**图像去雾系统**

**说明文档**

版本号：V1.0.0

编制人：武沛鑫

编制时间：2023年11月25日星期一

**文档修订版本**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 修订说明 | 修订时间 | 文档版本号 | 编制人 |
| 1 | 初次编制 | 2023-11-25 | V1.0.0 | 武沛鑫 |
|  |  |  |  |  |

**目录**

[系统需求分析 11](#_Toc19305)

[用户需求 11](#_Toc8712)

[场景需求 11](#_Toc11283)

[系统需求 12](#_Toc3757)

[非功能需求 12](#_Toc9573)

[性能需求 12](#_Toc9112)

[系统设计 13](#_Toc10269)

[系统架构设计 13](#_Toc2728)

[业务架构 13](#_Toc2664)

[代码架构 14](#_Toc28147)

[部署架构 15](#_Toc20512)

[技术栈 15](#_Toc9120)

[前端 15](#_Toc6583)

[JAVA后端 15](#_Toc2458)

[Python后端 16](#_Toc6060)

[功能模块设计 17](#_Toc6071)

[去雾模块 17](#_Toc3714)

[图片上传 17](#_Toc9294)

[图片下载 18](#_Toc6030)

[指标评价 18](#_Toc30294)

[去雾处理 18](#_Toc27927)

[历史记录 18](#_Toc22902)

[用户模块 18](#_Toc17569)

[操作记录 18](#_Toc19385)

[菜单模块 18](#_Toc4182)

[权限模块 18](#_Toc22978)

[角色模块 18](#_Toc2279)

[数据分析模块 18](#_Toc9646)

[API接口设计 18](#_Toc11109)

[认证接口 18](#_Toc13838)

[POST 登录 18](#_Toc29400)

[请求参数 18](#_Toc21913)

[GET 获取验证码 19](#_Toc5029)

[DELETE 注销 19](#_Toc20873)

[用户接口 19](#_Toc28410)

[PUT 修改用户 19](#_Toc5052)

[请求参数 19](#_Toc26340)

[POST 新增用户 20](#_Toc16150)

[请求参数 20](#_Toc16306)

[POST 导入用户 20](#_Toc26549)

[请求参数 20](#_Toc25044)

[PATCH 修改用户状态 21](#_Toc3570)

[请求参数 21](#_Toc10074)

[PATCH 修改用户密码 21](#_Toc31304)

[请求参数 21](#_Toc31419)

[GET 用户表单数据 21](#_Toc18839)

[请求参数 21](#_Toc30867)

[GET 用户导入模板下载 22](#_Toc13407)

[GET 用户分页列表 22](#_Toc5229)

[请求参数 22](#_Toc12513)

[GET 获取当前登录用户信息 23](#_Toc221)

[GET 导出用户 23](#_Toc20303)

[请求参数 23](#_Toc23928)

[DELETE 删除用户 24](#_Toc26736)

[请求参数 24](#_Toc357)

[角色接口 24](#_Toc14941)

[PUT 修改角色状态 24](#_Toc12190)

[请求参数 24](#_Toc11341)

[PUT 分配菜单权限给角色 24](#_Toc10781)

[请求参数 24](#_Toc31105)

[PUT 修改角色 25](#_Toc904)

[请求参数 25](#_Toc17616)

[POST 新增角色 25](#_Toc19218)

[请求参数 26](#_Toc24938)

[GET 获取角色的菜单ID集合 26](#_Toc19640)

[请求参数 26](#_Toc17089)

[GET 角色表单数据 26](#_Toc12283)

[请求参数 26](#_Toc21697)

[GET 角色分页列表 27](#_Toc20928)

[请求参数 27](#_Toc18109)

[GET 角色下拉列表 27](#_Toc23053)

[DELETE 删除角色 28](#_Toc26290)

[请求参数 28](#_Toc21681)

[菜单接口 28](#_Toc22491)

[PUT 修改菜单 28](#_Toc3095)

[请求参数 28](#_Toc1705)

[DELETE 删除菜单 29](#_Toc8151)

[请求参数 29](#_Toc3472)

[GET 菜单列表 29](#_Toc23754)

[请求参数 29](#_Toc6489)

[POST 新增菜单 30](#_Toc19107)

[请求参数 30](#_Toc4304)

[PATCH 修改菜单显示状态 31](#_Toc17240)

[请求参数 31](#_Toc8963)

[GET 菜单表单数据 31](#_Toc19177)

[请求参数 31](#_Toc14454)

[GET 路由列表 32](#_Toc21058)

[GET 菜单下拉列表 32](#_Toc26357)

[字典接口 33](#_Toc9804)

[PUT 修改字典 33](#_Toc17638)

[请求参数 33](#_Toc5226)

[PUT 修改字典类型 33](#_Toc8273)

[请求参数 34](#_Toc22518)

[POST 新增字典 34](#_Toc32497)

[请求参数 34](#_Toc16496)

[POST 新增字典类型 34](#_Toc14351)

[请求参数 35](#_Toc16012)

[GET 字典下拉列表 35](#_Toc15865)

[请求参数 35](#_Toc13883)

[GET 字典数据表单数据 35](#_Toc7237)

[请求参数 35](#_Toc9824)

[GET 字典类型表单数据 36](#_Toc6485)

[请求参数 36](#_Toc31446)

[GET 字典类型分页列表 36](#_Toc7334)

[请求参数 36](#_Toc17385)

[GET 字典分页列表 37](#_Toc18678)

[请求参数 37](#_Toc9854)

[DELETE 删除字典 38](#_Toc28745)

[请求参数 38](#_Toc12913)

[DELETE 删除字典类型 38](#_Toc31051)

[请求参数 38](#_Toc30969)

[部门接口 38](#_Toc18605)

[PUT 修改部门 38](#_Toc19214)

[请求参数 39](#_Toc5731)

[GET 获取部门列表 39](#_Toc2089)

[请求参数 39](#_Toc5625)

[POST 新增部门 40](#_Toc26579)

[请求参数 40](#_Toc21074)

[GET 获取部门表单数据 40](#_Toc15584)

[请求参数 40](#_Toc24712)

[GET 获取部门下拉选项 41](#_Toc26439)

[DELETE 删除部门 41](#_Toc27349)

[请求参数 41](#_Toc4598)

[文件接口 41](#_Toc23460)

[POST 文件上传 41](#_Toc1046)

[请求参数 42](#_Toc29241)

[DELETE 文件删除 42](#_Toc18841)

[请求参数 42](#_Toc5522)

[去雾接口 42](#_Toc21767)

[POST 上传图片 42](#_Toc8056)

[请求参数 42](#_Toc3301)

[GET 下载图片 43](#_Toc26832)

[请求参数 43](#_Toc2350)

[POST 图片去雾 43](#_Toc27046)

[请求参数 43](#_Toc10676)

[POST 评价指标 43](#_Toc4762)

[请求参数 44](#_Toc13411)

[GET 获取去雾模型列表 44](#_Toc29323)

[数据库设计 45](#_Toc25272)

[系统管理相关 45](#_Toc4786)

[用户管理 45](#_Toc16371)

[角色管理 46](#_Toc1170)

[部门管理 46](#_Toc1479)

[菜单管理 47](#_Toc14433)

[字典管理 48](#_Toc21397)

[文件管理 50](#_Toc10210)

[系统日志 51](#_Toc15477)

[其他 53](#_Toc185)

[会员管理相关 54](#_Toc19103)

[套餐管理相关 55](#_Toc4810)

[模型管理相关 56](#_Toc19176)

[订单管理相关 57](#_Toc10782)

[系统部署 57](#_Toc2425)

[操作系统环境搭建 57](#_Toc31074)

[docker搭建 57](#_Toc12520)

[安装docker 57](#_Toc8453)

[配置国内镜像 58](#_Toc18258)

[注意事项 58](#_Toc111)

[mysql搭建 58](#_Toc24886)

[redis搭建 58](#_Toc10746)

[nginx搭建 59](#_Toc8211)

[创建临时容器 59](#_Toc22408)

[创建和启动容器 60](#_Toc799)

[代码运行环境搭建 61](#_Toc27880)

[持续集成部署流程 61](#_Toc657)

[运行情况检查 61](#_Toc15991)

[用户手册 61](#_Toc29264)

[登录到系统 61](#_Toc17813)

[系统首页 61](#_Toc20231)

[去雾模块 62](#_Toc31202)

[VIP及支付模块 64](#_Toc28908)

[会员管理模块 65](#_Toc32108)

[模型管理模块 65](#_Toc14787)

[套餐管理模块 65](#_Toc31531)

[订单管理模块 65](#_Toc16348)

[用户管理模块 66](#_Toc21119)

[角色管理模块 67](#_Toc17951)

[菜单管理模块 68](#_Toc16276)

[部门管理模块 70](#_Toc4455)

[字典管理模块 71](#_Toc28620)

[系统界面以及设置 72](#_Toc28083)

# 系统需求分析

在现代社会中，摄像头已经无处不在，无论是高速公路上的交通监控摄像头，还是人们手中的智能手机，都为我们提供了丰富的图像信息。然而，当恶劣天气如雾霾出现时，图像采集设备可能会受到影响，导致采集的图像模糊，对比度降低，甚至无法辨别图像中的物体。因此，开发一款能够在线实时响应的图像去雾系统，将极大地方便人们的生活，特别是对生活在雾霾地区的用户来说，这是一项急需解决的问题。

本系统是一套基于深度学习的去雾系统，利用先进的人工智能技术，通过训练神经网络模型对图像进行处理，从而实现去雾的目标。这不仅是一种技术上的创新，也是对人们生活质量改善的有力贡献。我们期待这款系统能够在各种应用场景中发挥其独特的价值，为人们提供更清晰、更准确的图像信息。

## 用户需求

1. 普通手机拍照用户  
   实时去雾：用户希望在拍照时，系统能够实时地去除雾霾，提高照片的清晰度。  
   易用性：系统应该有简单易懂的用户界面，用户可以轻松地选择和调整去雾效果。  
   自动优化：系统应能自动识别图片中的雾霾，并自动进行去雾处理。
2. 专业摄影师  
   高质量去雾：对于专业摄影师来说，去雾效果的质量非常重要。系统应能提供高质量的去雾效果，保证照片的细节和色彩不受损失。  
   可调性：摄影师可能需要根据具体情况调整去雾效果，系统应提供灵活的调整选项。
3. 图像处理软件后期处理  
   批量处理：在后期处理中，可能需要对大量图片进行去雾处理，系统应支持批量处理功能。  
   无损处理：系统应能提供无损的去雾处理，不改变原始图片的其他属性。

## 场景需求

1. 日常生活随手拍照场景  
   快速去雾：在日常生活中，用户可能希望快速地去除照片中的雾霾，系统应能提供快速的去雾处理。
2. 地面道路状况实时分析场景  
   实时去雾：在道路状况分析中，实时的去雾处理非常重要，系统应能提供实时的去雾处理。  
   稳定性：由于道路状况分析对准确性要求很高，系统应能提供稳定可靠的去雾效果。
3. 室内室外监控场景  
   全天候去雾：监控系统通常需要全天候运行，系统应能在各种光照和天气条件下提供有效的去雾处理。
4. 遥感图像分析场景  
   大规模去雾：在遥感图像分析中，可能需要对大规模的图像进行去雾处理，系统应能有效处理大规模的图像数据。
5. 影视后期场景  
   高质量去雾：在影视后期处理中，去雾效果的质量直接影响到最终的视觉效果，系统应能提供高质量的去雾效果。

## 系统需求

1. 用户管理  
   用户管理模块作为图像去雾系统的基础模块，能够识别使用系统的各种用户的身份。主要负责用户的增删改查、密码重置、用户状态管理等功能。此外，还需要提供用户登录日志查询，以便于跟踪用户的操作行为。
2. 权限管理  
   权限管理模块主要负责系统权限的分配和管理。系统管理员可以通过此模块给用户分配不同的权限，如查看、编辑、删除等。此外，还需要提供权限的动态配置功能，以便于根据业务需求灵活调整用户权限。
3. 界面菜单管理  
   界面菜单管理模块主要负责系统菜单的配置和管理。系统管理员可以通过此模块配置菜单的层级结构，设置菜单的显示名称、链接地址、图标等属性。此外，还需要提供菜单的动态配置功能，以便于根据用户权限动态生成菜单。
4. 角色管理  
   角色管理模块主要负责系统角色的定义和管理。系统管理员可以通过此模块创建角色，并为角色分配权限。此外，还需要提供角色的动态配置功能，以便于根据业务需求灵活调整角色权限。
5. 系统数据分析  
   系统数据分析模块主要负责系统运营数据的统计和分析。系统管理员可以通过此模块查看系统的运营数据，如用户数量、活跃度、操作日志等。此外，还需要提供数据的可视化展示，以便于管理员直观了解系统运营状况。

## 非功能需求

### 性能需求

1. 文件系统要求  
   由于系统需要进行图片处理操作，需要
2. 网络带宽要求
3. 响应时间  
   系统应具有快速的响应时间。无论是页面加载、数据查询还是操作反馈，系统都应在最短的时间内给出响应。具体的响应时间可能会根据具体的操作和数据量有所不同，但一般来说，任何操作的响应时间都不应超过2秒。
4. 并发处理能力  
   系统应具有良好的并发处理能力。系统应能同时处理多个用户的请求，而不会出现延迟或错误。具体的并发处理能力可能会根据系统的硬件配置和软件优化有所不同，但一般来说，系统应能同时处理上百个用户的请求。
5. 数据处理能力  
   后台管理系统应具有强大的数据处理能力。系统应能快速处理大量的数据，包括数据查询、数据更新和数据分析等操作。具体的数据处理能力可能会根据数据的复杂性和数据量有所不同，但一般来说，系统应能在几秒钟内处理上万条数据。
6. 系统稳定性  
   后台管理系统应具有高度的稳定性。系统应能在各种情况下稳定运行，不会出现崩溃或错误。此外，系统还应具有良好的容错能力，即使在出现错误的情况下，也能保证系统的正常运行。
7. 安全性  
   后台管理系统应具有高度的安全性。系统应能防止各种安全威胁，包括数据泄露、非法访问和恶意攻击等。此外，系统还应具有完善的安全审计功能，能记录和分析所有的安全事件。

