

AIGC检测 · 全文报告单

NO:CNKIAIGC2026SJ_20260593218716

检测时间:2026-05-07 15:10:13

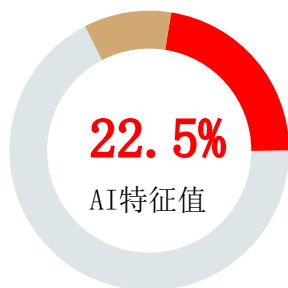
篇名: 基于SpringBoot的学校低值易耗品系统的设计与开发

作者: 丁亚洲

单位:

文件名:

全文检测结果



AI特征值: 22.5%
AI特征字符数: 6553
总字符数: 29127

- 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: 4.5%

AI特征字符数: 1310

中部60%

AI特征值: 13.5%

AI特征字符数: 3931

后部20%

AI特征值: 4.5%

AI特征字符数: 1310



分段检测结果










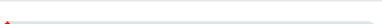


















序号	AI特征值	AI特征字符数/章节(部分)字符数	章节(部分)名称
1	22.5%	6553 / 29127	论文全文

1. 论文全文

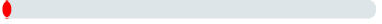



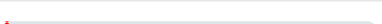




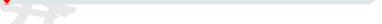
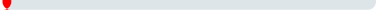
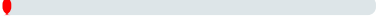




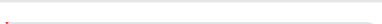



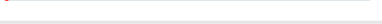
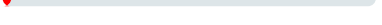
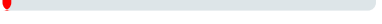
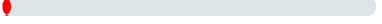
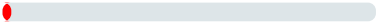







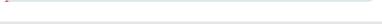

AI特征值: 22.5% AI特征字符数 / 章节(部分)字符数: 6553 / 29127


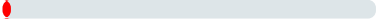







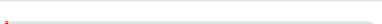









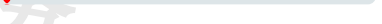

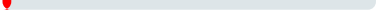

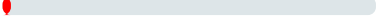









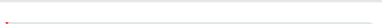







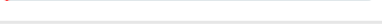

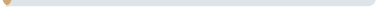

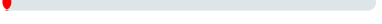

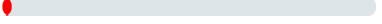

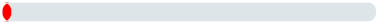















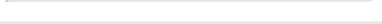


片段指标列表

序号	片段名称	字符数	
1	片段1	61	0.2%

2	片段2	37		0.1%
3	片段3	66		0.2%
4	片段4	158		0.5%
5	片段5	117		0.4%
6	片段6	156		0.5%
7	片段7	143		0.5%
8	片段8	46		0.2%
9	片段9	50		0.2%
10	片段10	90		0.3%
11	片段11	51		0.2%
12	片段12	85		0.3%
13	片段13	41		0.1%
14	片段14	67		0.2%
15	片段15	261		0.9%
16	片段16	198		0.7%
17	片段17	117		0.4%
18	片段18	163		0.6%
19	片段19	102		0.4%
20	片段20	56		0.2%
21	片段21	52		0.2%
22	片段22	43		0.1%
23	片段23	145		0.5%
24	片段24	130		0.4%
25	片段25	40		0.1%
26	片段26	95		0.3%
27	片段27	74		0.3%
28	片段28	85		0.3%
29	片段29	57		0.2%
30	片段30	39		0.1%
31	片段31	39		0.1%
32	片段32	61		0.2%
33	片段33	85		0.3%
34	片段34	83		0.3%
35	片段35	80		0.3%

36	片段36	79		0.3%
37	片段37	56		0.2%
38	片段38	63		0.2%
39	片段39	105		0.4%
40	片段40	216		0.7%
41	片段41	42		0.1%
42	片段42	64		0.2%
43	片段43	46		0.2%
44	片段44	55		0.2%
45	片段45	17		0.1%
46	片段46	50		0.2%
47	片段47	38		0.1%
48	片段48	38		0.1%
49	片段49	39		0.1%
50	片段50	111		0.4%
51	片段51	56		0.2%
52	片段52	15		0.1%
53	片段53	32		0.1%
54	片段54	40		0.1%
55	片段55	24		0.1%
56	片段56	40		0.1%
57	片段57	15		0.1%
58	片段58	61		0.2%
59	片段59	50		0.2%
60	片段60	87		0.3%
61	片段61	16		0.1%
62	片段62	41		0.1%
63	片段63	85		0.3%
64	片段64	60		0.2%
65	片段65	84		0.3%
66	片段66	48		0.2%
67	片段67	44		0.2%
68	片段68	55		0.2%
69	片段69	60		0.2%

70	片段70	55		0.2%
71	片段71	44		0.2%
72	片段72	17		0.1%
73	片段73	20		0.1%
74	片段74	56		0.2%
75	片段75	38		0.1%
76	片段76	58		0.2%
77	片段77	19		0.1%
78	片段78	16		0.1%
79	片段79	63		0.2%
80	片段80	68		0.2%
81	片段81	45		0.2%
82	片段82	16		0.1%
83	片段83	18		0.1%
84	片段84	53		0.2%
85	片段85	54		0.2%
86	片段86	61		0.2%
87	片段87	16		0.1%
88	片段88	18		0.1%
89	片段89	65		0.2%
90	片段90	45		0.2%
91	片段91	42		0.1%
92	片段92	53		0.2%
93	片段93	17		0.1%
94	片段94	18		0.1%
95	片段95	88		0.3%
96	片段96	41		0.1%
97	片段97	50		0.2%
98	片段98	17		0.1%
99	片段99	20		0.1%
100	片段100	61		0.2%
101	片段101	75		0.3%
102	片段102	72		0.2%
103	片段103	49		0.2%

104	片段104	19			0.1%
105	片段105	17			0.1%
106	片段106	17			0.1%
107	片段107	17			0.1%
108	片段108	20			0.1%
109	片段109	70			0.2%
110	片段110	94			0.3%
111	片段111	58			0.2%
112	片段112	53			0.2%
113	片段113	57			0.2%
114	片段114	24			0.1%
115	片段115	18			0.1%
116	片段116	51			0.2%
117	片段117	61			0.2%
118	片段118	122			0.4%
119	片段119	46			0.2%
120	片段120	22			0.1%
121	片段121	20			0.1%
122	片段122	68			0.2%
123	片段123	97			0.3%
124	片段124	58			0.2%
125	片段125	82			0.3%
126	片段126	66			0.2%
127	片段127	54			0.2%
128	片段128	57			0.2%
129	片段129	58			0.2%
130	片段130	39			0.1%
131	片段131	19			0.1%
132	片段132	55			0.2%
133	片段133	65			0.2%
134	片段134	24			0.1%
135	片段135	38			0.1%
136	片段136	46			0.2%
137	片段137	21			0.1%

138	片段138	54	<div><div></div></div>	0.2%
139	片段139	80	<div><div></div></div>	0.3%
140	片段140	23	<div><div></div></div>	0.1%
141	片段141	60	<div><div></div></div>	0.2%
142	片段142	51	<div><div></div></div>	0.2%
143	片段143	21	<div><div></div></div>	0.1%
144	片段144	80	<div><div></div></div>	0.3%
145	片段145	71	<div><div></div></div>	0.2%
146	片段146	28	<div><div></div></div>	0.1%
147	片段147	58	<div><div></div></div>	0.2%
148	片段148	35	<div><div></div></div>	0.1%
149	片段149	21	<div><div></div></div>	0.1%
150	片段150	93	<div><div></div></div>	0.3%
151	片段151	102	<div><div></div></div>	0.4%
152	片段152	150	<div><div></div></div>	0.5%
153	片段153	121	<div><div></div></div>	0.4%
154	片段154	139	<div><div></div></div>	0.5%
155	片段155	101	<div><div></div></div>	0.3%

原文内容

系统设计
系统设计
毕业设计(论文)
题目名称 基于SpringBoot的学校 低值易耗品领用系统的设计与开发
分院 信息工程学院
专业班级 计算机科学与技术2202班
学生姓名 丁亚洲
指导教师 姚文礼/讲师
辅导教师
时间 2025年10月至2026年5月
目录
中文摘要 I
外文摘要 II
1. 引言 1
1.1 研究的背景与意义 1
1.2 研究现状 1
1.3 研究目标与问题 2

1.4 论文结构	3
2. 文献综述	4
2.1 SpringBoot技术研究	4
2.2 Vue前端框架	4
2.3 MySql数据库技术	4
2.4 前后端分离与RESTful接口设计	5
2.5 Spring Data JPA与事务一致性研究	5
2.6 Excel批量处理与数据校验	6
3. 系统设计	6
3.1 总体设计	6
3.2 系统功能需求分析	8
3.3 系统功能模块设计	9
3.3.1 用户权限管理模块	9
3.3.2 基础信息管理模块	10
3.3.3 领用申请模块	11
3.3.4 审批管理模块	12
3.3.5 统计分析与报表模块	13
3.3.6 Excel导入导出模块	14
3.4 数据库设计	15
3.4.1 系统E-R图	15
3.4.2 数据表字段设计	16
4. 系统实现	18
4.1 登录模块	19
4.2 主页模块	20
4.3 用户申请模块	22
4.4 审批进度模块	23
4.5 管理审批模块	24
4.6 物品信息管理模块	26
4.7 批量导入导出模块	27
4.8 报表功能模块	30
4.9 智能分析功能	33
5. 系统测试	35
5.1 测试目的	35
5.2 测试方法	36
5.3 测试用例	37
6. 总结与展望	40
6.1 工作总结	40
6.2 存在的问题	41
6.3 未来展望	42
参考文献	43

