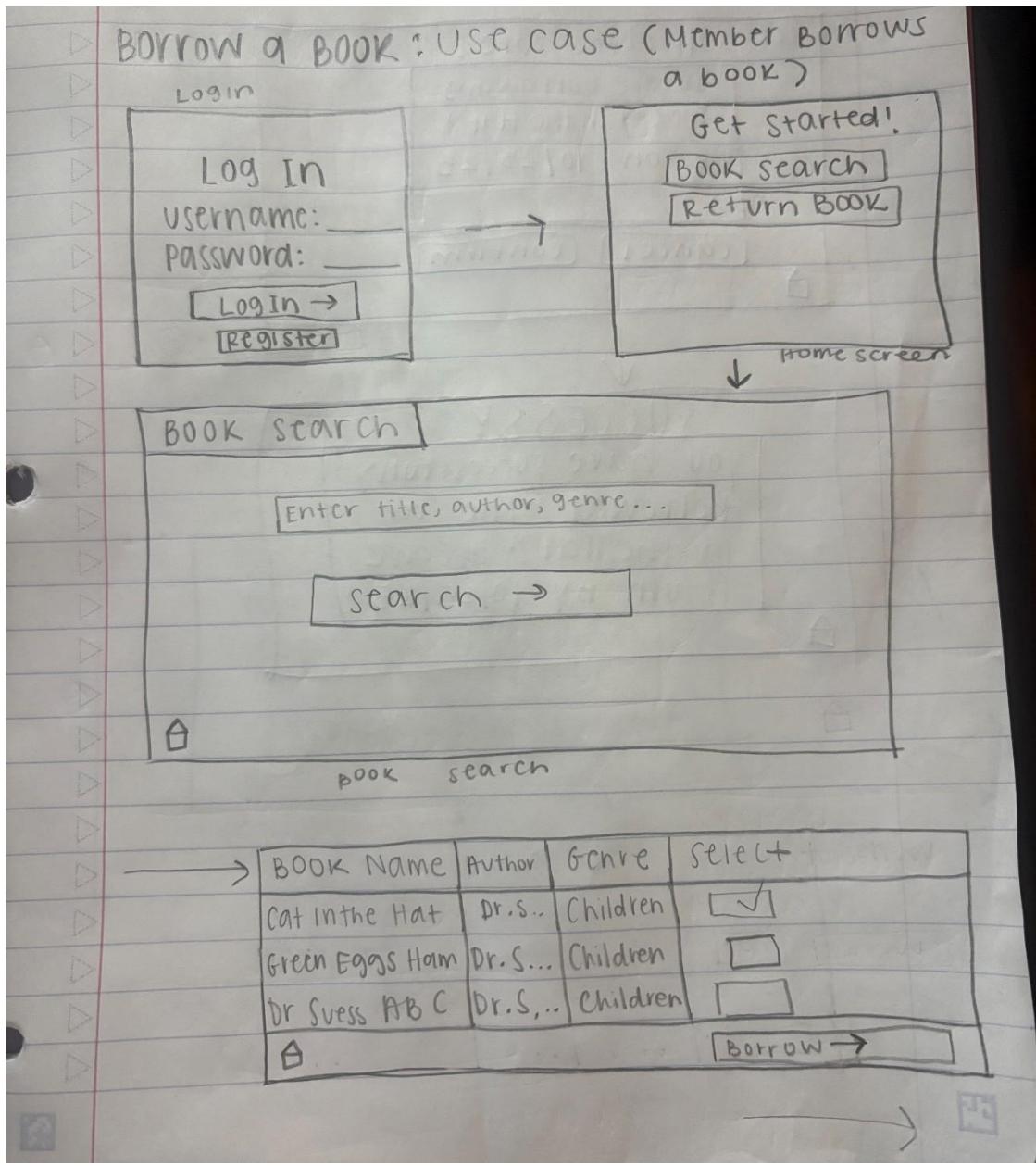


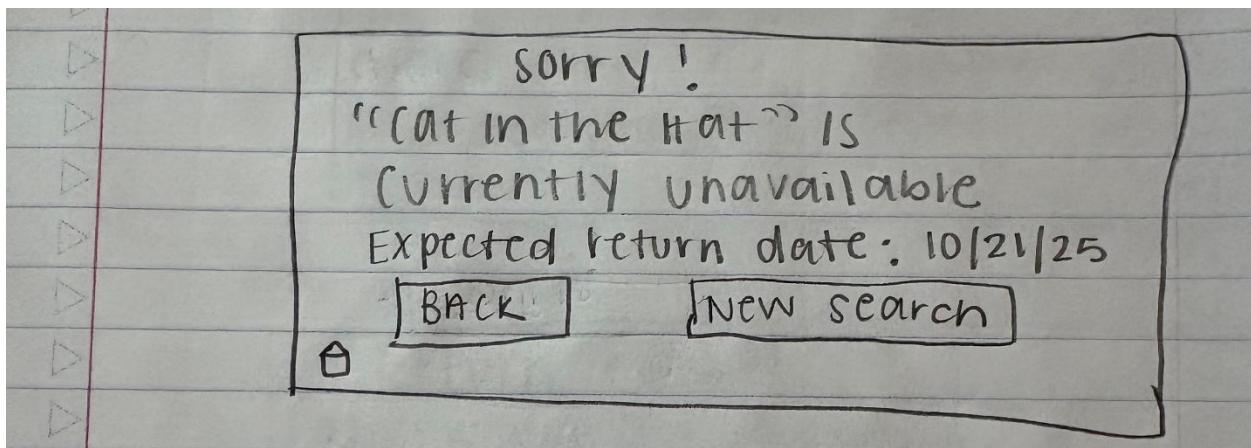
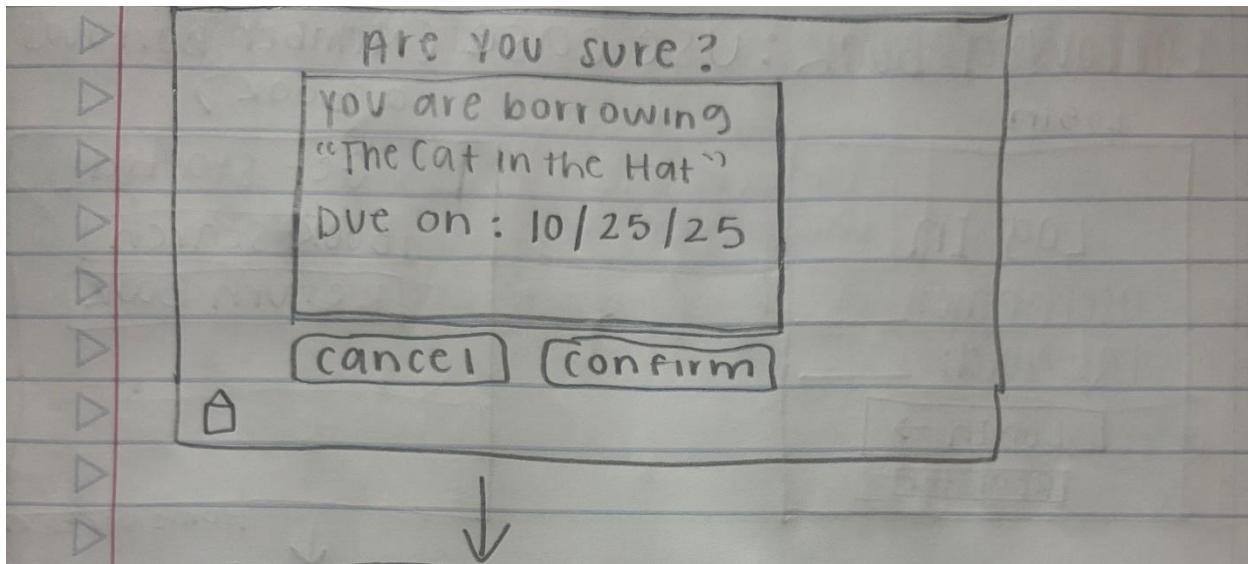


Use Case 2: Non-member creates an account

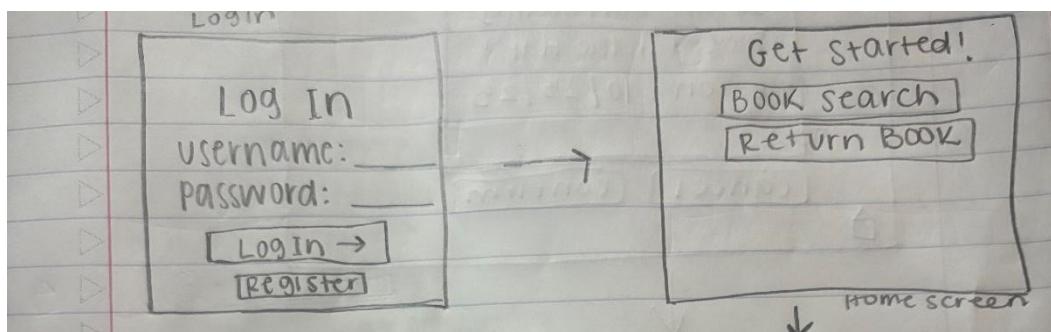


### Use case 3: Book is unavailable for member to