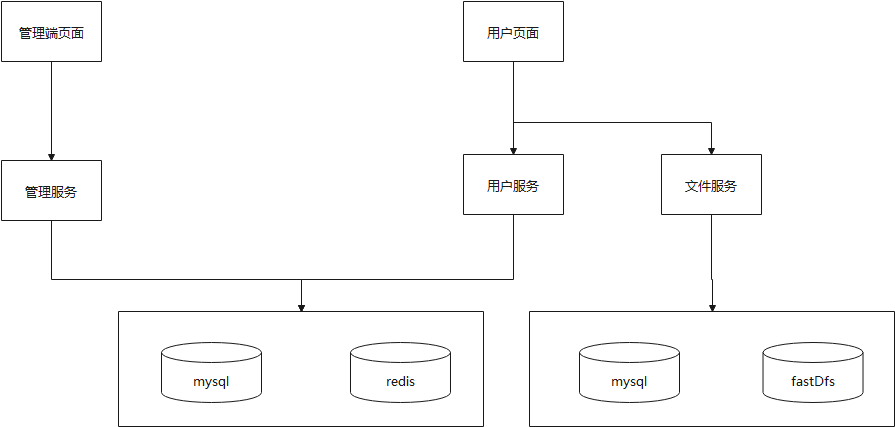
# 文件上传平台

## 项目背景

一个支持文件上传的平台，前端使用Vue构建，后端使用SpringMVC框架，实现前后端分离。数据库采用 MySQL，Redis。

## 模块功能分析

### 2.1 整体架构图



### 2.2 用户模块

2.2.1 基础功能

用户服务主要提供给外部用户使用，支持用户登录、注册、上传和下载文件，并支持删除自己上传的指定文件，查看自己上传的文件列表。

**个人信息**

显示个人头像和名称

**我的存储**

显示我的可用文件列表，支持下载和删除操作，删除时需要用户确认是放入垃圾桶还是彻底删除

同时支持文件名搜索和上传时间范围查询

界面显示：文件名称、文件大小、文件类型、上传时间和操作

使用分页方式查看数据

**传输列表(暂时不做具体业务)**

分为三种状态，分别为正在上传、正在下载、传输完成

界面上显示：文件名称、文件大小、文件类型、上传时间、上传进度，暂不允许中断上传

若是在传输完成界面，上传进度不显示，显示操作栏，操作栏有：下载、删除，后期增加分享功能  
使用分页方式查看数据

**垃圾桶**

垃圾桶显示用户放入垃圾桶的文件列表

界面显示：文件名称、文件大小、文件类型、上传时间、操作

操作提供：

恢复：将文件从垃圾桶移除，放回我的存储列表中

彻底删除：将文件从文件服务器中删除，并清理掉这条记录

2.2.2 后期拓展

JWT token 验证

短信平台接入

SSO单点登录

第三方登录、接入微信

支付模块

第三方存储平台，如七牛云

### 2.3 文件服务模块

2.3.1 基础功能

支持用户上传/下载文件

支持用户删除文件 => 事务

支持管理端删除/禁用一些文件

#### 2.3.2 后期拓展

支持管理端直接上传一些文件

公链/私链分享

实时进度返回

断点续传

## 项目流程设计

### 3.1 注册

支持用户通过手机号注册，为了能后期兼容短信验证码方式，手机号格式暂时限定于中国手机号格式。密码暂无限制

### 3.2 登录

支持用户通过账号、密码登录，但以后要兼容第三方登录以及短信验证方式登录

### 3.3 找回密码

一期先让用户输入账号就能修改密码，后期接入短信平台，通过验证手机号的方式登录

### 3.4 上传/下载

支持用户上传和下载文件，暂不做格式限制，但上传大小限制为100M，后期增加断点续传和实时进度反馈

### 3.5 查看上传列表

支持查看自己上传了的文件列表，并能对其进行操作。

支持文件名搜索，和限定上传时间范围

### 3.6 删除自己上传的文件

支持用户将自己上传的文件删除，但应提供给用户两种，一种是彻底删除，另一种是放进回收站。二者都不显示在自己的上传文件列表中，但彻底删除的文件不能恢复，且不会在垃圾桶列表中展示

