# Library Management System Diagrams

This document contains the Data Flow Diagram (DFD), Entity-Relationship Diagram (ERD), and all required UML diagrams for the Library Management System (LMS).
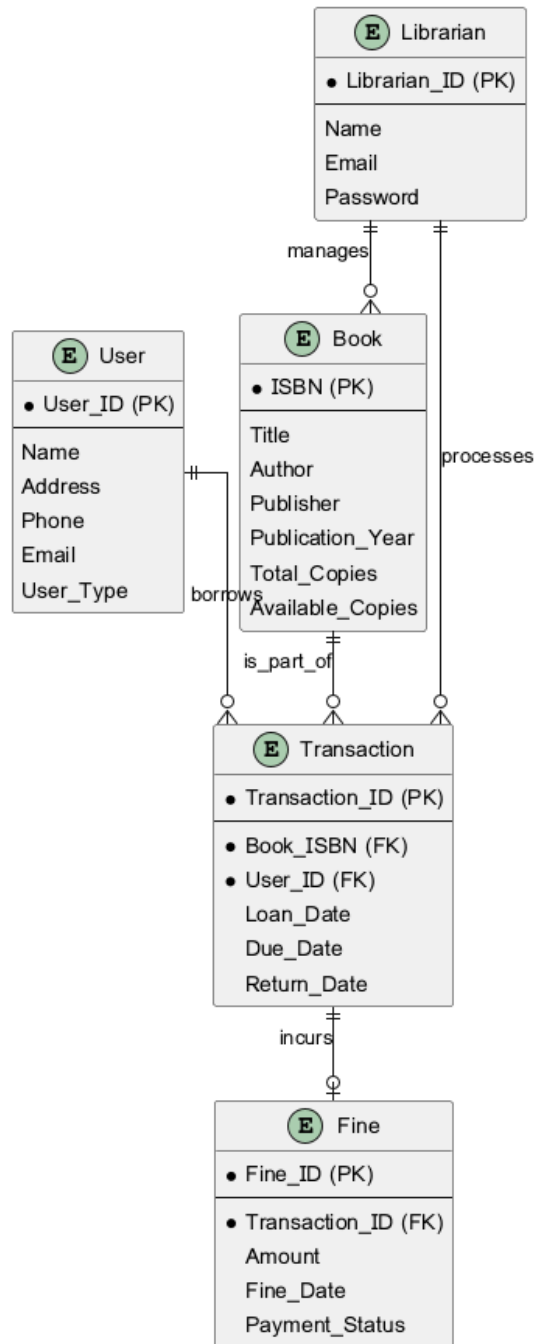
1. Data Flow Diagram (DFD) - Context Level

