**SE-Course Project (PhasEs 1 & 2 COVER sHEET)**

**Discussions Scheduled for Week 14 *(Monday | Tuesday)***

* Print 1 copies of this cover sheet and attach both to a printed copy of the documentation *(SRS, … etc.)*. You must submit a CD including softcopies of all your documents and Project implementation.
* Please write all your names in English.
* Please make sure that your students’ IDs are correct.
* Handwritten Signatures for the attendance of all team members should be filled in the cover sheet copy before the discussion.
* Please attend the discussion on time *(announced separately)*, late teams will lose 3 grades.

**Project Name: LIBRARY MANGEMNT SYSTEM**

**Project Leader Name: Youssef Hamdi Abdelaziz**

**Team Information *(typed not handwritten, except for the attendance signature)*:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ID**  **[Ordered by ID]** | **Full Name**  **[In English]** | **Attendance**  **[Handwritten Signature]** | **Final Grade** |
| **1** | 320220133 | Youssef hamdi abdelaziz |  |  |
| **2** | 320220137 | Abdelrahman Mahmoud Ramadan |  |  |
| **3** | 320220151 | Hazem Hossam eldin |  |  |
| **4** | 320220158 | Rowan ibrahim |  |  |
| **5** |  |  |  |  |
| **6** |  |  |  |  |

**Grading Criteria:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | | **Actual Grade** | **Notes** |
| **Functional Requirements** | **1** |  |  |
| **Non-Functional Requirements** | **1** |  |  |
| **Use-Case Diagram(s)** *including general use-cases for the system, and the detailed use-cases description* | **1** |  |  |
| **Activity Diagram(s)** | **1** |  |  |
| **Database Specification** *(ERD, Tables)* | **1** |  |  |
| **System Architecture –** *including applied Architectural Pattern(s)* | **1** |  |  |
| **Sequence Diagram(s)** | **3** |  |  |
| **System Sequence Diagrams (SSDs)** | **2** |  |  |
| **Collaboration/Communication Diagram(s)** | **3** |  |  |
| **Class Diagram *(2 versions)***   1. **An initial version based on the requirements and Use-Case/Activity diagrams.** 2. **An intermediate version based on the interaction diagrams.** | **4** |  |  |
| **Object Diagrams** *(Including object diagrams that illustrate the preconditions and the post-conditions of selected functions)* | **2** |  |  |
| **Package Diagram(s)** | **2** |  |  |
| **Self-Study Component 1: State-Machine Diagrams** *(for selected state-dependent objects)* | **0.5** |  |  |
| **Self-Study Component 2: Deployment diagram(s)** | **0.5** |  |  |
| **Front End Design for all Functions** Desk-Top **or** *(HTML, Bootstrap).* | **2** |  |  |
| **Implementation based on the submitted Requirements & Design. Should include at least 4 of the following modules (in addition of course to modules specific to your individual projects):**   1. **User Role Management Module.** 2. **User manipulation Module** *(Login, Add / Delete / Update / Search, List).* 3. **Controlling Resources Module** *(Rooms, Orders, Products, ... etc.).* 4. **Reservation and Rescheduling Module.** 5. **Generating Reports Module** *(PDFs, … etc.).* 6. **Sending Emails or Notifications Module.** | **6** |  | **1** *per module* |
| **Presentation Skills** | **2** |  |  |

