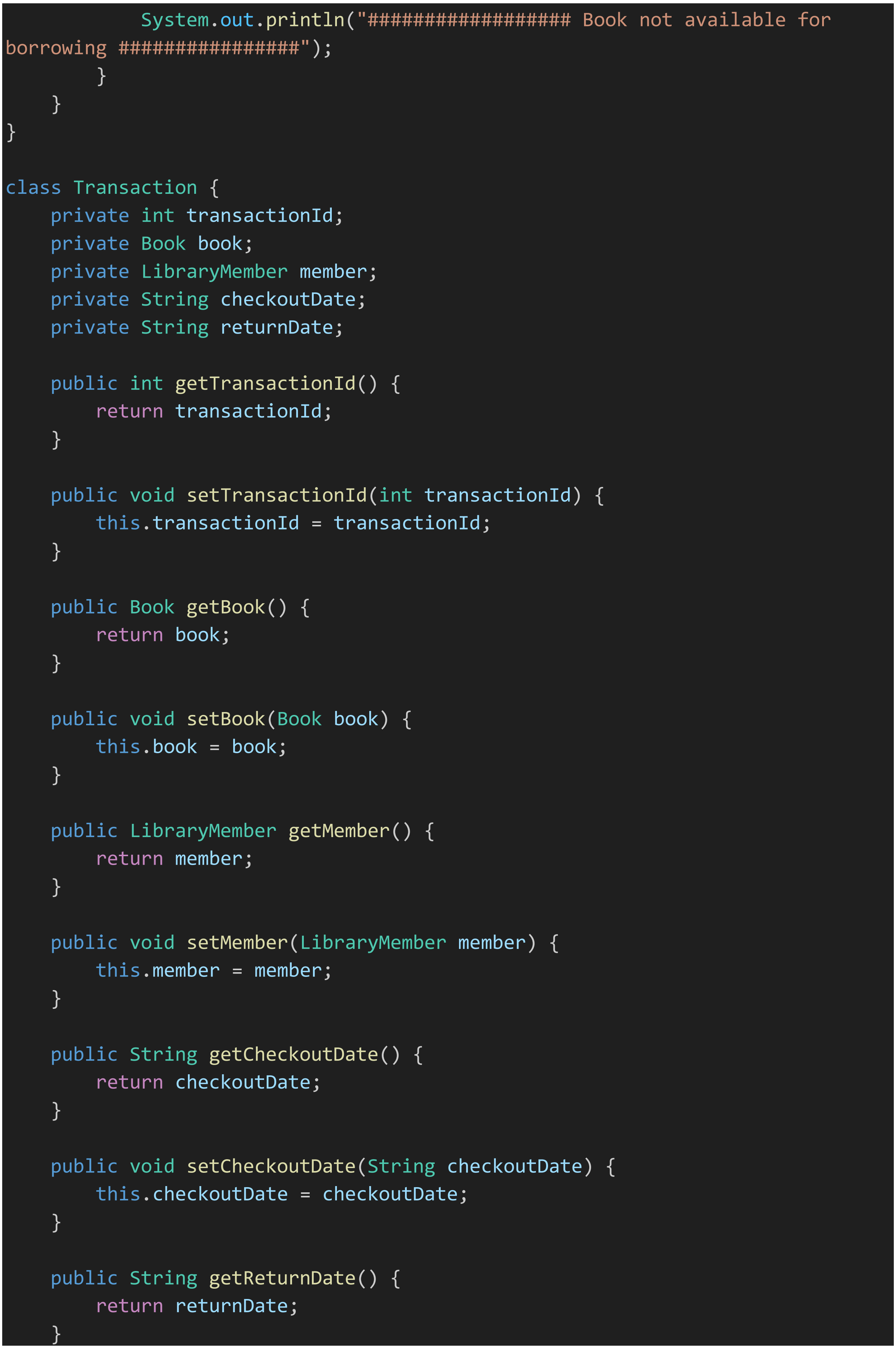
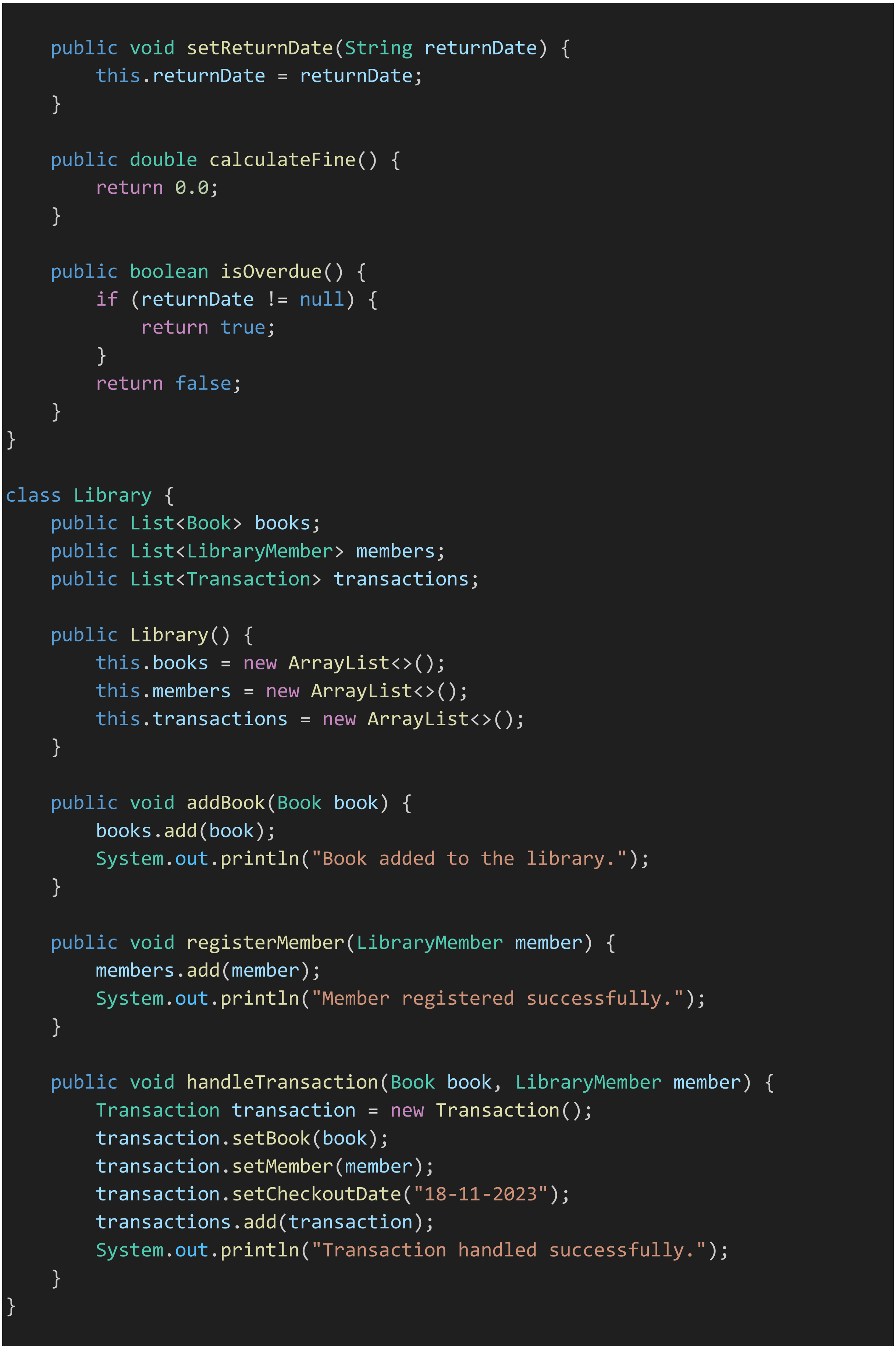
OOP with java

Name: Shashank Banait

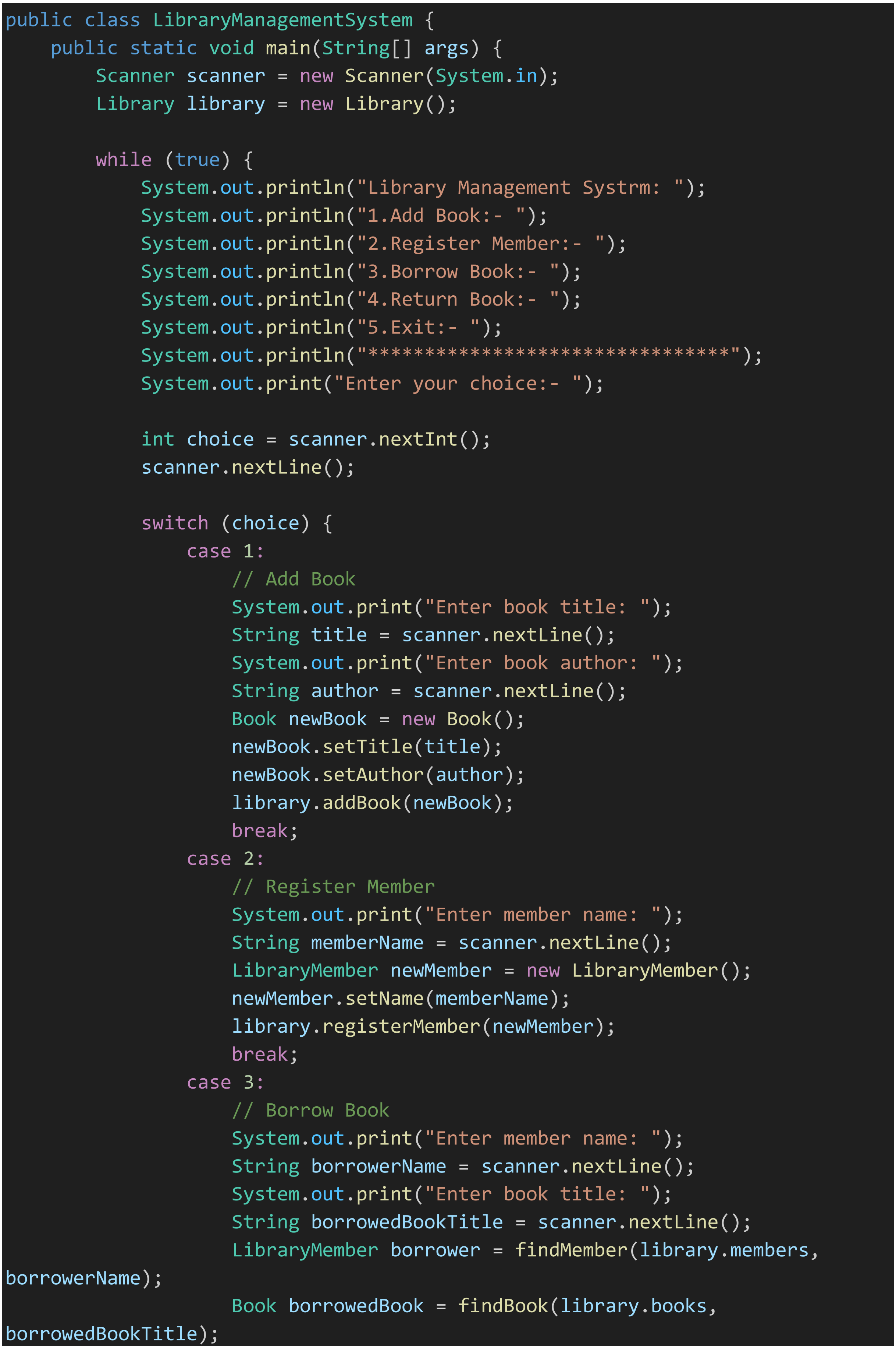
```java
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Book {
    private String title;
    private String author;
    private boolean availability;

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public boolean isAvailability() {
        return availability;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setAvailability(boolean availability) {
        this.availability = availability;
    }

    public void checkOut() {
        if (availability) {
            availability = false;
            System.out.println("################## Book checked out 
successfully ###################33");
        } else {
            System.out.println("################ Book not available for 
checkout ##############3");
        }
    }

    public void returnBook() {
        availability = true;
        System.out.println("################### Book returned successfully
###################");
    }
}

class LibraryMember {
    private int memberId;
    private String name;
    private List<Book> booksBorrowed;

    public int getMemberId() {
        return memberId;
    }

    public void setMemberId(int memberId) {
        this.memberId = memberId;
    }

    public String getName() {
        return name;
    }

    public List<Book> getBooksBorrowed() {
        return booksBorrowed;
    }

    public void setBooksBorrowed(List<Book> booksBorrowed) {
        this.booksBorrowed = booksBorrowed;
    }

    public void returnBook(Book book) {
        booksBorrowed.remove(book);
        book.returnBook();
        System.out.println("Book returned successfully.");
    }

    public void setName(String memberName) {
        this.name = memberName;
    }

    public void borrowBook(Book book) {
        if (book.isAvailability()) {
            booksBorrowed.add(book);
            book.checkOut();
            System.out.println("################## Book borrowed successfully
######################");
