

**NANCHANG UNIVERSITY**

**Web开发设计大作业报告**



**题 目:** 书店信息管理系统

**学 院：** 软件学院

**专 业：** 信息安全

**班 级：** 193班

**完成人数：** 1人

**人 员：** 8003119100 丁俊

**起讫日期：** 2021.5.23-2021.6.18

**任课教师：** 罗 铭 职称： 教 授

**完成时间：** 2021.6.18

**填表日期：** 2021 年 6 月 18 日

目录

[一、选题和需求介绍 3](#_Toc74996786)

[1.1选题 3](#_Toc74996787)

[1.2需求介绍 3](#_Toc74996788)

[1.2.1系统功能需求 3](#_Toc74996789)

[1.2.2系统环境 3](#_Toc74996790)

[二、结构设计和技术要求 4](#_Toc74996791)

[2.1程序文件结构 4](#_Toc74996792)

[2.2整体模块和功能实现 6](#_Toc74996793)

[2.2.1Util模块 6](#_Toc74996794)

[2.2.2poji模块 8](#_Toc74996795)

[2.2.3Dao模块 9](#_Toc74996796)

[2.2.4Service模块 12](#_Toc74996797)

[2.2.5 web模块 17](#_Toc74996798)

[2.2.6功能实现 23](#_Toc74996799)

[三、界面设计 29](#_Toc74996800)

[四、系统测试 31](#_Toc74996801)

[五、思考与总结 36](#_Toc74996802)

# 一、选题和需求介绍

## 1.1选题

该项目是从图书和购买图书信息层面来设计的，用于处理书籍信息和购买图书信息并且可以按要求显示图书的信息。可以方便管理员管理书籍和用户购买指定的图书。

## 1.2需求介绍

## 1.2.1系统功能需求

该系统是书店信息管理系统，主要模块分为两个方面，分别是管理员和用户顾客两个层面设计。管理员模块其中管理员可以管理图书进行相关操作，如添加、增加和修改，核心内容还包括分页显示。用户模块需要登录和注册账号并进行登录操作后可以购买图书、查看我的书架、多组合条件查询图书、分页显示、修改密码。

## 1.2.2系统环境

开发语言: java语言，面向对象编程

操作系统: Microsoft Windows 10

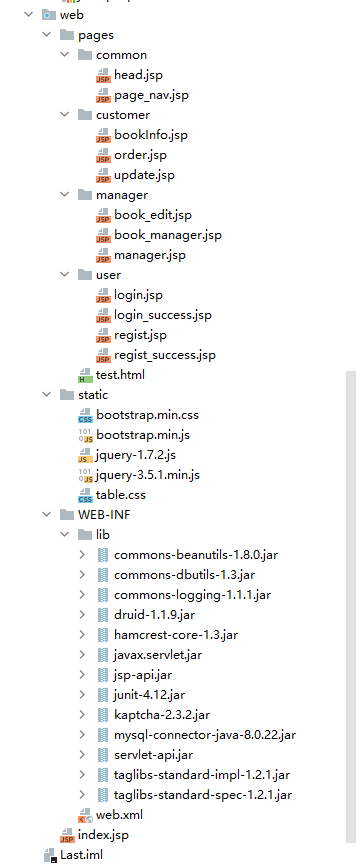
JAVA开发包: JDK1.8

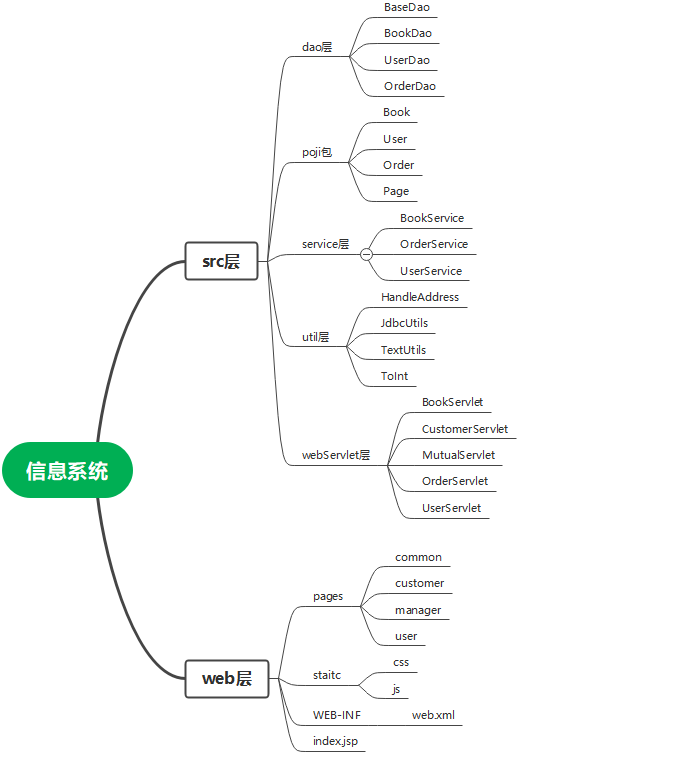
开发平台: IDEA 2020版本

所用jar包: commons-beanutils-1.8.0.jar、commons-dbutils-1.3.jar、druid-1.1.9.jar、hamcrest-core-1.3.jar、javax.servlet.jar、jsp-api.jar、mysql-connector-java-8.0.22.jar、servlet-api.jar、taglibs-standard-impl-1.2.1.jar等

# 二、结构设计和技术要求

## 2.1程序文件结构

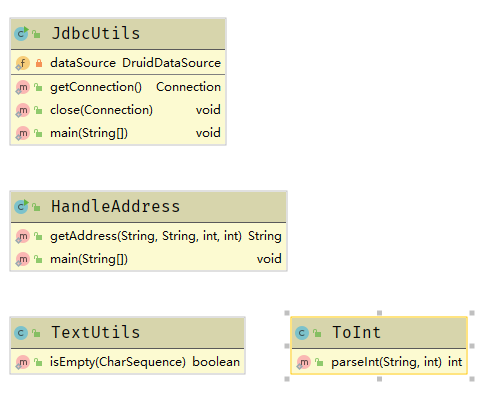




Index.jsp是主界面

## 2.2整体模块和功能实现

### 2.2.1Util模块



先在mysq中创建数据库和t\_user、t\_book、t\_order表

create database book;

use book;

create table t\_user(

`id` int primary key auto\_increment,

`sname` varchar(42) not null,

`username` varchar(20) not null unique,

`password` varchar(32) not null,

`email` varchar(200)