**33**

**Teaching-Assistant’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

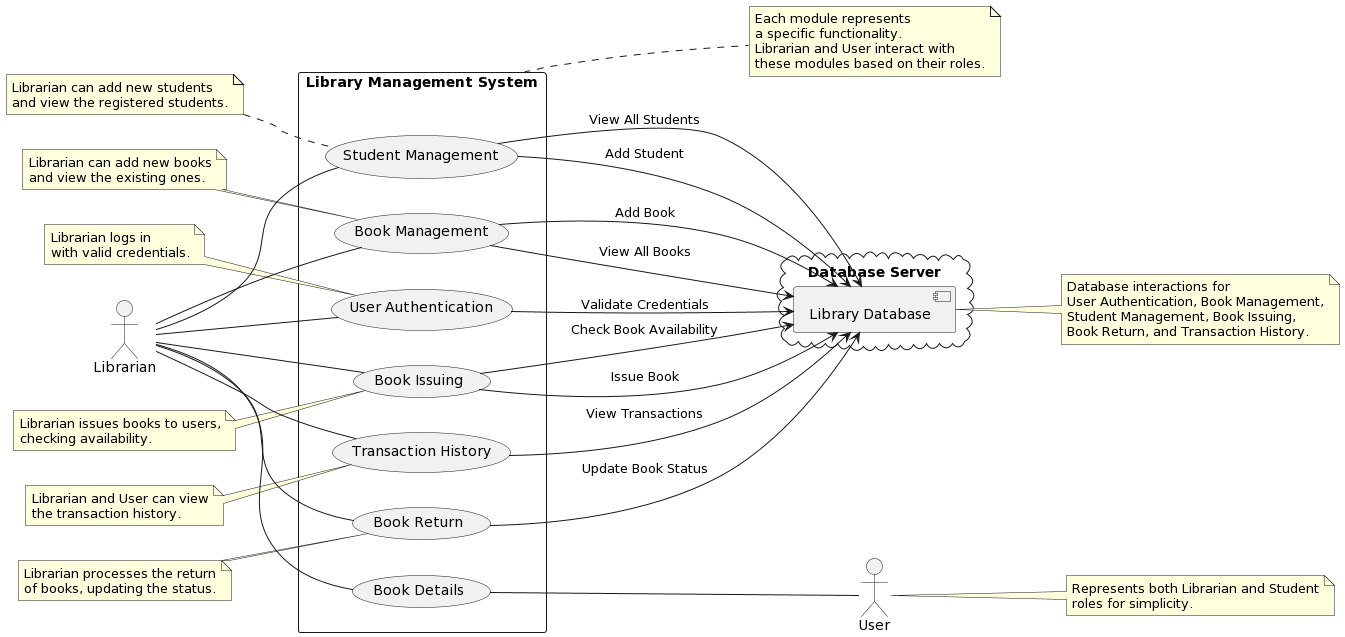
**Functional Requirements:**

1. **User Authentication:**
   * Librarian should be able to log in using a valid username and password.
   * Librarian should have the option to sign up for a new account.
2. **Book Management:**
   * Librarian can add new books to the system with details such as title, author, ISBN, genre, and quantity available.
   * Librarian can view the list of all books available in the library.
   * Each book should have a unique identifier.
3. **Student Management:**
   * Librarian can add new students to the system with details such as student ID, name, contact information, and class.
   * Librarian can view the information of all registered students.
4. **Book Issuing:**
   * Librarian can issue a book to a student, with a maximum limit of 3 books per student.
   * The system should track the issuance date and due date for each book.
5. **Book Return:**
   * Librarian can process the return of a book, updating the system to reflect that the book is now available.
   * The system should track the return date.
6. **Book Details:**
   * Librarian can view detailed information about a specific book, including its availability status and the students to whom it is currently issued.
7. **Transaction History:**
   * The system should maintain a transaction history, logging details of each book issuance and return, including the student and librarian involved, as well as dates.

