****Springboot学生宿舍管理系统****

****摘  要****

科技进步的飞速发展引起人们日常生活的巨大变化，电子信息技术的飞速发展使得电子信息技术的各个领域的应用水平得到普及和应用。信息时代的到来已成为不可阻挡的时尚潮流，人类发展的历史正进入一个新时代。在现实运用中，应用软件的工作规则和开发步骤，采用Springboot框架建设学生宿舍管理系统。

本设计主要实现集人性化、高效率、便捷等优点于一身的学生宿舍管理系统，完成首页、用户管理（管理员）更多管理（学生、宿舍信息、更换寝室、体温上报、维修申请管理、物品借用、宿舍缴费管理）等功能模块。系统通过浏览器与服务器进行通信，实现数据的交互与变更。本系统通过科学的管理方式、便捷的服务提高了工作效率，减少了数据存储上的错误和遗漏。学生宿舍管理系统开发是采用java语言，基于[MVVM](https://so.csdn.net/so/search?q=MVVM&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_blank)模式进行开发，采取MySQL作为后台数据的主要存储单元，采用Springboot框架实现了本系统的全部功能。

关键词 ：宿舍管理；数据库；Springboot技术

****Springboot student dormitory management system****

****Abstract****

  The rapid development of scientific and technological progress has caused great changes in people's daily life. The rapid development of electronic information technology has popularized and applied the application level of electronic information technology in various fields. The advent of the information age has become an irresistible fashion trend, and the history of human development is entering a new era. In practical application, according to the working rules and development steps of the application software, the springboot framework is used to build the student dormitory management system.

This design mainly realizes the student dormitory management system with the advantages of humanization, high efficiency and convenience, and completes the functional modules such as home page, user management (administrator) and more management (students, dormitory information, dormitory replacement, body temperature reporting, maintenance application management, article borrowing, dormitory payment management). The system communicates with the server through the browser to realize the interaction and change of data. Through scientific management and convenient service, the system improves work efficiency and reduces errors and omissions in data storage. The development of student dormitory management system adopts Java language and MVVM mode. MySQL is used as the main storage unit of background data, and the springboot framework is used to realize all the functions of the system.

****Key words****:Dormitory management; database Springboot Technology

****目    录****

[1 概述](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317764" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[1.1课题背景及意义](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317765" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[1.2 研究现状](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317766" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[1.3 主要工作内容](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317767" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[2 系统开发环境](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317768" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[2.1 java简介](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317769" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[2.2 Mysql数据库](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317770" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[2.3 B/S结构](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317771" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[2.4 Spring boot框架](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317771" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3 系统分析 6](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317772" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.1 可行性分析 6](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317773" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.1.1 技术可行性 6](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317774" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.1.2操作可行性 6](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317775" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.1.3 经济可行性 7](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317776" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.1.4 法律可行性 7](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317777" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2系统流程分析 7](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317778" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.1系统开发流程 7](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317779" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.2 用户登录流程](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317780" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.3 系统操作流程](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317781" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.4 添加信息流程](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317782" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.5 修改信息流程](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317783" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.2.6 删除信息流程](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317784" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.3系统用例分析](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317785" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.3.1管理员用例图](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317786" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[3.3.2用户前台用例图](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317787" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4 系统设计](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317788" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4.1 系统概述](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317789" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4.2 系统结构设计](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317790" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4.3数据库设计 1](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317791" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)4

[4.3.1 数据库设计原则](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317792" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4.3.2 数据库实体](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317793" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[4.3.3 数据库表设计](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317794" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[5系统统详细设计 2](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317795" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)5

[5.1用户前台功能模块 2](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317796" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)5

[5.2管理员功能模块 2](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317797" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)8

[6系统测试 3](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317800" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)3

[6.1系统测试的意义](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317801" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[6.2 测试方法 3](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317802" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)4

[6.3测试分析](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317803" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)

[结    论 3](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317804" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)5

[致    谢 3](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317805" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)6

[参考文献 3](https://blog.csdn.net/WeiXin_DZbishe/article/details/126707106" \l "_Toc55317806" \t "https://blog.csdn.net/WeiXin_DZbishe/article/details/_self)7

## ****1 概述****

### ****1.1课题背景及意义****

对于学校宿舍处而言，最主要的是管理宿舍的信息。如果使用一般的方法来管理，会比较繁琐，管理起来也很有可能出错。为了方便宿舍人员的管理工作，提高工作效率，同时为了更好地为学生提供服务，有必要开发学生宿舍管理系统，使学校的宿舍管理走上信息化之路，克服人为的种种弊端。

学生宿舍管理系统的开发不仅可以减少人力、物力和财力资源的浪费，更重要的是有助于提高宿舍管理的效率。宿舍管理人员管理学生信息、管理宿舍信息时是一项复杂的组织工作，这种复杂性不仅仅指学生学籍变更快，变更人数众多，更突出地表现在宿舍管理主要对象（即学生）的数据量大，管理起来带来不便，所以开发一个实用、高效的宿舍管理信息系统是很有必要的。

### ****1.2 研究现状****

目前我国各行各业的信息化建设开展得如火如荼，这种信息化的发展已经影响到了各学校。学校在对学生宿舍进行管理时，学生宿舍的管理一直都是很麻烦的事情，不仅仅管理起来不易修改和更新，而且学生需要了解本宿舍的信息时，也需要去学校宿舍处找专门人员。由于这些操作的繁琐性，学生宿舍管理系统的开发成为各学校必须的事情，特别是各个高校。

然而目前许多学生宿舍管理系统在使用时，效率低下容易出错，学生、教师资料不易整理，大量丢失，这些无疑都已成为管理学生宿舍的障碍。这就要求学校能够建立高效的学生宿舍管理系统，对学生的宿舍、管理等操作及流程进行规范化管理，简化业务流程，提高工作效率并防止中间的漏洞；迅速、准确地捕捉用户要求，并加以高效回应。同时需要不断完善系统，增加模块，更好的满足用户需求，简化宿舍人员的管理工作，尽量做到一切信息化。

### ****1.3 主要工作内容****

学生宿舍管理系统的开发及实现，所需要的工作内容：

（1）首先是确定选题，确定好所要做的系统，并对系统的背景及现在面临的一些问题等进行系统的初步确认。

（2）系统确认完成后，结合系统开发的需求进行确认系统开发所使用的技术，学生宿舍管理系统的开发使用Springboot框架，数据库进行平台的搭建开发，确认好使用的技术进行技术分析，所使用的技术是否可以完成系统的实现。

（3）确定好系统使用的技术，进行在线确认系统所划分的用户角色，并且根据用户角色划分确定所要设计的功能模块，对学生宿舍管理系统的设计主要划分别为管理员和用户角色，并所使用的功能模块也相应不同，但是系统的数据库实现的内容是交互的，用户可以随时根据自己的需求进行课程信息，对于系统工作人员可以根据自己的分管内容进行在线信息的处理及操作，管理员获取到所有用户的详细数据信息，并根据需求进行第一时间处理解决。

（4）系统的功能模块确认完成后进行程序及界面的设计，设计完成后，并且通过测试来判断程序是否完善，对于系统测试，需要不同的用户进行不同的内容编辑及提交，及使用不同的测试方式找出程序中存在的漏洞，并对程序出现的漏洞问题进行在线解决处理，如果测试系统没有任何问题时，可以将系统上传进行正式操作使用。

## ****2 系统开发环境****

### ****2.1 java简介****

Java语言是在二十世纪末由Sun公司发布的，而且公开源代码，这一优点吸引了许多世界各地优秀的编程爱好者，也使得他们开发出当时一款又一款经典好玩的小游戏。Java语言是纯面向对象语言之一，从发布初期到现今，可以说有将近20多年的历史，已发展成为人类计算机编程语言发展史上的一个深远影响。

Java语言具有非常多种的特性：（1）跨平台的无关性；（2）面向对象； （3）安全性得以保障；（4）支持多个任务；（5）多种编写方式，代码编写简单。对比其他的低级语言、高级语言，Java语言具有明显的显著优势以及未来开阔的前景，可以广泛的应用在个人笔记本电脑、大数据、大型游戏等等。

首先，Java语言具有面向对象的特性，并且易于理解。关于对象，其实可以理解成每一种事物都是一种对象，包括我们人类自身都是一种对象。利用面向对象语言的基本特征来解决软件开发中的实际问题，为有效软件开发提供了技术支持。

其次，Java 语言具有很好的跨平台无关性。所编写出来的应用程序是Java语言编写的，那么就无需再使用编译器来修改程序代码，可以直接在任何计算机系统中运行，Windows系统可以运行，在Linux系统中也可以，也就是经过一次编译，可以到处运行，所以Java语言具有卓越的可移植性，可以很好的跨平台实现。

### ****2.2 Mysql数据库****

Mysql是一个多用户、多线程的服务器，采用SQL的数据库，数据库管理系统是基于SQL的用户以及服务器模式的关系，它的优点有强大的功能、操作简单、管理方便、可靠安全、运行较快、多线程、跨平台性、完全网络化、稳定性等，非常适合Web站点或者其他应用软件，在数据库后端的开发。此外，利用许多语言，会员可以编写和访问Mysql数据库的程序。Mysql数据库也是开放源代码的，开发者越来越喜欢使用Mysql关系数据库，应用范围也被推而广之。这是由于速度快和易用性， Web站点或应用软件的数据库后端的开发也都在使用它。

mysql 数据库它有很多的优点，例如它在操作上能够让人通俗易懂、功能强大、信息储存量高等优点。所以被人们广泛应用，对于mysql数据库来说它一般主要是对数据进行编码和查询，而且在很多的设计当中都应用到了该数据库，在此过程当中我们可以对常规的数据进行查询和组合，所以我们在进行使用mysql数据库的时候只要对编写一小段的数据就能实现相应的功能。数据库，就是数据存储的储藏室，只不过数据是存储在计算机上的，而不是现实中的储藏室，数据的存放是按固定格式，而不是无序的，则定义就是 ：长期有固定格式，可以共享的存储在计算机存储器上。数据库管理主要包括数据表的建立，数据存储、修改和增加数据，为了使数据库系统能够正常运行，相关人员进行的管理工作。数据表的建立，可以对数据表中的数据进行调整，数据的重新组合及重新构造，保证数据的安全性。

