

AIGC检测 · 全文报告单

NO:CNKIAIGC2025FJ_202505102016583

检测时间: 2025-05-10 23:03:59

篇名: 基于SpringBoot与Vue的自习室管理系统的设计与实现

作者: 林少煌

单位: 厦门理工学院

文件名: 基于SpringBoot与Vue的自习室管理系统的设计与实现.pdf

全文检测结果



AI特征值: 16.1%

AI特征字符数: 4609

总字符数: 28632

- AI特征显著 (计入AI特征字符数)
- AI特征疑似 (未计入AI特征字符数)
- 未标识部分

AIGC片段分布图

前部20%

AI特征值: 62.6%

AI特征字符数: 3583

中部60%

AI特征值: 2.1%

AI特征字符数: 361

后部20%

AI特征值: 11.6%

AI特征字符数: 665



分段检测结果

序号	AI特征值	AI特征字符数 / 章节(部分)字符数	章节(部分)名称
1	37.2%	3803 / 10225	基于SpringBoot与Vue的自习室管理系统的设计与实现_第1部分
2	0.0%	0 / 10755	基于SpringBoot与Vue的自习室管理系统的设计与实现_第2部分
3	10.5%	806 / 7652	基于SpringBoot与Vue的自习室管理系统的设计与实现_第3部分

1. 基于SpringBoot与Vue的自习室管理系统的设计与实现_第1部分

AI特征值: 37.2%

AI特征字符数 / 章节(部分)字符数: 3803 / 10225

片段指标列表

序号	片段名称	字符数	AI特征		
1	片段1	2815	显著		27.5%
2	片段2	234	显著		2.3%
3	片段3	255	疑似		2.5%
4	片段4	467	疑似		4.6%
5	片段5	534	显著		5.2%
6	片段6	220	显著		2.2%
7	片段7	436	疑似		4.3%
8	片段8	480	疑似		4.7%

原文内容

基于Spring Boot与Vue的自习室管理系统的设计与实现

摘要

随着教育资源的日益丰富和学习需求的多样化，自习室作为重要的学习场所，其管理效率和用户体验成为了提升学习效率的关键因素。传统自习室管理方式存在诸多不便，严重制约了自习室的利用率和学习者的学习体验。本文通过引入Spring Boot后端框架和Vue 前端框架，可以实现更加智能化的用户体验。Spring Boot以其简洁、高效、易于集成的特性，成为构建后端服务的理想选择，能够迅速搭建稳定可靠的服务端架构。而Vue作为现代前端框架，以其数据驱动、组件化的设计理念，能够为用户提供流畅、响应迅速的用户界面体验。两者的结合，不仅提升了系统的开发效率，也确保了系统的高可用性和可扩展性。在自习室管理系统的架构设计中，充分利用了Spring Boot和Vue的优势，构建了一个高可用、可扩展的系统架构。

关键词：Vue；SpringBoot；自习室管理；资源合理分配的应用研究

The Design and implementation of Self-study Room Management System Based On Spring Boot and Vue

27.5%(2815)

Abstract

With the increasing abundance of educational resources and the diversification of learning demands, self-study rooms, as important learning spaces, have seen their management efficiency and user experience become key factors in enhancing learning effectiveness. Traditional self-study room management methods are fraught with numerous inconveniences, severely restricting the utilization rate of self-study rooms and the learning experience of learners. This paper introduces the Spring Boot backend framework and the Vue frontend framework to achieve a more intelligent user experience. Spring Boot, with its simplicity, efficiency, and ease of integration, is an ideal choice for building backend services, capable of rapidly establishing a stable and reliable server-side architecture. Meanwhile, Vue, as a modern frontend framework, with its data-driven and component-based design philosophy, can provide users with a smooth and responsive user interface experience. The combination of the two not only boosts the development efficiency of the system but also ensures its high availability and scalability. In the architectural design of the self-study room management system, the advantages of Spring Boot and Vue are fully utilized to construct a highly available and scalable system architecture.

Keywords: Vue; Spring Boot; user experience; system architecture design; applied research on resource allocation

1 前言

1.1 项目背景

随着社会发展，教育资源的优化配置与高效利用成为了提升教育质量与学习体验的关键所在。自习室，作为校园内外重要的学习空间，其管理效率与服务质量直接影响到学习者的学习效率与满意度。然而，传统自习室管理模式往往面临座位信息不透明、预约流程繁琐、资源分配不均等问题[1-2]，这些问题不仅限制了自习室的利用率，也影响了学习者的学习体验与积极性。利用现代信息技术手段，构建智能化、信息化的自习室管理系统，已成为解决传统管理痛点、提升自习室管理效率与用户体验的重要途径。

1.2 系统开发意义

针对自习室管理系统的现代化架构设计：采用Spring Boot与Vue这一前沿技术组合，旨在设计一个高效、可扩展且用户友好的自习室管理系统。通过Spring Boot构建后端服务，利用其强大的依赖注入、自动配置和嵌入式服务器等功能，实现后端服务的快速开发和部署。同时，Vue作为前端框架，提供了丰富的组件库和响

应式设计能力，使得前端界面更加美观且易于交互。这一架构设计不仅提升了系统的开发效率，还确保了系统的稳定性和可扩展性。

解决自习室管理系统的复杂业务需求：自习室管理系统需要处理用户注册登录、自习室信息查询、预订管理、通知公告发布等一系列复杂业务需求。通过深入分析这些需求，利用SpringBoot与Vue的技术优势，实现了功能的模块化划分和前后端分离的开发模式。这种设计方式不仅提高了系统的可维护性，还使得各个功能模块能够独立开发和迭代，从而更灵活地应对业务变化。

提升自习室管理系统的性能和用户体验：针对自习室管理系统可能面临的高并发访问和数据量大的问题，引入了Spring Cloud的组件（如Nacos、Sentinel等）来优化系统性能。Nacos作为服务注册与配置中心，能够动态管理服务的注册与发现，提高系统的可扩展性；Sentinel则提供了限流熔断与降级功能，确保系统在高并发场景下仍能稳定运行。此外，还通过优化数据库查询、使用缓存等技术手段，进一步提升系统的响应速度和用户体验。

1.3 国内外研究现状

1.3.1国内研究现状

近年来，随着国内升学和就业压力的增大，自主学习的人越来越多，相比与错综复杂的学习环境，越来越多的人更喜欢去更有学习氛围的自习室，更向往安静的学习环境。何佳，张越等人指出，传统自习室无法满足日益增长的市场需求，迫切需要一种多元化模式为一体的学习环境，来迎合不同群体的纷繁需求，共享自习室孕育而生[3]。柏锦诗琪,邹俊,刘敏娜针对共享自习室的运营方提供座位预约和分配、费用管理、数据监测与安全、运营数据分析等[4]，且为了满足不同用户群体的需求，国内自习室管理系统还在功能上不断进行完善。

魏文璿基于用户体验视角的高校图书馆学习空间价值进行评估研究，指出信息技术的飞速发展、教学模式和学习方式的巨大转变引发了高校图书馆空间再造的热潮，引用了用户体验的视角，将高校图书馆的空间价值研究置于用户学习体验的层面展开[5]，尽管国内自习室管理系统在市场需求、技术发展和功能完善等方面取得了显著进展，但是随着自习室的快速发展，越来越多的企业和个人开始进入这一领域，导致市场竞争日益激烈。为了保持竞争优势，自习室管理系统需要不断创新和升级功能，提升用户体验和服务质量。

2.3%(234)

1.3.2国外研究现状

本文针对这一问题查阅了相关文献资料，发现国外研究主要集中在ai深度学习等方面，旨在打造出一个智能自习室，Perera Damith等人探讨了俄克拉荷马州立大学（OSU）图书馆自习室预订的转变。通过开发了一个自定义的开源用户界面来简化预订效率，事实证明这些改进有助于证明定制解决方案在满足用户需求和为所有利益相关者改进系统功能方面的有效性[61]。日本的大川航平等人通过深度学习的