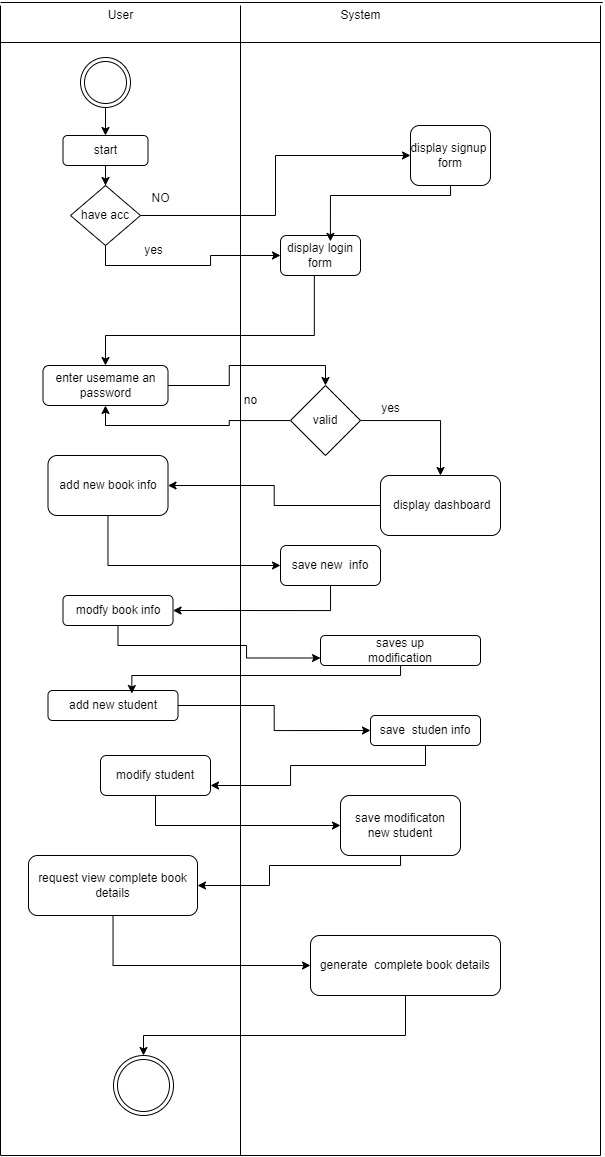
**Non-functional Requirements:**

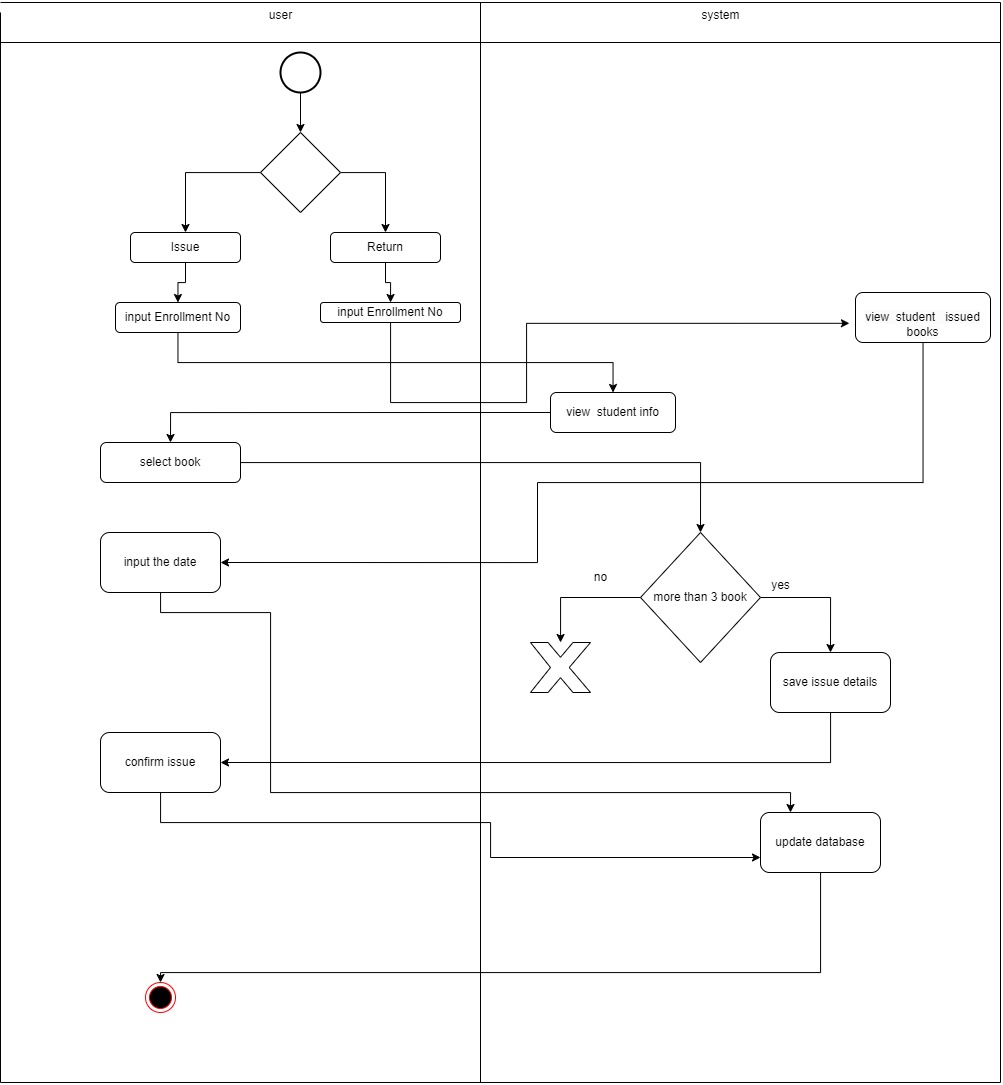
1. **Security:**
   * User authentication information should be securely stored and transmitted.
   * Access to sensitive information (such as student details and transaction history) should be restricted to authorized users.
2. **Performance:**
   * The system should respond to user requests promptly, ensuring a smooth and efficient experience for the librarian.
3. **Scalability:**
   * The system should be designed to handle a growing number of books, students, and transactions without a significant degradation in performance.
4. **Reliability:**
   * The system should be reliable, with minimal downtime for maintenance and updates.
5. **User Interface:**
   * The user interface should be intuitive and user-friendly to facilitate easy navigation for the librarian.
6. **Data Backup:**
   * Regular automated backups of the system's data should be performed to prevent data loss in case of system failures.
7. **Compatibility:**
   * The application should be compatible with various devices and browsers commonly used by librarians.
8. **Reporting:**
   * The system should have reporting capabilities, allowing librarians to generate reports on book availability, overdue books, and other relevant statistics

