**学号： 19402216**

**常 州 大 学**

**毕业设计（论文）**

**（ 2023 届）**

**题 目 基于VGG神经网络的花卉识别系统的设计与实现**

**学 生 高梦强**

**学 院 计算机与人工智能学院专 业 班 级 计算机195**

**校内指导教师 潘操 专业技术职务 副教授**

**校外指导老师 专业技术职务**

**二○二三年五月**

**基于VGG神经网络的花卉识别系统的设计与实现**

**摘要：**

花卉识别系统是一种常见的管理系统，它的主要目的是进行花卉识别并对用户信息进行管理。花卉管理系统提供了菜单管理模块，角色管理模块，用户管理模块，微信用户管理模块和取样记录模块，从而实现了花卉基于Web的识别查询与交互。通过花卉管理系统，管理员能够做到以用户为中心，简化繁琐的手工统计、信息汇总等大量的人工工作。

在生活中，花卉识别是一件繁杂而又耗时的工作，因为其中涉及到花卉多方面的信息，包括花卉基本形状以及花卉色彩等等。目前，集实用性、综合性于一体的花卉识别系统，实现实时在线、系统化、规范化的管理，大大提高了效率，在数据库安全性和系统维护方面，更是具有比较大的优势。

基于Web的花卉识别系统更好的迎合了现代信息化的发展需求，使得人们的生活变得更加的便利。适合了信息实时代的发展要求，跟上了信息技术潮流，在信息化的背景下，系统具有了更好的人机界面，方便的数据查询，方便的数据处理，同时实现了数据的稳定性与可靠性。如此的适应性、易用性、稳定性在现代化的发展过程中更是越来越多的受到大众的欢迎。花卉识别信息化的发展方便了对花卉相关情况的统一管理，为构建并实现信息化管理的发展起到了巨大的推动作用。

本系统采用当下流行的前后端分离方式开发，后端使用Spring Boot开发，前端使用Vue+ElementUI来构建单页面应用SPA。SPA通过动态重写当前页面与用户交互，而非传统的从服务器重新加载整个新页面，从而避免了页面之间切换打断用户体验，使应用程序更像一个桌面应用程序。一个后端可以对应多个前端，后端只负责提供数据，前后端的交互通过JSON数据完成。系统主要实现菜单管理，角色管理，用户管理，微信用户管理和取样记录管理功能。其中涉及到各个模块信息的录入、处理、多条件查询，通过设置管理员、用户两种权限，从而实现不同的管理功能。

本文从需求分析、概要设计、详细设计等几个方面详细的介绍了整个系统的设计与实现。该系统用户界面友好，操作流畅，功能较好，具有一定的应用价值。

**关键词：** SPA；VGG；模块化；花卉识别系统

**Design and Implementation of Student Records Management System Based on Web**

**Abstract:**

Flower recognition system is a common management system, whose main purpose is to identify flowers and manage user information. The flower management system provides a menu management module, role management module, user management module, WeChat user management module, and sampling record module, thus achieving web-based recognition, query, and interaction of flowers. Through the flower management system, administrators can achieve a user centric approach, simplifying tedious manual statistics, information aggregation, and other manual tasks.

In daily life, flower recognition is a complex and time-consuming task, as it involves various aspects of flower information, including the basic shape and color of flowers. At present, a flower recognition system that integrates practicality and comprehensiveness has achieved real-time online, systematic, and standardized management, greatly improving efficiency. It also has significant advantages in database security and system maintenance.

The web-based flower recognition system better caters to the development needs of modern information technology, making people's lives more convenient. Suitable for the development requirements of the information age and keeping up with the trend of information technology, in the context of informatization, the system has a better human-machine interface, convenient data query, convenient data processing, and achieved data stability and reliability. Such adaptability, ease of use, and stability are increasingly popular among the public in the process of modernization development. The development of flower recognition informatization has facilitated the unified management of flower related situations, playing a huge role in promoting the construction and implementation of information management.

This system is developed using the popular front-end and back-end separation method, with the backend developed using Spring Boot and the front-end using Vue+ElementUI to build a single page application SPA. SPA interacts with users by dynamically rewriting the current page, rather than traditional reloading the entire new page from the server, thereby avoiding page switching that interrupts the user experience and making the application more like a desktop application. A backend can correspond to multiple frontends, and the backend is only responsible for providing data. The interaction between the front and back ends is completed through JSON data. The system mainly implements menu management, role management, user management, WeChat user management, and sampling record management functions. It involves the input, processing, and multi condition query of information in various modules, and different management functions are achieved by setting administrator and user permissions.

This article provides a detailed introduction to the design and implementation of the entire system from several aspects such as requirement analysis, summary design, and detailed design. The system has a friendly user interface, smooth operation, good functionality, and certain application value.

**Keywords:**  SPA; VGG; Modularization; Flower recognition system

**目录**

[1 绪论 7](#_Toc135603677)

[**1.1** **课题背景** 7](#_Toc135603678)

[**1.2** **课题研究的现状** 7](#_Toc135603679)

[**1.3** **主要研究内容** 8](#_Toc135603680)

[2 需求分析和可行性分析 8](#_Toc135603681)

[**2.1系统需求分析** 8](#_Toc135603682)

[**2.1.1 功能需求分析** 8](#_Toc135603683)

[**2.1.2非功能性需求分析** 9](#_Toc135603684)

[**2.2可行性分析** 10](#_Toc135603685)

[**2.2.1技术可行性** 10](#_Toc135603686)

[**2.2.2经济可行性** 11](#_Toc135603687)

[**2.2.3操作可行性** 11](#_Toc135603688)

[**2.3 开发平台** 11](#_Toc135603689)

[3 系统概要设计 12](#_Toc135603690)

[**3.1 系统概述** 12](#_Toc135603691)

[**3.2 功能模块划分** 12](#_Toc135603692)

[**3.3 数据库设计** 13](#_Toc135603693)

[**3.3.1 逻辑设计** 13](#_Toc135603694)

[**3.3.2 表设计** 17](#_Toc135603695)

[4 详细设计与实现 19](#_Toc135603696)

[**4.1 核心技术简介** 19](#_Toc135603697)

[**4.1.1 SPA** 19](#_Toc135603698)

[**4.1.2 Spring Boot** 19](#_Toc135603699)

[**4.1.3 云对象存储** 20](#_Toc135603700)

[**4.1.4 Vue** 20](#_Toc135603701)

[**4.1.5 VGG** 20](#_Toc135603702)

[4.1.6 Flask 21](#_Toc135603703)

[**4.2 系统登陆模块** 21](#_Toc135603704)

[**4.2.1模块设计思路** 21](#_Toc135603705)

[**4.2.2 系统登录主页图** 22](#_Toc135603706)

[**4.3菜单管理模块** 23](#_Toc135603707)

[**4.3.1模块设计思路** 23](#_Toc135603708)

[**4.3.2菜单界面图** 25](#_Toc135603709)

[**4.4角色管理模块** 27](#_Toc135603710)

[**4.4.1模块设计思路** 27](#_Toc135603711)

[**4.4.2角色界面图** 31](#_Toc135603712)

[**4.5用户管理模块** 33](#_Toc135603713)

[**4.5.1模块设计思路** 33](#_Toc135603714)

[**4.5.2用户管理界面图** 39](#_Toc135603715)

[**4.6微信用户管理模块** 40](#_Toc135603716)

[**4.6.1模块设计思路** 40](#_Toc135603717)

[**4.5.2用户管理界面图** 44](#_Toc135603718)

[5 软件测试 46](#_Toc135603719)

[**5.1软件测试概述** 46](#_Toc135603720)

[**5.2测试用例** 46](#_Toc135603721)

[**5.3用户登录用例测试** 46](#_Toc135603722)

[5.4 用户权限测试 47](#_Toc135603723)

[6 总结与展望 49](#_Toc135603724)

[参 考 文 献 50](#_Toc135603725)

[致 谢 51](#_Toc135603726)

# **1 绪论**

## **课****题背景**

随着Internet的发展，改变了人们的生活模式，通过网络可以帮助人们解决很多问题。在生活中，为了方便人们对自然中的陌生花草有个更好的了解，基于Web的花卉识别系统发挥着越来越重要的作用，给人们的生活带来了方便。该系统的使用将改变以往的书本查询模式，使得繁琐的信息资料通过电子的形式得以充分的存储与修改，大大提高了人们的工作效率，同时也迎合了信息化时代的发展趋势。广义的花卉可指一切具有观赏价值的植物（或人工栽插的盆景），而狭义上则单指所有的开花植物的生殖器官。花卉由于常具有各种诱人的形状、美丽的颜色、芬芳的气味，而一直广受人们的喜爱和使用，主要用于观赏，还能当食材或提炼原料。在现代社会，人们经常因为鲜花宜人的外观和香味而以各种方式种植、购买和佩戴花卉在世界各地，人的一生中都会在各种各样的活动和场合中使用鲜花。在生活中，花卉识别是一件繁杂而又耗时的工作，因为其中涉及到花卉多方面的信息，包括花卉基本形状以及花卉色彩等等。目前，集实用性、综合性于一体的花卉识别系统，实现实时在线、系统化、规范化的管理，大大提高了效率，在数据库安全性和系统维护方面，更是具有比较大的优势。基于Web的花卉识别系统更好的迎合了现代信息化的发展需求，使得人们的生活变得更加的便利。适合了信息实时代的发展要求，跟上了信息技术潮流，在信息化的背景下，系统具有了更好的人机界面，方便的数据查询，方便的数据处理，同时实现了数据的稳定性与可靠性。如此的适应性、易用性、稳定性在现代化的发展过程中更是越来越多的受到大众的欢迎。花卉识别信息化的发展方便了对花卉相关情况的统一管理，为构建并实现信息化管理的发展起到了巨大的推动作用。

## **课题研究的现状**

花卉视觉是计算机视觉的一项现实应用。目前计算机视觉行业增速持续保持较高的增长趋势，深度学习和卷积神经网络技术的推动着计算机视觉技术，同时也驱动着整个人工智能行业的迅速发展，计算机视觉技术已在各行各业有着举足轻重的作用。根据Tractica预测，到2025年，全球计算机视觉市场规模（包括软件、硬件和服务收入）将从2016年的11亿美元增长到262亿美元。近年来，巨量数据的不断涌现与计算能力的快速提升，给以非结构化视觉数据为研究对象的计算机视觉带来了巨大的发展机遇与挑战性难题，计算机视觉也因此成为学术界和工业界公认的前瞻性研究领域，部分研究成果已实际应用，催生出人脸识别、智能视频监控等多个极具显示度的商业化应用。

计算机视觉的研究目标是使计算机具备人类的视觉能力，能看懂图像内容、理解动态场景，期望计算机能自动提取图像、视频等视觉数据中蕴含的层次化语义概念及多语义概念间的时空关联等。自 2012 年以来，计算机视觉领域不断涌现出很多激动人心的研究成果，例如，人脸识别、物体识别与分类等方面的性能已接近甚至超过人类视觉系统。因此，可以说计算机视觉当前发展已进入了一个新的阶段。目前，提供花卉识别的服务包括百度AI开放平台，可识别超过2万种常见植物和近8千种花卉，接口返回植物名称和置信度信息且支持获取识别结果的百科信息，返回百科词条URL、图片和描述，可自定义返回词条数。

## **主要研究内容**

花卉识别着力于提高识别准确性及识别种类多样性。花卉识别是计算机视觉的现实应用。计算机视觉是人工智能 (AI) 的一个领域，是指让计算机和系统能够从图像、视频和其他视觉输入中获取有意义的信息，并根据该信息采取行动或提供建议。计算机视觉需要大量数据。 它一遍又一遍地运行数据分析，直到能够辨别差异并最终识别图像为止。这个过程会用到两种关键技术：一种是机器学习，叫做 深度学习，另一种是卷积神经网络 (CNN)。机器学习使用算法模型，让计算机能够自行学习视觉数据的上下文。 如果通过模型馈入足够多的数据，计算机就能"查看"数据并通过自学掌握分辨图像的能力。 算法赋予机器自学的能力，而无需人类编程来使计算机能够识别图像。CNN 将图像分解为像素，并为像素指定标记或标签，从而使机器学习或深度学习模型能够"看"到物体。 它使用标签来执行卷积运算（用两个函数产生第三个函数的数学运算）并预测它"看到"的东西。 该神经网络运行卷积运算，并通过一系列迭代检验预测准确度，直到预测开始接近事实。 然后它以类似于人类的方式识别或查看图像。基于VGG的花卉识别管理系统主要围绕用户信息、识别记录等问题进行展开，方便用户进行多条件查询和处理基本信息。

# **2 需求分析和可行性分析**

## **2.1系统需求分析**

需求分析也称为软件需求分析、系统需求分析或需求分析工程等，是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。需求分析是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节，该阶段是分析系统在功能上需要“实现什么”，而不是考虑如何去“实现”。需求分析的目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作。此外，软件的一些非功能性需求(如软件性能、可靠性、响应时间、可扩展性等)，软件设计的约束条件，运行时与其他软件的关系等也是软件需求分析的目标 需求分析的内容是针对待开发软件提供完整、清晰、具体的要求，确定软件必须实现哪些任务。具体分为功能性需求、非功能性需求与设计约束三个方面。[]。

基于VGG的花卉识别管理系统是在数据的快速查询、检索需要以及管理科学化的基础上做出的系统需求分析，达到功能完善、操作便捷、及时性好、适用面高的系统需求，从而满足用户对于花卉信息的录入、检索和处理的便捷化需求。

### **2.1.1 功能需求分析**

功能性需求即软件必须完成哪些事，必须实现哪些功能，以及为了向其用户提供有用的功能所需执行的动作。功能性需求是软件需求的主体。开发人员需要亲自与用户进行交流，核实用户需求，从软件帮助用户完成事务的角度上充分描述外部行为，形成软件需求规格说明书。

花卉识别管理系统主要提供用户查询信息的便利，方便对网上信息浏览进行管理，系统总体任务是让用户及查询信息变得系统化、自动化、规范化，主要功能包括：

用户、管理员、角色、和菜单信息的录入、修改和查询操作，其中主要涉及到用户基本信息、管理员基本信息、菜单基本信息、角色等内容。

本系统的功能需求根据使用者的不同而有所不同，主要有两种用户：普通用户、系统管理员。为了便于整个系统的组织管理，把整个用户享有的不同功能模块分为两大块：普通用户模块和管理员模块。对于普通用户模块来说，有浏览属于本用户的一些基本信息、菜单信息和角色信息，但却不可以对个人的基本信息进行修改。而管理员模块，则可以录入、修改、删除、查询所有普通用户和系统管理员的基本信息，可以对菜单情况和角色情况进行增、删、改、查询操作，还可以查看所有用户的查询情况，同时，对于显示出的报表信息还具备打印功能。