This diagram illustrates the Library Management System as a single process, showing its interaction with external entities (Library User and Librarian).
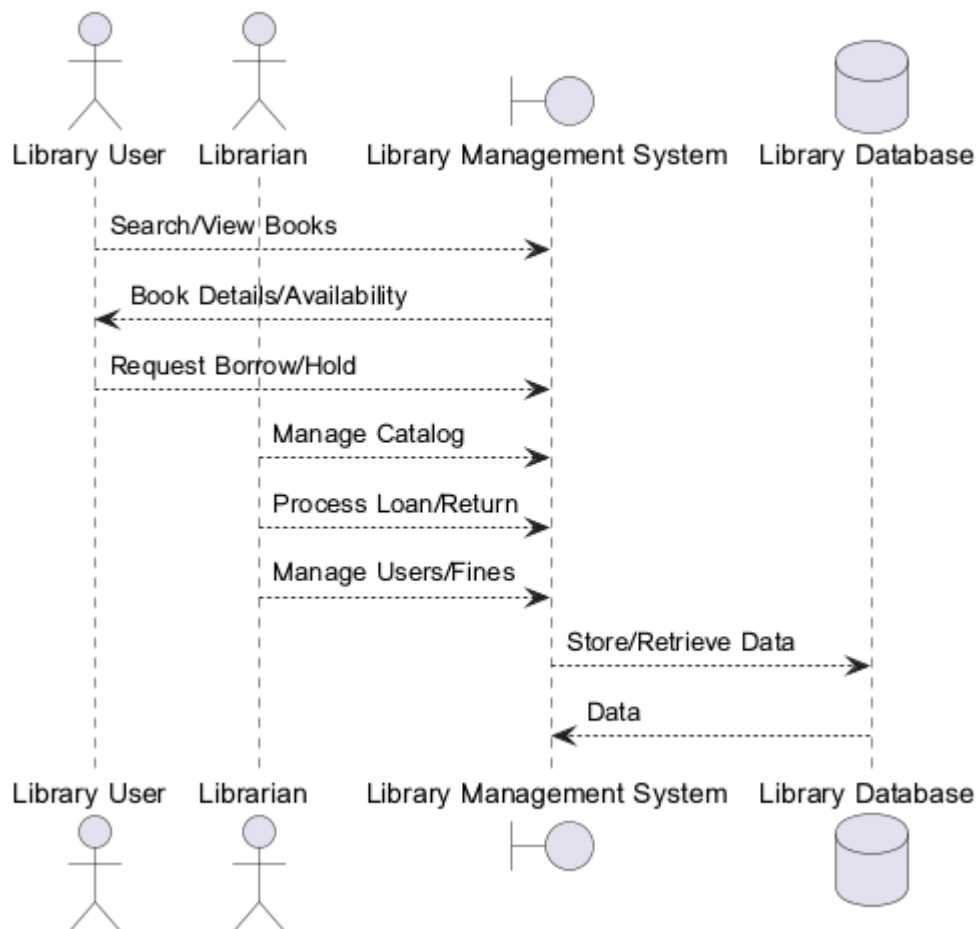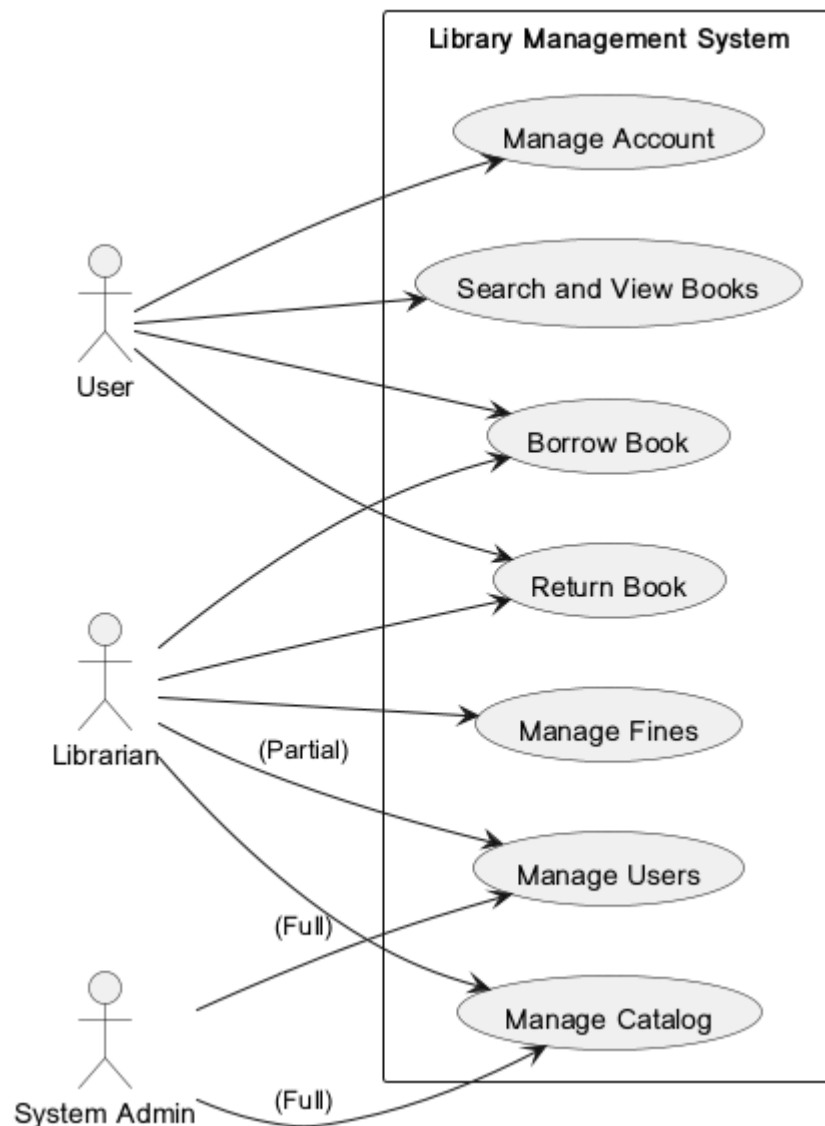
2. Entity-Relationship Diagram (ERD)

The ERD models the data structure of the system, showing the main entities and the relationships between them.

