



闽江学院

本科毕业论文（设计）

题 目	人脸识别考勤系统
学 生 姓 名	施铭锴
学 号	3207103138
学 院	计算机与大数据学院
年 级	2020
专 业	软件工程（闽台合作）
指 导 教 师	曹永忠
职 称	教授
完 成 日 期	

闽江学院毕业论文（设计）诚信声明书

本人郑重声明：

兹提交的毕业论文（设计）人脸识别考勤系统，是本人在指导老师 曹永忠 的指导下独立研究、撰写的成果；论文（设计）未剽窃、抄袭他人的学术观点、思想和成果，未篡改研究数据，论文（设计）中所引用的文字、研究成果均已在论文（设计）中以明确的方式标明；在毕业论文（设计）工作过程中，本人恪守学术规范，遵守学校有关规定，依法享有和承担由此论文（设计）产生的权利和责任。

声明人（签名）：

2024年4月28日

摘 要

随着社会的发展，大学校园也在不断追求智能化、信息化的管理方式，基于人脸识别的考勤签到系统正是这种发展趋势的体现。它不仅符合现代学校管理的需求，也为学生提供了更便利、高效的考勤方式，对于推动大学校园管理水平的提升和现代化建设具有积极的意义。此外，基于人脸识别的考勤系统还可以提供大量的考勤数据，学校可以通过对这些数据的分析，更好地了解学生的出勤情况和学习习惯，为学校管理和教学改进提供参考依据。这对于提升学校管理效率和教学质量具有重要意义。

在大学校园中，基于人脸识别的考勤签到系统具有重要意义和广泛应用前景[1]。这种系统可以极大地提高学生考勤的效率和准确性，避免了传统考勤方式中可能出现的作弊和误差问题。同时，基于人脸识别的考勤系统也能够改善学生的考勤体验，无需排队等候，只需站在设备前即可完成签到，提高了签到的便利性和效率。基于人脸识别的考勤系统还可以提供大量的考勤数据，学校可以通过对这些数据的分析，更好地了解学生的出勤情况和学习习惯，为学校管理和教学改进提供参考依据。这对于提升学校管理效率和教学质量具有重要意义。

所以为了满足上述需求该系统采用Java语言作为开发基础，利用SpringBoot、Vue等框架，微信小程序进行开发。数据库则是采用主流的Mysql作为存储数据库。详细介绍Unity引擎的使用方法，游戏需求分析，以及游戏功能实现和测试的整个开发流程。

关键词：智能化；人脸识别；考勤；SpringBoot；Vue；微信小程序

Abstract

With the development of society, university campuses are continuously pursuing intelligent and digitized management methods. The attendance system based on facial recognition is a manifestation of this trend. It not only meets the demands of modern school management but also provides students with a more convenient and efficient way of attendance. It holds positive significance for enhancing and modernizing the management level of university campuses. Additionally, the attendance system based on facial recognition can provide a large amount of attendance data. Schools can analyze these data to better understand students' attendance and study habits, providing reference for school management and teaching improvement. This is of great importance for improving school management efficiency and teaching quality.

In university campuses, the attendance system based on facial recognition holds significant importance and broad application prospects. This system can greatly improve the efficiency and accuracy of student attendance, avoiding cheating and error problems that may occur in traditional attendance methods. At the same time, the attendance system based on facial recognition can also enhance students' attendance experience. Without the need to queue, students can simply stand in front of the device to complete the check-in, thus improving the convenience and efficiency of attendance. Additionally, this system can provide a large amount of attendance data, which schools can analyze to better understand students' attendance and study habits, providing reference for school management and teaching improvement. This is crucial for improving school management efficiency and teaching quality.

Therefore, to meet the above requirements, the system adopts Java language as the development foundation, utilizing frameworks such as SpringBoot, Vue, and WeChat Mini Program for development. The database used is the mainstream MySQL as the storage database.

Key words: Intelligentization; facial recognition; attendance; SpringBoot; Vue; WeChat Mini Program

目录

目录.....	5
图目录.....	7
表目录.....	9
1 绪论.....	- 1 -
1.1 研究背景.....	- 1 -
1.2 研究目的.....	- 1 -
1.3 研究限制.....	- 2 -
1.4 论文结构.....	- 3 -
2 相关文献综述.....	- 5 -
2.1 开发技术选择.....	- 5 -
2.2 人脸识别技术选择.....	- 12 -
2.3 国内外相关研究情况.....	- 13 -
3 系统分析与功能设计.....	- 15 -
3.1 系统需求分析.....	- 15 -
3.2 系统功能设计.....	- 15 -
3.2.1 角色功能设计.....	- 17 -
3.3 数据库设计.....	- 19 -
4 系统实现.....	- 29 -
4.1 前台功能实现.....	- 29 -
4.2 后台功能实现.....	- 35 -
5 系统展示.....	- 45 -
5.1 系统功能展示.....	- 45 -
6 总结与展望.....	- 60 -

6.1 结论	- 60 -
参考文献	- 63 -
致谢	- 65 -

图目录

图 1	Java SE 的组成概念图	- 5 -
图 2	Vue 前端架构设计图	- 6 -
图 3	微信小程序架构图	- 7 -
图 4	常用 SpringBoot 系统框架图	- 9 -
图 5	使用 Maven 进行依赖管理的生命周期	- 9 -
图 6	Mybatis 架构图	- 10 -
图 7	数据库管理系统	- 11 -
图 8	Redis 单机模式架构图	- 12 -
图 9	阿里云人脸识别	- 13 -
图 10	前台业务流程图	- 16 -
图 11	后台业务流程图	- 16 -
图 12	用户用例图	- 17 -
图 13	教师用例图	- 18 -
图 14	功能组织架构图	- 18 -
图 15	数据库主要 ER 图	- 28 -
图 16	小程序登录功能	- 29 -
图 17	个人信息编辑功能	- 30 -
图 18	定位签到功能	- 31 -
图 19	人脸上传	- 32 -
图 20	人脸识别	- 33 -
图 21	普通签到	- 34 -
图 22	签到请假	- 35 -
图 23	后台登录	- 36 -
图 24	新建班级	- 36 -
图 25	修改班级	- 37 -
图 26	删除班级	- 37 -
图 27	新增课程	- 38 -
图 28	修改课程	- 38 -
图 29	删除课程	- 39 -
图 30	新建签到	- 39 -
图 31	修改签到	- 40 -
图 32	删除签到	- 40 -
图 33	新建用户	- 41 -
图 34	修改用户	- 41 -
图 35	删除用户	- 42 -
图 36	后台新增人脸	- 42 -
图 37	后台删除人脸	- 43 -
图 38	新建签到信息	- 43 -
图 39	修改签到信息	- 44 -
图 40	删除签到信息	- 44 -
图 41	查看各班出勤率	- 45 -
图 42	小程序登录功能演示	- 46 -
图 43	修改个人信息演示	- 46 -

图 44	定位签到演示	- 47 -
图 45	人脸签到演示	- 47 -
图 46	人脸录入展示	- 48 -
图 47	普通签到演示	- 48 -
图 48	签到请假演示	- 49 -
图 49	查看请假信息	- 50 -
图 50	后台登陆演示	- 50 -
图 51	新建班级信息演示	- 51 -
图 52	修改班级信息演示	- 51 -
图 53	删除班级信息演示	- 52 -
图 54	新建课程演示	- 52 -
图 55	修改课程演示	- 53 -
图 56	删除课程演示	- 53 -
图 57	发起签到演示	- 54 -
图 58	修改签到演示	- 54 -
图 59	删除签到演示	- 55 -
图 60	新建用户演示	- 55 -
图 61	修改用户演示	- 56 -
图 62	删除用户演示	- 56 -
图 63	后台新增人脸演示	- 57 -
图 64	后台删除人脸演示	- 57 -
图 65	新建签到信息演示	- 58 -
图 66	修改签到信息演示	- 58 -
图 67	删除签到信息演示	- 59 -
图 68	查看各班出勤率演示	- 59 -

表目录

表 1	微信用户信息表	- 19 -
表 2	学生人脸信息表	- 20 -
表 3	签到信息表 1	- 21 -
表 4	签到信息表 2	- 23 -
表 5	课题信息表	- 24 -
表 6	系统角色表	- 24 -
表 7	系统用户表	- 25 -
表 8	系统用户-角色表	- 26 -

1 绪论

1.1 研究背景

研究背景：

随着社会的发展，人们对于工作学习效率和管理效率的要求越来越高，基于人脸识别的考勤签到系统正是符合这一需求的创新解决方案。基于信息技术的不断发展和普及，基于人脸识别的考勤系统在近年来得到了广泛关注和应用。这种系统以其高效、准确的特点，逐渐取代传统的考勤方式，在各行各业都得到了应用，尤其在大学校园中更是发挥着重要作用。

基于人脸识别的考勤系统利用人脸识别技术，可以实现对员工或学生的自动识别和签到，不仅提高了考勤的效率和准确性，还避免了传统考勤方式中可能出现的作弊和误差问题。例如，在大学校园中，学生可以通过站在人脸识别设备前完成签到，无需排队等候，提高了签到的便利性和效率。同时，基于人脸识别的考勤系统还可以提供大量的考勤数据，学校可以通过对这些数据的分析，更好地了解学生的出勤情况和学习习惯，为学校管理和教学改进提供参考依据。这对于提升学校管理效率和教学质量具有重要意义。

另外，基于人脸识别的考勤系统也符合现代学校管理的智能化、信息化需求，推动了学校管理水平的提升和现代化建设。随着这种系统不仅可以提高企业和学校的考勤效率和准确性，还能够改善员工和学生的工作学习体验，提升企业和学校的管理效率。随着社会的发展，人们对于工作学习效率和管理效率的要求越来越高，基于人脸识别的考勤签到系统正是符合这一需求的创新解决方案。