# 系统设计

## 系统架构设计

### 业务架构

业务架构主要考虑到软件的功能需求，进行问题域的划分和领域建模等工作。架构模式如下：



### 代码架构

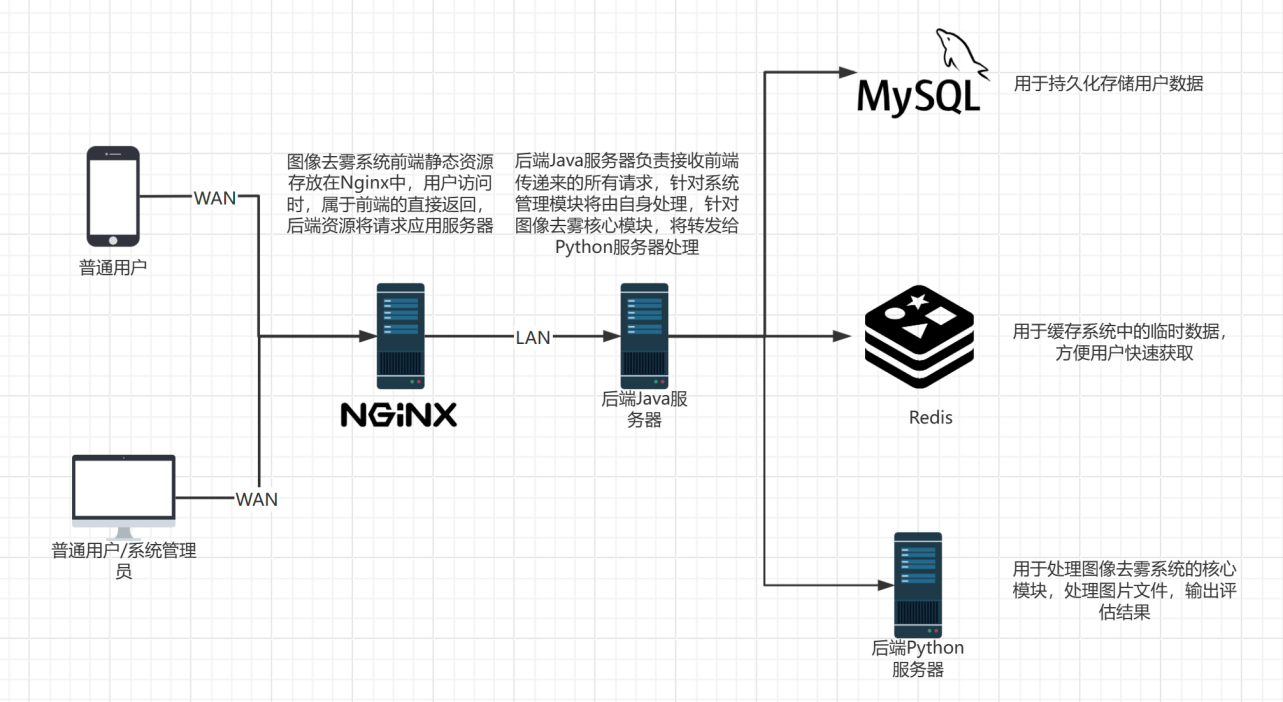
在代码层面上，系统主要分为3块。

第一部分是前端，前端是图像去雾系统的前台部分，运行在PC端，移动端等浏览器上展现给用户浏览的网页。随着互联网技术的发展，HTML5，CSS3，前端框架的应用，跨平台响应式网页设计能够适应各种屏幕分辨率，完美的动效设计，给用户带来极高的用户体验。本系统主要依靠前端网页和用户进行交互。

第二部分是JAVA后端，运行在Java Web服务器Tomcat上，采用SpringBoot Java Web技术进行开发。java后端主要用作系统的基础管理如用户管理、角色管理、会员管理和日志记录上。由于Java在网页开发上的优势，本系统后端的绝大部分内容采用Java来开发。

第三部分是Python后端，由于深度学习技术主要基于Python、pytorch框架执行。因此系统最核心的部分采用Python进行编写，同时向外提供API接口以供Java后端调用。

### 部署架构



## 技术栈

### 前端

| 环境 | 版本 |
| --- | --- |
| node | v18.18.0 |
| npm | 9.8.1 |
| axios | ^1.4.0 |
| echarts | ^5.2.2 |
| element-plus | ^2.3.9 |
| pinia | ^2.0.33 |
| vue | ^3.3.4 |
| vue-router | ^4.2.0 |
| vite | ^4.4.2 |
| 更详细内容 | 见前端代码package.json |

### JAVA后端

**Java 环境要求**

| 环境 | 版本 |
| --- | --- |
| Java | 17 |
| maven | 3.9.5 |
| SpringBoot | 3.1.0 |
| hutool | 5.8.15 |
| mysql driver | 8.0.28 |
| druid | 1.2.16 |
| mybatis-plus | 3.5.3.1 |
| knife4j | 4.0.0 |
| mapstruct | 1.5.3.Final |
| lombok-mapstruct-binding | 0.2.0 |
| xxl-job | 2.4.0 |
| jjwt | 0.11.5 |
| easyexcel | 3.2.1 |

**中间件要求**

| 软件 | 版本 |
| --- | --- |
| mysql | latest |
| redis | latest |

### Python后端

**软件环境要求**

| 环境 | 版本 |
| --- | --- |
| python | 3.8.16 |
| torch | ~=1.8.1+cu101 |
| opencv-python | ~=4.8.1.78 |
| torchvision | ~=0.9.1+cu101 |
| timm | ~=0.9.8 |
| numpy | ~=1.24.4 |
| Pillow | ~=10.0.1 |
| scikit-image | ~=0.21.0 |
| thop | ~=0.1.1.post2209072238 |
| einops | ~=0.7.0 |
| PyYAML | ~=6.0.1 |
| yacs | ~=0.1.8 |
| Django | ~=4.2.7 |

**硬件环境要求**

| 要求 | 详情 |
| --- | --- |
| 操作系统 | Ubuntu 18.04.6 LTS |
| CPU | 暂无要求 |
| 内存 | 16G以上 |
| GPU | Nvidia GPU 显存16G以上 |
| GPU Driver | 450.119.03 |
| CUDA | 10.1 |
| gcc | 9.3.0 |

## 功能模块设计

### 去雾模块

去雾模块是整个系统中最核心的模块，根据需求分析，该模块一共分为图片上传、图片下载、指标评价和去雾处理四个功能。

用户在本系统进行去雾操作时需要先选择系统中现有的去雾模型，不同的模型针对的去雾场景不同，去雾效果也有所差别。在选好去雾模型后上传有雾图像到系统中，随后点击按钮开始去雾。系统将会把请求发送给后端进行处理。随后将去雾结果返还给前端。此时前端系统会自动根据结果选择从后端获取模型去雾后的图像，或者是出现错误，将错误原因通知前端。

用户如果想要评估该模型的去雾效果，则需要提前准备好与有雾图像尺寸大小对应的无雾图像，点击效果评估按钮，将无雾图像上传至系统后系统就会调用评价算法评估模型去雾后的图像和真实无雾图像之间的相似度，并且针对去雾效果给出客观评价。

#### 图片上传

图像上传包含有雾图像和无雾基准图像的上传，

#### 图片下载

#### 指标评价

#### 去雾处理

#### 历史记录

### 用户模块

#### 操作记录

### 菜单模块

### 权限模块

### 角色模块

### 数据分析模块

# API接口设计

## 认证接口

### POST 登录

POST /api/v1/auth/login

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| username | query | string | 是 | 用户名 |
| password | query | string | 是 | 密码 |

返回示例

{  
 "code": "string",  
 "data": {  
 "accessToken": "string",  
 "tokenType": "Bearer",  
 "refreshToken": "string",  
 "expires": 0  
 },  
 "msg": "string"  
}

### GET 获取验证码

GET /api/v1/auth/captcha

返回示例

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

### DELETE 注销

DELETE /api/v1/auth/logout

返回示例

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

## 用户接口

### PUT 修改用户

PUT /api/v1/users/{userId}

Body 请求参数

{  
 "id": 0,  
 "username": "string",  
 "nickname": "string",  
 "mobile": "string",  
 "gender": 0,  
 "avatar": "string",  
 "email": "string",  
 "status": 0,  
 "deptId": 0,  
 "roleIds": [  
 0  
 ]  
}

### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| userId | path | integer | 是 | 用户ID |
| body | body | [UserForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### POST 新增用户

POST /api/v1/users

Body 请求参数

{  
 "id": 0,  
 "username": "string",  
 "nickname": "string",  
 "mobile": "string",  
 "gender": 0,  
 "avatar": "string",  
 "email": "string",  
 "status": 0,  
 "deptId": 0,  
 "roleIds": [  
 0  
 ]  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [UserForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### POST 导入用户

POST /api/v1/users/\_import

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| deptId | query | integer | 是 | 部门ID |
| file | query | string | 是 | none |

返回示例

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

### PATCH 修改用户状态

PATCH /api/v1/users/{userId}/status

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| userId | path | integer | 是 | 用户ID |
| status | query | integer | 是 | 用户状态(1:启用;0:禁用) |

返回示例

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

### PATCH 修改用户密码

PATCH /api/v1/users/{userId}/password

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| userId | path | integer | 是 | 用户ID |
| password | query | string | 是 | none |

返回示例

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

### GET 用户表单数据

GET /api/v1/users/{userId}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| userId | path | integer | 是 | 用户ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "username": "string",  
 "nickname": "string",  
 "mobile": "string",  
 "gender": 0,  
 "avatar": "string",  
 "email": "string",  
 "status": 0,  
 "deptId": 0,  
 "roleIds": [  
 0  
 ]  
 },  
 "msg": "string"  
}

### GET 用户导入模板下载

GET /api/v1/users/template

### GET 用户分页列表

GET /api/v1/users/page

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(用户名/昵称/手机号) |
| status | query | string | 否 | 用户状态 |
| deptId | query | string | 否 | 部门ID |
| pageNum | query | string | 否 | 页码 |
| pageSize | query | string | 否 | 每页记录数 |

返回示例

{  
 "code": "string",  
 "data": {  
 "list": [  
 {  
 "id": 0,  
 "username": "string",  
 "nickname": "string",  
 "mobile": "string",  
 "genderLabel": "string",  
 "avatar": "string",  
 "email": "string",  
 "status": 0,  
 "deptName": "string",  
 "roleNames": "string",  
 "createTime": "2019-08-24T14:15:22Z"  
 }  
 ],  
 "total": 0  
 },  
 "msg": "string"  
}

### GET 获取当前登录用户信息

GET /api/v1/users/me

返回示例

{  
 "code": "string",  
 "data": {  
 "userId": 0,  
 "nickname": "string",  
 "avatar": "string",  
 "roles": [  
 "string"  
 ],  
 "perms": [  
 "string"  
 ]  
 },  
 "msg": "string"  
}

### GET 导出用户

GET /api/v1/users/\_export

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| pageNum | query | integer | 否 | 页码 |
| pageSize | query | integer | 否 | 每页记录数 |
| keywords | query | string | 否 | 关键字(用户名/昵称/手机号) |
| status | query | integer | 否 | 用户状态 |
| deptId | query | integer | 否 | 部门ID |

### DELETE 删除用户

DELETE /api/v1/users/{ids}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| ids | path | string | 是 | 用户ID，多个以英文逗号(,)分割 |

返回示例

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

## 角色接口

### PUT 修改角色状态

PUT /api/v1/roles/{roleId}/status

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| roleId | path | integer | 是 | 角色ID |
| status | query | integer | 是 | 状态(1:启用;0:禁用) |

返回示例

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

### PUT 分配菜单权限给角色

PUT /api/v1/roles/{roleId}/menus

Body 请求参数

[  
 0  
]

### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| roleId | path | integer | 是 | none |
| body | body | array[integer] | 否 | none |

返回示例

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

### PUT 修改角色

PUT /api/v1/roles/{id}

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "sort": 0,  
 "status": 0,  
 "dataScope": 0  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | string | 是 | none |
| body | body | [RoleForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### POST 新增角色

POST /api/v1/roles

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "sort": 0,  
 "status": 0,  
 "dataScope": 0  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [RoleForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### GET 获取角色的菜单ID集合

GET /api/v1/roles/{roleId}/menuIds

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| roleId | path | integer | 是 | 角色ID |

返回示例

{  
 "code": "string",  
 "data": [  
 0  
 ],  
 "msg": "string"  
}

### GET 角色表单数据

GET /api/v1/roles/{roleId}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| roleId | path | integer | 是 | 角色ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "sort": 0,  
 "status": 0,  
 "dataScope": 0  
 },  
 "msg": "string"  
}

### GET 角色分页列表

GET /api/v1/roles/page

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(角色名称/角色编码) |
| pageNum | query | string | 否 | 页码 |
| pageSize | query | string | 否 | 每页记录数 |

返回示例

{  
 "code": "string",  
 "data": {  
 "list": [  
 {  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "status": 0,  
 "sort": 0,  
 "createTime": "2019-08-24T14:15:22Z",  
 "updateTime": "2019-08-24T14:15:22Z"  
 }  
 ],  
 "total": 0  
 },  
 "msg": "string"  
}

### GET 角色下拉列表

GET /api/v1/roles/options

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "value": {},  
 "label": "string",  
 "children": [  
 {  
 "value": {},  
 "label": "string"  
 }  
 ]  
 }  
 ],  
 "msg": "string"  
}

### DELETE 删除角色

DELETE /api/v1/roles/{ids}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| ids | path | string | 是 | 删除角色，多个以英文逗号(,)分割 |

返回示例

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

## 菜单接口

### PUT 修改菜单

PUT /api/v1/menus/{id}

Body 请求参数

{  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "type": "NULL",  
 "path": "string",  
 "component": "string",  
 "perm": "string",  
 "visible": 0,  
 "sort": 0,  
 "icon": "string",  
 "redirect": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | string | 是 | none |
| body | body | [MenuForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### DELETE 删除菜单

DELETE /api/v1/menus/{id}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | 菜单ID，多个以英文(,)分割 |

返回示例

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

### GET 菜单列表

GET /api/v1/menus

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(菜单名称) |
| status | query | string | 否 | 状态(1->显示；0->隐藏) |

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "type": "NULL",  
 "path": "string",  
 "component": "string",  
 "sort": 0,  
 "visible": 0,  
 "icon": "string",  
 "redirect": "string",  
 "perm": "string",  
 "children": [  
 {  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "type": "NULL",  
 "path": "string",  
 "component": "string",  
 "sort": 0,  
 "visible": 0,  
 "icon": "string",  
 "redirect": "string",  
 "perm": "string",  
 "children": [  
 {}  
 ]  
 }  
 ]  
 }  
 ],  
 "msg": "string"  
}

### POST 新增菜单

POST /api/v1/menus

Body 请求参数

{  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "type": "NULL",  
 "path": "string",  
 "component": "string",  
 "perm": "string",  
 "visible": 0,  
 "sort": 0,  
 "icon": "string",  
 "redirect": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [MenuForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### PATCH 修改菜单显示状态

PATCH /api/v1/menus/{menuId}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| menuId | path | integer | 是 | 菜单ID |
| visible | query | integer | 是 | 显示状态(1:显示;0:隐藏) |

返回示例

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

### GET 菜单表单数据

GET /api/v1/menus/{id}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | 菜单ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "type": "NULL",  
 "path": "string",  
 "component": "string",  
 "perm": "string",  
 "visible": 0,  
 "sort": 0,  
 "icon": "string",  
 "redirect": "string"  
 },  
 "msg": "string"  
}

