

**设计题目：**学生宿舍管理系统

**专业年级：**信息与计算科学2018（2）

**指导教师：**邵庆

**姓名：**何俊言

**学号：**18024040224

设计时间：2020年7月14日

目录

[第一章 开发背景和开发环境工具 3](#_Toc13342)

[1.1开发背景 3](#_Toc26207)

[1.2开发环境及开发工具 4](#_Toc22637)

[第二章 学生宿舍管理系统的设计与分析 6](#_Toc16197)

[2.1系统需求分析 6](#_Toc3429)

[2.2系统可行性分析 7](#_Toc11774)

[2.3 系统总体设计目标 7](#_Toc12921)

[2.3.1系统设计目标 7](#_Toc7330)

[2.3.2开发设计思想 7](#_Toc14349)

[2.3.3系统功能模块设计 8](#_Toc22025)

[第三章 系统详细设计与实现 11](#_Toc29072)

[3.1各功能关键代码设计 11](#_Toc11254)

[3.1.1登录页面及系统操作页面 11](#_Toc19846)

[3.1.2数据库操作基础类 15](#_Toc20354)

[3.1.3学生服务端部分代码 20](#_Toc19130)

[3.2数据库设计 23](#_Toc30201)

[3.2.1数据库表 23](#_Toc31797)

[3.2.2宿舍楼表 23](#_Toc30089)

[3.2.3宿舍房号表 23](#_Toc14760)

[3.2.4宿管表 24](#_Toc8300)

[3.2.5 入住表 24](#_Toc18538)

[3.2.6 学生表 24](#_Toc6782)

[3.3 界面设计与功能展示 24](#_Toc26948)

[3.3.1登陆功能 24](#_Toc1114)

[3.3.2欢迎页面 25](#_Toc22931)

[3.3.3添加信息功能 26](#_Toc28778)

[3.3.4查询功能 26](#_Toc2701)

[3.3.5删除功能 27](#_Toc31134)

[3.3.6修改功能 28](#_Toc10160)

[第四章 总结分析 29](#_Toc20635)

# 第一章 开发背景和开发环境工具

## 1.1开发背景

随着高校规模的不断扩大，学生宿舍数量也急剧增加，有关学生宿舍的各种信息量也成倍增长为了能够为高校学生宿舍信息管理提供一种更加高效实用的管理手段，为学生宿舍信息的存储、计算、统计、分析、交流提供一种更加安全快捷的信息平台，并且能够减少大量的人工操作，以及在人工操作中由于人为因素而引起的数据错误，保证学生宿舍信息数据的安全性和完整性，使学生宿舍管理人员能够轻松,正确无误地完成各项工作，为学生宿舍管理工作服务。

意义：

1. 保证信息的准确性和时效性，随时通过计算机对系统进行管理，使得学生宿舍信息处于最新状态，保证统计数据和分析的准确。
2. 减轻劳动强度、提高工作效率、增加学生宿舍管理的透明度。高校学生宿舍管理信息系统可以发挥计算机的强大功能，让管理人员从大量繁琐的手工劳动中解放出来，将单调枯燥的纸张涂写变成灵活的电子信息操作。并充分利用网络优势,加快学校内部学生宿舍信息和文档的发布、传送和获取，改善和优化学生宿舍工作，使各个部门之间的工作联系紧密、井然有序、清楚明晰，大大减轻工作负担，提高工作效率，增加学生宿舍管理的透明度。
3. 降低管理成本，由于采用本系统的原因，使辅导员管理工作一改往日埋在各种表格中的被动局面，基本实现了无纸化办公，根除了很多日常管理中的浪费现象，体现出现代化学生宿舍管理的优势。

4、规范化管理，由于采用了计算机统计分析学生宿舍信息，一些原始信息在录入时就必须要求准确录入，这就要求管理人员在提供信息时要准确，同时要求在录入时也要十分认真，这样做是为了将错误降到最低，提高系统工作的准确率，从而实现学生宿舍管理工作规范化。

## 1.2开发环境及开发工具

后端使用Jsp、Servlet、JDBC等技术。数据库使用MySQL 8.0 ，Tomcat 8，JDK 8

开发环境使用Eclipse JavaEE编写Java，Navicat管理数据库。

前端部分用EasyUi框架（ajax+接口）

1. Eclipse 是一个开放[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "https://baike.baidu.com/item/eclipse/_blank)的、基于[Java](https://baike.baidu.com/item/Java/85979" \t "https://baike.baidu.com/item/eclipse/_blank)的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java[开发工具](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7" \t "https://baike.baidu.com/item/eclipse/_blank)（Java Development Kit，[JDK](https://baike.baidu.com/item/JDK/1011" \t "https://baike.baidu.com/item/eclipse/_blank)）。
2. 本系统是为高校设计，旨在解决高校管理员处理数量庞大的学生住宿以及相关信息的问题，降低其繁琐性。因此，若想投入使用，必须能够清晰的展现所有的学生住宿及管理数据，但又因它只在学校使用，因此不需要太高的交互能力，考虑体积大小以及成本，本系统选用MySQL数据库。MySQL是一个关系型数据库管理系统，是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的RDBMS应用软件之一。MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。
3. Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由[Apache](https://baike.baidu.com/item/Apache/6265" \t "https://baike.baidu.com/item/tomcat/_blank)、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。
4. HTML5、CSS3、JSP、Java Servlet

本系统的后端是采用Java语言编写的，而前台页面则采用HTML5、 CSS3和JSP以及Java Servlet四大技术。[HTML](https://baike.baidu.com/item/HTML" \t "_blank)是[万维网](https://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91" \t "_blank)的核心语言，是一种[超文本标记语言](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/6972570" \t "_blank)，[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/6805073" \t "_blank)下的一个应用。HTML 不是一种[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80/9845131" \t "_blank)，而是一种标记语言，可以包含图片、[链接](https://baike.baidu.com/item/%E9%93%BE%E6%8E%A5/2665501" \t "_blank)，甚至音乐、[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F/71525" \t "_blank)等非文字元素。，是网页制作所必备的。CSS层叠样式表是一种用来表现[HTML](https://baike.baidu.com/item/HTML" \t "_blank)或[XML](https://baike.baidu.com/item/XML" \t "_blank)等文件样式的计算机语言。

CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。CSS 能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。

