

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  class Book {
5      String title;
6      String author;
7      String genre;
8      boolean available;
9
10     // Constructor and other methods for Book class
11     public Book(String title, String author, String genre) {
12         this.title = title;
13         this.author = author;
14         this.genre = genre;
15         this.available = true;
16     }
17 }
18
19 class Patron {
20     String name;
21     String contactInfo;
22     List<Book> borrowingHistory;
23
24     // Constructor and other methods for Patron class
25     public Patron(String name, String contactInfo) {
26         this.name = name;
27         this.contactInfo = contactInfo;
28         this.borrowingHistory = new ArrayList<>();
29     }
```



```
28         this.borrowingHistory = new ArrayList<>();
29     }
30 }
31
32 class Library {
33     List<Book> books;
34     List<Patron> patrons;
35
36     // Other methods for book and patron management
37
38     void borrowBook(Patron patron, Book book) {
39         if (book.available) {
40             book.available = false;
41             patron.borrowingHistory.add(book);
42             System.out.println(patron.name + " borrowed " + book.title);
43         } else {
44             System.out.println("Book not available for borrowing.");
45         }
46     }
47
48     void returnBook(Patron patron, Book book) {
49         if (patron.borrowingHistory.contains(book)) {
50             book.available = true;
51             patron.borrowingHistory.remove(book);
52             System.out.println(patron.name + " returned " + book.title);
53         } else {
54             System.out.println(patron.name + " did not borrow " + book.title);
55         }
56     }
57 }
```



```
55     }  
56 }  
57 }  
58  
59 public class LibraryManagementSystem {  
60     public static void main(String[] args) {  
61         // Example usage of the library management system  
62         Library library = new Library();  
63  
64         Book book1 = new Book("The Great Gatsby", "F. Scott Fitzgerald",  
65                               "Fiction");  
66         Book book2 = new Book("To Kill a Mockingbird", "Harper Lee", "Classic"  
67                               );  
68  
69         Patron patron1 = new Patron("John Doe", "john@example.com");  
70  
71         library.books = new ArrayList<>();  
72         library.patrons = new ArrayList<>();  
73  
74         library.books.add(book1);  
75         library.books.add(book2);  
76         library.patrons.add(patron1);  
77  
78         library.borrowBook(patron1, book1);  
79         library.returnBook(patron1, book1);  
80  
81         // Additional functionality can be added based on requirements  
82     }  
83 }
```


John Doe borrowed The Great Gatsby

John Doe returned The Great Gatsby