|  |  |
| --- | --- |
| 成绩 |  |



**课程设计报告**



**题 目** 图书管理系统需求文档

**课 程 名 称** 暑期Java实训

**院 部 名 称**  软件工程学院

**专 业** 软件工程

**班 级** 15软件工程2班

**学 生 姓 名** 吴跟强

**学 号** 1512001066

**课程设计地点**  A203

**课程设计学时** 120

**指 导 教 师** 张明星

金陵科技学院教务处制

图书管理系统需求文档

目录

[一、语言和环境 3](#_Toc488154117)

[二、系统需求分析 3](#_Toc488154118)

[1、功能需求： 3](#_Toc488154119)

[2、性能需求： 3](#_Toc488154120)

[三、数据库设计 3](#_Toc488154121)

[1、数据库概要设计 3](#_Toc488154122)

[1.1数据流图 3](#_Toc488154123)

[1.2实体关系E-R图 4](#_Toc488154124)

[1.3功能模块图 5](#_Toc488154125)

[2、数据库系统详细设计 5](#_Toc488154126)

[2.1数据库逻辑设计 5](#_Toc488154127)

[四、系统实现 6](#_Toc488154128)

[1、框架实现：采用SpringMVC+Spring+MyBtis方法 6](#_Toc488154129)

[1.1用户登录功能。 6](#_Toc488154130)

[1.2用户注册功能。 8](#_Toc488154131)

[1.3图书信息列表显示及图书增删改查更新排序功能。 9](#_Toc488154132)

[1.4实现翻页功能，界面展示。 12](#_Toc488154133)

[2、具体实现系统功能代码展示及步骤说明。 12](#_Toc488154134)

[2.1前端部分代码实现。 12](#_Toc488154135)

[2.2后端部分代码实现。 20](#_Toc488154136)

[2.3 Spring部分代码实现。 31](#_Toc488154137)

[五、图书管理系统设计实现步骤最终说明 42](#_Toc488154138)

[六、图书管理系统实现注意事项 42](#_Toc488154139)

图书管理系统需求文档

# 一、语言和环境

1. 实现语言：Java
2. 环境要求：JDK、Eclipse、Tomcat、MySql

# 二、系统需求分析

## 1、功能需求：

1.1功能描述：

（1）book表：反映图书管理系统里所有的书籍信息；

（2）user表：系统用户和管理员的基本信息及登录密码。

（3）注册系统用户信息

（4）登录：登录时提示普通用户登录or系统管理员登录

（5）管理员实现对图书增删改查更新，普通用户实现查看搜索功能

（6）图书信息列表显示，并实现翻页功能（进入图书信息列表界面，默认显示首页，默认按图书ID升序列出所有的图书）

（7）增加图书信息排序功能，实现管理员权限保护功能。

## 2、性能需求：

图书管理系统用户在使用本系统的过程中能自行根据实际情况进行精度设定，在时间特性上，当普通用户/管理员发出命令请求时的服务器的响应时间、对数据更新处理、查询检索等功能上，实现响应时间不会超过0.5秒时间。

# 三、数据库设计

## 1、数据库概要设计

### 1.1数据流图

根据图书管理系统要求及具体的情况，得到以下数据流图。

注：系统普通用户只能进行查看搜索功能

系统管理员可以进行增删改查更新搜索功能

修改图书

删除图书

添加图书

book表

管理员

图1.1管理员第一层数据流图

### 1.2实体关系E-R图

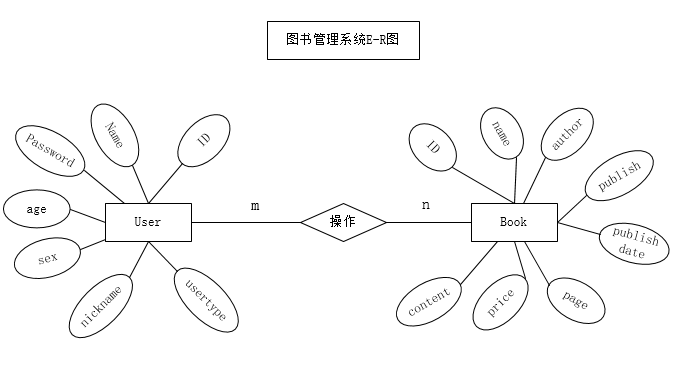


图1.2实体关系E-R图

### 1.3功能模块图

图书管理系统

系统模块

图书管理

用户登陆

用户注册

## 2、数据库系统详细设计

### 2.1数据库逻辑设计

2.1.1数据库关系模式：

根据实体关系图（E-R）图，建立数据库关系模式

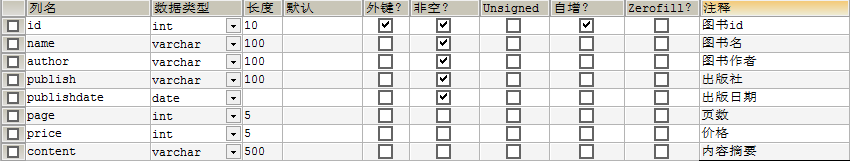
book（id，name,author,publish,publishdate,page,price,content）

