**大连理工大学本科毕业设计（论文）**

软件学院宿舍综合管理系统设计与实现

Design and Implementation of Dormitory Management System for School of Software

学 院（系）： 软件学院

专 业： 软件工程（日语强化）

学 生 姓 名： 包伟强

学 号： 201493069

指 导 教 师： 马瑞新

评 阅 教 师： 徐博

完 成 日 期： 2018.6.1

大连理工大学

Dalian University of Technology

摘 要

随着软件学院学生人数的不断增多，学生信息也变得更加复杂，这就使原本人工处理学生信息的方式变得局限性越来越大。为了更好的使用互联网技术和数据信息来帮助学院，本文设计并实现了一套符合软件学院业务需求的宿舍管理系统，能够更好地帮助学院后勤进行宿舍信息的管理。

本文对软件学院宿舍管理系统的业务流程和实际需求进行了详细分析，使用了SSM这个重要框架，运用软件工程的思想设计并实现了软件学院宿舍管理系统。该管理系统分为6个功能模块，其中宿舍管理用于处理宿舍楼信息，房间信息管理用于对房间信息进行增删改查等处理，学生信息管理用于对学生的居住信息、学号等进行管理，房间类型管理模块管理宿舍楼中所有房间类型，主要用于房间信息管理中，导员信息管理也主要是实现对导员信息的增删改查，审批管理是处理学生提交的退寝、申请入住、调换寝室的申请。本文首先分析了软件学院目前在人工进行宿舍信息管理方面的劣势，以及使用管理系统来管理宿舍信息后的优势，再从总体架构分析、用例分析和数据流分析多方面进行系统分析，确定出主要功能模块，然后详细设计出了系统数据库和核心模块实现流程，最后展示了系统界面实现和部分测试用例。

该系统已经在该软件学院内部正式运行，得到良好反馈，目前系统运行正常且满足基本需求，大大提高了工作效率。

**关键词：管理信息系统；信息存储；三层架构**

**Design and Implementation of Dormitory Management System for School of Software**

# Abstract

With the increasing number of students in School of Software, the student information becomes more complex, which makes the original way of manually managing student information become more and more limited.In order to better use of the Internet technology and the data and information to help college, this paper designs and implements a dormitory management system with the business needs of the school of software, better able to help the management of the dormitory information of the college logistics.

In this paper, school of software the dormitory management system of business process and the actual demand are analyzed in detail, using the SSM framework, based on the idea of software engineering school of software the dormitory management system was designed and implemented. The management system is divided into six functional modules, including dormitory management to handle a dormitory building information, room information management to add ,delete ,modify and search the information of rooms,student information management to manage the students living information, student id, etc,room type management to manage all the room types in the dormitory, it is mainly used for room information management, counsellor information management is also mainly to add ,delete,modify and search the information of counsellors, approval management to handle students' submit of checking out ,application for registration and exchanging the dormitory application.This article first analyzed the disadvantage of manually managing the information in school of software, and the advantage of using management information system to manage the dormitory after, and then determined the main function module according to overall architecture analysis, case analysis and data flow analysis. Then this paper designed the system database and core module realization process and finally showed the interface of system some the test cases for the software testing.

The system has been tested in a small scale within the school of software and received good feedback. Currently, the system operates normally and meets the basic requirements, which greatly improves the working efficiency.

Key Words：Management Information System;Information Storage;3-tier architecture

目 录

[摘 要 I](#_Toc22605)

[Abstract II](#_Toc18621)

[1 绪论 1](#_Toc7088)

[1.1 开发背景与现状 1](#_Toc12546)

[1.2 国内外发展现状 2](#_Toc5565)

[1.3 论文组织结构 2](#_Toc19431)

[2 相关技术 4](#_Toc17672)

[2.1 SSM框架 4](#_Toc27827)

[2.2 Echarts图表 5](#_Toc19129)

[2.3 三层架构 6](#_Toc18180)

[3 系统分析 7](#_Toc3774)

[3.1 总体架构分析 7](#_Toc17069)

[3.2 系统用例分析 7](#_Toc696)

[3.3 数据流分析 9](#_Toc30594)

[3.4 功能模块分析 11](#_Toc24853)

[3.4.1 宿舍管理 12](#_Toc22646)

[3.4.2 房间信息管理 12](#_Toc29239)

[3.4.3 学生信息管理 13](#_Toc4280)

[3.4.4 房间类型管理 14](#_Toc28570)

[3.4.5 导员信息管理 15](#_Toc27317)

[3.4.6 审批管理 15](#_Toc7505)

[3.5 运行环境分析 16](#_Toc6183)

[4 系统设计 17](#_Toc13401)

[4.1 数据库设计 17](#_Toc11158)

[4.1.1 概念模型设计 17](#_Toc19600)

[4.1.2 主要数据表设计 18](#_Toc22271)

[4.2 核心模块设计 21](#_Toc11088)

[4.2.1 宿舍管理 21](#_Toc24688)

[4.2.2 房间信息管理 22](#_Toc17411)

[4.2.3 学生信息管理 23](#_Toc1253)

[4.2.4 房间类型管理 25](#_Toc25124)

[4.2.5 导员信息管理 26](#_Toc678)

[4.2.6 审批管理 27](#_Toc28816)

[5 系统实现 28](#_Toc4083)

[5.1 宿舍管理 29](#_Toc12307)

[5.2 房间信息管理 30](#_Toc26571)

[5.3 学生信息管理 32](#_Toc32652)

[5.4 房间类型管理 33](#_Toc9728)

[5.5 导员信息管理 33](#_Toc10794)

[5.6 审批管理 34](#_Toc13577)

[6 系统测试 36](#_Toc5257)

[结 论 40](#_Toc25378)

[参 考 文 献 41](#_Toc10168)

[致 谢 42](#_Toc8643)

# 1 绪论

## 1.1 开发背景与现状

当今社会，科技迅速进步发展，计算机技术应用的范围也越来越广。当下对信息管理系统的使用已经覆盖了各行各业，包括教育、科研、金融等领域。所以为了保证组织机构的发展性，设计一个适合的信息管理系统就变得的尤为重要。而对学校来说，学生宿舍管理也是日常管理学生的一个重要组成部分。目前仍然有许多学校是由后勤人员由手工来记录与处理学生住宿情况的数据的，对于小规模的学校来说还勉强能够接受后勤人员手工记录数据，但是对于部分学生数据规模比较大的高校来说，人工记录与处理就会变得极其麻烦。而且根据需求需要定向查找某条数据记录的时候，只能依靠人工在庞大的数据资料中一条一条地查找，这样不仅会消耗大量的人力物力，并且也会大大降低工作效率。如果能够使用计算机开发一个管理信息系统，用它来处理宿舍以及学生的信息，相比较于人工处理，会有许多优势，包括提高查询信息记录的速度、节约人力物力资源、大幅提高工作效率、减少处理错误、容易查询处理记录。今天，计算机已经应用在许多领域，计算机被应用来处理信息的主要原因包括几个方面：（1）计算机可以进行许多复杂、数据量大的信息处理；（2）计算机可以节省大量人力与物力资源；（3）相比较与人工，计算机可以大幅度提高工作效率；（4）计算机可以提高信息保密性。

管理信息系统（Management Information System，简称MIS）是为了实现计算机管理、维护、处理信息，在计算机和管理相关学科基础上实现的技术。它首先研究管理信息时对信息的维护、处理的流程，并将其在计算机上实现[1]。它通过统一人、计算机、信息三个载体，实现信息的智能化管理，将信息的收集、加工、存储、维护在计算机上实现，提高了管理人员的工作效率。

随着管理信息系统越来越广泛的应用，管理工作的效率也不断提升。作为落后的人工管理方式的一种替代，管理信息系统实现了管理工作的规范化与现代化，使现代化信息管理形成了一个有效率的系统[2]。使用人工操作时，受限于人工能力的局限性，尽管信息管理人员投入大量时间精力，也难免会出现效率低下、容易出错等情况。相较之下，采用管理信息系统则可以避免大部分上述情况。它将信息管理和计算机相统一，对人和计算机的特点取长补短，使得管理信息系统实现信息管理现代化成为可能。

当下，使用计算机来进行数据的处理与分析的管理信息系统已经是各个企业与机构的信息管理部门不可或缺的一部分，这也正是信息管理逐步走向现代化的一个证明。而可以预见的是，未来信息资源会越来越多、越来越复杂，实现一个高效而实用的管理信息系统将会是信息管理技术发展的一个趋势。

## 1.2 国内外发展现状

由于高校办学规模的不断扩大，在校大学生的信息也逐渐呈现多样化、复杂化的趋势，这在很大程度上让学生信息管理工作出现了许多新的问题。由于学生人数不断增长，学生宿舍管理的信息量与数据处理量也变得日益庞大，人工处理学生信息的工作模式有许多不足之处，包括处理信息速度比较慢、查阅困难、准确性差，很难适应当今信息化时代的学生宿舍信息管理工作。因此，如何应用现代技术使对学生宿舍信息的管理变得更科学、更高效将会是各个高校后勤管理工作者的首要目标。

在传统模式下利用人工的方式进行宿舍信息管理，存在着许多缺点，如：信息处理效率低下、数据信息保密性差、随着时间推移将产生大量的冗余文件，更加不便于查找、更新、维护[3]。诸如这些情况，令后勤管理人员对学生的信息管理带来了很大困难，严重影响了后勤管理人员的工作效率，针对这些情况，我们可以选择使用计算机技术，使用一个信息管理系统来代替传统的人工处理模式，来实现宿舍信息的现代化管理，这一技术相比较于人工管理信息有着无法代替的优点，例如：检索更加迅速、查找信息更加方便、有很强的保密性、信息存储量更大、寿命更长、成本更加低廉等。使用计算机处理学生宿舍信息后，这些优点能够被放大，极大提高了学生宿舍信息管理的效率，使得我们的学校能够在市场经济的发展过程中具有一定竞争优势，这也是学校实现科学化、正规化、现代化的一个重要条件。使用宿舍信息管理系统，实现了科学、统一管理，避免人工操作容易产生的错误，大大的提高了工作效率。

为了提高高校后勤宿舍管理部门的工作效率、充分利用资源、减少不必要的人力物力消耗、方便宿舍管理部门的后勤工作人员全面地掌握学生住宿情况，为宿舍管理部门开发了一个宿舍信息管理系统，实现宿舍信息管理的标准化、规范化与现代化。

## 1.3 论文组织结构

本论文将对大连理工大学软件学院宿舍管理系统进行系统分析和具体实现的阐述说明，论文的组织结构如下：

1. 绪论：本章主要介绍本项目的来源以及项目背景，分析了宿舍管理信息化的实际意义。此外，还对信息化管理的国内外发展现状做了调查研究。
2. 相关技术：本章主要介绍了在系统实现过程所用到的相关技术，系统实现主要使用了SSM（Spring+Spring MVC+Mybatis）框架、Echarts图表、三层架构等。
3. 系统分析：本章将从总体架构分析、系统用例分析、数据流分析和功能模块四个维度对系统平台进行系统分析。
4. 系统设计：本章首先依据数据库系统设计的思想对系统数据库进行设计，主要从概念模型和数据库表设计两方面对系统数据库进行描述。然后介绍系统核心模块的详细设计。
5. 系统实现：本章展示系统各个功能模块的部分实现界面以及核心代码。
6. 结论：对整个系统开发和全文进行总结并对下一步工作进行展望。

