Task: Library Database Implementation

Objective:

Design and implement a layered architecture for a Library Database, focusing on the Book and Member tables, and optionally extending to additional tables.

Instructions:

1. **Create an ER-Model:**
   - Use draw.io to create an Entity-Relationship diagram for the Library Database.
   - Define relationships between entities (e.g., one-to-many, many-to-many).
2. **Complete the Layered Architecture for the Book Table:**
   - **BookDAOImpl**:
     - Finish the implementation of the BookDAOImpl class based on the lecture
   - **Service Class**:
     - Create a BookService class.
     - Implement and test two methods in the BookService class (e.g., addBook, getBookById).
3. **Create the Layered Architecture for the Member Table:**
   - **Member Model:**
     - Define the Member class with attributes such as memberId, name, email, membershipDate.
   - **MemberDAO Interface:**
     - Create an interface MemberDAO with CRUD operations.
   - **MemberDAOImpl**:
     - Implement the MemberDAO interface in the MemberDAOImpl class.
   - **Service Class:**
     - Create a MemberService class.
     - Implement and test two methods in the MemberService class (e.g., addMember, getMemberById).

4. **Extend to One More Table:**

    - Choose another table (e.g., Author).
    - Repeat the steps for creating the model, DAO interface, DAO implementation, and service class.

5. **Optional**:

    - **Complete the Layered Architecture for the Last Table**:
        - If time permits, complete the layered architecture for any remaining table in the database.
    - Create a Main-Class, which represents an Application. The Application wants to create a Member and borrow a book.
      Use the Service-Classes to complete this task in the Main-Method of the Main-Class