        } else {
```



```
            System.out.println("################## Book not available for 
borrowing ################");
        }
    }
}

class Transaction {
    private int transactionId;
    private Book book;
    private LibraryMember member;
    private String checkoutDate;
    private String returnDate;

    public int getTransactionId() {
        return transactionId;
    }

    public void setTransactionId(int transactionId) {
        this.transactionId = transactionId;
    }

    public Book getBook() {
        return book;
    }

    public void setBook(Book book) {
        this.book = book;
    }

    public LibraryMember getMember() {
        return member;
    }

    public void setMember(LibraryMember member) {
        this.member = member;
    }

    public String getCheckoutDate() {
        return checkoutDate;
    }

    public void setCheckoutDate(String checkoutDate) {
        this.checkoutDate = checkoutDate;
    }

    public String getReturnDate() {
        return returnDate;
    }
```



```
    public void setReturnDate(String returnDate) {
        this.returnDate = returnDate;
    }

    public double calculateFine() {
        return 0.0;
    }

    public boolean isOverdue() {
        if (returnDate != null) {
            return true;
        }
        return false;
    }
}

class Library {
    public List<Book> books;
    public List<LibraryMember> members;
    public List<Transaction> transactions;

    public Library() {
        this.books = new ArrayList<>();
        this.members = new ArrayList<>();
        this.transactions = new ArrayList<>();
    }

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

    public void registerMember(LibraryMember member) {
        members.add(member);
        System.out.println("Member registered successfully.");
    }

    public void handleTransaction(Book book, LibraryMember member) {
        Transaction transaction = new Transaction();
        transaction.setBook(book);
        transaction.setMember(member);
        transaction.setCheckoutDate("18-11-2023");
