

LIBRARY MANAGEMENT SYSTEM



Scope Definition

- The Library Management System focuses on the Book Borrowing and Returning subsystem, enabling users to borrow and return books, manage due dates, calculate overdue fines, and send notifications for overdue items.



Book Borrowing Functionality

Tasks:

- **Search for Books**
- **Check Availability**
- **Borrow Books**
- **Due Date Generation**

Book Returning Functionality

Tasks:

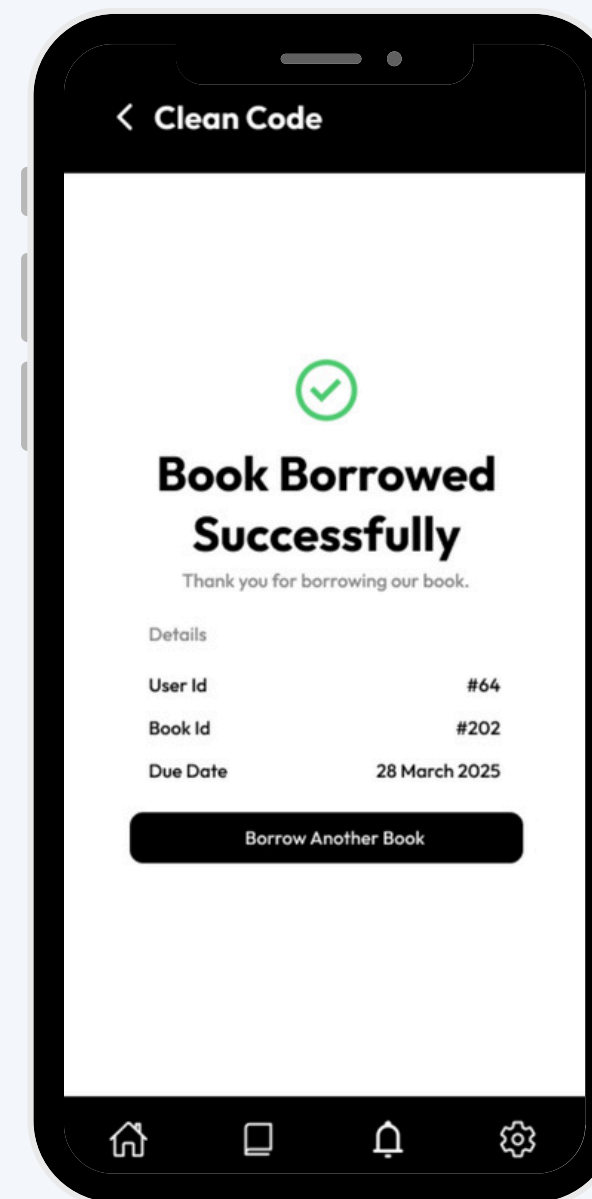
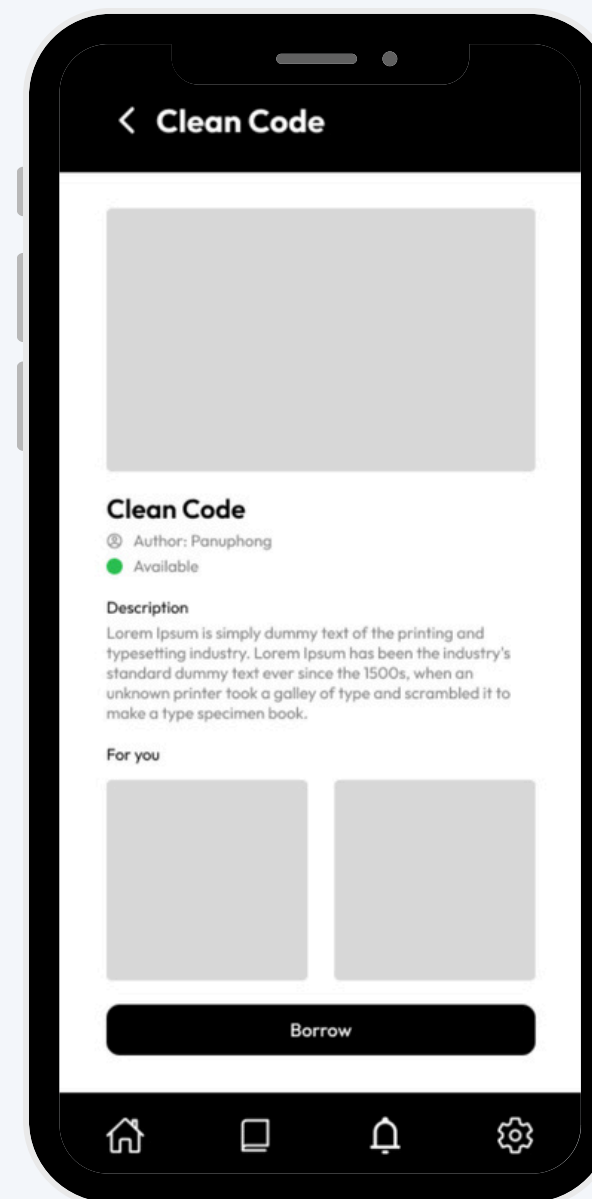
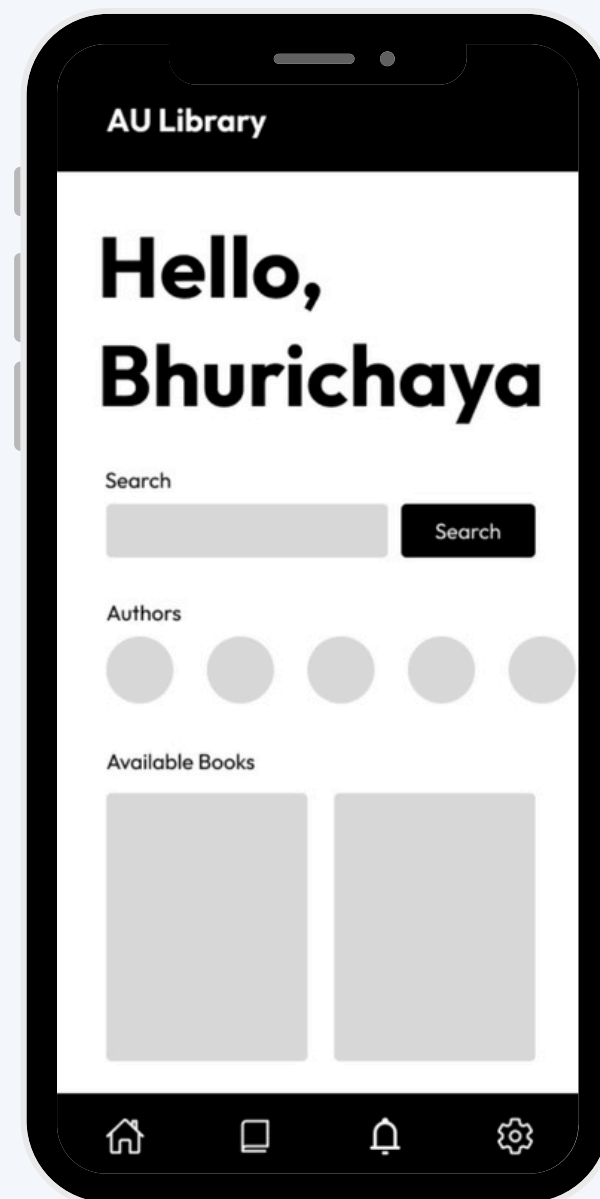
- **Record Book Returns**
- **Overdue Fine Calculation**
- **Update Book Availability**

