

视频可视化智能分析系统

V1.0

—设计说明书

第一章 前言

1.1 安装与初始化及软件主要参数设置

软件环境：

JDK1.8

Maven: 3.3.9

Mysql:5.7

软件编译：

```
cd /videoSource/
```

```
mvn clean install
```

软件部署：

```
./start.sh
```

1.2 软件故障排除

查看服务进程：

```
ps -ef | grep videoSource
```

查看进程是否存在：

如果不存在，进入/videoSource/

启动服务：./start.sh

查看日志：

```
cd /logs
```

```
tail -f logs.log
```

1.3 运行环境

硬件环境(主机):

主机配置: i7 9 代, 内存 8G, 硬盘 256G

软件环境(主机):

服务器环境: 操作系统: CentOS7.5

CPU: 4cores

内存: 32GB

DISK: 500GB

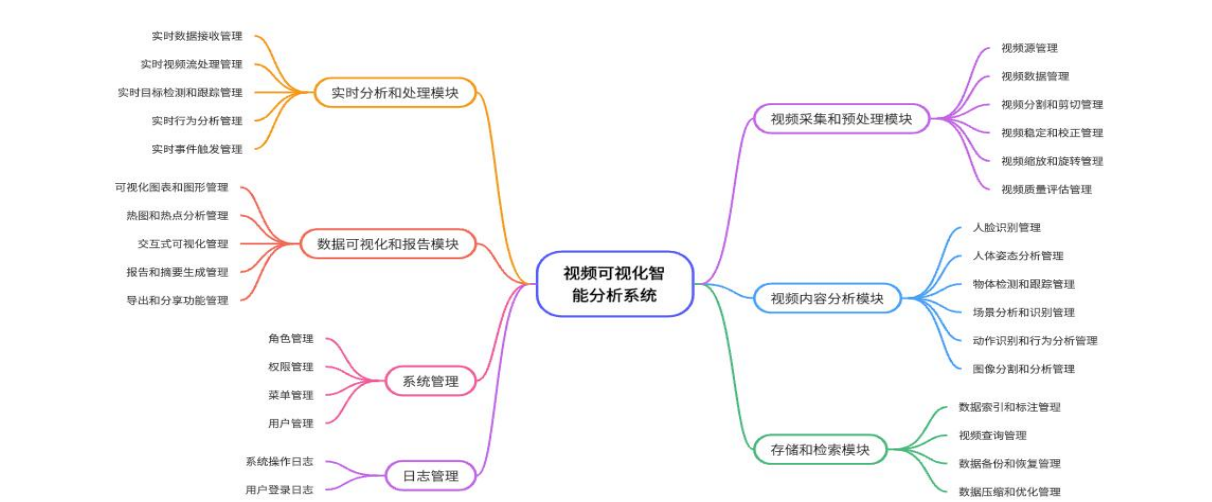
客户端: 终端

1.4 系统概述

视频可视化智能分析系统是一种利用人工智能和计算机视觉技术对视频进行实时分析和提取有用信息的系统。该系统可以识别视频中的对象、场景和动作, 并提供实时的统计分析和可视化展示。