### ****2.3 B/S结构****

BROWSER/SERVER程序架构方式是使用电脑中安装的各种浏览器来进行访问和使用的，相比C/S的程序结构不需要进行程序的安装就可以直接使用。BROWSER/SERVER架构的运行方式是在远程的服务器上进行安装一个，然后在任何接入互联网的电脑上访问和使用。BROWSER/SERVER架构的开发方式给用户带来了极大的便利，在任何时间和地点都可以使用开发的程序系统。

在B/S的结构中，用户可以在任何可以上网的地方访问和使用系统网站的功能，没有地域和时间等方面的限制，B/S结构是把程序完整放置到计算机网络的服务器上，通过计算机互联网给用户提供远程的网络服务。在三层体系结构的B/S系统中，通过浏览器，会员可以对很多服务器发出请求， B/S系统会很大程度的降低工作量，用户只需要安装运行较少的用户端即可，大量的工作将由服务器承担，另外，服务器也完成包括访问数据库，执行应用程序的工作等等。

B/S结构主要有三层，分别为数据层、控制逻辑层和视图层。用户通过视图层，让控制层调用数据层的数据，从而达到整个访问过程。三层相互独立，维护方便，使用安全，三层有互相调用，提高效率。

### **2.4 Spring boot框架**

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \o "Java平台)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \o "EJB)（EnterpriseJavaBeans）模型的补充，甚至是替补。Spring框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的Web框架方便开发等等。Spring框架具有控制反转（IOC）特性，IOC旨在方便项目维护和测试，它提供了一种通过Java的反射机制对Java对象进行统一的配置和管理的方法。Spring框架利用容器管理对象的生命周期，容器可以通过扫描XML文件或类上特定Java注解来配置对象，开发者可以通过依赖查找或依赖注入来获得对象。Spring框架具有[面向切面编程](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%88%87%E9%9D%A2%E7%BC%96%E7%A8%8B/6016335" \o "面向切面编程)（AOP）框架，SpringAOP框架基于代理模式，同时运行时可配置；AOP框架主要针对模块之间的交叉关注点进行模块化。Spring框架的AOP框架仅提供基本的AOP特性，虽无法与AspectJ框架相比，但通过与AspectJ的集成，也可以满足基本需求。Spring框架下的事务管理、远程访问等功能均可以通过使用SpringAOP技术实现。Spring的事务管理框架为Java平台带来了一种抽象机制，使本地和全局事务以及嵌套事务能够与保存点一起工作，并且几乎可以在Java平台的任何环境中工作。Spring集成多种事务模板，系统可以通过事务模板、XML或Java注解进行事务配置，并且事务框架集成了消息传递和缓存等功能。Spring的数据访问框架解决了开发人员在应用程序中使用数据库时遇到的常见困难。它不仅对Java:JDBC、iBATS/MyBATIs、Hibernate、Java数据对象（JDO）、ApacheOJB和ApacheCayne等所有流行的数据访问框架中提供支持，同时还可以与Spring的事务管理一起使用，为数据访问提供了灵活的抽象。Spring框架最初是没有打算构建一个自己的Web[MVC框架](https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/9241230" \o "MVC框架)，其开发人员在开发过程中认为现有的StrutsWeb框架的呈现层和请求处理层之间以及请求处理层和模型之间的分离不够，于是创建了SpringMVC。

## ****3 系统分析****

所谓系统分析就是，需求人员通过与用户的沟通，所获取的信息，然后把这些信息通过需求说明书的方式展示给用户和开发人员。在软件功能发展的历史长河中，很长时间，特别是最开始的时候，需求分析的重要性并不被人们所认同，例如当时美国IBM公司为英国电信公司开发一套信息管理系统，在需求不明确的情况下开始开发，最初的工期为一年，由于需求获取不清晰导致工期推迟了半年多，造成巨大损失。我们很多软件公司也存在这种情况，边需求，边开发，甚至与用户没有沟通清楚的情况下，直接照搬同类型的项目进行更改，导致到系统验收的时候，重新更改，造成了人力、物力的极大浪费。而导致这一切后果的原因就是需求获取不及时、不清楚、不全面。

### ****3.1 可行性分析****

学生宿舍管理系统主要目标是实现首页、站点管理、（轮播图、公告栏、）用户管理（管理员）更多管理（更换寝室管理、体温上报、维修申请、缺勤信息、物品借用、宿舍缴费）系统管理的相关信息管理服务。在确定了目标后，我们从以下四方面对能否实现本系统目标进行可行性分析。

#### **3.1.1 技术可行性**

学生宿舍管理系统主要采用Springboot框架，基于B/S结构，Mysql数据库，对于应用程序的开发要求具备完整功能，使用简单的特点，并建立一个数据完整安全稳定的数据库。学生宿舍管理系统的开发技术具有很高可行性，且开发人员掌握了一定的开发技术，所以系统的开发具有可行性。

#### **3.1.2操作可行性**

学生宿舍管理系统的登录界面简单易于操作，采用常见的界面窗口来登录界面，通过电脑进行访问操作，会员只要平时使用过电脑都能进行访问操作。此系统的开发采用java语言开发，基于B/S结构，这些开发环境使系统更加完善。本系统具有易操作、易管理、交互性好的特点，在操作上是非常简单的。因此本系统可以进行开发。

#### **3.1.3 经济可行性**

学生宿舍管理系统是基于B/S模式，采用Mysql数据库储存数据，所要求的硬件和软件环境，市场上都很容易购买，程序开发主要是管理系统的开发和维护。所以程序在开发人力、财力上要求不高，而且此系统不是很复杂，开发周期短，在经济方面具有较高的可行性。

#### **3.1.4 法律可行性**

此学生宿舍管理系统是自己设计的管理系统，具有很大的实际意义。因为无论是软件还是数据库，采用的都是开源代码，因此这个系统的开发和设计，并不存在侵权等问题，在法律上完全具有可行性。

综上所述，学生宿舍管理系统在技术、经济、操作和法律上都具有很高的可行性，开发此程序是可行的。

### ****3.2系统流程分析****

#### **3.2.1系统开发流程**

学生宿舍管理系统开发时，首先进行需求分析，进而对系统进行总体的设计规划，设计系统功能模块，数据库的选择等，本系统的开发流程如图3-1所示

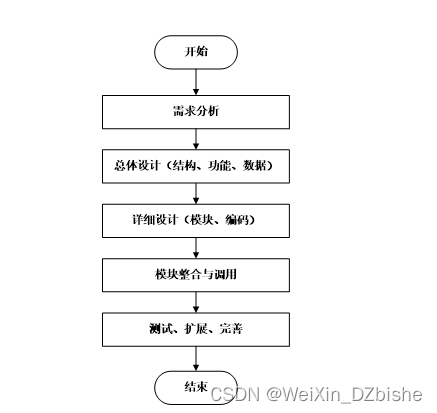


图3-1系统开发流程图

#### **3.2.2 用户登录流程**

为了保证系统的安全性，要使用本系统对系统信息进行管理，必须先登陆到系统中。如图3-2所示。

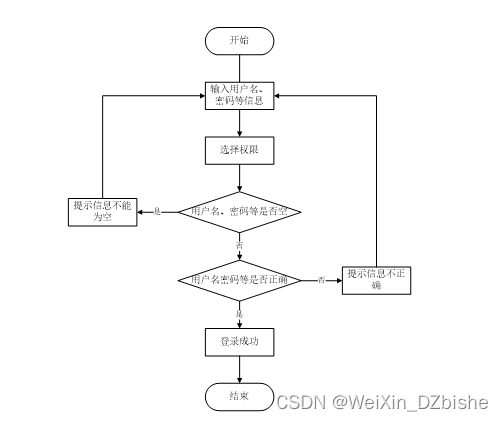


图3-2 登录流程图

#### **3.2.3 系统操作流程**

用户打开并进入系统后，会先显示登录界面，输入正确的用户名和密码，系统自动检测信息，若信息无误，则用户会进入系统功能界面，进行操作，否则会提示错误无法登录，操作流程如图3-3所示。

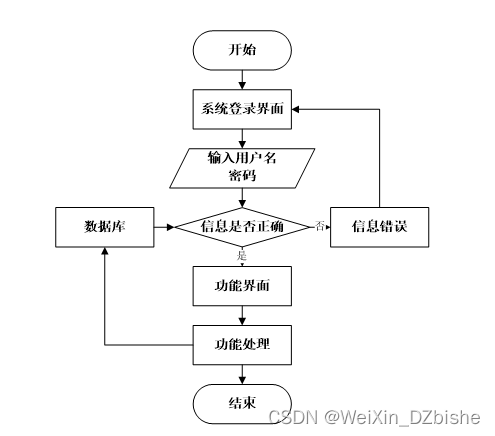


图3-3 系统操作流程图

#### **3.2.4 添加信息流程**

管理员可以对书籍类型管理进行信息的添加，用户可以对自己权限内的信息进行添加，输入信息后，系统会自行验证输入的信息和数据，若信息正确，会将其添加到数据库内，若信息有误，则会提示重新输入信息，添加信息流程如图3-4所示。

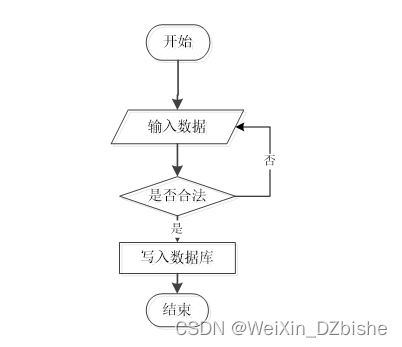


图3-4 添加信息流程图

#### **3.2.5 修改信息流程**

管理员可以对书籍信息管理进行的修改，用户可以对自己权限内的信息进行修改，首先进入修改信息界面，输入修改信息数据，系统进行数据的判断验证，修改信息合法则修改成功，信息更新至数据库，信息不合法则修改失败，重新输入。修改信息流程图如图3-5所示。

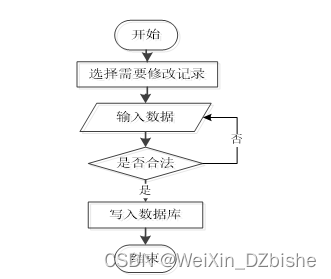


图3-5 修改信息流程图

#### **3.2.6 删除信息流程**

管理员可以对留言板管理进行信息的删除，对要删除的信息进行选中后，点击删除按钮，系统会询问是否确定，若点击确定，则系统会删除掉选中的信息，并在数据库内对信息进行删除，删除信息流程图如图3-6所示。