**Use case Diagram:**

****

**Activity Diagram:**

****

****

**Database Specification *(ERD, Tables)***

***Erd:***

***A diagram of a computer

Description automatically generated***

***Tables:***

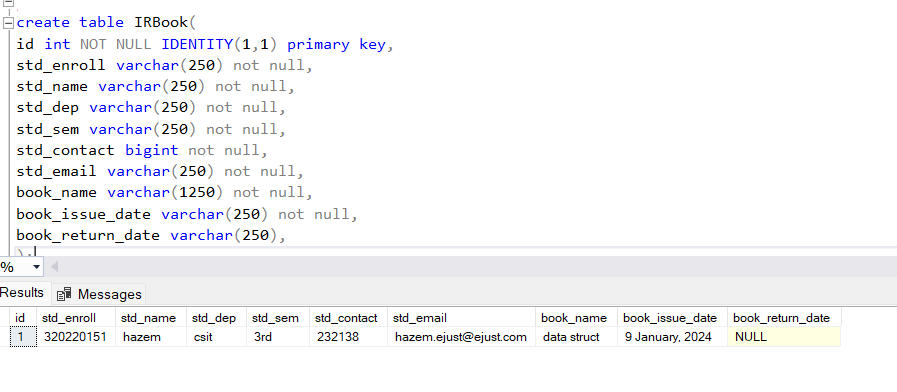
***A screenshot of a computer

Description automatically generated*** ***A screenshot of a computer

Description automatically generated***

**A screenshot of a computer

Description automatically generated**

****** **A screenshot of a computer

Description automatically generated**

**System Architecture**

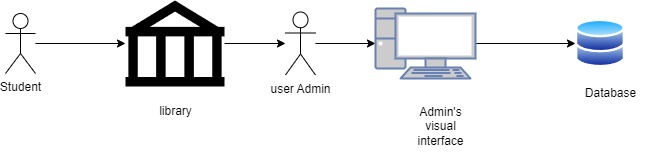
This layer will have different interfaces for each user type.

**Librarian Interface:** Allows librarians to log in, manage books, manage students, issue and return books, and view transaction history.

**Student Interface:** Enables students to log in, view available books, check their issued books, and return books.

**Library User Interface:** Provides library users with restricted access to search and view available books.

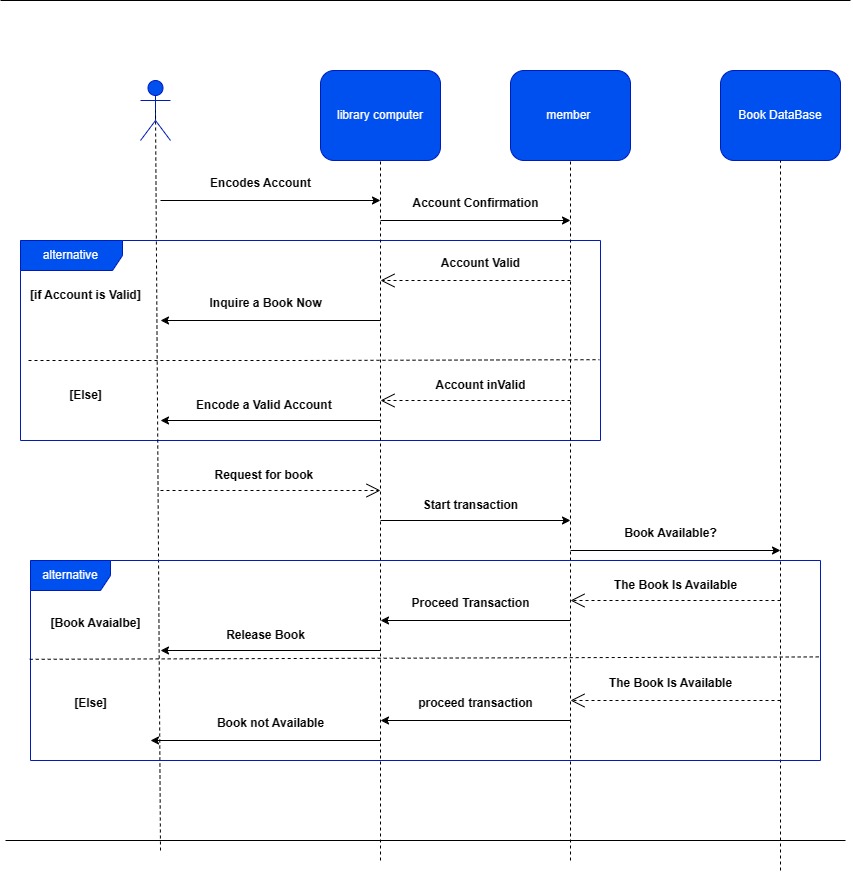
**Admin Interface**: Allows admin admins to manage librarian accounts and perform system-wide configurations

