

# 微电网谐波智能管理系统

## 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 /harmonic/
```

```
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 harmonic
```

查看进程是否存在：

如果不存在，进入/harmonic/

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

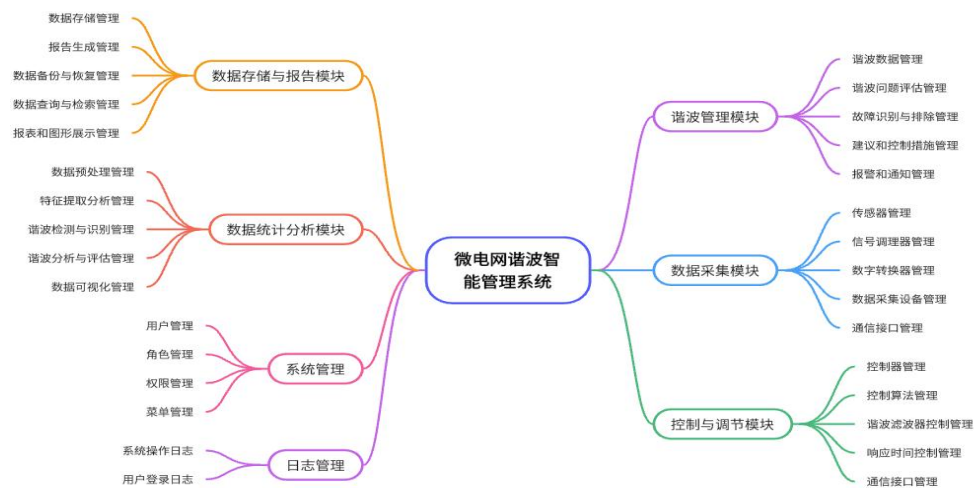
客户端: 终端

## 1.4 系统概述

微电网谐波智能管理系统是一种用于管理微电网中谐波问题的智能系统。它可以实时监测电压和电流的波形, 分析谐波成分并识别问题。