Library Management System Report

The goal of this assignment is to create a Library Management System using Java's Object-Oriented Programming (OOP) principles. The system enables users to manage books and members, track the borrowing and returning of books, and display information about available books and registered members. By leveraging OOP concepts such as abstraction, inheritance, and encapsulation, the system provides a structured and efficient solution for library operations.

Approach

To achieve the required functionality, the system follows a systematic approach based on OOP principles:

1. Abstract Class - LibraryItem

- Declared fields: title, author, itemID.
- The constructor initializes these fields.
- It has an abstract method displayDetails().

2. Subclasses - Book and Magazine

- **Book:** Adds a field ISBN and overrides displayDetails().
- **Magazine:** Adds a field issueNumber and overrides displayDetails().

3. Interface - LibraryOperations

- Specifies borrowItem(), returnItem(), and displayAvailableItems() methods.

4. Class - Library

- Implements LibraryOperations.

- Maintains lists of books and magazines.

---

5. **Syed Muhammad Oan Kazmi**

- Manages item borrowing and return while ensuring members are registered.

6. Class - Member

- Declares fields name and memberID.
- Has a displayMemberDetails() method.

7. Main Method Implementation

- Instantiates different Book and Magazine objects.
- Registers members.
- Demonstrates borrowing and returning functionality.
- Displays available books and member details.

Problems Faced

- **Managing Borrowed Items:**

  Ensuring an item is not borrowed twice required the addition of a HashMap.

- **User Input Validation:**

  Verifying user inputs for borrowing and returning books required error handling.

- **Encapsulation:**

Maintaining data integrity using getter methods and private fields.

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;



//Main class to demonstrate functionality

public class LibraryManagementSystem {

public static void main(String[] args) {

  Library library = new Library();


  // Creating Books and Magazines

  Book book1 = new Book("Atomic Habits", " ali ahmed", 10, "noedf");

  Book book2 = new Book("The Catcher in the Rye", "Rye", 20, "7");

  Magazine mag1 = new Magazine("In Search of Gold", "antoni
spears",30 , 8);


  library.addItem(book1);

  library.addItem(book2);

  library.addItem(mag1);


  // Creating Members
```

```
    Member member1 = new Member("saim", 1);
    Member member2 = new Member("saeed", 2);

    // Display available items
    library.displayAvailableItems();

    // Borrowing and Returning
    library.borrowItem(101, "saim");
    library.displayAvailableItems();
    library.returnItem(101, "saeed");
    library.displayAvailableItems();

    // Displaying Member Details
    member1.displayMemberDetails();
    member2.displayMemberDetails();
}
}


//Abstract class LibraryItem
abstract class LibraryItem {
 protected String title;
 protected String author;
```

```
 protected int itemID;

 public LibraryItem(String title, String author, int itemID) {
    this.title = title;
    this.author = author;
    this.itemID = itemID;
 }

 public abstract void displayDetails();
}


//Book subclass
class Book extends LibraryItem {
 private String ISBN;

 public Book(String title, String author, int itemID, String ISBN) {
    super(title, author, itemID);
    this.ISBN = ISBN;
 }

 @Override
 public void displayDetails() {
```

```
    System.out.println("Book: " + title + ", Author: " + author + ", ID: " +
itemID + ", ISBN: " + ISBN);
 }
}



//Magazine subclass
class Magazine extends LibraryItem {
 private int issueNumber;

 public Magazine(String title, String author, int itemID, int
issueNumber) {
    super(title, author, itemID);
    this.issueNumber = issueNumber;
 }

 @Override
 public void displayDetails() {
    System.out.println("Magazine: " + title + ", Author: " + author + ", ID:
" + itemID + ", Issue: " + issueNumber);
 }
}
```

```
//Interface LibraryOperations
interface LibraryOperations {
 void borrowItem(int itemID, String memberName);
 void returnItem(int itemID, String memberName);
 void displayAvailableItems();
}



//Library class implementing LibraryOperations
class Library implements LibraryOperations {
 private List<LibraryItem> items = new ArrayList<>();
 private Map<Integer, String> borrowedItems = new HashMap<>();

 public void addItem(LibraryItem item) {
    items.add(item);
 }

 @Override
 public void borrowItem(int itemID, String memberName) {
    for (LibraryItem item : items) {
        if (item.itemID == itemID &&
!borrowedItems.containsKey(itemID)) {
            borrowedItems.put(itemID, memberName);
```

```
            System.out.println(memberName + " borrowed " + item.title);
            return;
        }
    }
    System.out.println("Item not available for borrowing.");
}

@Override
public void returnItem(int itemID, String memberName) {
    if (borrowedItems.containsKey(itemID) &&
borrowedItems.get(itemID).equals(memberName)) {
        borrowedItems.remove(itemID);
        System.out.println(memberName + " returned item " + itemID);
    } else {
        System.out.println("Invalid return attempt.");
    }
}

@Override
public void displayAvailableItems() {
    System.out.println("Available Items:");
    for (LibraryItem item : items) {
        if (!borrowedItems.containsKey(item.itemID)) {
            item.displayDetails();
```

```
        }
    }
}
}



//Member class
class Member {
 private String name;
 private int memberID;

 public Member(String name, int memberID) {
    this.name = name;
    this.memberID = memberID;
 }

 public void displayMemberDetails() {
    System.out.println("Member Name: " + name + ", ID: " + memberID);
 }
}
```

Screenshots of output:

```
Available Items:
Book: Atomic Habits, Author:  ali ahmed, ID: 10, ISBN: noedf
Book: The Catcher in the Rye, Author: Rye, ID: 20, ISBN: 7
Magazine: In Search of Gold, Author: antoni spears, ID: 30, Issue: 8
Item not available for borrowing.
Available Items:
Book: Atomic Habits, Author:  ali ahmed, ID: 10, ISBN: noedf
Book: The Catcher in the Rye, Author: Rye, ID: 20, ISBN: 7
Magazine: In Search of Gold, Author: antoni spears, ID: 30, Issue: 8
Invalid return attempt.
Available Items:
Book: Atomic Habits, Author:  ali ahmed, ID: 10, ISBN: noedf
Book: The Catcher in the Rye, Author: Rye, ID: 20, ISBN: 7
Magazine: In Search of Gold, Author: antoni spears, ID: 30, Issue: 8
Member Name: saim, ID: 1
Member Name: saeed, ID: 2

=== Code Execution Successful ===
```