```
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;



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

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

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

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

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

    public String getAuthor() {
        return author;
    }

    public boolean isAvailable() {
        return availability;
    }
}

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

    public LibraryMember(int memberId,String name){
        this.memberId = memberId;
        this.name = name;
        this.booksBorrowed = new ArrayList<>();
    }

    public void borrowBook(Book book){
        if (booksBorrowed.size() < 3) {
            booksBorrowed.add(book);
            System.out.println("Book borrowed successfully");
        } else {
            System.out.println("Maximum books borrowed reached");
```

```
        }
    }

    public void returnBook(Book book){
        if (booksBorrowed.remove(book)) {
            System.out.println("Book returned successfully");
        } else {
            System.out.println("You did not borrow this book from the library");
        }
    }

    public int getMemberid(){
        return memberId;
    }

    public String getName(){
        return name;
    }

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

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

```
    public Transaction(int transactionId, Book book, LibraryMember member, Date checkoutDate) {
        this.transactionId = transactionId;
        this.book = book;
        this.member = member;
        this.checkoutDate = checkoutDate;
    }

    public double calculateFine(){
        double fine=0;

        return fine;
    }

    public boolean isOverdue(){
        boolean overdue =false;

        return overdue;
    }
}

class Library {
    private List<Book> books;
    private List<LibraryMember> members;
    private 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, String checkoutDate) {
        if (books.contains(book) && members.contains(member)) {
            Transaction transaction = new Transaction(transactions.size() + 1, book, member,
checkoutDate);
            transactions.add(transaction);
            member.borrowBook(book);
            book.checkout();
            System.out.println("Transaction completed successfully");
        } else {
            System.out.println("Invalid book or member");
        }
    }
}

public class LibraryManagementSystem{


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

        Book book1 = new Book("Book Title 1", "Author 1");
```

```
        Book book2 = new Book("Book Title 2", "Author 2");

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

        LibraryMember member1 = new LibraryMember(1, "Member 1");
        LibraryMember member2 = new LibraryMember(2, "Member 2");

        library.registerMember(member1);
        library.registerMember(member2);



        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        try {
            Date checkoutDate1 = dateFormat.parse("2023-01-01");
            Date checkoutDate2 = dateFormat.parse("2023-02-01");

            library.handleTransaction(book1, member1, checkoutDate1);
            library.handleTransaction(book2, member2, checkoutDate2);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```