一旦发现谐波问题, 系统可以自动采取措施来控制 and 调整微电网系统, 以减少谐波的影响。通过微电网谐波智能管理系统, 用户可以更好地监测和管理谐波问题, 保证微电网系统的稳定运行, 延长设备寿命, 并提高能源利用效率。

## 第二章 总体设计

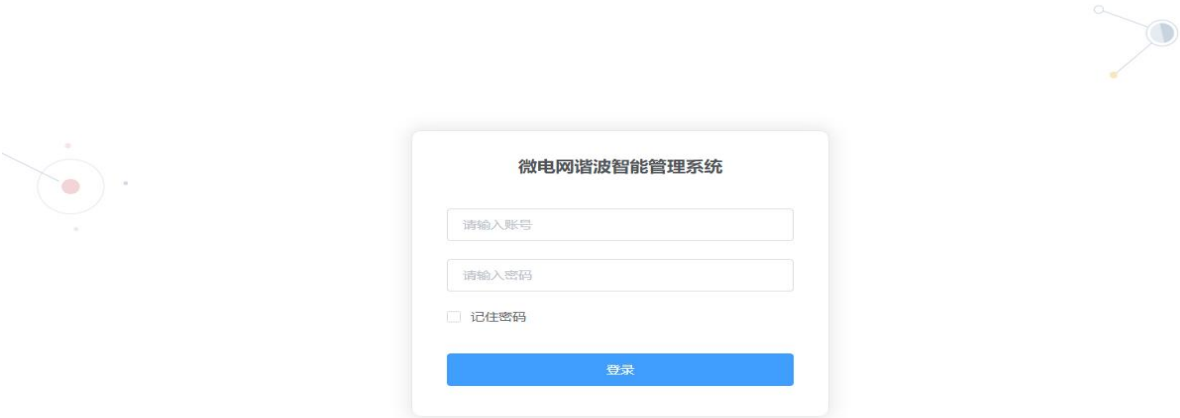
### 2.1 功能结构图



### 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，进入微电网谐波智能管理系统。



2.2.2 谐波数据管理

点击【谐波管理模块】菜单，再点击谐波数据管理，展示谐波数据管理列表，展示信息：谐波数据编号、谐波数据名称、谐波数据类型、谐波数据状态、创建时间、创建人。

谐波管理模块

谐波数据管理

谐波问题评估管理

故障寻踪与排除管理