2.5%(255)

方法，通过面部表情来估计学习者的困惑状态，结合了认知学徒理论和困惑估计方法，实现教育型自习室[61]。

1.4 论文组织结构

第一部分，前言，介绍了课题的研究背景、本论文的研究意义和目标等。

第二部分，开发环境，介绍了与本系统实现相关的技术和工具，包括常用的VUE前端框架、Spring Boot和MyBatis三大开源框架、MySQL等。

第三部分，系统分析，主要是系统流程设计和系统用例设计及法律、经济等可行性分析。

第四部分，系统设计，依照系统设计规范，绘制各个的流程图和用例图等。

第五部分，界面设计与功能实现，主要是展示系统具体功能和系统界面开发结果。

第六部分，系统测试，对完成的系统功能进行黑盒测试，编写并列举了相关测试用例表。

第七部分，分析与总结，对完成的系统功能进行相关总结，并说明了完成此次系统过程的心得体会等。

2 开发工具与技术

2.1 VUE

VUE是目前市场最主流的前端框架之一，用于构建用户界面的JavaScript框架。它基于标准HTML、CSS和JavaScript构建，并提供了一套声明式的、组件化的编程模型，帮助你高效地开发用户界面。无论是简单还是复杂的界面，Vue都可以胜任。

2.2Element UI

Element UI是一套基于Vue.

js的桌面端组件库，由饿了么前端团队开发并开源。Element UI提供了丰富的组件，使得前端开发者能够轻松地创建漂亮且功能强大的用户界面。

2.3Spring Boot

Spring Boot是由Pivotal团队提供的一套开源框架，可以简化spring应用的创建及部署。它提供了丰富的Spring模块化支持，可以帮助开发者更轻松快捷地构建出企业级应用。Spring Boot通过自动配置功能，降低了复杂性，同时支持基于JVM的多种开源框架，可以缩短开发时间，使开发更加简单和高效。

2.4MyBatis-Plus

MyBatis-Plus是一个基于MyBatis的增强工具，旨在简化开发、提高效率。它在MyBatis的基础上进行增强，但不改变其原有功能。MyBatis-Plus可以让开发者在不编写SQL语句的情况下，快速完成单表的操作。

2.5Uniapp

UniApp是基于Vue.js框架的一个开源前端开发框架，它允许开发者编写一套代

4.6%(467)

5.2%(534)

码，然后部署到iOS、Android、Web（响应式）以及各种小程序和快应用等多个平台。UniApp 的核心优势在于它的跨平台能力，开发者无需为每个平台编写不同的代码，从而节省了大量的开发时间和成本。

2.6MySQL

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性，是全球广受欢迎的开源数据库。

2.7 本章小结

本章概述了开发此系统时使用的一些技术、框架和开发时所用的一些可视化工具。

3 系统分析

3.1 可行性分析

自习室管理系统旨在实现校园自习空间资源的智能化管理与服务。系统开发前需从技术实现、社会效益及经济成本三个维度进行可行性论证。

3.1.1 技术可行性

本系统采用SpringBoot+MyBatisPlus构建后端服务框架，配合Vue3+ElementPlus实现管理员前端交互界面，Uniapp为用户前端交互界面。SpringBoot的自动装配机制简化了项目配置，Vue的组件化开发模式有利于功能模块复用，Uniapp实现用户的便捷操作。数据库选用MySQL8.0保障事务处理能力，整体技术架构成熟稳定，满足系统开发需求。

3.1.2 社会可行性

当前高校普遍存在自习座位资源紧张、占座现象严重等问题。本系统通过实时座位状态监控、预约时段管理等手段，可有效提升空间资源利用率。系统内置积分机制能规范用户行为，减少违规占座现象。电子化预约方式替代传统人工登记，既提高管理效率又避免人员聚集，符合智慧校园建设的发展趋势。

3.1.3 经济可行性

采用开源技术框架避免商业授权费用，后期维护可通过学生技术团队承接降低人力成本。系统实施后预计可使自习室日均使用人次将会提升，管理人员工作量相应减少，具有显著的成本效益优势。

3.2需求分析

3.2.1 系统场景描述

系统需支持用户实时查看各楼层自习室空位分布、完成在线预约/取消操作、可签到/签离、超时未签到自动释放座位。管理员可动态调整开放区域、设置特殊时段（如考试周延长开放）、对后台进行权限管理。系统需提供座位使用统计报表，并设置黑名单限制违规用户预约权限。

3.2.2 系统角色功能分析

本系统的用户分类有三类，分别是用户，系统管理员，次级管理员。普通用户可以查看座位空闲情况、预约/取消座位、查看预约记录、修改个人信息等，系统管理员具备系统最高控制权的超级用户，可以权限动态配置，将权限下发给次级管理员方便进行管理。次级管理员具有系统管理员下发的权限，例如用户管理、自习室管理、公告发布、失物招领管理等。

3.3 系统用例建模

用例建模指的是找到参与者和用例，通过不同的功能编写出不同的用例规约、用例描述等。系统分为管理员和用户。管理员包括系统管理员和次级管理员。具体系统用例图如图3.1所示。

图3.1 用户模块用例图

管理员登录后台后可以对其他进行操作，例如对用户进行增删改查等操作。管理员用例图如图3.2所示。

图3.2 管理员用例图

次级管理员与管理员不同，次级管理员只能使用系统管理员下发的权限，实际功能由系统管理员分配，次级管理员的功能被包含在系统管理员功能内。

3.4 系统主要用例规约

本小节主要介绍了此系统的主要用例规约。

(1) 角色管理用例规约如表3.1所示

表3.1 角色管理用例规约
用例编号：01
用例名称：角色管理
用例描述：系统管理员创建次级管理员账户并分配细粒度权限，被授权者仅可见被分配的功能菜单或功能按钮。
执行者：系统管理员
前置条件：管理员通过双因素认证登录系统且具有超级权限。
后置条件：权限变更记录写入审计日志，新账户即时生效。
基本流程：1. 进入角色管理模块。
2. 管理员对角色数据执行查询，查询到对应的数据。执行新增、编辑、删除，成功后数
据库得到更新。
2. 选择角色进行点击“分配菜单”。
3. 勾选可操作模块。
4. 生成带权限标签的账户。
业务规则：权限分配遵循最小化原则，本用例仅为系统管理员使用。

(2) 自习室管理用例规约如表3.2所示。

表3.2 自习室管理用例规约

用例编号：02

用例名称：自习室管理

用例描述：管理员管理自习室信息，包括座位布局图更新、设施状态标记。

执行者：系统管理员/次级管理员（需具备权限）

前置条件：操作者拥有空间管理权限且通过身份验证。

后置条件：更新后的座位同步至所有终端设备。

基本流程：1. 上传座位布局

2. 标注特殊座位

3. 设置区域开放时段

备选流程：检测到自习室ID冲突时修改失败。

业务规则：本用例为具备权限的用户使用。

（3）失物招领管理用例规约如表3.3所示。

表3.3 失物招领管理用例规约

用例编号：03

用例名称：失物招领管理

用例描述：管理员进行对失物的发布。

执行者：系统管理员/次级管理员（需具备权限）

前置条件：操作者拥有权限且通过身份验证。

后置条件：完成处置的记录同步至数据库。

基本流程：1. 管理员成功登入系统

2. 管理员点击失物招领管理功能模块

3. 管理员进行添加，选择失物或招领并编辑相应信息进行发布

4. 管理员对认领物品或找回物品进行状态更新

业务规则：本用例为具备权限的用户使用。

（4）预约座位用例规约如表3.4所示。

表3.4 预约座位用例规约

用例编号：04

用例名称：预约座位

用例描述：用户选择目标座位及使用时段，系统实施冲突检测。

执行者：普通用户

前置条件：用户为系统中存在的系统用户且成功登录系统。

后置条件：用户进行座位预约操作之后，系统保留其结果。

表3.4 预约座位用例规约（续）

基本流程：1. 查看实时座位空闲图

2. 选择预约时段

3. 进行预约确认

备选流程：检测到重复预约自动拦截并提示冲突时段。

业务规则：本用例为已进行登录的用户使用。

（5）座位签到用例规约如表3.5所示。

表3.5 座位签到用例规约

用例编号：05

用例名称：座位签到

用例描述：用户对已预约座位进行签到。

执行者：普通用户

前置条件：用户为系统中存在的系统用户且成功登录系统

后置条件：完成处置的记录同步至数据库

基本流程：1. 用户成功登入系统

2. 用户点击我的预约

3. 用户对已进行预约座位执行查询，查询到对应的数据。可选择签到或取消预约操作，

执行成功后数据库得到更新。

业务规则：本用例为已进行登录的用户使用

3.5 本章小结

本章通过多维度可行性论证确认系统实施价值，采用用例分析方法梳理核心业务流程，建立领域模型明确系统数据结构。特别针对高校自习场景设计信用管理体系与智能调度算法，为后续系统设计与实现奠定需求基础。

