实验 15、16 Spring 综合编程

一、实验目的

- 1、理解 Spring 容器的概念。
- 2、掌握 Struts 2、Hibernate 和 Spring 整合方法。

二、实验内容

综合运用 Struts 2、Hibernate 和 Spring 三大框架技术,设计并实现一个简单的图书应用系统,实现管理员对图书的管理功能和用户对图书的查询借还功能。

三、实验过程及要求

1、JavaBean 设计及 ORM 映射文件

如下图所示,设计各个实体类,并编写其与数据库表的 ORM 映射文件。

- - Author.java
 - ▶ Book.java
 - ▶ J Lend.java
 - - Author.hbm.xml
 - Book.hbm.xml
 - Lend.hbm.xml
 - Warang Paragraph Paragr

(1) 用户类 User 及 ORM 映射文件。

```
public class User {
  private String name;
  private String password;
  private String role;
  public User(String name, String password, String role) {
      this.name = name;
      this.password = password;
      this.role = role;
  }
  public User() {
 //setter 和 getter 方法
<hibernate-mapping>
  <class name="model.User" table="usr" schema="app">
      <id name="name" type="string">
          <column name="name" />
          <generator class="assigned" />
      </id>
```

```
cproperty name="password" type="java.lang.String">
             <column name="password" />
          </property>
          cproperty name="role" type="java.lang.String">
             <column name="role" />
          </property>
      </class>
   </hibernate-mapping>
(2) 作者类 Author 及 ORM 映射文件。
   public class Author {
      private String name;
      private String tel;
      private String email;
     //两个构造方法
    //setter 和 getter 方法
   <hibernate-mapping>
      <class name="model.Author" table="author" schema="app"
   lazy="false">
          <id name="name" type="string" column="name">
             <generator class="assigned" />
          </id>
          cproperty name="tel" type="string" column="tel" />
          cproperty name="email" type="string" column="email" />
      </class>
   </hibernate-mapping>
(3) 书籍类 Book 及 ORM 映射文件。
   public class Book {
      private String isbn;
      private String title;
      private int price;
      private String intro;
      private Author author;
     //两个构造方法
    //setter和 getter 方法
   <hibernate-mapping>
      <class name="model.Book" schema="app">
          <id name="isbn" type="java.lang.String" column="isbn">
             <generator class="assigned"></generator>
          </id>
          cproperty name="title" type="string" column="title" />
          roperty name="price" type="int" column="price" />
          column="intro" type="string" column="intro" />
```

```
cascade="all"
          <many-to-one
                               name="author"
    class="model.Author" column="name" />
       </class>
    </hibernate-mapping>
 (4) 借阅信息类 Lend 及 ORM 映射文件。
    public class Lend {
       private int bookld;
      private String name;
      private String isbn;
      private Date Itime;
      //两个构造方法
     //setter和 getter 方法
    <hibernate-mapping>
       <class name="model.Lend" table="lend" schema="app">
          <id name="bookId" type="int">
              <column name="bookid" />
              <generator class="identity" />
          </id>
          cproperty name="name" type="java.lang.String">
              <column name="name" />
          cproperty name="isbn" type="java.lang.String">
              <column name="isbn" />
          </property>
          cproperty name="Itime" type="java.util.Date">
              <column name="ltime" />
          </property>
       </class>
    </hibernate-mapping>
2、数据库表设计
(1) 用户表 usr。其中,角色 role 只能取"读者"和"管理员"。
    create table app.usr(
       name varchar(20),
      password varchar(6) not null,
      role varchar(10) not null default '读者',
      primary key(name)
    ):
    insert into app.usr values('李明', '123456', '管理员');
                       NAME PASSWORD ROLE
                       李明 123456
                                      管理员
                       Rose rose
                                      读者
                             123456
                                      读者
                       Tom
                       admin admin
                                      管理员
```

(2) 作者表 author。

```
create table app.author(
  name varchar(10),
  tel varchar(20),
  email varchar(50),
  primary key(name)
);
```

NAME	TEL	EMAIL
谢希仁	7788	xie@qq.com
王珊	5566	wang@qq.com
赵刚	8877	zg@qq.com
李平	6655	zhao@qq.com
周伟	5588	zw@qq.com