2 相关技术

## 2.1 SSM框架

SSM框架由Spring、Spring MVC、MyBatis三个部分组成，它是继SSH（Struts、Spring、hibernate）之后，又流行起来的Java EE企业级框架，在搭建各种各样的企业级应用系统方面有自己独特的优势[4]。  
 Spring是一个开源框架，最初是由Rod Johnson的著作《Expert One-On-One J2EE Development and Design》中陈述的一些理念衍生出来的结果。创建它的目的是解决在企业级应用开发容易遇到的一些复杂情况。Spring框架使用了一些基础的JavaBean来代替EJB完成那些以前只能由它完成的事情。但是，Spring不仅仅可以用在服务器端的开发。从软件工程的简单性、可测试性、松耦合性的目的来说，不仅限于企业级应用，任何简单的Java应用也都能应用Spring框架。简单而言，Spring是轻量级的、实现了控制反转（IOC）和面向切面（AOP）的一个容器框架[5]。  
        控制反转（IOC）也称为依赖注入。它利用工厂模式，即把对象交给容器来管理，只需编码者在spring配置文件事先中配置好相应需要用到的bean、配置好相关所需的属性，让spring容器启动后就可以自动生成相应类的实例对象和管理对象[6]。在spring容器启动后，bean就会被spring按照事先在配置文件中配置的那样初始化好，在编码者需要使用时，就会将这些已经初始化好的bean分配给需要调用的类，利用类的setter方法来注入，不需要编码者大量重复的重新创建这些bean了。  
        Spring的另一个特性是面向切面（AOP），和面向对象编程（OOP）类似，只是一种编程思想，而并不是某种技术，AOP某种程度上来讲就是对OOP的一个补充和完善[7]。OOP通过引入封装、继承、多态等众多新颖的概念建立了一种独特的对象层次结构，目的主要是用来形成一个集合来模拟代码中容易用到的公共重复行为。但是大量使用OOP的弊端是当我们需要为某些分散、难以定义重复行为的对象引入公共行为时，OOP则会显得不那么合适。换言之，编码者可以通过使用OOP定义从上到下的对象层次关系，但却难以用OOP来定义从左到右的关系，例如日志功能。设计日志代码时往往都是被编码者水平散布在所有对象层次中的，这部分日志相关的代码一般都与所被散布到的对象中的核心功能毫无关系。在设计OOP时，这种情况会导致大量代码的重复，而不利于将这部分核心代码在各个模块中重用。而代替OOP，使用AOP则可以避免这种情况，它可以将程序中的横向的交叉业务逻辑（安全、日志、事务等）封装成一个切面，然后由编码者注入到所有会使用到该部分代码的目标对象（即具体业务逻辑）中。目前AOP的实现方式主要分为两大类：一是静态织入，通过引入特定使用的语法创建“方面”，让编译器在编译期间织入“方面”相关的代码；二是采用动态代理的方式，通过截取消息，对截取到的消息进行修饰，藉以代替来执行原对象行为。

Spring MVC（Model、View、Controller）是Spring框架的一个后续产品，它支持原生的Spring特性，并且已经被融合在Spring Web Flow中，这就让编码者对项目的开发与支持变得十分简单规范。Spring MVC成功实现了控制器、对象模型、视图、拦截器和处理程序对象角色的分离，这种分离能够使开发者更容易进行项目的架构设计[8]。 MyBatis原本叫做iBatis，是apache的一个开源项目，2010年iBatis被从apache software foundation转移到了google code，并且更名为MyBatis 。MyBatis是一个持久层框架，基于JAVA。MyBatis提供了包括SQL Maps和Data Access Objects（DAO）在内的各种持久层框架，主要实现了数据连接层的设计[9]。它消除了几乎所有的JDBC相关代码，包括数据库查询参数的手工设置和结果集的检索。MyBatis通过使用简单的XML文件或在代码中使用注解用于实现配置和原始映射，将接口和JAVA中的POJO（Plain Old Java Objects，普通的 Java对象，即实体）映射为数据库中的相应记录。换言之，MyBatis框架主要能实现数据连接层，帮你管理数据的增删改查。

## 2.2 Echarts图表

ECharts作为一个纯 Javascript 的图表库，兼容市面上绝大多数的浏览器（包括IE8/9/10/11、Chrome、Firefox、Safari等），可以在大部分 PC 和移动设备上流畅运行。Echarts底层设计为依赖轻量级的Canvas类库 ZRender，能为用户提供直观、可交互、可高度个性化定制、生动的可视化数据图表[10]。它拥有以下优势：（1）Echarts拥有丰富的图表类型 。Echarts可以提供给用户各种各样的图标，包括常规的折线图、柱状图、散点图，地理数据方面的地图、线图、热力图，统计学的盒型图，数据关系方面的关系图等，并且也能支持各个图的混搭；（2）动态数据 。Echarts与数据实时交互，由数据驱动，数据的改变能实时体现在图表上。因此实现动态数据也十分简单，只需要从后台获取数据、输入数据，Echarts就能自动找到数据间的差异并使用最合适的动画来展现数据的对比或是变化；（3）移动端的优化。考虑到移动端的流量问题，在移动端的图表库设计为体积尽量小。Echarts和ZRender代码的重构能实现核心部分图表体积的缩小。并且Echarts也提供了按需打包的能力，可以按照用户需求提供不同体积与精细度的图表；（4）多维数据支持和丰富的视觉编码手段 。Echarts拥有对多维数据的支持。除了传统的多维数据可视化工具（如平行坐标等）的加入，Echarts对于传统的散点图也能实现多维度的数据传入。而为了将数据展现的更直观，配合VisualMap这种视觉映射组件，Echarts也能将不同维度的数据映射为颜色、大小、透明度、明暗度等方面的差异。

## 2.3 三层架构

三层架构(3-tier architecture)就是将整个业务应用划分为三个层次：表现层、业务逻辑层、数据访问层。区分层次与软件工程“高内聚，低耦合”的思想一致[11]。表现层（UI）也与用户交互的界面，大部分情况下就是指系统页面，主要用于用户输入数据和获取经过处理的数据。业务逻辑层（BLL）是针对具体业务逻辑的操作，处于表现层和数据访问层中间，主要将具体的业务逻辑转换为对数据层的操作并进行处理，在表现层和数据访问层之中起一个桥梁的作用。数据访问层（DAL）直接操作数据库，实现对数据库的连接与增删改查等操作的实现，主要是将数据库中的数据提交给业务层，以及将业务层处理完的数据存储到数据库中。实体（Entity）虽然不属于三层架构中的任何一个，但是也穿插在三个层次中，也是必不可少的一部分。

相比较于双层或单层结构，三层架构从开发角度和应用角度看有更大的优势。它适合企业项目开发，每个人分工不同，协作效率比较高，只需要少数人对系统有比较全面的了解，从一定程度上降低了开发难度。三层架构的另一个优点是易于后期维护修改，因为代码是分层实现的，耦合性比较低，因而后期维护时相比较于双层或单层架构会更加容易一些。三层架构的最大优点是它的安全性。用户端只能通过逻辑层来访问数据层，减少了入口点，系统更加安全。

3 系统分析

## 3.1 总体架构分析

本系统是为软件学院开发的，使用范围主要是在院内。根据需求分析，系统采用B/S架构。B/S架构把系统服务都汇集到服务器端，服务器端接收用户的数据并进行处理，然后返还给用户端，这样可以降低用户端的硬件配置要求，减轻用户端压力。学生只需要有浏览器和互联网就可以实现随时随地使用系统[12]。根据实际情况，我们设计了如图3.1所示的网络拓扑图。

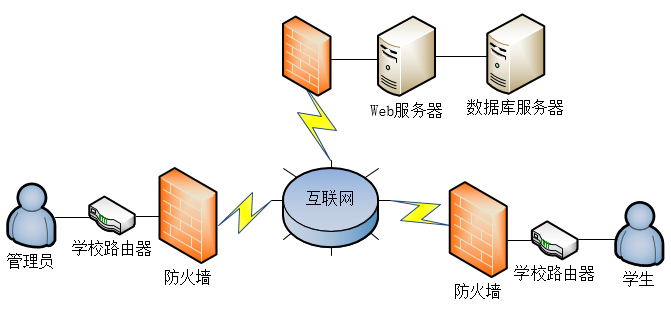


图3.1 系统拓扑图

## 3.2 系统用例分析

本系统是一个宿舍管理系统，具备的基础功能包括宿舍管理、房间信息管理、学生信息管理、导员信息管理、房间类型管理、审批信息管理等。本节将对系统中的管理人员、学生这两个比较重要的人员角色进行用例分析。

宿舍管理员是项目的核心人员，主要管理宿舍楼、房间信息与学生的住宿情况。如图3.2是管理员的用例图。宿舍管理员能实现的功能有宿舍管理、房间信息管理、学生信息管理、导员信息管理、房间类型管理、审批信息管理等。其中宿舍管理主要是实现了添加宿舍楼的功能，房间信息管理包括添加、删除、查询房间等功能，学生信息管理包括添加学生、导入学生信息、导出学生信息、下载导入模板、删除学生信息、修改学生信息等功能，导员信息管理包括添加与修改导员信息的功能，房间类型管理包括添加与删除房间类型的功能，审批信息管理包括审批入住、退寝、换寝申请的功能。

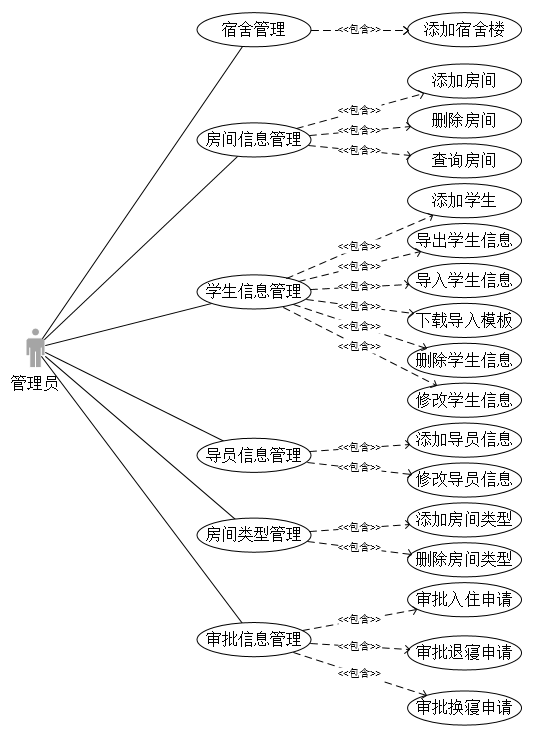


图3.2 管理员用例图

如图3.3是学生用例图。学生用例主要能实现申请入住、退寝、调换房间的功能。实现这三个功能需要与管理员进行交互，先提交申请，再由管理员进行审批，最后获取结果确认是否申请成功。

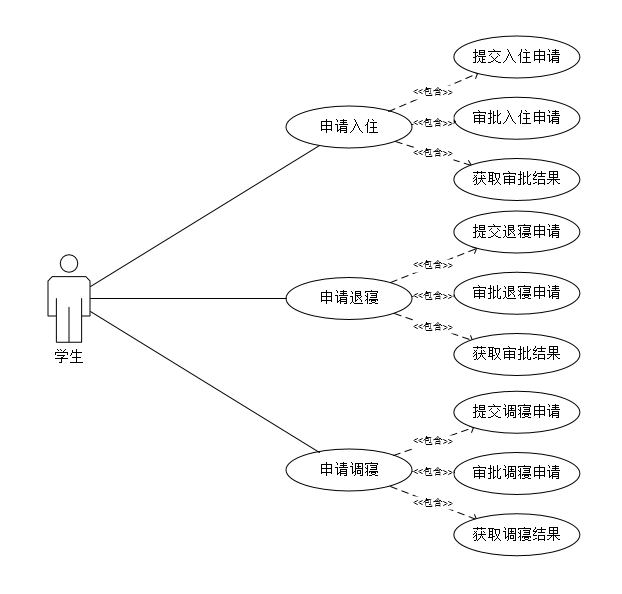


图3.3 学生用例图

## 3.3 数据流分析

数据流图又称DFD图，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，能够便于用户表达功能需求和数据需求及其联系、清晰表达数据流的情况、有利于系统建模[13]。

如图3.4所示是整个系统的顶层数据流图，从数据流图来看主要是对宿舍信息、房间信息、学生信息、导员信息、房间类型信息与审批信息等进行处理，其中还会用到宿舍楼信息、房间信息、学生信息、导员信息、房间类型信息等数据存储，而系统所需要处理的信息主要来源于系统管理员与学生两个外部交互方，学生会与审批信息管理模块进行交互，而管理员能跟所有模块进行交互。