4 系统设计

4.1系统总体结构设计

本系统采用B/S架构，基于前后端分离设计模式。前端采用Vue3+Element Plus实现管理员交互界面，uniapp开发工具实现用户交互界面。后端运用Spring Boot构建RESTful API服务，MyBatis Plus实现数据访问以及MySQL进行数据存储。

4.2系统功能设计

本系统基于需求分析阶段的用户业务场景，采用模块化设计方法，将功能划分为用户端与管理端两大子系统。

1，管理端提供的功能应包含对系统的全面管理，包括系统管理，班级管理，学生管理，公告自习室管理以及数据统计等模块，使管理员可以更加有效管理和监控整个系统的运作。

2，用户端首先需要提供管理账户的功能，包括登录、注册、个人信息管理等操作。其次需要使用自习室相关功能的使用，包括自习室的查看、预约、签到以及反馈等功能。

综上所述，应形成一个具有完善功能模块、可持续发展的自习室管理系统，为

2. 2%(220)

用户提供便捷高效的自习室预约服务，帮助用户更好的使用自习室，同时为管理员提供全面监控和管理系统运作的功能。

4.3系统主要活动图设计

1. 角色管理活动图根据用例规约表3.1设计，管理员用户通过在角色管理功能模块中对角色信息进行相应操作，管理员在系统中可以对系统内的角色信息进行全面管理，包含新增角色、编辑角色、查询角色、删除角色，以及分配菜单功能，可以对用户进行权限分配角色管理活动图如图4.1所示。

管理员点击角色管理进行CRUD 跳转到CRUD界面跳转到角色管理界面进行分配菜单操作跳转到角色管理界面操作是否正确是执行操作角色管理否提示错误信息生成带标签的账户

图4.1角色管理活动图

2. 自习室管理活动图根据用例规约表3.2设计，管理员用户通过在自习室管理功能模块中对自习室信息进行相应操作，新增自习室时需完整填写名称、类型、楼层等元数据并上传封面图片，系统校验数据有效性后持久化存储；编辑功能支持对自习室基础信息及图片的更新，特别强化类型与楼层的联动校验；查询模块提供名称关键字及类型组合检索， 返回结果按楼层自动排序。自习室管理活动图如图4.2所示。

管理员自习室管理模块管理员已登录点击自习室管理添加自习室修改自习室删除自习室查询自习室填写自习室信息修改自习室信息选择自习室输入查询条件上传照片修改照片确认删除显示查询结果提交信息是否删除成功是否提交成功删除成功显示失败信息提交成功显示失败信息

图4.2自习室管理活动图

3. 失物招领管理活动图根据用例规约表3.3设计，管理员可执行对失物招领物品的全生命周期管理：新增时系统强制校验必填字段，确保数据完整性；编辑功能支持对既有物品信息的动态调整；对已找回或已认领物品可进行状态确认，更新物品信息。失物招领管理活动图如图4.3所示。

管理员失物招领模块管理员已登录进入失物招领管理已丢失物品状态待招领登记失物信息登记招领信息物品状态已找回已招领更新为已找回状态更新为已招领状态

图4.3失物招领管理活动图

4. 预约座位活动图根据用例规约表3.4设计，用户查找自习室信息，选择相应的座位和时间段进行预约。预约座位活动图如图4.4所示。

管理员预约模块进入相应界面选择座位选择时间段检测时间段合法性合法非法确认座位提示错误信息用户是否已已预约预约座位未预约显示已预约座位信息检测座位状态占用空闲更新座位状态为已预约提示座位已被占用

图4.4预约座位活动图

5. 座位签到活动图基于用例规约表3.5设计，支持用户对已预约座位进行签到操作。当用户未签到时，可对座位进行取消预约操作，成功签到后可进行签退操作。座位签到活动图如图4.5所示。

用户预约管理用户已登录点击我的预约用户是否已预约座位是签到显示已预约座位列表签到选择操作签到是否成功是显示签到信息签退否显示失败信息取消预约否提示用户预约座位取消预约

图4.5座位签到活动图

4.4 系统时序图设计

1. 在角色管理时序图中，管理员通过角色管理模块进行角色权限配置的全流程交互。当执行新增或修改操作时，前端Vue组件实时收集角色名称、权限标识等字段信息，通过 RESTful API将完整数据包发送至Spring Boot服务端。服务端Controller层接收请求后，调用Service层进行数据校验及业务逻辑处理，将有效数据持久化至数据库，随后返回操作结果通知前端。前端接收到成功响应后，立即触发列表数据刷新，确保界面展示与数据库状态实时同步。菜单分配功能通过权限字段映射机制实现：前端将用户勾选的菜单权限ID数组传递至服务端，Service层将权限数据序列化存储至数据库权限字段。系统基于最新权限配置动态生成菜单树结构，通过递归算法构建多级菜单节点，最终转换为Vue Router 可识别的路由配置，实现菜单权限与访问路径的自动化关联。具体角色管理时序图如图4.6 所示。

sd Frame1 前端-角色管理页面: RoleList.

vue 数据库后端-控制层: RoomController.java 后端-业务逻辑层:

RoomServiceImpl.java 实体类: SysRole.java Object1 :管理员 1 :执行新增或修改操作() 2 :收集权限标识等字段信息() 3 :发送请求() 4 :处理请求() 5 :处理实体() 6 :数据持久化() 7 :返回数据 8 :返回处理结果 9 :返回操作结果 10 :返回结果 11 :刷新列表 12 :显示更新后的列表 13 :进行菜单分配操作() 14 :收集用户勾选的权限ID数组() 15 :发从请求() 16 :处理请求() 17 :处理实体() 18 :将权限数据序列化() 19 :存储数据() 20 :返回数据 21 :返回分配结果 22 :返回分配结果 23 :返回结果 24 :基于权限配置生成菜单树 25 :配置路由配置 26 :分配成功

图4.6角色管理时序图

2. 在自习室管理时序图中，完整呈现了管理员通过管理模块进行自习室信息维护的核心交互流程。当执行新增或修改操作时，前端Vue组件首先收集自习室名称、类型、楼层等基础字段信息，同时支持图片文件的上传，后端Spring Boot服务接收文件流后，将其存储至本地服务器指定目录，并生成可访问的URL路径返回给前端。随后，前端将完整数据包（含文件路径）通过RESTful API提交至服务端，Controller层调用Service进行业务校验，将有效数据持久化至数据库。数据库操作完成后，服务端返回成功状态码，触发前端列表数据刷新，确保界面展示与数据

4. 3%(436)

4. 7%(480)

库状态实时同步。删除操作采用精准ID匹配机制： 前端根据用户选择的自习室条目生成包含唯一ID的删除请求，服务端解析请求后执行数据库删除指令。删除成功后，服务端返回操作确认，前端刷新移除界面中的对应自习室卡片，完成视图层与服务端的联动更新。整个时序流程清晰展示了从文件处理到数据持久化的完整链路，体现了前后端分离架构下自习室信息管理的状态同步与异常处理机制。具体自习室管理时序图如图4.7所示。