建议和防控措施管理

报警和通知管理

控制与调节模块

数据采集模块

数据存储与报表模块

数据统计分析模块

系统管理

日志管理

谐波管理模块 > 谐波数据管理

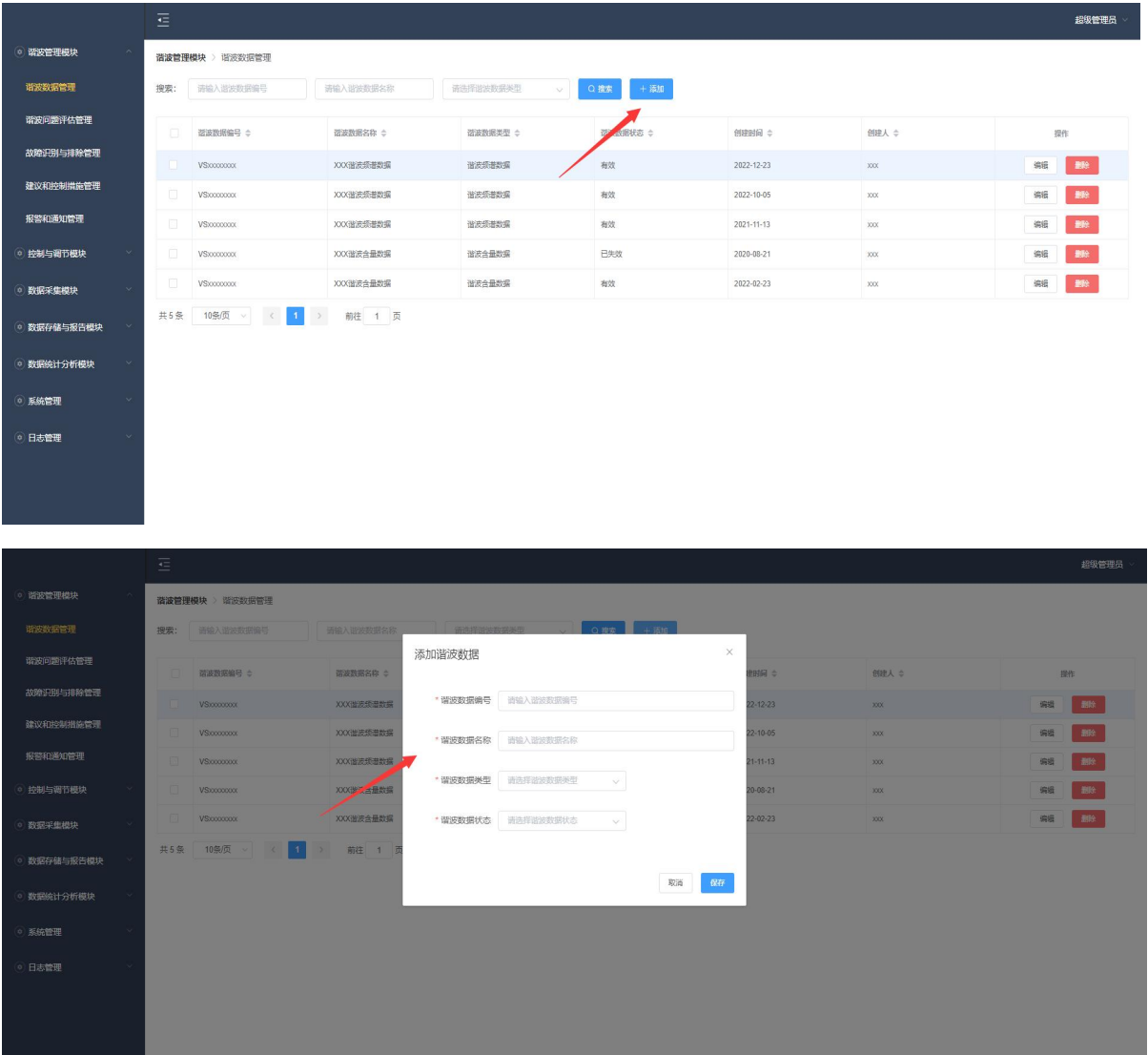
搜索:

<input type="checkbox"/>	谐波数据编号	谐波数据名称	谐波数据类型	谐波数据状态	创建时间	创建人	操作
<input type="checkbox"/>	VS00000001	XXX谐波电压数据	谐波电压数据	有效	2022-12-23	xxx	<input type="button" value="编辑"/> <input type="button" value="删除"/>
