## **基于Hadoop的云盘系统设计与实现系统功能设计文档**

目录

[基于Hadoop的云盘系统设计与实现系统功能设计文档 1](#_Toc461)

[一、系统基本情况 2](#_Toc15804)

[1.1 系统整体定位 2](#_Toc5926)

[1.2 技术栈详情 2](#_Toc25207)

[二、核心功能板块 2](#_Toc15433)

[2.1 用户管理相关功能 2](#_Toc29744)

[2.1.1 登录与注册 2](#_Toc21076)

[2.1.2 权限管理工作 3](#_Toc25738)

[2.2 文件管理相关功能 4](#_Toc11633)

[2.2.1 文件操作功能 4](#_Toc982)

[2.2.2 文件展示与搜索功能 4](#_Toc7759)

[2.3 系统布局与导航功能 5](#_Toc8983)

[2.3.1 头部导航设计 5](#_Toc20631)

[2.3.2 侧边栏导航设计 5](#_Toc11077)

[2.3.3 内容区设计 6](#_Toc28064)

[2.4 状态管理与路由功能 6](#_Toc31629)

[2.4.1 Redux状态管理 6](#_Toc11830)

[2.4.2 路由配置情况 6](#_Toc11934)

[2.5 接口与网络请求功能 6](#_Toc6962)

[三、待完善的扩展功能 7](#_Toc10045)

[3.1 分享功能 7](#_Toc31795)

[3.2 回收站管理功能 8](#_Toc1729)

[3.3 多端同步功能 8](#_Toc11447)

[四、系统架构与目录说明 8](#_Toc1060)

[4.1技术组件映射关系 8](#_Toc4963)

[五、性能与安全设计 8](#_Toc4289)

[5.1 性能优化措施 8](#_Toc5190)

[5.2 安全机制设计 9](#_Toc24581)

[六、部署与兼容性说明 9](#_Toc24142)

[6.1 部署方案规划 9](#_Toc20886)

[6.2 兼容性支持情况 10](#_Toc13641)

## **一、系统基本情况**

### **1.1 系统整体定位**

本系统是一个基于Hadoop HDFS的分布式云盘系统，采用前后端分离架构。前端用React 18搭配Ant Design 5.x进行组件化开发，后端基于Spring Boot 3.x和 Hadoop生态搭建，能实现文件存储、权限控制、多端同步等核心功能，主要面向企业和个人用户，提供安全高效的云存储服务。系统对标主流网盘产品，解决传统存储的容量限制与协作效率问题，支持高并发操作与 PB 级存储扩展。

### **1.2 技术栈详情**

****前端部分****：使用React 18、Ant Design 5.x、Redux Toolkit 2.x、React Router DOM 6.x、Vite 4.x等技术。

****后端部分：****基于 Spring Boot 3.x、Hadoop 3.x、MySQL 8.0、Redis 7.x等技术。

****部署方面：****采用Docker、Kubernetes、Nginx等工具。

## **二、核心功能板块**

### **2.1 用户管理相关功能**

### **2.1.1 登录与注册**

****登录功能****：支持用户名和密码登录，集成了JWT Token认证机制。Token存储在sessionStorage中，默认有效期1小时，在过期前10分钟会自动刷新。登录界面设有“记住密码”功能，通过localStorage实现长效存储。前端代码在src/views/login/index.tsx文件中，通过useAppDispatch调用Redux的login action来更新用户状态。

****注册功能：****注册表单包含用户名（需唯一）、手机号（需进行 11 位数字校验）、邮箱（需进行格式校验）、密码（6-16 位）、确认密码等字段。注册成功后，会自动在HDFS中为用户创建根目录，路径为/user/{userId}，之后会自动跳转至登录页面。前端代码位于src/views/register/index.tsx，注册逻辑通过Ant Design Form组件实现。

### **2.1.2 权限管理工作**

****角色体系设置****：

| **角色类型** | **文件操作权限** | **系统管理权限** |
| --- | --- | --- |
| 普通用户 | 可进行个人文件的上传、下载、删除操作，以及创建目录 | 无 |
| 管理员 | 可操作所有用户的文件 | 具备查看 HDFS 集群状态、进行用户管理等权限 |
| VIP 用户 | 个人文件操作+高速下载 | 无 |

****权限控制方式****：前端通过useAppSelector获取role和isVip字段，动态显示按钮（如管理员可见“用户管理”,VIP用户可见“无限存储”标识）。对于敏感操作，如删除文件，后续计划增加二次验证功能（如短信或邮箱验证码）。

### **2.2 文件管理相关功能**

### **2.2.1 文件操作功能**

****文件上传****：支单文件最大 5GB，自动分片上传（≥100MB拆分为512KB/片），通过Blob.slice()生成切片，计算MD5实现秒传（校验HDFS是否存在相同文件）。前端代码在src/layout/component/ContentMain.tsx文件中，上传逻辑与UploadOutlined图标按钮绑定,支持断点续传与失败重试。

****文件下载****：支持批量下载（ZIP打包）、断点续传（Range请求头，返回206 Partial Content）和多线程下载（Blob处理）。“离线下载”按钮预留接口，后端通过HdfsClient.readFileRange实现分块传输。

### **2.2.2 文件展示与搜索功能**

****列表展示形式****：使用 Ant Design Table 组件展示文件列表，列表列包括文件名（带有目录或文件图标）、类型、大小、修改日期等信息。支持文件的单选和全选，通过selectedRowKeys状态来管理选中的文件项。引入react-window实现虚拟滚动，渲染 1000 条数据内存占用≤150MB。