2. 基于SpringBoot与Vue的自习室管理系统的设计与实现_第2部分

AI特征值: 0.0%

AI特征字符数 / 章节(部分)字符数: 0 / 10755

片段指标列表

序号	片段名称	字符数	AI特征
----	------	-----	------

原文内容

sd自习室管理前端-自习室管理页面: RoomList.vue 后端-控制层: RoomController.java 后端-业务逻辑层: RoomServiceImpl.java 后端-实体类: Room.java 数据库 :管理员 1 :点击自习室管理() 2 :发起自习室列表请求() 3 :处理自习室列表数据() 4 :查询自习室数据() 5 :封装并返回处理结果 6 :返回处理结果 7 :返回处理结果 8 :返回自习室列表 9 :显示自习室列表 10 :点击新增自习室() 11 :填写表单信息() 12 :提交表单信息() 13 :处理表单信息() 14 :将表单信息转为实体类模型() 15 :存入数据库() 16 :返回处理结果 17 :返回结果码 18 :返回处理结果 19 :显示新增结果 20 :选择编辑自习室() 21 :填写表单信息() 22 :提交表单请求() 23 :处理表单数据() 24 :将表单信息转为实体类模型() 25 :存入数据库() 26 :返回处理结果 27 :返回结果码 28 :返回处理结果 29 :显示编辑结果 30 :点击删除自习室() 31 :携带ID提交删除请求() 32 :对数据进行删除更新() 33 :返回处理结果 34 :返回删除结果码 35 :返回删除结果 36 :显示删除结果

图4.7自习室管理时序图

3. 在失物招领时序图中，管理员对遗失物品或招领物品进行信息登记，前端界面通过表单验证规则强制校验必填字段（如内容、类型等）。编辑功能支持对物品属性的动态调整，特别是物品状态变更操作，确保流程的合理性。整个时序流程清晰展示了从数据校验、 业务处理到状态同步的完整链路。失物招领时序图如图4.8所示。

sd失物招领前端: Lost.vue 控制层: LostController.java 业务逻辑层: LostServiceImpl.java 实体类: Lost.java 数据库 :管理员 1 :发起物品登记() 2

:传递操作请求() 3 :调用业务逻辑() 4 :创建物品实体对象() 5 :执行数据库操作
() 6 :返回操作结果 7 :返回业务处理结果 8 :返回操作结果 9 :显示操作结果 10
:选择失物或招领类型() 11 :传递类型选择() 12 :处理类型选择() 13 :更新物品
类型信息() 14 :更新物品类型数据() 15 :返回更新结果 16 :返回处理结果() 17
:返回操作结果 18 :显示操作结果 19 :将物品信息改为已招领或已找回状态() 20
:传递状态更改请求() 21 :处理状态更改() 22 :更改物品状态信息() 23 :更新物
品状态数据() 24 :返回更新结果 25 :返回处理结果 26 :返回操作结果 27 :显示
操作结果

图4.8失物招领时序图

4. 在预约座位时序图中,完整呈现了用户从选择自习室到完成座位使用的完整交互流程。用户在前端界面加载自习室列表时,系统实时查询数据库获取座位状态信息,采用颜色标识区分可用/已预约两种状态。当用户选择具体座位后,前端触发预约请求,携带自习室ID、座位编号、预约时间段参数。服务端接收到请求后,查询该座位在指定时间段内是否存在冲突预约,校验通过后,服务端生成唯一预约ID,将预约记录写入数据库,并更新座位状态为"已预约"。前端接收到成功响应后,动态更新座位状态展示,并开放取消预约

按钮。预约座位时序图如图4.9所示。

sd预约座位前端-客户端页面: Room.vue 后端-控制层: RoomController.java
后端-业务逻辑层: RoomServiceImpl.java 后端-实体类: Room.java 数据库 :用户
1 :发起预约请求() 2 : API调用() 3 :传递预约请求() 4 :查询该时间段和座位是
否有冲突() 5 :返回查询结果 alt 6 :生成预约实体() [无冲突] 7 :插入预约数据
() 8 :返回插入结果 [有冲突] 9 :不生成实体,不插入数据 10 :返回处理结果 11
:返回数据 12 :展示预约结果

图4.9预约座位时序图

5. 在座位签到时序图中,详细描述了前后端协作实现签到功能的核心流程。当执行签到操作时,前端收集座位信息,通过RESTful API提交至服务端。服务端Controller层接收请求后,调用Service层进行业务校验,重点验证座位状态是否为已预约。校验通过后,更新数据库座位状态信息,返回前端。座位签到时序图如图4.10所示。

sd座位签到 :用户前端-客户端页面: Room.vue 1 :签到() 12 :签退() 后端-
控制层: RoomController.java 后端-业务逻辑层: RoomServiceImpl.java 后端-实
体类: Room.java 数据库 2 :发起签到请求() 11 :显示签到成功 13 :发起签退请
求() 21 :显示签退成功 3 :传递签到请求() 10 :返回签到成功结果 14 :传递签退
请求() 20 :返回签退成功结果 4 :获取座位预约信息() 7 :检查是否可签到 5 :查
询座位预约状态() 6 :返回座位预约状态 8 :更新座位签到状态() 9 :签到成功反
馈 15 :获取座位签到信息() 16 :查询座位签到状态() 17 :返回座位签到状态 18

:更新座位签退状态() 19 :签退成功反馈

图4.10座位签到时序图

4.5 数据库设计

本系统采用MySQL数据库作为核心存储方案，利用其事务处理与并发控制机制，确保多用户并发场景下的数据一致性。InnoDB引擎的行级锁和MVCC技术有效支持高并发访问，配合索引优化策略，保障关键查询的毫秒级响应。数据库架构采用读写分离和分库设计，为系统扩展奠定坚实基础。

4.5.1数据库关系模型设计

本系统通过科学的类模型设计，将面向对象概念精准映射为关系型数据库结构。采用UML建模语言定义类间继承、关联、依赖等关系，基于数据库范式理论消除冗余属性，将类属性拆解为原子化字段。通过单表继承策略实现类型扩展，利用外键约束维护参照完整性，最终形成包含用户体系、课程管理、订单日志等核心模块的数据库模型，如图4.11。该模型严格遵循第三范式，确保数据一致性的同时，为高频查询字段建立覆盖索引，兼顾性能与规范化要求。

图4.11数据库关系映射图

4.5.2数据库物理设计

(1) 公告信息表(notice) 主要包含公告ID、类别ID等5个字段。具体公告信息表如表4.1所示。

表4.1公告信息表

序号列名数据类型是否为空主键自增 1 notice_id int 否是是 2 category_id int 是否否 d 3 notice_title varchar(255) 是否否 4 notice_content text 是否否 t 5 create_time datetime 是否否 e 备注公告id 类型id 标题内容发布时间

(2) 公告类别信息表(notice_category) 主要包含公告类别ID、公告类别名称、公告类别序号3个字段。具体公告类别信息表如表4.2所示。

表4.2公告类别信息表

序号列名数据类型是否为空主键自增 1 category_id int 否是是 d 2 category_name varchar(64) 是否否 ame 3 order_num int 是否否备注类型id 类型名称序号

(3) 自习室信息表(room) 主要包含自习室ID、自习室名称等9个字段。具体自习室信息表如表4.3所示。

表4.3自习室信息表

序号列名数据类型是否为空主键自增 1 room_id int 否是是 2 floor_id int 是否否 3 category_id int 是否否 d 4 room_name varchar(64) 是否否 5 image varchar(256) 是否否 6 num_x int 是否否 7 num_y int 是否否 8 order_num int 是否否 9 area varchar(36) 是否否备注主键类型id 楼层id 阅览室名称图片行数 列数序号区域