1.2 研究目的

在传统的学校考勤管理中，使用刷卡式打卡系统是一种常见方式。然而，随着科技的进步，人脸识别打卡系统作为一种新兴的考勤管理工具，逐渐受到了广泛关注。本部分旨在探讨人脸识别打卡系统相较于传统刷卡式打卡系统的优势，并分析其在学校考勤管理中的应用价值。

识别度更高：人脸识别技术通过识别学生的面部特征进行打卡记录[2]。这

种技术的识别度通常比刷卡式打卡更高。由于每个人的面部特征都是独一无二的，人脸识别系统能够更准确地确认打卡人员的身份，从而大大减少了误判的可能性。这种高识别度可以有效地避免因识别错误而产生的考勤数据不准确的问题，提高了考勤管理的准确性和可靠性。

防止代打卡：人脸识别打卡系统可以有效防止代打卡现象的发生。在刷卡式打卡系统中，卡片可能会被他人盗用或借用，从而导致代打卡的风险。而人脸识别打卡系统只会识别已注册人员的面部特征，不是本人的面部无法通过验证进行打卡，因此可以有效地防止代打卡现象的发生，保障了考勤数据的真实性和可信度。

无需携带卡片：人脸识别打卡系统无需学生携带任何卡片，只需通过面部识别即可完成打卡。这大大减少了学生因忘记携带卡片或卡片丢失而无法打卡的困扰，提高了打卡的便捷性和灵活性。学生只需简单地面对识别设备进行识别，就能完成打卡，极大地节省了时间和精力。

可追溯性：人脸识别打卡系统可以记录每位学生的打卡记录，包括打卡时间、地点等信息。这为管理人员提供了更加详细和准确的考勤数据，便于进行后续的考勤管理和数据分析。管理人员可以随时查看学生的打卡情况，及时发现异常情况并进行处理，提高了考勤管理的效率和及时性。

节省时间：人脸识别打卡系统节约了大量的排队等待打卡的时间。由于无需接触和携带卡片，学生只需简单地通过面部识别即可完成打卡，大大提高了打卡的效率和便捷性。这对于学生来说是一种更加高效和便捷的打卡方式，有助于提升学生的学习和工作效率。

节约成本：人脸识别打卡系统节约了大量硬件成本和人力成本。无需额外的硬件设备例如打卡机和打卡卡片，也不需要额外的人力去维护打卡机等设备，因此能够节省大量的硬件成本和管理成本。

综上所述，人脸识别打卡系统相较于传统刷卡式打卡系统具有识别度更高、防止代打卡、无需携带卡片、可追溯性、节省时间和节约成本的优势。因此，在学校考勤管理中推广应用人脸识别打卡系统具有重要的现实意义和应用价值。

1.3 研究限制

时间限制：作为学生，时间有限，从头开始编写一个完整的人脸识别算法需要大量的时间和精力。尤其是需要考虑算法的设计、实现、调试和优化等多个阶

段，这些过程需要深入的理论基础和实践经验，因此时间上存在限制。

技术难度：人脸识别算法涉及复杂的图像处理和机器学习技术，对于学生来说，需要掌握的理论知识和编程技能较多。所以需要借助现有的开源算法或库。

资源限制：学生在编写算法时可能面临硬件和软件资源的限制。例如，需要高性能的计算机、GPU 加速、大量的数据集等资源来进行算法训练和测试，这对于学生来说可能不容易获得。

微信小程序上架费用：将微信小程序上架需要支付一定的费用，包括开发者认证费用和应用审核费用等。对于学生来说，存在经济上的限制，无法轻易支付这些费用。

实践环境限制：学校或个人的实践环境可能不具备搭建和部署微信小程序所需的条件。例如，需要搭建服务器、配置数据库、编写后台接口等，这对于学生来说可能需要额外的资源和技术支持。

法律和隐私限制：人脸识别技术涉及到用户的个人隐私和数据安全，需要遵守相关的法律法规和隐私政策。学生在研究中需要考虑到这些法律和道德问题，可能需要额外的时间和精力来保证系统的合法合规性。

综上所述，尽管有意愿进行人脸识别算法的研究和微信小程序的开发，但作为学生，面临着时间、技术、资源、经济和法律等多方面的限制，这些限制将对研究和开发过程带来一定的挑战。

1.4 论文结构

本文以人脸识别考勤系统为主体，从需求分析、系统设计、功能实现和系统测试这些方面对人脸识别考勤系统阐述。

本文分为六个章节：

第一章为绪论，主要介绍人脸识别考勤系统的研究背景、意义及国内外的研究情况。

第二章为相关技术介绍，主要介绍人脸识别考勤系统和人脸识别技术选择及其对应的基础理论。

第三章为系统需求分析，系统功能设计和数据库设计，其中包含业务流程图、功能模块设计图、数据表 and ER 关系图。

第四章为系统实现，主要介绍前台功能和后台功能的具体实现。

第五章为系统展示，主要对系统的功能进行展示。

第六章为总结以及展望，主要对人脸识别考勤系统的完成进行总结，以及对人脸识别考勤系统的展望。

2 相关文献综述

2.1 开发技术选择

(1) Java

Java 作为一种稳定、安全、跨平台、高性能的后端语言，非常适合用于支持基于人脸识别的考勤系统的后端开发。其丰富的生态系统和易学易用的特性能够提高系统的开发效率和可靠性，保证系统的稳定运行和良好的用户体验。经过多年发展，具有良好的稳定性和安全性，能够保证系统的稳定运行和数据安全。具有跨平台特性，可以在不同操作系统上运行，提高了系统的适用性和可移植性[3]。拥有庞大的开发者社区和丰富的生态系统，有大量的框架和工具可供选择，能够加速系统的开发和部署。因此，本人脸识别考勤系统选择使用 Java 语言是非常合适的。

图 1 Java SE 的组成概念图



图 1 Java SE 的组成概念图

(2) Vue

人脸识别的考勤系统的可行性分析显示，这种系统在技术方面具备一定的成熟度，需要一个稳定、高效的前端框架来支持其用户界面的设计和实现。在这种情况下，Vue 框架作为一种现代化、轻量级，灵活性强、易学易用、性能优越的前端框架，非常适合用于支持基于人脸识别的考勤系统的用户界面设计和实现。其优秀的性能和易用性将有助于提高系统的开发效率和用户体验，符合系统设计和实现的需求。其用了组件化的开发方式，能够更好地管理和复用代码，这对于

一个复杂的系统如基于人脸识别的考勤系统尤为重要。支持响应式设计，能够根据不同设备的屏幕大小和分辨率自动调整页面布局，保证用户界面的良好显示效果。学习曲线较为平缓，开发者可以快速上手，快速构建出符合需求的用户界面，提高开发效率。具有优秀的性能表现，能够有效减少页面渲染时的性能开销，保证系统的流畅性和稳定性。拥有庞大的开发者社区，有大量的插件和工具可供开发者使用，能够快速解决开发中遇到的问题。

图 2 Vue 前端架构设计图

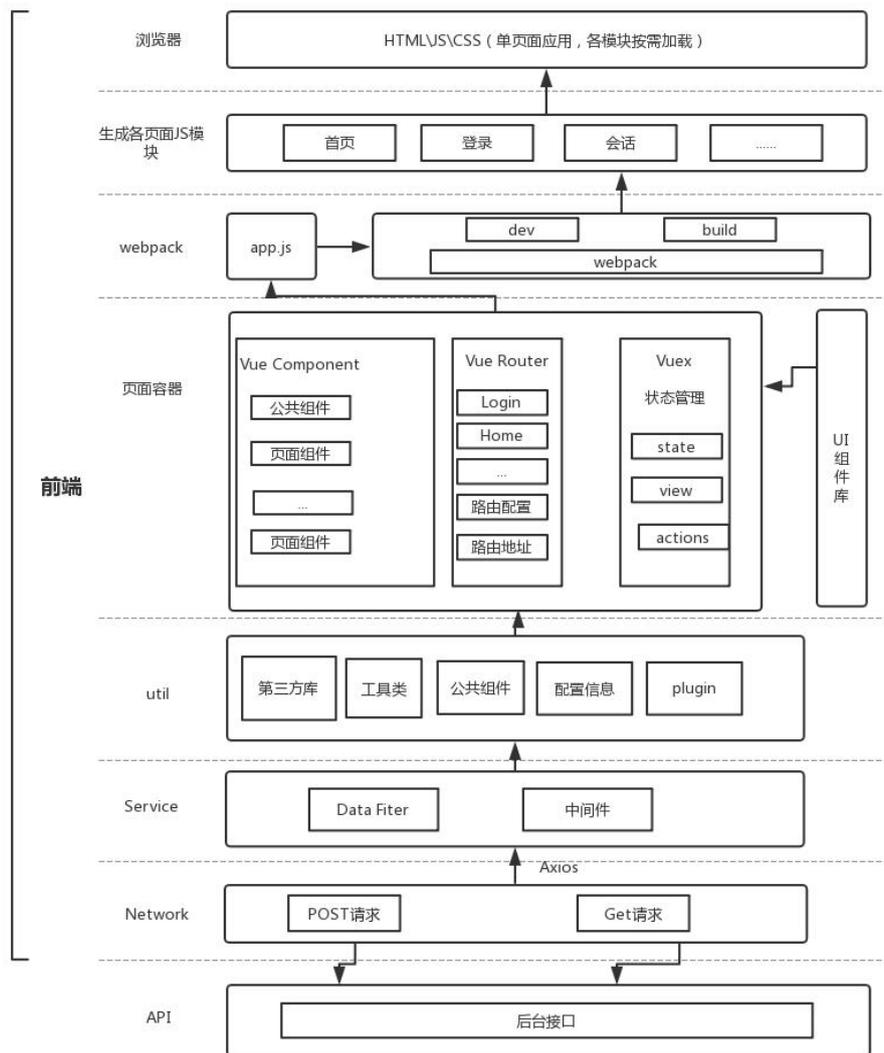


图 2 Vue 前端架构设计图

(3) 微信小程序

小程序框架作为一种快速开发、跨平台支持、低成本的前端框架，非常适合用于支持基于人脸识别的考勤系统的移动端用户界面设计和实现。其与 Vue 框架

具有良好的相性，能够很好地满足系统设计和实现的需求。与 Vue 框架相比，小程序框架具有良好的相性。两者都是轻量级、易学易用的前端框架，可以很好地结合使用。开发者可以利用 Vue 框架来构建基于人脸识别的考勤系统的管理后台，同时使用小程序框架来开发移动端的考勤签到小程序，两者能够很好地配合，提高开发效率和用户体验。小程序框架提供了丰富的组件和 API，开发者可以快速构建出符合需求的用户界面，快速迭代和发布。[14]小程序框架背靠腾讯，经过优化，具有良好的性能表现，能够保证系统的流畅性和稳定性。拥有庞大的开发者社区，有大量的插件和工具可供开发者使用，能够快速解决开发中遇到的问题。提高系统的适用性和覆盖面。