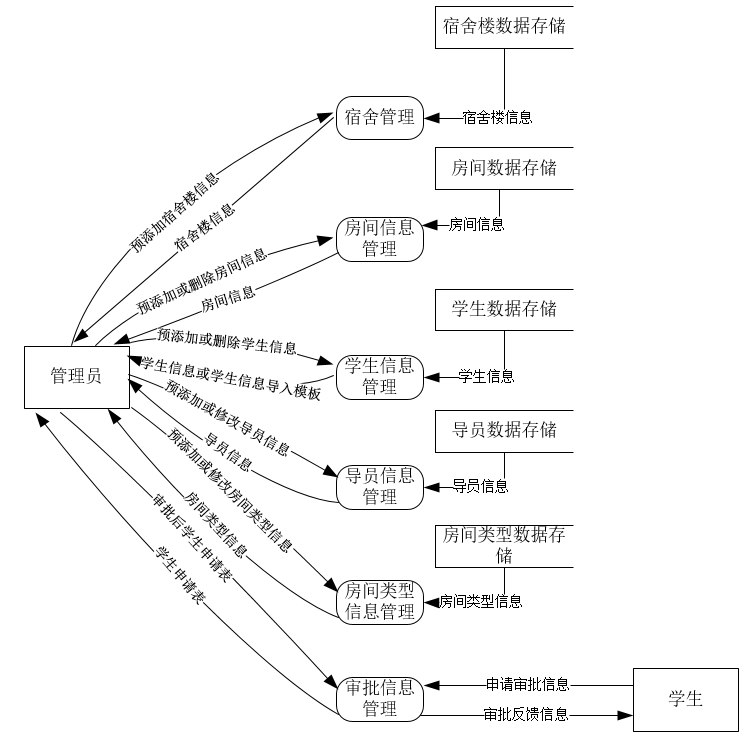


图3.4 系统顶层数据流图

如图3.5所示是学生信息管理的数据流图。管理员进行导入学生信息、添加学生、修改学生信息、删除学生信息时都是将对应的学生信息存储到学生数据表中，导出学生信息时则需要从学生数据表中获取学生信息并导出给管理员，而下载导入模板时则需要从模板数据存储中先获取再导出给管理员。

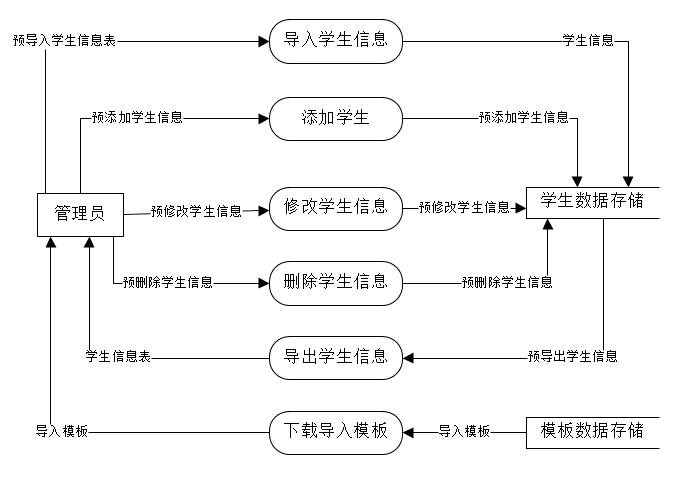


图3.5 学生信息管理数据流图

如图3.6所示是审批管理的数据流图。学生进行入住、退寝、换寝操作时需要先提交申请，申请会存入数据库中然后提交给管理员，管理员进行审批后将审批结果再返给学生，学生依情况可能申请成功，也可能需要修改后再次提交申请。

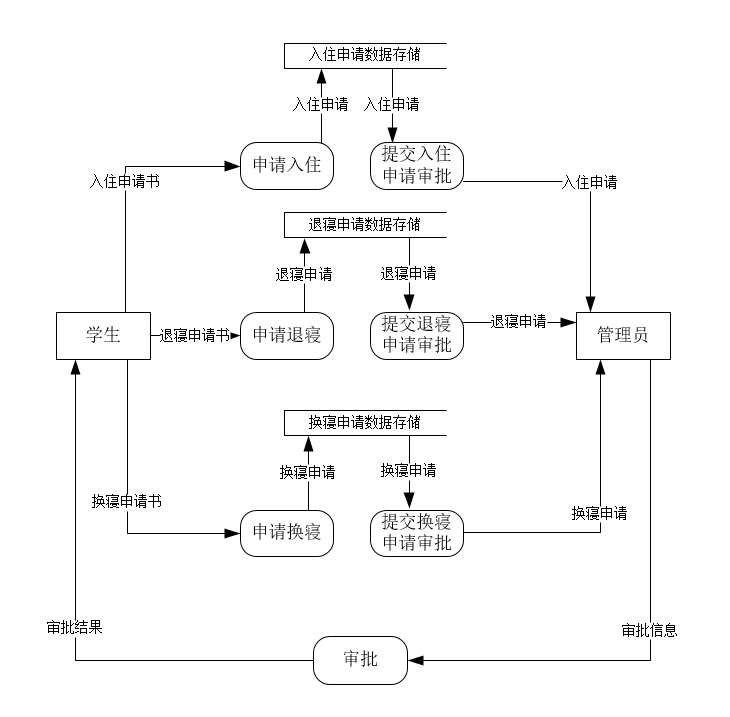


图3.6 审批管理数据流图

## 3.4 功能模块分析

根据上文对系统的用例和数据流分析，设计了一个系统功能结构图，如图3.7所示，下面将着重阐述图中几个功能模块。

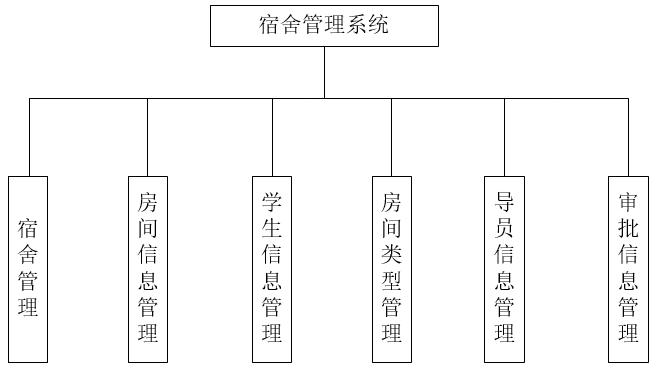


图3.7 系统功能结构图

### 3.4.1 宿舍管理

宿舍管理模块能够实现对宿舍楼信息的基本增删改查功能，如图3.8所示，主要功能如下。

1. 添加宿舍楼。管理员在页面上输入宿舍楼信息并确认后提交，即可将宿舍楼信息添加到数据库中。
2. 删除宿舍楼。管理员将不需要或多余的宿舍楼信息删除。
3. 修改宿舍楼信息。一般是修改楼号信息。
4. 查看宿舍楼信息。管理员可以直接查询所有宿舍楼的详细信息。

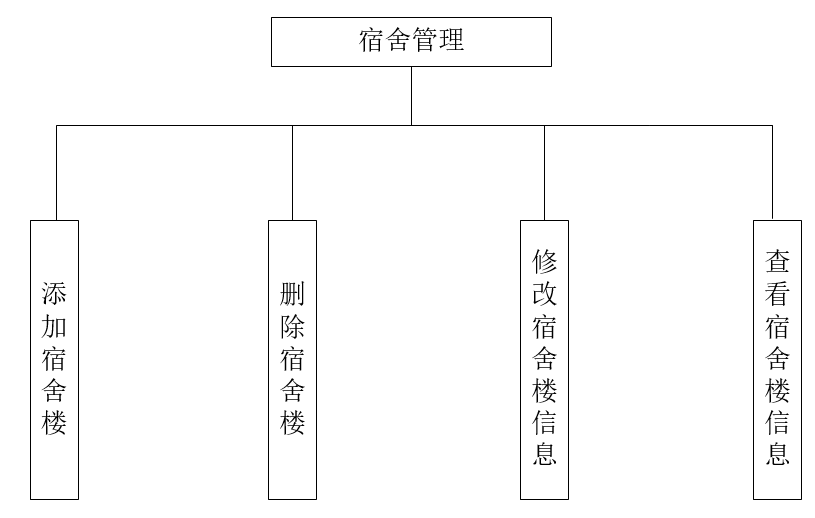


图3.8 宿舍管理功能结构图

### 3.4.2 房间信息管理

在管理宿舍时，会经常需要对房间进行一些操作，如图3.9所示就是一个房间信息管理的功能模块图，提供如下一些功能。

1. 添加房间。顾名思义，就是选定某个宿舍楼后可以进行房间的添加，添加时可以设置床位数之类的参数。
2. 删除房间。一般很少用到的功能，可以将房间信息删除。
3. 查看房间图表。可以以图表的形式查看房间，可以直观的看到有几个空床位的房间。
4. 查询房间。可以按照楼号、床位数、学号、宿舍号查询。

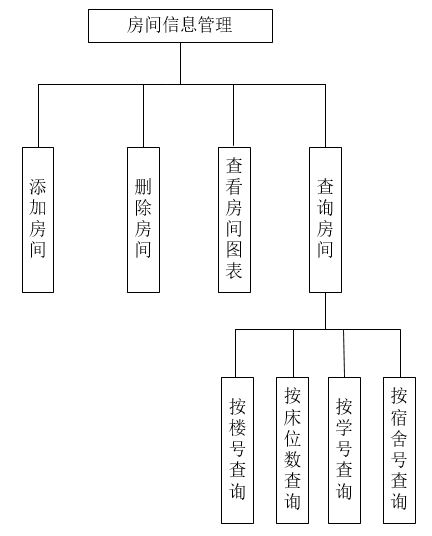


图3.9 房间信息管理功能结构图

### 3.4.3 学生信息管理

学生是整个宿舍管理系统的主要实体，对学生的信息管理也是必不可少的，如图3.10所示是学生信息管理的功能模块图，提供如下一些功能。

1. 添加学生。往数据库中添加学生数据有两种方式，一是批量导入，二是输入单个学生信息然后添加。
2. 删除学生信息。删除时可以选择按年级批量删除或选中删除。
3. 修改学生信息。对学生信息进行修改，一般是修改床位和宿舍号。
4. 下载导入模板。下载导入模板，用于批量导入学生信息。
5. 导出学生信息。按照年级或级队批量导出学生的信息，格式与导入模板相同。
6. 导入学生信息。按照下载的模板输入学生数据，选择该文件导入，可以实现批量导入。
7. 查询学生信息。实现模糊查询，能够按照给定关键字查询学生信息。

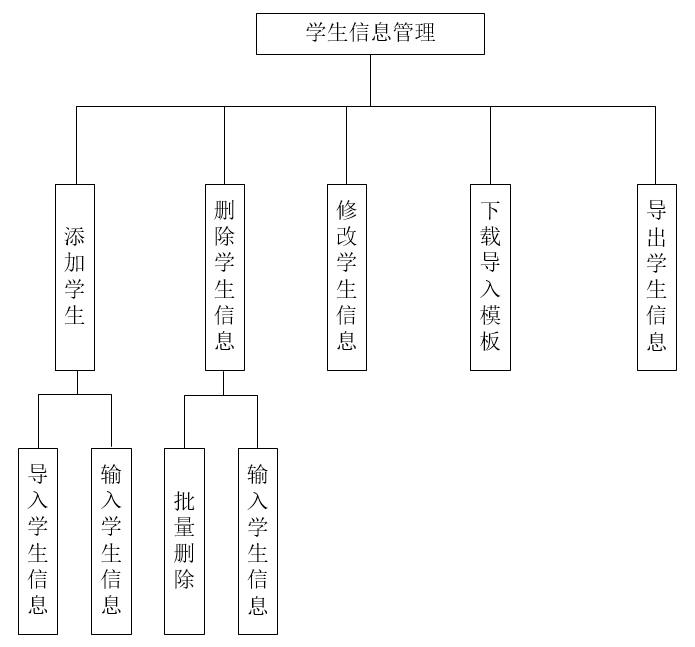


图3.10 学生信息管理功能结构图

### 3.4.4 房间类型管理

房间类型管理如图3.11所示，主要功能如下。

1. 添加房间类型。由管理员进行操作将房间类型数据添加到数据库中。
2. 删除房间类型。选择不需要的房间类型删除。
3. 修改房间类型信息。选中某条信息修改并确认后提交到数据库即可。
4. 查看房间类型信息。查看目前所有房间类型的信息。

