

# 系统设计说明书

学院：计算机与大数据学院

年级：2021 级

专业：软件工程

团队：流光悦影

# 目录

一、引言.....	3
1.1 编写目的.....	3
1.2 背景.....	3
二、体系结构设计.....	3
2.1 软件总体结构.....	3
2.2 技术设计.....	5
2.3 系统功能模块图.....	6
2.3.1 前台功能模块图.....	6
2.3.3 后台功能模块图.....	7
三、设计类图和接口设计.....	7
3.1 设计类图.....	7
3.2 用例图.....	11
3.3 接口设计.....	11
3.3.1 用户接口.....	11
3.3.2 视频接口.....	29
3.3.3 弹幕接口.....	42
3.3.4 通用接口.....	46
3.3.5 搜索接口.....	53
3.3.6 反馈接口.....	56
3.3.7 推荐接口.....	57
3.3.8 评论接口.....	63
四、系统安全和权限设计.....	67
4.1 系统安全.....	67
4.2 权限设计.....	69
五、验收标准.....	69
5.1 界面验收标准.....	69
5.1.1 注册登录和重置密码界面.....	69
5.1.2 首页.....	70
5.1.3 视频详情界面.....	70
5.1.4 搜索界面.....	70
5.1.5 个人主页界面.....	71
5.1.6 视频上传界面.....	71
5.1.7 反馈界面.....	71
5.1.8 举报审核界面.....	71
5.1.9 反馈处理界面.....	72
5.1.10 视频优先级调整界面.....	72
5.2 功能验收标准.....	72

# 一、引言

## 1.2 背景

（下列“亚文化”均指代“ACGN 亚文化”，后不再赘述。）

当前，多数大众化的视频播放平台，如 Bilibili、抖音和 AcFun，涵盖了丰富多样的视频内容，并且拥有庞大的用户群体。然而，对于一些核心的亚文化爱好者来说，这些广受欢迎的平台也有它们无法触及的痛点，例如社区环境的扩大带来的相关影响，以及信息传播机制对社区环境的破坏等。

鉴于此，我们小组提出了打造一个专为亚文化爱好者服务的视频分享及讨论平台的想法。该平台的宗旨是构建一个集视频分享、观看和社区讨论为一体的环境，专注于亚文化领域。通过使用先进的互联网技术进行视频搜索和整理，结合相关算法推荐系统，我们希望用户能够迅速而精准地找到他们感兴趣的视频内容。除了聚集和推广已有的视频资源，我们还鼓励用户自创内容，上传专属的视频作品，从而培养一个积极正向、充满热情和创造力的社区氛围。通过这样的方式，不仅能提供给亚文化爱好者一个定制化的内容观看体验，还能助力于社区内知识的传播和文化的交流。在这里，每个人都可以是观众，也可以是创作者，共同维护和丰富这个平台的多样性和独特性。

## 1.1 编写目的

本文档的目的是详细地介绍“moe”所包含的需求，以便客户能够确认产品的确切需求以及开发人员能够根据需求设计编码，以下叙述将结合文字描述、类图等来描述“moe”的功能、性能、用户界面、运行环境以及针对用户操作给出的各种响应。

# 二、体系结构设计

## 2.1 软件总体结构

1. 前端：

- ✧ 负责与用户直接交互的界面。
- ✧ 使用 Typescript、React 等技术栈实现动态的页面效果。
- ✧ Mobx 用于状态管理，Vite 和 Less 提高开发效率和样式处理。
- ✧ HTTPS 保障数据传输的安全。

## 2. 接入层:

✧ 使用 Nginx 作为反向代理服务器，增强安全性和负载能力，同时可以提供 CDN 加速服务来提高全球用户的访问速度。

## 3. CI/CD:

✧ 利用 Vercel、Github Action 等工具自动化代码的构建、测试和部署流程。

✧ Maven、Docker 等工具确保开发环境与生产环境的一致性。

✧ JUnit、JMeter 分别用于单元测试和性能测试，保障代码质量。

## 4. 后端部分:

✧ 实现业务逻辑，处理前端发来的请求。

✧ 可能包括特定的服务，如短信发送、消息队列（RabbitMQ）处理异步消息等。

## 5. 中间件:

✧ 提供附加的服务能力，如 CRUD 操作、搜索和文件服务。

✧ 阿里云 OSS 用于存储大量的静态文件。

## 6. 数据库技术:

✧ MySQL、Redis 和 Elasticsearch 分别用于数据存储、高速缓存和高效搜索。

✧ 保障数据的可靠性、快速访问和搜索能力。



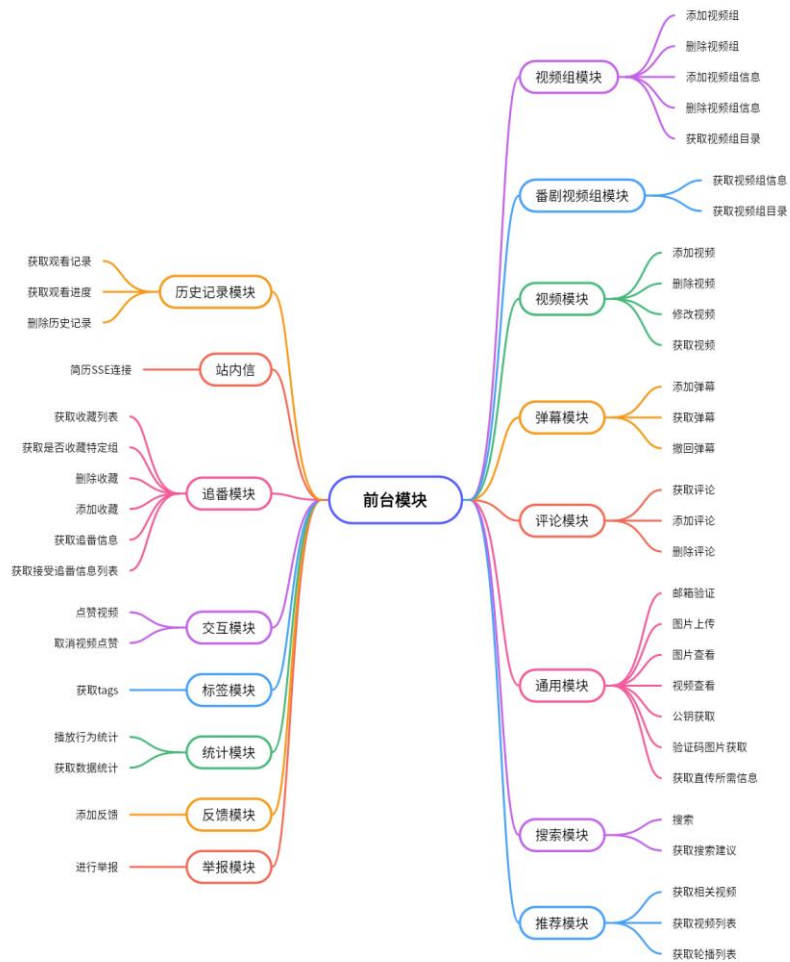
## 2.2 技术设计

✧ 前端: React、React Router、Less、Vite、Antd、ProComponent、TypeScript、Zustand

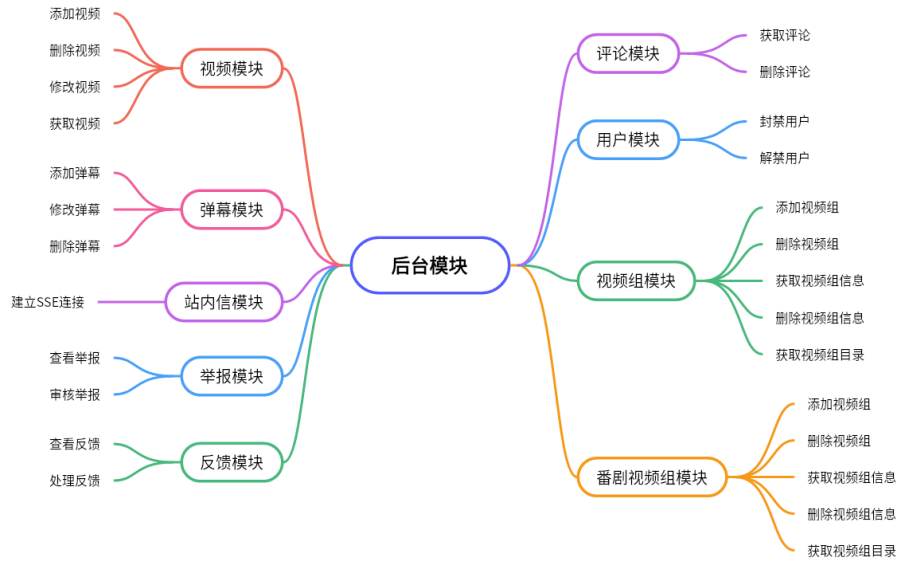
✧ 后端: SpringBoot、MySQL、Redis、Elastic Search、RabbitMq