borrow





Use case 4: Member returns book

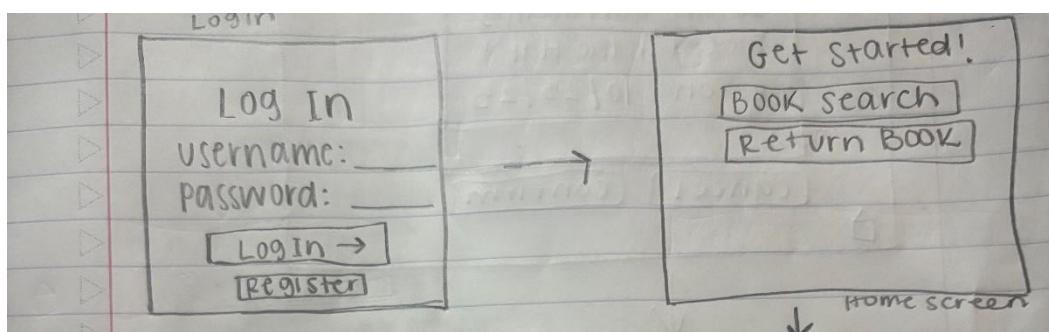


YOUR BOOKS		due date	items select
cat in the hat	10/25	overdue!	<input type="checkbox"/>
Green Eggs + Ham	10/30	none	<input checked="" type="checkbox"/>
		<b>Return →</b>	

