****

**毕 业 设 计**

**题 目**

**姓 名**

**学 号**

**学 院**

**专业年级**

**指导教师**

**2024 年 5月 30日**

中南林业科技大学涉外学院本科毕业设计诚信声明

本人声明：本人所交的中南林业科技大学涉外学院毕业设计，是在指导老师的指导下，独立完成的。该毕业设计不存在知识产权争议，除设计中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表 或撰写过的作品成果。对本设计的研究做出重要贡献的个人和集体均已在文中明确标明。本人完全意识到本声明的法律结果由本人承担。

本科毕业设计作者签名：

2024年 5月 30日

**摘 要**

随着校园和企业规模的不断扩大，传统的请假管理方式逐渐暴露出效率低下、信息不透明、流程复杂等问题。为了提高请假管理的效率和透明度，减少人为错误，开发一套高效、便捷的请假管理系统显得尤为重要。本系统旨在通过信息化手段，实现请假流程的自动化和规范化，提升校园和企业管理的现代化水平。

本系统采用前后端分离的设计模式，后端基于Sprinboot框架开发，提供稳定、高效的接口服务；前端采用UNIAPP框架，支持多端适配，为用户提供一致的使用体验；数据存储采用MySQL数据库，确保数据的安全性和可靠性。系统主要功能包括请假申请、审批流程、请假记录查询、统计分析等，能够满足校园和企业对请假管理的多样化需求。通过本系统，学生或员工可以便捷地提交请假申请，管理者可以高效地处理审批流程，同时系统还提供数据统计功能，帮助校园或企业更好地分析请假数据，优化管理策略。未来，随着智能化、大数据分析技术的进一步发展，基于UniApp的请假管理系统将在更多领域得到广泛应用，并推动相关领域的智能化转型。

**关键词**  请假管理系统；Sprinboot框架；UNIAPP框架；MySQL

**Title** Design and Implementation of a UniApp-based Leave Management System

**Abstract:**

With the continuous expansion of campus and enterprise scales, traditional leave management methods have gradually revealed problems such as inefficiency, lack of transparency, and complex processes. To improve the efficiency and transparency of leave management and reduce human errors, it is particularly important to develop an efficient and convenient leave management system. This system aims to automate and standardize the leave process through information technology, enhancing the modernization level of campus and enterprise management.

The system adopts a front-end and back-end separation design pattern. The back-end is developed based on the Springboot framework, providing stable and efficient interface services; the front-end uses the UNIAPP framework, supporting multi-terminal adaptation and providing users with a consistent experience; data storage relies on the MySQL database, ensuring data security and reliability. The main functions of the system include leave application, approval process, leave record query, statistical analysis, etc., which can meet the diversified needs of campuses and enterprises for leave management. Through this system, students or employees can conveniently submit leave applications, managers can efficiently handle approval processes, and the system also provides data statistics functions to help campuses or enterprises better analyze leave data and optimize management strategies.

**Keywords:** Leave management system; Springboot framework; UNIAPP framework; MySQL

目 录

[1 绪论 1](#_Toc18486)

[1.1 课题研究背景 1](#_Toc12874)

[1.2 课题研究现状 1](#_Toc20598)

[1.3 本文主要工作 2](#_Toc26279)

[1.4 本文组织结构 3](#_Toc35)

[2 技术简介 4](#_Toc21749)

[2.1 B/S架构 4](#_Toc4706)

[2.2 前端框架Vue 4](#_Toc1889)

[2.3 后端框架Spring Boot 5](#_Toc15809)

[2.4 数据库MySQL 5](#_Toc17139)

[2.5 开发工具IDEA 6](#_Toc16665)

[3 请假管理系统分析与设计 7](#_Toc18792)

[3.1 需求分析 7](#_Toc25704)

[3.2 系统可行性分析 1](#_Toc786)3

[3.3 系统数据流图 14](#_Toc19081)

[3.4 数据库设计 15](#_Toc971)

[4 请假管理系统实现 23](#_Toc20353)

[4.1 系统目录结构 23](#_Toc6107)

[4.2 系统前端功能模块的实现 23](#_Toc18055)

[4.3 系统后端功能模块的实现 32](#_Toc17608)

[5 系统测试 41](#_Toc1772)

[5.1 测试的原理 41](#_Toc14011)

[5.2 系统测试用例 41](#_Toc32200)

[5.3 测试结果分析 47](#_Toc14755)

[5.4 本章小结 47](#_Toc21951)

[6 总结与展望 48](#_Toc19028)

[6.1 总结 48](#_Toc10763)

[6.2 展望 48](#_Toc18929)

[致 谢 50](#_Toc1744)

[参考文献 51](#_Toc11634)

# 

# **1** 绪论

## **1.1** 课题研究背景

在当今社会，随着企业规模的不断扩大和校园管理需求的日益复杂，传统的请假管理方式逐渐暴露出效率低下、信息不透明、流程繁琐等问题。无论是企业还是校园，请假管理都是日常运营中不可或缺的一部分，其效率直接影响到整体的管理水平和用户体验。然而，传统的请假管理方式多依赖于纸质表单或简单的电子表格，容易造成数据丢失、审批流程冗长、统计困难等问题，难以满足现代管理的需求。

在企业管理中，员工请假涉及到人力资源调配、工作安排等多个方面，高效的请假管理系统能够帮助企业快速处理请假申请，优化资源配置，提升管理效率。在校园管理中，学生请假涉及到课程安排、考勤统计等，传统的管理方式不仅增加了教师和辅导员的工作负担，还容易出现信息不对称和统计错误。因此，开发一套高效、便捷、透明的请假管理系统，成为企业和校园管理现代化的重要需求。

随着信息技术的快速发展，尤其是移动互联网和云计算技术的普及，为请假管理系统的设计与实现提供了坚实的技术基础。通过引入先进的信息化手段，可以实现请假流程的自动化、数据的集中化管理以及审批流程的透明化，从而显著提升管理效率和用户体验。此外，请假管理系统的应用还能够为企业和校园提供数据支持，帮助管理者更好地分析请假数据，优化管理策略。

综上所述，设计和实现一套高效的请假管理系统，对于提升企业和校园的管理水平、优化资源配置、提高用户体验具有重要的研究意义和实践价值。

## **1.2** 课题研究现状

自近年来，随着信息化管理的普及，请假管理系统逐渐成为企业和校园管理的重要工具。国内外在这一领域的研究和应用已经取得了一定的成果，但仍存在诸多问题和挑战。

在国外，请假管理系统的研究和应用起步较早，尤其是在企业管理领域，许多大型企业已经采用了成熟的请假管理系统。这些系统通常基于云计算和SaaS（Software-as-a-Service）模式，能够实现多终端访问、实时数据同步和高效的审批流程。例如，国外的Workday、SAP SuccessFactors等系统，不仅支持请假管理，还集成了人力资源管理、绩效管理等功能，为企业提供了全面的管理解决方案。然而，这些系统通常价格昂贵，且功能复杂，难以满足中小企业和校园的特定需求。

在国内，请假管理系统的研究和应用相对较晚，但随着信息化建设的推进，越来越多的企业和校园开始重视请假管理系统的建设。目前，国内市场上已经出现了一些成熟的请假管理系统，例如钉钉、企业微信等平台提供的请假功能，能够满足基本的请假管理需求。然而，这些系统通常功能较为单一，缺乏针对性和灵活性，难以满足不同企业和校园的个性化需求。此外，部分系统在数据安全、用户体验等方面也存在一定的不足。

在校园管理领域，请假管理系统的应用主要集中在高校和中小学的日常管理中。一些学校通过自主研发或购买第三方系统，实现了请假流程的电子化和自动化。然而，由于校园管理的特殊性，许多系统在功能设计、用户体验等方面仍有较大的改进空间。

综上所述，尽管国内外在请假管理系统的研究和应用方面取得了一定的进展，但仍存在功能单一、价格昂贵、适应性差等问题。因此，开发一套高效、灵活、适应性强的请假管理系统，具有重要的现实意义和市场价值。

## 1.3 本文主要工作

总本文的主要工作分为两部分：系统的开发实现和论文的撰写。

首先，在系统的开发实现方面，本文基于B/S架构，采用Sprinboot框架进行后端开发，UNIAPP框架进行前端开发，MySQL数据库进行数据存储。系统的主要功能包括请假申请、审批流程、请假记录查询、统计分析等，能够满足企业和校园对请假管理的多样化需求。在开发过程中，注重系统的灵活性、可扩展性和用户体验，确保系统能够适应不同场景的需求。

其次，在论文撰写方面，本文首先对课题背景和研究现状进行了深入分析，明确了研究的必要性和可行性。接着，对系统的需求进行了详细分析，并进行了系统设计和数据库设计。在系统实现部分，详细介绍了前端界面和后端代码的实现过程。最后，对系统进行了全面的测试，确保其稳定性和可靠性，并对整个项目进行了总结和展望。

## 1.4 本文组织结构

本文共分为六章，具体组织结构如下：

第一章为绪论，主要介绍了请假管理系统的研究背景、研究现状、本文的主要工作以及论文的组织结构，为后续研究奠定基础。

第二章为技术简介，详细介绍了开发请假管理系统所需的技术，包括B/S模式、Sprinboot框架、UNIAPP框架以及MySQL数据库等。

第三章为系统分析与设计，首先对系统的需求进行了详细分析，接着对系统的可行性进行了评估，并进行了数据库设计和系统功能设计。

第四章为系统实现，重点介绍了前端界面和后端代码的实现过程，展示了系统的主要功能和实现细节。

第五章为系统测试，对系统进行了功能模块测试、整体系统测试以及验收测试，确保系统满足用户需求。

第六章为总结与展望，对本文的研究工作进行了总结，并提出了未来研究的方向和改进建议。

# **2** 技术简介

## 2.1 C/S架构

C/S架构即客户端/服务器架构，是一种经典的分布式计算模式，广泛应用于各类软件系统中。在C/S架构中，客户端负责用户界面的展示和用户交互，而服务器则负责数据处理和业务逻辑的执行。这种架构的优势在于能够充分利用客户端的计算能力，减轻服务器的负担，同时提供高效的响应速度和良好的用户体验。C/S架构通常适用于对性能要求较高、客户端功能复杂的场景，例如企业级应用、桌面软件等。然而，C/S架构也存在一些缺点，例如客户端的安装和维护成本较高，且难以实现跨平台兼容。随着互联网技术的发展，C/S架构逐渐演变为B/S架构，但其核心思想仍在许多系统中得以保留和应用。C/S架构在企业内部系统、游戏开发、金融交易等领域仍有广泛应用，特别是在对实时性和安全性要求较高的场景中表现出色。

