***Design Choices:***

Class Structure: The design follows a simple object-oriented structure with classes for Item, Book, CD, DVD, and Person. Each class represents real-world entities, and inheritance is used where applicable to model the "is-a" relationships.

Item Management: I've implemented functions to add, delete, check availability, check out, and return items. These operations are performed efficiently with the help of the library's data structures (vectors).

Account Management: The system allows creating and closing library accounts. A person can borrow items, and each person's account keeps track of their borrowed items and account details.

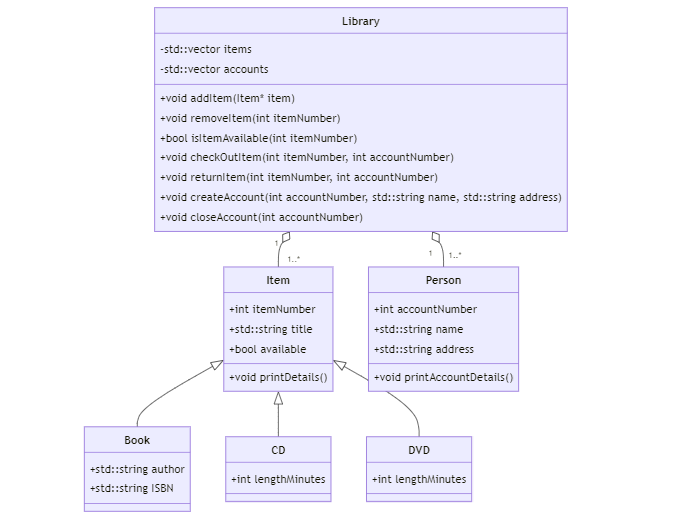
Search Functionality: Currently, the system can search for items by title, search for books by author, and search for items by type (Book, CD, DVD).

Error Handling: The code includes basic error checking and gracefully handles different error cases. For example, items cannot be deleted if they are currently checked out.

Input Validation: The code prompts the user for input and validates it to ensure the correct data type and range.

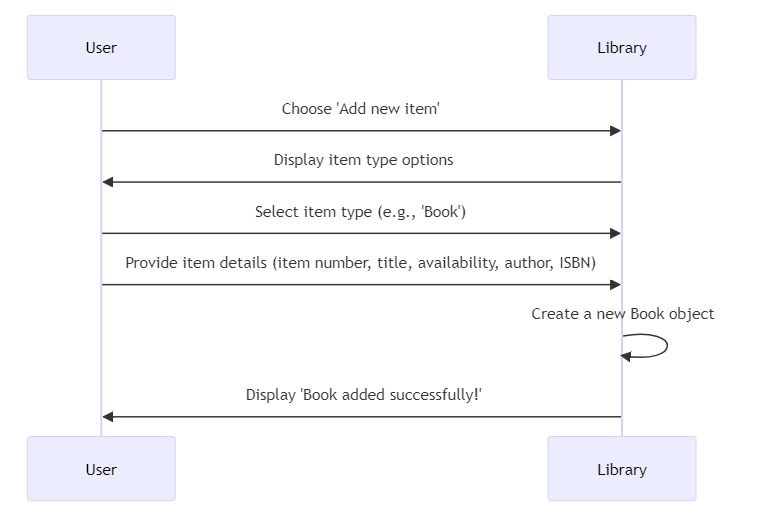
**UML Diagrams:**

***Class Diagram:***

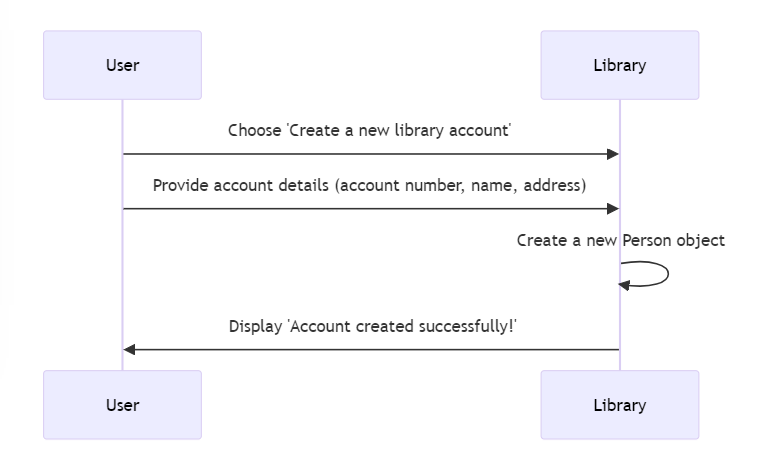


***Sequence Diagram:***

1) **Adding an Item Sequence Diagram:**



2) **Creating an Account Sequence Diagram:**



***Output:***

**Add a Book**