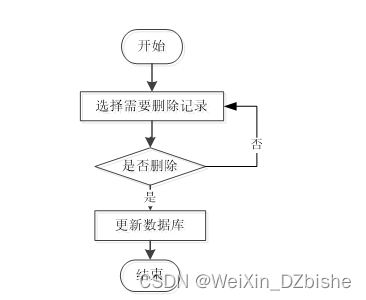


图3-6 删除信息流程图

### ****3.3系统用例分析****

#### **3.3.1管理员用例图**

系统中的核心用户是系统管理员，管理员登录后，通过管理员菜单来管理后台系统。主要功能有：首页、用户管理、宿舍信息管理、换寝申请管理、物品借用管理、晚归信息管理、体温上报管理、维修申请管理等功能。管理员用例如图3-7所示。

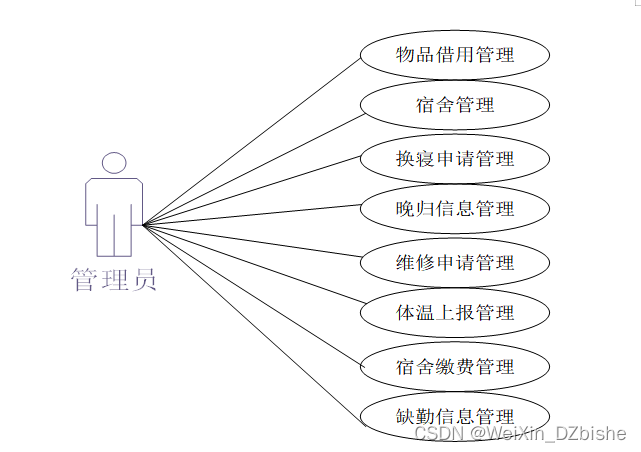


图3-7 管理员用例图

#### **3.3.2学生前台用例图**

学生前台进入系统可以进行首页、宿更换寝室。缺勤信息、维修申请、体温上报、宿舍缴费、物品借用等。用户前台用例如图3-8所示。

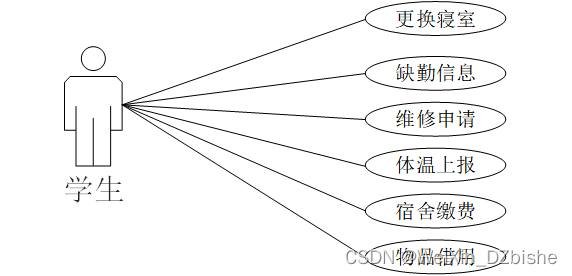


图3-8 学生前台用例图

## ****4 系统设计****

### ****4.1 系统概述****

进过系统的分析后，就开始记性系统的设计，系统设计包含总体设计和详细设计。总体设计只是一个大体的设计，经过了总体设计，我们能够划分出系统的一些东西，例如文件、文档、数据等。而且我们通过总体设计，大致可以划分出了程序的模块，以及功能。但是只是一个初步的分类，并没有真正的实现。

整体设计，只是一个初步设计，而且，对于一个项目，我们可以进行多个整体设计，通过对比，包括性能的对比、成本的对比、效益的对比，来最终确定一个最优的设计方案，选择优秀的整体设计可以降低开发成本，增加公司效益，从这一点来讲，整体设计还是非常重要的。

学生宿舍管理系统工作原理图如图4-1所示：

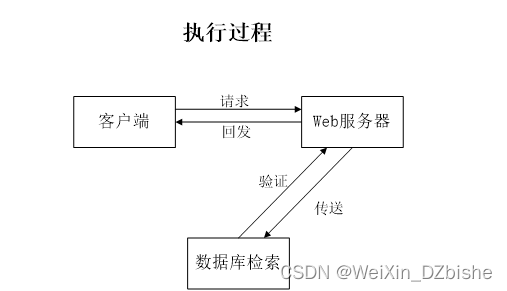


图4-1 系统工作原理图

### ****4.2 系统结构设计****

系统架构图属于系统设计阶段，系统架构图只是这个阶段一个产物，系统的总体架构决定了整个系统的模式，是系统的基础。学生宿舍管理系统的整体结构设计如图4-2所示。

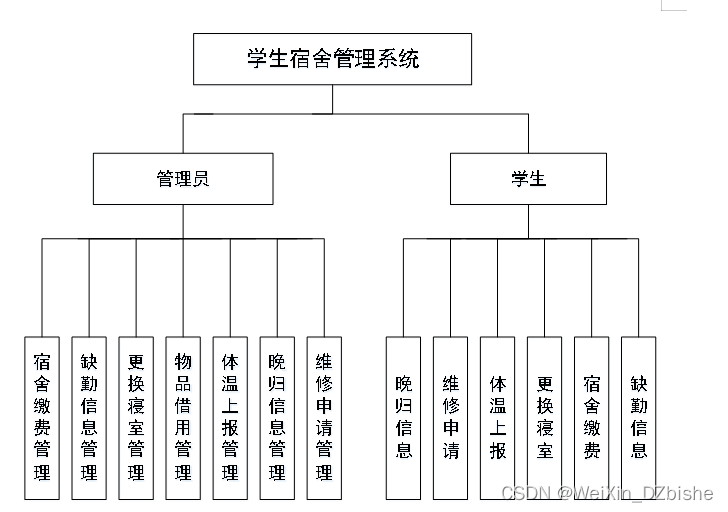


图4-2 系统结构图

### ****4.3数据库设计****

数据库是计算机信息系统的基础。目前，电脑系统的关键与核心部分就是数据库。数据库开发的优劣对整个系统的质量和速度有着直接影响。

#### ****4.3.1 数据库设计原则****

数据库的概念结构设计采用实体—联系（E-R）模型设计方法。E-R模型法的组成元素有：实体、属性、联系，E-R模型用E-R图表示，是提示用户工作环境中所涉及的事物，属性则是对实体特性的描述。在系统设计当中数据库起着决定性的因素。下面设计出这几个关键实体的实体—关系图。

#### ****4.3.2 数据库实体****

数据模型中的实体（Entity），也称为实例，对应现实世界中可区别于其他对象的“事件”或“事物”。

本系统的E-R图如下图所示：

1、缺勤信息实体图如图4-3所示：

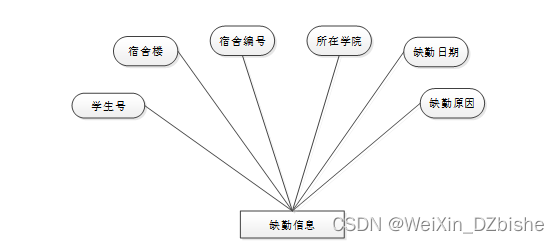


图4-3缺勤信息实体图

2、用户信息实体图如图4-4所示：

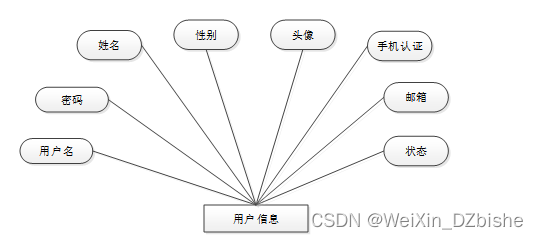


图4-4用户信息实体图

3、维修申请实体图如图4-5所示：

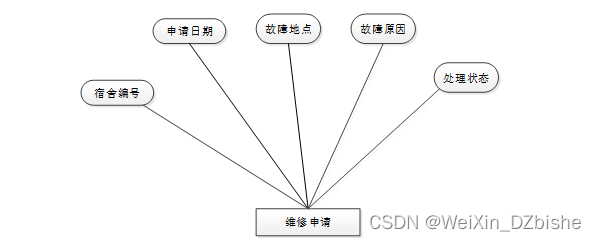


图4-5维修申请实体图

#### ****4.3.3 数据库表设计****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| student\_user\_id | int | 11 | 是 | 是 | 学生用户ID |
| student\_name | varchar | 64 | 否 | 否 | 学生姓名 |
| student\_number | varchar | 64 | 是 | 否 | 学生学号 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| college | varchar | 64 | 否 | 否 | 所在学院 |
| enrollment\_year\_ | varchar | 64 | 否 | 否 | 入学年份 |
| examine\_state | varchar | 16 | 是 | 否 | 审核状态 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| user\_id | int | 11 | 是 | 否 | 用户ID |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| temperature\_reporting\_id | int | 11 | 是 | 是 | 体温上报ID |
| student\_number | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | int | 11 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| college | varchar | 64 | 否 | 否 | 所在学院 |
| class\_information | varchar | 64 | 否 | 否 | 班级信息 |
| student\_telephone | varchar | 64 | 否 | 否 | 学生电话 |
| abnormal\_date | date | 0 | 否 | 否 | 异常日期 |
| temperature\_at\_that\_time | varchar | 64 | 否 | 否 | 当时体温 |
| physical\_condition | varchar | 64 | 否 | 否 | 身体状况 |
| processing\_status | varchar | 64 | 否 | 否 | 处理状态 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| student\_information\_id | int | 11 | 是 | 是 | 学生信息ID |
| student\_number | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | int | 11 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| college | varchar | 64 | 否 | 否 | 所在学院 |
| class\_information | varchar | 64 | 否 | 否 | 班级信息 |
| province\_of\_origin | varchar | 64 | 否 | 否 | 生源地省份 |
| student\_telephone | varchar | 64 | 否 | 否 | 学生电话 |
| dormitory\_building\_id | varchar | 64 | 否 | 否 | 宿舍楼ID |
| dormitory\_no | varchar | 64 | 否 | 否 | 宿舍编号 |
| dormitory\_electricity\_balance | varchar | 64 | 否 | 否 | 宿舍电费余额 |
| enrollment\_year\_ | varchar | 64 | 否 | 否 | 入学年份 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| maintenance\_application\_id | int | 11 | 是 | 是 | 维修申请ID |
| student\_number | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | int | 11 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| college | varchar | 64 | 否 | 否 | 所在学院 |
| class\_information | varchar | 64 | 否 | 否 | 班级信息 |
| student\_telephone | varchar | 64 | 否 | 否 | 学生电话 |
| dormitory\_building\_id | varchar | 64 | 否 | 否 | 宿舍楼ID |
| dormitory\_no | varchar | 64 | 否 | 否 | 宿舍编号 |
| application\_date | date | 0 | 否 | 否 | 申请日期 |
| fault\_location | varchar | 64 | 否 | 否 | 故障地点 |
| fault\_photos | varchar | 255 | 否 | 否 | 故障照片 |
| processing\_status | varchar | 64 | 否 | 否 | 处理状态 |
| cause\_of\_failure | longtext | 0 | 否 | 否 | 故障原因 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| late\_return\_id | int | 11 | 是 | 是 | 晚归信息ID |
| student\_number | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | int | 11 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| dormitory\_building\_id | varchar | 64 | 否 | 否 | 宿舍楼ID |
| dormitory\_no | varchar | 64 | 否 | 否 | 宿舍编号 |
| late\_return\_date | date | 0 | 否 | 否 | 晚归日期 |
| reasons\_for\_late\_return | longtext | 0 | 否 | 否 | 晚归原因 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 不是null | 主键 | 注释 |
| change\_dormitory\_id | int | 11 | 是 | 是 | 更换寝室ID |
| student\_number | varchar | 64 | 否 | 否 | 学生学号 |
| student\_name | int | 11 | 否 | 否 | 学生姓名 |
| student\_gender | varchar | 64 | 否 | 否 | 学生性别 |
| college | varchar | 64 | 否 | 否 | 所在学院 |
| class\_information | varchar | 64 | 否 | 否 | 班级信息 |
| student\_telephone | varchar | 64 | 否 | 否 | 学生电话 |
| dormitory\_building\_id | varchar | 64 | 否 | 否 | 宿舍楼ID |
| dormitory\_no | varchar | 64 | 否 | 否 | 宿舍编号 |
| application\_date | date | 0 | 否 | 否 | 申请日期 |
| processing\_status | varchar | 64 | 否 | 否 | 处理状态 |
| reason\_for\_changing\_sleep | longtext | 0 | 否 | 否 | 换寝原因 |
| recommend | int | 11 | 是 | 否 | 智能推荐 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 更新时间 |