C/S架构其设计理念是将计算任务分布到客户端和服务器两端，客户端负责处理用户交互和界面展示，而服务器则专注于数据处理和业务逻辑。这种分工明确的设计使得C/S架构在高性能计算和复杂业务场景中表现出色。此外，C/S架构还支持离线操作，客户端可以在没有网络连接的情况下继续工作，待网络恢复后再与服务器同步数据。这种特性使其在需要高可靠性和实时性的场景中具有显著优势。

## 2.2 前端框架UNIAPP

UNIAPP是一个基于Vue.js的跨平台前端框架，支持一次性开发，多端部署。开发者可以使用UNIAPP开发同时运行在iOS、Android、H5、微信小程序等多个平台的应用，极大地提高了开发效率和代码复用率。UNIAPP的核心优势在于其跨平台能力和丰富的生态系统。它提供了统一的API和组件库，使得开发者无需为不同平台编写重复代码。同时，UNIAPP支持原生插件扩展，能够满足复杂的业务需求。此外，UNIAPP还提供了强大的开发工具和调试支持，帮助开发者快速构建和测试应用。由于其轻量级和高性能的特点，UNIAPP在移动端开发领域得到了广泛应用，特别适合开发多端兼容的请假管理系统。UNIAPP还支持热更新功能，开发者可以在不重新发布应用的情况下更新代码，极大地提升了用户体验和维护效率。

## 2.3 后端框架Spring Boot

Spring Boot是一个由Pivotal Software开发的开源框架，设计理念是通过自动配置和约定优于配置的原则，简化Java应用程序的开发过程，它是基于Spring框架构建的轻量级、快速开发的框架。Spring Boot提供了“开箱即用”的开发体验，通过自动配置和约定优于配置的原则，减少了开发者的配置工作量。其核心特性包括内置Tomcat服务器、丰富的Starter模块、简化的依赖管理以及强大的集成能力。Spring Boot还支持微服务架构，能够轻松构建分布式系统。由于其简洁的设计和高效的开发模式，Spring Boot在企业级应用开发中得到了广泛应用，特别适合作为请假管理系统的后端框架。Spring Boot还提供了强大的监控和管理功能，例如Actuator模块，可以帮助开发者实时监控系统状态，快速定位和解决问题。此外，Spring Boot的社区活跃，文档丰富，为开发者提供了强大的支持。

此外，Spring Boot还支持多种数据库、消息队列、缓存等技术，开发者可以轻松集成这些技术，构建高性能的应用系统。其Actuator模块提供了丰富的监控和管理功能，开发者可以实时监控系统的健康状态、性能指标和日志信息，快速定位和解决问题。Spring Boot的社区活跃，开发者可以通过社区获取技术支持和新功能建议，进一步提升了开发体验

## 2.4 容器化平台Docker

Docker是一种开源的容器化平台，能够将应用程序及其依赖打包成一个轻量级、可移植的容器。与传统的虚拟化技术不同，Docker容器共享宿主机的操作系统内核，从而显著减少了资源占用和启动时间。Docker的核心优势在于其高效性、一致性和可移植性。通过Docker，开发者可以在本地开发环境中构建容器，并将其无缝部署到生产环境中，确保环境的一致性。此外，Docker还提供了强大的镜像管理、网络配置和存储管理功能，支持多容器协同工作。在请假管理系统的开发中，Docker可以用于快速部署和测试，提高开发效率和系统稳定性。Docker还支持与Kubernetes等容器编排工具集成，能够实现大规模应用的自动化部署和管理。其轻量级特性使得Docker在云计算和微服务架构中得到了广泛应用，成为现代软件开发的重要工具。

## 2.5 NavicatLite

Navicat Lite是一款轻量级的数据库管理工具，支持多种数据库系统，包括MySQL、PostgreSQL、SQLite等。它提供了直观的图形化界面，使得开发者能够轻松地管理数据库、执行SQL查询、导入导出数据以及进行数据库设计。Navicat Lite的核心优势在于其易用性和高效性。它支持多连接管理，能够同时连接多个数据库，并提供了强大的数据同步和备份功能。此外，Navicat Lite还支持SSH和SSL加密，确保数据的安全性。在请假管理系统的开发中，Navicat Lite可以用于数据库的设计、调试和维护，提高开发效率和数据管理的可靠性。Navicat Lite还提供了可视化查询构建器，开发者可以通过拖拽操作快速生成复杂的SQL语句，极大地简化了数据库操作。其跨平台支持特性使得开发者可以在Windows、macOS和Linux系统上无缝使用，进一步提升了开发的便捷性。

# **3** 请假管理系统分析与设计

## 3.1 需求分析

请假管理系统需要满足基本的一个校园与企业运作，本系统以高校为应用场景，为高校提供一套高效、便捷的请假管理解决方案，解决传统请假方式中存在的效率低下、信息不透明、流程繁琐等问题。通过信息化手段，实现请假流程的自动化、数据的集中化管理以及审批流程的透明化，从而提升管理效率和用户体验。

系统功能模块如图3.1所示：

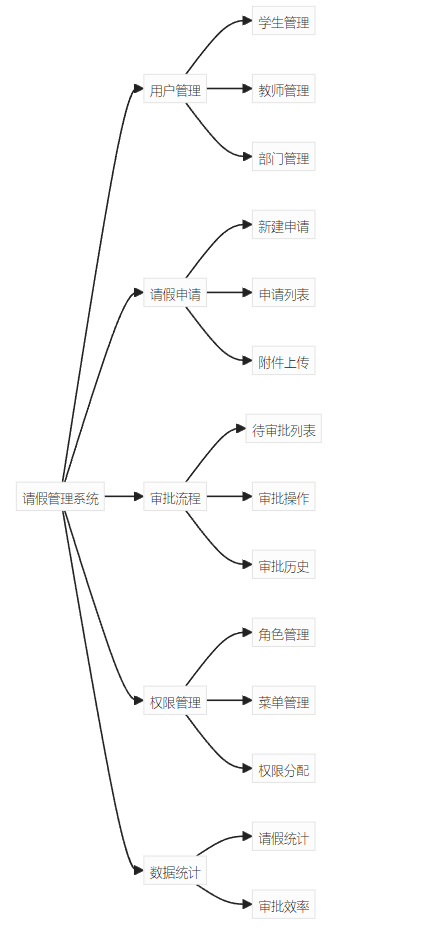


图3.1 系统功能模块图

以上是请假管理系统的一些主要功能模块，这些模块相互协作，共同构成了整个系统的核心功能。通过这些功能，请假管理系统能够为高校提供一整套高效便捷的管理方案，提高高校的运营效率和管理水平。具体的功能说明如下；

本系统以高校为应用场景，主要功能模块包括：

用户登录模块：学生、辅导员、教务处和考勤人员通过登录窗口输入用户名和密码，系统根据用户角色分配不同的权限。

学生模块：学生可以提交请假申请，查看请假历史，撤销未审批的请假申请，以及提交销假申请。

请假管理模块：辅导员可以查看学生的请假申请，根据实际情况批准或提交至教务处审批，同时处理学生的销假申请。

审批流程模块：教务处负责审批学生的请假申请，查看历史请假信息，并对请假

具体用例图如图3.2、图3.3、图3.4所示：

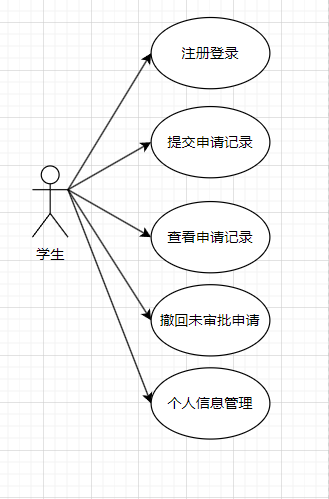


图3.2 学生用例图

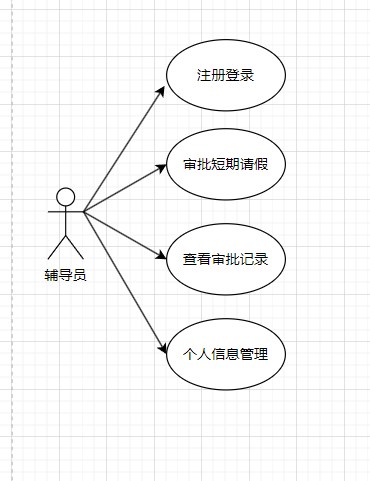


图3.3 辅导员用例图

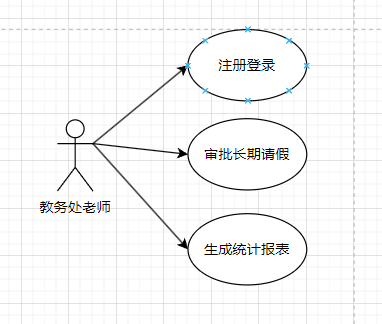


图3.4 教务处老师用例图

用例描述如表3.1、表3.2、表3.3所示：

表3.1 学生请假操作用例描述表

|  |  |
| --- | --- |
| **用例名称**： | 学生申请请假 |
| **参与者**： | 学生-用户 |
| **简要说明**： | 学生填写请假信息，提交请假信息 |
| **前置条件：** | 学生成功注册和登录应用 |
| 基本事件流： | 1. 学生填写请假相关信息 2. 学生提交请假信息 |
| 后置条件： | 显示本人请假记录的信息 |

表3.2辅导员审批用例描述表

|  |  |
| --- | --- |
| 用例名称： | 辅导员审批学生请假 |
| 参与者： | 辅导员-用户 |
| 简要说明： | 辅导员查看请假信息 |
| 前置条件： | 辅导员成功登录系统 |
| 基本事件流： | 1. 入驻应用查看申请请假信息 2. 对学生产生的请假信息进行回复和审批 |
| 后置条件： | 显示审批结果信息 |

表3.3管理员公告管理用例描述表

|  |  |
| --- | --- |
| 用例名称： | 管理员公告管理 |
| 参与者： | 管理员 |
| 简要说明： | 管理员管理系统公告 |
| 前置条件： | 管理员成功登录系统 |
| 基本事件流： | 1.管理员查看公告模块  2.管理员对公告进行更新 |
| 后置条件： | 显示更新后的公告信息 |

其中部分功能描述如下：

(1)未注册的用户：考虑到为高校师生提供服务，未注册的用户无法进入应用。如果没有拥有本系统的登录账号，系统将会提供便捷的账户注册功能，引导访客完成注册流程。

(2)用户登录注册：用户输入自己的用户名和密码登录该系统，如果登录成功即可显示系统主界面。如果显示账号或密码不正确则登录失败，则需要用户进行注册或联系管理员重置密码。

