Design a Library Management System

A Library Management System is a software built to handle the primary housekeeping functions of a library. Libraries rely on library management systems to manage asset collections as well as relationships with their members. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles.

Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.

System Requirements

checked-out by a specific library member.

```
Always clarify requirements at the beginning of the interview. Be sure to ask questions to find the exact
scope of the system that the interviewer has in mind.
```

We will focus on the following set of requirements while designing the Library Management System: 1. Any library member should be able to search books by their title, author, subject category as well by the

- publication date. 2. Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.
- 3. There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item.
- 5. There should be a maximum limit (5) on how many books a member can check-out. 6. There should be a maximum limit (10) on how many days a member can keep a book.

4. The system should be able to retrieve information like who took a particular book or what are the books

- 7. The system should be able to collect fines for books returned after the due date. 8. Members should be able to reserve books that are not currently available.

9. The system should be able to send notifications whenever the reserved books become available, as well

- as when the book is not returned within the due date. 10. Each book and member card will have a unique barcode. The system will be able to read barcodes from books and members' library cards.
- Librarian: Mainly responsible for adding and modifying books, book items, and users. The Librarian can
- also issue, reserve, and return book items. • Member: All members can search the catalog, as well as check-out, reserve, renew, and return a book.

- Add/Remove/Edit book: To add, remove or modify a book or book item. • **Search catalog:** To search books by title, author, subject or publication date.
- Here are the top use cases of the Library Management System:

Class diagram Here are the main classes of our Library Management System: like 'Name' to distinguish it from any other libraries and 'Address' to describe its location. Each book item will have a unique barcode. will be a librarian.

address: Address blockMember(): bool description: string unblockMember(): bool getaName(): string getAddress(): Address getTotalCheckedoutBooks(): int Extends Extends

0..5

0..5

-borrows

-reserves

Account

id: string

password: string

person: Person

status: AccountStatus

BookItem

barcode: string

borrowed: date

dueDate: date

price: double

Extends

isReferenceOnly: bool

LibraryCard

cardNumber: string

barcode: string

issuedAt: date

active: bool

Book

ISBN: string

title: string

subject: string

publisher: string

language: string

numberOfPages: int

resetPassword(): bool isActive(): bool format: BookFormat getTitle(): string status: BookStatus scans dateOfPurchase: date makes against publicationDate: date Rack 0..5 placed at number: int checkout(): bool BarcodeReader BookReservation id: string locationIdentifier: string creationDate: date registeredAt: date status: ReservationStatus active: bool Catalog getStatus(): ReservationStatus isActive(): bool creationDate: date fetchReservationDetails(): BookReservation <<interface>> totalBooks: int Search searchBvTitle(string) **BookLending** bookTitles: Map<string, list> searchByAuthor(string) * creationDate: date searchBySubject(string) bookAuthors: Map<string, list> searchByPubDate(datetime) dueDate: date bookSubjects: Map<string, list> returnDate: date bookPublicationDates: Map<date, list> Notification getReturnDate(): date updateCatalog(): bool notificationId: int createdOn: date 0..1 content: string sendNotification(): bool amount: double CreditCardTransaction getAmount(): double nameOnCard: string PostalNotification Extends CheckTransaction address: Address bankName: string **FineTransaction** Extends checkNumber: string creationDate: date EmailNotification Extends amount: double CashTransaction cashTendered: double initiateTransaction(): bool Class diagram for Library Management System **UML** conventions <<interface>> <u>Name</u> Interface: Classes implement interfaces, denoted by Generalization. method1() ClassName property_name: type Class: Every class can have properties and methods. Abstract classes are identified by their *Italic* names. method(): type Generalization: A implements B. Inheritance: A inherits from B. A "is-a" B. Use Interface: A uses interface B. Association: A and B call each other. Uni-directional Association: A can call B, but not vice versa. Aggregation: A "has-an" instance of B. B can exist without A. Composition: A "has-an" instance of B. B cannot exist without A. **Activity diagrams Check-out a book:** Any library member or librarian can perform this activity. Here are the set of steps to

> Member scans their library card through barcode reader

Member scans the barcode of the book

System checks if the book can be issued and that the book is not 'reference only'?