3. UML Use Case Diagram

The Use Case Diagram captures the functional requirements of the system and the interaction between the actors (User, Librarian, System Admin) and the system's use cases.

4. UML Sequence Diagram (Borrow Book)

This diagram shows the sequence of steps and interactions between objects to perform the "Borrow Book" use case.

Borrow Book Sequence Diagram
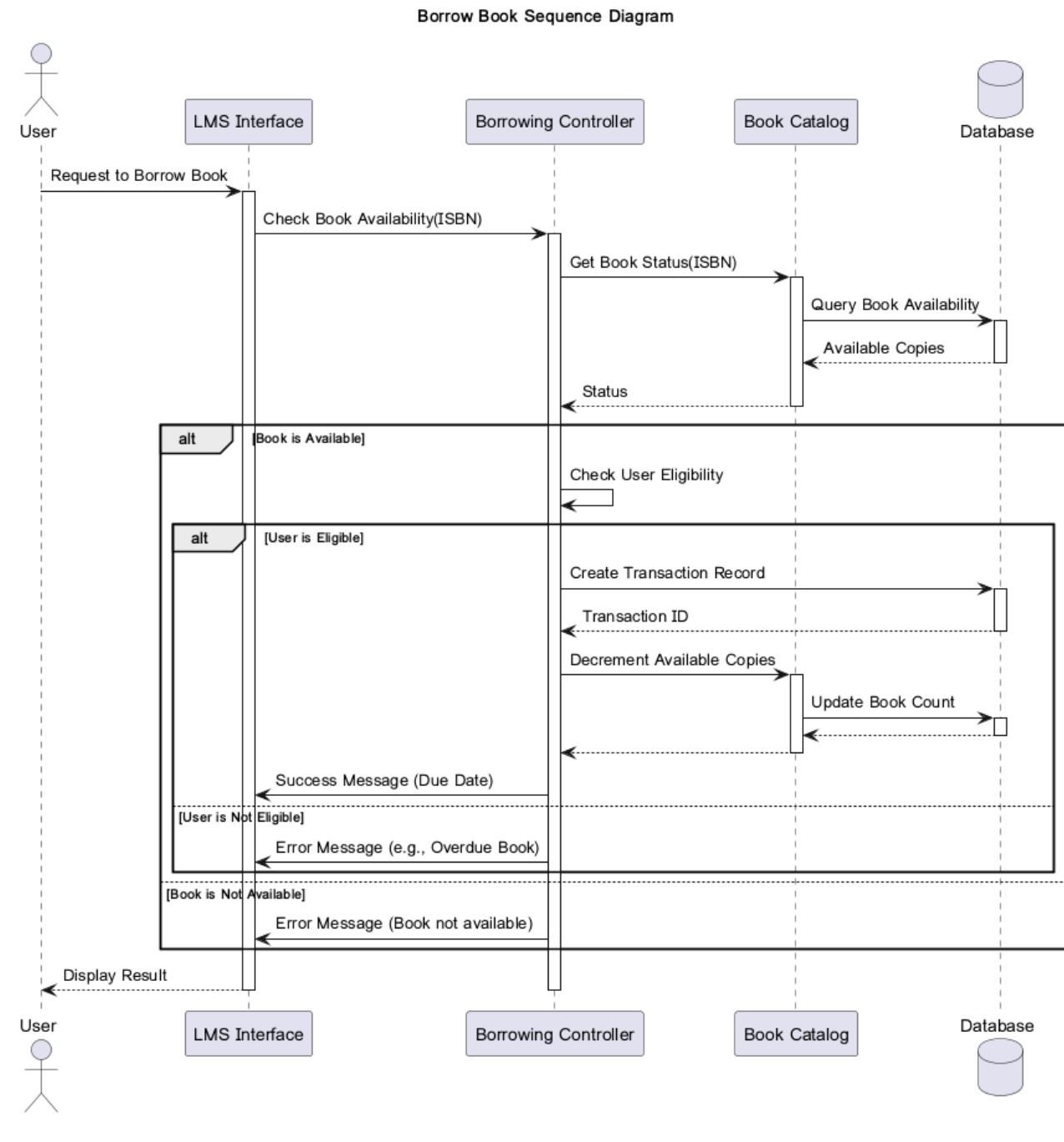
5. UML Activity Diagram (Return Book)

The Activity Diagram illustrates the flow of control and activities involved in the "Return Book" process, including the logic for fine calculation.