第二章 总体设计

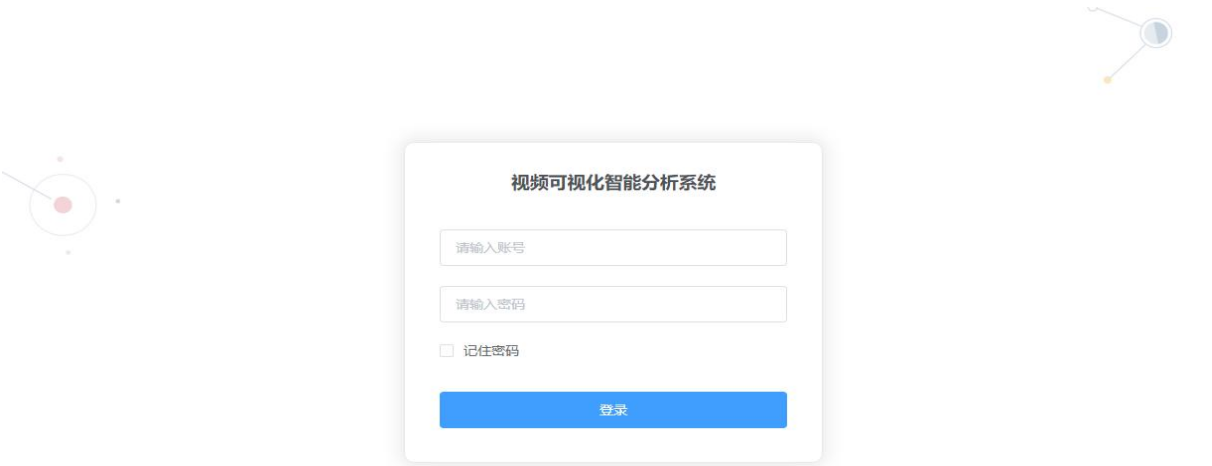
2.1 功能结构图



2.2 界面效果图

2.2.1 登录系统

在浏览器输入系统地址，点击【enter】键，进入系统登录页面，用户依次输入自己的账号、密码后，点击【登录】按钮，进入视频可视化智能分析系统。



2.2.2 视频源管理

点击【视频采集和预处理模块】菜单，再点击视频源管理，展示视频源管理列表，
展示信息：视频源编号、视频源名称、视频源类型、视频源状态、创建时间、创建人。

视频采集和预处理模块

视频源管理

视频数据管理

视频分割和剪切管理

视频融合和校正管理

视频缩放和旋转管理

视频质量评估管理

视频内容分析模块

存储和检索模块

实时分析和处理模块

数据可视化和报告模块

系统管理

日志管理

视频采集和预处理模块 > 视频源管理

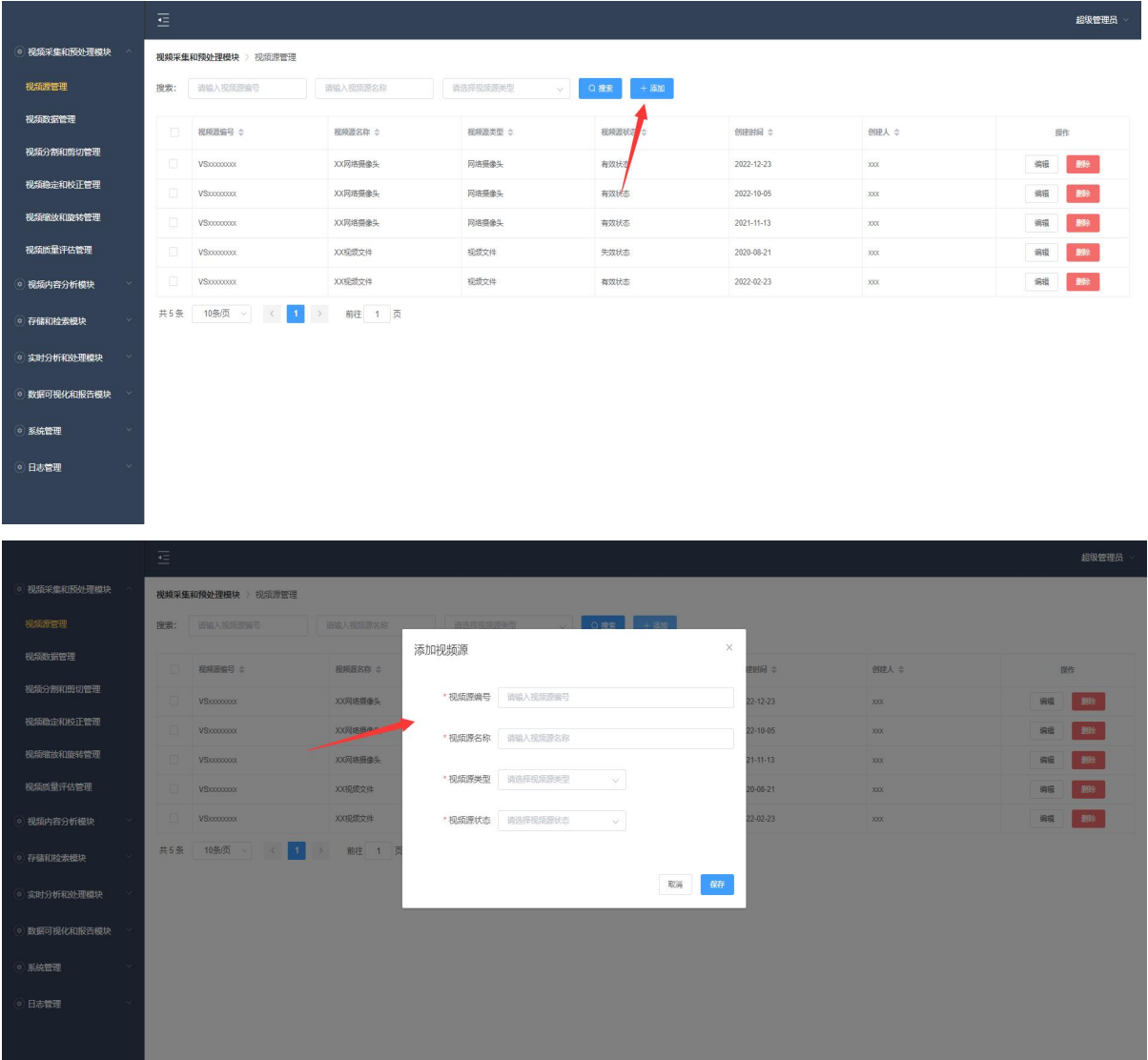
搜索:

| <input type="checkbox"/> | 视频源编号 | 视频源名称 | 视频源类型 | 视频源状态 | 创建时间 | 创建人 | 操作 |
|--------------------------|------------|---------|-------|-------|------------|-----|---|
| <input type="checkbox"/> | VS00000000 | XX网络摄像头 | 网络摄像头 | 有效状态 | 2022-12-23 | xxx | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | VS00000000 | XX网络摄像头 | 网络摄像头 | 有效状态 | 2022-10-05 | xxx | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | VS00000000 | XX网络摄像头 | 网络摄像头 | 有效状态 | 2021-11-13 | xxx | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | VS00000000 | XX视频文件 | 视频文件 | 失效状态 | 2020-08-21 | xxx | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | VS00000000 | XX视频文件 | 视频文件 | 有效状态 | 2022-02-23 | xxx | <input type="button" value="编辑"/> <input type="button" value="删除"/> |

共 5 条 前往 页

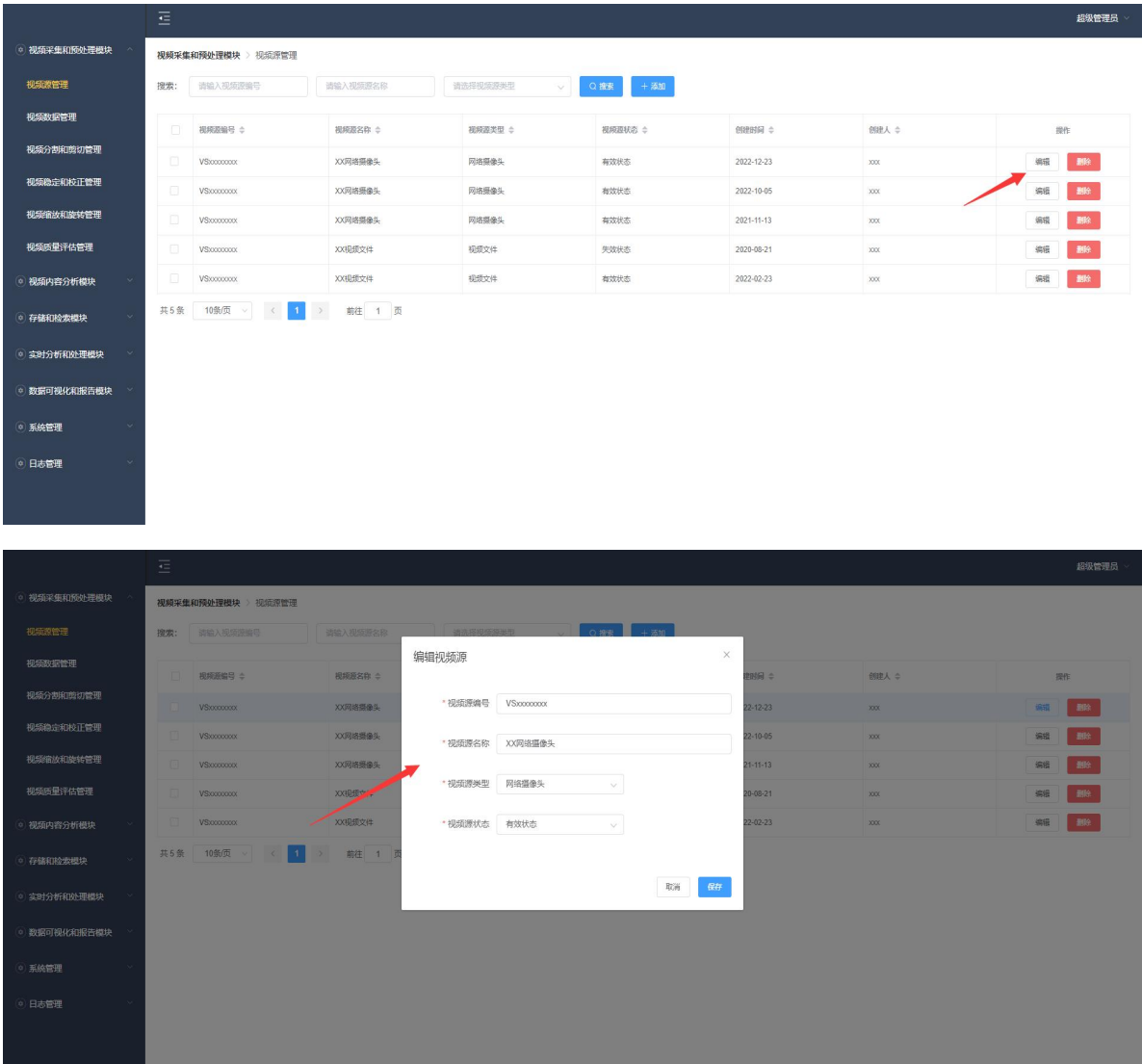
2.2.3 视频源添加

点击添加按钮，弹出视频源添加弹窗，填写信息添加即可。



2.3.4 视频源编辑

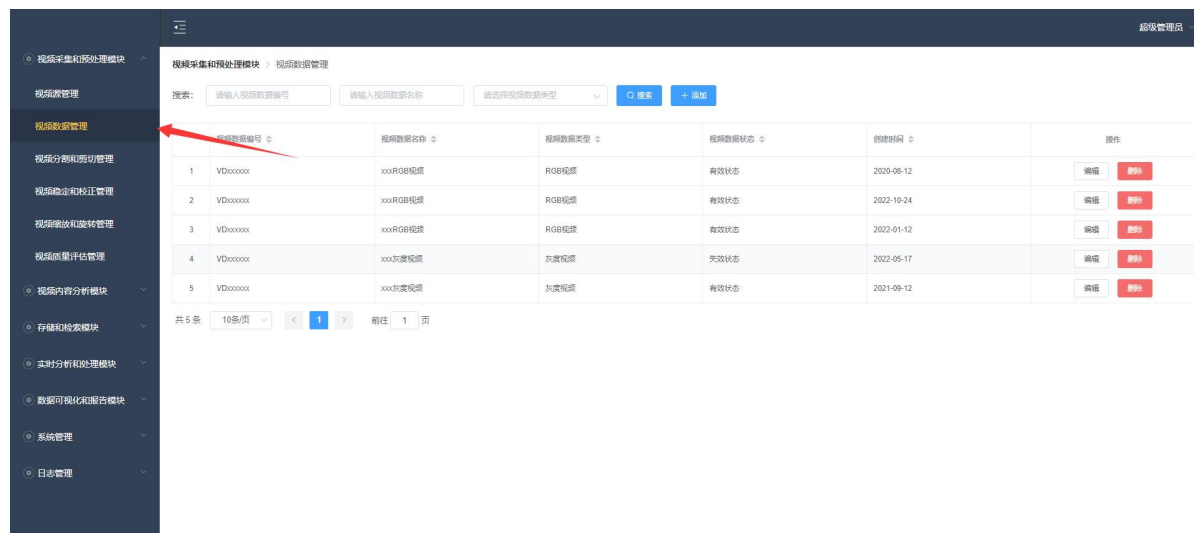