);

create table t\_book(

`id` int primary key auto\_increment,

`name` varchar(100),

`author` varchar(100),

`price` decimal(11,2),

`stock` int,

`type` varchar(100)

);

create table t\_order(

`id` int primary key auto\_increment,

`username` varchar(100),

`bookId` varchar(100),

`bookName` varchar(100),

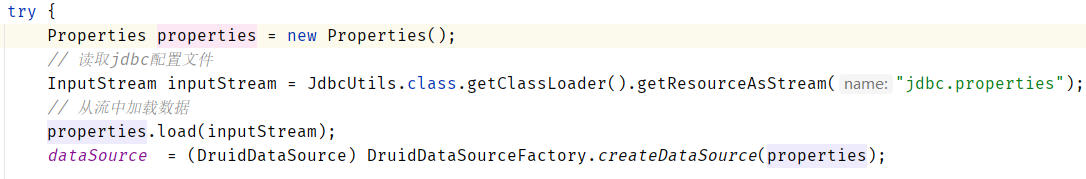
`author` varchar(100),

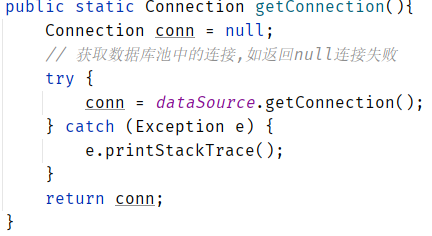
`bookType` varchar(100),

`bookNum` int

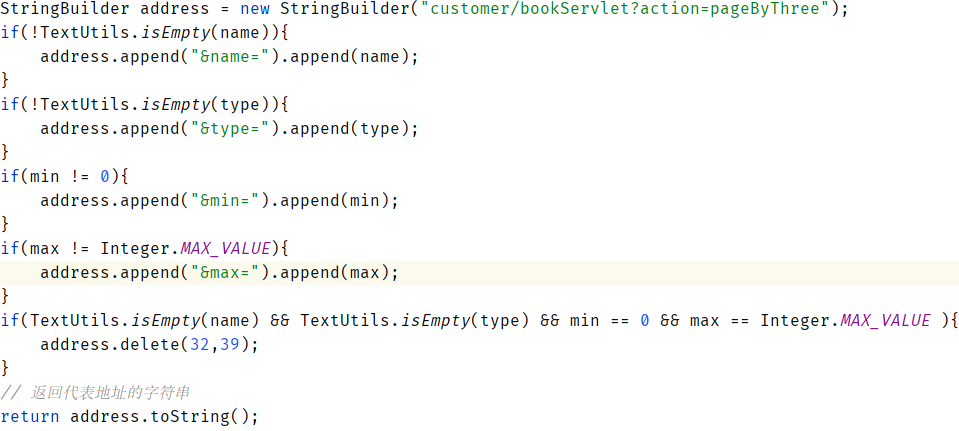
);

Util模块提供各种在此项目中频繁要使用到的工具类，分别是JdbcUtils数据库连接池，使用配置文件连接数据库连接池，使用到了dbutil核心jar包。

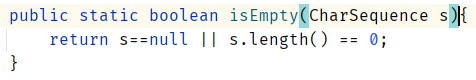




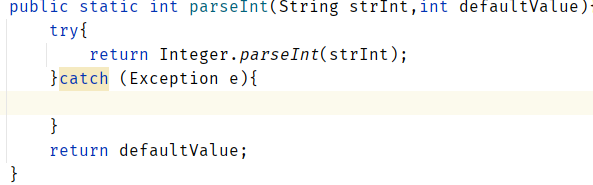
HandleArress类是专门用来处理记录浏览器跳转不同页面所获取的参数。



TextUtils类判断一个字符串是否为空，封装成一个方法。



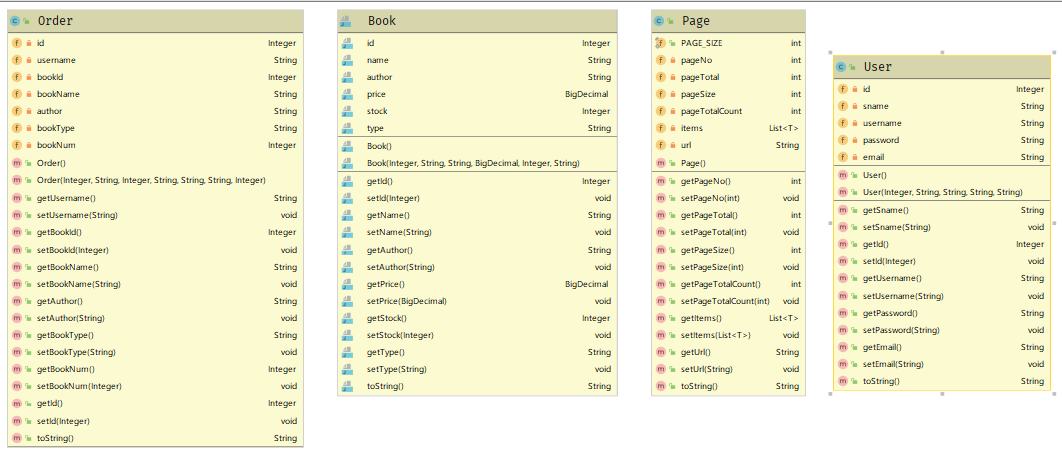
ToInt类将一个字符串转换成整型，否则转换成自定义的默认值。

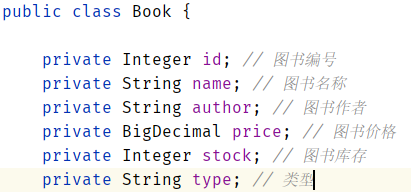


### 2.2.2poji模块

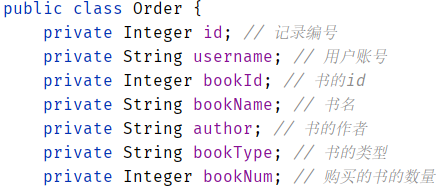
poji模块中定义了3个对象实物类User、Book、Order和1个分页类Page。