本系统可以提供相应的多条件查询功能，可以根据用户昵称、ID等关键信息进行查询，具备处理基本数据的能力。此外，还可以维护用户的基本信息，包括对管理员、普通用户密码进行管理，以达到系统运行的安全性。

### **2.1.2非功能性需求分析**

作为对功能性需求的补充，软件需求分析的内容中还应该包括一些非功能需求。主要包括软件使用时对性能方面的要求、运行环境要求。软件设计必须遵循的相关标准、规范、用户界面设计的具体细节、未来可能的扩充方案等。非功能性需求是系统必须具备的，通常并不改变系统的功能，它们可以让系统更有吸引力，实现快速、可靠、安全的性能，非功能性需求包括界面需求，易用性需求、操作需求、可维护性需求、安全性需求：

（1）界面需求：主要描述了对产品外观的期望、情绪和风格。这些需求规定了外观想要达到的目标，它和详细的界面设计还是有区别的，体现的是客户的感觉。界面需求还包括对控件进行规范和对控件的使用范围进行一个规定等方面的内容。可以考虑借用一个原型来描述。用户在使用系统过程中通过浏览系统外观获得对系统的第一感觉，体现的是用户的隐形需求，这些需求规定了外观需要达到的目标，作为学生成绩管理系统，它的界面设计应该要让用户觉得美观、简洁、清晰。

（2）易用性需求：易用性会使产品提高符合用户习惯的能力以及其对使用的期望。它会对用户使用产品的生产效率、错误率以及用户对新产品的接收程度产生很大的影响。它会对系统的处理效率以及用户对系统的适应程度产生影响，要尽量让各个功能模块显得合理有序，使用户容易适应各功能的操作，让用户易于操作。

（3）操作需求：是指产品可以在给定的时间或者特定的精确度来执行某些任务，或者在一段时间内的极端状态值。在考虑执行需求时，可以从完成任务的速度、结果的精确度、容量、允许值的范围、单位时间内完成的任务数、资源的使用效率、两次故障间的平均无故障时间、连续不停机时间等方面入手。它还应该包括对风险的控制内容

（4）可维护性需求：开发出来的程序需要有灵活性强的特点，并且便于后期的修改和维护。

（5）安全性需求：安全性指产品消除潜在风险的能力和对风险的承受能力。包含、保密性、可靠性和完整性三个子特性。保密性指的是数据不能被授权用户以外的任何人访问的能力。可靠性指的是授权用户可以不受阻止的访问数据、与其它软件的兼容的能力和产品的强壮度。完整性指的是安预期目标完成任务的能力。安全性指系统可以消除潜在风险和拥有抵抗风险的能力，具备不同管理权限，不同的用户享受不同查阅、处理成绩的权利。

## **2.2可行性分析**

可行性分析即可行性研究，是在进行系统需求分析的基础上，根据系统开发的必要性，对系统进行的技术、经济、操作方面的可行性研究，对系统进行抽象的分析和处理，从而能够用最小的代价保证问题的最终解决[]，提高软件的开发、管理效率。进行可行性分析才能明确系统是否值得做，避免投资损失，即能否带来经济效益、企业效益或社会效益。对于花卉识别管理系统，下面进行了具体的可行性论证。

### **2.2.1技术可行性**

主要考虑：在给定的限制范围内，能否设计出系统，并实现必须的功能和性能？是否有充足的熟练技术人员可以支配？其他必要的资源（软件和硬件）对建造系统可用么？相关的技术条件是否能够支持系统的开发？技术可行性是可行性分析中比较重要的环节，根据需求分析，从技术上进行划分，可以分为对普通用户、管理员信息等各个模块的增加、删除、修改、查询操作，通过不断的细分，实现各个代码语句对于各个功能的实现，从而构建出一个完善的系统功能。

VGG花卉识别系统采用前后端分离技术。在后端方面使用Spring Boot技术，通过自动配置功能，降低了复杂性，同时支持基于JVM的多种开源框架，可以缩短开发时间，使开发更加简单和高效。后端使用Vue.JS，Vue是渐进式JavaScript框架，易学易用，性能出色，适用场景丰富。经过编译器优化、完全响应式的渲染系统，几乎不需要手动优化。数据库使用MariaDB，数据缓存使用Redis并在Spring Boot中集成MyBatis以简化数据库操作。模型训练方面，采用谷歌花卉数据集，使用VGG16训练模型，通过Tensor Flow的keras库获取参数，并通过Flask库搭建起识别服务器。硬件方面，使用8GB内存、Nivdia MX150GPU，即Tensor Flow中采用CPU而非GPU进行训练。

综上所述，学生成绩管理系统的设计和开发在技术和硬件上都得到满足，所以它是具有技术可行性的。

### **2.2.2经济可行性**

如今，计算机价格已经在性能优越的前提下保持着合理的趋势，对于本系统的开发，它的经济成本是可以接受的，同时它的成功实施可以帮助用户实现信息化管理，提高花卉识别管理的工作效率[11-12]。体现在：

（1）减轻用户整理数据的繁重性，通过强大的数据库管理系统可轻松的进行存储和查阅。

（2）方便用户和管理及时的录入花卉查询信息，减轻繁重的信息负担。

（3）便于用户随时随地都可查阅自己的基本信息，以及查询记录情况。

所以，整个系统带来的经济隐形效益也是该系统的实用性所在，它是经济可行的。

### **2.2.3操作可行性**

对于花卉识别管理系统的运行，无论是管理员还是普通用户，它的操作都是简单易行的，该系统功能明确，享有查询、修改、删除等操作，对于不同用户进入该系统，拥有的权限也不同，在使用方面，它简单易学，能够使用户短期内明确各个模块的功能作用，从而方便对信息的及时查阅与处理。

对于整个系统在操作方面是很容易的，所以它的操作性是可行的。

## **2.3 开发平台**

根据功能需求分析，花卉识别管理系统采用的开发平台如下：

（1）开发环境：Manjaro Linux 22

（2）开发工具：IDEA，JDK8

（3）包管理器：Anaconda、Node.js

（4）数据库：Maria DB、Redis

（5）小程序：微信开发者工具、HBuildX

# **3 系统概要设计**

## **3.1 系统概述**

花卉识别管理系统设计的目的在于更好的方便用户及管理员对于信息的及时查阅与处理，该系统主要采用软件工程的设计思想进行设计，完成预定的任务。首先要从需求分析开始，构建系统功能的各个合理方案，选择最佳方案，进行结构设计，之后确定模块组成以及模块之间的调用关系。同时要注意设计时要遵循模块独立性，即软件由相对独立的子模块构成，设计尽可能简单的接口。概要设计是一个设计师根据用户交互过程和用户需求来形成交互框架和视觉框架的过程，其结果往往以反映交互控件布置、界面元素分组以及界面整体板式的页面框架图的形式来呈现。这是一个在用户研究和设计之间架起桥梁，使用户研究和设计无缝结合，将对用户目标与需求转换成具体界面设计解决方案的重要阶段。概要设计的主要任务是把需求分析得到的系统扩展用例图转换为软件结构和数据结构。概要设计建立的是目标系统的逻辑模型。软件结构设计的具体任务是将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。

## [**3.2 功能模块划分**](#_Toc233179546)

根据系统需求分析，从功能上来看，主要有如下五个模块构成，包括用户管理模块、角色管理模块、菜单管理模块、取样管理模块、微信用户管理模块：

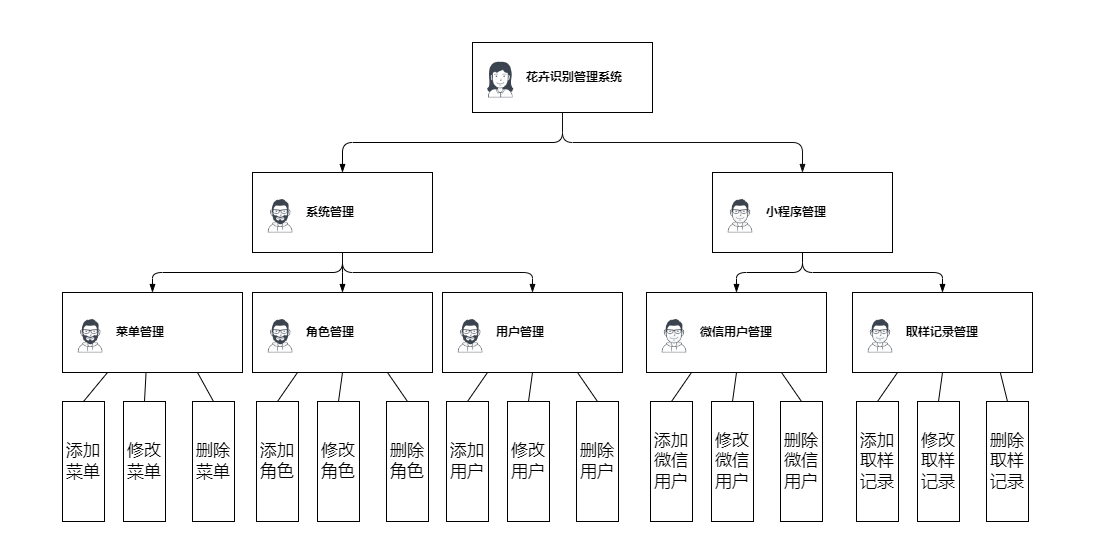
（1）用户管理模块：此模块主要包括新增用户、用户删除和修改搜索功能。用户分为两种类型，包括管理员、普通用户。用户包含ID、用户名、昵称、状态、创建时间以及操作等属性。普通用户对用户数据只有简单的查看功能。而管理员则比普通的用户有更高的权限，包括增删改查属性。如新增、删除、修改、查看菜单。用户权限的分离保障了系统的稳定性与安全性。

（2）菜单管理模块：此功能模块方便管理员和用户本身对菜单信息进行管理操作。菜单管理模块包含了对各模块操作的记录。包含菜单名称、图标、类型、排序、权限表示、路由地址、组件地址、状态、创建时间以及属性等字段。

（3）取样记录管理模块：此功能模块方便管理员和用户本身对用户在微信小程序端进行的识别记录进行管理操作。取样记录基本信息包括：ID、记录名、备注、图片、取样结果、经度、纬度、创建者、状态、创建时间以及操作。其中管理员和用户均可以通过对ID、记录名、备注的输入，实现取样记录信息的多条件查询，对于检索出来的信息具有打印功能。

（4）角色管理模块：此功能模块方便管理员和普通用户对相对应的角色进行管理，可以录入，更新信息，调整所在角色及其详细信息。通过对ID、记录名、备注，可以实现多条件查询。显示出的角色信息包括ID、记录名、备注、权限。此外还将显示出具体的各个角色信息，从而实现对角色信息的查询与处理，对于检索出来的信息具有打印功能。

（5）微信用户管理模块：此功能模块方便管理员及普通用户录入、更新微信用户的信息，通过对ID、用户名、名称的输入，可以实现学生成绩的多条件查询。。管理员可以查看所有班级所有微信用户信息，对于检索出来的信息具有打印功能。

****

**图3.1 系统功能图**

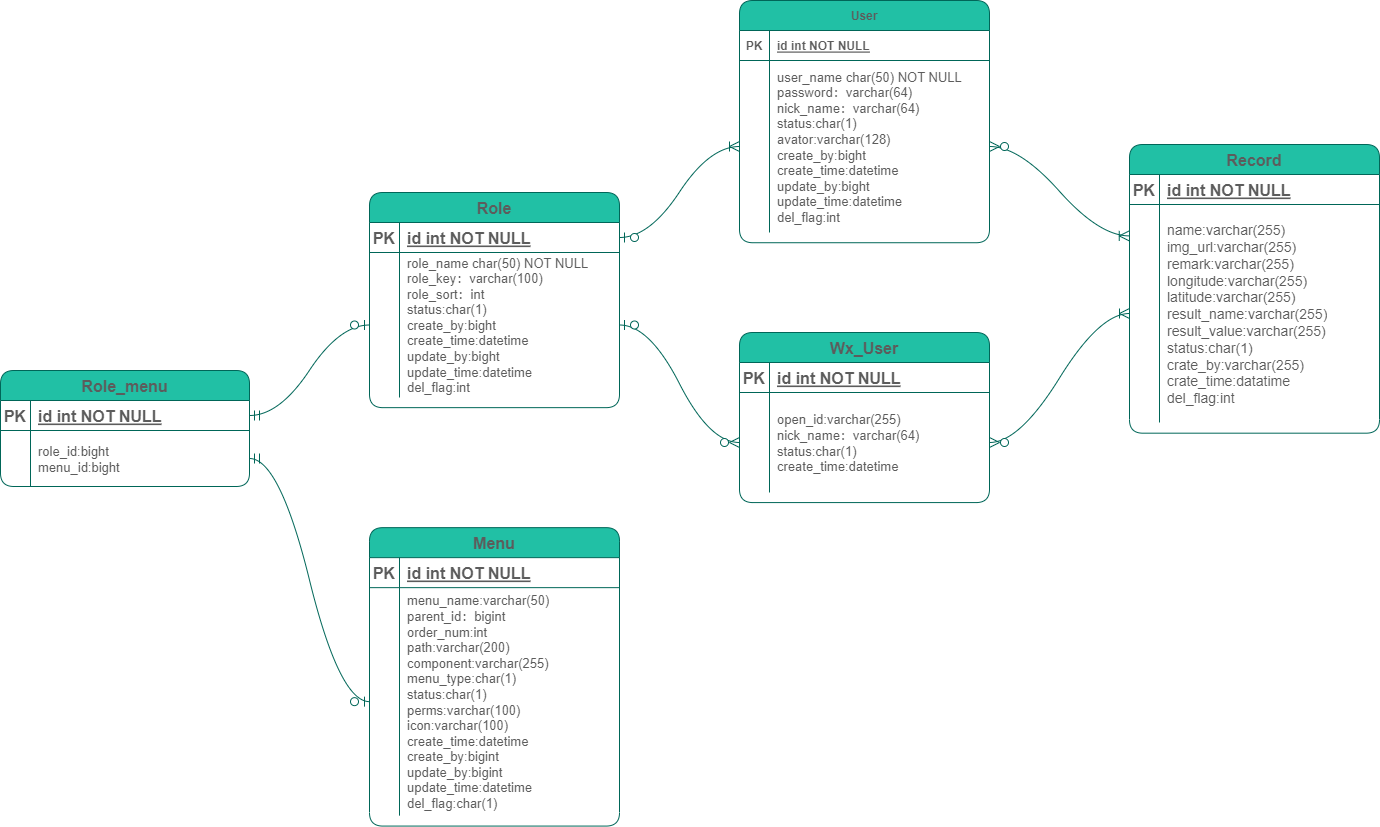
综上所述，花卉识别管理系统需要对五个功能模块对其进行合理的划分，来完成整体整个系统的运行，从而保证系统的稳定、高效运转，以下是整个花卉识别管理系统功能的图示，如图3.1所示。