## ****5系统详细设计****

### **5.1用户前台功能模块**

学生宿舍管理系统，在系统首页可以查看首页、更换寝室、体温上报、维修申请、晚归信息、宿舍缴费等内容，如图5-1所示。

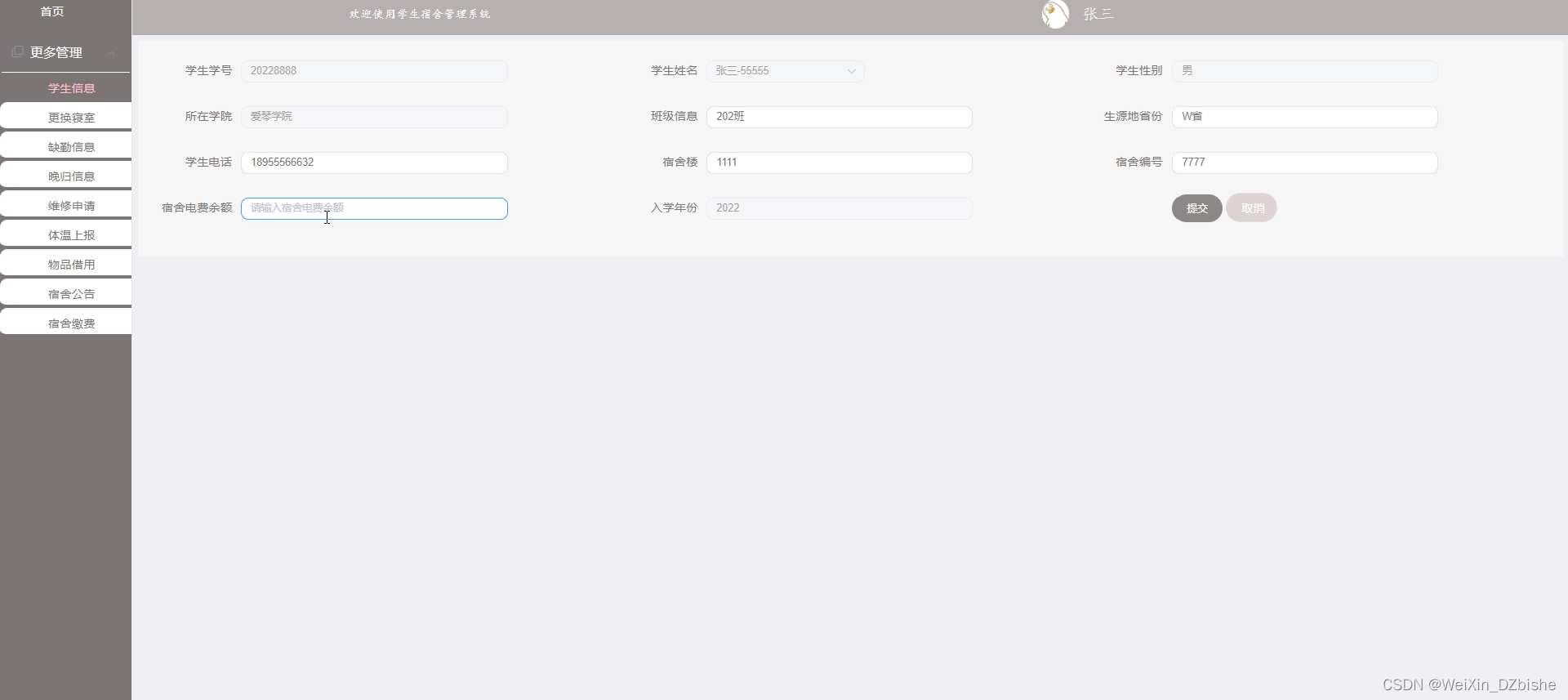


图5-1用户前台功能界面图

    用户注册，在用户注册页面可以填写账号、密码、昵称、邮箱、手机号码、身份、性别、身份证号码等详细内容进行注册，如图5-2所示。



图5-2用户注册界面图

用户注册关键代码如下所示。

/\*\*

     \* 注册

     \* @param user

     \* @return

     \*/

    @PostMapping("register")

    public Map<String, Object> signUp(@RequestBody User user) {

        // 查询用户

        Map<String, String> query = new HashMap<>();

        query.put("username",user.getUsername());

        List list = service.select(query, new HashMap<>()).getResultList();

        if (list.size()>0){

            return error(30000, "用户已存在");

        }

        user.setUserId(null);

        user.setPassword(service.encryption(user.getPassword()));

        service.save(user);

        return success(1);

}

/\*\*

     \* 用户ID：[0,8388607]用户获取其他与用户相关的数据

     \*/

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "user\_id")

    private Integer userId;

    /\*\*

     \* 账户状态：[0,10](1可用|2异常|3已冻结|4已注销)

     \*/

    @Basic

    @Column(name = "state")

    private Integer state;

    /\*\*

     \* 所在用户组：[0,32767]决定用户身份和权限

     \*/

    @Basic

    @Column(name = "user\_group")

    private String userGroup;

    /\*\*

     \* 上次登录时间：

     \*/

    @Basic

    @Column(name = "login\_time")

    private Timestamp loginTime;

    /\*\*

     \* 手机号码：[0,11]用户的手机号码，用于找回密码时或登录时

     \*/

    @Basic

    @Column(name = "phone")

    private String phone;

    /\*\*

     \* 手机认证：[0,1](0未认证|1审核中|2已认证)

     \*/

    @Basic

    @Column(name = "phone\_state")

    private Integer phoneState;

    /\*\*

     \* 用户名：[0,16]用户登录时所用的账户名称

     \*/

    @Basic

    @Column(name = "username")

    private String username;

    /\*\*

     \* 昵称：[0,16]

     \*/

    @Basic

    @Column(name = "nickname")

    private String nickname;

    /\*\*

     \* 密码：[0,32]用户登录所需的密码，由6-16位数字或英文组成

     \*/

    @Basic

    @Column(name = "password")

    private String password;

    /\*\*

     \* 邮箱：[0,64]用户的邮箱，用于找回密码时或登录时

     \*/

    @Basic

    @Column(name = "email")

    private String email;

    /\*\*

     \* 邮箱认证：[0,1](0未认证|1审核中|2已认证)

     \*/

    @Basic

    @Column(name = "email\_state")

    private Integer emailState;

    /\*\*

     \* 头像地址：[0,255]

     \*/

    @Basic

    @Column(name = "avatar")

    private String avatar;

    /\*\*

     \* 创建时间：

     \*/

    @Basic

    @Column(name = "create\_time")

    @JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")

    private Timestamp createTime;

    @Basic

    @Transient

    private String code;

}

登录，在登录页面通过填写账号、密码等信息完成登录，如图5-3所示。更换寝室页面通过填写学生信息、宿舍楼、宿舍编号、申请日期、换寝原因、处理状态等信息进行提交操作，如图5-4所示。



图5-3登录界面图

用户登录的关键代码如下。

 /\*\*

     \* 登录

     \* @param data

     \* @param httpServletRequest

     \* @return

     \*/

    @PostMapping("login")

    public Map<String, Object> login(@RequestBody Map<String, String> data, HttpServletRequest httpServletRequest) {

        log.info("[执行登录接口]");

        String username = data.get("username");

        String email = data.get("email");

        String phone = data.get("phone");

        String password = data.get("password");

        List resultList = null;

        Map<String, String> map = new HashMap<>();

        if(username != null && "".equals(username) == false){

            map.put("username", username);

            resultList = service.select(map, new HashMap<>()).getResultList();

        }

        else if(email != null && "".equals(email) == false){

            map.put("email", email);

            resultList = service.select(map, new HashMap<>()).getResultList();

        }

        else if(phone != null && "".equals(phone) == false){

            map.put("phone", phone);

            resultList = service.select(map, new HashMap<>()).getResultList();

        }else{

            return error(30000, "账号或密码不能为空");

        }

        if (resultList == null || password == null) {

            return error(30000, "账号或密码不能为空");

        }

        //判断是否有这个用户

        if (resultList.size()<=0){

            return error(30000,"用户不存在");

        }

        User byUsername = (User) resultList.get(0);

        Map<String, String> groupMap = new HashMap<>();

        groupMap.put("name",byUsername.getUserGroup());

        List groupList = userGroupService.select(groupMap, new HashMap<>()).getResultList();

        if (groupList.size()<1){

            return error(30000,"用户组不存在");

        }

        UserGroup userGroup = (UserGroup) groupList.get(0);