实体类对象



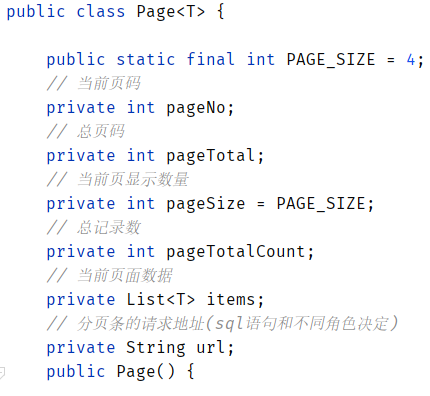




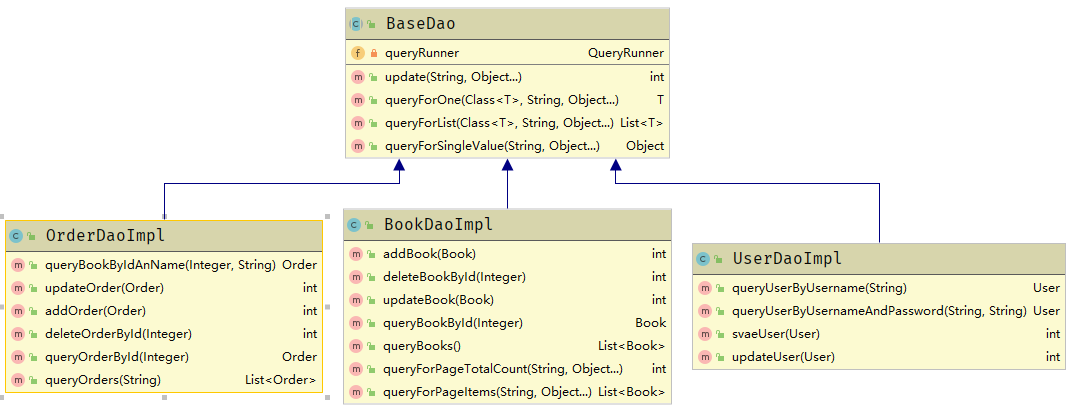


3个对象实体类都定义了相关的get和set方法、toString方法和初始化构造方法。

分类对象类

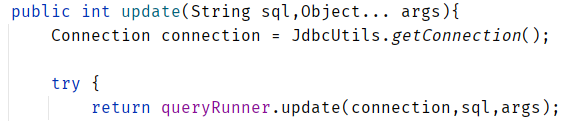


### 2.2.3Dao模块

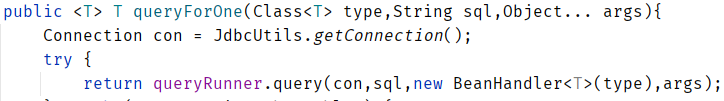


**BaseDao抽象类**：提供数据库查询操作接口，创建使用QueryRunner对象方法访问数库，还有ResultSetHandler结果处理器接收查询结果。BeanHandler表示把一行的数据封装到一个实体对象中，BeanListHandler 把多行的数据封装到一个List集合中，ScalarHandler结果返回单个值。

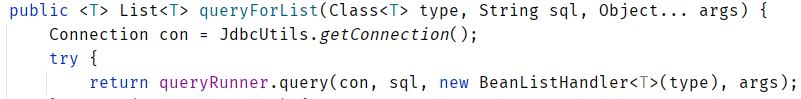
执行一条语句



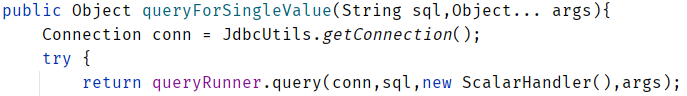
查询返回一条javabean



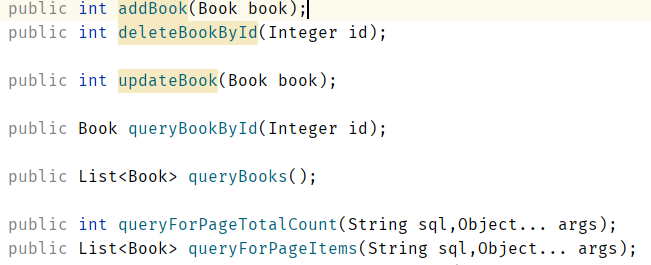
查询返回多个javabean



查询返回单个值、一行一列

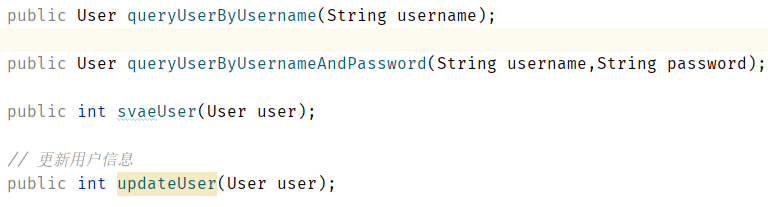


**BookDao接口和BookDaoImpl实现类**：提供对图书操作和管理的接口和实现方法，BookDaoImpl实现BookDao接口并且继承BaseDao抽象类。



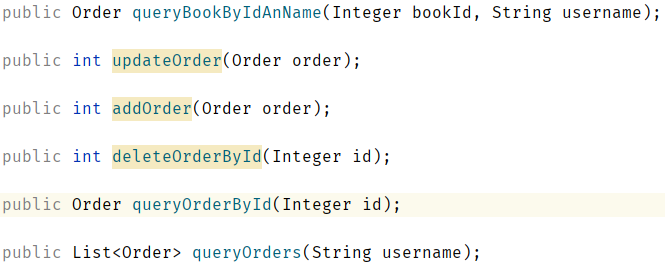
以上方法都直接使用封装好的BaseDao抽象类中的查询方法，把底层对数据库连接池的操作封装成函数直接进行调用进而形成新的封装方法。