## **3.3 数据库设计**

数据库设计是根据用户的需求，在某一具体的数据库管理系统上，设计数据库结构和建立数据库的过程。当数据库比较复杂（如数据量大，表较多，业务关系复杂）时，需要先设计数据库，因为良好的数据库设计可以节省数据的存储空间，保证数据的完整性，且方便进行数据库应用系统的开发。数据库设计就是根据业务系统的具体需求，结合我们所选用的数据库，建立好表结构及表与表之间的管理关系，为这个业务系统构造出最优秀的数据存储模型的过程。使之能有效的对应用的数据进行存储，并高效的对已经存储的数据进行访问。数据库设计是数据库系统中的重要组成部分。一个良好的数据库可以给系统带来清晰的数据统计与数据的详细分析，给系统带来方便直观的数据。不良的数据库设计，必然会造成很多问题，轻则增减字段，重则系统无法运行。数据库设计分为六个阶段：需求分析、概要设计、逻辑结构设计、物理设计、数据库实施以及数据库运行维护。因此，进行合理的数据库设计显得非常重要。

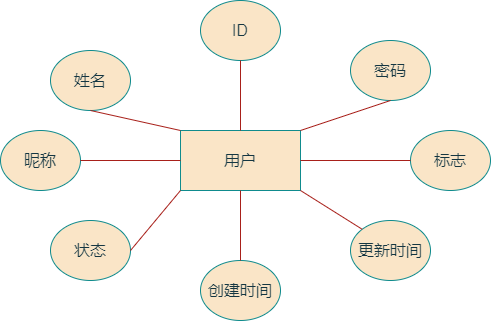
### **3.3.1 逻辑设计**

数据库逻辑结构设计即将 E-R 图转换为多张表，进行逻辑设计，确认各表的主外键，并应用数据库设计的三大范式进行审核，对其优化。根据系统功能需求分析，本系统由用户、角色、菜单、记录和微信用户五个实体构成。其中用户和记录是多对多的关系，微信用户和记录也是多对多的关系，教师和课程也是一对多的关系，角色和用户是一对多的关系，角色和微信用户是一对多的关系，角色菜单和角色是一对一的关系，角色菜单和菜单是一对一的关系。为了清楚的表示各个实体及之间的联系，下面采用E-R图来形象的表示：

****

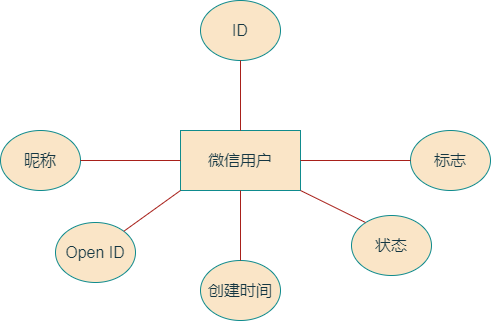
**图3.2 实体关系E-R图**

图 3.2详细展示了各个实体之间的关系。

****

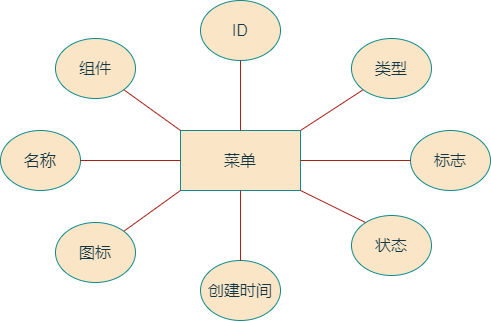
**图3.3 用户实体属性图**

图 3.3是用户实体属性图，包括姓名、密码、昵称、标志、状态、创建时间、更新时间几个属性。



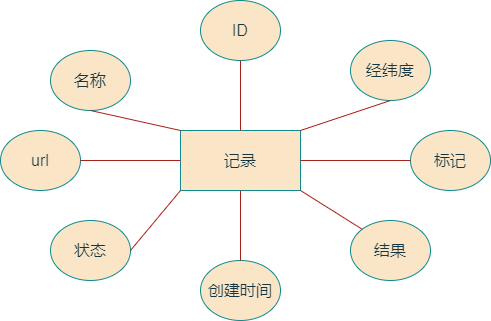
**图3.4 微信用户实体属性图**

图 3.4是微信用户实体属性图，包括昵称、标志、Open ID、状态、创建时间几个属性。



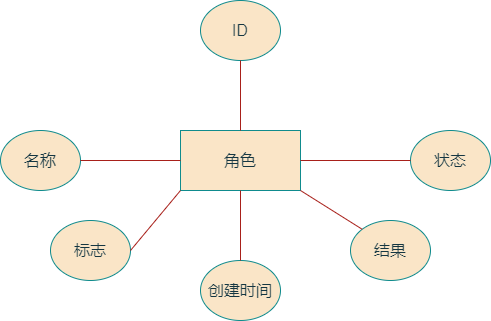
**图3.5 菜单实体属性图**

图 3.5是菜单实体属性图，包括组件、名称、创建时间、状态、图标、标志、类型几个属性。



**图3.6 记录实体属性图**

图 3.6是记录实体属性图，包括名称、经纬度、创建时间、标记、结果几个属性。



**图3.7 角色实体属性图**

图 3.7是角色实体属性图，包括名称、标志、创建时间、状态几个属性。

### **3.3.2 表设计**

通过逻辑设计，主要设计出了以下几个表，它们的字段信息如下：

（1）用户表user（如表3.1）

**表3.1 用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| id | bigint |  | Not null | id | 主键 |
| username | varchar | 64 | Not null | 用户名 |  |
| password | varchar | 64 | Not null | 密码 |  |
| nick\_name | varchar | 64 | Not null | 昵称 |  |
| create\_time | datatime |  | null | 创建时间 |  |
| status | char | 1 | null | 账号状态 |  |
| del\_flag | int |  | Not null | 删除标记 |  |
| avatar | varchar | 64 | null | 账号头像 |  |
| update\_time | datetime |  | null | 更新时间 |  |

用户表主要是存储管理员、普通用户用户名和密码，系统登录时需要用到它。

（2）菜单信息表(如表3.2)

**表3.2 菜单信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| id | int | 10 | Not null | 菜单ID | 主键 |
| parent\_id  menu\_name | bigint varchar | 50 | null  Not null | 菜单名称  父菜单ID |  |
| path | varchar | 200 | null | 路由地址 |  |
| component | varchar | 255 | null | 组件路径 |  |
| menu\_type  status  perms  icon  create\_by  create\_time | char  char  varchar  varchar  bigint  datetime | 1  1  100  100 | null  null  null  null  null  null | 菜单类型  菜单状态  权限标识  菜单图标  创建者  创建时间 |  |

菜单信息表主要是用于存储菜单信息的，与菜单相关的基本信息都要参照这张表做关联。

（3）记录信息表（如表3.3）

**表3.3记录信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| id | int |  | Not null |  | 主键 |
| name | varchar | 255 | null | 检测名称 |  |
| img\_url | varchar | 255 | null | 图片地址 |  |
| remark | varchar | 255 | null | 备注 |  |
| longitude  latitude  result\_name  result\_value  status  create\_by  create\_time  del\_flag | varchar  varchar  varchar  varchar  char  varchar  datetime  int | 255  255  255  255  1  255 | null  null  null  null  null  null  null  null | 经度  纬度  识别结果  相似度  记录状态  创建者  创建时间  删除标志 |  |

记录信息表主要用于存储记录信息，包括名称、地址、备注、经纬度等。

（4）角色信息表（如表3.4）

**表3.4角色信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| id | int |  | Not null |  | 主键 |
| role\_name | varchar | 255 | null | 角色名称 |  |
| role\_sort | int |  | null | 显示顺序 |  |
| status | char | 1 | null | 角色状态 |  |
| del\_flag  create\_time  update\_by  update\_time | char  datatime  bigint  datetime | 1 | null  null  null  null | 删除标志  创建时间  更新者  更新时间 |  |

角色信息表主要用于存储角色信息的，系统根据角色状态标志显示角色类型。

（5）微信用户信息表（如表3.5）

**表3.5 微信用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| id | int |  | Not null |  | 主键 |
| nick\_name | varchar | 255 | null | 用户昵称 |  |
| open\_id | varchar | 255 | null | 微信开放ID |  |
| status  create\_time | char  datetime | 1 | null  null | 状态  创建时间 |  |

微信用户信息表主要存储微信用户的个人信息，管理员和用户都可以查阅。

（6）角色菜单关联表（如表3.6）

**表3.6课程表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 | 主/外键 |
| role\_id  menu\_id | bigint  bigint |  | Not null  Not null | 角色ID  菜单ID |  |

角色菜单关联表表主要存储角色与菜单相关信息。

# **4 详细设计与实现**

详细设计阶段就是为每个模块完成的功能进行具体的描述，要把功能描述转变为精确的、结构化的过程描述。详细设计阶段常用的描述方式有：流程图、N-S图、PAD图、伪代码等根据软件工程的思想。在这个阶段，各个模块可以分给不同的人去并行设计。在详细设计阶段，设计者的工作对象是一个模块，根据概要设计赋予的局部任务和对外接口，设计并表达出模块的算法、流程、状态转换等内容。这里要注意，如果发现有结构调整（如分解出子模块等）的必要，必须返回到概要设计阶段，将调整反应到概要设计文档中，而不 能就地解决，不打招呼。详细设计文档最重要的部分是模块的流程图、状态图、局部变量及相应的文字说明等。一个模块一篇详细设计文档。在详细设计阶段，需要解决的问题是如何具体地实现这个系统，即设计实现每个模块功能所需算法和所需数据结构。

## **4.1 核心技术简介**

本系统基于VGG设计，采用了一些比较流行的技术WEB设计技术和CNN技术，这是本系统的核心技术，在具体实现功能模块前，有必要对这些技术做些简要的说明介绍。

### **4.1.1 SPA**

SPA 是 single page web application 的缩写，即单页Web应用，就是只有一张Web页面的应用，是加载单个HTML 页面并在用户与应用程序交互时动态更新该页面的Web应用程序。浏览器一开始会加载必需的HTML、CSS和JavaScript，所有的操作都在这张页面上完成，都由JavaScript来控制，通过JavaScript动态控制页面中显示的内容，从而模拟出多个页面的效果。

SPA在现在非常常见，当前流行的Vue、React等许多框架都使用了SPA。最直观的感受就是，当你、在当前页面触发了某个跳转使当前的网页URL改变了，但是依然停留在当前页面，并没有打开新窗口或者刷新到新页面。

### **4.1.2 Spring Boot**

Spring Boot是由Pivotal团队提供的一套开源框架，可以简化spring应用的创建及部署。它提供了丰富的Spring模块化支持，可以帮助开发者更轻松快捷地构建出企业级应用。Spring Boot通过自动配置功能，降低了复杂性，同时支持基于JVM的多种开源框架，可以缩短开发时间，使开发更加简单和高效。Spring Boot是在Spring的基础上面搭设的框架，目的是为了简化Spring项目的搭设和开发过程。不存在冲突的问题。Spring Boot自动配置 Spring-boot-starter 开箱即用依赖模块，简化了统一配置文件，内嵌了如Tomcat，Jetty,所有的依赖都打到一个jar包里面，可以直接运行。Spring Boot的设计目标是避免在Spring中进行复杂的XML配置，以更简单的方式开发生产就绪的Spring应用程序，减少开发时间并独立运行应用程序。Spring Boot会根据使用@EnableAutoConfiguration批注添加到项目中的依赖项自动配置应用程序。例如，如果MySQL数据库在类路径上，但尚未配置任何数据库连接，则Spring Boot会自动配置内存数据库。Spring Boot应用程序的入口点是包含@SpringBootApplication注释和main方法的类。Spring Boot使用@ComponentScan注释自动扫描项目中包含的所有组件。

### **4.1.3 云对象存储**

云对象存储（Cloud Object Storage，COS）是由腾讯云推出的无目录层次结构、无数据格式限制，可容纳海量数据且支持 HTTP/HTTPS 协议访问的分布式存储服务。腾讯云 COS 的存储桶空间无容量上限，无需分区管理，适用于 CDN 数据分发、数据万象处理或大数据计算与分析的数据湖等多种场景。COS 提供网页端管理界面、多种主流开发语言的 SDK、API 以及命令行和图形化工具，并且兼容 S3 的 API 接口，方便用户直接使用社区工具和插件。腾讯云对象存储提供数据跨多架构、多设备冗余存储，为用户数据提供异地容灾和资源隔离功能。COS 提供防盗链功能，可屏蔽恶意来源的访问；支持数据 SSL 加密传输，控制每个单独文件的读写权限。结合腾讯的攻击防御系统，能够有效抵御 DDoS 攻击、CC 攻击，保障您的业务正常运行。SSL 加密传输，控制每个单独文件的读写权限。结合腾讯的攻击防御系统，能够有效抵御 DDoS 攻击、CC 攻击，保障您的业务正常运行。使用 COS，无需传统硬件的采购、部署和运维，从而节省了运维工作和托管成本。COS 支持按需按量使用，无需预先支付任何预留存储空间的费用，通过生命周期管理进行数据降冷，进一步降低成本。COS 提供图形化程序、命令行工具、协议工具等多种途径对存储对象进行批量操作，让使用更为简单。COS 还提供能够将存储桶挂载到本地的工具，让您能像使用本地文件系统一样直接操作腾讯云对象存储。COS 可作为大数据计算的数据池，为大数据分析与计算提供数据源；也可以结合 SCF 服务可以实现事件通知及自动处理。

### **4.1.4 Vue**

Vue是渐进式JavaScript框架，易学易用，性能出色，适用场景丰富。经过编译器优化、完全响应式的渲染系统，几乎不需要手动优化。Vue 是一个框架，也是一个生态。其功能覆盖了大部分前端开发常见的需求。但 Web 世界是十分多样化的，不同的开发者在 Web 上构建的东西可能在形式和规模上会有很大的不同。考虑到这一点，Vue 的设计非常注重灵活性和“可以被逐步集成”这个特点。根据需求场景，可以用不同的方式使用 Vue：无需构建步骤，渐进式增强静态的 HTML；在任何页面中作为 Web Components 嵌入；单页应用 (SPA)；全栈 / 服务端渲染 (SSR)；Jamstack / 静态站点生成 (SSG)；开发桌面端、移动端、WebGL，甚至是命令行终端中的界面。