Overdue Management Functionality

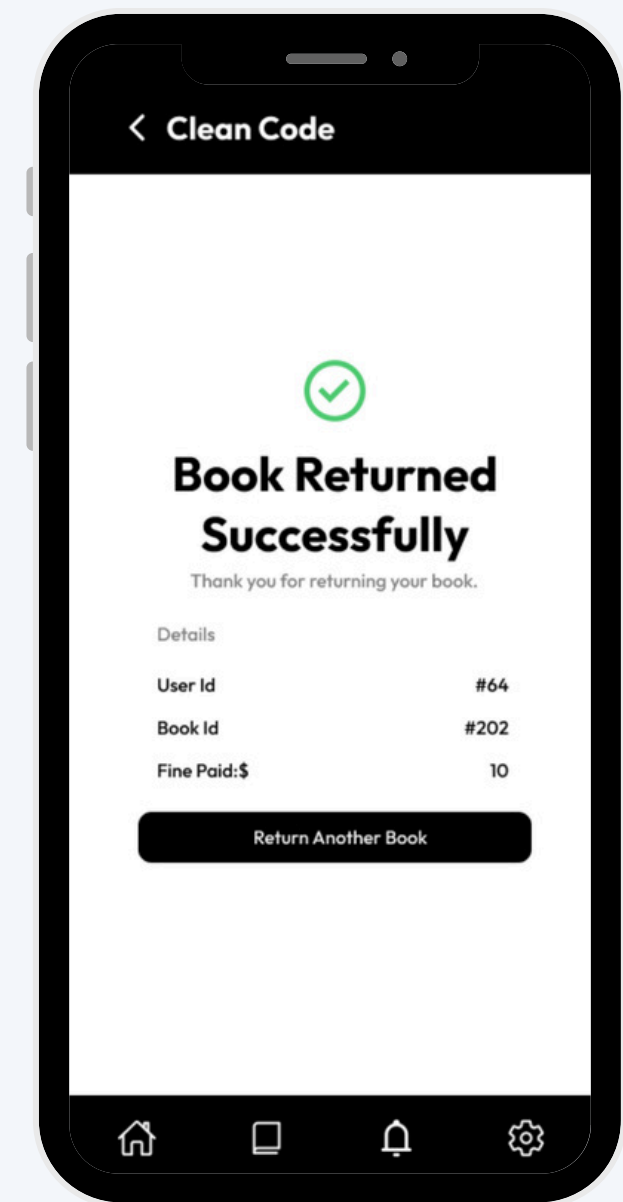
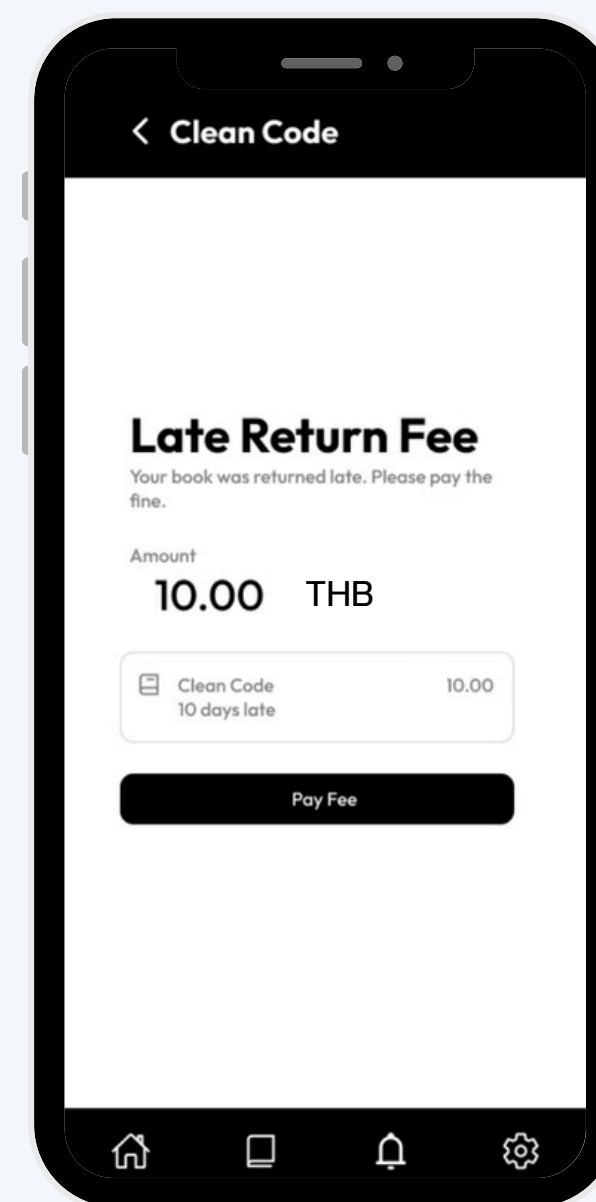
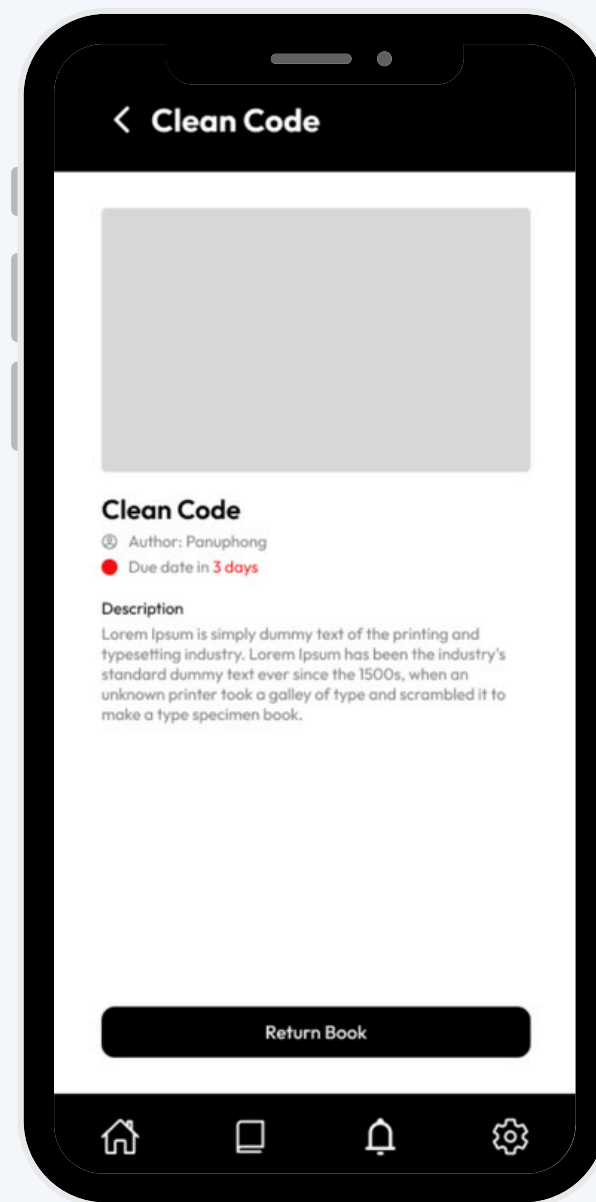
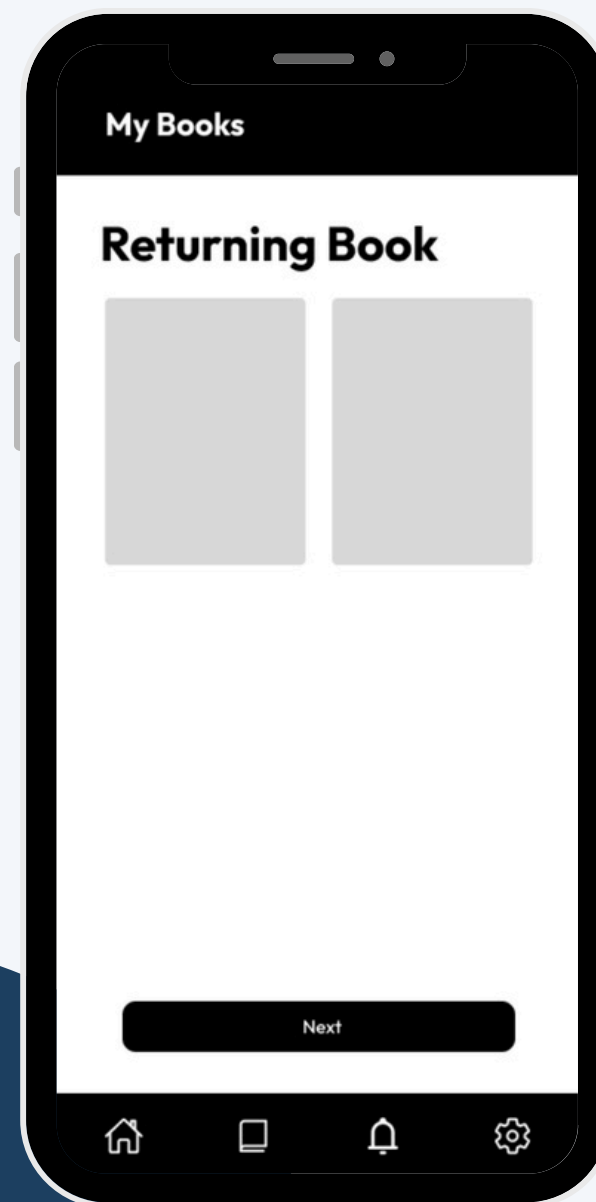
Tasks:

- **Notifications**
- **Fine Calculation**
- **Overdue Reports**

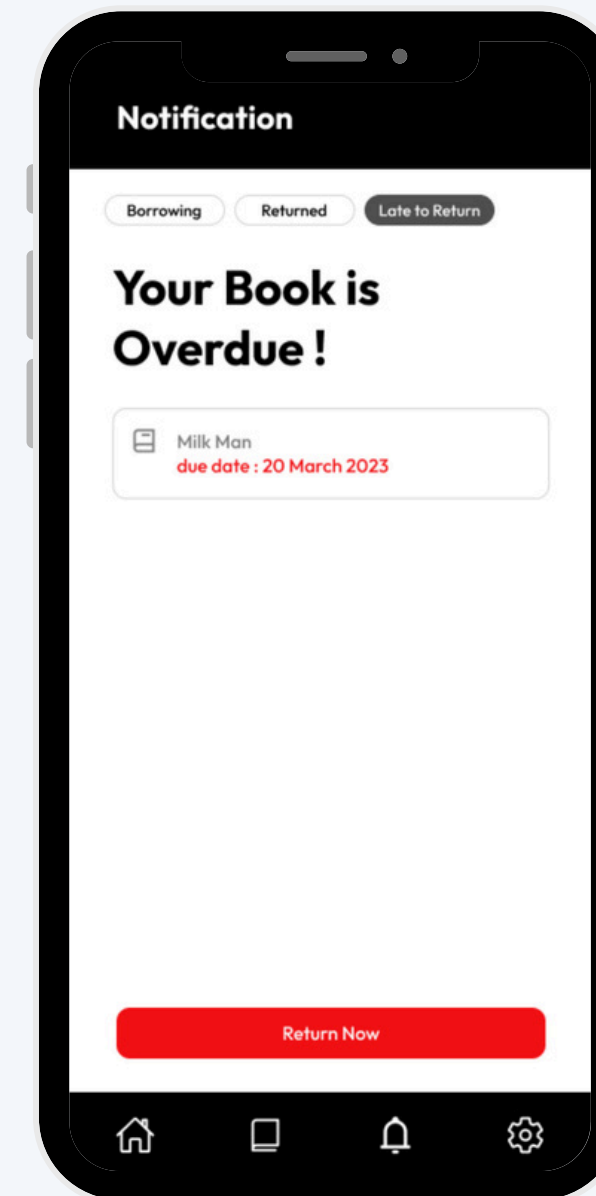
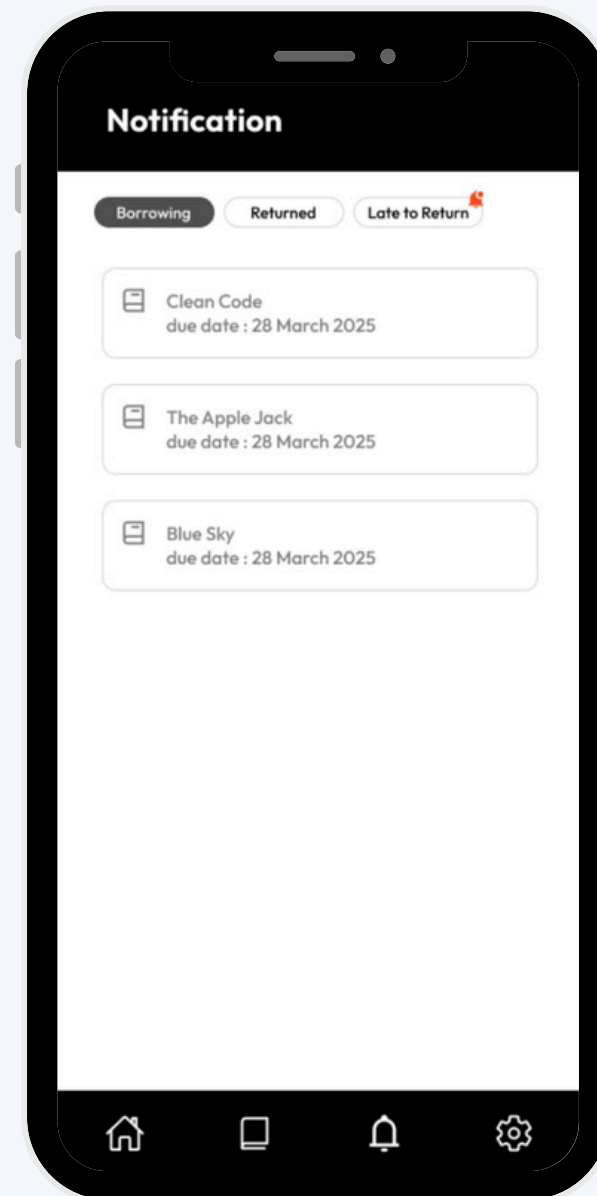
Prototype : Book Borrowing



