

Library 项目设计和说明

1 项目背景

Library 项目是基于 Java 的 Web 应用项目，利用 MVC（Model View Controller，模型 视图 控制器）设计模式。MVC 设计模式如图 1 所示，主要由三部分组成：

- 1) 视图（View），视图是程序的用户界面部分，用户能够看到并能够与之交互的系统应用界面。
- 2) 控制器（Controller），控制器处理用户输入，根据用户输入调用不同的模型，改变模型状态，并根据模型返回的结果跳转不同的视图界面或显示不同的数据给用户。
- 3) 模型（Model），模型是程序的主体部分，实现业务逻辑，是提供系统服务的根本所在。

这一模式的关键特性是实现了显示模块（视图）和功能模块（模型）的分离，从而提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性。

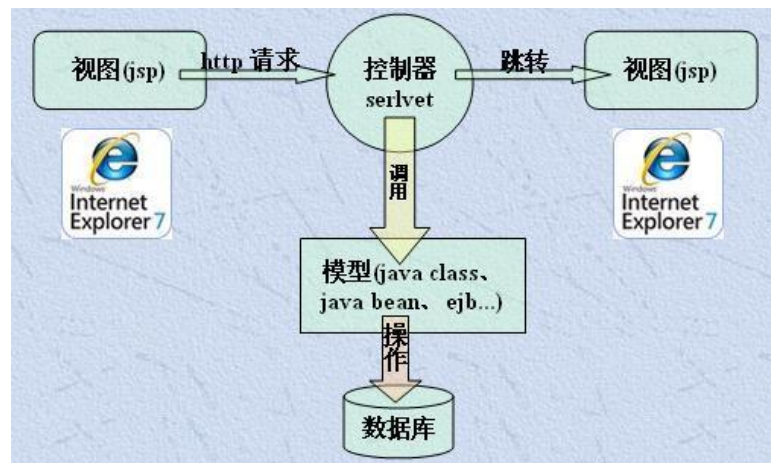


图 1 MVC 模式

Library 使用 JSP（Java Server Pages）实现视图，使用 Java Servlets 实现控制器，使用 Java Beans 实现模型。

图 1 中的 MVC 模式处理流程如下：

- 1) 客户端利用视图 View（用 JSP 实现）输入信息并发送一个 HTTP 请求。
- 2) 控制器 Controller（用 Servlets 实现）接受这个请求，将处理工作转至相应的业务逻辑处理模型（用 Java Bean 实现）。（需要说明的是：我们通常会将工作分工得更细，因此，先有一个总的分发控制器，称为 DispatcherServlet，将

接受用户请求并将处理工作委托给具体的处理控制器 `HandlerServlet`，后者负责将处理请求转至相应的业务逻辑处理模型）。

3) `Service` 层的业务逻辑处理模型完成业务处理后，将返回一个 `Model And View` 对象给 `Servlet`，在这个 `Model And View` 对象中包含了视图逻辑名和实现动态视图需要用到的模型数据对象。（如果细分，这个返回过程是由 `HandlerServlet` 再到 `DispatcherServlet` 的）。

4) `DispatcherServlet` 得到模型视图对象后，将请求的响应分派给某个 `View` 视图，当这个 `View` 呈现给客户端时，客户端就得到了返回的动态 `HTTP` 响应。

在 `Library` 项目中，所有的视图是用 `JSP` 实现的具有扩展名是 `.jsp` 的文件；所有的控制器是用 `Java Servlets` 实现的具有文件名形式为 `*Controller.java` 的文件，`*`代表具体的处理控制器名；所有的模型是用 `Java Bean` 实现的具有扩展名为 `.java` 的文件，并置于包 `library` 下。

用户请求通过 `JSPs` 发送至 `Servlets`，`Servlets` 利用 `Java Beans` 完成对数据库的操作，实现服务，任何模型状态的变化也将通过视图使用的 `Java Bean` 返回给用户。

2 数据库设计

数据库实体联系 `ER` 图如图 2 所示。使用 `PostgreSQL` 数据库管理系统，数据库用户名和密码均是 `webuser`，数据库关系模式如下：

1) 读者表，存储读者信息，主码是读者的社保号 `ssn`：

`member (ssn, driverlicstate, driverlicnum, fname, lname, street, city, state, pcode, phone, email, passwd)`