点击右侧编辑按钮，弹出视频源编辑弹窗，填写信息编辑即可。



2.2.5 视频数据管理

点击【视频采集和预处理模块】菜单，再点击视频数据管理进入列表页。展示信息：

视频数据编号、视频数据名称、视频数据类型、视频数据状态、创建时间。



视频采集和预处理模块 > 视频数据管理

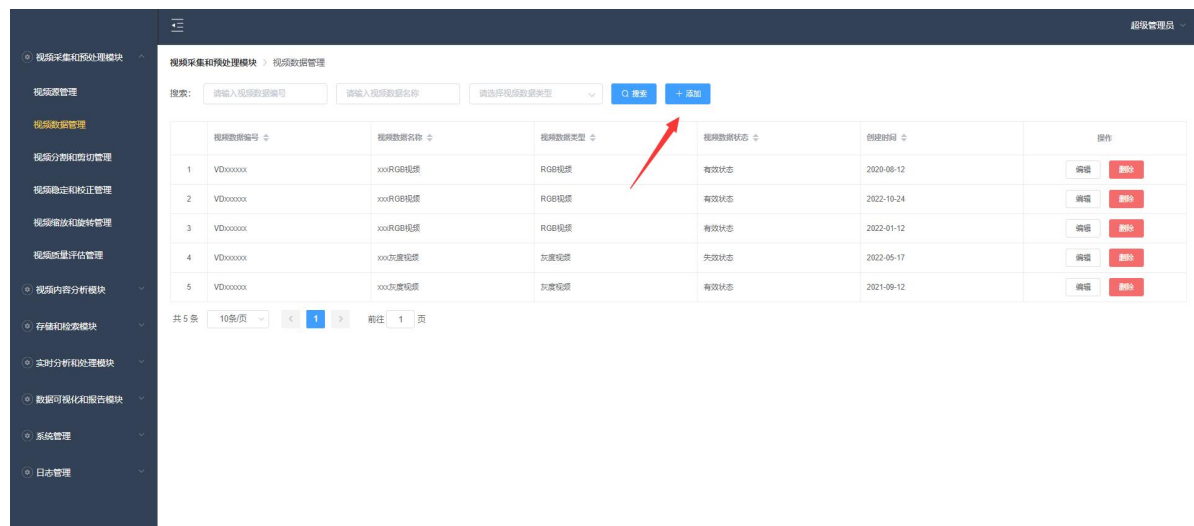
搜索: 请输入视频数据编号 请输入视频数据名称 请选择视频数据类型

| | 视频数据编号 | 视频数据名称 | 视频数据类型 | 视频数据状态 | 创建时间 | 操作 |
|---|----------|----------|--------|--------|------------|---|
| 1 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2020-08-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 2 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2022-10-24 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 3 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2022-01-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 4 | VD000000 | xxx灰度视频 | 灰度视频 | 失效状态 | 2022-05-17 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 5 | VD000000 | xxx灰度视频 | 灰度视频 | 有效状态 | 2021-09-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |

共 5 条 < 1 > 前往 1 页

2.2.6 视频数据添加

点击添加按钮，弹出视频数据添加弹窗，填写信息添加即可。

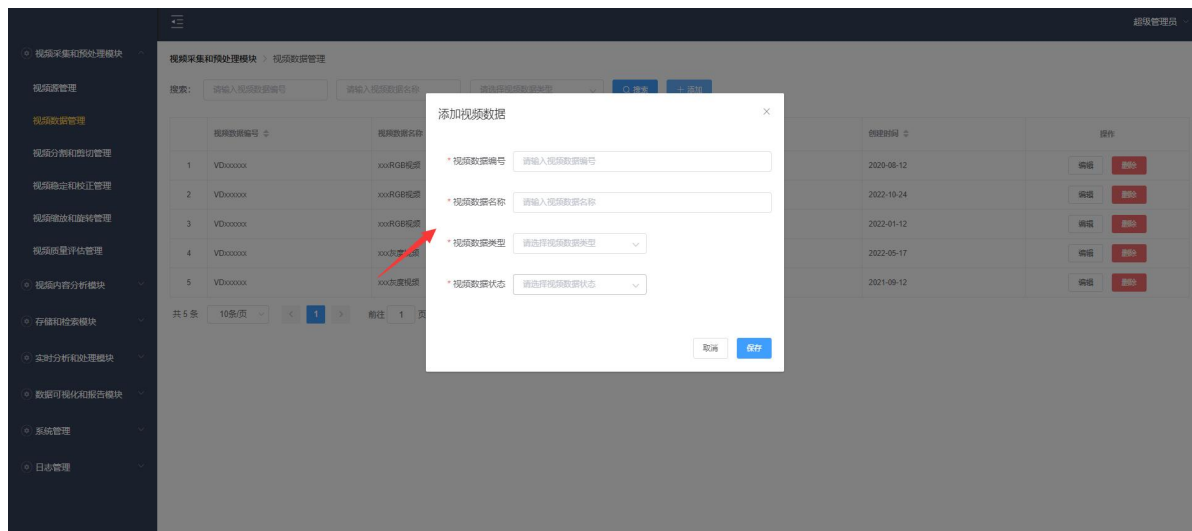


视频采集和预处理模块 > 视频数据管理

搜索: 请输入视频数据编号 请输入视频数据名称 请选择视频数据类型

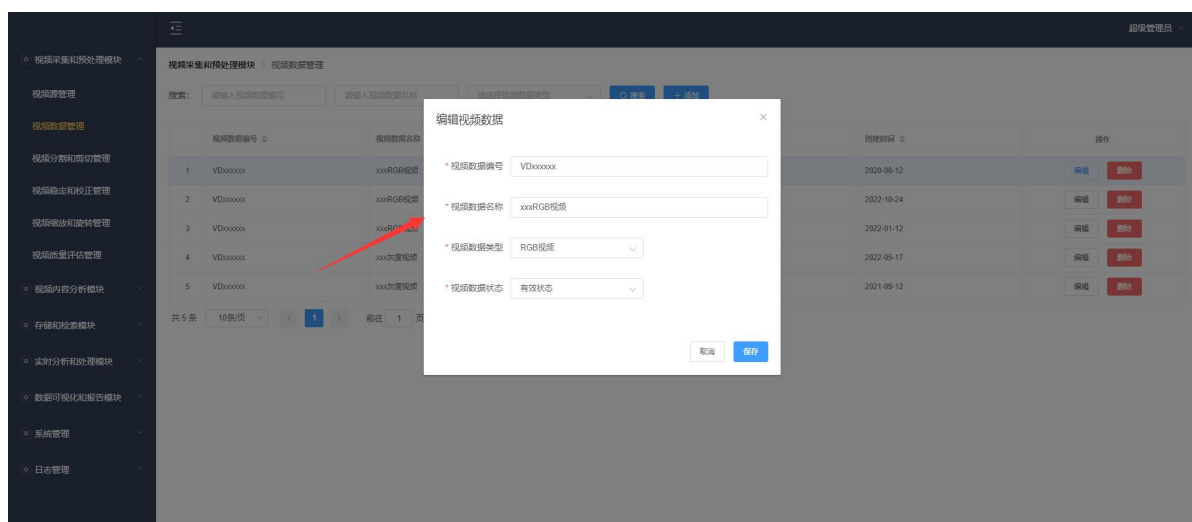
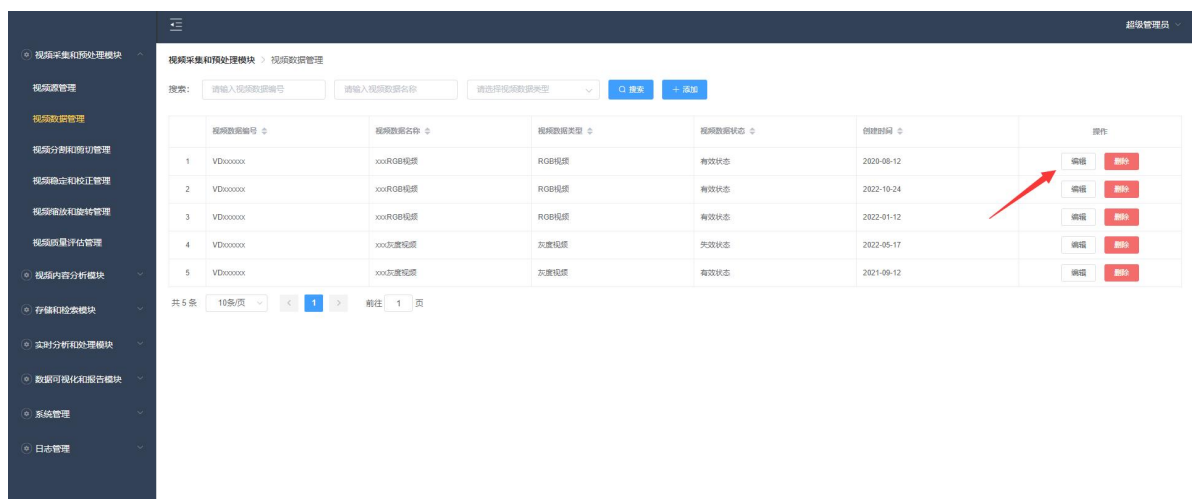
| | 视频数据编号 | 视频数据名称 | 视频数据类型 | 视频数据状态 | 创建时间 | 操作 |
|---|----------|----------|--------|--------|------------|---|
| 1 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2020-08-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 2 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2022-10-24 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 3 | VD000000 | xxxRGB视频 | RGB视频 | 有效状态 | 2022-01-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 4 | VD000000 | xxx灰度视频 | 灰度视频 | 失效状态 | 2022-05-17 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |
| 5 | VD000000 | xxx灰度视频 | 灰度视频 | 有效状态 | 2021-09-12 | <input type="button" value="编辑"/> <input type="button" value="删除"/> |

共 5 条 < 1 > 前往 1 页



2.2.7 视频数据编辑

点击右侧编辑按钮，弹出视频数据编辑弹窗，填写信息编辑即可。



第三章 接口设计

3.1 视频源和视频数据管理

3.1.1 视频源管理接口

请求路径

/farm/videoSourceList

请求方式 POST

json 参数

```
{
  "pageNo": 1,
  "pageSize": 10,
  "videoSourceNo": "xxxx",
  "videoSourceName": "xxxx",