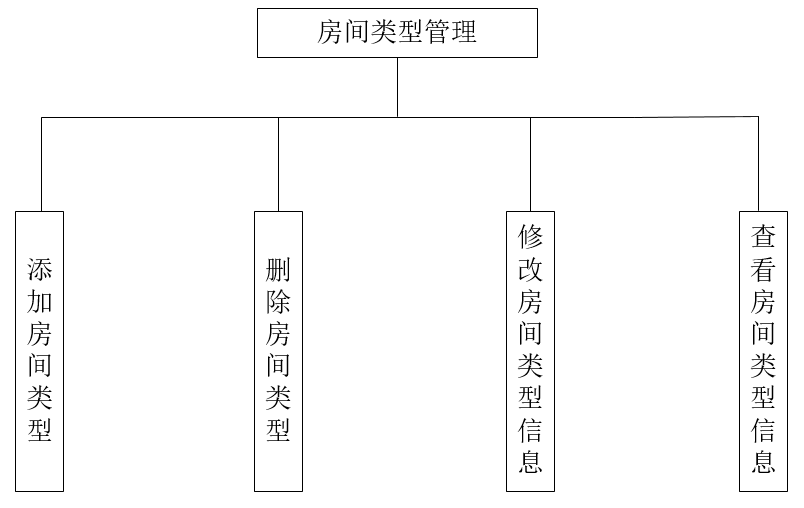


图3.11 房间类型管理功能结构图

### 3.4.5 导员信息管理

导员信息管理与房间类型管理类似，也是实现了增删改查四个功能。如图3.12所示，主要功能如下。

（1）添加导员。由管理员进行操作将导员数据添加到数据库中。

（2）删除导员。选择不需要的导员信息删除。

（3）修改导员信息。选中某条信息修改并确认后提交到数据库即可。

（4）查看导员信息。查看目前所有导员的信息。

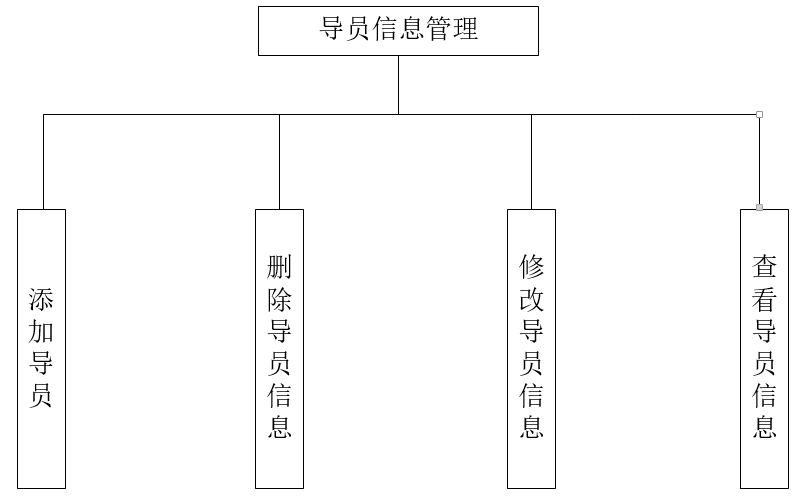


图3.12 导员信息管理功能结构图

### 3.4.6 审批管理

学生需要入住、退寝或是调换寝室时需要进行申请，此时就要用到审批管理这个功能模块，如图3.13所示，主要功能如下。

1. 审批入住申请。学生申请入住时需要先提交申请，然后经过管理员审批后反馈申请结果。
2. 审批退寝申请。和入住申请类似，学生申请退寝时需要先提交申请，经过管理员审批后反馈申请结果。
3. 审批调寝申请。和入住申请、退寝申请一样，也需要经过提交申请、管理员审批、反馈结果三个步骤。

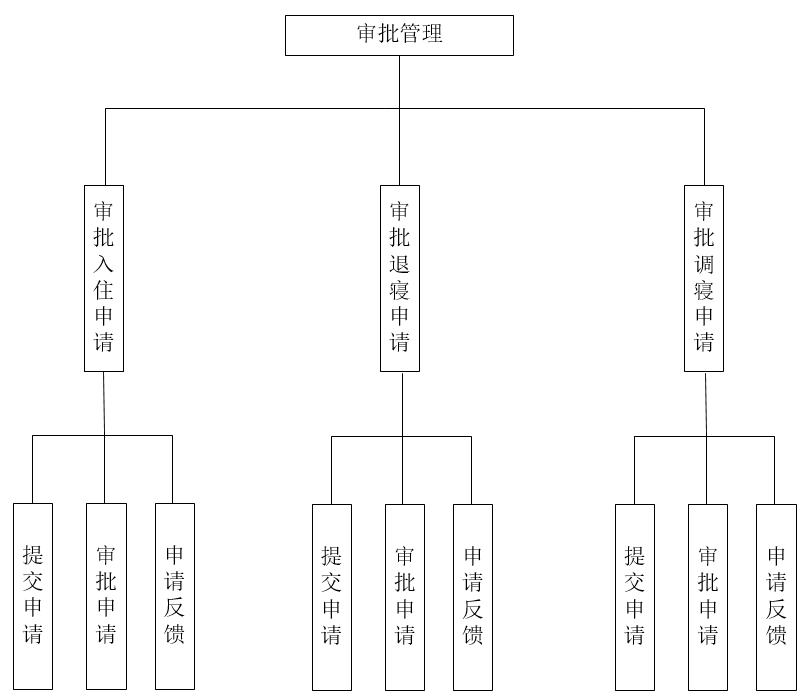


图3.13 审批管理功能结构图

## 3.5 运行环境分析

服务器端

操作系统：Windows Server 2008及以上

CPU：Intel core i3及以上处理器

内存： 4G及以上

硬盘：500G及以上

软件配置：MySQL 5.1及以上版本，.NET Framework 3.5及以上版本，IIS7.0及以上版本

用户端

操作系统：Windows XP及以上Windows系统版本

CPU：Intel core i3及以上处理器

内存：2 GHz及以上

硬盘：120G及以上

浏览器：IE 7.0及以上版本

4 系统设计

## 4.1 数据库设计

### 4.1.1 概念模型设计

经过对宿舍管理系统需求的分析，抽取出系统中一些重要的实体、属性与关联关系，设计了系统的概念模型，即E-R图。如图4.1所示就是一个总体E-R图，包括宿舍楼、房间、学生、级队、辅导员这五个实体对象。其中宿舍楼实体包含楼号与楼层数两个属性，房间实体包含宿舍楼号、楼层、房间类型、房间号、1号铺学生学号、2号铺学生学号、3号铺学生学号、4号铺学生学号等属性。宿舍楼与房间之间是一对多的关系，一个宿舍楼可以有多个房间。学生实体包含学号、姓名、性别、手机号、学位、班级、宿舍楼、房间号、床位等基础属性，辅导员实体包含编号、姓名、宿舍楼、房间号等属性，级队实体包含级队编号和年级两个属性。辅导员与级队之间是一对一的关系，一个辅导员管理一个级队；级队与学生之间是一对多的关系，一个级队包含多个学生。而学生和辅导员跟房间之间也都是一对一的关系。

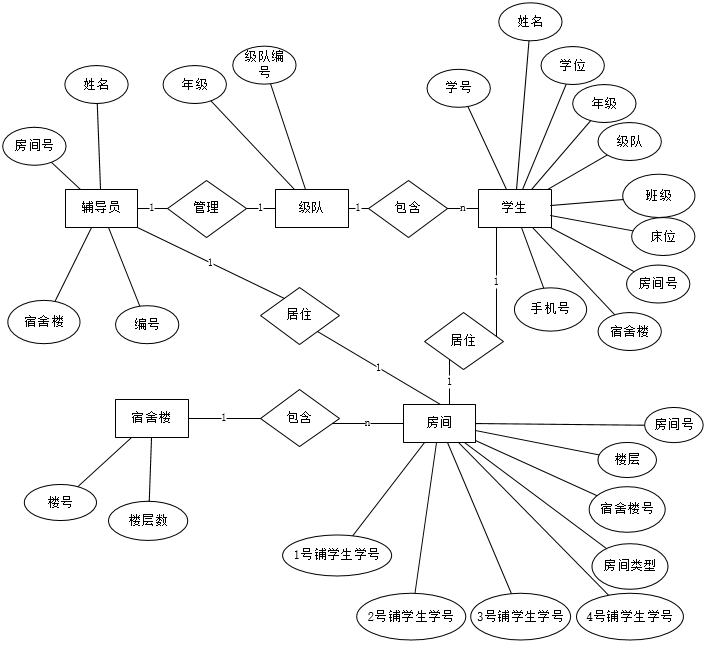


图4.1 总体E-R图

如图4.2所示是一个入住/退寝申请管理的E-R图，包括管理员、入住申请、退寝申请三个实体。管理员实体包含姓名与编号两个属性，入住申请实体包含学生学号、学生姓名、学生学位、学生班级、入住时间、入住楼号、入住房间、入住床位等属性，退寝申请实体包含学生学号、学生姓名、学生学位、学生班级、退寝时间、退寝楼号、退寝房间、退寝床位等属性。管理员与入住申请之间是一对多的关系，管理员可以审批多条入住申请。管理员与退寝申请之间也是一对多的关系，管理员同样可以审批多条退寝申请。

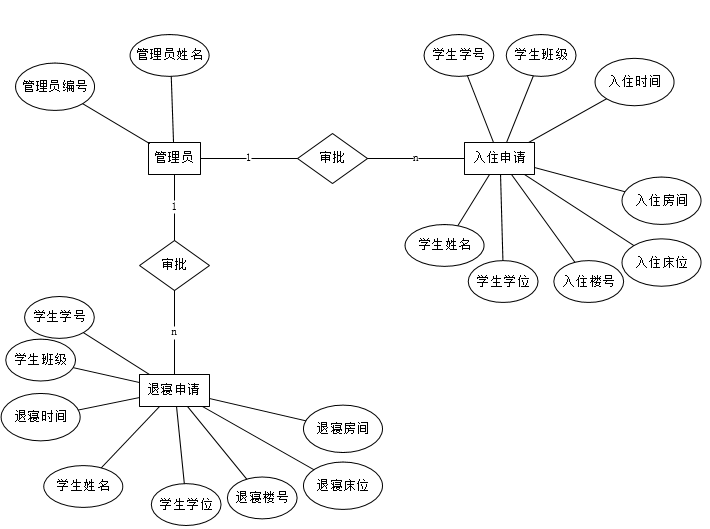


图4.2 入住/退寝申请管理E-R图

### 4.1.2 主要数据表设计

经过概念模型的设计，我们提取出主要实体之间的联系，然后生成数据库表。进行数据表设计时，要满足范式要求，去除数据冗余。此外，在进行数据表设计时还要考虑到开发的便利性和系统效率等因素，所以有时候一定程度的数据冗余可以给开发带来方便，提高系统运行速度。所以既要满足理论设计也要考虑实际开发，这样才能更好地实现系统的开发。

在本模块中将会对宿舍管理系统中主要用到的几个表的设计进行详细阐述说明，包括学生表、房间表、辅导员表、入住申请表、调换寝室申请表。

如表4.1所示是学生表，主要用于存储系统中的学生信息，是系统的基础表。学生表主要记录学生的各项信息，包括学号、姓名、性别、学位、年级、级队、班级、手机、辅导员、宿舍楼、房间、床位、入住时间等字段，其中学号为主键，标识独立的学生信息，辅导员是外键，指向辅导员表的主键Counsellor\_id，级队也是外键，指向级队表的主键Team\_id。

表4.1 学生表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 字段含义 | 数据名称 | 数据类型 | 空值 | 描述 |
| 1 | 学号 | Student\_id | nvarchar(9) | False | 主键 |
| 2 | 姓名 | Student\_name | nvarchar(10) | False |  |
| 3 | 性别 | Student\_gender | int | False |  |
| 4 | 学生类别 | Student\_degree | nvarchar(10) | False |  |
| 5 | 年级 | Student\_grade | int | False |  |
| 6 | 级队id | Team\_id | nvarchar(10) | False | 外键 |
| 7 | 班级 | Student\_class | nvarchar(30) | False |  |
| 8 | 手机 | Student\_phone | nvarchar(11) | False |  |
| 9 | 辅导员id | Counsellor\_id | nvarchar(10) | False | 外键 |
| 10 | 宿舍楼 | Student\_building | int |  |  |
| 11 | 房间 | Student\_room | int |  |  |
| 12 | 床位 | Student\_bed | int |  |  |
| 13 | 入住时间 | Student\_checkintime | datetime |  |  |