如图 3 微信小程序架构图所示为微信小程序的架构图，可以便于理解和编写微信小程序。

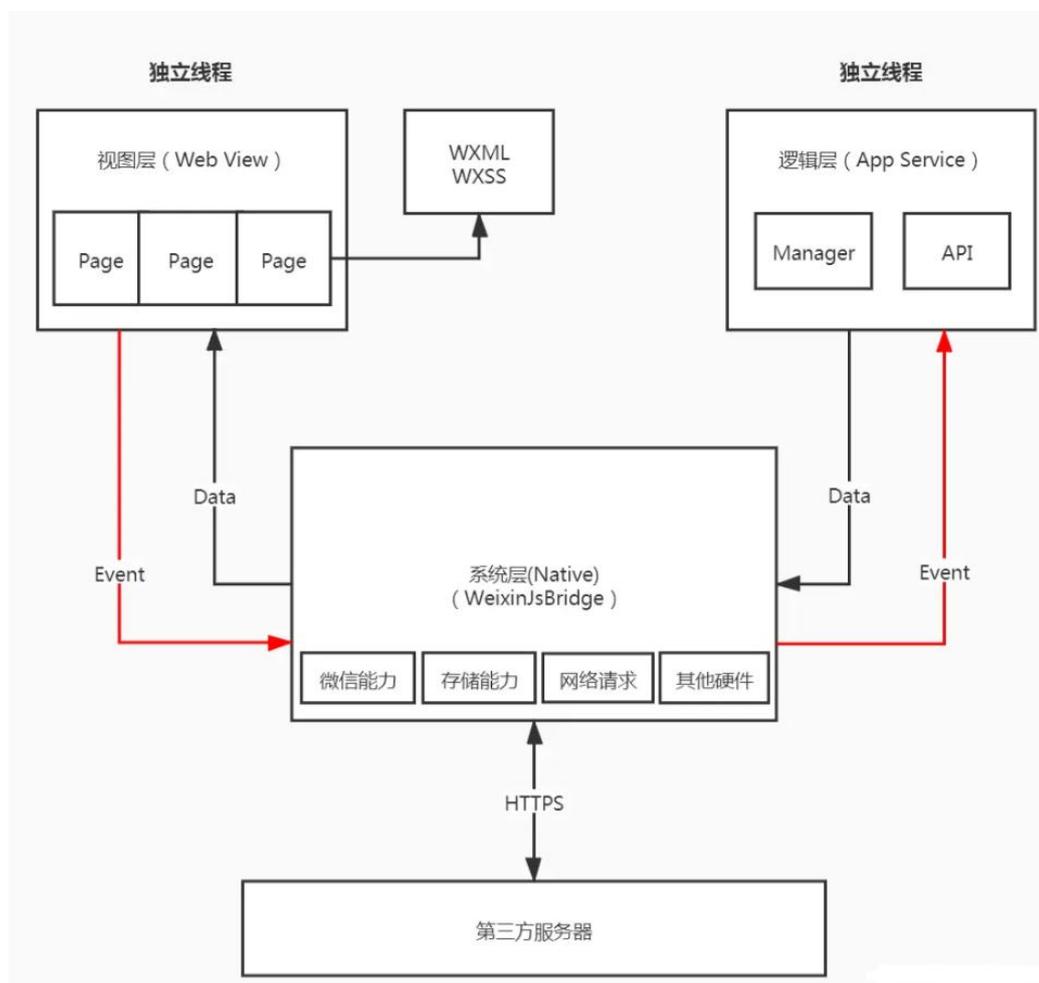


图 3 微信小程序架构图

(4) SpringBoot

Spring Boot 是一个基于 Spring Framework 的开发框架，旨在简化 Java 应用

程序的开发、部署和管理[4]。它提供了一种快速构建、易于扩展的方式，使得开发者能够更专注于业务逻辑的实现，而不必在配置文件上花费太多精力。SpringBoot 框架与 Java 语言的稳定性、安全性、快速开发性、可扩展性以及与前端框架的良好结合，能够确保考勤系统的稳定运行、安全性和良好的用户体验。并且 SpringBoot 框架简化了 Java 应用程序的开发过程，提供了许多开箱即用的功能和组件，可以加速系统的开发。同时，SpringBoot 框架与前端框架如 Vue 框架等良好结合，可以通过 RESTful API 等方式与前端进行交互，为用户提供优秀的体验。

由于 SpringBoot 框架与 Java 相结合的实用性得到广泛认可，所有在企业级应用开发领域得到了广泛应用，并有大量案例和实践经验可供参考。如图 4 常用 SpringBoot 系统框架图所示，这确保了考勤系统的质量和可靠性。因此，它们非常适合用于支持本人脸识别考勤系统的后端开发。

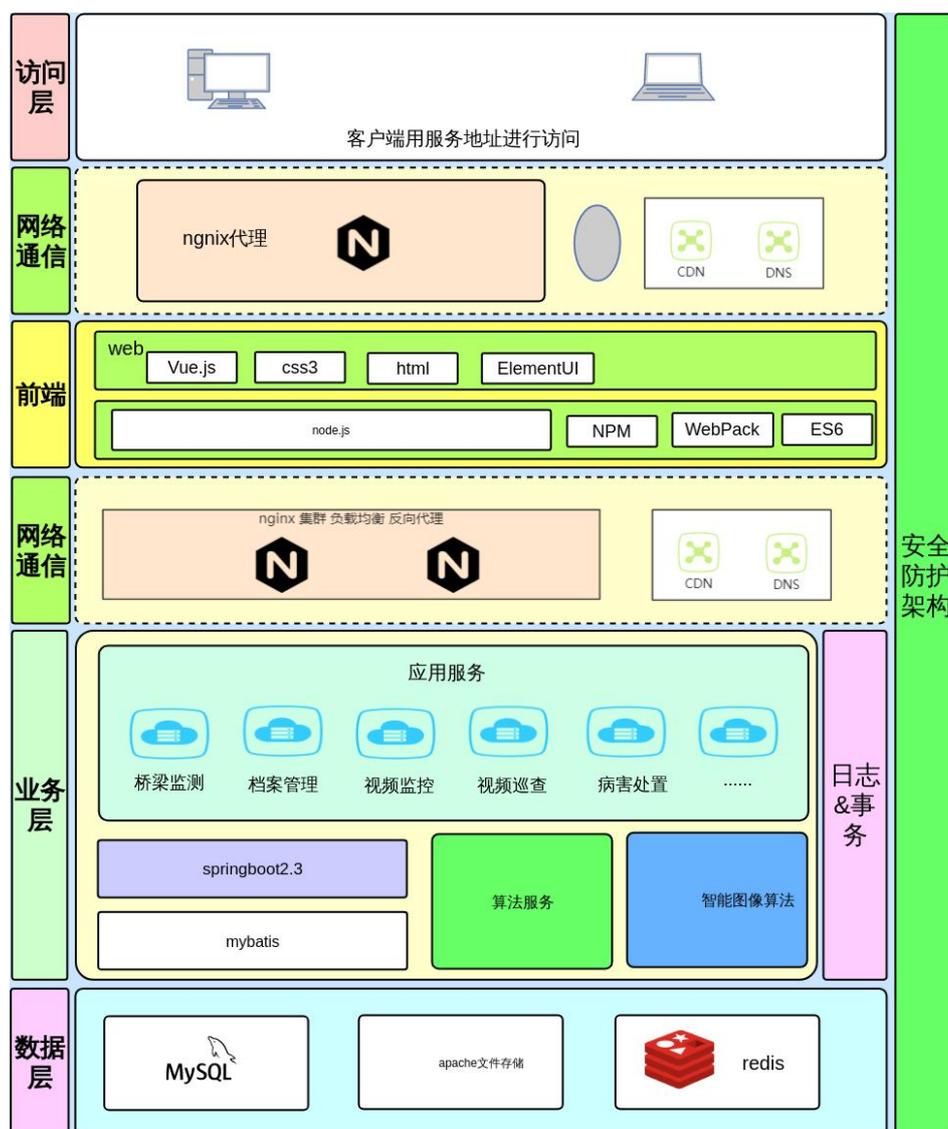


图 4 常用 SpringBoot 系统框架图

(5) Maven

基于人脸识别的考勤系统的开发和部署需要一个有效的项目管理工具，用以管理项目依赖并构建项目。Maven 是一个用于构建和管理 Java 项目的强大工具，它能够自动化项目的构建过程，管理项目依赖，以及生成项目文档等。作为 Java 开发中最流行的构建工具之一，Maven 极大地简化了项目的管理和维护，提高了开发效率。因此，作为一种优秀的项目管理工具，Maven 与 SpringBoot 框架具有良好的相性，而且 SpringBoot 本身就是基于 Maven 构建的，因此两者天生适配。Maven 能够精准地管理项目所需的各种依赖，包括第三方库、框架和插件，从而简化项目的配置和管理过程。同时，通过 Maven，开发者能够更加便捷地控制项目的构建流程，确保项目的稳定性和高效性。