Prototype : Book Returning



Prototype : Overdue Notification



Book Borrowing Pseudocode

```
books = []
availableBooks = []
title = Undefined
genre = Undefined
author = Undefined

IF title != Undefined OR genre != Undefined OR
author != Undefined THEN
  i=0
  WHILE (i < books.length) DO
    IF books[i].status != "borrow" AND (
books[i].title.includes(title) OR
books[i].genre == genre OR books[i].author ==
author ) THEN
      availableBooks[books[i].id] = book
    ENDIF
    i++
  ENDWHILE
ENDIF
```

Book Borrowing Pseudocode

```
availableBooks = []
borrowedBooks = []
currentDate = new Date()
bookId = Undefined
userId = Undefined
msg = Undefined

IF availableBooks[bookId] EXISTS THEN
  dueDate = currentDate + 14 days;
  availableBooks[bookId].status = "borrowed";

  borrowedBooks.append({
    userId: userId,
    bookId: bookId,
    status: "borrowed",
    dueDate: dueDate
  })

  msg = "Book borrowed successfully, due date: " +
dueDate;
ELSE
  msg = "Book is not available for borrowing";
ENDIF
```

Book Returning Pseudocode

```
borrowedBooks = []
bookId = Undefined
userId = Undefined
currentDate = new Date()
msg = Undefined

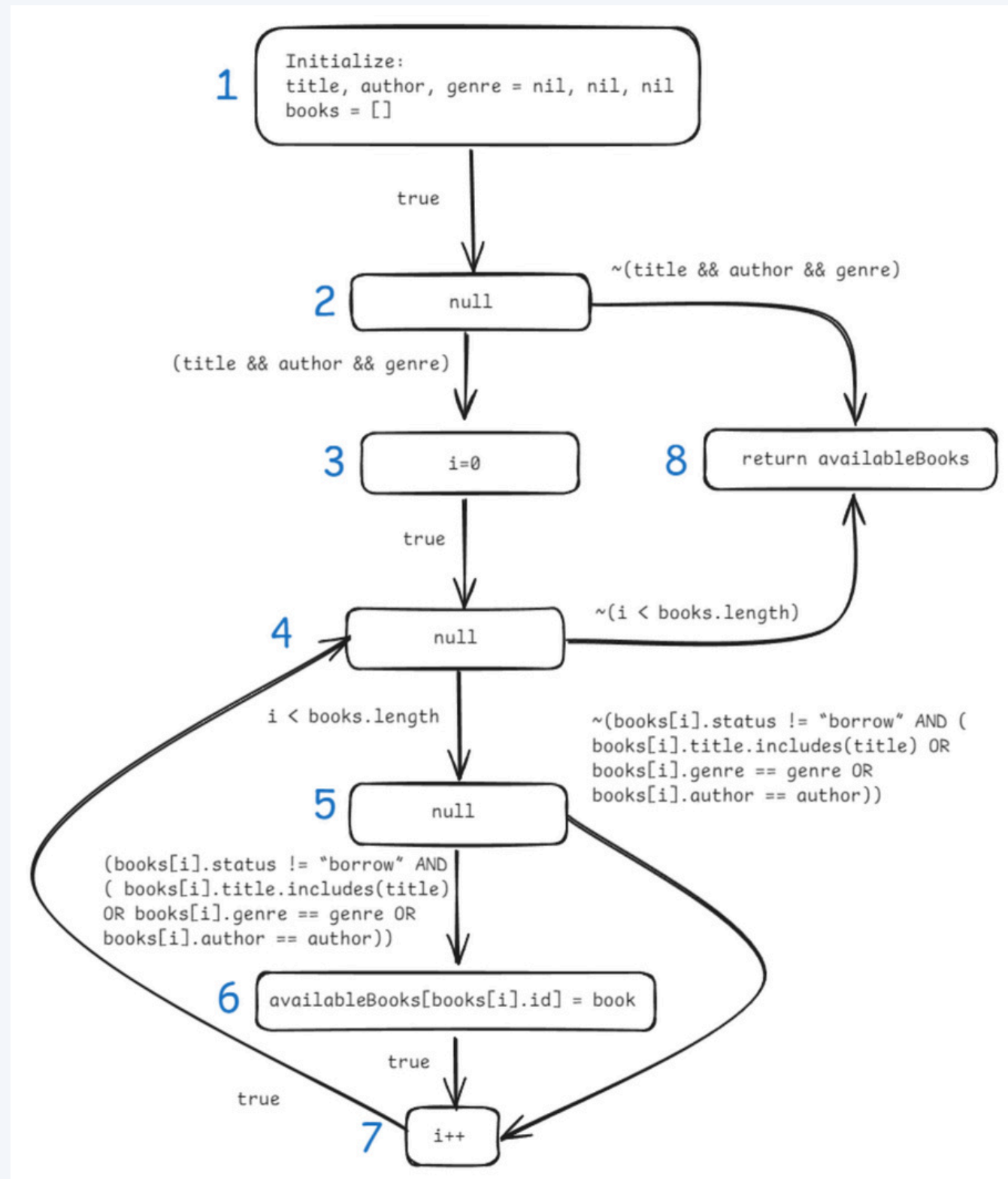
msg = "Book return failed. No matching record found.";
IF borrowedBooks.length > 0 THEN
    i = 0
    WHILE (i < borrowedBooks.length) DO
        IF borrowedBooks[i].bookId == bookId AND
        borrowedBooks[i].userId == userId THEN
            IF currentDate <= borrowedBooks[i].dueDate THEN
                fee = 0;
                msg= "Book returned successfully. No fee.";
            ELSE
                daysLate = (currentDate -
                borrowedBooks[i].dueDate).inDays();
                fee = daysLate * 10;
                msg = "Book returned late. Fee: THB " + fee;
            ENDIF
            availableBooks[bookId].status = "available";
            borrowedBooks.remove(borrowedBook);
        ENDIF
        i ++
    ENDFOR
ENDIF
```

