|  |  |  |
| --- | --- | --- |
|  | 论文编号： |  |

**广州商学院**

**本科毕业论文（设计）**

|  |  |
| --- | --- |
| **题目：** | **面向多端协同的智能文件收集与管理系统** |
|  | **的设计与实现** |

|  |  |
| --- | --- |
| **姓名：** | **苏增烨** |
| **学号：** | **202206120166** |
| **学院：** | **信息技术与工程学院** |
| **专业班级：** | **软工2204** |
| **指导教师：** | **廖俐鹃**  注：此页内容不需要任何改修，电子签名或者手写签名和日期即可。  阅后删除此文本框。 |

**2026年6月**

**本科毕业论文（设计）原创性声明**

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

作者签名： 日期：2026年6月 1 日

**本科毕业论文（设计）版权使用授权书**

本论文（设计）作者完全了解学校有关保留、使用毕业论文（设计）的规定，同意学校保留并向国家有关部门或机构送交论文（设计）的复印件和电子版，允许论文（设计）被查阅和借阅。本人授权广州商学院可以将本论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本论文（设计）。

本论文（设计）属于

1．保密□，在\_\_\_\_\_\_年解密后适用本授权书。

2．不保密☑。

（请在以上相应方框内打“√”）

作者签名： 日期：2026年6月1日

指导老师签名： 日期：2026年6月1日

摘要

**关键词：**

ABSTRACT

**Key words**:

**目录**

[摘要 I](#_Toc514957848)

[ABSTRACT II](#_Toc537453926)

[1 绪论 1](#_Toc671915000)

[1.1 课题背景 1](#_Toc1406389074)

[1.2 目的与意义 1](#_Toc1976147836)

[1.2.1 研究目的 1](#_Toc134595150)

[1.2.2 研究意义 1](#_Toc840405759)

[1.3 论文研究主要内容 1](#_Toc699645194)

[1.4 国内外研究现状 2](#_Toc1463808233)

[1.4.1 国外研究现状 2](#_Toc652311999)

[1.4.2 国内研究现状 2](#_Toc503749258)

[2 关键理论与技术介绍 4](#_Toc1133242732)

[2.1 理论介绍 4](#_Toc378131481)

[2.2 关键技术栈介绍 4](#_Toc851689694)

[2.2.1 后端技术栈 4](#_Toc1370179803)

[2.2.2 前端技术栈 4](#_Toc1144802240)

[2.3 本章小结 5](#_Toc1385254207)

[3 需求分析与总体设计 6](#_Toc1097239922)

[3.1 智能文件收集与管理系统概述 6](#_Toc869292265)

[3.2 智能文件收集与管理系统的需求分析 6](#_Toc863847314)

[3.2.1 功能性需求分析 6](#_Toc1692352678)

[3.2.2 非功能性需求分析 6](#_Toc2098038278)

[3.3 智能文件收集与管理系统总体设计 6](#_Toc47854606)

[3.3.1 前端视图层设计概述 6](#_Toc1133479064)

[3.3.2 数据交互层设计概述 7](#_Toc55196111)

[3.3.3 服务层设计概述 7](#_Toc2115585720)

[3.3.4 数据库层设计概述 7](#_Toc762452661)

[3.4 本章小结 7](#_Toc506951778)

[4 系统的详细设计与实现 9](#_Toc1270905197)

[4.1 系统设计概要 9](#_Toc1231292917)

[4.2 数据库层设计与实现 9](#_Toc1187633527)

[4.3 前端视图层设计与实现 9](#_Toc1843673071)

[4.4 数据交互层设计与实现 9](#_Toc571761734)

[4.5 服务层设计与实现 9](#_Toc1757626660)

[4.6 系统界面展示 9](#_Toc1793710135)

[4.7 9](#_Toc510802359)

[5 系统测试与优化 10](#_Toc1563110654)

[5.1 系统功能测试与分析 10](#_Toc1033308027)

[5.2 本章小结 10](#_Toc107756500)

[6 总结与展望 11](#_Toc734781079)

[6.1 论文总结 11](#_Toc1434624503)

