

NIDA KHAN

SP23-BSE-011

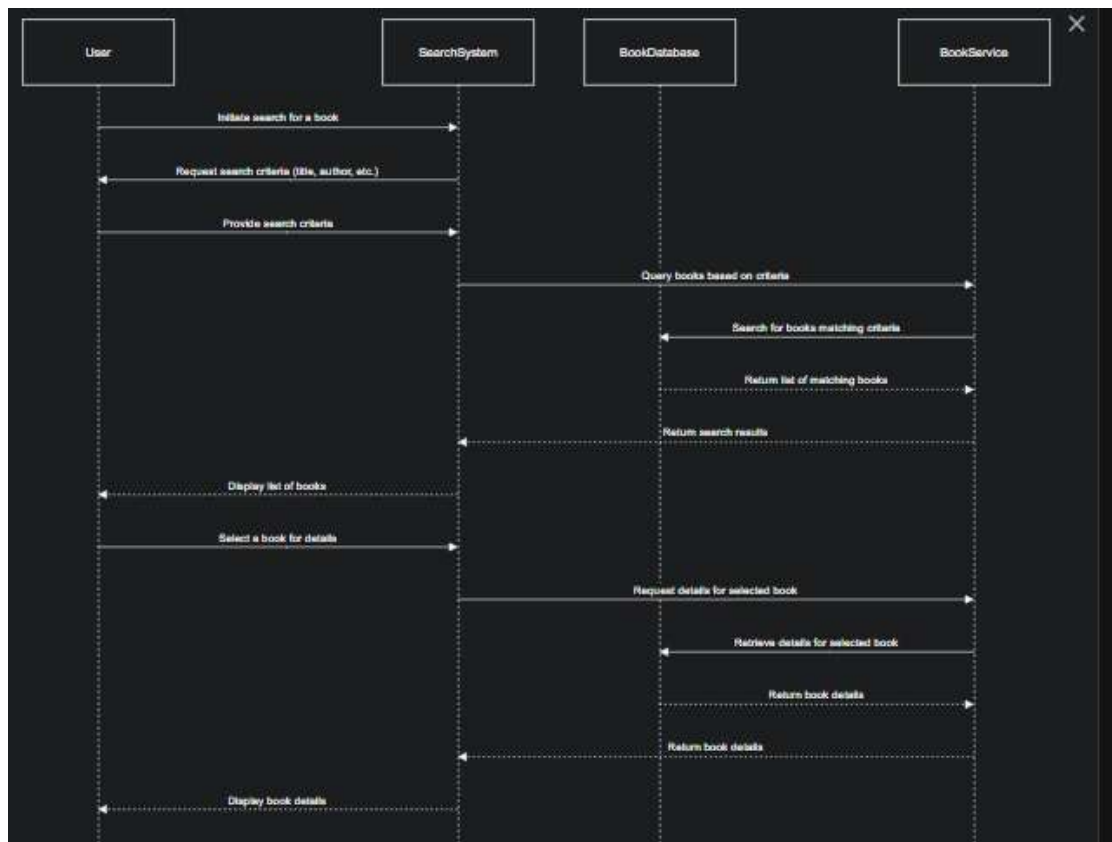
### Use Case: Search Book

Field	Details
Use Case ID	UC002
Use Case Name	Search Book
Brief Description	This use case describes how a library user searches for a book using the Library Management System by entering criteria such as title, author, etc.
Primary Actor	Library User
Secondary Actors	Librarian, Library Management System
Preconditions	• User must have access to the LMS (via web, mobile, or terminal).• The catalog must be up-to-date and indexed.
Postconditions	• The user sees matching books or is notified that no matches are found.• The search is logged for analytics or audit.
Trigger	The user selects the "Search Book" option from the Library Management System interface.
Basic Flow	1. User accesses the LMS.2. Navigates to the search page.3. Enters search criteria (e.g., title, author).4. System validates input.5. System queries the catalog.6. System displays matching results.7. User views the list.
Alternative Flows	<b>3a. Invalid Input Format:</b> – System displays an error if input (e.g., ISBN) is invalid.– User re-enters correct input. <b>6a. No Matching Results:</b> – System notifies user of no matches.– User is prompted to refine search.
Exception Flows	<b>a) System Downtime:</b> – Search unavailable; message shown. <b>b) Database Error:</b> – Error accessing catalog; log generated. <b>c) Authorization Issue:</b> – Restricted data access denied; message shown.
Business Rules	• ISBN must be 10 or 13 digits.• Search must return results in $\leq 5$ seconds.• Restricted books are only visible to authorized users.• All search activities are logged with user ID.• Catalog must be indexed at least every 24 hours.
Assumptions	• The system supports partial and multi-field searches.• The search uses current catalog data.• High availability is ensured during peak usage.