**UserDao接口和UserDaoImpl实现类**：提供对用户信息的管理方法，UserDaoImp实现UserDao接口和继承BaseDao抽象类。



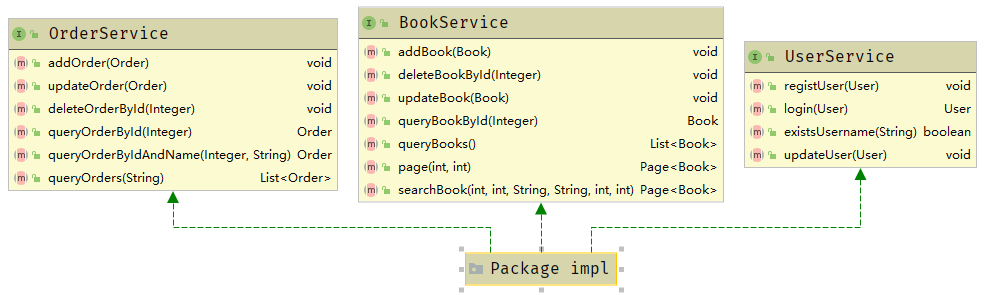
分别是根据账号查询用户、根据账号密码两条件查询、保存插入用户信息到数据库中、更新用户信息，上述方法直接调用BaseDao抽象类中的方法，也把函数进行封装提供后层更快捷的接口和函数方法。

**OrderDao接口和OrderDaoImp实现类**：提供对订单管理的方法，



分别是根据书的id和用户账号两个条件查询订单记录、更新修改订单、添加订单、根据订单id删除订单、根据id查询订单、根据账号查询订单记录

### 2.2.4Service模块



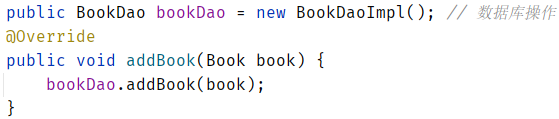
Service模块中定义的方法在Servlet方法充当直接被调用的对象方法类，在函数方法体中初始化了一个BookDao、OrderDao、UserDao对象，对Dao层的对象函数进行再一次封装。

BookService

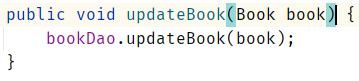
接口定义：



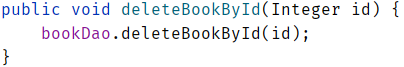
添加图书



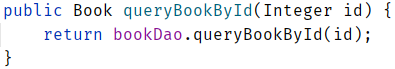
修改图书



删除图书



根据图书id查找单本书

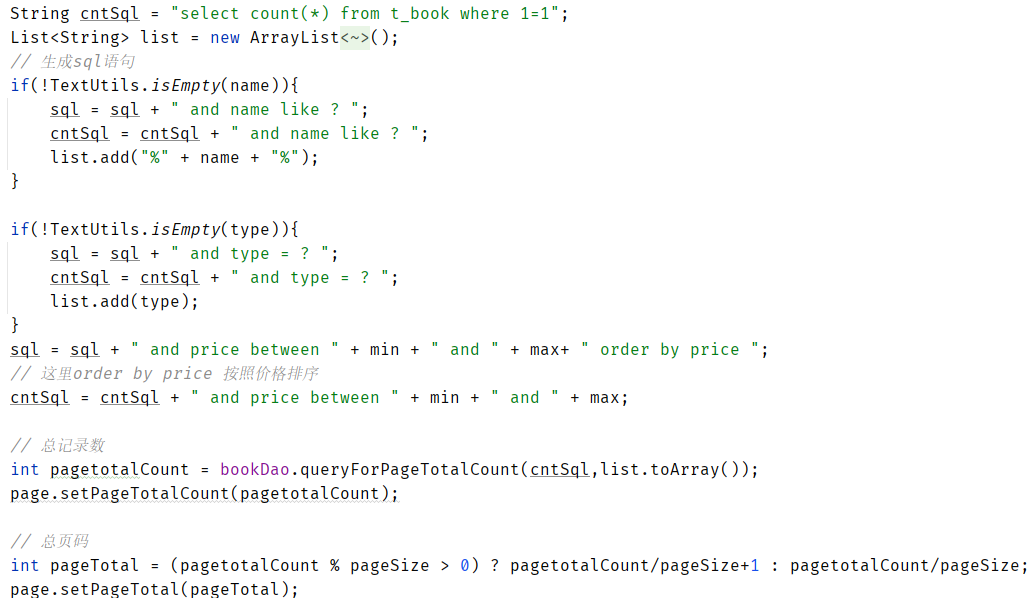


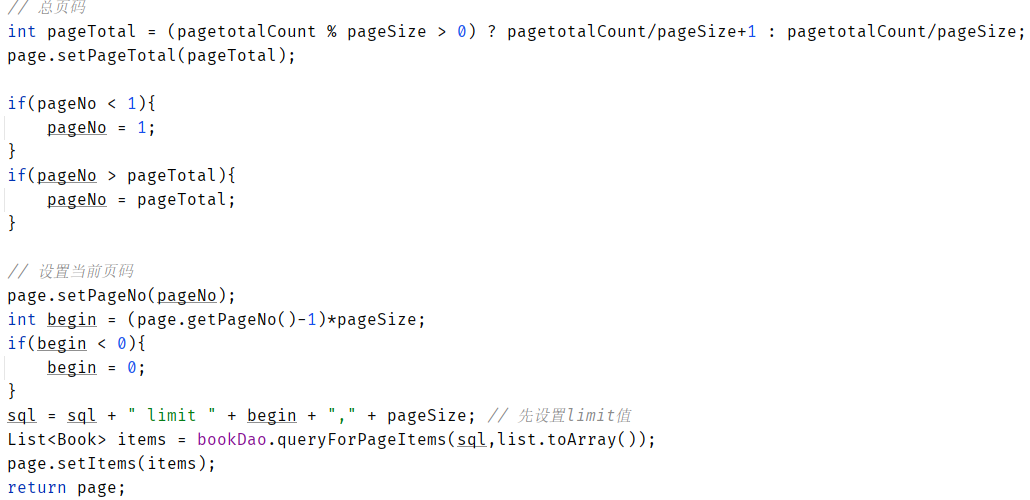
**图书分页方法(查询所有图书)**



函数参数是pageNo当前页和pageSize页面大小，“select count(\*) from t\_book”查询图书的数目，通过总记录数/每页大小=总页数(除不尽要加1)，查询起点begin = (当前页-1)\*每页大小，再通过“limit begin,pageSize”有限查询某页的多行记录返回一个列表设置为page对象中list集合，返回处理好的page对象。