JSP是java服务器页面，其根本是一个简化的Servlet设计，是一种动态网页技术标准。JSP技术有点类似ASP技术，它是在传统的网页HTML（文件(\*.htm,\*.html)中插入Java程序段和JSP标记，从而形成JSP文件，后缀名为(\*.jsp)。 用JSP开发的Web应用是跨平台的，既能在Linux下运行，也能在其他操作系统上运行。它实现了Html语法中的java扩展（以 <%, %>形式）。JSP与Servlet一样，是在服务器端执行的。通常返回给客户端的就是一个HTML文本，因此客户端只要有浏览器就能浏览。JSP技术使用Java编程语言编写类XML的tags和scriptlets，来封装产生动态网页的处理逻辑。网页还能通过tags和scriptlets访问存在于服务端的资源的应用逻辑。JSP将网页逻辑与网页设计的显示分离，支持可重用的基于组件的设计，使基于Web的应用程序的开发变得迅速和容易。 JSP是一种动态页面技术，它的主要目的是将表示逻辑从Servlet中分离出来。

Java Servlet是JSP的技术基础，而且大型的Web应用程序的开发需要Java Servlet和JSP配合才能完成。JSP具备了Java技术的简单易用，完全的面向对象，具有平台无关性且安全可靠，主要面向因特网的所有特点。

1. EasyUi是一种基于jQuery、Angular.、Vue和React的用户界面插件集合。为创建现代化，互动，JavaScript应用程序，提供必要的功能。完美支持HTML5网页的完整框架，简单但功能强大。AJAX（异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术。主要包含了以下几种技术：基于[web标准](https://www.baidu.com/s?wd=web%E6%A0%87%E5%87%86&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)XHTML+CSS的表示；使用 DOM进行动态显示及交互；使用 XML 和 XSLT 进行数据交换及相关操作；使用 XMLHttpRequest 进行异步数据查询、检索；使用 JavaScript 将所有的东西绑定在一起。AJAX不是指一种单一的技术，而是有机地利用了一系列相关的技术。Ajax应用程序的优势在于：1、 通过异步模式，提升了用户体验。2、 优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了带宽占用。3、Ajax引擎在客户端运行，承担了一部分本来由服务器承担的工作，从而减少了大用户量下的服务器负载。
2. Navicat是一套快速、可靠并价格相当便宜的[数据库管理](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86/10509024" \t "https://baike.baidu.com/item/navicat/_blank)工具，专为简化数据库的管理及降低系统管理成本而设。它的设计符合[数据库管理员](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E5%91%98/1216449" \t "https://baike.baidu.com/item/navicat/_blank)、开发人员及中小企业的需要。Navicat 是以直觉化的图形用户界面而建的，让你可以以安全并且简单的方式创建、组织、访问并共用信息。
3. Java

Java在本系统的开发中得到了极大的应用。 Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。因此，在本系统中几乎所有的用户与数据库的映射，交互技术都通过Java来实现。JDK是 [Java](https://baike.baidu.com/item/Java/85979" \t "https://baike.baidu.com/item/jdk/_blank) 语言的[软件开发工具包](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7%E5%8C%85/10418833" \t "https://baike.baidu.com/item/jdk/_blank)，主要用于[移动设备](https://baike.baidu.com/item/%E7%A7%BB%E5%8A%A8%E8%AE%BE%E5%A4%87/9157757" \t "https://baike.baidu.com/item/jdk/_blank)、嵌入式设备上的java应用程序。JDK是整个java开发的核心，它包含了JAVA的运行环境（JVM+Java系统类库）和JAVA工具。

# 学生宿舍管理系统的设计与分析

## 2.1系统需求分析

学生宿舍管理系统的开发主要是为了解决学生住宿及其相关问题，同时满足管理者方便管理的需求。随着各个高校不断扩招，与住宿的同学数量的增多，各种住宿问题大量出现，本着解决问题，便捷为主的目标开发了本系统。

## 2.2系统可行性分析

本系统方便宿管查询并管理住宿同学的信息，随时了解自己管理部分的动态内容、方便学生查询，增加或者修改删除自己的信息，进行实时缴费等以及方便管理员安排宿管的工作，随时了解各个宿管，楼宇，以及学生的动态信息并能做出修改。使宿舍管理员的管理工作更系统化、规范化、自动化，大大提高管理效率提高管理效率，降低管理成本，使得每个人各取所需，方便的找到自己授权内需要的信息。把安全、高效、便捷、服务和自律相结合,实现学生自主和学校引导相结合,管出水平,管出活力。

## 2.3 系统总体设计目标

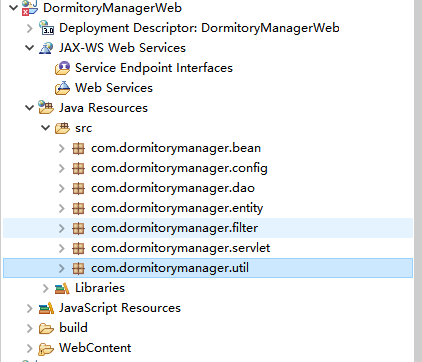
### 2.3.1系统设计目标

开发设计学生宿舍管理系统实质上是开发了一个基于JavaWeb的动态Web项目。

首先考虑系统安全性，本系统默认管理员已经将学生信息录入数据库中，管理员以及宿管都可以增加或者删除数据库中学生表的学生信息，但是学生不能自行添加注册信息，只能登陆管理自己的信息，这点是为了安全，也是比较贴近现实。通过B/S模式实现人员登陆管理，只有存在于数据库学生表中的学生，以及存在的宿管和管理员可以登录本系统，有效避免了非法人员登陆系统。此外本系统还对用户的权限进行了分级设置，用户在输入自己的姓名，密码以及验证码后，需要选择合适的身份才能登陆系统，用户级别分为学生、宿管和管理员，不同级别的用户有不同的操作权限。用户登录系统后才可以进行相应权限下的增、删、改、查操作。

### 2.3.2开发设计思想

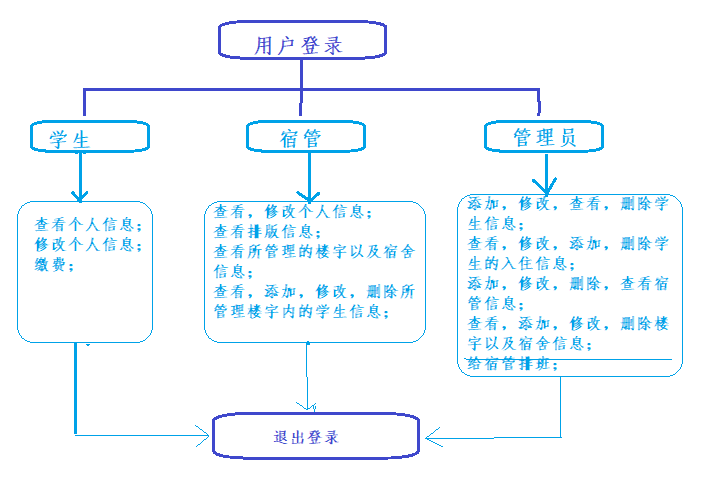
DAO层负责与与服务器的数据访问，Servlet负责web端业务逻辑。



### 2.3.3系统功能模块设计学生宿舍管理系统

**图1. 系统思维导图**

根据系统所要实现的功能进行编程设计，首先进行登录页面等前台页面的编写，然后再写内在逻辑，和涉及到数据库的操作，以上思维导图是编程的重点内容。从用户角度而言，学生宿舍管理系统，不同的用户拥有不同的权限，各自完成各自的管理功能，不同的用户看到不同的系统功能。用mySql创建后台数据库，然后利用JSP技术编写程序实现对数据库的操作，其中系统基本功能模块包括：人员的登录管理，并设置不同权限；学生的增、删、改、查功能；宿舍的增、删、改、查功能；缴费信息增、删、改、查功能；实现楼管排班功能；系统功能图如下图所示：