<input type="checkbox"/>	VS00000002	XXX谐波频率数据	谐波频率数据	有效	2022-10-05	xxx	<input type="button" value="编辑"/> <input type="button" value="删除"/>
<input type="checkbox"/>	VS00000003	XXX谐波电流数据	谐波电流数据	有效	2021-11-13	xxx	<input type="button" value="编辑"/> <input type="button" value="删除"/>
<input type="checkbox"/>	VS00000004	XXX谐波含量数据	谐波含量数据	已失效	2020-08-21	xxx	<input type="button" value="编辑"/> <input type="button" value="删除"/>
<input type="checkbox"/>	VS00000005	XXX谐波含量数据	谐波含量数据	有效	2022-02-23	xxx	<input type="button" value="编辑"/> <input type="button" value="删除"/>

共 5 条   前往  页

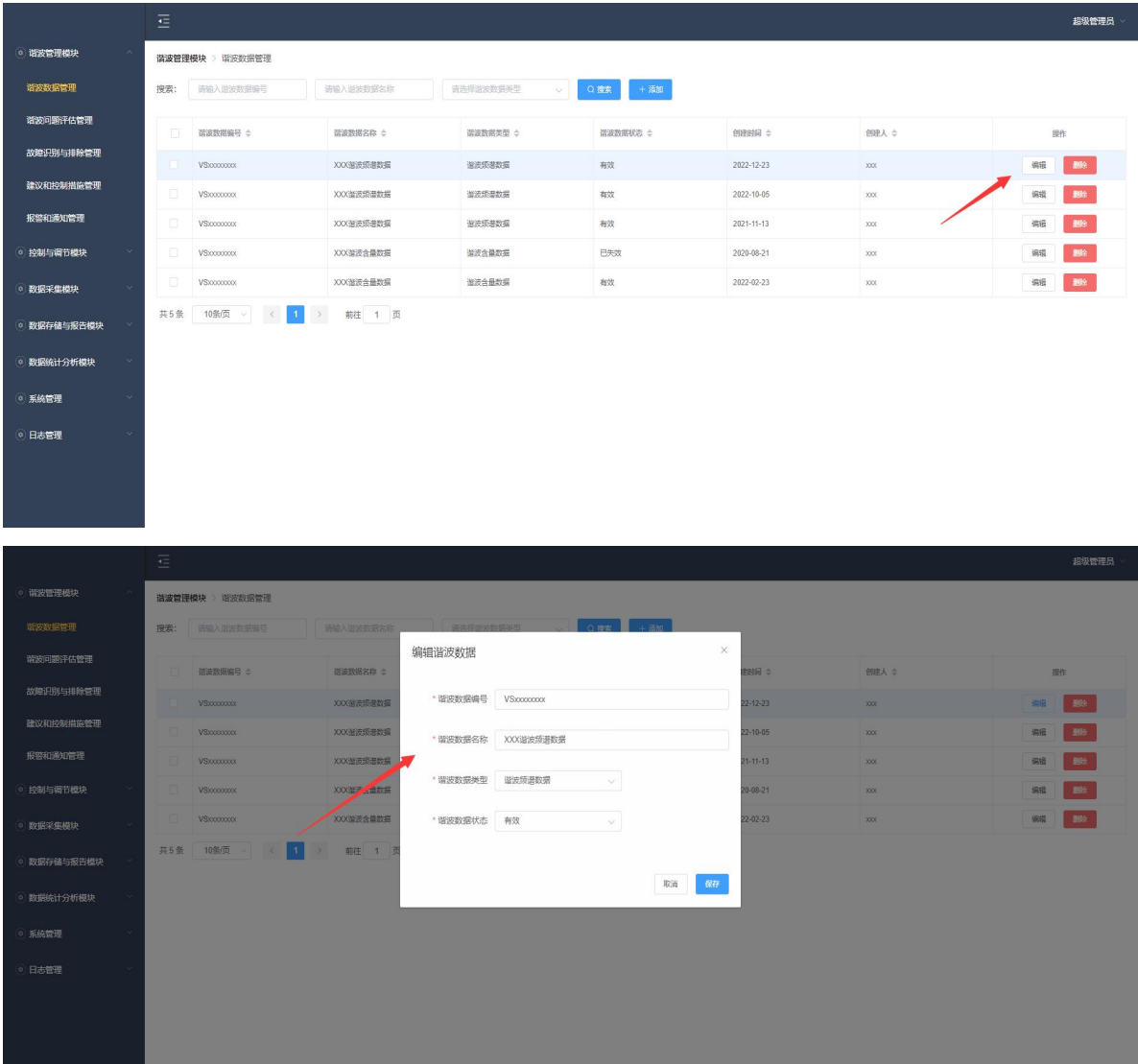
2.2.3 谐波数据添加

点击添加按钮，弹出谐波数据添加弹窗，填写信息添加即可。



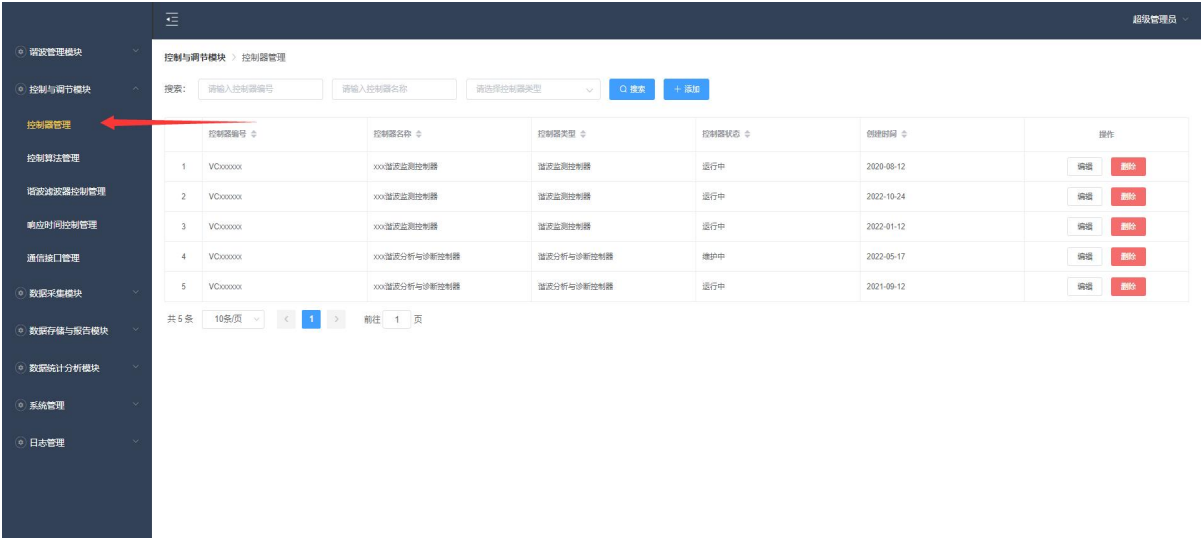
2.3.4 谐波数据编辑

点击右侧编辑按钮，弹出谐波数据编辑弹窗，填写信息编辑即可。



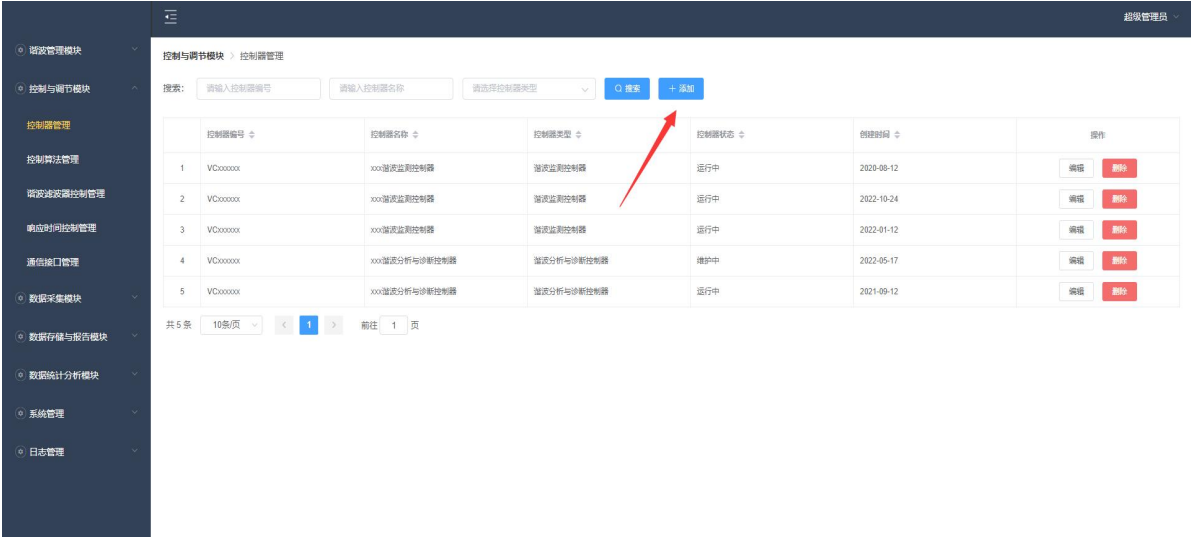
2.2.5 控制器管理