(3) 书籍表 book。

```
create table app.book(
   isbn varchar(10),
   title varchar(50) not null,
   price smallint not null,
   intro varchar(20) not null,
   name varchar(10),
   primary key(isbn)
);
```

ISBN TITLE PRICE INTRO NAME 9001 计算机网络 hello.txt 谢希仁 58 hello.txt 王珊 9002 数据库原理 37 9003 大数据 40 hello.txt 李平 9004 禽散数学 hello.txt 赵刚 36 hello.txt 周伟 9008 计算机组成原理 52

(4) 借阅表 lend。其中,bookid 自增长,ltime 为借阅时间。

```
create table app.lend(
```

```
bookid integer generated always as identity (start with 1, increment by 1) primary key,
   name varchar(20),
   isbn varchar(10),
   Itime date
);
```

BOOKID	NAME	ISBN	LTIME
25	Rose	9004	2018-11-10
27	Rose	9001	2018-11-10

3、JSP页面设计

(1) 用户登录页面 login.jsp

姓名:	李明
密码:	•••••
	登录
	重置

(2) 图书应用系统主页 main.jsp

如下图所示,根据不同的角色,显示不同的功能界面。

图书应用系统

当前用户:李明(管理员) [注销]	图书应用系统
	当前用户: Rose(读者) <u>[注销</u>]
2. 管理用户 3. 添加书籍 4. 管理书籍	1. 已借书籍 2. 借阅书籍

(3) 添加用户页面 addUser.jsp(管理员)

添加用户	添加用户
14-74117 147	姓名: 刘正
姓名: 刘正	密码: 123456
密码: 123456	角色: 读者 ▼
角色: 读者 ▼	保存
管理员 读者 保存	成功保存用户: 刘正(读者)
返回主页	返回主页

(4) 管理用户页面 listUser.jsp(管理员)

此处,仅能修改用户密码。

用户管理

搜索条件:		查询		
用户姓名	用户密码	用户角色	拼	操作
李明	123456	管理员	<u>修改</u>	删除
Rose	rose	读者	<u>修改</u>	删除
Tom	123456	读者	<u>修改</u>	删除
admin	admin	管理员	<u>修改</u>	删除
刘正	123456	读者	<u>修改</u>	删除

查询全部 返回主页

(5) 修改用户页面 updateUser.jsp(管理员)

用户姓名和角色为只读(readonly)。

修改用户信息

修改用户信息

用户姓名:	Rose
用户密码:	123456
用户角色:	读者
	修改

用户姓名: Rose 用户密码: 123456 用户角色: 读者

更新用户成功!

用户管理 返回主页

用户管理 返回主页

用户管理

搜索条件:		查询		
用户姓名	用户密码	用户角色	护	操作
李明	123456	管理员	<u>修改</u>	删除
Rose	123456	读者	<u>修改</u>	删除
Tom	123456	读者	<u>修改</u>	删除
admin	admin	管理员	<u>修改</u>	删除
刘正	123456	读者	修改	船線

查询全部 返回主页

删除用户 Tom,界面如下。

用户管理

搜索条件: [宣训		
用户姓名	用户密码	用户角色	护	操作
李明	123456	管理员	<u>修改</u>	删除
Rose	123456	读者	<u>修改</u>	删除
admin	admin	管理员	<u>修改</u>	删除
刘正	123456	读者	<u>修改</u>	删除

查询全部 返回主页

(6) 添加书籍页面 addBook.jsp(管理员)

添加书籍

图书编号:	
图书名称:	
图书价格:	
图书简介:	浏览
作者姓名:	
作者电话:	
作者邮件:	
	保存

返回主页

(7) 管理书籍页面 listBook.jsp(管理员)

此处,仅能修改用户密码。

管理书籍

搜索条件:		查询				
图书编号	图书名称	图书价格	作者姓名	作者电话	作者邮件	操作
9001	计算机网络	58	谢希仁	7788	xie@qq.com	简介 修改 删除
9002	数据库原理	37	王珊	5566	wang@qq.com	简介 修改 删除
9003	大数据	40	李平	6655	zhao@qq.com	简介 修改 删除
9004	离散数学	36	赵刚	8877	zg@qq.com	简介 修改 删除
9008	计算机组成原理	52	周伟	5588	zw@qq.com	简介 修改 删除