致谢 45

附录 46

基于SpringBoot的学校低值易耗品领用系统的设计与开发

学 生：丁亚洲，信息工程学院

指导教师：姚文礼，信息工程学院

摘 要：伴随着高校信息化建设的不断发展，校园物资管理也从原来的非系统化、非信息化的方式转变为现在的信息化、数字化的模式。低值易耗品是指在高校教学、科研及行政管理过程中使用频繁的一类物资，管理效率好坏会直接影响到校园资源利用效率和管理水平。传统低值易耗品管理大多依靠人工记载、纸笔审核和人工统计等手段来完成工作，其工作效率低，数据易出错、库存数不清，同时还会造成管理上的混乱状况产生。因此开发出高效的稳定的信息管理系统就显得十分必要了。

本文设计并实现了以SpringBoot为底层技术的校园低值易耗品领用系统。系统以前后端分离的架构为基础，前端用Vue来创建用户界面，后端用SpringBoot来实现业务逻辑的处理，并且使用MySQL做为数据存储系统。系统主要包含用户登录管理、低值易耗品管理、领用申请管理、审批管理、数据统计、Excel数据导出等几个主要功能模块。

可以对低值易耗品申请、审批、领用全过程进行信息化管理。可以提高学校物资管理效率，降低人工操作的成本，并且有良好的数据准确性、追溯性。测试结果说明本系统运行稳定、功能齐全，可以满足校园低值易耗品管理的需要。

关键词：SpringBoot；Vue；低值易耗品管理系统；MySQL；信息化管理

The Design and Development of a School Low-Value Consumables Issuance System Based on Spring Boot

Student: Ding Yazhou, School of Information Engineering

Supervisor: Yao Wenli, School of Information Engineering

Abstract:With the continuous advancement of information technology, school administration has also started to move towards informatization and digitalization. The low-value consumable is used in the teaching activities, researching activity, and administration activity of university. They use the resources on the campus effectively based on how well they manage it. Traditional management relies heavily on manual registration and paperwork for approvals. It's just not efficient and full of mistakes.

The paper is built by springboot and i have created the entire management system for low-value consumables on campus. Based on the front end division. The front end is made with vue and the back end is done by springboot. Use mysql to store date in the syste:

Mainly has 5 parts: User Login Management, Consumable Item Management, Application Management, Approval Management, Data Statistics and Excelexport of Data. With such a system, it is possible to manage all processes of consumables from applications, approvals until they are used.

From experimental analysis we can see that the system is running well, it can also help to make consumable management in campus more efficient.

Keywords: SpringBoot; Vue; Consumables Management System; MySQL

基于SpringBoot的学校低值易耗品领用系统的设计与开发

引言

研究的背景与意义

伴随着信息技术的飞速发展，各类信息管理系统已经被应用到教育、医疗、企业等各个领域当中[1]。低值易耗品指的是单价小、使用频率高的办公用品、实验耗材、教学用品等。物资虽然单价低，但是数量多、消耗频繁，所以管理起来十分重要。

高校教学和科研活动持续增加时，低值易耗品的消耗表现为品类繁杂、周转迅速、单价便宜但是总量庞大的特点。其管理难点主要表现在三个方面上，第一，耗材采购和领用频率高，传统的纸质登记方法不能及时更新库存数据，造成库存数据滞后，容易出现账实不符的情况；第二，领用审批涉及多个角色，如果缺少统一的平台支持，就会出现审批时间长、责任不明、信

息丢失等情况；第三，耗材种类繁多，库存变动较大，人工统计费时费力，容易出现重复录入、手写错误等问题，造成数据失真；第四，在预算管理、审计要求越来越严格的情况下，高校对于耗材的流向追踪以及数据可追溯性提出了更高的要求，单靠纸质台账很难满足日常管理和监督检查的需求。

就管理目标而言，低值易耗品系统不能只做“领用登记”这一个环节的工作，必须要有从“物品建档、库存维护、领用申请、审批流转、库存扣减、统计报表、Excel导入导出”这些环节完整的闭环。利用信息化系统可以对库存的安全情况、审批办理过程以及耗材消耗趋向有随时掌控的能力，进而达成更细密化的物资管理工作。

本课题面向学校实际业务场景，以日常办公用品、实验耗材领用为出发点，对校园低值易耗品领用进行设计和实现。系统使用前后端分离架构，后端用 SpringBoot 提供业务接口，前端用 Vue 实现交互界面，数据库用 MySQL 存储业务数据，提高耗材管理效率、降低人工成本、提高数据准确性和可追溯性。

研究现状

在国内外很多高校、企业都已使用信息化系统对物资进行管理。美国以及欧洲的一些高校用ERP系统对校园物资实施了统一的管理，利用信息系统来达到库存、采购、资产等各个方面的目的。这些系统一般自动化程度较高，用数据分析技术来提高管理效率。

国内随着高校信息化建设的推进，越来越多的高校开始开发或者引进物资管理系统[10]。一些研究者提出以Web为基础的物资管理平台，用B/S结构实现物资信息的在线管理。除此之外，随着SpringBoot等轻量级开发框架的发展，很多系统开始使用它们来完成系统的开发工作，进而提高系统的开发效率以及稳定性。

但是部分系统仍然存在着单一、体验不好、扩展性差等缺陷。因此，对校园低值易耗品管理系统进行结构化、系统化、易维护的研究有一定的研究意义。

从国内外的研究和已有的系统实践来看，现有的物资管理平台在库存管理和采购管理等方面已经形成了比较完善的方案，但是在低值易耗品领用场景中还存在着一些问题，即部分系统偏重于资产管理或者大型ERP，部署和维护成本较高，难以在中小规模的院校中迅速实施；有些轻量化系统只是实现了基础的增删改查，缺少对业务流程的审批流转以及日志跟踪，使得数据无法闭环；还有些系统缺少严格的、对Excel导入导出的数据校验策略，容易造成脏数据被录入到数据库里；报表统计功能一般只做简单的列表导出，缺少按照时间区间、物品排序、状态汇总等管理决策所用的指标。

因此本文以高校低值易耗品实际管理流程为主线，从以下几个方面展开研究。一是以流程闭环为出发点来设计申请和审批机制，使状态流转清楚、过程可追溯，二是以库存数据的一致性为基础来设计审批通过之后的库存扣减逻辑，防止出现审批通过但是没有扣库存或者扣库存但是没有记录的情况，三是提供 Excel 模板、导入校验、导出等实用功能，方便日常批量操作，四是要有统计报表输出功能，便于管理人员掌握消耗趋势及库存风险。

研究目标与问题

本文以学校低值易耗品领用管理为研究对象，就传统管理方式下出现的人工登记效率低下、审批流程不明晰、库存信息更新缓慢、统计分析能力欠缺、数据追溯不便等种种问题展开设计和开发，创建起一套依靠SpringBoot支撑的学校低值易耗品领用系统。系统以学校的日常管理为依托，用前后端分离的架构来完成低值易耗品从基础信息的录入、领用申请、审核操作、库存的变动、日志的留痕、统计报表制作等全过程信息化的管理工作。

本文的研究目标主要有两点：一是实现学校低值易耗品管理业务的信息化，提高管理效率；二是形成规范的申请审批程序，使业务办理有条理、有控制；三是保证库存数据变动的数据准确性、一致性，加强数据可信度；四是有日志、报表等功能来提高管理过程的可追踪、可统计能力；五是检验 SpringBoot 技术在校园低值易耗品管理系统开发的应用成果。

本文主要研究的是怎样按照学校低值易耗品管理需求完成系统功能的设计，怎样达到申请、审批、库存三者之间的业务闭环，保证重要业务数据的一致性以及可追溯性，通过数据库的设计与报表统计来提高系统实用性以及管理价值。经过以上研究，最后得到一个比较完整的、运行稳定的、适合学校场景使用的低值易耗品领用系统。

论文结构

本文主要对基于SpringBoot的学校低值易耗品领用系统进行设计和开发，全文共分六个章节，各章节内容如下所示：

第一章是前言部分，主要论述了本课题的研究背景、国内外研究现状、研究目的及研究内容，并对全文结构做了说明。通过对高校低值易耗品管理存在问题的分析，论述本课题研究的意义以及现实应用价值，为后面系统的设计和实现提供基础。

第二章为文献综述，对本系统开发过程中需要用到的技术及有关的研究成果进行整理并加以分析。内容包括 SpringBoot 技术、Vue 前端框架、MySQL 数据库技术、前后端分离架构、RESTful 接口设计、Spring Data JPA、事务一致性以及 Excel 批量处理等。对以上技术进行研究，为系统技术选型及实现方案的提出提供理论基础。

第三章为系统设计，主要是对系统的总体设计思路以及功能设计方案进行阐述。首先对系统整体架构和设计原则进行说明，再对系统功能需求进行分析，接着将系统功能模块划分为用户权限管理、物品信息管理、领用申请管理、审批管理、统计分析以及Excel导入导出等功能模块，并根据业务流程对这些模块进行详细设计。设计系统的数据库结构，对系统数据库进行实体关系分析、各个数据表字段结构设计，为系统的实现提供数据支持。

第四章为系统实现，主要对各个功能模块的实现过程进行详细的阐述。根据系统前后端开发的实际，从登录模块、主页模块、用户申请模块、审批进度模块、管理审批模块、物品信息管理模块、批量导入导出模块、报表功能模块以及智能分析功能四个方面分别阐述实现思路、业务逻辑以及代码。本文以该系统从需求设计到功能实现为主线，介绍系统开发的过程。