**图2. 程序功能设计图**

下面从三种用户的角度分别进行详细的功能设计：

1、登陆功能

已存在数据库中的用户输入自己的信息后可以进行登录，用户可以根据提示输入用ID、密码，并选择权限，点击登陆即可进入相应的管理界面。

2、学生功能

学生登录系统后权限是最小的，学生只能查看自己的姓名，性别，学号，ID，密码，缴费信息等信息，或者对自己的信息作出修改。

3、宿管功能

同样，也只有存在数据库中的宿管才能登录该系统。宿管登录系统后，权限比学生登录多很多。

（1）、学生管理

宿管可以查看其所管理宿舍的学生信息，并可以为学生分配宿舍，甚至可以修改学生的住宿信息；

（2）、宿管管理

宿管可登陆系统对自己的姓名，密码，ID等信息进行增、删、改，查；

（3）、楼宇管理

宿管可以登录系统查看其所管理的楼宇信息，比如楼宇的位置，名称，ID等信息。同时也可以对自己所管辖的楼宇进行检索。此功能是为了方便当一个宿管管辖的楼宇过多时，查看某座楼宇的信息。

（4）、宿舍管理

宿管可以登录系统查看其所管理的宿舍信息，比如宿舍的所属楼宇，宿舍号， ID，所属楼层，最大入住人数，已住人数等信息。同时也可以对自己所管辖的宿舍进行添加或者删除。也可以按要求对自己管辖的宿舍进行检索。此功能是为了方便当一个宿管管辖的宿舍过多时，查看某宿舍的信息。

（5）、住宿管理

宿管可以登录系统查看其所管理的学生宿舍信息，并能对宿舍信息进行增、删、改、查。比如添加住宿学生，调整住宿学生的宿舍信息，为学生退宿等。同时宿管还可以按照学生姓名查看学生的宿舍信息，或者按照宿舍信息查看宿舍中的学生入住信息。

4、管理员功能

管理员的权限是最大的，管理员登录系统后的功能最为全面，用到了前面所涉及的所有数据库的信息。这里在进行系统设计是将管理员数量预设为一名，预设其姓名为admin，密码也是admin。

（1）、学生管理

管理员不仅可以查看其所管理宿舍的学生信息，并可以为学生分配宿舍，甚至可以修改学生的住宿信息，他拥有最高的权限，可以向数据库添加一个新的学生信息，或者删除一个学生信息。

（2）、宿管管理

同样，管理员不仅可以对一个宿管的姓名，密码，ID等信息进行增、删、改，查。也可以向数据库中添加一个宿管的信息，或者在数据库中删除一个宿管的信息。

（3）、楼宇管理

管理员登录系统查看所有楼宇信息，包括楼宇的位置，名称，ID等信息。同时也可以按楼宇的ID或者位置对楼宇进行检索。同时可以对楼宇信息进行增、删、改、查。

（4）、宿舍管理

管理员可以登录系统查看所有楼宇的所有宿舍信息，包括宿舍的所属楼宇，宿舍号， ID，所属楼层，最大入住人数，已住人数等信息。当然可以所有宿舍进行添加或者删除。也可以按要求对宿舍进行检索。或者从数据库中添加或者删除一个宿舍的信息。

（5）、住宿管理

管理员登录系统可以查看所有学生的宿舍信息，并能对宿舍信息进行增、删、改、查。比如添加住宿学生信息，调整住宿学生的宿舍信息，为学生退宿等。同时管理员还可以按照学生姓名查看学生的宿舍信息，或者按照宿舍信息查看宿舍中的学生入住信息。不过管理员主要对住宿信息进行宏观调控，查看等操作。

（6）、系统管理

与学生和宿管登录该系统一样，管理员也能对自己的信息进行查看和修改，此外管理员还能查看当前管理员的数量，以及各个管理员的当前状态（是否可用）。

# 系统详细设计与实现

## 3.1各功能关键代码设计

学生宿舍管理系统总体上来讲是实现了不同状态（权限）下，对数据库中不同信息的增、删、改、查以及相应的前端页面设计。下面分别展示不同阶段重点部分的代码：

### 3.1.1登录页面及系统操作页面

<title>登录|学生宿舍管理系统</title>

<meta name=*"keywords"* content=*"学生宿舍管理系统"*>

</head>

<body>

<div class=*"header"* style="padding: *0*;">

<h2 style="color: *white*; width: *400px*; height: *60px*; line-height: *60px*; margin: *0 0 0 30px*; padding: *0*;">学生宿舍管理系统</h2>

</div>

<div class=*"loginWraper"*>

<div id=*"loginform"* class=*"loginBox"*>

<form id=*"form"* class=*"form form-horizontal"* method=*"post"*>

<div class=*"row cl"*>

<label class=*"form-label col-3"*>用户名：</label>

<div class=*"formControls col-8"*>

<input id=*"login-name"* name=*"name"* type=*"text"* placeholder=*"请输入用户名"* class=*"input-text size-L"*>

</div>

</div>

<div class=*"row cl"*>

<label class=*"form-label col-3"*>密&nbsp;&nbsp;码：</label>

<div class=*"formControls col-8"*>

<input id=*"login-password"* name=*"password"* type=*"password"* placeholder=*"请输入密码 "* class=*"input-text size-L"*>

</div>

</div>

<div class=*"row cl"*>

<div class=*"formControls col-8 col-offset-3"*>

<input id=*"login-vcode"* class=*"input-text size-L"* name=*"vcode"* type=*"text"* placeholder=*"请输入验证码"* style="width: *200px*;">

<img title=*"点击图片切换验证码"* id=*"vcodeImg"* src=*"CpachaServlet?method=loginCpacha&vl=4&fs=28"*></div>

