**诚 信 声 明**

我声明，所呈交的毕业论文是本人在老师指导下进行的研究工作及取得的研究成果。据我查证，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得其他教育机构的学位或证书而使用过的材料。我承诺，论文中的所有内容均真实、可信。

毕业论文作者签名： 签名日期： 年 月 日

高校请假管理系统

随着网络技术的发展，传统的高校请假模式也在发生着变化，如今随着网络，计算机，手机的普及，传统的纸质请假流程既需要去线下找到辅导员审批，又需要与任课教师沟通请假时间，存在着诸多弊端。在此背景下本论文旨在开发能够线上解决高校请假流程的系统，是一套简化学生请假流程、提高教师和行政人员工作效率的数字化解决方案。本系统通过集成自动化的请假申请、审批、记录功能，使得学生的请假流程更为透明、高效。系统支持多级审批流程，包括辅导员初审、院系审批以及超期请假学院院长最终审查。此外，系统还提供了任课教师接口，与传统拿到请假条报备任课教师不同，能够自动化的将学生对应课程请假信息给与到任课教师端，不仅可以减少传统纸质请假单的使用，还可以提高处理速度和准确性，确保了教育管理工作的顺畅进行。

请假管理；高校管理；；多级审批

University leave management system

**Abstract：**With the development of network technology, the traditional leave application model in colleges and universities is also changing. Now with the popularization of the Internet, computers, and mobile phones, the traditional paper leave application process requires not only going to the counselor offline for approval, but also contacting the teacher. There are many drawbacks to communicating about leave time. In this context, this paper aims to develop a system that can solve the leave process in colleges and universities online. It is a digital solution that simplifies the student leave process and improves the work efficiency of teachers and administrators. This system integrates automated leave application, approval, and recording functions to make the student leave process more transparent and efficient. The system supports a multi-level approval process, including preliminary review by counselors, departmental approval, and final review by the dean of the college for overdue leave. In addition, the system also provides a teacher interface. Different from the traditional method of obtaining leave requests to report to teachers, it can automatically send students' leave information for corresponding courses to the teachers, which not only reduces the use of traditional paper leave requests, but also It can improve the processing speed and accuracy and ensure the smooth progress of education management work.

**Keywords:** Leave management; university management; multi-level approval

**目 录**

[1 绪论 1](#_Toc14086)

1. [1.1 开发背景、意义 1](#_Toc9132)
2. [1.2 开发环境 2](#_Toc10797)
3. [1.3 本文工作和内容安排 2](#_Toc23042)

[2 相关技术介绍 4](#_Toc18024)

1. [2.1 前端技术框架 4](#_Toc31669)

[2.1.1 Layui 框架 4](#_Toc25027)

[2.1.2 Layuimini框架模版 4](#_Toc32108)

[2.1.3 jQuery 5](#_Toc25061)

1. [2.2 后端开发技术 6](#_Toc27465)

[2.2.1 SpringBoot 6](#_Toc21281)

[2.2.2 Maven 7](#_Toc22083)

[2.2.3 Activiti7 7](#_Toc2252)

1. [2.3 系统存储介质 8](#_Toc14734)

[2.3.1 MySQL 8](#_Toc6565)

[2.3.2 Redis 8](#_Toc32766)

1. [2.4 系统整体技术架构 9](#_Toc24588)

[3 需求分析 10](#_Toc20934)

1. [3.1 功能需求分析 10](#_Toc9560)

[3.1.1 管理员需求分析 10](#_Toc21774)

[3.1.2 学生用户需求分析 11](#_Toc6296)

[3.1.3 辅导员用户需求分析 12](#_Toc8228)

[3.1.4 系主任用户需求分析 13](#_Toc29812)

[3.1.5 院长用户需求分析 14](#_Toc21084)

[3.1.6 任课教师需求分析 14](#_Toc5392)

1. [3.2 系统非功能需求分析 15](#_Toc12510)

[4 系统概要设计 16](#_Toc1571)

1. [4.1 系统功能概要 16](#_Toc20210)
2. [4.2 系统工作流程 17](#_Toc4765)

[5 数据库设计 19](#_Toc17796)

1. [5.1 概念结构设计 19](#_Toc27274)

[5.1.1 系统实体E-R图 19](#_Toc23544)

[5.1.2 全局E-R图 21](#_Toc22813)

1. [5.2 逻辑物理结构设计 22](#_Toc21648)

[6 系统详细设计 26](#_Toc5126)

1. [6.1 系统整体架构 26](#_Toc24803)
2. [6.2 系统模块设计 27](#_Toc13952)

[6.2.1 登录模块 27](#_Toc10046)

[6.2.2 找回密码模块 27](#_Toc19893)

[6.2.3 导入信息模块 27](#_Toc18409)

[6.2.4 请假申请模块 28](#_Toc18801)

[6.2.5 请假审批模块 28](#_Toc2462)

[7 系统功能实现与展示 30](#_Toc31545)

1. [7.1 登录模块 30](#_Toc20152)

[7.1.1 用户登录模块 30](#_Toc24377)

[7.1.2 修改密码模块 32](#_Toc16615)

1. [7.2 管理员模块 36](#_Toc1464)

[7.2.1 组织结构管理模块 36](#_Toc20327)

[7.2.2 用户管理模块 37](#_Toc26849)

[7.2.3 请假管理模块 43](#_Toc15374)

1. [7.3 学生模块 45](#_Toc6695)
2. [7.4 审批人员模块 50](#_Toc5549)

[7.4.1 辅导员审批 51](#_Toc19351)

[7.4.2 系主任/院长审批 55](#_Toc12951)

1. [7.5 教师模块 59](#_Toc7369)

[结论 65](#_Toc5201)

[致谢 66](#_Toc4180)

[附录A 67](#_Toc9644)

[参考文献 68](#_Toc30194)

# **绪论**

## **开发背景、意义**

高校的请假管理是高校内部实施的一套规范学生请假流程的管理制度。它涵盖从请假申请提交、审核、批准到记录归档的整个流程，旨在确保学生的请假行为合理化、规范化，并保持教学管理的高效性和便利性。

随着网络化、信息化的发展,越来越多的通信技术和计算机技术被广泛地应用于高校的教学与管理中，传统的请假方式存在各种弊端,例如:手续复杂、耗时较长、辅导员审批效率低等[1]。

传统的请假流程通常涉及手写请假条、纸质文件的提交与审批，这不仅耗时而且容易出现信息丢失或延误的情况。数字化的请假管理系统可以显著提高处理速度，确保数据的准确传递，并提升整个流程的透明度。在传统管理方式下，学生需要线下到辅导员处签字，若请假流程需要多级审批，那么请假的复杂度将会更高，并且由于普通传统请假可能缺乏相关记录，教师和行政人员难以掌握学生的请假情况。对于任课教师来说，教师本身并不处于请假审批的流程之中，但传统的请假管理中需要学生将请假条交到任课教师手中，不但需要学生亲自或委派其他人来递交请假条，任课教师也需要自行记录请假信息，容易出现疏漏。

本系统的意义就在于通过自动化的请假申请和审批流程，减少了人为操作的环节，从而提高了处理速度和准确性。数字化记录也避免了纸质文档可能带来的信息丢失或误差，全程线上进行大大减少了师生的办理时间，也能充分实现多级审批的需求，不仅解决了管理的实际问题，也推动了教育管理现代化的进程，对学校的长远发展和日常运营均具有深远影响。

## **开发环境**

(1) 操作系统：WINDOWS 10

(2) 运行环境：JDK 1.8

(3) WEB服务器： Apache Tomcat 9.0.52

(4) 数据库：MySQL 8.0.22

(5) 云服务：七牛云

(6) 开发语言：Java8 +HTML+CSS+JavaScript

(7) 选用框架：SpringBoot 2.4.10+MyBatis 3.4.5+Redis 5.0.14+

Layui 2.5.5(前端)

(8) 开发工具：IntelliJ IDEA 2023.2.1 +apache-maven-3.6.3+Navicat Premium 16

## **本文工作和内容安排**

本文研究的工作是探讨高校请假管理系统的设计与实现，通过深入分析系统的目的、意义、功能需求以及技术实现等方面，为高校提供一套高效、便捷的请假管理解决方案。下面是各个章节的内容安排：

第一章：绪论，简要介绍项目的开发背景和目的意义，介绍高校请假管理的重要性及现状，以及介绍本系统开发环境和使用技术框架等。

第二章：相关技术框架，将本系统实现的技术框架和技术工具予以介绍。

第三章：需求分析，对参与请假管理的学生，辅导员，系主任，院长，管理员，教师等做不同的需求分析，分析需要实现的功能。

第四章： 概要设计，通过需求分析，设计出请假管理系统的整体流程以及其架构，并给出数据库相关设计。

第五章：数据库设计，通过概要设计的 结果设计出数据库概念结构设计以及具体的物理设计。

第六章：详细设计，根据概要设计以及数据库设计的结果，对各个角色的功能进行设计，完成拆分实现。

第七章：功能实现和展示，将对系统的代码实现，功能实现等进行说明，并进行系统展示。

# **相关技术介绍**

## **前端技术框架**

前端部分，本系统采用Layui框架作为基础框架来进行开发，简洁高效地开发前端界面，并使用Layuimini 后台框架模板进行整合开发，可以进行界面设计复用，并兼容手机端访问。开发符合Restful风格，前后端交互使用JSON串格式，以下是相关介绍：

### **Layui 框架**

Layui 是一个前端 UI 框架，由国内开发者贤心（昵称）主导和开发，其设计宗旨是“面向经典，拥抱未来”。Layui 以简洁、易用、快速的开发体验为核心，提供了丰富的组件和模块，帮助开发者能够快速地构建各种 Web 应用。它坚持使用原生 HTML/CSS/JS 进行开发，具有极低的学习门槛，易于上手。外观简洁而内核丰富，Layui 在轻巧的体积中提供了丰富的组件。从核心代码到 API 设计，每一个细节都经过仔细打磨，非常适合快速进行界面开发。Layui 的首个版本于2016年秋季发布，它与基于 MVVM 的UI框架不同，不追求技术上的复杂，而是倾向于简单实用。它主要面向服务端开发者，开发者不必深入繁杂的前端工具配置，只需要关注浏览器本身，从而使得开发过程更为简便。

### **Layuimini框架模版**

LayuiMini 是一个基于 Layui 构建的轻量级前端 UI 框架，专为构建后台管理系统而设计。它继承了 Layui 的模块化设计，提供了一套更简洁、更易于快速启动的界面解决方案。LayuiMini 主要特点是其简洁的界面和较小的加载体积，使其在后台管理系统中非常受欢迎，尤其是对于需要迅速搭建系统原型或中小型项目。LayuiMini 强调“轻量级”和“易用性”，集成了多种常用的界面布局和功能组件，如侧边菜单、选项卡切换、表单设计等，全部采用 Layui 风格。此外，它也提供了一些便利的功能，如自适应屏幕大小、色彩主题切换等，帮助开发者能够快速定制并优化用户界面。

由于其简便的配置和模块化的结构，LayuiMini 非常适合快速开发小到中型的后台管理系统，同时也为开发者提供了灵活性，可以根据项目需求轻松添加或修改组件。界面设计清新简约，完全响应式，优化了手机端浏览体验。仅需简单几行代码即可快速启动整个框架，操作简便。页面提供多种色彩方案供用户选择。支持多标签操作，允许同时打开多个窗口。拥有无限级菜单和对font-awesome图标库的深度整合。对于失效或错误的菜单项，系统不会允许打开，同时会通过弹窗提示优化用户体验。采用URL地址的哈希定位，便于用户识别和导航当前标签页的位置。页面刷新后能够保持当前窗口状态，并匹配左侧菜单栏对应位置。同时，系统支持font-awesome图标选择插件，增加可定制性。

### **jQuery**

Query是一个快速、小巧且功能丰富的JavaScript库。它使事情像文档遍历和操作、事件处理、动画和Ajax等变得更加简单，且易于使用API，能够在多种浏览器中工作。jQuery的核心特性可以概括为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的css选择器，并且可对HTML元素集做遍历和修改；拥有便捷的处理事件和动画效果；并且拥有丰富的插件。此外，jQuery兼容各种主流浏览器，如IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等。随着应用程序不断从桌面向浏览器转移,由于易于使用和功能完备,j Query逐渐从其它Java Script库中脱颖而出,成为很多开发人员的最佳选择。j Query是桌面特效等越来越复杂的跨浏览器的web应用程序的首选工具[2]。

## **后端开发技术**

后端部分，使用IDEA工具进行代码编写与构建，基于SpringBoot框架，MyBatis 框架等，实现后端功能的开发与测试，在与数据库交互方面使用MyBatis 作为ORM框架。

### **SpringBoot**

SpringBoot是一种基于 Java 的开源框架，旨在简化创建新的 Spring 应用程序的过程。Spring Boot 让开发者能够快速启动和开发自己的微服务项目，它内置了自动配置的特性，使得项目的配置过程极为简化。通过提供大量的起步依赖（starters），Spring Boot 自动配置您的应用程序的基础设施和依赖，这意味着开发者可以避免大量的模板代码和 XML 配置，直接专注于业务逻辑。

SpringBoot技术的特点有以下几点:可以类似点菜一样地选配、组装、生成一个独立的开发初始项目;在内部内置了如Tomcat服务器和Jetty服务器, 直接把项目打成jar包, 这应该是为Docker特意设计的;使用叫做starter POMs的配置方式使Maven配置大幅度简化;Spring Boot还为大中型项目提供了经常用到的非业务功能型的特点。比如:健康状态检测, 外部配置、指标、安全等;在项目中不再推荐使用xml的方式[3]。Spring Boot 适用于想要快速开发生产级 Spring 应用的开发者，尤其是微服务架构中的服务。通过简化配置和部署流程，它让开发和维护 Spring 应用更加轻松。

### **Maven**

Maven 是一个项目管理和自动化构建工具，主要用于Java项目的构建、依赖管理和项目信息管理。它基于项目对象模型（POM，Project Object Model）概念，使用一个中央信息文件（通常是pom.xml文件）来管理项目的构建、报告和文档的软件项目管理工具。它最突出的特点是，作为一款优秀的项目构建和管理工具，它把原本需要开发人员操作的繁杂的工作转变成了简单的命令。通过它，可以实现构建工作自动化，完成项目的清理、编译、打包、发送报告和部署等工作，简化构建过程，还可以避免造成构建重复[4]。

### **Activiti7**

Activiti7 是一个开源的工作流和业务流程管理系统（BPM），它允许组织设计、执行和监控业务流程。它是 Activiti 项目的最新版本，具有强大的流程引擎，支持业务流程建模标准 BPMN 2.0（Business Process Model and Notation）。Activiti7 专为现代微服务架构设计，提供了更好的集成和扩展能力。Activiti7所支持的规范丰富、对外通信能力优越、对Java语言主流框架Spring的支持程度高。Activiti组件BPMN以XML为载体，定义了包括流对象、连接对象、泳道、描述对象和数据5种流程的基本元素，作为流程定义时将传统业务流程抽象成电子流程的桥梁。BPMN提供了业务流程创建标准化，通过其画出的流程图为XML文件，便于数据库实现流程的流转，以及非开发人员对流程的理解。Activiti提供许多API用以操作任务流程，Activiti7则把原有的API进行了封装，使用新的ProcessRuntime和TaskRuntime进行流程与任务的控制[5]。

## **系统存储介质**

本系统的大多存储是在MySQL数据库中，包括各个角色的信息，请假信息，请假流程的信息等，数据库操作使用Navicat,界面简洁直观。在进行邮箱验证时使用到的存储介质为Redis,此外，请假流程中学生上传图片资料时，图片由于大小太大不适合储存在数据库中，使用了七牛云存储，上传到七牛云中。

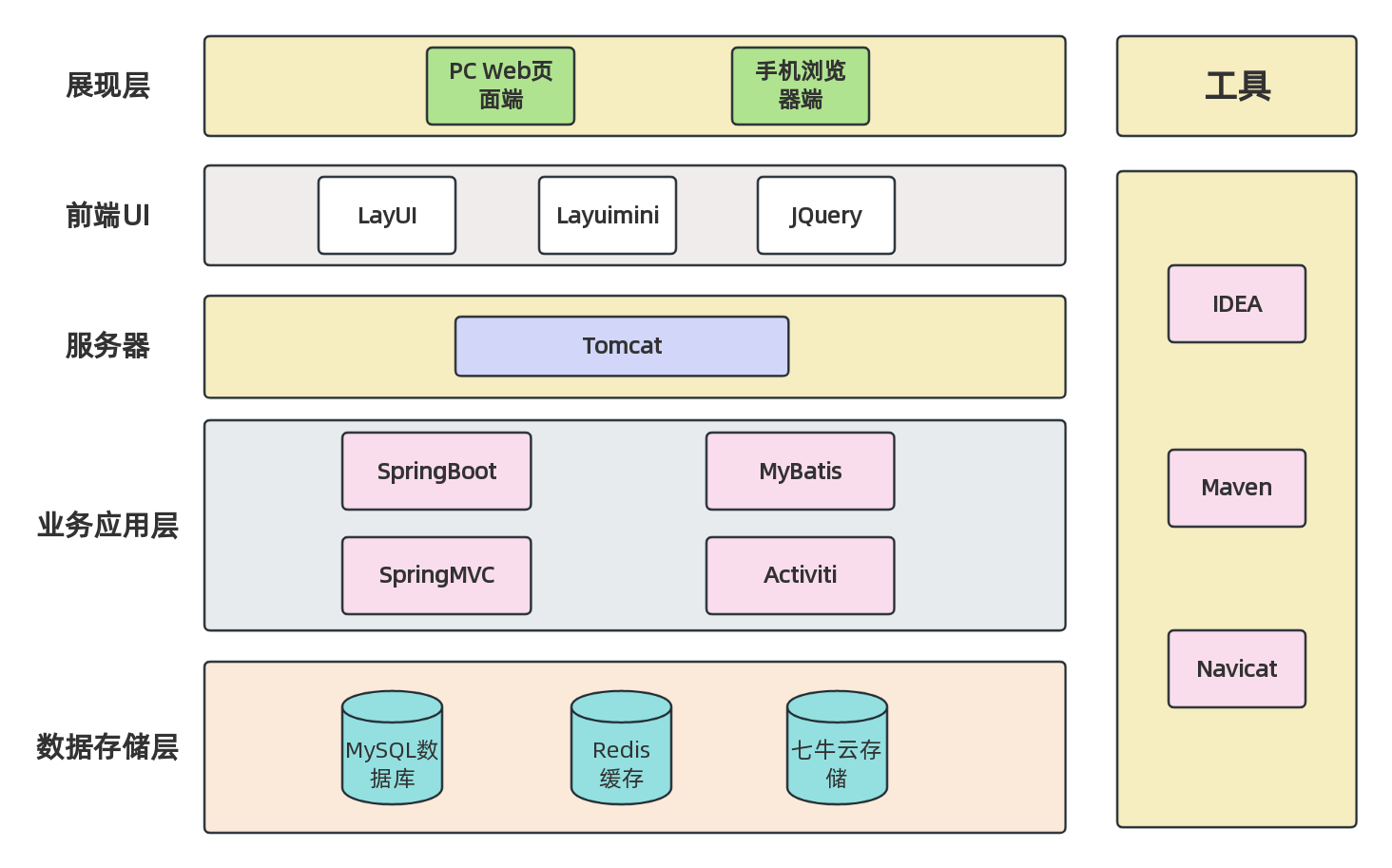
### **MySQL**

MySQL是广泛使用的数据库之一，适用于各种应用程序，特别是网页应用程序。它是许多动态网站的后台数据库组件。MySQL以其高性能、可靠性、简单的用户界面和相对较低的成本而闻名。MySQL支持大规模的数据库。它可以处理具有上千万条记录的庞大数据库。同时，它也非常适合处理小型应用程序，如个人网站或小型企业系统。