  "videoSourceType": "xxxx",
  "createTime": "xxxx",
  "creator": "xxxx",
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|-------------------|--------|----|-------|
| pageNo | String | 是 | 分页号 |
| pageSize | String | 是 | 分页数量 |
| videoSourceNo | String | 是 | 视频源编号 |
| videoSourceName | String | 是 | 视频源名称 |
| videoSourceStatus | String | 是 | 视频源状态 |
| createTime | Date | 是 | 创建时间 |

| | | | |
|---------|--------|---|-----|
| creator | String | 是 | 创建人 |
|---------|--------|---|-----|

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": "",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.1.2 视频源创建接口

请求路径

/farm/addVideoSource

请求方式 POST

json 参数

```
{
  "videoSourceNo": "***",
  "videoSourceName": "***",
  "videoSourceType": "***",
  "videoSourceStatus": "***",
}
```

```
        "createTime": "***",
        "creator": "***",
    }
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|-------------------|--------|----|-------|
| videoSourceNo | String | 是 | 视频源编号 |
| videoSourceName | String | 是 | 视频源名称 |
| videoSourceType | String | 是 | 视频源类型 |
| videoSourceStatus | String | 是 | 视频源状态 |
| createTime | Date | 是 | 创建时间 |
| creator | String | 是 | 创建人信息 |