user（id,name,password,age,sex,nickname,usertype）

注：标有下划线的 表示为该数据表的主码，即主关键字。

2.1.2数据库表

表一：图书信息表（book）：



表二：用户信息表（user）：



# 四、系统实现

## 1、框架实现：采用SpringMVC+Spring+MyBtis方法

### 1.1用户登录功能。

打开图书管理系统，默认进入用户登录界面，输入用户名和密码，会提示相应信息，点击登录：（1）登录成功后，系统会弹框提示系统普通用户登录还是管理员登录；（2）登录失败后，会弹框提示用户名或者密码错误信息。

1.1.1用户登录界面：



1.1.2管理员登录提示界面



1.1.3普通用户登录提示界面



1.1.3用户名或密码输入错误提示



### 1.2用户注册功能。

点击登录界面注册按钮，进行用户的注册





### 1.3图书信息列表显示及图书增删改查更新排序功能。

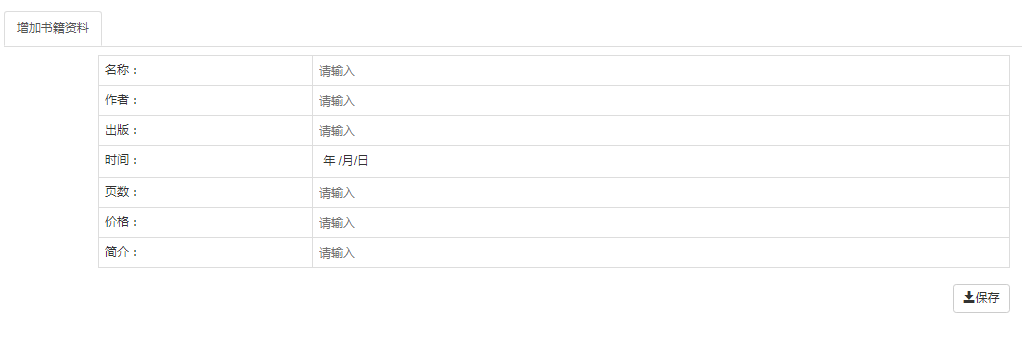
用户登录成功后，进入图书信息列表显示首页。默认按图书ID升序显示列出所有的图书。

#### 1.3.1管理员登录显示图书信息界面：



#### 1.3.2管理员具有图书增删改查更新排序功能

1.3.2.1新增书籍界面：



1.3.2.2删除书籍界面：

勾选书籍，点击删除按钮，删除所选书籍。



1.3.2.3修改书籍界面：



1.3.2.4查找书籍界面：



1.3.2.5书籍排序界面：



#### 1.3.3普通用户登录显示图书信息界面：

已经默认将增删功能隐藏



#### 1.3.4普通用户实现查看查找排序功能

1.3.4.1查找书籍界面：



1.3.4.2书籍排序界面：



### 1.4实现翻页功能，界面展示。

点击右边1，2即可实现跳转页面显示：





## 2、具体实现系统功能代码展示及步骤说明。

### 2.1前端部分代码实现。

此部分使用了html超文本标签语言，jsp等等。

2.1.1注册界面实现前端展示代码：

User\_register.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<title>用户注册</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"../css/bootstrap.min.css"*/>

<style type=*"text/css"*>

*.space*{

height: *10px*;

}

**table** **input**{

border: *0*;

width: *100%*;

height: *100%*;

}

</style>

</head>

<body>

<div class=*"space"*>

</div>

<div class=*"container"*>

<form action=*"/LibraryManageSystem/User\_register.do"* method=*"post"*>

<table id=*"insertable"* class=*"table table-bordered"*>

<tr>

<td>姓名：</td>

<td><input type=*"text"* name=*"name"* placeholder=*"请输入"* /></td>

</tr>

<tr>

<td>密码：</td>

<td><input type=*"text"* name=*"password"* placeholder=*"请输入"* /></td>

</tr>

<tr>

<td>年龄：</td>

<td><input type=*"number"* name=*"age"* placeholder=*"请输入"* /></td>

</tr>

<tr>

<td>性别：</td>

<td><input type=*"number"* name=*"sex"* placeholder=*"请输入"* /></td>

</tr>

<tr>

<td>昵称：</td>

<td><input type=*"text"* name=*"nickname"* placeholder=*"请输入"* /></td>

</tr>

<tr>

<td>用户类型：</td>

<td><input type=*"number"* name=*"usertype"* placeholder=*"请输入【1：管理员，0：用户】"* /></td>