(4) 自习室类别信息表 (room_category) 主要包含自习室类别ID、自习室类别名称、 自习室类别序号3个字段。具体自习室类别信息表如表4.4所示。

表4.4自习室类别信息表

序号列名数据类型是否为空主键自增 1 category_id int 否是是 2 category_name varchar(36) 是否否 备注主键类型名称 3 order_num int 是否否 序号

(5) 自习室楼层信息表 (room_floor) 主要包含自习室楼层ID、自习室楼层名称等4 个字段。具体自习室楼层信息表如表4.5所示。

表4.5自习室楼层信息表

序号列名数据类型是否为空主键自增备注 1 floor_id int 否是是主键 2 floor_name varchar(64) 是否否楼层名称 3 floor_num int 是否否楼层数量 4 order_num int 是否否序号

(6) 座位信息表 (room_seat) 主要包含自习室楼层ID、自习室楼层名称等4个字段。 具体座位信息表如表4.6所示。

表4.6座位信息表

序号列名数据类型是否为空主键自增备注 1 seat_id int 否是是座位id 2 room_id int 是否否自习室id 3 seat_row int 是否否行数 4 seat_column int 是否否列数 5 seat_name varchar(64) 是否否名称

(7) 班级信息表 (sys_class) 主要包含班级ID、专业ID等6个字段. 具体班级信息表如表4.7所示。

表4.7班级信息表

序号列名数据类型是否为空主键自增备注 1 class_id int 否是是班级id 2 major_id int 是否否专业id 3 class_name varchar(64) 是否否班级名称 4 class_num varchar(36) 是否否班级编号 5 order_num int 是否否序号 6 years varchar(36) 是否否年纪

(8) 专业信息表 (sys_major) 主要包含专业ID、专业名称等4个字段. 具体专业信息表如表4.8所示。

表4.8专业信息表

序号列名数据类型是否为空主键自增备注 1 major_id int 否是是专业id 2 major_name varchar(64) 是否否专业名称 3 major_num varchar(36) 是否否专业编号 4 order_num int 是否否序号

(9) 菜单信息表 (sys_menu) 主要包含菜单ID、父级菜单id等4个字段. 具体菜单信息表如表4.9所示。

表4.9菜单信息表

序号列名数据类型是否为空主键自增 1 menu_id int 否是是 2 parent_id int 是否否 3 title varchar(64) 是否否 4 code varchar(64) 是否否 5 name

varchar(36) 是否否 6 path varchar(36) 是否否 7 url varchar(128) 是否否 8
type varchar(2) 是否否 9 icon varchar(36) 是否否 10 parent_name
varchar(64) 是否否 11 order_num int 是否否 12 create_time datetime 是否否
13 update_time datetime 是否否备注菜单id 父级菜单id 菜单名称权限字段路由
name 路由path 组件路径类型(0目录1 菜单2按钮) 菜单图标上级菜单名称序号创建
时间更新时间

(10) 角色信息表 (sys_role) 主要包含角色ID、角色名称等6个字段. 具体角
色信息表如表4. 10所示。

表4. 10菜单信息表

序号列名数据类型是否为空主键自增 1 role_id int 否是是 2 role_name
varchar(64) 是否否 3 type varchar(10) 是否否 4 remark varchar(128) 否否否
5 create_time datetime 是否否 6 update_time datetime 是否否备注角色id 角
色名称拓展字段备注创建时间更新时间

(11) 角色菜单信息表 (sys_role_menu) 主要包含角色菜单ID、角色ID等3个
字段. 具体角色菜单信息表如表4. 11所示。

表4. 11角色菜单信息表

序号列名数据类型是否为空主键自增 1 role_menu_id int 否是是 2 menu_id
int 是否否 3 role_id int 是否否备注主键菜单id 角色id

(12) 学生信息表 (sys_student) 主要包含学生ID、班级ID等8个字段. 具体学
生信息表如表4. 12所示。

表4. 12学生信息表

序号列名数据类型是否为空主键自增 1 stu_id int 否是是 2 class_id int
是否否备注学生id

表4. 12学生信息表 (续)

序号列名数据类型是否为空主键自增 3 stu_name varchar(36) 是否否 4 sex
varchar(2) 是否否 5 phone varchar(18) 是否否 6 stu_num varchar(64) 是否否
7 password varchar(128) 是否否 8 status varchar(2) 是否否备注学生姓名性别
电话密码状态

(13) 用户角色信息表 (sys_user_role) 主要包含用户学生ID、用户ID等3个
字段. 具体用户角色信息表如表4. 13所示。

表4. 13用户角色信息表

序号列名数据类型是否为空主键自增 1 user_role_id int 否是是 2 user_id
int 否否否 3 role_id int 否否否备注主键用户id 角色id

(14) 用户信息表 (sys_user) 主要包含用户ID、用户名称等14个字段. 具体用
户信息表如表4. 14所示。

表4. 14学生信息表

序号列名数据类型是否为空主键自增备注 1 user_id int 否是用户id

表4.14学生信息表（续）

序号列名数据类型是否为空主键自增备注 2 username varchar(36) 是否否登录账户 3 password varchar(128) 是否否登录密码 4 phone varchar(18) 是否否电话 5 email varchar(36) 是否否邮箱 6 sex varchar(4) 是否否性别是否是超级管理 7 is_admin tinyint 是否否员(1是0否) 账户是否过期 is_account_n 8 tinyint 是否否 (1未过期0已过 on_expired 期) 账户是否被锁定 is_account_n 9 tinyint 是否否 (1未锁定0已锁 on_locked 定) is_credentia 密码是否过期 10 ls_non_expir tinyint 是否否 (1未过期0已过 ed 期) 账户是否可用 11 is_enabled tinyint 是否否 (1可用0已删除用户) 12 nick_name varchar(36) 是否否姓名 13 create_time datetime 是否否创建时间 14 update_time datetime 是否否更新时间

4.6系统主要类模型设计

类模型设计通过规范化对象特征与交互关系构建系统结构框架。本系统采用面向对象设计原则，通过四个核心实现类，包括用户对象实现类

(SysUserServiceImpl)、角色对象实现类(SysRoleServiceImpl)、自习室对象实现类(RoomServiceImpl)、菜单对象实现类(SysMenuServiceImpl)，构建了四个核心模块，分别为用户模块、角色模块、自习室模块、菜单模块，这些核心模块相互交织，共同构成了系统的整体类模型结构。

1. 用户模块以SysUserServiceImpl类为核心，在业务逻辑层面，SysUserServiceImpl 类实现了SysUserService接口的5个公开方法。通过调用这些方法，以此处理关于用户的相关操作。具体用户模块的主要类图如图4.12所示。

图4.12用户模块主要类图

2. 角色模块以SysRoleServiceImpl类为核心，在业务逻辑层面，SysRoleServiceImpl 类实现了ServiceImpl接口的公开方法。通过调用这些方法，以此处理关于角色权限的相关操作。具体角色模块的主要类图如图4.13所示。

图4.13角色模块主要类图

3. 自习室模块以RoomServiceImpl类为核心，在业务逻辑层面，RoomServiceImpl类实现了RoomService接口的公开方法。通过调用这些方法，以此处理关于自习室模块的相关操作。具体自习室模块的主要类图如图4.14所示。

图4.14自习室模块主要类图

4. 菜单模块以SysMenuServiceImpl类为核心，在业务逻辑层面，SysMenuServiceImpl 类实现了SysMenuService接口的公开方法。通过调用这些方法，以此处理关于菜单模块的相关操作。具体菜单模块的主要类图如图4.15所示。

图4.15菜单模块主要类图

4.7 本章小结

本章重点完成自习室管理系统设计，运用活动图梳理用户预约全流程，时序图细化模块交互逻辑，类图抽象实体关系，并设计数据库表结构，确保系统流程清晰、数据规范。

5 系统实现

5.1 系统功能说明

本系统共分为用户前台系统和管理员后台系统，其中用户前台系统拥有查询自习室、查看失物、查看公告、座位预约等功能，后台系统则实现了对自习室管理、系统管理、角色管理等功能。