第五章为系统测试，主要对系统的功能完整性、业务正确性和运行稳定性进行测试与分析。首先说明测试的目的与方法，再根据系统核心业务流程设计测试用例，对登录认证、物品管理、领用申请、审批处理、库存扣减、审批进度查询、Excel导入导出以及报表统计等主要功能进行测试，根据测试结果分析系统是否达到预期目的。

第六章为总结与展望，主要对本文所完成的工作进行总结，归纳出系统设计和实现中取得的成绩，分析目前系统还存在的安全问题、并发处理问题、功能拓展性问题等，根据实际情况提出未来改进的方向和发展预测。

文献综述

SpringBoot技术研究

SpringBoot是由Pivotal团队基于Spring框架推出的快速开发框架，它的主要目的就是简化传统Spring应用的开发和配置过程[16]。按照“约定优于配置”的理念，SpringBoot可以依照项目的依赖自行完成大部分的基础配置，大大简化了开发的工作量。此框架自带Web服务器(Tomcat)，使得应用可以不必额外部署就可独立运行，并且有诸多Starter依赖可以简化项目构建过程。开发过程中，SpringBoot可以大幅度提高开发速度、减少XML配置文件的应用、通过统一项目结构改善代码可读性、可维护性。此外，SpringBoot具备良好的扩展性与生态支持，能够与Spring Data JPA、Spring Security等组件无缝集成，适用于构建稳定、高效的Web应用系统[13]。因此，在本系统开发中选择SpringBoot作为后端开发框架，可以提高系统开发效率、保证系统的稳定性和可扩展性。

Vue前端框架

Vue是渐进式的JavaScript框架，用来构建用户界面[9]。Vue具备组件化开发、数据双向绑定和虚拟DOM等特性，从而使得前端开发变得更加高效，用户体验也得到改善。

Vue框架在Web管理系统开发中应用较广，在前后端分离架构中，Vue能够与后端的RESTful接口实现高效的交互。

MySql数据库技术

数据库属于信息管理系统的重要组成部分，主要用来存储备份系统运转时所产生的数据。Web应用开发中关系型数据库管理系统被广泛使用，MySQL数据库因为开源、稳定、性能好等特性而被广泛使用[11]。

MySQL数据库使用标准SQL语句来执行数据的增删改查，有较好的并发处理能力。在Web应用系统当中，MySQL一般会与Java的后端框架结合起来使用，借助ORM框架或者数据访问层来完成数据库的操作。

校园物资管理系统当中，数据库设计就显得尤其重要。合理的数据库结构能保证系统数据的完整性、一致性，也可以提高系统的查询速度。因此在系统设计阶段需要根据业务需求来设计合理的数据表结构，建立各个数据表之间的关系。

前后端分离与RESTful接口设计

前后端分离属于一种比较普遍的Web系统架构模式，它把界面呈现同业务逻辑分离开来，前端担当交互和页面渲染的任务，而后端承担起业务规则，数据处理以及安全性方面的职责[14]。该模式可以提高团队的协作效率，前端可以独自做页面的迭代

工作，后端也可以只顾着做好接口的稳定性和数据的一致性。校园低值易耗品管理系统采用前后端分离的方式有利于后期添加新增加的功能，例如新增报表、导入导出等，并能满足系统部署和运维的要求。

RESTful 风格接口强调资源（Resource）与统一的 HTTP 方法语义，例如 GET 用于查询、POST 用于新增、PUT用于修改、DELETE 用于删除。合理设计URL，使用约定好的参数可以保证接口的可读性、可测试性以及可扩展性。本文系统中的物品管理、领用申请、审批处理以及报表统计等各个功能都可以被抽象为对资源的操作，采用RESTful风格可以提高接口的一致性，减小前端的调用复杂度。

另外，为了便于前端统一处理响应结果，在实际工程中一般都会用统一的返回结构，对于异常统一封装并给出提示，从而使得前端页面无法显示难以理解的错误信息，提高用户体验。

Spring Data JPA与事务一致性研究

Spring Data JPA是Spring生态中常用的实现数据访问的方案，通过使用方法命名的方式就可以得到常见的查询，可以降低样板代码的产生，提高开发效率。对本系统来说，物品、用户、申请单、申请明细和审批日志之间具有典型的“一对多/多对一”的关系，用JPA可以以较低的成本实现实体映射和持久化操作。

业务一致性的体现就是执行“校验库存-扣减库存-更新申请状态-写入审批日志”一系列的操作，该类操作要求是强一致的。若其中某一步失败（库存不足），则保证系统状态回滚，防止出现部分扣减库存或者状态已更改但是库存未扣的数据不一致情况。因此在工程实现的时候一般会使用事务机制(@Transactional)来保证多个数据库操作的原子性。本文系统中虽然使用了分步骤校验和扣减的方式，但是也可以通过引入事务控制来提高一致性、并发安全性。

Excel批量处理与数据校验

在校园管理中，Excel依然是使用最广泛的数据交换载体之一，新学期物品初始化、批量修改库存、批量导入用户账号等都是其应用场景。基于 Web 系统具有Excel导入导出的功能，从而大大减少管理人员的操作成本。Apache POI是Java生态中常用的Office文档处理库，可以用来读取写入xlsx文件，并且可以控制xlsx文件的格式，适合生成模板文件、导出报表以及解析上传文件。

Excel 导入的难点就是数据质量控制。由于Excel单元格会出现空值、字符和数字混用、格式不一致的情况，因此系统应该使用DataFormatter这样的数据读取方式，并对必填项、数值区间、角色合法等作出检查。对于一行不符合条件的数据，尽量做到记录错误信息但是不影响全部导入，从而提高批量处理的速度。本文在用户导入和物品导入模块中都有了inserted、updated、skipped、errors等统计结果，方便管理员迅速找到导入问题。

系统设计

总体设计

系统使用经典的B/S架构，完全按照前后端分离的原则来建站[8]。前端表现层使用 Vue 3 + Vite + Axios 构建响应式页面，并与后端API进行交互，数据层采用MySQL数据库实现事务的保证和并发处理，后端用Spring Boot框架设计成controller-service-repository的三层模式，并采用JPA进行数据的操作，在service中定义业务服务的方法，在repository里定义数据仓库接口。

整体架构可分为三层：

表现层即用户的交互界面，对表单进行验证并显示数据可视化信息，采用路由和实时刷新技术。

应用层主要负责业务逻辑处理工作，库存校验、审批状态机以及统计查询等都用到了Spring Boot的自动配置和Starter组件。

数据层主要是对实体进行映射和持久化，使用JPA自动生成了DML和DDL，有懒加载和事务管理。

分层设计具有高内聚低耦合的特点，便于维护和扩展，用CORS配置解决跨域问题，适合实验室或者办公环境的实际使用。就结构上来说，该类耗材管理的结构相似性较高，在库存和申请程序等环节都存在运作情况。

在设计原则方面，系统遵循以下思路：

- （1）模块化，把登录、物品管理、申请管理、审批管理、导入导出、统计报表等拆分为独立模块，降低耦合度；

- (2) 一致性：关键业务过程（如审批通过扣库存）需要保证数据一致性与可追溯性；
- (3) 易用性：面向管理人员与普通用户，界面操作应尽量简单清晰，减少学习成本；
- (4) 可维护性：接口与数据库设计尽可能规范化，便于后续新增业务字段与功能扩展。

系统主要的数据流为：用户登录之后查询物品并提交领用申请，系统会生成申请单和明细，申请状态是待审批；管理员查看待审批列表并进行审批；审批通过之后系统扣减库存，并写入审批日志；最后利用进度查询和报表模块实现过程跟踪和统计分析。

系统总体架构图如图1所示：

图1 系统架构图

系统功能需求分析

本系统为校园低值易耗品管理系统，实现物品信息登记，领用申请，审核发放，库存管理，数据分析等全部过程的信息化管理。系统应该有用户登录和权限设置，普通用户可以查看物品、提交领用申请、查看个人申请记录，管理员可以增删改查物品、库存和安全库存、审批处理和报表导出。业务流程中申请单要具备清楚的状态流转（待审批、已通过、已驳回），审批通过之后需要做库存充足性的校验并自动扣除库存，还要留下审批日志以便查询。从数据上要保证库存变动的准确性、一致性，提供按时间区间领用统计、状态统计、Top消耗物品分析等功能，支持Excel导入导出提高数据整理、上报效率。

系统总体功能结构图如图2所示：

图2 总体功能结构图

系统功能模块设计

用户权限管理模块

系统设有登录认证以及角色权限控制，分为管理员和普通用户两种身份。

主要功能为用户登录/退出、用户信息修改（导入/导出）、角色分配、启用/禁用账号。关键规则有如下几点，登录需要用户名和密码，返回会话标识（token）、用户角色信息；管理员可以访问管理端功能（审批、物品维护、报表），普通用户只能申请和查询。

异常处理：密码、账号错误或不存在时返回明确提示。

登录流程图如图3所示：

图3 登录流程图

基础信息管理模块

基础信息管理模块主要是对物品台账信息进行管理的工作，它是业务活动数据的主要来源。管理员可以对物品进行新增、修改、删除和查询，可以按照关键字检索快速找到物品；物品信息包含名称、规格型号、单位、库存数量、安全库存等主要字段，保证物品数据结构化、标准化，为后续申请、审批、统计、报表导出提供准确的数据支撑。