### GET 路由列表

GET /api/v1/menus/routes

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "path": "user",  
 "component": "system/user/index",  
 "redirect": "https://www.youlai.tech",  
 "name": "string",  
 "meta": {  
 "title": "string",  
 "icon": "string",  
 "hidden": true,  
 "roles": "['ADMIN','ROOT']",  
 "keepAlive": true  
 },  
 "children": [  
 {  
 "path": "user",  
 "component": "system/user/index",  
 "redirect": "https://www.youlai.tech",  
 "name": "string",  
 "meta": {  
 "title": null,  
 "icon": null,  
 "hidden": null,  
 "roles": null,  
 "keepAlive": null  
 },  
 "children": [  
 {}  
 ]  
 }  
 ]  
 }  
 ],  
 "msg": "string"  
}

### GET 菜单下拉列表

GET /api/v1/menus/options

返回示例

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

## 字典接口

### PUT 修改字典

PUT /api/v1/dict/{id}

Body 请求参数

{  
 "id": 0,  
 "typeCode": "string",  
 "name": "string",  
 "value": "string",  
 "status": 0,  
 "sort": 0,  
 "remark": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | none |
| body | body | [DictForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### PUT 修改字典类型

PUT /api/v1/dict/types/{id}

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "status": 0,  
 "remark": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | none |
| body | body | [DictTypeForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### POST 新增字典

POST /api/v1/dict

Body 请求参数

{  
 "id": 0,  
 "typeCode": "string",  
 "name": "string",  
 "value": "string",  
 "status": 0,  
 "sort": 0,  
 "remark": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [DictForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### POST 新增字典类型

POST /api/v1/dict/types

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "status": 0,  
 "remark": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [DictTypeForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### GET 字典下拉列表

GET /api/v1/dict/{typeCode}/options

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| typeCode | path | string | 是 | 字典类型编码 |

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "value": {},  
 "label": "string",  
 "children": [  
 {  
 "value": {},  
 "label": "string"  
 }  
 ]  
 }  
 ],  
 "msg": "string"  
}

### GET 字典数据表单数据

GET /api/v1/dict/{id}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | 字典ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "typeCode": "string",  
 "name": "string",  
 "value": "string",  
 "status": 0,  
 "sort": 0,  
 "remark": "string"  
 },  
 "msg": "string"  
}

### GET 字典类型表单数据

GET /api/v1/dict/types/{id}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| id | path | integer | 是 | 字典ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "status": 0,  
 "remark": "string"  
 },  
 "msg": "string"  
}

### GET 字典类型分页列表

GET /api/v1/dict/types/page

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(类型名称/类型编码) |
| pageNum | query | string | 否 | 页码 |
| pageSize | query | string | 否 | 每页记录数 |

返回示例

{  
 "code": "string",  
 "data": {  
 "list": [  
 {  
 "id": 0,  
 "name": "string",  
 "code": "string",  
 "status": 0  
 }  
 ],  
 "total": 0  
 },  
 "msg": "string"  
}

### GET 字典分页列表

GET /api/v1/dict/page

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(字典项名称) |
| typeCode | query | string | 否 | 字典类型编码 |
| pageNum | query | string | 否 | 页码 |
| pageSize | query | string | 否 | 每页记录数 |

返回示例

200 Response

{  
 "code": "string",  
 "data": {  
 "list": [  
 {  
 "id": 0,  
 "name": "string",  
 "value": "string",  
 "status": 0  
 }  
 ],  
 "total": 0  
 },  
 "msg": "string"  
}

### DELETE 删除字典

DELETE /api/v1/dict/{ids}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| ids | path | string | 是 | 字典ID，多个以英文逗号(,)拼接 |

返回示例

200 Response

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

### DELETE 删除字典类型

DELETE /api/v1/dict/types/{ids}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| ids | path | string | 是 | 字典类型ID，多个以英文逗号(,)分割 |

返回示例

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

## 部门接口

### PUT 修改部门

PUT /api/v1/dept/{deptId}

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "parentId": 0,  
 "status": 0,  
 "sort": 0  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| deptId | path | integer | 是 | none |
| body | body | [DeptForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

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

### GET 获取部门列表

GET /api/v1/dept

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| keywords | query | string | 否 | 关键字(部门名称) |
| status | query | string | 否 | 状态(1->正常；0->禁用) |

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "sort": 0,  
 "status": 0,  
 "children": [  
 {  
 "id": 0,  
 "parentId": 0,  
 "name": "string",  
 "sort": 0,  
 "status": 0,  
 "children": [  
 {}  
 ],  
 "createTime": "2019-08-24T14:15:22Z",  
 "updateTime": "2019-08-24T14:15:22Z"  
 }  
 ],  
 "createTime": "2019-08-24T14:15:22Z",  
 "updateTime": "2019-08-24T14:15:22Z"  
 }  
 ],  
 "msg": "string"  
}

### POST 新增部门

POST /api/v1/dept

Body 请求参数

{  
 "id": 0,  
 "name": "string",  
 "parentId": 0,  
 "status": 0,  
 "sort": 0  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | [DeptForm](#Xa39a3ee5e6b4b0d3255bfef95601890afd80709) | 否 | none |

返回示例

200 Response

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

### GET 获取部门表单数据

GET /api/v1/dept/{deptId}/form

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| deptId | path | integer | 是 | 部门ID |

返回示例

{  
 "code": "string",  
 "data": {  
 "id": 0,  
 "name": "string",  
 "parentId": 0,  
 "status": 0,  
 "sort": 0  
 },  
 "msg": "string"  
}

### GET 获取部门下拉选项

GET /api/v1/dept/options

返回示例

{  
 "code": "string",  
 "data": [  
 {  
 "value": {},  
 "label": "string",  
 "children": [  
 {  
 "value": {},  
 "label": "string"  
 }  
 ]  
 }  
 ],  
 "msg": "string"  
}

### DELETE 删除部门

DELETE /api/v1/dept/{ids}

### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| ids | path | string | 是 | 部门ID，多个以英文逗号(,)分割 |

返回示例

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

## 文件接口

### POST 文件上传

POST /api/v1/files

Body 请求参数

{  
 "file": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | object | 否 | none |
| » file | body | string(binary) | 是 | 表单文件对象 |

返回示例

{  
 "code": "string",  
 "data": {  
 "name": "string",  
 "url": "string"  
 },  
 "msg": "string"  
}

### DELETE 文件删除

DELETE /api/v1/files

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| filePath | query | string | 是 | 文件路径 |

返回示例

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

## 去雾接口

### POST 上传图片

POST /upload/

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | string(binary) | 否 | none |

返回示例

{  
 "code": "string",  
 "msg": "string",  
 "data": {  
 "image\_name": "string"  
 }  
}

### GET 下载图片

GET /download/{image\_name}/

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| image\_name | path | string | 是 | none |

### POST 图片去雾

POST /dehazeImage/

Body 请求参数

{  
 "haze\_image": "string",  
 "model\_name": "C2PNet/OTS.pkl"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | object | 否 | none |
| » haze\_image | body | string | 是 | 有雾图片的文件名 |
| » model\_name | body | string | 是 | none |

##### 枚举值

| 属性 | 值 |
| --- | --- |
| » model\_name | C2PNet/OTS.pkl |

返回示例

{  
 "code": "string",  
 "msg": "string",  
 "data": {  
 "image\_name": "string"  
 }  
}

### POST 评价指标

POST /calculateIndex/

获取图像去雾的评价指标，PSNR和SSIM

Body 请求参数

{  
 "haze\_image": "string",  
 "clear\_image": "string"  
}

#### 请求参数

| 名称 | 位置 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- | --- |
| body | body | object | 否 | none |
| » haze\_image | body | string | 是 | none |
| » clear\_image | body | string | 是 | none |

返回示例

{  
 "code": "string",  
 "msg": "string",  
 "data": {  
 "psnr": 0,  
 "ssim": 0  
 }  
}

### GET 获取去雾模型列表

GET /model/

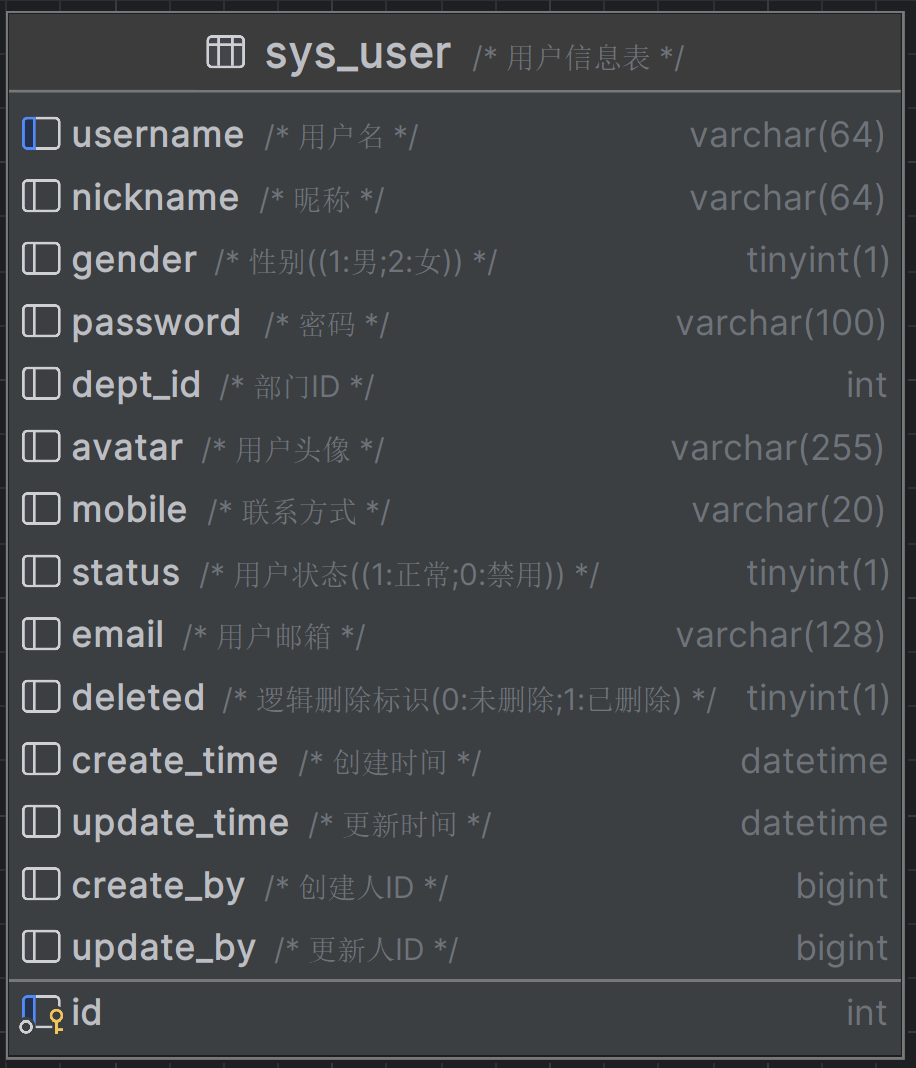
返回示例

{
  
 "code": "string",
  
 "msg": "string",
  
 "data": [
  
 {
  
 "value": "string",
  
 "label": "string",
  
 "children": [
  
 {
  
 "value": "string",
  
 "label": "string",
  
 "children": [
  
 {}
  
 ]
  
 }
  
 ]
  
 }
  
 ]
  
