Problem Objective

Managing library resources efficiently while ensuring security and streamlined user interactions is challenging. Traditional systems often struggle with authentication security, flexible book browsing, and effective admin-user communication. Automated notifications may lack personalization, leading to inefficiencies in overdue book handling and user engagement.

The objective of this project is to design and develop a **Library Management System (LMS)** that enhances security, streamlines book and user management, and improves admin oversight. The system should allow:

- **Secure user authentication** with two-factor authentication (2FA) via email OTP.
- **Admins** to manage books, update user accounts, and enforce security token verification for critical actions.
- **Manual notification system** for overdue books, renewals, and other updates.
- **Flexible book browsing** based on title, author, ISBN, or genre for easy discovery.
- **User engagement features**, including borrowing history, cart management, book requests, and feedback submission.
- **Admin access to user feedback** for continuous service improvement and data-driven decisions.

System Specifications:

- **User Authentication:**
    - **Secure Authentication**: Users will authenticate with their credentials (username, password) along with **two-factor authentication (2FA)** via email OTP.
    - **Role-based Access Control**: Different roles (Admin, Library Staff, User) will have specific access to resources and operations.
- **Admin Features:**
    - **Library Management**: Admin can manage library details, including assigning staff and overseeing library operations.
    - **Book Management**: Admin has complete control over adding, updating, and deleting book details.
    - **User Management**: Admin can create, update, and delete user accounts, manage user roles, and oversee overdue notifications.
    - **Security**: Admin can enforce security token verification for critical actions, such as deleting books or updating library settings.
- **Library Staff Features:**
    - **Book Maintenance**: Library staff can manage book details and availability but cannot delete books or perform other high-security tasks.
    - **User Support**: Library staff can assist users with their book requests, queries, and reservations.
- **User Features:**
    - **Personalized Book Browsing**: Users can search for books by title, author, ISBN, or genre.

    - **Cart Management**: Users can add books to their cart and manage the list before borrowing or reserving.
    - **Borrowing and Reservations**: Users can borrow books for a limited time, with options to reserve unavailable books.
    - **Book Feedback and Ratings**: Users can rate books and provide feedback on their experiences.
    - **Overdue Notifications**: Users receive automated notifications about overdue books, upcoming renewals, or changes in availability.
- **Notifications:**

    - **Manual Overdue Notification System**: Users will receive manual notifications regarding overdue books and renewal reminders via email.
- **System Security and Data Privacy:**

    - **Data Encryption**: Sensitive user data (e.g., personal information, passwords) will be encrypted.
    - **Access Logs**: All admin and staff activities will be logged for auditing and security purposes.

Assumptions:

- **Books and User Records:** It is assumed that all books in the system are linked to at least one publisher and one author. Users must be registered to borrow or reserve books.
- **Overdue Handling**: If a book is not returned on time, an automated reminder will be sent to users, but manual follow-up may still be necessary.
- **User Data**: User data such as feedback, borrowing history, and cart contents will be stored securely.

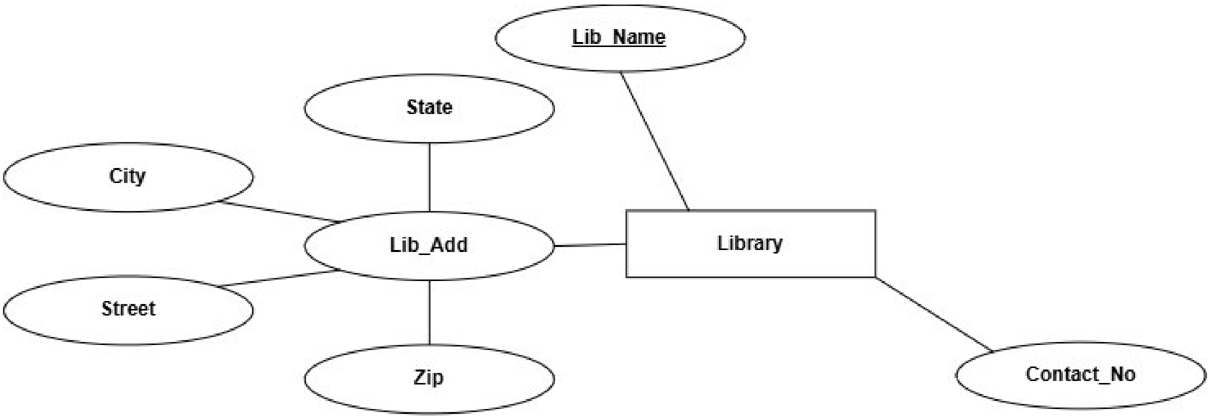
- **Publisher and Author Info**: A book can have multiple authors, but every book must have a publisher.

Steps Involved in Designing the System

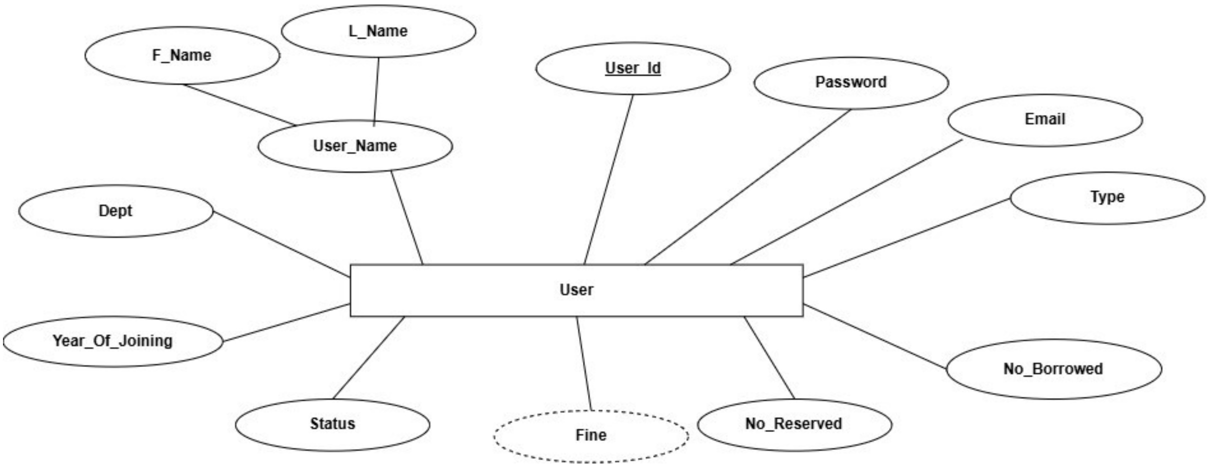
- Identification of Entities
- Drawing the ER Diagram
- Mapping ER Diagram to Relational Schema
- Identifying Functional Dependencies (FDs)