### **Redis**

Redis是一个开源的高性能键值数据库，以其出色的速度和灵活性而闻名，常被用作数据结构服务器。Redis支持多种数据结构，包括字符串、列表、集合、有序集合、散列等等。它允许用户通过键快速存储、访问和管理数据，这些数据都存储在内存中，以实现极高的读写速度。由于Redis具有高性能和低延迟的特点，它被广泛应用于缓存系统。在许多Web应用中，数据库查询是非常耗时的操作，这样会严重影响用户体验。与此同时，假如有成千上万条请求同一时间访问关系型数据库并进行增删改查操作，则势必会给数据库服务器造成严重的压力，万一处理不当就会使其崩溃，导致软件系统的瘫痪。为了避免这种情况，软件开发人员可以通过将查询结果缓存到Redis中，让所有请求先在缓存中进行查询，有结果就直接返回，没有查到，再到关系型数据库中进行查询。这样可以大大提升系统的性能，提升处理请求的效率，避免服务器因压力过大而崩溃[6]。

## **系统整体技术架构**



* + - * 1. 图 2-1 系统技术框架

# **需求分析**

本系统由于需要实现多级的请假审批以及完成与任课教师端的数据交互，需要实现的用户设置较多，按照功能的需求，应该分别有管理员，学生，辅导员，系主任，院长，任课教师共6个角色，其中，管理员与任课教师不直接参与请假审批，管理员对用户，请假时间限制等进行管理，也对请假管理有着较高的查询功能；教师作为请假管理系统的一员，主要功能应该是了解学生的请假信息以便课程结束后得出考勤成绩；辅导员与系主任的功能应当较为相似，都是对于请假信息进行审批；院长则负责超期审批。

## **功能需求分析**

### **管理员需求分析**

管理员是系统的管理者，需要进行对请假系统和用户的整体管理。

（1）组织结构管理，添加，修改和删除

学校组织结构的导入与编辑，想要实现多级的请假管理，就需要先建立起学校的组织结构，以学院，系，班级的层级结构构建一个学校的组织结构，管理员可以对组织结构进行变更和编辑。组织结构的添加要实现导入功能，由文件可直接导入系统。

1. 学生信息管理

管理员需要对全部的学生账户管理，管理员有查看所有学生账户的权限，并且可以编辑和删除学生信息，在实际应用中，往往学生信息是大量的，较少进行手动添加，需要实现由execl文件直接导入到数据库中的功能，实现学生的批量导入，个别学生特殊情况由管理员手动导入。管理员应当具有高权限，能够修改删除学生信息，需要有查询功能，能够根据姓名等对特定账户进行查询。

1. 教职工信息管理

对于教职工，包括普通教师，班主任，系主任和院长，需要对教师的信息进行增加删除功能，其功能与学生信息管理相似，也需要根据如execl表格的导入，能够简易的大范围导入教职工信息，信息展示应该以分页的形式，能够简洁进行信息展示，同时也需要实现查询对应的用户功能。

1. 请假信息管理

管理员应该能够对于所有学生的请假信息予以管理，能够查询学生的请假信息，并为了处理一些如误操作请假等情况，管理员能够删除对应请假信息，为了删除请假信息需要实现查询的功能。此外，本系统是一个多级请假审批的系统，对于超过一定期限的请假，应当给与院长审批，管理员应当能够修改这个期限时间。

### **学生用户需求分析**

学生用户是请假系统的主要使用者之一，能够完成线上请假。

1. 个人信息管理

学生需要对自己的个人信息管理，包括修改自己的账户，邮箱，密码，联系方式等。

1. 请假功能实现

学生作为请假的发起者，应当能够完成发起请假，编辑请假信息，包括请假天数，离开时间，请假原因以及能够上传图片作为请假证明材料的功能；编辑成功的请假信息应该能够有学生进行提交，一旦提交后不可撤回，提交后请假信息就会被发送到对应班级的辅导员处。

1. 请假历史信息查看功能

学生需要对自己请假的历史信息进行查看，包括查看请假当前的审批流程过程和查看审批结果，学生需要知道请假是否已经被批准，且若被驳回能够看到对应请假审批的驳回原因。

1. 邮箱修改密码功能

学生不能仅通过进入系统来进行密码的修改，若忘记登录密码，需要学生自行根据留存邮箱，通过向邮箱发送验证码来修改密码。

### **辅导员用户需求分析**

辅导员作为请假的初审者，主要需要审批请假信息。

1. 用户信息管理

辅导员需要查看自己对应班级内的学生用户信息，并可以对自己班级内的学生信息进行修改，考虑到学生信息收集可能以班级为单位，辅导员也可以批量导入学生信息，并查看对应的学生信息。

1. 请假任务办理

由学生提交的请假申请，第一时间直接发送到辅导员端，由辅导员审批办理，辅导员需要办理对应的请假申请，办理请假申请时应当通过学生提供的请假时间，请假原因和证明材料等来进行审批，同时给出审批意见。

1. 任务记录查询

已经被辅导员审批过的请假记录，需要被记录，已办任务功能记录已经办理过的请假记录，并且能够简洁地展示已经审批过的请假记录，以便辅导员事后进行记录查询。

1. 邮箱修改密码

若辅导员忘记密码，可根据在系统中存留的邮箱来进行密码修改。

### **系主任用户需求分析**

系主任功能与辅导员类似，是多级审批中的一环。

1. 系用户信息管理

系主任需要管理本系的学生信息，对系内学生的信息进行修改，并支持搜索功能，可以通过用户名，姓名，邮箱地址等多种方式查询。

1. 请假任务办理

当学生提交的请假申请通过辅导员的初步审核后，该申请将被转发至系主任审批。系主任在审批过程中需要仔细评估学生提供的请假时间、具体原因以及相应的证明材料。系主任在考虑请假理由以及辅导员审批是否合适后，做出合理的决定，并提供具体的审批意见。

1. 任务记录查询

经通过辅导员，系主任审批的请假记录，系主任可以进行查询和记录。同时能够以简洁明了的方式展示这些已处理的记录。确保了请假流程的透明度和追踪性，方便系主任回顾和管理学生的请假情况，同时保证教学和行政工作的效率。

1. 根据班级统计

系主任利用管理系统应当可以根据班级进行请假信息的统计，此功能允许系主任快速获取每个班级的请假概况，借助柱形图展示，以实现直观的对各个班级的请假有所了解。

### **院长用户需求分析**

1. 用户信息管理

院长负责管理学院内的学生信息，包括对学生信息的编辑和更新。系统提供了高效的搜索功能，允许院长通过多种关键信息如用户名、姓名或电子邮件等进行查询。

1. 请假任务管理

院长负责请假任务的最终审批，如果请假流程中请假天数小于限制天数，应无需院长审批，但是若学生请假的天数大于限制天数，即使辅导员与系主任都已经经过审批，仍然需要院长进行审批，院长在结合辅导员和系主任的审批意见，以及学生提交的请假原因后，应当在不影响教学秩序的情况下进行同意或者驳回。

1. 任务记录查询

经由审核完成的请假信息最终被记录，院长可以查看自己审批过的特殊长时间请假记录，以便后续信息核对等，保证请假流程顺利。

### **任课教师需求分析**

任课教师不参与审核阶段，可以查看对应请假数据。

1. 个人信息管理

教师可以管理自己的个人信息，包括修改联系方式，邮箱，密码等

1. 请假信息获取

教师在传统线下请假流程中，需要手动的记录学生给到教师的请假条，并由学期末统一进行核对以确定学生的考勤成绩。教师在系统中只需要简单选择自己所授课程并导出，即可得到学生对应课程请假信息。

## **系统非功能需求分析**

（1）准确性：此系统请假流程，时间，与任课教师的交互需要数据准确，以免因系统问题导致学生请假数据异常。

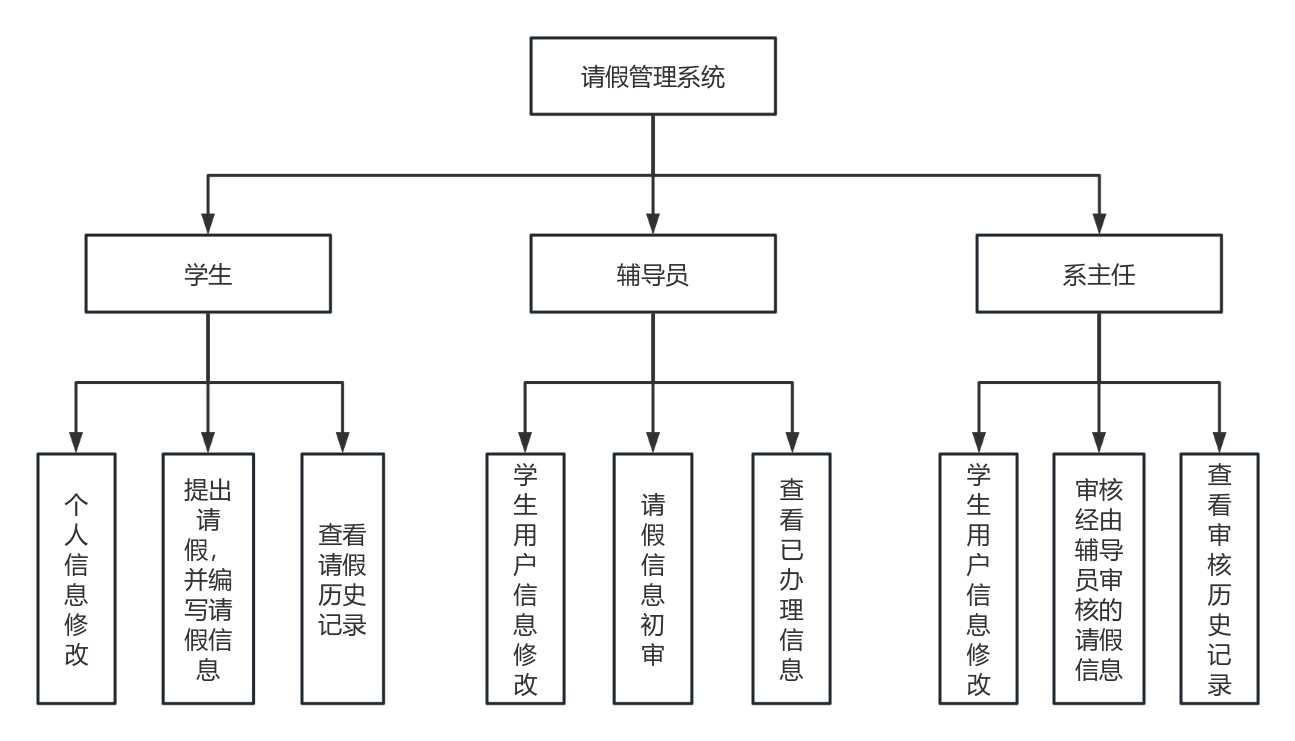
（2）可靠性：系统必须保证运行稳定可靠，避免系统崩溃，数据库异常等情况。

（3）便易性：请假系统的操作界面应当简单方便，对用户友好，以简洁舒适为宜。

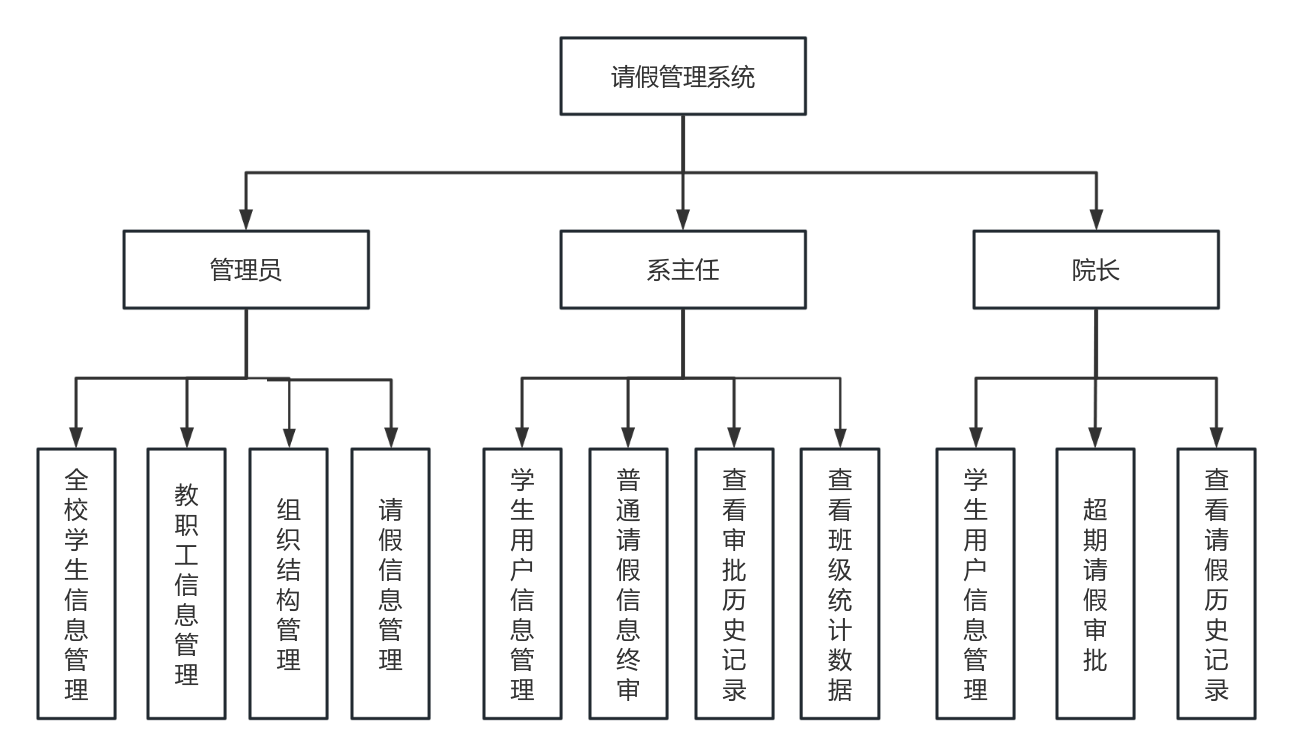
# **系统概要设计**

## **系统功能概要**

根据已经完成的需求分析，已经可以大致上设计系统的概要功能，根据分析，可以得到系统的功能层次图如下：



* + - * 1. 图 4-1 系统功能层次图（1）

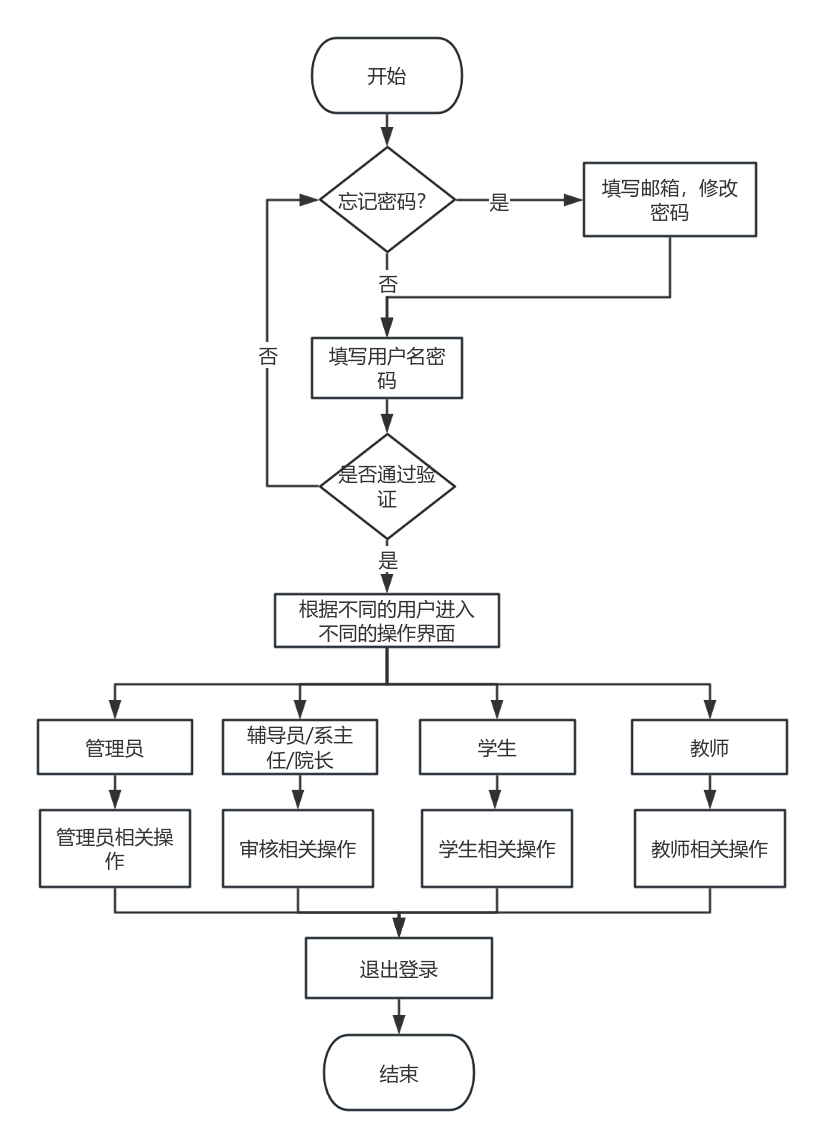


* + - * 1. 图 4-2系统功能层次图（2）

由图4-1，图4-2可知，由需求分析，概要设计可以确定，系统分为管理员，学生，辅导员，系主任，院长，教师六个角色，学生，辅导员，系主任，院长参与请假流程审批，管理员进行系统整体管理，并动态调控一些请假限制信息。教师根据学生请假信息得到所授课程的对应请假信息，各角色主要功能如图所示。

## **系统工作流程**

系统的工作流程比较简单，用户使用web浏览器或手机端浏览器，从前端输入对应的用户名和密码，如果验证成功后直接进入对应的用户角色下的页面中，如果忘记密码可以先进行密码找回然后再去返回登录页面进行系统登录，对应进入对应系统后，完成特定的功能后即可退出系统，至此系统工作流程完成。如图所示：



* + - * 1. 图 4-3高校请假系统工作流程

# **数据库设计**

## **概念结构设计**

此系统中，不但有六个角色作为实体，此外，课程，请假任务，也都是实体部分，而课程组织信息，请假信息，角色信息等都是属性，选课信息，授课信息等是关系。

### **系统实体E-R图**

在信息管理系统设计过程中,利用E-R图对数据库进行设计,反映数据库中表间的关系,避免产生多余的数据,降低数据库运行速度。数据库建立过程中,若数据库中表之间的关系模糊不清,将增加程序设计人员的工作量和成本投入,甚至无法实现信息管理系统功能设计,导致项目设计过程中失败,或者重新设计信息管理系统中的数据库、功能、程序等[7]。