### **4.1.5 VGG**

VGG是Oxford的Visual Geometry Group的组提出的。该网络是在ILSVRC 2014上的相关工作，主要工作是证明了增加网络的深度能够在一定程度上影响网络最终的性能。VGG有两种结构，分别是VGG16和VGG19，两者并没有本质上的区别，只是网络深度不一样。VGG16相比AlexNet的一个改进是采用连续的几个3x3的卷积核代替AlexNet中的较大卷积核（11x11，7x7，5x5）。对于给定的感受野（与输出有关的输入图片的局部大小），采用堆积的小卷积核是优于采用大的卷积核，因为多层非线性层可以增加网络深度来保证学习更复杂的模式，而且代价还比较小（参数更少）。在VGG中，使用了3个3x3卷积核来代替7x7卷积核，使用了2个3x3卷积核来代替5\*5卷积核，这样做的主要目的是在保证具有相同感知野的条件下，提升了网络的深度，在一定程度上提升了神经网络的效果。VGGNet的结构非常简洁，整个网络都使用了同样大小的卷积核尺寸（3x3）和最大池化尺寸（2x2）。几个小滤波器（3x3）卷积层的组合比一个大滤波器（5x5或7x7）卷积层好：验证了通过不断加深网络结构可以提升性能。但VGG耗费更多计算资源，并且使用了更多的参数，导致更多的内存占用，其中绝大多数的参数都是来自于第一个全连接层。

### 4.1.6 Flask

Flask是一个轻量级的WSGI web应用程序框架。它旨在使入门变得快速而简单，并能够扩展到复杂的应用程序。它最初是一个围绕Werkzeug和Jinja的简单包装器，现已成为最流行的Python web应用程序框架之一。Flask提供了建议，但没有强制执行任何依赖项或项目布局。由开发人员来选择他们想要使用的工具和库。社区提供了许多扩展，可以轻松添加新功能。Flask本身相当于一个内核，其他几乎所有的功能都要用到扩展，都需要用第三方的扩展来实现。用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。可以选择MySQL，也可以用NoSQL。其 WSGI 工具箱采用 Werkzeug（路由模块），模板引擎则使用 Jinja2 。Flask相对于Django而言是轻量级的Web框架。和Django不同，Flask轻巧、简洁，通过定制第三方扩展来实现具体功能。可定制性，通过扩展增加其功能，这是Flask最重要的特点。Flask的两个主要核心应用是Werkzeug和模板引擎Jinja。Flask程序运行过程：所有Flask程序必须有一个程序实例。Flask调用视图函数后，会将视图函数的返回值作为响应的内容，返回给客户端。一般情况下，响应内容主要是字符串和状态码。用户向浏览器发送http请求，web服务器把客户端所有请求交给Flask程序实例，程序用Werkzeug来做路由分发，每个url请求，找到具体的视图函数。路由的实现是通过route装饰器实现的，调用视图函数，获取数据后，把数据传入模块中，模块引擎渲染响应的数据，由Flask返回给浏览器。

## **4.2 系统登陆模块**

### **4.2.1模块设计思路**

该界面主要是为了对整个系统进行安全管理，系统的用户名以及密码都存储在数据库user用户表中，根据不同用户身份而设置不同的权限功能，包括管理员、普通用户，用户可以根据实际情况选择合适的身份登录该系统。

相关代码：

//前端账号

String userName = userLoginDto.getUserName();

//前端的密码

String password = userLoginDto.getPassword();

//查看这个账号是不是存在的

User user = userMapper.selectOne (Wrappers.<User> lambdaQuery(). eq(User::getUserName, userName));

if (Objects.isNull(user)) {

//账号不存在

throw new SystemException (AppHttpCode Enum .LOGIN\_ERROR);

}

if(user.getStatus().equals(SysConstant.USER\_STATE\_BAN)){

//账号停用

throw new SystemException(AppHttpCodeEnum.USER\_BAN);

}

//加密

String newPassword = SaSecureUtil.md5(password);

//对前端提交的账号和密码

if (user.getUserName().equals(userName) && user.getPassword().equals(newPassword)) {

//根据账号id，进行登录

StpUtil.login(user.getId());

//获取token

String token = StpUtil.getTokenValue();

//登录成功响应前端

UserLoginVo userLoginVo = new UserLoginVo(token,user.getUserName());

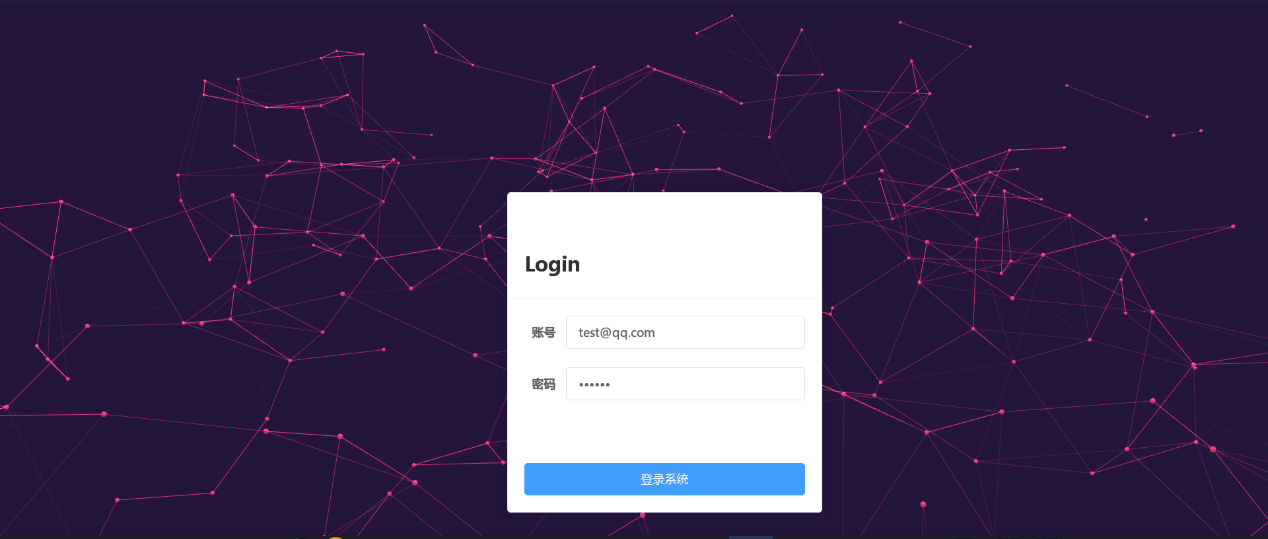
return Result.okResult(userLoginVo);

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

### **4.2.2 系统登录主页图**

请根据类型管理员、普通用户输入相应的用户名和密码，主界面如图4.1所示：



**图4.1登陆页面**

## **4.3菜单管理模块**

### **4.3.1模块设计思路**

该模块主要是对菜单进行管理和操作，管理员可以添加、修改、删除菜单信息。同时还可以通过对名称、类型、状态的输入实现多条件查询。模块之间相互独立，从而完成对整个菜单的管理操作。

相关代码：

（1）新增菜单：

管理员可以对菜单进行管理，可以新增菜单，具体代码如下：

@ApiOperation("新增菜单")

@SaCheckPermission("menu::add")

@PostMapping

public Result addMenu(@RequestBody Menu menu) {

boolean save = menuService.save(menu);

if (save) {

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

public static Result okResult() {

Result result = new Result();

return result;

}

（2）修改菜单：

管理员可以对菜单进行管理，可以对菜单信息进行修改，具体代码如下：

@ApiOperation("修改菜单")

@SaCheckPermission("menu::put")

@PutMapping

public Result updateMenu(@RequestBody Menu menu) {

boolean res = menuService.updateById(menu);

if (res) {

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

（3）查询菜单

管理员可以对菜单进行管理，可以对班级信息按照名称、类型、状态进行多条件查询，查询到的信息包括名称、类型、状态、路径，同时可以对查询到的报表信息进行打印操作。

具体代码如下：

@ApiOperation("查看所有菜单")

@SaCheckPermission("menu::query")

@GetMapping("/list")

public Result list(String keywords) {

return menuService.selectMenuPage(keywords);

}

@ApiOperation("根据id查询菜单")

@SaCheckPermission("menu::query")

@GetMapping

public Result geMenuById(Long id) {

Menu menu = menuService.getById(id);

return Result.okResult(menu);

}

@ApiOperation("根据角色id查询拥护的菜单")

@SaCheckPermission("menu::query")

@GetMapping("/role/ids")

public Result roleMenuIds(Long id) {

return menuService.selectRoleMenuIds(id);

}

管理员可以对菜单信息按照名称、类型、状态进行多条件查询，其中，查询到的信息还包括创建时间、更新时间、路径、图标、删除标志。可以对查询到的报表信息进行排序操作。

@Override

public Result selectMenuPage(String keywords) {

LambdaQueryWrapper<Menu> queryWrapper = new LambdaQueryWrapper<>();

//模糊查询

queryWrapper.like(Objects.nonNull(keywords), Menu::getMenuName, keywords);

//排序

queryWrapper.orderByAsc(Menu::getOrderNum);

List<Menu> list = list(queryWrapper);

return Result.okResult(list);

}

@Override

public Result selectRoleMenuIds(Long id) {

List<Long> roleMenuIds = getBaseMapper().selectRoleMenuList(id);

return Result.okResult(roleMenuIds);

}

（4）删除菜单信息

管理员可以对菜单进行删除操作，具体代码如下：

@ApiOperation("删除菜单")

@SaCheckPermission("menu::delete")

@DeleteMapping

public Result deleteMenu(Long id) {

boolean res = menuService.removeById(id);

if (res) {

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

@ApiOperation("根据角色id查询拥护的菜单")

@SaCheckPermission("menu::query")

@GetMapping("/role/ids")

public Result roleMenuIds(Long id) {

return menuService.selectRoleMenuIds(id);

}

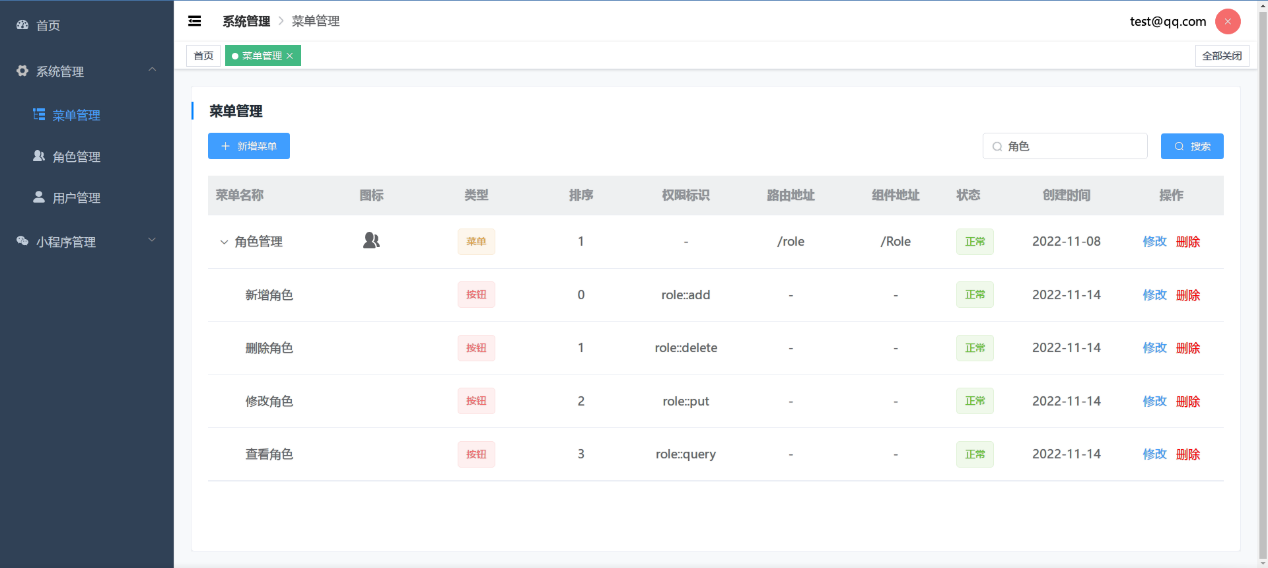
### **4.3.2菜单界面图**

菜单管理模块是管理员及普通用户可以进行操作的部分，该模块中可以新增菜单信息，新增界面如图4.4所示：



**图4.2菜单新增界面图**

菜单信息新增后，可以对菜单新增信息进行查询和删除、修改操作，界面如图4.5所示：



**图4.3菜单管理查询界面图**

****

**图4.4 菜单管理修改界面图**

****

**图4.5菜单管理删除界面图**

## **4.4角色管理模块**

### **4.4.1模块设计思路**

此模块主要是进行角色信息的管理操作，其中包括管理员、普通用户对菜单信息的新增、删除、修改操作，其中管理员和普通用户还可以按照名称、权限字符、状态和创建时间对角色进行多条件查询，从而使相关角色更快速的获取信息。

相关代码：

（1）新增角色信息：

管理员可以对角色信息进行管理，可以新增角色信息，具体代码如下：

@ApiOperation(value = "新增角色", notes = "新增角色的同时，分配菜单")

@SaCheckPermission("role::add")

@PostMapping

public Result addRole(@RequestBody AddRoleDto addRoleDto) {

return roleService.addRole(addRoleDto);

}

@Data

public class AddRoleDto {

//角色名称

private String roleName;

//角色权限字符串

private String roleKey;

//显示顺序

private Integer roleSort;

//角色状态（0正常 1停用）

private String status;

//勾选的菜单id

private List<Long> menuIds;

}

@Override

public Result addRole(AddRoleDto addRoleDto) {

//新增角色

Role role = BeanCopyUtils.copyBean(addRoleDto, Role.class);

save(role);

//根据新增角色返回的id，批量删除对应的菜单

Long id = role.getId();

roleMenuMapper.deleteById(id);

//将选择的菜单分配给角色

List<Long> menuIds = addRoleDto.getMenuIds();

for (Long menuId : menuIds) {

RoleMenu roleMenu = new RoleMenu(id, menuId);

roleMenuMapper.insert(roleMenu);

}

return Result.okResult();

