ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

Institute of Technology of Cambodia

Department of Information of Technology

Group: I4-GIC-C

Course : Software Engineering

Topic : Report week 3 Library Management tracking System

Name Student	ID of student	Score
RIN NAIRITH	e20221557	.....
THOU LAIHENG	e20220843	.....

Lecturer: **Mr. ROUEN Pacharoth**

Academic year 2025-2026

1. Introduction (Nairith part)

This report presents the Week 3 progress of the Library Management System project. The main objective of this stage is to analyze system requirements, identify core entities, define their relationships, and design a clear Entity Relationship (ER) Diagram.

The system is designed to support essential library operations such as book management, borrowing, reservation, fine calculation, notifications, and auditing. Emphasis in Week 3 is placed on data structure design rather than full system implementation.

2. System Scope

The Library Management System aims to:

- Manage books and their physical copies
- Handle borrowing and reservation processes
- Track fines for overdue items
- Notify users about important events
- Record user actions for accountability

At this stage, the following entities have been identified and designed:
Contact, Book, BookCategory, BookCopy, Borrow, Reservation, Fine, Notification, AuditLog

3. Entity Relationship (ER) Diagram Overview

The ER diagram models how data is stored and how entities interact within the system. The design follows normalization principles to avoid redundancy and ensure data consistency.

- Key design decisions
- Logical book data is separated from physical copies
- Borrowing is linked to specific book copies
- Fine and notification entities depend on borrowing activities
- Audit logging tracks system actions without affecting business logic

4. Entity Descriptions and Relationships

4.1 Contact Entity

The Contact entity stores user communication information.

Attributes (example): contact_id (PK), full_name, email, phone, address

Relationships:

- One Contact can be associated with multiple Borrow records
- One Contact can make multiple Reservations
- One Contact can receive multiple Notifications
- This entity ensures reliable communication and user identification.

4.2 BookCategory Entity

The BookCategory entity classifies books into categories.

Attributes: category_id (PK), category_name, description

Relationships:

- One BookCategory can have many Books
- Each Book belongs to one BookCategory

This relationship supports efficient searching and organization.

4.3 Book Entity

The Book entity represents general book information.

Attributes: book_id (PK), title, author, ISBN, publication_year

Relationships:

- One Book belongs to one BookCategory

- One Book can have many BookCopies

This separation allows the system to manage multiple physical copies of the same book.

4.4 BookCopy Entity

The BookCopy entity represents each physical copy in the library.

Attributes: copy_id (PK), book_id (FK), status (Available, Borrowed, Reserved, Lost)

Relationships:

- One BookCopy belongs to one Book
- One BookCopy can be involved in multiple Borrow records over time
- One BookCopy can be reserved through Reservation

This design allows precise tracking of inventory.

4.5 Borrow Entity

The Borrow entity manages lending transactions.

Attributes: borrow_id (PK), contact_id (FK), copy_id (FK), borrow_date, due_date, return_date

Relationships:

- One Contact can have many Borrow records
- One BookCopy can appear in many Borrow records
- One Borrow can generate one Fine
- One Borrow can trigger multiple Notifications

This entity is central to library operations.

4.6 Reservation Entity

The Reservation entity handles book reservations.

Attributes: reservation_id (PK), contact_id (FK), book_id (FK), reservation_date, status

Relationships:

- One Contact can make many Reservations
- One Book can have many Reservations

This supports fair access to popular books.

4.7 Fine Entity

The Fine entity records penalties for overdue or lost books.

Attributes: fine_id (PK), borrow_id (FK), amount, status (Paid / Unpaid)

Relationships:

- One Borrow record can generate one Fine
- One Fine is linked to only one Borrow

This ensures clear financial accountability.

4.8 Notification Entity

The Notification entity stores system messages sent to users.

Attributes: notification_id (PK) , contact_id (FK), message, sent_date, type (Reminder, Overdue, Reservation Ready)

Relationships:

- One Contact can receive many Notifications
- Notifications are often triggered by Borrow or Reservation events
- This entity improves user awareness and system communication.

4.9 AuditLog Entity

The AuditLog entity records system activities.

Attributes: log_id (PK), action, entity_name, entity_id, timestamp, performed_by

Relationships:

- AuditLog is linked logically to system users and actions
- It does not directly affect business transactions

This entity supports monitoring, debugging, and security.

