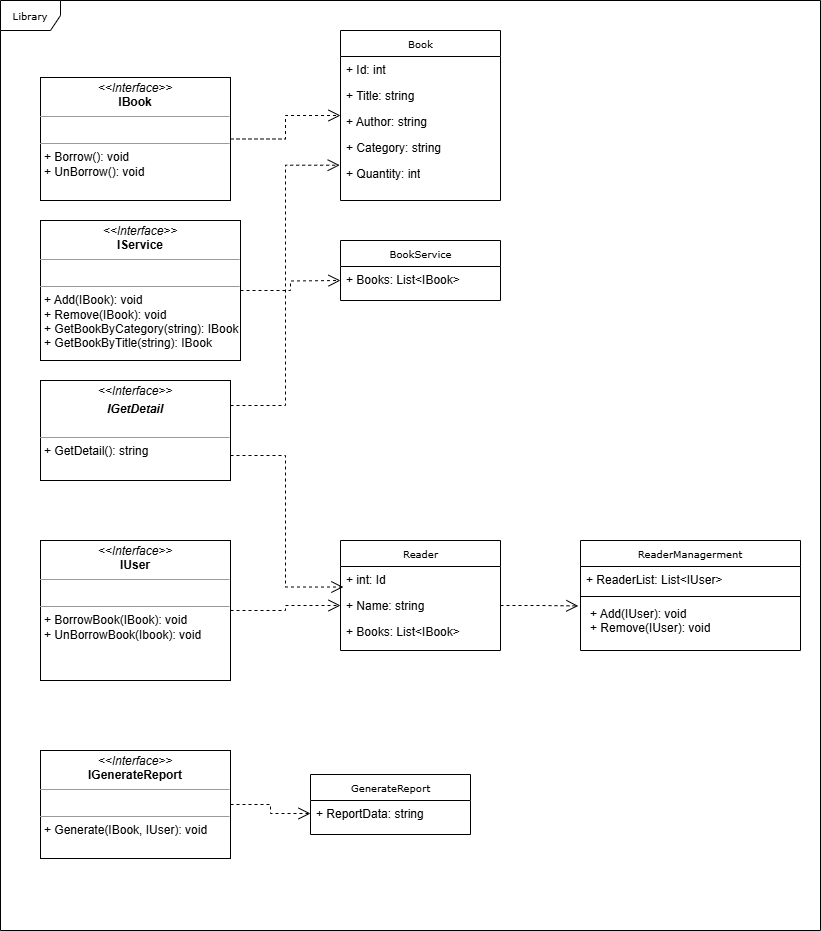
**REPORT**

1. UML Diagram



1. Explain how SOLID principles are applied in the design.
2. ***Single Responsibility Principle (SRP)***

Each class and interface in the diagram has a distinct responsibility:

* Book: Manages book attributes.
* Reader: Manages reader attributes.
* BookService: Manages a list of books.
* ReaderManagement: Manages a list of readers.
* GenerateReport: Handles report generation.

1. ***Open/Closed Principle (OCP)***

Allows for extension without modification:

* New functionalities can be added by implementing existing interfaces (like IBook, IService, IUser), without altering the existing code.

1. ***Liskov Substitution Principle (LSP)***

I uses interfaces to ensure derived classes can be substituted for their base interfaces:

* Any class implementing the IBook interface can be used wherever IBook is expected.

1. ***Interface Segregation Principle (ISP)***

I uses multiple specific interfaces rather than a single general-purpose interface:

* Separate interfaces like IBook, IService, IUser, IGetDetail, and IGenerateReport cater to different aspects of the library system.

1. ***Dependency Inversion Principle (DIP)***

My design depends on abstractions rather than concrete implementations:

* For example, the BookService class depends on the IBook interface rather than a specific Book class.
* Similarly, ReaderManagement depends on the IUser interface.

1. Ways to extend the system in the future.

* Extend the system to manage other resources such as e-books, magazine: Add classes like EBook, implementing a IBook interface.
* Enhance security by implementing user roles and permissions: Create a RoleManagement class to handle role-based access control.
* Implement advanced search and filtering capabilities to enhance user experience.
* Add functionality to notify users about due dates, the books that they like are borrowed.