物品信息管理流程图如图4所示：

图4 物品信息管理流程图

领用申请模块

领用申请模块主要是普通用户通过该模块提交领用申请，系统使用申请单主表和申请明细子表的结构来保存数据，一次可以申请多种物品。申请单包含申请人信息、备注说明、创建时间以及状态（默认为待审批），申请明细中包含各类物品的数量等信息；用户可以查看个人申请列表和详情，实现领用业务的线上化以及可追踪。

用户申请流程图如图5所示：

图5 用户申请流程图

审批管理模块

仓库分区模块主要就是管理员在审批模块里查看待审批的申请单及其明细，然后对申请进行通过或者驳回的操作。系统中建立了清晰的状态转换机制（待审批、已经通过、已经被驳回），只允许处在待审批状态下的申请可以被处理，防止出现重复

审批的情况发生；审批结果会带有意见说明部分，有利于申请人理解原因及后续的调整工作，进而产生规范的领用审批流程。

管理审批流程图如图6所示：

图6 管理审批流程图

统计分析报表模块

统计分析和报表模块主要是为管理者提供面向统计分析的功能，可以对申请的总量、通过的个数、驳回的个数以及待审批的个数进行汇总统计，也可以对物品消耗的排行和消耗的趋势进行统计分析。统计结果可以直观地反映出物资使用情况，为采购补充、库存优化、管理决策等提供依据，也可以和Excel导出结合起来形成可归档的管理报表。

统计分析与报表流程图如图7所示：

图7 统计分析与报表流程图

Excel导入导出模块

本模块主要用以提高数据初始化和日常上报的效率，系统加入了Excel导入导出的功能。导出库存报表、申请记录、账号信息等数据一键生成xlsx文件，导入支持批量导入用户、物品数据，对必填项、数据类型、角色合法性等进行校验，错误信息可以返回行号和原因，降低人工录入成本，提高数据处理效率。

导入导出流程图如图8所示：

图8 导入导出流程图

数据库设计

本系统使用MySQL作为关系型数据库，数据库的设计目标是保证系统的数据结构清晰、冗余控制、查询高效、可追踪、支持领用业务流程（申请-审批-扣库存-日志-统计-报表导出）。

系统E-R图

系统E-R图就是用来表示实体类型、属性以及关系的一种方法。可以采用矩形来表征实体，用椭圆型表示属性信息，用菱形表示联系。用图型可以直观地反映这三个量之间存在怎样的联系，便于我们理解。

系统E-R图如图9所示：

图9 系统E-R图

数据表字段设计

整个系统分为多个数据表，每个数据表都很重要，系统相关操作都会写入数据表中，主要包括用户信息表、物品信息表、领用申请表、领用申请明细表、申请日志表，具体的如图1.2所示：

用户信息表如表1所示：

表1 用户信息表

字段名称	数据类型	主键	是否为空	备注
------	------	----	------	----

UserId	bigint	是	否	用户编号
--------	--------	---	---	------

UserName	varchar(50)	否	否	登录账号
----------	-------------	---	---	------

Password	varchar(255)	否	否	用户密码
----------	--------------	---	---	------

RealName	varchar(50)	否	是	用户姓名/昵称
----------	-------------	---	---	---------

Role	varchar(20)	否	否	角色
------	-------------	---	---	----

UserRole	int	否	否	用户状态
----------	-----	---	---	------

CreateTime	datetime	否	否	创建时间
------------	----------	---	---	------

IsDeleted	int	否	否	删除标识
-----------	-----	---	---	------

物品信息表如表2所示：

表2 物品信息表

字段名称	数据类型	主键	是否为空	备注
------	------	----	------	----

ItemId bigint 是否 物品编号
 ItemName varchar(100) 否 否 物品名称
 ItemModel varchar(100) 否 是 规格/型号
 Unit varchar(20) 否 否 单位
 Stock int 否 否 当前库存数量
 IsDeleted int 否 否 删除标识
 SafetyStock int 否 否 安全库存
 CreateTime datetime 否 否 创建时间
 UpdateTime datetime 否 是 更新时间

领用申请表如表3所示:

表3 领用申请表

字段名称 数据类型 主键 是否为空 备注
 OrderId bigint 是否 申请单编号
 UserId bigint 否 否 申请人编号
 UserName varchar(50) 否 否 申请人账号
 Status varchar(20) 否 否 申请状态
 Remark varchar(255) 否 是 申请说明/备注
 CreateTime datetime 否 否 申请时间
 IsDeleted int 否 否 删除标识

领用申请明细表如表4所示:

表4 领用申请明细表

字段名称 数据类型 主键 是否为空 备注
 DetailId bigint 是否 明细编号
 OrderId bigint 否 否 申请单编号
 ItemId bigint 否 否 物品编码
 ItemName varchar(100) 否 否 物品名称
 Quantity int 否 否 申请数量
 IsDeleted int 否 否 删除标识

审批日志表如表5所示:

表5 审批日志表

字段名称 数据类型 主键 是否为空 备注
 LogId bigint 是否 日志编号
 OrderId bigint 否 否 申请单编号
 AdminId bigint 否 否 审批人编号
 AdminName varchar(50) 否 否 审批人账号
 Result varchar(20) 否 否 审批结果
 Comment varchar(255) 否 是 审批意见
 ApproveTime datetime 否 否 审批时间
 IsDeleted int 否 否 删除标识

系统实现

系统实现阶段就是根据前期的需求分析、总体设计和数据库设计，在具体实现各个功能模块的过程中进行的。本系统以用户申请、管理员审批、库存扣减、日志留痕、统计分析为主线，用前后端分离的方式对整个工程进行组织。前端使用 Vue 来构建用户交互页面，对表单进行输入，数据进行展示，实现页面的跳转以及结果的反馈；后端用 SpringBoot 提供 RESTful 接口，对身份进行认证，业务规则进行校验，数据进行持久化以及统计分析；数据层使用 MySQL 储存业务数据，并且通过实体映射方式来关联对象与数据表。整个系统的重点放在了模块之间衔接的关系上，核心的业务规则所造成的约束以及数据处理时的各个方面的可重复利用和一致性。

系统在工程实现上采用分层的设计思想，用控制器、服务层以及数据访问层对后台的代码进行组织。控制器层主要是接收前端请求、解析参数后得到统一格式的响应结果，服务层是对核心业务逻辑进行封装，是系统规则控制和模块联动的中心部分，repository 层是数据持久化的处理，使用 JPA 对常见的查询、更新以及统计等操作进行执行。前端使用组件化的架构来组织页面，用路由来区分普通用户端和管理员端的视图，并且采用统一的封装的请求模块同后端接口相联系。为了保证接口返回结果的规范统一，系统定义了统一的响应结构，用来封装状态码、提示信息和业务数据，使得前端可以在各个模块中使用相同的方式来完成结果的处理以及异常提示。

从业务控制角度来说，本系统所达成的是对各个页面、各个接口的完成不是单一环节，而是由一个个实际领用过程组成闭环。用户提交申请之后，系统不仅要产生申请主表的数据，还要产生申请明细数据，并且把状态初始化成“待审批”。管理员审批通过以后，系统不能只对状态字段进行更改，而必须同步完成库存校验、库存扣减和审批日志的记录。统计与报表模块在上述业务数据累积的基础上，需要对申请记录、审批结果和物品消耗情况实施聚合分析。因此系统实现阶段实际上是将业务流程的技术化、规则化的过程，它不仅仅是“实现功能”，更是保证流程正确、数据可靠、结果可追溯的过程。

登录模块

登录模块是系统运行的入口，它的主要任务就是完成用户的身份认证、角色的识别以及会话上下文的建立，为后面的功能访问做好基础。系统前端对用户名、密码进行收集之后，向后台认证接口发送请求。后端首先对输入参数做非空校验以及合法性检查，之后按照用户名来查询用户信息，再对密码进行匹配检查。当账号不存在、密码错误或者账号处于禁用状态的时候，系统会给出清楚的错误提示信息，前端按照响应的结果来结束后面的跳转，并且向用户显示反馈，防止无效的登录进入系统内部。

认证成功之后，系统就会把用户的编号、登录账号、姓名以及角色这些重要信息发送到前端，从而创建起当前的会话上下文。角色是实行权限边界划分的主要依据。普通用户登录之后会直接进入面向申请业务的功能界面，可以对物品进行查询、填写领用申请和查看审批进度；管理员登录之后进入管理界面，可以对物品进行维护、批准事务、导入导出和统计分析等。系统经由在登录环节对角色展开识别并划分访问边界之后，可从前端就达成功能权限隔离工作，从而缩减越权访问的可能性。

另外为了提高登录模块的可维护性、可扩展性，在实现过程中把认证逻辑和普通业务接口分开处理。登录接口只做身份校验和基本信息返回，不承载复杂的业务逻辑，有利于后面接更加完善的身份认证方式如基于Token的会话控制、统一拦截器认证、JWT认证等。总体上来说，登录模块既是系统的访问入口，又是之后的角色分发、权限管理以及业务隔离的根基环节。

系统登录页面实现效果如图10所示：

图10 系统登录页面效果图

系统登录页面模块的主要实现代码如下所示：

```
data.put("role", user.getRole());
data.put("token", token);
return ApiResult.ok(data);
```