(3)个人信息管理：成功登录到系统的用户可以进行个人信息管理，进入个人中心可以查看到自己的具体信息，并且可以对自己的个人信息进行修改。

(4)辅导员：辅导员可以对其学生的请假信息进行查看，能够查看请假类型、开始时间、结束时间等信息，还能通过请假管理对学生的请假进行审批操作。

学生请假时序图如图3.5所示：

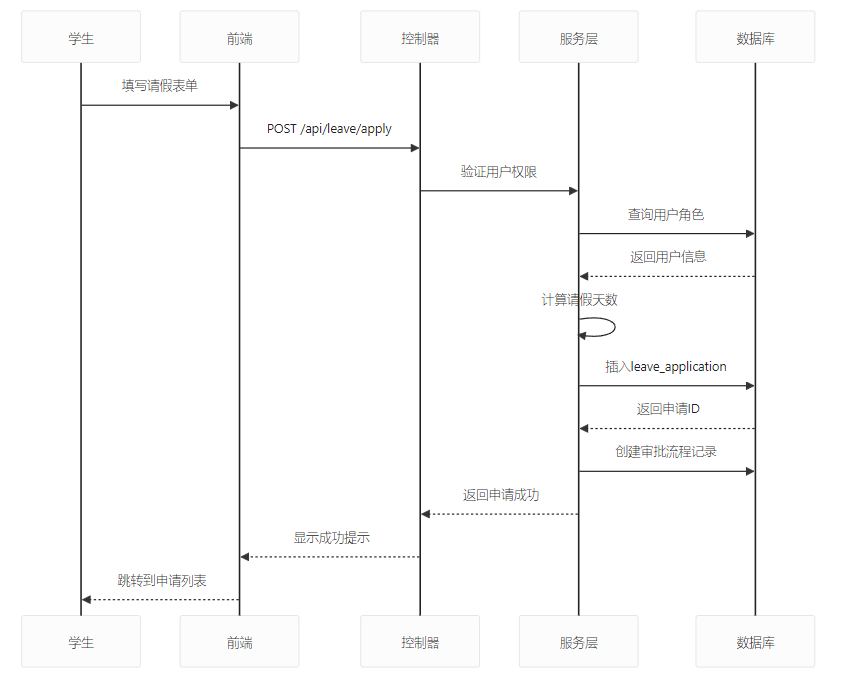
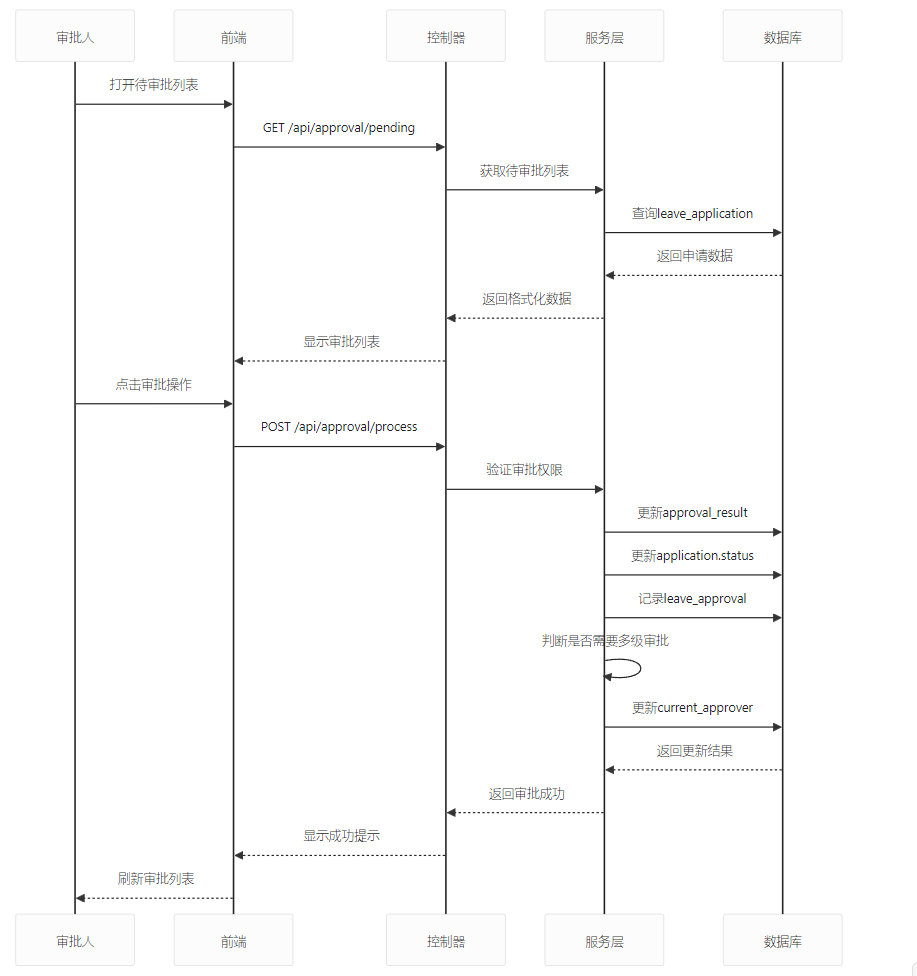


图3.5 学生请假时序图

请假审批流程图如图3.6所示。

图3.6 请假审批流程时序图

## 3.2 系统可行性分析

3.2.1 经济可行性

建立请假管理系统具有很高的经济可行性，经济可行性主要分析成本和收益。通过采纳请假管理系统的运用可以显著削减高校在人力资源及日常管理方面的开支，并提升整体运营效率。系统采用B/S架构,主要业务逻辑在服务器上实现,因此对使用系统的计算机和移动端设备的要求不高[10]。本系统主要依托于Spring Boot框架与MySQL数据库环境运行，核心投入主要是软件的开发与后续的维护。不过当系统正式投入运营后，其带来的效益便会远超过这些初始投入。它不仅为广大用户带来了前所未有的便捷体验，极大地节省了他们的时间与精力，同时，对于高校的运营者来说，这一系统更是显著地减轻了他们的工作负担，提高了运营效率。从长远来看，该系统在时间效率提升和经济利益累积上的正面效应，大大超越了其后期的维护与管理成本。具体而言，请假管理系统集成了公告发布、轮播图展示等功能，方便用户随时随地了解高校的课程和促销活动信息，这可以促进收入的增长。此外，通过系统对用户信息的精确管理，能够为用户提供更加个性化的服务，进而提升用户的满意度，从而进一步推动高校经济效益的提升。因此开发此系统是合适可行的。

3.2.2 技术可行性

在技术层面上，构建请假管理系统是完全可行的。技术可行性是衡量技术解决方案的实用性和技术资源的可用性[11]。随着互联网的迅猛发展和广泛普及，相关的网络技术如网站构建、服务器运营、数据储存等也得到了广泛的运用和推进，这使得请假管理系统的构建过程更加顺畅和高效。Vue等前端框架的崛起，为开发者提供了一种快速搭建美观且用户友好的网站界面的途径。Spring Boot等后端开发框架不仅能够提升开发效率，减少冗余工作，还能使开发者更加专注于项目逻辑的实现。在构建请假管理系统的过程中，安全技术是不可或缺的一环。系统必须精心设计用户数据的保护措施，如实施用户信息的加密传输策略、有效防范SQL注入等安全风险，以确保用户数据的安全与稳定。从技术角度来看，请假管理系统的构建具有极高的可行性。通过合理的技术架构和开发流程，以及运用先进的技术工具和框架，能够构建出一款功能完备、安全可靠且用户友好的请假管理系统。

3.2.3 社会可行性

请假管理系统的社会可行性也是很高的。随着健身行业的快速发展，社会健身消费也不断增长，传统的请假管理方式已经无法满足现代高校的需求。请假管理系统可以大大提高运营效率，减少人工成本，提供更优质的服务，满足社会的需求。同时也可以通过数据分析，帮助高校制定更科学的营销策略，提高高校经济收益。随着信息技术的不断发展，实现请假管理系统的技术已经非常成熟。利用现有的技术，完全可以开发出稳定、易用、功能全面的请假管理系统。在如今人们愈发注重健康和形体的社会，人们对健身的价值认知逐渐提高，有调查数据表明,上海莱美特高校的会员在对健身的价值认知这一问题上93%的会员认为健身对人们的贡献还是不容忽视的,只有少数人不能清楚的认识到健身对人们的重要性[12]。请假管理系统的引入，不仅能够显著提升高校的运营效率和服务品质，同时还对健身行业的蓬勃发展起到积极的推动作用，提高人们的健身意识，对社会产生积极的影响。从国家的健身规划来看，要达成健身消费的升级目标，关键在于深入理解并把握消费群体的实际需求，只有这样才能提升国民的健身投入，增强国民身体素质，才能推进国家健身战略的实施。

## 3.3 系统数据流图

数据流图是阐述数据在软件内部流转与处理逻辑过程的图示，采用特定的图形元素来呈现这些数据流。它详尽地揭示了系统内各个处理环节间的数据关联，是结构化系统分析手法中不可或缺的重要工具之一。通过数据流图能够更直观地理解数据在系统中的流动与处理情况。本系统的数据流图如图3.8所示：

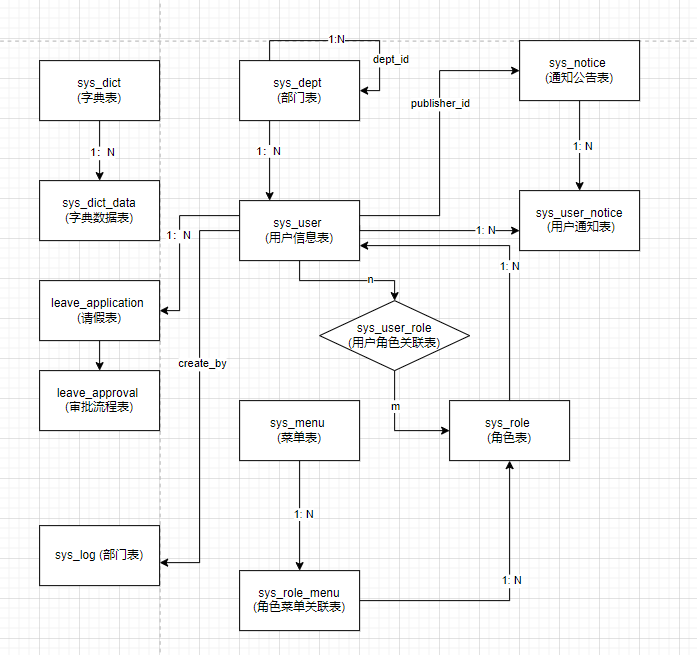


图3.8 系统数据流图

## 3.4 数据库设计

3.4.1 概念结构设计

在需求分析阶段，我们收集到了实现系统所需的基本数据和信息。为确保这些数据能更精确地反映系统特性，需要构建一个概念数据模型。此模型依据本系统的信息构建，能映射现实环境，进而促进对系统的深入理解和灵活调整。通常会采用E-R（实体-关系）模型来阐述这一概念模型，它由数据实体、实体的属性及实体间的关联关系共同构成。本系统的E-R图如图3.9所示：

