

本科生毕业设计

|  |
| --- |
| 基于Spring Boot的大学生宿舍自选管理系统的设计与实现 |

姓 名 刘仕恒

学 号 1008519070310

院 系 计算机系

专 业 软件工程

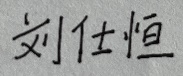
指导教师 崔晶

二〇二三 年 五 月 二十六 日

学位论文原创性声明

本人所提交的学位论文基于Spring Boot的大学生宿舍自选管理系统的设计与实现，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

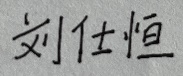
本声明的法律后果由本人承担。

论文作者（签名）： 指导教师确认（签名）：

年 月 日 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解河北水利电力学院有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北水利电力学院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

论文作者（签名）： 指导教师（签名）：

年 月 日 年 月 日

摘 要

本研究旨在设计一个大学生宿舍自选管理系统，加强大学生宿舍自主权、提高大学生住宿体验，为大学生提供方便快捷的宿舍管理方式。传统的管理方式有工作量大、复杂的特点，并且学生是被动选择宿舍。该系统采用了Spring Boot+MySQL等技术，实现了宿舍选择、离宿记录、报修等多个方面的自主选择，同时也方便管理人员进行宿舍管理。经过调查问卷、实地调研等方式，该系统得到了用户的积极评价。结果显示，该系统提高了学生对于宿舍生活的参与度，同时有效缩减了管理人员的工作量，促使宿舍管理的智能化、数字化。本文详细介绍了该系统的系统设计、技术实现、功能模块等方面，对于大学生宿舍管理的改革和提升具有一定的参考价值。

关键词：大学生宿舍自选管理系统；Spring Boot框架；MySQL数据库

ABSTRACT

The aim of this study is to design a self selection management system for college students' dormitories, strengthen their autonomy, improve their accommodation experience, and provide convenient and efficient dormitory management methods for college students. Traditional management methods have the characteristics of heavy workload and complexity, and students are passive in choosing dormitories. The system adopts technologies such as Spring Boot and MySQL to achieve independent selection in multiple aspects such as dormitory selection, departure records, and repair reporting, while also facilitating dormitory management for management personnel. After conducting surveys, on-site research, and other methods, the system has received positive feedback from users. The results show that the system improves students' participation in dormitory life, while effectively reducing the workload of management personnel, promoting the intelligence and digitization of dormitory management. This article provides a detailed introduction to the system design, technical implementation, functional modules, and other aspects of the system, which has certain reference value for the reform and improvement of college student dormitory management.

**Keywords**: College student dormitory self selection management system; Spring Boot framework; MySQL database

目 录

[摘 要 I](#_Toc19332)

[第1章 绪论 1](#_Toc17924)

[1.1 课题背景 1](#_Toc20205)

[1.2 目的和意义 1](#_Toc18082)

[1.2.1 课题目的 1](#_Toc29991)

[1.2.2 课题意义 1](#_Toc474)

[1.3 国内研究现状 2](#_Toc18282)

[1.4 本文主要研究内容 3](#_Toc22128)

[1.5 本文结构 3](#_Toc25133)

[第2章 需求分析 3](#_Toc19638)

[2.1 系统目标及流程分析 3](#_Toc31566)

[2.1.1 系统目标 3](#_Toc7837)

[2.1.1 系统主要流程分析 4](#_Toc31322)

[2.2 系统需求分析 5](#_Toc23898)

[2.2.1 功能需求分析 5](#_Toc15393)

[2.2.2 非功能性需求分析 8](#_Toc2340)

[2.3 本章小结 9](#_Toc9223)

[第3章 系统设计 10](#_Toc21451)

[3.1 系统总体设计 10](#_Toc28839)

[3.2 系统功能模块设计 10](#_Toc19781)

[3.2.1 用户登录模块设计 10](#_Toc14022)

[3.2.2 学生在线选宿舍模块设计 11](#_Toc29025)

[3.2.3 用户管理模块设计 12](#_Toc15530)

[3.2.4 学生管理模块设计 13](#_Toc2102)

[3.2.5 楼宇管理模块设计 13](#_Toc329)

[3.2.6 宿舍管理模块设计 13](#_Toc5914)

[3.2.7 宿舍预选设置模块设计 14](#_Toc22009)

[3.2.8 报修管理模块设计 14](#_Toc19990)

[3.3 数据库设计 14](#_Toc26240)

[3.3.1 数据库概念结构设计 14](#_Toc32307)

[3.3.2 数据库逻辑结构设计 17](#_Toc10929)

[3.4 本章小结 24](#_Toc18210)

[第4章 系统实现 25](#_Toc22018)

[4.1 开发和使用框架 25](#_Toc24314)

[4.2 相关技术介绍 25](#_Toc15547)

[4.2.1 LayUI简介 25](#_Toc11563)

[4.2.2 Spring Boot简介 25](#_Toc9134)

[4.2.3 MySQL简介 26](#_Toc31198)

[4.2.4 Redis 简介 26](#_Toc22166)

[4.3 功能模块设计实现 26](#_Toc15688)

[4.3.1 学生模块首页界面实现 26](#_Toc658)

[4.3.2 学生在线选宿舍页面实现 27](#_Toc14024)

[4.3.3 学生离宿记录页面实现 30](#_Toc32087)

[4.3.4 超级管理员首页界面实现 31](#_Toc29128)

[4.3.5 用户管理页面实现 34](#_Toc9996)

[4.3.6 机构管理页面实现 36](#_Toc20142)

[4.3.7 学生管理页面实现 38](#_Toc17390)

[4.3.8 宿舍编号设置页面的实现 41](#_Toc6250)

[4.3.9 宿舍管理页面的实现 42](#_Toc21550)

[4.3.10 公告管理页面的实现 44](#_Toc13157)

[4.3.11 菜单管理页面的实现 46](#_Toc25514)

[4.4 本章小结 48](#_Toc6353)

[第5章 系统测试 49](#_Toc22631)

[5.1 测试环境 49](#_Toc10678)

[5.2 功能测试 49](#_Toc15872)

[5.2.1 超级管理员/宿管员/学生登录功能测试 49](#_Toc32562)

[5.2.2 楼宇管理功能测试 49](#_Toc15712)

[5.2.3 宿舍编号设置管理功能测试 50](#_Toc18869)

[5.2.4 宿舍管理功能测试 50](#_Toc11144)

[5.2.5 宿舍分配功能测试 51](#_Toc8016)

[5.2.6 在线选宿舍功能测试 51](#_Toc12069)

[5.3 本章小结 52](#_Toc22862)

[结论 53](#_Toc15432)

[参考文献 54](#_Toc24197)

[致谢 55](#_Toc13601)

[附录A:系统核心代码 56](#_Toc19765)

[附录B：软件使用说明书 61](#_Toc15248)

# 绪论

## 课题背景

大学生宿舍自选管理系统的课题背景在于改善传统大学宿舍管理模式[[[1]](#endnote-1)]，将宿舍管理变得数字化和智能化，提高管理效率和学生的宿舍管理体验[[[2]](#endnote-2)]。主要背景包括以下几个方面：

传统宿舍管理模式存在缺陷，传统大学宿舍管理模式往往采用手工登记、纸质记录等方式，存在信息孤立、漏洞多、流程繁琐等问题，导致学生宿舍管理效率低下，学生宿舍管理信息难以及时公开、传达、反馈等等。然而这种情况会降低宿舍管理和学生宿舍生活的质地和满意度。

智能化改造是必要趋势，伴随数据库分析[[[3]](#endnote-3)]、人工智能技术和物联网技术等科技的不断更新，智能宿舍管理正成为一种新的趋势。操纵数字化和智能化技术，将学生和管理员之间的通讯渠道落实数字化，能够有效提高大学宿舍管理的效率和透明度，从而推动大学宿舍管理的改进。

满足学生个性化需求：在大学校园里，每个学生的个性以及对生活环境的期望不同。为了让学生更加方便合理地选择和管理自己的宿舍，提供不同的需求和选择，在宿舍自主选择和管理方面嵌入数字化管控也变得十分必要。

综上，大学生宿舍自选管理系统[[[4]](#endnote-4)]的课题支柱主要是基于传统宿舍管理模式存在的问题和科技的发展主旋律，通过数字化和智能化技术实施宿舍管理，以提高学生管理体验和宿舍使用效率。

## 目的和意义

### 课题目的

大学生宿舍自选管理系统是一款面向大学校园的宿舍管理的软件[[[5]](#endnote-5)]，主要为学生提供宿舍自主选择、离宿、报修和管理等服务，同时也提供管理员进行宿舍管理和监管的功能。该系统的优点在于，它可以提高宿舍管理的效率和透明度，让学生和管理员之间的沟通更加便捷，同时也为学生提供了更好的宿舍管理体验。大学宿舍是所有学生在校时间里都要经历的一大部分，毕竟学生的日常生活和学习起到了至关重要的作用。因此，大学生宿舍管理系统对于大学校园管理具有一定的推进作用。

### 课题意义

大学生宿舍自选管理系统的课题意义在于改进大学宿舍管理的方式[[[6]](#endnote-6)]，提高宿舍管理的效率和透明度，从而为学生提供更好的宿舍管理体验。具体来说，这个系统的意义和价值大概表现在以下几个方面：

（1）提升宿舍管理的效率：传统的宿舍管理方式其实存在各类信息不透明、相关流程繁琐等问题，而大学生宿舍自选管理系统则可以实现宿舍信息、学生信息、预约信息、调配或维修请求的快速处理和更加即时的反馈，从而提升宿舍管理的效率。

（2）提升宿舍管理的透明度：学生可以在系统上直观地了解宿舍床位状况、空床位信息，同时也可以查看自己的预约情况和宿舍调配信息。对于系统管理员，通过该系统可以及时了解学生预约、请求、投诉等方面的信息，以加强管理工作，提高透明度。

（3）促进宿舍生活的改善：将宿舍管理数字化、可视化，可以加强学生对自身宿舍生活的管理和把控。同时，宿舍管理人员可以在系统中了解宿舍状况，定位和解决管理问题，因此会更好地保障学生宿舍生活本身的品质的提升和需求。

（4）增加学生参与度和满意度：通过大学生宿舍自选管理系统，学生参与宿舍管理的过程变得更加主动与便捷，他们可以自主选择更合适的宿舍、查看当前宿舍成员和报修申请等，从而提高他们参与宿舍管理的积极性和满意度。

基于以上意义，大学生宿舍自选管理系统可以更好地推进大学校园管理[[[7]](#endnote-7)]的数字化与智能化，为未来的校园管理提供创新思路和借鉴。

## 国内外研究现状

目前，大学生宿舍自选管理系统已经成为宿舍管理领域的一个热门课题，国内外已经涌现出不少相关的研究。以下是目前国内外研究现状的概述：

（1）国内研究现状：

国内的相关研究主要聚焦于大学宿舍管理中应用信息技术和数字化技术的实际应用与推广。这些研究从不同角度和层面对大学生宿舍自选管理系统的开发和实践问题进行了深入探讨，包括系统架构设计、用户需求分析、数据信息和数据处理等方面。

（2）国外研究现状：

国外方面，美国、欧洲等地区的大学已经广泛应用数字化和智能化技术进行宿舍管理，比如在宿舍投资、床位管理、维修管理等方面开发和使用了大量的软件和应用程序。其中一些系统可以为学生提供更个性化的选择和服务，同时也可以帮助管理员更好地加强宿舍监管和管理。

总的来说，国内外相关研究都反映出数字化和智能化技术在大学宿舍管理中的重要性和应用前景。当前，随着科技的不断进步和互联网技术应用的广泛推广，数字化和智能化的宿舍管理势必会成为未来学校宿舍管理的必然趋势。

## 本文主要研究内容

大学生宿舍自选管理系统的主要研究内容中包括以下几个方面：

1. 宿舍自选管理系统的需求分析：

调查大学生对于宿舍管理系统的需求，包括在线选宿舍、报修申请、公告查看等方面的需求，为设计宿舍自选管理系统提供依据。

（2）宿舍自选管理系统的设计与实现:

根据需求分析结果, 设计宿舍自选管理系统的功能模块及其相关技术，并进行系统实现。

（3）宿舍自选管理系统的测试与评估：

对宿舍自选管理系统进行测试，检查其性能、稳定性、安全性、易用性等方面的表现，并进行用户满意度评估。

（4）系统优化升级：

对宿舍自选管理系统进行优化升级，提升系统的用户体验和性能，同时增强系统的安全性、稳定性等方面的保障。

## 本文结构

结合大学生宿舍自选管理系统的需求分析，进行系统设计和开发，直到最后实现大学生宿舍自选管理系统的开发。

本文共有五章，内容如下：

第一章为绪论，阐述了大学生宿舍自选管理系统的项目的开发背景、目的意义以及国内外研究现状。

第二章为需求分析，结合需求分析，确认系统主要的开发模块。

第三章为系统设计，介绍了系统的整体核心思路以及各模块和数据库的设计。

第四章为系统实现，主要模块的实现与核心Java代码的部分展示与描述。

第五章为系统测试，对系统进行全面测试，为系统排除已检测到的问题。

# 系统开发的平台与技术

* 1. 开发和使用框架

在系统实现的过程中，系统的开发环境如表2.1所示。

表2.1 开发环境

|  |  |
| --- | --- |
| 环境项 | 环境参数 |
| 操作系统 | Windows 11 |
| 数据库可视化工具 | Navicat |
| 开发工具 | IDEA |
| 浏览器 | Google Chrome |

在系统实现的过程中，系统的开发框架如表2.2所示。

表2.2 开发框架

|  |  |
| --- | --- |
| 前端 | 后端 |
| LayUI、jQuery | Spring Boot |

* 1. 相关技术介绍
     1. LayUI简介

LayUI 是一款由咱们国家自己开发的UI框架。使它的好处就是后端人员只需在前端配置好接口，后端则按照定义好的接口规则返回数据，就可完成页面的展示，大大降低了后端人员的开发成本。