</tr>

</table>

<button type=*"submit"* class=*"btn btn-default pull-right"*>

<span class=*"glyphicon glyphicon-save"* aria-hidden=*"true"*></span>保存

</button>

</form>

</div>

</body>

</html>

2.1.2用户登录部分：

User\_login.jsp

<%@page contentType=*"text/html"*%>

<%@page pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort() + path + "/";

%>

<!DOCTYPE html>

<html>

<!--将css样式引入-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/denglu.css"* />

<!--头部-->

<head>

<!--以防出现乱码-->

<!--标题-->

<title>图书管理系统</title>

</head>

<!--主体-->

<body>

<div class=*"content"*>

<!--内容-->

<h1 class=*"content-title"*>图书管理系统</h1>

<div class=*"content-form"*>

<form method=*"post"* action=*"/LibraryManageSystem/User\_login.do"* id=*"form"* >

<div id=*"change\_margin\_1"*>

<input class=*"user"* type=*"text"* name=*"name"* placeholder=*"请输入用户名"* onblur="oBlur\_1()" onfocus="oFocus\_1()">

</div>

<!--input的value为空时弹出提醒-->

<div class=*"text-item"*>

<p id=*"remind\_1"*></p>

</div>

<p id=*"change\_margin\_2"*>

<input class=*"password"* type=*"password"* name=*"password"* placeholder=*"请输入密码"*onblur="oBlur\_2()" onfocus="oFocus\_2()">

</p>

<div class=*"text-item"*>

<p id=*"remind\_2"*></p>

</div>

<div id=*"change\_margin\_3"*>

<input class=*"content-form-signup"* type=*"submit"* value=*"登录"* >

</div>

</form>

</div>

<div class=*"content-login-description"*>没有账户</div><br/>

<div>

<a class=*"content-login-link"* href=*"html/User\_register.html"*>注册</a>

</div>

</div>

</body>

<script src=*"js/User\_login.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

</html>

2.1.3图书管理系统主界面部分：

indexjsp

<%@page contentType=*"text/html"*%>

<%@page pageEncoding=*"UTF-8"*%>

<%@page session=*"true"* %>

<%@page import=*"com.wgq.entity.User"* %>

<%

String path = request.getContextPath();

String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort() + path + "/";

%>

<%

User user = (User)session.getAttribute("loginUser");

**if**(user==**null**||user.getName()==**null**){

response.sendRedirect(basePath+"User\_login.jsp");

**return**;

}

%>

<!DOCTYPE html>

<html>

<head>

<title>图书管理系统首页</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/style.css"* />

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap.min.css"* />

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap-table.min.css"* />

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap-editable.css"*/>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrapEx.css"*/>

<script src=*"js/jquery-3.2.1.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrap.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrap-table.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrap-table-locale-all.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrap-editable.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrap-table-editable.min.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrapTable.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

<script src=*"js/bootstrapEx.js"* type=*"text/javascript"* charset=*"utf-8"*></script>

</head>

<script type=*"text/javascript"*>

$(**function**(){

**var** oTable = **new** TableInit();

oTable.Init();

//管理员权限赋予

**if**(<%=user.getUsertype()%>==0){

$('#btn\_add').css({"display":"none"});//对普通用户进行增加功能隐藏

$('#btn\_delete').css({"display":"none"});//对普通用户进行删除功能隐藏

}

});

</script>

<body>

<div class=*"header"*>

<div class=*"topTxt pull-right"*>

<div class=*"container"*>

<a title=*"欢迎"* target=*"\_blank"*>欢迎回来</a> |

<a href=*"http://lib.jit.edu.cn"* title=*"金科院图书馆官网"* target=*"\_blank"*>金科院图书馆官网</a> |

<a href=*"html/exit.html"* title=*"退出"*>退出</a>

</div>

</div>

<div class=*"container"*>

<div class=*"topArea"*>

<h1 class=*"logo"*>

<a target=*"\_blank"*><img src=*"img/logo.jpg"* alt=*"金科院图书馆"*></a>

</h1>

</div>

</div>

</div>

<div class=*"container"*>

<div id=*"menu"*>

<nav class=*"navbar navbar-default"* role=*"navigation"*>

<div class=*"container-fluid"*>

<div class=*"navbar-header"*>

<button class=*"navbar-toggle"* data-toggle=*"collapse"* data-target=*"#example-navbar-collapse"*>