Identification of Entities

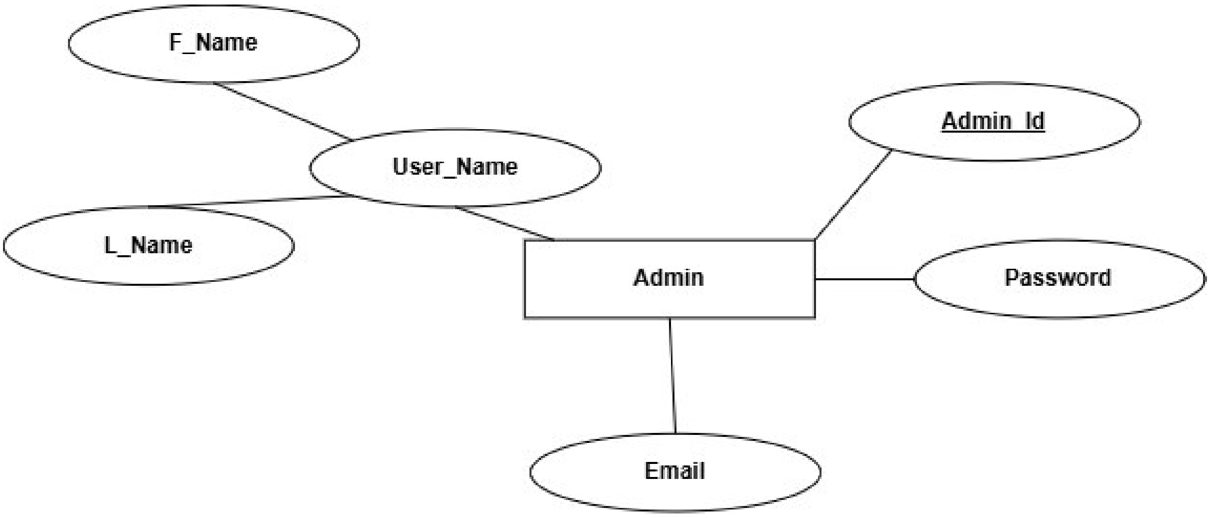
- **Library Entity**



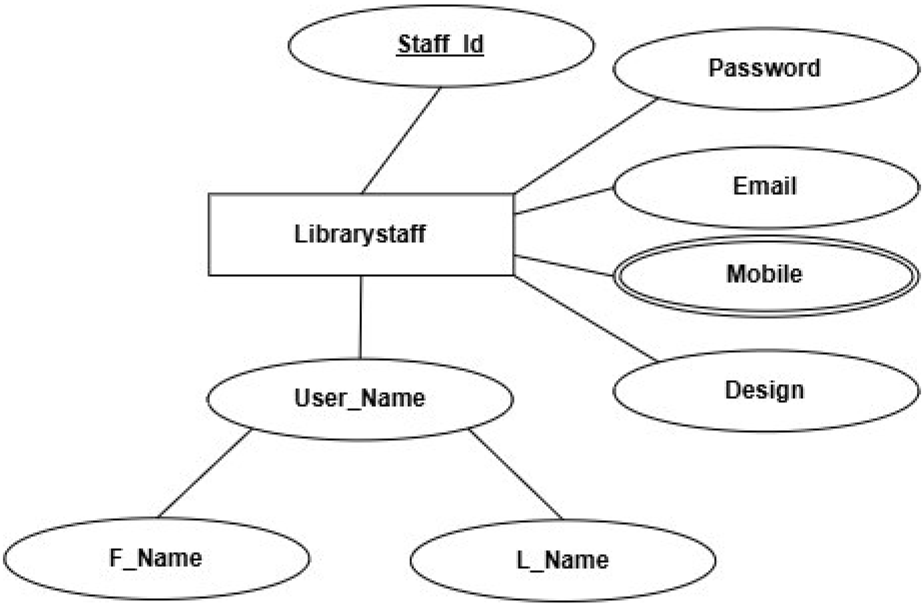
- **User Entity**



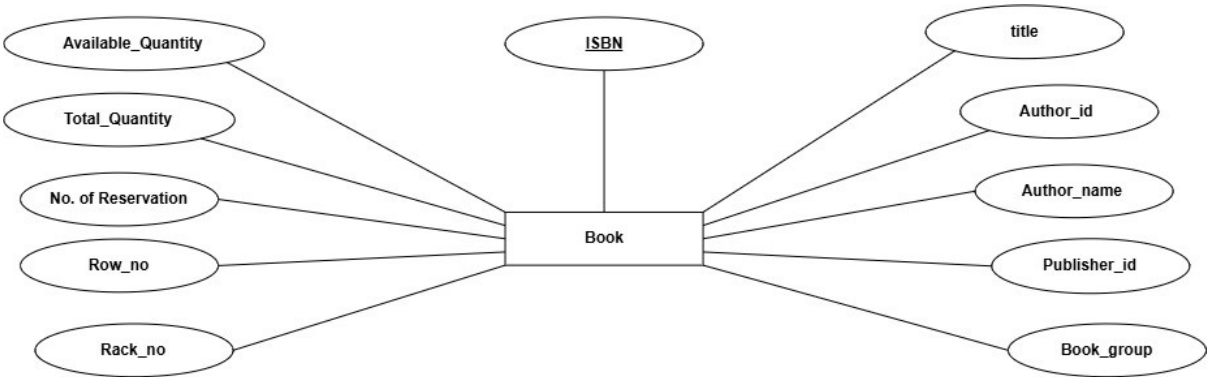
- **Admin Entity**



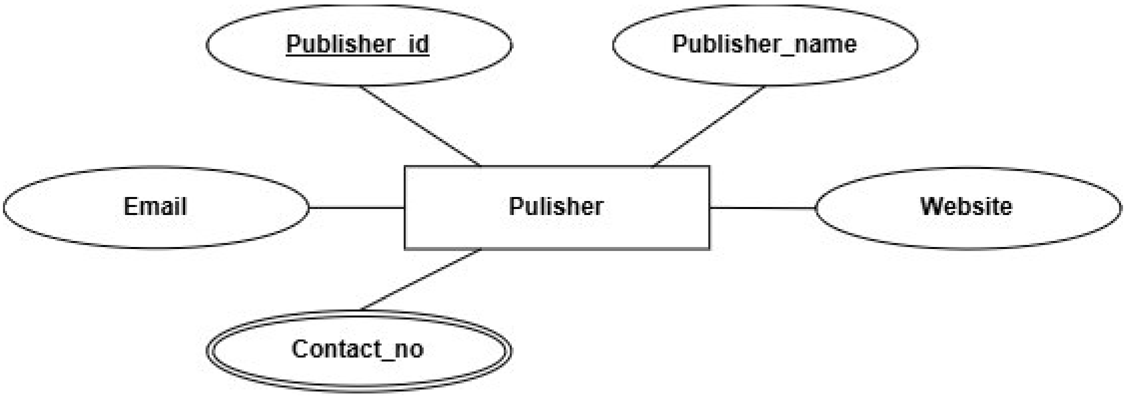
- **Library Staff Entity**



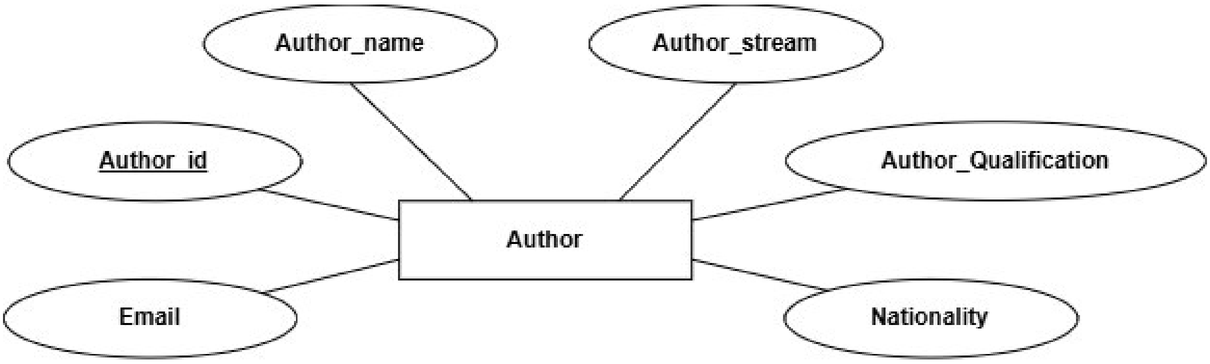
- **Book Entity**



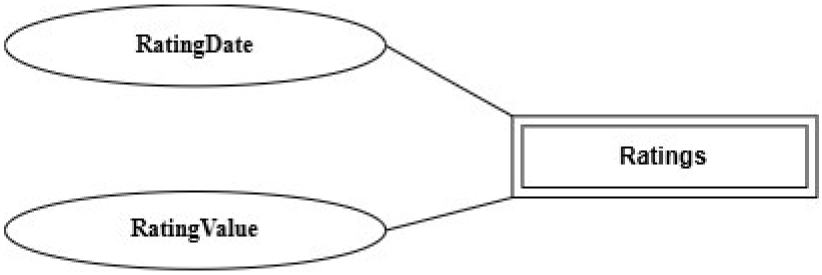
- **Publisher Entity**



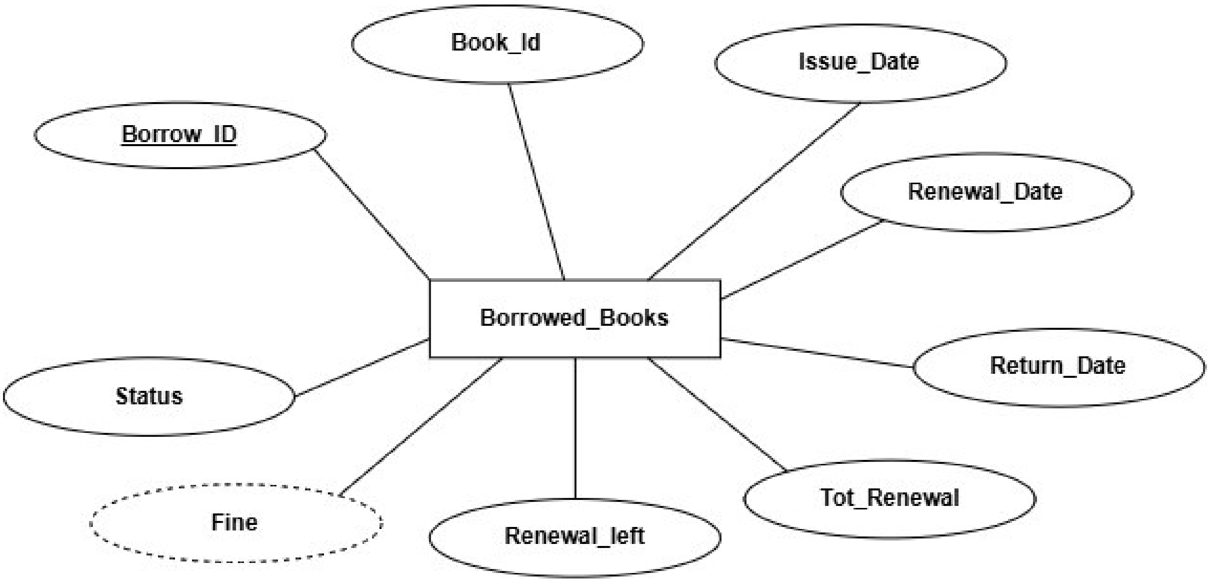
- **Author Entity**



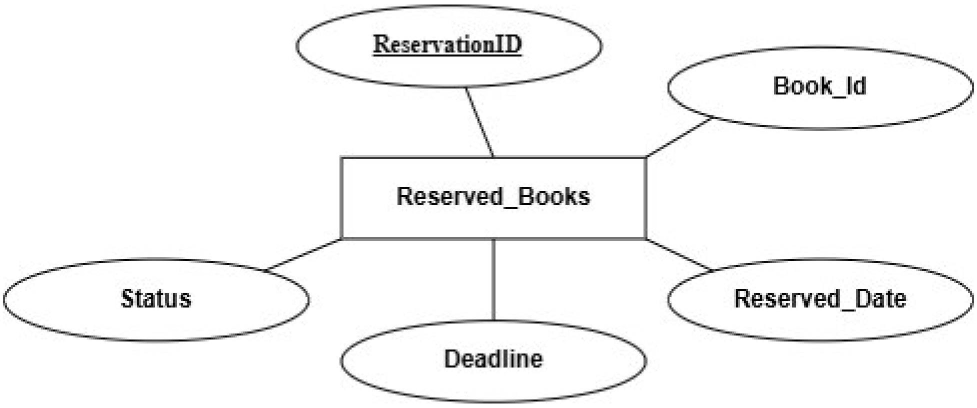
- **Ratings Entity**



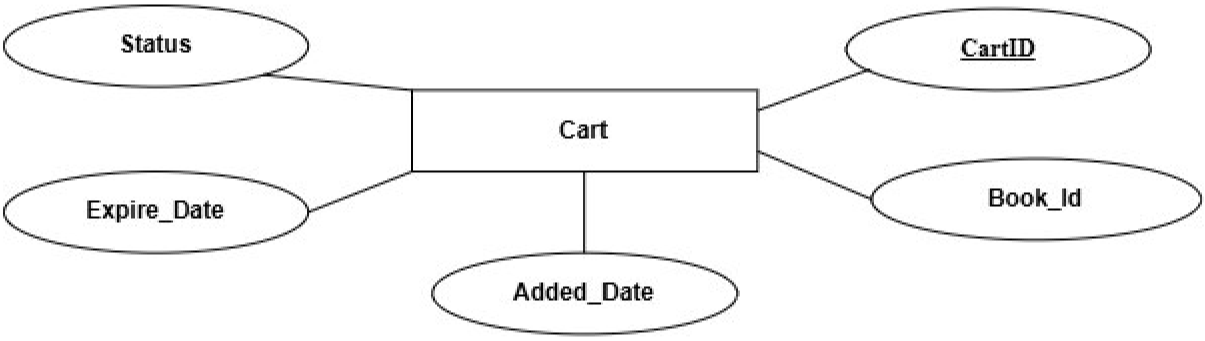
- **Borrowed Books Entity**



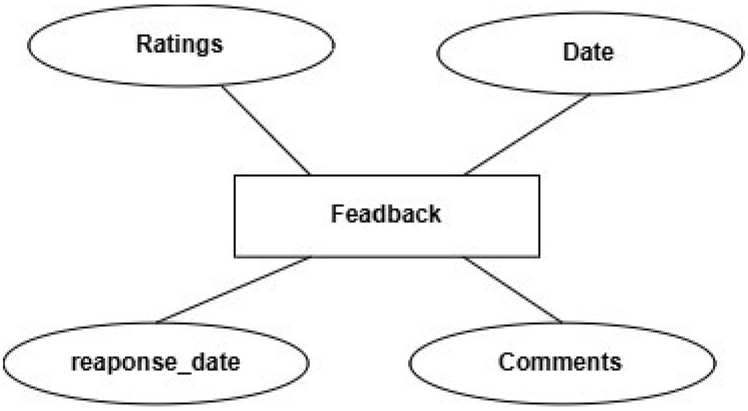
- **Reserved Books Entity**



- **Cart Collections Entity**



- **Feedbacks Entity**



Relationship Between Entities

1. Library contains Books
2. Book is published by Publisher
3. Book is written by Author
4. Library is managed by Admin
5. Library employs Library Staff
6. Book has Ratings
7. Library registers Users
8. Admin maintains Books
9. Library Staff maintains Books
10. Admin supervises Library Staff
11. User sends requests to Library Staff
12. User rates Books
13. User provides Feedback for Library
14. User requests New Books
15. User adds Books to Cart Collection
16. User reserves Books
17. User borrows Books

Cardinality Ratio and Participation (Partial/Total)

1. Library - Book (contains)

- **Cardinality**: 1 : N
- **Participation**:
  - **Library**: **Total** (Every book must belong to a library).
  - **Book**: **Partial** (A book may not belong to a library yet).

2. Book - Publisher (published by)

- **Cardinality**: N : 1
- **Participation**:
  - **Book**: **Total** (Every book must have a publisher).
  - **Publisher**: **Partial** (A publisher may not have published any books).

3. Book - Author (written by)

- **Cardinality**: N : M
- **Participation**:
  - **Book**: **Total** (Every book must have at least one author).
  - **Author**: **Partial** (An author may not have written any books).

4. Library - Admin (managed by)

- **Cardinality**: 1 : 1
- **Participation**:
  - **Library**: **Total** (Every library must have an admin).
  - **Admin**: **Partial** (An admin may not manage a library).

5. Library - Library Staff (employs)

- **Cardinality**: 1 : N
- **Participation**:
  - **Library**: **Total** (Every library has at least one staff member).
  - **Library Staff**: **Partial** (A staff member may not be employed in a library).

6. Book - Rating (has)

- **Cardinality**: 1 : N
- **Participation**:
  - **Book**: **Partial** (A book may not have ratings).
  - **Rating**: **Total** (Each rating belongs to a book).

7. Library - User (registers)

- **Cardinality**: 1 : N
- **Participation**:
  - **Library**: **Total** (A user must be registered to the library).
  - **User**: **Partial** (A library may not have registered users).

8. Admin - Book (maintains)

- **Cardinality**: 1 : N
- **Participation**:
  - **Admin**: **Partial** (An admin may not maintain any books).
  - **Book**: **Total** (Every book must be maintained by an admin).

9. Library Staff - Book (maintains)

- **Cardinality**: 1 : N
- **Participation**:
  - **Library Staff**: **Partial** (Not every staff member maintains books).
  - **Book**: **Total** (Every book must be maintained by staff).

10. Admin - Library Staff (supervises)

- **Cardinality**: 1 : N
- **Participation**:
  - **Admin**: **Total** (An admin supervises at least one staff).
  - **Library Staff**: **Partial** (A staff member may not be supervised).

11. User - Library Staff (sends requests to)

- **Cardinality**: N : N
- **Participation**:
  - **User**: **Partial** (A user may not send any requests).

    - **Library Staff**: **Partial** (A staff member may not receive any requests).

12. User - Book (rates)

- **Cardinality**: N : N
- **Participation**:
    - **User**: **Partial** (A user may not rate any books).
    - **Book**: **Partial** (A book may not have any ratings).

13. User - Feedback (provides)

- **Cardinality**: 1 : N
- **Participation**:
    - **User**: **Partial** (A user may not provide feedback).
    - **Feedback**: **Total** (Every feedback must belong to a user).

14. User - Book (requests new book)

- **Cardinality**: N : 1
- **Participation**:
    - **User**: **Partial** (A user may not request new books).
    - **Book**: **Partial** (A book may not have any requests yet).

15. User - Cart Collection (adds books to)

- **Cardinality**: 1 : N
- **Participation**:
    - **User**: **Partial** (A user may not have a cart collection).
    - **Cart Collection**: **Total** (Every cart collection belongs to a user).

16. User - Reservation (reserves books)

- **Cardinality**: 1 : N
- **Participation**:

    - **User**: **Partial** (A user may not reserve any books).
    - **Reservation**: **Total** (Every reservation must be linked to a user).

17. User - Borrowed Books (borrows books)

- **Cardinality**: 1 : N
- **Participation**:
    - **User**: **Partial** (A user may not borrow any books).
    - **Book**: **Total** (Every borrowed book is linked to a user).

18. Borrowed Books → Books (Borrowed Book)

- **Cardinality**: 1 : M
- **Participation**:
    - **Borrowed Books**: **Fully Participatory** (Every borrowed book must be linked to a book).
    - **Books**: **Partial** (A book may or may not be borrowed; it can exist without being borrowed).

19. Reserved Books → Books (Reserve Book)

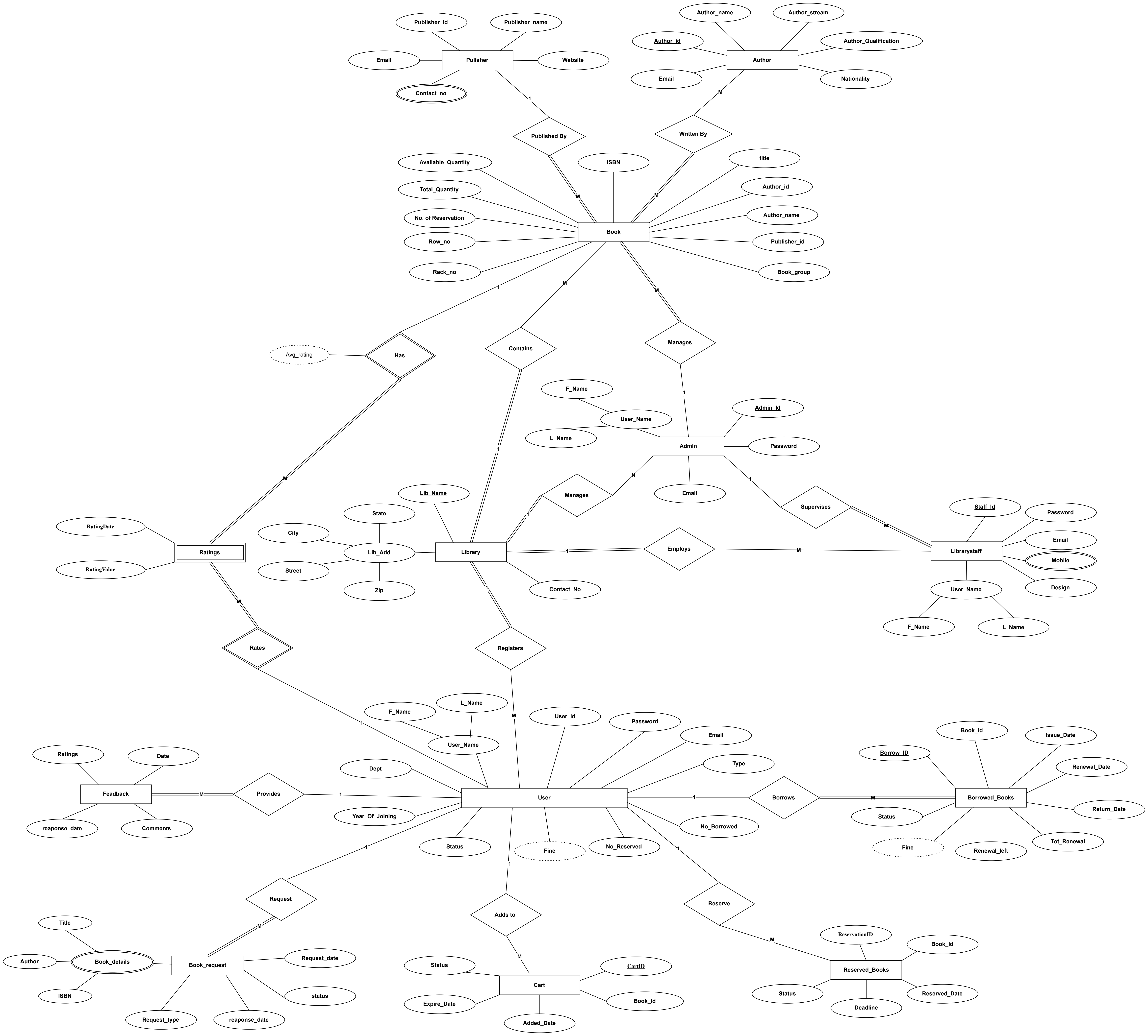
- **Cardinality**: 1 : M
- **Participation**:
    - **Reserved Books**: **Fully Participatory** (Every reservation must be linked to a book).
    - **Books**: **Partial** (A book may or may not be reserved; it can exist without being reserved).