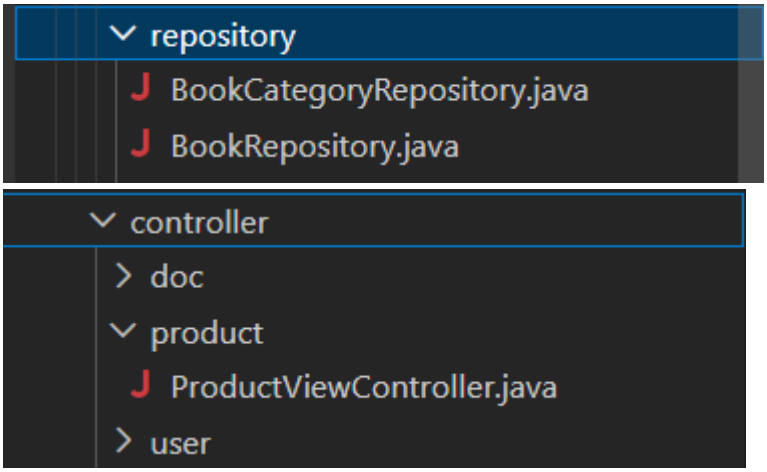
5. ER Diagram Relationship Summary

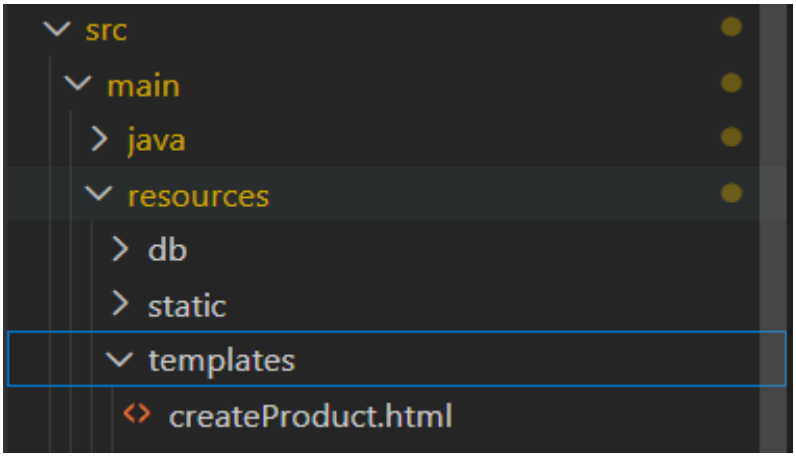
Relationship Type
BookCategory → Book One-to-Many
Book → BookCopy One-to-Many
Contact → Borrow One-to-Many
BookCopy → Borrow One-to-Many
Borrow → Fine One-to-One
Contact → Reservation One-to-Many
Book → Reservation One-to-Many
Contact → Notification One-to-Many

6. Product Management (Laiheng's part)

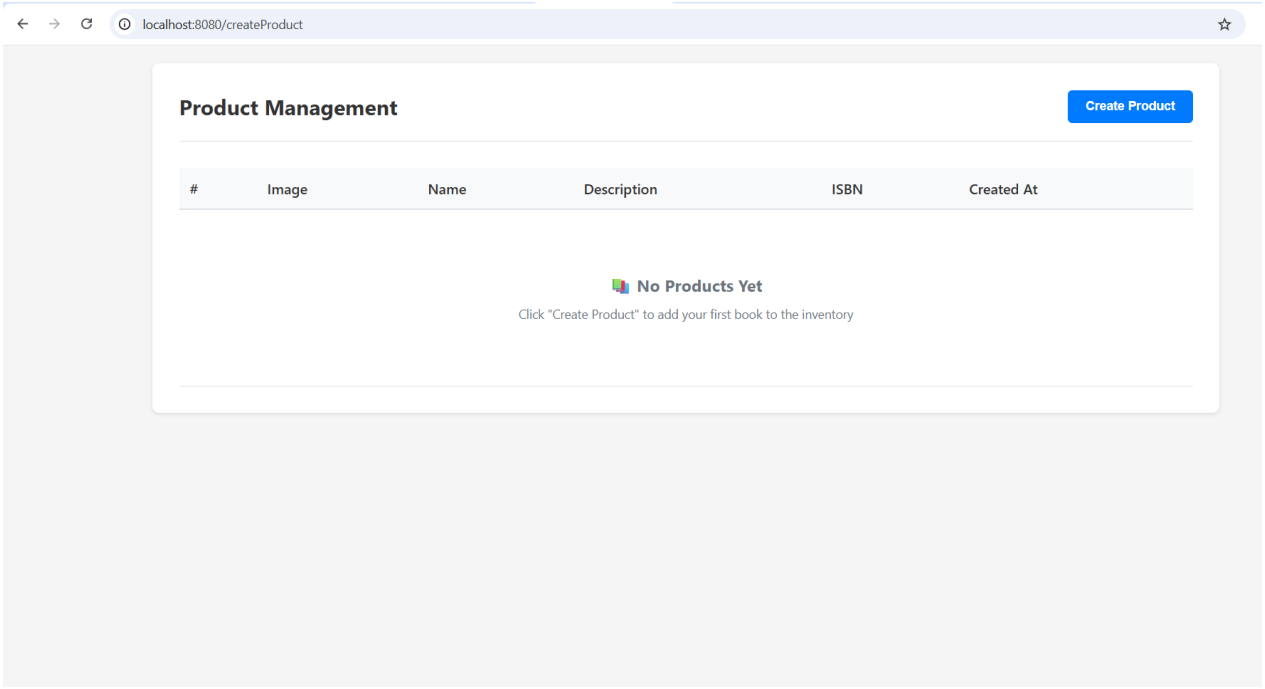
6.1 Overview
Developed a comprehensive product management interface for managing books in the library inventory.