[6.2 未来展望 11](#_Toc1935117052)

[参考文献 12](#_Toc2019942796)

[致谢 13](#_Toc1757080596)

[附录 14](#_Toc1205947075)

# 绪论

## 课题背景

随着信息技术的迅猛发展和数字化转型的不断加速，文件收集与管理已成为教育机构、企业组织及个人用户日常工作中不可或缺的核心环节。无论是教育机构的教学互动、企业的项目推进，还是科研团队的数据共享，都高度依赖于文件的收集与分发。然而，当前广泛使用的传统文件收集方式，如通过QQ群、微信群组或电子邮件进行传输，其弊端日益凸显。这些方式不仅导致文件收集效率低下、后续整理与归档极为困难，更普遍存在版本控制混乱、数据安全性不足以及难以支持多设备间协同作业等深层次问题。

随着“互联网+”和移动办公模式的深度普及，特别是远程办公和在线教育的常态化，用户对于能够跨越不同平台、在多种设备上无缝协同的文件管理需求变得空前迫切。因此，开发一套能够实现文件自动化收集、智能分类、多端实时同步和提供严密安全保障的管理系统，以从根本上解决传统文件管理的诸多痛点，提升工作效率与协作体验，已具备非常重要的现实意义和紧迫性。

## 目的与意义

### 研究目的

尽管现有的文件管理系统在基础功能上已相对完善，但在多端协同的流畅性、文件处理的智能化以及用户体验的精细化方面仍存在显著不足。许多系统在不同设备间的同步性和一致性较差，缺乏智能化的文件分类、标签推荐与内容审核功能，并在数据安全与隐私保护方面（如端到端加密、细粒度权限控制）存在短板。

### 研究意义

本课题的研究具有重要的理论价值和实际应用意义：

理论与技术意义：本研究将探索Node.js在处理高并发文件管理系统中的性能优化策略，研究Ant Design Pro在构建复杂企业级管理系统中的组件化开发模式，并设计一套健壮的支持多端协同的系统架构与数据同步机制。这不仅能为相关领域的技术发展提供有益的探索，也能为构建下一代智能协同办公系统提供有价值的技术方案和实践经验。

实际应用价值：系统的实现将为不同场景提供高效的解决方案。在教育领域，可帮助教师高效收集和管理学生作业；在企业应用中，能支持团队文档协作，提升项目管理效率；在科研领域，可为团队提供安全的数据收集与共享平台。

## 论文研究主要内容

为实现上述研究目标，本论文将围绕“idrop.in云寄”智能文件收集与管理系统的分析、设计、实现与测试展开系统性研究。论文的主要研究内容包括以下几个方面：

1. **系统需求分析与建模：**制定详细的需求规格说明书。
2. **多端协同架构设计与技术选型：**采用前后端分离模式，后端基于Node.js与Express.js构建RESTful API，前端采用Ant Design Pro框架开发响应式界面。数据库层面采用MongoDB与MySQL组合存储，并引入Redis提供缓存服务。利用WebSocket技术实现多端实时通信。
3. **智能文件处理算法研究与实现：**集成机器学习算法，根据文件特征实现智能分类。研发基于NLP技术的文档内容分析模块，支持关键词提取与主题识别。设计智能标签推荐算法，提升文件组织效率。

多端协同机制设计与优化：设计高效的数据同步协议，支持增量同步和冲突解决。实现离线优先操作和自动同步恢复。开发跨平台兼容的文件在线预览功能。

安全防护体系构建：设计多层次安全架构，包括用户身份认证、基于角色的权限访问控制（RBAC）、数据传输加密、存储加密、文件安全扫描及完整的操作审计日志系统。

用户体验优化与性能调优：基于UX原则设计直观易用的界面，实现响应式布局。通过数据库索引优化、查询调优、缓存策略、文件分片上传及断点续传等技术，全面提升系统响应速度和处理大文件的用户体验。