20. Cart Collection → Books (Add to Cart)

- **Cardinality**: M : M
- **Participation**:

  - **Cart Collection**: **Fully Participatory** (Every cart must contain at least one book).
  - **Books**: **Partial** (A book may or may not be in a user's cart; it can exist without being in any cart).

ER DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM



Mapping ER Diagram to Relational Schema

Rules used for conversion from ER to Relational Schema

• Entity to Table:

Each entity (Customer, Restaurant, Item, Order, etc.) becomes a table with its attributes as columns.

• Simple Attributes:

Directly stored as columns in their respective tables (e.g., Customer Name, Address in Customer table).

• Composite Attributes:

Split into individual columns (e.g., Address is broken into Street, City, State, Zip).

• Multivalued Attributes:

Stored in a separate table. Example: If a customer can have multiple phone numbers, create a CustomerPhone table with CustomerID as foreign key.

• Primary Key:

Each table has a primary key to uniquely identify records.

• 1:1 Relationship:

Foreign key can be added to either side (based on access needs) or merged if they always exist together.

• 1:N Relationship:

Primary key of the one side becomes foreign key on the many side.

• M:N Relationship:

Create separate table with both entity primary keys as foreign keys.

• Weak Entity:

Create a table with primary key including the owner entity's primary key + weak entity's discriminator.

Rationale for Primary and Foreign Key Selection

Entities and Their Attributes

Library (**LibraryName**, Location, ContactInfo)

Book (ISBN, Title, AuthorID, AuthorName, PublisherID, LibraryName, BookGroup, TotalCopies, AvailableCopies,NoOf Reservation,RowNo,RackNo)

Publisher (PublisherID, PublisherName, ContactInfo, Email,WebsiteLink)

Author (**AuthorID**, AuthorName, AuthrStream, AuthorQualification,Nationality,Email)

Admin (**AdminID**, PassWord,AdminName, Email, ContactInfo, LibraryName)

LibraryStaff (**StaffID**, PassWord, StaffName, Role, Email, ContactInfo, LibraryName, SupervisorID)

Users (**UserID**, PassWord, MembershipType, UserName, LibraryName, Department, JoinedYear, Email, NoBorrowedBooks, NoReservedBooks, Fine,Status)

Rating (**UserID**, **BookID**, RatingValue, RatingDate)

Feedback (**UserID**, **DateOfGiven**, Comments, Ratings,Response_Date)

RequestNewBook (**UserID**, BookDetails, RequestType, **RequestDate**,ResponseDate,Status)

CartCollection (**CartID**, UserID, BookID, AddedDate,ExpireDate,Status)

Reservation (**ReservationID**, UserID, BookID, ReservationDate, DeadLine, Status)

BorrowedBooks (**BorrowID**, UserID, BookID, IssueDate,RenewalDate, ReturnDate,TotalRenewals,RenewalsLeft,Fine, Status)

Primary and Foreign Key Mapping with Descriptions

1. Library

- **Primary Key**: LibraryName (Each library has a unique name that identifies it.)

- **Foreign Keys**: None

2. Book

- **Primary Key**: ISBN (Each book is uniquely identified by its ISBN.)

- **Foreign Keys**: PublisherID (Links book to its publisher), AuthorID (Links book to its author), LibraryName (Indicates which library holds the book.)

3. Publisher

- **Primary Key**: PublisherID (Each publisher is uniquely identified by an ID.)

- **Foreign Keys**: None

4. Author

- **Primary Key**: AuthorID (Each author is uniquely identified by an ID.)

- **Foreign Keys**: None

5. Admin

- **Primary Key**: AdminID (Each admin has a unique identifier.)
- **Foreign Keys**: LibraryName (Indicates which library the admin manages.)

6. LibraryStaff

- **Primary Key**: StaffID (Each library staff member is uniquely identified.)
- **Foreign Keys**: LibraryName (Links staff to a specific library), SupervisorID (References AdminID, indicating the admin who supervises the staff.)

7. User

- **Primary Key**: UserID (Each user is uniquely identified.)
- **Foreign Keys**: LibraryName (Indicates which library the user is registered under.)

8. Rating

- **Primary Key**: UserID, BookID (Composite key: A user can rate multiple books, and each book can have multiple ratings.)
- **Foreign Keys**: UserID (Links rating to the user who gave it), BookID (Links rating to the book being rated.)

9. Feedback

- **Primary Key**: UserID, Date (Composite key: A user can provide multiple feedback entries on different dates.)

- **Foreign Keys**: UserID (Links feedback to the user who provided it.)

10. RequestNewBook

- **Primary Key**: UserID, RequestDate (Composite key: A user can request multiple books on different dates.)

- **Foreign Keys**: UserID (Links request to the user making the request.)

11. CartCollection

- **Primary Key**: CartID (Each cart entry is uniquely identified.)

- **Foreign Keys**: UserID (Links cart to the user who added the book), BookID (Links cart to the book added.)

12. Reservation

- **Primary Key**: ReservationID (Each reservation is uniquely identified.)

- **Foreign Keys**: UserID (Links reservation to the user who made it), BookID (Links reservation to the book being reserved.)

13. BorrowedBooks

- **Primary Key**: BorrowID (Each borrow transaction is uniquely identified.)

- **Foreign Keys**: **UserID** (Links borrowing to the user who borrowed the book), **BookID** (Links borrowing to the borrowed book.)

System Specifications:

- **User Authentication:**

  - **Secure Authentication**: Users will authenticate with their credentials (username, password) along with **two-factor authentication (2FA)** via email OTP.
  - **Role-based Access Control**: Different roles (Admin, Library Staff, User) will have specific access to resources and operations.
- **Admin Features:**

  - **Library Management**: Admin can manage library details, including assigning staff and overseeing library operations.
  - **Book Management**: Admin has complete control over adding, updating, and deleting book details.
  - **User Management**: Admin can create, update, and delete user accounts, manage user roles, and oversee overdue notifications.
  - **Security**: Admin can enforce security token verification for critical actions, such as deleting books or updating library settings.
- **Library Staff Features:**

  - **Book Maintenance**: Library staff can manage book details and availability but cannot delete books or perform other high-security tasks.
  - **User Support**: Library staff can assist users with their book requests, queries, and reservations.
- **User Features:**

  - **Personalized Book Browsing**: Users can search for books by title, author, ISBN, or genre.

    - **Cart Management**: Users can add books to their cart and manage the list before borrowing or reserving.
    - **Borrowing and Reservations**: Users can borrow books for a limited time, with options to reserve unavailable books.
    - **Book Feedback and Ratings**: Users can rate books and provide feedback on their experiences.
    - **Overdue Notifications**: Users receive automated notifications about overdue books, upcoming renewals, or changes in availability.
- **Notifications:**

    - **Manual Overdue Notification System**: Users will receive manual notifications regarding overdue books and renewal reminders via email.
- **System Security and Data Privacy:**

    - **Data Encryption**: Sensitive user data (e.g., personal information, passwords) will be encrypted.
    - **Access Logs**: All admin and staff activities will be logged for auditing and security purposes.

Assumptions:

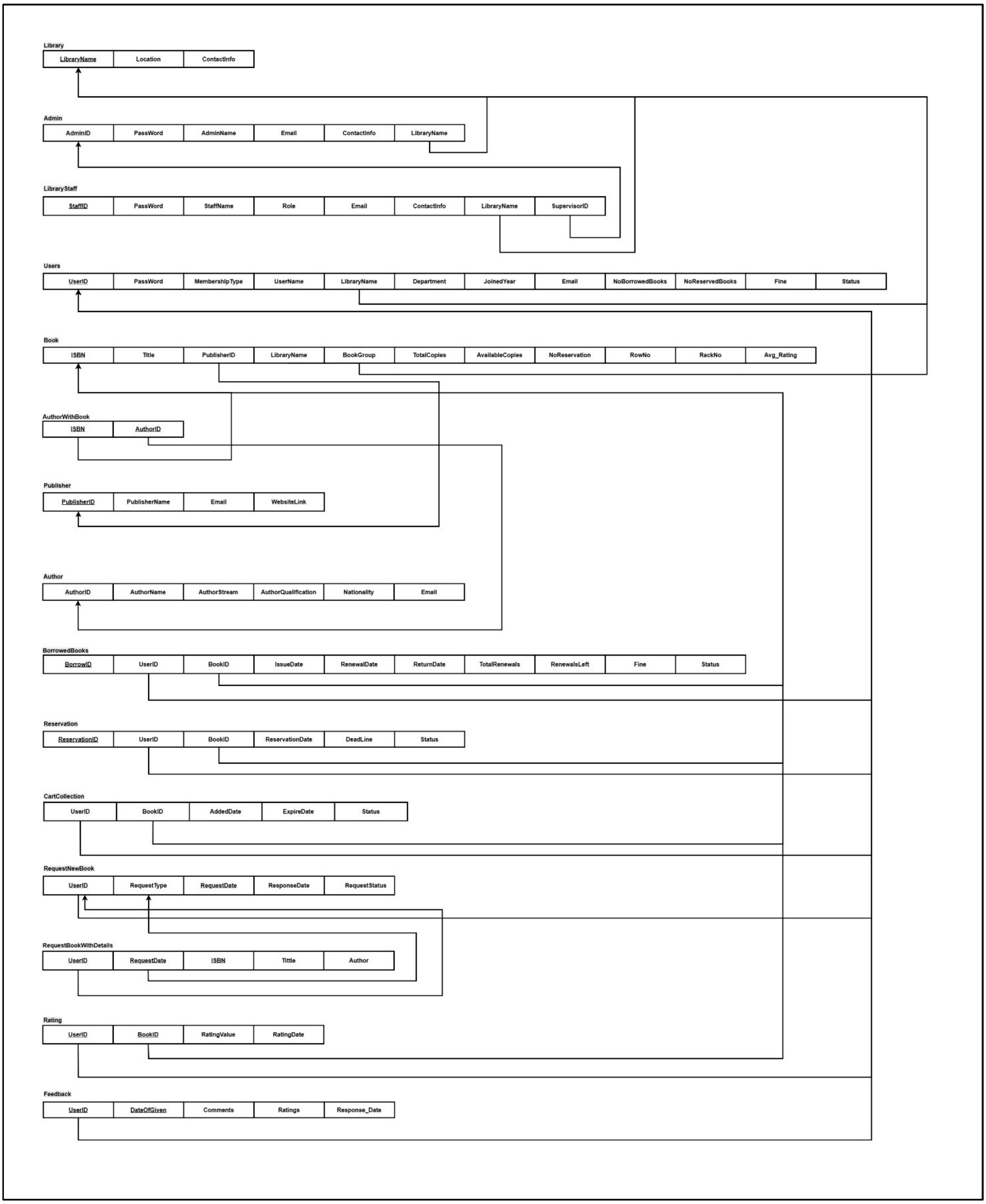
- **Books and User Records:** It is assumed that all books in the system are linked to at least one publisher and one author. Users must be registered to borrow or reserve books.
- **Overdue Handling**: If a book is not returned on time, an automated reminder will be sent to users, but manual follow-up may still be necessary.
- **User Data**: User data such as feedback, borrowing history, and cart contents will be stored securely.

- **Publisher and Author Info**: A book can have multiple authors, but every book must have a publisher.

Steps Involved in Designing the System

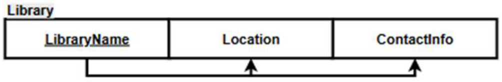
- Identification of Entities
- Drawing the ER Diagram
- Mapping ER Diagram to Relational Schema
- Identifying Functional Dependencies (FDs)

Schema Diagram

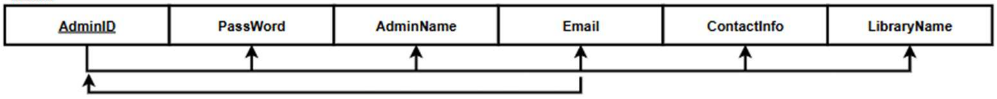


Functional Dependencies

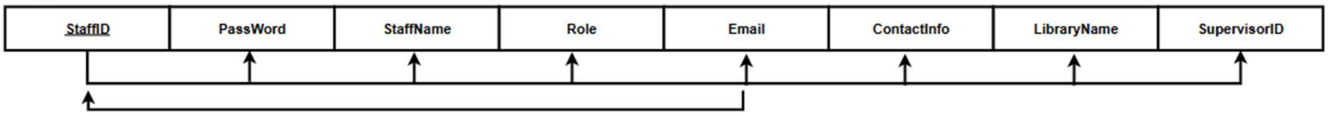
Library



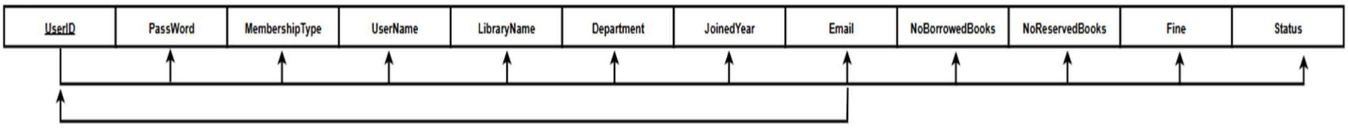
Admin



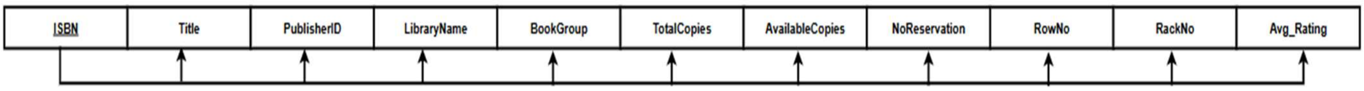
Library Staff



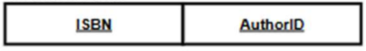
Users



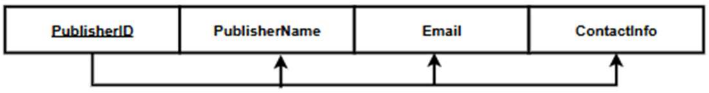
Book



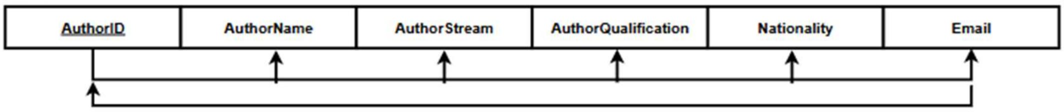
AuthorWithBook



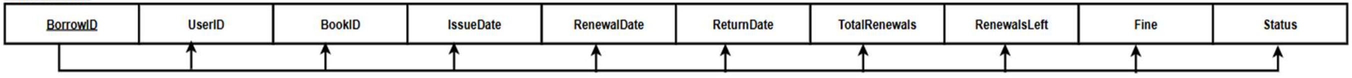
Publisher



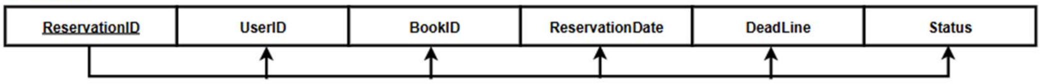
Author



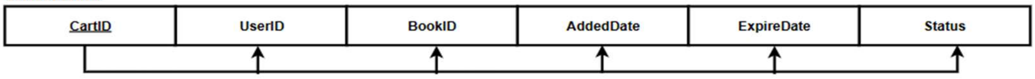
BorrowedBooks



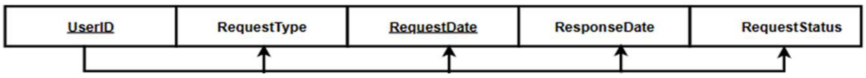
Reservation



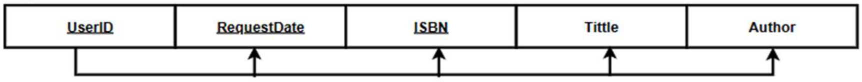
CartCollection



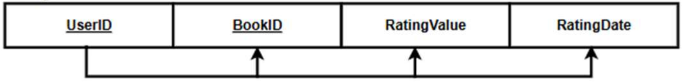
RequestNewBook



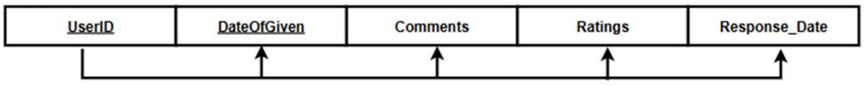
RequestBookWithDetails



Rating



Feedback



All Possible FDs

1. Library

- LibraryName → Location
- LibraryName → ContactInfo

3. LibraryStaff

- StaffID → Password
- StaffID → StaffName
- StaffID → Role
- StaffID → Email
- StaffID → ContactInfo
- StaffID → LibraryName
- StaffID → SupervisorID
- Email → Password
- Email → StaffName
- Email → Role
- Email → StaffID
- Email → ContactInfo
- Email → LibraryName
- Email → SupervisorID