## 2.3 系统功能模块图

### 2.3.1 前台功能模块图



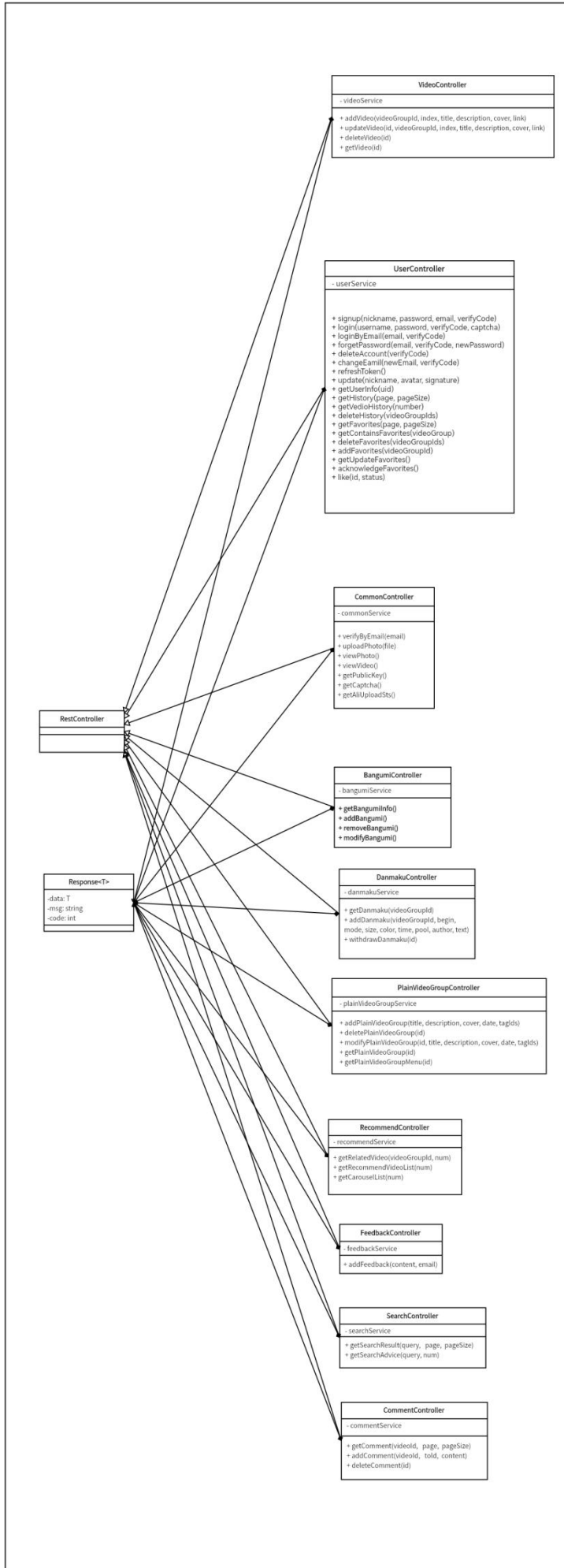
### 2.3.3 后台功能模块图



## 三、设计类图和接口设计

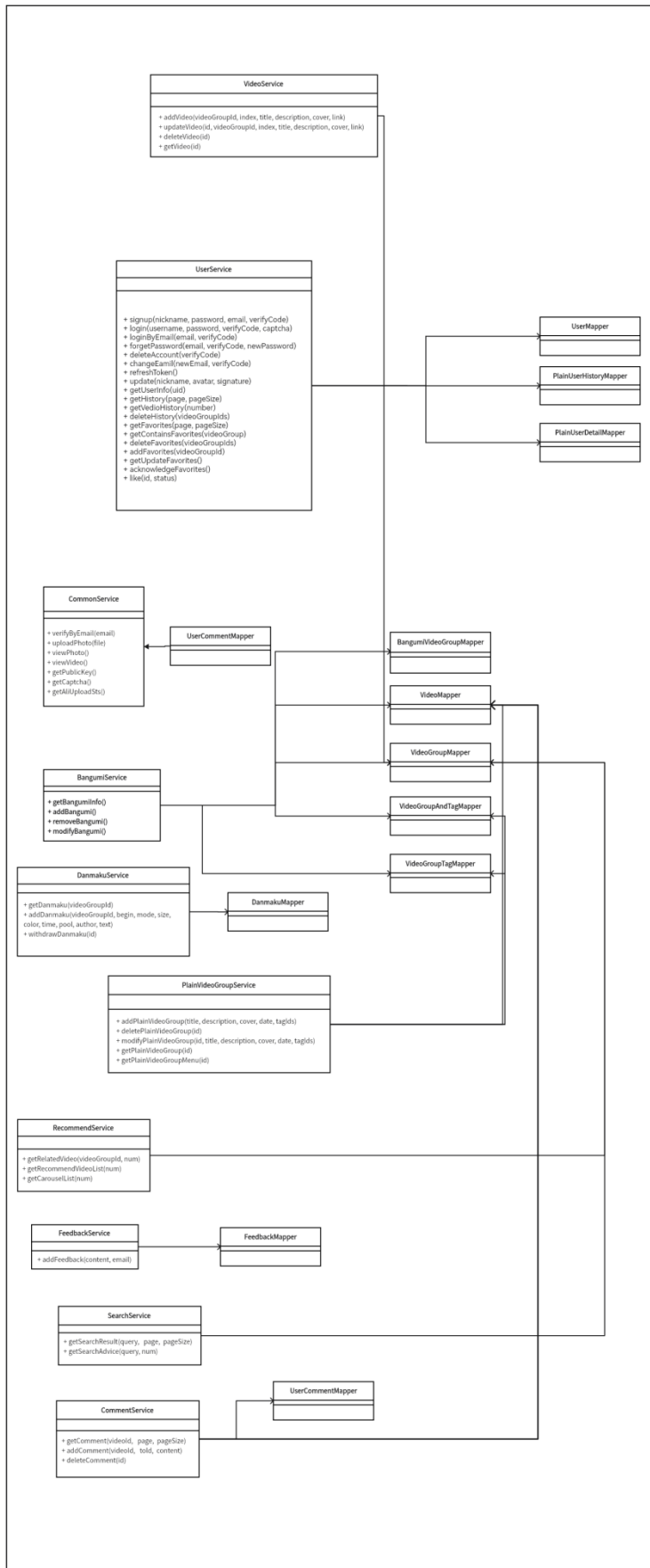
### 3.1 设计类图

Controller 层:

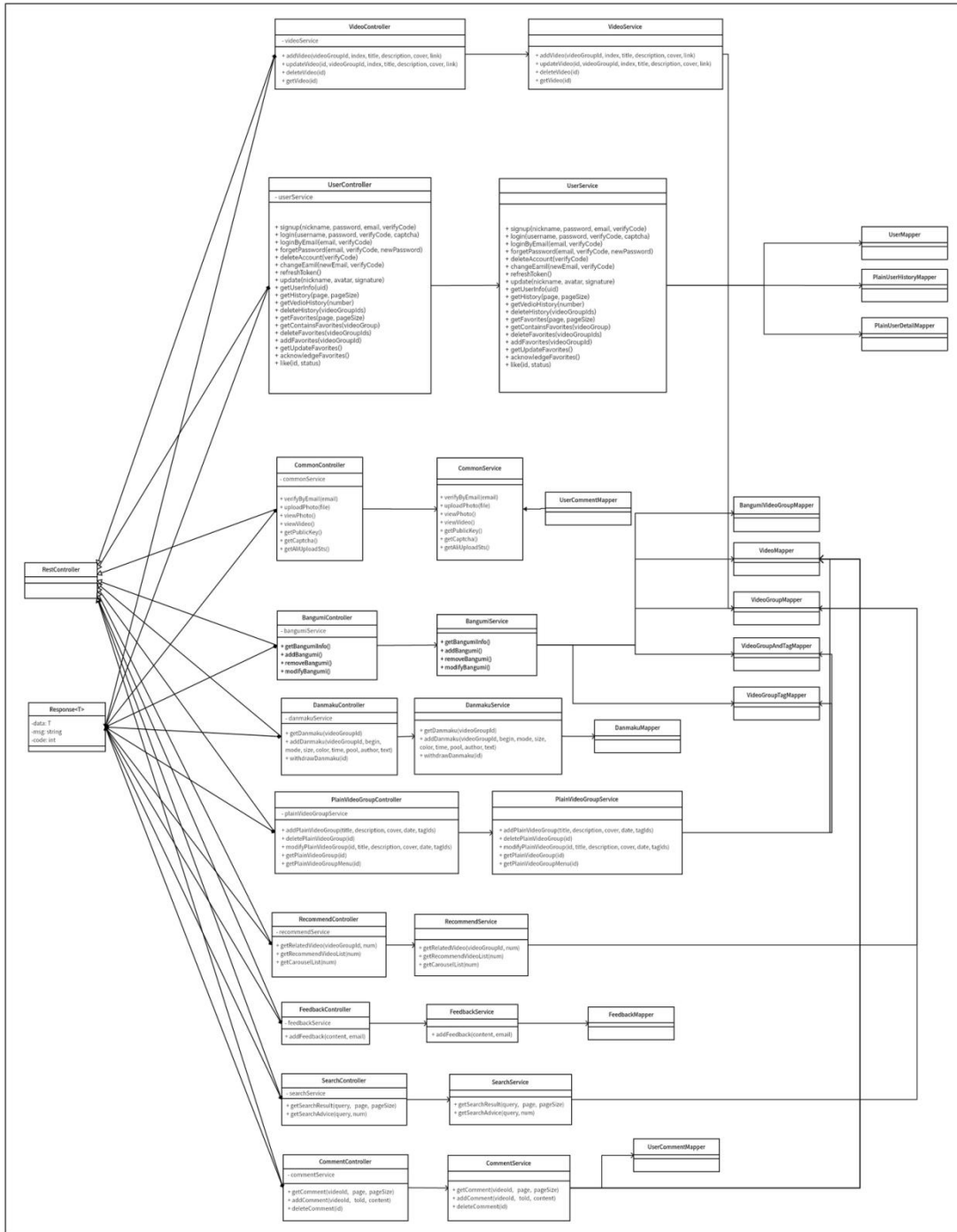




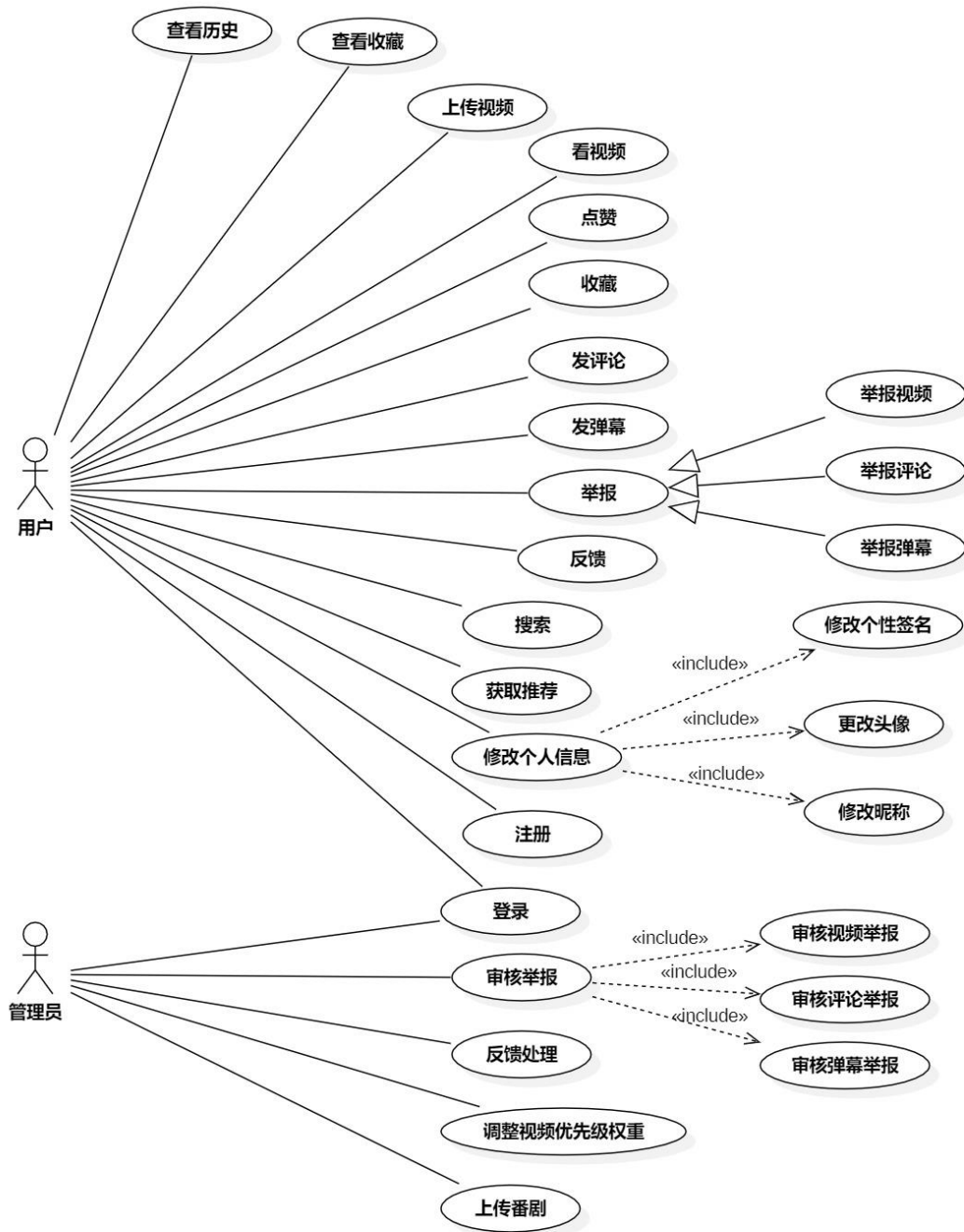
# Service / Mapper 层:



# 总体类图:



## 3.2 用例图



## 3.3 接口设计

### 3.3.1 用户接口

◇ 用户注册

请求 URL:

```
/user/signup
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	生成代码	示例
nickname	string	必需	<pre>{   "nickname": "曾丽",   "password": "magnasint",   "email": "c.yvbinooqii@qq.com",   "verifyCode": "176194" }</pre>
password	string	必需	
email	string	必需	
verifyCode	string 6位邮箱验证码	必需	

返回参数:

数据结构	生成代码
<p>code integer 编码</p> <p>200成功, 0和其它数字为失败</p> <p>默认值: <input type="text" value="200"/></p>	必需
<p>msg string 错误信息</p>	必需
<p>data string 数据</p> <p>默认值: <input type="text" value="ok"/></p>	必需

返回示例:

示例
<pre>{   "code": 200,   "msg": "string",   "data": "token" }</pre>

✧ 用户登录

请求 URL:

```
/user/login
```

请求方式:

POST

请求参数:

Body 参数	application/json	生成代码	示例
<code>username</code>	string	必需	<pre>{   "username": "武杰",   "password": "pariatorexipsum",   "verifyCodeId": "dfadsfaw",   "captcha": "xqru" }</pre>
<code>password</code>	string	必需	
<code>verifyCodeId</code>	string 图像验证码id	必需	
<code>captcha</code>	string 4位图像验证码	必需	

返回参数:

数据结构	生成代码
<code>code</code> integer 编码 200成功, 0和其它数字为失败 默认值: <input type="text" value="200"/>	必需
<code>msg</code> string 错误信息	必需
<code>data</code> string token 默认值: <input type="text" value="token"/>	必需

返回示例:

示例
<pre>{   "code": 200,   "msg": "string",   "data": "token" }</pre>

✧ 用户使用邮箱登录

请求 URL:

/user/login-by-email

请求方式:

POST

请求参数:

Body 参数	application/json	生成代码	示例
<code>email</code>	string	必需	<pre>{   "email": "d.cqvoxul@qq.com",   "verifyCode": "576838" }</pre>
<code>verifyCode</code>	string 6位邮箱验证码	必需	

返回参数:

数据结构	生成代码
<p><code>code</code> integer 编码 ----- 必需</p> <p>200成功, 0和其它数字为失败</p> <p>默认值: <input type="text" value="200"/></p>	
<p><code>msg</code> string 错误信息 ----- 必需</p>	
<p><code>data</code> string token ----- 必需</p> <p>默认值: <input type="text" value="token"/></p>	

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "token"
}
```

✧ 忘记密码

请求 URL:

/user/forget-password

请求方式:

POST

请求参数:

Body 参数	application/json	生成代码	示例
<code>email</code>	string -----	必需	<pre>{   "email": "y.ynfhbb@qq.com",   "verifyCode": "547317",   "newPassword": "aliquisin" }</pre>
<code>verifyCode</code>	string 6位邮箱验证码	必需	
<code>newPassword</code>	string	必需	

返回参数:

数据结构		生成代码
<code>code</code>	<code>integer</code> 编码	必需
200成功, 0和其它数字为失败		
默认值:	<code>200</code>	
<code>msg</code>	<code>string</code> 错误信息	必需
<code>data</code>	<code>string</code> 数据	必需
默认值:	<code>ok</code>	

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 用户注销账号

请求 URL:

```
/user/delete-account
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	生成代码	示例
<code>verifyCode</code>	<code>string</code> 6位邮箱验证码	必需	<pre>{   "verifyCode": "135149" }</pre>

返回参数:

数据结构

生成代码

`code` integer 编码 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

`data` string 数据 必需

默认值:

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 用户邮箱

请求 URL:

```
/user/change-email
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	生成代码	示例
<code>newEmail</code>	string	必需	<pre>{   "newEmail": "b.bghtgvokv@qq.com",   "verifyCode": "851339" }</pre>
<code>verifyCode</code>	string 新邮箱收到的6位验证码	必需	

返回参数:



<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	数据	必需
默认值:	<input type="text" value="ok"/>		

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

◇ 用户续登

请求 URL:

```
/user/login/refresh
```

请求方式:

```
GET
```

请求参数:

```
无
```

返回参数:

数据结构		生成代码
<code>code</code>	<code>integer</code> 编码 200成功, 0和其它数字为失败 默认值: <input type="text" value="200"/>	必需
<code>msg</code>	<code>string</code> 错误信息	必需
<code>data</code>	<code>string</code> token 默认值: <input type="text" value="token"/>	必需

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "token"
}
```

✧ 获取用户信息

请求 URL:

```
/plain-user
```

请求方式:

```
GET
```

请求参数:

Query 参数

```
uid string 必需
```

返回参数:

<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	<code>string</code>	错误信息	必需
▼ <code>data</code>	<code>object</code>	(PlainUserVO) 数据	必需
<code>userId</code>	<code>number</code>		必需
<code>nickname</code>	<code>string</code>		必需
<code>avatar</code>	<code>string</code>		必需
<code>signature</code>	<code>string</code>		必需

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "userId": 0,
    "nickname": "string",
    "avatar": "string",
    "signature": "string"
  }
}
```

✧ 获取用户观看历史记录

请求 URL:

```
/plain-user/history
```

请求方式:

```
GET
```

请求参数:

Query 参数

`page` `integer` 必需

`pageSize` `integer` 必需

返回参数:

<code>code</code>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	string	错误信息	必需
▼ <code>data</code>	object {2}	数据	必需
└─ <code>total</code>	integer		必需
└─ <code>records</code>	array[object (HistoryVO) {8}]		必需

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "total": 0,
    "records": [
      {
        "id": 0,
        "videoGroupId": 0,
        "videoId": 0,
        "lastWatchTime": "string",
        "videoGroupTitle": "string",
        "uploaderId": "string",
        "videoGroupCover": "string",
        "videoTitle": "string"
      }
    ]
  }
}
```

✧ 获取用户视频组上次看到哪集

请求 URL:

```
/plain-user/history/video
```

请求方式:

```
GET
```

请求参数:

## Query 参数

videoid number 可选

返回参数:

数据结构

生成代码

<code>code</code>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	string	错误信息	必需
▼ <code>data</code>	object (HistoryVO)	数据	必需
<code>id</code>	integer		可选
<code>videoGroupId</code>	integer		可选
<code>videoid</code>	integer		可选
<code>lastWatchTime</code>	string		可选
<code>videoGroupTitle</code>	string		可选
<code>uploaderId</code>	string		可选
<code>videoGroupCover</code>	string		可选
<code>videoTitle</code>	string		可选

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "id": 0,
    "videoGroupId": 0,
    "videoId": 0,
    "lastWatchTime": "string",
    "videoGroupTitle": "string",
    "uploaderId": "string",
    "videoGroupCover": "string",
    "videoTitle": "string"
  }
}
```

✧ 删除用户观看历史

请求 URL:

/plain-user/history/delete

请求方式:

POST

请求参数:

Query 参数

`videoGroupIds` array[string] 必需

返回参数:

数据结构

[生成代码](#)

`code` integer 编码 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

`data` string 数据 必需

默认值:

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 获取用户收藏列表

请求 URL:

/plain-user/favorites

请求方式:

GET

请求参数:

## Query 参数

`page` integer 必需

`pageSize` integer 必需

## 返回参数:

数据结构

[生成代码](#)

`code` integer 编码 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

▼ `data` object (2) 数据 必需

├── `total` integer 必需

└── `records` object (VideoGroupInfoVO) 必需

## 返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "total": 0,
    "records": {
      "videoGroupId": 0,
      "title": "string",
      "uploader": {
        "userId": 0,
        "nickname": "string",
        "avatar": "string"
      },
      "cover": "string",
      "description": "string",
      "type": 0,
      "tags": [
        {
          "id": 0,
          "tagName": "string"
        }
      ]
    }
  }
}
```

✧ 获取用户是否收藏特定视频组

请求 URL:

```
/plain-user/favorites/contains
```

请求方式:

GET

请求参数:

Query 参数

videoGroupId integer 可选

返回参数:

数据结构

[生成代码](#)

code integer 编码 必需

200成功, 0和其它数字为失败

默认值: 200

msg string 错误信息 必需

data string 数据 必需

默认值: ok

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 删除用户收藏

请求 URL:

/plain-user/favorites/delete

请求方式:

POST

请求参数:



## Query 参数

`videoGroupIds` array[string] 必需

### 返回参数:

数据结构

[生成代码](#)

`code` integer 编码 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

`data` string 数据 必需

默认值:

### 返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

### ✧ 添加用户收藏

请求 URL:

```
/plain-user/favorites/add
```

请求方式:

```
GET
```

请求参数:

Query 参数

`videoGroupId` integer 必需

返回参数:

`code` integer 编码 ----- 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

`data` string 数据 必需

默认值:

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 获取追番更新信息

请求 URL:

```
/plain-user/favorites/get-updates
```

请求方式:

```
GET
```

请求参数:

