

Domain

```

classDiagram
    class Loan {
        o id: Long
        o book: Book
        o member: Member
        o loanDate: LocalDate
        o expectedReturnDate: LocalDate
        o actualReturnDate: LocalDate
        o status: LoanStatus
        o loan()
        o Loan(book: Book, member: Member, loanDays: int)
        o getID(): Long
        o setID(id: Long): void
        o getBook(): Book
        o setBook(book: Book): void
        o getMember(): Member
        o setMember(member: Member): void
        o getLoanDate(): LocalDate
        o setLoanDate(date: LocalDate): void
        o getExpectedReturnDate(): LocalDate
        o setExpectedReturnDate(date: LocalDate): void
        o getActualReturnDate(): LocalDate
        o setActualReturnDate(date: LocalDate): void
        o getStatus(): LoanStatus
        o setStatus(status: LoanStatus): void
        o isOverdue(): boolean
        o overdueDays(): int
        o equals(obj: Object): boolean
        o hashCode(): int
        o toString(): String
    }
    class Book {
        o id: Long
        o isbn: String
        o title: String
        o publisher: String
        o year: Year
        o availableStock: int
        o totalStock: int
        o Book()
        o Book(isbn, title, author, publisher, year, totalStock)
        o getID(): Long
        o setID(id: Long): void
        o getISBN(): String
        o setISBN(isbn: String): void
        o getTitle(): String
        o setTitle(title: String): void
        o getAuthor(): String
        o setAuthor(author: String): void
        o getPublisher(): String
        o setPublisher(publisher: String): void
        o getYear(): Year
        o setYear(year: Year): void
        o getAvailableStock(): int
        o setAvailableStock(availableStock: int): void
        o getTotalStock(): int
        o setTotalStock(totalStock: int): void
        o isAvailable(): boolean
        o equals(obj: Object): boolean
        o hashCode(): int
        o toString(): String
    }
    class Member {
        o id: Long
        o name: String
        o email: String
        o memberNumber: String
        o status: MemberStatus
        o registrationDate: LocalDate
        o role: UserRole
        o Member()
        o Member(name, email, memberNumber)
        o getID(): Long
        o setID(id: Long): void
        o getName(): String
        o setName(name: String): void
        o getEmail(): String
        o setEmail(email: String): void
        o getMemberNumber(): String
        o setMemberNumber(memberNumber: String): void
        o getStatus(): MemberStatus
        o setStatus(status: MemberStatus): void
        o getRegistrationDate(): LocalDate
        o setRegistrationDate(date: LocalDate): void
        o getRole(): UserRole
        o setRole(role: UserRole): void
        o isActive(): boolean
        o equals(obj: Object): boolean
        o hashCode(): int
        o toString(): String
    }
    class LoanStatus {
        ACTIVE
        RETURNED
        OVERDUE
    }
    class MemberStatus {
        ACTIVE
        INACTIVE
    }
    class UserRole {
        MEMBER
        LIBRARIAN
        ADMIN
    }
    Loan --> Book
    Loan --> Member
    Loan --> LoanStatus
    Member --> MemberStatus
    Member --> UserRole
  
```

The diagram illustrates the Domain model for a library system, showing the relationships between various entities and their attributes.

Loan Entity:

- Attributes: `id` (Long), `book` (Book), `member` (Member), `loanDate` (LocalDate), `expectedReturnDate` (LocalDate), `actualReturnDate` (LocalDate), `status` (LoanStatus).
- Operations: `loan()`, `Loan(book: Book, member: Member, loanDays: int)`, `getID()` (Long), `setID(id: Long)` (void), `getBook()` (Book), `setBook(book: Book)` (void), `getMember()` (Member), `setMember(member: Member)` (void), `getLoanDate()` (LocalDate), `setLoanDate(date: LocalDate)` (void), `getExpectedReturnDate()` (LocalDate), `setExpectedReturnDate(date: LocalDate)` (void), `getActualReturnDate()` (LocalDate), `setActualReturnDate(date: LocalDate)` (void), `getStatus()` (LoanStatus), `setStatus(status: LoanStatus)` (void), `isOverdue()` (boolean), `overdueDays()` (int), `equals(obj: Object)` (boolean), `hashCode()` (int), `toString()` (String).