主页模块

主页模块是系统的业务入口组织和角色化的导航，在页面展示的同时还具有把不同的身份用户直接导向各自的业务区域的功能。系统会依照登录后的用户角色动态显示主页的内容，使得普通用户和管理员进入系统之后看到不一样的菜单结构和功能入口。普通用户的主页是基于查看物品、提交申请、查询进度三个功能展开的，比较简单直观；管理员的主页则是以管理物品、

审批处理、报表统计、批量导入导出为主，侧重在业务办理上的统一操作。

从技术实现的角度来说，主页模块用路由和组件联动的方式来完成各个业务区的切换。前端根据用户的权限来显示不同的菜单项以及相应的页面组件，以降低无关功能出现而引起的操作干扰。主页的数据展示不是静态的，它是后端接口返回的业务数据在页面上动态渲染，首页物品列表、库存信息、待审批数量等等都是这样的形式。采用这样的实现方式以后，主页不再是简单的导航页，而成了系统业务状态的总览页，有利于用户和管理员在进入系统之后马上知道当前的业务状况。

从业务流程角度来讲，主页模块是将登录认证的结果和后面的业务逻辑功能调用进行联系的模块，使系统内部的功能入口更加集中。用户端主页是领用申请流程的起始点，管理员主页是审批、维护、统计流程的进入口。所以，主页模块在整个系统中起到的是流程分发、操作聚合的作用，是整个业务闭环高效运转的重要支撑部分。

主页模块的效果图如图11、图12所示：

图11 管理员主页模块效果图

图12 用户主页模块效果图

主页模块的主要实现代码如下所示：

```
public ApiResult<List<Item>> list(@RequestParam(required = false) String keyword)
return ApiResult.ok(itemService.list(keyword));
```

用户申请模块

用户申请模块属于系统业务运转的开端部分，它的主要工作就是把用户的领用需求转变成系统的业务数据，为后面的审批处理赋予全面的支撑。考虑到实际领用场景下一次申请一般会涉及多种物品，所以系统用“申请主表+申请明细表”这种结构来组织数据。申请主表主要用来保存申请人的身份信息、申请时间、备注说明和申请状态等整体属性，申请明细表用来记录每一种申请物品的编号、名称、数量等具体内容。该设计同时满足数据库规范化的要求，并且给后续的审批、统计和报表模块的数据拆分以及聚合工作提供了一个很好的数据支撑。

在实现过程中，用户提交申请的时候，后端会对前端传入的申请数据进行合法性校验，即申请项目是否为空、申请数量是否为正数、所选物品是否存在等。校验通过之后，系统先保存申请主表信息，再分别写入申请明细，最后将每一条明细和对应的申请单编号进行关联。系统将新提交的申请状态统一初始化为“待审批”，保证申请数据一经产生就可以直接进入审批程序。采用这样的先主后从数据落库顺序，可以使得系统的申请记录完整、有联系，不会造成明细数据孤立或者主从关系丢失。

从业务逻辑上来说，用户申请模块除了可以完成数据的录入之外，还能将领用需求进行规范化的表述。系统以结构化方式保存申请数据，使得以后审批模块可以按照申请单维度去查看完整的业务信息，统计模块可以按照物品维度去汇总消耗趋势，审批进度查询模块可以面向用户展示清晰的申请处理过程。因此，用户申请模块在整个系统中起着承前启后的作用，是用户需求 and 后台管理之间联系的纽带。

用户申请模块效果图如图13所示：

图13 用户申请效果图

用户申请模块的主要实现代码如下所示：

```
ApplyOrder order = new ApplyOrder();
order.setStatus("PENDING");
orderRepository.save(order);
ApplyItem ai = new ApplyItem();
ai.setOrderId(order.getId());
itemRepository.save(ai);
```

审批进度模块

审批进度模块主要是对普通用户，可以解决传统的线下管理中，申请提交之后状态无法查看，处理结果难以追踪的问题。系统将申请单的状态字段同审批日志的数据一起形成申请处理轨迹，从而让用户可以查看自己提交的申请处在何种状态，并知

道申请在审核过程中出现的关键处理信息。该模块实现可以提高业务透明度，也可以减少由于用户对审批信息不明造成的重复咨询、重复提交的情况。

系统在功能上采用用户编号来区分各个申请数据，在查询时只会显示自己提交的申请记录，不会显示其他用户的申请信息。查询结果一般由申请单基本信息、申请明细信息和审批日志信息三大部分组成。其中申请单信息体现总体情况，待审批、已通过、已驳回均属于其类别，申请明细显示申请的物品及数量，审批日志记载审批人、审批结果、审批意见和审批时间等过程性信息。用这种主信息和过程信息相结合的方式呈现申请从提交到处理完毕的全部过程。

就系统的实现而言，审批进度模块加强了业务流程可视化展示的效果，使系统由原来的“申请提交工具”变成现在的“全过程跟踪平台”。普通用户对该模块有更高的感知度，对于整个系统来说，该模块借助日志联动加强了业务留痕能力，有利于问题排查、责任划分及流程审计。

审批进度模块效果图如图14所示：

图14 审批进度模块效果图

审批进度模块的主要实现代码如下所示：

```
List<ApplyOrder> orders = orderRepo.findByUserIdOrderByCreateTimeAsc(userId);
List<ApplyItem> items = itemRepo.findByOrderId(o.getId());
List<ApproveLog> logs = logRepo.findByOrderId(o.getId());
```

管理审批模块

管理审批模块是系统业务闭环中的关键控制节点，它的功能主要是根据用户所提交的申请信息作出审批决定，在审批通过之后再行进行库存联动操作。与一般的修改数据的行为不同的是，审批行为本质上是一个带有多业务约束的过程控制环节，它的结果会直接导致申请状态、库存数据和日志信息等发生同步的变化。因此本模块在实现过程中着重对状态流转控制、库存一致性和审批过程留痕加以考虑。

系统管理员进入审批模块以后，先按申请状态将待处理的申请记录选出，可以查看申请单和申请明细。管理员核对完申请内容之后可以执行通过或者驳回两种操作。为避免由于重复审批导致的数据异常，在处理审批请求之前会先判断申请当前的状态是否仍然为“待审批”。只有处在审批状态的申请才能继续下一步的操作，若在提交之后立刻被拒绝或者被退回，则系统就会提示用户错误的信息并终止操作流程。可以避免同一个申请因为页面重复提交造成状态混乱的情况出现。

驳回的业务处理方式比较直接，主要对已提交申请的状态进行修改，将审批日志记录下来即可。当场景发生时，就必须要做更严格的业务校验以及联动更新。系统逐行读取申请明细所对应的物品库存信息，对每一项物品进行库存是否足够检查。只有所有申请明细都达到库存标准之后，系统才会对库存进行扣减、状态更新以及日志记录等工作。若任一物资库存不足，整个审批流程就会立即停止，回退给前端明确告知信息。该种的先校验后扣减再更新状态最后写日志的处理方式，可以有效地防止出现已经扣减但是部分成功的情况，提高审批处理的一致性可靠性。

从数据库操作的角度来讲，审批模块主要是对申请表状态字段进行改变，对物品库存数量进行更新，并且记录审批日志。由于这些操作在业务上对原子性有要求，因此该模块的实现应该尽量做到统一处理、有序约束，必要时还可以加入事务控制手段来加强系统在异常情况以及高并发环境下对数据一致性的保证。从整体上看，管理审批模块属于系统规则最集中、业务约束最严格的模块，其实现的质量好坏直接影响到整个系统的数据可信度和流程的规范性。

管理审批模块的效果图如图15所示：

图15 管理审批模块效果图

管理审批模块的主要实现代码如下所示：

```
if (!"PENDING".equals(order.getStatus()))
    throw new RuntimeException("不能重复审批");
if (it.getStock() < d.getQuantity())
    throw new RuntimeException("库存不足：" + it.getName());
```

物品信息管理模块

物品信息管理模块属于系统业务运行的根基数据支撑模块，主要作用是对低值易耗品实行标准化的台账信息管理，给后续申请提交、审批处理、库存核算以及报表统计等各项工作赋予一致的数据来源。该模块由管理员负责维护，可以对物品的新增、修改、删除以及查询等基本操作进行控制，并且会维护好库存数量、安全库存等重要字段的信息，使得物品的数据一直处在完整的、正确的、可以使用的状态之中。

系统会把物品名称、规格型号、单位、当库存、安全库存这些字段当作基本属性来统一保存。新增、修改操作既要保证字段本身的完整性，也要保证业务语义上合理，库存数量、安全库存要满足数值的合法范围，物品名称等标识信息要尽量保持一致。删除操作适合使用逻辑删除的方式，用删除标识控制记录的可见性，防止直接物理删除造成的数据追溯困难和历史记录关联失效的问题。查询操作可以按照关键字进行搜索，有利于管理员对大量物品信息进行查找定位，从而加快后台维护的速度。

从业务联动的角度来看，物品信息管理模块虽然只属于基础数据维护，但是它与很多重要的业务模块有密切联系。用户申请的时候，会根据这个模块给出的物品信息来挑选；审批通过之后还要依靠库存字段来完成库存减少的任务；统计分析的时候还得靠物品维度来进行汇总消费情况的处理。所以，物品信息管理模块不是孤立的增删改查页面，它是系统正常运行的底层数据基础模块，它的实现是否规范、准确会直接影响到后面所有的业务处理结果。

物品管理模块的效果图如图16所示：

图16 管理审批模块效果图

物品信息管理模块的主要实现代码如下所示：

```
return ApiResult.ok(itemService.list(keyword));  
return ApiResult.ok(itemService.update(id, item));
```