        //查询用户审核状态

        if (!StringUtils.isEmpty(userGroup.getSourceTable())){

            String sql = "select examine\_state from "+ userGroup.getSourceTable() +" WHERE user\_id = " + byUsername.getUserId();

            String res = String.valueOf(service.runCountSql(sql).getSingleResult());

            if (res==null){

                return error(30000,"用户不存在");

            }

            if (!res.equals("已通过")){

                return error(30000,"该用户审核未通过");

            }

        }

        //查询用户状态

        if (byUsername.getState()!=1){

            return error(30000,"用户非可用状态，不能登录");

        }

        String md5password = service.encryption(password);

        if (byUsername.getPassword().equals(md5password)) {

            // 存储Token到数据库

            AccessToken accessToken = new AccessToken();

            accessToken.setToken(UUID.randomUUID().toString().replaceAll("-", ""));

            accessToken.setUser\_id(byUsername.getUserId());

            tokenService.save(accessToken);

            // 返回用户信息

            JSONObject user = JSONObject.parseObject(JSONObject.toJSONString(byUsername));

            user.put("token", accessToken.getToken());

            JSONObject ret = new JSONObject();

            ret.put("obj",user);

            return success(ret);

        } else {

            return error(30000, "账号或密码不正确");

        }

}

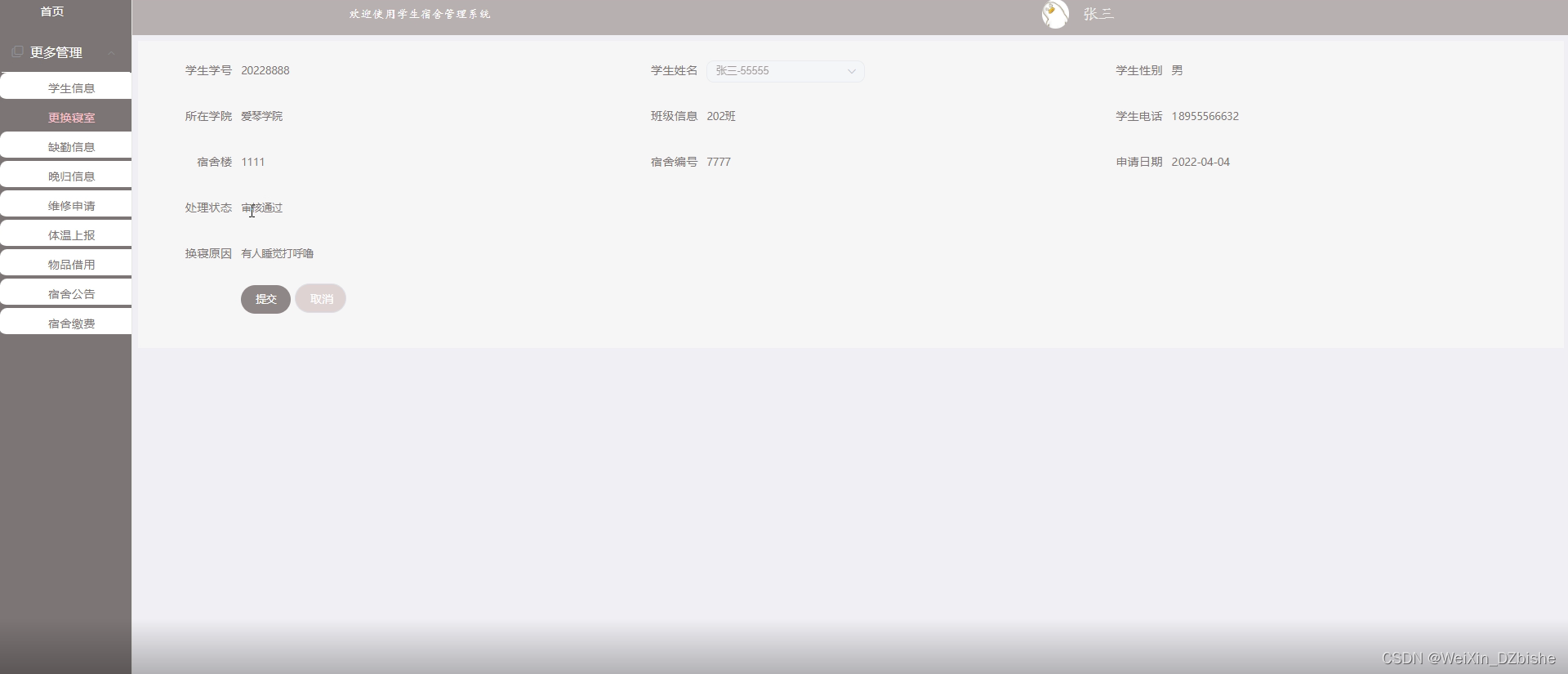


图5-4更换寝室界面图

维修申请，在申请页面通过填写学生信息、宿舍楼、宿舍编号、处理状态、故障原因等信息进行立即提交申请，如图5-5所示。体温上报页面填写具体学生信息、身体状况、体温、处理状态进行提交，如图5-6所示。

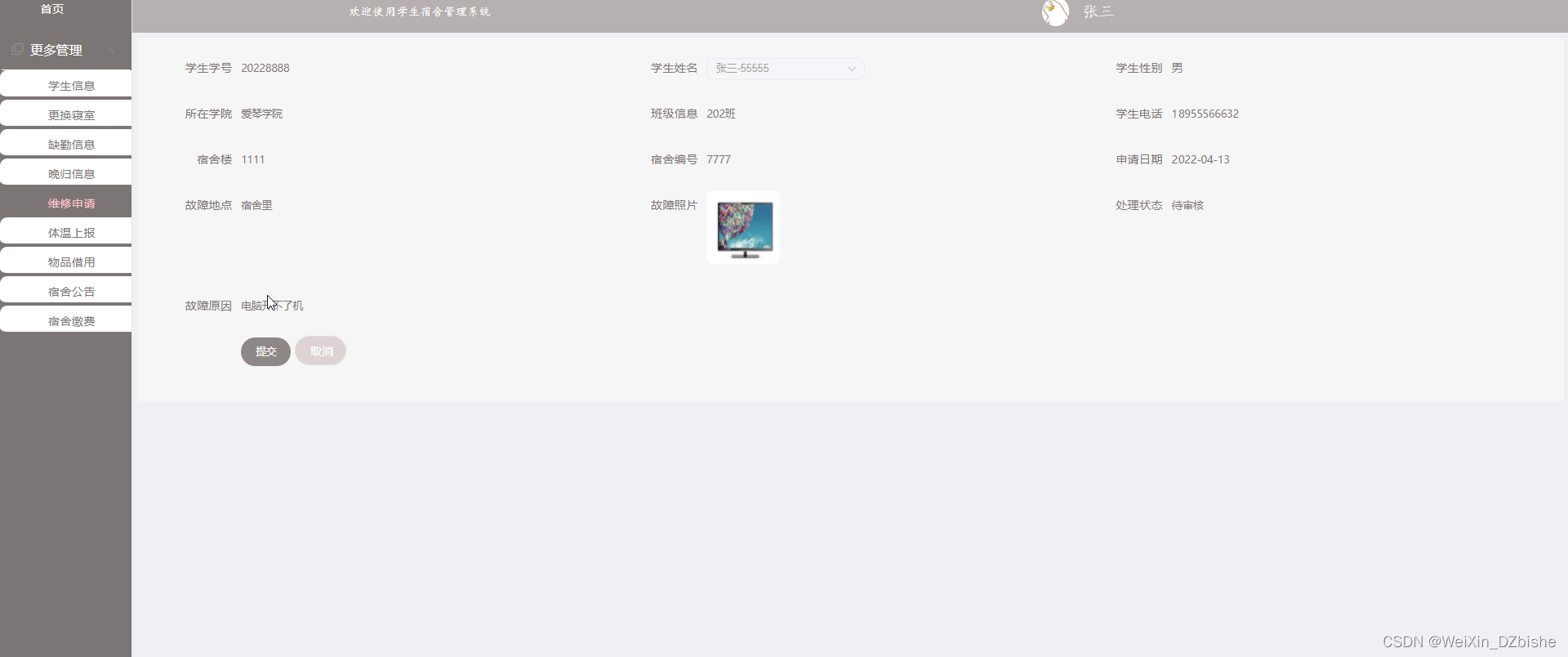


图5-5维修申请界面图

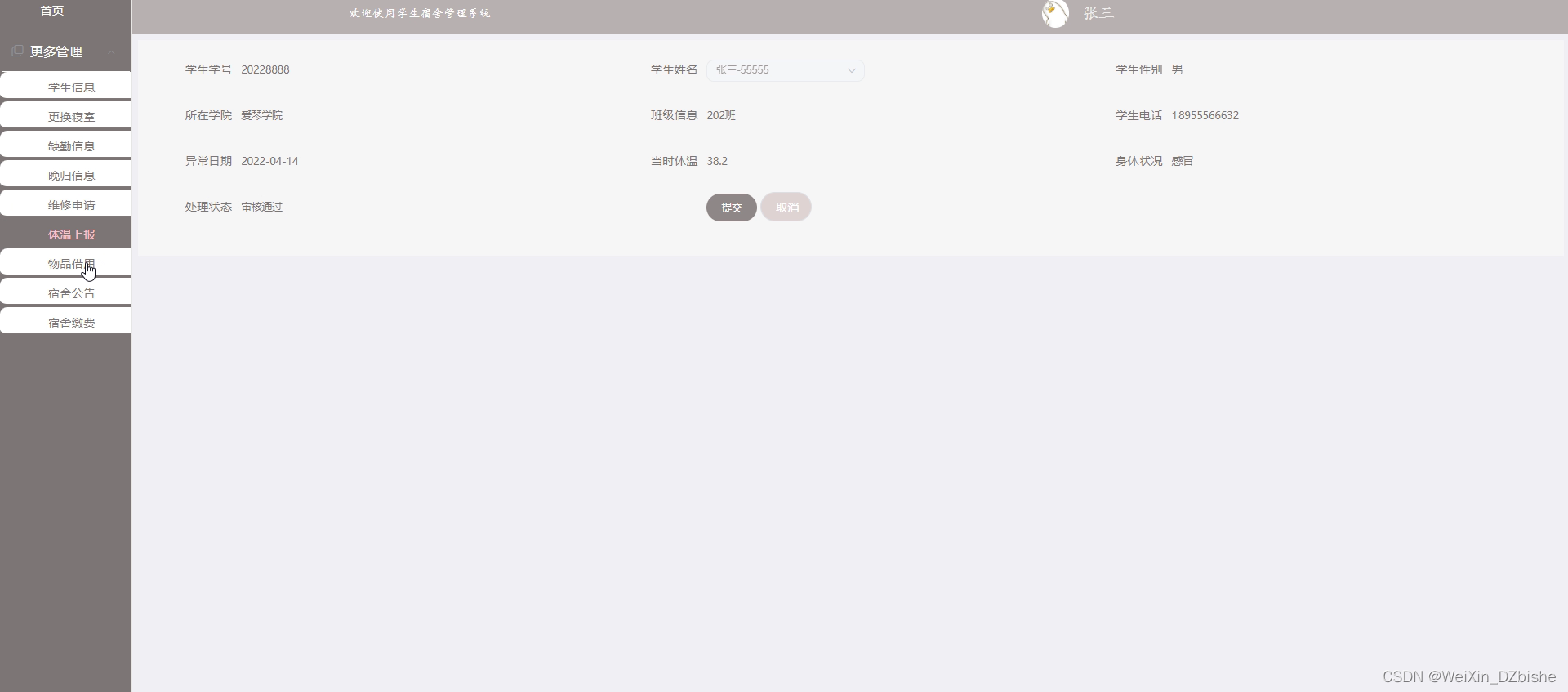


图5-6体温上报界面图

体温上报的逻辑代码如下：

  @PostMapping("/add")