如表4.2所示是房间表，主要用于存储系统中的房间信息，包括宿舍楼、楼层、房间号、房间类型、4个床铺分别的学生学号等字段，其中房间ID为主键，唯一标识房间。

表4.2 房间表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 字段含义 | 数据名称 | 数据类型 | 空值 | 描述 |
| 1 | 房间ID | Room\_id | int | False | 主键 |
| 2 | 宿舍楼 | Room\_building | int | False |  |
| 3 | 楼层 | Room\_floor | int | False |  |
| 4 | 房间号 | Room\_number | int | False |  |
| 5 | 房间类型 | Room\_type | int | False |  |
| 6 | 1铺学生学号 | Room\_bed1 | nvarchar(9) |  |  |
| 7 | 2铺学生学号 | Room\_bed2 | nvarchar(9) |  |  |
| 8 | 3铺学生学号 | Room\_bed3 | nvarchar(9) |  |  |
| 9 | 4铺学生学号 | Room\_bed4 | nvarchar(9) |  |  |

如表4.3所示是辅导员表，主要用于存储系统中的辅导员信息，包括辅导员编号、姓名、性别、手机号、管理级队、宿舍楼、房间号、床位等信息，其中辅导员编号为主键，唯一标识辅导员，管理级队为外键，指向级队表的主键Team\_id。

表4.3 辅导员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 字段含义 | 数据名称 | 数据类型 | 空值 | 描述 |
| 1 | 编号 | Counsellor\_id | nvarchar(10) | False | 主键 |
| 2 | 姓名 | Counsellor\_name | nvarchar(10) | False |  |
| 3 | 性别 | Counsellor\_gender | int | False |  |
| 4 | 手机号 | Counsellor\_phone | nvarchar(11) | False |  |
| 5 | 管理级队id | Team\_id | int | False | 外键 |
| 6 | 宿舍楼 | Counsellor\_building | int |  |  |
| 7 | 房间号 | Counsellor\_room | int |  |  |
| 8 | 床位 | Counsellor\_bed | int |  |  |

如表4.4所示是入住申请表，主要用于存储学生的入住申请信息，包括入住信息ID、学生学号、学生姓名、学生性别、学生学位、入住楼号、入住房间、入住床位、入住时间、入住信息等信息，其中入住信息ID为主键，学生学号为外键，指向学生表的主键Student\_id。

表4.4 入住申请表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 字段含义 | 数据名称 | 数据类型 | 空值 | 描述 |
| 1 | 入住信息ID | Checkin\_id | nvarchar(10) | False | 主键 |
| 2 | 学生学号 | Student\_id | nvarchar(9) | False | 外键 |
| 3 | 学生姓名 | Student\_name | nvarchar(10) | False |  |
| 4 | 学生性别 | Student\_gender | int | False |  |
| 5 | 学生学位 | Student\_degree | nvarchar(10) | False |  |
| 6 | 入住楼号 | Checkin\_building | int | False |  |
| 7 | 入住房间 | Checkin\_room | int | False |  |
| 8 | 入住床位 | Checkin\_bed | int | False |  |
| 9 | 入住时间 | Checkin\_time | datetime | False |  |
| 10 | 备注 | Checkin\_detail | nvarchar(50) |  |  |

如表4.5所示是调换寝室申请表，主要用于存储学生的调换寝室申请信息，包括调换信息ID、调换学生学号、调换学生姓名、旧宿舍楼号、旧房间号、旧床位、新宿舍楼号、新房间号、新床位、调换日期、备注等信息，其中调换信息ID为主键，调换学生学号为外键，指向学生表的主键Student\_id。

表4.5 调换寝室申请表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 字段含义 | 数据名称 | 数据类型 | 空值 | 描述 |
| 1 | 调换信息ID | Exchange\_id | nvarchar(10) | False | 主键 |
| 2 | 调换学生学号 | Student\_id | nvarchar(9) | False | 外键 |
| 3 | 调换学生姓名 | Exchange\_stu\_name | nvarchar(10) | False |  |
| 4 | 旧宿舍楼号 | Exchange\_oldbuilding | int | False |  |
| 5 | 旧房间号 | Exchange\_oldroom | int | False |  |
| 6 | 旧床位 | Exchange\_oldbed | int | False |  |
| 7 | 新宿舍楼号 | Exchange\_newbuilding | int | False |  |
| 8 | 新房间号 | Exchange\_newroom | int | False |  |
| 9 | 新床位 | Exchange\_newbed | int | False |  |
| 10 | 调换日期 | Exchange\_date | datetime | False |  |
| 11 | 备注 | Exchange\_note | nvarchar(50) |  |  |

如表4.5所示是调换寝室申请表，主要用于存储学生的调换寝室申请信息，包括调换信息ID、调换学生学号、调换学生姓名、旧宿舍楼号、旧房间号、旧床位、新宿舍楼号、新房间号、新床位、调换日期、备注等信息，其中调换信息ID为主键，调换学生学号为外键，指向学生表的主键Student\_id。

## 4.2 核心模块设计

在本部分将会对宿舍管理系统中用到的主要功能模块进行阐述说明，包括宿舍管理、房间信息管理、学生信息管理、房间类型管理、导员信息管理、审批管理等6个模块。

### 4.2.1 宿舍管理

宿舍管理模块主要就是对宿舍楼信息的管理。管理员能够依据需求对宿舍楼的信息进行相应的操作。如图4.3所示是宿舍管理流程图，其具体步骤如下：

1. 进入宿舍管理模块时首先会显示所有宿舍楼的信息；
2. 选择是否新增宿舍楼信息，如果选择是则填写数据并提交后返回宿舍信息页面，流程结束，选择否则进入第三步；
3. 选择是否修改宿舍楼信息，如果选择是则填写数据、提交，返回宿舍信息页面，流程结束，选择否则进入第四步；
4. 选择是否删除宿舍楼信息，如果选择是则选中需删除的数据并提交，之后返回宿舍信息页面，流程结束，选择否则流程直接结束。

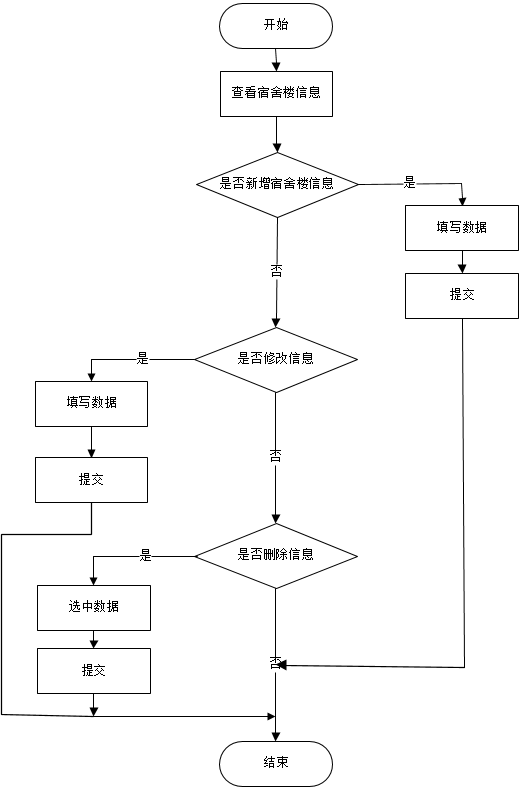


图4.3 宿舍管理流程图

### 4.2.2 房间信息管理

房间信息管理模块是系统中一个重要组成部分，主要管理房间数据，包括居住学生、床位数等，由管理员负责。如图4.4所示，其具体步骤如下：

1. 进入房间信息管理模块时首先会显示房间详细信息页面；
2. 选择是否新增房间信息，选择是则提交数据后结束，选择否则进入第三步；
3. 选择是否查询房间信息，选择否则流程图结束，选择是则可选择查询方式，包括按照楼号、床位数、学号、宿舍号查询，提交后显示结果页面。之后可选择是否删除房间信息，选择是则选中欲删除的数据并提交即可，之后返回房间详细信息页面。

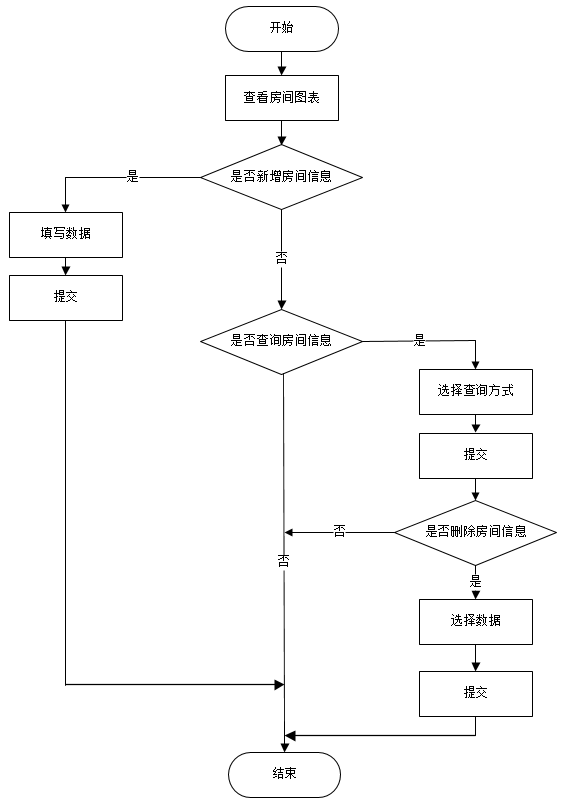


图4.4 房间信息管理流程图

### 4.2.3 学生信息管理

学生信息管理是宿舍管理系统中比较重要的一个模块，其流程图如图4.5所示。

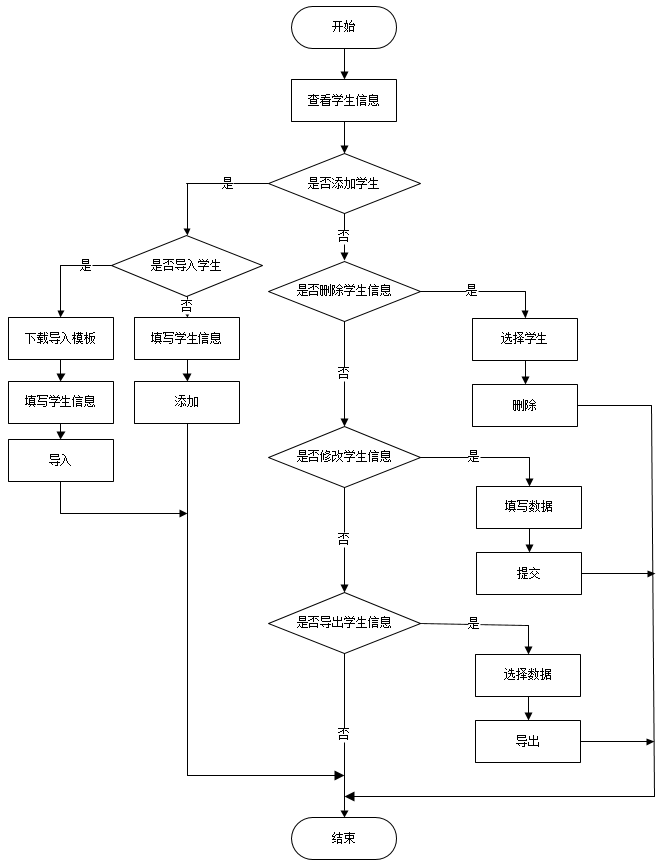


图4.5 学生信息管理流程图

具体步骤如下：

1. 进入学生信息管理模块后，显示学生信息页面；
2. 选择是否添加学生，选择是则继续选择是否选择导入方式，如果选择以导入方式则需要下载导入模板后填写之后导入即可，选择不以导入方式则需要在页面上填写学生信息并添加；
3. 如果选择不添加学生则进入下一步，选择是否删除，选择是则选中欲删除的学生信息并删除即可，之后返回学生信息界面；
4. 如果选择不删除学生信息则进入下一步，选择是否修改学生信息，选择是则填写修改数据并提交即可，之后返回学生信息界面；
5. 如果选择不修改学生信息则进入下一步，选择是否导出学生信息，选择是需要选中欲导出的数据并导出，之后返回学生信息界面；选择否则流程结束。