批量导入导出模块

批量导入导出模块是解决系统实际应用过程中出现的基础数据初始化工作量大、日常数据维护频繁、报表资料需要统一留档等问题的模块。相对于逐条手工录入的方式来说，Excel批量处理更契合管理场景下高效率、低重复劳动的实际情况。因此系统将模板下载、批量导入、批量导出以及导入结果反馈等几个功能结合在一起，使管理员可以很方便地对用户的资料、物品的台账、业务数据等进行维护和整理。

在导入功能的实现中，系统会先给出一个标准的Excel模板来规定字段的排列顺序、必填项以及数据的格式等，从而降低由于源数据不规范而引起的导入失败的风险。管理员上传文件之后，系统会把文件解析组件按行逐行读取数据，并且会对每行的内容进行字段级和业务级的双重校验。字段级校验主要是对空值、数据类型、格式规范等基本问题进行检验；业务级校验则是对角色取值合法性、库存数量合理性、账号重复性、物品是否存在等规则进行检验。检测出的错误数据不会中断整个导入过程，只会把异常行号以及错误原因保存起来，并且会与成功的导入条数、更新的条数、被跳过的条数这些统计结果一起返回。既可以提高批量处理速度，又可以提高问题定位准确率。

导出功能主要是把系统内用户的用户数据、物品数据、申请记录、统计结果以指定格式导出为Excel文件，供管理员归档、报告使用以及进行在线之外的分析。根据不同的导出类型，会调用对应的数据库查询逻辑，将结果按照统一的表头、字段顺序导出成规范的文件。为了使导出结果更易阅读，在文件内容中会尽量保持字段的语义清楚、结构整齐，并对重要时间字段、状态字段做一定的格式化。

从总体上看，批量导入导出模块可以提高系统数据维护效率，也可以加强系统同现实办公流程之间的衔接能力。使系统既可以承担在线管理功能，又能满足批量初始化、集中维护、资料归档等现实业务需要，提高系统的实用性和推广性。

批量导入导出页面的效果图如图17所示：

图17 批量导入导出模块效果图

批量导入模块的效果图如图18所示：

图18 批量导入模块效果图

批量导出模块的效果图如图19所示：

图19 批量导出模块效果图

报表生成模块的效果图如图20所示：

图20 报表生成模块效果图

批量导入导出模块的主要实现代码如下所示：

```
String username = df.formatCellValue(r.getCell(0)).trim();
if (username.isEmpty() || name.isEmpty() || pwd.isEmpty()) {
    throw new RuntimeException("必填缺失");
}
if (!role.equals("ADMIN") && !role.equals("USER")) {
    throw new RuntimeException("角色非法");
}
```

报表功能模块

报表功能模块是在业务数据不断积累的基础上构建的综合分析模块，主要作用就是把申请、审批、库存、消耗等运行数据转化为可以用于管理决策的统计结果。相较于基础业务模块只注重流程处理的模式而言，报表模块更看重历史数据的汇总、分析以及表达的作用，它主要用以使管理者从总体上把握系统的运行状况，找出物资使用的规律，并给采购的补充、库存的改善及制度的调整赋予量化的支撑。

系统具有按时间段区间对领用业务做多种统计的功能。首先对申请单总数进行汇总，再将之拆分为待审批的数量、审批通过的数量以及审批驳回的数量，以此来表现某一时间段内业务处理的规模和审批执行的情况。其次，根据申请明细数据可以统计出各种物品的领用量，按数量或次数排列，得到高频消耗物品排行表，找出重点消耗物品。其次可以按照日期进行分类整理，形成每天的消耗趋势数据，可以反映各个时间段内物品的领用状况，进而为采购计划阶段性的制定、教学实验周期的分析或者办公消耗高峰期的识别等提供数据支撑。

为了提高报表模块的分析价值，在实现的时候，并不是将所有的申请记录进行简单的累加计算，而是按照业务语义来明确统计口径。一般情况下，只有审批通过的申请才会对实际库存造成影响，因此消耗类统计应该以已经提交申请为有效数据，不能将已经驳回或者未处理的数据算入实际消耗数据中。对统计口径加以限制，可使系统所给出的报表成果更具真实性并且更能具备管理方面的借鉴意义。基于业务规则的聚合分析方式，就是报表模块和普通列表展示有区别的地方。

在报表展示形式上，系统不但可以给出基本数值统计的结果，而且可以根据不同的分析目的将更多的业务描述组织起来。管理方面可以显示申请规模、审批速度、被拒绝的比例、库存压力；物品上能显示出高频率领用的物品、低周转的物品、安全库存临界点上的物品以及季节性消耗增加的物品；时间上可以从日、周或者月度数据上进行比较，找出业务高峰期以及资源消耗的规律。经过上述功能扩展之后，报表模块就从单纯的用来输出数据的工具，变成可以进行业务分析的工具了。

报表模块还可以和导出功能结合在一起，将统计结果以Excel文件的形式导出，方便管理部门进行归档、上报或者二次分析。从后面扩展的角度来说，本模块可以再加入图表可视化展现，分类统计，同比环比比较，库存预警联动以及审批时效分析等特性，进而使得整个系统具有更加强大的管理决策支撑作用。报表功能模块是系统从基础业务管理向数据驱动分析转变的标志，也是论文中可以体现系统综合应用价值的部分。

报表功能模块效果图如图21、图22、图23所示：

图21 报表功能模块效果图

图22 报表功能模块效果图

图23 报表功能模块效果图

报表功能模块的主要实现代码如下所示：

```
data.put("topItems", applyItemRepo.statTopItems(st, et));
data.put("daily", applyItemRepo.statDaily(st, et));
```

```
List<ApplyOrder> all = applyOrderRepo.findByCreateTimeBetween(st, et);
data.put("approved", all.stream().filter(x -> "APPROVED".equals(x.getStatus())).count());
```

智能分析功能

在传统的统计分析的基础上，系统又增加了智能分析的功能来加强系统对于库存状态以及业务数据变化的解释能力。本功能不会对原来的业务流程进行改变，只是在已有的数据基础上对信息进行更高的层次上的提炼，从而使系统不但能显示结果，还能对结果作出说明，进而提高管理人员对于数据的了解速度以及做出决定的速度。

智能分析模块主要是库存预警分析、报表总结分析这两块。库存预警分析以物品的当量库存、安全库存为依据，来实现自动判定库存状况。系统一旦发现某种物品的库存少于安全库存、临近安全库存或者出现缺货的情况，就会发出有关的预警提示信息。智能分析不能只简单地把数字展示出来，而应该把数据的状态转变成更有助于理解的文字说明，即哪些物品有短缺的风险，哪些物品需要先补齐，哪些物品会影响到后面的教学或者工作使用。这样有利于管理人员及时发现风险的关键点，使库存管理更加积极主动。

报表总结分析是建立在统计模块输出结果之上的。系统会先将指定时间段内所申请的总数量以及被审核过的成功率为多少，驳回情况如何，重点使用的物品有哪些，在哪个时间段上消费最多等进行结构化的整理，然后用规则化的方法对这些信息进行归纳，并将其变成文字符号。系统可以从业务规模的变化、审批处理情况、物品消耗集中度和库存风险趋势等各方面得出综合分析结论，并给出简单的管理建议。因此管理员看报表的时候既能看到具体数字又可以理解这些数字背后所表示的意义，使得数据结果更具有业务可读性。

就实现的意义而言，智能分析功能属于系统由“信息记录型系统”转向“辅助决策型系统”的重要表现。它把系统原来已有的数据采集与统计能力又往解释层、决策层方向推进了一步，使系统在实际管理中应用得更深更广。虽然目前实现还以规则分析、文本化总结为主，但是其设计思路已经为以后接入更复杂的模型、智能推荐以及风险预测留有余地。

智能分析功能模块效果图如图24、图25所示：

图24 AI预警分析效果图

图25 AI报表总结模块效果图

智能分析功能模块的主要实现代码如下所示：

```
public String buildWarningAnalysis(List<Item> items) { ... }
public String buildReportSummary(LocalDate start,
LocalDate end,
List<ApplyOrder> orders,
List<ItemConsumeStat> topItems) { ... }
```

系统测试

测试目的

低值易耗品领用系统的系统测试目的主要是保证整个系统的整体功能正常、准确地工作，在实际的使用过程中给用户和管理员提供一个良好的操作体验。由于本系统分为用户端和管理员端两种角色，并且核心业务包含申请、审批、扣库存、进度查询、统计报表、Excel导入导出等环节，所以需要通过系统测试来检验各个功能模块之间是否存在衔接不合理的地方，数据是否一致，异常处理是否完善。对普通用户来说，系统应当在登录之后能够进入到用户的主页，进行物品的查询和选择，提交领用申请，查看申请的过程和结果等操作，提示要明确，状态变动要及时；对管理员而言，在管理员主页可以完成物品信息的维护，待审批申请的处理，审批通过后的库存校验与扣减，审批日志的记录，批量导入导出，消耗统计报表的查询等功能，保证管理流程准确并且可追溯。经过系统测试，可以找到并改正可能存在的功能缺陷、数据错误等问题，例如账号密码错误处理、重复审批拦截、库存不足时提示回滚、Excel导入格式异常、统计报表时间区间计算错等问题，提高系统可靠性和可使用性，使系统可以满足校园低值易耗品领用管理的基本需求，并且能够稳定地上线运行。