## 国内外研究现状

### 国外研究现状

近年来，国外在协同办公与智能文件管理领域的研究与实践取得了显著进展。相关研究表明，多智能体办公协作系统通过整合人工智能、机器学习及自然语言处理技术，有效实现了任务智能分配与进度动态监控；同时，基于云架构的内容协作平台专为协作学习场景设计，依托分布式系统与实时通信机制保障了多设备间的数据同步效率；在企业级应用层面，基于大语言模型的业务文档工作流系统「Fok」能够深度解析文档内容并自动提取关键信息，显著提升了文档处理的智能化水平。这些进展为构建高效、智能的个人与团队信息管理工具提供了重要技术参考。参考文献

### 国内研究现状

国内在协同办公与文件管理系统领域发展迅速，成果显著。以致远互联为代表的协同管理软件企业，在"AI+协同办公"领域进行了大量实践探索，有效推动了协同系统在内容生产、智能助手与流程自动化等维度的智能化升级。在学术研究层面，相关成果深入探讨了开放式创新社区中用户知识贡献的采纳机制[1]，并通过智能制造项目案例系统[2]分析了跨学科团队知识整合的关键影响因素，这些研究为构建更贴合用户实际需求的协同系统提供了扎实的理论支撑与实践参考。

然而，纵观国内外研究与产品实践，现有系统在多端协同体验、智能化程度及安全隐私保护方面仍存在显著不足：多端协同中数据同步延迟与一致性问题导致用户体验不够流畅；智能化功能普遍局限于基础文件存储与共享，缺乏深度内容分析、智能分类及用户行为预测等高级能力；安全防护方面则普遍缺失端到端加密、细粒度权限控制及数据操作溯源等关键特性。基于此，本课题聚焦于通过集成AI技术优化实时同步机制、构建多层次安全防护体系，旨在开发一个新型智能文件收集与管理系统，有效填补上述研究空白并提升用户综合体验。

# 关键理论与技术介绍

本章将介绍支撑“面向多端协同的智能文件收集与管理系统”（idrop.in云寄）设计与实现的核心理论基础和所需的主要技术。

## 理论介绍

本系统的研发在方法论上遵循敏捷开发与用户导向的设计思想，强调快速迭代和以用户体验为中心。在系统架构上，采用了成熟的“前后端分离”模式，通过标准API实现客户端与服务器的解耦，为多端协同提供了灵活性。同时，在数据存储方面采用了混合存储理论，针对结构化数据（如用户信息）与非结构化数据（如文件元数据）分别选用最适合的数据库技术，以兼顾系统性能、数据一致性与未来的扩展性。

## 关键技术栈介绍

基于上述理论和系统需求，本项目在技术选型上力求成熟、高效且具备前瞻性。关键技术栈分为后端、前端和通信技术三大部分。

### 后端技术栈

Node.js：本系统选用Node.js作为后端开发的主要运行时环境。Node.js基于Chrome V8引擎，采用了异步、事件驱动的I/O模型，使其在处理高并发、I/O密集型任务（如本系统中的文件上传、下载和实时消息推送）时具有显著的性能优势。

Express.js：作为Node.js生态中最流行、最精简的Web应用框架，Express.js提供了强大的路由功能和中间件机制，是构建本系统RESTful API的核心基础，具有极高的灵活性和可扩展性。

MySQL：选用MySQL这一全球最流行的开源关系型数据库管理系统。在本系统中，它主要负责存储用户账户、密码、角色以及基于RBAC（Role-Based Access Control）模型的权限配置信息。

MongoDB：作为领先的NoSQL文档数据库，MongoDB被用于存储系统的核心业务数据——文件元数据。其灵活的BSON文档模型非常适合存储包含文件名、大小、上传者、智能标签、版本历史等动态变化的非结构化信息。

Redis：Redis是一个高性能的内存数据库（In-Memory Database）。在本系统中，它主要扮演两个角色：一是作为高速缓存层，缓存热点数据（如用户信息、常用文件列表），大幅降低数据库压力，提升响应速度；二是用于会话管理（Session Management），确保用户在多端登录时的一致性。