1. 角色实体：用户分为管理员，学生，辅导员，系主任，院长，教师，包含用户名，密码，邮箱，联系方式，组织结构信息，角色信息等多个属性。

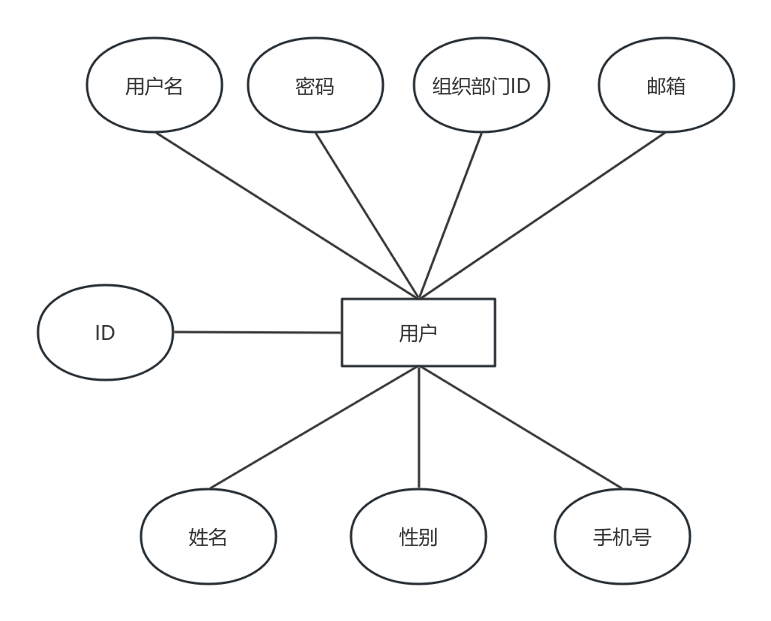


图 5-1用户实体E-R图

1. 课程实体：课程本身作为一个实体，有着课程号，课程时间信息，课程所对应教师ID等。

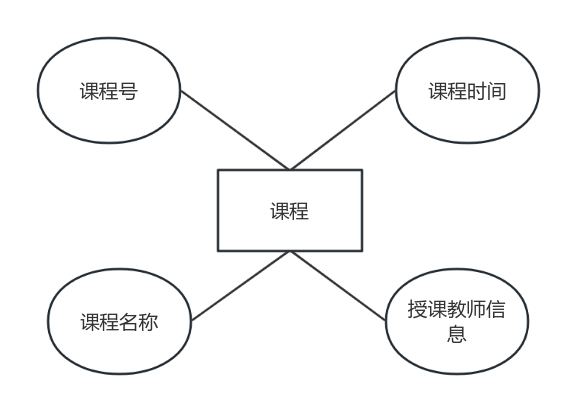


图 5-2课程实体E-R图

1. 组织部门实体：部门应作为一个实体存在，部门是一个树状的组织结构，分为学院，系，以及系下对应的班级。

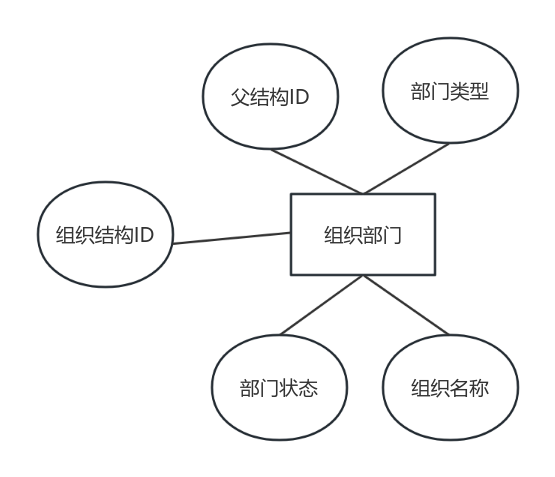


图 5-3组织部门实体E-R图

1. 请假任务实体，学生提出的请假申请，作为一个请假任务的实体存在。包括着任务流程信息，请假时间，请假原因等信息。

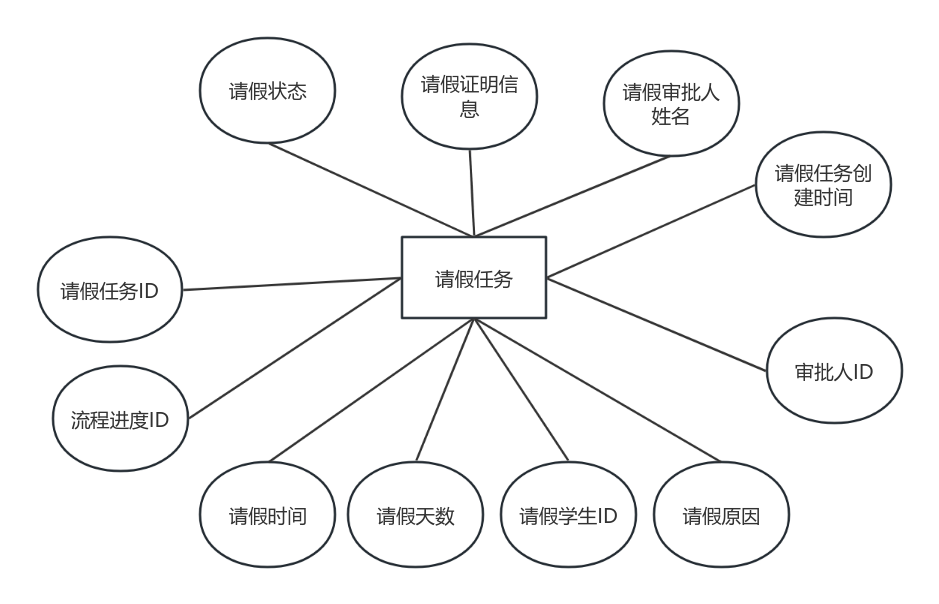
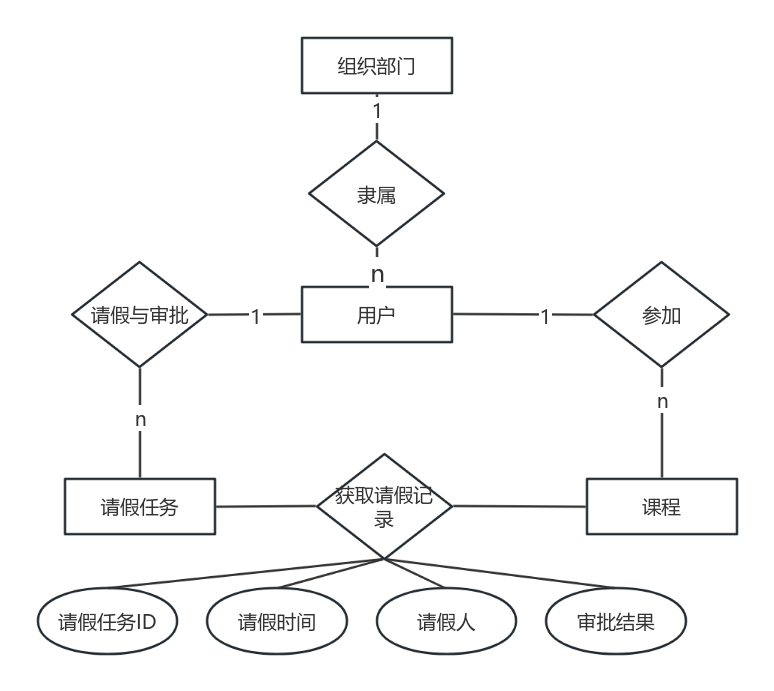


图 5-4请假任务实体E-R图

### **全局E-R图**

5.1.1所展示的E-R图，是系统各实体的E-R图，将以上各个图进行结合并，将多个局部E-R图进行综合，以形成全局E-R图是一个关键的步骤。这个过程能够揭示实体之间的关联，并在更高层次上表达数据模型的整体概念结构。由于每个实体可能具有大量的属性，在全局E-R图中展示所有属性可能会导致图形过于复杂，因此，这里在全局E-R图中省略实体属性，而专注于展示实体和它们之间的关系，此外为了简化数据库设计以及系统设计的复杂性，多个角色的用户信息都设计在了用户表中，使用不同的角色ID进行区分，展示全局E-R图时，统一使用用户实体，在详细功能实现时予以区分。



* + - * 1. 图 5-5系统全局E-R图

## **逻辑物理结构设计**

结合图5-5，以及数据库的概要设计，我们可以设计出6个数据库，分别是用户表，课程表，学生选课表，请假信息表，角色信息表，组织部门表。此外，由于系统使用Activiti7工作流实现请假功能的流程控制，不分请假任务信息由Activiti7自动生成的act数据库表储存。下面是各数据库表的设计：

1. 用户表（user表）：用于储存用户基础信息，包括姓名，邮箱，账户密码等。

表 5-1用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| Id | int | × | √ | 用户的唯一标识 |
| username | varchar(20) | × | × | 用户的登录名称，用户使用此账户登录 |
| password | varchar(20) | √ | × | 密码 |
| dept\_id | int | √ | × | 组织部门的ID，外键 |
| role\_name | varchar(20) | √ | × | 用户角色名称，区分学生，教师，系主任等 |
| realname | varchar(20) | √ | × | 用户真实姓名 |
| sex | varchar(20) | √ | × | 用户性别信息 |
| tel | varchar(20) | √ | × | 用户手机号 |
| email | varchar(50) | × | × | 用户邮箱地址 |

1. 课程表（course表）：用于储存课程实体信息，课程相关内容。

表 5-2课程表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| id | int | × | √ | 课程id，课程唯一标识符 |
| name | varchar(20) | √ | × | 课程名称 |
| course\_time | datetime | √ | × | 课程开始时间 |
| teacher\_id | int | × | × | 本课程教师id |

1. 学生选课表（stu\_course表）：记录学生选课信息。
   * + - 1. 表 5-3学生选课表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| id | int | × | √ | 选课信息唯一标识 |
| stu\_id | int | √ | × | 选课学生信息id |
| course\_id | int | √ | × | 选课课程id |

1. 角色信息表（role\_info表）：记录系统角色信息。
   * + - 1. 表 5-4角色信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| id | int | × | √ | 角色信息唯一标识符 |
| name | varchar(20) | √ | × | 角色信息名称 |
| remarks | varchar(20) | √ | × | 角色信息备注 |

1. 请假信息表（leave\_info表）：记录请假信息，包括请假时间，请假学生，请假原因等。
   * + - 1. 表 5-5 请假信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| id | int | × | √ | 请假信息唯一标识 |
| user\_id | int | × | × | 请假用户id |
| leave\_date | date | × | × | 请假开始时间 |
| leavedays | int | × | × | 请假天数 |
| reason | varchar(255) | √ | × | 请假原因 |
| state | int | √ | × | 请假审批状态 |
| processInstanceId | varchar(50) | × | × | 流程id |
| img\_name | varchar(255) | √ | × | 请假图片证明名称（以url形式存入） |

1. 组织信息表（dept\_info表）：记录组织结构信息，以及组织部门对应信息。
   * + - 1. 表 5-6组织信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 可空 | 主键 | 备注说明 |
| id | int | × | √ | 组织部门唯一标识符 |
| name | varchar(20) | × | × | 组织部门名称 |
| parent\_id | int | × | × | 上级组织id |
| type | int | × | × | 部门类型 |
| status | int | √ | × | 部门状态 |

通过以上设计，我们在MySQL数据库中建立以上表格，可以在数据库中导出如下模型图：

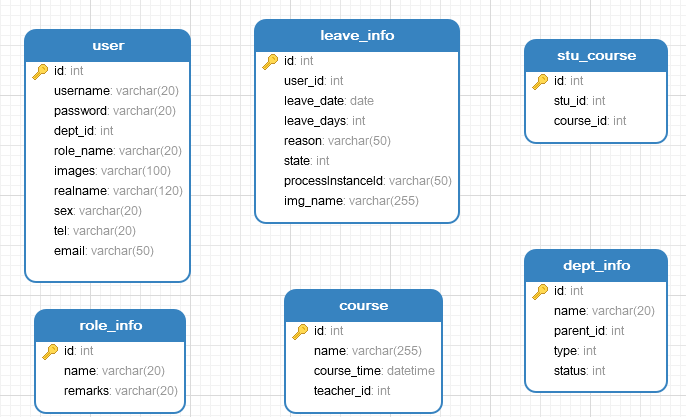


图 5-6数据库表模型图

# **系统详细设计**

## **系统整体架构**

系统架构主要使用SpringBoot 和layui采用B/S架构来实现，其中客户端通过网络浏览器访问服务器上的应用程序。此架构能够简化客户端的维护和部署，因为主要的应用逻辑和数据处理都在服务器端进行。这种模型以Web浏览器作为用户界面，与后端服务器进行交互，以完成各种业务处理和数据操作。在B/S架构中，客户端通常只需要一个浏览器即可，所有的业务处理逻辑都在服务器端完成，客户端与服务器之间通过HTTP或HTTPS协议进行通信，数据格式通常为HTML,CSS和JavaScript等Web标准技术，这使得B/S架构的应用系统具有良好的跨平台和易扩展性[8]。

1. 浏览器端

浏览器端主要是基于Web的界面，用户通过任何标准的网络浏览器如Chrome、Firefox或Edge来访问系统。用户界面设计可以适配不同的设备和屏幕尺寸，提供直观的操作界面来提交请假申请、查看请假状态及历史记录等功能。比如用户交互，用户通过表单输入请假信息，包括请假时间、原因及证明材料等。数据提交，通过浏览器提交数据至服务器，如Ajax请求，以实现无刷新的数据交互。界面由Layui编写。

1. 服务器端

服务器端负责处理业务逻辑、数据存储和安全验证。系统后端可能采用如Java,语言开发，并通过RESTful API与前端进行通信。服务器处理请假申请的审批流程，如自动审批、转发给上级审批等。服务器还管理一个数据库，用于存储用户信息、请假记录和审批状态等数据。系统通过RESTful API实现前后端的数据交互，API提供标准的HTTP方法来接收前端请求和发送响应。这些API处理前端发来的各种请求，如获取用户信息、提交请假申请、审批流程控制等，并返回JSON格式的数据。

## **系统模块设计**

### **登录模块**

登录模块是请假管理系统的入口，主要负责验证用户的身份和授权用户访问系统的权限。该模块有以下功能：

1. 用户验证：用户通过输入用户名和密码进行身份验证。系统将输入的信息与数据库中的记录进行匹配，以确认用户身份。
2. 会话管理：一旦用户登录成功，系统会为用户创建一个会话，并在用户会话期间维持其登录状态。
3. 用户界面：提供一个用户友好的界面，包括错误消息提示（如“用户名或密码错误”）、密码可见性切换功能以及忘记密码链接。

### **找回密码模块**

找回密码模块允许用户在忘记密码时重置密码，增强系统的可用性。该模块需要在后端实现通过邮箱发送验证码的功能，并由后端验证验证码，并实现验证身份后，用户可以设置新的密码。为了确保验证码的安全性，设置验证码的有效期限为三分钟。

### **导入信息模块**

导入信息模块用于向系统批量导入用户或组织结构信息，通常用于系统初次设置或大规模更新。实现以下功能：

1. 文件上传

用户可以上传格式化的文件（Excel文件），包含必要的数据。

1. 数据验证

系统自动验证导入数据的格式和完整性，确保数据的准确性和可用性。

1. 后台处理

大量数据的导入通常在后台进行处理，不影响前端用户操作。

### **请假申请模块**

请假申请模块是学生请求休假的功能区域，实现以下功能：

1. 申请表单

学生填写请假类型、开始日期、请假时长、请假原因等。

1. 申请提交

申请提交后，自动发送给对应的辅导员进行审批，后续参与多级审批。

1. 状态跟踪

学生可以查看其请假申请的当前状态（如审核中、已批准、已拒绝）。

1. 历史记录

系统保存用户的请假历史，供未来查询和记录。

### **请假审批模块**

请假审批模块使得辅导员系主任和院长能够查看、审核和处理学生的请假请求。实现以下功能：

1. 审批界面

展示待处理的请假申请，并提供必要的详情，如申请人、请假时长、理由等

1. 操作功能

参与审核的老师可以批准或拒绝请假申请，并附带审批意见。

1. 统计数据

提供一定的统计数据供系主任等了解请假趋势和分布。

# **系统功能实现与展示**

展示系统的功能实现，并相应使用代码，图片，文字等解释介绍具体的实现细节，功能展示与实现主要分为登录功能，管理员功能，学生功能，辅导员功能，系主任功能，院长功能，教师功能几个模块。登录功能由全体用户共享，进行统一说明。

## **登录模块**

### **用户登录模块**

用户登录界面如下图所示：



* + - * 1. 图 7-1用户登录界面

用户登录需要三个对应的数据，用户名，密码以及对应的用户类型，用户类型的选择由一个下拉框来进行选择，这样能避免用户的错误输入，当用户输入用户名，密码以及对应的用户类型后，点击立即登录，前端系统会对是否缺失数据进行检查，如果出现如角色类型为空的情况，前端会弹出“角色不能为空”提升，经过检验之后将输入的数据以json格式发给后端接口处理。

后端对应Controller接收对应的用户名，密码和用户类型，接收数据后，先对输入的明文密码进行MD5加密，之后controller 代码调用service层userservice来进行密码核对的业务流程操作，userservice调用dao层userMapper相关代码来实现查找数据库对应用户信息，从而返回是否正确登录的信息，若正确登录即将对应的用户信息存储到一个session中并返回正确信息,将code设置为200，否则返回错误信息，将code设置为400。

数据传回前端后，若返回值的code为200，则将用户数据储存在localstorage中，若code不为200则提示“登录失败”，以下是登录模块后端controller层中的代码实现：



* + - * 1. 图 7-2登录功能实现代码

此外，未登录的用户应该不能访问系统，系统的请假功能与查询功能是只对系统用户开放的功能，为了实现组织未登录用户访问系统的功能，系统设置了一个登录拦截器，除了对应的文件与修改密码模块不进行拦截以外，将其余模块一概拦截，只有当session中的用户名不为空时，拦截器才予以放行。这样即能够实现对非登录用户的全部拦截。以下是拦截功能实现核心代码：



* + - * 1. 图 7-3拦截器配置文件



* + - * 1. 图 7-4拦截功能实现代码

### **修改密码模块**

由图7-1可见，在登录界面的右下角，由一个蓝色的“忘记密码”按钮，在系统用户忘记登录密码时，能够使用验证密码功能，自行根据在系统中留存的email邮箱收取验证码，经由验证码验证修改密码。



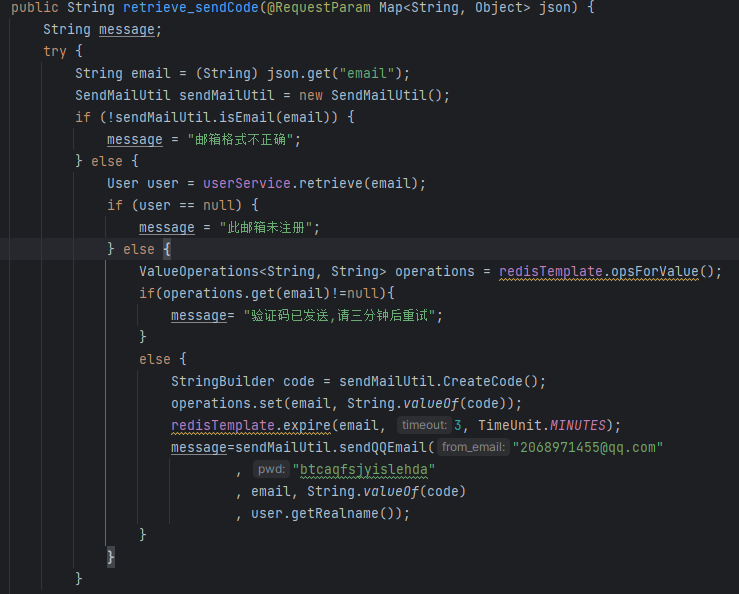
* + - * 1. 图 7-5找回密码界面

找回密码的流程为用户在登录界面选择“忘记密码”链接，系统提供一个表单（如图7-5所示）让用户输入注册账户时使用的电子邮件地址。用户提交信息后，系统后端验证提供的电子邮件地址是否与现有用户账户匹配。如果信息匹配，系统将发送一条包含验证码的邮件到用户的注册电子邮件上。