<span class=*"sr-only"*></span>

<span class=*"icon-bar"*></span>

<span class=*"icon-bar"*></span>

<span class=*"icon-bar"*></span>

</button>

<a class=*"navbar-brand"* href=*"#"*>首页</a>

</div>

<div class=*"collapse navbar-collapse pull-right"* id=*"example-navbar-collapse"*>

<ul class=*"nav navbar-nav"*>

<li class=*"dropdown"*>

<a href=*"#"* class=*"dropdown-toggle"* data-toggle=*"dropdown"*> 图书类别 <b class=*"caret"*></b></a>

<ul class=*"dropdown-menu dropdown-menu-right"*>

<li><a href=*"#"*>JAVA网站开发</a></li>

<li class=*"divider"*></li>

<li><a href=*"#"*>PHP网站开发</a></li>

<li class=*"divider"*></li>

<li><a href=*"#"*>新媒体UI设计</a></li>

<li class=*"divider"*></li>

<li><a href=*"#"*>互联网跨境电商</a></li>

<li class=*"divider"*></li>

<li><a href=*"#"*>大数据分析</a></li>

</ul>

</li>

</ul>

</div>

</div>

</nav>

</div>

</div>

<div class=*"container"*>

<div id=*"toolbar"* class=*"btn-group"*>

<button id=*"btn\_add"* type=*"button"* class=*"btn btn-default"* data-dismiss=*"modal"* aria-hidden=*"true"*>

<span class=*"glyphicon glyphicon-plus"* aria-hidden=*"true"*></span>新增

</button>

<button id=*"btn\_delete"* type=*"button"* class=*"btn btn-default"*>

<span class=*"glyphicon glyphicon-minus"* aria-hidden=*"true"*></span>删除

</button>

</div>

</div>

<div class=*"container"*>

<table class=*"table table-bordered"* id=*"table"* data-toggle=*"table"*>

<caption class=*"text-center"*><big>图书详细信息列表</big></caption>

</table>

</div>

</body>

</html>

### 2.2后端部分代码实现。

2.2.1实现JDBC链接数据库MySQL

jdbc.properties

driver=com.mysql.jdbc.Driver

url=jdbc\:mysql\://localhost\:3306/bookdb?useUnicode=true&characterEncoding=UTF-8

username=root

password=123456

#定义初始连接数

initialSize=0

#定义最大空闲

maxIdle=20

#定义最小空闲

minIdle=1

#连接池的最大值

maxTotal=15

#定义最长等待时间

maxWait=60000

2.2.2bootstrpTable.js编写，实现前端与后端的数据传递

//主机+地址

**var** host = "http://"+window.location.host+"/LibraryManageSystem/";

**var** TableInit = **function**(t\_data) {

**var** oTableInit = **new** Object();

//初始化Table

oTableInit.Init = **function**() {

//---先销毁表格 ---

$('#table').bootstrapTable('destroy');

$('#table').bootstrapTable({

url: host+'booksPage.do', //请求后台的URL（\*）

method: 'post', //请求方式（\*）

locale: 'zh-CN',

toolbar: '#toolbar', //工具按钮用哪个容器

striped: **true**, //是否显示行间隔色

cache: **false**, //是否使用缓存，默认为true，所以一般情况下需要设置一下这个属性（\*）

pagination: **true**, //是否显示分页（\*）

sortable: **true**, //是否启用排序

sortOrder: "asc", //排序方式

sidePagination: "server", //分页方式：client客户端分页，server服务端分页（\*）

pageNumber: 1, //初始化加载第一页，默认第一页

pageSize: 5, //每页的记录行数（\*）

pageList: [5,10, 25, 50, 100], //可供选择的每页的行数（\*）

search: **true**, //是否显示表格搜索，此搜索是客户端搜索，不会进服务端，所以，个人感觉意义不大

strictSearch: **false**,//设置为true启用全匹配

searchOnEnterKey:**true**,

//editable:true,//开启编辑模式

showColumns: **true**, //是否显示所有的列

showRefresh: **true**, //是否显示刷新按钮

showHeader: **true**, //是否显示头部

showFooter: **false**, //是否显示底部

showColumns: **true**, //是否显示 内容列下拉框

showToggle: **true**, //是否显示详细视图和列表视图的切换按钮

minimumCountColumns: 2, //最少允许的列数

clickToSelect: **true**, //是否启用点击选中行

//height: 500, //行高，如果没有设置height属性，表格自动根据记录条数觉得表格高度

uniqueId: "ID", //每一行的唯一标识，一般为主键列

cardView: **false**, //设置为 true将显示card视图，适用于移动设备。否则为table试图，适用于pc

//showExport: true, //是否显示导出

//exportDataType: "basic", //basic', 'all', 'selected'