* + 1. Spring Boot简介

Spring Boot是Spring全家桶的一员，其诞生以来以简化配置著称，这样使后端开发人员不再需要定义很多的xml配置。提升开发速度[9]。

* + 1. MySQL简介

MySQL是现在比较常用的数据库，使用它进行网站数据整体管理，它既能方便地进行数据[存储](http://storage.it168.com/)，也能使数据更新变得方便，操作简单，体积小[3]。

* + 1. Redis 简介

Redis是一款的基于内存的key-value形式的通常被作为缓存使用的数据库。

# 需求分析

## 系统目标及流程分析

### 系统目标

大学生宿舍自选管理系统是一个方便学生在线选择宿舍的宿舍管理系统，通过系统的设计能够增强学生对于宿舍的选择权和提高宿舍的管理工作的运转效率和准确性，与此同时推进大学的教育发展。

### 系统主要流程分析

学生登录系统后可进行宿舍的选择。超级管理员登录系统后添加楼宇信息，然后为每栋楼宇设置宿舍编号，并对宿舍进行初始化。之后需要为班级手动添加预选信息，包括班级名字、选择开始时间和结束时间。在为宿舍预选班级分配对应的宿舍后，学生就能在指定时间内进行宿舍选择了。若没有为班级设置预选宿舍信息，则不能进行宿舍选择。

管理员使用本系统的主要流程图如图所示：

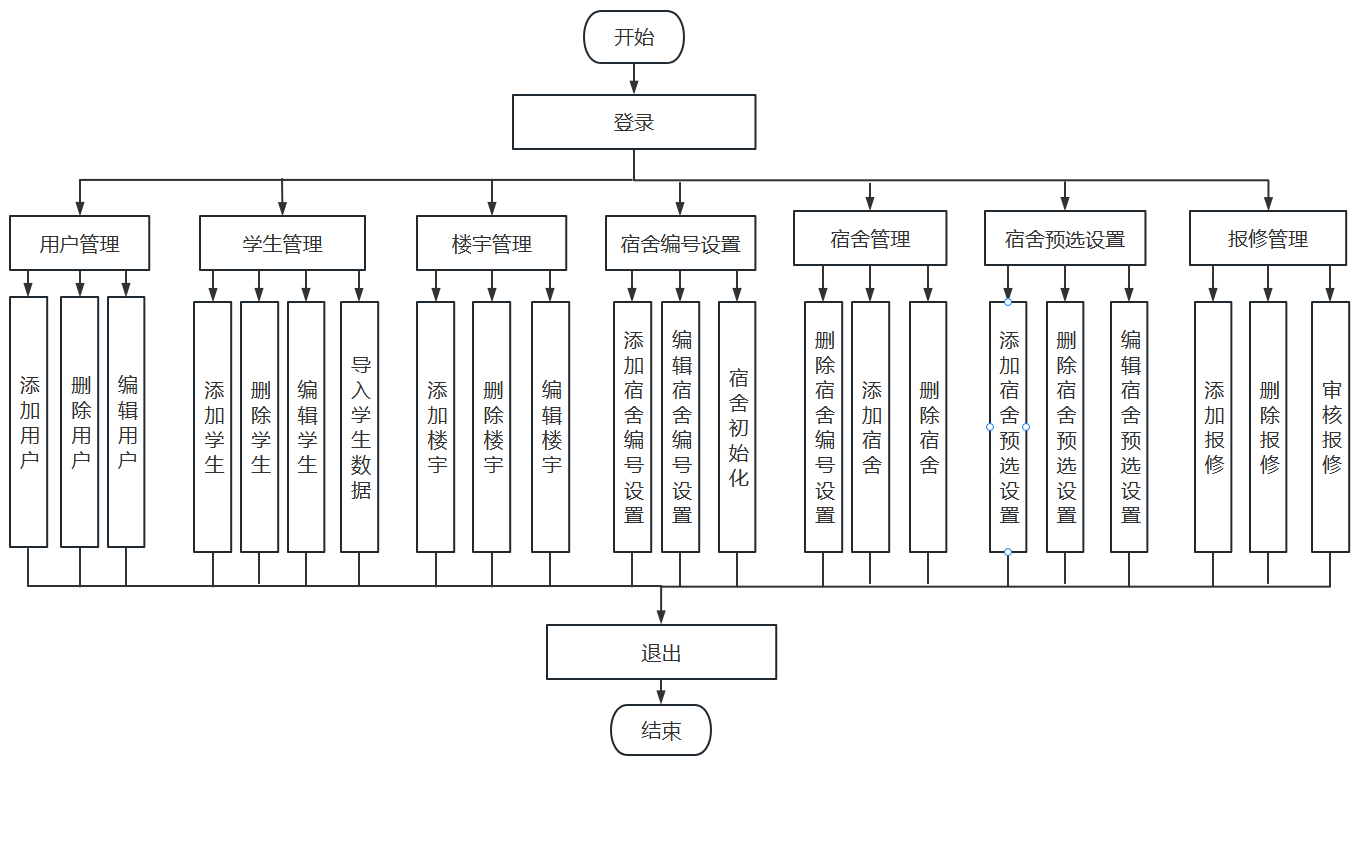


图 3.1 管理员流程图

学生使用本系统的主要流程如图所示

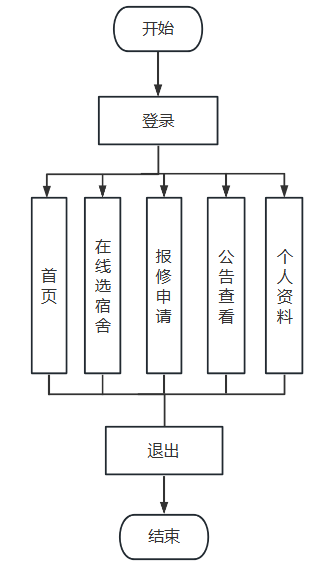


图 3.2 学生流程图

## 系统需求分析

### 功能需求分析

（1）注册登录功能：学生需要注册并登录该系统才能进行宿舍的选择。

（2）在线选择宿舍功能：学生需要在系统中线上选择可用的空床位。

（3）宿舍信息管理功能：超级管理员可以添加编辑和删除宿舍信息，包括宿舍号、楼宇号、楼层数等。

（4）学生管理功能：超级管理员可以添加和删除学生信息，并可以查看和编辑学生的详细信息，例如姓名、手机号、性别、学号等。

（5）床位分配调整功能：超级管理员和宿管员可选择需要调整的床位进行床位的调整。

（6）宿舍报修功能：学生可以在系统中提交宿舍报修求和必要的信息。超级管理员和宿管员会收到申请并及时有效审核。

（7）宿舍评价功能：超级管理员和宿管员可对宿舍进行评价。

（8）数据报表功能：管理端的首页展示各种报表，包括宿舍床位使用率、学生入住情况、宿舍闲置数量等。

本系统主要分为学生模块、超级管理员模块和宿管员模块。

学生模块展示

学生模块主要针对于学生。主要包括登录注册、首页、在线选宿舍、公告查看、报修申请。学生模块的功能主要满足学生在线选宿舍的需求，同时增加宿舍周边功能。

学生用例图如图3.3所示：

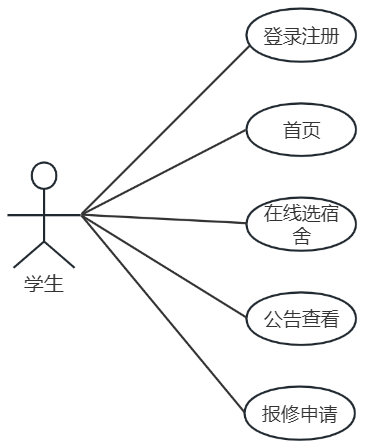


图3.4学生用例图

超级管理员模块

超级管理员进行后台管理，主要有登录注册、基础管理、宿舍管理、报修管理、公告管理、系统管理等功能。

超级管理员用例图如图3.5所示：

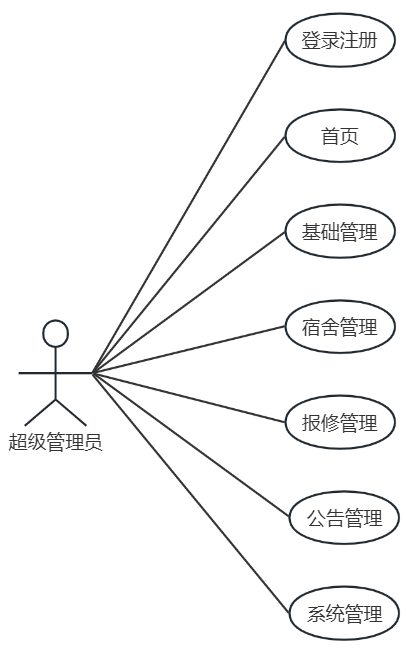


图3.6超级管理员用例图

宿管员模块

宿管员进行后台管理，主要功能有登录注册，首页，宿舍管理、报修管理、公告管理。

宿管员用例图如图3.7所示：



图3.8宿管员用例图

### 非功能性需求分析

（1）易用性

大学生自选管理系统对用户友好，具有良好的UI设计，易于学习、实用、安全、易于个性化定制。

（2）技术可行性

在技术方面，大学生宿舍自选管理系统采用前后端分离的方式[[[8]](#endnote-8)]实现。后端使用Spring Boot[[[9]](#endnote-9)]和MyBatis框架[[[10]](#endnote-10)]实现。数据库使用MySQL实现数据存储，使用Redis实现部分数据的缓存。服务端运行环境采用Tomcat服务器作为系统运行环境。网络通信技术采用HTTP协议，确保信息的安全性问题和可靠性。

（3）经济可行性

大学生自主选择管理系统的运营相当方便，节约了大量人力物力，减少了大量的经济成本。与此同时，该系统的实施增强了学校的宣传力度，提高了学校的影响力，吸引更多学生到该校就读。该系统所使用的技术均来源于网络开源平台，并且开发工具也是免费的，因此并不会增加系统开发的成本。由此看来，大学生宿舍自选管理系统在经济方面可行。

（4）性能

大学生自选管理系统使用Redis实现数据的快速响应，使用锁机制在关键流程和高峰选择时间内，保证系统的稳定性、高效性和准确性。

（5）可伸缩性

大学生自选管理系统能够应对不断增长的用户需求，应该提供有效的扩展性，以支持更多的用户并保持其顺畅运行。

经过以上几点的非功能性需求分析可以看出，大学生宿舍自选管理系统的设计是可行的，值得开发。

## 本章小结

本章从细化分是对大学生宿舍自选管理系的需求予以分析，从软件的功能需求和非功能需求进行分析，以提高对系统独特的设计，为后续的系统设计打好基础。

# 系统设计

## 系统总体设计

该系统在用户角度下进行主体设计，包括超级管理员模块、宿舍管理员模块和学生模块。超级管理员模块用于管理用户信息、学生信息、楼宇信息、宿舍信息、宿舍编号设置、宿舍预选和宿舍分配等。学生模块则用于根据超级管理员所规划的宿舍和床位信息实施宿舍选择。此外，还设计了精美的网页样式布局和数据图表展示，以呈现出良好的视觉效果。 系统功能结构图如图3.1所示：

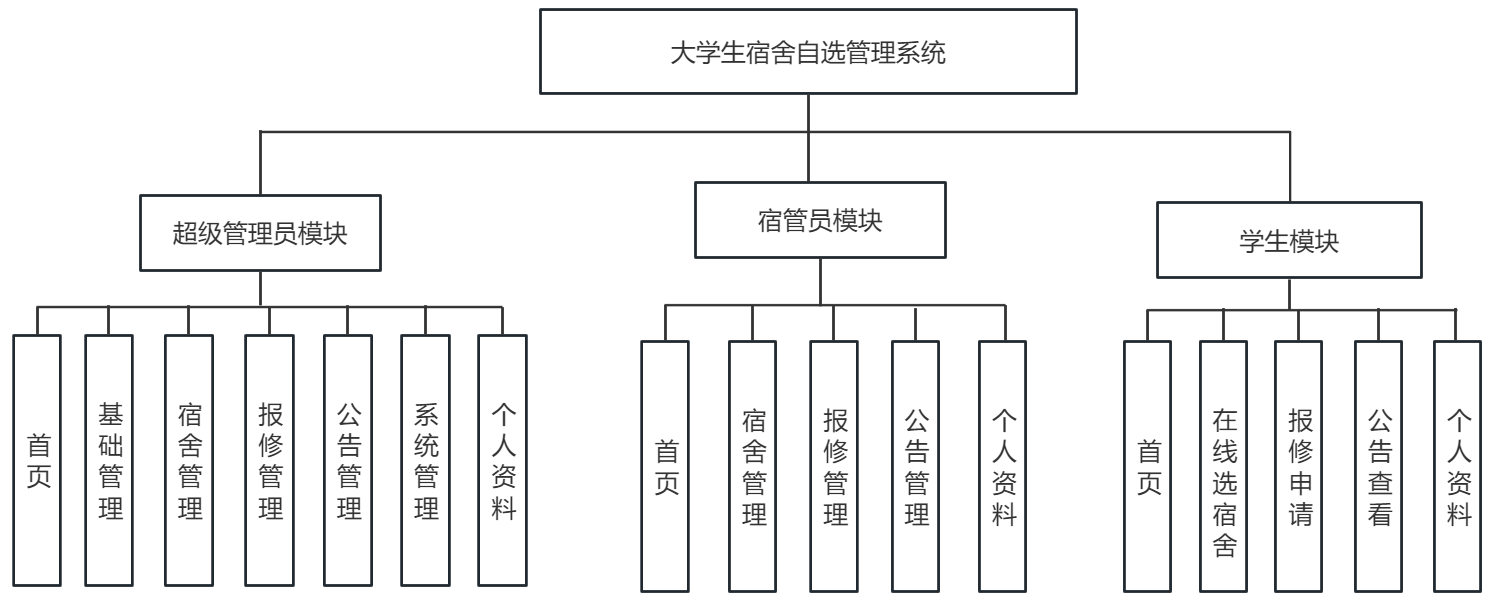


图 4.1 系统功能结构图

## 系统功能模块设计

本系统围绕宿舍展开，学生可在线选择宿舍，管理员可管理学生信息、宿舍信息。

### 用户登录模块设计

用户分为三种。学生、宿管员、超级管理员。用户登录时需要选择自己的身份，以及输入账号、密码，如果账号、密码正确，将跳转到网站的首页，否则将提示登录失败。用户需要重新登录。