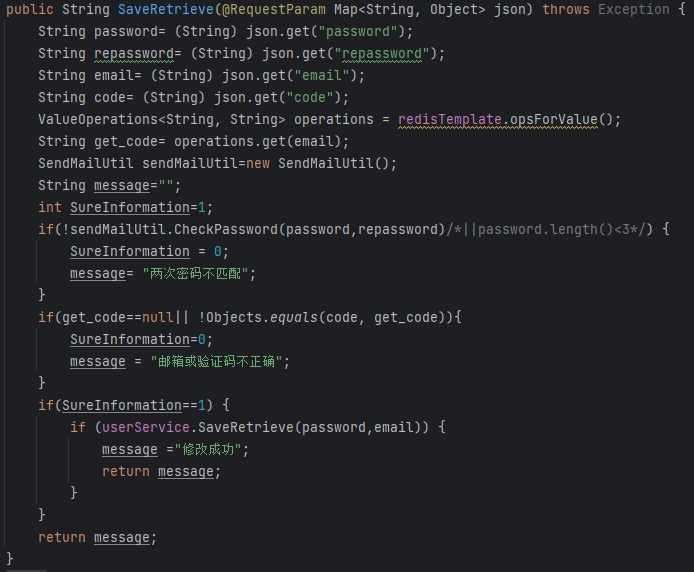
得到验证码信息后，在表单输入验证码处输入验证码，验证码信息存入Redis并设置三分钟的时间限制，同时将前端获取验证码的按钮置为不可用，

并设置倒计时，三分钟结束后可重新获取验证码。保证了验证码的安全性，提交修改密码申请后进行找回密码功能。

找回密码验证方法首先从请求中获取用户输入的新密码、确认密码、邮箱和验证码，并使用 Redis 验证邮箱关联的验证码是否正确。接下来，通过 SendMailUtil 校验两次输入的密码是否一致。如果验证码正确且密码一致，调用 userService 的 SaveRetrieve 方法更新密码并返回成功消息；如果有错误，如密码不匹配或验证码不正确，则返回相应的错误消息。修改密码成功后，自动跳转到登录页面，不需要手动点击返回登录。



* + - * 1. 图 7-6验证码发送代码实现



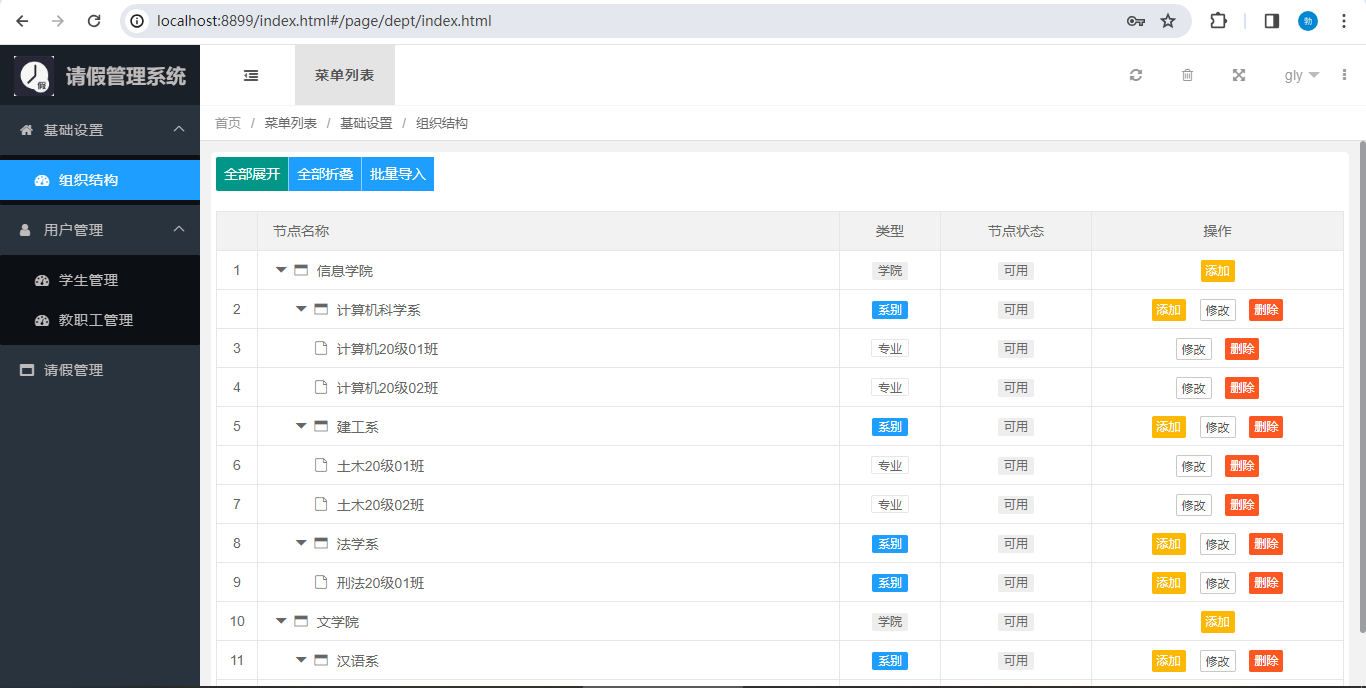
* + - * 1. 图 7-7验证修改密码代码实现



* + - * 1. 图 7-8验证码界面

## **管理员模块**

### **组织结构管理模块**



* + - * 1. 图 7-9组织结构界面

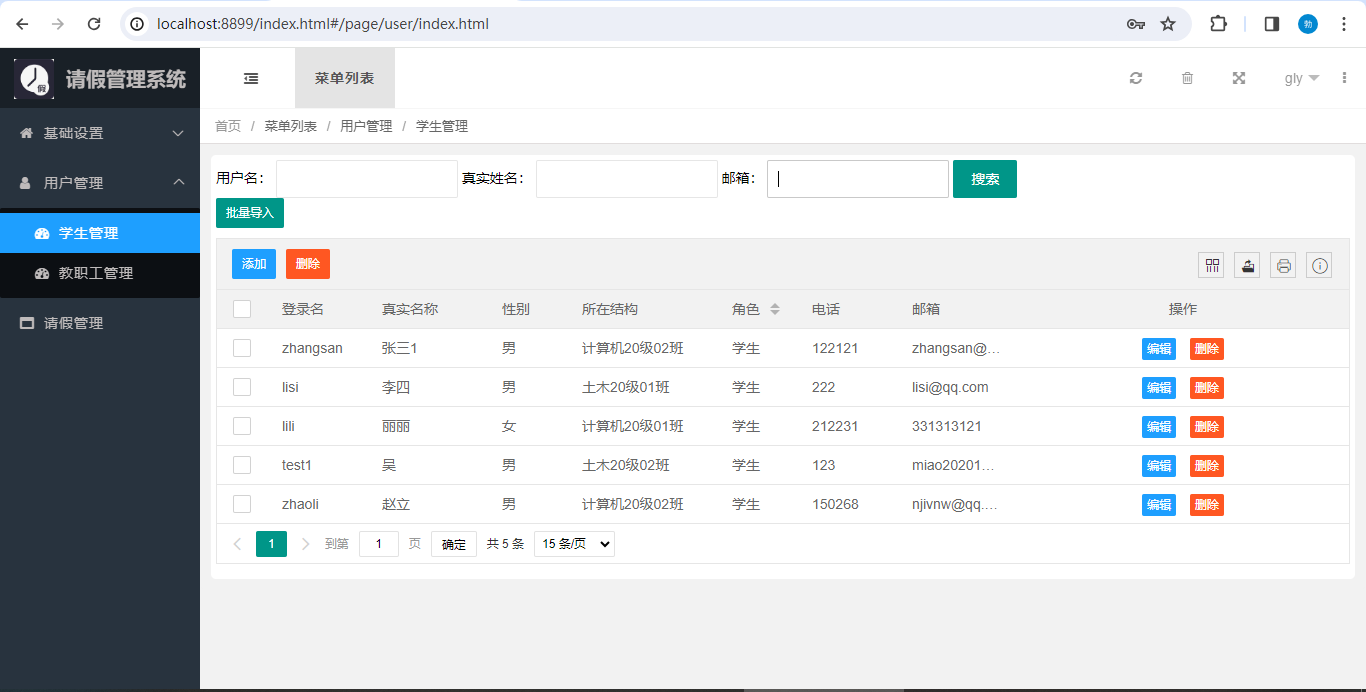
管理员作为系统的管理者，首先是对学院的组织结构的建立，由于系统需要实现多级审批的功能，管理员需要管理学院，学系，班级这样的组织结构，组织结构管理功能允许管理员在系统中维护组织的层级结构，一般组织结构信息不会大幅变更，可以进行由文件批量导入的功能，也可以进行单独修改，点击添加按钮后会在表格的树形结构下直接添加对应位置的班级或学系等，点击删除功能会弹出判断框最后询问是否进行删除操作，也可以通过手动设置节点是否可用在不删除数据的情况下禁止节点的相关功能。组织结构相关的Controller层以及Service层，主要实现树形结构的删除，修改，查询结构信息的功能等。组织结构的树形展示使用了前端的treetable组件实现。以树形结构节点的删除为例，如图7-10中代码所示，如果删除模块的deleteById，接受一个部门ID作为参数。首先，它通过调用 deptDao.selectByPrimaryKey(id) 方法获取到对应ID的部门信息。接着根据部门的类型进行不同的处理，如果部门是一个院系，则首先查询该院系下所有的子部门（即查询该院系作为父ID的所有部门）。接着，遍历这些子部门，并逐一调用 deptDao.deleteByPrimaryKey 方法删除每一个子部门。在删除完所有子部门后，再删除该院系本身，如果部门是班级（即 type 不为 2），则直接删除该部门本身，最后根据num的数值判断是否删除成功。



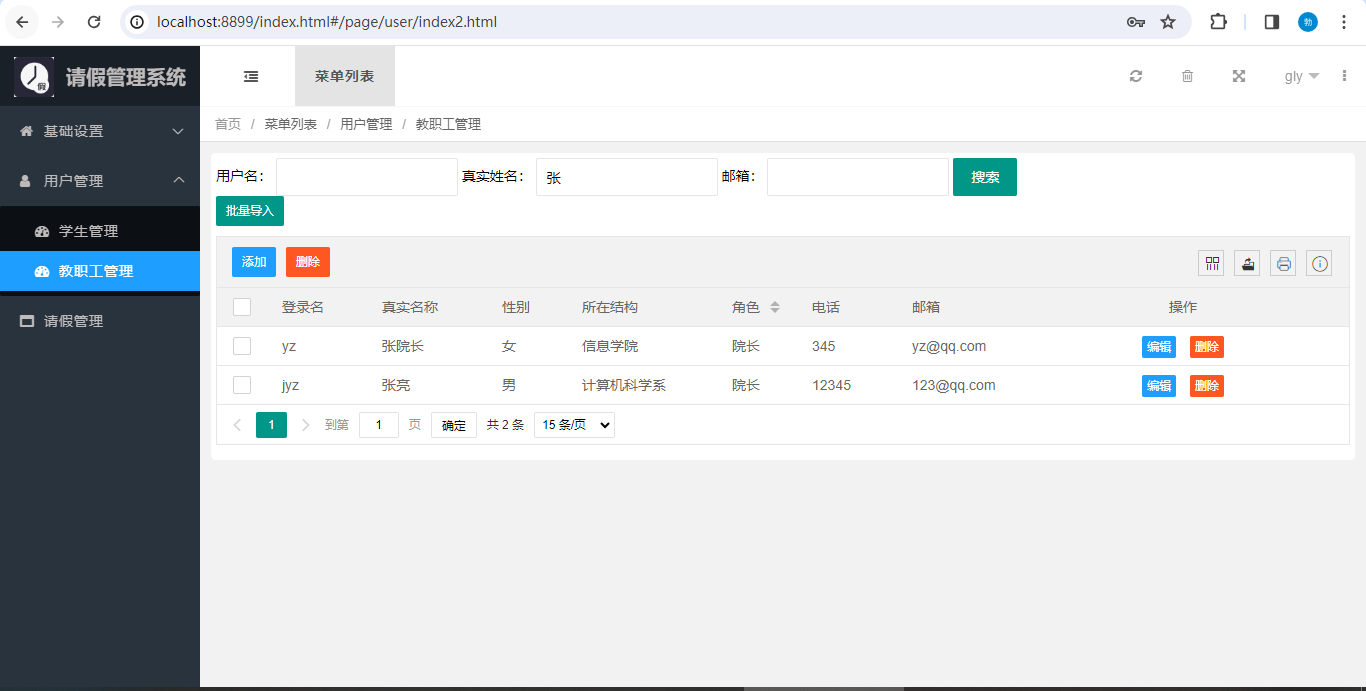
* + - * 1. 图 7-10删除组织节点功能

### **用户管理模块**

用户管理模块主要对用户的信息进行管理。



* + - * 1. 图 7-11学生信息管理界面



* + - * 1. 图 7-12教职工信息管理页面

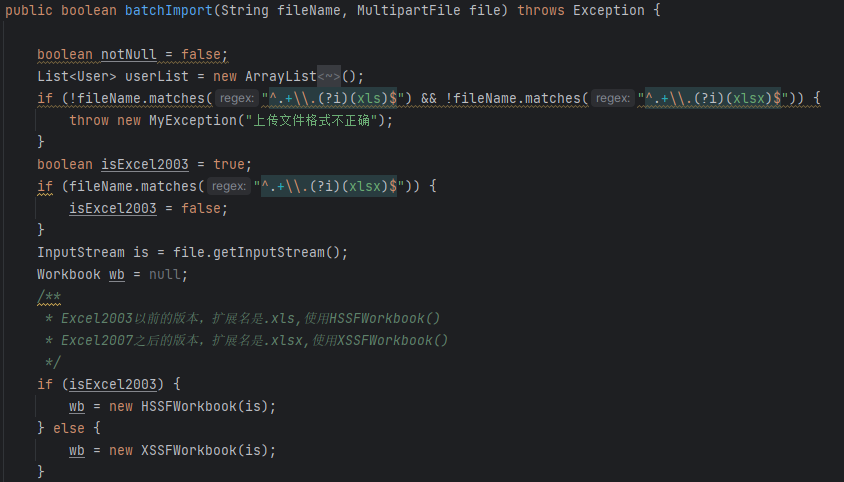
如图所示，在学生管理页面，展示了一系列表格，每个表格代表一个学生的详细信息。这些表格包括了学生的姓名、真实姓名、性别、所在结构、角色、电话、邮箱等关键信息，学生管理功能还提供了编辑和删除学生的操作。通过点击对应学生信息后的“编辑”按钮，用户可以对学生信息进行修改或更新。如果需要删除某个学生的信息，可以点击“删除”按钮进行操作。这些功能使得用户能够灵活地管理学生信息。由于学生信息插入往往是大量数据，学生管理模块还引入了学生数据导入的功能，由execl表格数据直接导入到系统中，便于管理员操作。此外，为了方便查询用户信息，系统设置了由用户名，真实姓名，邮箱来查询特定用户的方法，如图7-12所示通过模糊查询，可以简单高效的查询用户信息，模糊查询的底层实现细节见图。方便管理员管理特定人员信息。教职工管理与学生信息管理基本类似，只是在批量导入数据信息的接口以及展示的信息不同，教职工以教职工的格式来导入信息，下面以部分学生信息管理的代码为例简单介绍底层实现。



* + - * 1. 图 7-13导入学生数据Controller层

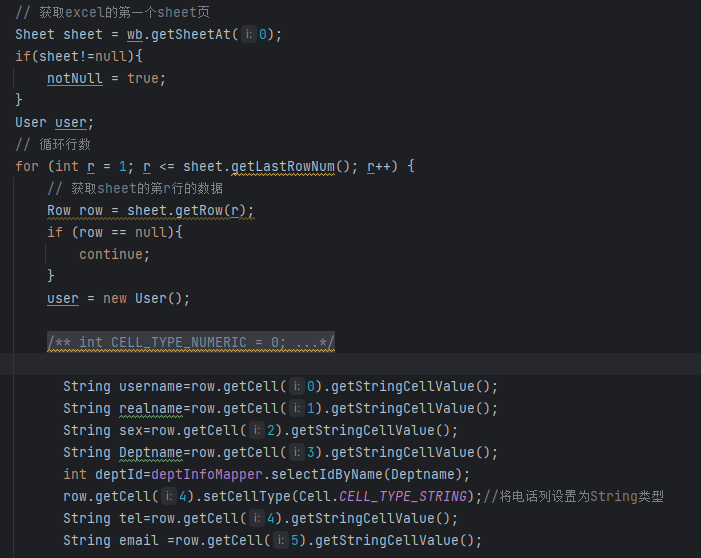
如图7-13，学生数据导入的Controller层比较简单，只是使用MultipartFile类型的参数从前端接收对应的文件信息，然后使用batchImport方法处理，

在batchImport 方法中，如图7-14所示，图7-14展示了batchImport方法的前半部分，首先通过检查文件扩展名来确定文件是否为Excel文件，并区分处理Excel 2003（.xls）和Excel 2007及以上版本（.xlsx）。根据文件类型，它使用不同的类HSSFWorkbook 或 XSSFWorkbook来处理对应的Excel文件。文件第一行是标题，代码从文件的第二行开始读取，创建用户实例，并将其添加到列表中。如果导入过程中没有读取到任何数据，它将抛出一个异常。然后关闭相关资源并返回用户列表。



* + - * 1. 图 7-14批量上传用户（1）

文件判断完成后，batchImport方法进行execl文件（.xls或.xlsx）进行处理，从第一个工作表中，从第二行开始逐行读取学生的用户名、真实姓名、性别、部门名称、电话和电子邮件，将电话列设置为字符串类型以避免格式错误。部门名称通过一个查询转换为部门ID，同时为每个学生设置默认密码和角色名。最后，所有读取的用户信息被存储在一个列表中，准备进一步的处理或存入数据库。最后由batchImport方法调用对应的mapper层方法将数据存入数据库。完成用户的批量导入。



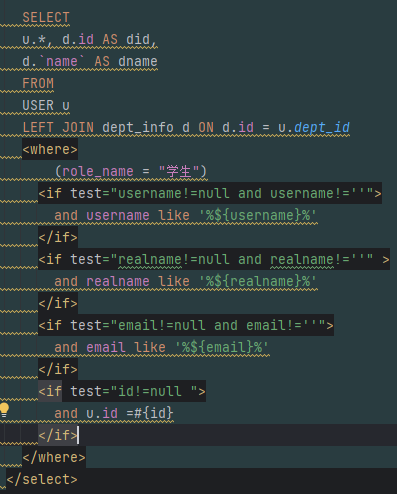
* + - * 1. 图 7-15批量上传用户（2）

用户信息模糊查询是通过前端传递的用户名，真实姓名，和邮箱地址结合来进行模糊查询，如图7-16，是前端实现模糊查询用户信息的核心代码，代码中，定义了一个active对象，其中包含一个reload方法。此方法根据用户输入的用户名、真实姓名和电子邮件地址重新加载表格数据。通过调用table.reload函数，并设置当前页为第一页，并将输入框中的值作为查询参数传递给后端。此外，还通过事件监听器绑定点击事件，当点击具有layui-btn类的按钮时，会检查active对象中是否有与按钮data-type属性相对应的方法，如果有，则调用该方法。在这里按钮为图7-12中的“搜索”按钮。