</div>

<div class=*"mt-20 skin-minimal"* style="text-align: *center*;">

<div class=*"radio-box"*>

<input type=*"radio"* id=*"radio-2"* name=*"type"* checked value=*"2"* />

<label for=*"radio-1"* style="font-weight:*bold*; color:*#000066*;">学生</label>

</div>

<div class=*"radio-box"*>

<input type=*"radio"* id=*"radio-3"* name=*"type"* value=*"3"* />

<label for=*"radio-2"* style="font-weight:*bold*; color:*#000066*;">宿管</label>

</div>

<div class=*"radio-box"*>

<input type=*"radio"* id=*"radio-1"* name=*"type"* value=*"1"* />

<label for=*"radio-3"* style="font-weight:*bold*; color:*#000066*;">管理员</label>

</div>

</div>

]};

</script>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"* />

<title>学生宿舍管理系统 管理员后台</title>

<link rel=*"shortcut icon"* href=*"xuexiao.png"*/>

<link rel=*"bookmark"* href=*"xuexiao.png"*/>

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

<link rel=*"stylesheet"* type=*"text/css"* href=*"easyui/themes/default/easyui.css"* />

<link rel=*"stylesheet"* type=*"text/css"* href=*"easyui/themes/icon.css"* />

<script type=*"text/javascript"* src=*"easyui/jquery.min.js"*></script>

<script type=*"text/javascript"* src=*"easyui/jquery.easyui.min.js"*></script>

<script type=*"text/javascript"* src=*'easyui/js/outlook2.js'*> </script>

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

**var** \_menus = {"menus":[

{"menuid":"1","icon":"","menuname":"学生管理",

"menus":[

{"menuid":"11","menuname":"学生列表","icon":"icon-user-student","url":"StudentServlet?method=toStudentListView"}

]

},

<c:**if** test="${userType != 2}">

{"menuid":"2","icon":"","menuname":"宿管管理",

"menus":[

{"menuid":"21","menuname":"宿管列表","icon":"icon-user-teacher","url":"DormitoryManagerServlet?method=toDormitoryManagerListView"},

]

},

{"menuid":"3","icon":"","menuname":"楼宇管理",

"menus":[

{"menuid":"31","menuname":"楼宇列表","icon":"icon-house","url":"BuildingServlet?method=toBuildingListView"},

]

},

</c:**if**>

<c:**if** test="${userType == 1 || userType == 3}">

{"menuid":"4","icon":"","menuname":"宿舍管理",

"menus":[

{"menuid":"41","menuname":"宿舍列表","icon":"icon-house","url":"DormitoryServlet?method=toDormitoryListView"},

]

},

</c:**if**>

{"menuid":"5","icon":"","menuname":"住宿管理",

"menus":[

{"menuid":"51","menuname":"住宿列表","icon":"icon-house","url":"LiveServlet?method=toLiveListView"},

]

},

<c:**if** test="${userType == 1}">

{"menuid":"6","icon":"","menuname":"系统管理",

"menus":[

{"menuid":"61","menuname":"管理员列表","icon":"icon-set","url":"AdminServlet?method=toAdminListView"},

]

}

</c:**if**>

]};

</script>

</head>

<body class=*"easyui-layout"* style="overflow-y: *hidden*" scroll=*"no"*>

<noscript>

<div style=" position:*absolute*; z-index:*100000*; height:*2046px*;top:*0px*;left:*0px*; width:*100%*; background:*white*; text-align:*center*;">

<img src=*"images/noscript.gif"* alt=*'抱歉，请开启脚本支持！'* />

</div>

</noscript>

<div region=*"north"* split=*"true"* border=*"false"* style="overflow: *hidden*; height: *30px*;

background: *url(images/layout-browser-hd-bg.gif) #CC3333 repeat-x center 50%*;

line-height: *20px*;color: *#fff*; font-family: *Verdana, 微软雅黑,黑体*">

<span style="float:*right*; padding-right:*20px*;" class=*"head"*><span style="color:*#FFFFCC font-weight*:*bold*;font-size:*150%*">${user.name}&nbsp;</span>您好&nbsp;&nbsp;&nbsp;<a href=*"SystemServlet?method=LoginOut"* id=*"loginOut"*>安全退出</a></span>

<span style="padding-left:*10px*; font-size: *16px*; ">宿舍管理系统</span>

</div>

<div region=*"south"* split=*"true"* style="height: *30px*; background: *#D2E0F2*; ">

<div class=*"footer"*></div>

</div>

<div region=*"west"* hide=*"true"* split=*"true"* title=*"导航菜单"* style="width:*180px*;" id=*"west"*>

<div id=*"nav"* class=*"easyui-accordion"* fit=*"true"* border=*"false"*>

<!-- 导航内容 -->

</div>

</div>

<div id=*"mainPanle"* region=*"center"* style="background: *#eee*; overflow-y:*hidden*">

<div id=*"tabs"* class=*"easyui-tabs"* fit=*"true"* border=*"false"* >

<jsp:include page=*"/view/welcome.jsp"* />

</div>

</div>

</body>

</html>

### 3.1.2数据库操作基础类

数据库操作基础类， 利用泛型和反射机制来抽象数据库基本的增删该查操作。

所有新增插入操作抽象封装代码如下：

public boolean add(T t){

if(t== null)return false;

String buildSql = buildSql(CURD\_ADD);

try {

PreparedStatement prepareStatement = con.prepareStatement(buildSql);

Field[] fields = t.getClass().getDeclaredFields();

for(int i=1; i < fields.length; i++){

fields[i].setAccessible(true);

prepareStatement.setObject(i, fields[i].get(t));

}

return prepareStatement.executeUpdate() > 0;

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

抽象封装所有的分页查询列表操作代码如下：

**public** Page<T> findList(Page<T> page){

String sql = buildSql(CURD\_SELECT);

sql += buidSearchSql(page);

sql += " limit " + page.getOffset() + "," + page.getPageSize();

//System.out.println(sql);

**try** {

PreparedStatement prepareStatement = con.prepareStatement(sql);

prepareStatement = setParams(page, prepareStatement);

ResultSet executeQuery = prepareStatement.executeQuery();

List<T> conten = page.getConten();

**while**(executeQuery.next()){

T entity = t.~~newInstance~~();

Field[] declaredFields = t.getDeclaredFields();

**for**(Field field :declaredFields){

field.setAccessible(**true**);

field.set(entity, executeQuery.getObject(StringUtil.*convertToUnderLine*(field.getName())));

}

conten.add(entity);

}

page.setConten(conten);

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

page.setTotal(getTotal(page));

**return** page;

}

获取符合条件的所有记录数代码如下：

**public** **int** getTotal(Page<T> page){

String sql = buildSql(CURD\_COUNT);

sql += buidSearchSql(page);