大学生宿舍管理系统的用户登录流程如图3.2所示：

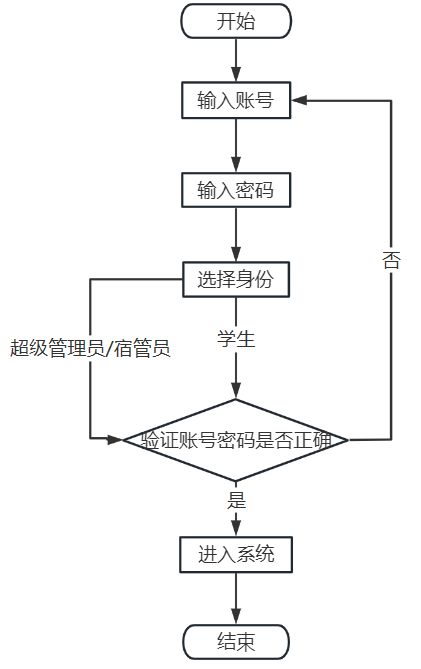


图3.2 系统登录流程图

### 学生选宿舍模块设计

学生登录成功进入本系统后即可使用本系统，本系统的学生模块包括首页、在线选宿舍、报修申请、公告查看、更改个人资料等。

学生选择宿舍流程图如图.3所示：

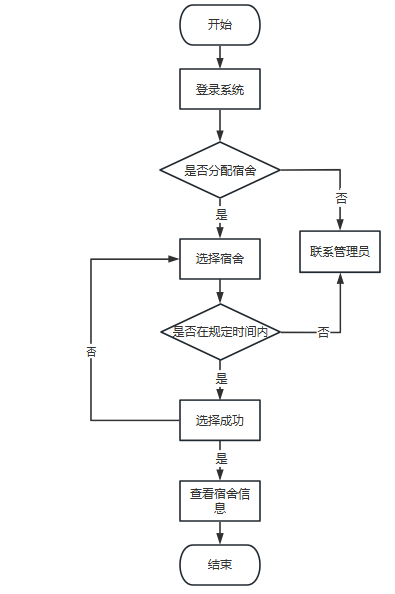


图3.4 学生选择宿舍流程图

### 管理员分配宿舍模块设计

管理员分配宿舍流程图如图3.5所示：

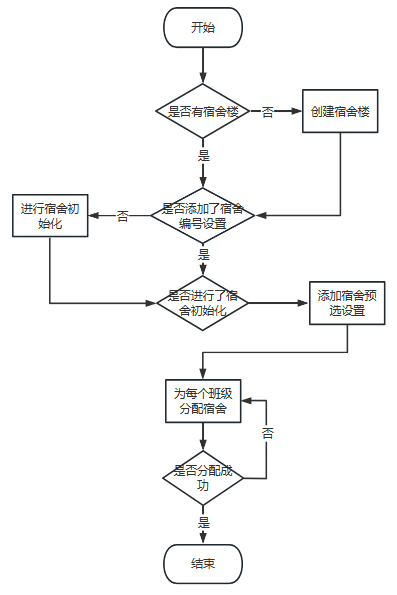


图3.6 管理员分配宿舍流程图

### 用户管理模块设计

用户管理主要的功能是超级管理员可以管理用户信息。用户管理功能模块划分及描述如表3.1所示：

表3.1 用户管理功能模块表

|  |  |
| --- | --- |
| 模块名称 | 模块功能描述 |
| 添加用户 | 超级管理员添加用户信息 |
| 修改用户 | 超级管理员能修改用户信息 |
| 删除用户 | 超级管理员删除所选用户信息 |
| 重置密码 | 超级管理员可以重置用户密码 |

### 学生管理模块设计

学生管理模块的基本管理，主要对学生信息进行管理，导入导出学生数据，其功能模块的具体划分及描述如表3.2所示：

表3.2 学生管理功能模块表

|  |  |
| --- | --- |
| 模块名称 | 模块功能描述 |
| 添加学生信息 | 超级管理员添加学生信息 |
| 修改学生信息 | 超级管理员修改学生的信息 |
| 删除学生信息 | 超级管理员删除学生的信息 |
| 导入学生数据 | 超级管理员导入学生数据 |
| 导出学生数据 | 超级管理员导出学生数据 |
| 重置密码 | 超级管理员重置学生密码 |

### 楼宇管理模块设计

楼宇的基本管理，主要提供超级管理员对楼栋的信息增删改，其功能模块的具体划分及描述如表3.3所示：

表3.3 楼宇管理功能模块表

|  |  |
| --- | --- |
| 模块名称 | 模块功能描述 |
| 添加楼宇信息 | 超级管理员添加楼宇信息 |
| 修改楼宇信息 | 超级管理员修改楼宇的信息 |
| 删除楼宇信息 | 超级管理员删除楼宇的信息 |

### 宿舍管理模块设计

宿舍的基本管理，主要是对宿舍信息的展示以及宿舍信息的增删改，其功能模块的具体划分及描述如表3.4所示：

表3.4 宿舍管理功能模块表

|  |  |
| --- | --- |
| 模块名称 | 模块功能描述 |
| 添加楼层 | 超级管理员添加楼层 |
| 添加宿舍 | 超级管理员添加宿舍 |
| 添加床位 | 超级管理员添加床位 |
| 删除宿舍 | 超级管理员删除宿舍 |
| 删除床位 | 超级管理员删除床位 |
| 分配调整 | 超级管理员对床位进行分配调整 |

### 报修管理模块设计

报修管理是审核学生对于宿舍的一些报修申请，其功能模块划分及描述如表3.6所示：

表3.6报修功能模块表

|  |  |
| --- | --- |
| 模块名称 | 模块功能描述 |
| 添加报修信息 | 学生添加报修信息 |
| 审核报修信息 | 超级管理员对报修信息进行审核 |
| 删除审核信息 | 超级管理员删除已经审核完毕的报修信息 |

## 数据库设计

### 数据库概念结构设计

将大学生宿舍管理系统的数据库概念模型分为成各个实体，其中包括：管理员实体、学生实体、楼宇实体、楼层实体、宿舍实体、床位实体等。

管理员实体的属性有id、用户名、密码、姓名、手机号、类型、性别等。管理员实体图如图3.4所示：

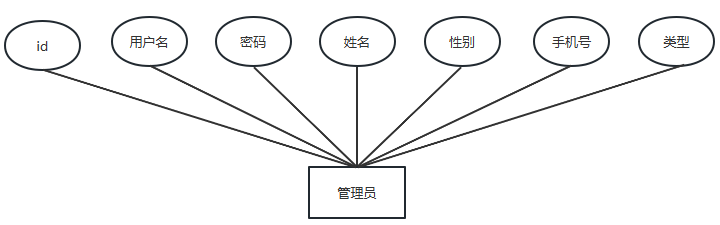


图3.7 管理员实体图

学生实体的属性有：id、学号、密码、姓名、性别、身份证号、手机号等。学生实体图如图3.5所示：

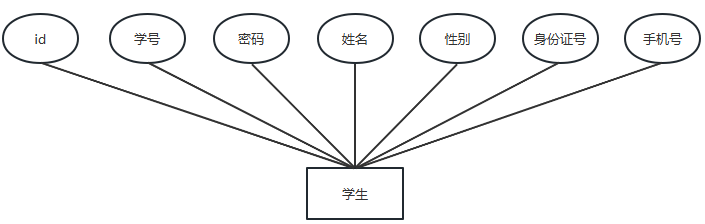


图3.8 学生实体图

楼宇实体的属性有：id、名称、层数、性别、宿管员id。楼宇实体图如图3.6所示：

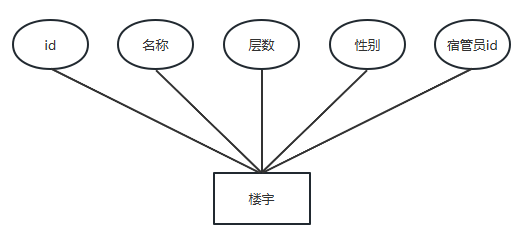


图3.9 楼宇实体图

楼层实体的属性有：id、名称、楼宇id、创建时间等。楼层实体图如图3.7所示：

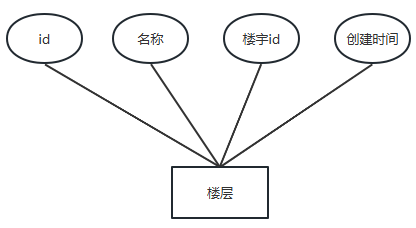


图3.10 楼层实体图

宿舍实体的属性有：id、宿舍号、性别、类型、容量等。宿舍实体图如图3.8所示：

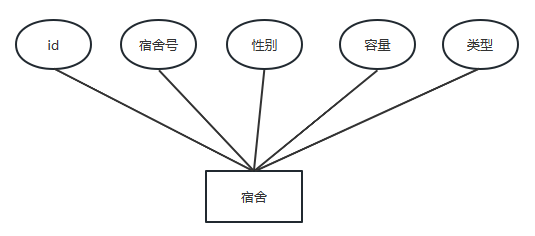


图3.11 宿舍实体图

床位实体的属性有：id、床位号、宿舍id等。床位实体图如图3.9所示：

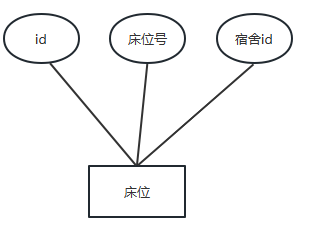


图3.12 床位实体图

通过以上分析可以得到系统管理员、学生、宿舍关系的E-R图，如图3.10所示：

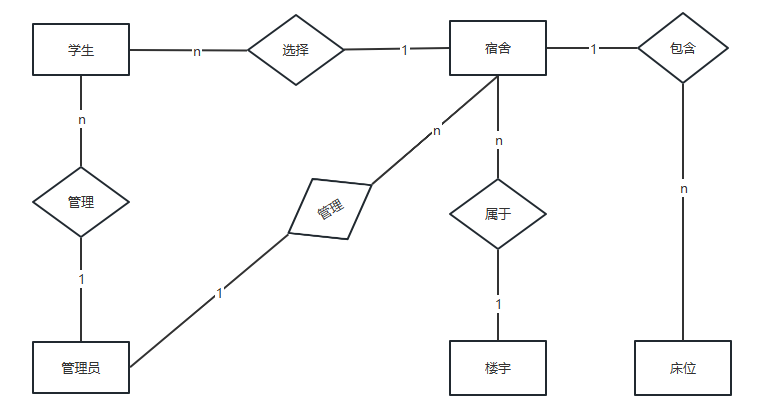


图3.13 系统E-R图

### 数据库逻辑结构设计

本系统数据库包括：学生信息表（tb\_student）、用户信息表（tb\_user）、年级表（tb\_grade）、组织结构表（tb\_org）、楼宇表（tb\_building）、楼层表（tb\_storey）、宿舍表（tb\_dormitory）、

床位表（tb\_bed）、宿舍编号设置表（tb\_dormitory\_set）、宿舍学生关联表（tb\_dormitory\_student）、宿舍预选表（tb\_selection）、预选宿舍班级关联表（tb\_selection\_dormitory）、预选班级关联表（tb\_selection\_joiner）、报修表（tb\_repair）、公告表（tb\_notice）、公告接收表（tb\_notice\_receive）、来访者记录表（tb\_visit）、缺勤表（tb\_absence）、宿舍评价表（tb\_assess）、菜单表（tb\_menu）、用户菜单关联表（tb\_user\_menu）。各表单的数据结构如下所示：

（1）学生表：学生表存储学生id、学号、姓名、身份证号等信息，如表3.7所示：

表. 学生信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 学生id |
| sto\_no | varchar | 20 |  | 学号 |
| name | varchar | 20 |  | 姓名 |
| id\_card | varchar | 50 |  | 身份证号 |
| sex | int | 10 |  | 性别  （0-女/1-男） |
| phone | varchar | 50 |  | 学生手机号 |
| password | varchar | 50 |  | 密码 |
| clazz\_id | Int | 10 |  | 学生所在班级id |
| file | varchar | 100 |  | 头像名称 |

（2）用户信息表：用户信息表存储用户id、用户名、密码等信息，如表3.8所示：

表3.8 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 用户id |
| user\_name | varchar | 100 |  | 用户名 |
| password | varchar | 100 |  | 密码 |
| phone | varchar | 100 |  | 手机号 |
| name | varchar | 100 |  | 姓名 |
| type | int | 10 |  | 用户类型   1. 超级管理员   /1-宿管员） |
| status | int | 10 |  | 用户状态  （0-禁用/1-正常） |
| file | varchar | 100 |  | 头像名称 |

（3）年级表：年级表存储年级id、年纪名称、类型等信息，如表3.9 所示：

表3.10 年级表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 年级id |
| name | varchar | 50 |  | 年级名称 |

（4）组织结构表：组织结构表存储组织id、组织名称、类型等信息，如表3.11所示：

表3.12 组织结构表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 组织id |
| name | varchar | 100 |  | 组织名称 |
| type | int | 10 |  | 类型  （1-学院/2-系/3-专业/4-班级） |
| grade\_id | int | 10 |  | 年级id |
| parent\_id | int | 10 |  | 父级组织id |
| status | int | 10 |  | 状态  （0-正常/1-不正常） |
| deleted | int | 10 |  | 删除标示  （0-删除、1-未删除） |