如图 5 使用 Maven 进行依赖管理的生命周期所示，使用 Maven 对 SpringBoot 进行依赖管理的生命周期。

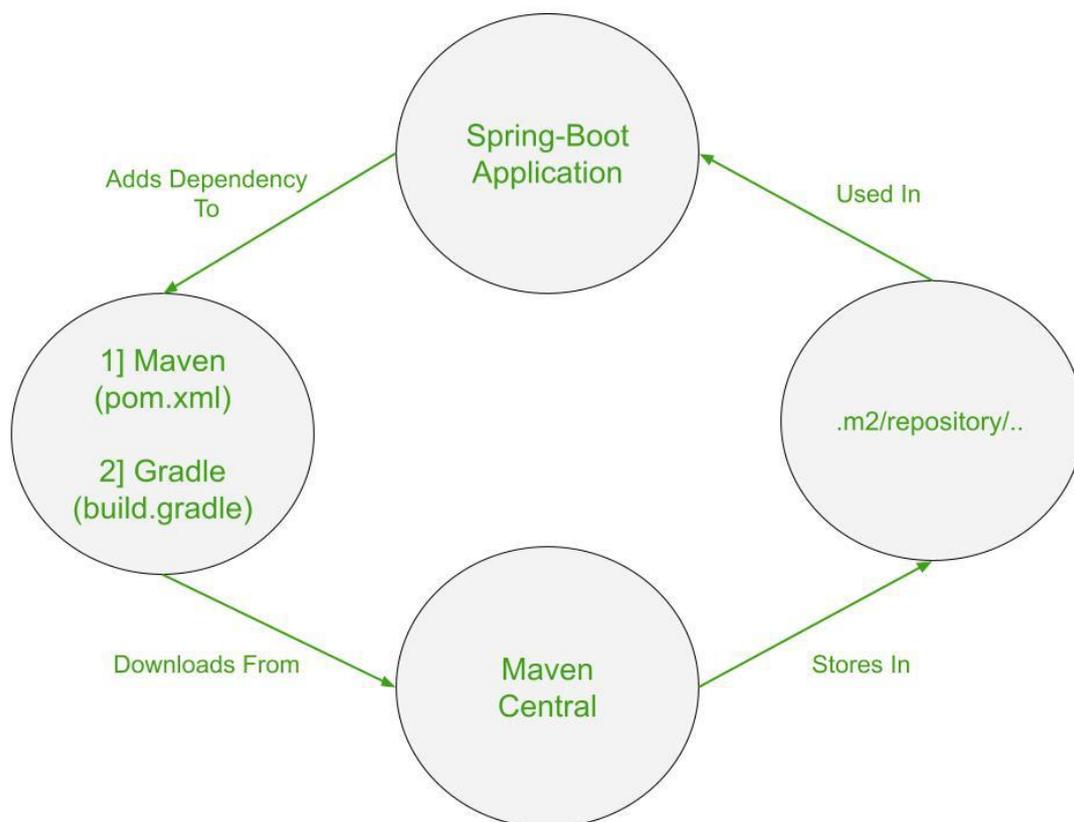


图 5 使用 Maven 进行依赖管理的生命周期

(6) Mybatis

MyBatis 是一个基于 Java 的持久层框架，它简化了数据库操作，并提供了灵

活的映射机制，使得开发者能够更容易地编写和维护数据访问层代码。MyBatis 将 SQL 语句与 Java 代码解耦，通过映射文件将数据库表中的记录映射为 Java 对象，从而实现了面向对象的数据库操作。Mybatis 作为一种优秀的持久化框架，与 SpringBoot 框架紧密结合，为人脸识别考勤系统的数据访问层开发提供了高效支持，其特性包括简化数据库操作、灵活性、高效性以及与 SpringBoot 的无缝集成。这些特性使开发者能够快速构建稳定且高效的考勤系统。Mybatis 通过 XML 或注解的方式提供了简洁的 SQL 映射配置，极大地简化了数据库操作，降低了开发的复杂度。开发者可以自定义 SQL 语句，以满足各种复杂查询和数据操作需求，这提供了灵活的 SQL 执行方式。此外，Mybatis 还提供了丰富的插件机制，满足了项目在数据访问层面的各种扩展需求。

如图 6 Mybatis 架构图

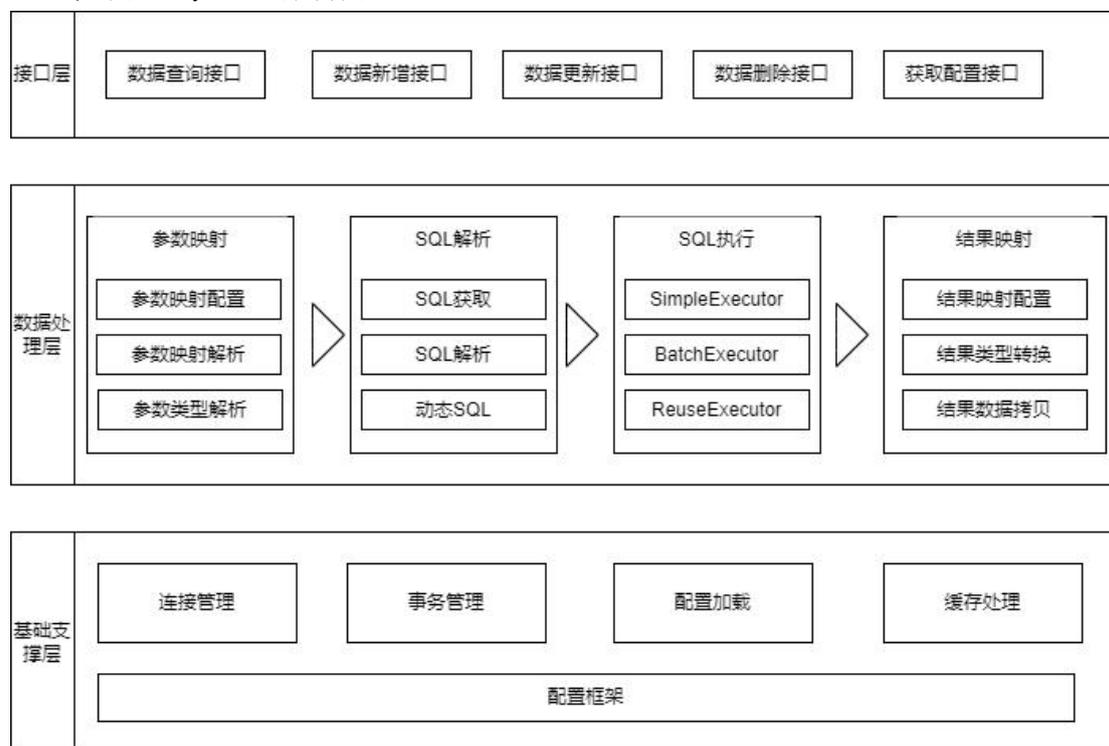


图 6 Mybatis 架构图

(7) MySQL

SQL (Structured Query Language) 是结构化查询语言，用于管理数据库管理系统，它可以帮助我们在庞大的数据中，快速找到想要的信息[5]。当我们在操作应用程序、网页或触发器时，数据会被 DBMS 数据库管理系统记录，并存放 Database 数据库中，接着通过 SQL 结构化查询语言向用户显示、提取所需的数据与信息，如图 7 数据库管理系统所示。

MySQL 也是是 DBMS 的一种。MySQL 是一个开源的关系型数据库管理系统，其稳定性、高性能、易用性等特点使其成为支持基于人脸识别的考勤系统的理想数据存储和管理解决方案[6]。MySQL 与 Java 框架的良好兼容性，为考勤系统的开发和运行提供了有效的支持。经过多年的发展和广泛应用，MySQL 在稳定性和可靠性方面表现卓越，可确保考勤系统数据的安全存储和可靠访问。其优秀的性能优化特性能够支持高并发访问和大规模数据存储，从而保证系统在高负载情况下的稳定性和出色的性能表现。MySQL 的这些特性使其成为考勤系统的首选数据库引擎，为系统的稳定运行和高效管理提供了可靠保障。