```
无
```

返回参数:

<code>code</code>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	string	错误信息	必需
▼ <code>data</code>	array[object (VideoVO) {7}]	数据	必需
<code>id</code>	string	ID 编号	必需
<code>videoGroupId</code>	string		必需
<code>index</code>	integer		必需
<code>title</code>	string		必需
<code>description</code>	string		必需
<code>cover</code>	string		必需
<code>link</code>	string		必需

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "id": "string",
      "videoGroupId": "string",
      "index": 0,
      "title": "string",
      "description": "string",
      "cover": "string",
      "link": "string"
    }
  ]
}
```

✧ 确认已经接收最新番剧信息

请求 URL:

```
/plain-user/favorites/acknowledge
```

请求方式:

```
POST
```

请求参数:

无

返回参数:

数据结构

生成代码

**code** integer 编码 必需

200成功, 0和其它数字为失败

默认值: 200

**msg** string 错误信息 必需

**data** string 数据 必需

默认值: ok

返回示例:

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 视频组点赞/取消点赞

请求 URL:

/plain-user/update

请求方式:

POST

请求参数:

Body 参数 application/json

生成代码

示例

**id** string ID 编号 必需

**status** integer 状态 必需

枚举值: 0 1

```
{
  "id": "string",
  "status": 0
}
```

返回参数:

`code` integer 编码 ..... 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

`data` string 数据 必需

默认值:

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

### 3.3.2 视频接口

✧ 添加视频

请求 URL:

/video/add

请求方式:

POST

请求参数:

Body 参数	application/json	生成代码	示例
<code>videoGroupId</code>	string	必需	<pre>{   "videoGroupId": "string",   "index": "string",   "title": "string",   "description": "string",   "cover": "string",   "link": "string" }</pre>
<code>index</code>	string	必需	
<code>title</code>	string	必需	
<code>description</code>	string	必需	
<code>cover</code>	string	必需	
<code>link</code>	string	必需	

返回参数:

## 数据结构

<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	视频id	必需

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "string"
}
```

## ✧ 修改视频

### 请求 URL:

/video/update

### 请求方式:

POST

### 请求参数:

Body 参数	application/json	示例
<code>id</code>	<code>string</code> ID 编号	必需
<code>videoGroupId</code>	<code>string</code>	可选
<code>index</code>	<code>string</code>	可选
<code>title</code>	<code>string</code>	可选
<code>description</code>	<code>string</code>	可选
<code>cover</code>	<code>string</code>	可选
<code>link</code>	<code>string</code>	可选

```
{
  "videoGroupId": "string",
  "index": "string",
  "title": "string",
  "description": "string",
  "cover": "string",
  "link": "string"
}
```

### 返回参数:

### 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	数据	必需
默认值:	<input type="text" value="ok"/>		

### 返回示例:

#### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

### ✧ 删除视频

#### 请求 URL:

```
/video/delete
```

#### 请求方式:

```
POST
```

#### 请求参数:

Body 参数	application/json	</> 生成代码	示例
<b>id</b>	string	ID 编号	必需
			<pre>{   "id": "string" }</pre>

#### 返回参数:

## 数据结构

**code** integer 编码

200成功, 0和其它数字为失败

默认值:

必需

**msg** string 错误信息

必需

**data** string 数据

默认值:

必需

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## ✧ 获取视频

### 请求 URL:

/video

### 请求方式:

GET

### 请求参数:

#### Query 参数

**id** string ID 编号

### 返回参数:



## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
▼ <b>data</b>	object (VideoVO)	数据	必需
<b>id</b>	string	ID 编号	必需
<b>videoGroupId</b>	string		必需
<b>index</b>	integer		必需
<b>title</b>	string		必需
<b>description</b>	string		必需
<b>cover</b>	string		必需
<b>link</b>	string		必需

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "id": "string",
    "videoGroupId": "string",
    "index": 0,
    "title": "string",
    "description": "string",
    "cover": "string",
    "link": "string"
  }
}
```

## ✧ 添加普通视频组

请求 URL:

```
/plain-video-group/add
```

请求方式:

```
POST
```

请求参数:

Body 参数		multipart/form-data
title	string	必需
description	string	必需
cover	file	视频封面 必需
date	string	出版日期 必需
tagIds	array[string]	必需

返回参数:

数据结构		
code	integer	编码 必需
200成功, 0和其它数字为失败		
默认值:	200	
msg	string	错误信息 必需
data	integer	视频组id 必需

返回示例:

示例
<pre>{   "code": 200,   "msg": "string",   "data": 0 }</pre>

✧ 删除普通视频组

请求 URL:

```
/plain-video-group/delete
```

请求方式:

```
POST
```

请求参数:

## Query 参数

**id** integer ID 编号 必需

## 返回参数:

### 数据结构

**code** integer 编码 必需

200成功, 0和其它数字为失败

默认值:

**msg** string 错误信息 必需

**data** string 数据 必需

默认值:

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## ✧ 修改普通视频组信息

请求 URL:

```
/plain-video-group/update
```

请求方式:

```
POST
```

请求参数:

Body 参数 multipart/form-data

**id** string 必需

**title** string 可选

**description** string 可选

**cover** string 视频封面 可选

**date** string 出版日期 可选

**tagIds** string 可选

返回参数:

数据结构

<b>code</b> integer 编码	必需
200成功, 0和其它数字为失败	
默认值: 200	
<b>msg</b> string 错误信息	必需
<b>data</b> string 数据	必需
默认值: ok	

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 获取普通视频组信息

请求 URL:

/plain-video-group

请求方式:

GET

请求参数:

## Query 参数

`id` `string` ID 编号 **必需**

## 返回参数:

数据结构 生成代码

- `code` `integer` 编码 **必需**  
200成功, 0和其它数字为失败  
默认值:
- `msg` `string` 错误信息 **必需**
- `data` `object` (VideoGroupInfoVO) 数据 **必需**
  - `videoGroupId` `integer` **必需**
  - `title` `string` **必需**
  - `uploader` `object` (3) **必需**
    - `userId` `integer` **必需**
    - `nickname` `string` **必需**
    - `avatar` `string` **必需**
  - `cover` `string` **必需**
  - `description` `string` **必需**
  - `type` `integer` **必需**  
枚举值:
  - `tags` `array` [object (2)] **必需**
    - `id` `integer`
    - `tagName` `string`

## 返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "videoGroupId": 0,
    "title": "string",
    "uploader": {
      "userId": 0,
      "nickname": "string",
      "avatar": "string"
    },
    "cover": "string",
    "description": "string",
    "type": 0,
    "tags": [
      {
        "id": 0,
        "tagName": "string"
      }
    ]
  }
}
```

✧ 获得普通视频组目录

请求 URL:

```
/plain-video-group/contents
```

请求方式:

```
GET
```

请求参数:

Query 参数

**id** string ID 编号 可选

返回参数:

数据结构			
<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值: <input type="text" value="200"/>			
<code>msg</code>	<code>string</code>	错误信息	必需
▼ <code>data</code>	<code>array</code> [object {5}]	数据	必需
<code>videoId</code>	<code>string</code>		必需
<code>videoGroupId</code>	<code>string</code>		必需
<code>index</code>	<code>string</code>		必需
<code>title</code>	<code>string</code>		必需
<code>videoCover</code>	<code>string</code>		必需

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "videoId": "string",
      "videoGroupId": "string",
      "index": "string",
      "title": "string",
      "videoCover": "string"
    }
  ]
}
```

✧ 获取番剧视频组信息

请求 URL:

`/bangumi-video-group`

请求方式:

GET

请求参数:

Query 参数

`id` `string` ID 编号 必需

## 返回参数:

数据结构		生成代码
<b>code</b>	integer 编码 200成功, 0和其它数字为失败 默认值: <input type="text" value="200"/>	必需
<b>msg</b>	string 错误信息	必需
▼ <b>data</b>	object (BangumiVideoGroupVO) 数据	必需
<b>videoGroupId</b>	integer	必需
<b>title</b>	string	必需
▼ <b>uploader</b>	object {3}	必需
<b>userId</b>	integer	必需
<b>nickname</b>	string	必需
<b>avatar</b>	string	必需
<b>cover</b>	string	必需
<b>description</b>	string	必需
<b>type</b>	integer 枚举值: <input type="text" value="0"/> <input type="text" value="1"/>	必需
<b>releaseTime</b>	string	必需
<b>updateAtAnnouncement</b>	string	必需
<b>status</b>	integer 枚举值: <input type="text" value="0"/> <input type="text" value="1"/>	必需
▼ <b>tags</b>	array [object {2}]	必需
<b>id</b>	integer	可选
<b>tagName</b>	string	可选