（5）楼宇表：楼宇表存储楼宇id、层数、性别等信息，如表3.11所示：

表3.13 楼宇表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 楼宇id |
| storey\_num | int | 10 |  | 层数 |
| sex | int | 10 |  | 性别  （0-女/1-男） |
| user\_id | int | 10 |  | 宿管员id |
| remark | varchar | 100 |  | 备注 |

（6）楼层表：楼层表存储楼层id、楼层名称、创建时间等信息，如表3.12所示：

表3.14 楼层表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 楼层id |
| name | varchar | 100 |  | 名称 |
| building\_id | int | 10 |  | 楼宇id |
| create\_time | datetime |  |  | 创建时间 |

（7）宿舍表：宿舍表存储宿舍id、宿舍号、性别、类型等信息，如表3.13所示：

表3.15 宿舍表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 宿舍id |
| no | varchar | 100 |  | 宿舍号 |
| sex | int | 10 |  | 性别  （0-女/1-男） |
| type | int | 10 |  | 类型  （4/6/8人间） |
| capacity | int | 10 |  | 容量 |
| storey\_id | int | 10 |  | 楼层id |
| building\_id | int | 10 |  | 楼宇id |
| version | int | 10 |  | 版本号 |

（8）床位表：床位表存储床位相关信息，如表3.16所示：

表3.17 床位表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 床位id |
| bno | varchar | 100 |  | 床位号 |
| dormitory\_id | int | 10 |  | 宿舍id |

（9）宿舍编号设置表：宿舍编号设置表宿舍编号设置的相关信息，如表3.18所示：

表3.19 宿舍编号设置表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 宿舍编号设置id |
| prefix | varchar | 50 |  | 前缀 |
| start | int | 10 |  | 开始值 |
| end | int | 10 |  | 结束值 |
| building\_id | int | 10 |  | 楼宇id |
| storey\_id | int | 10 |  | 楼层id |
| capacity | int | 10 |  | 容量 |

（10）宿舍学生关联表：宿舍学生关联表存储宿舍和学生的关联的相关信息，如表3.20所示：

表3.21 宿舍学生关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 关联id |
| bed\_id | int | 10 |  | 床位id |
| dormitory\_id | int |  |  | 宿舍id |
| student\_id | int | 10 |  | 学生id |
| checkin | datetime |  |  | 选择时间 |
| status | int |  |  | 状态  （0-待入住/1-已入住） |

（11）宿舍预选表：宿舍预选表存储宿舍预选的相关信息，如表3.22所示：

表3.23 宿舍预选表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 宿舍预选id |
| name | varchar | 100 |  | 名称 |
| start\_time | datetime |  |  | 开始时间 |
| end\_time | datetime |  |  | 结束时间 |
| remark | varchar | 100 |  | 备注 |

（12）预选宿舍班级关联表：预选宿舍班级关联表存储预选宿舍班级关联的相关信息，如表3.24所示：

表3.25 预选宿舍班级关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
|  | varchar | 100 |  | 床位号 |
| dormitory\_id | int | 10 |  | 宿舍id |
| clazz\_id | int | 10 |  | 班级id |

（13）预选班级关联表：预选班级关联表存储预选id、班级id等信息，如表3.26所示：

表3.27 预选班级关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 关联id |
| selection\_id | int | 10 |  | 预选id |
| clazz\_id | int | 10 |  | 班级id |

（14）报修表：报修表报修id、报修描述、宿舍id、审核人、审核意见等信息，如表3.28所示：

表3.29 报修表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 报修id |
| description | varchar | 100 |  | 保修描述 |
| dormitory\_id | int | 10 |  | 宿舍id |
| student\_id | int | 10 |  | 申请报修的学生id |
| processor | varchar | 20 |  | 审核人 |
| processor\_idea | varchar | 100 |  | 审核意见 |
| status | int | 10 |  | 状态  （0-待审核/1-审核通过/2-未通过） |
| create\_date | datetime |  |  | 申请时间 |

（15）公告表：公告表存储公告id、标题、内容、创建人等信息，如表3.30所示：

表3.31 公告表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | id |
| title | varchar | 100 |  | 标题 |
| content | varchar | 100 |  | 内容 |
| user\_id | int | 10 |  | 创建人 |
| type | int | 10 |  | 类型  （1-公告/2-通知） |
| create\_time | datetime |  |  | 创建时间 |

（15）公告接收表：公告接收表存储公告接收的相关信息，如表3.32 所示：

表3.33 公告接收表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | id |
| notice\_id | int | 10 |  | 公告id |
| building\_id | int | 10 |  | 楼宇id |

（16）来访者记录表：来访者记录表存储来访者记录的相关信息，如表3.34所示：

表3.35 来访者记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 记录id |
| visitor | varchar | 50 |  | 来访人 |
| phone | varchar | 10 |  | 手机号 |
| id\_card | varchar | 100 |  | 身份证号 |
| sex | int | 10 |  | 性别  （0-女/1-男） |
| student\_id | int | 10 |  | 学生id |
| building\_id | int | 10 |  | 楼宇id |
| visit\_time | datetime |  |  | 来访时间 |
| leave\_time | datetime |  |  | 离开时间 |

（17）缺勤表：缺勤表存储宿舍缺勤的相关信息，如表3.36 所示：

表3.37 缺勤表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 缺勤id |
| student\_id | int | 10 |  | 学生id |
| dormitory\_id | int |  |  | 宿舍id |
| building\_id | int | 10 |  | 楼宇id |
| status | int | 10 |  | 状态  （0-离宿中/1-已回/2-缺勤） |
| state\_time | datetime |  |  | 开始时间 |
| end\_time | datetime |  |  | 结束时间 |

（18）宿舍评价表：宿舍评价宿舍评价的相关信息，如表3.38所示：

表3.39 宿舍评价表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 宿舍评价id |
| description | varchar | 100 |  | 评价内容 |
| dormitory\_id | int | 10 |  | 宿舍id |
| user\_id | int | 10 |  | 评价人id |
| type | int | 10 |  | 等级  （0-差/1-良好/2-优秀） |
| create\_time | datetime |  |  | 评价时间 |

（19）菜单表：菜单表存储菜单id、标题、图标、地址等信息，如表3.40所示：

表3.41 菜单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| id | int | 4 | 是 | 菜单id |
| title | varchar | 100 |  | 标题 |
| icon | varchar | 20 |  | 图标 |
| href | varchar | 20 |  | 地址 |
| type | int | 10 |  | 类型  （0-超级管理员；宿管员/1-学生） |
| parent\_id | int | 10 |  | 父级菜单id |

（20）用户菜单关联表：用户菜单关联表存储用户id、菜单id的信息，如表3.42所示：

表3.43 用户菜单关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 大小 | 是否主键 | 描述 |
| user\_id | int | 10 |  | 用户id |
| menu\_id | int | 10 |  | 菜单id |

## 本章小结

本章详细介绍了该系统的整体方案，并根据大学生宿舍自选管理系统的需求分析，设计了主要的功能模块。在数据库设计中，首先利用实体图来描述各实体，并对其进行概念结构设计与逻辑架构设计，最后对大学生宿舍自选管理系统的数据进行完善。

# 系统实现

## 功能模块设计实现

### 学生模块首页页面实现

本系统的学生首页由两个部分组成，包括宿舍概览和当前宿舍选择。该页面在main.html中实现。

界面展示如下图所示：



图 . 学生模块首页展示

宿舍概览使学生在选择完宿舍后清晰准确地查看所选宿舍的相关信息。当前宿舍选择则可让大学生浏览其所选宿舍的成员信息。

### 学生在线选宿舍页面实现

学生在线选宿舍页面分为两部分，即宿舍信息和床位信息。此页面在stu/select.html中编写。宿舍信息包括宿舍所在的楼栋、楼层和门牌号，以及该宿舍的可容纳人数和已入住人数。床位信息包括床位号和可供学生选择的复选框。已入住人数由后端计算得出，其余信息由后端查询数据库进行获取。

学生在线选宿舍页面展示如图所示：



图. 在线选宿舍页

在线选宿舍的页面是我们系统中最重要的页面之一。该页面的主要目的是让学生能够基于学校分配的宿舍自由选择。页面清晰地展示了宿舍和床位的信息，并提供复选框供学生进行选择。未被选择的床位是灰色的，但一旦学生成功选择了床位，它就会变为绿色。这样学生就可以清楚地知道哪些床位可以选择，哪些已经被选择。为此，我们通过将学生已选择的床位id与所有床位id进行比较来确定何时显示绿色图片。如果它们一致，我们将显示绿色图片，否则将显示灰色图片。

部分前端代码如图所示：



图. 学生选择床位图片切换部分前端代码

部分后端代码如所示：



图. 学生选择床位后端代码

超级管理员可以通过添加预选配置来规划哪些班级在何时选择宿舍。为每个预选宿舍的班级分配宿舍，然后该班级的学生即可登录该系统进行宿舍选择。但是学生需要留意选择宿舍的时间，一旦时间过期，就不能再进行选择。如果超级管理员没有为特定班级设置预选配置，则该班级的学生无法登录该系统进行宿舍选择。

宿舍预选页面展示如所示：



图. 宿舍预选页面展示

分配宿舍页面如图所示：

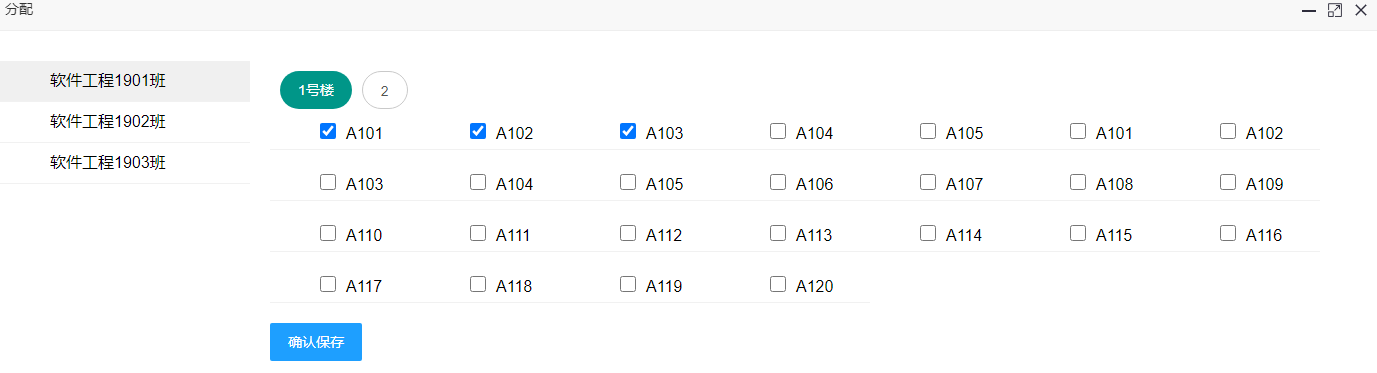


图4.6 分配宿舍页面展示

前端部分代码如所示：



图4.7 宿舍分配前端部分代码

后端部分代码如所示：

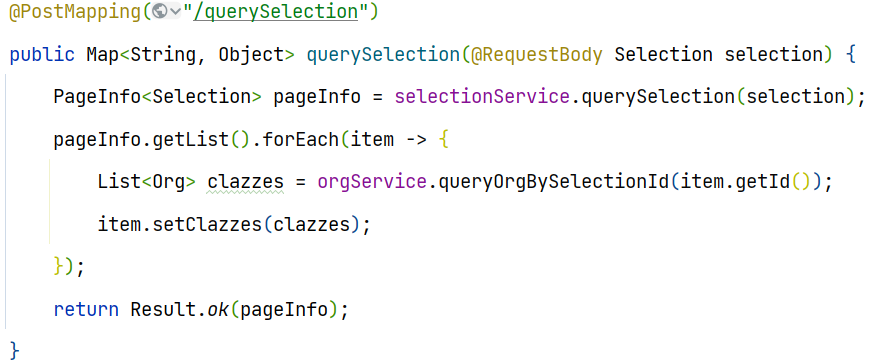


图. 宿舍分配后端部分代码

### 学生离宿记录页面实现

该页面由一个表格组成，表格有姓名，宿舍，开始时间，结束时间，状态，备注几个属性。清晰的展示了学生的离宿记录。学生还可更改离宿记录，方便学生在回宿后进行登记。

学生离宿记录页面如所示：



图4.9 学生离宿页面展示

学生的离宿记录管理作用在于对当天晚上离宿的学生的管理。其由超级管理员或者宿管员添加。也可对学生的离宿记录进行编辑和删除。

学生离宿页面前端部分代码如所示：



图4.10 学生离宿前端部分代码

学生离宿页面后端部分代码如图所示：



图. 学生离宿后端部分代码

### 管理员首页界面实现

管理员首页界面由4个部分组成的，其中包括数据统计、系统公告、报表统计、使用率统计。数据统计模块是一个表格，展示楼宇、宿舍数量、入住人数、闲置数量、使用率的数据。而报表统计是对这几个数据呈折线图展示，更加形象。其中宿舍使用率更是使用了饼图对其展示。用户很清晰的就可以看到几栋楼的使用率的情况。用户体验贼好。系统公告模块是对发布的所有的公告的一个展示。

超级管理员首页所示：



图4.12 超级管理员首页展示

管理员首页前端部分代码展示如所示：



图. 超级管理员首页前端部分代码

超级管理员首页后端部分代码如图所示：

图 . 超级管理员首页部分部分代码

### 用户管理页面实现

用户管理页包括用户列表页面添加用户页面、和用户编辑页面，分别在user/list.html，user/add.html，user/edit.html中编写。

用户管理页如所示：



图4.15 用户管理页面展示

超级管理员可以在管理后台添加该系统的用户，添加用户的时候可以给用户分配角色，有超级管理员和宿管员两种角色。选择角色同时给相应的角色添加相应的菜单权限。宿管员没有用户管理这个权限。超级管理员也可对用户进行编辑、删除、重置密码。