        transactions.add(transaction);
        System.out.println("Transaction handled successfully.");
    }
}
```



```java
public class LibraryManagementSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Library library = new Library();

        while (true) {
            System.out.println("Library Management Systrm: ");
            System.out.println("1.Add Book:- ");
            System.out.println("2.Register Member:- ");
            System.out.println("3.Borrow Book:- ");
            System.out.println("4.Return Book:- ");
            System.out.println("5.Exit:- ");
            System.out.println("********************************");
            System.out.print("Enter your choice:- ");

            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    // Add Book
                    System.out.print("Enter book title: ");
                    String title = scanner.nextLine();
                    System.out.print("Enter book author: ");
                    String author = scanner.nextLine();
                    Book newBook = new Book();
                    newBook.setTitle(title);
                    newBook.setAuthor(author);
                    library.addBook(newBook);
                    break;
                case 2:
                    // Register Member
                    System.out.print("Enter member name: ");
                    String memberName = scanner.nextLine();
                    LibraryMember newMember = new LibraryMember();
                    newMember.setName(memberName);
                    library.registerMember(newMember);
                    break;
                case 3:
                    // Borrow Book
                    System.out.print("Enter member name: ");
                    String borrowerName = scanner.nextLine();
                    System.out.print("Enter book title: ");
                    String borrowedBookTitle = scanner.nextLine();
                    LibraryMember borrower = findMember(library.members,
borrowerName);
                    Book borrowedBook = findBook(library.books,