主码 `pk = ssn`

2) 馆员表，存储馆员信息，主码是馆员的社保号 `ssn`：

`librarian (ssn, fname, lname, street, city, state, pcode, phone, email, passwd)`

主码 `pk = ssn`

3) 书目表，存储图书信息，主码是图书的调用号 `callnumber`：

`booktitle (callnumber, name, author, edition, isbn, year, publisher)`

`pk = callnumber`

4) 图书表，存储读者借阅的图书信息，主码是 `bookid`，外码一是 `callnumber`，引用书目表的 `callnumber`，外码二是 `borrowerssn`，引用读者表的 `ssn`，外码三是 `librarianssn` 引用馆员表的 `ssn`：

`book (bookid, callnumber, borrowerssn, duedate, librarianssn)`

`pk = bookid`

`fk(callnumber) → booktitle(callnumber)`

`fk(borrowerssn) → member(ssn)`

`fk(librarianssn) → librarian(ssn)`

5) 预订表，存储读者想预订的图书信息，主码可以设置成 `ssn`、`callnumber` 组合或 `ssn`、`callnumber`、`holddatetime` 组合，视情况而定，外码一是 `ssn`，引用读者表的 `ssn`，外码二是 `callnumber` 引用书目表的 `callnumber`：

hold (ssn, callnumber, holddatetime)
 fk(ssn) → member(ssn)
 fk(callnumber) → booktitle(callnumber)

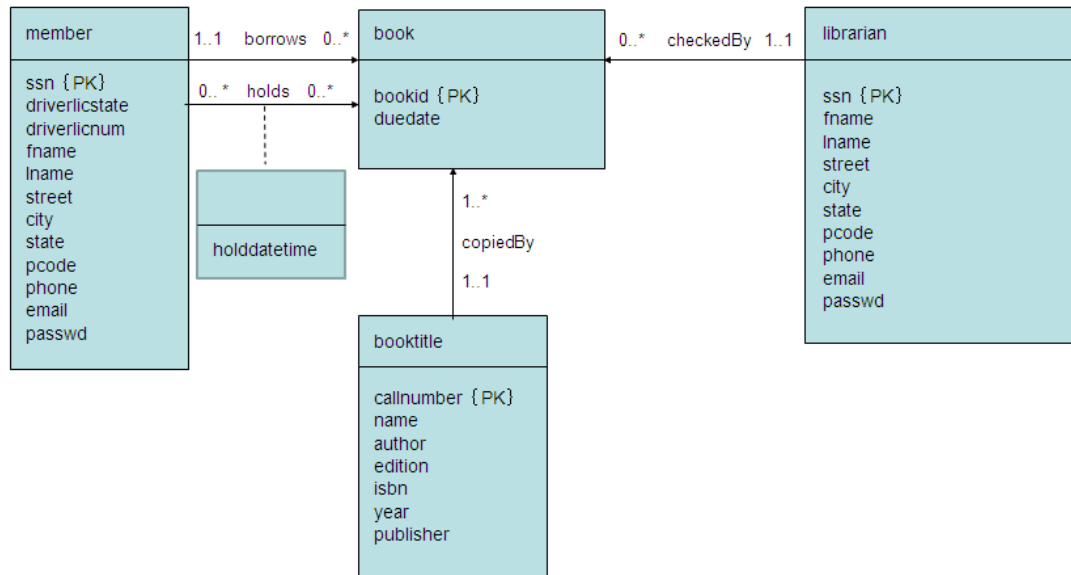


图 2 用户数据库实体联系 ER 图

3 对象和类

Library 系统有以下三种类型的用于实现模型的 Java Bean:

1) DBWrapper, 维护与数据库的连接, 包含访问数据库的方法。这个 Bean 采用单例设计模式, 在任一时刻有唯一的类对象维护到数据库的连接, 可利用 instance()方法获取这个连接, 当这个连接存在时返回, 不存在, 则创建后返回。这个 Bean 的细节和与其他类的关系在图 3 中给出。

2) 第二种类型的 Java Bean 与数据库连接无关, 但与管理对应数据库表的对象有关, 有 Member、Librarian、Book Title、Book、Hold、LibraryUser、Book TitleSet、BookSet 和 HoldSet 共九个 Bean, 前六个 Bean 有 accessor 和 mutator 方法, 使得应用系统可以获取、设置数据库对象的值, LibraryUser 是 Member 和 Librarian 类的父类, 后三个 Bean 用于实现简单数据库对象的集合操作, 这一操作涉及系统多个功能的实现, 这九个 Bean 的细节和与其他类的关系在图 3 中给出。