**try** {

PreparedStatement prepareStatement = con.prepareStatement(sql);

prepareStatement = setParams(page, prepareStatement);

ResultSet executeQuery = prepareStatement.executeQuery();

**if**(executeQuery.next()){

**return** executeQuery.getInt("total");

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** 0;

}

构造查询sql语句代码如下：

**private** String buidSearchSql(Page<T> page){

String sql = "";

List<SearchProperty> searchProperties = page.getSearchProperties();

**for**(SearchProperty searchProperty : searchProperties){

**switch**(searchProperty.getOperator()){

**case** ***GT***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " > ?";

**break**;

}

**case** ***GTE***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " >= ?";

**break**;

}

**case** ***EQ***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " = ?";

**break**;

}

**case** ***LT***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " < ?";

**break**;

}

**case** ***LTE***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " <= ?";

**break**;

}

**case** ***LIKE***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " like ?";

**break**;

}

**case** ***NEQ***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " <> ?";

**break**;

}

**case** ***IN***:{

sql += " and " + StringUtil.*convertToUnderLine*(searchProperty.getKey()) + " in (" + searchProperty.getValue() + ") ";

**break**;

}

}

}

sql = sql.replaceFirst("and", "where");

System.***out***.println(sql);

**return** sql;

}

构造一般查询语句代码如下：

**private** String buildSql(**int** type) {

// **TODO** Auto-generated method stub

String sql = "";

**switch** (type) {

**case** CURD\_ADD:{

String sql1 = "insert into " + StringUtil.*convertToUnderLine*(t.getSimpleName()) + "(";

Field[] declaredFields = t.getDeclaredFields();

**for**(Field field : declaredFields){

sql1 += StringUtil.*convertToUnderLine*(field.getName()) + ",";

}

sql1 = sql1.substring(0,sql1.length()-1) + ")";

String sql2 = " values(null,";

String[] params = **new** String[declaredFields.length-1];

Arrays.*fill*(params, "?");

sql2 += StringUtils.*join*(params, ",") + ")";

sql = sql1 + sql2;

**break**;

}

**case** CURD\_SELECT:{

sql = "select \* from " + StringUtil.*convertToUnderLine*(t.getSimpleName());

**break**;

}

**case** CURD\_COUNT:{

sql = "select count(\*) as total from " + StringUtil.*convertToUnderLine*(t.getSimpleName());

**break**;

}

**case** CURD\_UPDATE:{

sql = "update " + StringUtil.*convertToUnderLine*(t.getSimpleName()) + " set ";

Field[] declaredFields = t.getDeclaredFields();

**for**(Field field : declaredFields){

**if**(!"id".equals(field.getName())){

sql += StringUtil.*convertToUnderLine*(field.getName()) + " =?,";

}

}

sql = sql.substring(0,sql.length()-1) + " where id = ?";

**break**;

}

**case** CURD\_DELETE:{

sql = "delete from "+StringUtil.*convertToUnderLine*(t.getSimpleName())+" where id in(";

**break**;

}

**default**:

**break**;

}

System.***out***.println(sql);

**return** sql;

}

关闭数据库链接代码如下：

**public** **void** closeConnection(){

**if**(con != **null**){

**try** {

con.close();

System.***out***.println(t.getSimpleName()+"Dao建立的数据库链接成功关闭!");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

### 3.1.3学生服务端部分代码

删除学生数据代码如下：

**private** **void** deleteStudent(HttpServletRequest req, HttpServletResponse resp) {

// **TODO** Auto-generated method stub

String[] ids = req.getParameterValues("ids[]");

StudentDao studentDao = **new** StudentDao();

String msg = "";

**if**(studentDao.delete(ids)){

msg = "success";

}

studentDao.closeConnection();

**try** {

resp.getWriter().write(msg);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

查看学生信息代码如下：

**private** **void** getStudentList(HttpServletRequest req, HttpServletResponse resp) {

// **TODO** Auto-generated method stub

String from = req.getParameter("from");

//如果来自下拉框查询

**if**("combox".equals(from)){

returnByCombox(req,resp);

**return**;

}

**int** pageNumber = Integer.*parseInt*(req.getParameter("page"));

**int** pageSize = Integer.*parseInt*(req.getParameter("rows"));

String name = req.getParameter("name");

**if**(StringUtil.*isEmpty*(name)){

name = "";

}

Student student = **new** Student();

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

student.setName(name);

StudentDao studentDao = **new** StudentDao();

Page<Student> page = **new** Page<Student>(pageNumber, pageSize);

page.getSearchProperties().add(**new** SearchProperty("name", "%"+name+"%", Operator.***LIKE***));

//判断当前用户是否是学生

**int** userType = Integer.*parseInt*(req.getSession().getAttribute("userType").toString());

**if**(userType == 2){

//如果是学生，则只能查看他自己的信息

Student loginedStudent = (Student)req.getSession().getAttribute("user");

page.getSearchProperties().add(**new** SearchProperty("id", loginedStudent.getId(), Operator.***EQ***));

}

Page<Student> findList = studentDao.findList(page);

ret.put("rows", findList.getConten());

ret.put("total", findList.getTotal());

studentDao.closeConnection();

resp.setCharacterEncoding("utf-8");

**try** {

resp.getWriter().write(JSONObject.*fromObject*(ret).toString());

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

添加学生信息代码如下：

**private** **void** addStudent(HttpServletRequest req, HttpServletResponse resp) **throws** IOException {

// **TODO** Auto-generated method stub

String name = req.getParameter("name");

String password = req.getParameter("password");

String sex = req.getParameter("sex");

resp.setCharacterEncoding("utf-8");

**if**(StringUtil.*isEmpty*(name)){

resp.getWriter().write("姓名不能为空!");

**return**;

}

**if**(StringUtil.*isEmpty*(password)){

resp.getWriter().write("密码不能为空!");

**return**;

}

**if**(StringUtil.*isEmpty*(sex)){

resp.getWriter().write("性别不能为空!");

**return**;

}

Student student = **new** Student();

student.setName(name);

student.setPassword(password);

student.setSex(sex);

student.setSn(StringUtil.*generateSn*("S", ""));

StudentDao studentDao = **new** StudentDao();

String msg = "添加失败!";

**if**(studentDao.add(student)){

msg = "success";

}

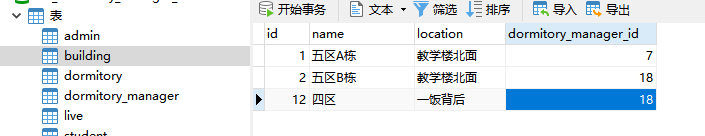
studentDao.closeConnection();

resp.getWriter().write(msg);