* + - * 1. 图 7-16前端用户模糊查询加载代码

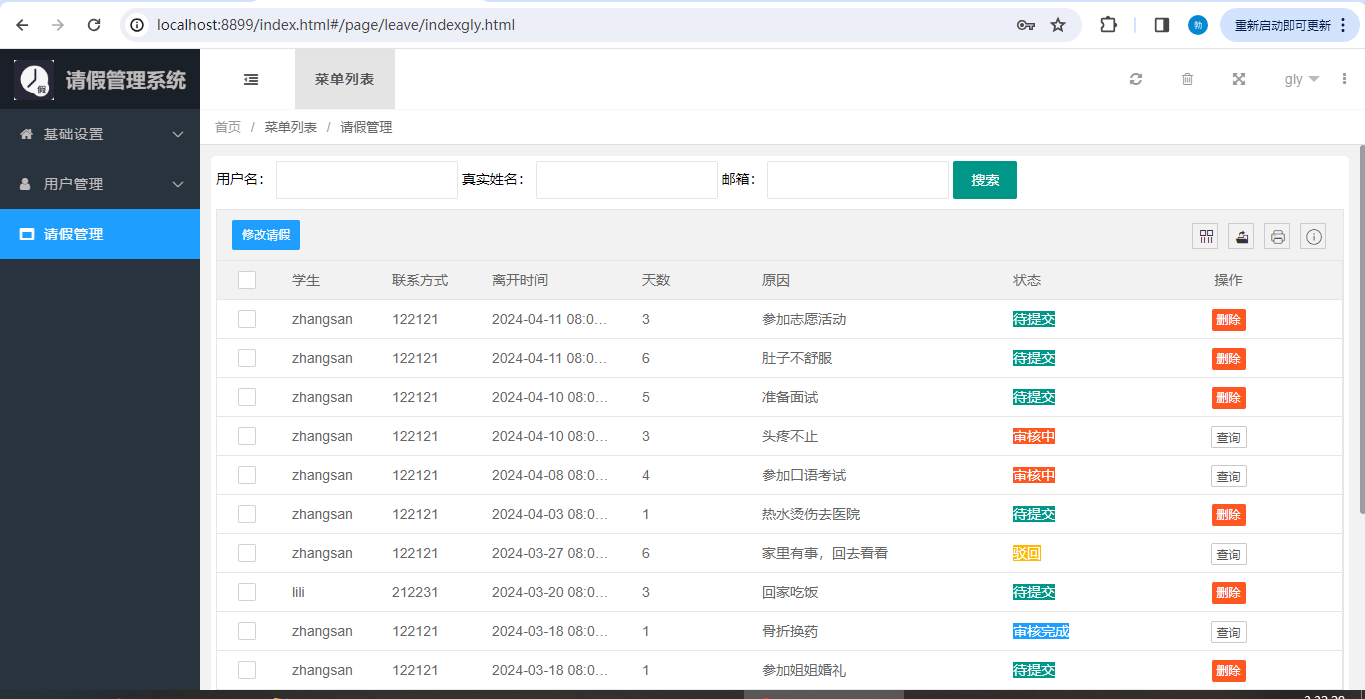
前端重新加载页面后，会调用user/queryUserstu方法进行重新查询用户数据，以实现查询的功能。后端接收到前端的username，realname,email后最终调用如图7-17的数据库查询方法，该方法使用了动态SQL元素<where>和<if>来构造条件语句。使得查询很灵活，只有在相应的参数非空且不为空字符串时，相应的条件才会加入到查询语句中，即根据提供的用户名、真实姓名、电子邮件地址进行筛选。



* + - * 1. 图 7-17用户信息模糊查询mapper实现

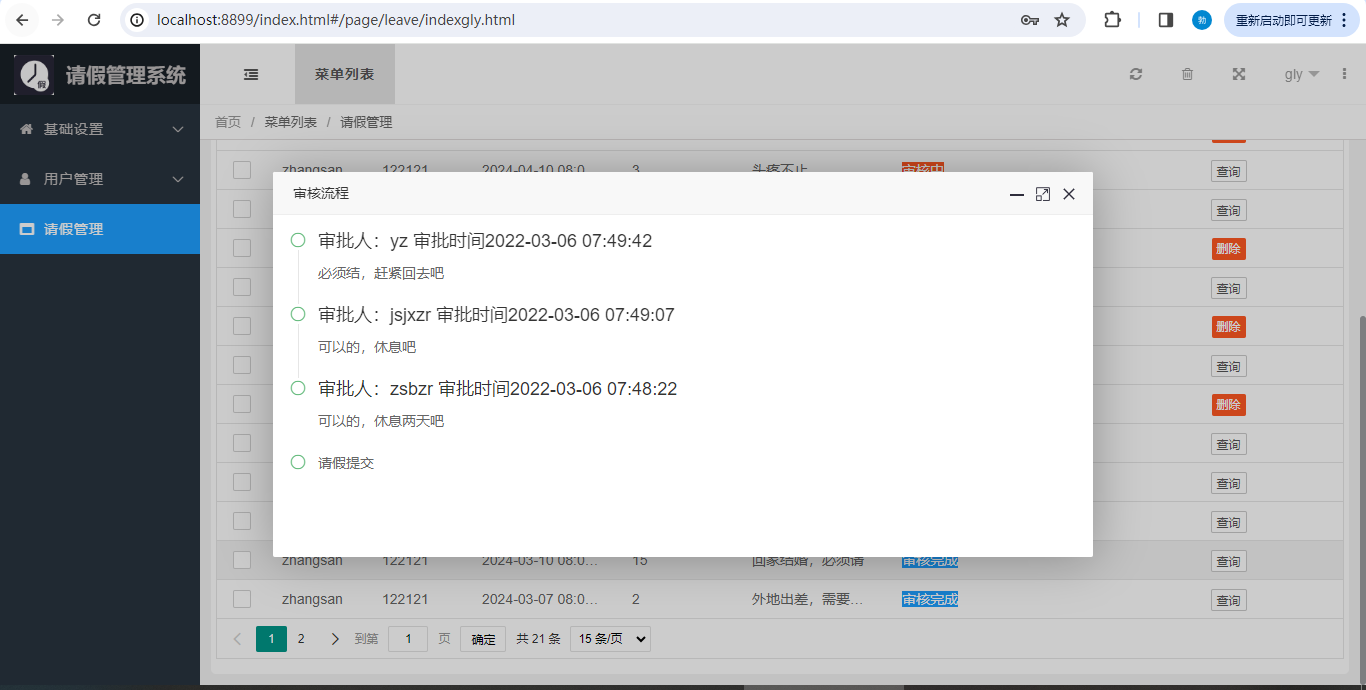
### **请假管理模块**

管理员对请假信息以及请假限制条件进行管理和修改。其中请假数据也可以根据学生对应的用户名，姓名，邮箱来进行查询，并且设置了修改请假限制的按钮，可以对多级审批中的超期请假判断天数进行修改，完成对请假规则的修改。



* + - * 1. 图 7-18管理员请假管理界面

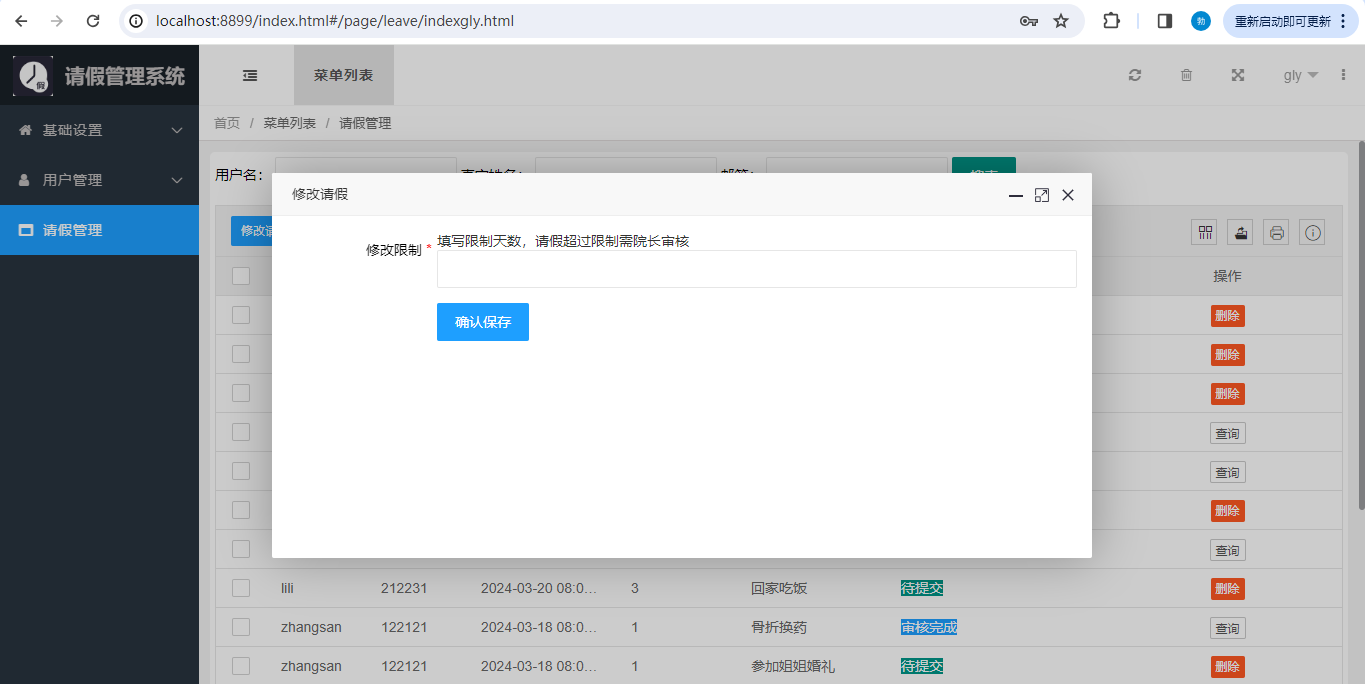
管理员可以对请假数据进行查看，点击查询按钮时，会得到请假审批流程信息，此处的后端数据处理涉及到整个请假流程的具体实现，包括Activiti7的使用，在后面的请假模块中予以细致说明。这里给出具体展示。如下图：



* + - * 1. 图 7-19审批流程展示

通过查询按钮可以看到具体多级审批的各级审批细节，帮助管理员了解请假具体信息。

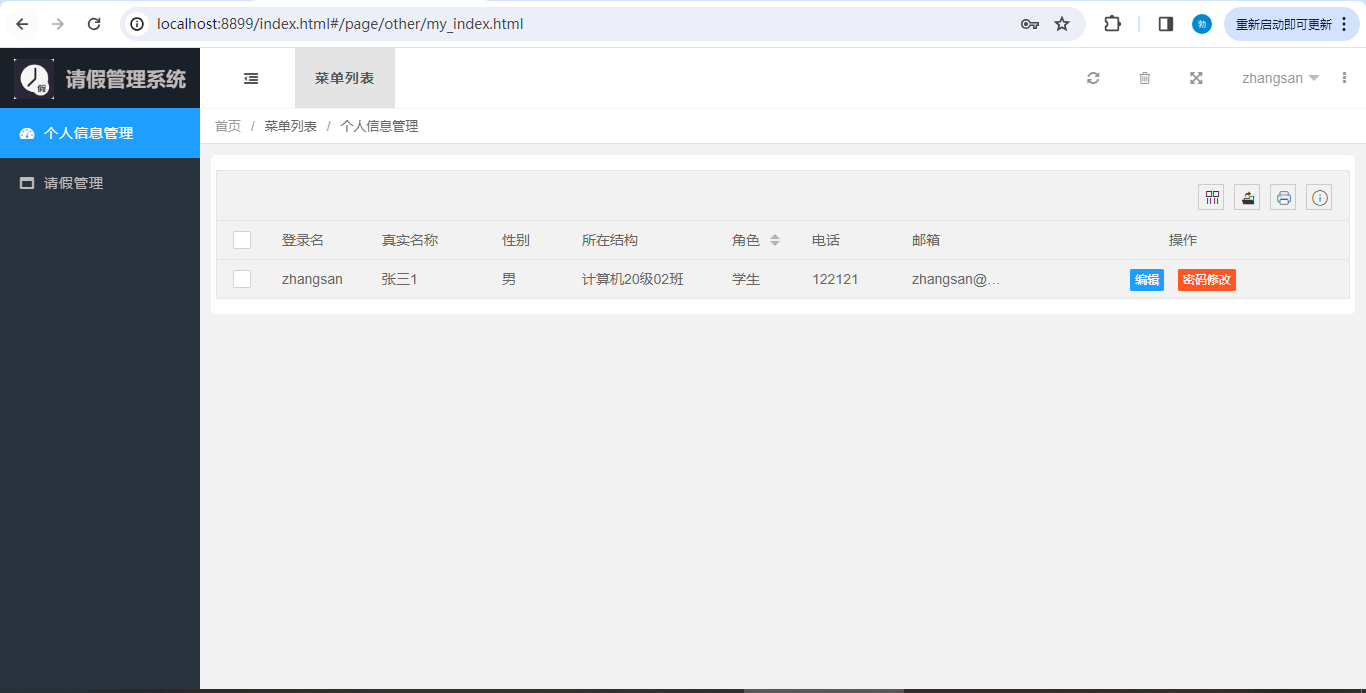
在修改功能模块中，如图所示，管理员拥有修改请假限制天数的权限，如默认值为超过5天的请假需要再请假流程中发送给院长审批，修改模块的实现是通过前段传入数据来动态修改BPMN中的判断条件来进行的。



* + - * 1. 图 7-20修改请假限制天数界面

## **学生模块**

学生模块的功能主要是进行请假申请的提交，也是学生最核心的功能，学生登录页面后，可以查看到自己的个人信息，通过编辑按钮也可以进行个人信息的编辑操作，如图7-22所示，编辑操作会生成一个信息修改的表单，将学生原本的个人信息进行渲染，并展示在表单上，学生可以此为基础修改自己的信息。学生填写好自己的信息后，只需点击“确认修改”按钮系统会将学生信息修改保存。在系统内，也提供了对应的修改密码的按钮。



* + - * 1. 图 7-21学生模块界面

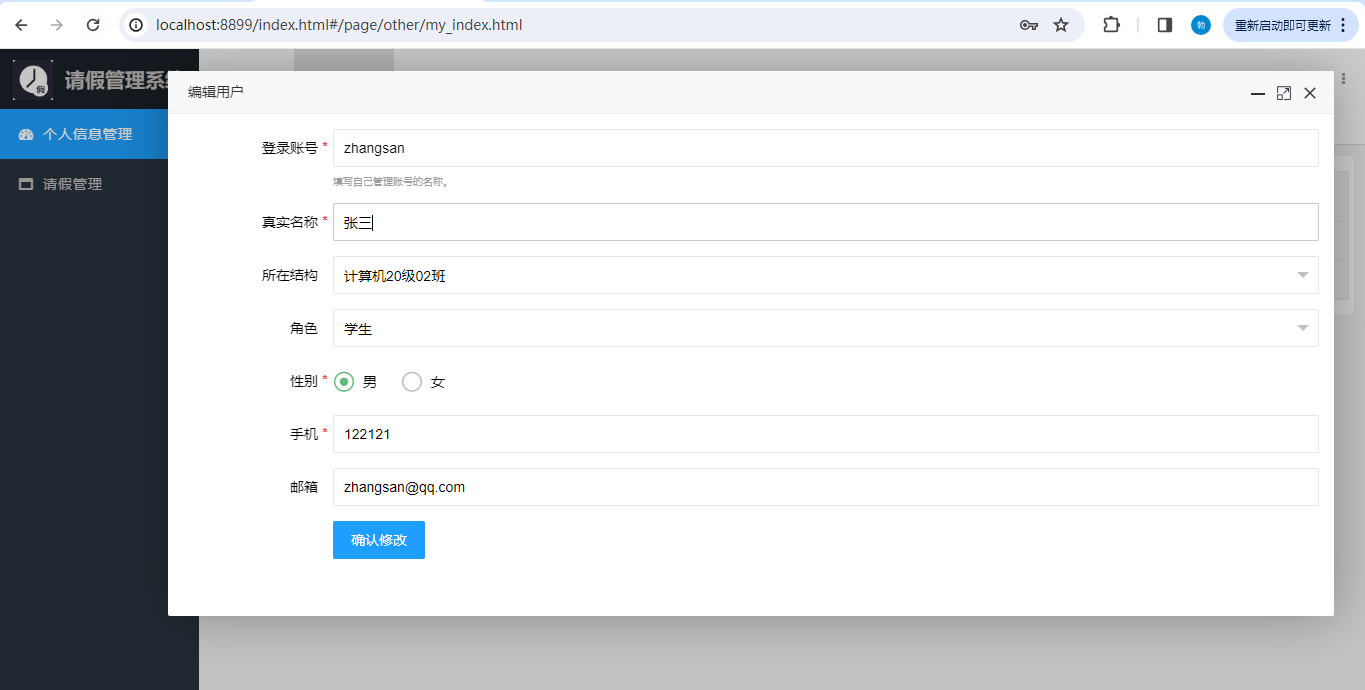


图 7-22学生修改个人信息

学生的请假管理界面可以进行历史请假的展示，在如图7-23中，可以看到，学生自己可以查询自己的请假记录，请假状态共分为4个状态，待提交，审核中，审核完成，驳回，学生可以点击“我要请假”按钮，生成一个请假信息填写的表单，如图7-24所示，学生填写离开天数，请假开始时间，以及请假原因，为了使系统更加具有可用性，可以上传证明图片，图中所示上传了医院病历单，后续审批过程中，流程进行到哪一环，哪一环的审批老师就能够看到对应的请假原因和图片资料，以衡量对应学生请假的请假资质，便于审批人员更快更合理地进行审批工作。