    @Transactional

    public Map<String, Object> add(HttpServletRequest request) throws IOException {

        service.insert(service.readBody(request.getReader()));

        return success(1);

    }

    @Transactional

    public Map<String, Object> addMap(Map<String,Object> map){

        service.insert(map);

        return success(1);

}

    public Map<String,Object> readBody(BufferedReader reader){

        BufferedReader br = null;

        StringBuilder sb = new StringBuilder("");

        try{

            br = reader;

            String str;

            while ((str = br.readLine()) != null){

                sb.append(str);

            }

            br.close();

            String json = sb.toString();

            return JSONObject.parseObject(json, Map.class);

        }catch (IOException e){

            e.printStackTrace();

        }finally{

            if (null != br){

                try{

                    br.close();

                }catch (IOException e){

                    e.printStackTrace();

                }

            }

        }

        return null;

}

    public void insert(Map<String,Object> body){

        StringBuffer sql = new StringBuffer("INSERT INTO ");

        sql.append("`").append(table).append("`").append(" (");

        for (Map.Entry<String,Object> entry:body.entrySet()){

            sql.append("`"+humpToLine(entry.getKey())+"`").append(",");

        }

        sql.deleteCharAt(sql.length()-1);

        sql.append(") VALUES (");

        for (Map.Entry<String,Object> entry:body.entrySet()){

            Object value = entry.getValue();

            if (value instanceof String){

                sql.append("'").append(entry.getValue()).append("'").append(",");

            }else {

                sql.append(entry.getValue()).append(",");

            }

        }

        sql.deleteCharAt(sql.length() - 1);

        sql.append(")");

        log.info("[{}] - 插入操作：{}",table,sql);

        Query query = runCountSql(sql.toString());

        query.executeUpdate();

    }

### **5.2管理员功能模块**

管理员登录，通过填写注册时输入的用户名、密码、进行登录，如图5-7所示。



图5-7管理员登录界面图

管理员登录进入学生宿舍管理系统后的主要功能是：首页、用户管理、更换寝室管理、体温上报管理、维修申请管理、晚归信息、宿舍缴费管理等信息。5-8所示。还可以对更换寝室信息等进行管理操作，如图5-9所示。



图5-8首页展示界面图



图5-9更换寝室管理界面图

更换寝室管理界面逻辑代码如下：

@RequestMapping(value = {"/avg\_group", "/avg"})

public Map<String, Object> avg(HttpServletRequest request) {

        Query count = service.avg(service.readQuery(request), service.readConfig(request));

        return success(count.getResultList());

}

体温上报管理，在体温上报管理页面中可以对学生信息、上报体温、异常情况等信息进行管理，并可以对报修申请进行提交等操作，如图5-10所示。

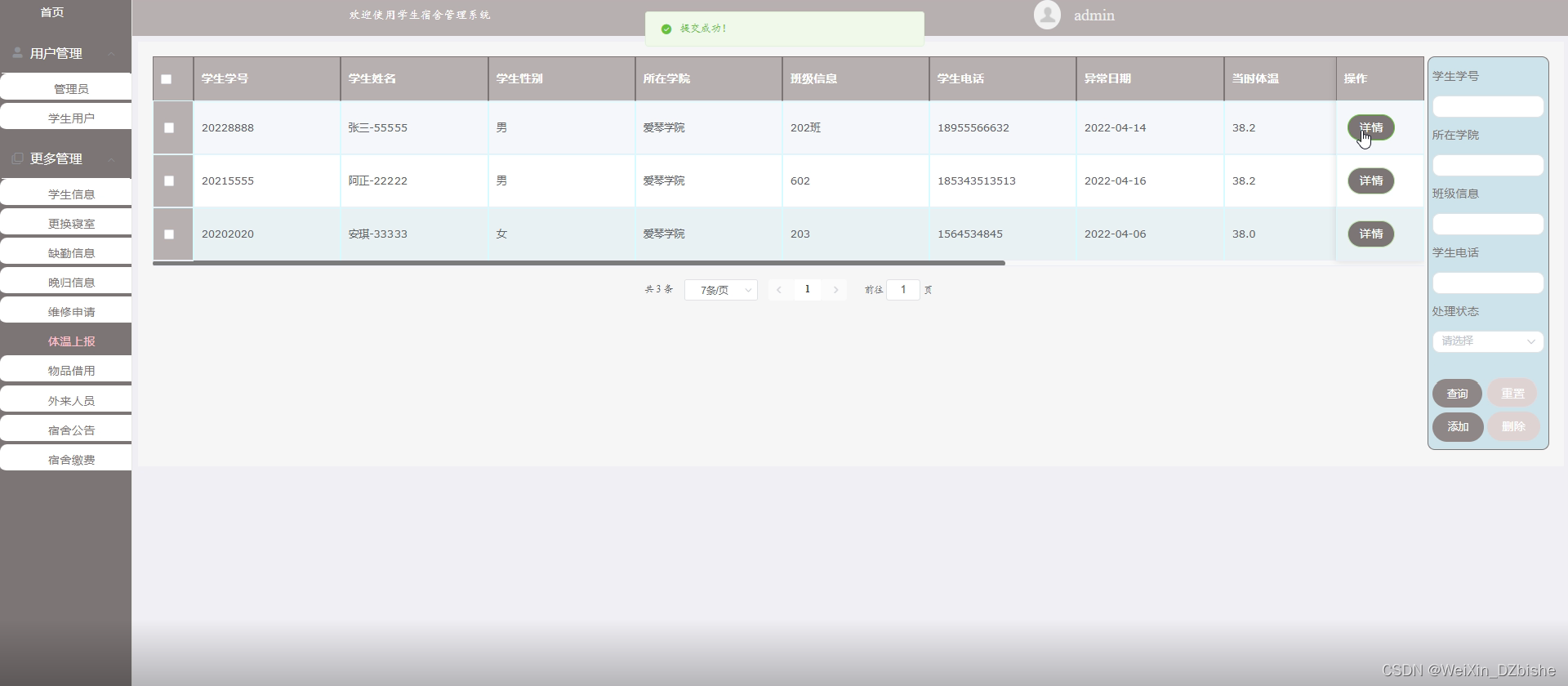


图5-10体温上报管理界面图

缺勤信息管理，在缺勤信息页面中可以对：学生信息、宿舍楼、所在学院、所在学校、缺勤信息等维护管理，并可以对需要修改的信息进行修改或删除等详细操作，如图5-11所示。

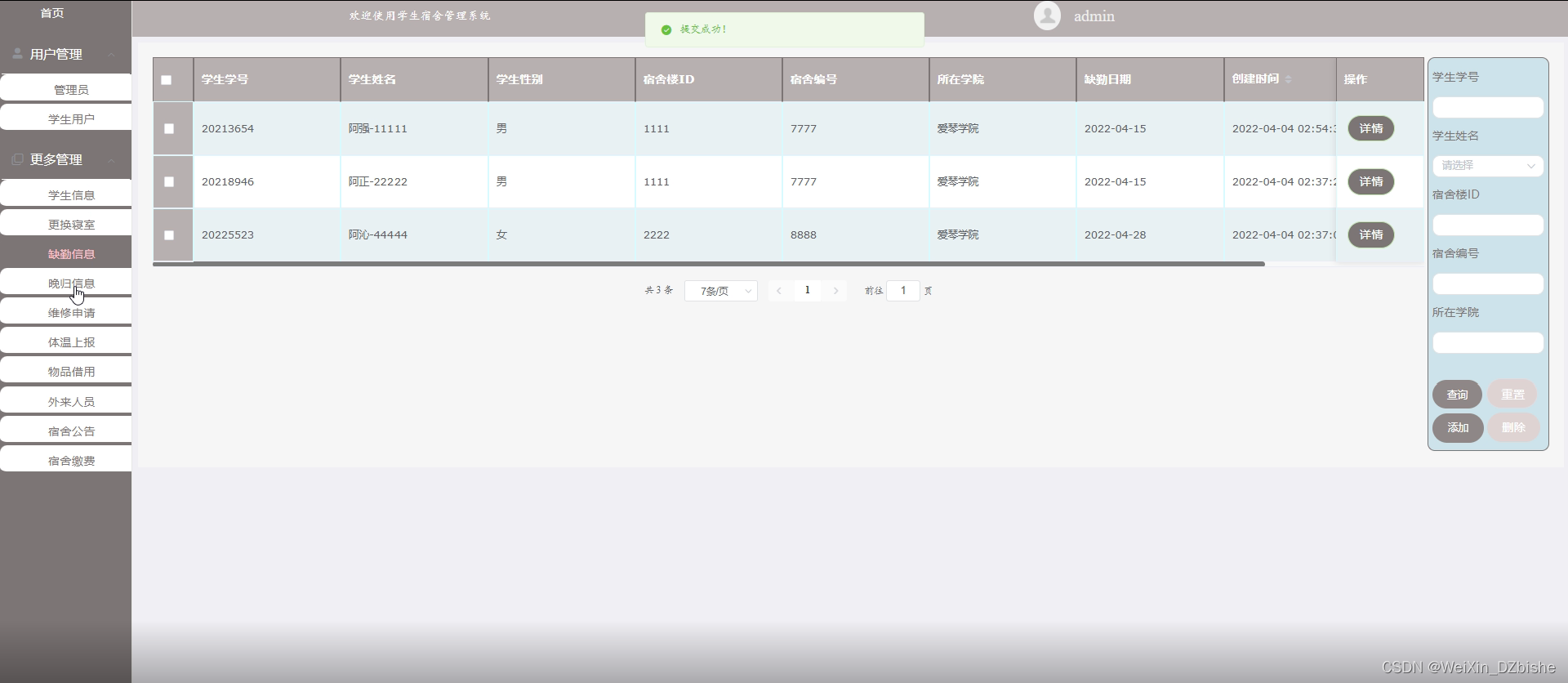


图5-11缺勤信息管理界面图

缺勤信息管理界面逻辑代码如下：

@RestController

@RequestMapping("auth")

public class AuthController extends BaseController<Auth, AuthService> {

    /\*\*

     \* 服务对象

     \*/

    @Autowired

    public AuthController(AuthService service) {

        setService(service);