borrowedBookTitle);
```

```
                    if (borrower != null && borrowedBook != null) {
                        borrower.borrowBook(borrowedBook);
                        library.handleTransaction(borrowedBook, borrower);
                    } else {
                        System.out.println("#################### Member or
book not found #######################");
                    }
                    break;
                case 4:
                    // Return Book
                    System.out.print("Enter member name: ");
                    String returnerName = scanner.nextLine();
                    System.out.print("Enter book title: ");
                    String returnedBookTitle = scanner.nextLine();
                    LibraryMember returner = findMember(library.members,
returnerName);
                    Book returnedBook = findBook(library.books,
returnedBookTitle);
                    if (returner != null && returnedBook != null) {
                        returner.returnBook(returnedBook);
                    } else {
                        System.out.println("########### Member or book not
found #############");
                    }
                    break;
                case 5:
                    // Exit the program
                    System.out.println("############################# Exiting
######################");
                    System.exit(0);
                default:
                    System.out.println("####################### Please enter a
valid option ###################333");
            }
        }
    }

    private static LibraryMember findMember(List<LibraryMember> members,
String name) {
        for (LibraryMember member : members) {
            if (member.getName().equals(name)) {
                return member;
            }
        }
        return null;
    }

    private static Book findBook(List<Book> books, String title) {
        for (Book book : books) {
            if (book.getTitle().equals(title)) {
                return book;
            }
        }
        return null;
    }
}
```