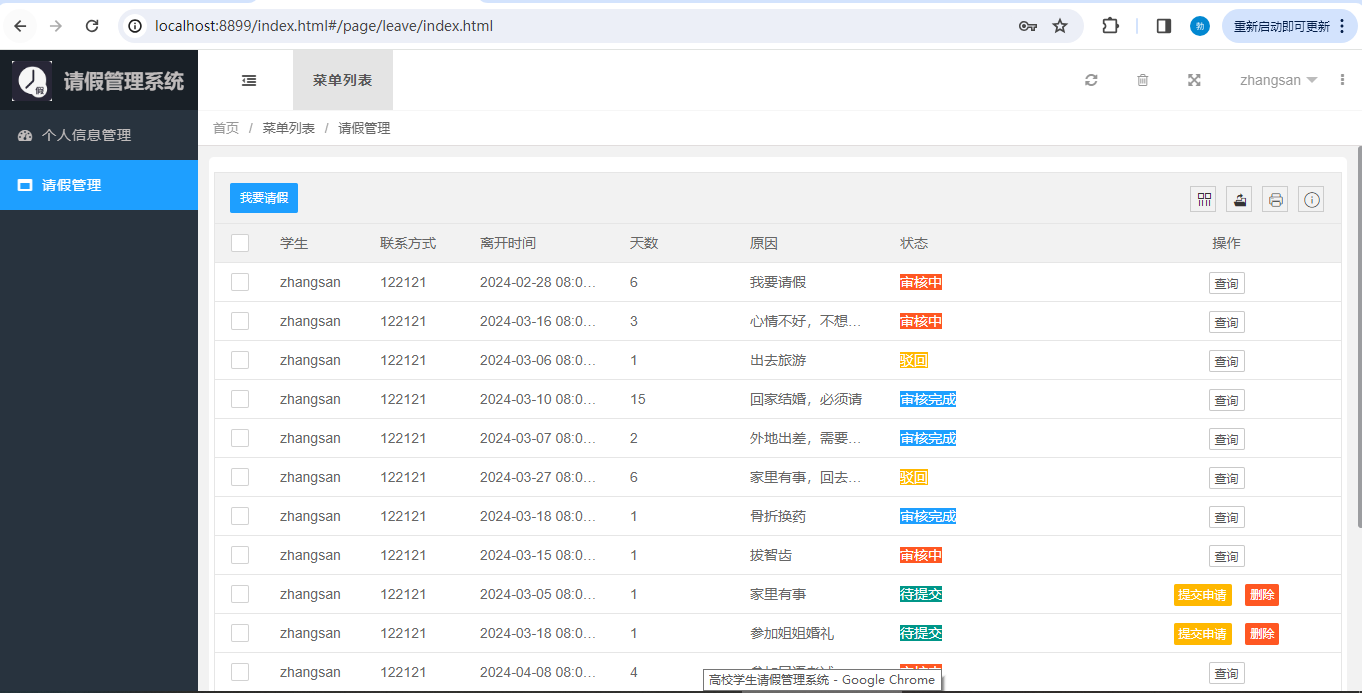
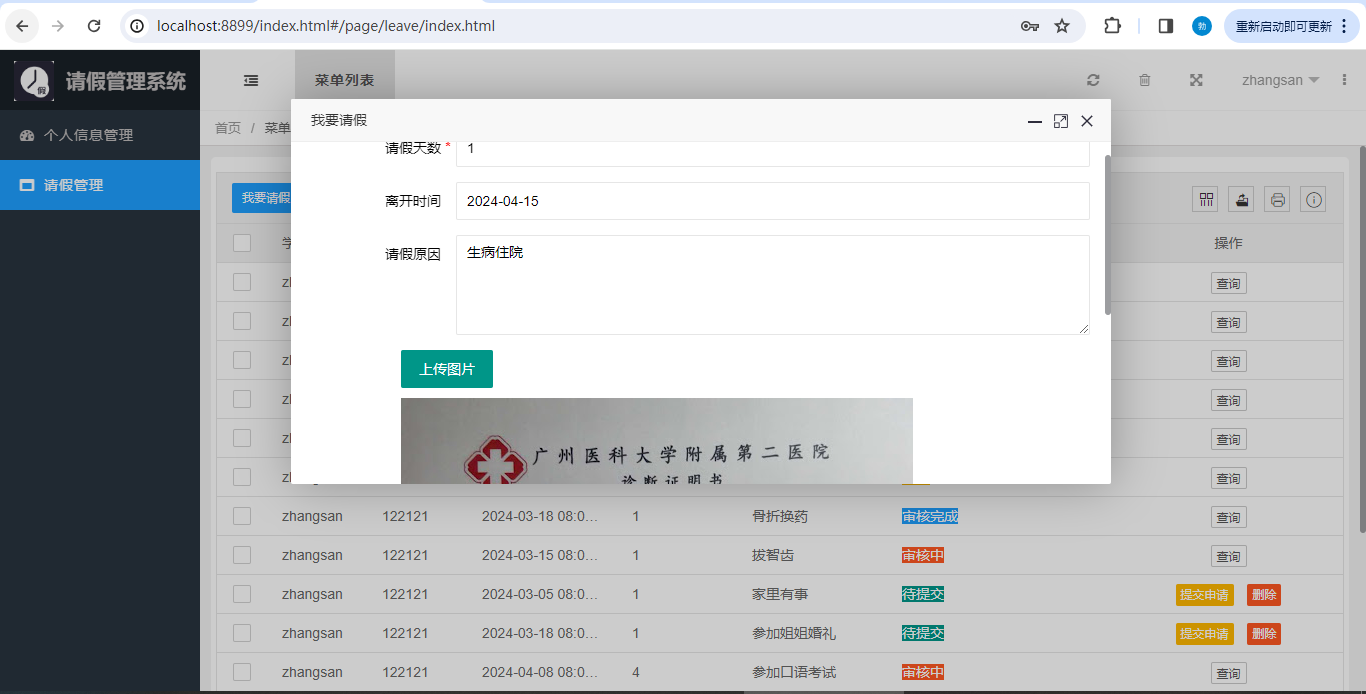
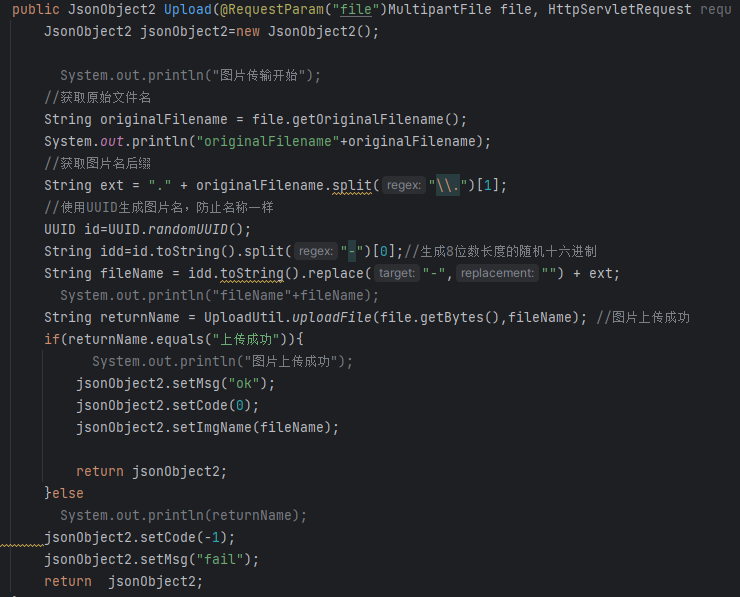


图 7-23学生请假主页面

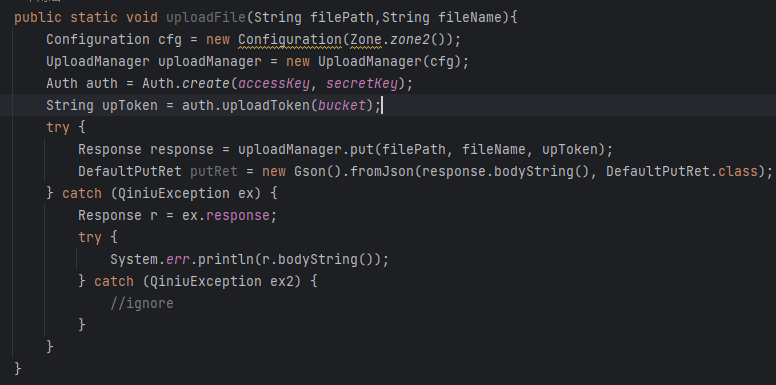


* + - * 1. 图 7-24请假申请表单

在请假单的上传图片功能中，我们不能将图片直接存储在数据库当中，因为图片的占有空间太大，会消耗过多的数据库资源，在本系统中，我们通过上传云储存空间来解决，本系统中，我们上传七牛云存储，为了避免上传图片的名称在七牛云中重复，借助java.util的UUID，生成唯一的图片名，以此名称存入七牛云后通过七牛云对应存储区域的外链域名即可访问。

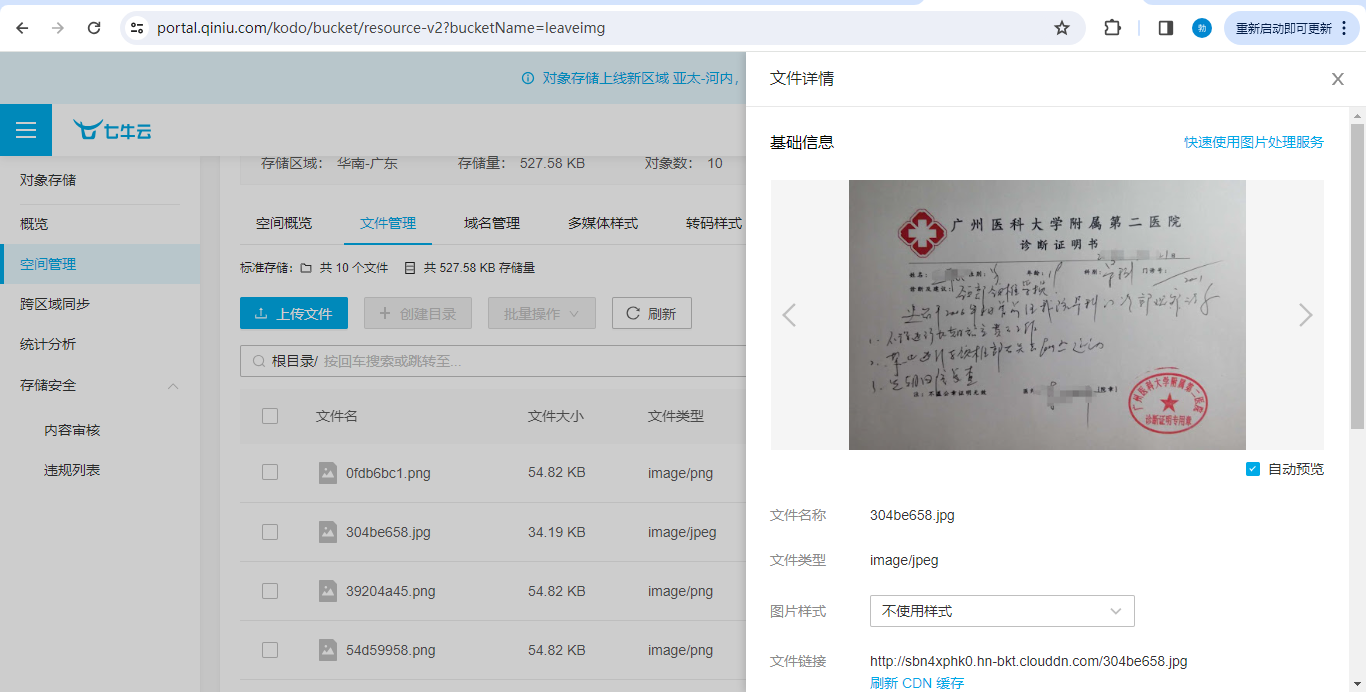


* + - * 1. 图 7-25上传图片后端实现



* + - * 1. 图 7-26七牛云上传代码

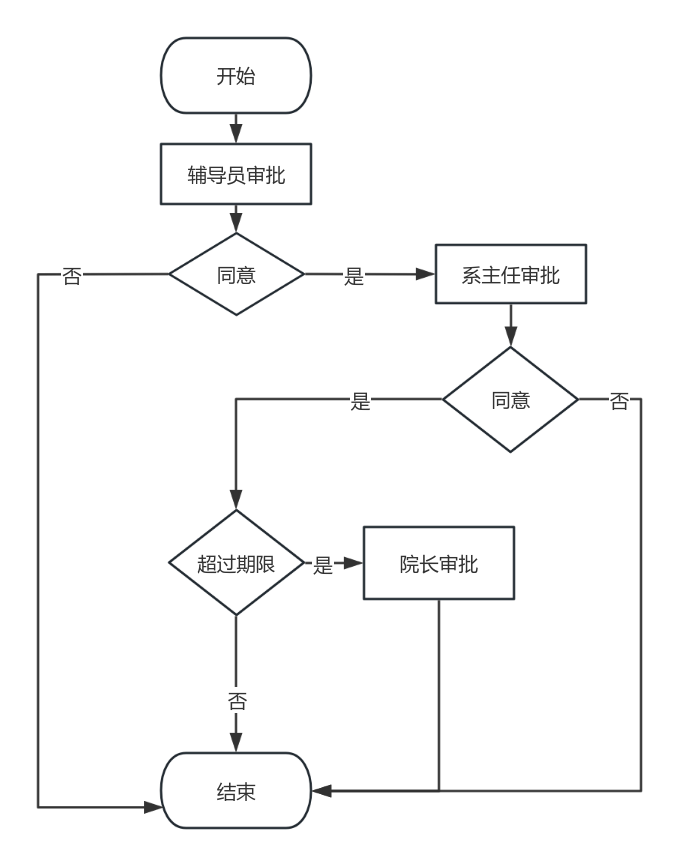
上传七牛云存储需要配置accessKey, secretKey和对应的存储空间名称bucket，其中accessKey是访问密钥，是七牛云账户的标识，用来验证用户的身份。它类似于用户名，用于识别账户。在上传图片至七牛云时，系统通过 Access Key 识别调用者的身份。secretKey是秘密密钥，Secret Key 用于生成签名，保证了请求的安全性。它类似于密码，是与 Access Key 配对使用的，用于生成对API调用进行认证的签名。Secret Key 必须保密，只存储在服务器端，不应暴露给客户端。Bucket则是存储数据的容器，指定 Bucket 是为了明确告诉七牛云系统要将数据存储在哪个特定的存储空间内，是存储空间的名称。此外，如图中所示，也应该配置存储器所在地区，本系统中是Zone.zone2()，表示华南地区服务器。在后续的请假审批时，审批人员进行审批的时候系统前端会直接通过外链域名和对应的图片UUID名称来访问图片，将图片数据传到审批页面中，图7-27展示了在图7-24中学生申请所上传的病历图片证明。



* + - * 1. 图 7-27七牛云存储空间展示

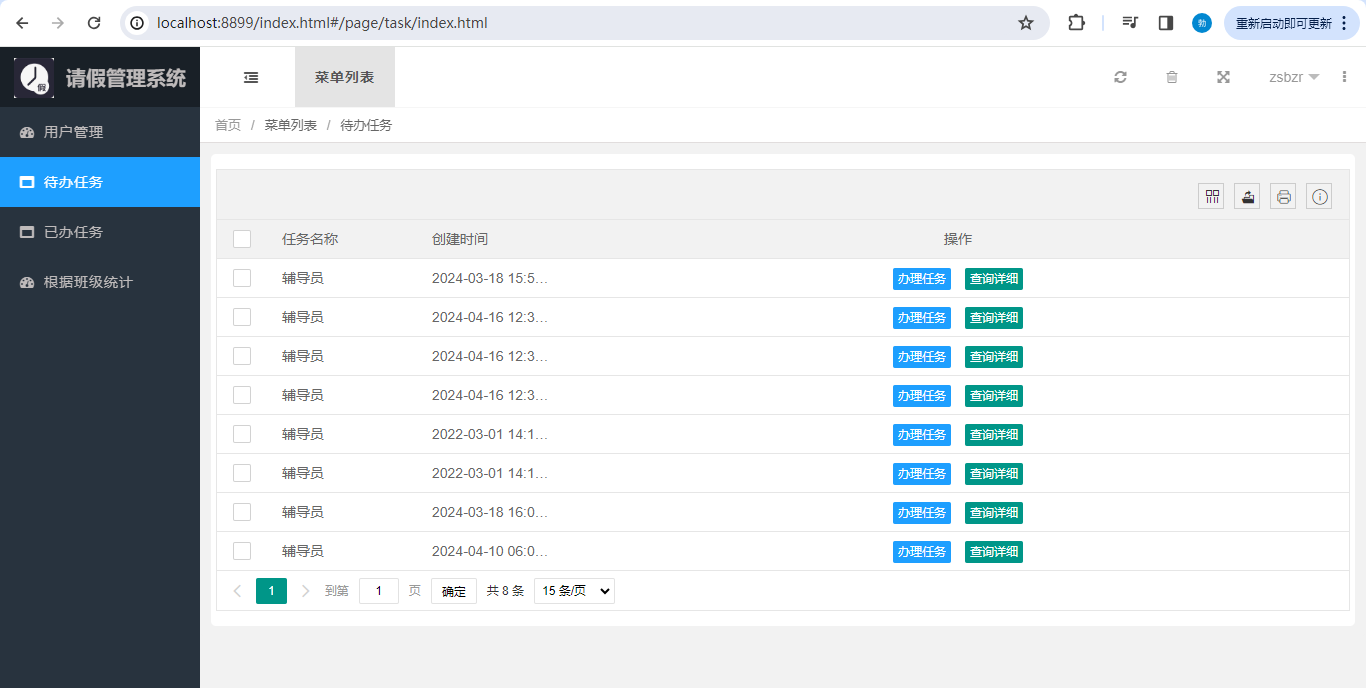
## **审批人员模块**

系统的多级审批包括辅导员，系主任，院长，三者的实现以及系统界面有相似之处，作为一个模块进行介绍，审批流程的BPMN流程图如下图7-28所示，系统开始后，由学生申请请假作为整个请假流程的开始，随后请假审批由辅导员进行初审，辅导员审核通过后交给系主任进行审批，若请假天数不大于管理员设置的限定天数（默认值为5天），那么审批完成，如果超过限制审批天数，请假申请会送到对应的院长处处理。



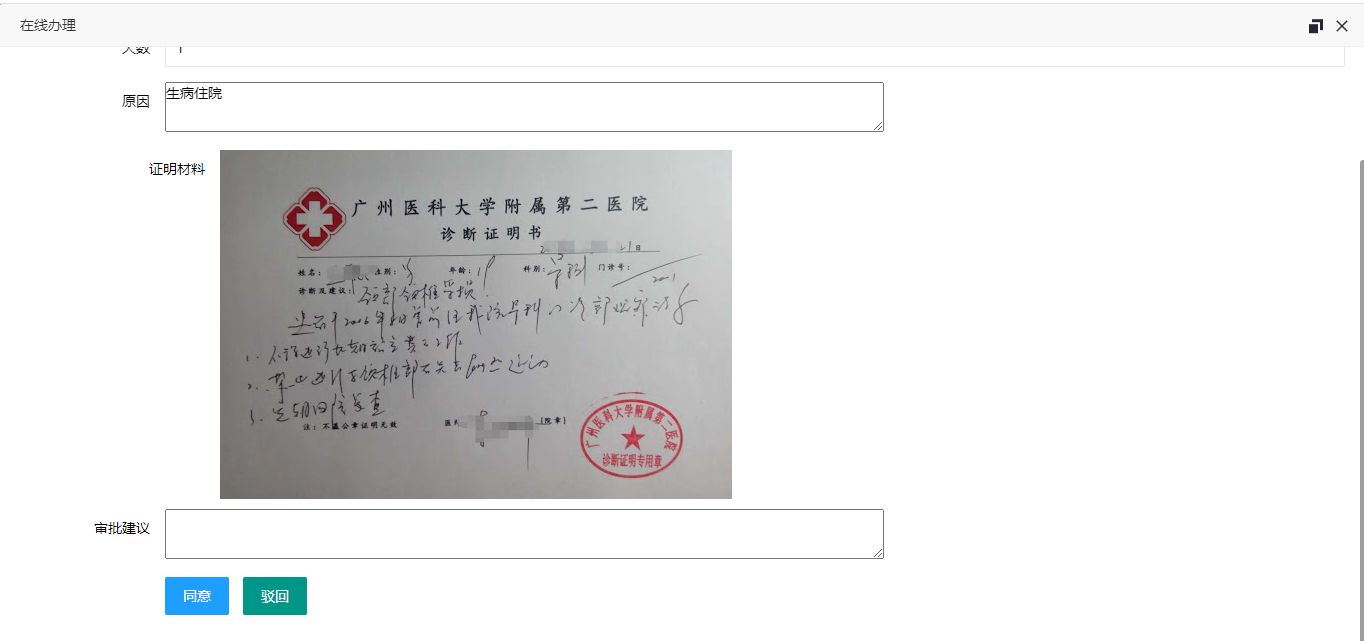
* + - * 1. 图 7-28请假流程图

### **辅导员审批**

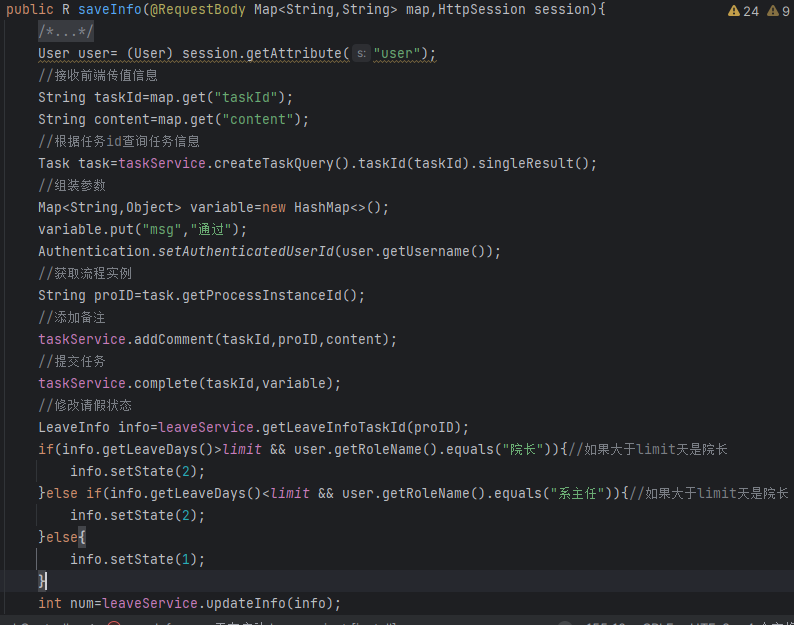


* + - * 1. 图 7-29辅导员代办任务

首先审批由辅导员进行，辅导员可以由代办任务栏查询自己的代办任务，并点击办理任务按钮进行任务办理，由图7-30可以看到，辅导员进行审批时，可以看到学生进行审批时填写的请假时间，请假原因，请假天数信息，并且可以从在线办理表格中直接看到学生上传的图片证明材料，此证明材料通过直接leave\_info表中记录的img\_name直接向七牛云进行访问，得到图片信息并回显在表单之中，辅导员可以在表单中填写审批建议，如“我已了解同意审批”或“不符合要求不同意审批”等批语，此批语能够让审批流程后面的系主任以及院长，以及提交申请的学生查看，在学生方面来看，审批成功与通过与否有理有据，不会出现审批不知什么原因莫名不通过而与辅导员沟通上产生矛盾，对审批流程后面的系主任和院长来说，可以看到更加了解学生辅导员的审批批语，能够帮助系主任和院长决定是否应该同意审批。