5.2 主要功能实现

5.2.1 角色管理功能

角色分配菜单权限功能是将权限分配完成的角色分配给次级管理员，令次级管理员仅能使用被分配权限。主要是对菜单树进行查询，将角色表与菜单表进行联立查询并生成动态路由，并在前端和后端进行双重验证，前端使得次级管理员无法看到未被授权的权限字段，后端使得次级管理员无法使用未被授权的权限字段。角色分配菜单权限功能如图5.1所示。

图5.1 角色分配菜单权限功能界面

对于角色分配菜单权限功能模块，后端权限查询代码详细描述：

```
@Override
public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException { SysUser user =
sysUserService.loadUser(username);
    在!"); {
}

if (user==null){ throw new CustomerAuthenticationException("用户名错
误或账户不存 }

//查询菜单权限List<SysMenu> menuList = null;
//判断当前用户是否是超级管理员 if
(StringUtils.isEmpty(user.getIsAdmin()) &&
"1".equals(user.getIsAdmin()))

//超级管理员，直接全部查询 menuList = sysMenuService.list();
} else { menuList = sysMenuService.getMenuByUserId(user.getUserId());
}

//获取菜单表的code字段List<String> collect =
Optional.ofNullable(menuList).orElse(new ArrayList<>()) .stream()
    .filter(item -> item != null &&
StringUtils.isEmpty(item.getCode()))
```

```

.map(item -> item.getCode())
.collect(Collectors.toList());
//将权限字段交给spring security进行管理
String[] strings = collect.toArray(new String[collect.size()]);
List<GrantedAuthority> authorityList =
AuthorityUtils.createAuthorityList(strings); //权限集合
user.setAuthorities(authorityList); //设置权限 return user;

```

5.2.2 自习室管理功能

自习室管理通过的新增、修改功能通过前端填写表单后发送post请求到后端持久层中对相关非复杂性业务通过Mybatis-Plus提供的方法进行处理，在指定ID删除通过获取前端ID回传后端进行处理，对上传自习室图片等业务在前端通过Element Plus实现。而对自习室的分页，采用了Mybatis-Plus中的分页插件功能，查询以及数据回显通过 TypeScript进行实现。自习室管理功能如图5.2所示。

图5.2 自习室管理展示功能界面

对于自习室管理模块，部分实现代码详细描述：

```

@Override public void edit(Room room) { //编辑自习室 int insert =
this.baseMapper.updateById(room); //查看座位是否发生变化 if (insert>0) {
//把原来座位号先删除 QueryWrapper<RoomSeat> query=new QueryWrapper<>();
query.lambda().eq(RoomSeat::getRoomId,room.getRoomId());
roomSeatService.remove(query); //重新插入座位 List<RoomSeat> list=new
ArrayList<>(); //生成座位号
for (int i=1;i<=room.getNumX();i++) {
for (int j=1;j<=room.getNumY();j++) { RoomSeat roomSeat=new
RoomSeat(); roomSeat.setSeatRow(i); roomSeat.setSeatColumn(j);
roomSeat.setRoomId(room.getRoomId());
roomSeat.setSeatName(String.valueOf(i)+ "排"+String.valueOf(j)+"列");
list.add(roomSeat); } } if (list.size()>0) {
roomSeatService.saveBatch(list); } } }

```

5.2.3 失物招领管理功能

失物招领系统聚焦于物品信息全生命周期管理，管理员通过可视化表单录入失物或招领信息，系统自动生成包含物品特征、拾取/丢失时间地点、联系方式等结构化数据存入数据库，并支持上传多维度佐证图片。后台提供状态流转工作台，管理员可一键切换物品状态为已认领或已找回，系统实时更新物品标签颜色与前端展示优先级。显著提升失物处理效率与信息透明度。失物招领管理功能如图5.3所示。

图5.3 失物招领管理展示功能界面

对于失物招领管理模块，部分实现代码详细描述：

```
@Override
    public List<LostItemVO> listPendingItems(String statusFilter) {
        LambdaQueryWrapper<LostItem> queryWrapper = new LambdaQueryWrapper<>();
        if (StringUtils.isNotBlank(statusFilter)) {
            queryWrapper.eq(LostItem::getStatus, statusFilter);
        } else {
            queryWrapper.in(LostItem::getStatus, Arrays.asList("pending", "claimed", "found"));
        }
        queryWrapper.orderByDesc(LostItem::getCreateTime);
        List<LostItem> items = itemMapper.selectList(queryWrapper);
        return items.stream().map(item -> {
            LostItemVO vo = new LostItemVO();
            BeanUtils.copyProperties(item, vo);
            vo.setImages(JSON.parseArray(item.getImages(), String.class));
            return vo;
        }).collect(Collectors.toList());
    }
```

5.2.4 预约座位功能

预约座位系统是用户对座位进行预约的功能模块，管理员针对自习室录入座位数据， 用户可查看空座并提交预约，系统自动核验时间冲突与身份权限，判断座位实时状态，预约成功后锁定座位并同步状态至管理端。管理员可调整开放时段，超时未签到时自动释放座位资源，同时通过数据统计使用高峰，优化空间分配。预约功能如图5.4所示。

3. 基于SpringBoot与Vue的自习室管理系统的设计与实现_第3部分

AI特征值: 10.5% AI特征字符数 / 章节(部分)字符数: 806 / 7652

片段指标列表

序号	片段名称	字符数	AI特征		
9	片段1	242	显著		3.2%
10	片段2	262	显著		3.4%
11	片段3	317	疑似		4.1%
12	片段4	234	疑似		3.1%
13	片段5	237	疑似		3.1%
14	片段6	302	显著		3.9%

原文内容

图5.4预约座位功能界面

对于预约座位功能模块，部分实现代码详细描述：

```
@Override
```

```
public boolean saveReservation(SeatReservation reservation) { //检查座位是否可用 List<String> seats = reservation.getSeatList(); for (String seat : seats) { if (!isSeatAvailable(seat, reservation.getDate(), reservation.getTimeSlot())) { throw new RuntimeException("座位" + seat + "在指定时段已被预约"); } } return save(reservation); }
```

5.2.5座位签到功能

座位签到用于用户预约座位后，选择签到后，系统自动标记座位为“使用中”状态并记录签到时间；使用完毕后进行签离释放座位，系统更新状态为“空闲”。管理员可通过后台查看实时座位占用情况及历史使用记录，处理未正常签离的异常订单，保障座位资源循环利用。座位签到如图5.5所示。

图5.5座位签到功能界面

对于座位签到功能模块，部分实现代码详细描述：

```
@Override public SeatReservation validateReservation(Long reservationId, String seatNumber) {  
    SeatReservation reservation =  
reservationMapper.selectById(reservationId); if (reservation == null) {  
throw new RuntimeException("未找到对应的预约记录"); }  
    //检查座位是否匹配  
List<String> reservedSeats =  
Arrays.asList(reservation.getSeats().split(",")); if  
(!reservedSeats.contains(seatNumber)) { throw new RuntimeException("该座位不在预约范围内"); }  
    //检查预约时间  
LocalDate today = LocalDate.now(); if  
(!reservation.getDate().equals(today)) { throw new RuntimeException("只能在预约当天签到"); }  
    //检查是否已签到  
QueryWrapper<SeatCheckIn> queryWrapper = new QueryWrapper<>();  
queryWrapper.eq("reservation_id", reservationId)  
.eq("seat_number", seatNumber)  
.eq("status", "checked-in"); if (count(queryWrapper) > 0) { throw new  
RuntimeException("该座位已签到"); } return reservation; }
```

5.3 本章小结

本章节旨在概要阐述该系统的核心功能模块，通过逐一剖析各个功能模块的设计理念、工作原理及其在整个系统架构中的作用，并结合实际的代码示例与直观的效果图进行详尽的解读与分析。

