

# **Library Booking System: Database**

## **Week 3 Report – CRUD Demonstration**

**Database CS A**



**Made by:**

<b>Klara Ahinta Daniswara</b>	<b>24/532751/PA/22531</b>
<b>Daffa Aryza Pasya</b>	<b>24/532884/PA/22549</b>
<b>Shadiq Arif Aryasatya</b>	<b>24/536702/PA/22763</b>

**DEPARTEMEN ILMU KOMPUTER dan ELEKTRONIKA  
FAKULTAS MATEMATIKA dan ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA  
2025**

## **Tables of Content**

<b>Objective.....</b>	<b>2</b>
<b>CRUD Coverage Matrix.....</b>	<b>2</b>
<b>CRUD Functionality Summary.....</b>	<b>3</b>
<b>Entity Breakdown &amp; Logic.....</b>	<b>5</b>
<b>Observed Behavior.....</b>	<b>7</b>
<b>Notes.....</b>	<b>8</b>

# Objective

Week 3 focus: Develop core CRUD features for the main entities (add, view, update, delete).

## CRUD Coverage Matrix

DBProject_Group4_LibraryBorrowingTracker / week3_CRUD_demo /	
Name	Last commit message
...	
adding a book to the library.png	CRUD demo for adding a book to the library
create a borrow record.png	CRUD demo to create a borrow record
creating a book copy.png	CRUD demo for adding a book copy to the library
creating a member.png	CRUD demo for creating a member
deleting a book.png	CRUD demo for deleting a book
getting a book based on keyword (search feature).png	CRUD demo for search feature of book
getting a book by isbn.png	CRUD demo for getting a book by ISBN
getting all book copies.png	CRUD demo for getting all book copies
getting all books from the library.png	CRUD demo for getting all book from the library
getting all borrow records.png	CRUD demo to get all borrow records
updating a book content.png	CRUD demo for updating a book
updating a borrow (returning a book).png	CRUD demo for borrow update (book return)

## CRUD Functionality Summary

Entity	Create	Read	Update	Delete
Book	Successfully executed via ISBN	Retrieval, keyword search, and ISBN lookup verified	Full book detail updates confirmed	Removal executed with successful response
Book Copy	Individual copy insertion working	Full retrieval and ID lookup verified	Detail updates functioning and keyword search active	Delete logic implemented and confirmed
Member	Creation successful via JSON body	Retrieval and keyword search (name/email) verified	Partial/Full update logic functioning	Removal executed with successful response
Borrow Record	Record insertion success via borrow logic	Full listing and ID lookup verified	Return date updates functioning	Removal executed with successful response
Fine	Fine creation via borrow/staff ID verified	Retrieval and ID lookup verified	Status update (mark as paid) functioning	Removal executed with successful response

Staff	Creation successful via JSON body	Full retrieval and email lookup verified	Detail updates functioning; keyword search active	Removal executed with successful response
-------	-----------------------------------	--	---	---

# Entity Breakdown & Logic

## Entity: Book

- **Create:** Uses `@PostMapping` to accept a book object. It includes `@Valid` to ensure that constraints (like ISBN format) are met before the service layer processes the addition.
- **Read:** Offers three access points: `getAllBooks()` for the full catalog, `getBook(isbn)` for specific details, and a `/search` endpoint to filter books by keyword.
- **Update:** Uses `@PutMapping` targeting the ISBN. It allows for a full replacement of book metadata for a specific ISBN.
- **Delete:** A standard `@DeleteMapping` that removes the book record based on its unique ISBN.

## Entity: Book Copy

- **Create:** Adds specific physical instances of a book to the database.
- **Read:** Provides a list of all copies and a specific lookup by the internal ID. The `/search` function allows filtering by copy-specific attributes.
- **Update:** Updates the status or details of a specific copy
- **Delete:** Removes a specific copy from the inventory using its unique ID.

## Entity: Member

- **Create:** Registers new library users. It takes a JSON object containing member details like name and contact info.
- **Read:** Supports broad retrieval and a targeted `/search` endpoint typically used for finding members by name or email.
- **Update:** The `updateMember` method allows modifying user profiles
- **Delete:** Permanently removes a member's profile from the system.

## Entity: Borrow Record

- **Create:** Unlike standard CRUD, this `borrowBook` method maps specific IDs
- **Read:** Tracks the history of borrows through list views and ID lookups.
- **Update:** Primarily focused on the "Return" logic. Instead of a generic update, the `returnBook` endpoint specifically updates the `returnDate`.
- **Delete:** Allows for the removal of a borrow transaction record.

### **Entity: Fine**

- **Create:** Fines are generated by linking a borrowId and a staffId with a specific currency amount.
- **Read:** Admins can view all fines or search for specific ones using keywords (e.g., searching for "UNPAID").
- **Update:** Features a specialized markPaid endpoint that updates the payment status and sets the datePaid.
- **Delete:** Removes a fine record from the system.

### **Entity: Staff**

- **Create:** Used to onboard new library employees.
- **Read:** In addition to standard ID lookup, it features a unique getStaffByEmail endpoint, which is essential for authentication or staff-specific profile lookups.
- **Update:** Allows for the modification of staff records, such as job titles or contact information.
- **Delete:** Removes a staff member from the system via their unique ID.

## **Observed Behavior**

- All tested endpoints return HTTP 200 responses.
- Data persistence confirmed across multiple operations.
- Update and search functions manipulate and retrieve expected records.
- Borrow record lifecycle demonstrates multi-entity logic (member, staff, book).

## Notes

- Missing operations (update/delete for members/copies) may be implemented later.
- Role-based access control not tested here.
- The system is functionally complete for CRUD demonstration.