图3.9 系统ER图

1. 部门实体（sys\_dept）​​

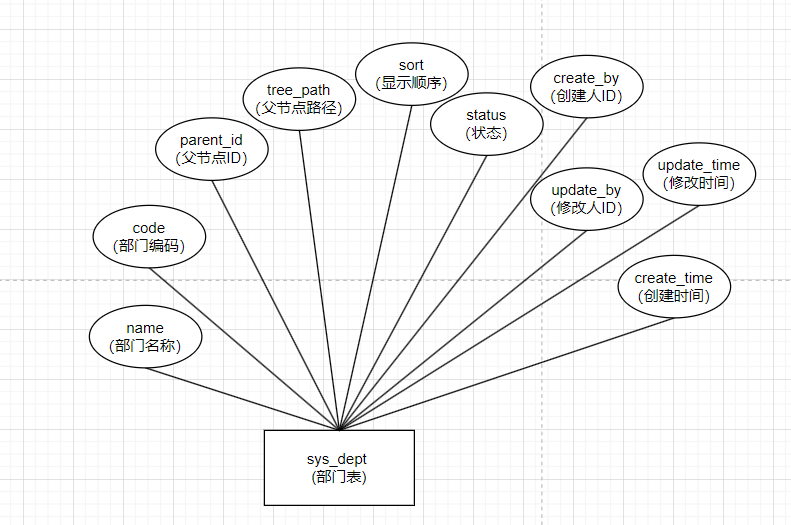
主要包含部门名称、部门编号、父节点ID、层级路径、显示顺序、状态等属性，用于实现组织架构的树形管理。  


图3.10 部门实体属性图

1. 字典实体（sys\_dict）​​

包含字典编码、类型名称、状态、备注等属性，支持系统动态配置字典类型，如性别、通知类型等。

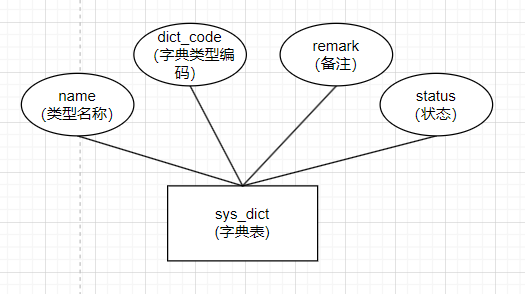


图3.11 字典实体属性图

1. 菜单实体（sys\_menu）​​

包含菜单名称、类型（目录/菜单/外链/按钮）、路由路径、组件路径、权限标识、图标等属性，用于构建动态路由和权限控制。

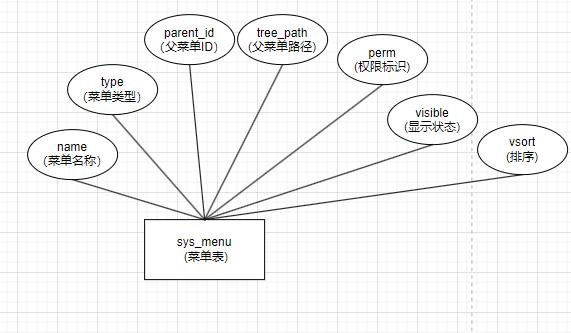


图3.12 菜单实体属性图

1. 角色实体（sys\_role）​​

包含角色名称、角色编码、数据权限范围、状态等属性，支持基于角色的访问控制（RBAC）。

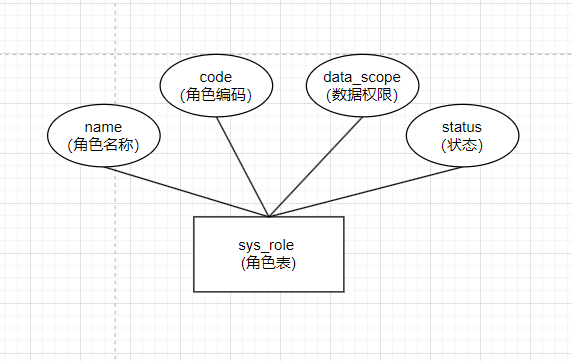


图3.13 角色实体属性图

1. 用户实体（sys\_user）​​

包含用户名、密码、昵称、性别、部门ID、头像、联系方式等属性，实现用户身份认证与基础信息管理。

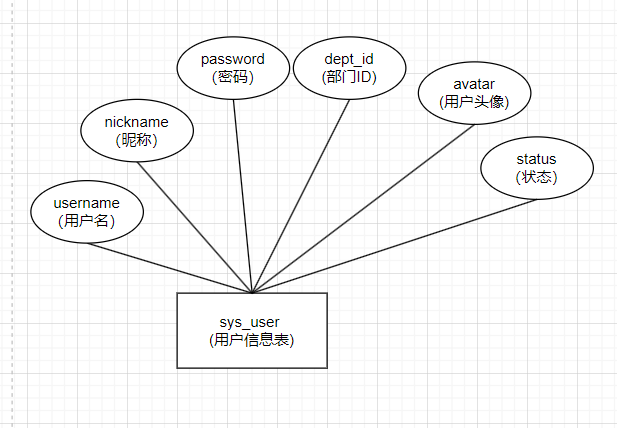


图3.14 用户实体属性图

1. 系统日志实体（sys\_log）​​

包含操作模块、请求参数、IP地址、执行时间、浏览器信息等属性，记录用户操作和系统运行日志。

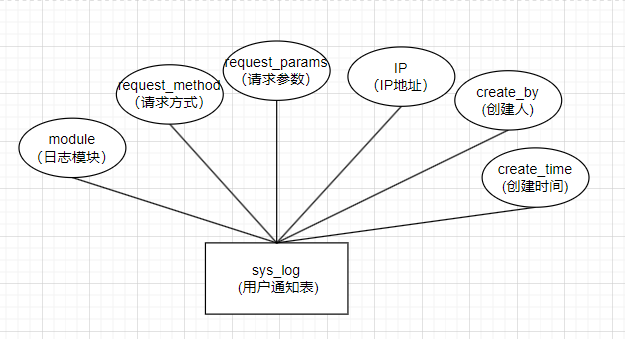


图3.15 用户通知实体属性图

1. 通知公告实体（sys\_notice）​​

包含标题、内容、类型、发布状态、目标用户等属性，支持系统内通知的发布与管理。

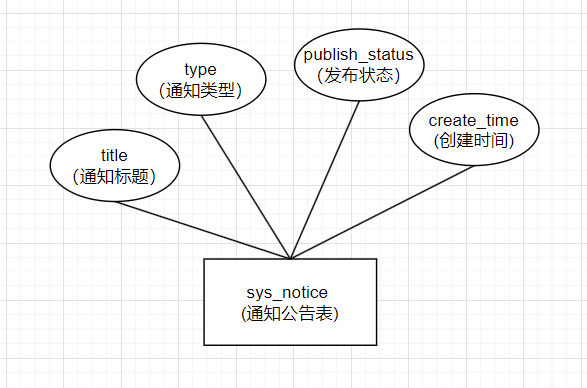


图3.16 通知公告实体属性图

1. 请假申请实体（leave\_application）

包含学生ID、请假类型（短期/长期）、起止时间、请假天数、事由说明、证明材料、审批状态、当前审批人、时间戳等属性，支持学生在线提交请假申请、审批状态实时跟踪、附件材料上传及流程进度管理。

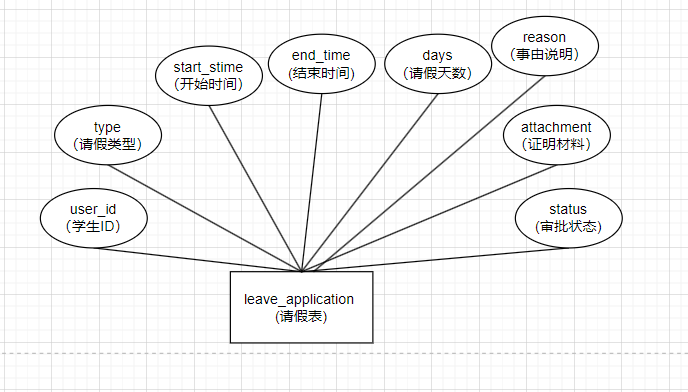


图3.17 请假表实体属性图

1. 审批流程实体（leave\_approval）

包含关联请假单ID、审批人ID、审批结果（同意/驳回）、审批意见、审批时间等属性，支持多级审批流程记录（如辅导员初审→教务处终审）、审批意见可视化反馈及审批操作全链路追溯。

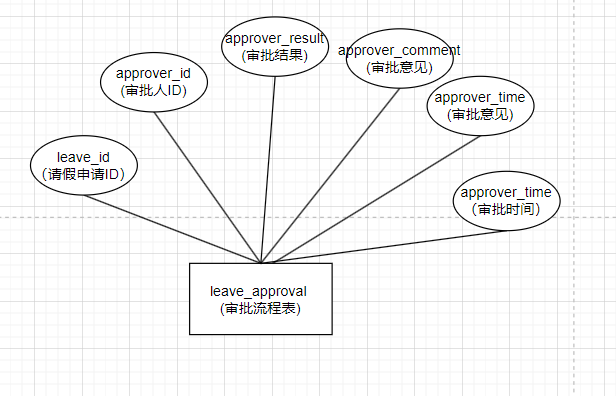


图3.16 审批流程实体属性图

3.4.2 逻辑结构设计

在此阶段需要将概念设计阶段获取的E-R图（实体-关系图）转化为与所选数据库管理系统兼容的逻辑结构。首先将E-R图转化为相应的关系模式，接着将其适配到数据库管理系统所支持的具体数据模型中。然后进行数据表的构建，并在表结构构建完成后，对它们进行详尽的优化调整，这些调整包括但不限于确立数据间的相互依赖关系，以及对关系模式实施恰当的分解等。以确保数据结构的合理性和高效性。本系统的主要数据库表如表3.4-表3.11所示：