返回示例

```
{
    "code": 0,
    "data": {},
    "flag": true,
    "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3. 1. 3 视频数据管理接口

请求路径

/farm/videoDataList

请求方式 POST

json 参数

```
{
  "pageNo": 1,
  "pageSize": 10,
  "videoDataNo": xxx,
  "videoDataName": "xxx",
  "videoDataType": "xxxx",
  "createTime": "xxxx",
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|---------------|--------|----|--------|
| pageNo | String | 是 | 分页号 |
| pageSize | String | 是 | 分页数量 |
| videoDataNo | String | 是 | 视频数据编号 |
| videoDataName | String | 是 | 视频数据名称 |
| videoDataType | String | 是 | 视频数据类型 |
| createTime | Date | 是 | 创建时间 |

返回示例

```
{
  "code": 0,
  "data": {},
}
```

```
    "flag": true,
    "msg": " ",
  }
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3. 1. 4 视频数据创建接口

请求路径

/farm/addVideoData

请求方式 POST

json 参数

```
{
  "videoDataNo": "***",
  "videoDataName": "***",
  "videoDataType": "***",
  "videoDataStatus": "***",
  "createTime": "***",
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|-----|----|----|----|
|-----|----|----|----|

| | | | |
|-----------------|--------|---|--------|
| videoDataNo | String | 是 | 视频数据编号 |
| videoDataName | String | 是 | 视频数据名称 |
| videoDataType | String | 是 | 视频数据类型 |
| videoDataStatus | String | 是 | 视频数据状态 |
| createTime | String | 是 | 创建时间 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": "",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3. 2 系统用户管理

3. 2. 1 登录接口

请求路径

/farm/user/login

请求方式 POST

json 参数

```
{
  "username": 1,
  "password": "123456",
  "address": "xxxx"
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|----------|--------|----|------|
| password | String | 是 | 密码 |
| username | String | 是 | 用户名 |
| address | String | 是 | 用户地址 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.2.2 注销接口

请求路径

/farm/user/layout

请求方式 POST

json 参数

```
{
  "username": 1,
  "password": "123456"
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|----------|--------|----|-----|
| password | String | 是 | 密码 |
| username | String | 是 | 用户名 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|------|------|
|------|------|------|

| | | |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.2.3 注册接口

请求路径

/farm/user/register

请求方式 POST

json 参数

```
{
  "username": 1,
  "password": "123456",
  "mobile": "xxxx",
  "address": "xxxx",
  "Role": "xxxx",
  "Picture": "xxxx",
  "File": "xxxx",
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|----------|--------|----|-----|
| password | String | 是 | 密码 |
| username | String | 是 | 用户名 |
| mobile | String | 是 | 手机号 |

| | | | |
|---------|--------|---|----|
| Role | String | 是 | 角色 |
| Picture | String | 是 | 照片 |
| address | String | 是 | 区域 |
| File | String | 是 | 附件 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": "",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3. 2. 4 信息修改接口

请求路径

/farm/user/changeUserInfo

请求方式 POST

json 参数

```
{
```

```
    "username": 1,
    "password": "123456",
    "mobile": "xxxx",
    "address": "xxxx",
    "Role": "xxxx",
    "Picture": "xxxx",
    "File": "xxxx",
  }
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|------------|--------|----|-----|
| password | String | 是 | 密码 |
| username | String | 是 | 用户名 |
| Sex | String | 否 | 性别 |
| Role | String | 否 | 角色 |
| Picture | String | 否 | 照片 |
| password | String | 否 | 密码 |
| address | String | 否 | 区域 |
| File | String | 否 | 附件 |
| Department | String | 否 | 部门 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.3 权限管理

3.3.1 权限新增接口

请求路径

/farm/security/addPermission

请求方式 POST

json 参数

```
{
  "username": 1,
  "Wechatnumber": "123456",
  "Sex": "xxxx",
  "Role": "xxxx"
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|--------------|--------|----|-----|
| username | String | 是 | 用户名 |
| Wechatnumber | String | 是 | 微信号 |

| | | | |
|------|--------|---|----|
| Sex | String | 是 | 性别 |
| Role | String | 是 | 角色 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.3.2 权限删除接口

请求路径

/farm/security/deletePermission

请求方式 POST

json 参数

```
{
  "username": 1,
  "Wechatnumber": "123456",
}
```

```
    "Sex": "xxxx",
    "Role": "xxxx"
  }
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|--------------|--------|----|-----|
| username | String | 是 | 用户名 |
| Wechatnumber | String | 是 | 微信号 |
| Sex | String | 是 | 性别 |
| Role | String | 是 | 角色 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": "",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.3.3 权限列表查询接口