3) 第三种类型的 Java Bean 用于为前四个第二种类型 Bean 的对象装配数据, 有 LibraryMember、LibraryLibrarian、LibraryBook Title 和 LibraryBook, 每一个 Bean 包含类方法用于实例化对应的第二类 Bean 的对象。例如, LibraryMember 包含类方法 getMember(), 通过输入的读者 ssn 号获得该读者在数据库中保存的完整信息, 这四个 Bean 的细节在图 4 中给出。

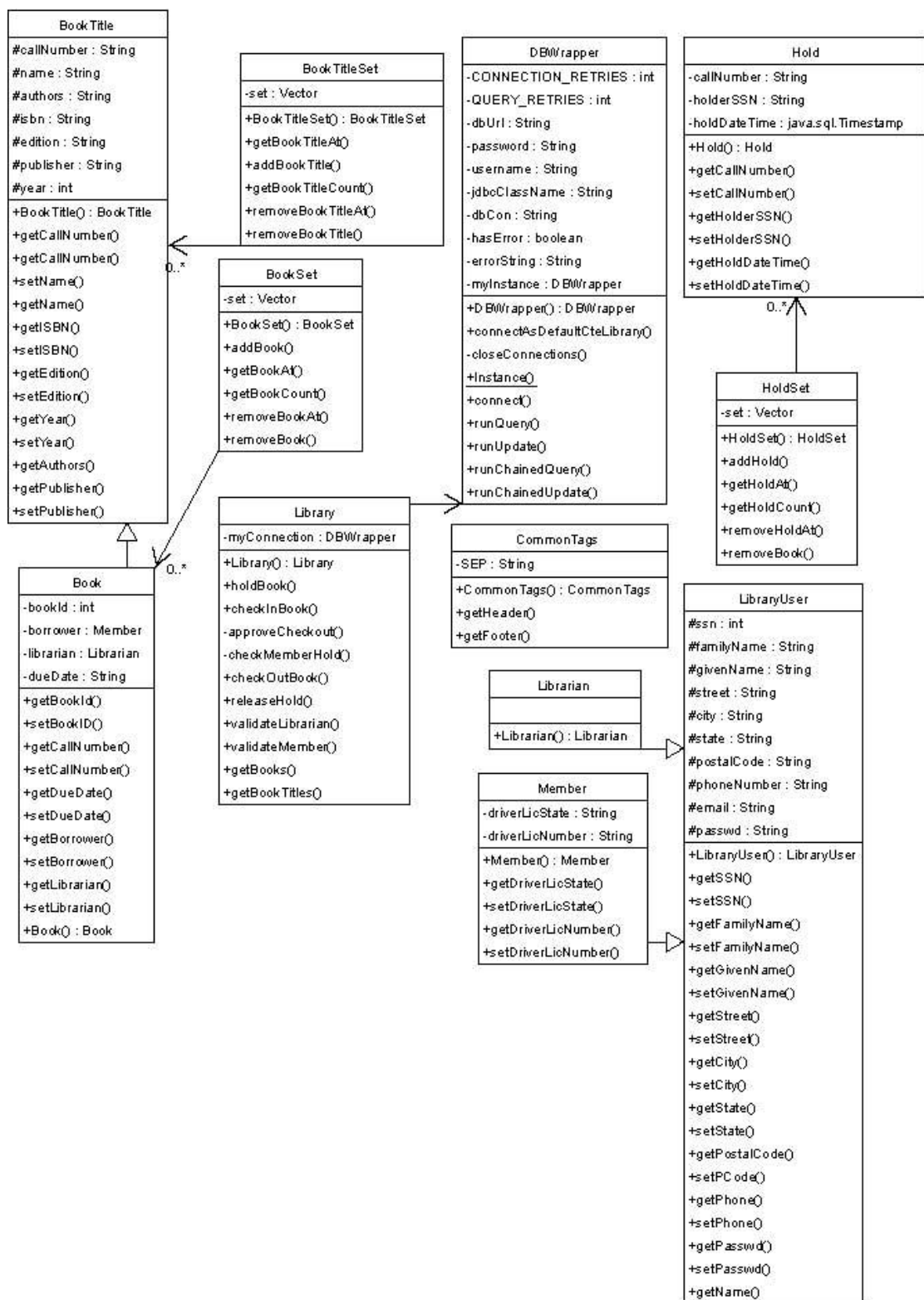


图3 数据库对象管理和两个公共服务类

Library 有以下两个类用于实现一些公共服务功能:

- 1) **CommonTags** 类，用于为应用的每个页面的显示提供共同风格的头部和页脚，这个类的细节在图 3 中给出。

2) **Library** 类，提供多种重要的系统服务功能，诸如，读者和馆员的登录、图书的借出和归还，控制器通过创建 **Library** 类的对象使用这些功能。这个类的细节在图 3 中给出。

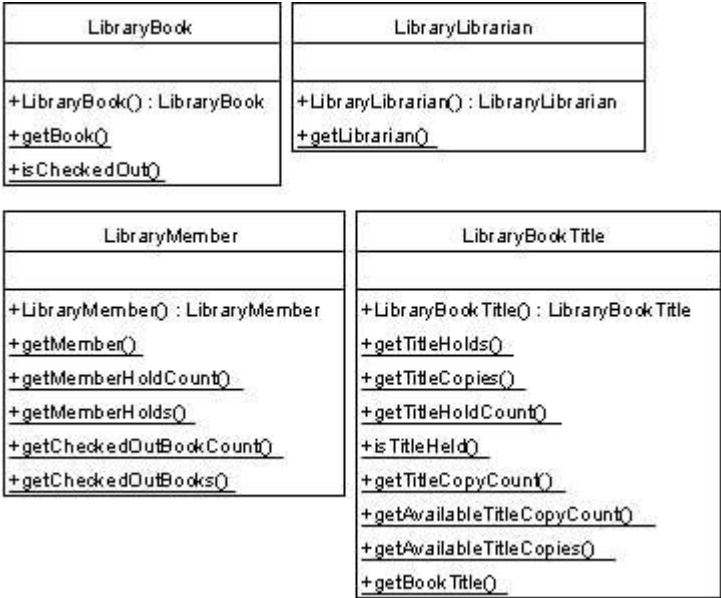


图 4 数据库对象装配类

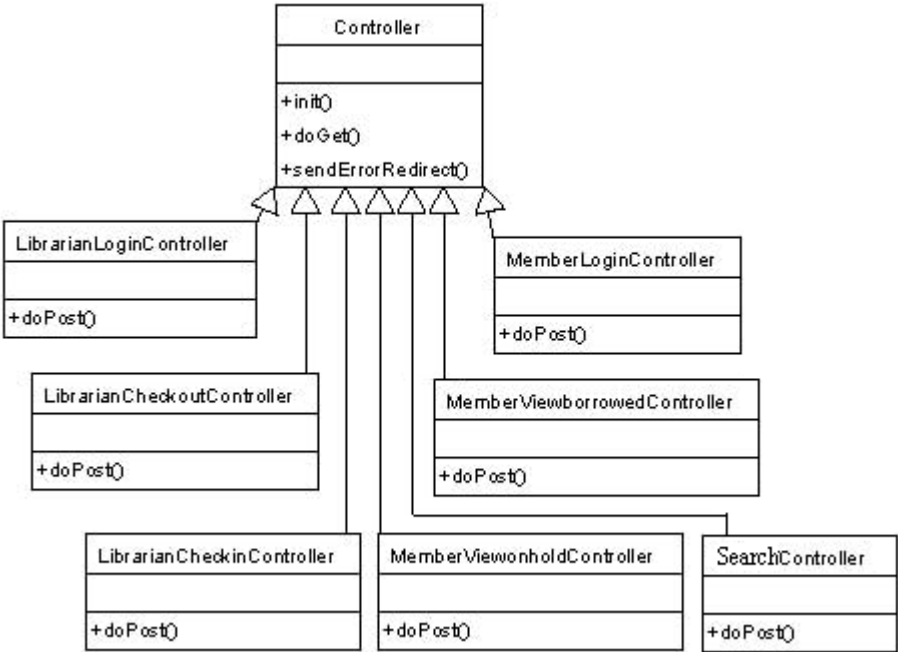


图 5 控制器类

Library 用以下 Servlets 实现用户请求的处理：

1)Servlets 类 Controller 是一个总的分发控制器,通过继承 HttpServlet 创建,包含 doGet()和 sendErrorRedirect()方法,doGet()方法将任何由 Get 表单获取的请求转发给相应的处理控制器,如 MemberLoginController 控制器的 doPost()中,以获得统一形式的实际的处理。Controller 的细节在图 5 中给出。