表3.4 sys\_dept（部门表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| name | varchar | 100 | NO |  | 部门名称 |
| code | varchar | 100 | NO |  | 部门编号（唯一） |
| parent\_id | bigint |  | YES | 0 | 父节点ID |
| tree\_path | varchar | 255 | NO |  | 父节点路径 |
| sort | smallint |  | YES | 0 | 显示顺序 |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-禁用） |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES |  | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES |  | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

说明：部门表采用树形结构存储，支持多层级组织架构管理，通过tree\_path字段实现快速层级查询。

表3.5 sys\_dict（字典表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| dict\_code | varchar | 50 | YES |  | 字典类型编码 |
| name | varchar | 50 | YES |  | 类型名称 |
| status | tinyint |  | YES | 0 | 状态（0-正常 1-禁用） |
| remark | varchar | 255 | YES |  | 备注 |
| create\_time | datetime |  | YES |  | 创建时间 |

说明：字典表用于管理系统枚举值（如性别、通知类型），通过dict\_code字段与字典数据表关联。

表3.6 sys\_menu（菜单表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| parent\_id | bigint |  | NO |  | 父菜单ID |
| tree\_path | varchar | 255 | YES |  | 父节点路径 |
| name | varchar | 64 | NO |  | 菜单名称 |
| type | tinyint |  | NO |  | 菜单类型（1-菜单 2-目录 3-外链 4-按钮） |
| route\_path | varchar | 128 | YES |  | 路由路径 |
| component | varchar | 128 | YES |  | 组件路径 |
| perm | varchar | 128 | YES |  | 权限标识 |
| visible | tinyint |  | YES | 1 | 显示状态（1-显示 0-隐藏） |
| sort | int |  | YES | 0 | 排序 |
| icon | varchar | 64 | YES |  | 菜单图标 |

说明：菜单表支持多级嵌套结构，通过type字段区分菜单类型，perm字段实现细粒度权限控制。

表3.7 sys\_role（角色表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| name | varchar | 64 | NO |  | 角色名称（唯一） |
| code | varchar | 32 | NO |  | 角色编码（唯一） |
| data\_scope | tinyint |  | YES |  | 数据权限（0-全部 1-本部门） |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-停用） |

说明：角色表通过data\_scope字段实现数据权限控制（如部门隔离），支持RBAC权限模型。

表3.8 sys\_user（用户信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| username | varchar | 64 | YES |  | 用户名（唯一） |
| nickname | varchar | 64 | YES |  | 昵称 |
| password | varchar | 100 | YES |  | 密码 |
| dept\_id | int |  | YES |  | 部门ID |
| avatar | varchar | 255 | YES |  | 用户头像 |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-禁用） |

说明：用户表通过dept\_id关联部门表，支持组织架构归属管理，密码采用BCrypt加密存储。

表3.9 sys\_user\_role（用户角色关联表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| user\_id | bigint |  | NO |  | 用户ID（联合主键） |
| role\_id | bigint |  | NO |  | 角色ID（联合主键） |

说明：用户角色关联表是中间表，使用多对多关联设计，用于关联用户与角色。

表3.10 sys\_log（系统日志表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| module | varchar | 50 | NO |  | 日志模块 |
| request\_method | varchar | 64 | NO |  | 请求方式 |
| ip | varchar | 45 | YES |  | IP地址 |
| execution\_time | bigint |  | YES |  | 执行时间（ms） |
| create\_time | datetime |  | YES |  | 创建时间 |

说明：日志表记录系统操作流水，支持通过execution\_time字段分析接口性能瓶颈。

表3.11 sys\_notice（通知公告表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| title | varchar | 50 | YES |  | 通知标题 |
| type | tinyint |  | NO |  | 通知类型（字典notice\_type） |
| publish\_status | tinyint |  | YES | 0 | 发布状态（0-未发布 1-已发布） |
| create\_time | datetime |  | NO |  | 创建时间 |

说明：通知公告表记录系统公告，支持多级发布状态管理和目标用户定向推送。

表3.12 leave\_application（请假申请表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint | 20 | 否 | - | 主键 |
| student\_id | bigint | 20 | 否 | - | 学生ID（关联user表） |
| type | tinyint | 1 | 否 | - | 请假类型（1-短期病假，2-长期事假，3-特殊事假，字典leave\_type） |
| start\_time | datetime | - | 否 | - | 开始时间 |
| end\_time | datetime | - | 否 | - | 结束时间 |
| days | int | 4 | 否 | - | 请假天数（自动计算） |
| reason | varchar | 500 | 否 | - | 请假事由（需填写详细说明） |
| attachment | varchar | 255 | 是 | NULL | 证明材料（存储文件路径，支持PDF/JPG格式） |
| status | tinyint | 1 | 是 | 0 | 审批状态（0-待审批，1-已批准，2-已拒绝，3-已撤回） |
| current\_approver | bigint | 20 | 是 | NULL | 当前审批人ID（关联user表，动态推进审批链） |
| create\_time | datetime | - | 否 | - | 创建时间 |
| update\_time | datetime | - | 是 | NULL | 最后更新时间 |

表3.13 leave\_approval（审批流程表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint | 20 | 否 | - | 主键 |
| leave\_id | bigint | 20 | 否 | - | 请假申请ID（关联leave\_application表） |
| approver\_id | bigint | 20 | 否 | - | 审批人ID（关联user表） |
| approval\_result | tinyint | 1 | 是 | 0 | 审批结果（0-待处理，1-同意，2-拒绝） |
| approval\_comment | varchar | 255 | 是 | NULL | 审批意见（驳回时必须填写） |
| approval\_time | datetime | - | 是 | NULL | 审批时间 |
| create\_time | datetime | - | 否 | - | 流程节点创建时间 |

# **4** 请假管理系统实现

## 4.1 系统目录结构

该请假管理系统主要为学生、辅导员、领导使用，借助前端界面来与系统实现交互，辅导员和相关领导则通过后端界面来与系统进行连接，从而实现对请假管理系统各个功能模块的管理。系统目录图如4.1图所示：

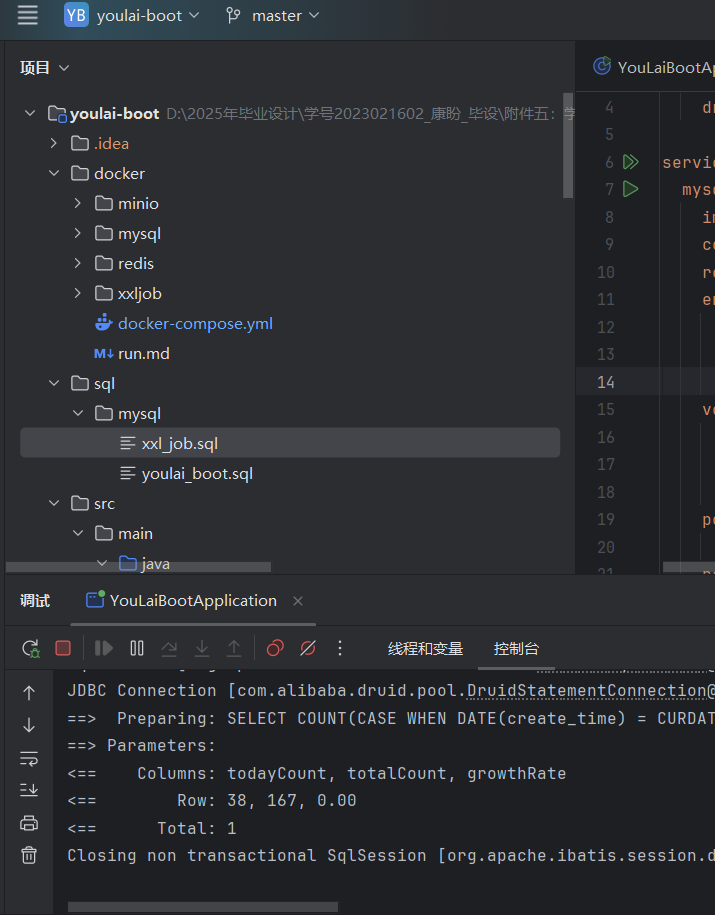


图4.1 系统目录图

## 4.2 系统前端功能模块的实现

4.2.1 前端的主界面模块

用户打开浏览器搜索网址http://localhost:3000/进入请假管理系统的首页，用户可以点击“登录”进入自己账号的系统首页。首页的设计布局方式基于开源项目youlaiboo，进入系统前端首页，可以查看到首页，系统管理，代码生成等功能模块，并可以进行详细的操作，在首页可以查看到公告。界面如图4.2所示：

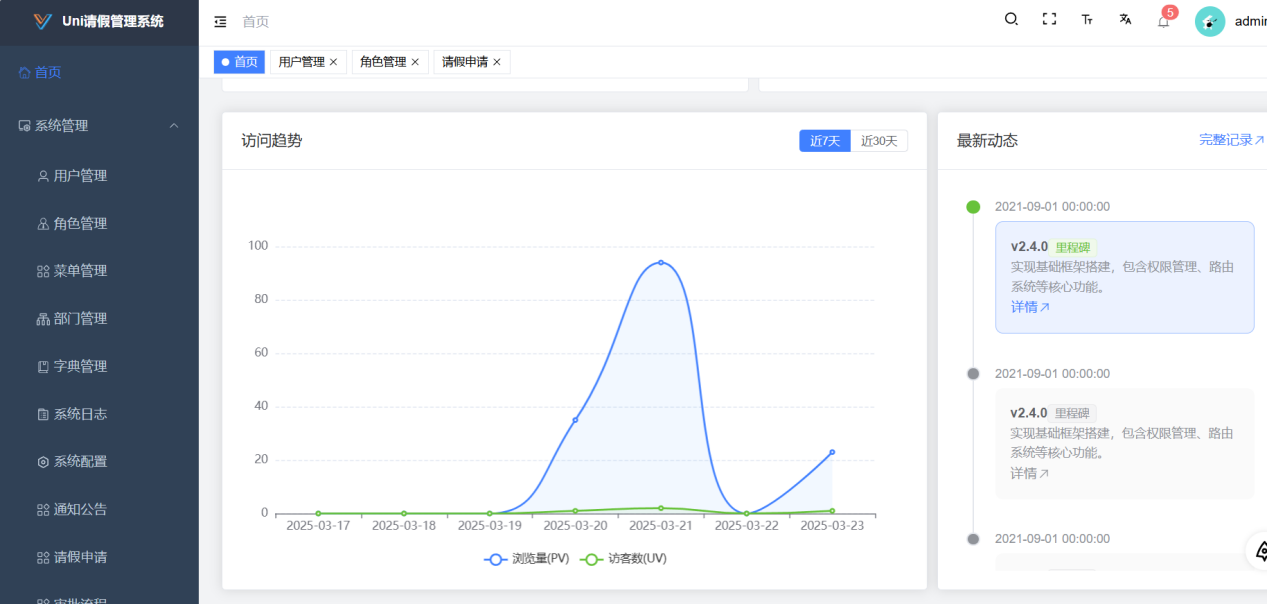


图4.2 系统前端首页

4.2.2 登录模块

当用户使用请假管理系统时，需要先登录本系统方可进行操作。如果用户尚未注册，必须先完成账号录入功能。在此过程中，用户需点击“注册”按钮，随后按照要求填写包括账号、密码、姓名、性别、联系方式、身份证号和邮箱等在内的个人信息。用户登录流程图如下图4.3所示，界面如图4.4、图4.5所示：