点击【控制与调节模块】菜单，再点击控制器管理进入列表页。展示信息：控制器编号、控制器名称、控制器类型、控制器状态、创建时间。

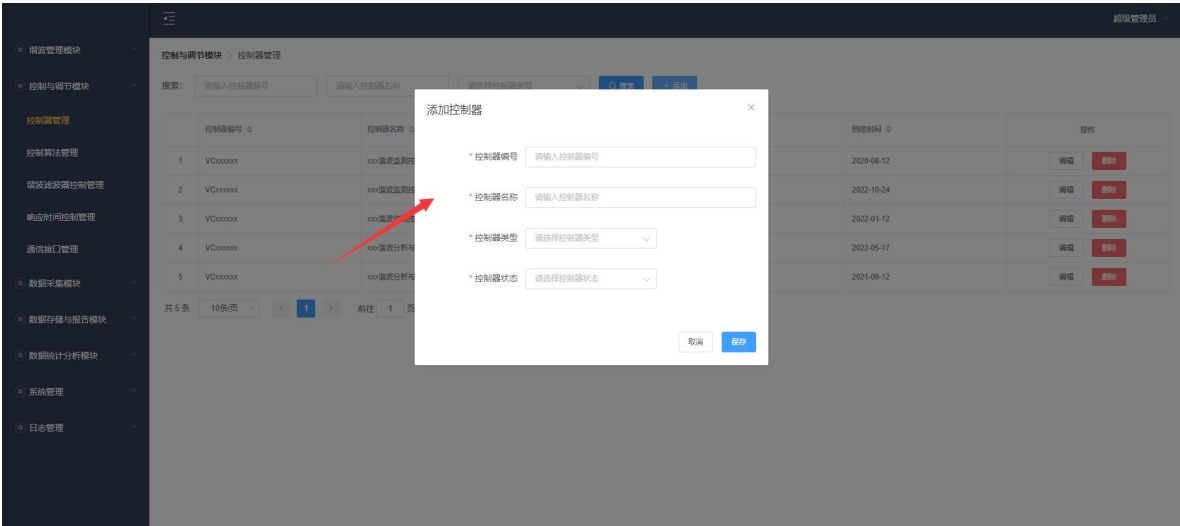


2.2.6 控制器添加

点击添加按钮，弹出控制器添加弹窗，填写信息添加即可。

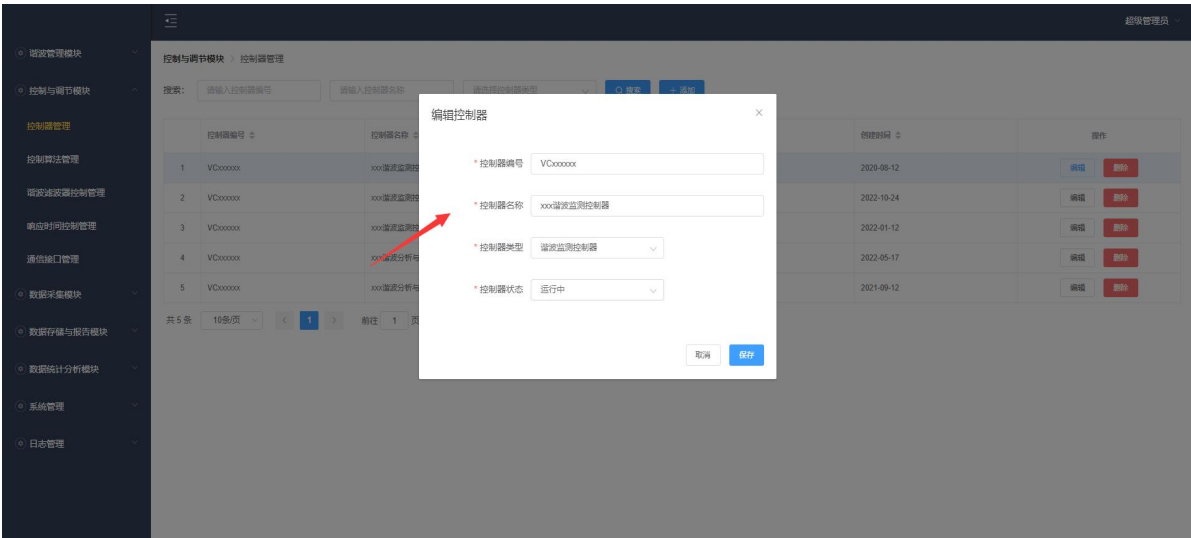
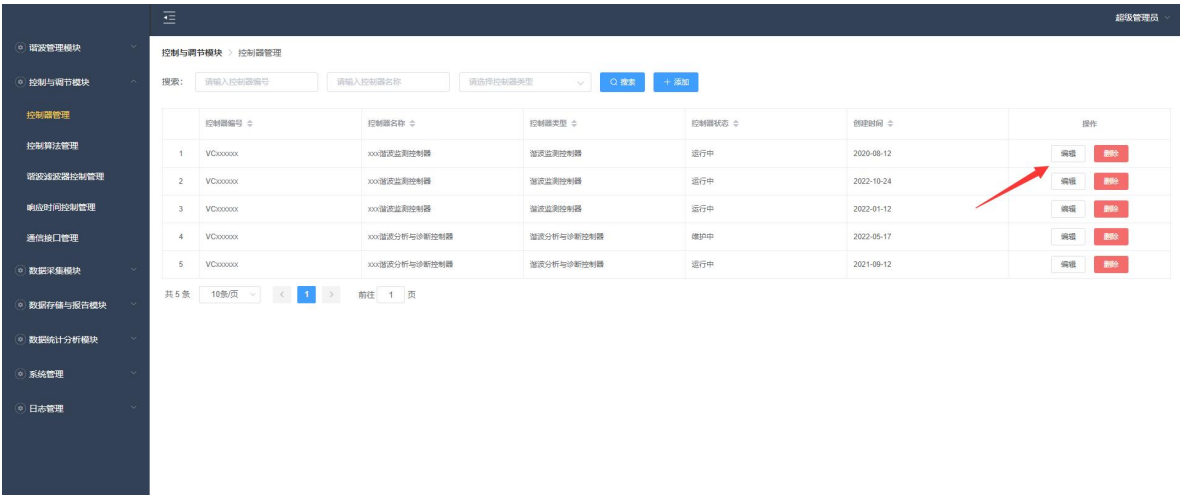






2.2.7 控制器编辑

点击右侧编辑按钮，弹出控制器编辑弹窗，填写信息编辑即可。



### 第三章 接口设计

#### 3.1 谐波数据和控制器管理

##### 3.1.1 谐波数据管理接口

请求路径

/common/harmonicList

请求方式 POST

json 参数