2) 所有其它的处理控制器,如 MemberLoginController 通过继承 Controller 创建。这些类的细节在图 5 中给出。

4 图书搜索功能

任意的 Library 用户(无须注册登陆)都可以使用系统图书搜索功能,可用完整或部分的作者名、图书名搜索图书。搜索服务流程如图 6 所示,如下所述:

1)在 index.jsp 中通过 form 表单 input 的 name 属性 str 和 searchBy 将用户的图书搜索参数发送至 SearchController。

2)在 SearchController 创建一个动态的 SQL 语句,并将其发送给 DBWrapper 对象,利用 runQuery()方法从数据库获得搜索图书的结果集 ResultSet 返回 SearchController,SearchController 据此结果集并利用 BookTitle 创建 BookTitleSet 对象 Bean。

3) 随后这个 BookTitleSet 对象被加入到 HttpSession 对象中,以备其它 jsp 使用。

4) SearchController 重定向至 search.jsp。

5) search.jsp 通过访问 HttpSession 对象获取 BookTitleSet。

6) search.jsp 显示搜索结果,每页最多 10 本。每本书的标题处有链接至 bookdetails.jsp。bookdetails.jsp 显示每本书的完整细节信息。

关于 **bookdetails.jsp**, bookdetails.jsp 依据获得的图书的 callNumber 通过 LibraryBookTitle 的 getBookTitle()方法得到图书的完整信息,然后利用 BookTitle 的多个 accessor 方法获得图书的细节信息并加以显示。

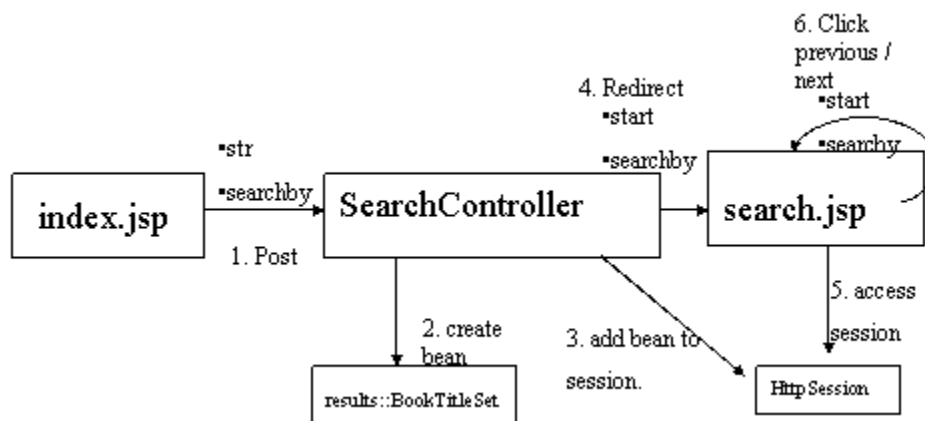


图 6 图书搜索流程图

5 读者和馆员登陆功能

读者和馆员需要登陆才能使用图书的借阅、归还功能，读者登陆的流程如图 7 所示，如下所述：

1) 在 `member.jsp` 中通过 form 表单 input 的 name 属性 `uname` 和 `passwd` 将用户的登陆信息发送至 `MemberLoginController`。

2) 在 `MemberLoginController` 中，首先使当前的 `HttpSession` 无效，让先前登陆的读者登出。然后调用类 `Library` 的 `validateMember()` 方法验证当前读者的登陆信息，返回值是 `Member` 对象，为 `null` 时，`MemberLoginController` 回送一个 `errMsg` 信息至 `member.jsp`，告知用户输入信息有误，否则，创建一个 `Member` 对象 Bean。

3) 随后这个 `Member` 对象被加入到 `HttpSession` 对象中，以备其它 `jsp` 使用。

4) `MemberLoginController` 重定向至 `memberaccount.jsp`。

5) `memberaccount.jsp` 通过访问 `HttpSession` 对象获取 `Member`。

关于 `memberaccount.jsp`，`memberaccount.jsp` 通过从 `HttpSession` 对象中获取的 `Member` 对象，利用 `MemberViewborrowedController` 和 `MemberViewonholdController` 可查看读者借阅的图书和预订的图书。流程分别如图 8、9 所示。