}

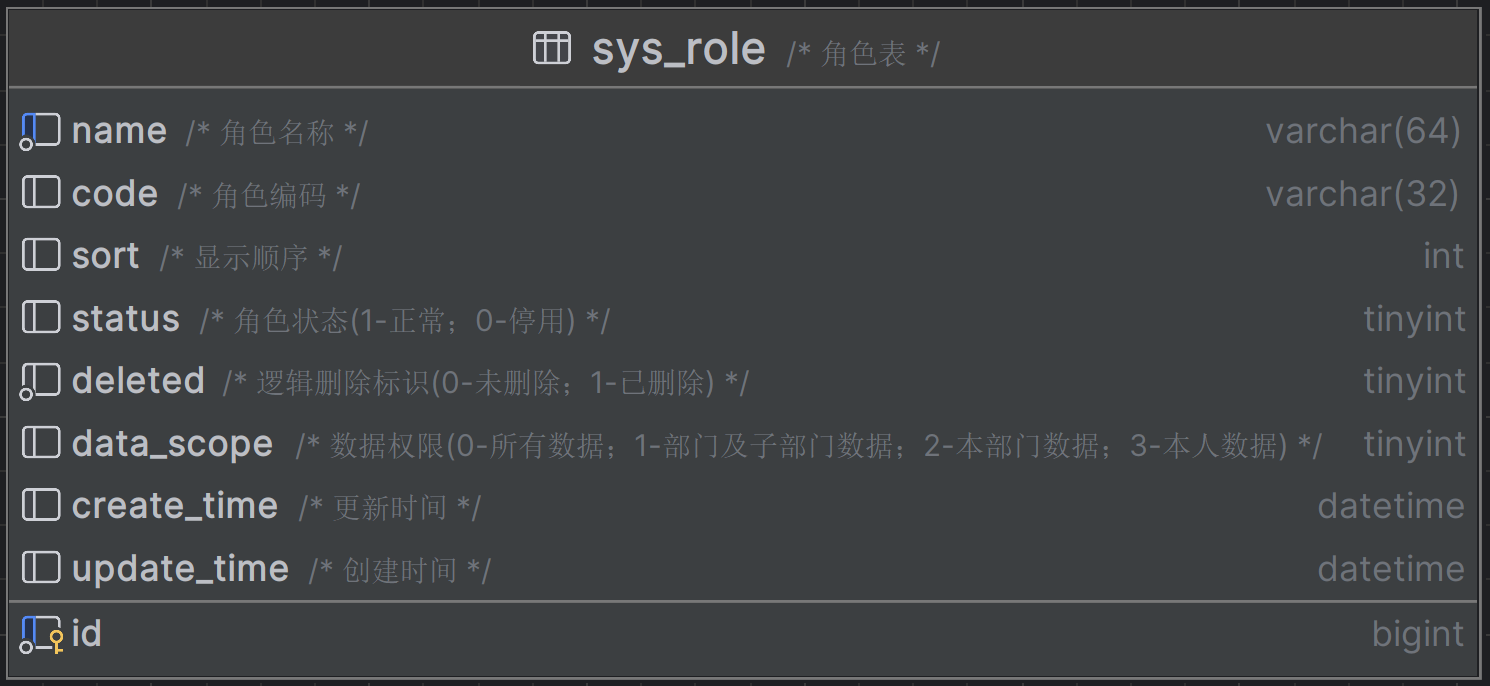
# 数据库设计

## 系统管理相关

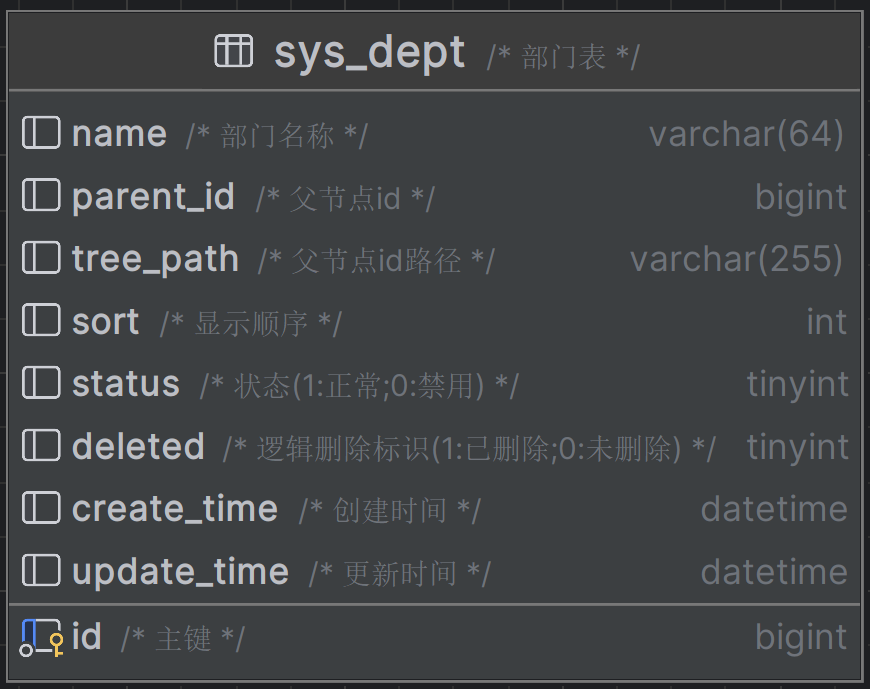
### 用户管理



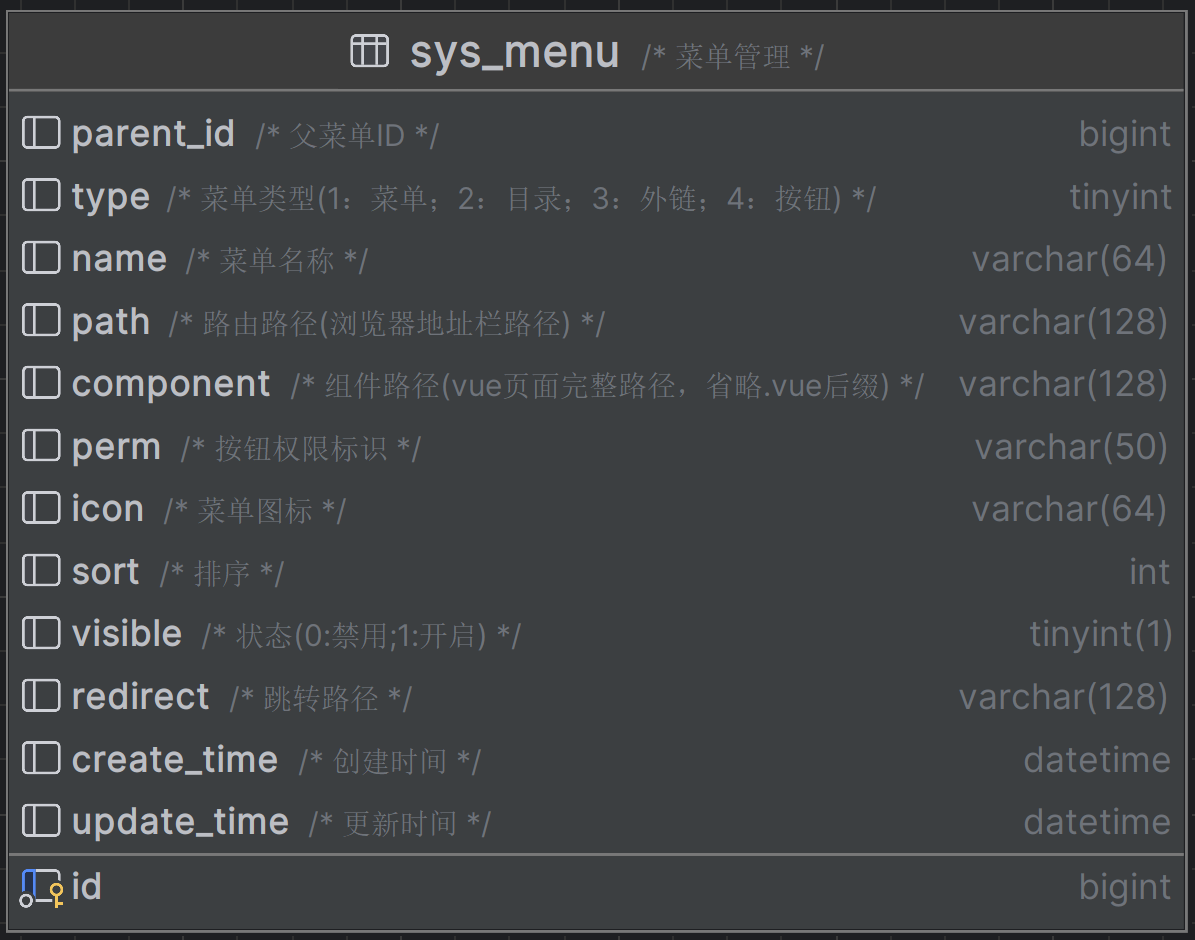
### 角色管理



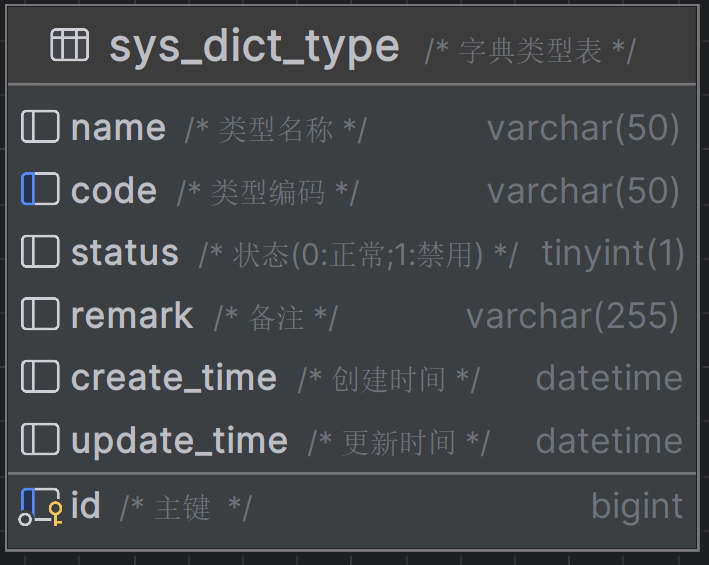
### 部门管理

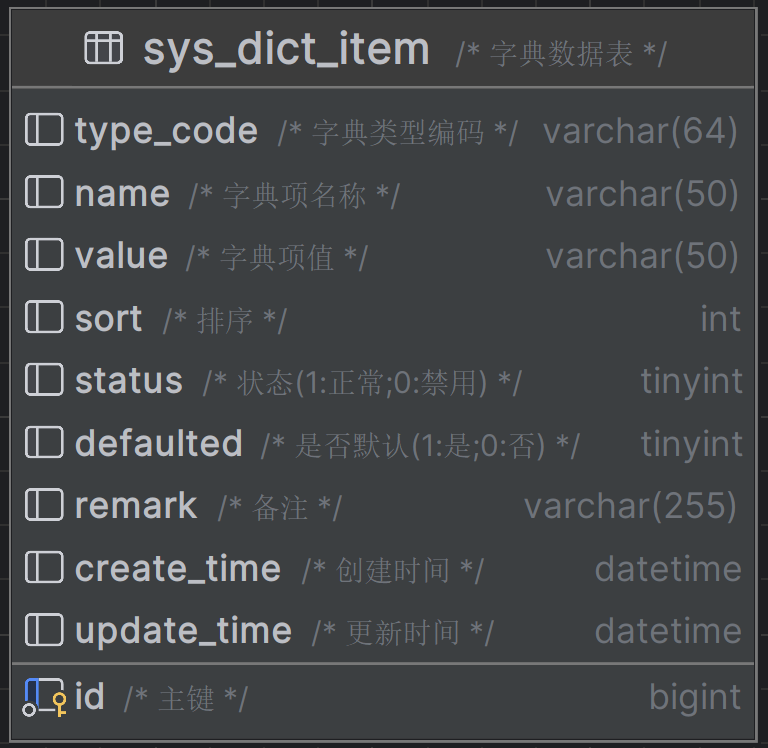


### 菜单管理

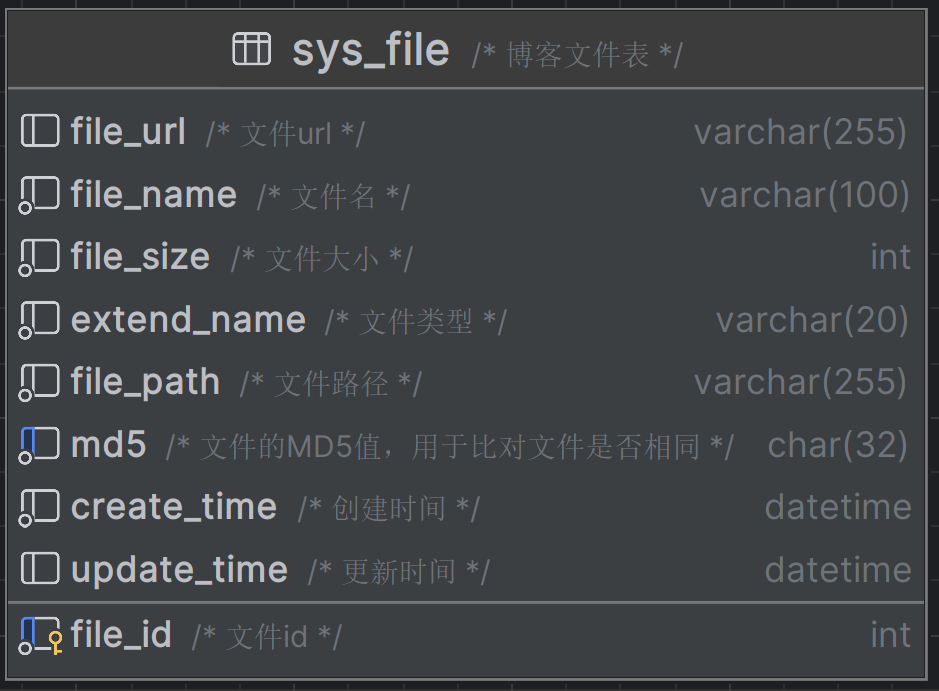


### 字典管理



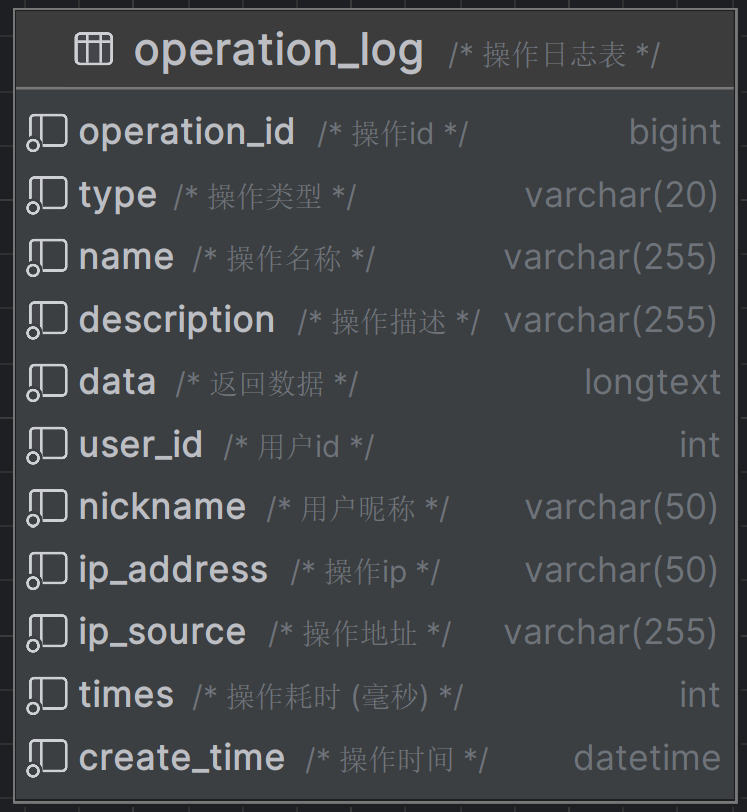


### 文件管理



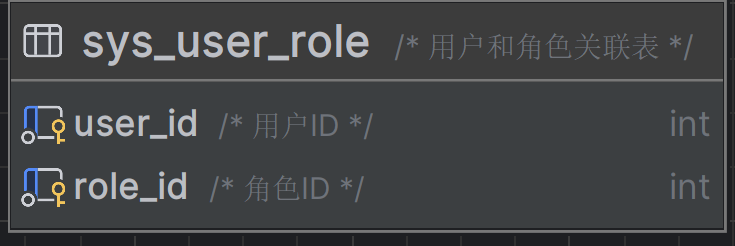
### 系统日志



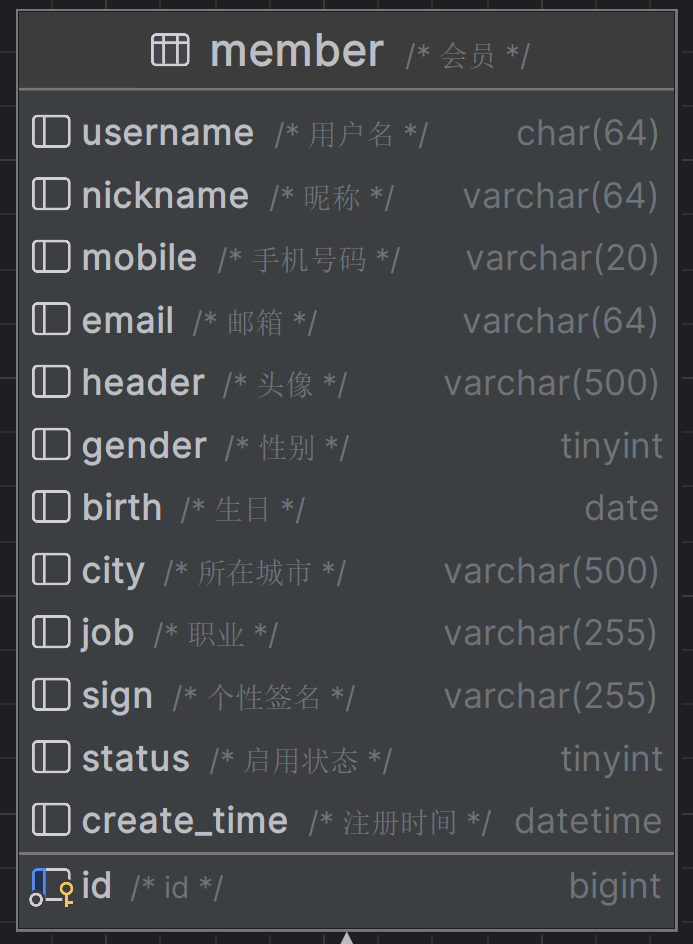


### 其他

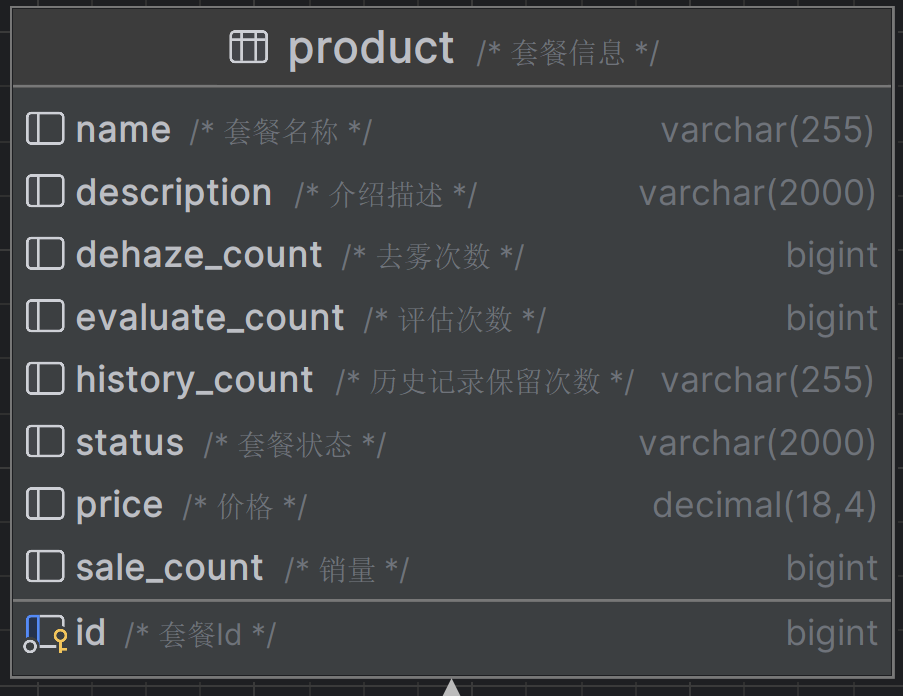




## 会员管理相关



## 套餐管理相关



## 模型管理相关



## 订单管理相关



# 系统部署

程序代码行数：14000+

## 操作系统环境搭建

### docker搭建

#### 安装docker

使用官方给出的脚本，<https://get.docker.com/>

# 获取并运行安装脚本  
curl -fsSL get.docker.com -o get-docker.sh  
sudo sh get-docker.sh  
# 启动docker服务  
systemctl start docker  
# 配置docker开机自启  
systemctl enable docker

#### 配置国内镜像

修改配置文件

sudo vim /etc/docker/daemon.json

添加内容

{  
 "registry-mirrors":[  
 "https://o65lma2s.mirror.aliyuncs.com",  
 "http://hub-mirror.c.163.com"  
 ],  
 "insecure-registries" : ["192.168.210.100:5000"]  
}

重启docker

systemctl daemon-reload && systemctl restart docker

#### 注意事项

如果以非 root 用户可以运行 docker 时，  
需要执行 sudo usermod -aG docker cmsblogs 命令然后重新登陆，  
否则会有如下报错

docker: Cannot connect to the Docker daemon.   
Is the docker daemon running on this host ?

### mysql搭建

docker run --restart=always \  
-d \  
-p 3307:3306 \  
--privileged=true \  
-v /opt/docker\_volume/mysql/log:/var/log/mysql \  
-v /opt/docker\_volume/mysql/data:/var/lib/mysql \  
-v /opt/docker\_volume/mysql/conf:/etc/mysql/conf.d \  
-e MYSQL\_ROOT\_PASSWORD=123456 \  
--name mysql \  
mysql

### redis搭建

官网获取 redis.conf : <http://www.redis.cn/download.html>

mkdir -p /opt/docker\_volume/redis

解压后可在根目录查看到 redis.conf 配置文件，将其内容复制到 /opt/docker\_volume/redis/redis.conf，然后修改以下内容：

* bind 127.0.0.1 ： 注释掉，redis可以外部访问
* --requirepass "123456" ： 设置Redis密码为123456，默认无密码
* --appendonly yes ： AOF持久化
* tcp-keepalive 60 ： 默认300，调小防止远程主机强迫关闭一个现有连接错误

docker run --restart=always \  
--log-opt max-size=100m \  
--log-opt max-file=2 \  
-p 6379:6379 \  
--name redis \  
-v /opt/docker\_volume/redis/redis.conf:/etc/redis/redis.conf \  
-v /opt/docker\_volume/redis/data:/data \  
-d redis redis-server /etc/redis/redis.conf \  
--appendonly yes \  
--requirepass 142536aA