Book Entity:

- Attributes: `id` (Long), `isbn` (String), `title` (String), `publisher` (String), `year` (Year), `availableStock` (int), `totalStock` (int).
- Operations: `Book()`, `Book(isbn, title, author, publisher, year, totalStock)`, `getID()` (Long), `setID(id: Long)` (void), `getISBN()` (String), `setISBN(isbn: String)` (void), `getTitle()` (String), `setTitle(title: String)` (void), `getAuthor()` (String), `setAuthor(author: String)` (void), `getPublisher()` (String), `setPublisher(publisher: String)` (void), `getYear()` (Year), `setYear(year: Year)` (void), `getAvailableStock()` (int), `setAvailableStock(availableStock: int)` (void), `getTotalStock()` (int), `setTotalStock(totalStock: int)` (void), `isAvailable()` (boolean), `equals(obj: Object)` (boolean), `hashCode()` (int), `toString()` (String).

Member Entity:

- Attributes: `id` (Long), `name` (String), `email` (String), `memberNumber` (String), `status` (MemberStatus), `registrationDate` (LocalDate), `role` (UserRole).
- Operations: `Member()`, `Member(name, email, memberNumber)`, `getID()` (Long), `setID(id: Long)` (void), `getName()` (String), `setName(name: String)` (void), `getEmail()` (String), `setEmail(email: String)` (void), `getMemberNumber()` (String), `setMemberNumber(memberNumber: String)` (void), `getStatus()` (MemberStatus), `setStatus(status: MemberStatus)` (void), `getRegistrationDate()` (LocalDate), `setRegistrationDate(date: LocalDate)` (void), `getRole()` (UserRole), `setRole(role: UserRole)` (void), `isActive()` (boolean), `equals(obj: Object)` (boolean), `hashCode()` (int), `toString()` (String).

LoanStatus Enum:

- Values: `ACTIVE`, `RETURNED`, `OVERDUE`.

MemberStatus Enum:

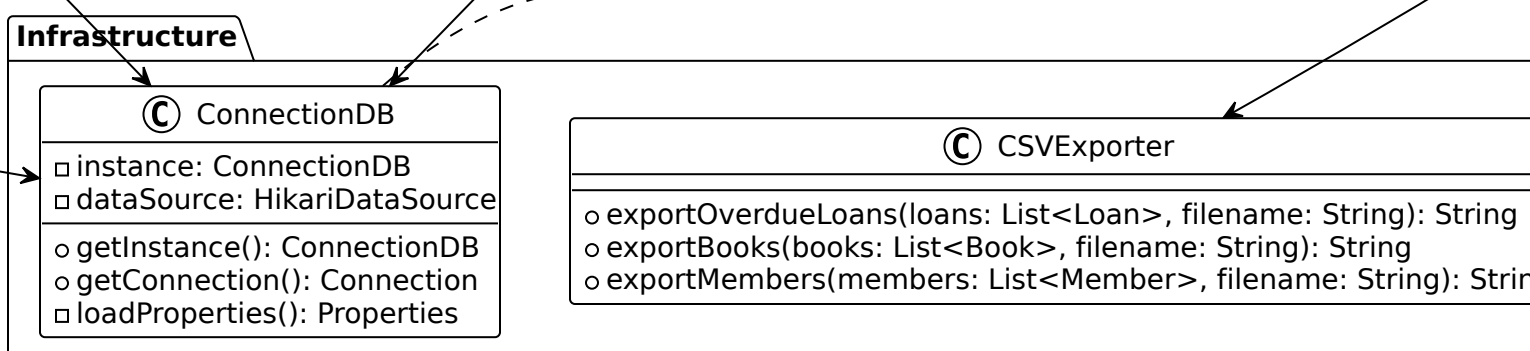
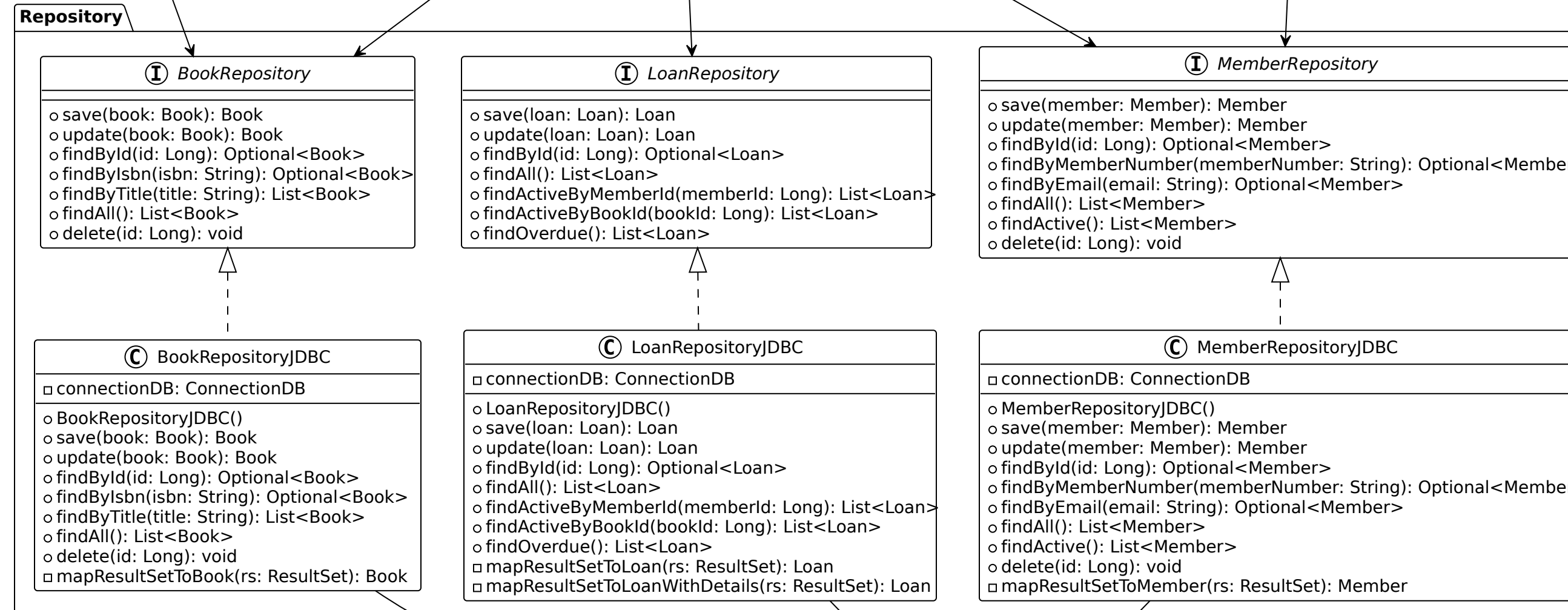
- Values: `ACTIVE`, `INACTIVE`.

UserRole Enum:

- Values: `MEMBER`, `LIBRARIAN`, `ADMIN`.

Relationships:

- Loan** is associated with **Book**, **Member**, and **LoanStatus**.
- Member** is associated with **MemberStatus** and **UserRole**.



- Manages loan operations:
 - Creates loans with validation
 - Processes book returns
 - Calculates fines (\$1/day)
 - Stock adjustments

Singleton pattern for database connection using HikariCP pool