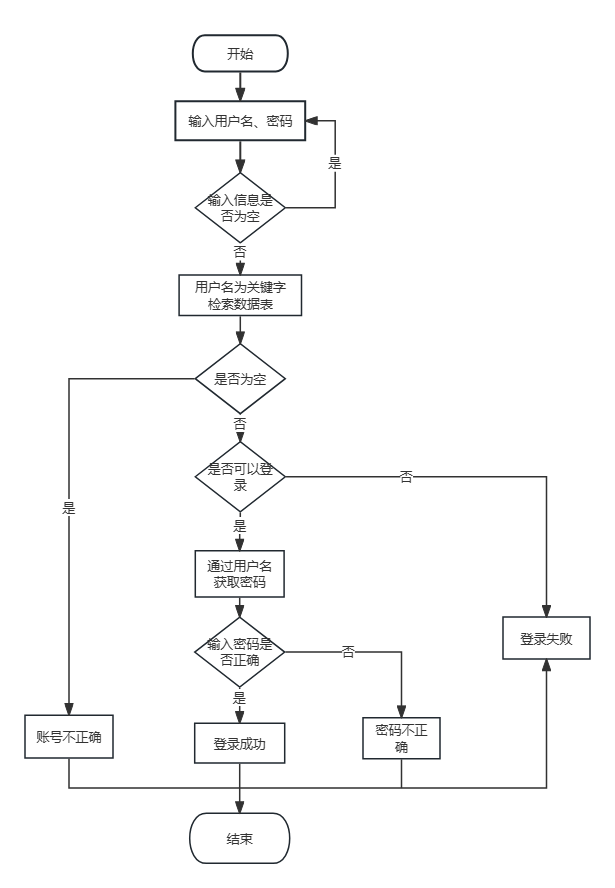


图4.3 用户登录流程图



图4.4 用户登录界面

4.2.3 管理员信息模块

经由RABC权限机制分配权限，管理员进行登录操作后，在UniApp前台首页的课程信息模块可以查看到系统所有在移动端前端的详细信息，包含首页轮播图、用户管理、角色管理等。管理员作为系统的最高权限拥有者，可以对系统所有可操作的内容进行操作。点击“用户管理”即可立即进入到用户管理界面，对系统用户进行操作。如图4.6、图4.7、图4.8所示：



图4.6 首页信息展示界面

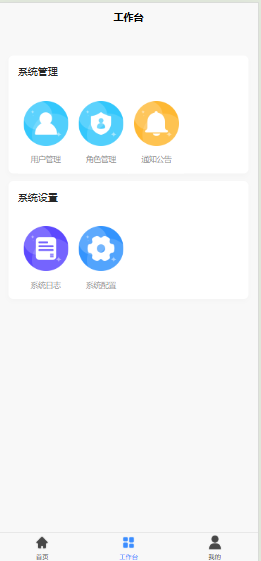


图4.7 系统管理相关

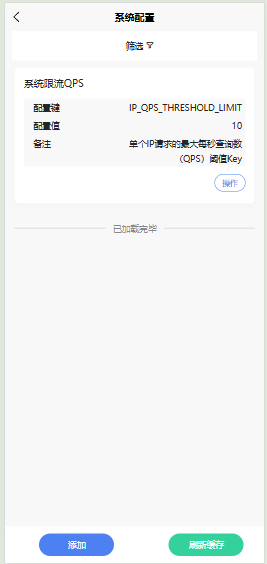


图4.8 系统设置相关

4.2.4 辅导员领导信息模块

经由RABC权限机制分配权限，辅导员和领导所属的角色，在前台首页中并没有管理员权限所拥有的用户管理等功能，但也可以查看到系统相关的用户数据，可以通过请假管理来对申请请假的同学进行“通过”或“拒绝”操作，用户可以点击学生的请假信息进行查看，根据所提交的信息，选择是否通过学生的请假。具体如图4.9、图4.10、4.11所示：



图4.9 辅导员领导展示界面



图4.10 辅导员领导批准界面

4.2.5 学生模块

经由RABC权限机制分配权限，所属学生角色权限的用户，在首页模块可以查看到对应的信息，并对该些功能模块进行详细的操作，同时，由于学生作为请假的主力单位，主要使用在于申请请假这个流程，用户可以填写请假表单使用应用进行提交请假单。具体如图4.12、4.13所示：

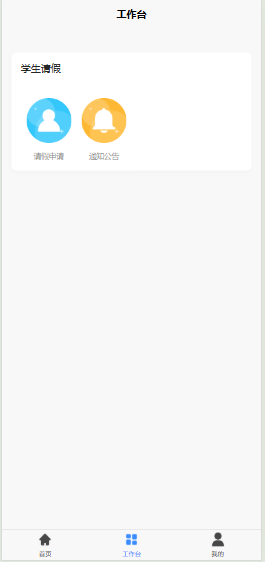


图4.12 学生详情信息

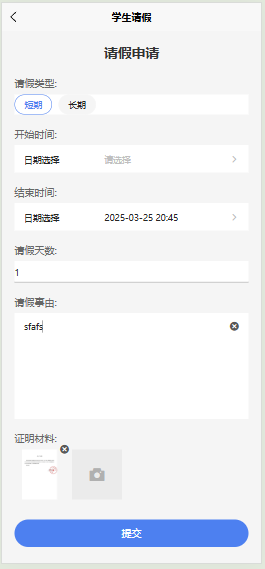


图4.13 订单信息详情

4.2.6 个人中心模块

个人中心模块，使用用户来对个人、应用的信息进行查看已经修改的地方，包含个人信息修改、请假记录查询、关于我们等一系列内容。具体如图4.14、图4.15所示：

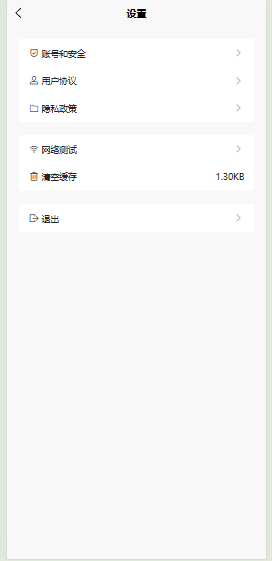


图4.14 交流论坛界面

## 4.3 系统后端功能模块的实现

4.3.1 登录模块

管理员登录后台模块进行详细的操作，主要是输入用户名和密码，选择登录身份进行登录操作。登录系统流程图如图4.17所示，相关界面展示如图4.18所示：

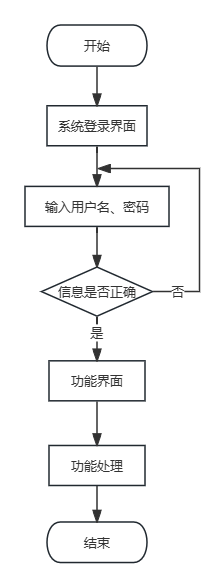


图4.17 管理员、管理员登录流程图



图4.18 登录界面

4.3.2 用户管理功能模块

管理员在用户管理模块可以查看到系统用户所有的姓名，头像，性别，联系方式，等信息，可以对会员信息进行查看详情，修改，删除，重置密码，在该模块可以新增一位用户到系统中来。