### nginx搭建

#### 创建临时容器

启动前需要先创建Nginx外部挂载的配置文件，之所以要先创建 , 是因为Nginx本身容器只存在/etc/nginx 目录 , 本身就不创建 nginx.conf 文件  
当服务器和容器都不存在 nginx.conf 文件时, 执行启动命令的时候 docker会将nginx.conf 作为目录创建 , 这并不是我们想要的结果 。

（ /home/nginx/conf/nginx.conf）

# 创建挂载目录  
mkdir -p /opt/docker\_volume/nginx/conf && mkdir -p /opt/docker\_volume/log && mkdir -p /opt/docker\_volume/nginx/html  
 # 或  
sudo mkdir -p /home/earthyzinc/nginx/conf && sudo mkdir -p /home/earthyzinc/nginx/log && sudo mkdir -p /home/earthyzinc/nginx/html  
# 创建文件  
touch /opt/docker\_volume/nginx/conf/nginx.conf  
# 生成容器  
docker run --name nginx -p 9527:80 -d nginx  
# 将容器nginx.conf文件, conf.d文件夹下内容, html文件夹复制到宿主机  
docker cp nginx:/etc/nginx/nginx.conf /opt/docker\_volume/nginx/conf/nginx.conf  
  
docker cp nginx:/etc/nginx/conf.d /opt/docker\_volume/nginx/conf/conf.d  
  
docker cp nginx:/usr/share/nginx/html /opt/docker\_volume/nginx  
# 删除临时容器  
docker rm -f nginx

#### 创建和启动容器

创建容器互联网络

docker network create -d bridge earthy-net

docker run --restart=always \  
--name nginx \  
-p 9531:9531 \  
-p 9532:9532 \  
-v /opt/docker\_volume/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \  
-v /opt/docker\_volume/nginx/conf/conf.d:/etc/nginx/conf.d \  
-v /opt/docker\_volume/nginx/log:/var/log/nginx \  
-v /opt/docker\_volume/nginx/html:/usr/share/nginx/html \  
-d nginx  
  
docker run --restart=always \  
--name nginx \  
--network host \  
-v /home/earthyzinc/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \  
-v /home/earthyzinc/nginx/conf/conf.d:/etc/nginx/conf.d \  
-v /home/earthyzinc/nginx/log:/var/log/nginx \  
-v /home/earthyzinc/nginx/html:/usr/share/nginx/html \  
-v /home/earthyzinc/nginx/ssl:/usr/share/nginx/ssl \  
-d nginx

* -v /opt/docker\_volume/nginx/conf/nginx.conf:/etc/nginx/nginx.conf 挂载nginx.conf配置文件
* -v /opt/docker\_volume/nginx/conf/conf.d:/etc/nginx/conf.d 挂载nginx配置文件
* -v /opt/docker\_volume/nginx/log:/var/log/nginx 挂载nginx日志文件
* -v /opt/docker\_volume/nginx/html:/usr/share/nginx/html 挂载nginx内容

## 代码运行环境搭建

## 持续集成部署流程

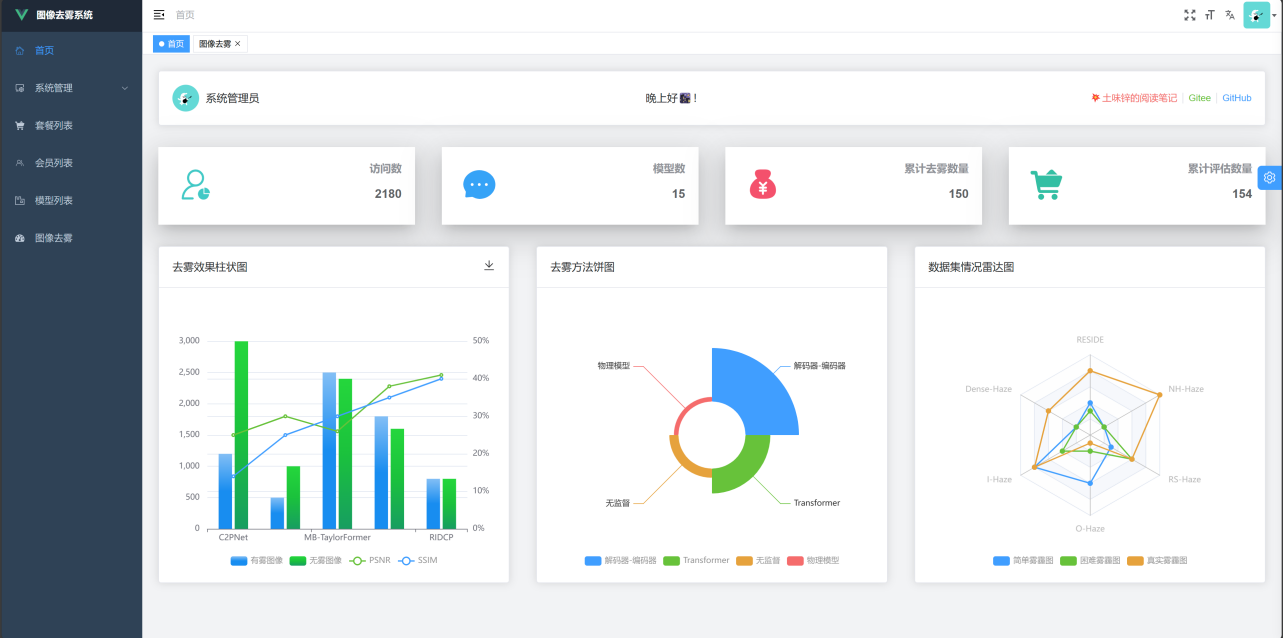
## 运行情况检查

# 用户手册

## 登录到系统

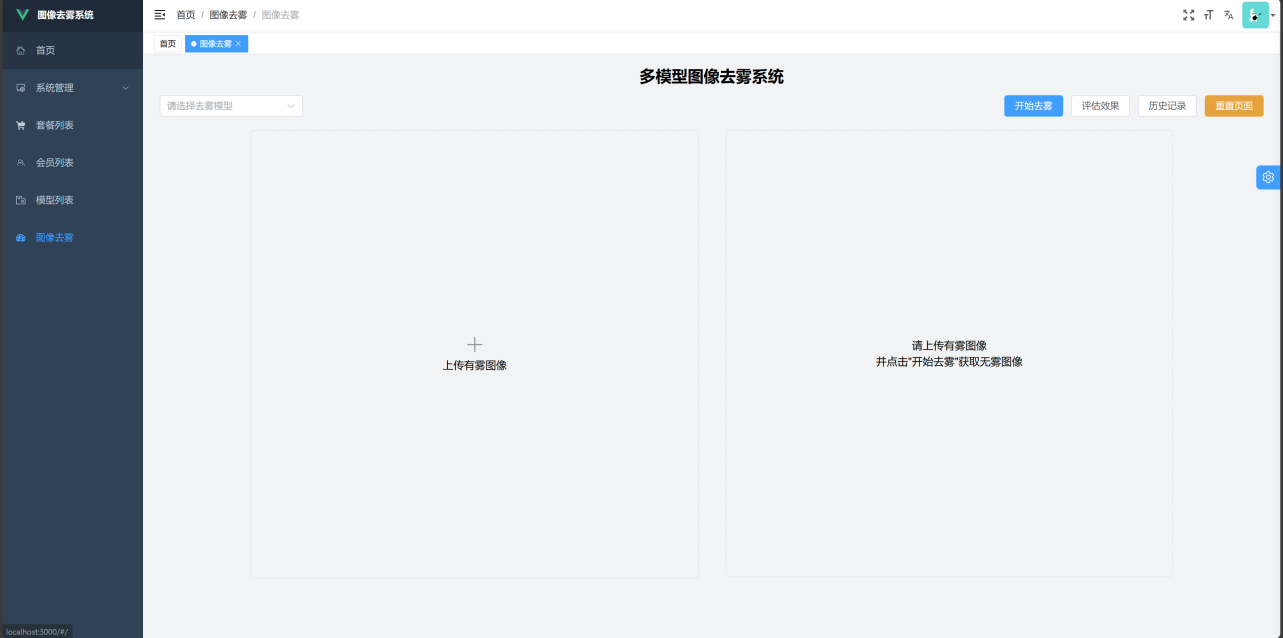
用户可以通过用户名和密码登陆进本系统  


## 系统首页

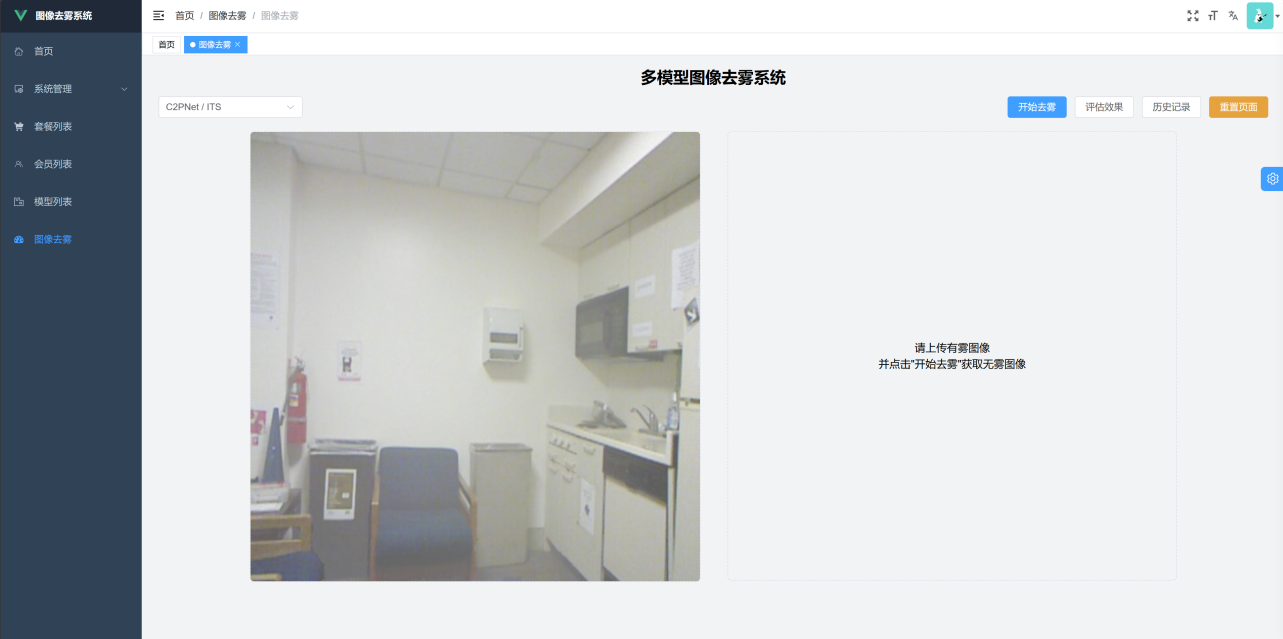
使用管理员账号登录后，系统默认会显示首页数据分析模块，展示了当前系统的总访问次数、去雾模型个数、系统累计进行图像去雾的张树、累计评估去雾效果的数量。以及可视化图标界面。  