请求路径

/farm/security/queryPermissionList

请求方式 POST

json 参数

```
{
  "username": 1,
  "Wechatnumber": "123456",
  "Sex": "xxxx",
  "Role": "xxxx",
  "isActivate": "xxxx"
  "Permissionlist": "xxxx"
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|----------------|--------|----|------|
| username | String | 是 | 用户名 |
| Wechatnumber | String | 是 | 微信号 |
| Sex | String | 是 | 性别 |
| Role | String | 是 | 角色 |
| isActivate | String | 是 | 是否激活 |
| Permissionlist | String | 是 | 权限列表 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
}
```



```
      "msg": "",
    }
  }
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

3.3.4 管理员权限赋予接口

请求路径

/farm/security/adminPermission

请求方式 POST

json 参数

```
{
  "username": 1,
  "Wechatnumber": "123456",
  "Sex": "xxxx",
  "Role": "xxxx"
}
```

json 参数字段说明

| 参数名 | 类型 | 必选 | 说明 |
|----------|--------|----|-----|
| username | String | 是 | 用户名 |

| | | | |
|--------------|--------|---|-----|
| Wechatnumber | String | 是 | 微信号 |
| Sex | String | 是 | 性别 |
| Role | String | 是 | 角色 |

返回示例

```
{
  "code": 0,
  "data": {},
  "flag": true,
  "msg": " ",
}
```

返回参数说明

| 字段名称 | 字段类型 | 注释说明 |
|------|---------|------|
| flag | boolean | 异常标识 |
| code | int | 错误码 |
| msg | string | 错误信息 |
| data | Object | 返回数据 |

第四章 表设计

4.1 视频源信息表

简要描述:

- 视频源信息

| 字段 | 类型 | 空 | 默认 | 注释 |
|-------------------|--------|------|------|-------|
| Id | String | true | Null | 主键 id |
| videoSourceNo | String | true | Null | 视频源编号 |
| videoSourceName | String | true | Null | 视频源名称 |
| videoSourceType | String | true | Null | 视频源类型 |
| videoSourceStatus | String | true | Null | 视频源状态 |
| createTime | String | true | Null | 创建时间 |
| creator | String | true | Null | 创建人 |

- 备注: 无

4.2 视频数据信息表

简要描述:

- 视频数据

| 字段 | 类型 | 空 | 默认 | 注释 |
|----|--------|-------|------|-------|
| Id | String | flase | Null | 主键 id |

| | | | | |
|-----------------|--------|-------|------|--------|
| videoDataNo | String | true | Null | 视频数据编号 |
| videoDataName | String | true | Null | 视频数据姓名 |
| videoDataType | String | flase | Null | 视频数据类型 |
| videoDataStatus | String | true | Null | 视频数据状态 |
| createTime | String | flase | Null | 创建时间 |

备注：无

4.3 系统用户表

简要描述：

- 描述储存系统员工信息

| 字段 | 类型 | 空 | 默认 | 注释 |
|------------|--------|-------|------|------|
| password | String | flase | Null | 密码 |
| username | String | flase | Null | 用户名 |
| mobile | String | flase | Null | 手机号 |
| gender | String | true | Null | 性别 |
| role | String | true | Null | 角色 |
| picture | String | true | Null | 照片 |
| isActivate | String | true | Null | 是否激活 |
| file | String | flase | Null | 附件 |

备注：无

第五章 非功能性设计

5.1 性能设计

数据库采用主从方式；

合理的应用程序架构：MVC

数据缓存：redis

搭建两套应用服务器，应用 SL[改成 SLB]B 负载均衡，提高系统性能，应对多用户、高并发业务场景；

平均响应时间 小于 100ms；最大响应时间不超过 500ms

资源利用率：cpu 平均利用率 30%，最高不超过 80%； 内存利用率平均 45%，最高不超过 80%；

请求并发量 ≈ 90

5.2 安全性设计

定义基于角色的权限管理机制，不同角色视频数据访问权限内数据

服务器只开放需要的端口

数据库操作使用预编译预计，防止 SQL 注入[增加 XSS 注入、CSRF 注入]、

非登录用户不允许进入系统

接口添加签名校验

5.3 其他非功能性设计

易用性、界面友好性：

用户交互界面绘图考虑不同层面用户对信息使用的不同需求，要求页面美观大方，操作简便，界面术语名称一致。

兼容性

操作系统：系统服务端及选用的数据库、中间件，应兼容 Linux 不同发行版或 Windows 不同版本；

产品版本：系统升级可兼容以前版本所有数据，不会造成数据丢失；

浏览器：应兼容主流浏览器&版本，系统监测报告中应明确体现浏览器兼容列表。

可移植性

提高代码可移植性，降低代码重复开发工作量。