5. Book

- ISBN → Title
- ISBN → AuthorID
- ISBN → PublisherID
- ISBN → LibraryName
- ISBN → BookGroup
- ISBN → TotalCopies
- ISBN → AvailableCopies
- ISBN → NoReservation
- ISBN → RowNo
- ISBN → RackNo
- ISBN → Avg_Rating

2. Admin

- AdminID → Password
- AdminID → AdminName
- AdminID → Email
- AdminID → ContactInfo
- AdminID → LibraryName
- Email → Password
- Email → AdminName
- Email → AdminID
- Email → ContactInfo
- Email → LibraryName

4. Users

- UserID → Password
- UserID → MembershipType
- UserID → UserName
- UserID → LibraryName
- UserID → Department
- UserID → JoinedYear
- UserID → Email
- UserID → NoBorrowedBooks
- UserID → NoReservedBooks
- UserID → Fine
- UserID → Status
- Email → Password
- Email → MembershipType
- Email → UserName
- Email → LibraryName
- Email → Department
- Email → JoinedYear
- Email → UserID
- Email → NoBorrowedBooks
- Email → NoReservedBooks
- Email → Fine
- Email → Status

6. Publisher

- PublisherID → PublisherName
- PublisherID → Email
- PublisherID → ContactInfo
- Email → PublisherName
- Email → PublisherID
- Email → ContactInfo

7. Author

- AuthorID → AuthorName
- AuthorID → AuthorStream
- AuthorID → AuthorQualification
- AuthorID → Nationality
- AuthorID → Email
- Email → AuthorName
- Email → AuthorStream
- Email → AuthorQualification
- Email → Nationality
- Email → AuthorID

8. AuthorWithBookName

- BookID, AuthorID → (No non-key attributes)

9. BorrowedBook

- BorrowedID → UserID
- BorrowedID → BookID
- BorrowedID → IssuedDate
- BorrowedID → RenewedDate
- BorrowedID → ReturnDate
- BorrowedID → TotalRenewals
- BorrowedID → RenewalsLeft
- BorrowedID → Fine
- BorrowedID → Status

10. Reservation

- ReservationID → UserID
- ReservationID → BookID
- ReservationID → ReservationDate
- ReservationID → Deadline
- ReservationID → Status

11. UserCart

- UserID, BookID → DateAdded
- UserID, BookID → Status

12. RequestBook

- UserID, RequestDate → RequestType
- UserID, RequestDate → ResponseDate
- UserID, RequestDate → RequestStatus

13. RequestBookWithDetails

- UserID, RequestDate, ISBN → Title
- UserID, RequestDate, ISBN → Author

14. Rating

- UserID, BookID → RatingValue
- UserID, BookID → DateOfRating

15. Feedback

- UserID, DateOfComment → Comments
- UserID, DateOfComment → Ratings
- UserID, DateOfComment → Response_Date

Minimal FDs

1. Library

- LibraryName → Location
- LibraryName → ContactInfo

3. LibraryStaff

- StaffID → Password
- StaffID → StaffName
- StaffID → Role
- StaffID → Email
- StaffID → ContactInfo
- StaffID → LibraryName
- StaffID → SupervisorID

5. Book

- ISBN → Title
- ISBN → AuthorID
- ISBN → PublisherID
- ISBN → LibraryName
- ISBN → BookGroup
- ISBN → TotalCopies
- ISBN → AvailableCopies
- ISBN → NoReservation
- ISBN → RowNo
- ISBN → RackNo
- ISBN → Avg_Rating

6. Publisher

- PublisherID → PublisherName
- PublisherID → Email
- PublisherID → ContactInfo

2. Admin

- AdminID → Password
- AdminID → AdminName
- AdminID → Email
- AdminID → ContactInfo
- AdminID → LibraryName

4. Users

- UserID → Password
- UserID → MembershipType
- UserID → UserName
- UserID → LibraryName
- UserID → Department
- UserID → JoinedYear
- UserID → Email
- UserID → NoBorrowedBooks
- UserID → NoReservedBooks
- UserID → Fine
- UserID → Status

7. Author

- AuthorID → AuthorName
- AuthorID → AuthorStream
- AuthorID → AuthorQualification
- AuthorID → Nationality
- AuthorID → Email

8. AuthorWithBookName

- BookID, AuthorID → *(No non-key attributes)*

9. BorrowedBook

- BorrowedID → UserID
- BorrowedID → BookID
- BorrowedID → IssuedDate
- BorrowedID → RenewedDate
- BorrowedID → ReturnDate
- BorrowedID → TotalRenewals
- BorrowedID → RenewalsLeft
- BorrowedID → Fine
- BorrowedID → Status

10. Reservation

- ReservationID → UserID
- ReservationID → BookID
- ReservationID → ReservationDate
- ReservationID → Deadline
- ReservationID → Status

11. UserCart

- UserID, BookID → DateAdded
- UserID, BookID → Status

12. RequestBook

- UserID, RequestDate → RequestType
- UserID, RequestDate → ResponseDate
- UserID, RequestDate → RequestStatus

13. RequestBookWithDetails

- UserID, RequestDate, ISBN → Title
- UserID, RequestDate, ISBN → Author

14. Rating

- UserID, BookID → RatingValue
- UserID, BookID → DateOfRating

15. Feedback

- UserID, DateOfComment → Comments
- UserID, DateOfComment → Ratings
- UserID, DateOfComment → Response_Date

Normalization

1. **Find Candidate Key (C.K)**
   - Prove that it is a Super Key (C.K → R).
   - Prove that **no proper subset** is a Super Key.
2. **Prove that it is the Only Candidate Key**
   - Check RHS of all FDs.
   - If no **prime attribute** appears on RHS → this is the **only Candidate Key**.

Normal Form	**Required Conditions**
1NF	Atomic attributes (no multi-valued or composite values)
2NF	1NF + No partial dependency (Proper subset of C.K → NPA)
3NF	2NF + No transitive dependency (NPA → NPA)
BCNF	3NF + For every FD X → Y, X is a **Super Key**

Relation: Library(LibraryName, Location, ContactInfo)
FDs: LibraryName → Location, ContactInfo

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {LibraryName}
2. **Proof only C.K**:
   → No other attribute or combo determines all attributes. Only LibraryName.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (LibraryName is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → LibraryName → Location, ContactInfo; X = C.K → BCNF holds

Relation: Admin(AdminID, Password, AdminName, Email, ContactInfo, LibraryName)
FD: AdminID → Password, AdminName, Email, ContactInfo, LibraryName

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {AdminID}
2. **Proof only C.K**:
   → No other attribute or combo determines all others. Only AdminID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (AdminID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → AdminID → all; X = C.K → BCNF holds

Relation: LibraryStaff(StaffID, Password, StaffName, Role, Email, ContactInfo, LibraryName, SupervisorID)
FD: StaffID → Password, StaffName, Role, Email, ContactInfo, LibraryName, SupervisorID

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {StaffID}
2. **Proof only C.K**:
   → No other attribute or combo determines all others. Only StaffID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
     → No proper subset of C.K → NPA (StaffID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
     → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
     → StaffID → all; X = C.K → BCNF holds

Relation: Users(UserID, Password, MembershipType, UserName, LibraryName, Department, JoinedYear, Email, NoBorrowedBooks, NoReservedBooks, Fine, Status)
FD: UserID → all other attributes

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {UserID}
2. **Proof only C.K**:
   → No other attribute or combo determines all others. Only UserID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
     → No proper subset of C.K → NPA (UserID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
     → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
     → UserID → all; X = C.K → BCNF holds

Relation: Book(ISBN, Title, PublisherID, LibraryName, BookGroup, TotalCopies, AvailableCopies, NoReservation, RowNo, RackNo, Avg_Rating)
FD: ISBN → all other attributes

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {ISBN}
2. **Proof only C.K**:
   → No other attribute or combination determines all attributes. Only ISBN.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (ISBN is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → ISBN → all; X = C.K → BCNF holds

Relation: Publisher(PublisherID, PublisherName, Email, ContactInfo)
FD: PublisherID → PublisherName, Email, ContactInfo

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {PublisherID}
2. **Proof only C.K**:
   → No other attribute or combo determines all attributes. Only PublisherID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (PublisherID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → PublisherID → all; X = C.K → BCNF holds

Relation: Author(AuthorID, AuthorName, AuthorStream, AuthorQualification, Nationality, Email)
FD: AuthorID → AuthorName, AuthorStream, AuthorQualification, Nationality, Email

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {AuthorID}
2. **Proof only C.K**:
   → No other attribute or combination determines all others. Only AuthorID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (AuthorID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → AuthorID → all; X = C.K → BCNF holds

Relation: AuthorWithBook(BookID, AuthorID)
FDs: (No non-trivial FDs other than from candidate key)
PK: {BookID, AuthorID}

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {BookID, AuthorID}
2. **Proof only C.K**:
   → Neither BookID nor AuthorID alone determines the other → Only composite key
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   ✓ No NPA exists → No dependency of a non-prime on subset of CK
5. **Check 3NF**:
   → 2NF + No transitive dependency
   ✓ No NPA → NPA FD exists
6. **Check BCNF**:
   → All FDs have LHS as super key
   ✓ Only CK exists → satisfies BCNF

Relation: BorrowedBook(BorrowedID, UserID, BookID, IssueDate, RenewedDate, ReturnDate, TotalRenewals, RenewalsLeft, Fine, Status)
FD: BorrowedID → all other attributes

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {BorrowedID}
2. **Proof only C.K**:
   → No other attribute or combination determines all others. Only BorrowedID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (BorrowedID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → BorrowedID → all; X = C.K → BCNF holds

Relation: Reservation(ReservationID, UserID, BookID, ReservationDate, Deadline, Status)
FD: ReservationID → all other attributes

Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {ReservationID}
2. **Proof only C.K**:
   → No other attribute or combination determines all others. Only ReservationID.
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA (ReservationID is single attribute)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → ReservationID → all; X = C.K → BCNF holds

Relation: CartCollection(UserID, BookID, AddedDate, ExpireDate, Status)
FD: (UserID, BookID) → AddedDate, ExpireDate, Status
PK: {UserID, BookID}
Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {UserID, BookID}
2. **Proof only C.K**:
   → Neither UserID nor BookID alone determines all others → Only the composite key
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA
   (UserID ↛ NPA, BookID ↛ NPA)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → (UserID, BookID) → others; X = C.K → BCNF holds

Relation:RequestBook(BookID, RequestedDate, RequestType, ResponseDate, RequestStatus)
FD: (BookID, RequestedDate) → RequestType, ResponseDate,Status
PK: {BookID, RequestedDate}
Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {BookID, RequestedDate}
2. **Proof only C.K**:
   → BookID alone or RequestedDate alone cannot determine all others
   → Only the composite key
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA
   (BookID ↛ NPA, RequestedDate ↛ NPA)
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → (BookID, RequestedDate) → all others; X = C.K → BCNF holds

Relation: RequestBookWithDetails(UserID, RequestDate, BookID, Title, Author)
FD: (UserID, RequestDate, BookID) → Title, Author
PK: {UserID, RequestDate, BookID}
Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {UserID, RequestDate, BookID}
2. **Proof only C.K**:
   → No proper subset of {UserID, RequestDate, BookID} determines all other attributes
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → (UserID, RequestDate, BookID) → Title, Author; X = C.K → BCNF holds

Relation: Rating(UserID, BookID, RatingValue, RatingDate)
FD: (UserID, BookID) → RatingValue, RatingDate
PK: {UserID, BookID}
Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {UserID, BookID}
2. **Proof only C.K**:
   → Neither UserID nor BookID alone determines all others → Only the composite key
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → (UserID, BookID) → RatingValue, RatingDate → X = C.K → BCNF holds