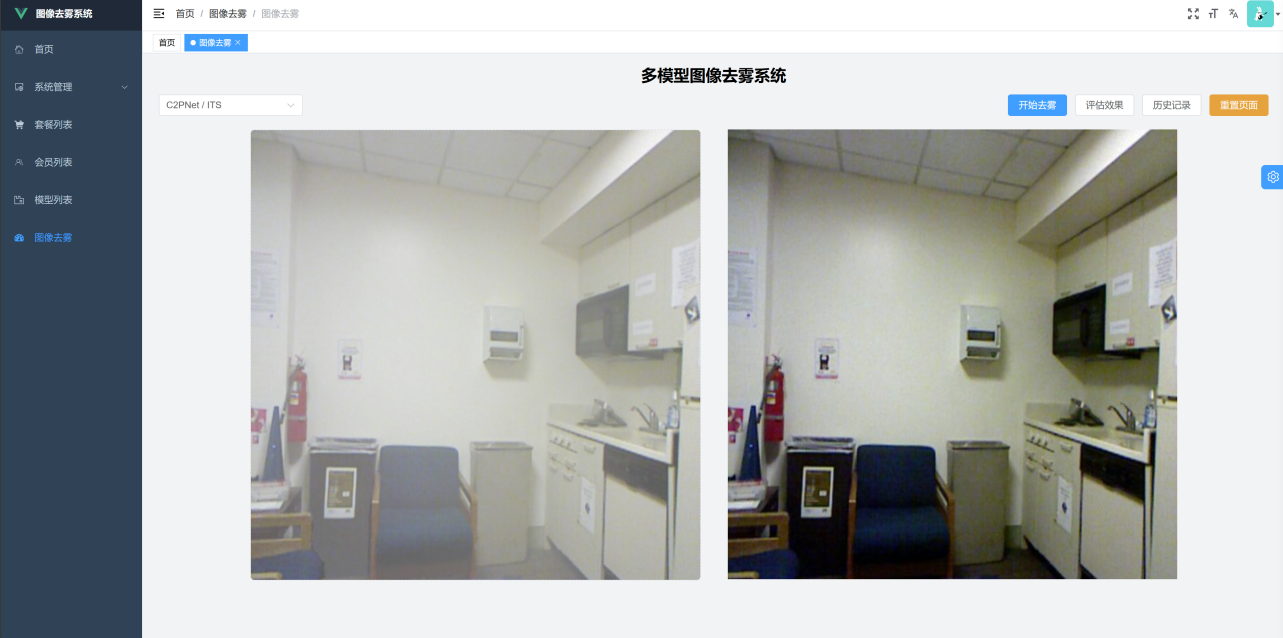
## 去雾模块

去雾模块是本系统的核心，任何已登录用户都可以进入去雾页面使用本系统进行去雾。但是不同的用户具有不同的去雾权益，详见-VIP及支付模块。

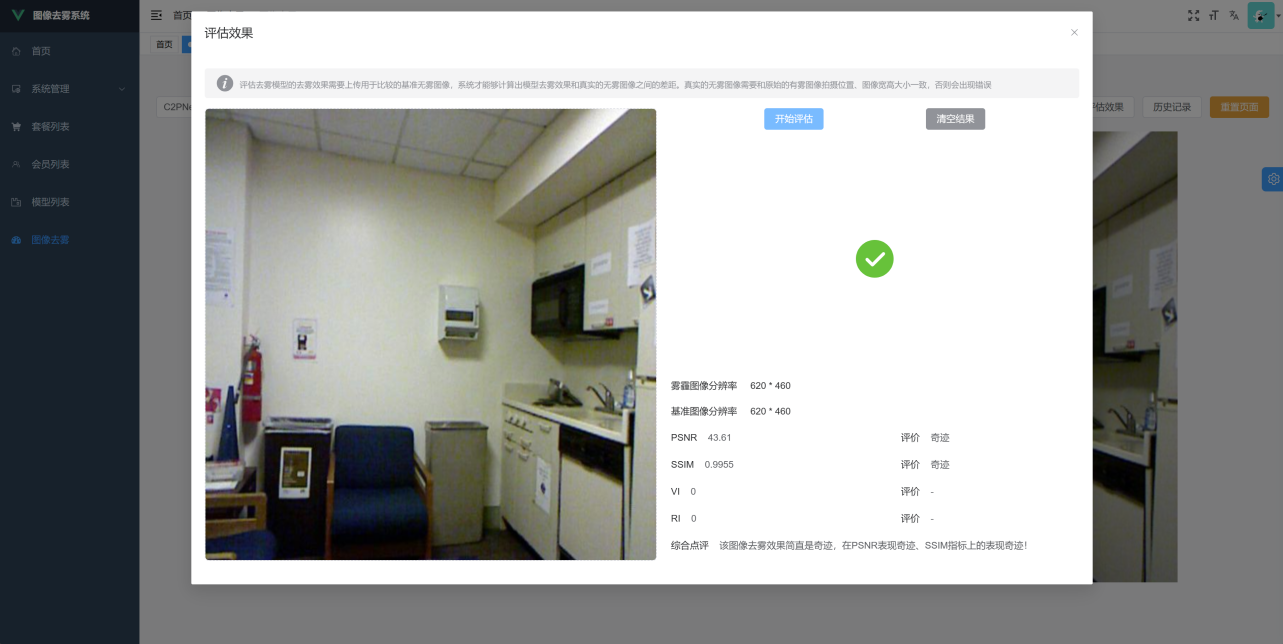


用户在该界面选择合适的去雾模型，然后点击左侧 + 号方框上传有雾图像，等到上传完成后，点击右上角蓝色“开始去雾”按钮去雾。去雾后的结果将展示在右侧的方框中，用户可以右键保存去雾后的图片。

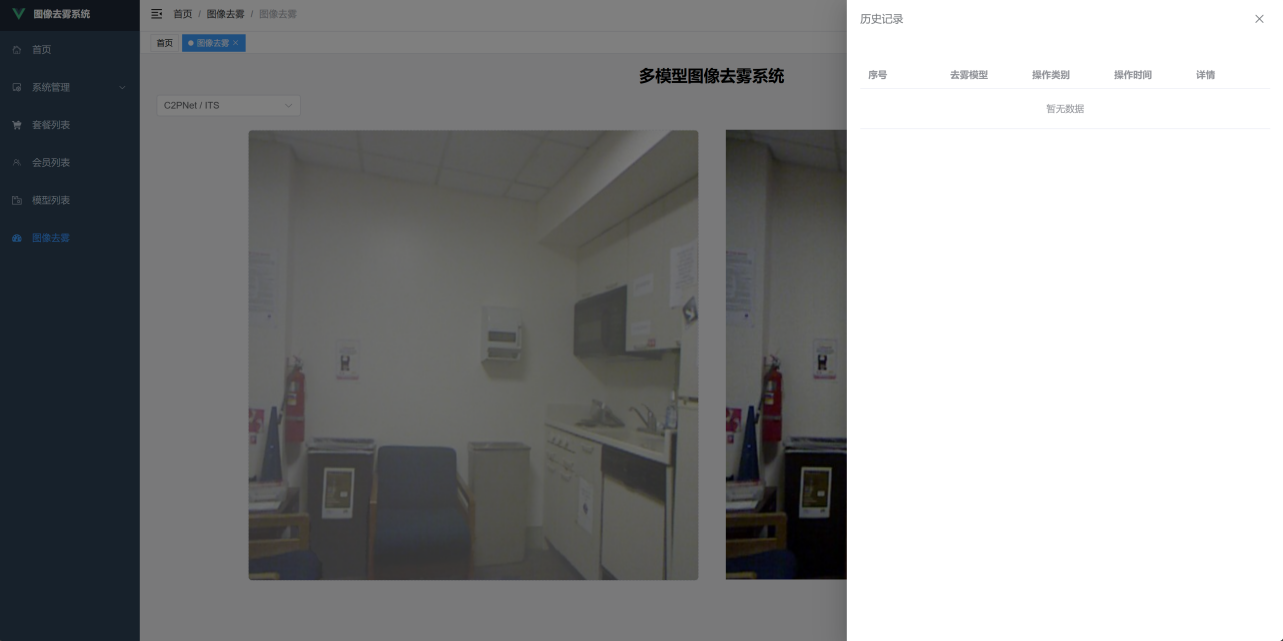




如果用户希望评估去雾效果，并且具有有雾图像对应的无雾基准图像，则可以点击“评估效果”按钮，打开新的对话框。在该对话框中上传基准无雾图像，然后点击开始评估。系统会计算出当前模型去雾后的图像和基准无雾图像之间的峰值信噪比（Peak Signal to Noise Ratio，PSNR）和结构相似度（Structural SIMilarity，SSIM）。并且做出评价。



用户在每次操作之后，都会在系统中留下历史记录，用户如果需要查看和获取之前的去雾结果，则可以点击“历史记录”按钮获取之前的操作记录。对于不同VIP级别的用户，系统保留的记录数量不定。



## VIP及支付模块

普通已登录用户使用本系统具有一定的限制，如下表所展示。但是用户可以通过付费购买VIP来解锁更多的权益。

用户可以通过主动购买VIP来获得对应的权益，如果当前用户去雾次数/评估次数已达上限，则会自动跳转到VIP购买页面。如果用户选择购买，则会跳转到支付界面进行支付，支付完成后，则可以继续使用本系统。

| 项目 | 普通用户 | VIP1 | VIP2 | SVIP |
| --- | --- | --- | --- | --- |
| 去雾次数 | 每月20次 | 每月100次 | 每月500次 | 每月3000次 |
| 评估次数 | 每月20次 | 每月100次 | 每月500次 | 每月3000次 |
| 历史记录保留条数 | 10条 | 50条 | 100条 | 500条 |
| 包月价格 | 免费 | 10元/月 | 20元/月 | 50元/月 |

更多VIP权益敬请期待

## 会员管理模块

系统管理员对于付费用户，系统专门开发了对应的管理模块，用来对会员用户进行管理。该模块可以展示会员的ID、名称、会员等级、剩余去雾次数、剩余评估次数、会员开通时间、会员到期时间等。方便对VIP用户针对性的管理。

## 模型管理模块

系统管理员可以通过模型管理来对系统存在的所有去雾模型进行浏览。去雾模型展示的信息有去雾模型的ID（唯一标识一个去雾模型）、模型名称（模型所属的神经网络名称）、针对方向（针对去除哪种类型的雾霾）、模型的参数量、模型大小、模型描述、添加时间、是否启用等信息。

## 套餐管理模块

针对不同的VIP套餐，系统设置了套餐管理模块。用来调整不同VIP的权益。系统最初数据库中是没有任何VIP套餐的，系统管理员可以通过手动添加套餐来实现VIP以及支付模块表格中对应的VIP权益。

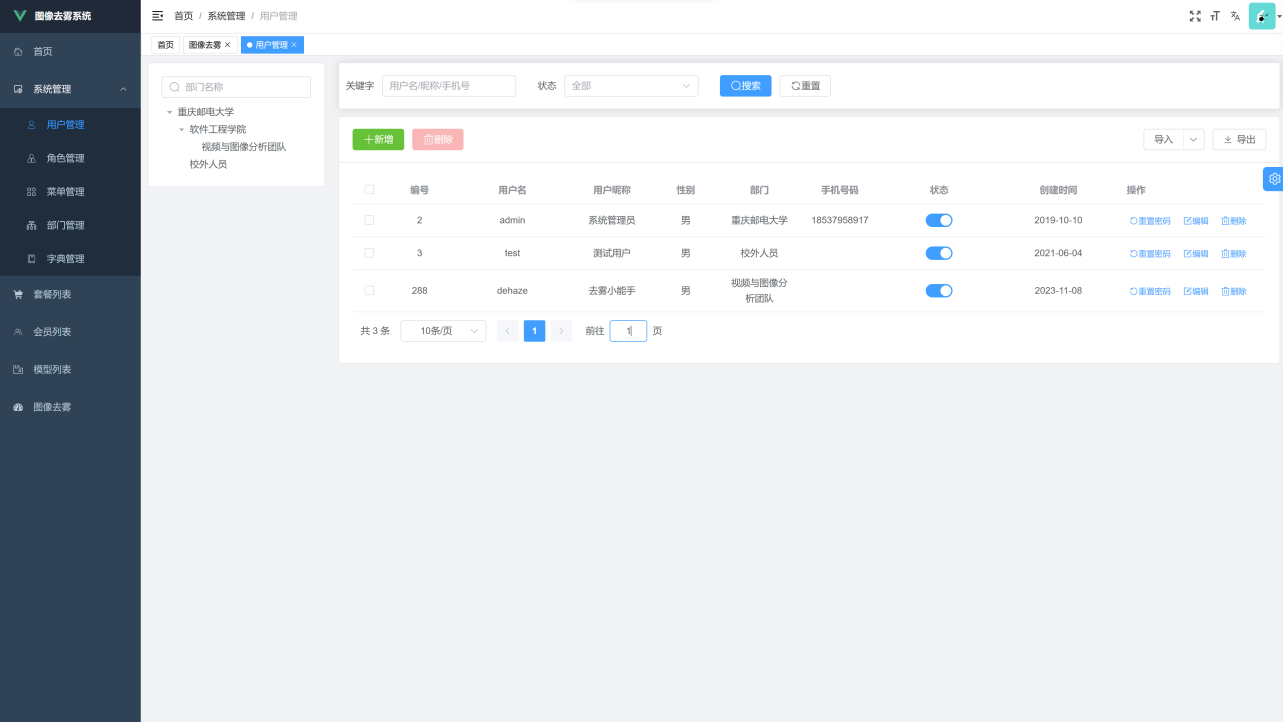
添加套餐，管理员可以通过输入套餐名（如VIP1）、套餐包含的去雾次数、评估次数、历史记录保留条数、包月价格来完成套餐的添加。同时管理员还可以删除套餐、启用停用套餐、查看套餐当前使用人数等。