Overdue Notification Pseudocode

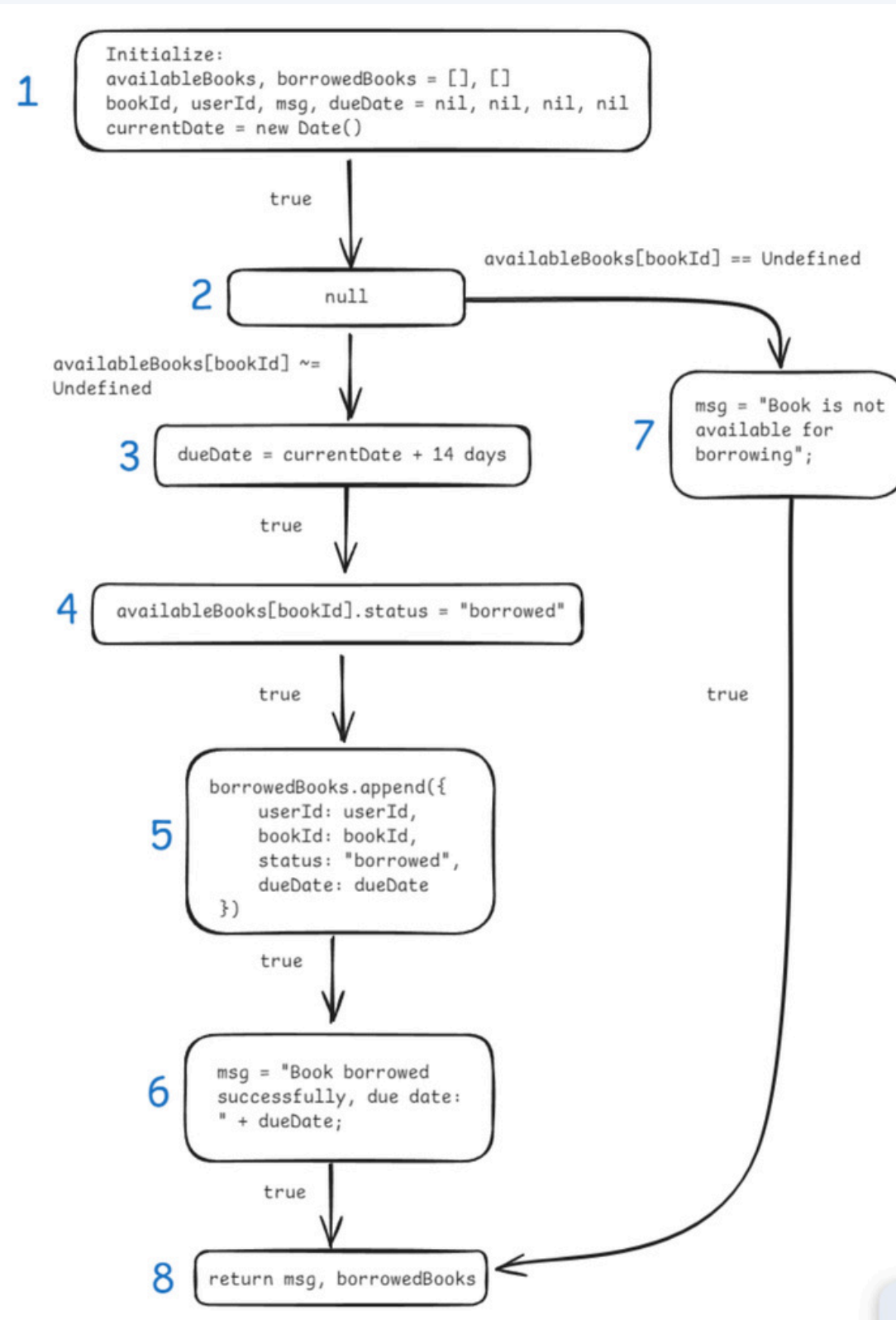
```
borrowedBooks = []
currentDate = new Date()
dayLate = 0
fee =
i = 0
WHILE (i < borrowedBooks.length) DO
    IF borrowedBooks[i].status == "borrowed"
    AND currentDate > borrowedBooks[i].dueDate
    THEN
        daysLate = (currentDate -
        borrowedBooks[i].dueDate).inDays(); //
        Calculate how many days late
        fee = daysLate * 10;

        sendNotifcation(borrowedBooks[i].userId,
        "Dear User " + borrowedBooks[i].userId + ",
        your borrowed book (ID: " +
        borrowedBooks[i].bookId + ") is overdue by " +
        daysLate + " days. The current fee is THB " +
        fee + ". Please return the book as soon as
        possible.");
    ENDIF
    i++
ENDFOR
```