6.2. Components Created
Files Created:





6.3 Features Implemented



6.4. Backend Functionality:

- **Data Retrieval :** Fetches all books and categories from the database
- **Validation:**
  - Client-side validation using HTML5 and JavaScript
  - Server-side validation for all required fields
  - ISBN uniqueness check to prevent duplicates
- **Database Integration:**
  - Saves books with encrypted data
  - Automatically links books to selected categories
  - Displays category names from `book_categories` table

6.5. **Technical Details:**

- **Endpoint:**`GET/POST /createProduct`

- **Security:** Public access (no authentication required for development)
- **Design:**Responsive table layout with clean, modern styling
- **Error Handling:** User-friendly error messages with form data preservation

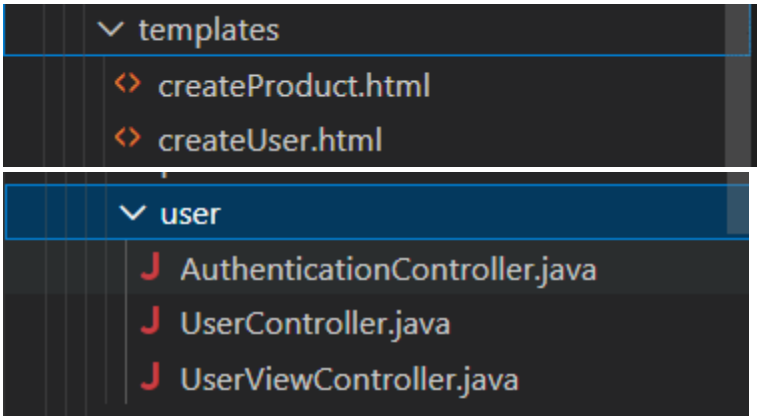
7. User Management

7.1 Overview

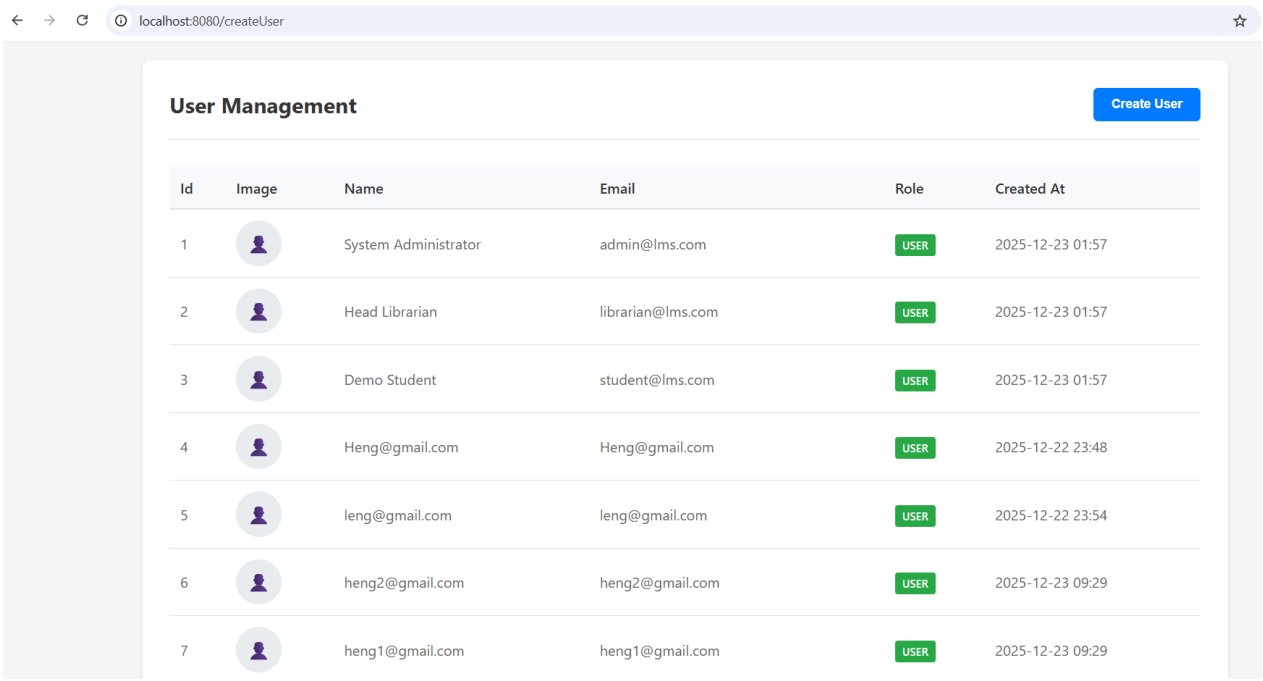
Developed a user management interface for creating and viewing system users.

7.2 Components Created

Files Created:



7.3 Features Implemented



7.4. Backend Functionality:

- **Data Retrieval:** Fetches all users from the database

- **Validation:**
  - Client-side validation for all fields
  - Server-side validation with detailed error messages
  - Email uniqueness check
  - Password strength requirement (minimum 6 characters)
- **Security:**
  - Password encryption using BCryptPasswordEncoder
  - Default role assignment (USER)
  - Account enabled by default

7.5. **Technical Details:**

- **Endpoint:** `GET/POST /createUser`
- **Security:** Public access (configured for development)
- **Design:** Consistent with product management UI
- **Role Display:** Color-coded badges (red for ADMIN, green for USER)

8. Testing & Deployment

8.1 Testing Performed

- Form submission with valid data
- Form submission with invalid data
- Duplicate prevention (ISBN and Email)
- Client-side validation
- Server-side validation
- Modal open/close functionality