## 返回示例:



## 示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "videoGroupId": 0,
    "title": "string",
    "uploader": {
      "userId": 0,
      "nickname": "string",
      "avatar": "string"
    },
    "cover": "string",
    "description": "string",
    "type": 0,
    "releaseTime": "string",
    "updateAtAnnouncement": "string",
    "status": 0,
    "tags": [
      {
        "id": 0,
        "tagName": "string"
      }
    ]
  }
}
```

## ✧ 获得番剧视频组目录

请求 URL:

```
/bangumi-video-group/contents
```

请求方式:

```
GET
```

请求参数:

### Query 参数

**id** string ID 编号 可选

返回参数:

## 数据结构

<code>code</code>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	string	错误信息	必需
▼ <code>data</code>	array [object {5}]	数据	必需
├── <code>videoid</code>	string		必需
├── <code>videoGroupId</code>	string		必需
├── <code>index</code>	string		必需
├── <code>title</code>	string		必需
└── <code>videoCover</code>	string		必需

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "videoId": "string",
      "videoGroupId": "string",
      "index": "string",
      "title": "string",
      "videoCover": "string"
    }
  ]
}
```

### 3.3.3 弹幕接口

◇ 获取弹幕

请求 URL:

```
/video/bulletchat
```

请求方式:

```
GET
```

请求参数:

### Query 参数

`videoGroupId` string 必需

返回参数:

#### 数据结构

`code` integer 编码 必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息 必需

∨ `data` array[object (BulletChatVO) {8}] 数据 必需

`begin` number 必需

`mode` number 必需

`size` number 必需

`color` string 必需

`time` number 必需

`pool` number 必需

`author` string 必需

`text` string 必需

返回示例:

## 示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "begin": 0,
      "mode": 0,
      "size": 0,
      "color": "string",
      "time": 0,
      "pool": 0,
      "author": "string",
      "text": "string"
    }
  ]
}
```

## ✧ 添加弹幕

请求 URL:

```
/video/bulletchat/add
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	示例	
videoGroupId	string	必需	<pre>{   "videoGroupId": "string",   "begin": 0,   "mode": 0,   "size": 0,   "color": "string",   "time": 0,   "pool": 0,   "author": "string",   "text": "string" }</pre>
begin	number	必需	
mode	number	必需	
size	number	必需	
color	string	必需	
time	number	必需	
pool	number	必需	
author	string	必需	
text	string	必需	

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	弹幕id	必需

返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "string"
}
```

✧ 撤回弹幕

请求 URL:

```
/video/bulletchat/delete
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	示例		
<b>id</b>	string	ID 编号	必需	{ "id": "string" }

返回参数:

数据结构			
<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<code>200</code>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	数据	必需
默认值:	<code>ok</code>		

返回示例:

示例
<pre>{   "code": 200,   "msg": "string",   "data": "ok" }</pre>

### 3.3.4 通用接口

✧ 邮箱验证

请求 URL:

`/common/verify-email`

请求方式:

POST

请求参数:

Body 参数	application/json	必需	示例
<code>email</code>	<code>string</code>	必需	示例 <pre>{   "email": "string" }</pre>

返回参数:

## 数据结构

`code` integer 编码

必需

200成功, 0和其它数字为失败

默认值:

`msg` string 错误信息

必需

`data` string 数据

必需

默认值:

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## ✧ 图片上传

### 请求 URL:

/common/upload-image

### 请求方式:

POST

### 请求参数:

Body 参数 multipart/form-data

`file` file 必需

### 返回参数:

## 数据结构

<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<code>200</code>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	数据	必需
默认值:	<code>ok</code>		

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## ◇ 图片查看

### 请求 URL:

```
/common/image/**
```

### 请求方式:

```
GET
```

### 请求参数:

```
无
```

### 返回参数:



## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	数据	必需
默认值:	<input type="text" value="ok"/>		

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

◇ 视频查看

请求 URL:

```
/common/video/**
```

请求方式:

```
GET
```

请求参数:

```
无
```

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	数据	必需
默认值:	<input type="text" value="ok"/>		

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

✧ 获得公钥

请求 URL:

```
/common/public-key
```

请求方式:

```
GET
```

请求参数:

```
无
```

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	公钥	必需
返回Base64编码的公钥 为pem格式去掉开头和结尾的-----BEGIN PUBLIC KEY-----和-----END PUBLIC KEY-----			

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "string"
}
```

◇ 获取验证码图片

请求 URL:

```
/common/captcha
```

请求方式:

```
GET
```

请求参数:

```
无
```

返回参数:

## 数据结构

<code>code</code>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	string	错误信息	必需
▼ <code>data</code>	object (2)	数据	必需
<code>verifyCodeId</code>	string		必需
<code>captcha</code>	string	验证码base64图片	必需

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "verifyCodeId": "string",
    "captcha": "string"
  }
}
```

## ✧ 获取 ali 直传所需信息

### 请求 URL:

```
/common/get-ali-upload-sts
```

### 请求方式:

```
GET
```

### 请求参数:

```
无
```

### 返回参数:

数据结构			
<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值: <input type="text" value="200"/>			
<code>msg</code>	<code>string</code>	错误信息	必需
▼ <code>data</code>	<code>object</code>	(4) 数据	必需
<code>AccessKeyId</code>	<code>string</code>		必需
<code>AccessKeySecret</code>	<code>string</code>		必需
<code>SecurityToken</code>	<code>string</code>		必需
<code>endpoint</code>	<code>string</code>		必需

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "AccessKeyId": "string",
    "AccessKeySecret": "string",
    "SecurityToken": "string",
    "endpoint": "string"
  }
}
```

### 3.3.5 搜索接口

◇ 搜索

请求 URL:

/search

请求方式:

GET

请求参数:

## Query 参数

**q** string 必需

**page** string 必需

**pageSize** string 必需

## 返回参数:

数据结构

[生成代码](#)

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	object (2)	数据	必需
<b>total</b>	integer		必需
<b>records</b>	array [object (VideoGroupInfoVO) (7)]		必需
<b>videoGroupIid</b>	integer		必需
<b>title</b>	string		必需
<b>uploader</b>	object (3)		必需
<b>userId</b>	integer		必需
<b>nickname</b>	string		必需
<b>avatar</b>	string		必需
<b>cover</b>	string		必需
<b>description</b>	string		必需
<b>type</b>	integer		必需
枚举值:	<input type="text" value="0"/> <input type="text" value="1"/>		
<b>tags</b>	array [object (2)]		必需
<b>id</b>	integer		可选
<b>tagName</b>	string		可选

## 返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "total": 0,
    "records": [
      {
        "videoGroupId": 0,
        "title": "string",
        "uploader": {
          "userId": 0,
          "nickname": "string",
          "avatar": "string"
        },
        "cover": "string",
        "description": "string",
        "type": 0,
        "tags": [
          {
            "id": 0,
            "tagName": "string"
          }
        ]
      }
    ]
  }
}
```

◇ 获得搜索建议

请求 URL:

```
/search/suggestion
```

请求方式:

```
GET
```

请求参数:

Query 参数

**q** string 必需

**num** string 必需

返回参数:

## 数据结构

<b>code</b>	integer 编码	必需
200成功, 0和其它数字为失败		
默认值:	<input type="text" value="200"/>	
<b>msg</b>	string 错误信息	必需
<b>data</b>	array[string] 数据	必需

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    "string"
  ]
}
```

## 3.3.6 反馈接口

✧ 添加反馈

请求 URL:

```
/feedback/add
```

请求方式:

```
POST
```

请求参数:

Body 参数 application/json	示例
<b>content</b> string <span>必需</span>	<pre>{   "content": "string",   "email": "string" }</pre>
<b>email</b> string <span>可选</span>	

返回参数:



## 数据结构

<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<code>200</code>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	数据	必需
默认值:	<code>ok</code>		

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

### 3.3.7 推荐接口

✧ 获取相关视频

请求 URL:

```
/video-group/related
```

请求方式:

```
GET
```

请求参数:

Query 参数

<code>videoGroupId</code>	<code>string</code>	必需
<code>num</code>	<code>string</code>	必需

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
√ <b>data</b>	array[object (VideoGroupInfoVO) {7}]	数据	必需
├─ <b>videoGroupId</b>	integer		必需
├─ <b>title</b>	string		必需
├─ √ <b>uploader</b>	object {3}		必需
│   └─ <b>userId</b>	integer		必需
│   └─ <b>nickname</b>	string		必需
│   └─ <b>avatar</b>	string		必需
├─ <b>cover</b>	string		必需
├─ <b>description</b>	string		必需
├─ <b>type</b>	integer		必需
│   枚举值:	<input type="text" value="0"/> <input type="text" value="1"/>		
├─ √ <b>tags</b>	array [object {2}]		必需
│   └─ <b>id</b>	integer		<input type="text" value="可选"/>
│   └─ <b>tagName</b>	string		<input type="text" value="可选"/>

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "videoGroupId": 0,
      "title": "string",
      "uploader": {
        "userId": 0,
        "nickname": "string",
        "avatar": "string"
      },
      "cover": "string",
      "description": "string",
      "type": 0,
      "tags": [
        {
          "id": 0,
          "tagName": "string"
        }
      ]
    }
  ]
}
```

✧ 获取推荐视频列表

请求 URL:

```
/video-group/recommend
```

请求方式:

```
GET
```

请求参数:

Query 参数

```
num string 必需
```

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
▼ <b>data</b>	array[object (VideoGroupInfoVO) {7}]	数据	必需
├─ <b>videoGroupId</b>	integer		必需
├─ <b>title</b>	string		必需
├─ ▼ <b>uploader</b>	object {3}		必需
│   └─ <b>userId</b>	integer		必需
│   └─ <b>nickname</b>	string		必需
│   └─ <b>avatar</b>	string		必需
├─ <b>cover</b>	string		必需
├─ <b>description</b>	string		必需
├─ <b>type</b>	integer		必需
│   枚举值: <input type="text" value="0"/> <input type="text" value="1"/>			
└─ ▼ <b>tags</b>	array [object {2}]		必需
│   └─ <b>id</b>	integer		<input type="text" value="可选"/>
│   └─ <b>tagName</b>	string		<input type="text" value="可选"/>

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "videoGroupId": 0,
      "title": "string",
      "uploader": {
        "userId": 0,
        "nickname": "string",
        "avatar": "string"
      },
      "cover": "string",
      "description": "string",
      "type": 0,
      "tags": [
        {
          "id": 0,
          "tagName": "string"
        }
      ]
    }
  ]
}
```

◇ 获取轮播列表

请求 URL:

```
/video-group/carousel
```

请求方式:

```
GET
```

请求参数:

Query 参数

```
num string 必需
```

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
√ <b>data</b>	array[object (VideoGroupInfoVO) (7)]	数据	必需
├─	<b>videoGroupId</b>	integer	必需
├─	<b>title</b>	string	必需
├─	√ <b>uploader</b>	object (3)	必需
│	├─ <b>userId</b>	integer	必需
│	├─ <b>nickname</b>	string	必需
│	└─ <b>avatar</b>	string	必需
├─	<b>cover</b>	string	必需
├─	<b>description</b>	string	必需
├─	<b>type</b>	integer	必需
│	枚举值:	<input type="text" value="0"/> <input type="text" value="1"/>	
└─	√ <b>tags</b>	array [object (2)]	必需
│	├─ <b>id</b>	integer	<input type="text" value="可选"/>
│	└─ <b>tagName</b>	string	<input type="text" value="可选"/>

返回示例:

示例

```
{
  "code": 200,
  "msg": "string",
  "data": [
    {
      "videoGroupId": 0,
      "title": "string",
      "uploader": {
        "userId": 0,
        "nickname": "string",
        "avatar": "string"
      },
      "cover": "string",
      "description": "string",
      "type": 0,
      "tags": [
        {
          "id": 0,
          "tagName": "string"
        }
      ]
    }
  ]
}
```

### 3.3.8 评论接口

✧ 获取评论

请求 URL:

```
/video/comment
```

请求方式:

```
GET
```

请求参数:

## Query 参数

**videoid** string 必需

**page** string 必需

**pageSize** string 必需

## 返回参数:

数据结构 </> 生成代码

<b>code</b> integer 编码	必需
200成功, 0和其它数字为失败	
默认值: <input type="text" value="200"/>	
<b>msg</b> string 错误信息	必需
▼ <b>data</b> object {2} 数据	必需
<b>total</b> integer	必需
▼ <b>records</b> array[array (CommentBlockVO) {1}]	必需
▼ object {6}	可选
<b>id</b> integer	可选
▼ <b>userDetail</b> object {3}	可选
<b>id</b> integer	必需
<b>nickname</b> string	必需
<b>avatar</b> string	必需
<b>told</b> integer	可选
▼ <b>toUserDetail</b> object {3}	可选
<b>id</b> integer	必需
<b>nickname</b> string	必需
<b>avatar</b> string	必需
<b>content</b> string	可选
<b>timestamp</b> string	可选

## 返回示例:



## 示例

```
{
  "code": 200,
  "msg": "string",
  "data": {
    "total": 0,
    "records": [
      [
        {
          "id": 0,
          "userDetail": {
            "id": 0,
            "nickname": "string",
            "avatar": "string"
          },
          "toId": 0,
          "toUserDetail": {
            "id": 0,
            "nickname": "string",
            "avatar": "string"
          },
          "content": "string",
          "timestamp": "string"
        }
      ]
    ]
  }
}
```

## ◇ 添加评论

请求 URL:

```
/video/comment/add
```

请求方式:

```
POST
```

请求参数:

Body 参数	application/json	示例	
<code>videoid</code>	string	必需	{ "videoId": "string", "toId": "string", "content": "string" }
<code>toId</code>	string 目标评论id, 对视频写-1	必需	
<code>content</code>	string	必需	

返回参数:

## 数据结构

<b>code</b>	integer	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<b>msg</b>	string	错误信息	必需
<b>data</b>	string	数据	必需
默认值:	<input type="text" value="ok"/>		

## 返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## ✧ 删除评论

### 请求 URL:

```
/video/comment/delete
```

### 请求方式:

```
POST
```

### 请求参数:

Body 参数	application/json	示例		
<b>id</b>	string	ID 编号	必需	{ "id": "string" }

### 返回参数:

## 数据结构

<code>code</code>	<code>integer</code>	编码	必需
200成功, 0和其它数字为失败			
默认值:	<input type="text" value="200"/>		
<code>msg</code>	<code>string</code>	错误信息	必需
<code>data</code>	<code>string</code>	数据	必需
默认值:	<input type="text" value="ok"/>		

返回示例:

### 示例