* + - * 1. 图 7-30辅导员审批在线办理
        2. 审批有同意和驳回两个选择，以同意审批为例，介绍同意审批的后端代码实现，如图7-31所示。



* + - * 1. 图 7-31同意审批代码实现

同意审批也是工作流中处理任务提交的后端接口，需要对工作流进行操作，同意审批的流程操作大概实现，获取登录信息，添加审核备注信息，提交审核信息，修改请假信息表状态这几个部分，从而使得流程顺利进行，

流程开始使用httpsession获取当前用户和任务信息，先从会话中获取当前登录的用户信息，并从请求体中获取任务ID 和任务执行时的评论内容 。然后利用taskService根据任务ID查询具体的任务实例并设置流程变量，这里设置了一个变量 msg，值为“通过”。而后设置当前操作的用户身份，用于流程审计和追踪。最后为当前任务添加备注，备注内容来自传入的 content。完成当前任务，同时传入流程变量，推动流程向下一步移动。

流程相关内容操作完成后修改相关业务数据：查询与当前任务关联的请假信息 (LeaveInfo)。根据请假天数和用户角色（如院长、系主任），更新请假单的状态。如果请假天数超过设定的限制并且操作用户是院长，或者请假天数未超过限制但操作用户是系主任，则将请假状态设置为2（代表审批同意），否则状态为1（代表审核中）。最后更新请假信息到数据库。

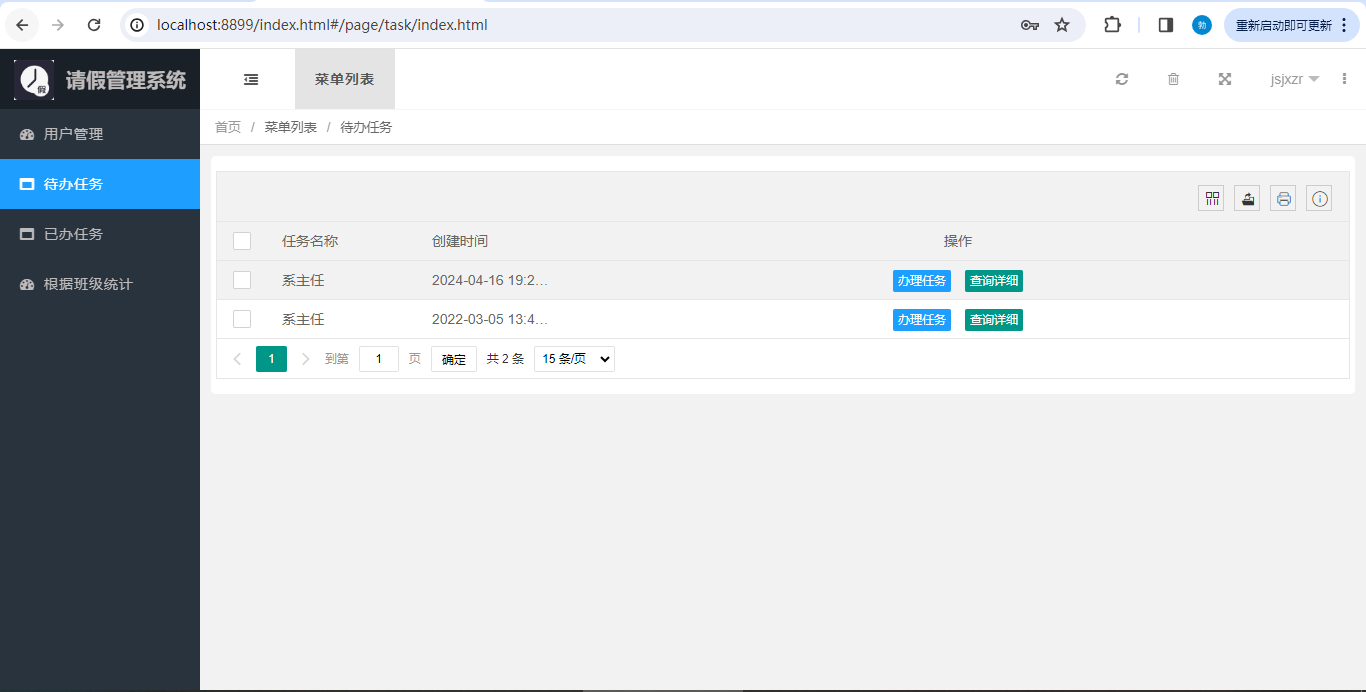
驳回操作与同意 操作类似，如图7-32是驳回操作核心代码，原理与同意类似，实现流程为获取登录信息，添加审核备注信息，提交审核信息，修改请假信息表状态，任务流向的处理。



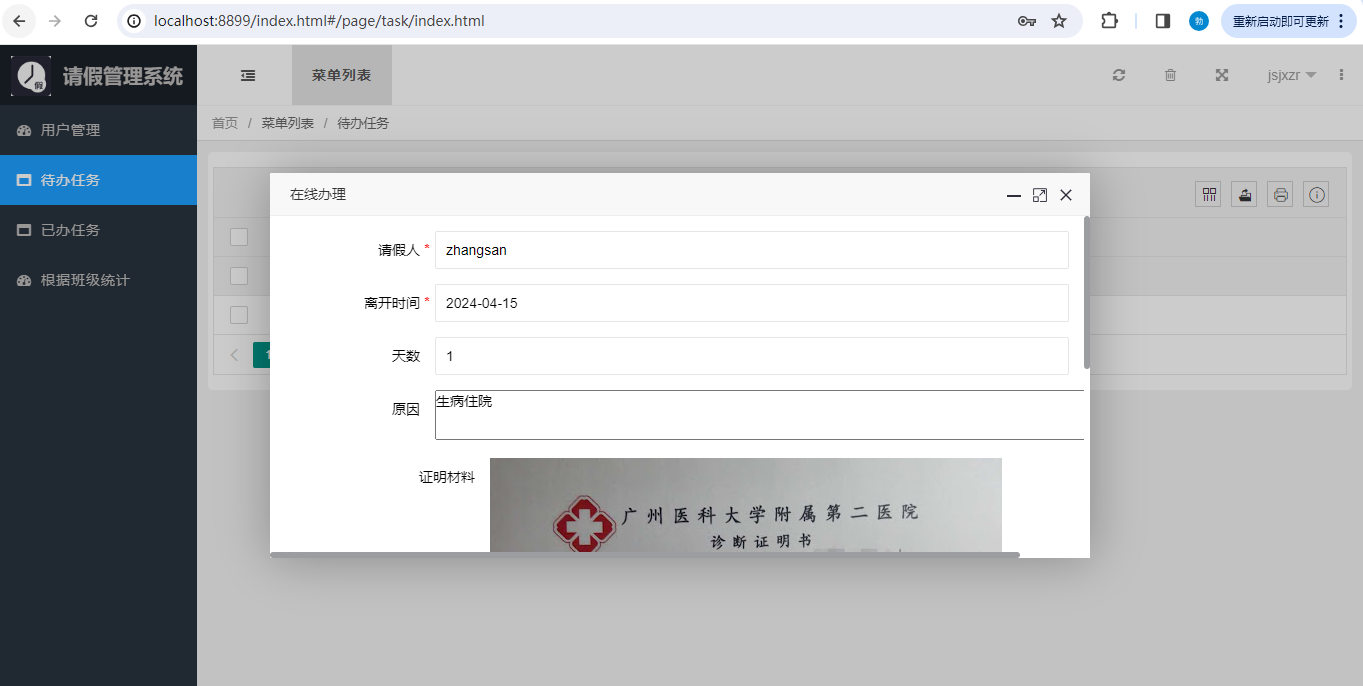
* + - * 1. 图 7-32驳回审批代码实现

### **系主任/院长审批**

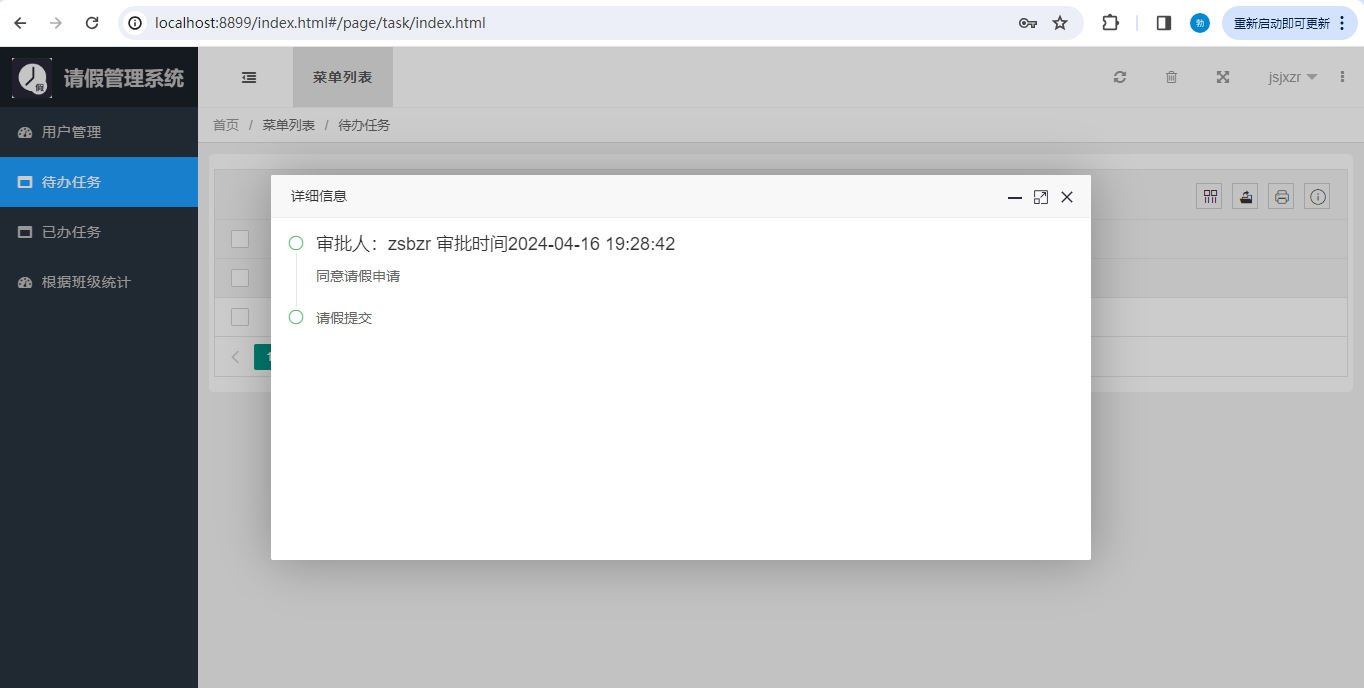
系主任和院长是审批的最后角色，由于两者的审批界面以及实现较为相似，所以一起进行展示。



* + - * 1. 图 7-33系主任审批页面



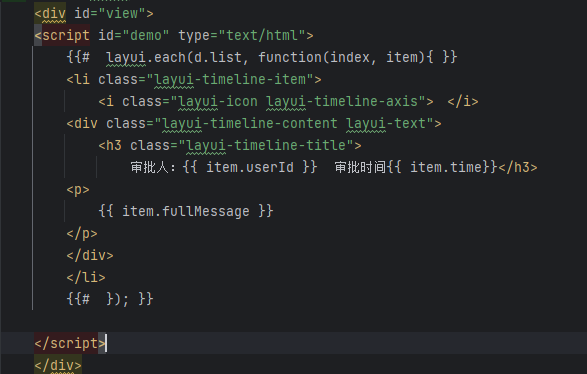
* + - * 1. 图 7-34系主任办理任务



* + - * 1. 图 7-35系主任查看审批信息

由图7-33所示，系主任也拥有与辅导员类似的用户管理，代办任务和已办任务界面，其中已办任务记录系主任已经办理的请假申请，代办任务处允许系主任办理在流程中，经由辅导员审核并已经通过的请假申请，系主任进行审批。若请假的时间天数少于管理员设置的限制期限，那么由系主任审核结束之后审核完毕，若请假时间天数多余管理员设置的限制期限，那么系主任审核成功后请假申请交由院长再进行最后审核。

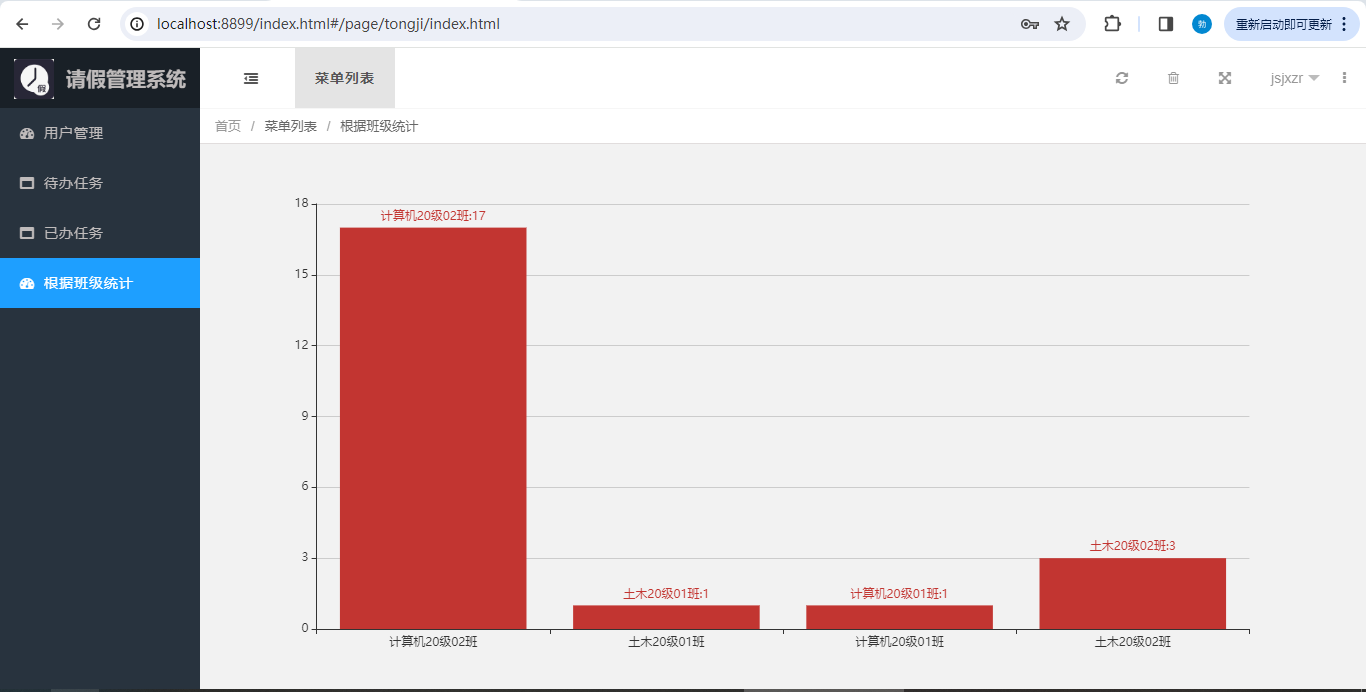
由图7-35所示，系主任可以点击查询按钮，查看上一级的审批信息，在图7-35中，系主任可以看见辅导员在4月16日19点钟的审批信息“同意请假申请”，如图7-34所示，系主任在审批时也能够查看学生天填写的请假时间，请假天数和请假原因以及上传的证明材料信息，系主任在填写审核批语后可以对学生请假信息进行审核。



* + - * 1. 图 7-36前端动态生成审批信息实现

图7-36是实现查询上一个流程审批批语的前端代码，这段代码是Layui框架中使用Laytpl模板引擎编写的一个脚本模板，用于动态生成审批人，审批时间，审批原因。它主要用来显示一个时间轴，展示审批流程中的各个节点信息。其中{{# layui.each(d.list, function(index, item){ }} 是Laytpl模板的数据遍历语句。遍历从服务器获取的数据 d.list，每个元素用 item 表示，index 是当前元素的索引。每次遍历生成一个 <li class="layui-timeline-item"> HTML元素，用于表示时间轴上的一个节点。

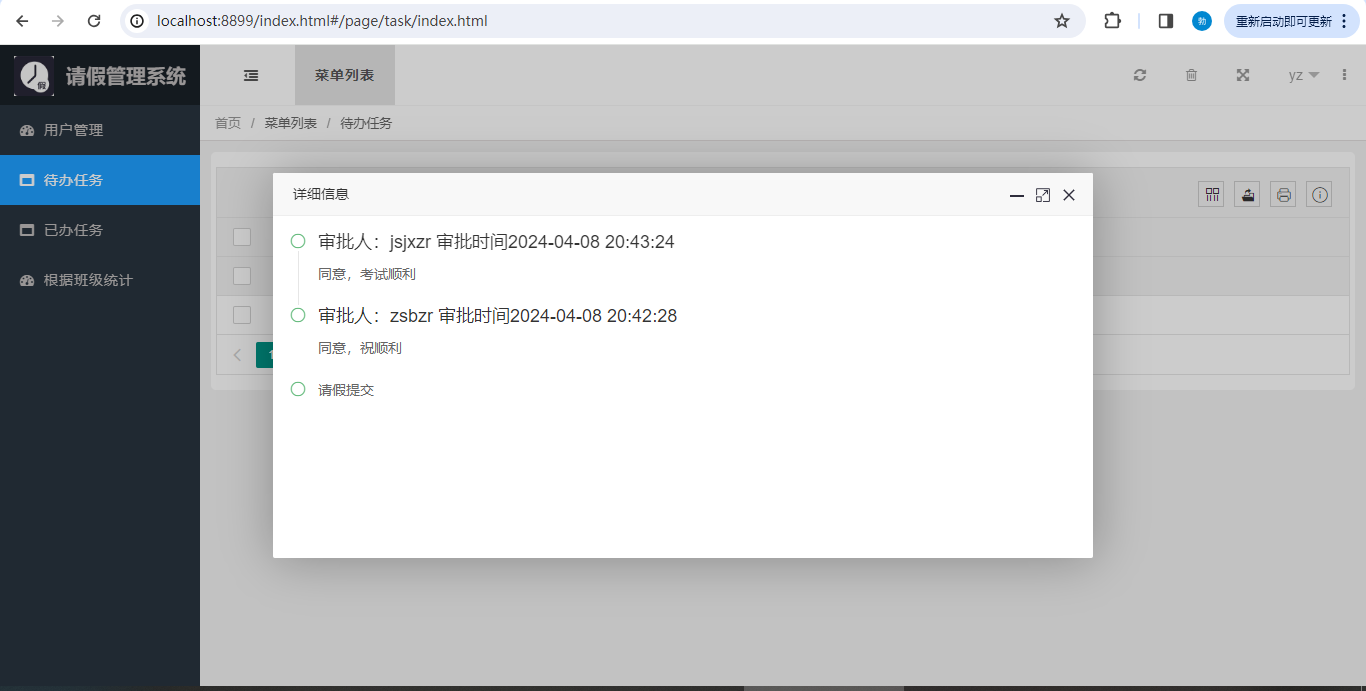
<div class="layui-timeline-content layui-text"> 定义了内容区域，用于展示审批人，审批时间以及审批批语。使用 {{ item.userId }} ， {{ item.time }} 和{{ item.fullMessage }}动态插入数据。