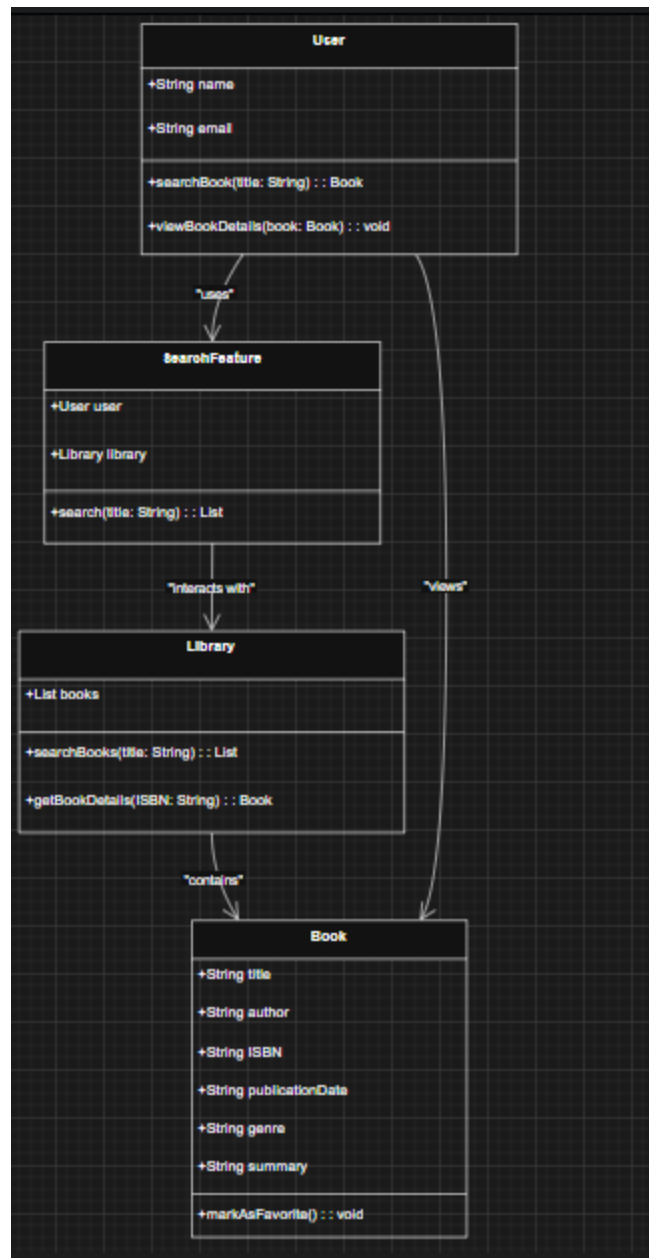
## System event design:

Actor Action (Trigger)	System Event	System Description / Response
User selects "Search Book" option	initiateSearch()	System loads the search page and prepares for input.
User enters search criteria (title, author, etc.)	—	Data entered manually — not a system event (user input).
User clicks "Search"	validateSearchInput(searchData)	System validates input, ensuring criteria are in the correct format (e.g., ISBN, title, etc.).
System validates search input	checkSearchCriteria()	System checks for the correct format (e.g., ISBN is 10 or 13 digits) and that all necessary fields are filled.
System queries the catalog for matches	queryCatalog(searchData)	System queries the database using the search criteria (e.g., title, author) to find matching books.
System processes search results	processSearchResults()	System processes the query results and prepares them for display, sorting results if needed.
System displays search results	displaySearchResults()	System displays the matching books or a message indicating no matches were found.
User views results	—	User can view the list of results or be prompted to refine search if no matches are found.
System logs search activity	logSearchActivity(userID, searchData)	System logs the search action with user ID and search criteria for analytics or audit.