6 系统测试

6.1 测试方法

黑盒测试也称功能测试，将软件视为不透明的封闭单元，仅通过输入输出验证功能实现。检测软件功能能否按照需求规格说明书的规定正常工作、是否有功能遗漏、检测是否有人机交互错误、是否有数据结构和外部数据库访问错误、检测行为、性能等特性是否满足要求等，它是从用户角度出发，确保系统各功能模块按预期进行。白盒测试也称结构测试，基于代码可见性，验证程序内部逻辑是否符合设计规范，按照程序内部的结构测试程序，检验每条代码执行路径是否按预期运行、检查函数调用、变量状态等内部行为与设计文档的一致性、确保控制流、数据流的逻辑严密性，排除隐藏缺陷等。本文使用黑盒测试确保系统的可用性和有效性。

6.2 功能测试

功能测试是软件系统测试中最基础、最重要的测试类型之一。其主要目标是确保软件系统的各项功能符合需求和设计要求，满足用户的使用需求。通过功能测试，可以发现软件系统在功能方面的缺陷和问题，提高软件的质量和可靠性。该系统面向用户提供了一系列服务，例如自习室查询、自习室预约等，而管理员则肩负着管理自习室、维护用户信息以及其他关联信息的职责。在此背景下，对该系统的功能测试核心聚焦于以下几个方面：对角色管理功能的有效性测试、对自习室预约的功能验证，以及菜单管理功能的测试。

6.3 主要测试用例设计

(1) 管理员可通过输入用户姓名关键字或电话号码查询用户信息，若数据库中无匹配记录，系统将显示空数据集；在新增或编辑用户信息时，需校验姓名、电话、角色、账户、密码等必填字段是否为空值；执行删除操作前，系统应弹出确认提示框以避免误操作；重置用户密码时，需触发系统提示框进行二次确认；当完成角色分配后，系统需自动刷新用户权限配置，确保新分配的角色权限即时生效，整套流程需覆盖数据校验、权限管控及操作反馈等核心环节。用户管理测试用例如表6.1所示。

表6.1 用户管理模块功能测试用例

功能	用户管理	测试编号	01	测试用例	检测管理员是否正常使用用户管理功能	预置条件	管理员进入用户管理界面	操作描述	测试数据	预期结果	实际结果	结论
新增用户	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功	新增用户成功
编辑用户	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功	编辑用户成功
删除用户	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功	删除用户成功

3. 2%(242)

户成功删除用户成功测试通过显示用户名字中含显示用户名字中 “测试”的用户，含“测试”的用户， 用户名称：测试，电查询用户显示电话是显示电话是测试通过话：“13799493610” “13799493610”的 “13799493610”的用户用户查询不存在用户用户名称：孙显示空数据集显示空数据集测试通过重置密码用户名称：测试2重置密码成功重置密码成功测试通过用户名称:王五,角分配角色 分配角色成功分配角色成功测试通过色：区域管理员

在用户管理模块测试中，首先验证了查询功能的准确性：当输入数据库中存在的用户精确信息时，系统能正确检索并完整展示该用户信息；若查询条件与数据库记录不匹配，系统则准确反馈空数据集。针对用户信息的新增与修改操作，测试表明系统能稳定执行数据写入及更新流程，确保信息变更即时生效。删除功能测试结果显示，系统严格遵循预设删除策略，精准移除目标用户数据。角色分配验证环节，通过以分配角色身份登录系统并核查权限配置，确认角色权限绑定逻辑有效。整体测试过程中，用户信息管理各功能模块均表现出符合预期的设计规范与业务需求，系统表现稳定可靠。

3. 4%(262)

对用户增删改查操作的测试均通过，测试结果如图6.1和6.2所示。

图6.1测试编号01增删改查截图

图6.2测试编号01增删改查结果截图

查询不存在用户功能测试的目的是验证当查询数据库不存在数据时，后端是否会返回空数据. 测试结果如图6.3所示。

图6.3测试编号01查询不存在用户结果截图

重置密码功能测试的目的是验证查询数据库不存在数据时，后端是否会返回空数据. 测试结果如图6.4所示。

图6.4测试编号01重置密码结果截图

分配角色功能测试的目的是验证对用户进行角色分配后，后端是否正确应用角色权限。 测试结果如图6.5所示。

图6.5测试编号01分配角色结果截图

(2) 管理员可通过输入自习室名称或类别发起查询，系统需准确展示符合条件的自习室信息；在新增或编辑自习室信息时，需验证上传封面图片的功能是否正常运行，同时确保自习室类型与楼层表的关联准确无误；若搜索不存在的自习室，系统应明确提示"暂无数据"；执行删除操作后，目标自习室需从数据库及前端界面彻底移除；点击重置按钮时，搜索栏内容应清空，系统需重新加载完整数据集并默认返回第一页结果，确保各项交互逻辑与数据展示符合预期。自习室管理测试用例如表6.2所示。

4. 1%(317)

表6.2自习室管理模块功能测试用例

功能测试编号测试用例预置条件操作描述新增自习室编辑自习室删除自习室自

自习室管理 02 检测管理员是否正常使用自习室管理功能管理员进入自习室管理界面
测试数据预期结果实际结果结论自习室名称：测试自习室，类别：舒适简约，自习
室楼层：集美馆一层，座位行数：10，座位列数： 10，图片：任意图片自习室名称
：测试2， 类别：明朗开阔自习室名称名称：测试2 新增自习室成功编辑自习室成
功删除自习室成功新增自习室成功测试通过编辑自习室成功测试通过删除自习室成
功测试通过

表6.2自习室管理模块功能测试用例（续）
功能自习室管理测试编号 02 测试用例检测管理员是否正常使用自习室管理功
能预置条件管理员进入自习室管理界面操作描述测试数据预期结果实际结果结论显
示自习室名称中显示自习室名称中自习室名称：测试2，含“测试”的用户，含“测
试”的用户， 查询自习室 测试通过类别：明朗开阔显示类别是“明朗显示类别是
“明朗开阔”的自习室开阔”的自习室查询不存在自习类别名称：精巧从容显示空
数据集显示空数据集测试通过室给搜索栏清空，查询搜索栏清空，查询重置自习室
名称：测试所有数据集，返回所有数据集，返回测试通过第一页第一页

在自习室管理模块，通过对查询、新增、修改和删除操作的综合验证，管理员
能够准确查询并展示符合条件的自习室记录，同时上传自习室图片资源功能正常。
自习室管理的各项测试用例均执行成功，功能表现符合设计规范与业务需求。

新增自习室功能测试的目的是验证添加自习室后，相对应的属性图片能否正确
存入数据库中。测试结果如图6.6所示。

图6.6测试编号02新增自习室结果截图
编辑自习室功能测试的目的是验证修改自习室相关信息，数据库对应属性是否
同步修改。测试结果如图6.7所示。

图6.7测试编号02编辑自习室结果截图
删除自习室功能测试的目的是验证删除自习室后，数据库相应数据是否正确被
删除。 测试结果如图6.8所示。

图6.8测试编号02删除自习室结果截图
重置功能测试的目的是验证重置后，是否查询全部数据且搜索栏清空。测试结
果如图 6.9所示。

图6.9测试编号02重置结果截图
(3) 在反馈模块功能的测试中，管理员需要验证用户能否通过意见反馈功能提
交反馈到管理员反馈列表界面。反馈管理测试用例如表6.3所示。

表6.3反馈管理模块功能测试用例
功能测试编号测试用例预置条件操作描述反馈管理 03 检测用户，管理员是否
正常使用反馈功能用户进入意见反馈，管理员进入反馈管理界面提交反馈测试数据
预期结果实际结果结论标题：测试，内容测试用户提交反馈成用户提交反馈成功
，管理员查看到功，管理员查看到测试通过后台反馈数据后台反馈数据