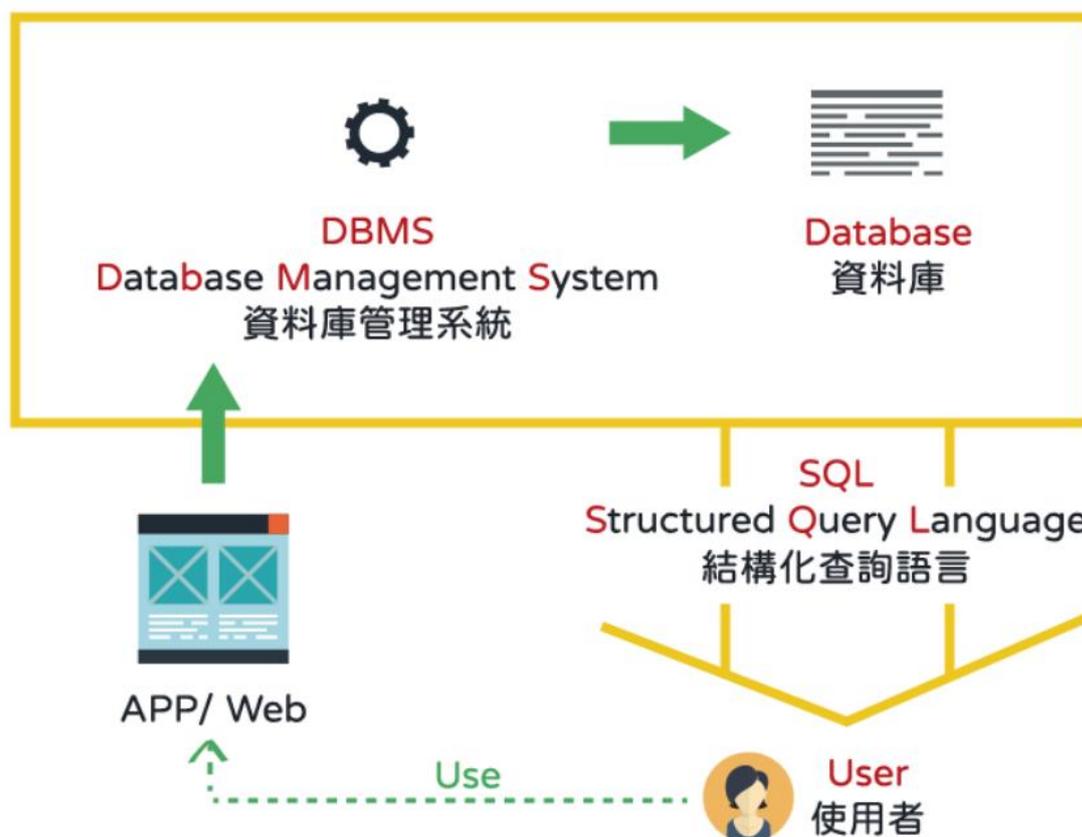


图 7 数据库管理系统

(8) Redis

Redis 作为一款高性能、拥有丰富数据结构以及提供持久化支持的内存数据库，具备极佳的适用性，可作为支撑基于人脸识别的考勤系统数据缓存与管理的理想选择。其作为内存数据库，数据存储于内存中，保证了读写操作的高速执行，有力满足系统高并发访问的要求。Redis 的卓越性能、多样化数据结构、可靠的持久化支持以及与 Java 和 MySQL 的无缝集成，为考勤系统的开发与运行提供了有力支持，实现了优异的整合与协作。以 Java 为例，通过 Redis 等客户端库连接

与操作 Redis 数据库，实现数据的高效缓存与管理；而 MySQL 与 Redis 的联合应用，进一步提升了系统性能与响应速度。

如图 8 Redis 单机模式架构图

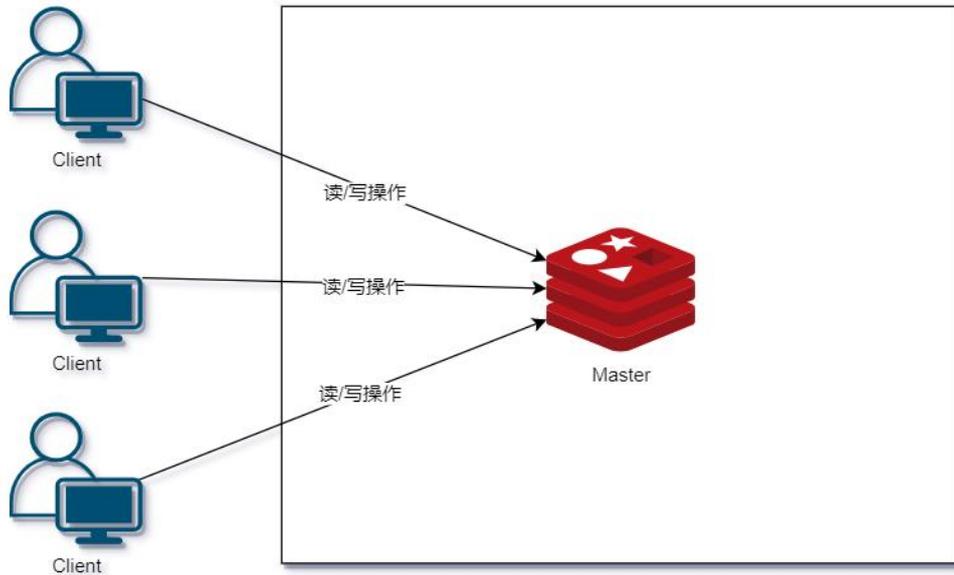


图 8 Redis 单机模式架构图

2.2 人脸识别技术选择

(1) 阿里云人脸识别

阿里云人脸识别 API 作为阿里云提供的一项重要服务，在当今数字化环境中扮演着至关重要的角色。这项服务以其出色的安全性、高度准确的识别能力以及成本效益高的特点而备受推崇。尤其是在支持基于人脸识别的考勤系统方面，它展现出了其无可比拟的优势。通过利用该 API，考勤系统得以确保用户的人脸数据得到安全存储和隐私保护，这对于维护个人信息安全至关重要。此外，该 API 所具备的高精度识别功能能够快速而准确地识别人脸信息，从而提高了考勤系统的准确性和效率。值得一提的是，该服务提供了按需付费和灵活的计费方式，使用户能够根据实际需求灵活调整费用，这进一步增强了其在实践中的可操作性和经济性。与自建人脸识别系统相比，利用阿里云人脸识别 API 不仅能够降低成本，还能够提高资源的有效利用率，从而实现了更加经济、高效的运营管理。

图 9 阿里云人脸识别



图 9 阿里云人脸识别

2.3 国内外相关研究情况

在中国，基于人脸识别的考勤系统在教育领域的应用具有重要背景和积极意义。中国教育系统庞大，学生数量众多，传统的考勤方式往往效率低下、易受作弊等问题影响。基于人脸识别的考勤系统可以有效解决这些问题，提高教育管理的效率和质量。

首先，中国教育系统中存在着考勤管理繁琐、耗时的问题。传统的考勤方式需要学生排队刷卡或签到，消耗了大量时间和人力资源。基于人脸识别的考勤系统可以实现自动识别和签到，节省了时间和精力，提高了考勤效率。

其次，中国教育系统中也存在着考勤数据准确性不高的问题。传统的考勤方式容易出现学生代签、漏签等情况，导致考勤数据的不准确。基于人脸识别的考勤系统具有高精度和安全性，可以减少这些问题，确保考勤数据的准确性和可靠性。

另外，基于人脸识别的考勤系统也符合中国教育现代化建设的需要。随着信息技术的不断发展，教育管理也在向智能化、信息化方向发展[7]。基于人脸识别的考勤系统可以推动教育管理的现代化进程，提高教育管理水平，符合中国教育现代化建设的战略目标。

中国，已经有许多学校和企业开始使用基于人脸识别的考勤系统，取得了显著的效果。例如，北京大学、清华大学等一些知名高校采用了这种系统，通过人脸识别技术实现学生的考勤签到，提高了考勤的准确性和效率。在企业方面，一些大型企业如阿里巴巴、腾讯等也开始尝试使用基于人脸识别的考勤系统，取代传统的考勤方式，提高了考勤管理的效率和便利性。

基于人脸识别的考勤系统在中国教育领域的应用具有重要背景和积极意义。通过提高考勤效率和准确性，推动教育管理的现代化进程，这种系统有望成为中

国教育管理的重要工具，为教育事业的发展做出积极贡献[1]。

3 系统分析与功能设计

3.1 系统需求分析

(1) 前台（用户端）：

前台主要包括学生用户模块、班级管理模块、请假管理模块、考勤签到模块和二维码签到模块。学生用户模块用于录入和管理学生基本信息，班级管理模块管理学生所属班级信息，请假管理模块提供学生请假功能，考勤签到模块结合人脸识别技术实现学生考勤签到，二维码签到模块教师发布二维码，学生扫码签到。学生定位签到，系统可以结合定位服务，例如 GPS 定位或 Wi-Fi 定位，实现学生在特定位置的签到功能。

(2) 后台（管理端）：

后台主要包括教师管理模块、教师模块，发布签到模块和可视化模块[8]。教师管理模块管理教师信息和职务，教师模块包括创建班级和分配学生功能，发布签到模块，可用于进行事实签到，增加签到的精确性和可靠性。可视化模块发布签到用于对考勤数据进行分析 and 统计，生成考勤报表，为学校提供完善的考勤管理解决方案。