**条件查询并分页**

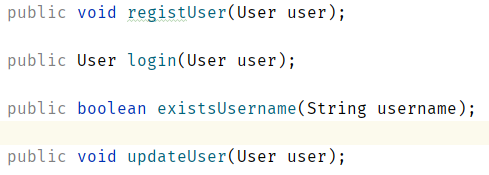




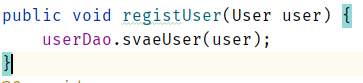
对传进来的查询参数进行判空，list集合是用来记录sql语句的查询参数的，作为查询参数传到对应的数据库操作函数中，cntsql是专门用来查询该条件对应所得记录的总数目，sql是多条件组合查询的语句参数最后加上所求得的pageNo和pageSize作为limit子句中的参数，注意这里的多组合查询语句可能查不到几条数据从而导致begin<0，所以要进行判断并保证begin>0。最终把链接的sql语句传给queryForPageItems函数返回一个集合作为page对象的list数据并返回page对象。

UserService

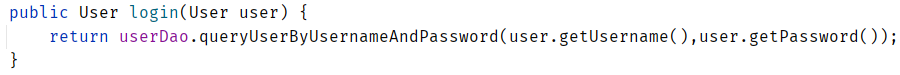
接口定义：



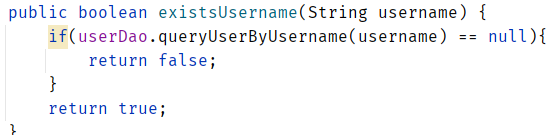
注册：也即保存用户对象信息到数据库中。



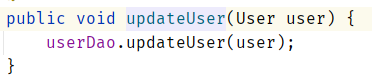
登录：判断输入的账号和密码在数据库中是否能匹配，如果存在就返回这个User对象，否则返回null。