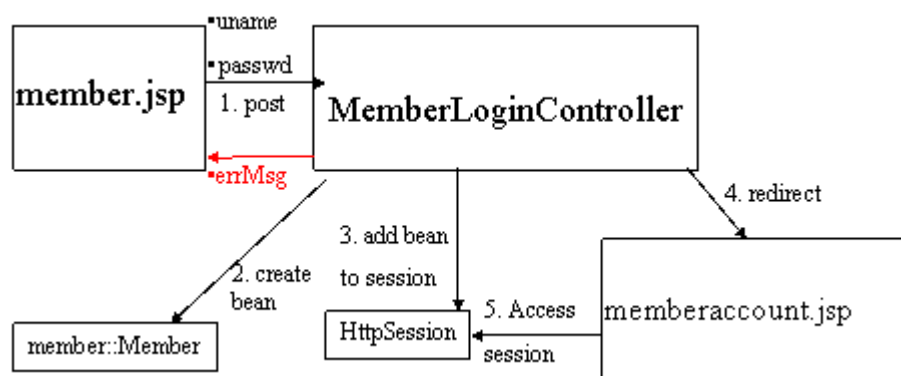


图 7 读者登陆流程图

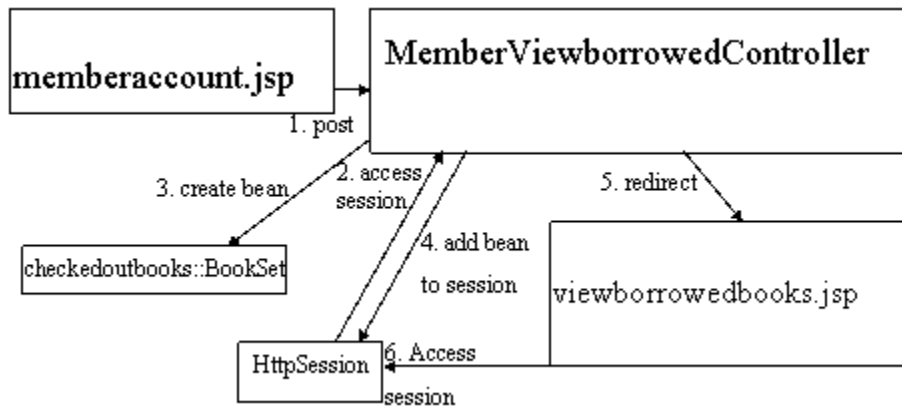


图 8 查看登陆读者借阅的图书

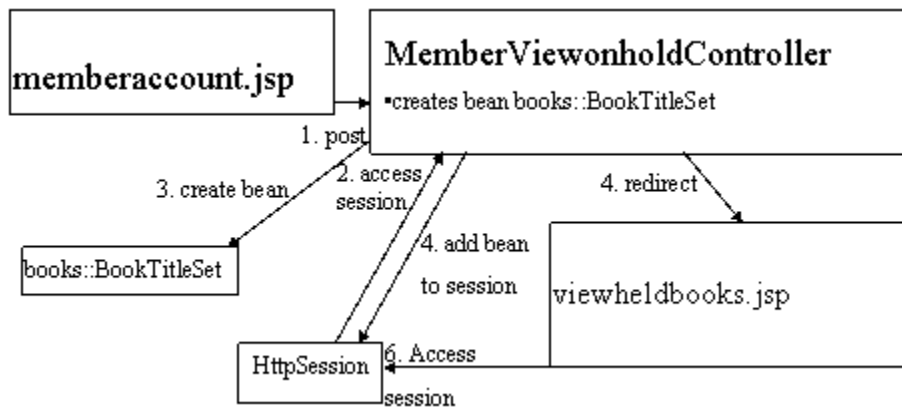


图 9 查看登陆读者预订的图书

馆员登陆过程与读者登陆类似。利用 `librarian.jsp` 获取馆员信息并发送至 `LibrarianLoginController`，使用 `Library` 对象的 `validateLibrarian()` 方法验证馆员信息，如果验证成功，创建 `Librarian` 对象并将其添加至 `HttpSession` 对象中，以备其它 `jsp` 使用，否则，回送一个 `errMsg` 信息至 `librarian.jsp`。

