**Hadoop云端网盘**

**系**

**统**

**分**

**析**

**文**

**档**

**目 录**

[一、项目概述 2](#_Toc17237)

[1.1 项目背景 2](#_Toc5193)

[1.2 项目目标 2](#_Toc25012)

[二、系统角色与权限 2](#_Toc14668)

[2.1 角色划分 2](#_Toc12527)

[2.2 权限矩阵 2](#_Toc21746)

[三、功能结构设计 2](#_Toc30099)

[3.1 用户系统 2](#_Toc9569)

[3.2 文件系统 2](#_Toc31188)

[3.3 文件夹系统 3](#_Toc17943)

[3.4 回收站管理 3](#_Toc25832)

[3.5 系统设置 3](#_Toc7261)

[四、技术选型与架构 3](#_Toc7733)

[4.1 前端技术栈 3](#_Toc13367)

[4.2 架构设计要点 3](#_Toc8961)

[五、系统流程图 3](#_Toc31200)

[六、非功能性需求 4](#_Toc10881)

[6.1 安全性 4](#_Toc23569)

[6.2 性能 4](#_Toc4986)

[6.3 可扩展性 4](#_Toc1448)

[七、附录 4](#_Toc12240)

**一、项目概述**

1.1 项目背景

Hadoop 云端网盘是一套基于 Web 的分布式云存储系统，对标百度网盘等主流产品，旨在解决个人及组织用户在文件存储、管理与协作场景下的核心需求。当前传统本地存储面临容量限制、跨设备访问不便及协作效率低下等问题，而分布式存储技术（如 HDFS）的成熟为海量文件管理提供了可靠方案。本系统基于 Hadoop 生态构建，支持高并发文件操作与弹性存储扩展，适用于教育机构、中小型企业及个人用户的文件云端管理场景。

1.2 项目目标

功能性目标：  
实现文件上传 / 下载、分类浏览、文件夹层级管理、多角色权限控制、回收站机制及系统配置功能，支持文件预览（图片 / 文档 / 视频）、链接分享等核心场景。

非功能性目标：

安全性：采用 Token 认证（JWT）、数据传输加密（HTTPS）、存储加密（AES-256），满足等保三级安全要求；

易用性：界面交互符合主流网盘操作习惯，支持面包屑导航、批量操作等便捷功能；

可扩展性：底层存储支持 HDFS 与对象存储（如 MinIO）无缝切换，系统容量可扩展至 PB 级。

**二、系统角色与权限**

2.1 角色划分

| 角色 | 描述 |
| --- | --- |
| 管理员 | 系统级权限持有者，负责用户与资源的全局管理 |
| 普通用户 | 个人文件管理角色，权限限于自身资源操作 |

2.2 权限矩阵

| 功能模块 | 管理员权限 | 普通用户权限 |
| --- | --- | --- |
| 用户管理 | 增 / 删 / 改 / 查所有用户 | 无 |
| 文件管理 | 操作系统内所有文件（含删除 / 恢复） | 仅操作自己上传的文件 |
| 文件夹管理 | 管理所有文件夹结构 | 管理个人创建的文件夹 |
| 系统设置 | 配置 VIP 策略、上传限制等全局参数 | 自定义显示风格、查看个人配额 |

**三、功能结构设计**

3.1 用户系统

核心功能：注册 / 登录、第三方登录（预留微信 / QQ 接口）、权限校验（RBAC 模型）、角色状态管理（Redux 维护用户角色与 Token）。

交互流程：用户登录后通过 Redux 实时同步权限数据，路由组件根据角色动态显示可访问页面（如管理员可见 “用户管理” 菜单）。

3.2 文件系统

| 功能项 | 细节说明 |
| --- | --- |
| 上传 | 支持单文件最大 5GB，自动分片上传（≥100MB 拆分），断点续传（HTTP 308 状态码） |
| 下载 | 支持批量下载（ZIP 打包），大文件分块传输 |
| 预览 | 图片 / 视频在线预览，文档调用 LibreOffice 转换为 PDF 预览 |
| 分享 | 生成带时效的分享链接，支持密码保护与权限控制（仅预览 / 可下载） |

3.3 文件夹系统

层级管理：支持无限层级嵌套，通过parentId字段递归构建树状结构；

导航机制：面包屑导航实时显示当前路径，支持点击路径节点快速跳转上级目录。

3.4 回收站管理

逻辑删除：文件删除后转入回收站，默认保留 30 天，管理员可配置清理策略；

操作权限：用户仅能恢复 / 彻底删除自己的回收站文件，管理员可批量清理所有过期文件。

3.5 系统设置

VIP 功能：识别 VIP 用户并提供无限存储空间、高速传输通道等特权；

个性化配置：支持暗黑 / 亮色主题切换、文件列表视图切换（列表 / 网格模式）。

**四、技术选型与架构**

4.1 前端技术栈

| 类别 | 技术 | 版本 | 说明 |
| --- | --- | --- | --- |
| 框架 | React | 18.x | 组件化开发，支持 Concurrent Mode |
| 状态管理 | Redux Toolkit | 2.x | 替代传统 Redux，简化状态逻辑 |
| 路由 | React Router DOM | v6 | 支持动态路由加载与 Suspense 懒加载 |
| 样式 | SCSS + Ant Design | 5.x | 组件库与自定义样式结合 |
| 构建工具 | Vite | 4.x | 原生 ES Module，快速冷启动 |
| 模块化 | ES6 Module + 路径别名 | - | @指向src目录，优化导入路径 |

4.2 架构设计要点

状态管理：通过 Redux 集中管理用户登录态（token、role）、文件选中状态等全局数据，实现组件间数据同步；

性能优化：使用 React.lazy + Suspense 实现组件懒加载，配合 Vite 的 Tree Shaking 减少打包体积；

兼容性：基于现代浏览器（Chrome 80+、Edge Chromium）开发，适配 1366×768 及以上分辨率屏幕。

**五、系统流程图**



具体过程：

A[用户登录] --> B[校验用户名/密码]

B -->|通过| C[生成JWT Token，存入localStorage]

C --> D[根据Token获取用户权限与文件结构]

D --> E[展示文件列表界面]

E --> F[用户操作文件]

F -->|删除| G[标记文件为逻辑删除，移入回收站]

G --> H[触发Redux状态更新，刷新文件列表]

H --> I[后台定时任务检查回收站文件（每日凌晨2点）]

I -->|超过保留期限| J[物理删除文件，释放存储资源]

**六、非功能性需求**

6.1 安全性

认证机制：JWT Token 有效期 24 小时，刷新 Token 机制防止会话固定攻击；

数据加密：文件传输使用 HTTPS+TLS 1.3，存储层对敏感字段（如密码）进行 BCrypt 哈希处理。

6.2 性能

首屏加载：通过代码分割与懒加载，首屏 JS 资源加载时间≤3 秒；

文件操作响应：小文件（≤10MB）上传 / 下载延迟≤5 秒，大文件分片传输支持并行加速。

6.3 可扩展性

存储扩展：底层文件存储抽象为接口，支持 HDFS、MinIO、阿里云 OSS 等存储后端切换；

功能扩展：预留文件协作编辑、AI 识别（图片 OCR）等功能接口，方便后续迭代开发。

**七、附录**

术语表：

RBAC：基于角色的访问控制（Role-Based Access Control），通过角色关联权限实现细粒度授权；

分片上传：将大文件拆分为多个数据块依次传输，解决网络中断导致的全量重传问题。

界面原型参考：可附加文件管理界面、登录页等关键页面的线框图或高保真设计图。