- Table data display
- Empty state display

9 . Conclusion

In Week 3, the Library Management System has been successfully analyzed and modeled using an ER diagram. All major entities and their relationships have been clearly identified, ensuring data integrity, scalability, and maintainability. Also, Successfully implemented two complete CRUD interfaces using Thymeleaf for the Library Management System. Both interfaces follow consistent design patterns, provide excellent user experience, and integrate seamlessly with the existing Spring Boot application. The implementation demonstrates modern web development practices with proper validation, security measures, and responsive design. This ER design provides a strong foundation for database implementation in later weeks, as well as for building application logic such as borrowing rules, fine calculation, and notifications.

10 . Reference

- **Dependencies reference**: https://mvnrepository.com/artifact/org.springframework.boot
- **Software Development Life Cycle (SDLC).** *Requirement Analysis and Design Phase Protocols.*
- **Library Management Standards.** *ISBN Data Structures and Classification Systems.* (Used as a reference for the Book and BookCategory entity attributes).

- **Java 21 Documentation**: https://docs.oracle.com/en/java/javase/21/
- **Maven Documentation:** https://maven.apache.org/guides/
- **Lombok Documentation:** https://projectlombok.org/features/
- **Elmasri, R., & Navathe, S. B. (2015).** *Fundamentals of Database Systems (7th Edition).* Pearson. (Used for ER modeling principles and normalization techniques)
- **Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019).** *Database System Concepts (7th Edition).* McGraw- Hill. (Applied for understanding transaction management and relationship mapping).
- **Martin, R. C. (2017).** *Clean Architecture: A Craftsman's Guide to Software Structure and Design.* Prentice Hall. (Used for designing the AuditLog and Notification separation of concerns).