### 4.2.4 房间类型管理

房间类型管理模块主要就是对房间信息的管理。管理员能够依据需求对房间类型的信息进行相应的操作。如图4.6所示是房间类型管理流程图。

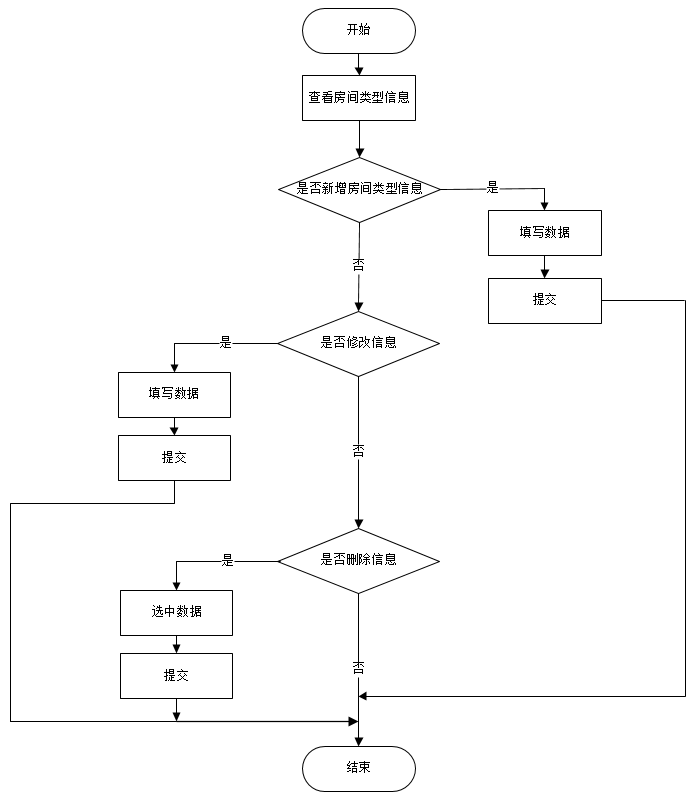


图4.6 房间类型管理流程图

### 4.2.5 导员信息管理

导员信息管理模块主要就是对房间信息的管理。管理员能够依据需求对导员信息进行相应的操作。如图4.7所示是导员信息管理流程图，其具体步骤如下：

1. 进入导员信息管理模块时首先会显示所有导员的信息；
2. 选择是否新增导员信息，如果选择是则填写数据并提交后返回导员信息页面，选择否则进入第三步；
3. 选择是否修改导员信息，如果选择是则填写数据、提交，返回导员信息页面，选择否则进入第四步；
4. 选择是否删除导员信息，如果选择是则选中需删除的数据并提交，之后返回导员信息页面，选择否则流程结束。

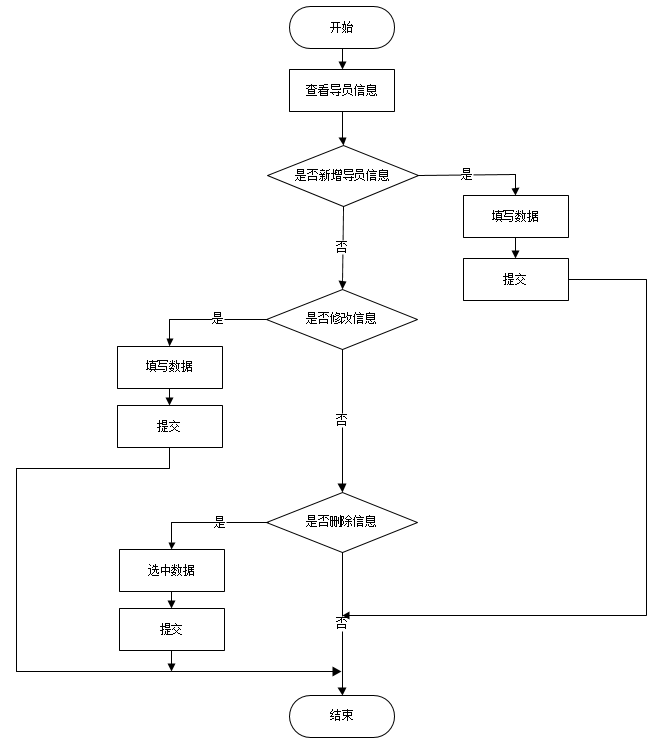


图4.7 导员信息管理流程图

### 4.2.6 审批管理

审批管理是需要管理员与学生交互的一个模块，主要功能是实现学生的申请提交与管理员的申请审批、反馈。如图4.8所示是审批管理流程图，其主要步骤如下：

1. 首先学生选择申请审批的类型，包括入住申请、退寝申请、调寝申请；
2. 选择完毕后需要下载对应的申请表；
3. 学生填写对应的申请表并提交申请；
4. 管理员审批申请后反馈结果；
5. 如果反馈结果为申请未成功则需要学生重新填写申请表并提交，如果成功则流程结束。

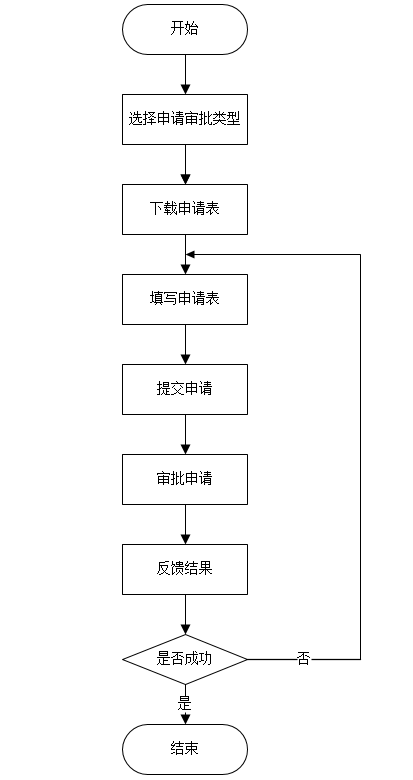


图4.8 审批管理流程图

5 系统实现

软件学院宿舍管理系统实现了宿舍的信息化管理，规避了没有此系统时的众多问题。管理员能够简单的一览所有宿舍的信息，并使用了Echarts图表来更加直观的表示宿舍信息，大大提高了工作效率。

如图5.1所示是系统的主界面，系统主要分为宿舍管理、房间信息管理、学生信息管理、房间类型管理、导员信息管理、审批管理六个功能模块，其中审批管理又分为申请入住、申请退寝、申请调换房间等三个部分，而入住与退寝又分别分为本科生、研究生两个部分。

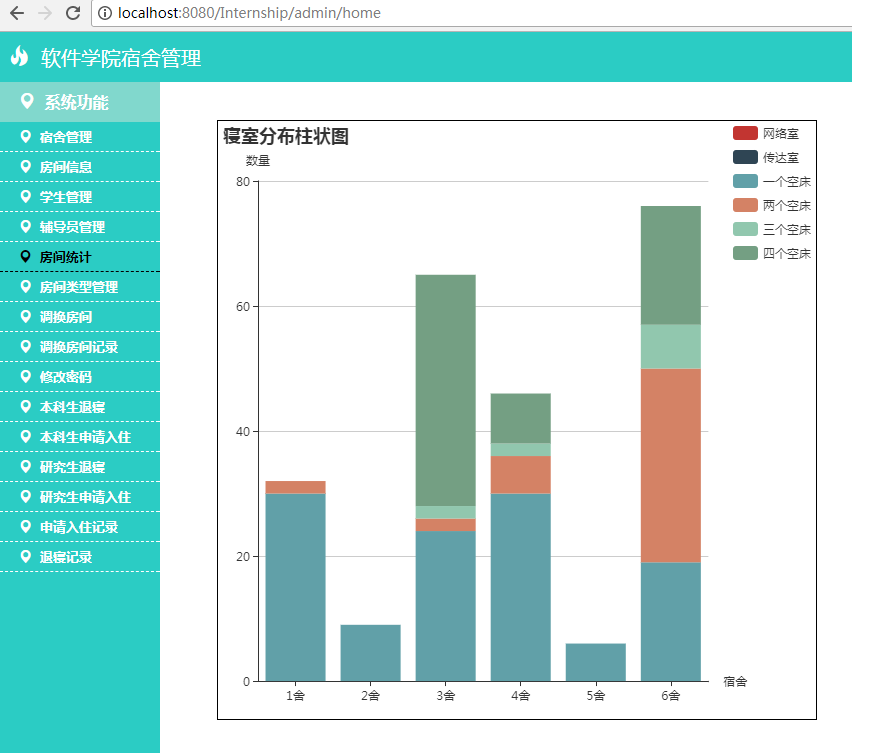


图5.1 系统主界面

## 5.1 宿舍管理

点击左侧宿舍管理按钮即可进入宿舍管理模块，如图5.2所示。模块宿舍管理是对系统中所有宿舍楼信息的管理，点击左上角的加号能执行添加宿舍楼的功能，每个宿舍楼左边有一个管理房间按钮，点击即可查看每栋宿舍楼里的房间信息，如图5.3所示。可以一览房间大致信息，也可查询对应房间号查看详细信息。



图5.2 宿舍管理界面



图5.3 房间信息界面

房间信息查询核心代码如下：

List<Room> list=roomService.getAllRoomByStu(sid);

for(int i=0;i<list.size();i++){

Room ro=list.get(i);

if(ro.getRoomType()==0){

int count=0;

if(ro.getRoomBed1()==0){

count++;

}

if(ro.getRoomBed2()==0){

count++;

}

if(ro.getRoomBed3()==0){

count++;

}

if(ro.getRoomBed4()==0){

count++;

}

ro.setRoomBed1(count);

}

list.set(i, ro);

}

return list;

## 5.2 房间信息管理

如图5.4所示，点击左侧房间信息按钮即可进入房间信息管理模块，主要可以通过不同条件查询对应的宿舍信息，可以通过楼号、楼层、空床位数、所住学生学号等信息查询房间信息，房间小图中红色的格子代表已经住了学生，绿色的就是空床位。点击对应的房间可以出现一个弹框，显示居住学生的信息，包括姓名、学号、电话、年级、辅导员等信息。与宿舍管理模块中的房间信息查看方式，这样子显得更加直观。

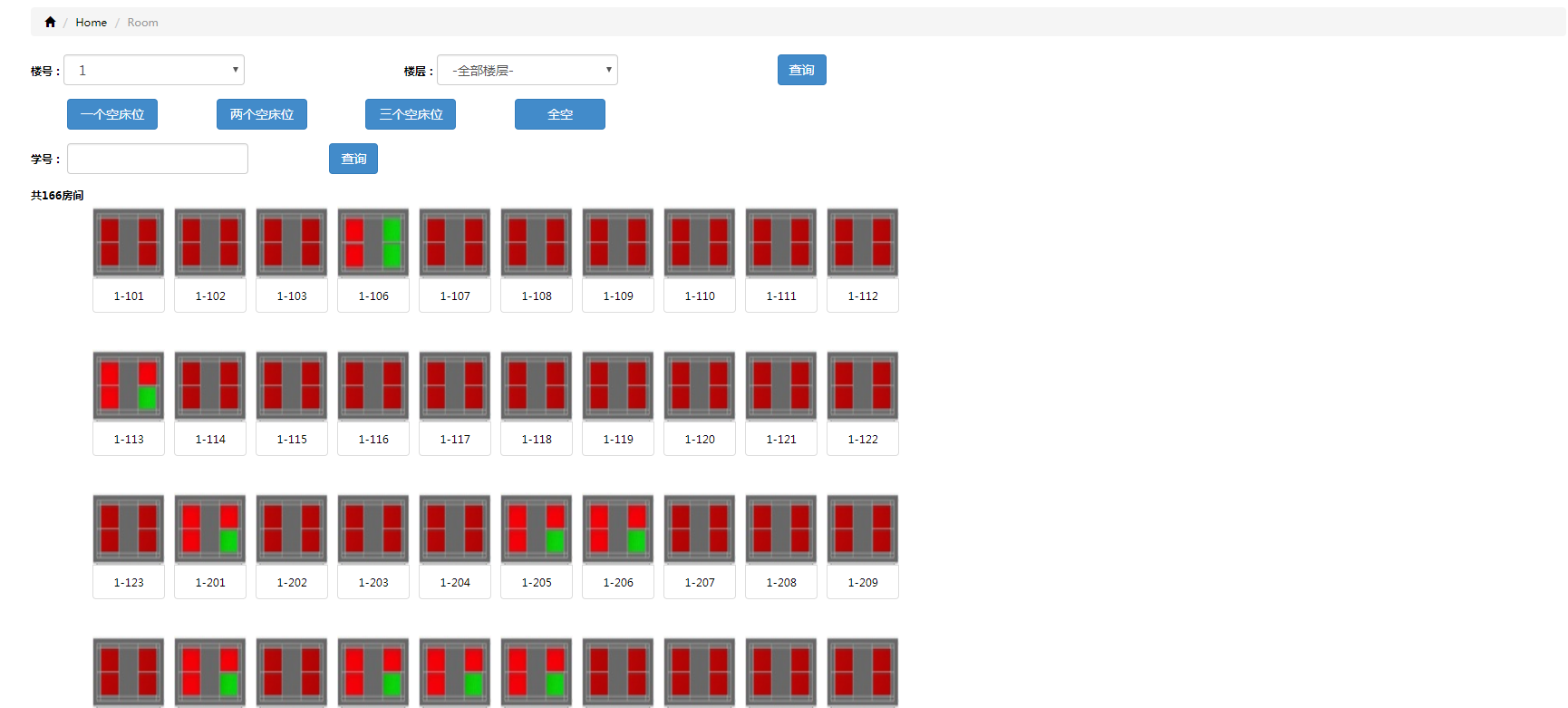


图5.4 房间信息管理界面

而如图5.5所示则是用Echarts中的柱状图表示各个宿舍楼的房间信息。使用了不同的色块表示不同的空床位数，将鼠标悬浮在上边可以显示详细的数据。这样能让后勤管理人员更加直观的查看房间信息，提高工作效率。