添加用户页面如所示：



图4.16 添加用户页面展示

用户管理页前端部分代码如所示：



图.用户管理页面前端部分代码

用户管理页面后端部分代码如图所示：

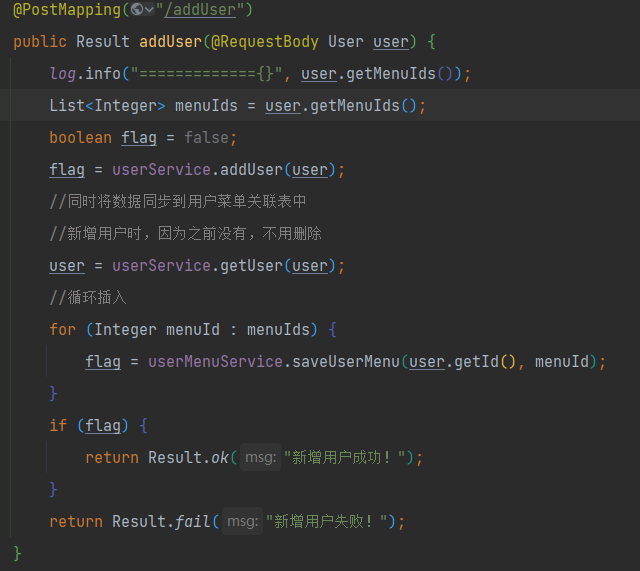


图4.18 用户管理页面后端部分代码

### 机构管理页面实现

机构管理页面包括组织列表页面、添加组织页面和编辑组织页面。分别在org/list.html、org/add.html和org/edit.html中编写。

机构管理页面如图所示：

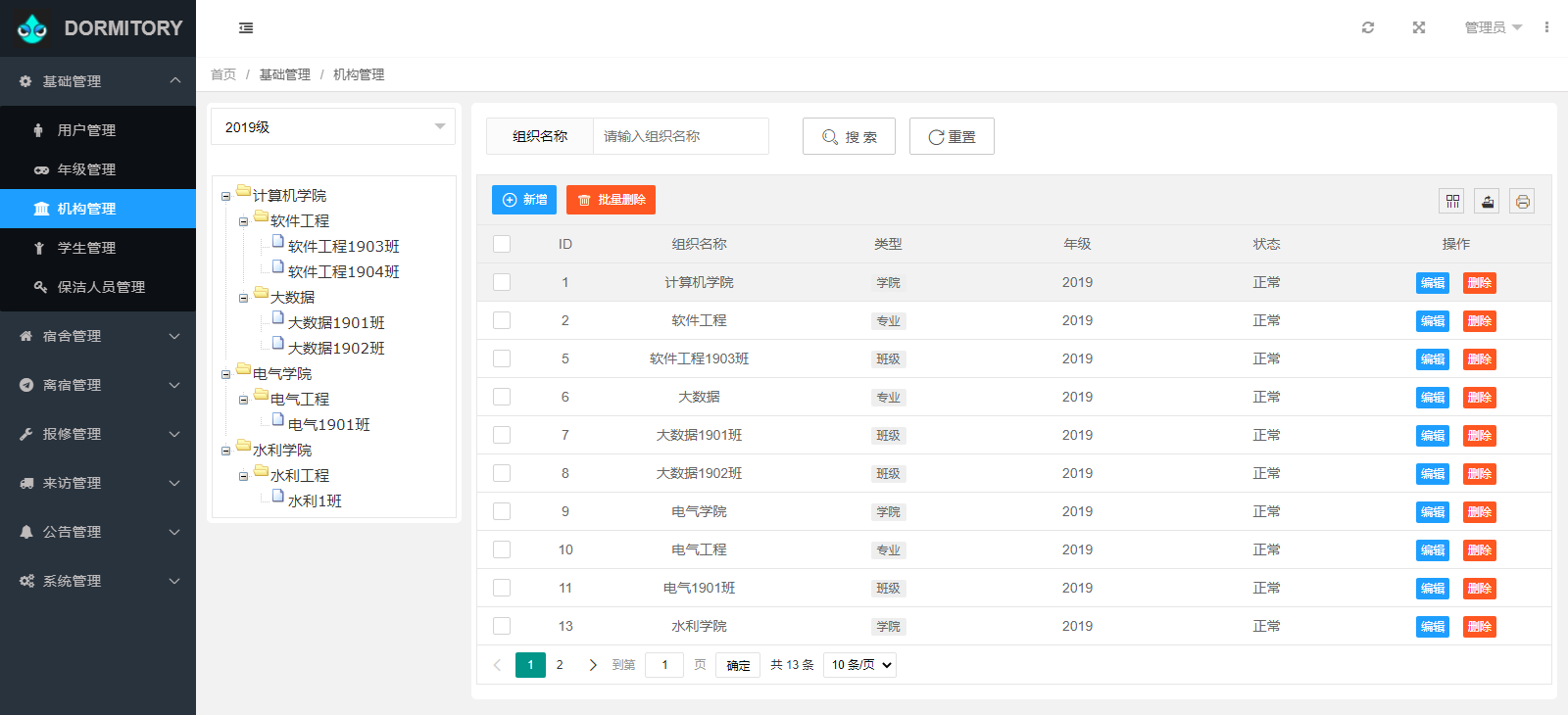


图. 机构管理页面展示

超级管理员可以在机构管理页面查看结构列表，由树形结构，也有表格结构。两种方式呈现，展示更加丰满。可以根据年级选择对应年级的结构列表，可以根据机构名称搜索相应的机构。超级管理员可以添加机构、编辑机构和删除机构。

添加机构页面如图所示：

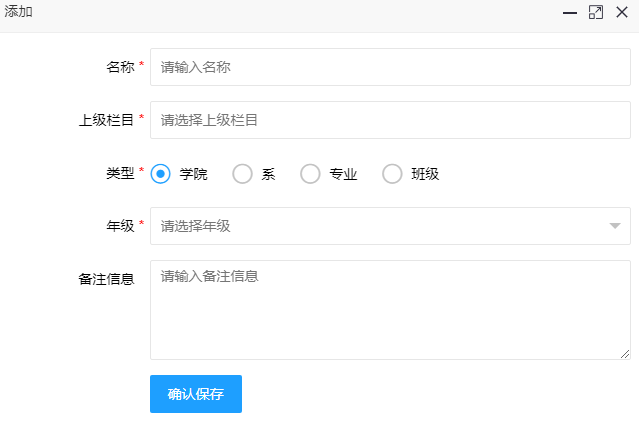


图. 机构管理添加页面展示

机构管理前端部分代码如图所示：



图.机构管理页面前端部分代码

后端主要代码如图所示：

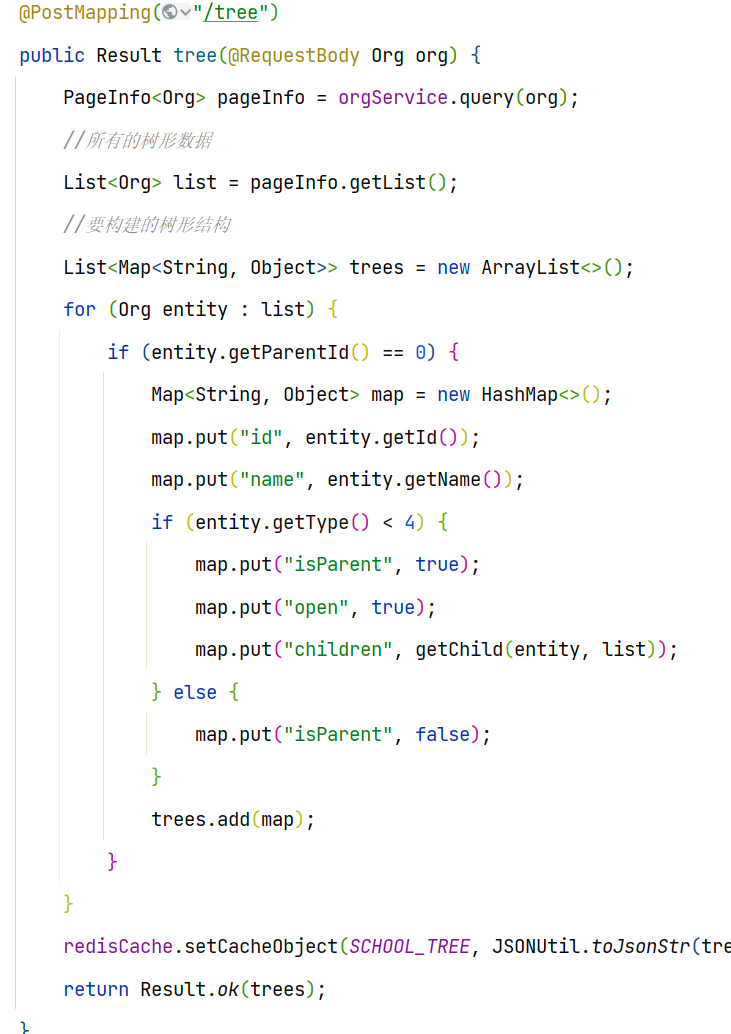


图.机构管理页面后端部分代码

### 学生管理页面实现

学生管理页面由学生列表页面、添加学生页面、编辑学生页面组成。分别在student/list.html、student/add.html、student/edit.html页面编写。

学生管理页面如图所示：

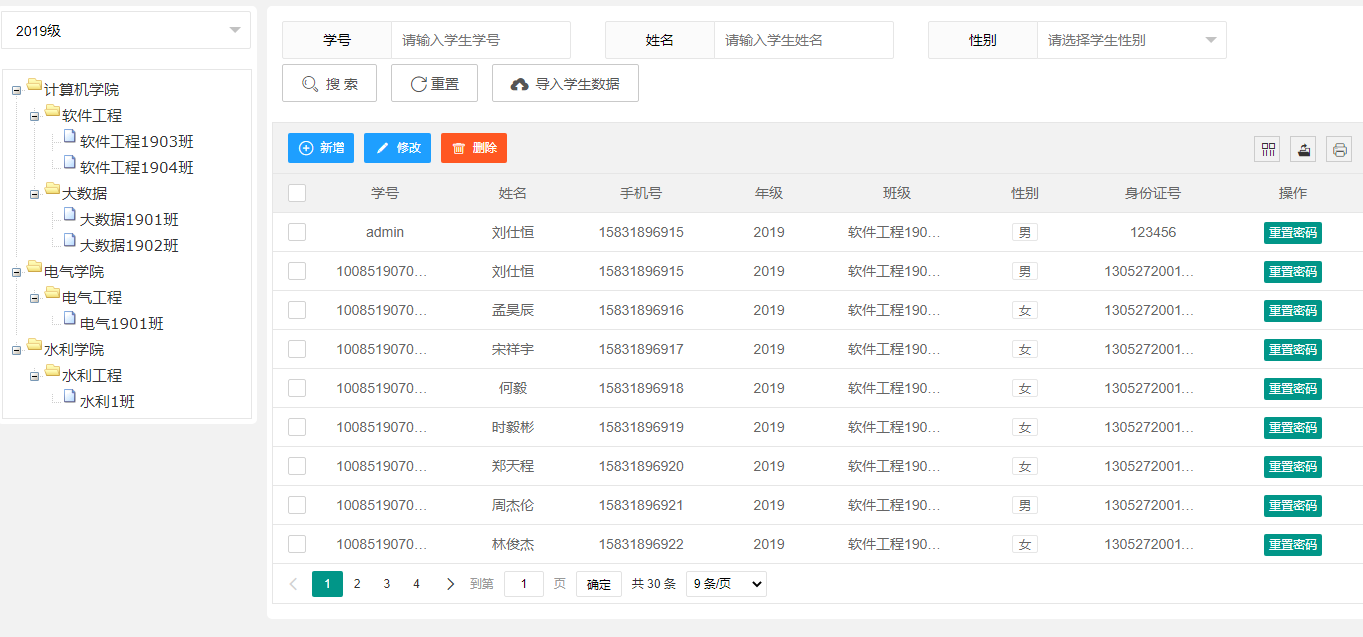


图. 学生管理页面展示

超级管理员在该页面可以查看学生的列表信息，跟机构管理相似，也有表格结构跟树形结构两种，可以添加学生、编辑学生、删除学生、重置学生的密码的功能。还有导入导出学生数据的功能，可以根据学生的学号、姓名、性别去搜索学生。

添加学生的页面如图所示：

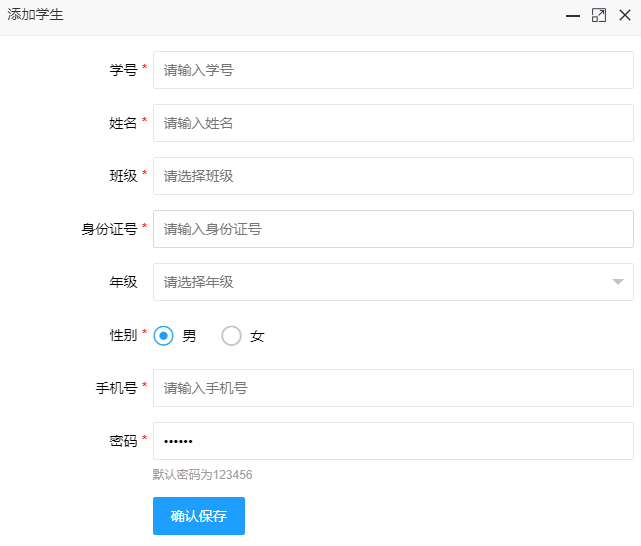


图. 学生管理添加学生页面展示

### 宿舍编号设置页面的实现

宿舍编号设置页面由宿舍编号列表页面、新增宿舍编号页面和编辑宿舍编号页面组成。分别在dormitoryset/list.html、dormitoryset/add.html和dormitoryset/edit.html中编写。宿舍 宿舍编号设置页面如图所示：