3.2 系统功能设计

此系统采用前后端分离的形式开发的，并且分为用户前台、管理员后台。

其中用户前台业务流程图如图 10 前台业务流程图

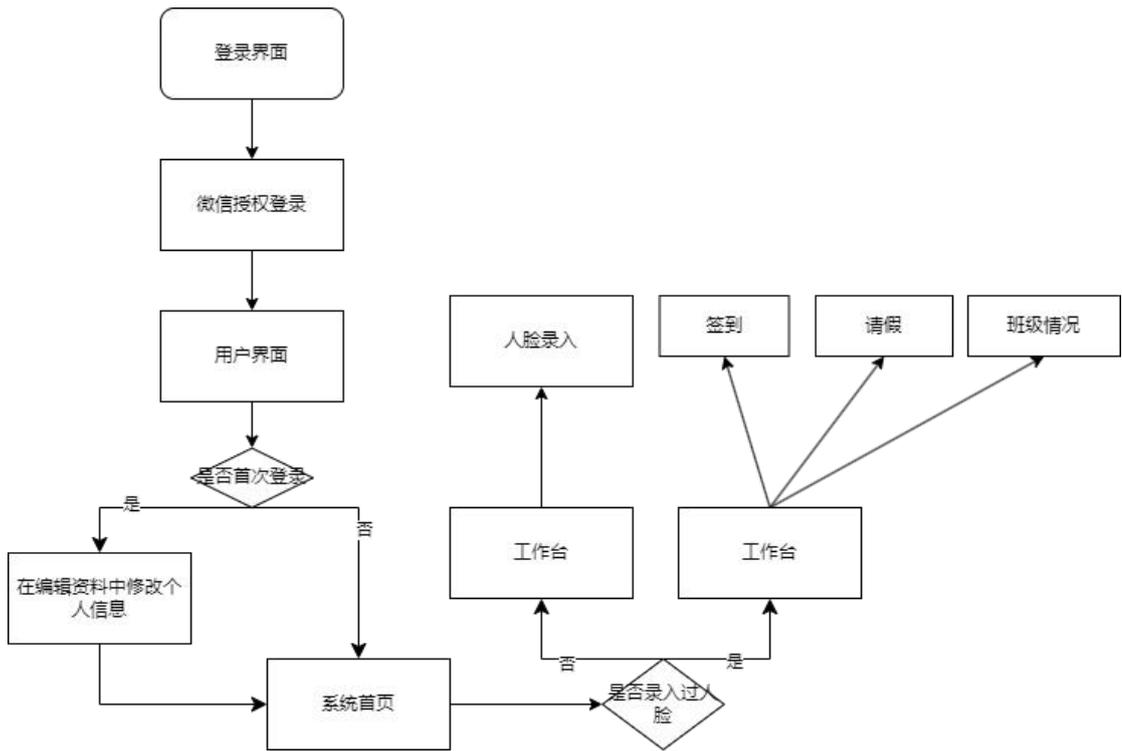


图 10 前台业务流程图

后台业务流程图如图 11 后台业务流程图

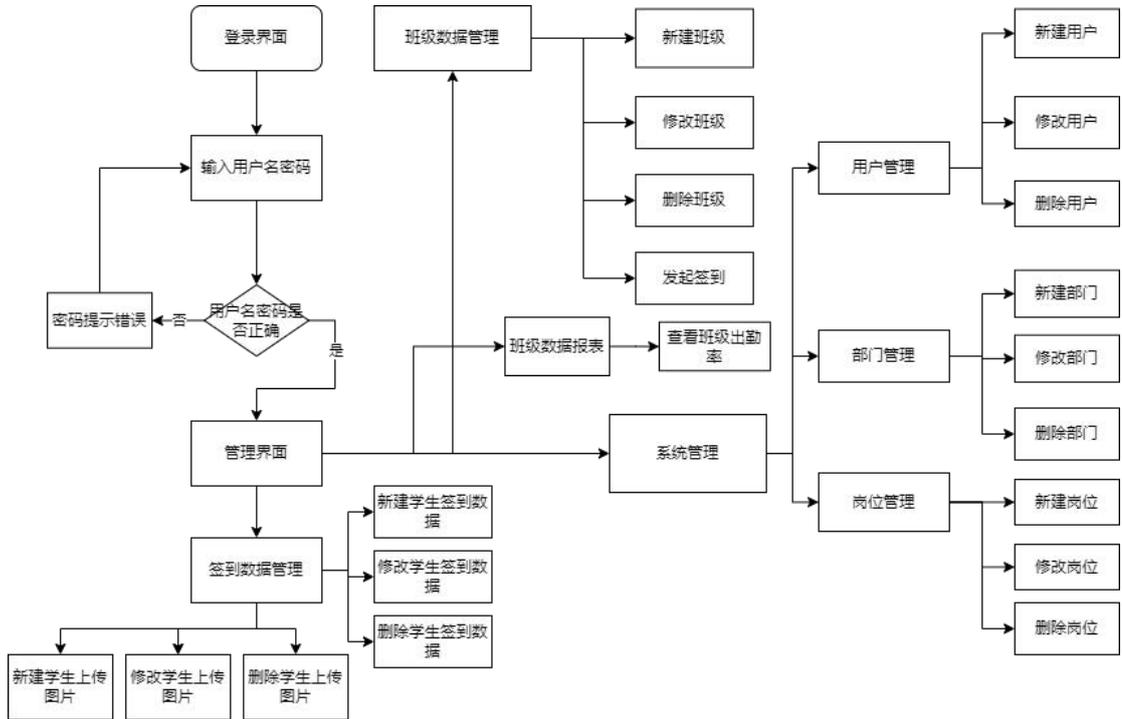


图 11 后台业务流程图

3.2.1 角色功能设计

其中用户的功能设计如图 12 用户用例图

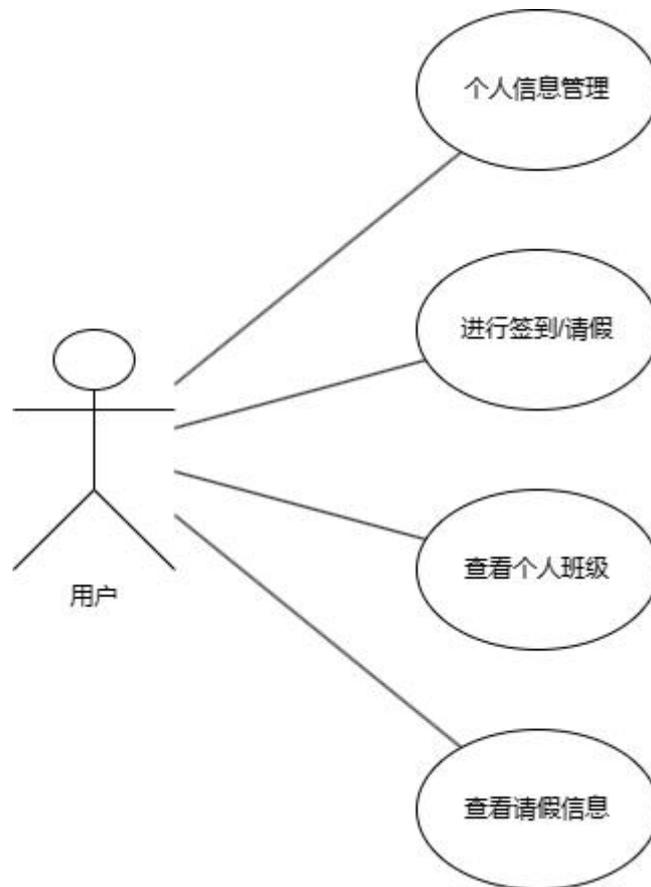


图 12 用户用例图

其中教师的功能设计如图 13 教师用例图

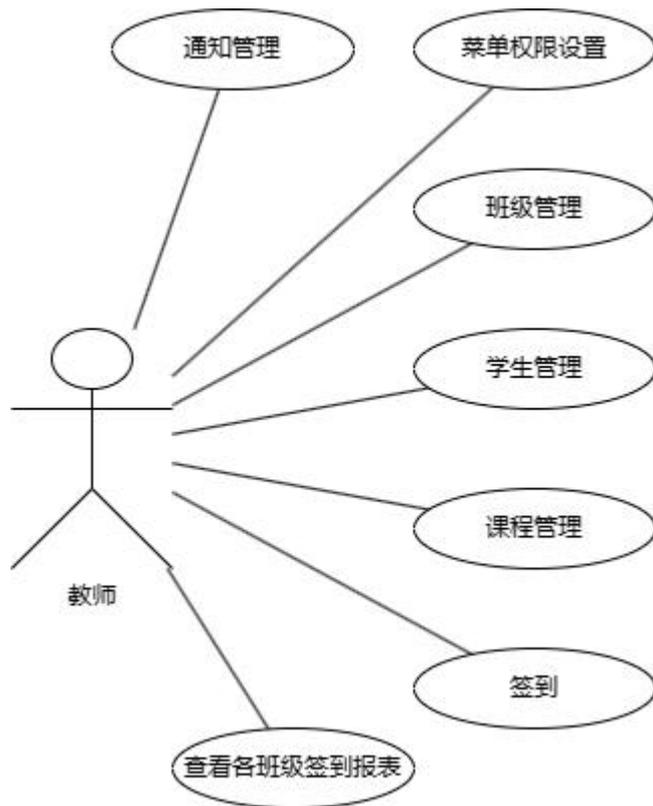


图 13 教师用例图

3.2.2 功能组织架构图

功能组织架构图如图 14 所示。

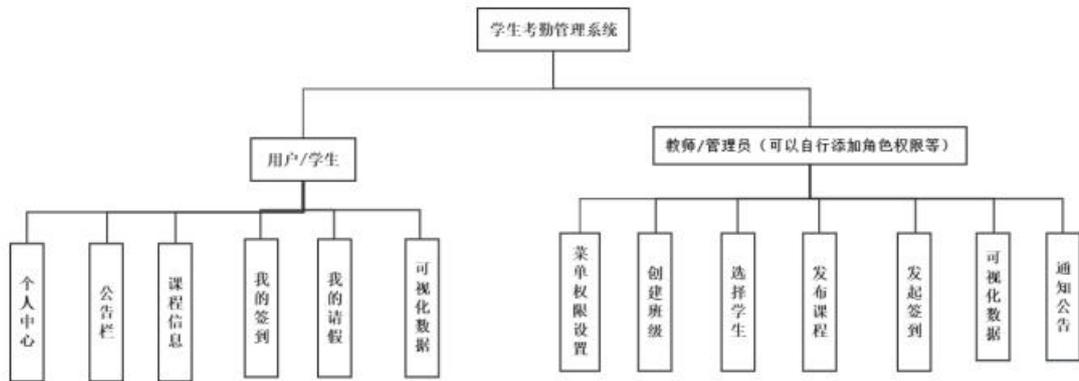


图 14 功能组织架构图

1. 学生用户模块：用于录入和管理学生的基本信息，包括姓名、学号、班级、性别等[9]。
2. 班级管理：用于管理学生所属的班级信息，包括班级名称、班级成员等。
3. 教师管理：对教师信息进行管理，包括教师的基本信息、职务等[10]。
4. 教师模块--创建班级--分配学生-
5. 请假管理：提供学生请假功能

6. 可视化模块 用于对考勤数据进行分析 and 统计，生成考勤报表，包括学生出勤率、请假次数、迟到次数等。
7. 二维码签到--教师发布二维码--学生扫码签到数据

3.3 数据库设计

(1) 数据库设计：

a) 微信用户信息表如表 1

表 1 微信用户信息表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
clazz_id	int		NO	Y		班级 ID
clazz_name	varchar	29	YES			班级名称
user_id	int		YES			用户 ID(学生)
user_name	varchar	255	YES			用户名称 (学生)
clazz_sum	varchar	24	YES			班级人数
teacher_id	int		YES			老师 ID
teacher_name	varchar	255	YES			老师姓名
status	char	1	YES		0	帐号状态 (0 正常 1 停用)
create_by	varchar	64	YES			创建者

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者
update_time	datetime		YES			更新时间
remark	varchar	500	YES			备注

b) 学生人脸信息表如表 2

表 2 学生人脸信息表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
image_id	int		NO	Y		图片 id
image_name	varchar	25	YES			图片关联学号
image_url	varchar	255	YES			图片路径
status	char	1	YES		0	帐号状态 (0 正常 1 停用)
create_by	varchar	64	YES			创建者
create_time	datetime		YES			创建时间
update_time	datetime		YES			更新时间

