Creating library Management System using oop (minor project)

```
import java.util.ArrayList;
import java.util.List;

class Book {
    private int bookId;
    private String bookname;
    private boolean available;

    public Book(int bookId, String bookname) {
        this.bookId = bookId;
        this.bookname = bookname;
        this.available = true;
    }

    public int getBookId() {
        return bookId;
    }
```

```
    public String getTitle() {
        return bookname;
    }

    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }
}

class Member {
    private int memberId;
    private String name;
    private List<Book> borrowedBooks;

    public Member(int memberId, String name) {
        this.memberId = memberId;
        this.name = name;
```

```
        this.borrowedBooks = new ArrayList<>();
    }

    public int getMemberId() {
        return memberId;
    }

    public String getName() {
        return name;
    }

    public List<Book> getBorrowedBooks() {
        return borrowedBooks;
    }

    public void borrowBook(Book book) {
        if (book.isAvailable()) {
            borrowedBooks.add(book);
            book.setAvailable(false);
            System.out.println(name + " has borrowed '" + book.getTitle() +
"'.");
        } else {
```

```
            System.out.println("Sorry, '" + book.getTitle() + "' is not available for borrowing.");
        }
    }

    public void returnBook(Book book) {
        if (borrowedBooks.contains(book)) {
            borrowedBooks.remove(book);
            book.setAvailable(true);
            System.out.println(name + " has returned '" + book.getTitle() + "'.");
        } else {
            System.out.println("Error: Book '" + book.getTitle() + "' is not in the possession of " + name + ".");
        }
    }
}

class Library {
    private List<Book> books;
    private List<Member> members;

    public Library() {
```

```
        this.books = new ArrayList<>();
        this.members = new ArrayList<>();
    }

    public void addBook(Book book) {
        books.add(book);
        System.out.println("Book '" + book.getTitle() + "' has been added to the library.");
    }

    public void addMember(Member member) {
        members.add(member);
        System.out.println("Member '" + member.getName() + "' has been added to the library.");
    }

    public void displayAvailableBooks() {
        System.out.println("Available Books:");
        for (Book book : books) {
            if (book.isAvailable()) {
                System.out.println("  " + book.getTitle());
            }
        }
```

```
    }
}

public class LibraryManagementSystem {
    public static void main(String[] args) {
        Library library = new Library();

        Book book1 = new Book(1, "Introduction to Java");
        Book book2 = new Book(2, "Data Structures and Algorithms");
        Book book3 = new Book(3, "Object-Oriented Programming");

        library.addBook(book1);
        library.addBook(book2);
        library.addBook(book3);

        Member member1 = new Member(101, "Sameer");
        Member member2 = new Member(102, "Sasuke");

        library.addMember(member1);
        library.addMember(member2);

        library.displayAvailableBooks();
```

```
        // Simulating borrowing and returning books
        member1.borrowBook(book1);
        member2.borrowBook(book2);

        library.displayAvailableBooks();

        member1.returnBook(book1);
        member2.returnBook(book2);

        library.displayAvailableBooks();
    }
}
```

Output:

```
Book 'Introduction to Java' has been added to the library.
Book 'Data Structures and Algorithms' has been added to the library.
Book 'oop' has been added to the library.
Member 'Sameer' has been added to the library.
Member 'Sasuke' has been added to the library.
Available Books:
  Introduction to Java
  Data Structures and Algorithms
```

oop

Sameer has borrowed 'Introduction to Java'.

Sasuke has borrowed 'Data Structures and Algorithms'.

Available Books:

oop

Sameer has returned 'Introduction to Java'.

Sasuke has returned 'Data Structures and Algorithms'.

Available Books:

Introduction to Java

Data Structures and Algorithms

Oop

Flow Chart of Program

| Start of Program |

|

V

| Initialize Library |

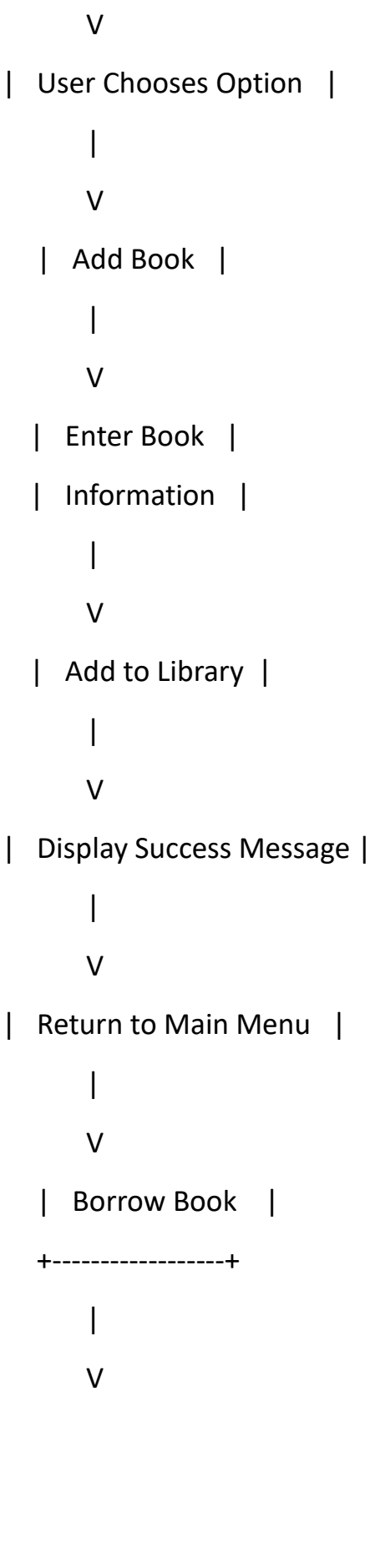
| and Member Objects |

|

V

| Display Main Menu |

|


V
| User Chooses Option |
|
V
| Add Book |
|
V
| Enter Book |
| Information |
|
V
| Add to Library |
|
V
| Display Success Message |
|
V
| Return to Main Menu |
|
V
| Borrow Book |
+----------+
|
V

| Enter Member ID and |

| Book ID for Borrowing |

|

V

| Check Book Availability |

|

V

| Update Records |

|

V

| Display Success Message |

|

V

| Return to Main Menu |

|

V

| Return Book |

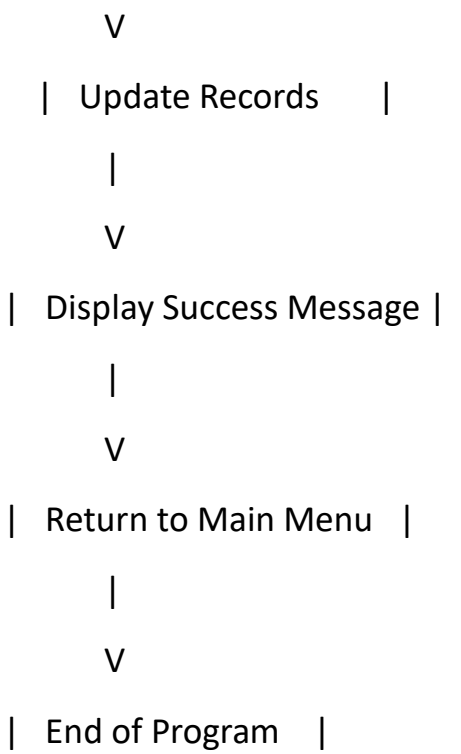
|

V

| Enter Member ID and |

| Book ID for Returning |

|



GitHub Link:

https://github.com/Sameerabd386/Assignment-day-1

LinkedIn Link:

https://www.linkedin.com/in/mohammad-sameer-6aba4222a