图. 宿舍编号设置页面展示

超级管理员在该页面可以查看宿舍编号设置的详情。最上方是宿舍管理显示的是当前选择的楼宇，有高亮显示默认是第一栋楼。左侧是该楼宇的哪一层。默认是第一层。中间部分是宿舍编号设置的列表，超级管理员可以新增宿舍编号设置、编辑宿舍编号设置、删除宿舍编号设置。最重要的一个功能是宿舍初始化，在确认宿舍编号设置无误后，可以点击这个按钮实现宿舍编号的初始化。

新增宿舍编号的页面如所示：

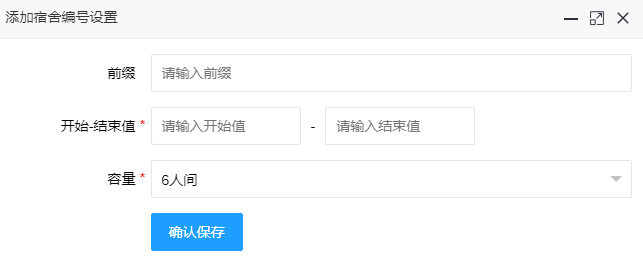


图. 宿舍管理新增宿舍页面展示

宿舍初始化的前端部分代码如图所示：



图 .宿舍初始化的前端部分代码

宿舍初始化的后端部分代码如图所示：



图.宿舍初始化的后端部分代码

### 宿舍管理页面的实现

宿舍管理页面由宿舍列表页面、添加宿舍页面、添加床位页面和床位页面分配调整组成，分别在dormitory/list.html、dormitory/addDormitory.html、dormitory/addBed.html和dormitory/edit.html中编写。

宿舍管理页面如图所示：

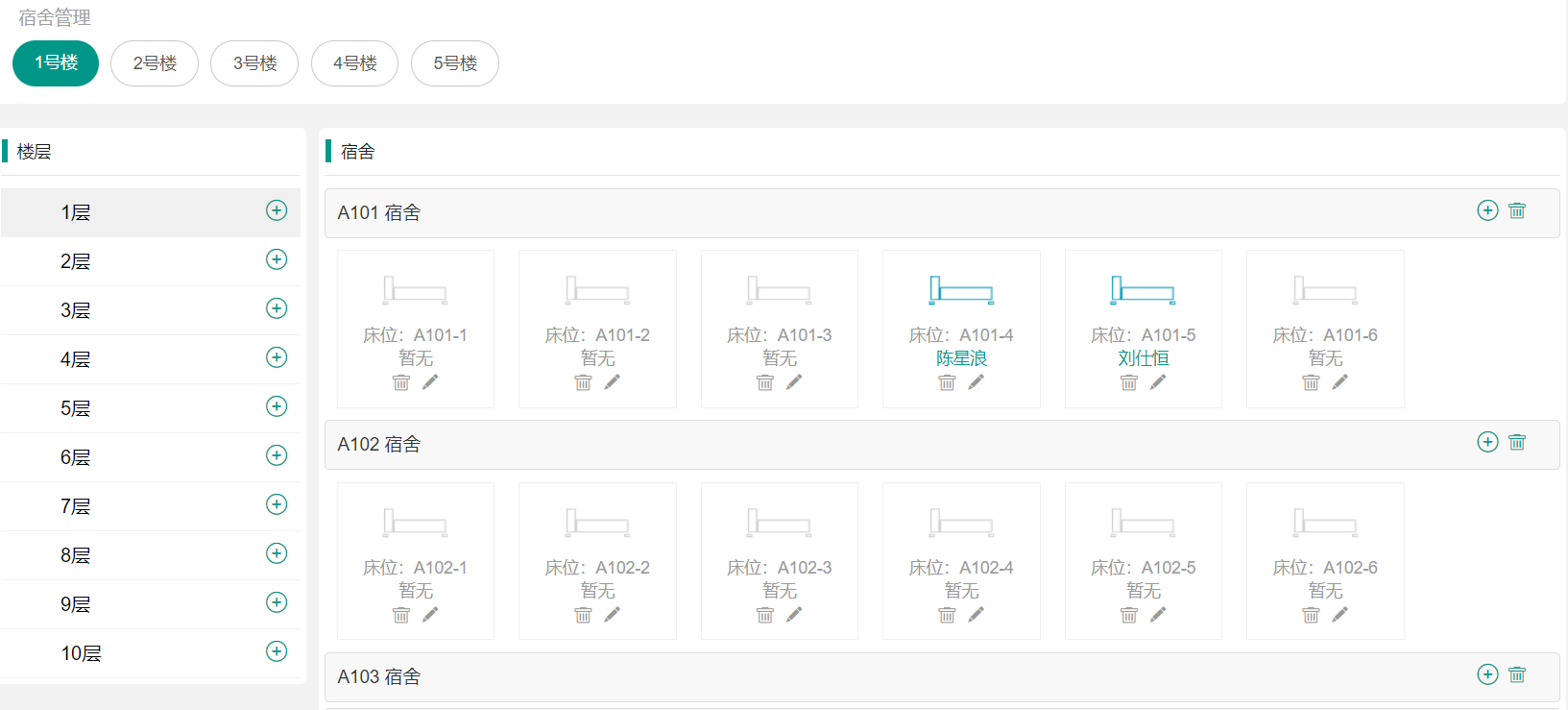


图. 宿舍管理页面展示

本页面超级管理员跟宿管员都有权限，区别是超级管理员可以管理所有楼宇，而宿管员只能管理自己负责的楼宇。宿舍管理页面上侧展示的楼宇的数据，左侧展示的是对应楼宇的每一层，中间部分是对对应楼宇对应楼层宿舍列表的展示。点击对应宿舍显示的是该宿舍的床位信息，超级管理员可以添加宿舍、删除宿舍、添加床位、删除床位、床位的分配调整的操作。

宿舍管理页面前端部分代码如图所示：



图.宿舍管理页面前端部分代码

宿舍管理页面后端部分代码如图所示：



图. 宿舍管理页面后端部分代码

### 公告管理页面的实现

公告管理页面由公告列表页面、添加公告页面、编辑公告页面组成。分别在notice/list.html、notice/add.html和notice/edit.html中编写。

公告管理页面如图所示：



图. 公告管理页面展示

公告管理页面超级管理员跟宿舍员都可查看、编辑、删除。区别是，超级管理员的权限是所有楼宇，而宿管员的权限是自己负责的楼宇。超级管理员跟宿管员在公告管理页面可对新增公告、编辑公告和删除公告。新增公告时，需要注意选择公告的类型：公告或者通知，和勾选通知范围。

添加公告的页面如图所示：



图. 公告管理添加公告页面展示

添加公告的前端部分代码如图所示：



图. 添加公告的前端部分代码

添加公告的后端部分代码如图所示：



图. 添加公告的后端部分代码

## 本章小结

本章主要内容为对大学生宿舍自选管理系统的功能模块的实现进行展示，简单的呈现了系统主要功能界面的效果图和部分程序代码。

# 系统测试

## 测试环境

大学生宿舍自选管理系统主要用黑盒测试方法进行测试，黑盒测试主要针对于系统整体功能进行测试，检测系统功能是否能够实现预期的效果，对系统不完善或不严谨的地方进行完善和修改。测试主要在Window11操作系统、Tomcat服务器以及Google Chrome浏览器的环境下进行测试。

## 功能测试

### 超级管理员/宿管员/学生登录功能测试

超级管理员/宿管员/学生登录功能测试主要是对超级管理员/宿管员/学生进行系统登录测试。主要的测试内容为：分别输入超级管理员/宿管员/学生账号和密码，测试该系统是否正常进行。

超级管理员/宿管员/学生登录模块测试表，如表5.1所示：

表. 超级管理员/宿管员/学生登录测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 是否提示账号输入不正确 | 输入错误的账号 | 请输入正确的账号和密码 | 达到预期的效果，测试通过 |
| 2 | 是否提示密码输入不正确 | 输入错误的密码 | 请输入正确的账号和密码 | 达到预期的效果，测试通过 |
| 3 | 是否提示密码输入不正确 | 账号和密码为空 | 请输入正确的账号和密码 | 达到预期的效果，测试通过 |
| 4 | 能否成功登录 | 填写正确信息，选择身份，点击登录 | 登录成功 | 达到预期的效果，测试通过 |

### 楼宇管理功能测试

楼宇管理功能测试主要对超级管理员是否能对楼宇信息的添加、修改、删除操作进行测试。主要测试内容为：添加、修改、删除楼宇信息，测试系统是否正常进行相关操作。

楼宇管理功能测试表，如表.3所示：

表. 楼宇管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 楼宇列表是否正常显示 | 查看系统是否显示已有的楼宇信息 | 正常显示 | 达到预期的效果，测试通过 |
| 2 | 能否添加楼宇 | 点击“添加楼宇”，填写正确信息并保存 | 添加成功 | 达到预期的效果，测试通过 |
| 3 | 能否修改楼宇信息 | 填写正确信息，提交修改信息 | 修改成功 | 达到预期的效果，测试通过 |
| 4 | 能否删除楼宇点 | 选中相关楼宇，点击删除 | 删除成功 | 达到预期的效果，测试通过 |

### 宿舍编号设置管理功能测试

宿舍编号设置管理功能测试主要对超级管理员是否能对宿舍编号设置的添加、修改、删除、宿舍初始化操作进行测试。

宿舍编号设置管理功能模块测试，如表5.5所示：

表5.6 宿舍编号设置管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 宿舍编号设置是否正常显示 | 查看是否显示已有的宿舍编号设置列表 | 正常显示 | 达到预期的效果，测试通过 |
| 2 | 能否添加宿舍编号设置 | 点击“新增设置”，填写并保存 | 添加成功 | 达到预期的效果，测试通过 |
| 3 | 能否修改宿舍编号设置 | 填写正确信息，提交修改信息 | 修改成功 | 达到预期的效果，测试通过 |
| 4 | 能否删除宿舍编号设置 | 选中宿舍编号设置，点击删除 | 删除成功 | 达到预期的效果，测试通过 |
| 5 | 能否进行宿舍初始化 | 选中宿舍编号设置。点击宿舍初始化 | 初始化成功 | 达到预期的效果，测试通过 |

### 宿舍管理功能测试

宿舍管理功能测试主要对超级管理员/宿管员能否对添加宿舍、删除宿舍、添加床位、删除床位，床位分配调整操作进行测试。

宿舍管理功能模块测试，如表5.7所示：

表5.8 宿舍管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 楼宇、楼层、宿舍、床位能否正常显示 | 查看是否显示已有的信息 | 正常显示 | 达到预期的效果，测试通过 |
| 2 | 能否添加宿舍 | 点击“添加宿舍”，填写并保存 | 添加成功 | 达到预期的效果，测试通过 |
| 3 | 能否删除宿舍 | 选中相关宿舍，点击删除 | 删除成功 | 达到预期的效果，测试通过 |
| 4 | 能否添加床位 | 点击“添加床位”，填写并保存 | 添加成功 | 达到预期的效果，测试通过 |
| 5 | 能否删除床位 | 选中相关床位，点击删除 | 删除成功 | 达到预期的效果，测试通过 |
| 6 | 能否进行床位分配调整 | 选择相关床位，点击“分配调整”，填写并保存 | 调整成功 | 达到预期的效果，测试通过 |

### 宿舍分配功能测试

宿舍分配功能测试主要对超级管理员是否能对已经设置宿舍预选的班级的分配宿舍的操作进行测试。其功能模块测试如表5.9所示：

表5.10 宿舍分配测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 分配班级和宿舍能否正常显示 | 查看是否显示已有的班级和宿舍信息 | 正常显示 | 达到预期的效果，测试通过 |
| 2 | 能否分别为每个班级进行宿舍分配 | 选择相应班级，并选择为该班级分配的宿舍并提交 | 分配成功 | 达到预期的效果，测试通过 |

### 在线选宿舍功能测试

在线选宿舍功能测试主要对学生是否能对自己所在班级分配的宿舍中进行宿舍的选择的操作进行测试，其功能模块测试，如表5.11所示：

表5.12 在线选宿舍测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 测试内容 | 测试步骤 | 期望结果 | 结论 |
| 1 | 是否显示宿舍 | 查看是否显示已分配的宿舍信息 | 正常显示 | 达到预期的效果，测试通过 |
| 2 | 能否进行宿舍的选择 | 选中相应的床位进行选择 | 选择成功 | 达到预期的效果，测试通过 |

## 本章小结

本章的主要内容是对系统功能测试内容进行了相关阐述，各测试结果显示功能使用正常，未出现与预期不一致的结果，测试结果表明大学生宿舍自选管理系统的设计与实现符合用户需求，系统运行正常。

# 结论

本论文立足与大学生宿舍自选管理系统的开发，最重要的是增强的了大学生选择宿舍的自选权和提高宿舍管理的工作效率和准确性，避免了纯手工管理中可能存在的错误和遗漏。

通过宿舍自选功能，大学生可以更加自主地选择自己喜欢的宿舍，增加了宿舍住宿的满意度，提升了住宿体验。通过宿舍管理系统，可以更加方便地管理宿舍的报修、公告的发布等，保证了住宿信息的共享。

综上所述，大学生宿舍自选管理系统的开发将如实体现大学生宿舍的管理的需求，同时也提升了宿舍住宿的满意度，推动大学生宿舍管理方式的数字化，有望在未来得到更广泛的应用和推广。

但由于自身的研发经验和技术水平的欠缺，使得本系统的设计还存在着许多有待完善和提高的地方。在未来，我会适时地关注本系统的功能点，并不断地进行更新和维护，并在必要的时候添加一些新的功能，使页面更加好看，功能逐步健全[[[11]](#endnote-11)]，更全面地满足用户需求。

# 参考文献

1. [1] 黄强.基于Web的高职院校学生宿舍管理系统设计与结构分析[J].无线互联科技,2022(10):78-80 [↑](#endnote-ref-1)
2. [] 陈佳玲.高校学生宿舍管理信息系统设计与实现[D].四川:电子科技大学,2011 [↑](#endnote-ref-2)
3. [] 郑智方,李彬.探究MySQL的运用实例-对数据库的宏观把握[J] .信息科技,2022(02):129 [↑](#endnote-ref-3)
4. [] 李运平,吴素芹,刘艳华.基于Web的毕业设计管理系统设计与实现[J].软件导刊,2016(11):123-125 [↑](#endnote-ref-4)
5. [] 吴丹林,李梦雅.B/S模式宿舍管理系统设计与实现[J].电脑编程技巧与维护,2021(01):98-99+105 [↑](#endnote-ref-5)
6. [] 唐瑞明,李论,陈珊.高校宿舍管理系统综述[J].电子技术与软件工程,2020(02):64-66 [↑](#endnote-ref-6)
7. [] 任初明,徐延宇,付清香.基于学生视角的大学校园文化认同调查[J].教育理论与实践,2022(12) [↑](#endnote-ref-7)
8. [] 吴昌政.基于前后端分离技术的Web开发框架设计[D].南京:南京邮电大学,2020 [↑](#endnote-ref-8)
9. [] 于昕,廖晨伶,周卫丽.基于MVC的软件架构重构与优化研究[J].吉林化工学院学报,2021(07):49-52 [↑](#endnote-ref-9)
10. [] 欧阳宏基,葛萌,程海波.MyBatis框架在数据持久层中的应用研究[J].微型电脑应用,2020(03) [↑](#endnote-ref-10)
11. [] 何煜琳.基于网站制作的Web前端开发优化[J].软件,2021(02):112-114

    Influence Of Virtual Imaging Technology Based On Html5 Technology On Digital Painting[J]Microprocessors And MicrosystemsVolume 82, 2021.