[yes]

[else]

[no]

System creates book checkout transaction

System marks any reservation 'Completed' that the member

members if they return books after the due date. Here are the steps for returning a book:

System checks the number of books issued to the member

System checks if the book has been reserved by any other member?

[no]

[max quota exceeded]

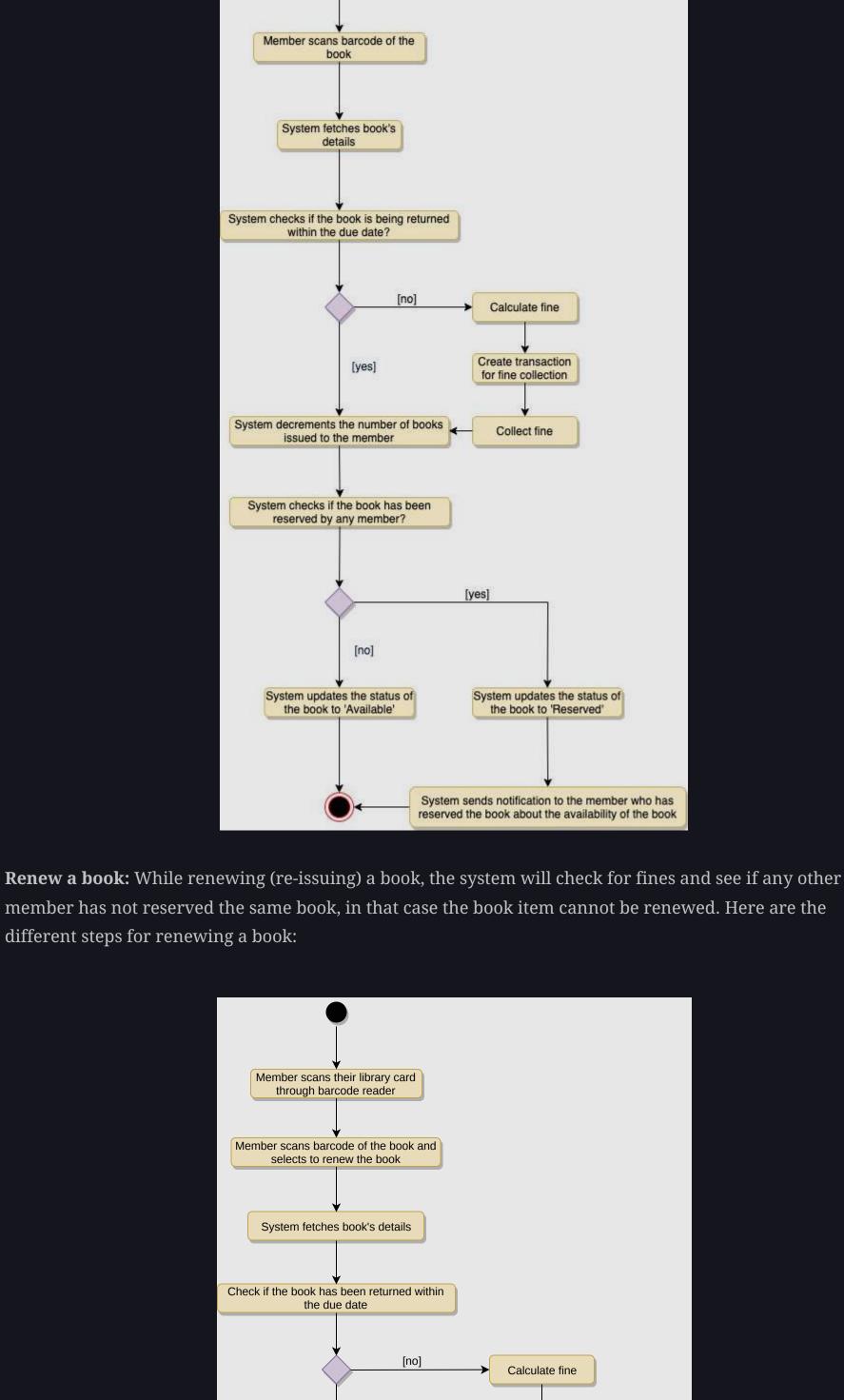
[yes]