contentType: "application/x-www-form-urlencoded",//必须的，没有这个后端收不到post数据

queryParamsType: 'limit', //默认值为 'limit' ,在默认情况下 传给服务端的参数为：offset,limit,sort

// 设置为 '' 在这种情况下传给服务器的参数为：pageSize,pageNumber

queryParams: oTableInit.queryParams,//传递参数（\*）

columns: oTableInit.columns,//设置表格显示的字段

responseHandler: **function**(result) {

//在ajax获取到数据，渲染表格之前，修改数据源

**return** result;

},

onEditableSave: oTableInit.onEditableSave,

onEditableHidden: **function**(field, row, $el, reason) { // 当编辑状态被隐藏时触发

**if**(reason === 'save') {

} **else** **if**(reason === 'nochange') {

}

},

});

};

//查询的参数

oTableInit.queryParams = **function**(params) {

**var** temp = { //这里的键的名字和控制器的变量名必须一直，这边改动，控制器也需要改成一样的

offset: params.offset, //第几页

limit: params.limit, //每页条数

search:params.search,//搜索字段

sortOrder: params.order,//排序方式

maxrows: params.limit,

pageindex: params.pageNumber,

};

**return** temp;

};

//编辑后操作

oTableInit.onEditableSave = **function**(field, row, oldValue, $el){

$.ajax({

type: "post",

url: host+'bookUpdate.do',

async: **false**,

data: row,

//dataType: 'JSON',

traditional: **true**,

success: **function** (data, status) {

**if** (status == "success") {

alert('数据保存成功');

}

},

error: **function** () {

alert('数据保存失败');

},

complete: **function** () {

//alert('complete');

}

});

};

//列表信息

oTableInit.columns = [{

checkbox: **true**

}, {

field: 'id',

title: 'ID',

align: 'center',

valign: 'middle',

width: '5%',

visible: **false**,

}, {

field: 'name',

title: '图书名',

align: 'center',

valign: 'middle',

sortable: 'true',

searchable:'true',

width: '20%',

editable:{

required:**true**,

validate: **function** (v) {

**if** (!v) **return** '图书名不能为空';

}

}

}, {

field: 'author',

title: '图书作者',

align: 'center',

valign: 'middle',

width: '10%',

editable: {

//mode: 'inline',

type: 'text',

title: '图书作者',

validate: **function** (v) {

**if** (!v) **return** '图书作者不能为空';

}

}

}, {

field: 'publish',

title: '出版社',

align: 'center',

valign: 'middle',

width: '20%',

editable:{

type:'text',

title: '出版社',

source: **function**(){

**if** (!v) **return** '出版社不能为空';

}

},

}, {

field: 'publishdate',

title: '出版日期',

align: 'center',

valign: 'middle',

width: '10%',

editable:{

type:'date',//日期

title: '出版日期',

source: **function**(){

**if** (!v) **return** '出版日期不能为空';

}

},

}, {

field: 'page',

title: '页数',

align: 'center',

valign: 'middle',

width: '5%',

editable:{

type:'text',

title: '页数',

source: **function**(){

**if** (!v) **return** '页数不能为空';

}

},

}, {

field: 'price',

title: '价格',

align: 'center',

valign: 'middle',

width: '5%',

editable:{

type:'text',

title: '价格',

source: **function**(){

**if** (!v) **return** '价格不能为空';

}

},

}, {

field: 'content',

title: '内容摘要',

align: 'center',

valign: 'middle',

width: '25%',

editable:{

type:'text',

title: '内容摘要',

source: **function**(){

**if** (!v) **return** '内容摘要不能为空';

}

},

}];

**return** oTableInit;

};

window.operateEvents = {

'click .save': **function**(e, value, row, index) {

$.ajax({

type: "post",

data: row,

url: '/Analyze/EditMeterMeasureHistoryData',

success: **function**(data) {

alert('修改成功');

}

});

},

'click .remove': **function**(e, value, row, index) {

$.ajax({

type: "post",

data: row,

url: '/Analyze/DeleteMeterMeasureHistoryData',

success: **function**(data) {

alert('删除成功');

$('#querylist').bootstrapTable('remove', {

field: 'MeterMeasureHistoryID',

values: [row.MeterMeasureHistoryID]

});

}

});

}

};

$(**function**(){

//新增

$('#btn\_add').on('click', **function** () {

$(document.body).append('<div id="modal"></div>');

Modal = **new** Modal();

Modal.title = "";

Modal.renderto = "#modal";

Modal.InitShow = **true**;

//Modal.btns = [{id:"btn1",text:"保存"}, { id: "closebtn" }];

tab = **new** Tab();