    # 致谢

    在本次毕业论文的完成过程中，我得到了很多人的协助和支持，在此我要向他们表达最真挚的谢意。

    先要感谢我的指导老师，他在我有难题的时候给予了耐心的指导。他不仅在论文撰写上提供了许多宝贵的意见和建议，还常常与我探讨专业问题，使我受益匪浅。

    感谢我的爸爸妈妈和家人，他们一直以来默默地支持着我，无论我遇到多少困难和挫折，他们都从不放弃地支持和鼓励我，给予我无私的关爱和支持，让我在学习和生活中觉得踏实稳定和安心。

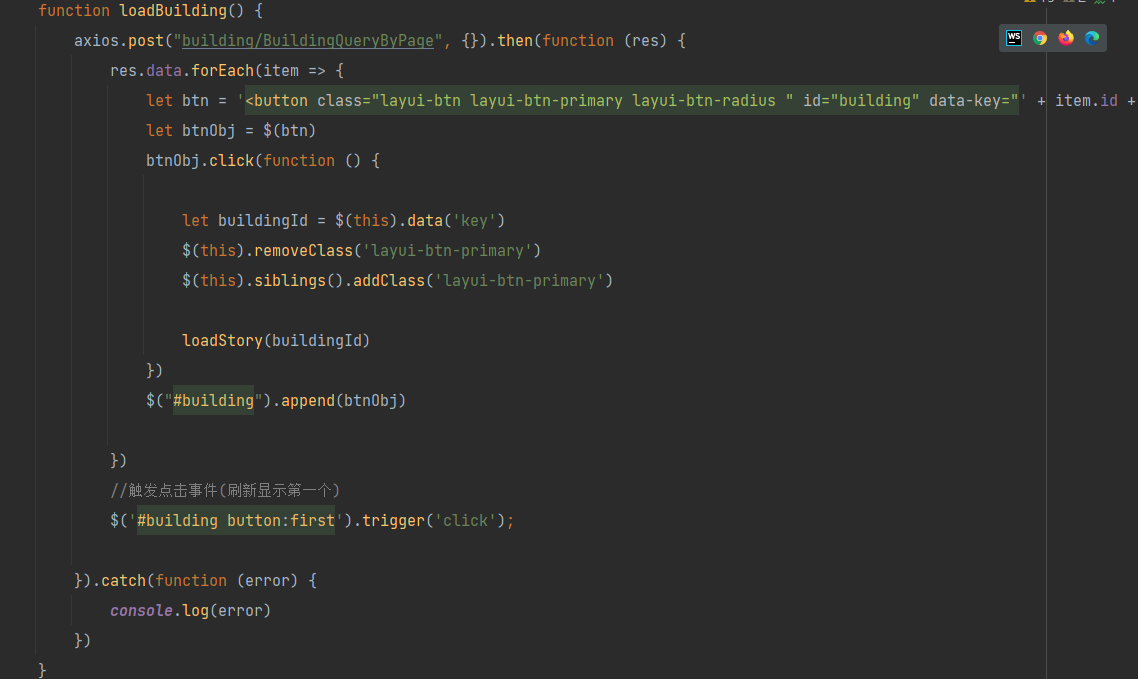
    感谢我的同学和朋友，他们陪伴我度过了美好的大学时光，在我的成长过程中给予了我很多的帮助和支持，与他们相处的时光让我感到无比快乐和幸福。

    最后，我要感谢所有为本次毕业论文提供帮助和支持的人士，你们的帮助和支持是我完成这篇论文的重要动力，谢谢你们！

    # 附录A:系统核心代码

    A.1 前端展示主要代码

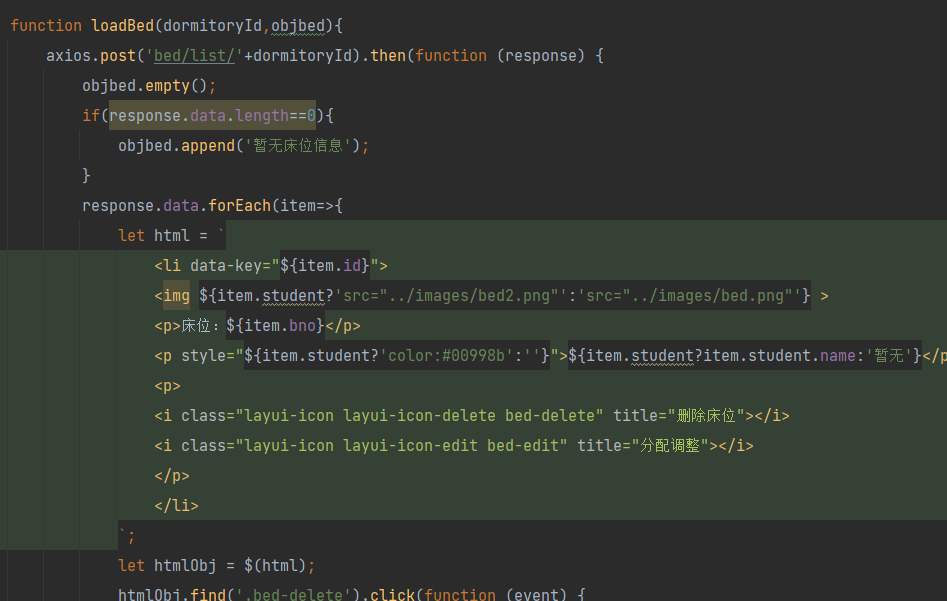
    （1）楼宇的展示的主要代码如下:

    （2）宿舍的展示的主要代码如下:

    （3）床位的展示的主要代码如下:

    （4）宿舍展示部分css代码如下：

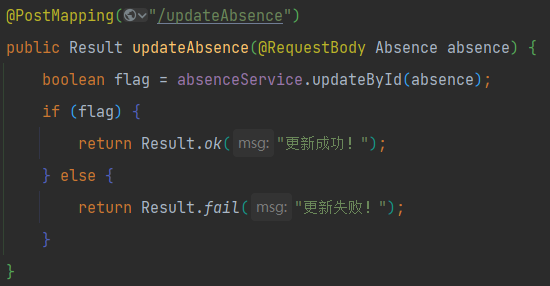
    

    A.2 后端管理主要代码

    （1）后端添加信息主要代码如下:

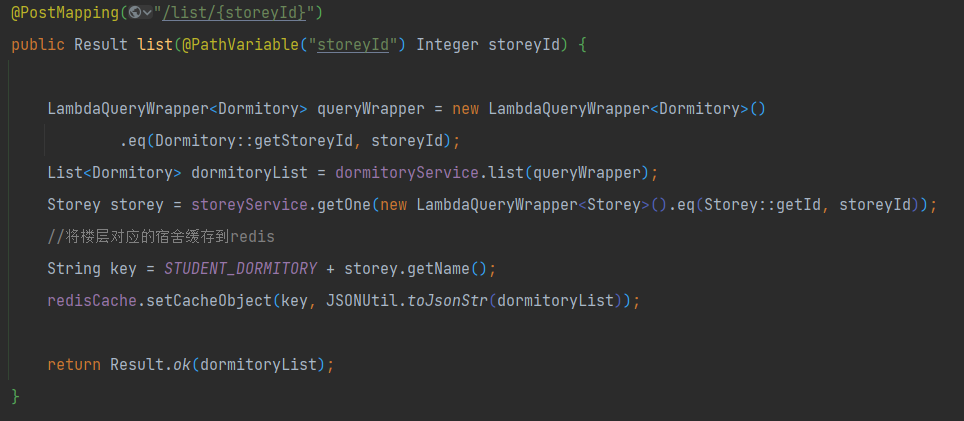
    （2）后台更新信息主要代码截图如下:

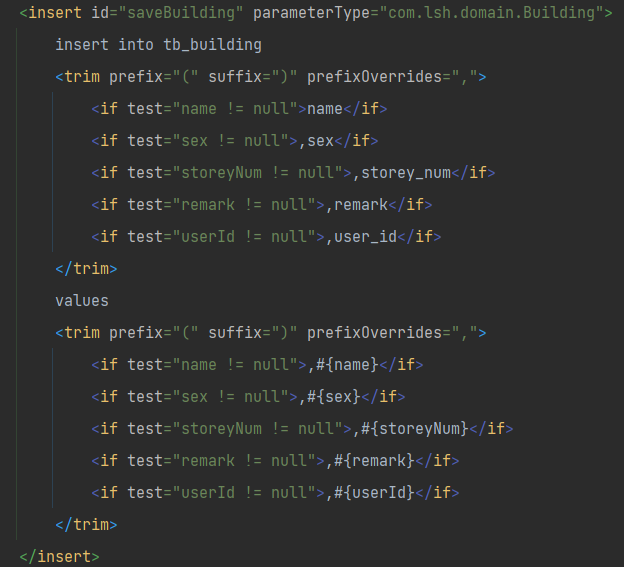
    （3）后台删除信息主要代码如下:

    （4）后端实现查询的主要代码如下:

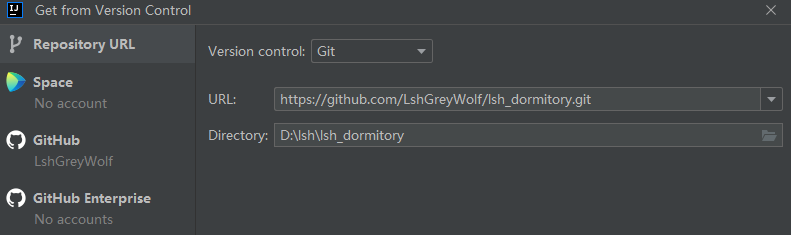
    （5）后端MyBatis的主要代码如下：

    # 附录B：软件使用说明书

    B.1 项目启动

    1. 打开IDEA和WebStorm，分别拉取Git上面的代码，如图B.1、图B.2所示:

    图B.1 拉取Git前端代码

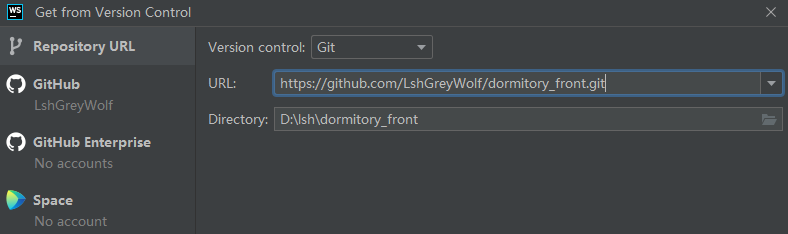
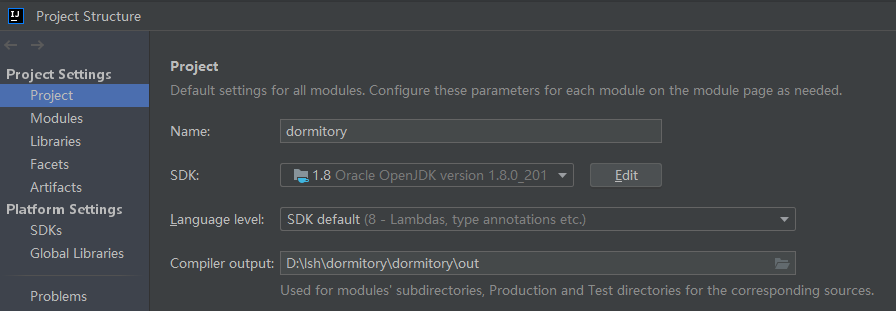
    

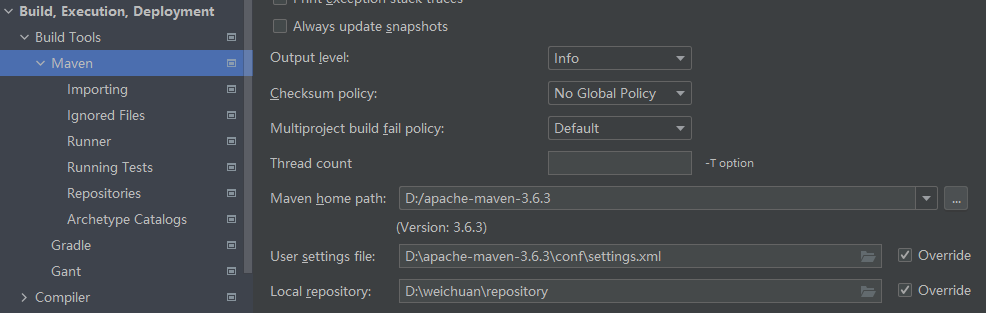
    图 B.2 拉取Git后端代码

    1. 打开IDEA项目结构，配置JDK，使用JDK1.8，如图B.3所示:

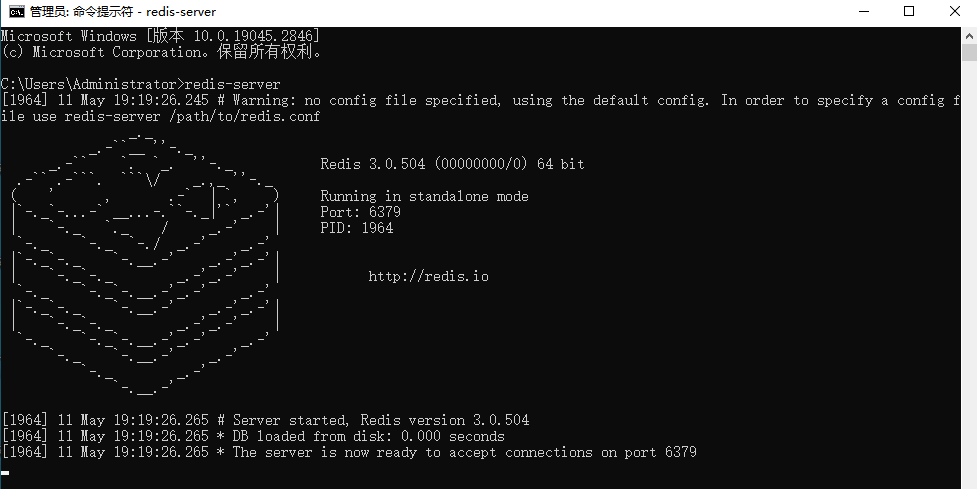
    图B.3 配置JDK

    1. 打开IDEA设置，配置本地Maven仓库，如图B.4所示:

    图B.4 配置Maven仓库

    1. 连接MySQL数据库，如图所示，输入账号：root，密码：123成功登录即可。如图所示：
    2. 连接Redis数据库，打开电脑终端输入redis-server命令，成功启动即可。如图B.6所示：

    图B.6 启动Redis

    1. 启动项目，在IDEA中找如图所示按钮，看到控制台“项目启动成功”字样即可。如图B.7所示：

    图B.7 启动后端代码

    1. 然后在浏览器输入http://localhost:63344/dormitory-front/page/login.html地址。如图B.8所示：

    图B.8访问浏览器地址

    1. 本系统的用户分为三类，分别为超级管理员、宿管员、学生，对应身份的账号可以使用不同的功能。其测试账号如下：

    超级管理员账号：admin，管理员密码：admin

    用户账号：张三，用户密码：123456

    学生账号：admin，学生密码：123456

    B.4 软件功能介绍

    B.4.1学生端介绍

    大学生宿舍自选管理系统的学生端主要是为学生提供自主选择宿舍的功能，其余还有报修申请、公告查看、离宿记录等功能供学生使用。学生登录时，需要输入账号、密码。如果账号、密码错误，会提示用户“输入错误，请重新正确的账号、密码”；输入如果账号、密码正确，将跳转到网站的首页。如图B.9所示：

    图B.9 登录注册

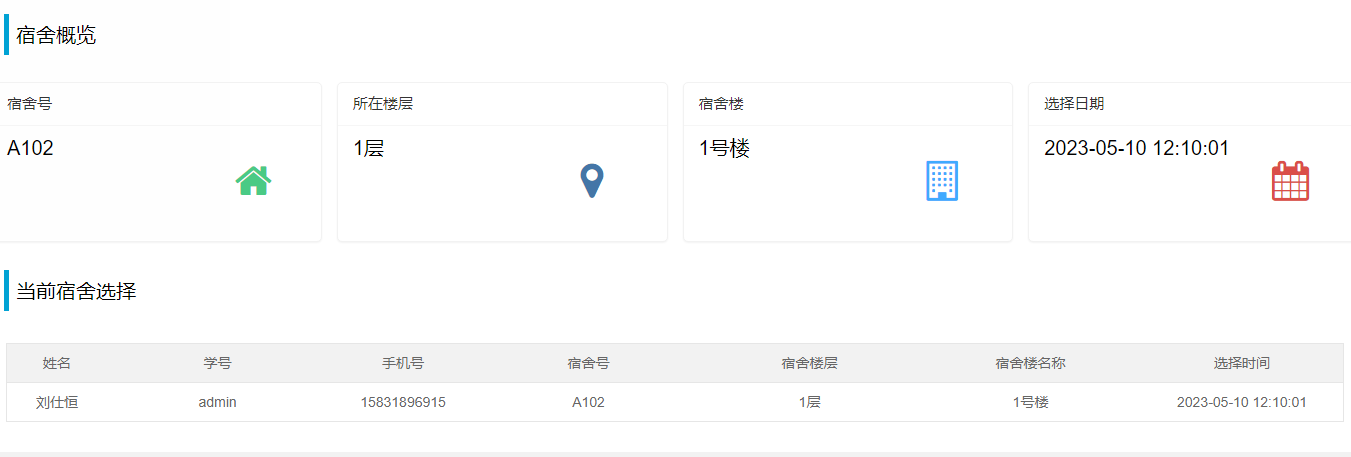
    学生成功进入系统后便可进行宿舍的选择。

    如图B.10所示：

    图B.10 宿舍选择

    学生选择完宿舍后，学生端的首页将展示该宿舍的相关信息。如图B.11所示：

    图B.11 旅游景点栏目页

    学生可以更改自己的离宿状态。学生在离宿后管理员将状态该学生改为离宿。如果学生在规定时间回来，自己可更改离宿状态为已回，超出时间不可更改。若规定时间未回，系统自动将状态改为缺勤。如图B.12所示：

    图B.12 离宿记录页

    学生可以申请报修。在申请时需要填写报修描述，管理员登录系统看到后会审核。如图B.13所示：

    图B.13 报修申请

    学生可以进行公告查看。如图B.14所示：

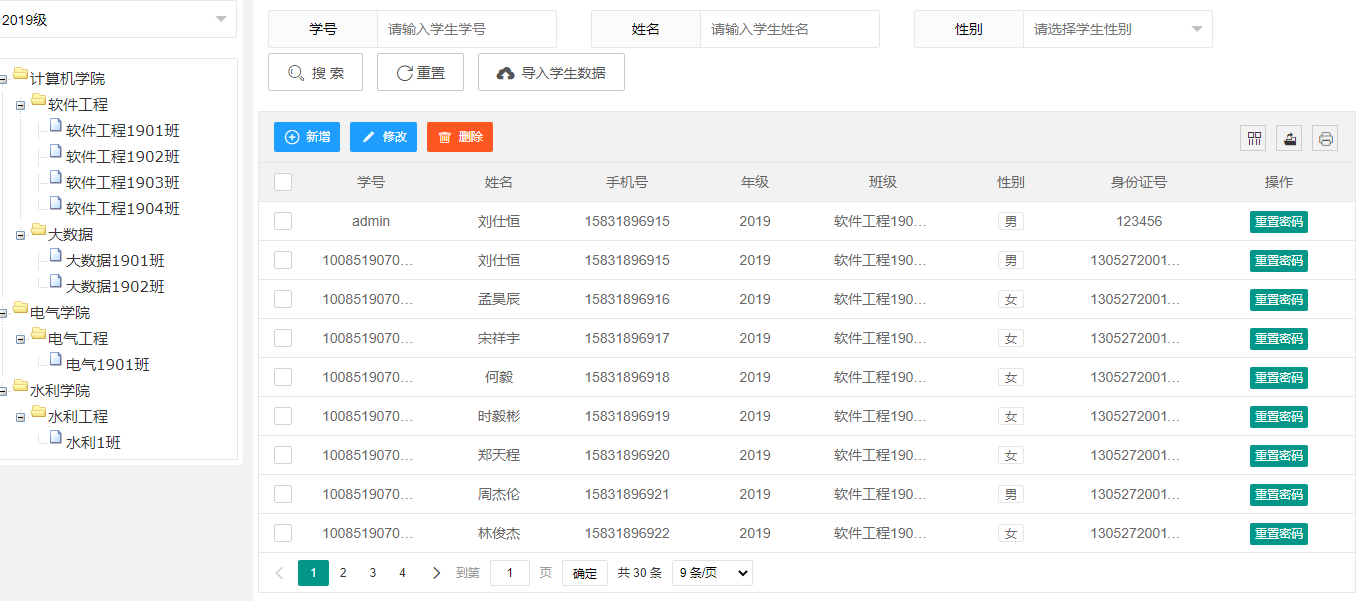
    图B.14 公告查看

    B.4.2管理员端介绍

    美丽邢台网站的后台主要对前台历史文化栏、旅游景点栏、特产美食栏、新闻栏四个栏目实现管理，可以对四个的图片文字实现增删改。

    大学生宿舍自选系统主要对学生、楼宇、宿舍、床位等管理。可以实现对他们的增删改查。同时还有公告管理、报修管理、系统管理等。

    后台管理员可以对学生进行管理。可以对学生的信息进行增删改查、重置密码、导入导出学生数据，界面展示如图B.15所示：

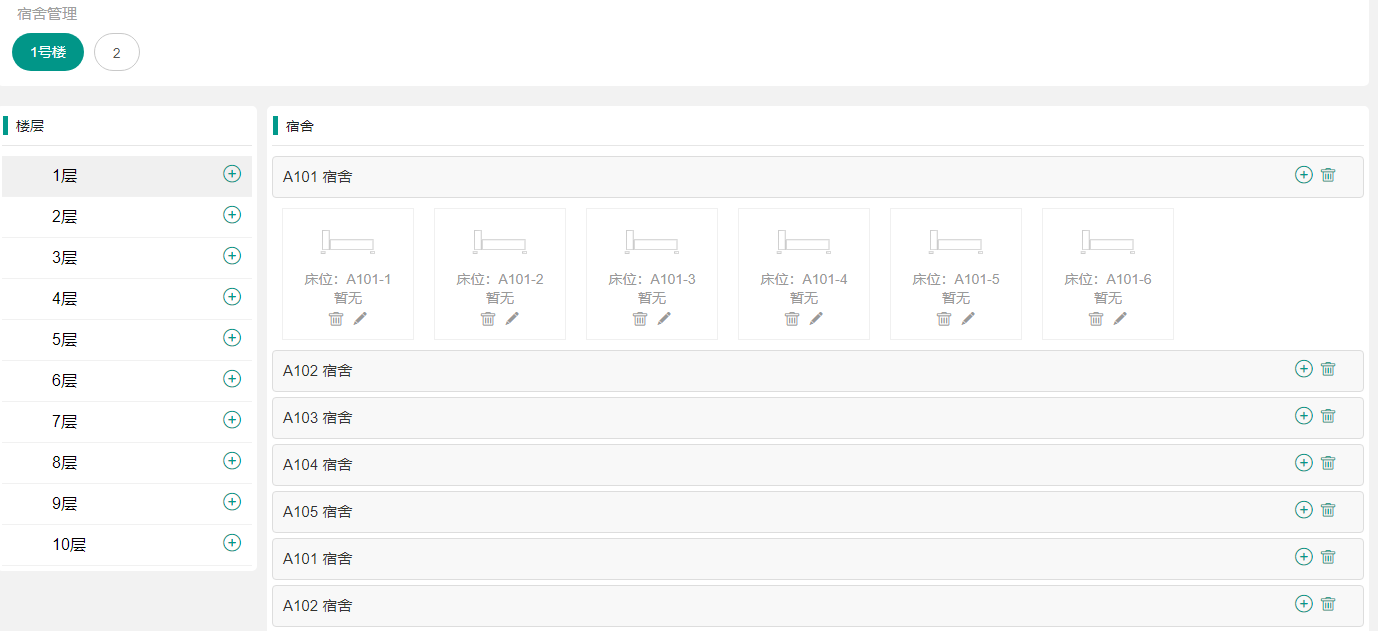
    图B.15 学生管理展示

    后台管理员可以对楼宇进行管理。界面展示如图B.16所示：

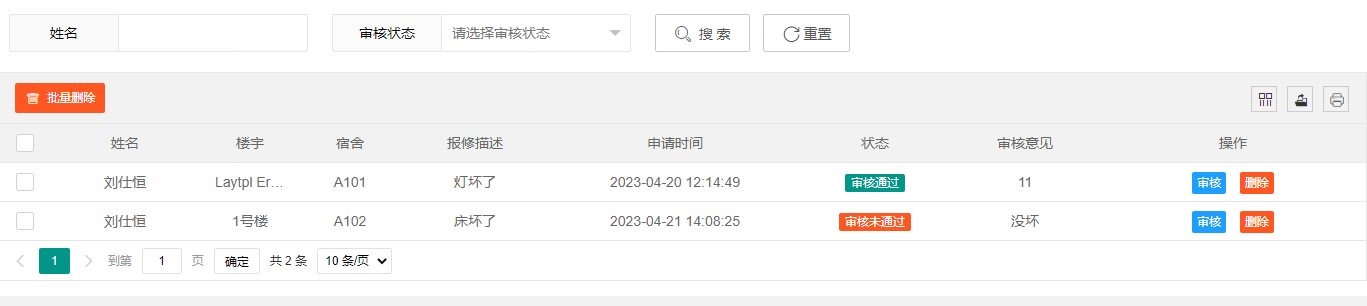
    图B.16 楼宇管理展示

    管理员对宿舍管理。界面如图B.17所示：

    图B.17 宿舍管理展示

    管理员可对报修审核。如图B.18所示：

    图B.18 报修管理展示

    审核界面展示如图B.19所示：

    图B.19 审核报修展示 [↑](#endnote-ref-11)