测试方法

在进行系统测试时，可以掌握一些测试的方法来对系统进行验证和评价。本系统主要用到手工测试和机器测试相结合的方式，增加测试覆盖率以及问题定位的效率，进而更加全面的保证系统的各项功能是正确的、稳定的。

手动测试指的是测试人员按照系统的实际需求以及业务流程，有条不紊地操作系统的各个界面来检验系统各项功能是否能够正常工作。测试时重点模拟真实使用场景，即用户端从登录到主页、选择物品、输入数量、提交申请、查看申请审批进度/我的申请，在申请审批进度/我的申请中查看状态的变化，管理员端从登录进入管理主页、查看待审批列表、进入申请明细核对信息、执行审批通过或者驳回操作、查看审批通过后库存是否正确的扣减、申请单状态是否正确的流转、审批日志是否写入成功；异常场景的反复验证，数量填写为0或负数、库存不足、重复审批、导入文件缺少字段等，系统给出明确提示且保持数据不被破坏。手动测试的优点是直观、接近真实用户的操作方式，可以很快地发现页面交互和业务流程上存在的问题，但是缺点就是执行效率低，重复场景的回归测试成本高。

机器测试主要是对系统接口以及逻辑处理做更规范、可重复的测试，一般使用Postman等接口测试工具对重要的接口进行请求，查看参数校验、返回结构、业务结果是否符合预期。机器测试当中包含黑盒测试和白盒测试这两种方式。

黑盒测试把系统当作一个整体看待，不去考虑系统的内部实现细节，而从需求的角度出发，根据输入和输出之间的关系来判断系统功能是否符合预期。本系统黑盒测试主要针对登录接口的正确、错误输入返回进行测试，对申请提交后是否产生申请单、明细，并进入待审批状态进行测试，管理员审批通过、驳回后申请状态是否发生变化进行测试，审批通过时库存是否按照申请数量扣减，库存不足时是否可以阻止扣减并提示错误，导入导出是否能生成或者解析Excel文件，报表统计在不同的时间区间下是否返回合理的统计结果。

白盒测试：白盒测试主要关注的是系统内部结构和逻辑是否正确，主要对条件判断、循环控制、关键业务逻辑等进行检验。本系统白盒测试主要是对审批模块的重要逻辑进行测试，比如只允许处于PENDING状态才能进行审批，保证不造成重复工作，审核批准后先进行库存核对再执行统一扣款，防止扣款部分遗漏的情况发生，导入模块对必填项以及角色合法性的检查（admin/user）是否正确，统计模块对时间区间转化、聚合计算及返回的字段是否一致等情况进行测试。采用手动测试和机器测试相结合的方式，在黑盒和白盒两种验证方法下进行系统的检验，以达到较好的效果。

测试用例

测试用例是系统测试过程规范化记载的一种，主要用以说明某个功能在某种输入、操作条件下，要如何进行检验以及检验的结果。本系统按照要求，将测试目的、测试步骤、预期结果、实际结果整理成一张统一的表格形式，对各个模块的功能逐一进行检验和追踪，使系统是否达到了预期的目的一目了然。测试用例覆盖了系统重要的业务链路和主要的功能点，包括登录认证、用户/管理员主页功能、物品查询与管理、领用申请提交、审批进度查询、管理员审批处理（包含库存校验和扣减）、Excel批量导入导出、统计报表查询和导出等内容，在用例里也包含一些异常以及边界场景的测试，即账号密码错误、必填项空缺、库存不够、重复审批、导入文件类型错、统计区间没有数据等，来保证系统在各种情况下都能给出合理的回答，并且数据一致。关于本系统中编写的测试用例如下表所示：

表7 登录模块测试表

测试目的 测试系统登录功能是否能够正常完成身份认证，并正确返回用户信息与角色字段，用于区分进入用户端/管理员端主页。

执行步骤描述 1）打开系统登录页面，输入正确的用户名与密码，点击“登录”按钮。 2）观察是否跳转到对应主页（普通用户进入用户主页，管理员进入管理员主页）。 3）使用错误密码再次登录，观察系统提示信息。 4）将用户名或密码置空后登录，观察系统提示信息。

预期效果 能够有效完成用户的登录操作。

实际效果 和预期效果一致。

表8 物品查询和管理模块功能测试表

测试目的 测试物品列表查询、关键字搜索、物品详情查询以及管理员对物品的新增/修改/删除等基础管理功能是否正常。

执行步骤描述 1) 用户进入主页，查看物品列表是否能正常加载。 2) 在搜索框输入关键字（如“笔”），点击搜索，观察返回列表是否变化。 3) 点击某个物品查看详情，观察库存、单位等信息是否完整。 4) 管理员进入物品管理页面，点击新增，填写物品名称/单位/库存/安全库存等信息并保存。 5) 管理员对某物品执行修改库存或安全库存并保存。 6) 管理员删除某物品后刷新列表，再次查询该物品验证是否删除成功。

预期效果 能够有效完成物品查找和管理的各项操作。

实际效果 和预期效果一致。

表9 用户申请模块功能测试表

测试目的 测试用户提交领用申请功能是否正常，是否能生成申请单并保存申请明细。

执行步骤描述 1) 用户登录进入用户主页，选择一种物品，填写申请数量与备注，点击“提交申请”。 2) 重复操作一次，选择多种物品分别填写数量后提交。 3) 提交成功后进入“我的申请/审批进度”页面，查看是否出现新申请记录。 4) 测试异常输入：将 items 置空或不传 items 提交，观察接口返回。

预期效果 能够有效完成用户申请的各项操作。

实际效果 和预期效果一致。

表10 审批进度模块功能测试表

测试目的 测试用户端审批进度查询是否正常，是否能同时返回申请单信息、申请明细以及审批日志，从而实现进度追踪。

执行步骤描述 1) 用户提交一条申请后，访问“审批进度/我的申请”页面。 2) 观察返回结构中是否包含 order（申请单）、items（明细）、logs（日志）。 3) 当申请处于待审批状态时，查看 logs 是否为空列表。 4) 管理员完成审批后，用户刷新进度页面，再次查看 logs 是否出现审批记录。

预期效果 能够有效完成查询审批进度的各项操作。

实际效果 和预期效果一致。

表11 管理审批模块功能测试表

测试目的 测试管理员待审批列表、查看申请明细、审批通过/驳回、库存校验扣减、日志写入以及重复审批拦截等关键逻辑是否正确。

执行步骤描述 1) 管理员进入待审批列表页面，查看待审批申请。 2) 选择一条申请进入详情，查看申请明细。 3) 执行驳回：填写驳回意见，提交审批。 4) 执行通过：确保库存充足后提交审批，观察库存变化。 5) 制造库存不足场景：将某物品库存调小后再尝试审批通过，观察提示与库存是否被扣减。 6) 对已处理过的申请再次审批，观察是否能被系统拦截。 7) 审批完成后查询日志。

预期效果 能够有效完成审批申请的各项操作。

实际效果 和预期效果一致。

表12 Excel批量导入导出模块功能测试表

测试目的 测试系统 Excel 模板下载、批量导入（用户/物品）以及批量导出功能是否正常，验证必填校验、角色合法性校验、重复数据处理与错误信息返回是否符合预期。

执行步骤描述 1) 管理员下载物品导入模板，检查表头、示例行和说明行。 2) 管理员上传物品Excel进行导入，分别测试：新增物品、同名物品更）。 3) 构造错误Excel导入，观察 errors 是否返回“行号+原因”。 4) 管理员下载用户导入模板。 5) 管理员上传用户Excel导入，测试：正常新增、重复账号跳过、角色非法、必填缺失。 6) 执行导出：物品导出、用户导出，检查文件是否可打开且字段完整。

预期效果 能够有效完成Excel批量导入导出的各项操作。

实际效果 和预期效果一致。

表13 报表统计模块功能测试表

测试目的 测试管理员统计报表查询与报表导出功能是否正常，验证时间区间统计、状态汇总、消耗排行与趋势数据输出是否正确。

执行步骤描述 1) 管理员进入报表统计页面，选择起止日期，触发统计查询。 2) 检查返回字段是否齐全。 3) 选择一个无数据时间区间再次查询，观察是否正常返回空列表且不报错。 4) 执行报表导出：库存报表、账号报表、申请报表，检查Excel可打开且内容完整。

预期效果 能够有效完成报表统计的各项操作。

实际效果 和预期效果一致。

总结与展望

工作总结

本次毕业设计完成了面向校园场景的低值易耗品领用管理系统分析、设计、实现工作，以用户申请、管理员审批、库存扣减、日志留痕、统计报表为主线，把以前线下登记、人工汇总的流程转变成线上闭环处理。前期对实际业务痛点进行梳理之后，确定了系统要同时支持普通用户和管理员两种角色，对物品台账、领用申请、审批处理、库存管理、批量导入导出、统计分析等主要功能进行了需求分析及总体设计，得出比较完整的功能模块划分和数据结构设计。

后端使用SpringBoot搭建接口服务，按照模块化思想把登录认证、物品查询和管理、申请单和申请明细的创建与查询、审批处理和审批日志记录、统计报表查询、Excel模板下载和批量导入导出等主要功能实现出来。系统从技术层面保证业务逻辑中的关键链路一致性的要求，管理员的审核通过采用的是“先校验库存，再统一扣除”的做法来防止库存扣除一半造成数据异常的情况发生，并且对申请单的状态流转以及审批日志的记录可以进行追溯以备查核和审计之用。前端依据登录所得到的用户角色信息进行页面分发，用户侧侧重于物品浏览、提交申请及进度显示，管理员侧重于审批办理、台账保存、报表制作和批量数据更新，从而达到操作通道清楚、功能入口明确的目的。