Modal.OnfirstInited = **function** (thismodal) {

tab.tabs = [{

id: "tab1",

title: "增加书籍资料",

active: **true**,

isiframe: **true**,

url: 'html/insertBook.html',

}];

tab.renderto = thismodal.body;

tab.Init();

}

Modal.show();

});

//删除

$('#btn\_delete').on('click', **function** () {

**var** selections = $('#table').bootstrapTable('getSelections');

**var** ids;

//alert(typeof selections);

//alert(JSON.stringify(selections));

**var** ids = $.map(selections, **function**(row) {

**return** row.id;

});

//alert(selections[0].name);

**if**(ids.length == 0){

alert("没有勾选任何内容");

}**else**{

$.ajax({

type: "post",

//data: $(ids).serializeArray(),

data: {id:ids},

url: host+'bookDelete.do',

async: **false**,

//dataType: 'JSON',

traditional: **true**,

success: **function** (data, status) {

**if** (status == "success") {

alert('数据删除成功');

$('#table').bootstrapTable('remove', {field: 'id', values: ids});

}

},

error: **function** () {

alert('数据删除失败');

},

complete: **function** () {

//alert('complete');

}

});

}

});

});

2.2.3用户登录界面完善

User\_login.js

// 全局变量a和b，分别获取用户名框和密码框的value值

**var** a = document.getElementsByTagName("input").item(0);

**var** b = document.getElementsByTagName("input").item(1);

**function** submitTest() {

**if**(!a.value) {

document.getElementById("remind\_1").innerHTML = "请输入用户名！";

**return** **false**;

}

**if**(!b.value) {

document.getElementById("remind\_2").innerHTML = "请输入密码！";

**return** **false**;

}

**return** **true**;

}

//用户框验证value值

**function** oBlur\_1() {

**if**(!a.value) {

document.getElementById("remind\_1").innerHTML = "请输入用户名！";

}

**else** {

document.getElementById("remind\_1").innerHTML = "";

}

}

//密码框验证value值

**function** oBlur\_2() {

**if**(!b.value) {

document.getElementById("remind\_2").innerHTML = "请输入密码！";

}

**else** {

document.getElementById("remind\_2").innerHTML = "";

}

}

//用户框获得焦点的隐藏提醒

**function** oFocus\_1() {

document.getElementById("remind\_1").innerHTML = "";

}

//密码框获得焦点的隐藏提醒

**function** oFocus\_2() {

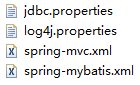
document.getElementById("remind\_2").innerHTML = "";

}

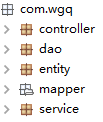
### 2.3 Spring部分代码实现。

Spring框架实现：

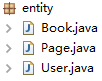
2.3.1配置文件



2.3.2项目目录



2.3.3在entity包中 Page、Book、User实体Bean：



2.3.4创建dao层：数据持久层

**package** com.wgq.dao;

**import** java.util.List;

**import** com.wgq.entity.Book;

**import** com.wgq.entity.Page;

//申明Mybatis操作的方法

//只做数据处理

**public** **interface** BookDataBaseDao {

**public** List<Book> queryBookListByPage(Page page);

**public** **int** insertBook(Book book);

**public** **int** deleteBook(Integer id);

**public** **int** updateBook(Book book);

}

2.3.5创建mapper配置文件：

BookDataBaseMapper.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"com.wgq.dao.BookDataBaseDao"*>

<!-- 对象参数传递到了语句中，id、username 和 password 属性将会被查找，然后将它们的值传入预处理语句的参数中 -->

<select id=*"queryBookListByPage"* resultType=*"com.wgq.entity.Book"*

parameterType=*"com.wgq.entity.Page"*>

select `id`,`name`,`author`,`publish`,`publishdate`,`page`,`price`,`content`

from `book`

<where>

<if test=*"search!=null"*>

name like '%${search}%'

</if>

</where>

<if test=*"sortOrder!=null"*>

order by id ${sortOrder}

</if>

<if test=*"offset>=0 and limit!=0"*>

limit #{offset},#{limit}

</if>

</select>

<!-- 添加一本书 -->

<insert id=*"insertBook"* parameterType=*"com.wgq.entity.Book"*>

INSERT into `book` (`name`,`author`,`publish`,`publishdate`,`page`,`price`,`content`) values (#{name},#{author},#{publish},#{publishdate},#{page},#{price},#{content})