}

（2）修改角色信息：

管理员和普通用户登录系统时，可以对角色信息进行修改操作，具体代码如下：

@ApiOperation(value = "修改角色", notes = "修改角色的同时，修改菜单")

@SaCheckPermission("role::put")

@PutMapping

public Result updateRole(@RequestBody UpdateRoleDto updateRoleDto) {

if(Objects.isNull(updateRoleDto.getMenuIds())){

Role role = BeanCopyUtils.copyBean(updateRoleDto, Role.class);

roleService.updateById(role);

return Result.okResult();

}

return roleService.updateRole(updateRoleDto);

}

@Override

public Result updateRole(UpdateRoleDto updateRoleDto) {

//修改角色

Role role = BeanCopyUtils.copyBean(updateRoleDto, Role.class);

updateById(role);

//批量删除对应的菜单

Long id = role.getId();

roleMenuMapper.deleteById(id);

//将选择的菜单分配给角色

List<Long> menuIds = updateRoleDto.getMenuIds();

for (Long menuId : menuIds) {

RoleMenu roleMenu = new RoleMenu(id, menuId);

roleMenuMapper.insert(roleMenu);

}

return Result.okResult();

}

（3）查询角色信息

管理员和普通用户可以对角色信息进行管理，可以对角色按照名称、权限字符、状态和创建时间进行多条件查询，查询到的信息包括名称、权限字符、状态、显示顺序和创建时间同时进行模糊查询和排序操作。

具体代码如下：

@ApiOperation("查看所有角色")

@SaCheckPermission("role::query")

@GetMapping("/roles")

public Result list() {

List<Role> list = roleService.list();

return Result.okResult(list);

}

@ApiOperation("根据id查询角色")

@SaCheckPermission("role::query")

@GetMapping

public Result geRoleById(Long id) {

Role role = roleService.getById(id);

return Result.okResult(role);

}

@Autowired

private RoleService roleService;

@ApiOperation("分页查看所有角色")

@SaCheckPermission("role::query")

@GetMapping("/list")

public Result list(Integer pageNum, Integer pageSize, String keywords) {

return roleService.roleList(pageNum, pageSize, keywords);

}

@Autowired

private RoleMenuMapper roleMenuMapper;

@Override

public Result roleList(Integer pageNum, Integer pageSize, String keywords) {

LambdaQueryWrapper<Role> queryWrapper = new LambdaQueryWrapper<>();

//模糊查询

queryWrapper.like(Objects.nonNull(keywords), Role::getRoleName, keywords);

//排序

queryWrapper.orderByAsc(Role::getRoleSort);

//分页

Page<Role> rolePage = new Page<>(pageNum, pageSize);

page(rolePage, queryWrapper);

PageVo pageVo = new PageVo(rolePage.getRecords(), rolePage.getTotal());

return Result.okResult(pageVo);

}

（4）删除角色信息

管理员和用户可以对角色信息进行删除操作，具体代码如下：

@ApiOperation(value = "删除角色")

@SaCheckPermission("role::delete")

@DeleteMapping("/{ids}")

public Result deleteRole(@PathVariable List<Long> ids) {

boolean b = roleService.removeByIds(ids);

if (b) {

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

public static Result okResult() {

Result result = new Result();

return result;

}

public static Result okResult(int code, String msg) {

Result result = new Result();

return result.ok(code, null, msg);

}

public static Result okResult(Object data) {

Result result = setAppHttpCodeEnum(AppHttpCodeEnum.SUCCESS, AppHttpCodeEnum.SUCCESS.getMsg());

if (data != null) {

result.setData(data);

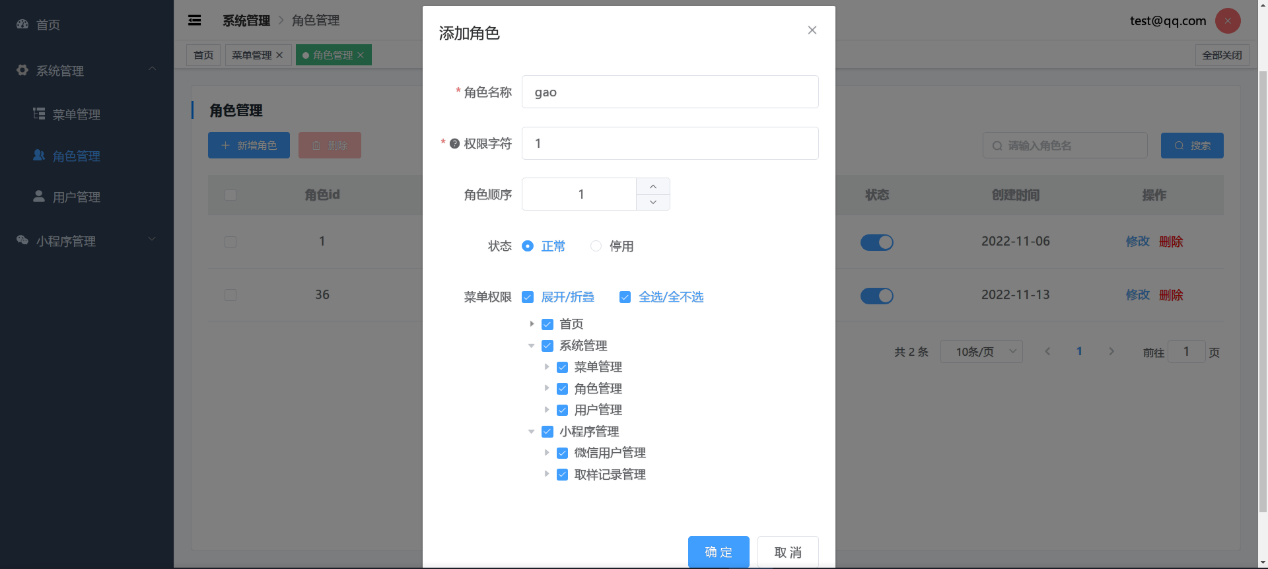
}

return result;

}

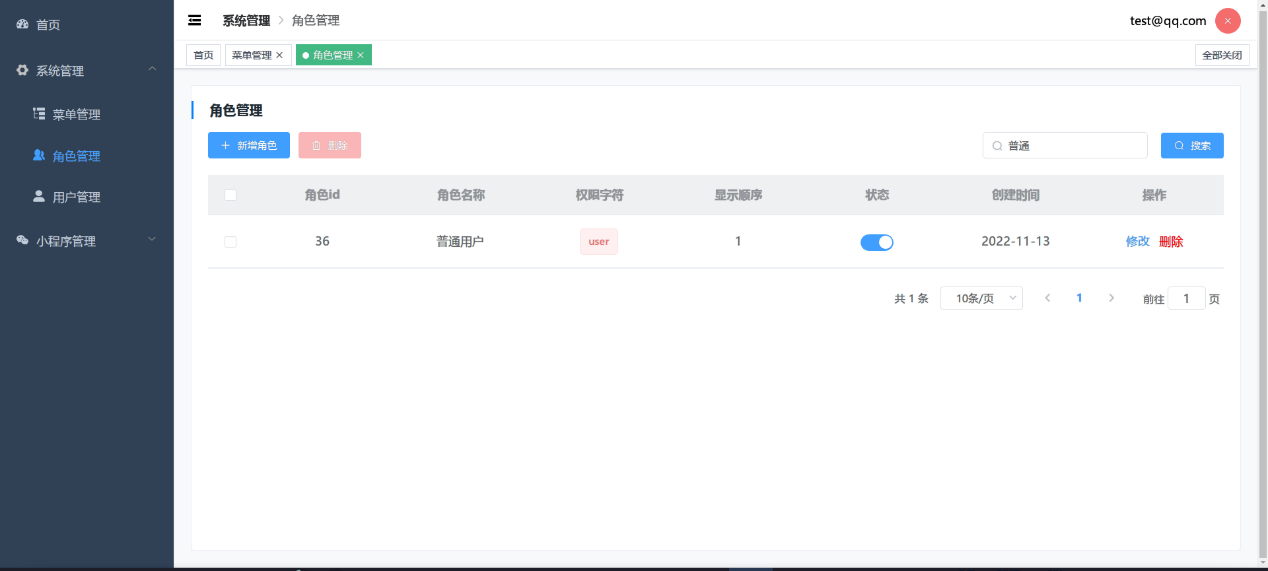
### **4.4.2角色界面图**

角色管理模块中管理员可以新增角色信息，具体界面如图4.6所示：

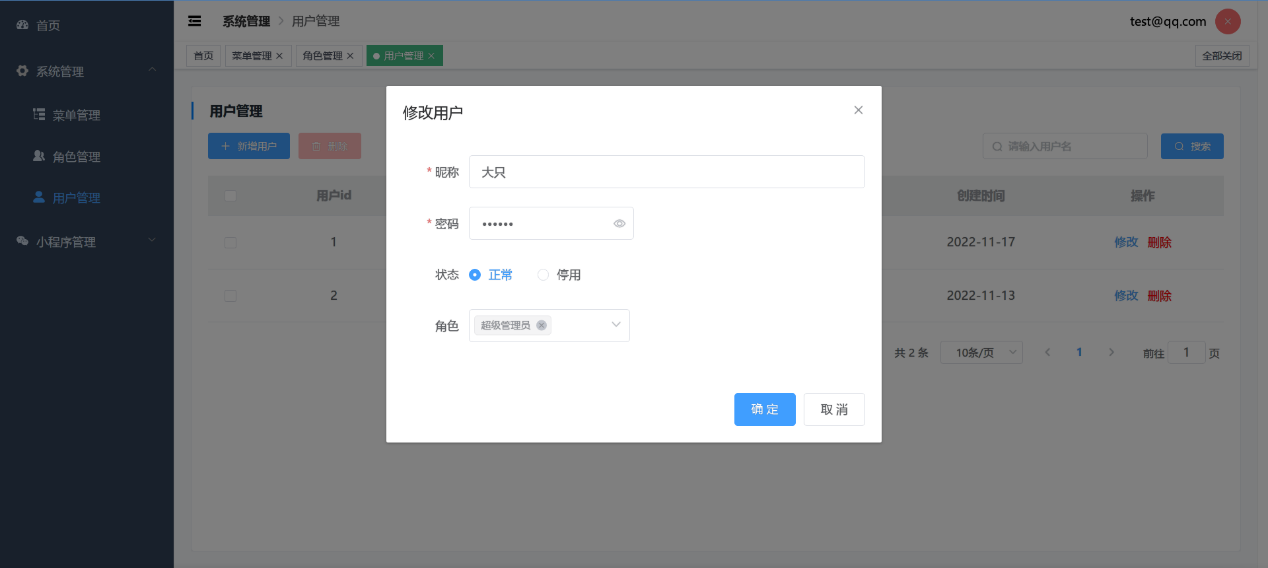


**图4.6学生信息新增界面图**

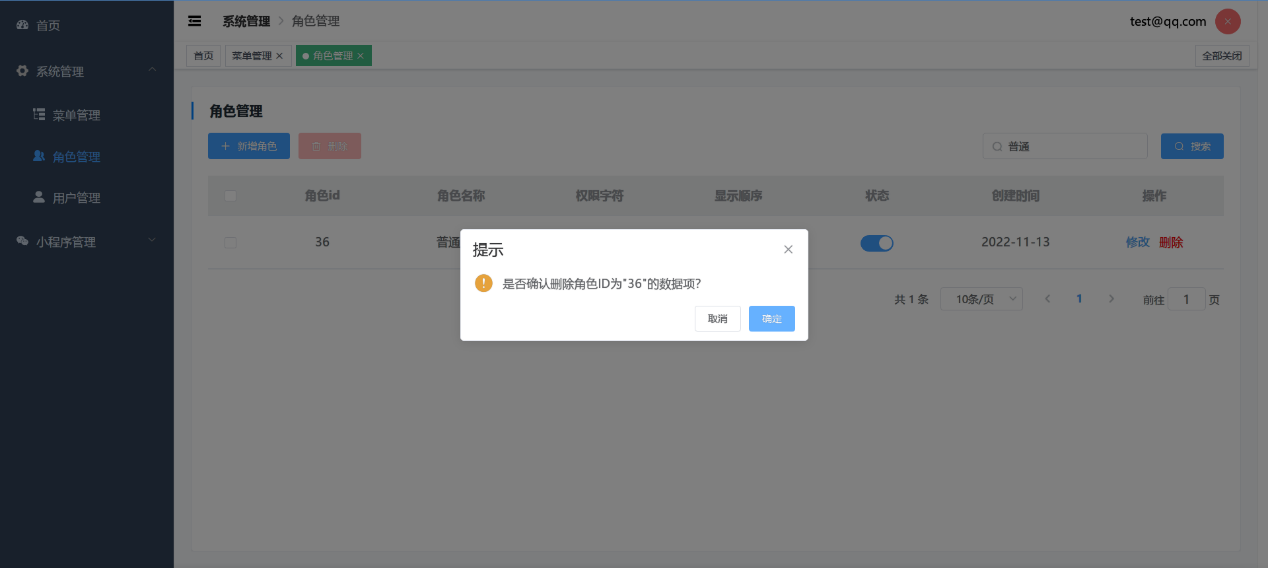
角色信息新增后，管理员和普通用户还可以对角色信息进行多条件查询，同时还可以进行角色信息的修改和删除操作，具体界面如图4.7所示：



**图4.7角色信息查询界面图**

****

**图4.8角色信息修改界面图**

****

**图4.9角色信息删除界面图**

## **4.5用户管理模块**

### **4.5.1模块设计思路**

该模块主要实现管理员对用户的管理操作，其中包括对用户信息的新增、修改、删除操作，还可以根据用户编号进行查询。整个模块使管理员对相关用户情况进行有效的管理。

相关代码：

（1）管理员新增用户信息：

管理员可以对用户信息进行管理，可以新增用户信息，具体代码如下：

@ApiOperation(value = "新增用户", notes = "新增用户的同时，分配角色")

@SaCheckLogin

@SaCheckPermission("user::add")

@PostMapping

public Result addRole(@RequestBody AddUserDto addUserDto) {

return userService.addUser(addUserDto);

}

@Override

public Result addUser(AddUserDto addUserDto) {

//新增用户

User user = BeanCopyUtils.copyBean(addUserDto, User.class);

//加密

String newPassword = SaSecureUtil.md5(user.getPassword());

user.setPassword(newPassword);

save(user);

//根据新增角色返回的id，批量删除对应的菜单

Long id = user.getId();

userRoleMapper.deleteById(id);

//将选择的角色分配给用户

List<Long> roleIds = addUserDto.getRoleIds();

for (Long roleId : roleIds) {

UserRole userRole = new UserRole(id, roleId);

userRoleMapper.insert(userRole);