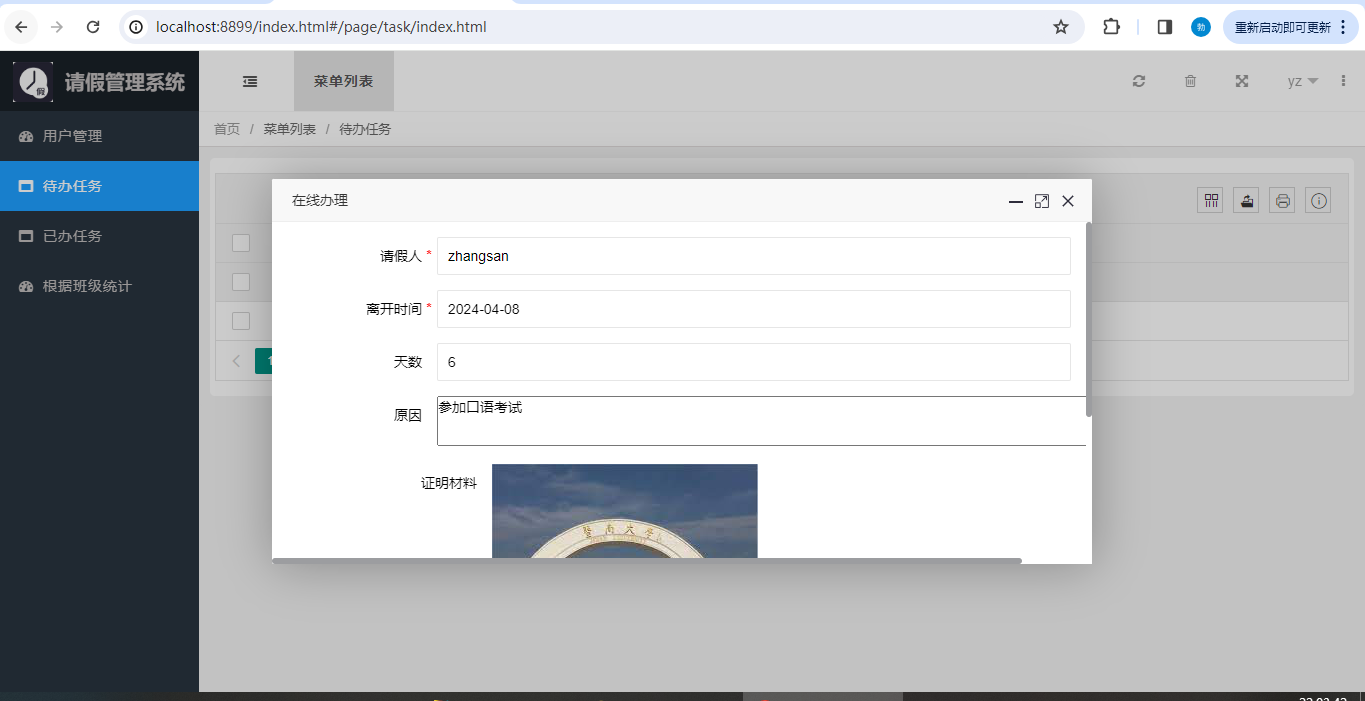
* + - * 1. 图 7-37系主任统计数据

此外，如图7-37，以柱状图的形式简单表示了系各个班级的请假次数情况，便于系主任简单了解请假趋势，更合理的安排教学内容。统计模块使用echart组件，由后端的接口计算出统计数据之后传递到前端，前端由echart绘制出对应的柱状图。

下面展示院长审批相关内容，院长审批与系主任审批类似，但是只会收到经由辅导员，系主任已经审批成功的请假。如图7-39所示，请假天数为6天的数据由院长进行审批。点击查询详情按钮，如图7-38所示，可以看到辅导员和系主任分别的审批批语。在考察辅导员和系主任的审批批语后院长可以结合自身判断和院系相关规定对请假予以最后的审批，审批通过后将请假申请状态改为审批通过，若审批不通过将状态改为驳回，具体的后端实现与系主任相似，不再展示源码。



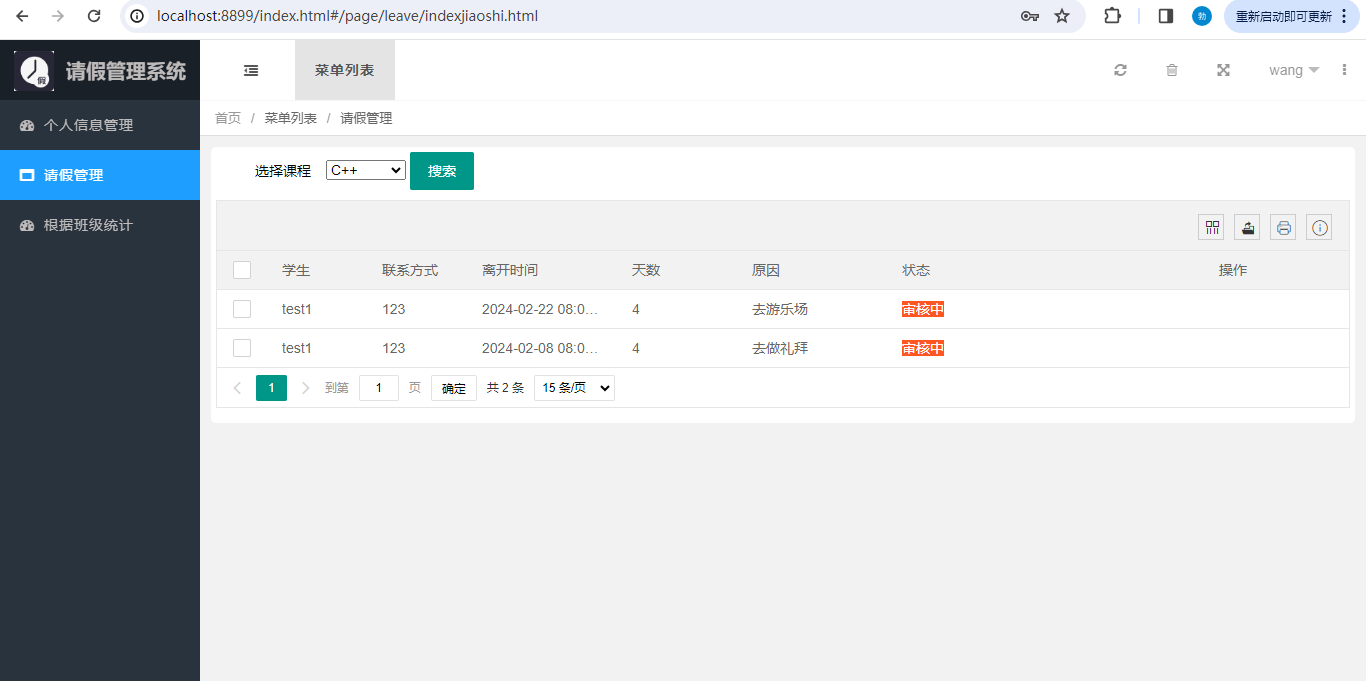
* + - * 1. 图 7-38院长查询请假批语



* + - * 1. 图 7-39院长办理任务

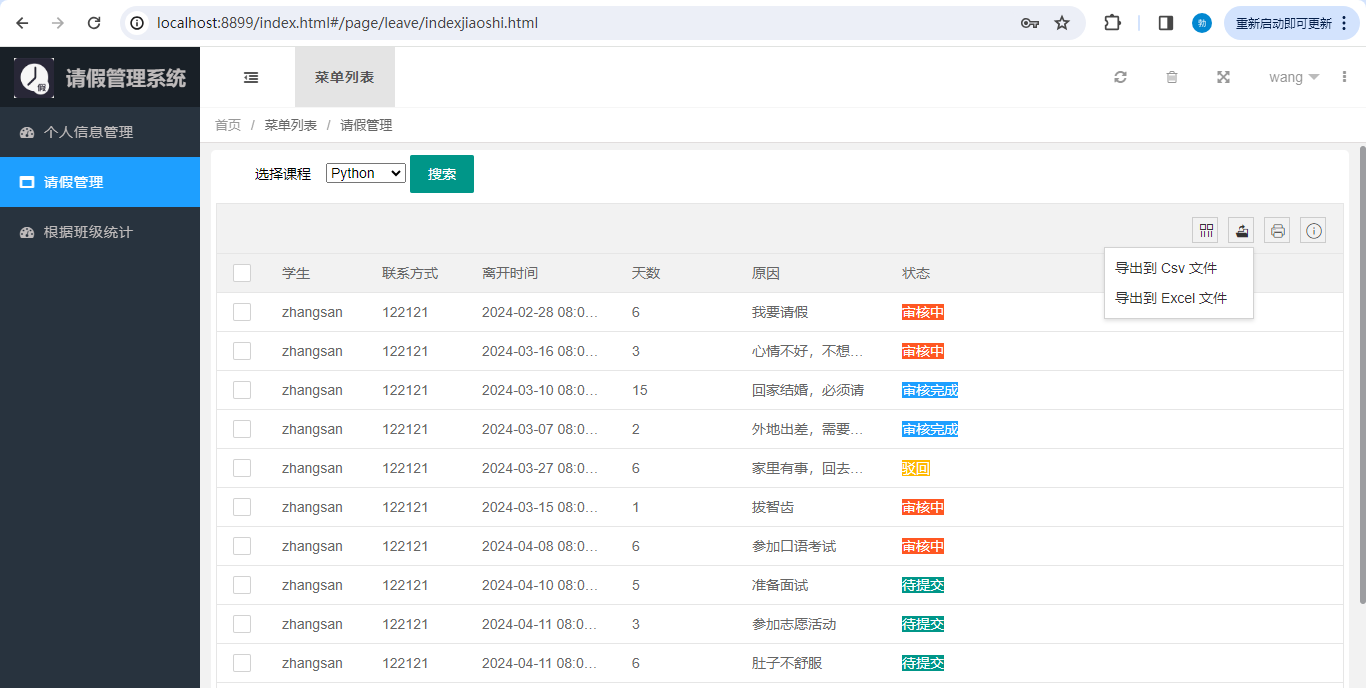
## **教师模块**

教师不直接参与审批流程，功能由个人信息管理以及请假信息的查询，查询按照教师所授的课程进行查询，如果对应选课学生的请假天数覆盖到了课程的上课时间，那么请假记录就会被教师端查询到，否则学生的请假信息不会被查询。



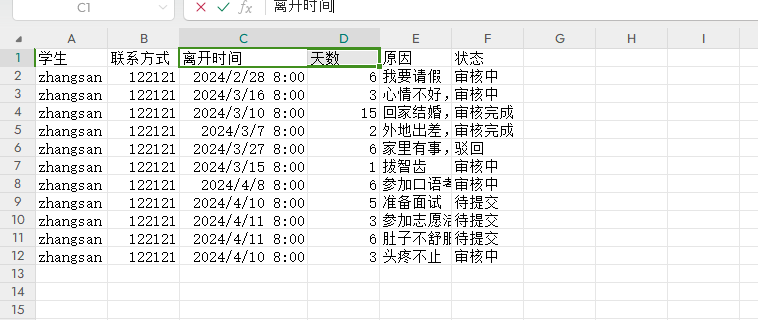
* + - * 1. 图 7-40教师端请假查询

教师在系统中的功能比较简单，主要需求就是获取课程的对应的请假信息以便快速高效的给出学生的出勤成绩。选择课程的选择框由前端动态实现，通过查询教师所授课程来进行动态渲染。

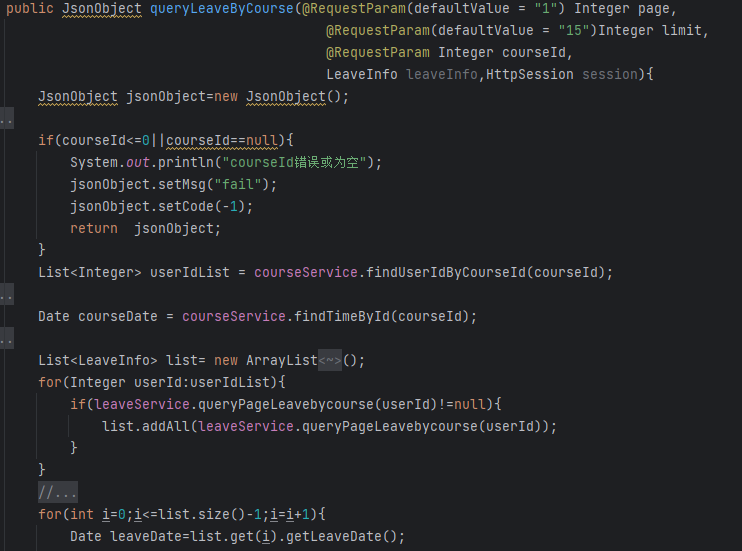


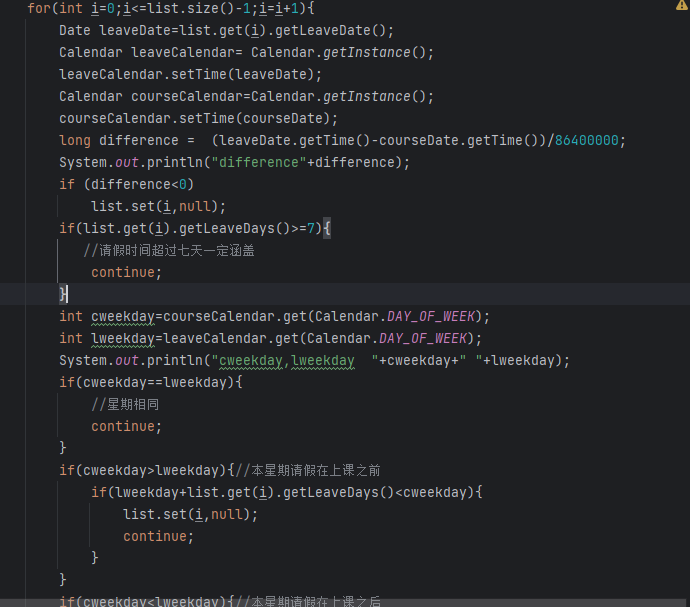
* + - * 1. 图 7-41教师导出按钮

为了方便教师统计，可以导出所有的请假申请到execl文件中，如图7-42所示，教师导出后能够存留纸质材料并且便于出勤成绩统计。



* + - * 1. 图 7-42教师导出的execl数据



* + - * 1. 图 7-43教师查询实现（1）
        2. 图 7-44教师查询实现（2）



* + - * 1. 图 7-45教师查询实现（3）

图7-43，7-44，7-45展示了教师查询请假数据的方法，代码首先检查传入的课程ID是否有效。如果课程ID小于等于0或为空，打印错误信息并返回失败响应。第一步获取所需数据通过课程ID从courseService获取对应的用户ID列表（可能有多个用户与一个课程关联）。获取该课程的具体日期，对每个相关的用户ID，调用leaveService相关方法获取其请假信息，然后将这些信息累加到一个列表中。遍历这个请假信息列表，对每条请假信息执行日期比较和处理，计算请假日期与课程日期之间的天数。如果请假日期早于课程日期，则该请假记录被设置为null。检查请假天数是否超过7天，如果超过则假设请假覆盖了课程日期，不做进一步检查。比较请假日期和课程日期在周中的位置（星期几），根据周中位置和请假天数判断是否覆盖课程日期。若没有覆盖则将其设置为null,若覆盖到则保留信息，不做操作，最后删除所以置空的节点，保留的数据即为教师查询到的数据。将返回的结果返回给前端后结束操作。

# **结论**

本论文成功设计并实现了一个高效的请假管理系统，该系统旨在简化学校中的请假流程，提高请假审批的透明度与效率。通过采用现代的Web技术和工作流管理技术（如Activiti工作流引擎），系统能够提供一个用户友好、响应迅速的平台，使得用户能够方便地提交请假申请，同时管理者可以高效地处理这些申请。

系统的主要功能包括请假申请的提交、审批流程的自动化处理、请假记录的查询与统计以及请假政策的管理。实际运行结果表明，该系统在用户体验、数据处理效率和系统稳定性方面均达到了预期目标。

但是，研究过程中也暴露出一些问题和不足。例如，在处理高并发请求时，系统并未对此有对应的方法。此外，系统的可扩展性和模块间的耦合度也是未来优化的重点。

未来的工作目标计划对系统功能与架构进行优化，比如更多的数据分析模块，更加便捷的统一批量请假信息处理，优化系统展示以提高用户体验等等。

# **致谢**

以下是部分修改密码前端代码

//验证码180秒倒计时  
function setTime (elem) {  
 if (countdown === 0) {  
 // 如果倒计时为0，重置按钮内容为"获取验证码"  
 elem.innerHTML = "获取验证码";  
 countdown = 180;  
 elem.disabled = false; // 按钮可点击  
 clearTimeout(anima);// 清除定时器  
 } else {  
 countdown--;  
 // 如果倒计时不为0，更新按钮内容为"重新获取(倒计时)"  
 elem.innerHTML = "重新获取(" + countdown + ")";  
 elem.disabled = true;  
 // 如果倒计时不为0，更新按钮内容为"重新获取(倒计时)"  
 var anima = setTimeout(function () {  
 setTime(elem);  
 }, 1000);  
 }  
}

# **参考文献**

1. 夏婷婷.基于Web的高校学生请假管理系统的设计与研究[J].洛阳师范学院学报,2016,35(08):52-55.DOI:10.16594/j.cnki.41-1302/g4.2016.08.014.
2. 周玲余.基于jQuery框架的页面前端特效的设计与实现[J].计算机与现代化,2013(01):61-63.
3. 张峰.应用SpringBoot改变web应用开发模式[J].科技创新与应用,2017(23):193-194.
4. 邓志强,邓林强.Maven在Java项目开发中的应用[J].电子元器件与信息技术,2019,3(05):1-4.DOI:10.19772/j.cnki.2096-4455.2019.5.001.
5. 刘玉凯,袁梅梅,童涛,等.基于Activiti工作流引擎的IP管理系统[J].电脑编程技巧与维护,2023(04):169-172.DOI:10.16184/j.cnki.comprg.2023.04.040.
6. 顾少伟,井波.Redis在软件项目中的应用[J].电脑编程技巧与维护,2023(11):16-19+32.DOI:10.16184/j.cnki.comprg.2023.11.046.
7. 李志.论E-R图在数据库建模过程中的重要性[J].信息记录材料,2020,21(06):143-145.DOI:10.16009/j.cnki.cn13-1295/tq.2020.06.085.
8. 王畅.基于BS架构的应用系统国产化环境适配策略及实践[J].网络安全技术与应用,2024(04):17-20.

-------自动修改批注-------：  
文档页眉页脚页边距设置已自动修改为标准模式。