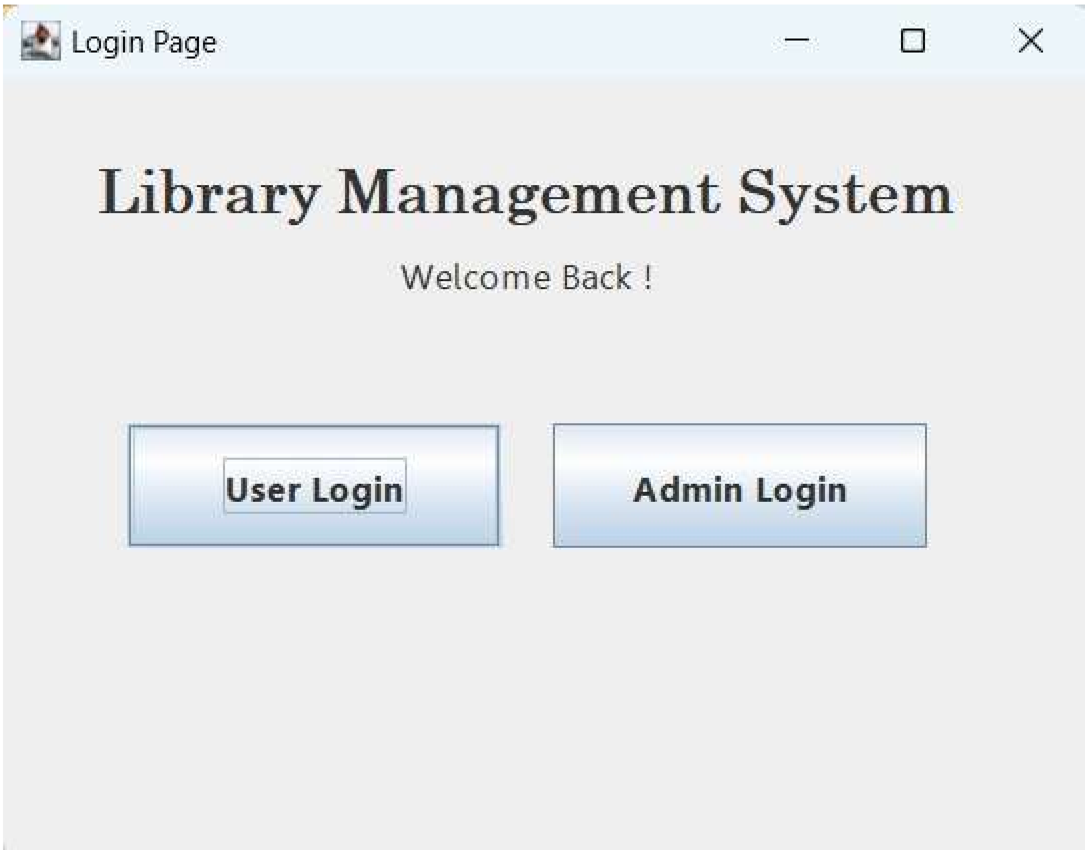
Relation: Feedback(UserID, DateOfComment, Comments, Ratings, Response_Date)
FD: (UserID, DateOfComment) → Comments, Ratings, Response_Date
PK: {UserID, DateOfComment}

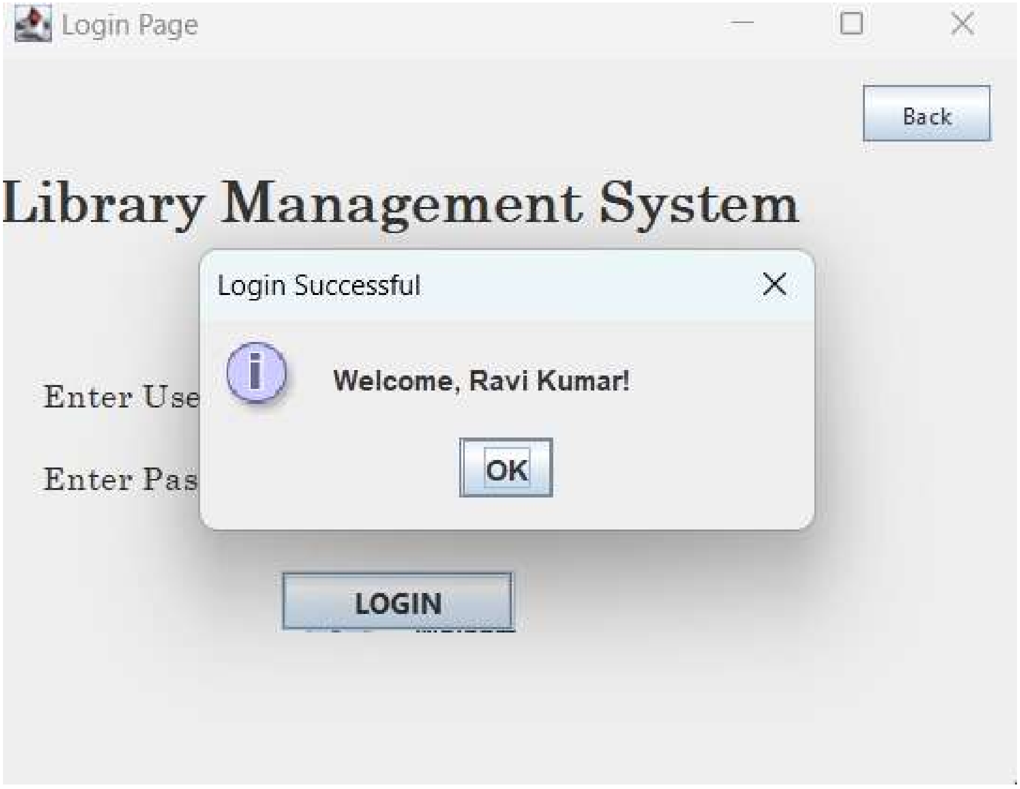
Step-by-step Normalization:

1. **Candidate Key (C.K)**:
   → {UserID, DateOfComment}
2. **Proof only C.K**:
   → Neither UserID nor DateOfComment alone determines all others → Only the composite key
3. **Check 1NF**:
   → All attributes atomic → 1NF satisfied
4. **Check 2NF**:
   → 1NF + No partial dependency
   → No proper subset of C.K → NPA
5. **Check 3NF**:
   → 2NF + No transitive dependency
   → No NPA → NPA FD exists
6. **Check BCNF**:
   → For every FD X → Y, X is a super key
   → (UserID, DateOfComment) → all others → X = C.K → BCNF holds

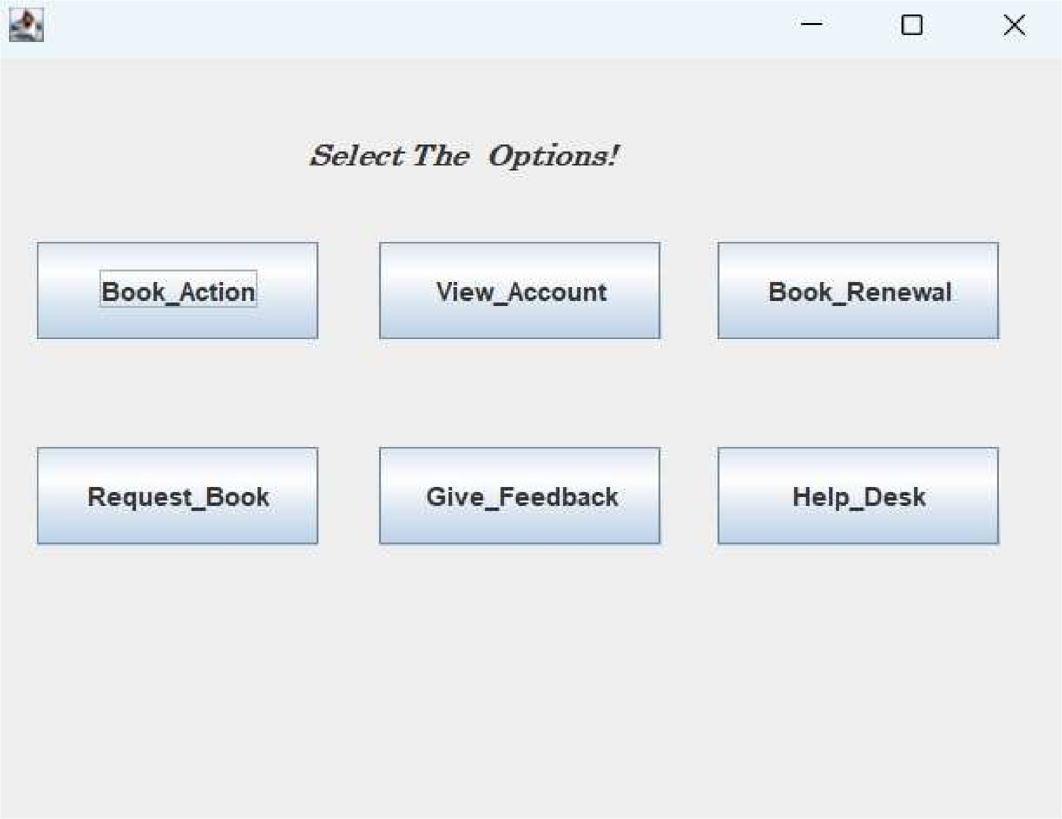
Relation Name	**Candidate Key(s)**	**Normal Form**
Library	LibraryName	**BCNF**
Admin	AdminID	**BCNF**
LibraryStaff	StaffID	**BCNF**
Users	UserID	**BCNF**
Book	ISBN	**BCNF**
Publisher	PublisherID	**BCNF**
Author	AuthorID	**BCNF**
AuthorWithBook	BookID + AuthorID	**BCNF**
BorrowedBook	BorrowedID	**BCNF**
Reservation	ReservationID	**BCNF**
CartCollection	UserID + BookID	**BCNF**
RequestBook	BookID + RequestedDate	**BCNF**
RequestBookWithDetails	UserID + RequestDate + BookID	**BCNF**
Rating	UserID + BookID	**BCNF**
Feedback	UserID + DateOfComment	**BCNF**

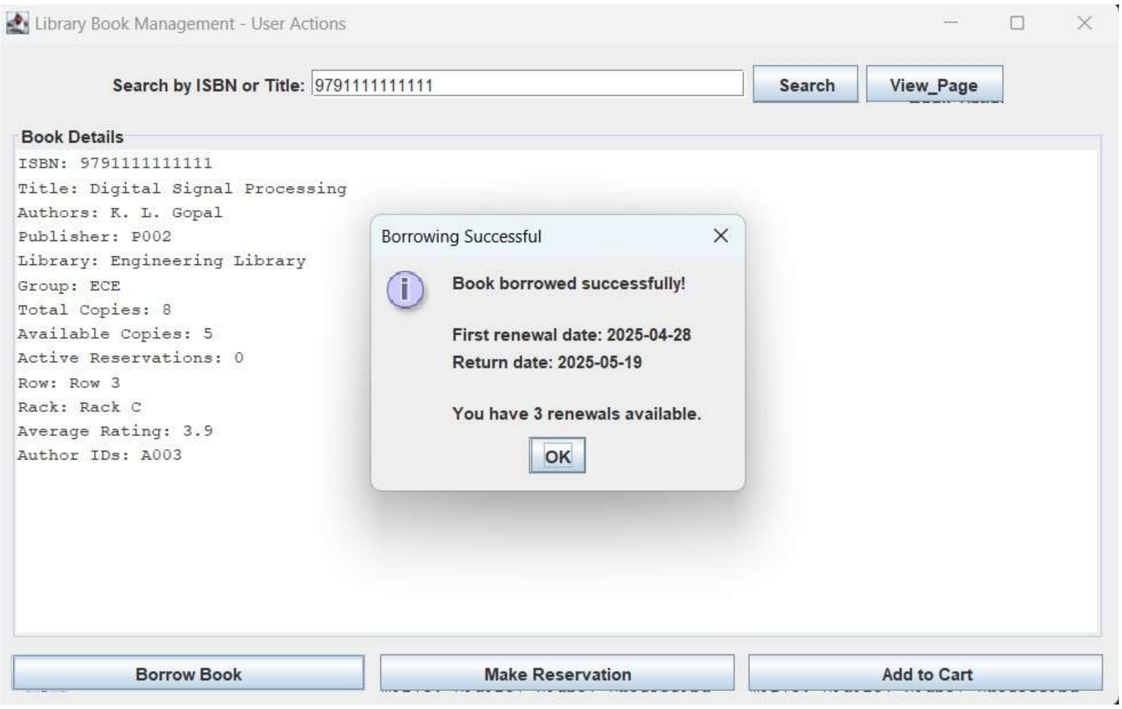
SNAPSHOTS





Welcome, Ravi Kumar!




Library Book Management - User Actions

Search by ISBN or Title: 9791111111111

Book Details

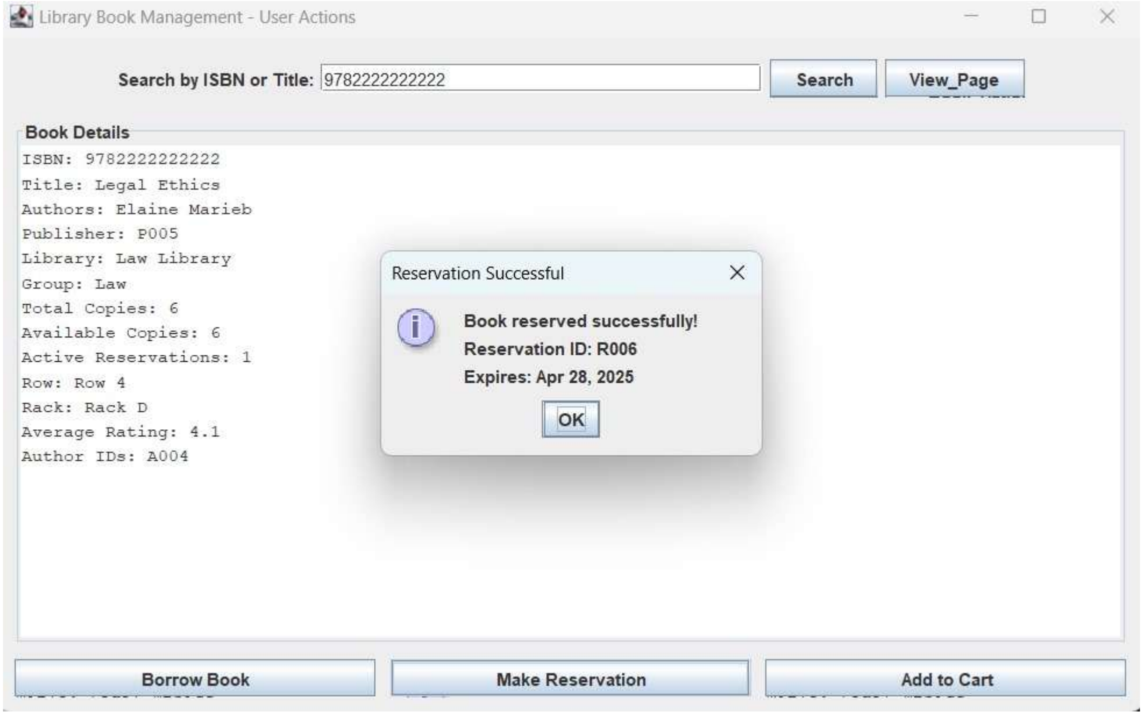
```
ISBN: 9791111111111
Title: Digital Signal Processing
Authors: K. L. Gopal
Publisher: P002
Library: Engineering Library
Group: ECE
Total Copies: 8
Available Copies: 5
Active Reservations: 0
Row: Row 3
Rack: Rack C
Average Rating: 3.9
Author IDs: A003
```

Borrowing Successful

Book borrowed successfully!

First renewal date: 2025-04-28
Return date: 2025-05-19

You have 3 renewals available.