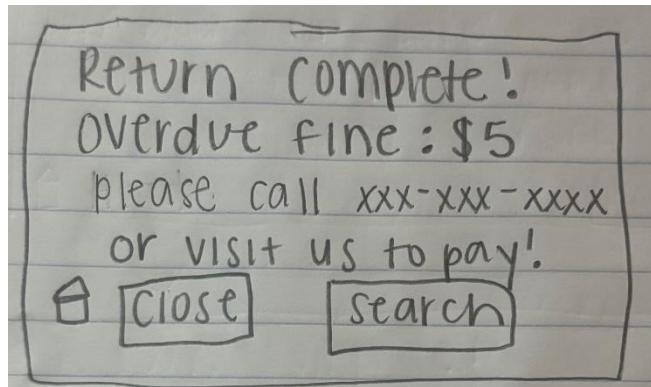
↓

Return complete! Thank you for returning this book on time!		
	<b>CLOSE</b>	<b>Search →</b>

Use case 5: Member returns overdue book



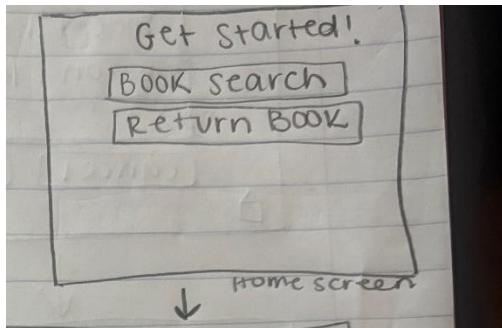
YOUR BOOKS		due date	items select
cat in the hat	10/25	overdue!	<input checked="" type="checkbox"/>
Green Eggs + Ham	10/30	none	<input type="checkbox"/>
		<b>Return →</b>	



#### User Effort Estimation :

Use Case	Steps	Clicks	Keystrokes	Total Actions
Log into system	Enter username, password, click login	3	>10	>13
Search for book	Enter search term, click search, select book, click borrow	4	>10	>14
Borrow a book	Click to confirm borrow	1	0	1
Return a book	Select return book, choose book to return, hit return	3	0	3

Additional Information:



This is the home screen that the small home icon leads to at the bottom left corner of most screens.

#### System Requirements:

Req Number	Priority Weight (1 lowest, 5 highest)	Description
Req1	5	Users must be able to log into and out of system
Req2	2	Non-members should be able to create an account
Req3	5	Librarians/faculty can log into and out of system
Req4	4	Member can search for books and see if they are available to be borrowed
Req5	4	Members can borrow books
Req6	4	Members can return books
Req7	3	System shows what books are overdue and what fine is associated
Req8	3	System updates what books are available/borrowed/returned automatically immediately after transaction
Req9	1	Members can reserve/waitlist books not yet returned

#### Use Cases:

Number	Description
1	Member borrows a book

2	Non-member creates an account
3	Book is unavailable for member to borrow
4	Member returns book
5	Member returns overdue book
6	Member logs into system
7	Member searches for a book
8	System updates after book is returned/borrowed

Traceability Matrix:

Req	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
Req1	5	X		X	X	X	X	X	
Req2	2		X						
Req3	5						X		
Req4	4	X		X				X	
Req5	4	X							
Req6	4				X	X			
Req7	3					X			
Req8	3	X		X	X	X			X
Req9	1			X					
Total PW	16	2	13	12		15	10	9	3
Max PW	5	2	5	5		5	5	5	3

### Objectives of system:

1. Provide easy way to search for books to see if they are borrowed or available
2. Track which books are borrowed, who they're borrowed by, and when that person will return the book by
3. Allow librarians to add/remove books from list

The system will allow members to search for books, borrow/return books, and view the books that they are currently borrowing. It will also allow librarians to add/remove books from system, track when books are borrowed/returned, and check if books are available or not.

### Typical customers:

- Members of library/people checking out books

- Librarians

**Project planning and development approach:**

- Software: Visual Studio Code, Java, MySQL
- Hardware: PC/laptop
- Network: Local network

**Development plan:**

W1-2: Set up project structure and determine framework

W3-4: Build system login and registration for members/librarians

W5-7: Implement basic structure for librarians: add, edit, search for book

W8: Test features and record demo for midterm

W9-11: Improve features and continue adding basic features for members: borrow, return  
view book

W12-14: Write test cases for current features and continue adding new features

W15: Record demo for final presentation