    }

}

宿舍公告管理，在宿舍公告页面中可以查看标题、内容等内容，并且可以对新的宿舍公告进行发布，修改或删除等详细操作，如图5-12所示。

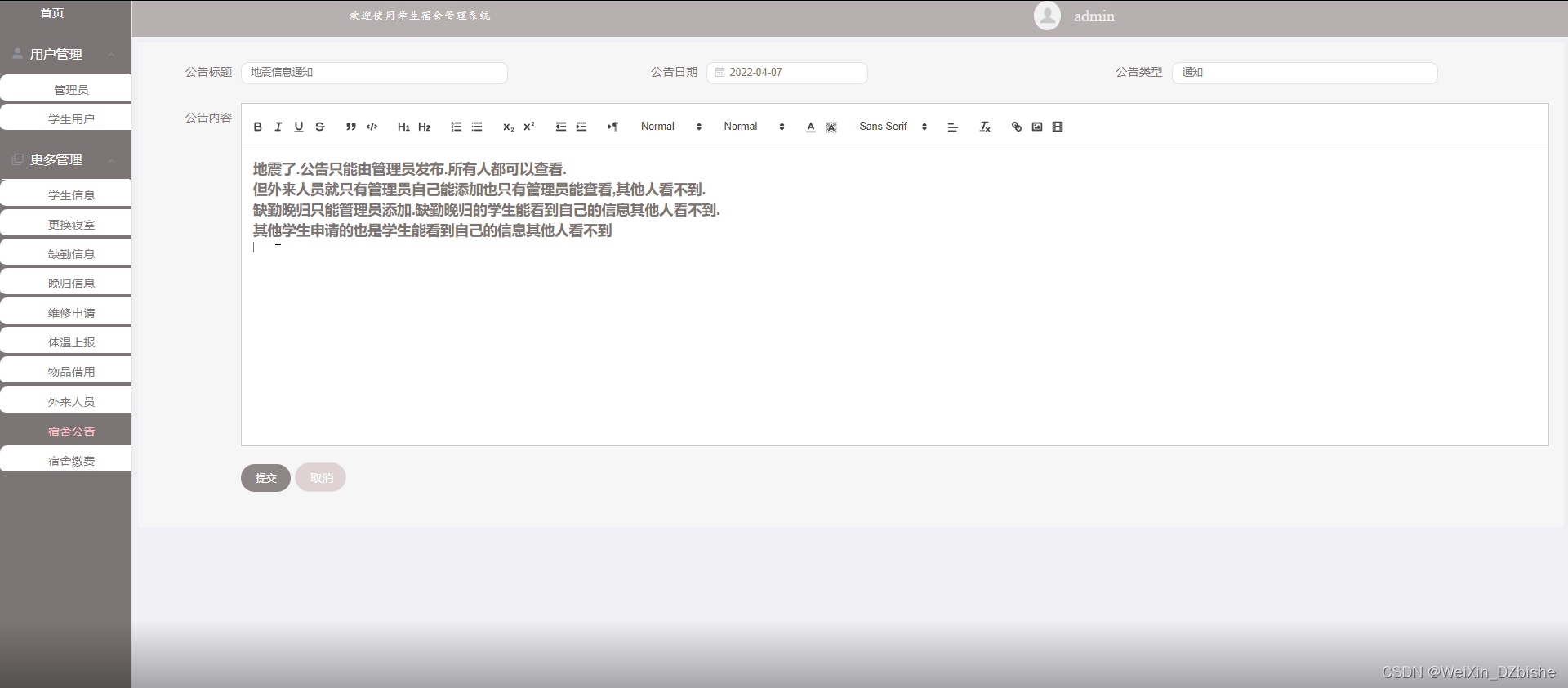


图5-12宿舍公告管理界面图

物品借用页面中主要包括：学生信息、所在学校、班级信息、借用日期等信息，如图5-13所示。

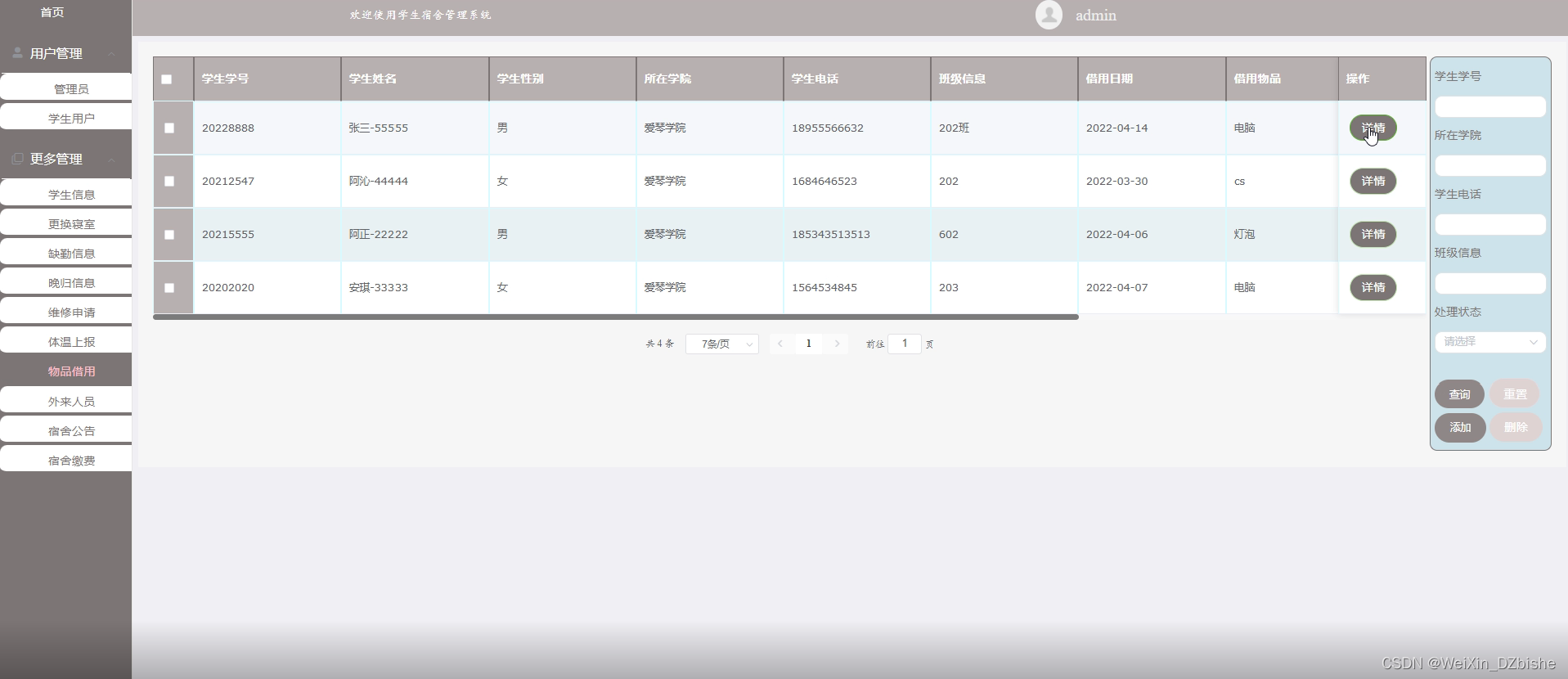


图5-13物品借用管理界面图

学生管理，在学生管理页面中可以查看学生姓名、密码、姓名、性别、头像等内容，并且根据需要对已有学生管理进行详情，修改或删除等详细操作，如图5-14所示。

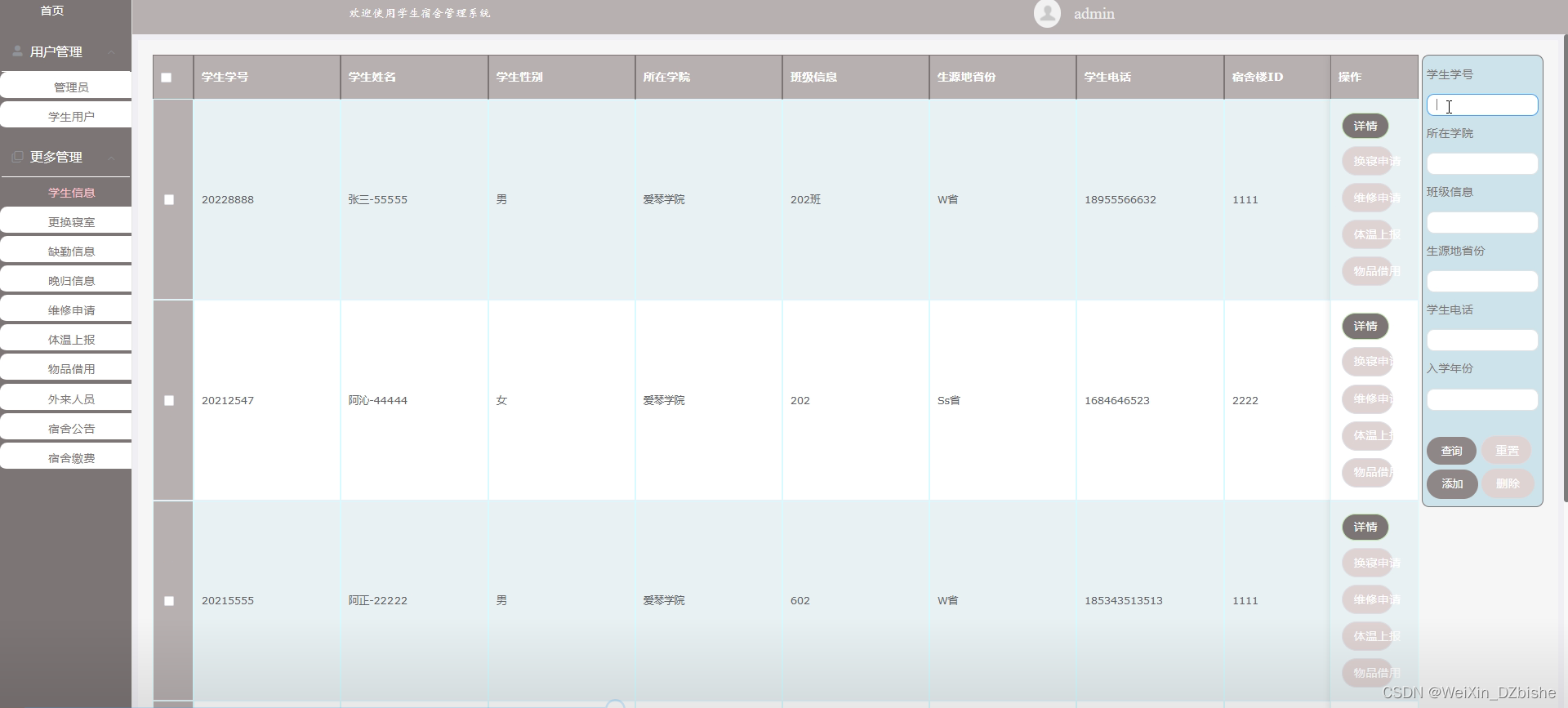


图5-14学生管理界面图

学生管理界面关键代码如下：

@RequestMapping("/get\_list")

    public Map<String, Object> getList(HttpServletRequest request) {

        Map<String, Object> map = service.selectToPage(service.readQuery(request), service.readConfig(request));

        return success(map);

}

## ****6系统测试****

系统开发的最后一个步骤就是系统测试，系统测试也是整个系统十分重要的一个环节，测试的好坏关系到产品的发展。用户对软件的质量、性能和可靠性等需求就要通过测试来实现。测试过程要必须遵循严谨性、完善性、规范性的原则，测试的主要目的就是看看在系统运行中，是否会出现bug，然后对出现的bug进行调试，直到程序完美运行。但是软件的测试只能尽可能的减少bug，理论上来说是无法达到消除bug。但是bug越少，系统出错的几率就越低，用户使用起来也更方便、更安全。

近年来，软件包含测试从现在的检验当中来看，系统接近预期目标可能出现的问题，并对这些错误做出相应的修正，假如我们不进行早期的测试错误就会延续下去，最后所做出的成品就会有很大的困难。

我们要在这个测试的过程当中找出错误。测试成软件开发的主要一部分，自从有了程序的设计那天开始，它就成为了重要的组成部分。经过统计来看，软件测试可以占据这个系统45%的工作量，而在软件开发的成本当中，对于测试成本来说它包含了很多的测试工作。每个程序测试时都会出现和遇到错误。在整个程序的开发过程当中，人为去查找错误是非常复杂和困难的，所以我们一般都会找一些测试的工具来进行测试。

### ****6.1系统测试的意义****

随着现代信息的快速发展，在社会各大领域中已经都开始应用网络信息技术，在应用网络技术的同时人们也开始把软件的质量问题作为了一个重要焦点来关注，因为一个软件的好与坏它决定着这个系统在市场上的生存，所以我们必须要把软件质量来做好，这样才有一定的生存能力。对于用户来说它们首先选用的都是保证这个系统软件的质量问题，因为一个系统的软件质量决定着用户在后期上成本经济的问题。

### ****6.2 测试方法****

具体测试方法包括：黑盒测试和白盒测试。

黑盒测试又被人们称作为功能测试，通常是在程序的接口来做一些测试的方法，它一般包括对程序的功能和使用的方法来做出一些数据的接受和输出，同时还可以做出正确的输出信息，并保证与外部信息的完整性。

白盒测试通常被人们称作为结构测试，在整个程序的结构和处理当中它是由程序当中的逻辑测试和检验程序来完成一些正确的工作。

具体的功能测试它是包括：系统的适用性、准确性、安全性等功能测试。

### ****6.3测试分析****

学生宿舍管理系统满足相关信息的管理需求，在设计时借鉴了国内外优秀网站的优点，从界面到系统设计都保证了管理员以及用户能够方便操作。系统的主要特点和优点归纳如下：

（1）本系统用的移置性和针对性都比较高，因为针对性高可以提供更好的服务而移置性可以在多个系统上运行，更给用户带来了极大的方便。

（2）该学生宿舍管理系统内容全面，管理方便可以及时的全面的处理各种错误，异常，这样避免了很多因用户的马虎操作而出现的失误，其操作方便，用户界面友好，能够上网的人都可以很好的进行操作。

经过对上述的测试结果分析，所有基本功能齐全，操作简单，系统运行性能良好，系统安全可靠，能促进学生宿舍管理系统的发展，发展前景广阔。

## ****结    论****

本系统通过对java和Mysql数据库的简介，从硬件和软件两反面说明了学生宿舍管理系统的可行性，本文结论及研究成果如下：实现了java与Mysql相结合构建的学生宿舍管理系统，网站可以响应式展示。通过本次学生宿舍管理系统的研究与实现，我感到学海无涯，学习是没有终点的，而且实践出真知，只有多动手才能尽快掌握它，经验对系统的开发非常重要，经验不足，就难免会有许多考虑不周之处。比如要有美观的界面，更完善的功能，才能吸引更多的用户。