6 图书借出功能

只有登陆的馆员才能使用图书借出功能，借阅流程如图 10 所示，如下所述：

1) 在 `librarianaccess.jsp` 中，通过 `form` 表单 `input` 的 `name` 属性 `bookid` 和 `ssn` 将预备借出的图书的 `bookid` 和读者的 `ssn` 发送至 `LibrarianCheckoutController`。

2) 在 `LibrarianCheckoutController` 中，首先使用 `HttpSession` 对象获得当前馆员对象，然后调用类 `LibraryBook` 的 `getBook()` 获得当前 `Book` 对象，调用类 `LibraryMember` 的 `getMember()` 获得当前 `Member` 对象。

3) 如果当前 `Book` 对象或 `Member` 对象有效：

如果借出成功，`LibrarianCheckoutController` 重定向至 `entrycomplete.jsp`；

如果借出失败, LibrarianCheckoutController 回送 chenkoutbook 信息至 librarianaccess.jsp, 告知用户想要借出的图书已被借出。

如果当前 Book 对象为 null 或 Member 对象为 null:

LibrarianCheckoutController 回送 chenkoutbook 或 checkoutmember 信息并重定向至 librarianaccess.jsp, 告知用户借阅的图书不存在或读者不存在。

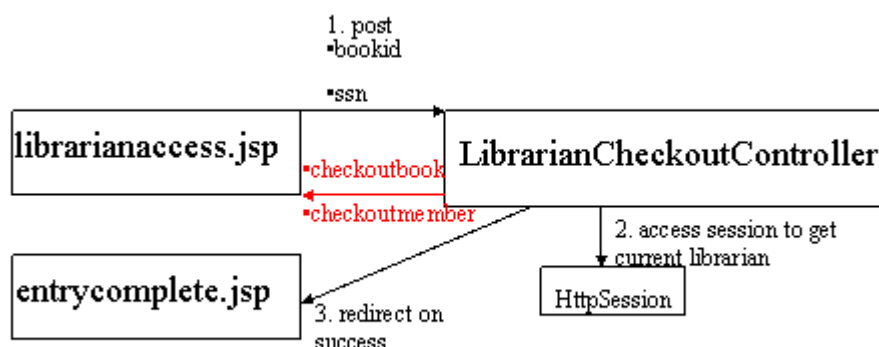


图 10 图书借阅流程图

7 图书归还功能

只有登陆的馆员才能使用图书归还功能, 归还流程如图 11 所示, 如下所述:

1) 在 librarianaccess.jsp 中, 通过 form 表单 input 的 name 属性 bookid 将预备归还的图书的 bookid 发送至 LibrarianCheckinController。

2) 在 LibrarianCheckinController 中, 首先使用 HttpSession 对象获得当前馆员对象, 然后调用类 LibraryBook 的 getBook()获得当前 Book 对象。

3) 如果当前 Book 有效:

如果归还成功, LibrarianCheckinController 重定向至 entrycomplete.jsp;

如归归还失败, LibrarianCheckoutController 重定向至 librarianaccess.jsp。

如果当前 Book 对象为 null:

LibrarianCheckoutController 回送 chenkinbook 信息并重定向至 librarianaccess.jsp, 告知用户归还的图书不存在。

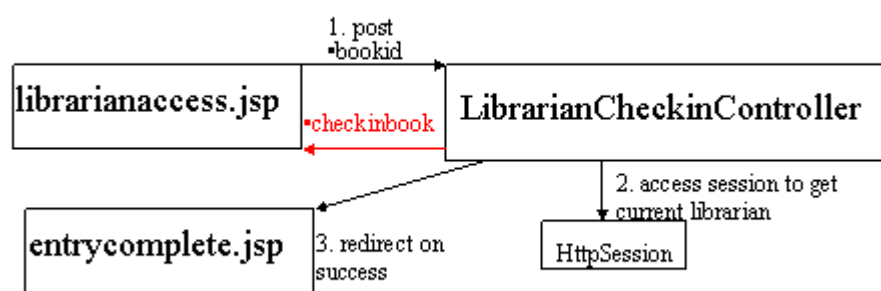


图 11 图书归还流程图