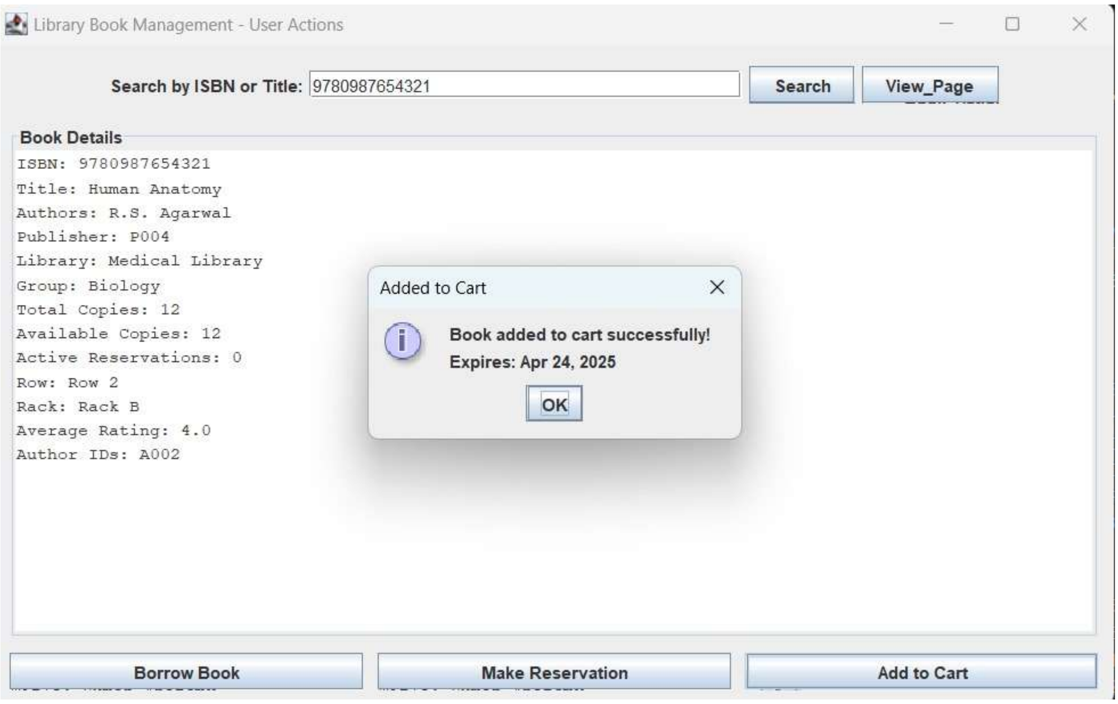
Search by ISBN or Title: 9782222222222

Book Details

```
ISBN: 9782222222222
Title: Legal Ethics
Authors: Elaine Marieb
Publisher: P005
Library: Law Library
Group: Law
Total Copies: 6
Available Copies: 6
Active Reservations: 1
Row: Row 4
Rack: Rack D
Average Rating: 4.1
Author IDs: A004
```

Reservation Successful

Book reserved successfully!
Reservation ID: R006
Expires: Apr 28, 2025



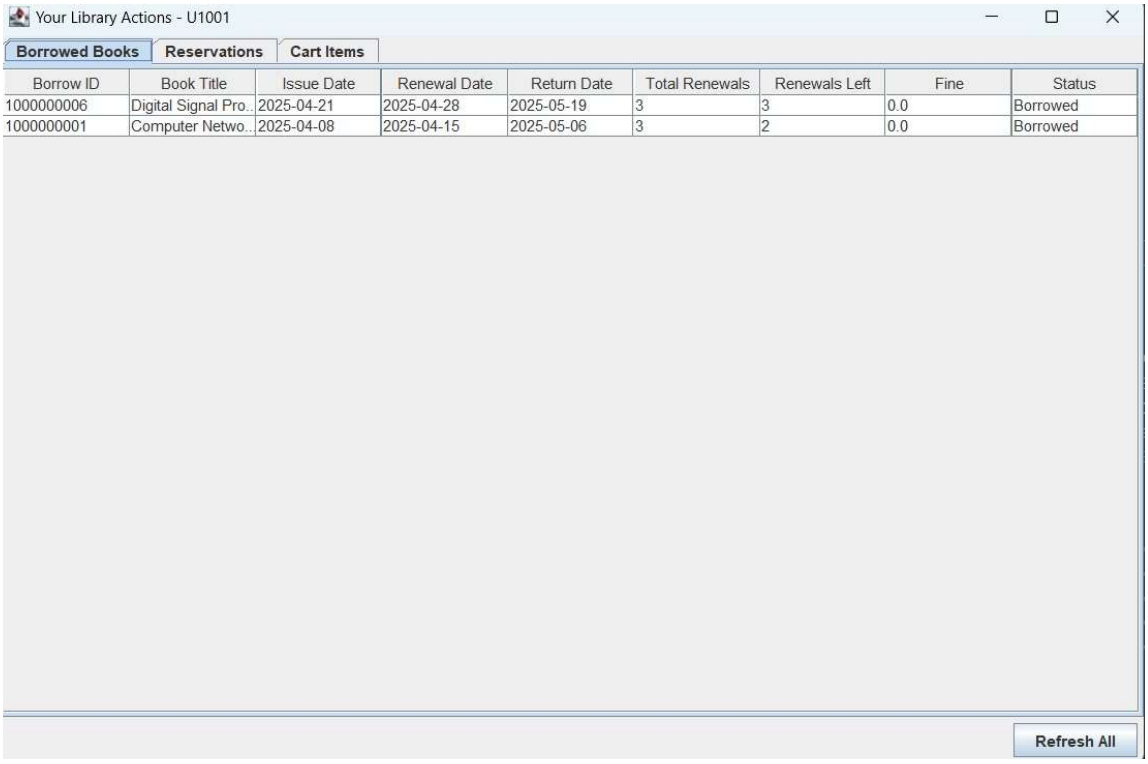
Search by ISBN or Title: 9780987654321

Book Details

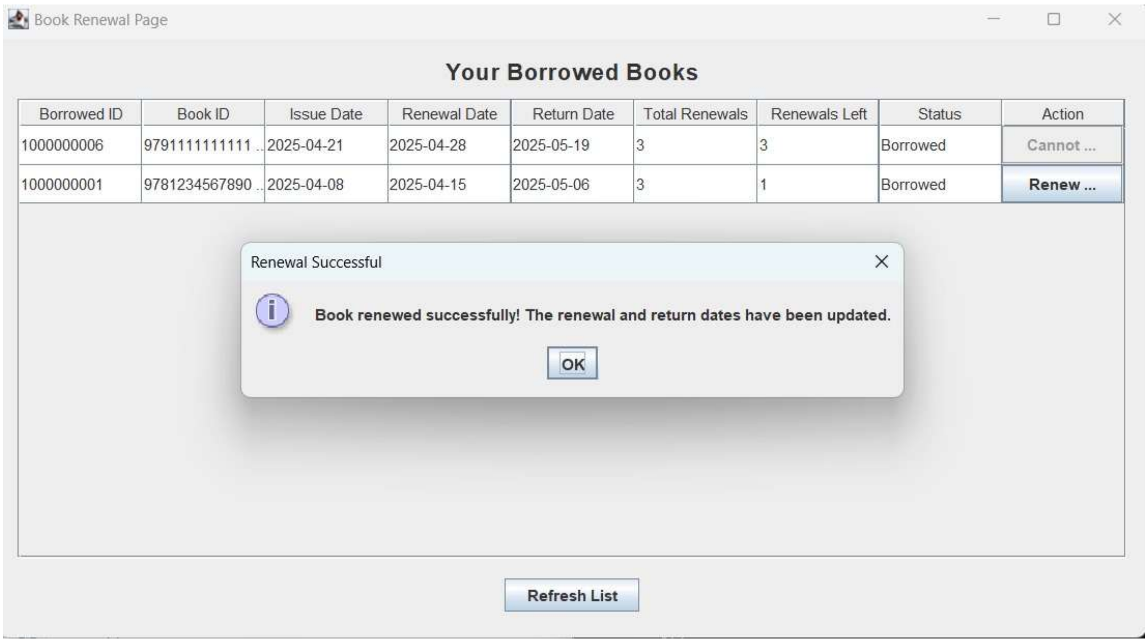
```
ISBN: 9780987654321
Title: Human Anatomy
Authors: R.S. Agarwal
Publisher: P004
Library: Medical Library
Group: Biology
Total Copies: 12
Available Copies: 12
Active Reservations: 0
Row: Row 2
Rack: Rack B
Average Rating: 4.0
Author IDs: A002
```

Added to Cart

Book added to cart successfully!
Expires: Apr 24, 2025



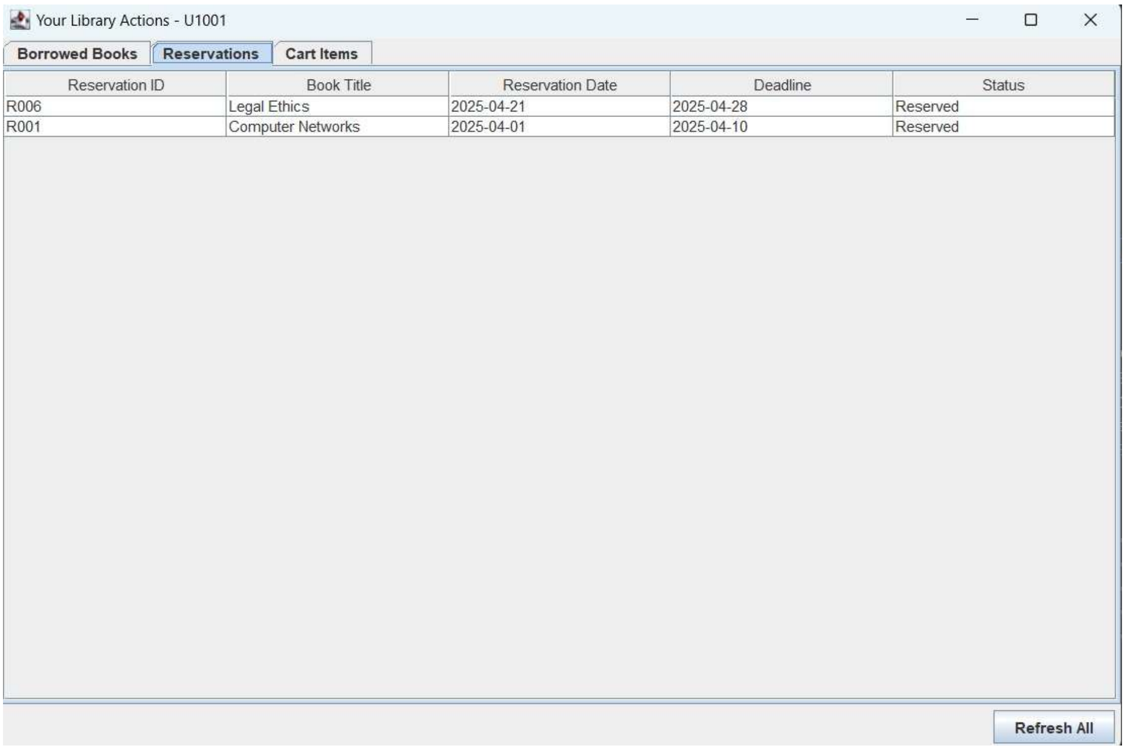
Borrow ID	Book Title	Issue Date	Renewal Date	Return Date	Total Renewals	Renewals Left	Fine	Status
1000000006	Digital Signal Pro...	2025-04-21	2025-04-28	2025-05-19	3	3	0.0	Borrowed
1000000001	Computer Netwo...	2025-04-08	2025-04-15	2025-05-06	3	2	0.0	Borrowed



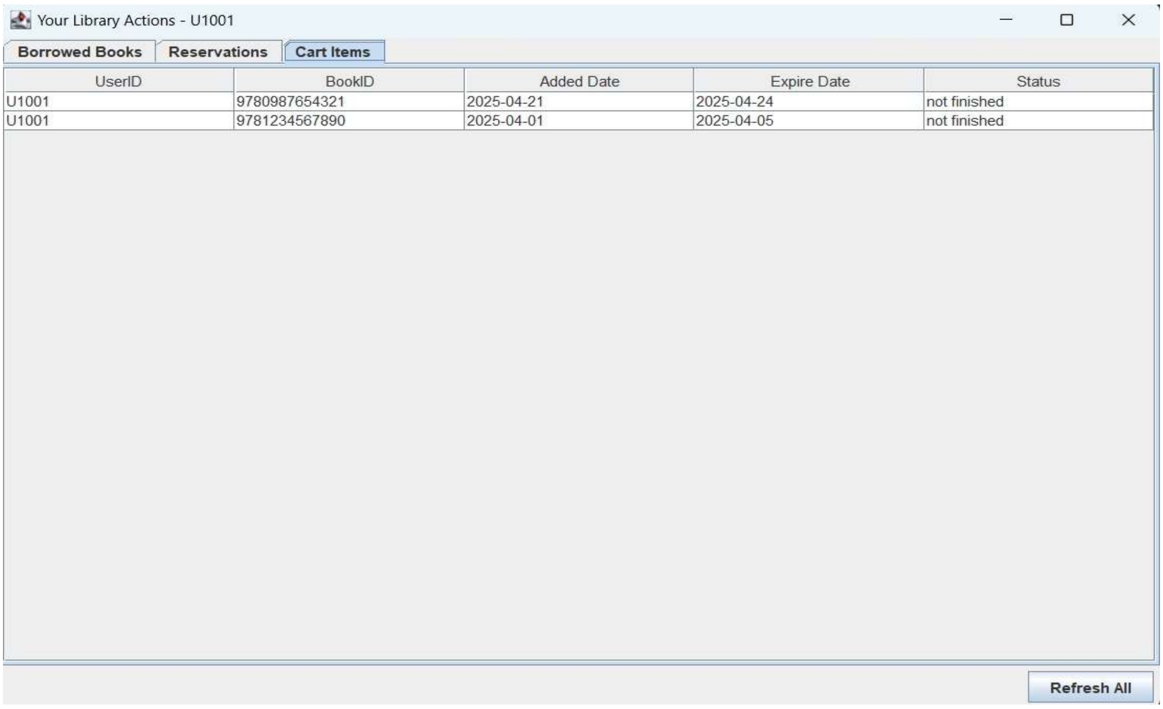
Your Borrowed Books

Borrowed ID	Book ID	Issue Date	Renewal Date	Return Date	Total Renewals	Renewals Left	Status	Action
1000000006	9791111111111 ...	2025-04-21	2025-04-28	2025-05-19	3	3	Borrowed	Cannot ...
1000000001	9781234567890 ...	2025-04-08	2025-04-15	2025-05-06	3	1	Borrowed	Renew ...