3.1%(234)

提交反馈测试的目的是输入数据提交后，验证后端存入数据后，前端能否成功返回。测试结果如图6.10所示。

图6.10测试编号03反馈列表结果截图

(4) 在公告模块功能的测试中，管理员在公告管理上传公告内容，用户可在公告查看公告信息。公告管理测试用例如表6.11所示。

表6.4公告模块功能测试用例

功能测试编号测试用例预置条件操作描述新增公告公告管理 04 检测管理员是否正常使用公告管理功能管理员进入公告管理界面，用户成功登录测试数据预期结果实际结果结论公告类型：普通公告，公告标题：测试新增公告成功，前新增公告成功，前 888，公告内容：测台显示公告数据台显示公告数据试888 测试通过

新增公告测试的目的是提交公告信息后，验证前端能否返回相应数据。测试结果如图 6.11所示。

图6.11测试编号04查看公告结果截图

(5) 预约模块，是自习室系统的核心功能，用户可在预约列表选择自习室并查看自习室信息。可对自习室座位进行选择预约，选择预约时间后，判断预约时间是否小于当前时间，如果小于当前时间则跳出“预约时间不能小于当前时间”。用户点击确认预约后成功预约。用户可选择对预约进行取消预约或签到，如未到半小时签到时间则跳出“未到签到时间，不能签到”。对已预约座位再进行预约则跳出“您存在未签到或者签退的预约，请先签到或签退”。签到完成后可对预约进行签退。管理员可在预约列表中查看预约数据。 预约管理测试用例如表6.5所示。

表6.5预约模块功能测试用例

功能预约管理测试编号 05 测试用例检测用户否正常使用预约管理功能预置条件用户登录前台系统操作描述测试数据自习室名称：测试预约座位 999，预约时间：“13: 24”，座位：1排2 列取消预约自习室名称：测试6， 预约座位：1排1列预期结果座位预约成功取消预约成功实际结果结论座位预约成功测试通过取消预约成功测试通过

预约座位测试的目的是选择座位进行预约后，验证座位是否成功被预约。测试结果如图6.12所示。

图6.12测试编号05预约座位结果截图

取消预约测试的目的是用户取消已预约座位的预约时，被取消座位是否能再被预约， 对应信息是否在我的预约中。测试结果如图6.13所示。

图6.13测试编号05取消预约结果截图

(6) 失物招领模块，是自习室管理系统的主要功能之一。对于丢失物品或者招领物品，管理员可对其进行数据登记。对已找回物品或已招领物品，管理员可对已登记信息进行状态更新。失物招领管理测试用例如表6.6所示。

表6.6失物招领模块功能测试用例

功能失物招领管理测试编号 06 测试用例检测管理员否正常使用失物招领管理功能预置条件管理员进入失物招领管理页面操作描述测试数据预期结果实际结果结论标题：捡到水杯一个，内容：本人在图新增物品书馆2楼鱼跳楼捡到新增物品成功新增物品成功测试通过水杯一个，类型：招认领人：测试，认领认领物品 认领物品成功认领物品成功测试通过人电话：12424

新增物品测试的目的是管理员新增物品后，用户是否可以正确查看失物招领的信息。测试如图6. 14所示。

图6. 14测试编号06新增物品结果截图

认领物品测试的目的成功认领物品后，物品的状态是否正确发生改变。测试结果如图 6. 15所示。

图6. 15测试编号06认领物品结果截图

(7) 座位签到模块，用户可对已预约座位进行签到，如未到半小时签到时间则跳出 “未到签到时间，不能签到”。对已预约座位再进行预约则跳出“您存在未签到或者签退的预约，请先签到或签退”。签到完成后可对预约进行签退。座位签到测试用例如表6. 7所示。

表6. 7座位签到模块功能测试用例

功能测试编号测试用例预置条件操作描述座位签到座位签到 07 检测用户否正常使用座位签到功能用户成功预约座位测试数据预期结果实际结果结论自习室名称：测试 999，预约时间：“13： 24”，座位：1排2 列座位预约成功座位预约成功测试通过

表6. 7座位签到模块功能测试用例(续)

功能座位签到测试编号 07 测试用例检测用户否正常使用座位签到功能预置条件用户成功预约座位操作描述测试数据预期结果实际结果结论自习室名称：测试 999，预约时间：“13： 座位签离 取消预约成功取消预约成功测试通过 24”，座位：1排2 列

座位签到测试的目的是用户对已预约座位进行签到，后台是否有对应信息。测试结果如图6. 16所示。

图6. 16测试编号05座位签到结果截图

座位签离测试的目的是用户签到座位后，能否成功进行签离。测试结果如图 6. 17所示。

图6. 17测试编号05座位签退结果截图

(8) 角色管理模块可方便管理员下发权限给次级管理员，管理员创建角色后，可对角色进行菜单权限的分配，针对不同的需求分配对应的权限。角色管理模块可实现精细化权限管控。角色管理用例如表6. 8所示。

表6. 8角色管理模块功能测试用例

功能角色管理测试编号 07 测试用例检测管理员否正常使用角色管理功能预置

条件管理员进入角色管理页面操作描述测试数据预期结果实际结果结论角色名称：教师，分分配菜单配菜单：专业管理， 座位预约成功分配成功测试通过班级管理分配菜单测试的目的是管理员针对不同的业务需求分配相应的权限，将权限下发给次级管理员，是否可以成功分配。测试结果如图6.18所示

图6.18测试编号05分配菜单结果截图

6.4 测试总结

基于Spring Boot与Vue技术栈构建的自习室管理系统，在综合测试中展现了稳定的技术实现与流畅的用户体验。后端Spring Boot框架确保了自习室管理、用户权限控制及业务逻辑处理的准确性，API接口在并发场景下仍保持高效响应；前端Vue组件实现了数据动态渲染与交互逻辑，座位预约、公告发布及反馈管理等功能模块界面操作流畅，状态同步及时。系统成功通过了功能完整性、边界条件及异常流程测试，前后端协作稳定，数据一致性得到保障，整体表现符合设计预期。

3.1%(237)

7 总结与展望

7.1 总结

本文基于Spring Boot与Vue技术栈，结合Uniapp跨平台框架，设计并实现了一套高效、可扩展的自习室管理系统。系统后端通过Spring Security与JWT实现动态权限控制，前端采用Vue与Uniapp分别构建管理端和用户端界面，支持Web、iOS、Android及微信小程序多端访问。通过模块化设计与前后端分离架构，系统解决了传统自习室管理的座位预约低效、信息不透明等问题，并利用Uniapp的跨平台特性降低多端开发成本，提升用户体验。功能测试验证了预约、签到、权限管理等核心模块的稳定性与可用性，为校园及公共场所自习室管理提供了完整的数字化解决方案。

3.9%(302)

7.2展望

未来可进一步探索微服务架构提升系统扩展性，结合物联网技术实现设备能耗智能监控，优化节能策略；引入机器学习算法预测高峰时段座位需求，动态调整资源分配；同时集成人脸识别或智能签到技术提高安全性与管理效率。此外，可深化数据分析模块，为管理者提供多维度报表支持，推动自习室管理向智能化、生态化方向发展。

致谢

求学数载，幸得良多。感谢学校，给了我学习的机会！感谢我的舍友，陪伴了我的大学时光。特别感谢我的指导老师，论文的完成离不开她无微不至的帮助。最后，将最深沉的感谢献给始终默默支持的父母。谢谢你们！

说明:

- 1、支持中、英文内容检测;
- 2、AI特征值=AI特征字符数/总字符数;

- 3、红色代表AI特征显著部分，计入AI特征字符数；
- 4、棕色代表AI特征疑似部分，未计入AI特征字符数；
- 5、检测结果仅供参考，最终判定是否存在学术不端行为时，需结合人工复核、机构审查以及具体学术政策的综合应用进行审慎判断。



关注微信公众号

知网AIGC检测服务

知网AIGC检测服务