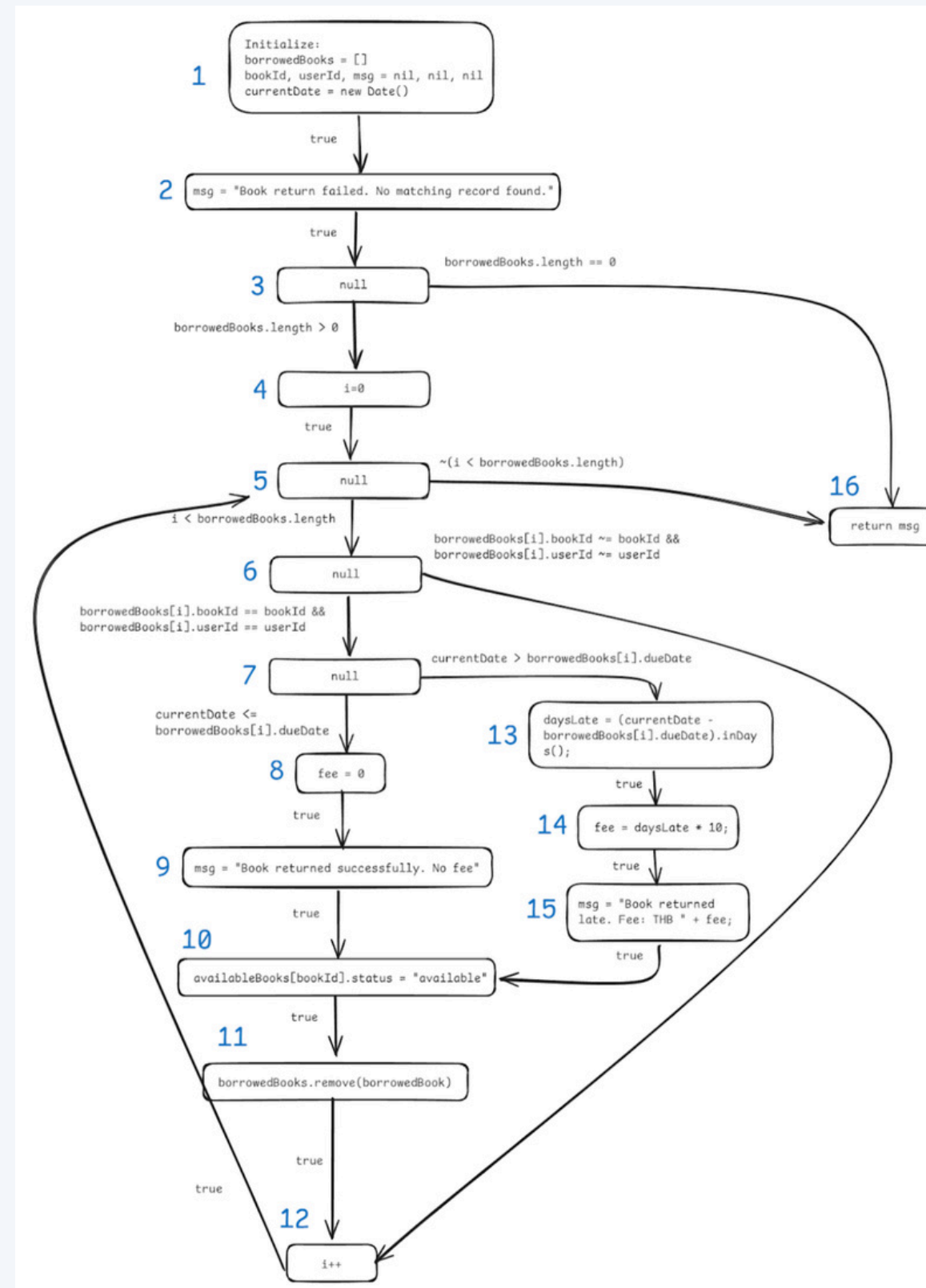
Data Flow : Book Searching Flow



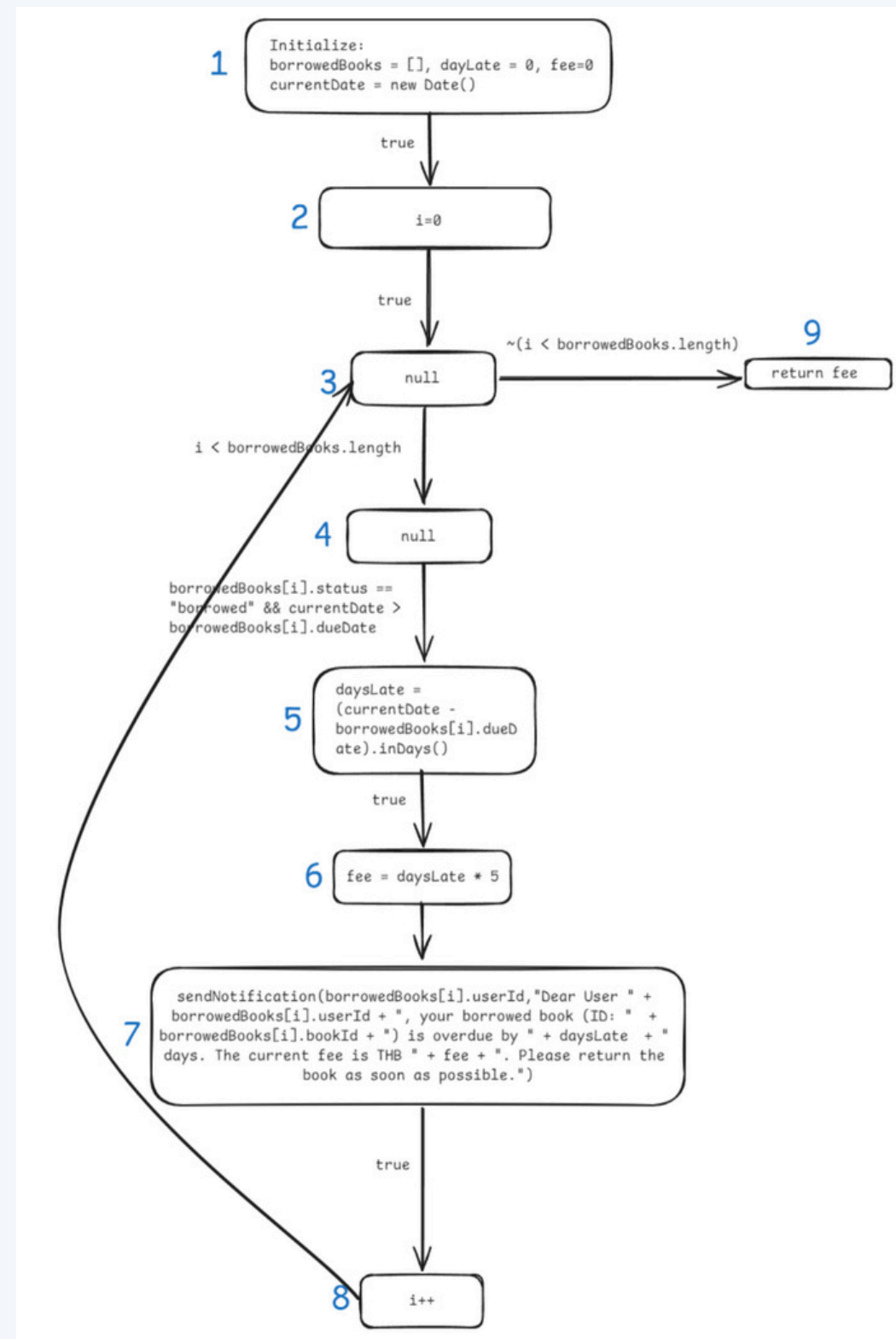
Data Flow : Book Borrowing Flow



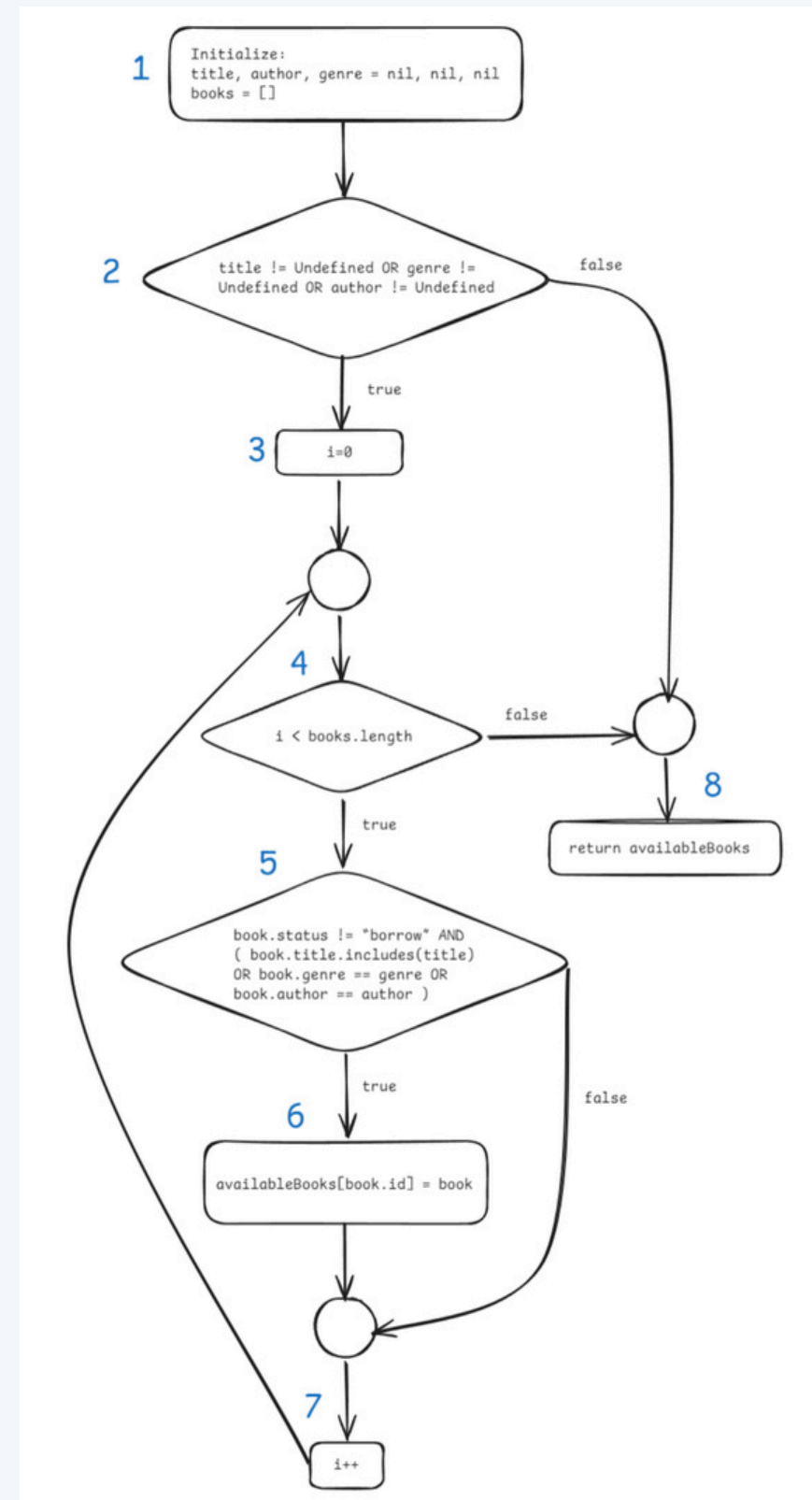
Data Flow : Book Returning Flow



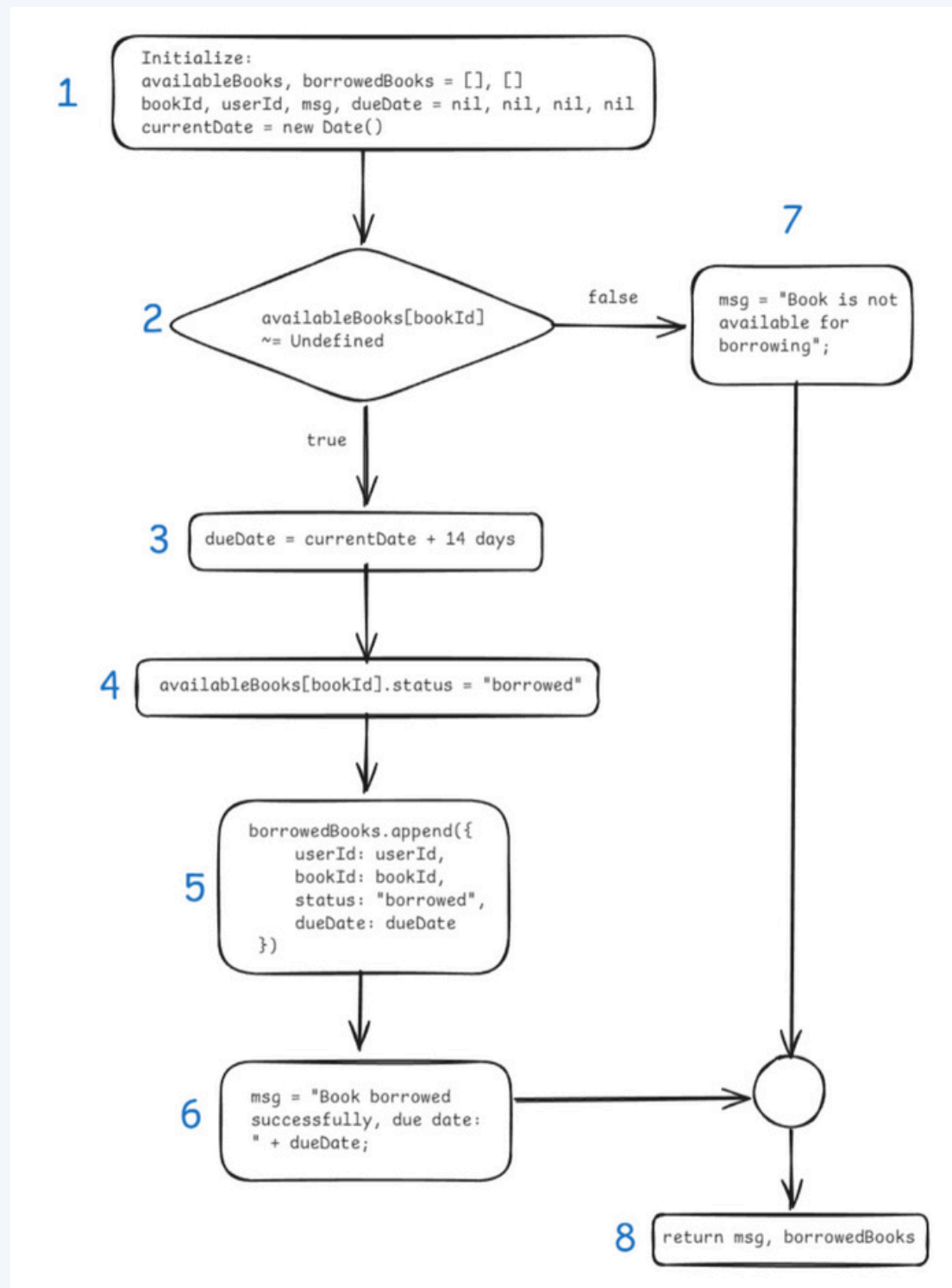
Data Flow : Overdue Notification Flow



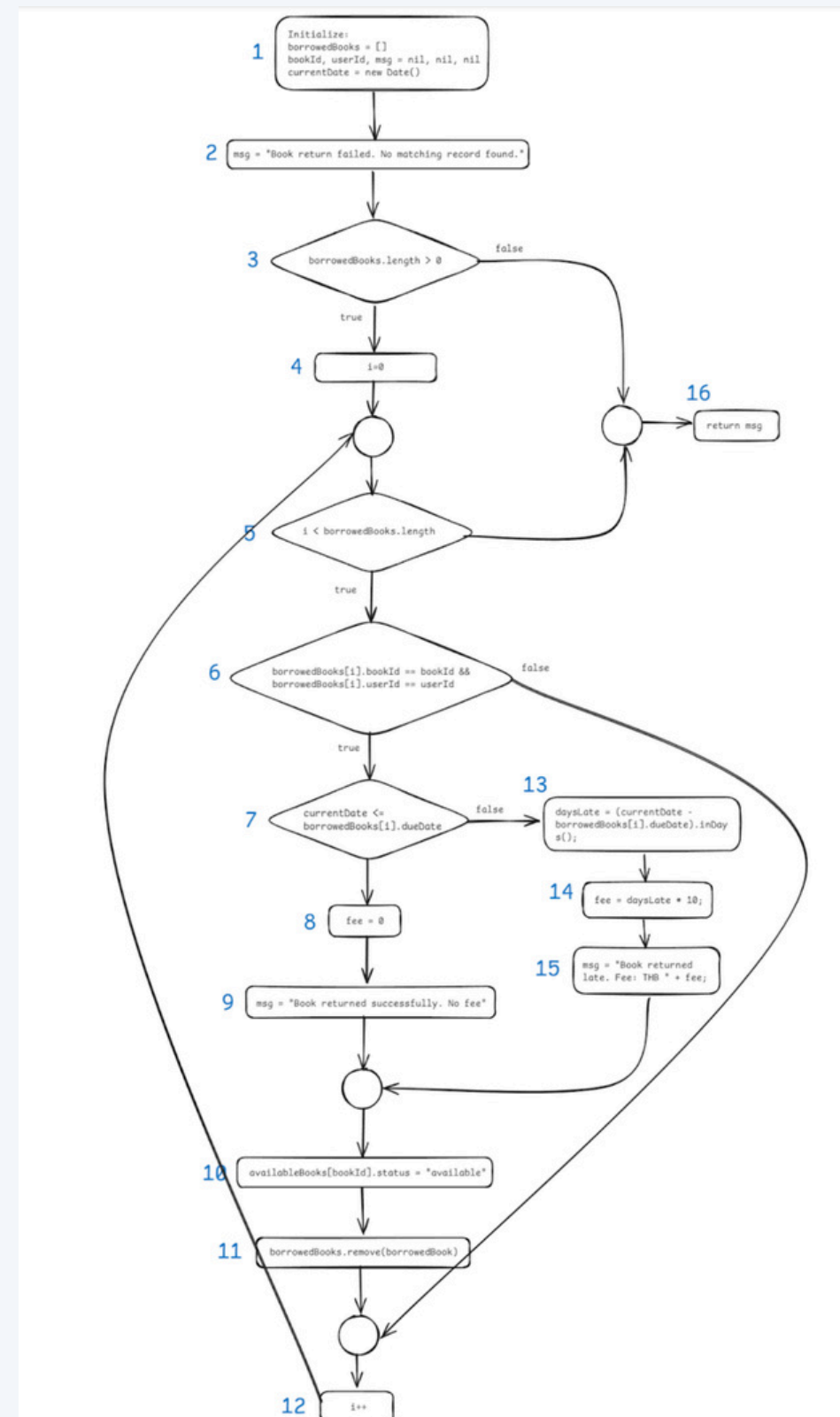
Control Flow : Book Searching



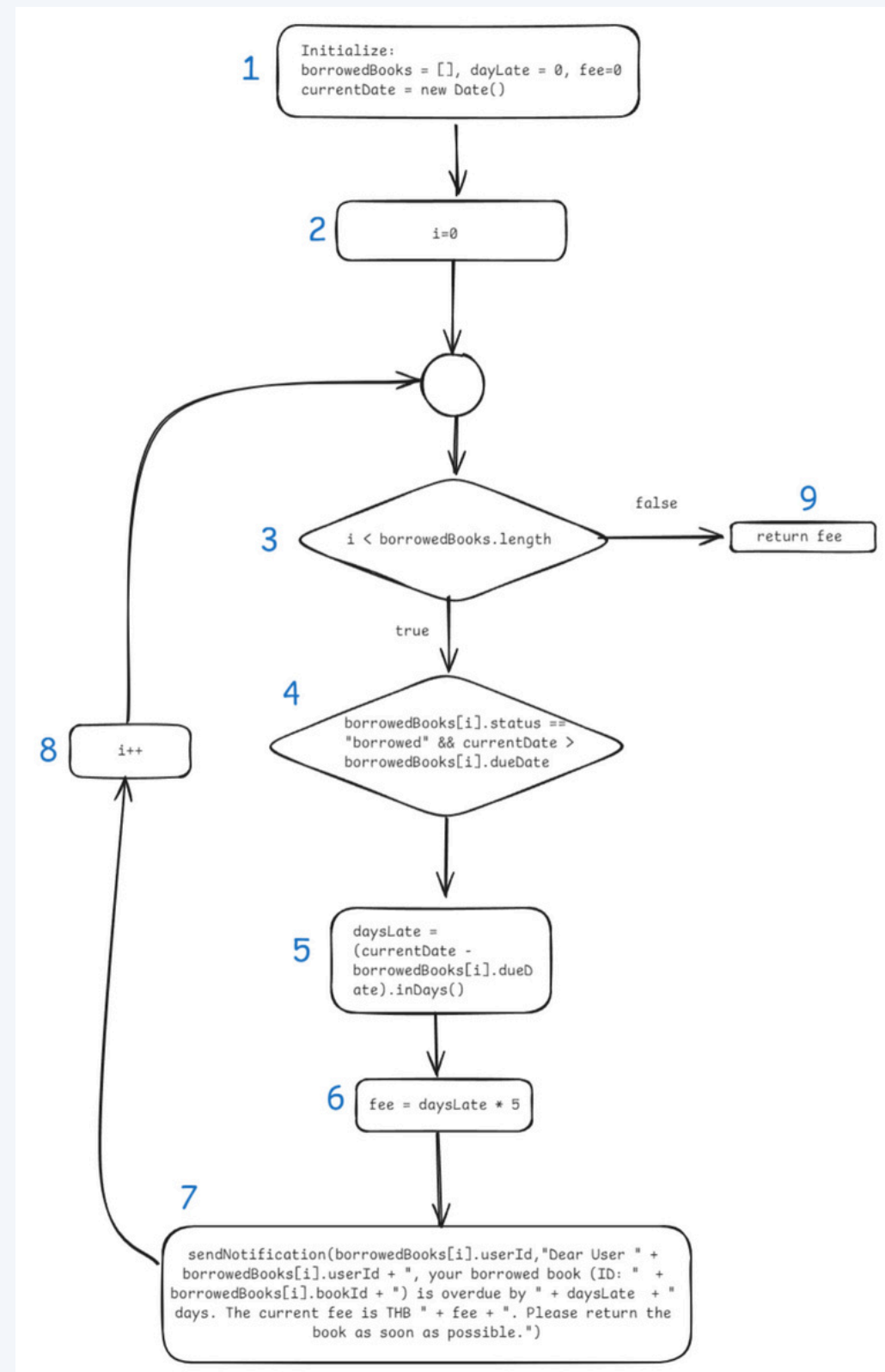
Control Flow : Book Borrowing



Control Flow : Book Returning



Control Flow : Overdue Notification



Test Case ID	Steps to Execute	Description	Expected Result
UAT-001	1. Search for "Clean Code" 2. Check availability	Validate book search functionality.	Correct books are displayed with availability.
UAT-002	1. Borrow the book "Clean Code"	Validate book borrowing.	Book is borrowed, status updated, due date generated.