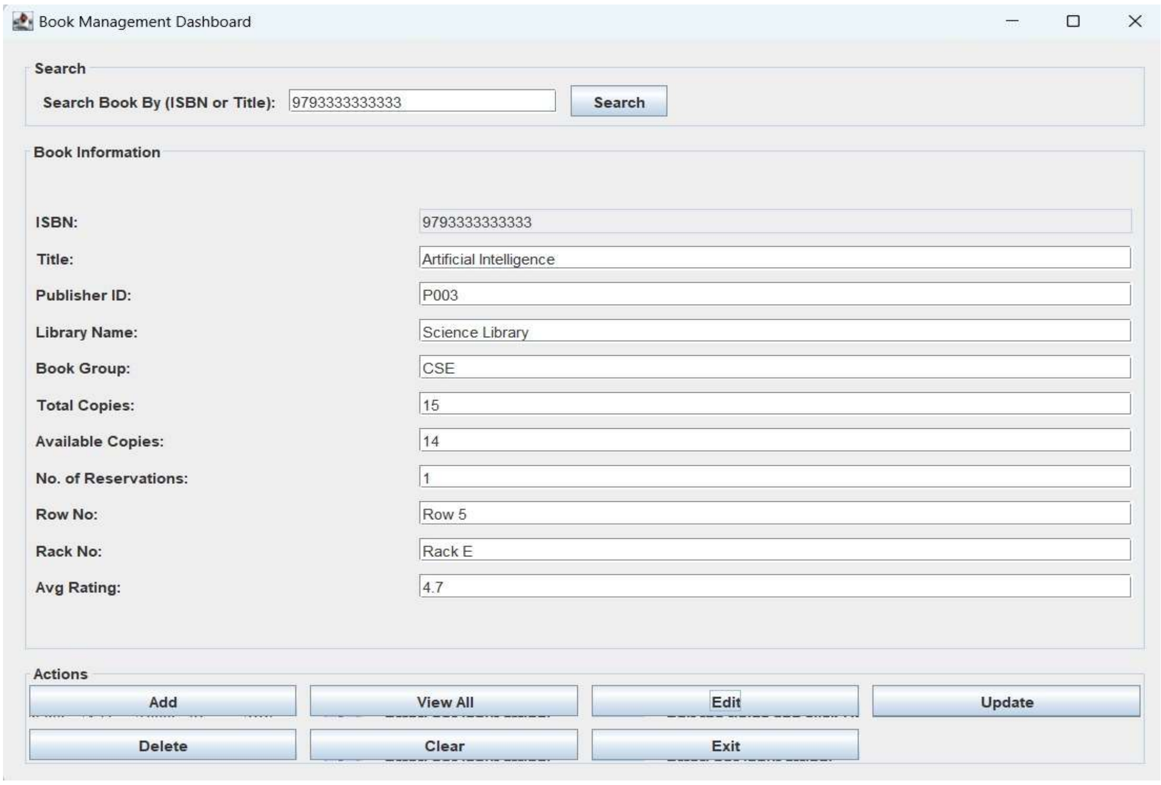
Book renewed successfully! The renewal and return dates have been updated.



Reservation ID	Book Title	Reservation Date	Deadline	Status
R006	Legal Ethics	2025-04-21	2025-04-28	Reserved
R001	Computer Networks	2025-04-01	2025-04-10	Reserved

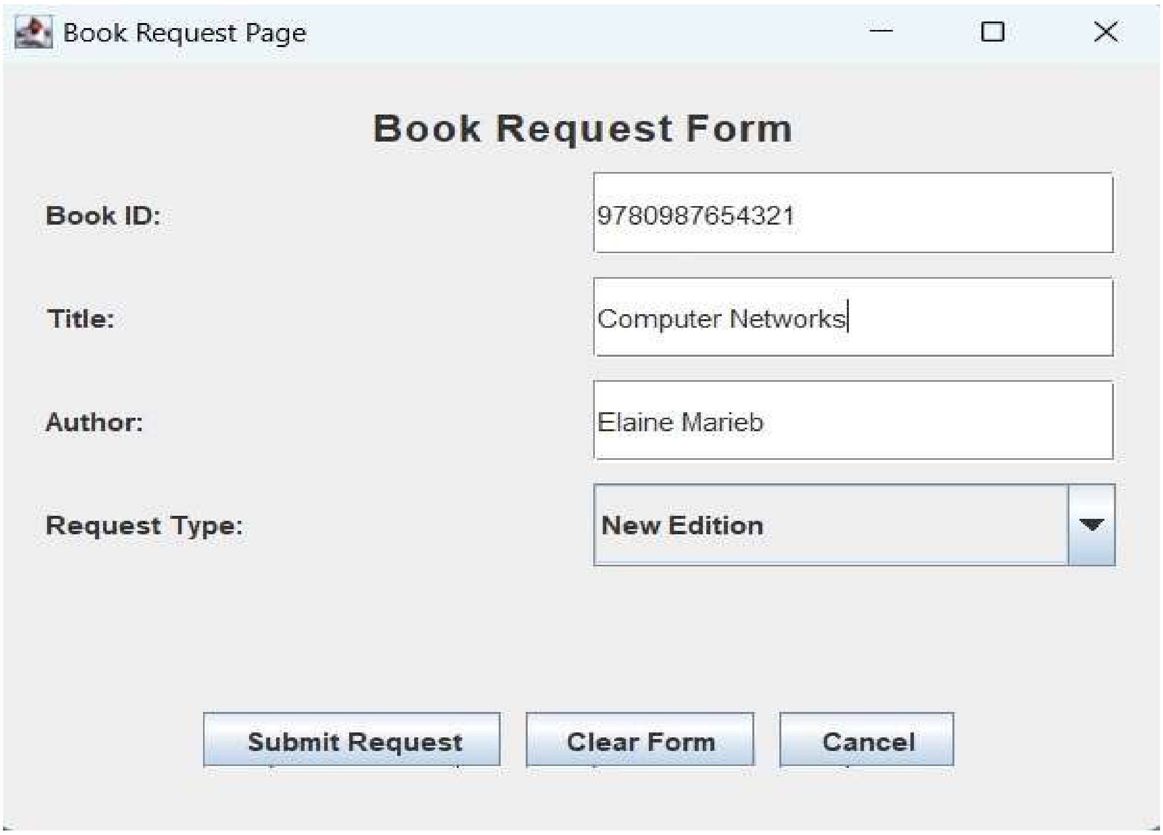


UserID	BookID	Added Date	Expire Date	Status
U1001	9780987654321	2025-04-21	2025-04-24	not finished
U1001	9781234567890	2025-04-01	2025-04-05	not finished



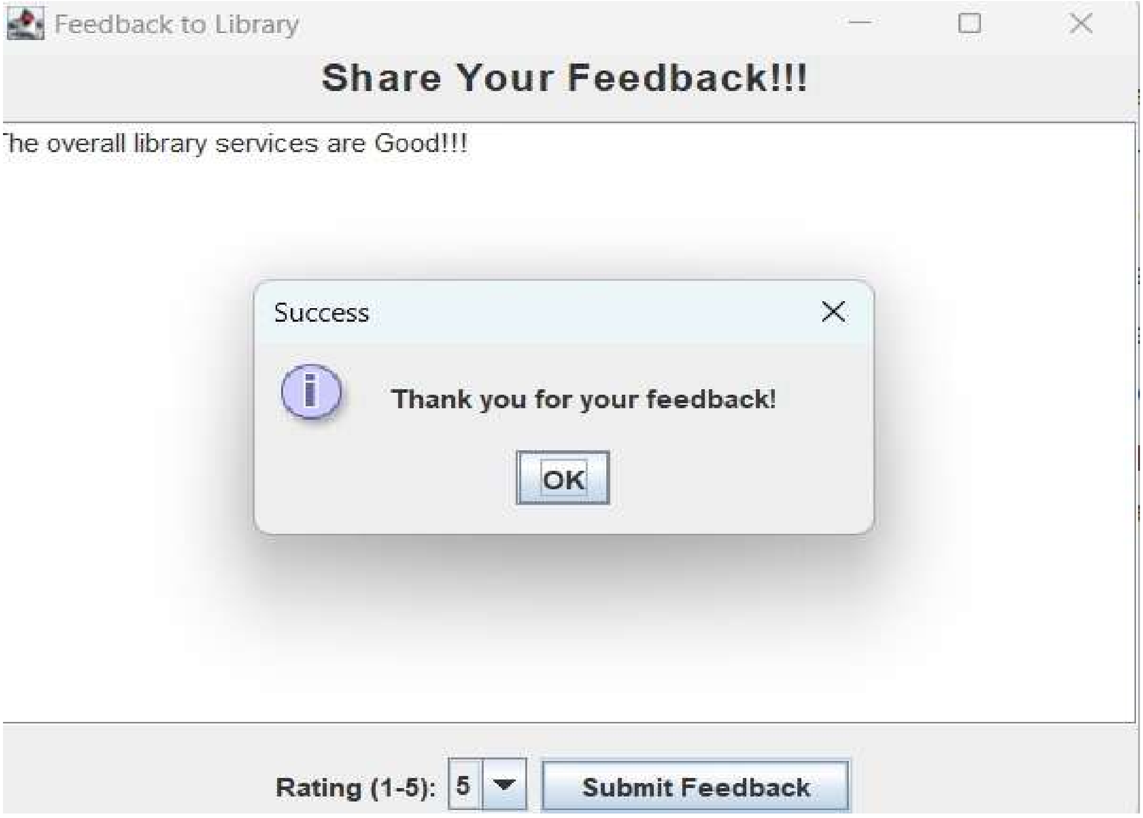
Search Book By (ISBN or Title): 9793333333333

ISBN:	9793333333333
Title:	Artificial Intelligence
Publisher ID:	P003
Library Name:	Science Library
Book Group:	CSE
Total Copies:	15
Available Copies:	14
No. of Reservations:	1
Row No:	Row 5
Rack No:	Rack E
Avg Rating:	4.7



Book Request Form

Book ID:	9780987654321
Title:	Computer Networks
Author:	Elaine Marieb
Request Type:	New Edition

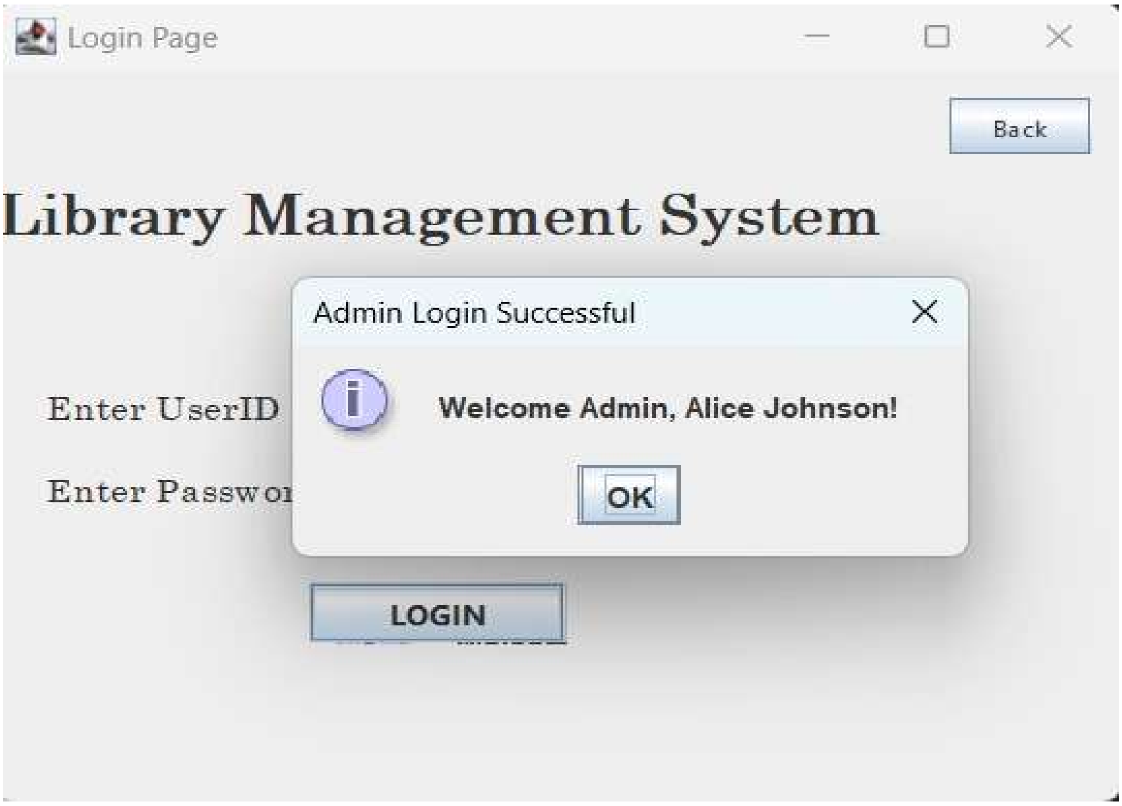


Share Your Feedback!!!

'he overall library services are Good!!!

Thank you for your feedback!