查询全部 返回主页

(8) 修改书籍页面 updateBook.jsp(管理员)

图书编号为只读(readonly)。

修改书籍

图书编号:	9008	
图书名称:	计算机组成原理	
图书价格:	52	
图书简介:		浏览
作者姓名:	周伟	
作者电话:	5588	
作者邮件:	zw@qq.com	
		修改

图书管理 返回主页

(9) 已借书籍页面 lendedBook.jsp(读者)

此处,用户Rose登录,显示其所借的图书。点击"还书",实现还书功能。

Rose已借阅的书籍

图书编号	图书名称	图书价格	作者姓名	操作
9001	计算机网络	58	谢希仁	<u>还书</u>
9004	离散数学	36	赵刚	<u>还书</u>

返回主页

Rose已借阅的书籍

 图书编号
 图书名称
 图书价格
 作者姓名
 操作

 9004
 离散数学
 36
 赵刚
 还书

还书成功: 计算机网络

返回主页

(10) 借阅书籍页面 lendBook.jsp (读者)

此处,用户 Rose 登录,显示其未借的所有图书。点击"借书",实现借书功能。 借阅书籍

搜索条件:		查询		
图书编号 9001 9002 9003	图书名称 计算机网络 数据库原理 大数据	图书价格 58 37 40	作者姓名 谢希仁 王珊 李平	操作 简介 借书 简介 借书 简介 借书
9008 查询全部	计算机组成原理 返回主页	52	周伟	简介 借书

借阅书籍

搜索条件:		查询		
图书编号 9001 9003 9008	图书名称 计算机网络 大数据 计算机组成原理	图书价格 58 40 52	作者姓名 谢希仁 李平 周伟	操作 简介 借书 简介 借书 简介 借书
借书成功:	数据库原理			

查询全部_ 返回主页

4、Action 类设计

(1) UserAction 类。

```
public class UserAction extends ActionSupport {
    private String message;
    private User user;
    private ArrayList<User> users;
    private String condition = "";
    private IUserDAO userDAO;
```

(2) BookAction 类。

```
public class BookAction extends ActionSupport {
    private String message;
    private Book book;
    private ArrayList<Book> books;
    private String condition = "";
    private File bookIntro;
    private String bookIntroFileName;
    private String bookIntroContentType;
    private String bookIntroContent = "";
    private BookDAO bookDAO;
```

(3) LendAction 类。

```
public class LendAction extends ActionSupport {
    private String message;
    private Book book;
    private User user;
    private ArrayList<Book> books;
    private String condition = "";
    private String bookIntroContent = "";
    private LendDAO lendDAO;
```

5、Spring 的 bean 定义

```
在目录 src 下,创建 XML 文件 applicationContext.xml。
```

6、拦截器设计

同实验14。

7、数据库连接

编写 Hibernate 配置文件 hibernate.cfg.xml。

```
<hibernate-configuration>
    <session-factory>
        cproperty name="dialect">org.hibernate.dialect.DerbyD
        property name="connection.driver class">org.apache.d
        cproperty name="connection.url">jdbc:derby://localhos
        property name="connection.username">admin/property>
        property name="connection.password">admin/property>
        cproperty name="show_sql">true
        <mapping resource="model/Book.hbm.xml" />
        <mapping resource="model/User.hbm.xml" />
        <mapping resource="model/Author.hbm.xml" />
        <mapping resource="model/Lend.hbm.xml" />
    </session-factory>
</hibernate-configuration>
8、web.xml 文件
在 web.xml 文件中增加以下代码,实现 Spring 容器。
tener>
    <listener-class>
        org.springframework.web.context.ContextLoaderListener
    </listener-class>
</listener>
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/classes/applicationContext.xml
    </context-param>
9、struts.xml 文件
示例:还书功能。
 <action name="lendedBook" class="lendAction" method="listLendedBook">
    <result name="success">/lendedBook.jsp</result>
 <action name="returnBook" class="lendAction" method="returnBook">
    <result name="success" type="redirectAction">
        <param name="actionName">lendedBook</param>
    </result>
    <result name="error" type="redirectAction">
        <param name="actionName">lendedBook</param>
    </result>
</action>
```

- 10、在 Tomcat 服务器上部署、在浏览器中测试 Web 应用。
- 11、完成实验内容后,写出实验过程、结论和心得。