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
remark	varchar	500	YES			备注

c) 签到信息表如表 3

表 3 签到信息表 1

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
signin_id	int		NO	Y		签到 ID
signin_clazzid	int		YES			签到班级 id
signin_clazzname	varchar	25	YES			签到班级名称
signin_teachername	varchar	255	YES			签到老师的名称
signin_username	varchar	500	YES			签到的学生
signin_status	char	1	YES		0	学生签到状态 (0 失败默认 1 成功)

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
signin_type	varchar	25	YES			签到的类型（普通、二维码、位置）
signin_addimg	varchar	255	YES			签到具体类型实现(存放图片、地理位置数据)
signin_addimgreturn	varchar	120	YES			扫码返回的数据
signin_starttime	datetime		YES			发起签到时间
signin_endtime	datetime		YES			结束签到时间
status	char	1	YES		0	帐号状态(0正常 1停用)
create_by	varchar	64	YES			创建者

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者
update_time	datetime		YES			更新时间

d) 签到信息表 2 如表 4

表 4 签到信息表 2

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
wx_courseid	int		NO	Y		课程 id
wx_coursename	varchar	25	YES			课程名称
wx_course_code	varchar	25	YES			课程学分
wx_course_clazzid	varchar	100	YES			授课班级名称
wx_course_teacherid	int		YES			授课老师 id
wx_course_teachername	varchar	100	YES			授课老师姓名
create_by	varchar	64	YES			创建者
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
update_time	datetime		YES			更新时间
remark	varchar	500	YES			备注

e) 课程信息表如表 5

表 5 课题信息表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
clazz_id	int		NO	Y		班级 ID
clazz_name	varchar	29	YES			班级名称
user_id	int		YES			用户 ID(学生)
user_name	varchar	255	YES			用户名称(学生)
clazz_sum	varchar	24	YES			班级人数
teacher_id	int		YES			老师 ID
teacher_name	varchar	255	YES			老师姓名
status	char	1	YES		0	帐号状态 (0 正常 1 停用)
create_by	varchar	64	YES			创建者
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者
update_time	datetime		YES			更新时间
remark	varchar	500	YES			备注

f) 系统角色表如表 6

表 6 系统角色表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
-----	------	----	------	-------	-----	----

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
role_id	bigint		NO	Y		角色 ID
role_name	varchar	30	NO			角色名称
role_key	varchar	100	NO			角色权限字符串
role_sort	int		NO			显示顺序
data_scope	char	1	YES		1	数据范围 (1: 全部数据权限 2: 自定义数据权限 3: 本部门数据权限 4: 本部门及以下数据权限)
menu_check_strictly	tinyint		YES		1	菜单树选择项是否关联显示
dept_check_strictly	tinyint		YES		1	部门树选择项是否关联显示
status	char	1	NO			角色状态 (0 正常 1 停用)
del_flag	char	1	YES		0	删除标志 (0 代表存在 2 代表删除)
create_by	varchar	64	YES			创建者
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者
update_time	datetime		YES			更新时间
remark	varchar	500	YES			备注

g) 系统用户表如表 7

表 7 系统用户表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
user_id	bigint		NO	Y		用户 ID
dept_id	bigint		YES			部门 ID

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
user_name	varchar	55	NO			用户账号
nick_name	varchar	55	NO			用户昵称
user_type	varchar	2	YES		0	用户类型 (00 系统用户)
email	varchar	50	YES			用户邮箱
phonenumber	varchar	11	YES			手机号码
sex	char	1	YES		0	用户性别 (0 男 1 女 2 未知)
avatar	varchar	250	YES			头像地址
password	varchar	100	YES			密码
status	char	1	YES		0	帐号状态 (0 正常 1 停用)
del_flag	char	1	YES		0	删除标志 (0 代表存在 2 代表删除)
login_ip	varchar	128	YES			最后登录 IP
login_date	datetime		YES			最后登录时间
create_by	varchar	64	YES			创建者
create_time	datetime		YES			创建时间
update_by	varchar	64	YES			更新者
update_time	datetime		YES			更新时间
remark	varchar	500	YES			备注
open_id	varchar	50	YES			微信 open_id
union_id	varchar	50	YES			微信 union_id

h) 系统用户-角色表如表 8

表 8 系统用户-角色表

字段名	数据类型	长度	是否为空	是否为主键	默认值	备注
user_id	bigint		NO	Y		用户 ID
role_id	bigint		NO	Y		角色 ID

(2) ER 图：出数据库的主要 ER 图如图 15 数据库主要 ER 图

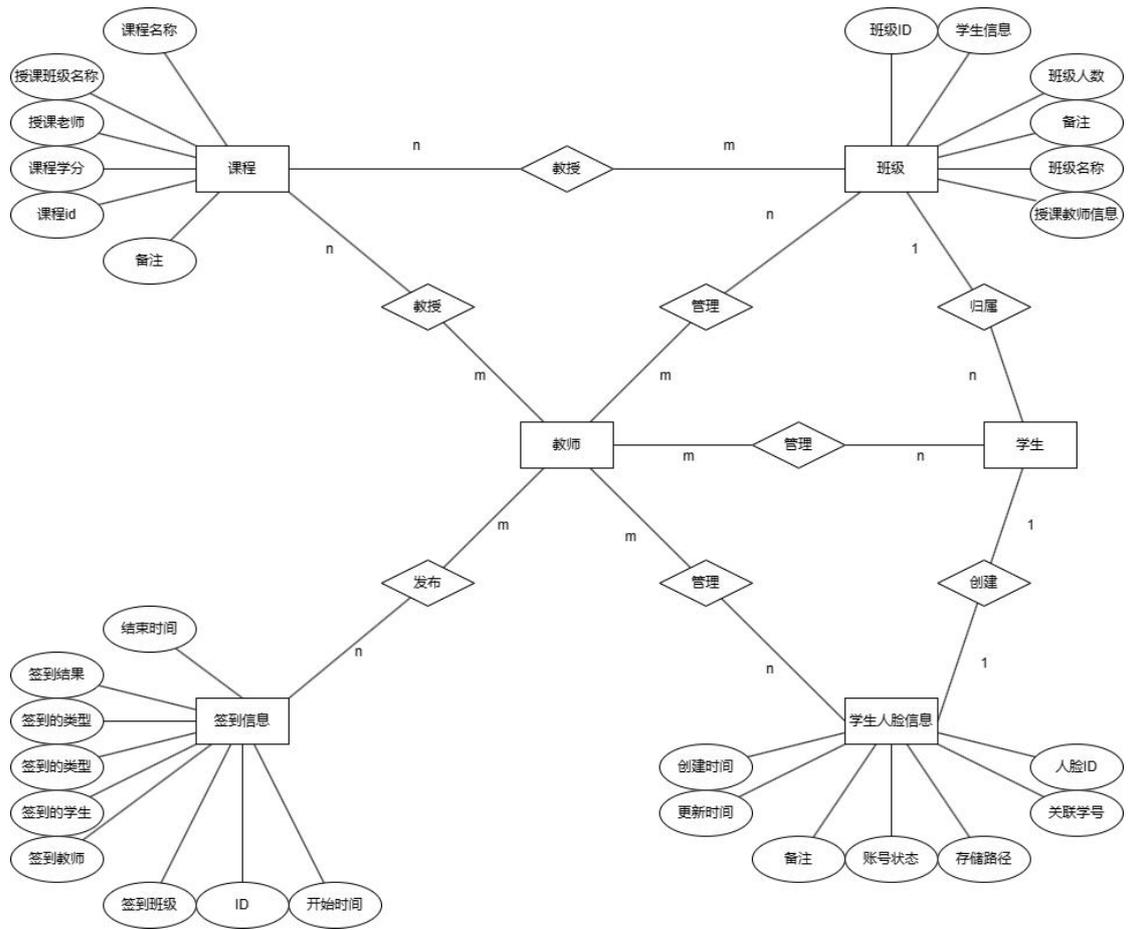


图 15 数据库主要 ER 图

4 系统实现

4.1 前台功能实现

(1) 小程序登录功能:

用户进入小程序后，点击登录，触发微信小程序的登录事件[11]，系统发起 POST 请求，将用户的数据发送到后端，后端通过对应接口接收到数据后，通过微信小程序自带的账号密码校验方法进行校验，登录成功则进入系统首页，登录失败返回错误信息以弹窗形式进行提示。

如图 16 小程序登录功能



图 16 小程序登录功能

(2) 个人信息编辑功能:

小程序发起修改个人信息请求，后端 Controller 控制层根据 url 进行接收，然后 WxUserinfoService.java 业务层进行业务处理，然后在持久层 (WxUserinfoMapper.java) 进行数据库修改，返回信息给前端，提示新建班级成

功。

如图 17 个人信息编辑功能



图 17 个人信息编辑功能

(3) 定位签到功能:

- 1.教师在后台 web 端发布签到任务，包括签到地点和有效时间等信息。
- 2.微信小程序接收到教师发布的签到任务，并显示在学生端的界面上。
- 3.学生打开微信小程序，在指定的时间签到。小程序 API 获取学生的位置信息。
- 4.然后发送给后台
- 5.在后台系统中记录学生的签到情况。小程序向学生反馈签到结果，显示签到成功或失败的提示。

如图 18 定位签到功能



图 18 定位签到功能

(4) 人脸文件上传:

1.准备阿里云 OSS 账号: 首先需要在阿里云注册账号, 并开通 OSS 服务, 获取 AccessKey ID 和 AccessKey Secret 等认证信息[12]。

2.集成 OSS SDK: 在项目中集成阿里云 OSS 的 Java SDK, 可以通过 Maven 等方式导入 SDK 依赖。

3.初始化 OSS 客户端: 使用 AccessKey ID 和 AccessKey Secret 初始化 OSS 客户端, 指定 OSS 服务的 Endpoint 和 Bucket 名称。