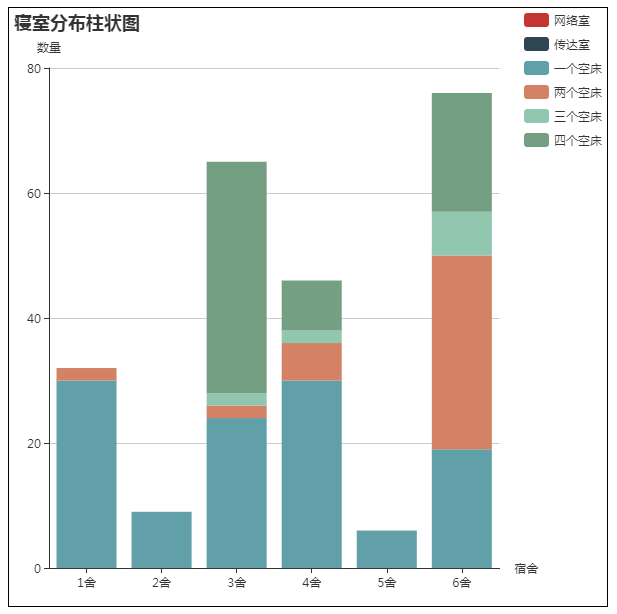


图5.5 Echarts图界面

Echarts柱状图核心代码如下：

title: {

text: '寝室分布柱状图'

},

tooltip : {

trigger: 'axis',

axisPointer : { // 坐标轴指示器，坐标轴触发有效

type : 'cross' // 默认为直线，可选为：‘line’|‘shadow’

}

},

legend: {

data:["网络室","传达室","一个空床","两个空床","三个空床","四个空床"],

right: 'right'

},

grid: {

left: '3%',

right: '18%',

bottom: '3%',

containLabel: true

},

## 5.3 学生信息管理

如图5.6所示，点击左侧学生管理按钮即可进入学生信息管理模块。主要包括添加学生信息、导入学生信息、导出学生信息、下载导入模板、删除学生信息、修改学生信息、查询学生信息等几个功能。其中查询功能使用了模糊查询，可按照学号、姓名、联系方式、房间号查询；而删除也可以实现批量删除和单条信息删除。

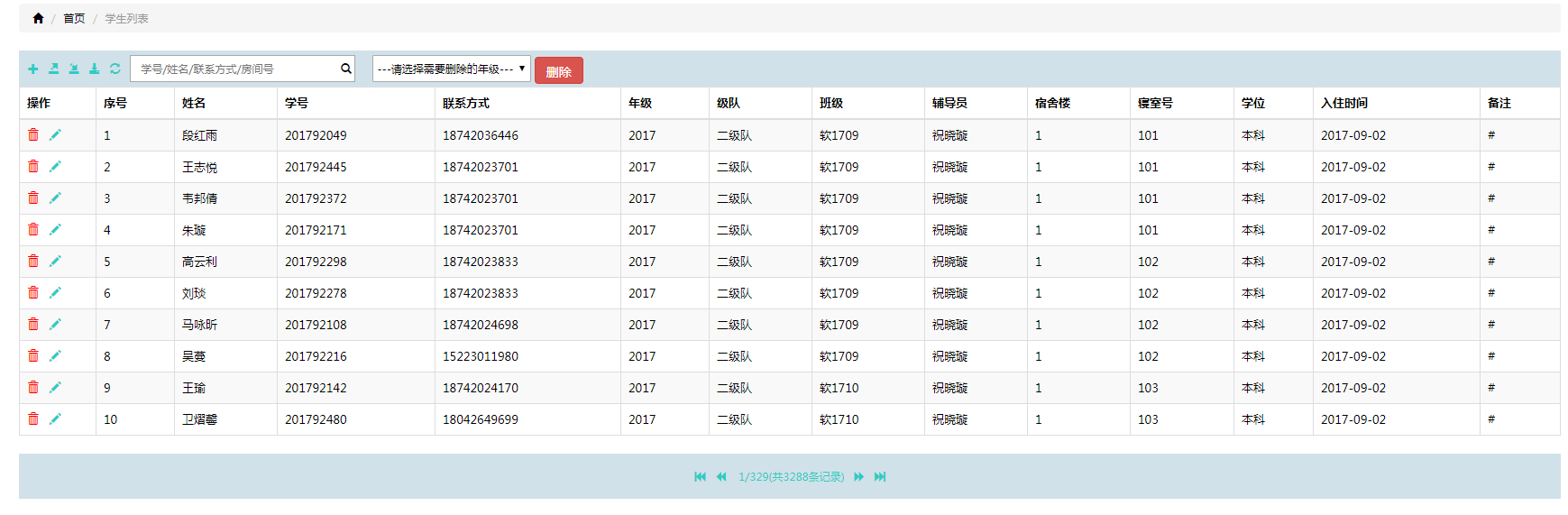


图5.6 学生信息管理界面

下载学生模板核心代码如下：

String filename = request.getParameter("filename");

String path = request.getServletContext().getRealPath("")+"/file";

File downloadFile = new File(path, filename);

request.setCharacterEncoding("utf-8");

BufferedInputStream bis = null;

BufferedOutputStream bos = null;

long fileLength = downloadFile.length();

response.setContentType("application/octet-stream");

response.setHeader("Content-disposition", "attachment;filename="+URLEncoder.encode(filename, "UTF-8"));

response.setHeader("Content-length",String.valueOf(fileLength));

bis = new BufferedInputStream(new FileInputStream(downloadFile));

bos = new BufferedOutputStream(response.getOutputStream());

byte[] buff = new byte[2048];

int byteRead;

while ( -1 != (byteRead = bis.read(buff, 0, buff.length))){

bos.write(buff,0,byteRead);}

bis.close();

bos.close();

## 5.4 房间类型管理

如图5.7所示，点击左侧房间类型管理按钮即可进入房间类型管理模块。这个模块的功能主要是添加、删除房间类型信息。其中添加房间类型时需要限定床位数，删除房间类型时要确保房间信息中不存在此房间类型。



图5.7 房间类型管理界面

## 5.5 导员信息管理

如图5.8所示，点击左侧辅导员管理按钮即可进入导员信息管理模块。这个模块的功能比较简单，主要是添加、修改辅导员信息。分别点击对应按钮即可进入相应页面。



图5.8 导员信息管理界面

## 5.6 审批管理

审批管理分为三个部分：退寝、入住和调换房间。如图5.9所示，点击研究生申请入住按钮会弹出一个页面，包含三个申请单，需要在页面上填写相应信息并打印下来提交给后勤管理人员，本科生入住退寝、研究生退寝和图5.9类似，提交后会将记录存入数据库。



图5.9 研究生申请入住界面

而调换房间则需要管理员在页面上进行信息输入，如图5.10所示，输入学号后可以查看学生基本信息确保输入正确，提交之后也会将记录存入数据库。

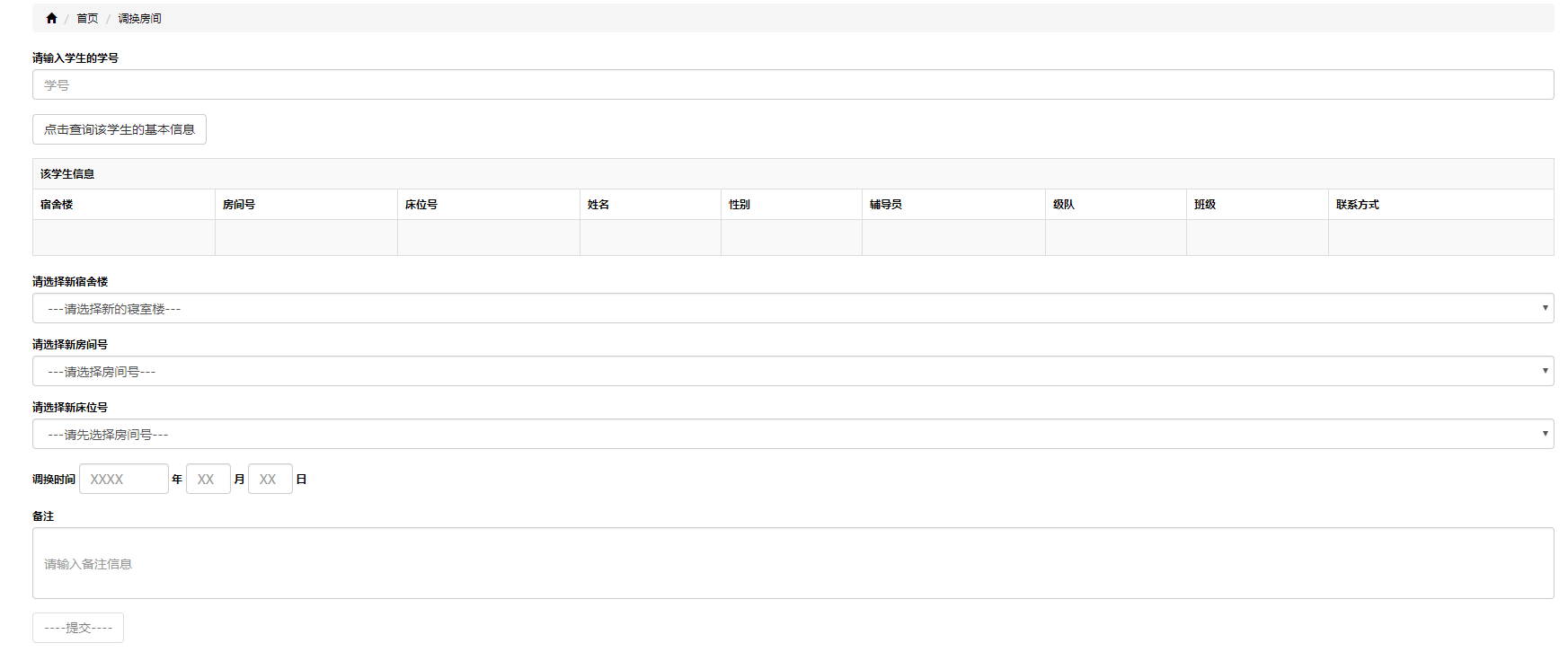


图5.10 查看学生基本信息界面

6 系统测试

经过系统详细设计与编码实现后，系统功能已全部实现，最后需要对整个系统进行软件测试。软件测试是软件开发的重要组成部分，其贯穿整个软件开发过程。软件测试是保证软件质量和可靠性的重要手段，通过在测试可以发现开发中的各种错误或者软件缺陷，以最少的时间和人力找出软件中潜在的错误和缺陷。软件测试是软件开发过程中必不可少的环节，如果缺乏测试，开发出来的低质量的软件会给客户在实际使用过程中带来糟糕的使用体验，并且会增加后期软件维护的成本。

在软件测试理论中存在四种经典的测试模型用以描述测试活动与开发阶段之间的关系，分别是V模型、W模型、X模型和H模型[14]。V模型是目前软件测试模型中较广泛使用的模型。V模型中测试活动开始于编码之后，单元测试对应于开发中的详细设计阶段，集成测试对应概要设计，确认测试与系统测试对应需求分析与系统设计，验收测试对应用户需求。将测试置于开发完成之后，需求分析上的错误不能尽早发现，会产生高时间和金钱成本bug修复的情况。W模型把各阶段的测试活动分布到开发的各阶段，伴随整个开发生命周期，开发和测试同步进行。但是W模型不适用于迭代开发，或者自发性和变更多的项目开发。X模型是以一个小的程序片段为单位进行独立测试，测试通过后将多个测试片段集成起来并进行集成测试，通过后进行封装形成模块或者提交给用户。X模型还提出了不事先计划的探索性测试，有助于发现难以发现的错误。H模型则将测试作为一个完全独立的活动，贯穿整个开发周期，与其他开发流程并行，只要条件准备充分，到达某个就绪点，测试则可以开始进行。