Show error message

System updates the status of the book to 'Loaned' System increments number of books issued to the member

check-out a book:

had made against the book Return a book: Any library member or librarian can perform this activity. The system will collect fines from



Update the status of the book to 'Reserved'

Note: This code only focuses on the design part of the use cases. Since you are not required to write a fully

executable code in an interview, you can assume parts of the code to interact with the database, payment

Enums and Constants: Here are the required enums, data types, and constants:

[yes]

[yes]

[no]

See if the book has been reserved

by any other member

Create book checkout transaction

with new due date

Code

system, etc.

1. Check-out a book,

2. Return a book, and

? Python

1 public enum BookFormat {

HARDCOVER, PAPERBACK, AUDIO_BOOK,

NEWSPAPER, MAGAZINE, JOURNAL

AVAILABLE,

LOANED,

19 WAITING,

21 CANCELED,

28 CANCELED, 29 BLACKLISTED,

private String password;

public boolean addBookItem(BookItem bookItem);

public boolean blockMember(Member member);

public boolean unBlockMember (Member member);

public int getTotalBooksCheckedout();

public boolean reserveBookItem(BookItem bookItem);

private void incrementTotalBooksCheckedout();

22 public class Member extends Account { private Date dateOfMembership;

collection, respectively.

🤁 Python

1 public class BookReservation {

private String bookItemBarcode;

private String memberId;

22 private Date creationDate;

24 private String memberId;

private Person person;

3. Renew a book.

Here is the code for the use cases mentioned above:

Create transaction

for fine collection

Collect fine

Show error message that the

book can't be issued

Send notification to the member who has reserved the book that the book has become available

```
18 public enum ReservationStatus{
Account, Member, and Librarian: These classes represent various people that interact with our system:
             🤁 Python
   6 private String id;
```

private Date creationDate; private ReservationStatus status; private String bookItemBarcode; private String memberId; public static BookReservation fetchReservationDetails(String barcode); 10 public class BookLending {

public static boolean lendBook(String barcode, String memberId); public static BookLending fetchLendingDetails(String barcode);

public static void collectFine(String memberId, long days) {}

BookReservation, BookLending, and Fine: These classes represent a book reservation, lending, and fine

```
BookItem: Encapsulating a book item, this class will be responsible for processing the reservation, return,
and renewal of a book item.
             🤁 Python
  🍨 Java
        private String ISBN;
        private String title;
        private String subject;
        private String publisher;
        private String language;
        private int numberOfPages;
        private String barcode;
        private boolean isReferenceOnly;
        private Date borrowed;
        private Date dueDate;
        private double price;
        private BookFormat format;
        private BookStatus status;
        private Date dateOfPurchase;
        private Date publicationDate;
        private Rack placedAt;
        public boolean checkout(String memberId) {
        if(bookItem.getIsReferenceOnly()) {
          if(!BookLending.lendBook(this.getBarCode(), memberId)){
          this undataRookItamStatus (RookStatus IOANFD) .
books.
```