是否存在该用户名：

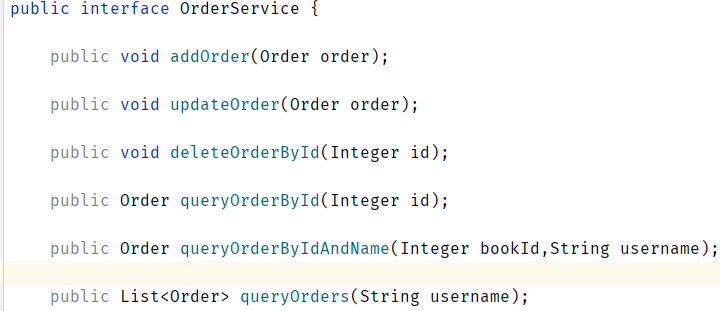


更新用户信息：

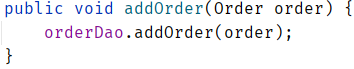


OrderService

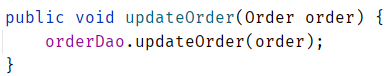
接口定义：



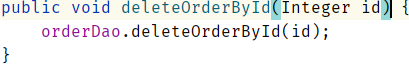
添加订单：



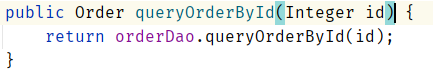
更新订单：



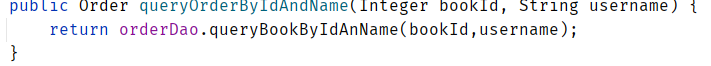
删除订单：



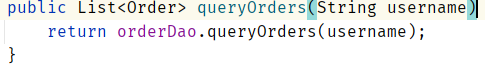
根据id查询订单：



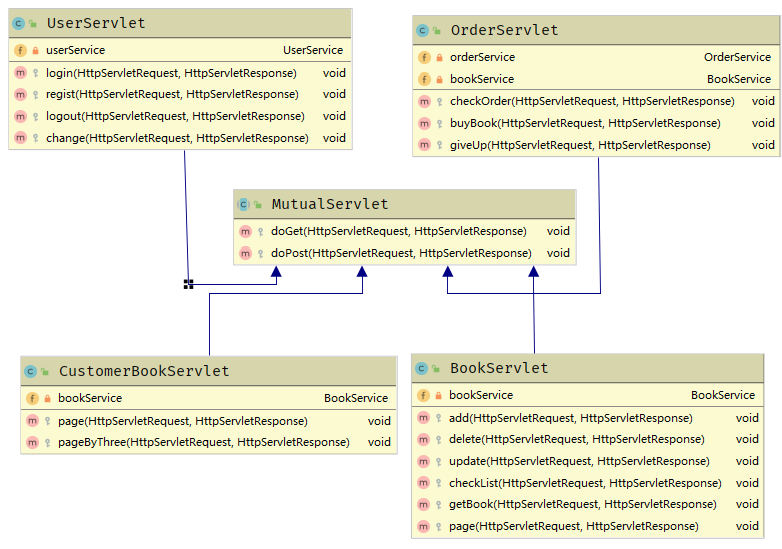
根据图书id和登录账号查询：



查询某个用户有关的图书：



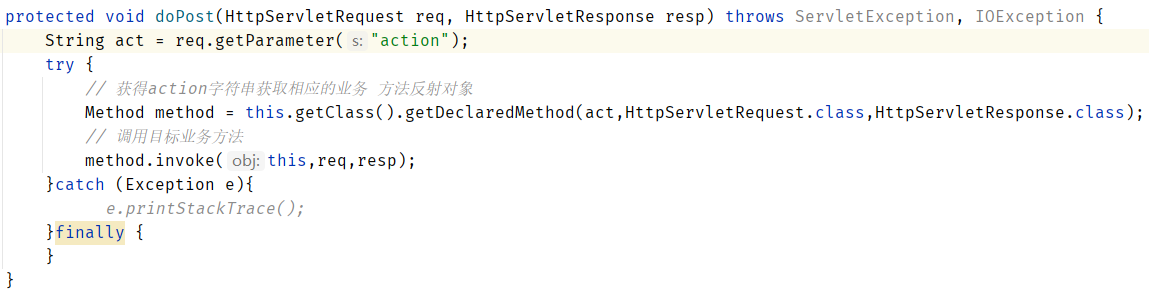
### 2.2.5 web模块



**Web.xml文件配置**



Web模块中其实有四个主要的servlet类，另外的MutualServlet类是公有的被继承的servlelt类，处理客户端向服务器发出的Post和Get请求，其中使用到了带有指定参数的指定对象调用由此Method对象表示的底层方法。先获取action参数的值，返回一个Metho对象，反映此Class对象所表示的类或接口的指定已声明方法。例如action = page表示调用方法名为“action”的函数方法。



**UserServlet**

处理与用户相关的业务比如登陆注册、注销、修改密码。

**登录函数**



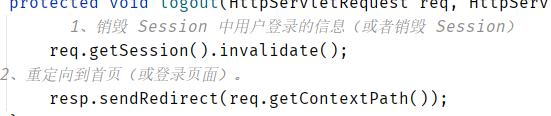
获取输入框的内容并在数据库中查找该用户，如果查不到说明不存在该用户，提示用户名或密码错误；否则利用session对象设置全局变量的对象值，然后跳转到登录成功界面，并显示用户数据。

**注册函数**



判断输入框的账号是否在数据库中存在，如果存在给出提示信息，反之跳转到注册成功界面。

**注销函数**



销毁Session中用户登录的信息，重定向跳转到主工程路径下的index.jsp文件。

**修改密码**



先更新用户信息，获取Session对象中原有的用户密码和现在输入的密码，如果一致说明没有修改密码，然后跳转到登录成功页面；如果密码修改过则跳转重新登陆界面。

**BookServlet**

处理与书籍相关的业务，比如添加、修改、删除、分页查询。

**添加图书**



**删除图书**



**更新图书**



**获得分页对象**



**CustomerBookServlet**

客户的分页显示和多条件显示功能。

多条件查询分页业务函数

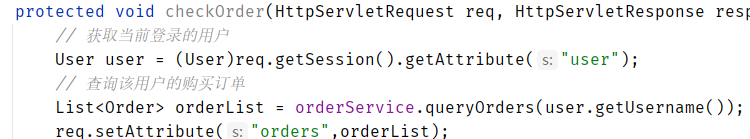


获取客户端传过来的分页参数值，有书籍名模糊查询、书籍价格范围、书籍类型等。

**OrderServlet**

处理订单相关的业务，比如查询当前用户所属的订单、买书(加入书架)、删除书架中的书。

**查询订单(书架)**



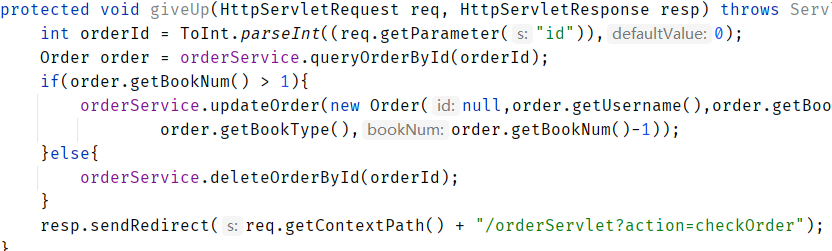
根据当前登录的Session中的User对象中的账号查询该用户所对应的订单，即该用户所购买的所以书籍，然后在request域中设置orders对象的值。

**买书(加入书架)**



查询该用户所属的订单(书架)中是否存在该书，如果存在则数量更新加1；不存在添加一条记录且所属的该书数量为1。

**删除书架中的书**



同理，当不要某本书时，如果该书数量大于1，更新数量减1；当等于1时，则直接删除有关该书的记录。

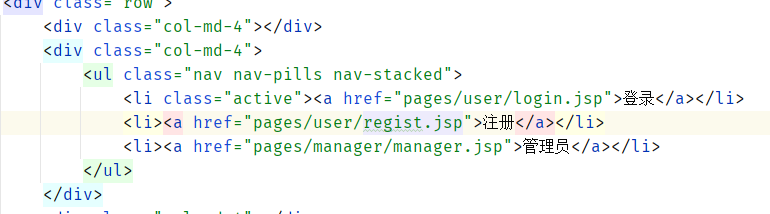
### 2.2.6功能实现

**公共文件head.jsp**



在head.jsp文件中包含了工程路径，<base>标签将当前文件的相对路径位置定位到web目录下，并且把必要的css和js文件包含，方便需要这些样式的其他jsp文件静态包含，利于代码复用功能，避免过多的冗余。

**主界面**

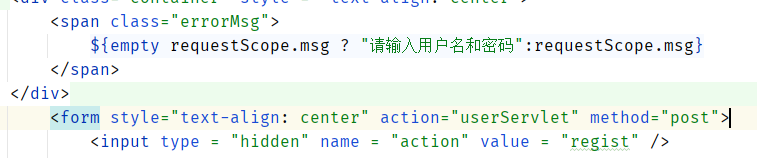


**注册功能**

1. 获取请求的参数
2. 检查用户名是否正确
3. 可用调用Service保存到数据库，跳到注册成功页面regist\_success.html
4. 不可用跳回注册页面

注册表单

达



表单提交的地址为userServlet，上部有一个提示框，当注册失败时，可以通过设置request中msg字符串的值进行错误显示。

**登录功能**

1. 请求的参数
2. 调用userservice处理业务，userService.login()登录
3. 根据login()方法返回结果判断登录是否成功，成功返回login\_success.html;失败跳回登录页面