点击修改按钮，可以对用户的性别，手机号码等信息进行修改。如图4.19、图4.20所示：



图4.19 用户管理



图4.20 修改用户信息界面

4.3.3 角色管理模块

管理员在角色管理模块可以对系统的所有角色详情进行操作，该系统基于RABC权限管理系统，管理员可以为不同的角色设置不同的权限。如下图4.21所示：

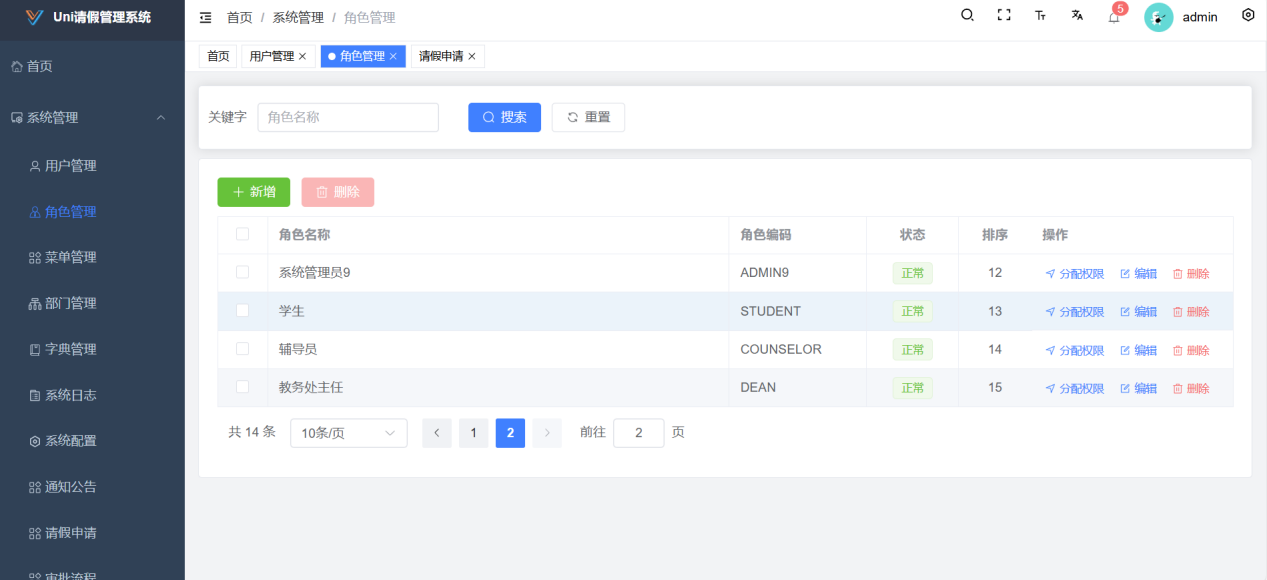


图4.21 角色管理界面

管理员在角色权限分配模块可以对系统所有角色进行查看、操作。如下图4.22所示：

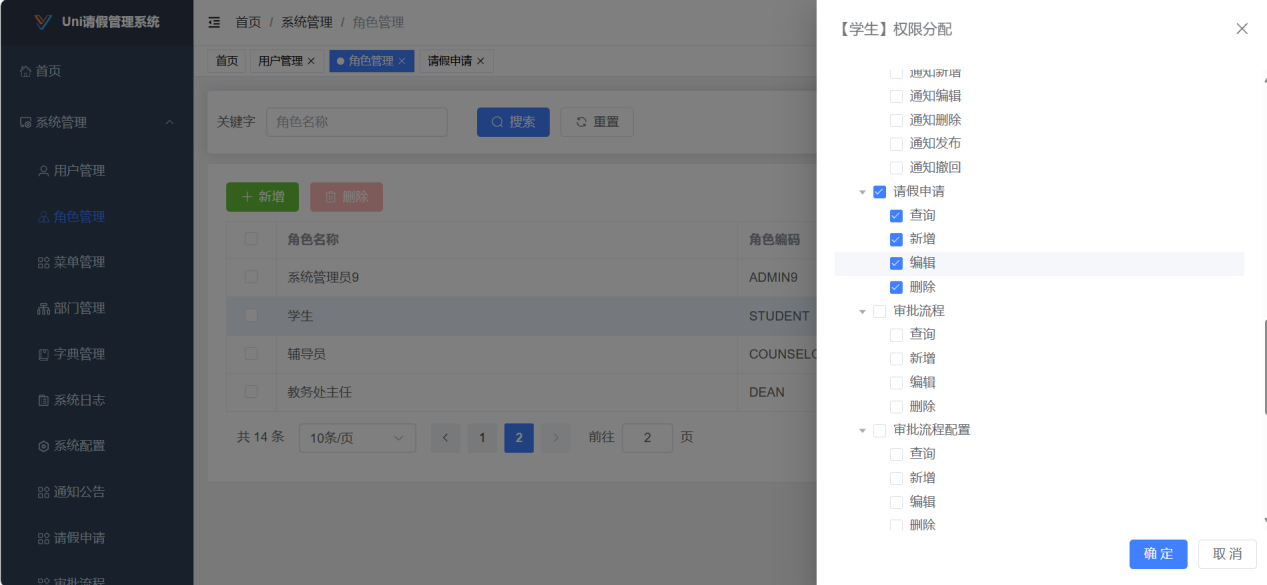


图4.22 角色权限分配界面

4.3.4 部门管理模块

管理员在部门管理模块可以对系统所有部门详细信息进行修改，删除和新增的操作。界面如图4.23所示：



图4.23 部门管理

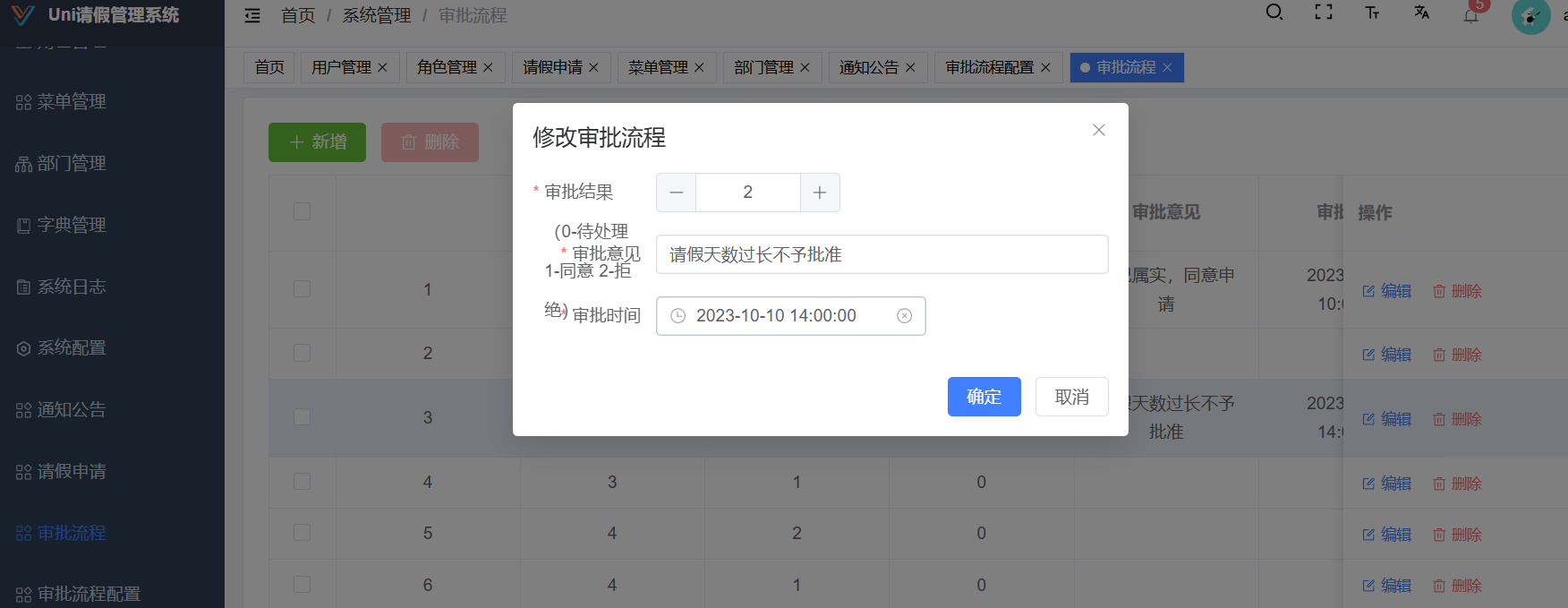
4.3.5 审批流程管理模块

管理员在审批流程信息管理列表可以查看到系统所有的审批详情，主要包括请假人、审批人、审批结果、审批意见等信息，可以对对应信息进行修改、删除、新增和查询的操作，在该模块可以对审批流程进行有效的修改。在Uniapp的移动端界面可以查看到请假后审批的评价。界面如图4.24所示：



图4.24 审批流程信息管理

管理员在该模块可以查看到修改审批流程。具体实现界面如图4.25所示：

图4.25 审批处理

审批流程自动配置管理，基于自动化操作，可以让不同的权限的用户自动分配不同的审批对象，比如特定班级的学生请假，自动为该班级安排指定的辅导员进行请假审批操作。界面如图4.26所示：



图4.26 审批流程自动配置管理

管理员可以修改审批流程的信息。界面如图4.27所示：



图4.27 修改审批流程配置

4.3.6 公告信息管理模块

管理员在公告信息模块可以对系统展示在前台界面的公告详情，公告类型，公告图片和公告时间进行查看详情，修改，删除和新增的操作。具体实现界面如图4.29所示：



图4.29 管理员公告管理

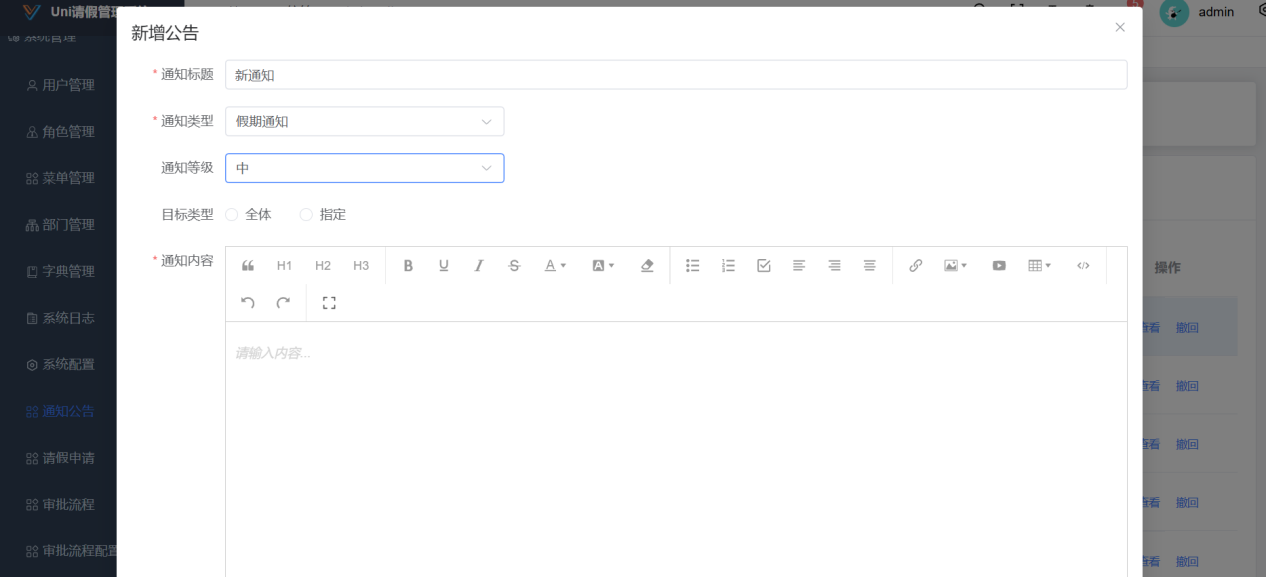
管理员在公告信息模块可以发布系统展示在前台界面的公告详情，界面如图4.30所示： 

图4.30 管理员公告发布

4.3.7 请假单管理模块

管理员在请假单管理模块可以查看到系统展示在前台的所有请假单详情主要包括学生ID、请假类型、开始时间、结束时间等请假单信息，可以对该信息进行修改、删除、新增的操作，在该界面可以对请假单进行修改的操作。

管理员在该模块可以查看到指定用户在请假单当前的状态，并且可以进行修改，也支持多种查询方式，方便管理员查找指定用户的请假记录。具体实现界面如图4.31、4.32所示：