}

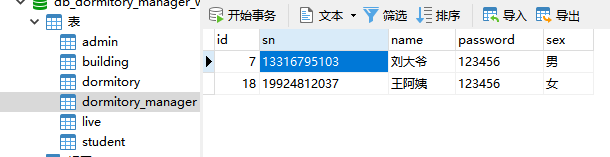
## 3.2数据库设计

### 3.2.1数据库表



### 3.2.2宿舍楼表

### 3.2.3宿舍房号表



### 3.2.4宿管表

### 3.2.5 入住表

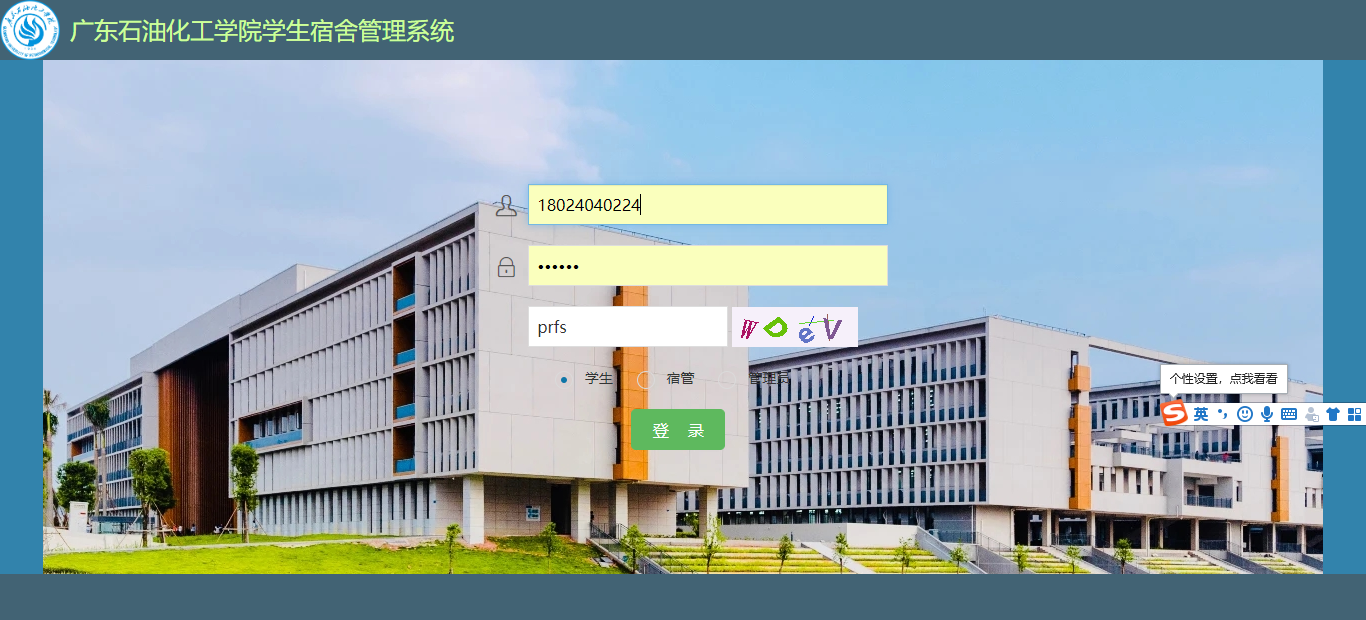
### 3.2.6 学生表

## 3.3 界面设计与功能展示

学生宿舍管理系统中，学生使用该系统的频率是最高的，下面以学生使用该系统为例介绍该系统的使用。

### 3.3.1登陆功能

输入自己的学号和密码以及验证码，选择相应的权限即可进入系统，这里应选择学生。如果是管理员和宿管输入对应账号登录。

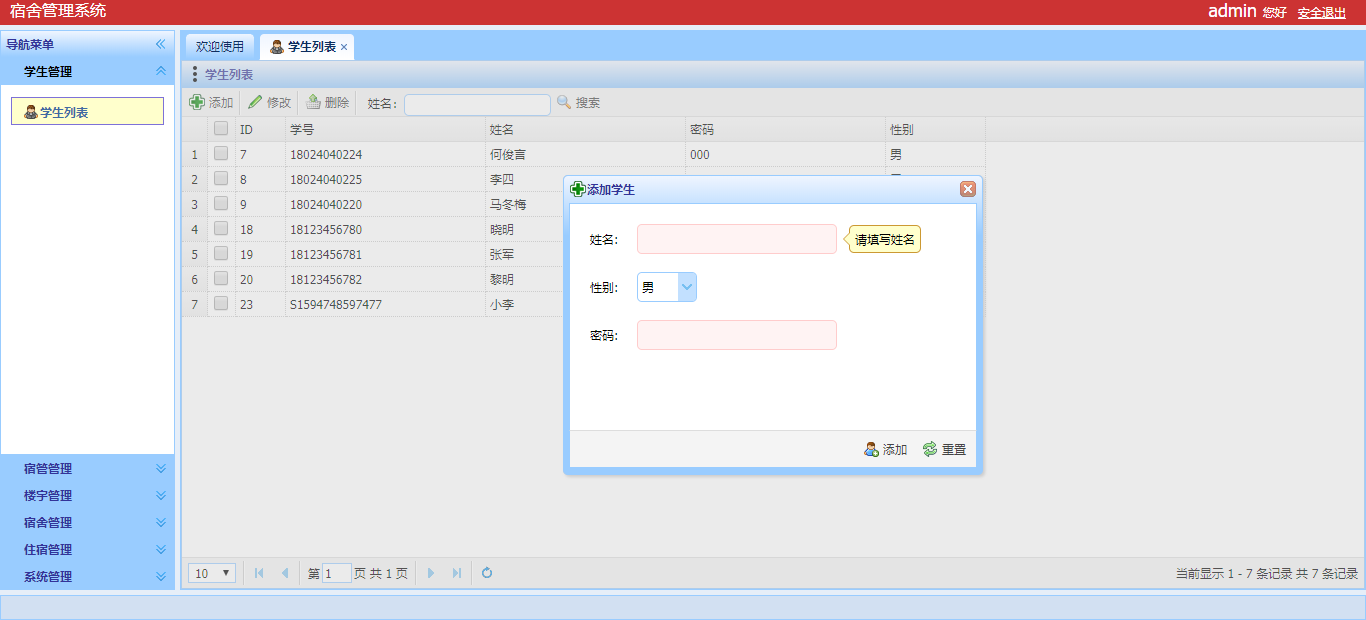