登录表单



设置action参数的值为regist，顶部的错误提示框是当登录失败时跳转回来的时候读取request域中的数据显示的。

**分页显示功能**

这里有个单独的分页条page\_nav.jsp，可以被静态包含

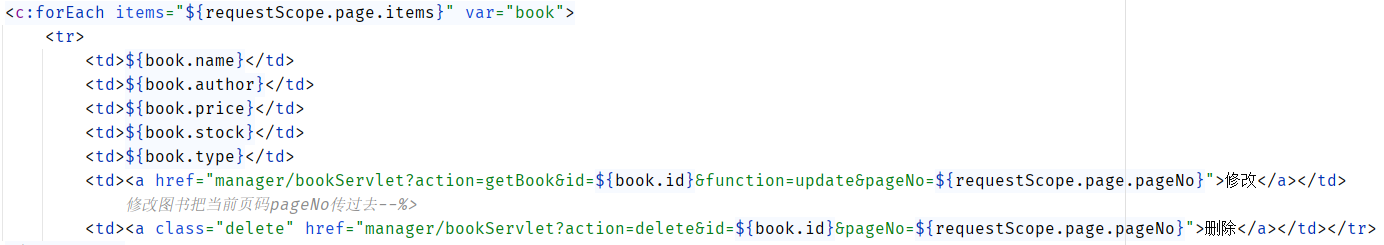


当此时在第一页时，上一页和首页的链接就利用c:if判断改成文字而不是链接；同理当此时在最后一页时，下一页和末页的链接也改成纯文字，以防发生越界而显示数据为空的异常。这个分页条是通用的，点击跳页链接可以跳转到分页处理业务然后又跳回来，并且显示了当前页和总页数，这些都已经被封装在page对象中，直接读取处理好的page对象中的数据即可。



给可自定义跳转按钮添加响应，判断输入的页面是否超过总页码，如果没有超过则提示不能跳转，反之跳转到指定页面。

**添加和修改图书功能**









value=“${param.function}”获取当前操作是添加还是修改，把值传给action。

利用c:forEach语句遍历page分页对象，增加图书记录数据，修改和删除三个链接。

manager/bookServlet?action=getBook&function=update

manager/bookServlet&action=function

book\_edit.jsp

function=add

function=update

/pages/manager/book\_manager.jsp

/manager/bookServletaction=page&pageNo=?

**删除图书功能**

/pages/manager/book\_manager.jsp

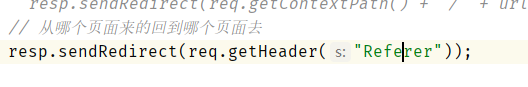
manager/bookServlet?action=delete$id=?&pageNo=${requestScope.pege.pegNo}

/manager/bookServlet?action=page&pageNo=

并且添加和更新完成后可以通过获取总页码在总页数的最后一页进行显示，删除可以在当前页面(获取当前页码)进行显示。

**用户买书和删除书**

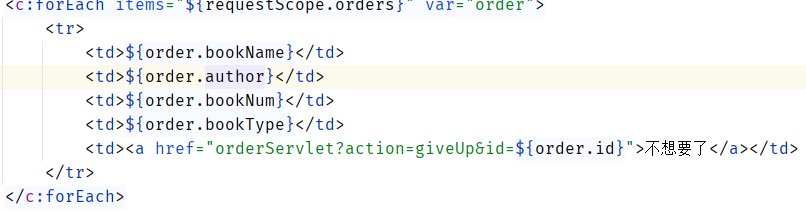




orderServlet?action=buyBook&bookId=?

返回上一页面

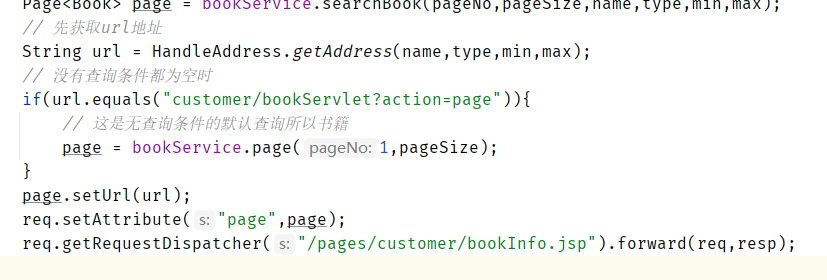
用户点击购买图书之后，跳转到orderServlet中的buyBook方法，处理完成后直接跳转回原来的页面，相当于一直在当前页面没动。



当用户点击查询订单后，即看我的书架界面，跳转到orderServlet?action=giveUp，删除书架中的书籍后，跳回我的订单显示并且需要更新，即地址/orderServlet?action=checkOrder。

**多条件查询**

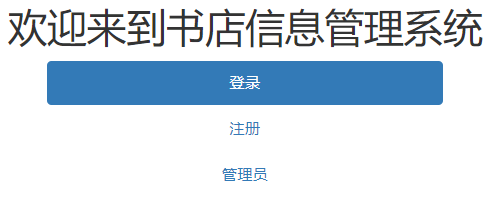




查询表单，因为要把查询条件记录下来，所以要在输入框中获取request域中设置查询前输入的数据并连续记录，这样就能实现多条件查询并分页显示，不记录的话查询条件就会无效。然后设置url的地址，记录在page对象中再次挑战到bookInfo.jsp文件。