****

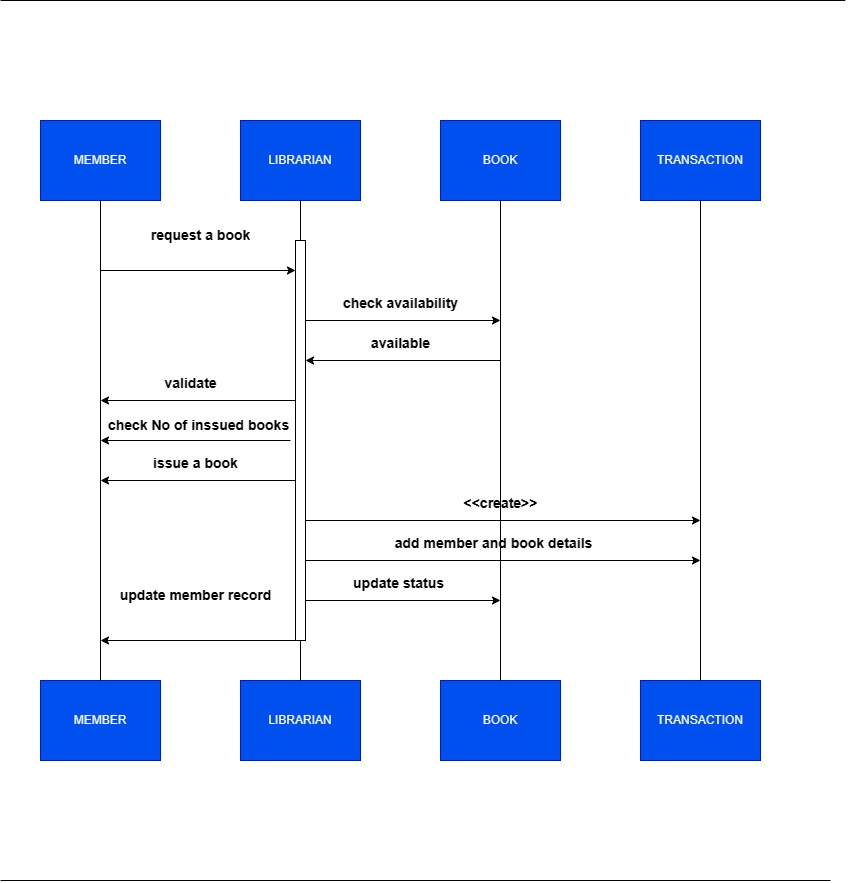
**A diagram of a diagram

Description automatically generated**

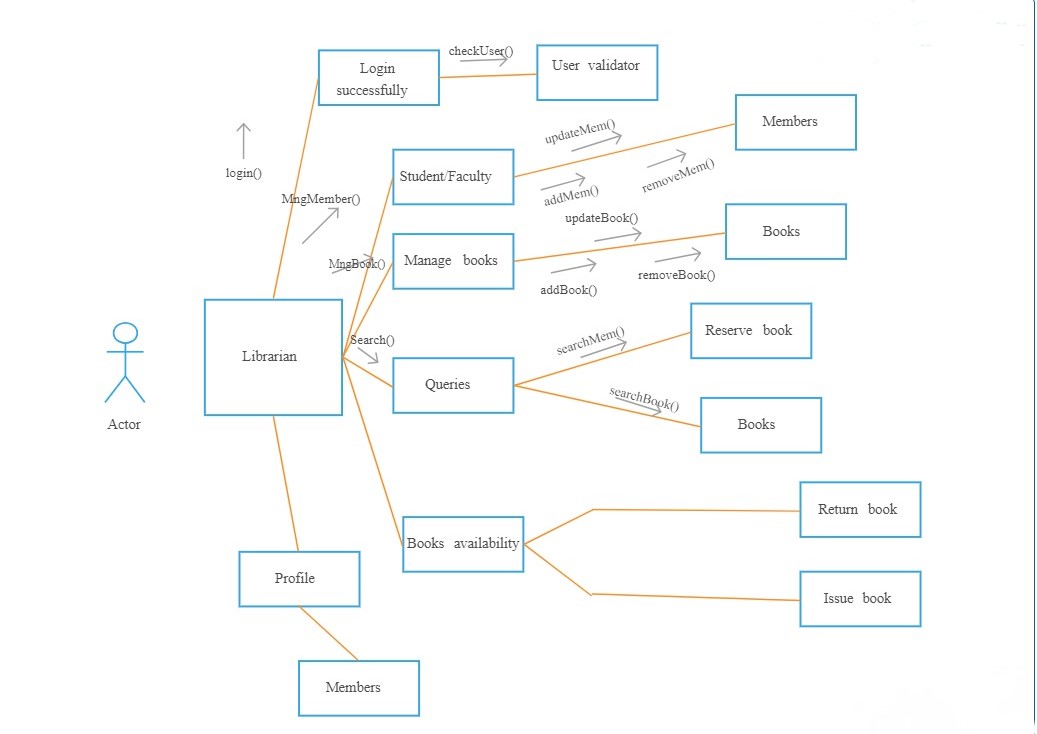
**Sequence Diagram(s)**

****

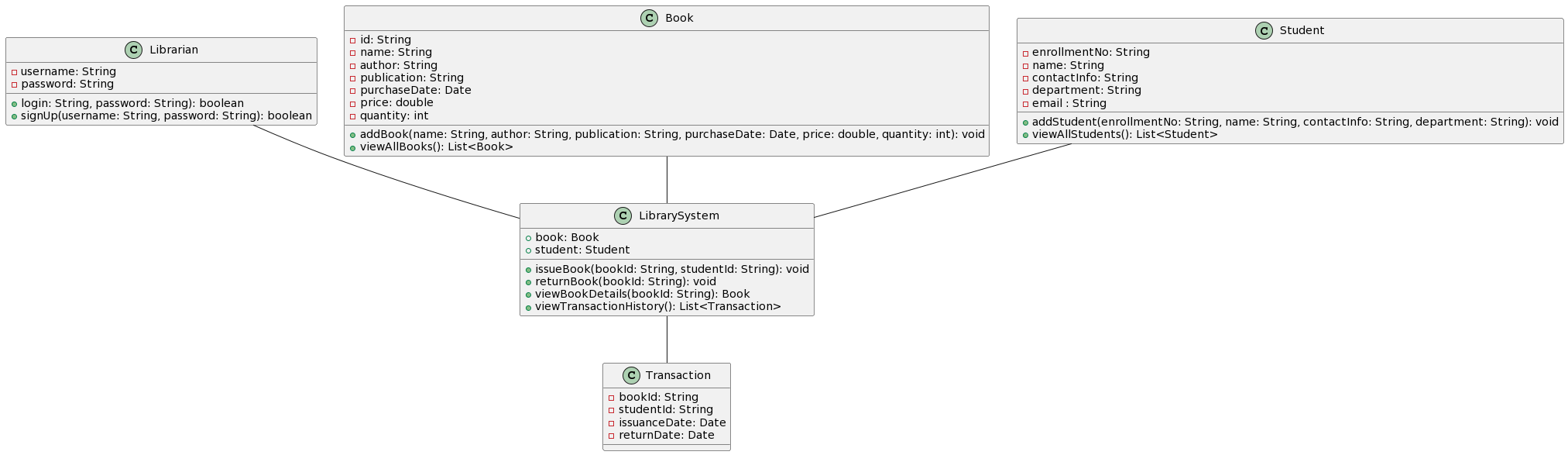
**System Sequence Diagram(s)**

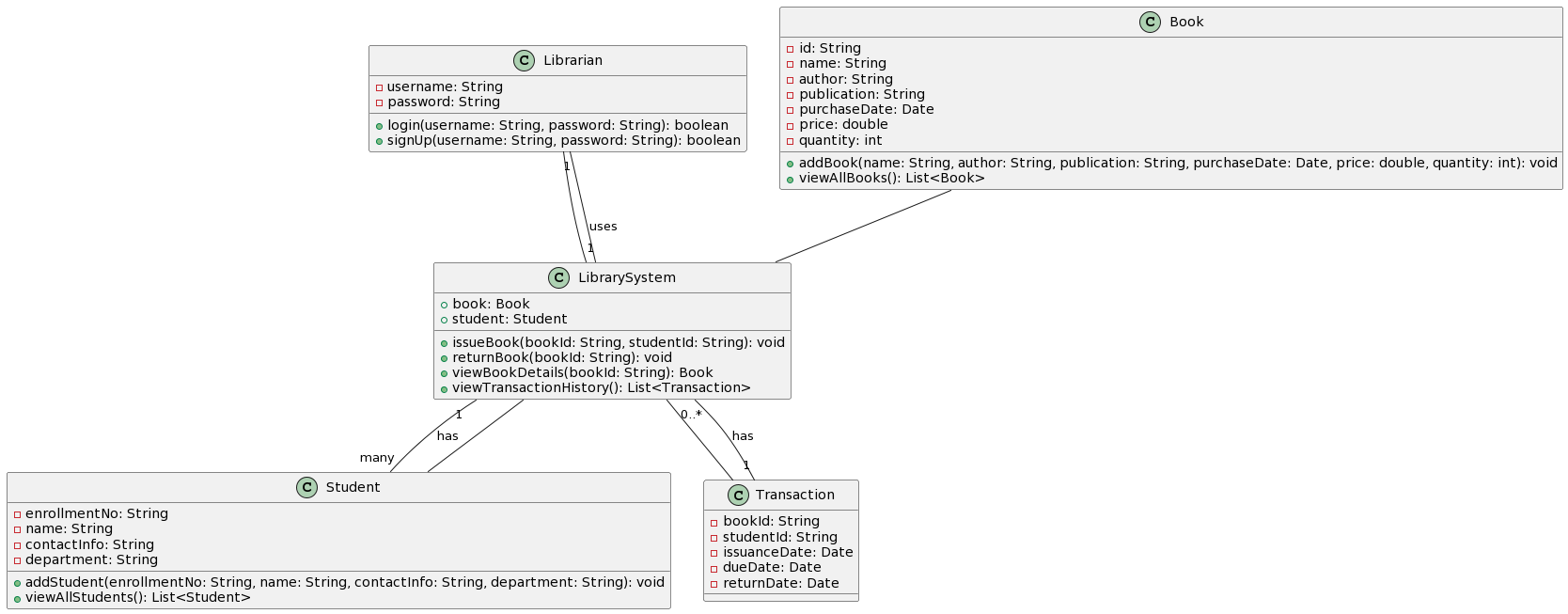