}

return Result.okResult();

}

（2）管理员查询用户信息：

管理员可以对新增的用户信息进行查询操作，具体代码如下：

@ApiOperation("查看所有用户")

@SaCheckLogin

@SaCheckPermission("user::query")

@GetMapping("/list")

public Result list(Integer pageNum, Integer pageSize, String keywords) {

return userService.userList(pageNum, pageSize, keywords);

}

@ApiOperation("根据id查询用户")

@SaCheckLogin

@SaCheckPermission("user::query")

@GetMapping

public Result geUserById(Long id) {

return userService.geUserById(id);

}

public Result getUserMenus() {

//获取用户id

long loginIdAsLong = StpUtil.getLoginIdAsLong();

//查询该用户拥有的菜单

List<Menu> userMenuList = getBaseMapper().selectUserByMenus(loginIdAsLong);

//安全校验

if (userMenuList.size() == 0) {

return Result.errorResult(400, "没有菜单,无法生成路由组件");

}

List<Menu> menuTree = builderMenuTree(userMenuList);

return Result.okResult(menuTree);

}

@Override

public Result userList(Integer pageNum, Integer pageSize, String keywords) {

LambdaQueryWrapper<User> queryWrapper = new LambdaQueryWrapper<>();

//模糊查询

queryWrapper.like(Objects.nonNull(keywords), User::getUserName, keywords);

//分页

Page<User> rolePage = new Page<>(pageNum, pageSize);

page(rolePage, queryWrapper);

PageVo pageVo = new PageVo(rolePage.getRecords(), rolePage.getTotal());

return Result.okResult(pageVo);

}

（3）管理员修改用户信息：

管理员可以对自己添加的用户的情况进行查询，具体代码如下：

@ApiOperation(value = "修改用户", notes = "修改用户的同时，修改菜单")

@SaCheckLogin

@SaCheckPermission("user::put")

@PutMapping

public Result updateUser(@RequestBody UpdateUserDto updateUserDto) {

if (StpUtil.getLoginIdAsLong() == updateUserDto.getId()) {

throw new SystemException(AppHttpCodeEnum.USER\_ERROR);

}

if (Objects.isNull(updateUserDto.getRoleIds())) {

User user = BeanCopyUtils.copyBean(updateUserDto, User.class);

//加密

String newPassword = SaSecureUtil.md5(user.getPassword());

user.setPassword(newPassword);

userService.updateById(user);

return Result.okResult();

}

return userService.updateUser(updateUserDto);

}

@Override

public Result updateUser(UpdateUserDto updateUserDto) {

//修改用户

User user = BeanCopyUtils.copyBean(updateUserDto, User.class);

//加密

String newPassword = SaSecureUtil.md5(user.getPassword());

user.setPassword(newPassword);

updateById(user);

//批量删除对应的角色

Long id = user.getId();

userRoleMapper.deleteById(id);

//将选择的角色分配给用户

List<Long> roleIds = updateUserDto.getRoleIds();

for (Long roleId : roleIds) {

UserRole userRole = new UserRole(id, roleId);

userRoleMapper.insert(userRole);

}

return Result.okResult();

}

（4）删除用户信息：

管理员用户的登录系统，可以对用户进行管理，对课程信息进行删除，具体代码如下：

@ApiOperation("删除角色")

@SaCheckLogin

@SaCheckPermission("user::delete")

@DeleteMapping("/{ids}")

public Result deleteUser(@PathVariable List<Long> ids) {

for (Long id : ids) {

if (StpUtil.getLoginIdAsLong() == id) {

throw new SystemException(AppHttpCodeEnum.USER\_ERROR);

}

}

boolean b = userService.removeByIds(ids);

if (b) {

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

（5）登录、注销用户

@Autowired

private LoginService loginService;

@Autowired

private UserService userService;

@PostMapping("/login")

@ApiOperation("用户登录")

public Result login(@RequestBody @Validated UserLoginDto userLoginDto) {

return loginService.login(userLoginDto);

}

@ApiOperation(value = "获取用户拥有的菜单")

@SaCheckLogin

@GetMapping("/menus")

public Result getUserMenus() {

return userService.getUserMenus();

}

@ApiOperation("注销用户")

@SaCheckLogin

@GetMapping("/logOut")

public Result userLogOut() {

if (StpUtil.isLogin()) {

StpUtil.logout();

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

@Autowired

private UserRoleMapper userRoleMapper;

@Override

public Result getUserMenus() {

//获取用户id

long loginIdAsLong = StpUtil.getLoginIdAsLong();

//查询该用户拥有的菜单

List<Menu> userMenuList = getBaseMapper().selectUserByMenus(loginIdAsLong);

//安全校验

if (userMenuList.size() == 0) {

return Result.errorResult(400, "没有菜单,无法生成路由组件");

}

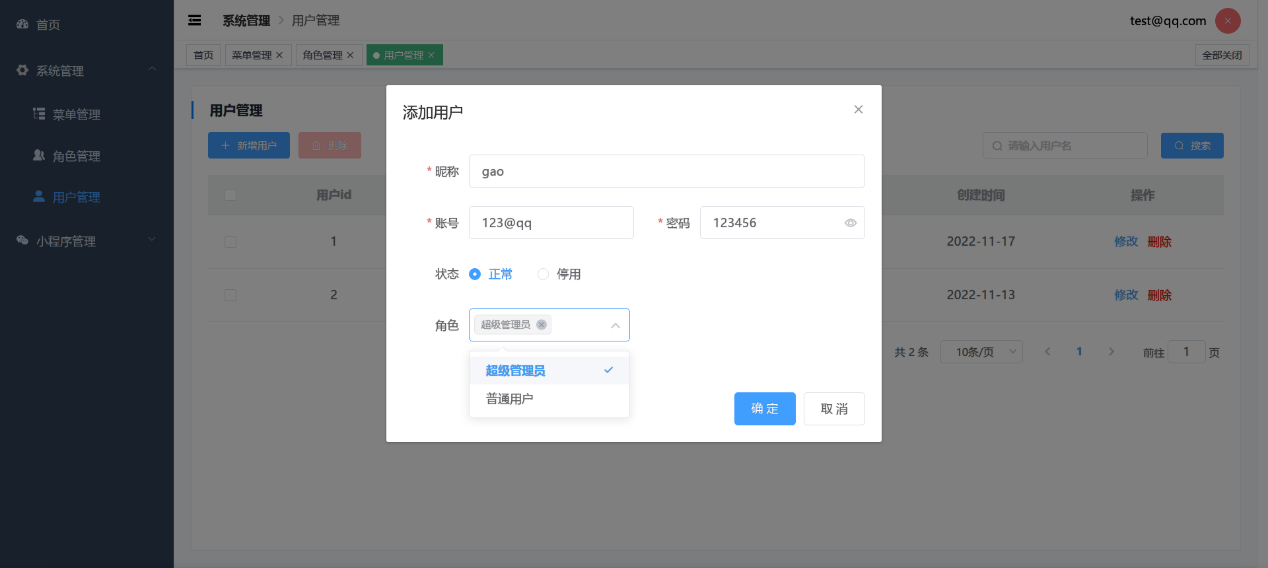
List<Menu> menuTree = builderMenuTree(userMenuList);

return Result.okResult(menuTree);

}

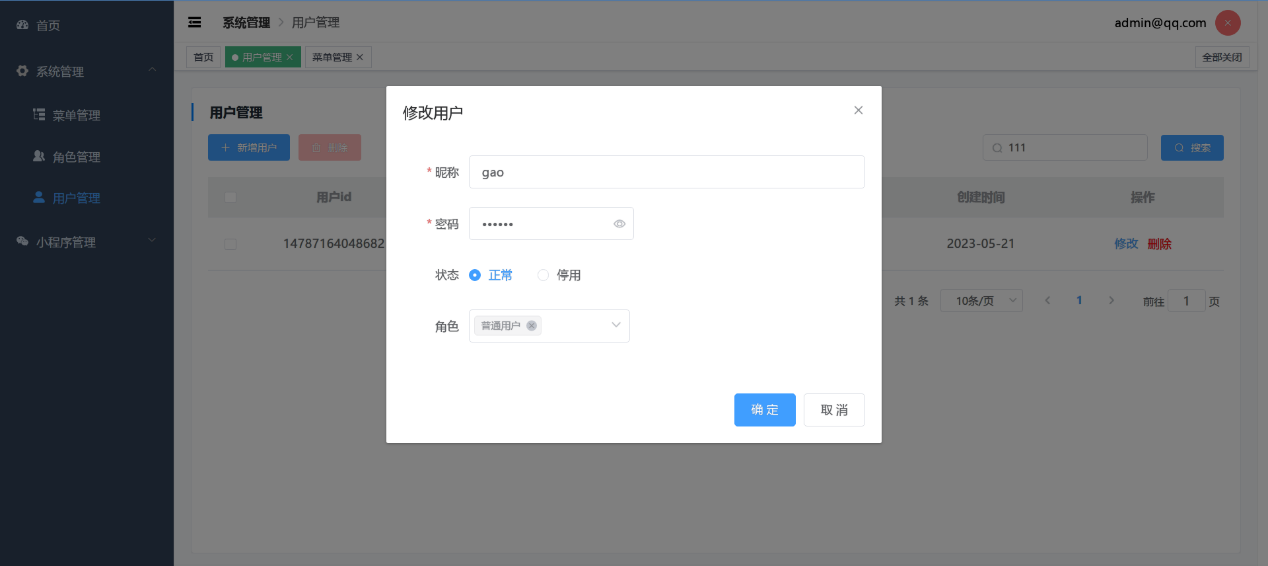
### **4.5.2用户管理界面图**

用户管理模块中管理员可以新增用户信息，具体界面如图4.10所示：

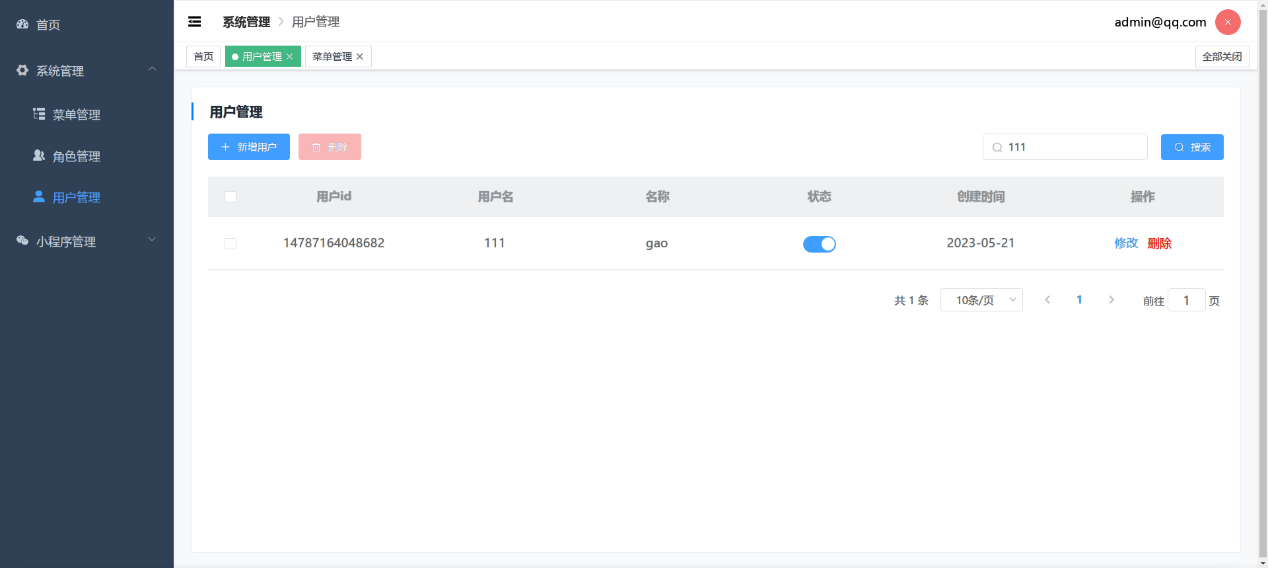


**图4.10用户新增界面图**

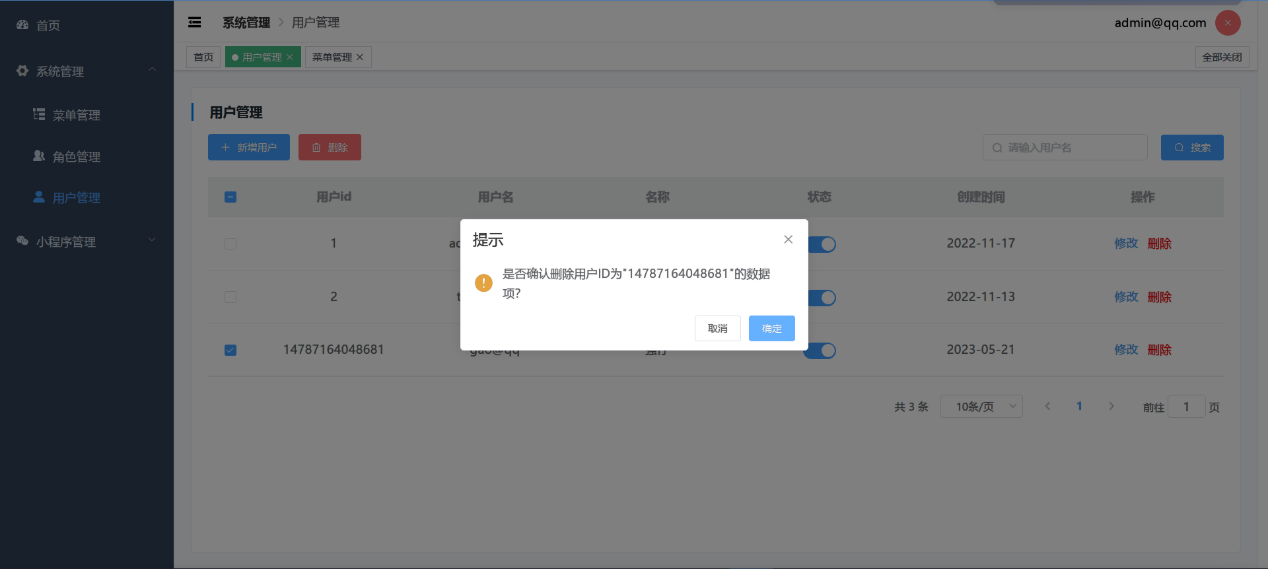
用户信息新增后管理员可以对用户信息进行查询，同时还可以进行用户信息的删除操作，管理员还可以按照用户名称对用户情况进行查询操作，具体界面如图所示：