UAT-003	1. Return the book on time.	Validate book return functionality.	Book returned, status updated, no fine.
UAT-004	1. Return the book late.	Validate overdue fine calculation.	Fine is calculated correctly based on the delay.
UAT-005	1. View overdue books.	Validate overdue notifications.	User is notified correctly for overdue books.

Unit Test Table for Library Management System

Test Case ID	Component	Test Description	Input	Expected Output
UNIT-001	Book Search	Verify that the book search functionality works correctly.	Search for title "Clean Code"	List of books matching the title with availability.
UNIT-002	Availability Check	Verify that the system checks book availability accurately.	Book ID of "Clean Code"	Status shows "available" or "not available".
UNIT-003	Book Borrowing	Verify that borrowing a book updates its status and due date.	User ID and Book ID	Status updated to "borrowed", due date generated.
UNIT-004	Book Returning	Verify that returning a book updates its status and calculates fines if late.	User ID and Book ID, return date	Status updated to "available", fine calculated if overdue.
UNIT-005	Overdue Notification	Verify that overdue notifications are sent correctly.	User ID and overdue book ID	Notification sent via email/SMS for overdue book.
UNIT-006	Fine Calculation	Verify that overdue fines are calculated correctly based on delay.	Due date, current date	Correct fine amount based on days late.

Thank you

6410521	Panuphong	Burakitphachai
6411390	Nanthanat	Ounma
6420134	Bhurichaya	Thuraphan
6420061	Nitipoom	Aumpitak