Return Book Activity Diagram
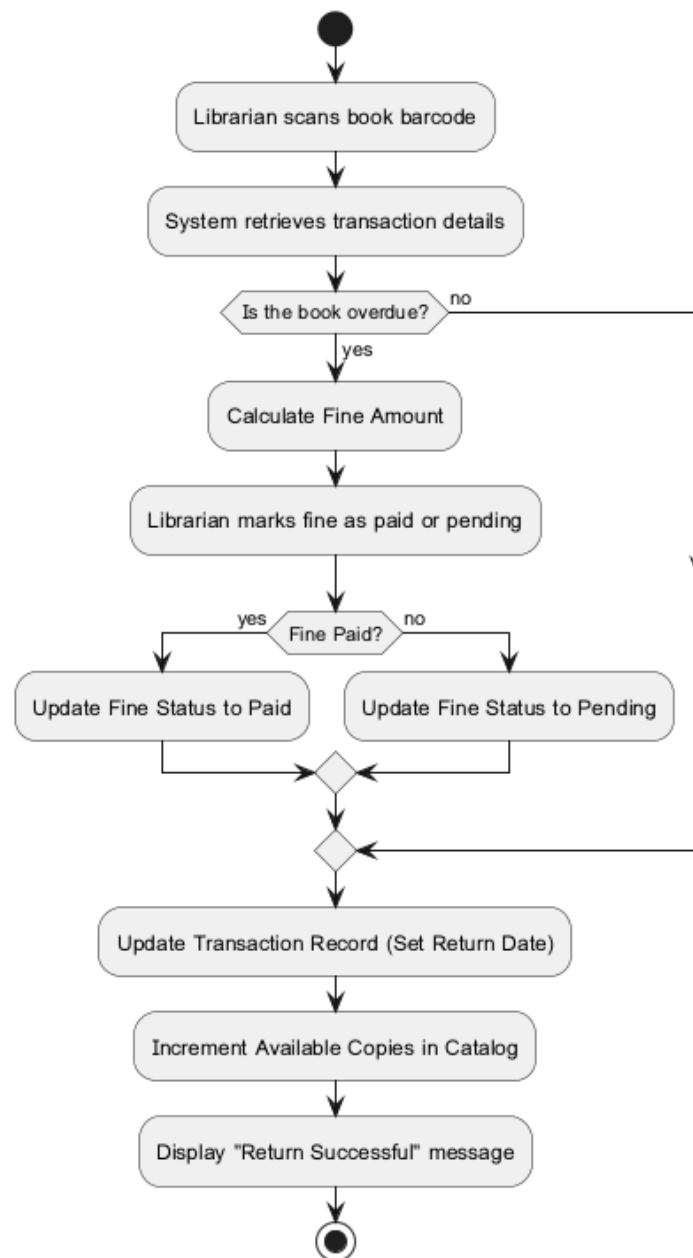
6. UML Class Diagram

The Class Diagram provides a static view of the system's structure, showing the classes, their attributes, methods, and the relationships between them.

Library Management System Class Diagram

Librarian
- □ librarianID : String
- □ name : String
- □ email : String

- ● processLoan(User user, Book book)
- ● processReturn(User user, Book book)
- ● manageCatalog()

User
- □ userID : String
- □ name : String
- □ email : String
- □ userType : String

- ● checkEligibility() : boolean
- ● getBorrowedBooks()

Book
- □ ISBN : String
- □ title : String
- □ author : String
- □ publisher : String
- □ publicationYear : int
- □ totalCopies : int
- □ availableCopies : int

- ● getDetails()
- ● updateCopies(int change)

Transaction
- □ transactionID : String
- □ loanDate : Date
- □ dueDate : Date
- □ returnDate : Date

- ● calculateFine() : double

Fine
- □ fineID : String
- □ amount : double
- □ paymentStatus : String

- ● processPayment()

1 ▼ manages 0..*

1 ▼ processes

1 ◄ borrows 0..*

1 ▼ is_part_of 0..*

0..*

1 ▼ incurs 0..1