**图4.11用户更改界面图**

****

**图4.12用户查询界面图**

****

**图4.13用户删除界面图**

## **4.6微信用户管理模块**

### **4.6.1模块设计思路**

该模块主要实现管理员对微信用户的管理操作，其中包括对微信小程序用户信息的新增、修改、删除操作，还可以根据用户编号进行查询。整个模块使管理员对相关用户情况进行有效的管理。

相关代码：

（1）管理员查看用户信息：

管理员及普通用户可以对用户信息进行管理，可以查看用户信息，具体代码如下：

@ApiOperation("查看所有用户")

@SaCheckLogin

@SaCheckPermission("wx::user::query")

@GetMapping("/list")

public Result list(Integer pageNum, Integer pageSize, String keywords) {

return wxUserService.userList(pageNum, pageSize, keywords);

}

@Override

public Result userList(Integer pageNum, Integer pageSize, String keywords) {

LambdaQueryWrapper<WxUser> queryWrapper = new LambdaQueryWrapper<>();

//模糊查询

queryWrapper.like(Objects.nonNull(keywords), WxUser::getOpenId, keywords);

//分页

Page<WxUser> wxUserPage = new Page<>(pageNum, pageSize);

page(wxUserPage, queryWrapper);

PageVo pageVo = new PageVo(wxUserPage.getRecords(), wxUserPage.getTotal());

return Result.okResult(pageVo);

}

（2）管理员更改用户信息：

管理员可以对新增的微信小程序用户信息进行更改操作，具体代码如下：

@ApiOperation("修改用户")

@SaCheckLogin

@SaCheckPermission("wx::user::put")

@PutMapping

public Result updateRole(@RequestBody WxUser wxUser) {

boolean b = wxUserService.updateById(wxUser);

if(b){

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

（3）微信用户登录、注销信息：

管理员可以对微信用户的登录、注销活动进行记录，具体代码如下：

@RestController

@RequestMapping("/wxUser")

@Api(tags = "微信用户模块")

public class WxUserController {

@Autowired

private WxUserService wxUserService;

@PostMapping("/login")

@ApiOperation("登录")

public Result wxUserLogin(@RequestBody WxUserLoginDto wxUserLoginDto) {

return wxUserService.wxUserLogin(wxUserLoginDto);

}

@SaCheckLogin

@GetMapping("/logout")

@ApiOperation("退出登录")

public Result wxUserLogin() {

if(StpUtil.isLogin()){

StpUtil.logout();

return Result.okResult();

}

return Result.errorResult(AppHttpCodeEnum.SYSTEM\_ERROR);

}

@Override

public Result wxUserLogin(WxUserLoginDto wxUserLoginDto) {

String code = wxUserLoginDto.getCode();

//拼接微信api请求地址，获取openID

String wxUrl = String.format(url, appid, secret, code);

restTemplate.getMessageConverters().add(new WxMappingJackson2HttpMessageConverter());

String forObject = restTemplate.getForObject(wxUrl, String.class);

//向微信小程序的服务器发起get请求

WxLoginVo wxLoginVo = JSON.parseObject(forObject, WxLoginVo.class);

if (Objects.isNull(wxLoginVo)) {

throw new SystemException(AppHttpCodeEnum.SYSTEM\_ERROR);

}

//得到openId

String openId = wxLoginVo.getOpenId();

//判断用户是否存在

WxUser user = getOne(new QueryWrapper<WxUser>().lambda().eq(WxUser::getOpenId, openId));

//自动注册

if (Objects.isNull(user)) {

//为空新增该用户

WxUser wxUser = new WxUser();

//生成随机昵称

String random = RandomUtils.getRandom(4);

String nickName = "微信用户" + random;

wxUser.setNickName(nickName);

wxUser.setOpenId(openId);

wxUser.setCreateTime(new Date());

if (save(wxUser)) {

//然后返回token

StpUtil.login(openId);

String token = StpUtil.getTokenValue();

return Result.okResult(new WxUserLoginVo(nickName,token));

}

}

//登录

if(user.getStatus().equals(SysConstant.USER\_STATE\_BAN)){

//账号停用

throw new SystemException(AppHttpCodeEnum.USER\_BAN);

}

//登录并直接返回token

StpUtil.login(openId);

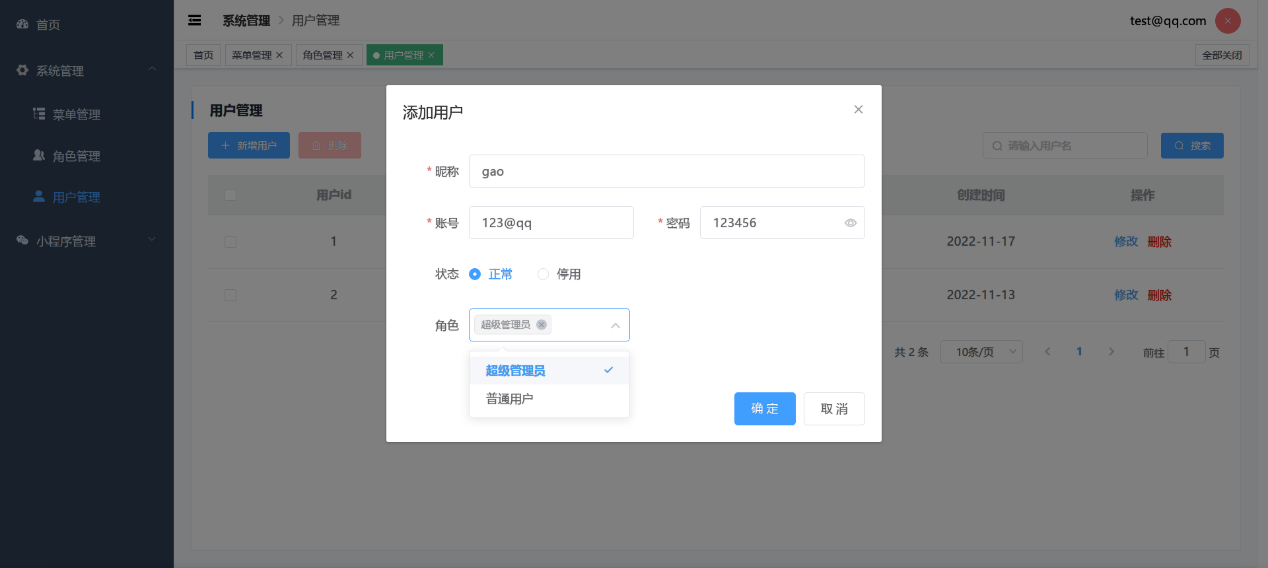
String token = StpUtil.getTokenValue();

return Result.okResult(new WxUserLoginVo(user.getNickName(),token));

}

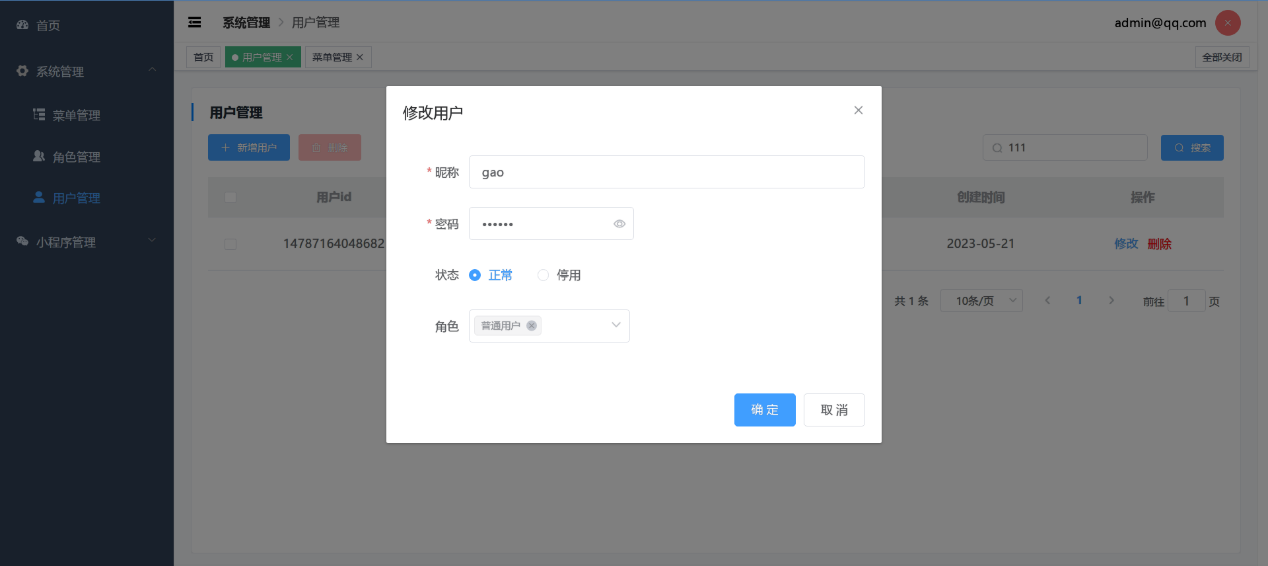
### **4.5.2用户管理界面图**

用户管理模块中管理员可以搜索微信小程序用户信息，具体界面如图4.14所示：

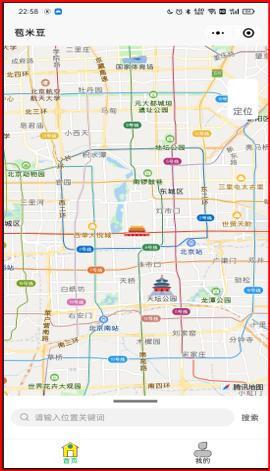


**图4.14小程序用户搜索界面图**

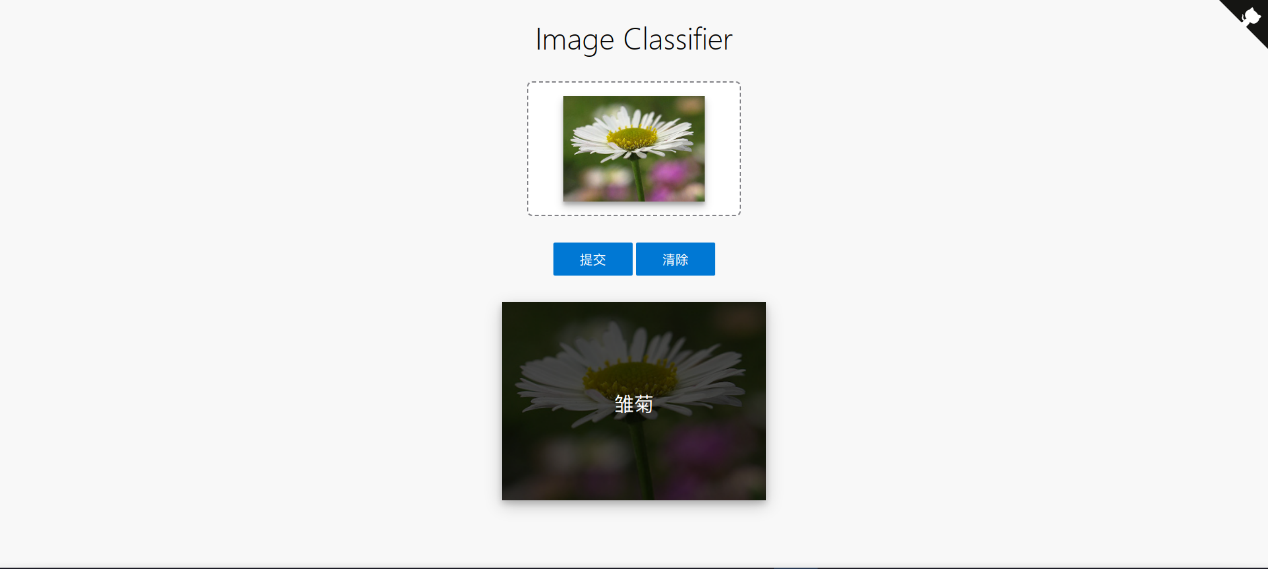
微信小程序用户信息新增后管理员可以对微信小程序用户信息进行查询，同时还可以进行用户信息的修改操作，具体界面如图所示：



**图4.15小程序用户信息更改界面图**

****

**图4.12小程序界面图**

****

**图4.13服务器识别界面图**

# **5 软件测试**

## **5.1软件测试概述**

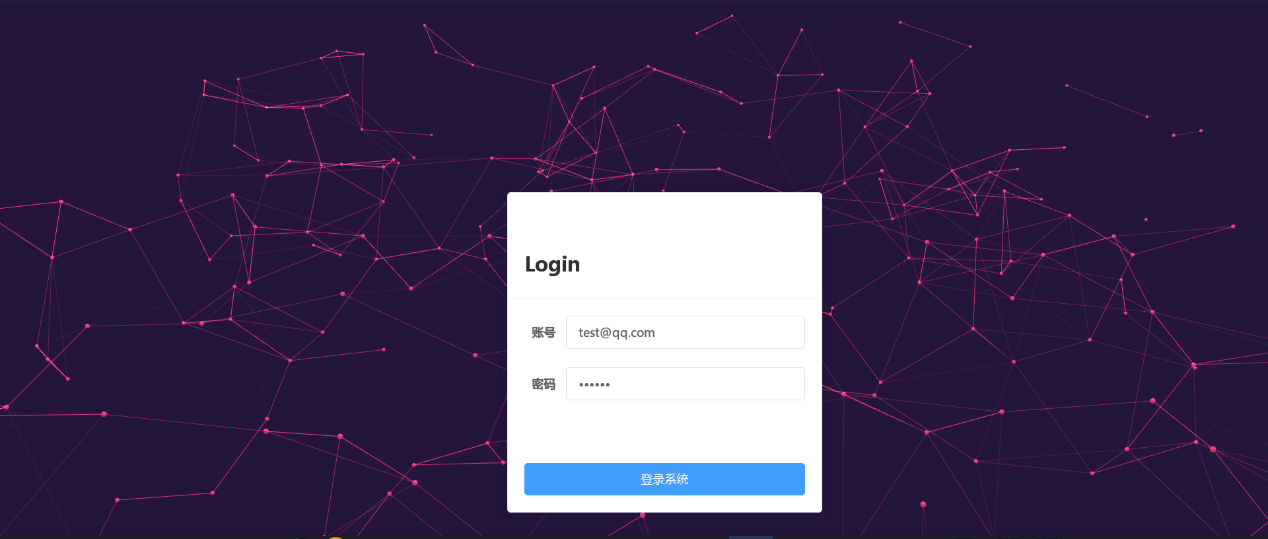
软件测试是评估和验证软件产品或应用程序是否按预期运行的过程。 测试的好处包括防止出现错误、降低开发成本和提高性能。目前存在多种不同类型的软件测试，而每种软件测试都具有特定的目标和策略：验收测试：验证整个系统是否按预期工作。集成测试：确保软件组件或功能可以一起运行。单元测试：验证每个软件单元是否按预期执行。 单元是应用程序中最小的可测试组件。功能测试：根据功能需求，通过模拟业务场景来检查功能。 黑盒测试是验证功能的常用方法。性能测试：测试软件在不同工作负载下的表现。 例如，负载测试用于评估真实负载条件下的性能。回归测试：检查新功能是否破坏或降低功能的效果。 如果没有时间进行完整的回归测试，那么可以使用健全测试在表面级别验证菜单、功能和命令。压力测试：测试系统在失败之前可以承受多大的压力。 这是一种非功能性测试。可用性测试：验证客户使用系统或 Web 应用程序完成任务的程度。优秀的测试方法应包括应用程序编程接口 (API)、用户界面和系统级别。 自动化测试越多，运行得越早，效果就越好。