### 前端技术栈

React.js：本系统采用由Facebook（Meta）维护的React.js库来构建用户界面。React的核心优势在于其声明式视图和高效的虚拟DOM（Virtual DOM）机制，使得复杂UI界面的开发和状态管理变得简单高效。其强大的组件化（Component-based）思想，极大地提高了代码的复用性和可维护性。

Ant Design Pro：Ant Design Pro是一个基于React和Ant Design组件库的企业级中后台前端解决方案。它提供了丰富的、高质量的预制组件（如表格、表单、图表）、统一的设计规范以及开箱即用的页面脚手架。选用Ant Design Pro可以极大缩短开发周期，确保系统UI的专业性、美观性和跨浏览器一致性，让开发团队能更专注于业务逻辑的实现。

## 本章小结

本章简要介绍了指导本系统研发的理论基础，以及详细介绍了为实现系统功能而选用的关键技术，涵盖了后端的Node.js、Express.js、MySQL、MongoDB、Redis，前端的React、Ant Design Pro，以及用于通信的RESTful API和WebSocket技术。这些理论和技术的有机结合，为后续构建“面向多端协同的智能文件收集与管理系统”奠定了坚实的技术基础。后续章节将在此基础上展开对系统的详细设计与实现。

# 需求分析与总体设计

## 智能文件收集与管理系统概述

本论文所设计的“面向多端协同的智能文件收集与管理系统”（后续简称“idrop.in云寄”），是一个旨在解决传统文件收集方式效率低下、管理混乱、安全性差等问题的网络应用系统。系统基于Node.js后端技术栈与Ant Design Pro前端框架开发，致力于通过现代化的技术手段，为教育、企业及科研团队提供一个集自动化收集、智能管理、多端实时同步与安全可控于一体的统一平台。用户不仅可以在Web端（PC）进行操作，也能在移动端上无缝衔接，实现真正高效、便捷的协同工作体验。