图4.31 请假单管理



图4.32 请假单修改

4.3.8 基础字典数据管理模块

管理员在基础数据管理模块可以管理系统展示在前台的字典数据。让高校运营的方向更明确，具体实现界面如图4.33所示：

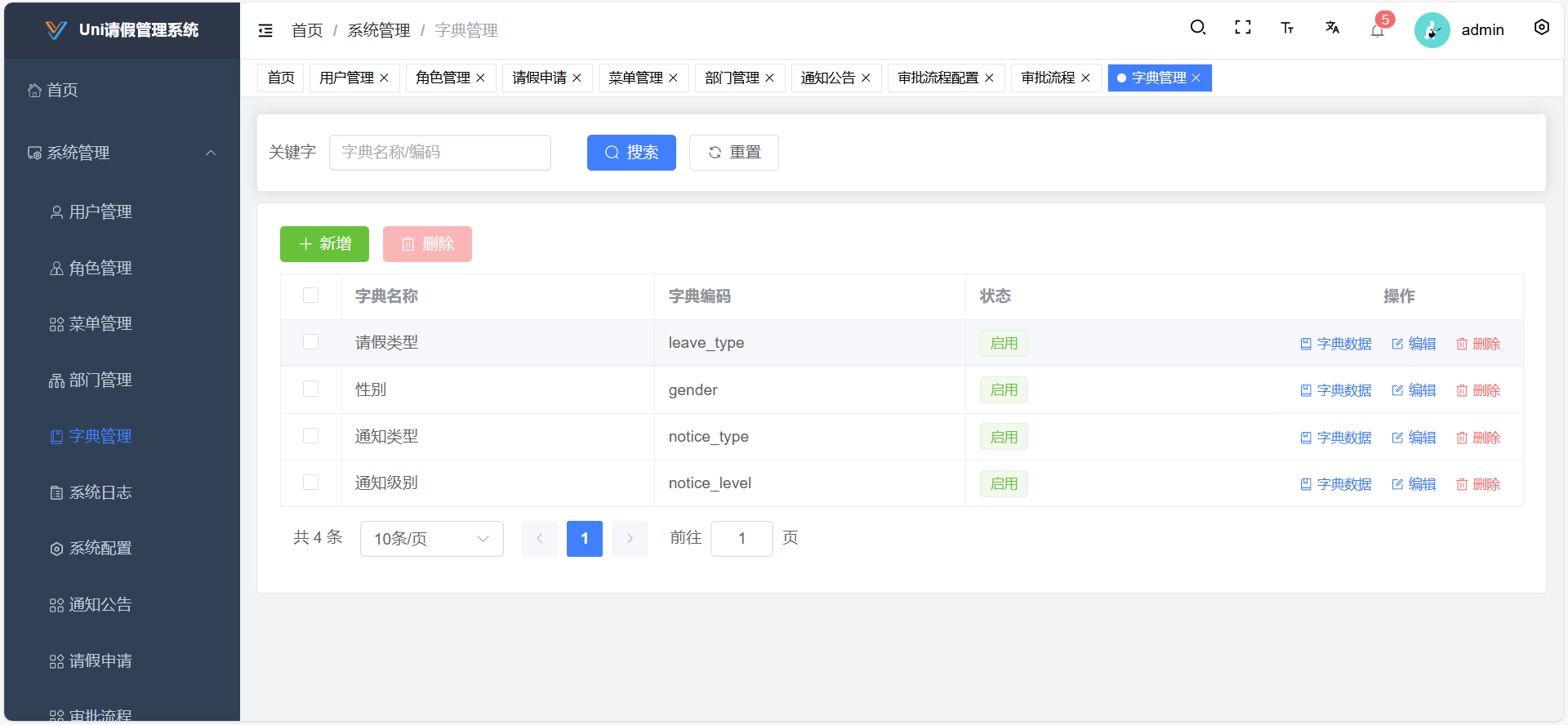


图4.33 基础字典数据管理

4.3.9 系统日志管理模块

管理员在日志管理模块可以管理系统展示在前台的日志。可以让管理员快速直观地了解，并且快速定位到具体问题，具体实现界面如图4.34所示：



图4.34 系统日志管理

# 5 系统测试

当前计算机系统中的软件测试是一个不可或缺的步骤，确保其正常运行是将此类系统推向实用市场的重要环节。经过开发者的充分设计与详细研究后，任何系统在正式投入使用前，均需经历专业的系统测试环节。这一环节不仅是对系统是否满足既定功能规范的验证，更是对系统设计潜在问题的有效识别和优化过程。测试系统的方法很多，本系统主要采用黑盒测试来检查系统的功能和性能表现。

## 5.1 测试的原理

在软件开发中，测试是确保程序正常执行的关键环节。系统的开发目标在于稳定、高效地提供优质服务，确保其在实际应用中不出故障。每个软件在初期阶段都可能存在各种问题，所以需要通过多种测试手段来精准识别系统中的缺陷并逐一解决，以构建出既高效又稳定的系统。这一过程是软件开发中不可或缺的一环。在进行测试时必须遵循既定的规则与准则。这些规则的遵循往往依赖于测试用例的具体要求，而测试用例则是根据详尽的需求规格或说明书精心设计的。本论文选择已实现的功能作为测试的核心要点，从而展开针对性的测试工作。

本论文基于实际的应用场景，选定了以功能为主导的黑盒测试法。此方法适用于系统功能明确的情况下，来验证各功能是否严格遵循需求分析中的定义正常运行。在软件测试中，黑盒测试将程序视为一个封闭系统，无需深入其内部机制。其核心目标在于检验系统的功能需求实现情况，从而发现并修复可能存在的功能缺陷、界面瑕疵以及性能问题等。因此它也常被称为功能验证测试。

## 5.2 系统测试用例

5.2.1 用户注册功能测试

用户录入注册信息后，点击“注册”会出现两种情况：注册成功跳转登录界面或提示请求填写正确的信息。注册功能测试用例表如表5.1所示，注册功能测试结果图如图5.1所示。

表5.1用户注册功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 输入不正确的手机号 | 未注册成功，提示信息为“请重新输入正确的手机号” | 与预期结果相同 | 通过 |
| 2 | 两次输入的密码不一致 | 注册失败，提示信息为“密码的俩次密码不一致” | 与预期结果相同 | 通过 |
| 3 | 输入正确用户名和密码、姓名、手机号、邮箱 | 注册成功，页面跳转到登录界面 | 与预期结果相同 | 通过 |



图5.1　注册功能测试结果图

5.2.2 用户登录功能测试

用户输入登录账号，密码，点击登录按钮，会出现两种情况：登录成功跳转请假管理系统首页或提示账号或密码不正确。登录功能测试用例表如表5.2所示，登录功能测试结果图如图5.2所示。

表5.2用户登录功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 输入正确用户名和密码 | 登录成功，页面跳转到系统首页 | 与预期结果相同 | 通过 |
| 2 | 输入不正确的密码 | 未登录成功，页面不跳转，提示信息为“账号或密码不正确” | 与预期结果相同 | 通过 |
| 3 | 输入不存在的用户名 | 未登录成功，页面不跳转，提示信息为“账号或密码不正确” | 与预期结果相同 | 通过 |

图5.2　登录功能测试结果图

5.2.3 个人中心功能测试

用户在个人信息管理界面可以对个人资料进行查看修改操作，主要包括用户名、账号、头像、手机号、身份证号、邮箱等详细信息进行修改，查看自己的课程订单和请假单兑换记录，对购买的课程进行评价或退款。个人中心功能测试用例表如5.3所示，个人中心功能测试结果图如图5.3所示。

表5.3个人中心功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 修改用户名，点击更新信息 | 显示更新后用户名，提示信息为“修改成功” | 与预期结果相同 | 通过 |
| 2 | 进入课程订单，选择课程点击评价，输入评价并提交 | 返回课程订单页面，订单类型由已支付更新为已评价 | 与预期结果相同 | 通过 |
| 3 | 进入课程订单，选择课程点击退款 | 弹出选项框“确认退款或取消”，确认后返回课程订单页面，订单类型由已支付更新为已退款 | 与预期结果相同 | 通过 |

图5.3　个人中心功能测试结果图

5.2.4 课程信息管理功能测试

在课程信息管理界面中，管理员能够看到系统中所有课程的信息以一个列表的形式展现出来，并且还可以对课程进行信息的修改，课程信息管理测试用例表如表5.4所示，课程信息管理测试结果图如图5.4所示。

表5.4 课程信息管理模块功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 在管理员页面点击课程信息管理 | 跳转到课程信息管理页面，以分页列表形式显示所有课程信息 | 跳转到课程信息管理页面，以分页列表形式显示所有课程信息 | 通过 |
| 2 | 在课程信息管理页面点击新增，在添加课程信息页面输入课程信息信息后点击提交 | 成功添加课程信息，跳转到课程信息管理页面并在课程信息列表中显示新增的课程信息 | 课程信息添加成功，跳转到课程信息管理页面并在课程信息列表中显示新增的课程信息 | 通过 |
| 3 | 在课程信息列表中点击修改，在课程信息信息页面修改完数据后单击提交 | 成功修改课程信息信 息，提示信息为“操作成功”，跳转至课程信息管理页面 | 成功修改课程信息信息，提示信息为“操作成功”，跳转至课程信息管理页面 | 通过 |
| 4 | 在课程信息列表中点击删除 | 成功删除课程信息，并刷新所有课程信息页面 | 成功删除课程信息，并刷新所有课程信息页面 | 通过 |

图5.4　课程信息管理功能测试结果图

5.2.5 用户管理功能测试

在用户管理界面，系统内的所有用户信息均以列表形式清晰呈现。可以对用户进行信息修改、新增用户或移除用户以及重置密码，确保了用户管理的全面性和灵活性。用户管理测试用例表如表5.5所示。用户管理测试结果图如图5.5所示。

表5.5 用户管理模块功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 在管理员登录页面输入管理员账户和密码，点击登录 | 跳转到用户管理页面，以分页列表形式显示所有用户 | 跳转成功，与预期结果相同 | 通过 |
| 2 | 在用户列表中单击重置密码 | 密码重置成功，提示信息为“重置成功，密码已重置为123456” | 成功重置密码，与预期结果相同 | 通过 |
| 3 | 在用户列表中单击修改，在用户信息页面修改完数据后点击提交 | 成功修改用户信息，提示信息为“操作成功”，更新用户信息列表 | 成功更新用户信息，与预期结果相同 | 通过 |
| 4 | 在用户管理页面点击“新增”，在添加用户页面输入用户信息后单击提交 | 成功添加用户，跳转到用户管理页面并在用户列表中显示新增的用户信息 | 成功添加用户信息，与预期结果相同 | 通过 |

图5.5　用户管理功能测试结果图

## 5.3 测试结果分析

经过以上测试后对请假管理系统的运行性能进行了分析。系统表现良好，不仅支持后台数据的增删改查操作，而且大部分功能均已成功实现。还具有一定的安全性保障，符合设计的各项要求。

## 5.4 本章小结

系统测试是完善系统必不可少的环节，采用常规的黑盒测试对系统各个功能模块进行了测试，不仅设计了多个测试用例，还在测试过程中严格要求系统是否符合需求。经过不断的测试与调整，使系统达到需求条件。