## **5.2测试用例**

测试用例是为项目需求而编制的一组包含测试输入、执行条件以及预期结果的文档，以便测试某个程序是否满足客户需求。测试用例是测试工作的指导，是软件测试质量稳定的根本保障，评估测试结果的基准。有一份用例来指导测试执行，可以在测试人员疲累的时候起到一个牵引作用。编写用例的过程中，通过熟悉需求，对系统架构或业务有更深入理解。编写测试用例大体思路分为三步：第 1 步：依据需求梳理功能及功能点；第 2 步：通过测试理论方法及经验，梳理测试点；第 3 步：挖掘隐性需求，覆盖非功能测试层面。

## **5.3用户登录用例测试**

该项是对管理员及用户进行的登录测试，管理员及用户进入登录模块，可以通过输入相应的账号、密码进行登录操作，通过输入正确的初始密码，可以完成登录操作进入主界面，界面显示如图5.1：

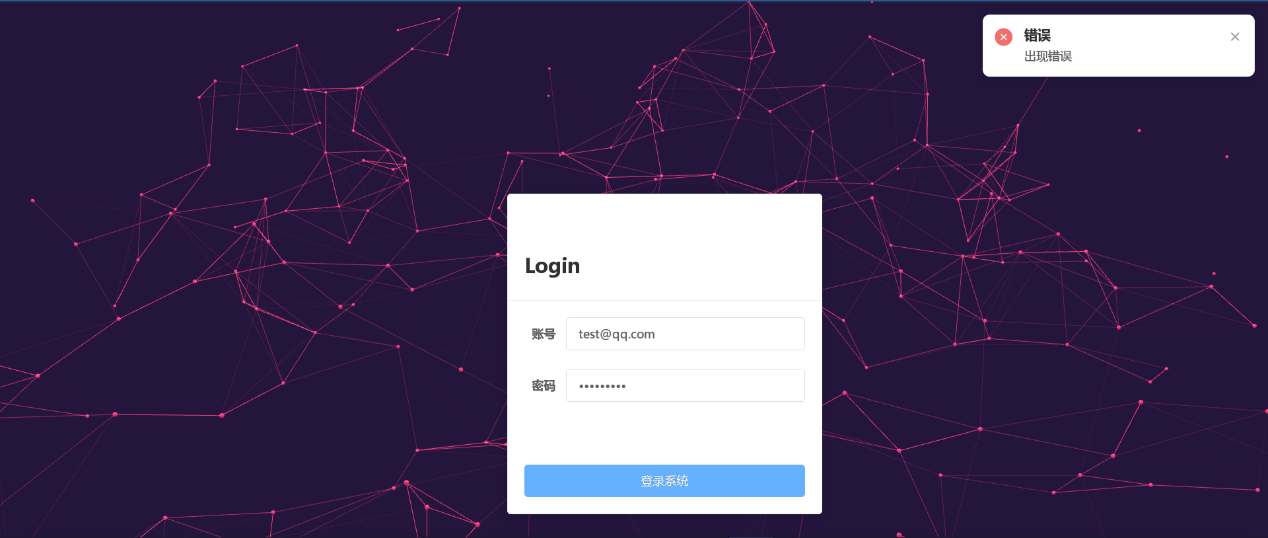


**图5.1用户登录界面**

****

**图5.2用户登录成功**

当输入错误的密码时，界面显示如图5.3：



**图5.3 登录失败界面**

## 5.4 用户权限测试

该项是对管理员及用户进行的账户权限测试，管理员及用户进入系统后，可对应用的各个模块具有不同的权限级别，因而可以进行不同的操作，管理员账户对系统的各个模块具有增删改查的全部操作权限，而普通用户只具有基本的查看操作权限，示例如图：

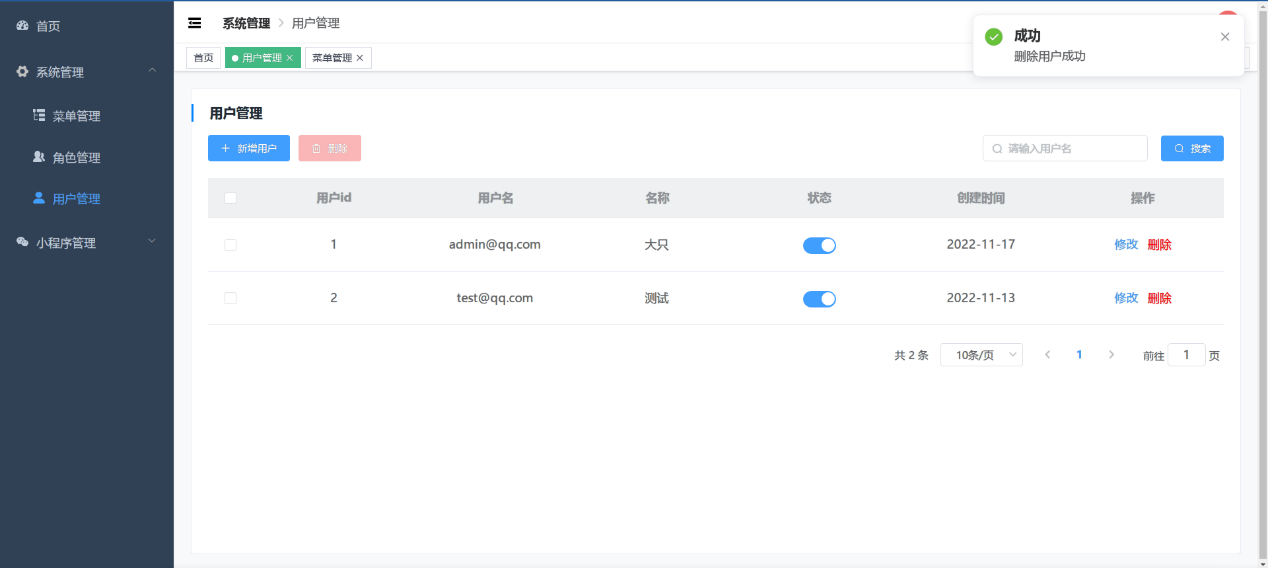
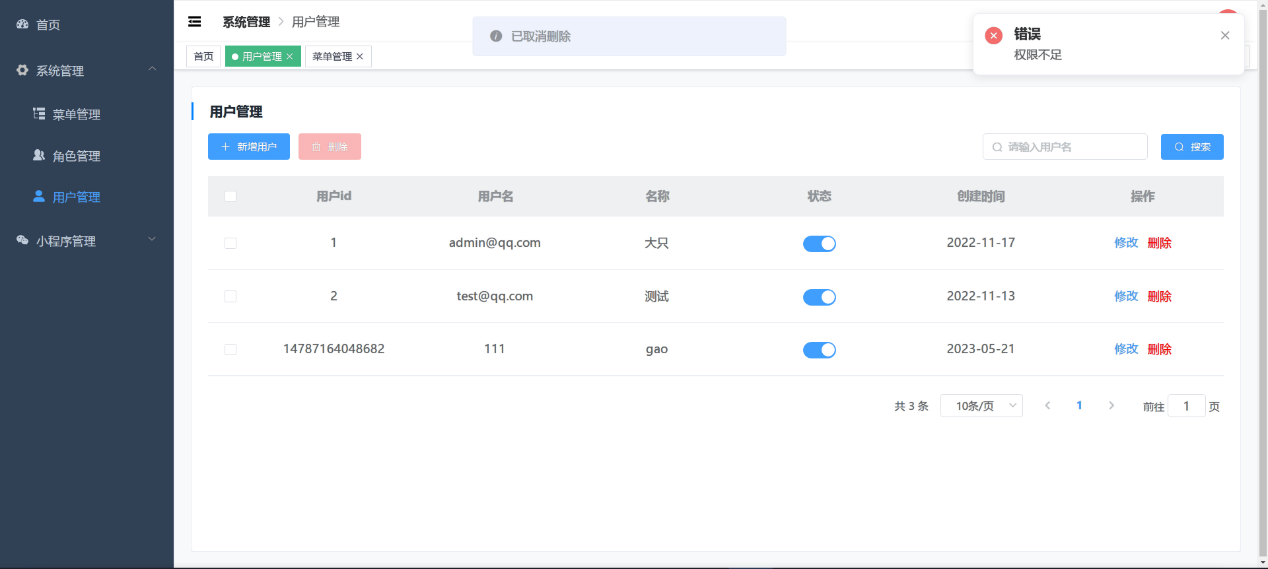


图5.4管理员删除成功界面

****

**图5.5普通用户删除失败**

# **6 总结与展望**

本系统基于B/S架构体系进行开发，采用Java语言进行编写，结合JSP技术实现基于Web的学生成绩管理系统的功能，整个系统的开发结构是按照软件工程设计思想进行，主要有可行性分析、需求分析、概要设计、详细设计、软件测试和系统实现，该系统总体上达到了整个系统功能需求，其中包括学生信息管理、成绩管理、班级管理、课程管理等，实现了以Internet网络为核心，通过数据库对采集的学生信息进行及时高效的管理。

系统总体上来说功能需求初步实现，系统层次分明，结构还算清晰，当然，该系统还有些不尽人意的地方，比如进一步优化DIV+CSS样式设计，改善用户界面，增强用户的使用体验度；使用更高效的系统开发平台；优化出错处理方案，增加系统数据定期备份等部分功能，这些都有待后期的进一步改进。

本系统基于B/S架构体系进行开发，采用Java语言进行编写，结合JSP技术实现基于Web的学生成绩管理系统的功能，整个系统的开发结构是按照软件工程设计思想进行，主要有可行性分析、需求分析、概要设计、详细设计、软件测试和系统实现，该系统总体上达到了整个系统功能需求，其中包括学生信息管理、成绩管理、班级管理、课程管理等，实现了以Internet网络为核心，通过数据库对采集的学生信息进行及时高效的管理。

系统总体上来说功能需求初步实现，系统层次分明，结构还算清晰，当然，该系统还有些不尽人意的地方，比如进一步优化DIV+CSS样式设计，改善用户界面，增强用户的使用体验度；使用更高效的系统开发平台；优化出错处理方案，增加系统数据定期备份等部分功能，这些都有待后期的进一步改进。

# **参 考 文 献**

[1]余华凌.基于Web学生信息管理系统的设计与实现[J].江西科学，2005，23（6）：755.

[2]曾惠芳.试论高校学生管理工作信息化存在的问题及解决方案[J].甘肃科技，2006，22

[3]马秀麟,郏示德.管理信息系统及其开发技术[M].北京：人民邮电出版社，2005:16.

[4]周光明,徐琳.基于b/s的学生信息管理系统[J].电脑与信息技术，2004,2:30－32.

[5]张志斌,朱小军.基于Web的学生管理系统的开发探究[J].兰州城市学院信息工程学院，2009，14卷第5期.

[6]张海藩,倪宁编著.软件工程[M].北京：人民邮电出版社，2010.

[7]常华.学生信息管理系统设计方法[J].辽宁工学院学报，2004,24(3)：28－29.

[8]耿建敏,吴文国编著.软件工程[M].北京：清华大学出版社出版，2009.

[9]王虎,张俊.管理信息系统[M].武汉：武汉理工大学出版社，2004.7.

[10]陈轶,姚晓昆编著.Java程序设计实验指导[M]..北京：清华大学出版社，2006.

[11]计磊,李里,周伟.J2EE整合应用案例[M].北京：人民邮电出版社，2007.

[12]Ian Sommerville.软件工程[M].北京：机械工业出版社，2003.

[13]数据库百度百科.<http://baike.baidu.com/view/1088.htm>.

[14]方振宇.Java Web从初学到精通[M].北京：电子工业出版社，2010.

[15] Walter Savitch.Absolute Java[M].北京：电子工业出版社，2005.

[16]徐明华编著.JSP基础与案例开发详解[M].北京：清华大学出版社，2009.

[17]盛仲飙.JSP中数据库访问的性能优化研究[M].微型电脑应用，2008.

[18]王珊,萨师煊编著.数据库系统概论[M].北京：高等教育出版社，2006.

[19]BruceEckel.JSP编程思想[M].北京：机械工业出版社，2003.

[20]Schwalbe .K..王金玉.IT项目管理[M].北京:机械工业出版社，2002.

# **致 谢**

首先向我的导师潘操老师表示衷心的感谢。本文是在潘老师的悉心指导和严格要求下完成的。我为有幸作为潘老师的学生感到非常的高兴，潘老师在我刚进大学的时候上了一堂计算机学业规划科课，老师严谨的治学态度和广博的知识使我印象深刻，指引着我学习道路上的方向，在此次设计中我也遇到了一些困难，期间，潘老师不但给予我很大的鼓励，而且在设计上提供了很多正确的建议，在他的指导下，我不仅学到了扎实的专业知识，也在老师身上学到了积极热情的态度。至此，我再一次向潘老师表示最诚挚的感谢！

在论文完成过程中，我还得到了授课老师和同窗同学的热心帮助，在此也非常感谢老师、同学在学习和生活上给予的指导和帮助。在以后的工作学习中，虽然没有了学校和老师的无微不至的照顾，我也要继续不断努力学习新的知识与技术，不辱学校和老师的悉心栽培，争取为母校增光添彩。我还要感谢我的母校，它为我提供了良好的学习环境和生活环境，让我的大学生活丰富多彩。

最后，向评审我论文的各位专家、老师以及本文所引用文献的各位作者表示衷心的谢意！