### 3.7 垃圾桶

用户在删除文件时，没有选择彻底删除，则会被放进垃圾桶中。垃圾桶中的的文件可以被恢复(即文件依旧先保存着)。同时提供彻底删除选项。

### 3.8 彻底删除

用户可以在我的存储中删除文件时，以及垃圾桶中，彻底删除某个文件，这个文件记录将不会再显示我的存储中

## 原型图

### 4.1 我的存储



### 4.2 传输列表

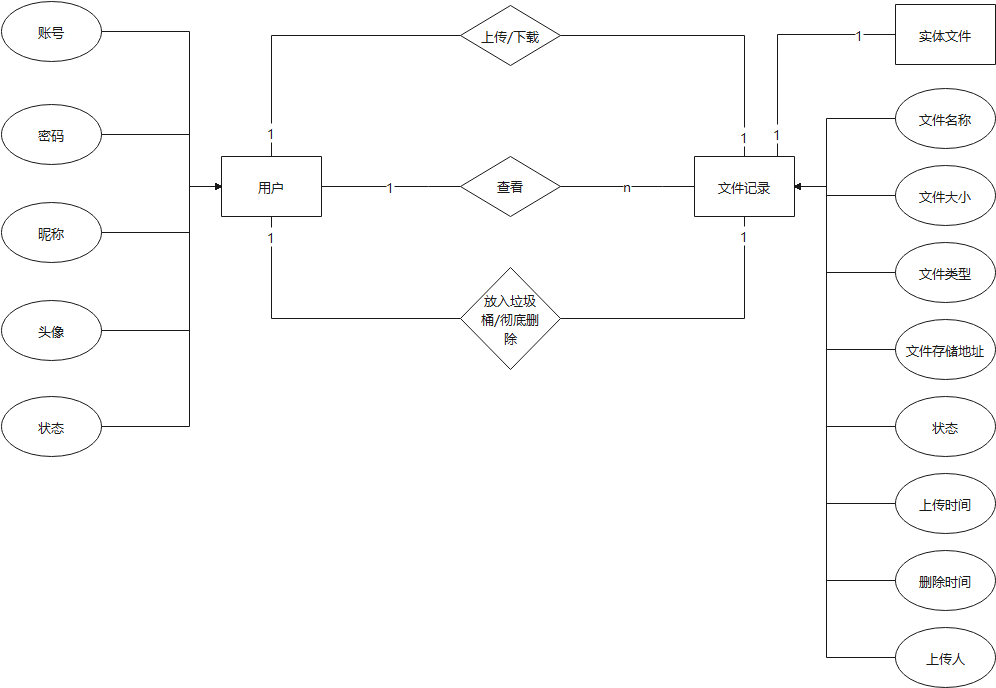


### 4.3 垃圾桶



## 数据库表设计

### 5.1 E-R图



### 5.2 表设计

#### 5.2.1 user(用户表)

用于存储用户账号相关

|  |  |  |
| --- | --- | --- |
| **字段名称** | **类型** | **说明** |
| id | bigint | 主键，唯一标识 |
| user\_id | varchar(100) | 用户id，唯一标识 |
| account | varchar(100) | 用户账号 |
| password | varchar(100) | 用户密码 |
| username | varchar(100) | 用户昵称 |
| avatar | varchar(255) | 用户头像地址 |
| state | int | 用户状态 0 正常 1 冻结 2 注销 |
| created | datetime | 创建时间 |
| updated | datetime | 更新时间 |

#### 5.2.2 upload\_file(文件存储表)

用于存储用户上传的文件记录

|  |  |  |
| --- | --- | --- |
| **字段名称** | **类型** | **说明** |
| id | bigint | 主键，唯一标识 |
| file\_id | varchar(100) | 文件id，唯一标识 |
| name | varchar(100) | 文件名称 |
| type | varchar(100) | 文件类型 |
| size | varchar(255) | 文件大小 |
| url | varchar(255) | 文件存储地址 |
| state | int | 文件状态 0 正常 1 垃圾桶 2 彻底删除 |
| created | datetime | 上传时间 |
| updated | datetime | 修改时间，如删除 |
| owner | bigint | 上传人的id |

#### 5.2.3 创建表语句

create table user(

`id` bigint not null auto\_increment primary key,

`user\_id` varchar(100) unique not null comment '用户id',

`account` varchar(100) unique not null comment '用户账号',

`password` varchar(100) not null comment '用户密码',

`username` varchar(100) not null comment '用户昵称',

`avatar` varchar(255) not null comment '用户头像地址',

`state` int DEFAULT 0 comment '用户状态 0 正常 1 冻结 2 注销',

`created` datetime not null comment '创建时间',

`updated` datetime not null comment '更新时间'

) ENGINE=InnoDB CHARSET=utf8;

create table upload\_record(

`id` bigint not null auto\_increment primary key,

`record\_id` varchar(100) unique not null comment '记录id',

`name` varchar(100) not null comment '文件名称',

`type` varchar(100) not null comment '文件类型',

`size` varchar(255) not null comment '文件大小',

`url` varchar(255) not null comment '文件存储地址',

`state` int DEFAULT 0 comment '文件状态 0 正常 1 放入垃圾桶 2 彻底删除',

`created` datetime not null comment '上传时间',

`updated` datetime not null comment '更新时间，如删除',

`owner` bigint not null comment '文件上传人的id'

) ENGINE=InnoDB CHARSET=utf8;

alter table upload\_record add index idx\_owner(owner);

alter table upload\_record add index idx\_state(state);

## 项目结构

前端： upload-client-web 用户页面

upload-manager-web 管理页面

后端： upload-parent

---- upload-user 用户服务

---- upload-file 文件服务

---- upload-manager 管理端服务