# 三、界面设计

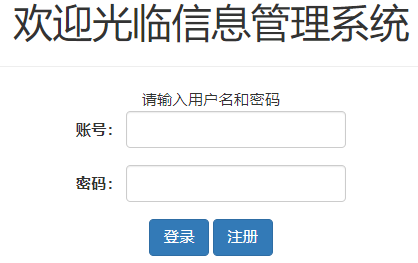
**主界面**



**管理员界面**



**登陆界面**



**注册界面**



**登录成功**



**购买图书和多条件查询界面**



**我的订单**



**管理界面**





# 四、系统测试

用户

**登录**

输入错误的账号或密码会有提示信息



登录成功后上部有多个选项，点击购买图书即开始分页显示内容，输入要查询的组合条件点击查询。



**注册**

当用户名已存在会提示这个用户名账号以前别的用户注册过。



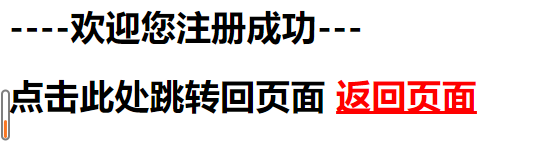
密码不一致



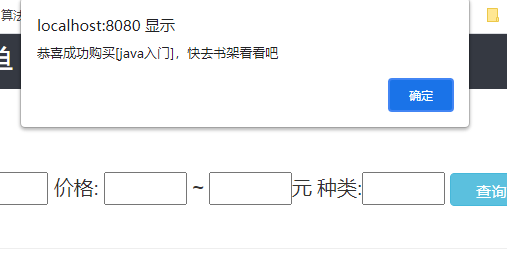
邮箱格式不合法



注册成功



购买图书





购买后新增图书或数量



删除书架图书



多组合条件查询

查询包含“c”字符的书籍



查询价格区间为30~120，种类为编程的书籍



查询编程类书籍

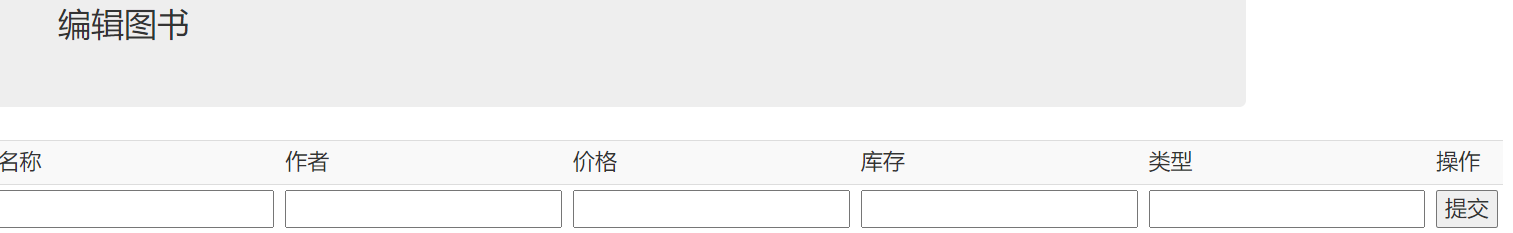


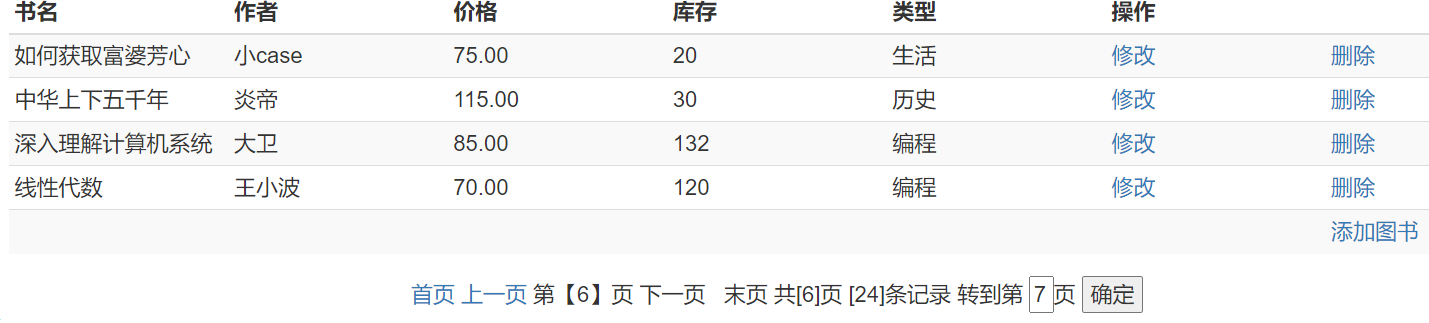
管理员

添加图书

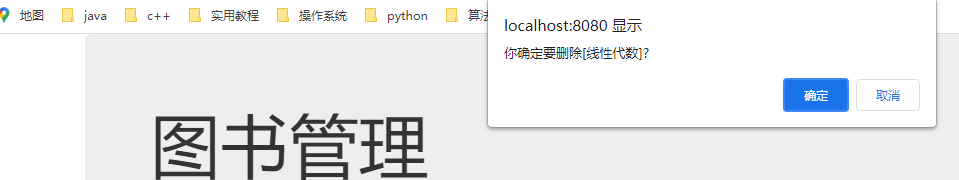


现在处于最后一页，当添加图书后书籍会自动在最后一页显示。



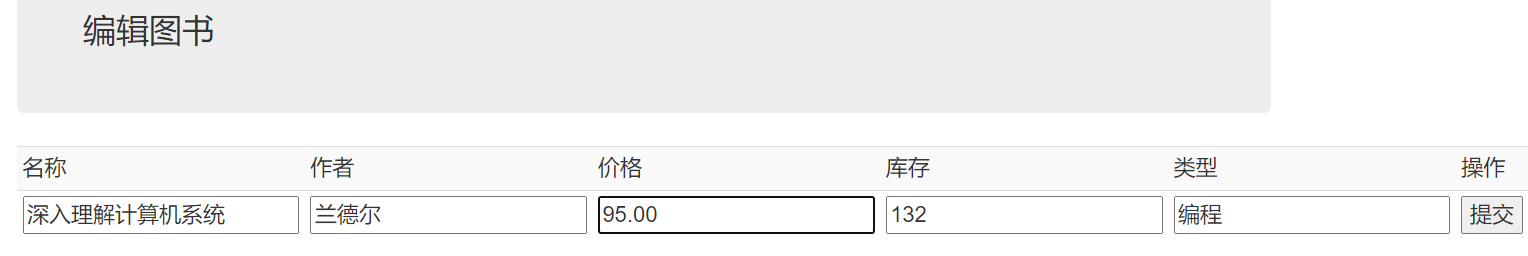


删除图书





修改图书





# 五、思考与总结

通过这次web大作业我深刻理解了网站前台和后台的服务和分工合作的关系，后台对于用户服务的需求实现至关重要，在实现页面跳转以及数据修改和保存方面我也遇到了很多问题，一直遇到的404、500问题也提升了我对代码的debug能力。让我的编程思维又进一步得到了提升。同时也收获到了有关于网页和网页响应方面的知识。