Search interface and Catalog: The Catalog class will implement the Search interface to facilitate searching of 🦰 Python public List<Book> searchByTitle(String title); public List<Book> searchByAuthor(String author); public List<Book> searchBySubject(String subject); public List<Book> searchByPubDate(Date publishDate); 8 public class Catalog implements Search { private HashMap<String, List<Book>> bookTitles; private HashMap<String, List<Book>> bookAuthors; private HashMap<String, List<Book>> bookSubjects; private HashMap<String, List<Book>> bookPublicationDates; public List<Book> searchByTitle(String query) { return bookTitles.get(query); public List<Book> searchByAuthor(String query) { return bookAuthors.get(query);

```
Use case diagram
We have three main actors in our system:
   • System: Mainly responsible for sending notifications for overdue books, canceled reservations, etc.
   • Register new account/cancel membership: To add a new member or cancel the membership of an
     existing member.
   • Check-out book: To borrow a book from the library.
   • Reserve book: To reserve a book which is not currently available.
   • Renew a book: To reborrow an already checked-out book.
   • Return a book: To return a book to the library which was issued to a member.
                                                                                 <<include>>
                                                                                              Add book
                                                                Add book item
                                                      <<include>>
                                                       <<include>>
                                        Update catalog
                                                                                  <<include>>
                                                                                             Remove book
                                                               Remove book item
                                                      <<include>>
                                                                                  <<include>>
                                                                                              Edit book
                                                                Edit book item
                                               Search by subject
                                 Search by author
                    Search by book
                        title
                                                <<extend>>
                                    <<extend>>
                          <<extend>>
                                                           Search by
                                                         publication date
                                   Search catalog
                                                                                  <<extend>>
                                                                                              Register new
                                                                   Issue library card
                                                                                                account
                                                              Cancel
                                                            membership
                                                           Register/Update
                                                              account
                                                            Login/Logout
                                                                                                                 Librarian
                                                              <<include>>
                                           Checkout book
                                                                             Issue book
                                                   <<include>>
                                                              Remove
                                                             reservation
                                                            Renew book
                                                                                           Send overdue
                                                            Reserve book
                                                                                            notification
                                                                                          Send reservation
                                                            View account
                                                                                         available notification
                                                                                                                 System
                                           <<extend>>
                                                                                          Send reservation
                                 Pay fine
                                                            Return book
                                                                                         canceled notification
                                                        Use case diagram
   • Library: The central part of the organization for which this software has been designed. It has attributes
   • Book: The basic building block of the system. Every book will have ISBN, Title, Subject, Publishers, etc.
   • BookItem: Any book can have multiple copies, each copy will be considered a book item in our system.
   • Account: We will have two types of accounts in the system, one will be a general member, and the other
   • LibraryCard: Each library user will be issued a library card, which will be used to identify users while
     issuing or returning books.
   • BookReservation: Responsible for managing reservations against book items.
   • BookLending: Manage the checking-out of book items.
   • Catalog: Catalogs contain list of books sorted on certain criteria. Our system will support searching
     through four catalogs: Title, Author, Subject, and Publish-date.
   • Fine: This class will be responsible for calculating and collecting fines from library members.
   • Author: This class will encapsulate a book author.
   • Rack: Books will be placed on racks. Each rack will be identified by a rack number and will have a
     location identifier to describe the physical location of the rack in the library.
   • Notification: This class will take care of sending notifications to library members.
            <<enumeration>>
                              <<enumeration>>
                                               <<enumeration>>
                                                                 <<enumeration>>
                                                                                     <<dataType>>
                                                                                                       <<dataType>>
              BookFormat
                                               ReservationStatus
                                                                  AccountStatus
                               BookStatus
                                                                                      Address
                                                                                                         Person
            Hardcover
                             Available
                                                                 Active
                                                                                  streetAddress: string
                                                                                                     name: string
                                               Pending
            Paperback
                                                                                  city: string
                                                                 Closed
                                                                                                     address: Address
                             Reserved
            Audiobook
                             Loaned
                                               Completed
                                                                 Canceled
                                                                                  state: string
                                                                                                     email: string
            Ebook
                             Lost
                                               Canceled
                                                                 Blacklisted
                                                                                  zipcode: string
                                                                                                     phone: string
            Newspaper
                                                                 None
                                                                                  country: string
                                               None
            Magazine
            Journal
            Author
                                           Library
                                                                     Librarian
                                                                                              Member
                                                              addBookItem(): bool
                                                                                      dateOfMembership: date
    name: string
                                   name: string
                                                                                      totalBooksCheckedout: int
```