****

**Collaboration/Communication Diagram(s)**

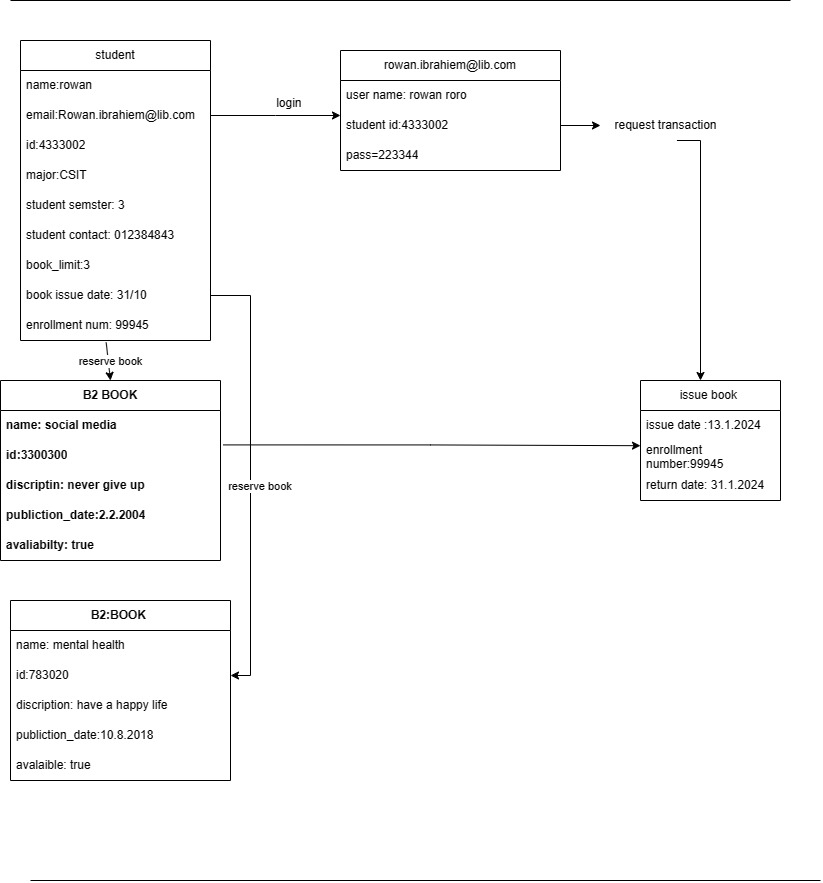
****

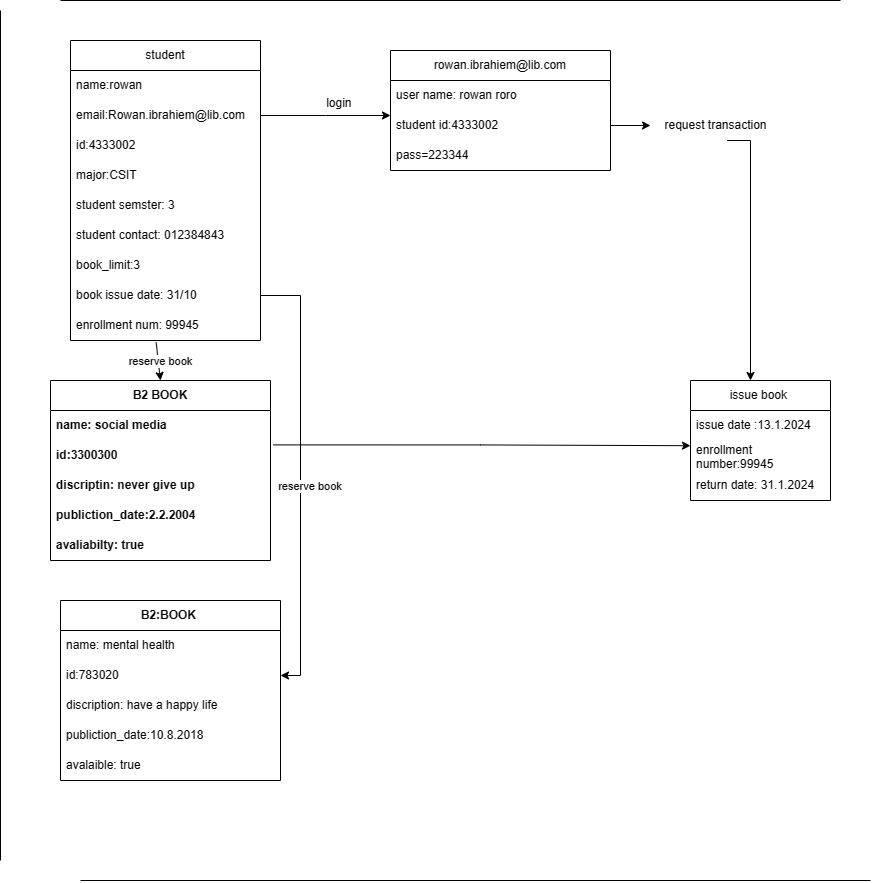
**Class Diagram:\*(initial)**

****

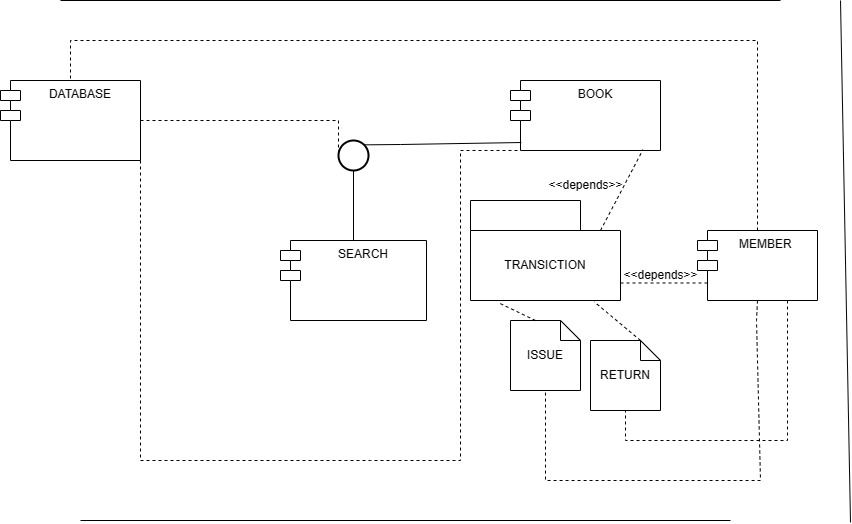
**(intermediate):**

**Object Diagrams:**

****

****

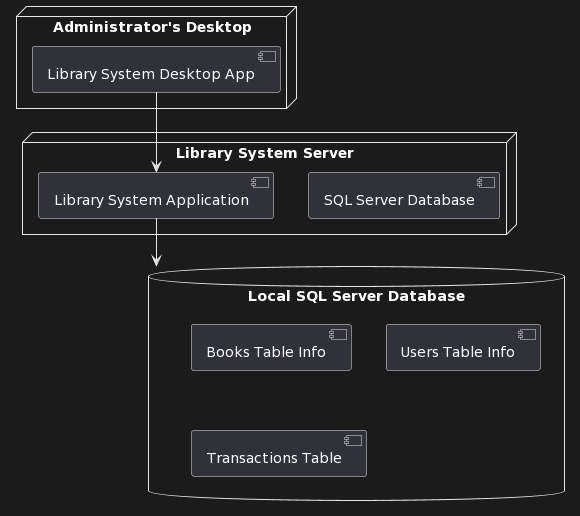
**Package Diagram(s):**

****

**Self-Study Component 1: State-Machine Diagrams** *(for selected state-dependent objects)*

**

**Self-Study Component 2: Deployment diagram(s)**

****

**System SnapShots:**

**A screenshot of a login screen

Description automatically generatedA screenshot of a login screen

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a book purchase form

Description automatically generated**

**A computer screen shot of a student

Description automatically generated**

**A screenshot of a computer

Description automatically generated**