## 订单管理模块

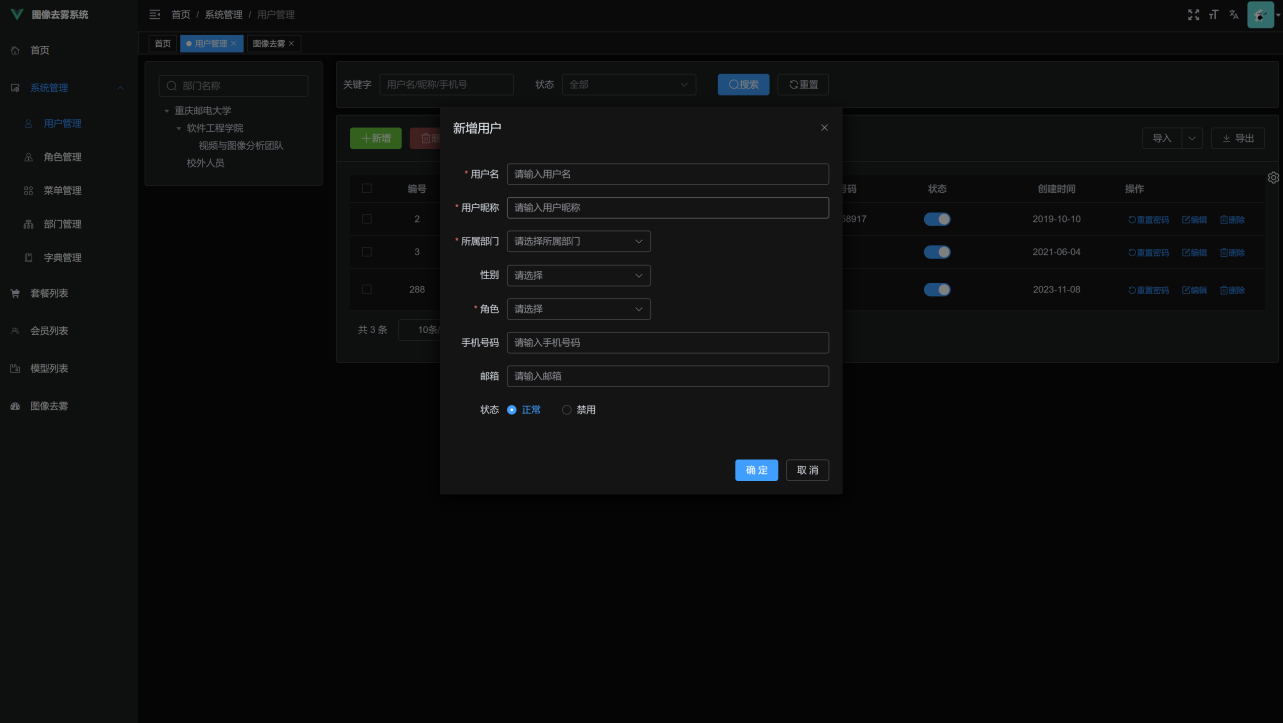
管理员功能。当用户购买对应的VIP套餐后，系统就会生成一个订单，显示了用户的购买情况。具体的内容有用户ID、用户昵称、购买套餐种类、订单状态、支付类型、订单创建时间、支付时间、购买时长等。

## 用户管理模块

系统管理员可以对当前系统中存在的用户进行管理，如启用/禁用某用户，删除某用户、重置用户密码、编辑用户信息等。

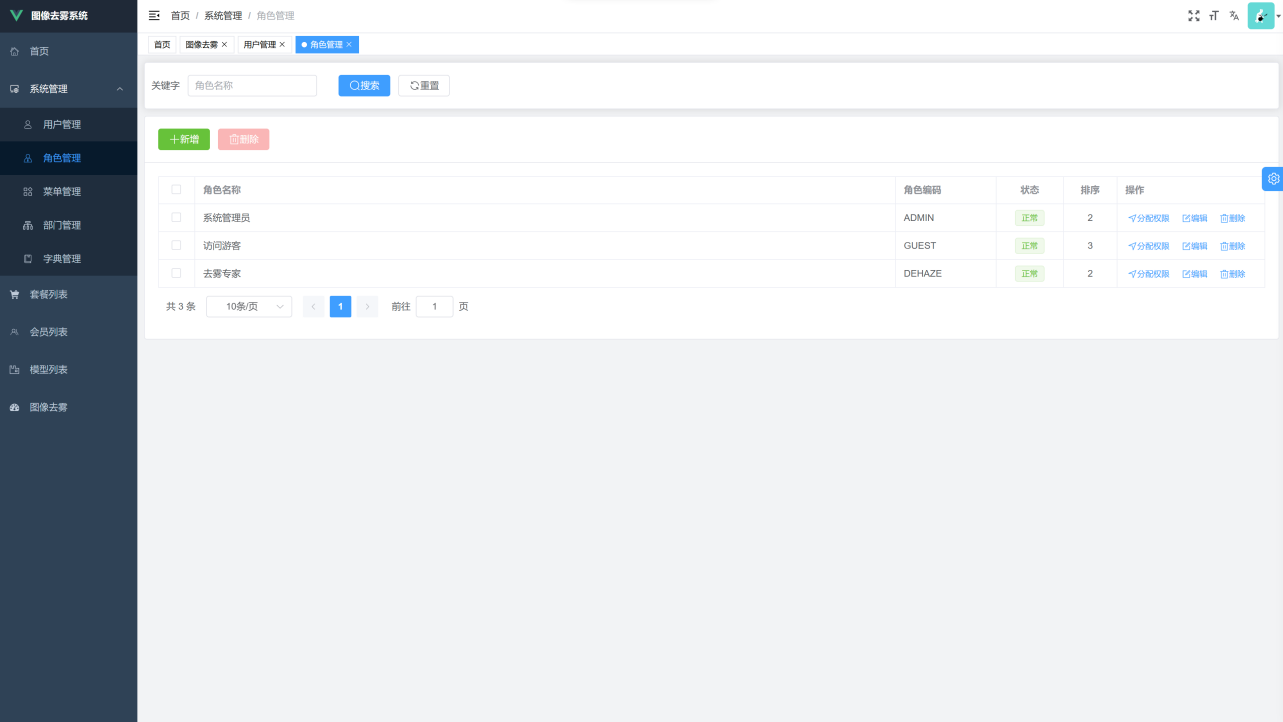


点击绿色按钮可以打开新增用户窗口

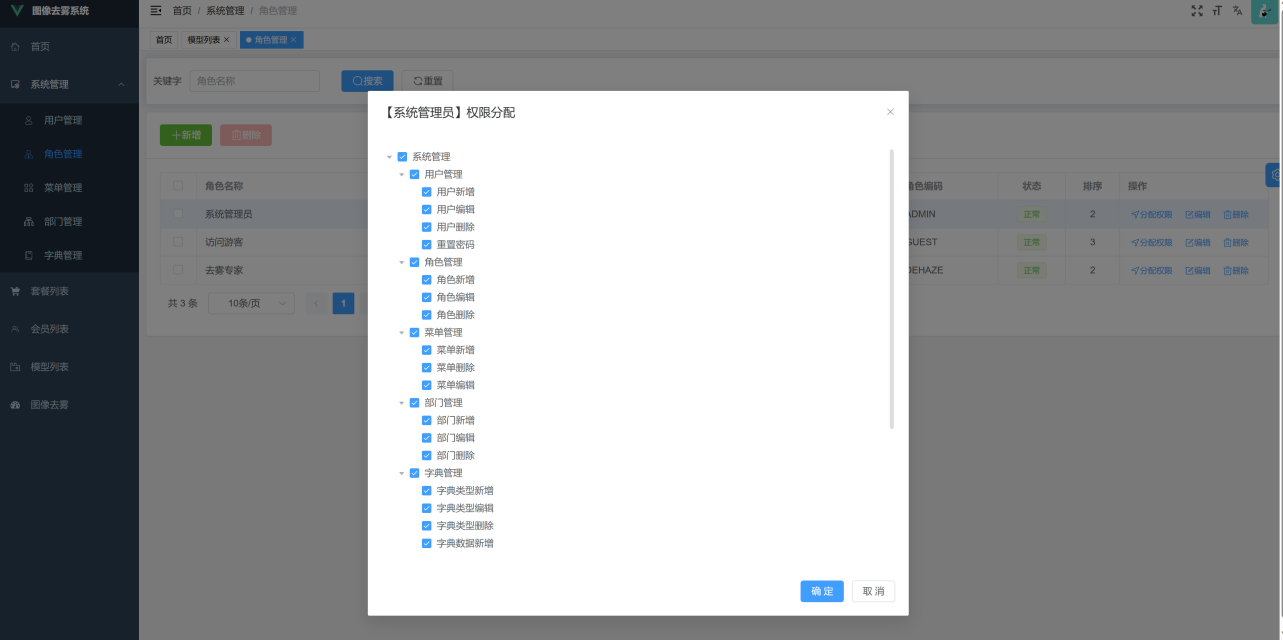


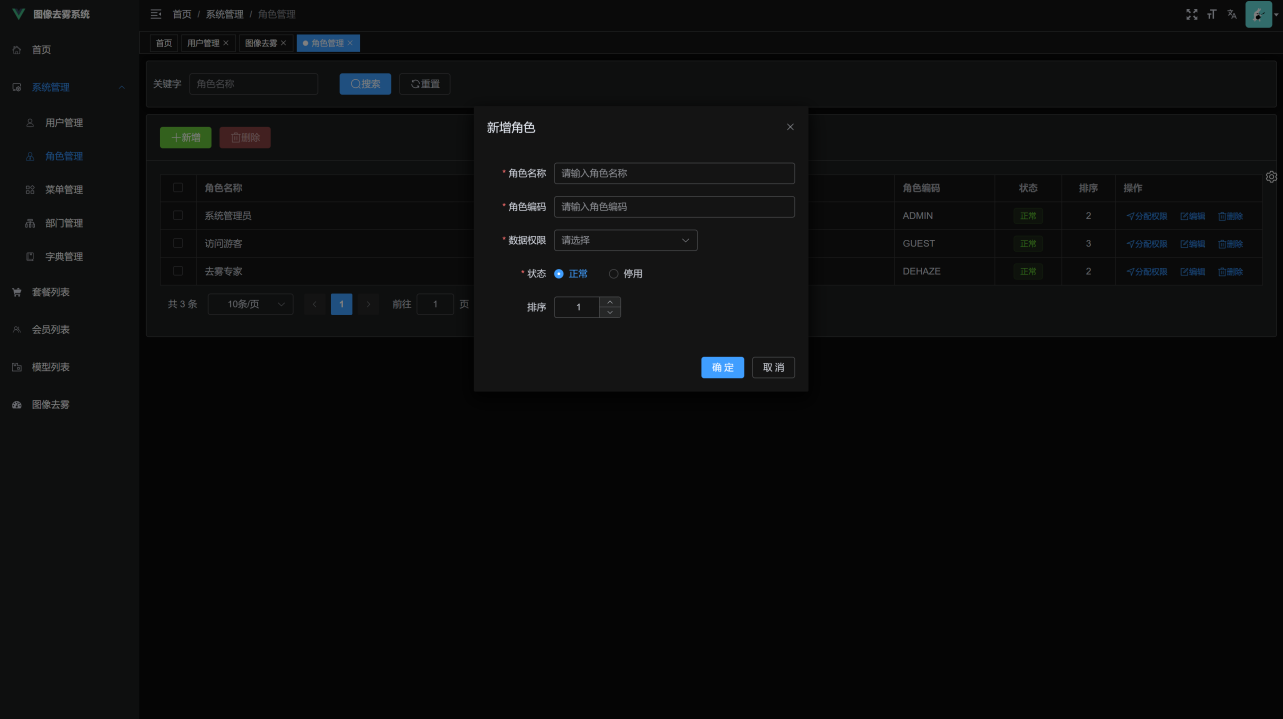
## 角色管理模块

系统管理员可以对当前系统存在的角色进行管理，不同的角色具有不同的权限。管理员可以使用角色管理来清晰划分每一位用户以及其他管理员应当具有的权限。避免越权操作。



针对某一具体角色，可以点击分配权限对该角色分配相应的权限，注意到角色编码为ROOT的超级管理员具有系统的任何权限。而其他角色则需要手动分配才能获取相应的权限。具有分配权限的用户则可以分配任意权限给任意用户，需要谨慎赋权。

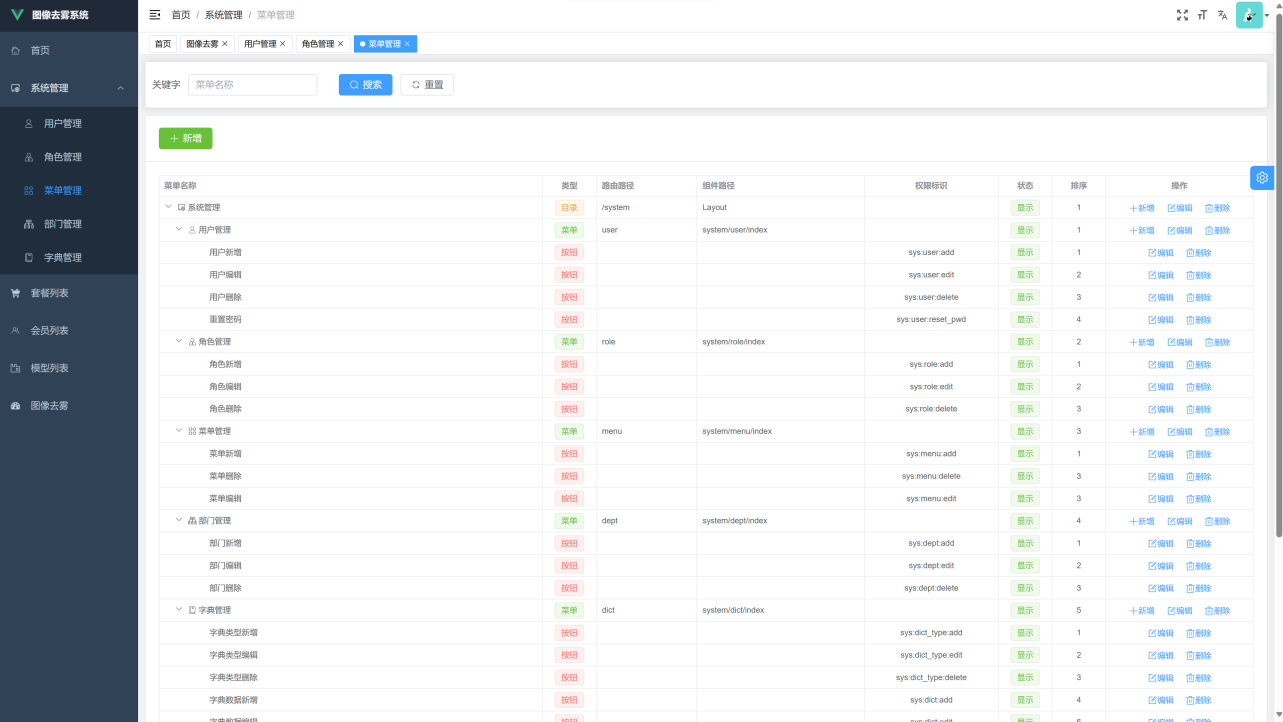


具有新增角色的管理员可以通过按钮新增角色，新增加的角色可以通过用户管理分配给某一具体用户，使某个用户具有该角色的所有权限。每个用户可以具有多个角色。用户的权限为多个角色权限之和。  