## 智能文件收集与管理系统的需求分析

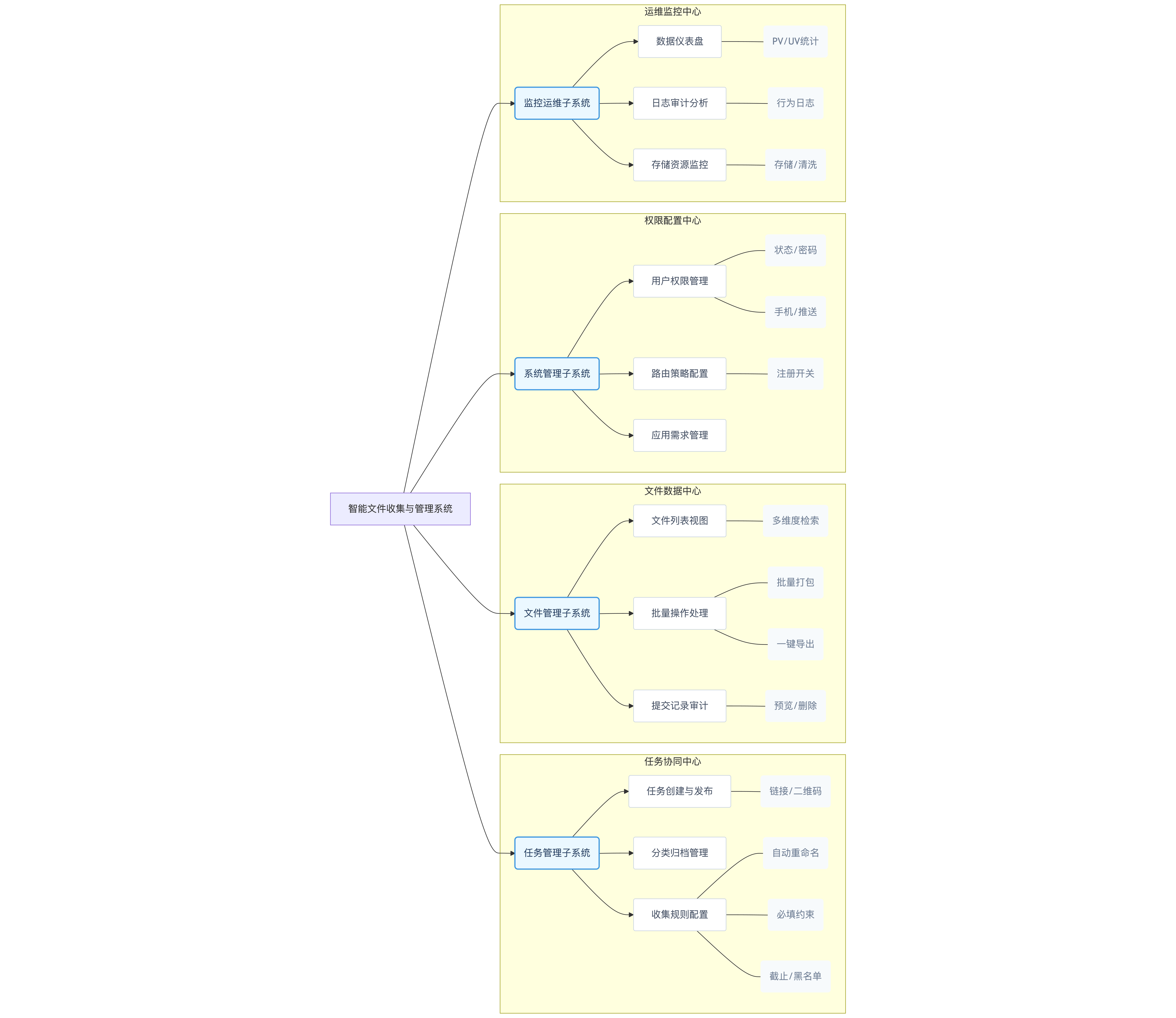
### 功能性需求分析

在功能性需求方面上，系统必须具备强大的文件收集任务管理能力。管理者（如教师或项目经理）应能便捷地创建文件收集任务，设置截止日期、文件格式要求等；参与者（如学生或团队成员）应能通过唯一的链接或入口，快速、批量地上传文件。并且系统需提供完善的文件管理功能，包括文件的在线预览、版本控制、拖拽上传、下载及分类归档。最为核心的是，系统必须集成智能处理能力，例如通过文件内容或元数据实现文件的自动分类、智能标签推荐和快速检索。同时，为满足多端协同的需求，系统必须包含一个可靠的用户身份认证与基于角色的权限控制（RBAC）模块，确保不同角色的用户（如管理员、普通用户、访客）拥有清晰的操作边界。

### 非功能性需求分析

在非功能性需求方面上，性能与实时性是首要考量。系统必须支持高并发的文件上传与下载，并采用分片上传、断点续传等技术优化大文件传输体验。多端协同要求数据同步必须是低延迟和实时的，当一个设备上的文件状态发生变更时，所有其他已登录的设备都应即时收到更新。在用户体验（UX）方面，界面必须遵循响应式设计，在PC和移动设备上均能提供直观、简洁、一致的操作流程。安全性是本系统的生命线，系统必须确保数据在传输（HTTPS）和存储（加密）过程中的绝对安全，并能抵御常见的网络攻击，建立完整的操作日志以备审计。