</insert>

<!-- 删除一本书 -->

<delete id=*"deleteBook"* parameterType=*"int"*>

DELETE FROM `book` WHERE id = #{id}

</delete>

<!-- 更新数据 -->

<update id=*"updateBook"* parameterType=*"com.wgq.entity.Book"*>

UPDATE `book`

<set>

<if test=*"name!=null"*>

`name`=#{name},

</if>

<if test=*"author!=null"*>

`author`=#{author},

</if>

<if test=*"publish!=null"*>

`publish`=#{publish},

</if>

<if test=*"publishdate!=null"*>

`publishdate`=#{publishdate},

</if>

<if test=*"page!=null"*>

`page`=#{page},

</if>

<if test=*"price!=null"*>

`price`=#{price},

</if>

<if test=*"content!=null"*>

`content`=#{content},

</if>

</set>

where id = #{id};

</update>

</mapper>

UserDataBaseMapper.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"com.wgq.dao.UserDataBaseDao"*>

<!-- 对象参数传递到了语句中，id、username 和 password 属性将会被查找，然后将它们的值传入预处理语句的参数中 -->

<select id=*"queryUser"* resultType=*"com.wgq.entity.User"*

parameterType=*"com.wgq.entity.User"*>

select `name`,`password`,`usertype`

from `user`

<where>

<if test=*"name!=null and password!=null"*>

and `name`=#{name}

and `password`=#{password}

</if>

</where>

</select>

<!-- 注册 -->

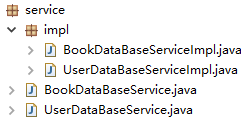
<insert id=*"registerUser"* parameterType=*"com.wgq.entity.User"*>

INSERT into `user` (`name`,`password`,`age`,`sex`,`nickname`,`usertype`) values (#{name},#{password},#{age},#{sex},#{nickname},#{usertype})

</insert>

</mapper>

2.3.6创建实现Service（接口）用来申明增删改查方法：



选取BookDataBaseServiceImpl.java说明:

**package** com.wgq.service.impl;

**import** java.util.List;

**import** javax.annotation.Resource;

**import** org.springframework.stereotype.Service;

**import** com.wgq.dao.BookDataBaseDao;

**import** com.wgq.entity.Book;

**import** com.wgq.entity.Page;

**import** com.wgq.service.BookDataBaseService;

@Service

**public** **class** BookDataBaseServiceImpl **implements** BookDataBaseService {

/\*\*

\* 表示Spring的自动注入

\* 依赖注入（DI）：依赖【需要某个类对象】

\* ,注入【Spring会自动创建并且注入到这个类对象文件中】

\*/

@Resource

**private** BookDataBaseDao dao;

/\*\*

\* 请求一页数据，根据page对象里面设置的起始位置和偏移量

\* **@return** 查询到的数据列表

\*/

**public** List<Book> queryBookListByPage(Page page) {

//第一次查询，只获取总条数

Page sp=**new** Page();

sp.setSearch(page.getSearch());

List<Book> bs = dao.queryBookListByPage(sp);

page.setCount(bs.size());

//第二次查询，查询指定的N条

**return** dao.queryBookListByPage(page);

}

/\*\*

\*添加指定ID的book

\*/

@Override

**public** **int** insertBook(Book book) {

**return** dao.insertBook(book);

}

/\*\*

\* 删除指定ID的book

\*/

@Override

**public** **int** deleteBook(Integer id) {

**return** dao.deleteBook(id);

}

/\*\*

\*更新指定ID的book

\*/

@Override

**public** **int** updateBook(Book book) {

**return** dao.updateBook(book);

}

}

2.3.7创建Controller：相当于servlet：

BookDataBaseController.java

package com.wgq.controller;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.context.annotation.Scope;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.wgq.entity.Book;

import com.wgq.entity.Page;

import com.wgq.service.BookDataBaseService;

/\*\*

\* spring mvc处理方法支持的返回方式：ModelAndView, Model, ModelMap, Map,View, String, void

\* ModelAndView构造方法可以指定返回的页面名称，也可以通过setViewName()方法跳转到指定的页面 ,

\* 使用addObject()设置需要返回的值，addObject()有几个不同参数的方法，可以默认和指定返回对象的名字。

\* 调用addObject()方法将值设置到一个名为ModelMap的类属性，ModelMap是LinkedHashMap的子类

\* @author STAR

\*

\*/

@ResponseBody

@Controller

@Scope("prototype")

public class BookDataBaseController {

@Resource

private BookDataBaseService service;

/\*\*

\* 获取一页数据

\* @param session

\* @param offset 起始页

\* @param limit 请求条数

\* @param map

\* @return

\*/

@RequestMapping(value = "/booksPage.do")

public Map<String,Object> toBookListPage(HttpSession session, Page page, ModelMap map) {

List<Book> list = null;

System.err.println("Limit:"+page.getLimit()+":"+"Offset:"+page.getOffset());

Map<String,Object> result = new HashMap<String,Object>();

list = service.queryBookListByPage(page);

result.put("total",page.getCount());

result.put("rows",list);

System.out.println("总共"+list.size()+"条数据");

return result;