### 3.3.2欢迎页面

登录成功后，首先看到的就是欢迎页面。



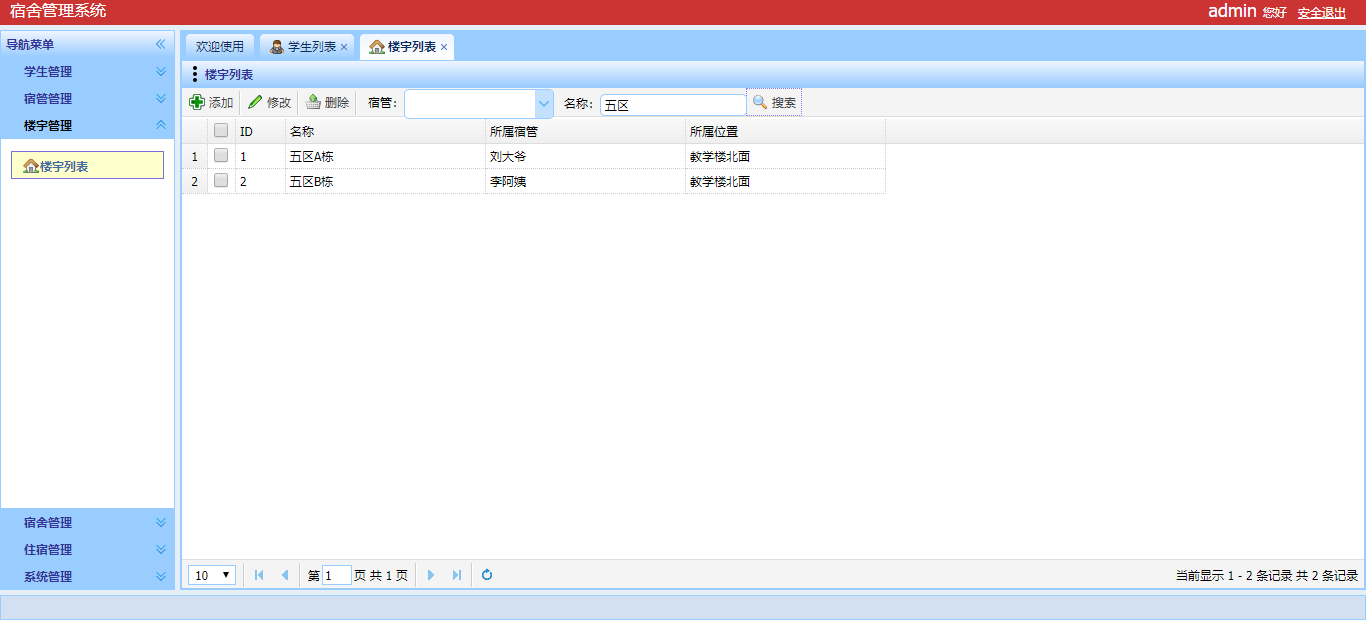
### 3.3.3添加信息功能



管理员登录系统后可以按照左边的菜单提示，查看学生信息，楼宇信息，宿舍信息，宿管信息，系统信息等内容，当进入某一功能后，可以对相应信息进行添加。如下是进入学生管理功能，可以向数据库中添加学生信息。

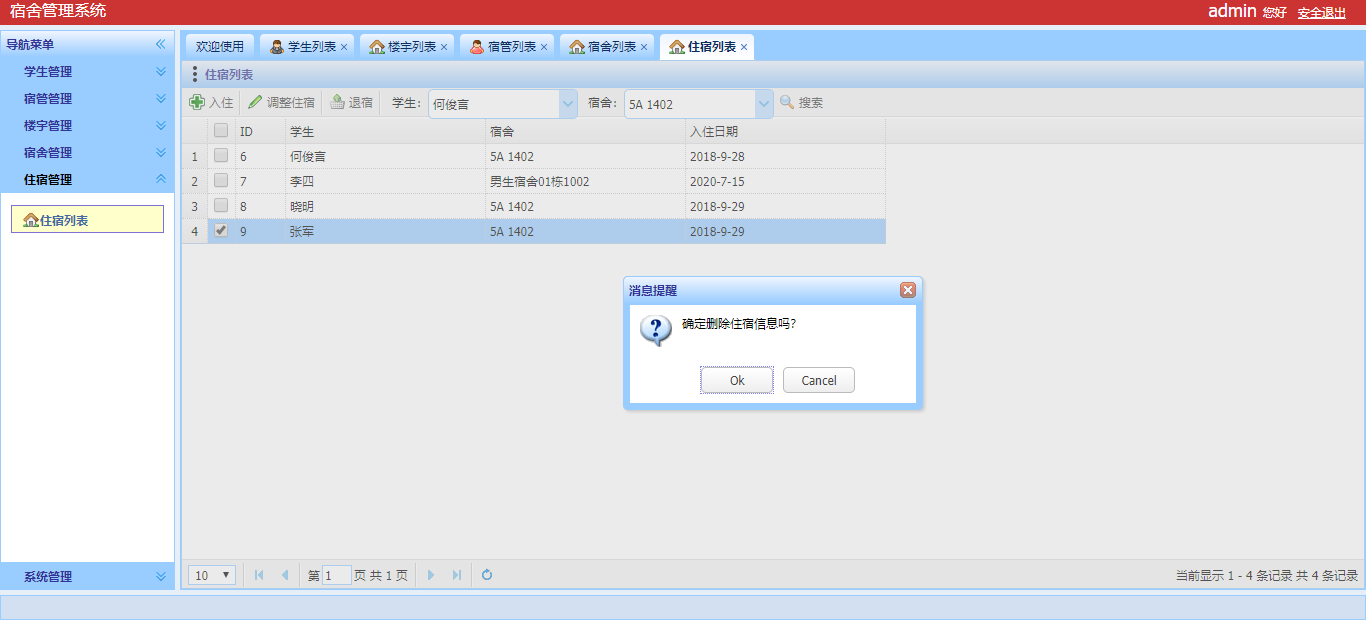
### 3.3.4查询功能

管理员登录系统后可以按照左边的菜单提示，查看学生信息，楼宇信息，宿舍信息，宿管信息，系统信息等内容，当进入某一功能后，都有相应的查询功能，在搜索框前面输入检索信息就能看到相应的符合要求的信息。