软件测试的方法分为黑盒测试和白盒测试，其中白盒测试侧重于测试代码逻辑，在开发各个单元阶段可以并行进行。黑盒测试则主要测试系统的功能，不关注程序内部结构和逻辑[15]。白盒测试包括代码检查、逻辑覆盖法、基本路径测试法和路径覆盖等方法，其中逻辑覆盖又包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖，这六种覆盖分别表示不同覆盖程度的测试，覆盖代码逻辑的能力由弱到强[16]。所以，白盒测试的主要目的是保证每一条代码语句都能被测试到，并且能判断出正确与否，从代码内部确保程序代码的正确性。黑盒测试包括等价类划分、边界值分析法、错误推测法、因果图法、判定表驱动法、正交实验设计法、功能图法和场景法等设计方法。其中等价类划分是将测试输入划分成不同的等价类，并在每个等价类中选取一些点进行测试。边界值分析法是选取测试输入域边界内外以及边界上的值进行测试。判定表驱动方法也是一种比较常用的测试方法，列举出所有的条件桩、动作桩以及由条件桩组合形成的条件项，以此推断出各条件项下所应该采取的动作项。

本章将采用黑盒测试方法对系统进行测试，测试用例表是测试的重要工具，用于记录测试操作、预期结果和实际结果。如表6.1是房间信息管理模块中进行的部分测试用例，主要测试新增、删除、查询、筛选、查看图表等功能能不能正常实现。

表6.1 房间信息管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试用例名称 | 测试数据及操作 | 预期结果 | 实际结果 |
| 1 | 新增房间测试 | 填写完整的房间信息 | 房间信息保存成功 | 弹出“添加房间成功”提示，房间信息保存成功 |
| 2 | 新增房间测试 | 不填写完整的房间信息 | 房间信息不能保存 | 提示信息未填完整，不能保存 |
| 3 | 删除房间测试 | 选择房间点击删除 | 房间信息被删除 | 弹出“房间删除成功”，房间信息删除成功 |
| 4 | 查询房间测试 | 输入房间号查询 | 获取对应房间信息成功 | 表中显示对应房间信息，查询成功 |
| 5 | 筛选房间测试 | 选择楼号、楼层、空床位数后点击查询 | 显示符合条件的所有房间 | 网页中显示符合条件的房间信息，筛选成功 |
| 6 | 房间图表测试 | 点击“房间统计”按钮 | 显示房间信息的柱状图且鼠标悬浮在对应位置能显示详细数据 | 房间柱状图成功显示，且鼠标悬浮处也能显示正确的数据 |

如表6.2所示是学生信息管理模块的测试用例，主要进行新增学生、删除单个学生、按年级删除学生、查询房间、下载导入模板、导入学生信息、导出学生信息、修改学生信息的测试。

表6.2 学生信息管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试用例名称 | 测试数据及操作 | 预期结果 | 实际结果 |
| 1 | 新增学生测试 | 填写完整的学生信息 | 学生信息保存成功 | 弹出“添加学生成功”提示，学生信息保存成功 |
| 2 | 新增学生测试 | 不填写完整的学生信息 | 学生信息不能保存 | 提示信息未填完整，不能保存 |
| 编号 | 测试用例名称 | 测试数据及操作 | 预期结果 | 实际结果 |
| 3 | 删除单个学生测试 | 选择学生点击删除 | 学生信息被删除 | 弹出“学生删除成功”，学生信息删除成功 |
| 4 | 按年级删除学生测试 | 选择年级点击删除 | 删除对应年级学生信息成功 | 弹出“按年级删除学生成功”，学生信息删除成功 |
| 5 | 查询房间测试 | 输入学号/姓名/房间号信息点击查询 | 显示符合条件的所有学生信息 | 网页中显示符合条件的学生信息，查询成功 |
| 6 | 下载导入模板测试 | 点击“下载学生信息导入模板”按钮 | 学生信息导入模板下载到浏览器文件夹中 | 模板成功下载 |
| 7 | 导入学生信息测试 | 点击“导入学生信息”按钮，选中模板表，选择导入 | 学生信息成功导入 | 弹出“学生信息导入成功”，导入成功 |
| 8 | 导出学生信息测试 | 点击“导出学生信息”按钮，选中导出路径，点击确定 | 学生信息按模板表的形式导出到选中的文件夹中 | 弹出“学生信息导出成功”，导出学生信息成功 |
| 9 | 修改学生信息测试 | 选中一条学生信息，点击修改按钮，填写新的学生信息并确定 | 学生信息修改成功 | 弹出“学生信息修改成功”，修改学生信息成功 |

如表6.3所示是审批管理模块的测试用例。

表6.3 审批管理测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试用例名称 | 测试数据及操作 | 预期结果 | 实际结果 |
| 1 | 本科生退寝测试 | 点击“本科生退寝”按钮，填写对应信息 | 弹出本科生退寝表格成功，能成功填写信息，签名部分不能填写 | 成功弹出表格并且除签名外的信息填写可以实现 |
| 2 | 研究生退寝测试 | 点击“研究生退寝”按钮，填写对应信息 | 弹出研究生退寝表格成功，能成功填写信息，签名部分不能填写 | 成功弹出表格并且除签名外的信息填写可以实现 |
| 编号 | 测试用例名称 | 测试数据及操作 | 预期结果 | 实际结果 |
| 3 | 本科生入住测试 | 点击“本科生入住”按钮，填写对应信息 | 弹出本科生入住表格成功，能成功填写信息，签名部分不能填写 | 成功弹出表格并且除签名外的信息填写可以实现 |
| 4 | 研究生入住测试 | 点击“研究生入住”按钮，填写对应信息 | 弹出研究生入住表格成功，能成功填写信息，签名部分不能填写 | 成功弹出表格并且除签名外的信息填写可以实现 |
| 5 | 调换寝室测试 | 点击“调换”按钮，填写对应信息后提交，且输入学号后能查询学生基本信息 | 能成功输入信息并提交，且能在输入学号后查询学生基本信息 | 提交信息成功，且能成功按照输入学号查询学生基本信息 |
| 6 | 查看申请入住信息测试 | 点击“申请入住记录”按钮 | 页面中显示学生申请入住信息 | 成功显示学生申请入住信息 |
| 7 | 查看申请退寝信息测试 | 点击“申请退寝记录”按钮 | 页面中显示学生申请退寝信息 | 成功显示学生申请退寝信息 |
| 8 | 查看调换寝室信息测试 | 点击“调换寝室记录”按钮 | 页面中显示调换寝室信息 | 成功显示调换寝室信息 |

以上是系统在测试阶段使用的部分测试用例，通过测试发现了问题并已经解决，整个系统运行正确，现已经投入运行。

结 论

宿舍管理系统功能基本已经实现，已在软件学院内部运行。在此期间与后勤负责人员多次沟通协商，确定完善系统需求，使得系统平台充分满足学院实际业务需求和日常办公。经过一段时间的试运行，我们得到比较好的反馈，极大提高了工作效率。

本文详细地阐述了软件学院宿舍管理系统的设计与实现，系统分为宿舍管理、房间信息管理、学生信息管理、房间类型管理、导员信息管理、审批管理等6个模块，本文对这六个模块进行了详细描述。宿舍管理主要用于对宿舍楼信息的增删改查，一般不会进行特别大的改动。房间信息管理主要是在对房间需要有操作时使用，其中使用了Echarts来更加直观的显示房间信息。学生信息管理包括对学生信息的增删改查，以及信息的导入导出功能。房间类型管理主要就是实现对房间类型信息的管理，在房间信息管理中会用到。导员信息管理也主要就实现了对导员信息的增删改查。审批管理用于管理学生提交的申请，包括退寝、申请入住、申请调寝。系统实现采用了SSM框架，框架的使用将系统代码进行分层，降低代码之间的耦合性，增强的系统的可扩展性和维护性。在开发过程中，充分利用的面向对象思想和软件工程中开发软件的知识，理论知识得以运用于实践应用。

系统平台的使用保证在项目从立项到最后一个环节，每个过程都严格经历项目计划、项目实施、项目检查和项目处理等过程。在后续的工作中我们将继续对系统功能进行完善和优化。

参 考 文 献

[1] 冯子荻,舒家先.管理信息系统的发展方向及实现技术的研究[J].北京印刷学院学报,2017, 25(07):156-158.

[2] 廖威.计算机管理信息系统现状及未来发展方向[J].信息与电脑(理论版),2017(09):31-3

3.

[3] 彭玉涛,蒋熔.学生宿舍管理系统的设计浅析[J].科技传播,2017,9(23):112-113.

[4] 李洋.SSM框架在Web应用开发中的设计与实现[J].计算机技术与发展,2016,26(12):190-1

94.

1. 任 仁. 基于 MVC 模式的 Struts 和 Spring 框架的分析及整合[J]. 硅谷, 2008(21): 5

1.

[6] Mahir Kaya, Altan Koçyiğit, P. Erhan Eren.An adaptive mobile cloud computing framework using a call graph based model[J]. Journal of Network and Computer Applications,2016,65.

[7] 楼程伟,陈丽红.关于计算机编程思想与AOP编程思想的研究[J].电脑知识与技术,2015,11

(24):52-53.

[8] 刘高军,夏景隆.基于 Spring MVC和iBATIS框架的研究与应用[J].计算机安全,2012(7):25

-30.

[9] 荣艳冬.关于Mybatis持久层框架的应用研究[J].信息安全与技术,2015,6(12):86-88.

[10] 王子毅,张春海.基于ECharts的数据可视化分析组件设计实现[J].微型机与应用,2016,3

5(14):46-48.

[11] 周从军.基于三层架构的软件体系设计与应用[J].信息与电脑(理论版),2017(15):98-10

0.

[12 ]陈伟鑫.基于B/S架构的文档管理系统在电力企业中的应用[J].工程技术研究,2017(03),

237-238.

[13] Shuli Yuwen,Kaifei Wang. A Method of Data Flow Diagram Drawing Based on Word Segmentation Technique[M].Berlin:Springer Netherlands,2014.

[14] 朱敏.计算机软件测试技术研究现状综述[J].电脑知识与技术,2017,13(24):86-87.

[15] 武昭宇,张月琴,阎华.软件测试方法的研究与应用[J].太原理工大学学报,2016,47(03):3

79-383.

[16] Sébastien Bardin,Omar Chebaro,Mickaël Delahaye,Nikolai Kosmatov.An All-in-One Toolkit for Automated White-Box Testing[M].Berlin:Springer International Publishing,2014.

致 谢

首先感谢马瑞新老师在论文写作期间对我的教导和照顾，在实验室的学习项目开发过程中老师对我的悉心教导、传授知识，给我机会能去学习各种新技术以及参加一些实际的项目开发，给我提供一个很好的学习平台，让我能在专业技能上得到很大的提升，也为我增加了一些实际项目开发经验，让我受益匪浅。此外，在软件学院宿舍管理系统的开发过程中，也感谢马瑞新老师在需求分析和系统设计的实现阶段对我提出了很多指导意见并帮我解决了许多复杂的问题。在论文撰写阶段，感谢老师悉心指导，多次对我的论文提出了修改意见。

其次，也要感谢我在实验室的学长学姐们对我的照顾。初入实验室时对大部分技术都不了解，学长学姐对我的悉心指导让我能快速融入实验室的氛围，大家浓厚的学习热情也感染了我，让我在学习新知识方面更加如鱼得水。

同时，我还要感谢我在本科生阶段学习期间教授课程的各位老师，他们深厚的专业知识和严谨的学术态度值得我一生去学习。其次，还要感谢一起学习的同学们，感谢他们在我的学习和工作过程中对我的帮助，帮我解决了很多困难的问题，从他们身上我也学习到了很多优秀的品质。还要感谢和我一起生活了四年的室友，在生活中能包容我、帮助我，陪伴我一起生活、学习、成长。

此外，感谢我的父母一直为我创造了良好的环境、让我接受良好的教育，在人生的道路上为我指引前进的方向，并且一直都会对我自己的决定给予支持。

最后，感谢各位评审老师对论文提出的意见和建议。