****搜索功能实现****：顶部设有搜索栏，支持文件名的模糊搜索，通过调用后端的/file/search接口实现搜索功能。前端代码在ContentMain组件的SearchOutlined图标输入框中。

****视图切换功能****：支持列表视图（默认视图）和网格视图的切换，通过 CSS控制布局，代码位于ContentMain组件。

### **2.3 系统布局与导航功能**

### **2.3.1 头部导航设计**

头部导航包含 Logo、导航菜单（包括网盘、分享、上传中、下载中等选项）、用户信息（头像、用户名、当前目录）、客户端下载入口以及退出按钮。退出功能通过 Redux 的logout action来清除 Token，并跳转至登录页面，相关代码在src/layout/component/HeaderMain.tsx文件中。

### **2.3.2 侧边栏导航设计**

侧边栏提供分类浏览功能，包括全部文件、图片、音频、视频、我的分享、回收站等分类。点击分类项会触发路由跳转，通过 React Router 实现页面的切换。“我的分享” 入口关联/share页面，显示用户创建的分享链接。

### **2.3.3 内容区设计**

内容区的操作栏有上传、新建文件夹、离线下载、搜索框等按钮。文件列表区域用于展示文件数据，文件列表支持点击（暂未实现预览，仅控制台打印日志，计划集成PDF.js和LibreOffice转换）。

### **2.4 状态管理与路由功能**

### **2.4.1 Redux状态管理**

****用户状态存储****：用户状态存储在store/modules/user.ts文件中，包含userNamerole、isVip、token字段，通过login和logout action 来更新用户状态。

文件状态管理：文件状态管理通过store/fileSlice.ts实现，包含文件列表files、总数total、加载状态loading。

### **2.4.2 路由配置情况**

****主路由设置****：系统主路由包括/login（登录页）、/register（注册页）、/主界面（包含文件管理、分享、回收站子路由）。

****路由守卫功能****：在App.tsx文件中，通过useEffect监听路由变化，未登录用户强制跳转登录页，管理员角色限制访问普通用户页面。

### **2.5 接口与网络请求功能**

****接口封装方式****：Axios 实例化代码在src/utils/request.ts文件中，包含请求拦截器（用于添加 Token）和响应拦截器（用于处理数据格式）。

****核心接口列表****：

| **接口路径** | **请求方法** | **功能描述** | **前端调用位置** |
| --- | --- | --- | --- |
| /user/login | POST | 用户登录功能 | Login组件的onFinish函数 |
| /file/upload | POST | 文件上传功能 | ContentMain组件的上传按钮 |
| /file/list | GET | 获取文件列表 | ContentMain组件表格数据 |
| /file/delete | POST | 删除文件 | 文件列表 “删除” 按钮 |
| /user/role/update | PUT | 更新用户角色 | 管理员后台 |

## **三、待完善的扩展功能**

### **3.1 分享功能**

计划实现生成分享链接的功能，支持密码保护（6-8 位）和权限控制（仅预览 / 可下载）。同时，将在侧边栏设置 “我的分享” 入口，用于管理分享链接。

### **3.2 回收站管理功能**

开发回收站界面，用于展示已删除的文件，支持对文件进行恢复或永久删除操，显示文件删除时间与大小。后端计划通过@Scheduled定时任务来清理过期的回收站文件。

### **3.3 多端同步功能**

在Header和Footer界面设置客户端下载入口，支持Android、Windows、Mac、iPhone等平台。客户端实现本地文件与云端自动同步，通过WebSocket实时通知变更，Electron构建桌面端，React Native 开发移动端。

## **四、系统架构与目录说明**

### **4.1技术组件映射关系**

****状态管理****：Redux Toolkit相关代码集中在store目录，通过useAppDispatch和useAppSelector来调用相关功能。

****路由管理****：React Router 6.x 的配置代码在routes/index.tsx文件中，通过useRoutes来配置路由表。

****UI 组件****：Ant Design组件主要在ContentMain、HeaderMain、SiderMain等组件中使用。

## **五、性能与安全设计**

### **5.1 性能优化措施**

****前端优化****：组件懒加载减少首屏 JS 体积 40%，首屏加载时间 2.8 秒（目标≤3 秒）。react-window实现列表虚拟滚动，提升大数据量渲染性能（1000 条数据渲染耗时≤500ms）。

****后端优化****：Redis 缓存用户权限信息（有效期 30 分钟），减少数据库查询次数。HDFS 多集群负载均衡（HDFS Federation），提升文件读写吞吐量至 50MB/s。

### **5.2 安全机制设计**

****身份认证机制****：WT Token包含sub（用户ID）、role、exp（过期时间），存储于localStorage，刷新Token 通过HttpOnly Cookie防 CSRF。

****数据加密处理****：传输层HTTPS+TLS 1.3加密，存储层密码BCrypt哈希（盐值16字节，迭代10轮），敏感文件AES-256分块加密。

****安全测试****：登录接口暴力尝试限制（5 次失败后封锁 IP），文件上传拦截恶意脚本。

## **六、部署与兼容性说明**

### **6.1 部署方案规划**

****前端部署****：使用Vite构建静态资源，通过Nginx进行部署。

****后端部署****：Spring Boot应用通过Kubernetes集群进行管理，并与 HDFS、MySQL、Redis 等服务对接，Prometheus+Grafana 监控CPU/内存/HDFS吞吐量。

### **6.2 兼容性支持情况**

****浏览器兼容性****：支持Chrome 80+、Edge Chromium、Firefox 90+，通过Autoprefixer处理CSS兼容性，适配 1366×768 及以上分辨率。

****客户端兼容性****：预留Electron/React Native入口，未来版本实现多平台适配，当前支持主流浏览器调试。