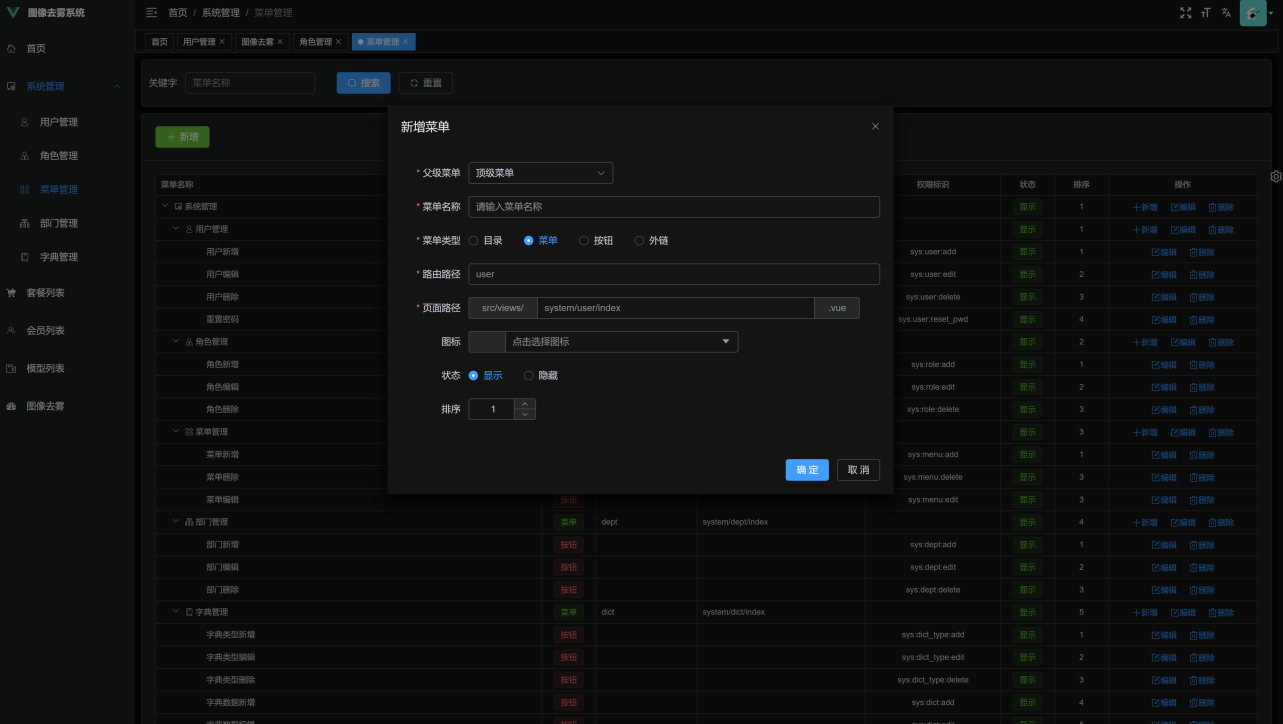
## 菜单管理模块

系统管理员可以对当前系统界面中的左侧菜单栏进行管理。前端的页面在写好之后，并不会自动添加进左侧的菜单栏中，而是需要通过新增菜单按钮来手动新增。菜单类型一共分为四种。目录、菜单、按钮、外链。

目录和菜单的关系类似于文件夹和文件的关系。目录将多个菜单聚合起来方便管理。菜单则指定了页面的路由地址、页面文件地址、控制左侧对应菜单栏的显示和隐藏等。按钮则是某个具体的菜单页面内部的某些按钮。可以通过添加按钮并赋予按钮权限来控制该按钮。实现不同用户、不同角色权限精细到按钮层面。外链则是系统外部的链接，可以通过添加外链再左侧菜单栏与外部系统整合，直接访问外部系统。

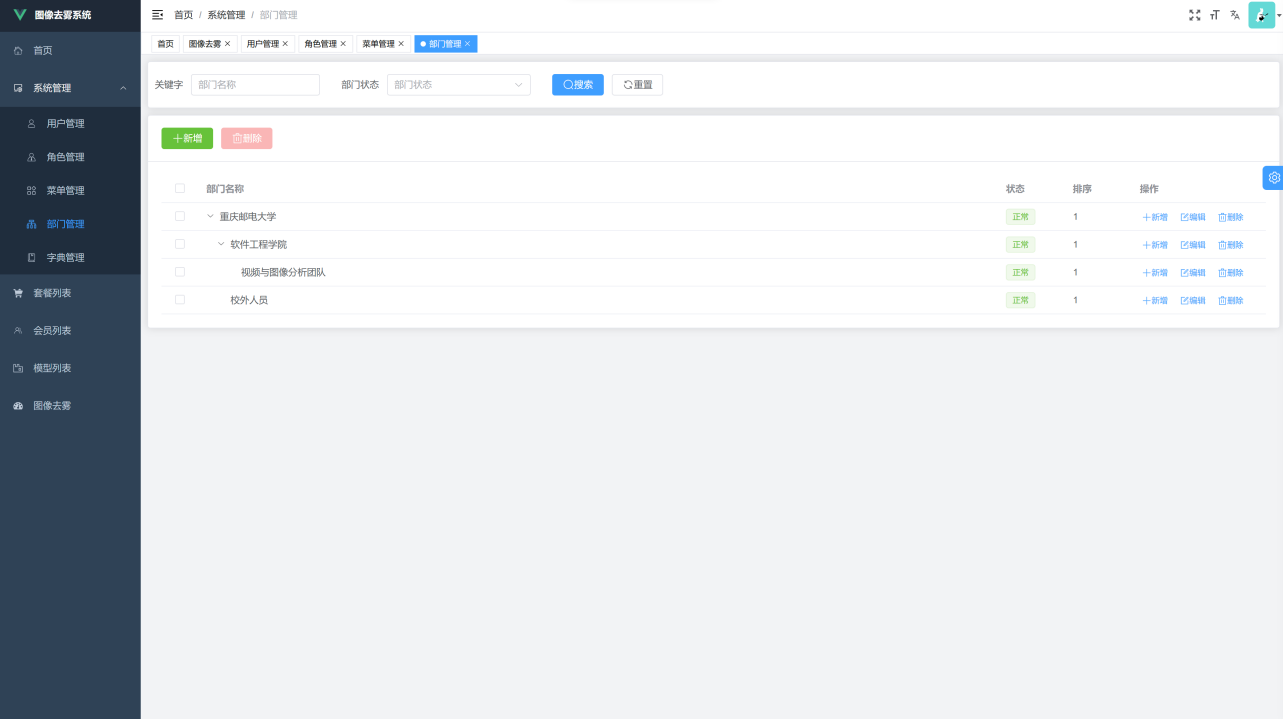


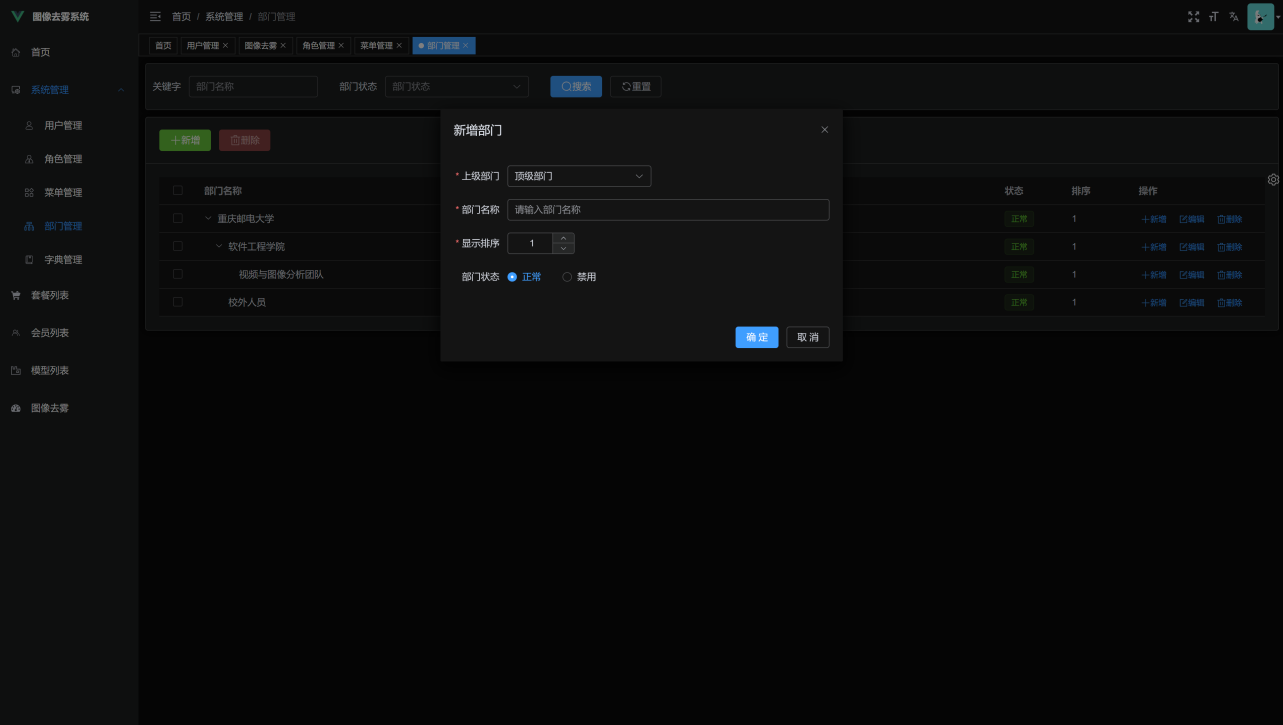
用户可以通过新增按钮新转增菜单，通过输入菜单名称、排序等操作新增。



## 部门管理模块

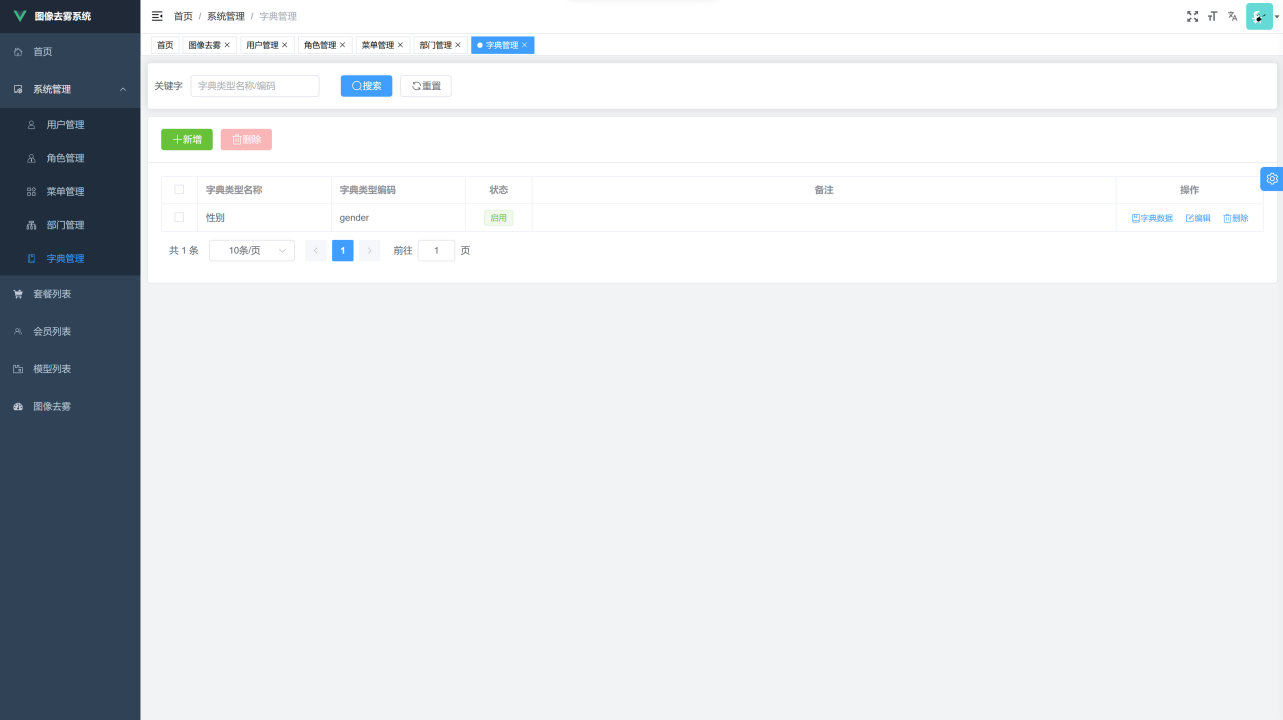
系统中所有用户都从属于某个部门，这里的部门可以是公司内部根据职责划分的不同的部门，也可以是使用系统的其他用户。通过部门管理可以很好的对不同的用户进行划分。



可以通过新增部门按钮来新增不同的部门  


## 字典管理模块

字典指的是系统内部设置的可选择项。通过字典来建立选择项和相应代码之间的映射管理，从而更好的实现数据存储。



## 系统界面以及设置

使用系统的所有用户都可以通过界面设置来调整自己的使用习惯。具体来说，可以调整为明亮模式或暗黑模式，也可以更换系统主题，调整系统页面布局。包括是否开启系统内部标签栏（Tags-View），是否固定头部标签。是否开启侧边菜单栏的Logo图像。

