



Software Requirements Specification (SRS) – Database Design Library Management System (LMS)

1. Problem

The Library Management System (LMS) automates the management of books, users (customers & employees), categories, branches, and transactions (borrowing & returning). The new design uses **inheritance (User → Customer / Employee)** to avoid data duplication and ensure scalability.

2. System Requirements

- **Functional Requirements**

The Library Management System should support the following functionalities:

1. Customers:

- Search Books
- Borrow Books
- Return Books
- List Available Books

2. Employees:

- Add Books
 - Edit Books
 - Remove Books
 - Issue Books
 - Return Books
-

3. Entities and Attributes

1. Books

- book_id (PK)
- title
- ISBN
- No_copies
- No_pages
- Language
- Released year

- category_id (FK → Category.category_id)
 - branch_id (FK → Branch.branch_id)
 - publisher_id(FK → Publisher.publisher_id)
 - author_id(FK → Author.author_id)
-

2. Category

- category_id (PK)
 - category_name
-

3. User

- user_id (PK)
- first_name
- last_name
- email
- phone_no
- username
- password

4. Customer

- customer_id (PK, FK -> User.user_id)
 - address
-

5. Employees

- employee_id (PK, FK -> User.user_id)
 - role
-

6. Branch

- branch_id (PK)
 - branch_name
 - location
-

7. Borrow

- issue_id (PK)
 - book_id (FK → Books.book_id)
 - customer_id (FK → Customer.customer_id)
 - issue_date
 - due_date
 - return_date
 - status
-

8. Publisher

- publisher_id (PK)
 - publisher_name
-

9. Author

- author_id (PK)
 - author_name
-

4. Relationships & Cardinality

1. Books – Category

Relationship: Each book belongs to one category.

A category can contain many books.

Cardinality: Category (1) — (M) Books

2. Books – Branch

Relationship: Each book is stored in one branch.

A branch can hold many books.

Cardinality: Branch (1) — (M) Books

3. Books – Publisher

Relationship: A book is published by one publisher.

A publisher can publish many books.

Cardinality: Publisher (1) — (M) Books

4. Books – Author

Each book is written by one author.
An author can write many books.
Cardinality: Author (1) — (M) Books

5. User – Customer

Relationship: A customer is a user (specialization).
Cardinality: User (1) — (1) Customer

6. User – Employee

Relationship: An employee is a user (specialization).
Cardinality: User (1) — (1) Employee

7. Borrow – Books

Relationship: A borrow record is for one book.
A book can appear in many borrow records (different times).
Cardinality: Book (1) — (M) Borrow

8. Borrow – Customer

Relationship: A borrow record belongs to one customer.
A customer can have many borrow records.
Cardinality: Customer (1) — (M) Borrow

9. Borrow – Employee

Relationship: A borrow record is issued/handled by one employee.
An employee can process many borrow records.
Cardinality: Employee (1) — (M) Borrow

5. Features of Users

1. User (General)

Common attributes: first_name, last_name, email, phone_no, username, password.
Base entity → can specialize into Customer or Employee.

2. Customer (Inherits from User.)

Features:

- Search books.
 - Borrow books.
 - Return books.
 - List available books.
-

3. Employee (Inherits from User)

Features (normal Employee):

- Add books.
- Edit books.
- Remove books.
- Issue books to customers.
- Handle book returns.

Features (Manager role):

- Manage categories.
- Manage branches.
- View reports.