Library Management System

```
package library_management_system;

import java.time.*;
import java.util.List;

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

    public Book(String author, String title, boolean availability, String borrowerId) {
        this.author = author;
        this.title = title;
        this.availability = availability;
        this.borrowerId = borrowerId;
    }

    public String getAuthor() {
        return author;
    }

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

    public String getTitle() {
        return title;
```

```
}

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

public boolean isAvailability() {
   return availability;
}

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

public String getBorrowerId() {
   return borrowerId;
}

public void setBorrowerId(String borrowerId) {
   this.borrowerId = borrowerId;
}

public void checkOut(String memberId){
   if (availability) {
      availability = false;
      borrowerId = memberId;
      System.out.println("Book "+title+" Author "+author+" Checkout by "+memberId);
   }
   else {
      System.out.println("Book "+title+" Author "+author+" is not available");
   }
```

```
    }

    public void returnBook(){
        if (!availability) {
            availability = true;
            borrowerId = null;
            System.out.println("Book "+title+" Author "+author+" returned");
        }
        else {
            System.out.println("Book "+title+" Author "+author+" is Available");
        }
    }

    @Override
    public String toString() {
        return "Books{" + "author=" + author + ", title=" + title + ", availability=" + availability + ",
borrowerId=" + borrowerId + '}';
    }

}

class LibraryMember{
    private String memberId;
    private String name;
    private int booksBorrowed;
    private static final int Max_books = 0;

    public LibraryMember(String memberId, String name, int booksBorrowed) {
        this.memberId = memberId;
        this.name = name;
        this.booksBorrowed = 0;
```

```
}

public String getMemberId() {
   return memberId;
}

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

public String getName() {
   return name;
}

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

public int getBooksBorrowed() {
   return booksBorrowed;
}

public void setBooksBorrowed(int booksBorrowed) {
   this.booksBorrowed = booksBorrowed;
}

public void borrowBook(Book book){
   if (booksBorrowed < Max_books) {
      book.checkOut(memberId);
      booksBorrowed++;
   }
```

```
        else {
            System.out.println("Member "+memberId+" has already borrowed maximum number of book");
        }
    }

    public void returnBook(Book book){
        if (booksBorrowed > 0) {
            book.returnBook();
            booksBorrowed--;
        }
        else {
            System.out.println("Member "+memberId+" has no books to return");
        }
    }

    @Override
    public String toString() {
        return "LibraryMember{" + "memberId=" + memberId + ", name=" + name + ", booksBorrowed=" + booksBorrowed + ", Max_books=" + Max_books + '}';
    }



}

class Transaction{
    private String transactionId;
    private Book book;
    private LibraryMember member;
    private LocalDate checkoutDate;
    private LocalDate returnDate;
    private static final int Max_days = 15;
```

```
    private static final double Fine_rate =0.5;

    public Transaction(String transactionId, Book book, LibraryMember member, LocalDate
checkoutDate, LocalDate returnDate) {
        this.transactionId = transactionId;
        this.book = book;
        this.member = member;
        this.checkoutDate = checkoutDate;
        this.returnDate = returnDate;
    }

    public String getTransactionId() {
        return transactionId;
    }

    public void setTransactionId(String 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 LocalDate getCheckoutDate() {
        return checkoutDate;
    }

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

    public LocalDate getReturnDate() {
        return returnDate;
    }

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

    public void calculateFine(){
        if (returnDate == null) {
            returnDate = LocalDate.now();
        }

        Period p = Period.between(returnDate, checkoutDate);
//      if (p>Max_days) {
//
//      } else {
//      }
    }
```

```
    public void isOverdue(){


    }


    @Override
    public String toString() {
        return "Transaction{" + "transactionId=" + transactionId + ", book=" + book.getTitle()+", By"+
book.getAuthor() + ", member=" + member.getName() + ", checkoutDate=" + checkoutDate + ",
returnDate=" + returnDate + '}';
    }




}


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


    public Library_Management_System(List<Book> books, List<LibraryMember> members,
List<Transaction> transactions) {
        this.books = books;
        this.members = members;
        this.transactions = transactions;
    }


    public List<Book> getBooks() {
        return books;
    }


    public List<LibraryMember> getMembers() {
```

```
        return members;
    }

    public List<Transaction> getTransactions() {
        return transactions;
    }

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

    public void registerMember(LibraryMember member){
        members.add(member);
        System.out.println("Member " + member.getName() + " registered to the library");
    }

    public void handleTransaction(Transaction transaction){
        transactions.add(transaction);
        System.out.println("Transaction " + transaction.getTransactionId() + " handled");
    }
    public static void main(String[] args) {

    }

}
```