}

/\*\*

\* 添加一条数据

\* @param session

\* @param book

\* @param map

\* @return

\*/

@RequestMapping(value = "/bookInsert.do",produces="text/html;charset=UTF-8")

public String bookInsert(HttpSession session, Book book, ModelMap map) {

if (service.insertBook(book) == 1) {

System.out.println("数据添加成功:"+book.toString());

return "<h2>数据添加成功</h2>";

}

return "{\"status\":\"success\",\"data\":\"\"}";

}

/\*\*

\* 删除数据

\* @param session

\* @param book

\* @param map

\* @return

\*/

@RequestMapping(value = "/bookDelete.do")

public String delete(HttpSession session, HttpServletRequest resq, ModelMap map) {

String[] ids = resq.getParameterValues("id");

for (String id : ids) {

if (service.deleteBook(new Integer(id)) == 1) {

System.out.println("数据删除成功：ID="+id);

}else{

return "{\"status\":\"error\",\"data\":\"数据删除失败\"}";

}

}

return "{\"status\":\"success\",\"data\":\"数据删除成功\"}";

}

/\*\*

\* 更新数据

\* @param session

\* @param book

\* @param map

\* @return

\*/

@RequestMapping(value = "/bookUpdate.do",produces="text/html;charset=UTF-8")

public String bookUpdate(HttpSession session, Book book, ModelMap map) {

if (service.updateBook(book) == 1) {

System.out.println("数据更新成功:"+book.toString());

return "<h2>数据更新成功</h2>";

}

return "{\"status\":\"success\",\"data\":\"\"}";

}

}

UserDataBaseController.java

package com.wgq.controller;

import javax.annotation.Resource;

import javax.servlet.http.HttpSession;

import javax.swing.JOptionPane;

import org.springframework.context.annotation.Scope;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import com.wgq.entity.User;

import com.wgq.service.UserDataBaseService;

@Controller

@Scope("prototype")

public class UserDataBaseController {

@Resource

private UserDataBaseService service;

@RequestMapping(value = "/User\_login.do",produces="text/html;charset=UTF-8")

public String updateUserPassword(HttpSession session, User user, ModelMap map) {

User u = service.queryUser(user);

if (u == null) {

map.put("name", user.getName());

map.put("message", "用户名或者密码错误！");

JOptionPane.showMessageDialog(null,"用户名或者密码错误！","错误提示！",JOptionPane.ERROR\_MESSAGE);

return "redirect:User\_login.jsp";

}

else{

map.put("userName",u.getName());

map.put("message","登陆成功！");

}

session.setAttribute("loginUser", u);

System.out.println("user:"+user.getName()+",password:"+user.getPassword());

if(u.getUsertype()==1)

JOptionPane.showMessageDialog(null,"管理员您好,欢迎登录图书管理系统！","正确提示！",0);

else

JOptionPane.showMessageDialog(null,"用户您好,欢迎登录图书管理系统！","正确提示！",0);

return "redirect:index.jsp";

}

//注册用户

@RequestMapping(value = "/User\_register.do",produces="text/html;charset=UTF-8")

public String registerUserPassword(HttpSession session, User user, ModelMap map) {

if (service.registerUser(user) == 1) {

map.put("message","注册成功！");

JOptionPane.showMessageDialog(null,"添加成功！","正确提示！",0);

System.out.println("用户添加成功:"+user.toString());

}

else{

JOptionPane.showMessageDialog(null,"添加失败！","错误提示！",0);

return "redirect:html/User\_register.html";

}

return "redirect:User\_login.jsp";

}

}

# 五、图书管理系统设计实现步骤最终说明

本次图书管理系统，建立步骤如下：

1、建立数据表，并且添加测试数据；

2、创建Web工程，命名为LibraryManageSystem；

3、创建entity建立实体bean

3、创建DAO实现相应的操作；

4、创建Service接口及其实现类，实现相应业务，并在类中加入DAO引用;

5、创建并配置控制器controller类，加入Service的引用并实现功能；

6、图书查看列表显示主页面设计：

A、实现系统管理员查看图书，进行增删改查搜索以及更新功能

B、实现系统普通管理员查看图书、搜索功能

C、默认按ID升序列出所有的帖子列表

7、进行用户注册界面设计，数据传送后台保存在数据库中。

8、进行用户登录界面设计，实现登录跳转到图书管理系统主界面。

9、进行调试，实现所想要实现的功能。

10、功能添加：实现图书查看排序功能等等。

# 六、图书管理系统实现注意事项

1、 请注意界面美观，控件摆放整齐；

2、 请注意代码的书写、命名符合规范，在代码中添加必要的注释；

3、 请注意操作数据库时进行必要的异常处理。