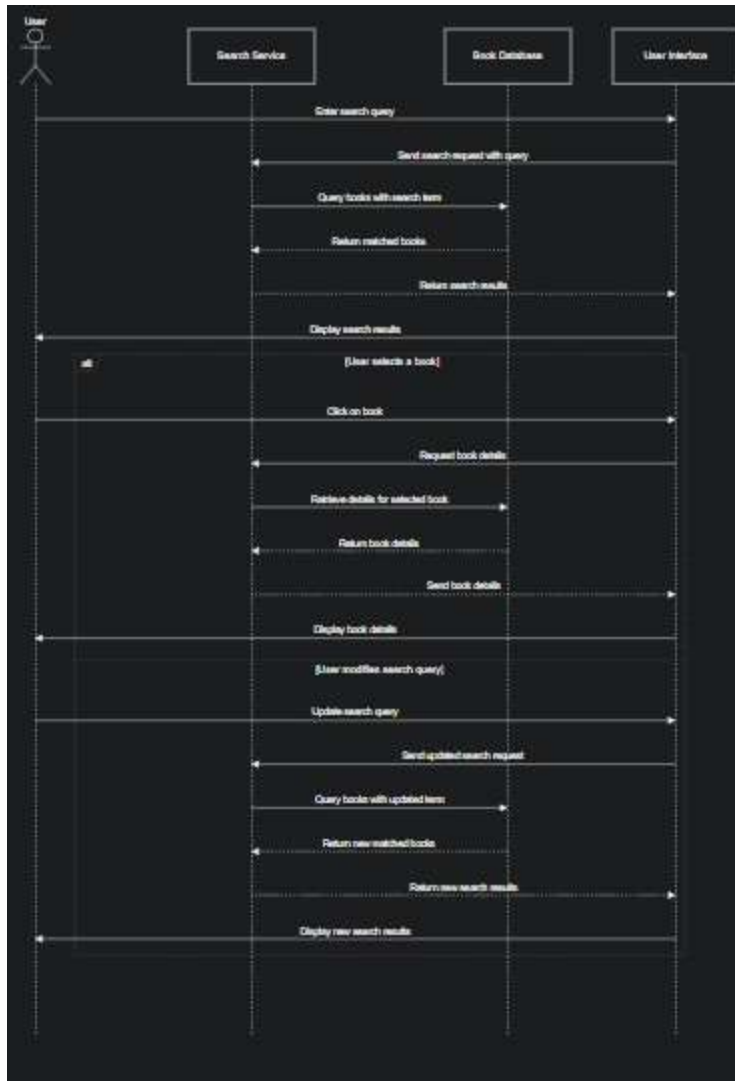
## System Sequence Diagram:



## CLASS DIAGRAM:



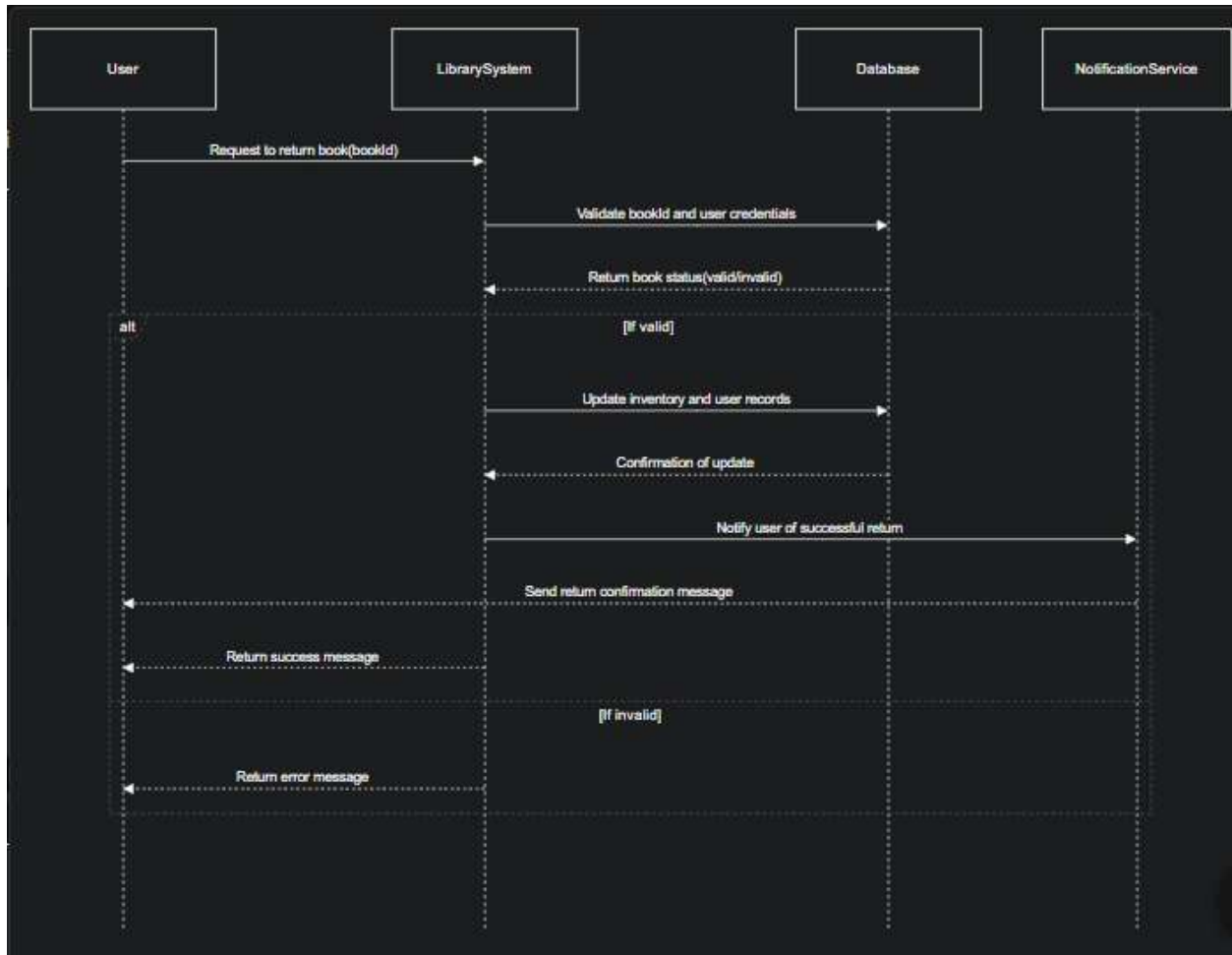
## SEQUENCE DIAGRAM:



Use Case: Return Book	
Field	Details
<b>Use Case ID</b>	UC003
<b>Use Case Name</b>	Return Book
<b>Brief Description</b>	This use case describes how a user (library member) returns a previously borrowed book using the Library Management System (LMS). The system verifies the return, updates records, and notifies the librarian if needed.
<b>Primary Actor</b>	Library User
<b>Secondary Actors</b>	Librarian, Library Management System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>The user must be logged into the LMS.</li> <li>The book must be issued to the user in the system.</li> <li>The system must be online and connected to the book database.</li> </ul>

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The book status is updated to "Available".</li> <li>• The user's account reflects the return.</li> <li>• Any fines (if applicable) are calculated and shown.</li> </ul>
<b>Trigger</b>	The user selects the "Return Book" option from the LMS or physically returns the book to the librarian.
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. User accesses the LMS and selects "Return Book".</li> <li>2. Enters or scans book ID/ISBN.</li> <li>3. System verifies the book is issued to the user.</li> <li>4. System updates the status of the book as returned.</li> <li>5. Fine (if any) is calculated based on due date.</li> <li>6. System updates the user's account and history.</li> <li>7. Confirmation is shown to the user.</li> </ol>
<b>Alternative Flows</b>	<p>3a. Book not issued to user:</p> <ul style="list-style-type: none"> <li>– System shows an error message.</li> <li>– User is asked to contact librarian.</li> </ul> <p>5a. Overdue Return:</p> <ul style="list-style-type: none"> <li>– System calculates fine.</li> <li>– Fine is shown and added to user's dues.</li> </ul> <p>a) System Offline:</p> <ul style="list-style-type: none"> <li>– Return cannot be processed; manual logging advised.</li> </ul>
<b>Exception Flows</b>	<p>b) Book Already Returned:</p> <ul style="list-style-type: none"> <li>– System shows status and blocks duplicate return.</li> </ul> <p>c) Invalid Book ID:</p> <ul style="list-style-type: none"> <li>– System notifies the user and requests re-entry.</li> </ul>
<b>Business Rules</b>	<ul style="list-style-type: none"> <li>• Returned books must update inventory instantly.</li> <li>• Fine is ₹10 per day after the due date.</li> <li>• Book can't be returned if not issued to the user.</li> <li>• All return activities are logged with timestamp and user ID.</li> </ul>
<b>Assumptions</b>	<ul style="list-style-type: none"> <li>• Barcode scanner or manual entry is available.</li> <li>• Fine payment can be settled later.</li> <li>• Return can be initiated online or at the help desk</li> </ul>

## System Sequence Diagram:



## Sequence Diagram:



**Class Diagram:**