```
{
  "pageNo": 1,
  "pageSize": 10,
  "harmonicNo": "xxxx",
  "harmonicName": "xxxx",
  "harmonicType": "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
harmonicNo	String	是	谐波数据编号
harmonicName	String	是	谐波数据名称
harmonicType	String	是	谐波数据类型
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 谐波数据创建接口

请求路径

/common/addHarmonic

请求方式 POST

json 参数

```
{
  "harmonicNo": "***",
  "harmonicName": "***",
  "harmonicType": "***",
  "harmonicStatus": "***",
}
```

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

json 参数字段说明

参数名	类型	必选	说明
harmonicNo	String	是	谐波数据编号
harmonicName	String	是	谐波数据名称
harmonicType	String	是	谐波数据类型
harmonicStatus	String	是	谐波数据状态
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 控制器管理接口

请求路径

/common/controllerList

请求方式 POST

json 参数

```
{
  "pageNo": 1,
  "pageSize": 10,
  "controllerNo": xxx,
  "controllerName": "xxx",
  "controllerType": "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
controllerNo	String	是	控制器编号
controllerName	String	是	控制器名称
controllerType	String	是	控制器类型
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 控制器创建接口

请求路径

/common/addController

请求方式 POST

json 参数

```
{
  "controllerNo": "***",
  "controllerName": "***",
  "controllerType": "***",
  "controllerStatus": "***",
  "createTime": "***",
}
```

json 参数字段说明

参数名	类型	必选	说明
-----	----	----	----

controllerNo	String	是	控制器编号
controllerName	String	是	控制器名称
controllerType	String	是	控制器类型
controllerStatus	String	是	控制器状态
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

/common/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

/common/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

/common/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

/common/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

/common/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

/common/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

/common/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

/common/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  谐波数据表

简要描述：

- 谐波数据信息

字段	类型	空	默认	注释
Id	String	true	Null	主键 id
harmonicNo	String	true	Null	谐波数据编号
harmonicName	String	true	Null	谐波数据名称
harmonicType	String	true	Null	谐波数据类型
harmonicStatus	String	true	Null	谐波数据状态
createTime	String	true	Null	创建时间
creator	String	true	Null	创建人

- 备注：无

### 4.2  控制器信息表

简要描述：

- 控制器

字段	类型	空	默认	注释
Id	String	false	Null	主键 id

controllerNo	String	true	Null	控制器编号
controllerName	String	true	Null	控制器名称
controllerType	String	false	Null	控制器类型
controllerStatus	String	true	Null	控制器状态
createTime	String	false	Null	创建时间

备注：无

4.3 系统用户表

简要描述：

- 描述储存系统用户信息

字段	类型	空	默认	注释
password	String	false	Null	密码
username	String	false	Null	用户名
mobile	String	false	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	false	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 $\approx 90$

### 5.2 安全性设计

定义基于角色的权限管理机制，不同角色控制器访问权限内数据

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。