围绕登录、主页、用户申请、审批进度、管理审批、批量导入导出、报表统计等模块设计和执行了正常的流程以及边界测试用例，对系统在账号密码错误、库存不足、重复审批、导入数据不规范、统计区间为空等情况下的错误提示及数据保护进行测试，整体运行结果满足需求，可以满足日常领用登记、审批管理、数据汇总的实际使用要求。

同时，本系统还有可以进一步改善之处，比如可以采用更加严格的鉴权策略（JWT加拦截器）来加强后端权限管控，在高并发情况下还可以通过事务、乐观锁或者原子扣减语句来保证库存的一致性，对功能方面可以扩展多级审批、消息提醒、更多的统计和可视化图表等等。本文完成一个结构清晰、功能齐全、业务闭环清楚的低值易耗品领用管理信息系统，给校园物资管理信息化、规范化提供了一个可行的实现方案，具有一定的扩展性、可优化性基础。

存在的问题

虽然本文已经完成基于SpringBoot的校园低值易耗品领用系统的开发，实现了登录认证、物品管理、领用申请、审批处理、库存扣减、审批日志、统计报表和Excel导入导出等功能，但系统测试结果表明，该系统还不能完全满足校园日常管理的基本需求，从实际使用角度出发还有待于进一步的完善。从系统安全角度而言，目前系统已经具有基本的登录认证以及角色划分功能，不过整体的鉴权体系还比较初级，以后还要进一步提升后端接口权限管控以及登录安全措施。其次，在并发处理以及数据一致性方面，系统虽然对库存先校验后统一扣减的策略进行了控制，但是在高并发的情况之下，库存更新的过程中的原子性以及一致性还存在着改善之处。另外在功能完备程度上，现有的系统主要实现了单级审批、基础报表统计、常规导入导出等一些基本功能，在消息提醒、多级审批、库存预警、更多的数据分析展示等方面还比较欠缺。除此之外，在移动端适配、界面交互的细节、异常提示的友好性、后期部署维护的便利性等各个方面还有待提高。

未来展望

就以上不足而言，未来可以在现有的基础上从以下几个方面进行改善。在安全性上可以使用更完整的鉴权方式，即JWT和拦截器或者过滤器配合使用来对系统接口进行统一的认证以及权限校验，进而提高系统安全性以及可靠性。其次，对库存一致性进行保证的时候可以使用事务控制、乐观锁或者原子扣减语句这些技术，从而提高系统在高并发环境中稳定性的表现，防止并发操作造成数据的不一致。再次，对功能进行延伸，之后还可以加入多级审批、消息提醒、库存预警等新功能，并且会加大统

计分析模块的建设力度,新增加消耗趋势图、分类统计图、排行分析图等可视化展示,从而更加有效地支持管理决策。而且可以继续改善前端界面的设计和交互体验,提高系统的易用性,研究移动端适配性能,让系统在更多的使用场景中发挥作用。本文所构建的系统基本上已经形成了比较成熟的功能架构以及扩充的能力,经过不断的改进和功能完善,有望对校园低值易耗品的管理达到更加智能化、规范化、信息化的效果。

基于springboot的学校低值易耗品领用系统的设计与开发

系统设计

参考文献

黄和. 校友信息化管理赋能大学生生涯教育的意义、难点与对策[J]. 湖南大众传媒职业技术学院学报, 2025, 25(03):111-116. DOI:10.16261/j.cnki.cn43-1370/z.2025.03.019.

周筱龙. 基于Spring Boot和Vue.js的物资验收管理系统的开发与应用[J]. 电脑知识与技术, 2025, 21(24):40-44. DOI:10.14004/j.cnki.ckt.2025.1201.

杨玉, 刘杰举. 基于Spring Boot与Vue的物业管理系统设计与实现[J]. 鞋类工艺与设计, 2025, 5(14):114-116.

谢俞秋, 刘权锋. 医院高值耗材信息化全流程管理系统构建与效能评估[J]. 电脑知识与技术, 2025, 21(20):45-48. DOI:10.14004/j.cnki.ckt.2025.0998.

吴享南. 计算机网络信息安全中数据加密技术的研究[J]. 信息系统工程, 2025, (04):125-128.

张成挺, 程超, 叶万兴, 等. 大数据时代计算机网络安全技术的应用策略[J]. 电脑知识与技术, 2025, 21(06):86-87+96. DOI:10.14004/j.cnki.ckt.2025.0254.

吕志强. 基于Vue前端开发框架的管理系统设计[J]. 电脑知识与技术, 2024, 20(36):56-59. DOI:10.14004/j.cnki.ckt.2024.1900.

胡荣, 羊雪玲. 基于Spring Boot前后端分离Web系统的设计与实现[J]. 新能源与智能网联, 2024, (01):88-97.

谢振华. 基于Vue.js与Spring Boot的教务管理系统设计[J]. 电脑与信息技术, 2024, 32(04):95-97+101. DOI:10.19414/j.cnki.1005-1228.2024.04.023.

张晨亮, 张亚周, 孙雪丽. 面向教学的实验耗材管理系统研究与设计[J]. 电气电子教学学报, 2024, 46(03):232-234.

张慧. 数据库设计与学生信息管理系统优化[J]. 信息与电脑(理论版), 2024, 36(11):244-246.

孙培双, 朱百禄. 基于微信公众平台的高校低值易耗品管理系统的设计与实现[J]. 电脑知识与技术, 2023, 19(04):50-52+55. DOI:10.14004/j.cnki.ckt.2023.0201.

喻佳, 吴丹新. 基于SpringBoot的Web快速开发框架[J]. 电脑编程技巧与维护, 2021, (09):31-33. DOI:10.16184/j.cnki.comp rg.2021.09.013.

吴昌政. 基于前后端分离技术的web开发框架设计[D]. 南京邮电大学, 2020. DOI:10.27251/d.cnki.gnjdc.2020.000727.

姚嘉宾. 光明公司物资管理系统的设计与研究[D]. 大连理工大学, 2020. DOI:10.26991/d.cnki.gdllu.2020.004684.

陈光剑. Spring Boot开发实战[M]. 机械工业出版社:201808:825.

赵佳佳. 华侨职业技术学校低值易耗品管理系统的设计与实现[D]. 南京理工大学, 2017. DOI:10.27241/d.cnki.gnjgu.2017.000138.

李光, 朱震华. 基于Web的低值易耗品管理系统设计和开发[J]. 数字技术与应用, 2014, (04):175. DOI:10.19695/j.cnki.cn12-1369.2014.04.118.

袁永国, 王月浩. 基于台账的实验室低值易耗品管理系统设计与实现[J]. 重庆文理学院学报, 2013, 32(03):99-101. DOI:10.19493/j.cnki.issn1673-8004.2013.03.027.

吴兵, 张伟, 杜菁, 等. 基于Web技术的高等医学院校低值易耗品管理系统的设计[J]. 中国医疗设备, 2011, 26(08):36-37+48.

致谢

在毕业设计圆满完成之时，我满怀真诚的感谢，向在学习、设计过程中给予我支持和帮助的老师、同学和家人表示最真挚的谢意。

首先要对指导老师表示深深的谢意。课题选题、需求分析、系统设计、功能实现、论文撰写等各个方面，老师对我都有耐心细致的指导，并且对论文的每一个环节都认真地进行审核。不论系统总体结构怎样确定、数据库以及各模块怎样分层设计、核心业务流程怎样考虑实现路径，只要结合现实问题提出建议，就能使我在第一时间发现问题并加以修正，也可以在后期改进系统的同时提高论文的水准。老师的严谨治学态度、求实工作作风和对细节的高标准要求，对本次毕业设计有很好的帮助，也必将对我以后的学习工作产生深刻的影响。向老师表示最崇高的敬意和最真挚的谢意。

此外，还要对在大学期间所有老师进行感谢。正是经过学习和积累相关课程之后，才具备了完成本次系统开发所需的基本能力。数据库原理、软件工程、Web开发、Java开发等课程给我的提供理论基础和实践方法，在需求分析、系统建模、接口设计、测试用例编写方面有更充足、更有思路。非常感谢老师细致地帮助指导我，给我打下了较好的专业基础。

在此也向我的同学和朋友们在毕业设计中对我提出的建议表示谢意，同时也要感谢他们在我毕业设计期间给予的关心和支持。系统开发调试期间，各位对于系统功能讨论、问题排查、测试反馈、论文格式规范等各方面的提议都给予了较多支持，并且对我的进步给予过鼓励和帮助。与同学们一起学习、相互促进的过程是我大学生活里很宝贵的回忆。

除此之外还要对家人表示最诚挚的谢意。感谢你们多年来对我的学习、生活予以的理解、包容和支持，在我备战毕业设计及论文写作的时候，给予我无微不至的关心和强有力的依靠，让我能够全心投入到学习当中并完成项目的实现工作，最后完成了毕业设计。

最后向学校为我们的学习创造了好的条件和实践平台，向学院组织毕业设计、提供资料以及答辩安排表示感谢。毕业设计是对大学阶段所学知识的一种最后检验证实，也是个人成长的一部分。将来会用更认真的态度来不断充实自己，以实践来不断完善自己，回报社会。再次向所有给我提供帮助的老师、同学、家人表示由衷的谢意！

附录

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