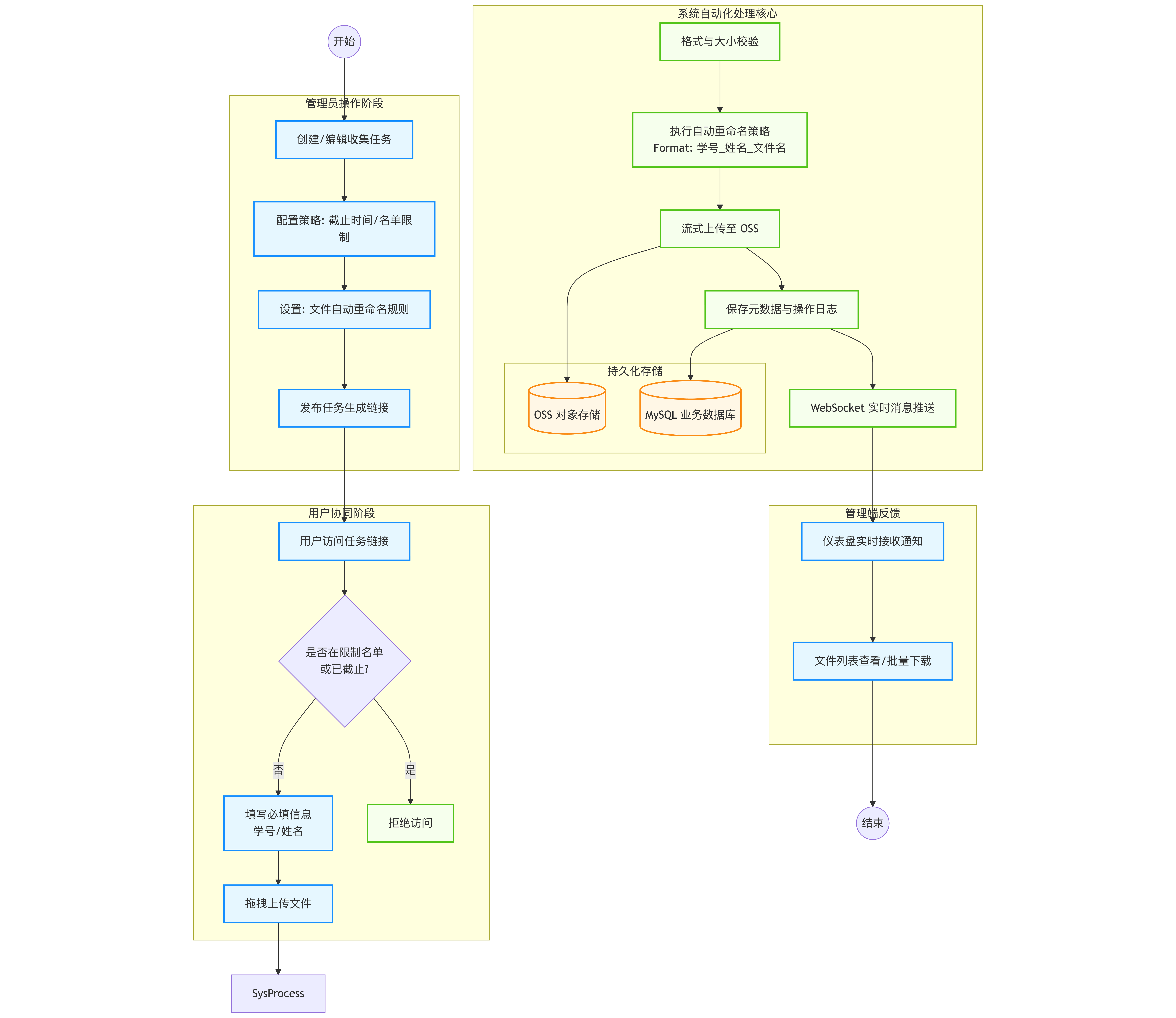
通过功能性需求的分析，本系统的功能如图1所示。



**图1 系统功能图**

## 智能文件收集与管理系统总体设计

为满足上述复杂的需求，并保证系统的高性能、高可用性和易扩展性，本系统在架构上采用了当前主流的前后端分离设计思想。该架构将整个系统在逻辑上划分为四个层次：前端视图层、数据交互层、服务层和数据库层，架构图如图2所示。



**图2 系统架构图**

### 前端视图层设计概述

前端视图层是用户与系统直接交互的界面，其设计的优劣直接决定了用户体验。本系统选用成熟的React.js作为核心框架，并基于此采用Ant Design Pro这一企业级前端解决方案。React的组件化思想使得UI界面的构建和复用变得高效；而Ant Design Pro提供了一套丰富、高质量的预制组件和统一的设计规范，极大地缩短了开发周期，确保了系统界面的专业性、美观性和响应式能力。该层负责处理所有的界面渲染、本地状态管理和用户交互事件，以实现对PC端和移动端的良好适配，满足“多端协同”的首要条件。