```
{
  "code": 200,
  "msg": "string",
  "data": "ok"
}
```

## 四、系统安全和权限设计

### 4.1 系统安全

安全问题:

- ✧ XSS 攻击
- ✧ CSRF 攻击
- ✧ SQL 注入
- ✧ DDOS
- ✧ 数据脱敏

解决办法:

1. 依托 React 自带的转义以及后端对风险内容的过滤来预防 XSS 攻击
2. 将 JWT 储存至 localStorage 来防止一定程度上的 CSRF 攻击。
3. 使用 Mybatis-Plus 框架从而使用其附带的 JDBC preparedStatement 调用来预防 SQL 注入攻击

4. 依托阿里云基础云盾服务来一定程度上防御 DDOS 攻击
5. 开启 HTTPS 实现服务端与客户端的加密通信，一定程度上保证数据传输安全
6. 对用户密码字段进行加盐 Hash 处理
7. 对用户邮箱等敏感字段进行数据脱敏处理
8. 进行简单的 IP 限流防止用户频繁恶意地调用接口

### 详细描述:

为了确保用户的隐私安全，我们将在用户注册和登录过程中采用 RSA 加密技术对密码进行传输。这样做可以确保即使在数据传输过程中被截获，攻击者也无法轻易获取到用户的明文密码。在后端接收到用户密码后，我们会对用户的敏感信息进行脱敏处理，以进一步保护用户隐私。接下来，我们将对密码进行加盐哈希处理，然后将处理后的密码存储到数据库中。

为了保证弹幕的匿名性，同时满足屏蔽特定用户发送的弹幕功能需求，我们将弹幕发送者的 ID 进行 CRC 32 循环校验。校验后，我们会保留 ID 的 8 位校验码作为识别用户身份的凭据。这样做既可以确保用户的匿名性，又可以让其其他用户屏蔽特定用户发送的所有弹幕。这种设计在保护用户隐私的同时，也提高了弹幕系统的实用性和安全性。

为了确保用户上传的视频内容符合法律法规以及平台政策，我们会采用先进的 AI 技术对其进行初步检测。在检测过程中，我们会对视频的标题、描述、内容等方面进行全面分析，以判断其是否涉及敏感信息、非法内容或违规行为。一旦检测到潜在问题，我们会立即采取相应措施，如限制用户观看、删除相关内容等，以确保平台的正常运行和用户的安全。

同时，我们非常重视用户的参与和监督。为此，我们提供了一个便捷的举报功能，允许用户对疑似不合法合规的视频进行举报。用户在举报时，只需填写相关信息，如举报原因、举报证据等，然后提交给我们。我们会对用户的举报进行严格保密，确保举报人的安全和隐私。

对于用户的举报，我们的后台管理团队会进行二次审查。在审查过程中，我们的专业团队会根据举报信息和 AI 检测结果，对视频进行更深入的分析 and 判断。一旦发现确实存在违规行为，我们会立即采取相应措施，如封禁账号、删除视频、追究法律责任等，以维护平台的公平、公正和法治环境。

通过这种双层检测机制，我们能够更有效地确保用户上传的视频内容合法合规，为用户提供一个安全、健康的网络环境。同时，我们也鼓励用户积极参与举报，共同维护平台的良好秩序。

## 4.2 权限设计

使用简化版 RBAC 权限模型进行设计，将角色分为普通用户和管理员，以及未登录的访客。

- ✧ 访客：可以使用视频观看、搜索等基本功能，但如果需要收藏、上传视频等则需要登录。
- ✧ 普通用户：可以使用访客的所有功能，还可以使用收藏、上传视频等登录后才可以使用的功能。
- ✧ 管理员：可以使用后台相关接口，包括 举报审核、删除特定评论、删除弹幕、下架视频、上传番剧等。

# 五、验收标准

## 5.1 界面验收标准

### 5.1.1 注册登录和重置密码界面

#### 1. 注册界面：

- ✧ 所有输入字段包括邮箱、昵称、密码和验证码均能正确接收并显示。
- ✧ 密码输入框能够掩盖用户输入的密码。
- ✧ 点击注册按钮后，系统能够验证输入的信息格式是否符合要求，如密码长度、邮箱格式等，并给予相应的提示。
- ✧ 验证码输入框能够正确接收用户输入的验证码，并进行验证。

#### 2. 登录界面：

- ✧ 所有输入字段包括邮箱或用户 id、密码和验证码均能正确接收并显示。
- ✧ 密码输入框能够掩盖用户输入的密码。
- ✧ 能够验证邮箱是否为有效邮箱。
- ✧ 点击登录按钮后，系统能够验证输入的信息格式是否符合要求，并给予相应的提示。
- ✧ 验证码输入框能够正确接收用户输入的验证码，并进行验证。

### 3. 重置密码界面：

- ✧ 用户能够通过提供注册时使用的邮箱来进行密码重置。
- ✧ 提供验证码输入框，能够正确接收用户输入的验证码，并进行验证。
- ✧ 能够验证用户新密码输入的格式是否符合要求，并给予相应的提示。
- ✧ 确认密码输入框，保证两次输入的密码一致。

## 5.1.2 首页

- ✧ 界面简洁美观，符合大众用户操作习惯，易于上手。
- ✧ 推荐的视频列表能够正确显示，并且大致符合用户的兴趣偏好。
- ✧ 顶栏能够准确显示并提供相应的功能按钮，如搜索、个人主页、收藏、历史记录等。
- ✧ 热门视频推荐的轮播图能够流畅地切换，展示不同的推荐视频。
- ✧ 点击视频项能够正确跳转到对应的视频详情界面，并能够正常播放视频。

## 5.1.3 视频详情界面

- ✧ 视频能够正常加载并播放，播放器界面清晰简洁，用户能够方便地控制播放进度、音量等。
- ✧ 评论区能够正常显示用户的评论，且评论内容符合输入输出格式的要求，字数在 1 到 1000 之间，不能全为空格。
- ✧ 用户能够在评论区输入评论，并且提交后能够及时显示在页面上。
- ✧ 用户能够发送弹幕，并且提交后能够及时显示在页面上。
- ✧ 用户能够对视频进行点赞和收藏操作，点赞和收藏状态能够正确地显示在界面上，并且用户的操作能够及时反映在数据库中。

## 5.1.4 搜索界面

- ✧ 搜索界面能够正确显示搜索框和搜索按钮，并且布局合理，美观大方。
- ✧ 用户能够在搜索框中输入关键字进行搜索，搜索按钮能够正确响应用户点击。
- ✧ 搜索结果能够准确显示符合关键字的视频列表，每个视频项包含视频标题、封面图等信息。
- ✧ 用户能够点击搜索结果中的视频项，跳转到对应的视频详情界面。

### 5.1.5 个人主页界面

- ✧ 个人主页能够准确显示用户的个人信息，包括头像、昵称、个性签名等。
- ✧ 用户能够方便地查看自己的收藏和历史记录，每个视频项能够显示视频的基本信息，并且能够正确跳转到对应的视频详情界面。
- ✧ 用户能够跳转到视频上传界面。

### 5.1.6 视频上传界面

- ✧ 视频上传界面能够准确显示上传相关的表单和按钮，包括视频标题、描述、选择视频分类标签等。
- ✧ 用户能够正确填写上传信息，并且提交后能够给出相应的提示，告知用户上传是否成功。

### 5.1.7 反馈界面

- ✧ 反馈界面能够准确显示反馈类型选择框、反馈内容输入框和提交按钮，布局合理美观。
- ✧ 用户能够方便地选择反馈类型，并且能够正确输入反馈内容，字符数量在 1 到 3000 之间，不能全为空格。
- ✧ 用户提交反馈后能够及时收到反馈提交成功的提示信息。

### 5.1.8 举报审核界面

- ✧ 管理员能够登录后台系统，进入举报审核界面。
- ✧ 举报审核界面能够准确显示待审核的举报列表，包括举报内容、举报者信息等。
- ✧ 管理员能够点击举报项进入举报审核页面，能够查看举报内容并进行审核操作。
- ✧ 如果举报情况属实，管理员能够对相关内容进行处理，例如将视频下架、内容删除、封禁账号等。

### 5.1.9 反馈处理界面

- ◇ 管理员能够登录后台系统，进入反馈处理界面。
- ◇ 反馈处理界面能够准确显示待处理的反馈列表，包括反馈类型、反馈内容、反馈者信息等。
- ◇ 管理员能够点击反馈项进入反馈处理页面，能够查看反馈内容并进行处理操作。

### 5.1.10 视频优先级调整界面

- ◇ 管理员能够登录后台系统，进入视频优先级调整界面。
- ◇ 视频优先级调整界面能够准确显示待调整的视频列表，包括视频标题、当前优先级等信息。
- ◇ 管理员能够通过界面操作对特定视频在搜索、推荐的优先级做出调整，并且调整后的优先级能够反映在用户界面上。

## 5.2 功能验收标准

实现界面介绍的所有功能。