由于在此之前对于java知识没有深入了解，所以从一开始就碰到许多困难，例如一开始的页面显示不规范、数据库连接有问题已经无法实现参数的传递等等，不过通过在网上寻找有关资料以及同学的帮助下最后都得到了解决，在此过程中，我不仅学到了很多知识，也提高了自己解决问题的能力，尤其是学会如何从大量的信息中筛选出所需有用的信息，同时我更加深刻的体会到了，虽然书本上的大部分知识都是有价值，正确的，但实际上每个人编程的思路和对数据处理的方法、思想都是不同的，这就要求我们一定要通过实践才能找到解决问题的方案。在此次毕业设计活动中，我不断的提高了自己，也得到了宝贵的经验，我相信这些对我以后的发展都会有很大帮助。

通过这次学生宿舍管理系统的开发，我参考了很多相关系统的例子，取长补短，吸取了其他系统的长处，逐步对该系统进行了完善，但是该系统还是有很多的不足之处，有待以后进一步学习。

实践证明，学生宿舍管理系统有着非常好的发展前景，经过测试运行，系统各项功能都十分完善，界面漂亮，使用方便，操作容易，在技术理论上已经成熟。

## ****致    谢****

大学的学习生活在这个季节将结束，但是在我的生命这仅仅只是一个逗号，我将面对的旅程的另一个的开始。通过这次毕业设计的整个开发过程，从需求分析到具体功能实现，再到最终测试和维护的理解有了很大的进步，让我对系统开发有了更深刻的认识，对我个人的实践能力和解决问题的能力，都有了很大的帮助。这是这次毕业设计最大的收获。

首先要感谢我的指导老师，他在论文写作上，给予了我各种无私的帮助，治学严谨，严格要求，使我获得了很大的收获。老师深厚的理论知识和丰富的实践经验，都深深的影响到我，在这方面，我谨表示衷心的感谢。

其次，我还要对所有的老师和同学表示感谢，他们在我学习的过程中，都积极的提供了很多帮助，无论是专业知识，还是实践操作技能，也能够让我在论文写作中，遇到的一些难题迎刃而解。

最后，对阅读和评审本论文的各位老师表示衷心的感谢!

## ****参考文献****

[1]孙妃,李可心,刘楠,王晓兰,孔荣荣,王慧.学生宿舍管理系统的设计与实现[J].江苏科技信息,2021,38(29):40-42+75.

[2]朱宏斌.智能控电管理系统在高校学生公寓中的应用——以宿迁学院为例[J].数字通信世界,2021(10):194-195.

[3]黄炜钢.基于移动端的中职学生宿舍管理系统的设计与实现[J].信息与电脑(理论版),2021,33(02):141-144.

[4]冯晓春,陈艳.基于web开发的学生公寓管理系统设计与实现[J].科技创新与应用,2020(31):77-78.

[5]王锋,金晶.高校学生宿舍水电表管理系统设计与实现[J].智能计算机与应用,2020,10(06):191-194.

[6]辛阅,黄浪尘,李家儿.学生宿舍远程智能管理系统设计[J].软件,2020,41(03):79-83.

[7]Xiaochen Geng,Sha Liu. Application of Modular Interface Design in Student Dormitory Management System[C]//Proceedings of 4th International Conference on Culture,Education and Economic Development of Modern Society(ICCESE 2020)(Advances in Social Science,Education and Humanities Research,VOL.416).Proceedings of 4th International Conference on Culture,2020:173-180.

[8]欧畅.宿舍管理系统的前端设计与实现[J].湖北农机化,2020(02):186-187.

[9]贾树刚.学生宿舍管理系统的设计与实现[J].辽宁师专学报(自然科学版),2019,21(04):44-46+100.

[10]初庆东,张敏霞,崔卫韬.一种新型高校学生信息管理系统的设计与实现[J].信息技术,2019,43(06):69-71+76.

[11]王青.高校宿舍智能管理系统设计[J].微型电脑应用,2019,35(06):135-137.

[12]俞玲.基于数据挖掘的高校学生宿舍管理系统研究与设计[J].信息与电脑(理论版),2019(04):121-122.

[13]徐洋.关于“互联网+高校学生宿舍”的研究[J].长江丛刊,2019(04):147-148.

[14]黄敏.基于B/S架构的学生宿舍管理系统数据库设计与实践[J].电脑知识与技术,2018,14(35):83-84+92.

[15]Wang Xi,Wang De. Research of Student Management Mechanism under the Background of Credit System Based on Student Dormitory[C]//Proceedings of the 14th International Conference on Innovation and Management.,2017:1676-1680.