### 数据交互层设计概述

数据交互层是连接前端视图层与后端服务层的“通信总线”，负责规范两者之间的数据交换。本设计采用了“RESTful API + WebSocket”相结合的双通道模式。对于常规的、无状态的数据操作，如获取用户信息、请求文件列表、创建或删除收集任务等，系统通过遵循RESTful风格的HTTP API进行通信。而对于需要高实时性的场景，如文件上传状态的即时反馈、多端设备间的数据同步通知（例如“某某某已提交文件”），则采用WebSocket技术。WebSocket建立了一条持久化的全双工通信通道，允许服务器主动向客户端推送消息，从而以最低的延迟实现数据的实时同步。

### 服务层设计概述

服务层是整个系统的核心，负责处理所有核心的业务逻辑、数据运算和智能分析。本系统基于Node.js运行时环境，利用其异步、事件驱动的I/O模型来构建高性能后端。核心业务逻辑（如用户认证、权限校验、文件处理、任务管理）将通过Express.js框架以API的形式暴露给数据交互层。为了提升系统响应速度和处理高并发的能力，本层还将引入Redis内存数据库，用于缓存热点数据和管理用户会话（Session）。此外，开题报告中规划的智能分类、标签推荐等算法模块，也将在此层中实现或集成，作为独立的微服务供主流程调用。

### 数据库层设计概述

鉴于本系统需要处理的数据类型复杂多样，单一的数据库模型难以同时满足性能和一致性的要求。因此，本设计采用了“关系型 + 非关系型”的混合数据存储方案。对于结构化强、一致性要求高的数据，如用户账户信息、角色、权限配置（RBAC）等，选用成熟稳定的MySQL关系型数据库进行存储。而对于数量庞大、结构灵活、读写频繁的数据，如文件的元数据（Metadata）、版本信息、智能标签、用户操作日志等，则选用高性能的MongoDB文档型数据库，利用其灵活的BSON模型和强大的水平扩展能力。

## 本章小结

本章首先对文件收集与管理系统进行了简单的概述，接着分别从功能性和非功能性两个方面进行需求分析，而后介绍了可视化系统的总体架构设计。系统的架构包括四层：前端视图层、数据交互层、服务器层和数据库层。最后对系统架构中的各层进行了简单的概述。

本章首先对智能文件收集与管理系统系统进行了简要的概述，明确了“idrop.in云寄”项目的建设目标。接着通过对典型用户场景的分析，详细阐述了系统必须满足的功能性需求和非功能性需求。基于这些需求，最后介绍了该系统的总体架构设计，系统的架构分为四层：视图层、交互层、服务层以及数据库层，并对每个层次的核心职责与技术选型进行了简单概述。

# 系统的详细设计与实现

## 系统设计概要

## 数据库层设计与实现

## 前端视图层设计与实现

## 数据交互层设计与实现

## 服务层设计与实现

## 系统界面展示

## 

# 系统测试与优化

## 系统功能测试与分析

## 本章小结

# 总结与展望

## 论文总结

## 未来展望

参考文献

1. 洪闯,李贺,毛太田.开放式创新社区用户知识贡献的采纳机理研究[J].现代情报,2020,40(05):33-40.
2. 邢飞,彭国超,贾怡晨,等.跨学科团队知识整合影响因素研究——以智能制造项目为例[J].现代情报,2020,40(05):41-50.

致谢

2026年 6 月 1 日

附录

此处为正文部分，请替换为自己的内容。以下红色字为相关格式说明，请理解相关要求后，自行删除。

**【注】**

**附录是正文主体的补充。附录属于可选部分，如果没有附录，请删除该部分。**

**下列内容可以作为附录：**

**(1)攻读学位期间发表的（含已录用，并有录用通知书的）与学位论文相关的学术论文。**

**(2)由于篇幅过大，或取材于复制件不便编入正文的材料、数据。**

**(3)对本专业同行有参考价值，但对一般读者不必阅读的材料。**

**(4)论文中使用的符号意义、单位缩写、计算机程序全文及有关说明书。**

**(5)附件：光盘、与论文相关的鉴定证书、获奖奖状或专利证书的复印件等。**