Rating (1-5): 5

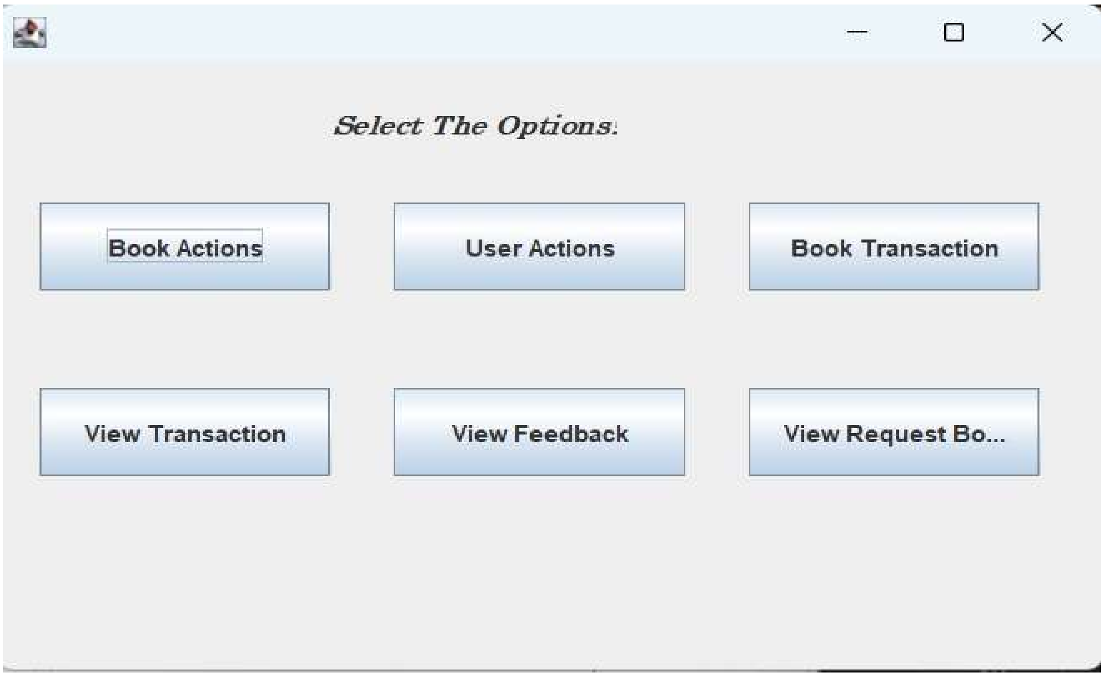


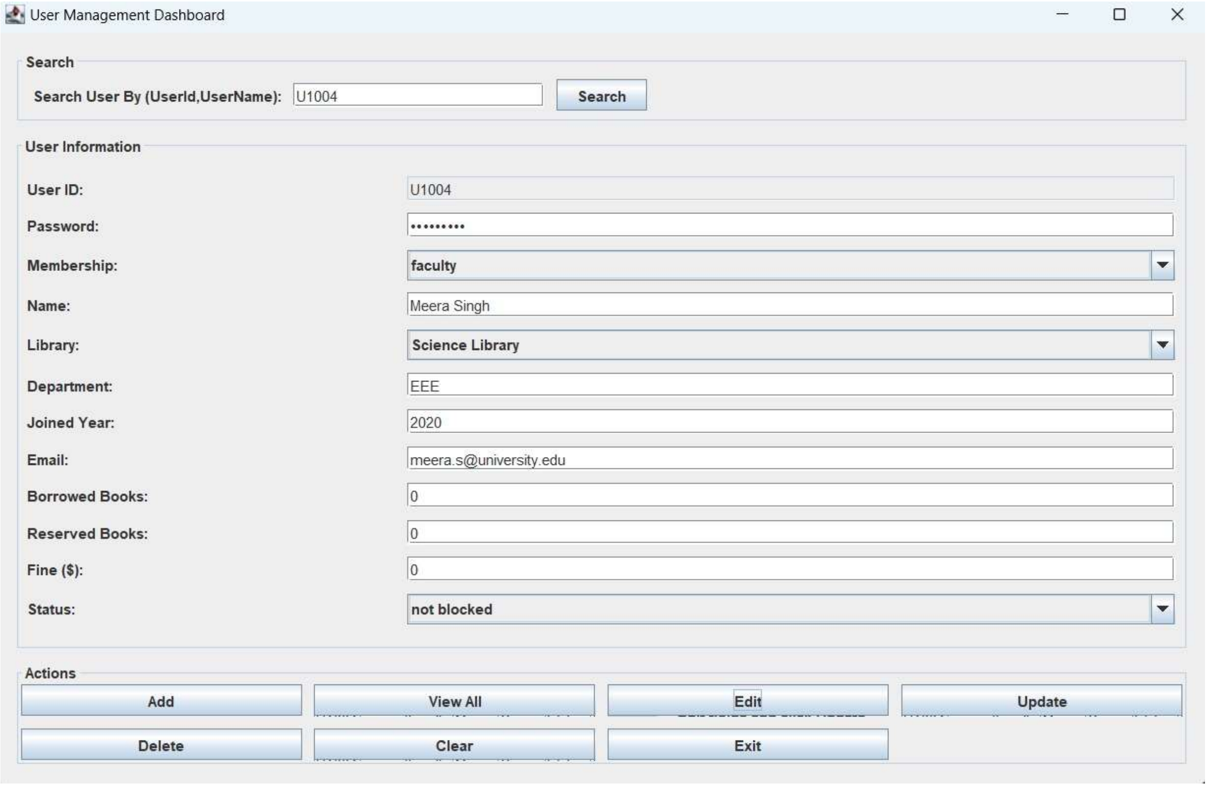
Library Management System

Enter UserID

Enter Passwo

Welcome Admin, Alice Johnson!




User Management Dashboard

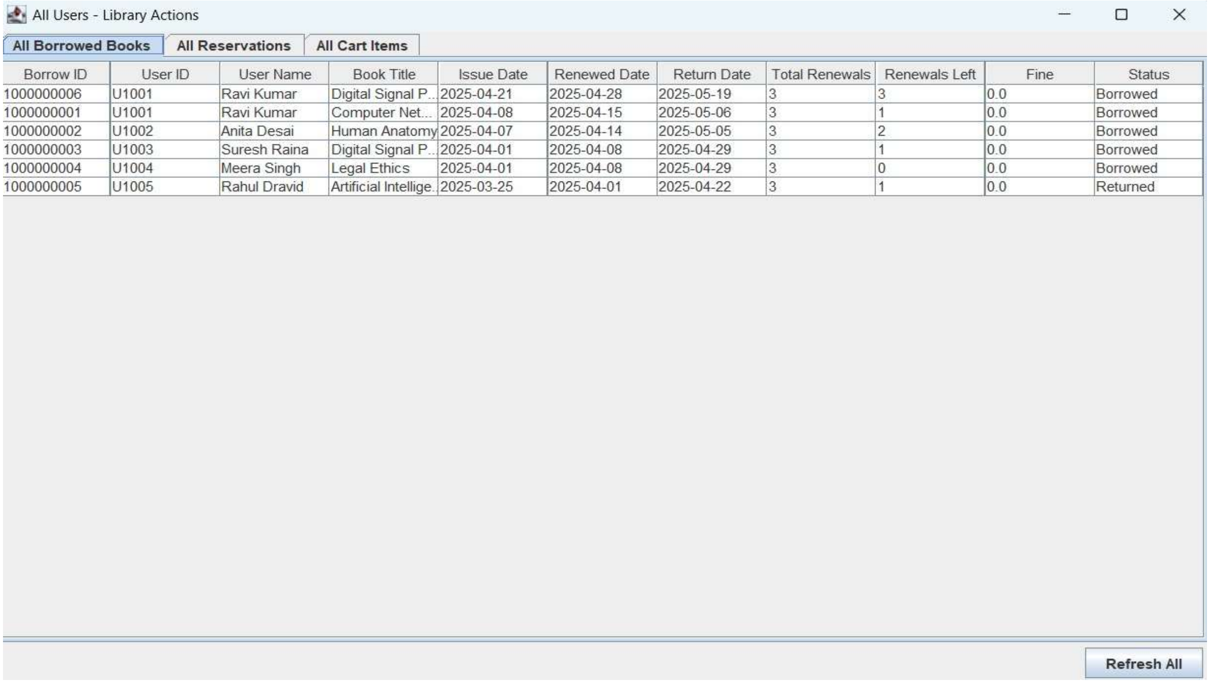
Search

Search User By (UserId,UserName): U1004

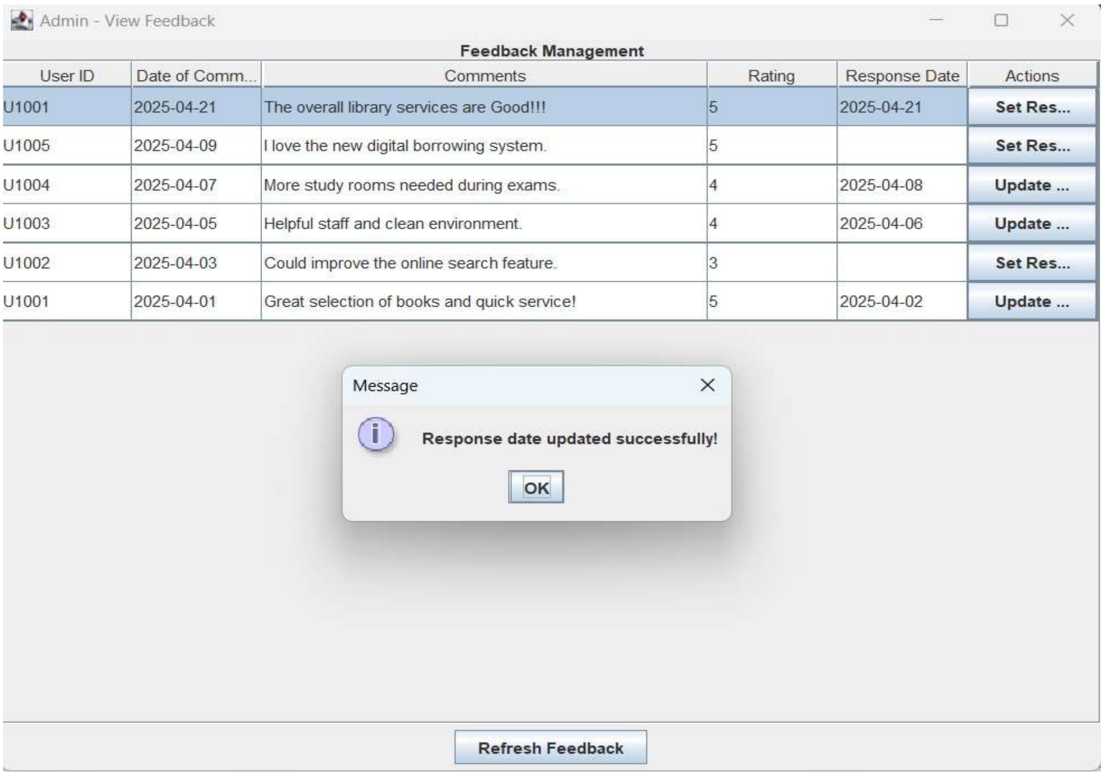
User Information

User ID:	U1004
Password:	•••••••••
Membership:	faculty
Name:	Meera Singh
Library:	Science Library
Department:	EEE
Joined Year:	2020
Email:	meera.s@university.edu
Borrowed Books:	0
Reserved Books:	0
Fine ($):	0
Status:	not blocked

Actions



Borrow ID	User ID	User Name	Book Title	Issue Date	Renewed Date	Return Date	Total Renewals	Renewals Left	Fine	Status
1000000006	U1001	Ravi Kumar	Digital Signal P...	2025-04-21	2025-04-28	2025-05-19	3	3	0.0	Borrowed
1000000001	U1001	Ravi Kumar	Computer Net...	2025-04-08	2025-04-15	2025-05-06	3	1	0.0	Borrowed
1000000002	U1002	Anita Desai	Human Anatomy	2025-04-07	2025-04-14	2025-05-05	3	2	0.0	Borrowed
1000000003	U1003	Suresh Raina	Digital Signal P...	2025-04-01	2025-04-08	2025-04-29	3	1	0.0	Borrowed
1000000004	U1004	Meera Singh	Legal Ethics	2025-04-01	2025-04-08	2025-04-29	3	0	0.0	Borrowed
1000000005	U1005	Rahul Dravid	Artificial Intellige...	2025-03-25	2025-04-01	2025-04-22	3	1	0.0	Returned



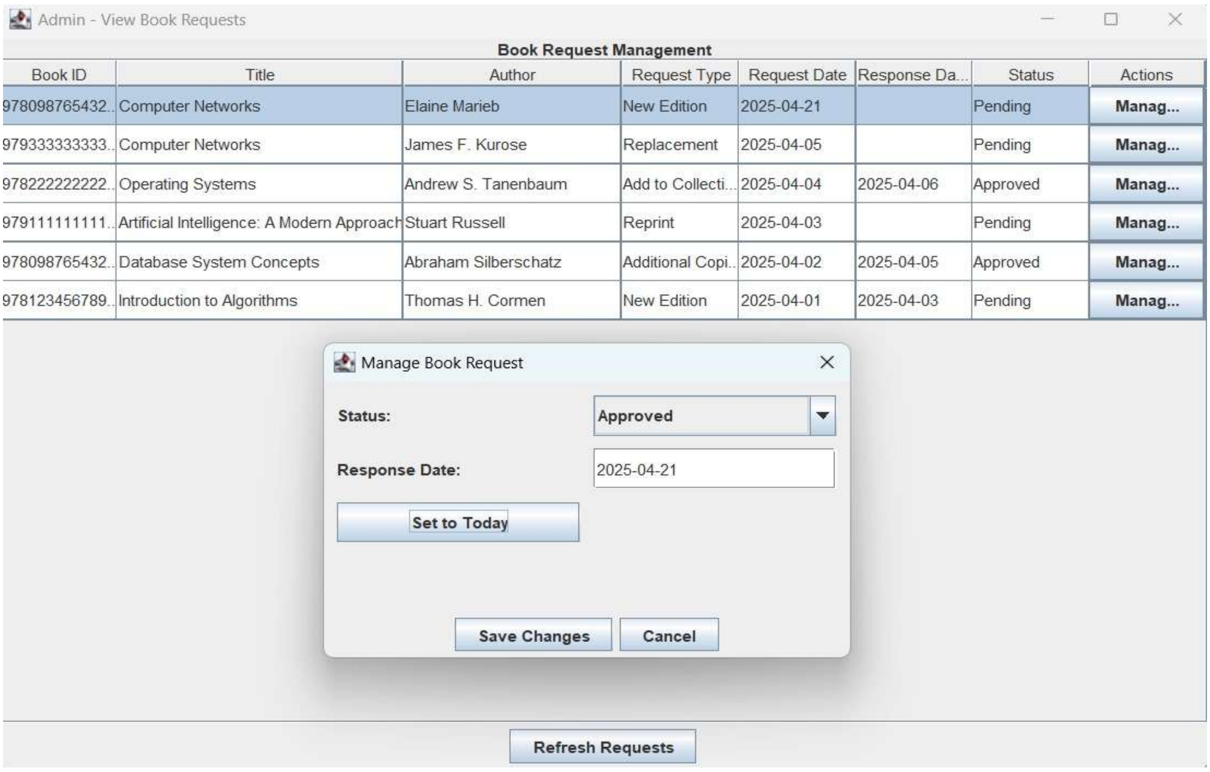
Feedback Management

User ID	Date of Comm...	Comments	Rating	Response Date	Actions
U1001	2025-04-21	The overall library services are Good!!!	5	2025-04-21	Set Res...
U1005	2025-04-09	I love the new digital borrowing system.	5		Set Res...
U1004	2025-04-07	More study rooms needed during exams.	4	2025-04-08	Update ...
U1003	2025-04-05	Helpful staff and clean environment.	4	2025-04-06	Update ...
U1002	2025-04-03	Could improve the online search feature.	3		Set Res...
U1001	2025-04-01	Great selection of books and quick service!	5	2025-04-02	Update ...

Message

Response date updated successfully!

OK


Admin - View Book Requests

Book Request Management

Book ID	Title	Author	Request Type	Request Date	Response Da...	Status	Actions
978098765432..	Computer Networks	Elaine Marieb	New Edition	2025-04-21		Pending	Manag...
979333333333..	Computer Networks	James F. Kurose	Replacement	2025-04-05		Pending	Manag...
978222222222..	Operating Systems	Andrew S. Tanenbaum	Add to Collecti...	2025-04-04	2025-04-06	Approved	Manag...
979111111111..	Artificial Intelligence: A Modern Approach	Stuart Russell	Reprint	2025-04-03		Pending	Manag...
978098765432..	Database System Concepts	Abraham Silberschatz	Additional Copi...	2025-04-02	2025-04-05	Approved	Manag...
978123456789..	Introduction to Algorithms	Thomas H. Cormen	New Edition	2025-04-01	2025-04-03	Pending	Manag...

Manage Book Request

Status: Approved

Response Date: 2025-04-21

Learning Outcomes

1. Understood the complete database system design lifecycle, including ER modeling, relational mapping, and normalization.

2. Gained hands-on experience in implementing a secure Library Management System using Java JFrame and NetBeans.

3. Strengthened skills in database normalization, ensuring data integrity and eliminating redundancy in relational design.

4. Improved skills in backend integration, user interface design, and managing user/admin functionalities.

5. Developed team collaboration, documentation, and real-world problem-solving abilities through this project.