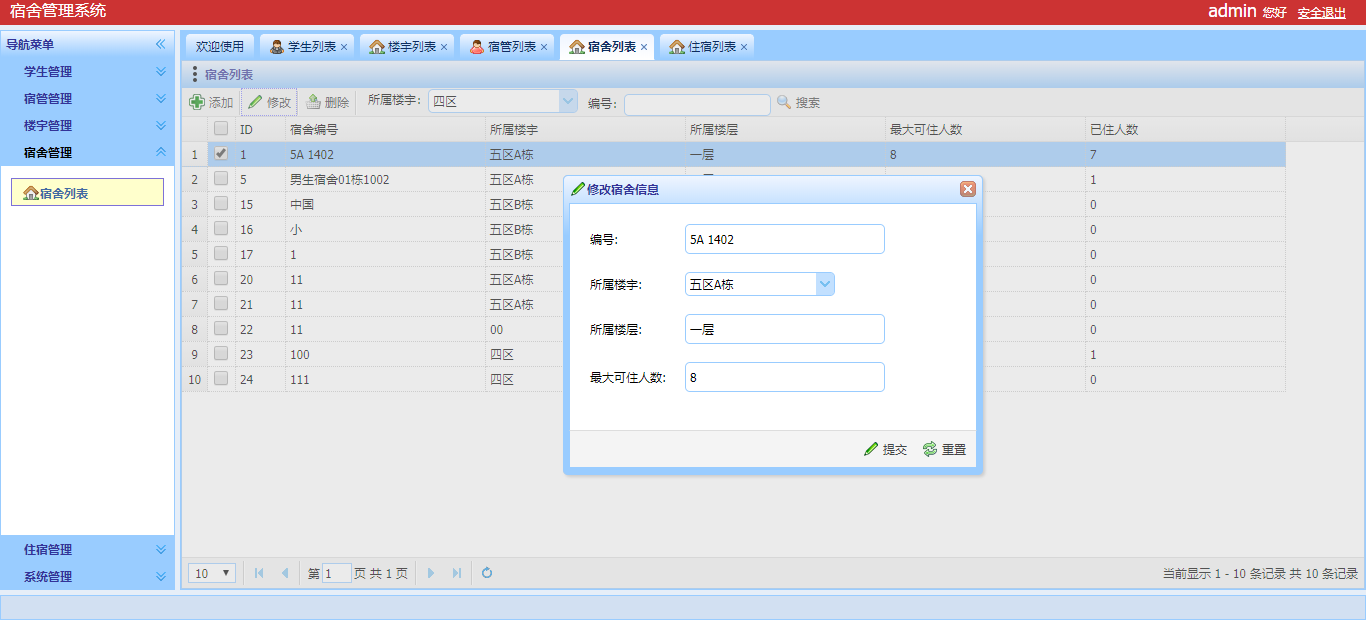
### 3.3.5删除功能

管理员登录系统后可以按照左边的菜单提示，查看学生信息，楼宇信息，宿舍信息，宿管信息，系统信息等内容，当进入某一功能后，都有相应的删除操作，操作方法一样的，下面以管理员删除住宿信息为例，选中一个或者多个学生，点击删除键，就会弹出如下确认删除的页面：



### 3.3.6修改功能

管理员登录系统后可以按照左边的菜单提示，查看学生信息，楼宇信息，宿舍信息，宿管信息，系统信息等内容，当进入某一功能后，都有相应的修改操作，操作方法一样的，下面以管理员修改宿舍信息为例，选中一个宿舍，点击修改键，就会弹出如下的重置页面：



# 总结分析

在一学期的Java Web课程之后，我初步了解了Web。 认为它是进行客户端与服务器已经数据库之间交互的体系，主要用来进行动态网站的设计开发。而在本次开发设计了宿舍管理系统之后我对java web 有了更进一步的理解。这次课程设计最重要的就是上机操作，我自己编写程序，在Eclipse上运行，首先进行了tomcat的配置，以及数据库的连接等准备工作。刚开始出现错误，比如数据库连接不成功、后台接收不到页面发来的信息以及各种逻辑错误，经过分析改正后，终于能够运行了，就觉得特别激动。课程设计是就应该是程序编写、程序调试、撰写报告结合为—体的过程。在这个过程中，不仅锻炼了我缉密的思维和坚持不解的毅力，也锻练了我写代码的能力。

在这一周的课程设计中，我遇到了不少问题，发现程序设计中我遇到问题并解决问题的过程，使得我独自探索并解决问题的能力了有了一个提高，这有利千我以后的学习。同时这整个课设过程，也使我对程序编写的整个过程有了一个统筹全局的思想，因为课设的需求分析、程序编写、程序调试、撰写报告这些过程是环环相扣的，绝对不可能独立进行。下面是我的一些收获：

1、登录，这里涉及到了很多。登录：不能为空，数据库连接，sql查询，cookie，sql防注入，用户名格式（我没有做多种方式登录，只做了用户名登录，没email登录等等），ajax，页面跳转，session，验证码等等。

2、模糊搜索（我直接用了sql语句的模糊搜索），分类功能菜单导航页面，功能表展示页面，修改、添加信息等都是数据的增、删、改、查。

3、网站的组装，比如：include ‘commontop.php’。例如：你每个二级页面都包含了顶部导航和菜单，那么你没个页面都再写个导航和菜单吗？显然不可能，那么自然而然的想到要将这些部分封装起来，写到一个独立的php里面，要用的时候就include。

4、Session的使用：在使用session前要初始化先，在php文件顶部session\_start();还有session的注销等等。

5、页面间的数据传递：post，get，session，cookie，ajax等。submit，button+函数，href，location.href等等。

6、利用泛型和反射机制实现添加操作的抽象封装.

7、登录后的主页设计及拦截器功能实现。

此外，一个小细节也非常重要，就是在设计的时候一定要把参数什么的定义好，否则真的很乱，特别是命名规则。这点在今后的开发中我会特别注意。

**参考文献**

[1]郭克华主编的《Java Web》 2016.1

[2]李博，徐云杰，翁丹。 基于Java的汽车企业产能管理系统设计[J]. 汽车实用技术，2015,01:151-153+156.

[3]李养胜，李俊。 基于JAVA的在线考试系统的设计与实现[J]. 物联网技术，2015,01:107-108.

[4]凌方。 基于Java的图书管理系统的设计与实现[J]. 计算机光盘软件与应用，2014,21:290+292.

[5]刘建。 基于Java和EJB的信息管理系统开发框架[J]. 计算机光盘软件与应用，2014,22:167+169.

[6]施阳，张海燕，戴德伟。 基于JavaEE的毕业设计管理系统设计与实现[J]. 软件导刊，2015,02:86-88.

[7]莫前玲。 基于Java语言的手机软件开发技术分析[J]. 计算机光盘软件与应用，2014,24:295-296.

[8]李佳，毛岽博。 基于Java局域网内实时通信系统设计[J]. 计算机光盘软件与应用，2014,24:283-284.

[9]崔佳佳。 基于JAVA的测试自动化设计应用[J]. 电子技术与软件工程，2015,05:253.