4.文件上传: 使用 OSS 客户端的 putObject 方法上传文件。可以先将本地文件转换为 InputStream, 然后调用 putObject 方法上传到指定的 Bucket 中[13]。

5.处理上传结果: 根据上传结果进行相应的处理, 例如成功则返回文件的 URL, 用于后续的人脸识别等操作。

如图 19 人脸上传



图 19 人脸上传

(5) 人脸识别：

1.基于人脸识别的考勤系统可以结合阿里云 OSS 实现文件上传,并利用 OSS 的文件实现人脸识别。

2.文件上传：学生在考勤签到时，系统将学生的人脸照片通过阿里云 OSS 上传到指定的 Bucket 中。

3.人脸识别：使用阿里云的人脸识别服务，通过调用 API 对上传的人脸照片进行识别。可以使用人脸识别服务的比对功能，将学生上传的人脸照片与系统中存储的人脸信息进行比对，判断是否匹配。

4.处理识别结果：根据人脸识别的结果进行相应的处理[14]。如果识别成功且匹配，则认为学生签到成功；如果识别失败或者不匹配，则认为签到失败，需要重新签到或者进行其他处理。

如图 20 人脸识别



图 20 人脸识别

(6) 普通签到

- 1.教师在后台 web 端发布签到任务，包括签到有效时间等信息。
- 2.微信小程序接收到教师发布的签到任务，并显示在学生端的界面上。
- 3.学生打开微信小程序，在指定的时间签到。小程序获取学生的签到信息。
- 4.小程序发送给后台
- 5.在后台系统中记录学生的签到情况。小程序向学生反馈签到结果，显示签到成功或失败的提示。

如图 21 普通签到



图 21 普通签到

(7) 签到请假:

- 1.教师在后台 web 端发布签到任务，包括签到有效时间等信息。
- 2.微信小程序接收到教师发布的签到任务，并显示在学生端的界面上。
- 3.学生打开微信小程序，在小程序上请假。小程序获取学生的请假信息。
- 4.小程序发送给后台
- 5.在后台系统中记录学生的请假情况。小程序向学生反馈请假结果，显示请假成功的提示。

如图 22 签到请假



图 22 签到请假

4.2 后台功能实现

(1) 后台登录:

管理员输入账号密码以及正确的验证码后点击登录,管理系统前端利用封装的 axios 库发起 POST 请求,将管理员账号参数发送到后端。后端接收到账号参数后,生成 Token 字符串,并将 Token 字符串进行封装后发送给前端。前端接收到 Token 字符串后,将其设置到 Cookie 中,以便后续的身份验证和权限控制。

如图 23 后台登录

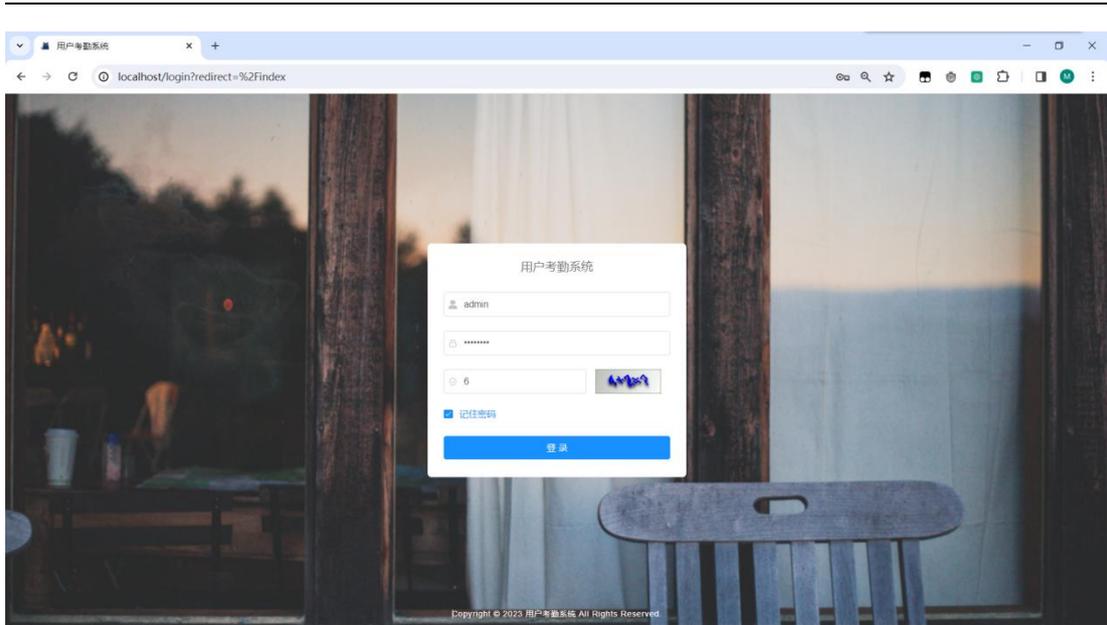


图 23 后台登录

(2) 增加班级:

前端发起增加班级请求，后端 Controller 控制层根据 url 进行接收[15]，然后 WxClazzService.java 业务层进行业务处理，然后在持久层 (WxClazzMapper.java) 进行数据库修改，返回信息给前端，提示新建班级成功。

如图 24 新建班级

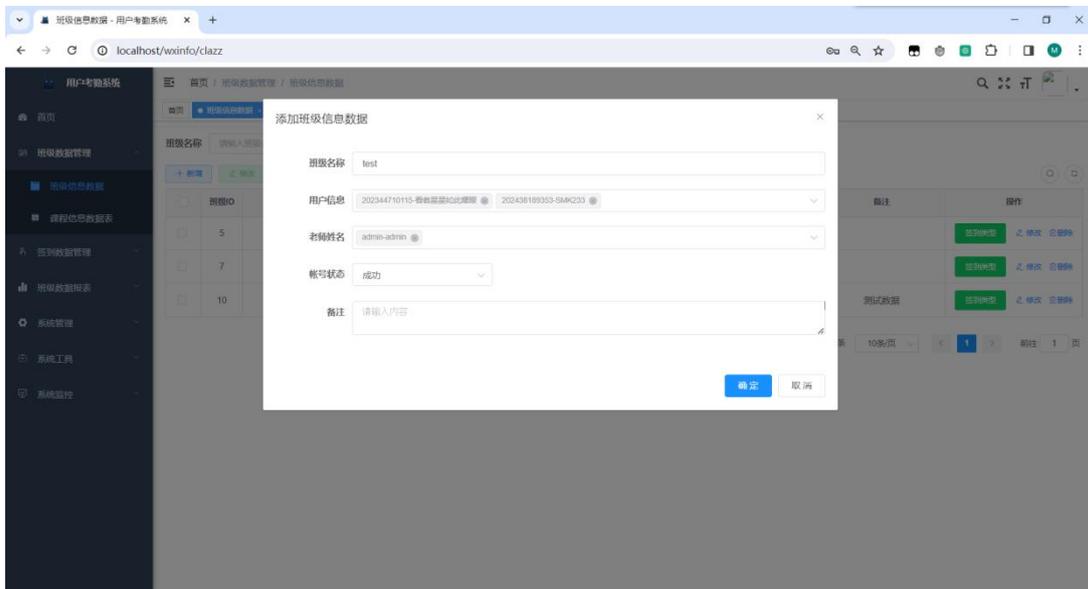


图 24 新建班级

(3) 修改班级:

前端发起修改班级请求，后端 Controller 控制层根据 url 进行接收，然后 WxClazzService.java 业务层进行业务处理，然后在持久层 (WxClazzMapper.java)

进行数据库修改，返回信息给前端，提示修改班级成功。

如图 25 修改班级

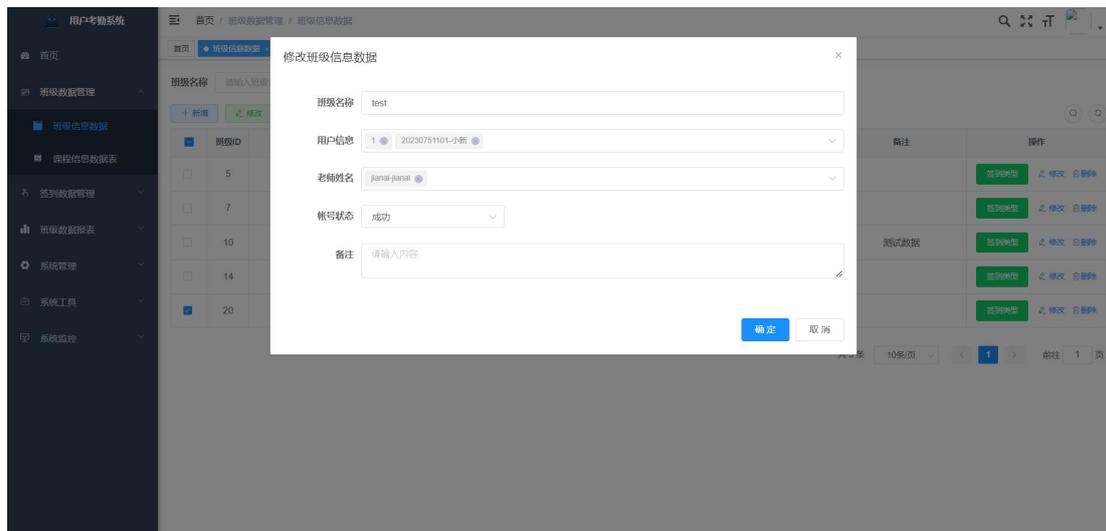


图 25 修改班级

(4) 删除班级:

前端发起删除班级请求，后端 Controller 控制层根据 url 进行接收，然后 WxClazzService.java 业务层进行业务处理，然后在持久层 (WxClazzMapper.java) 进行数据库修改，返回信息给前端提示删除班级成功。

如图 26 删除班级

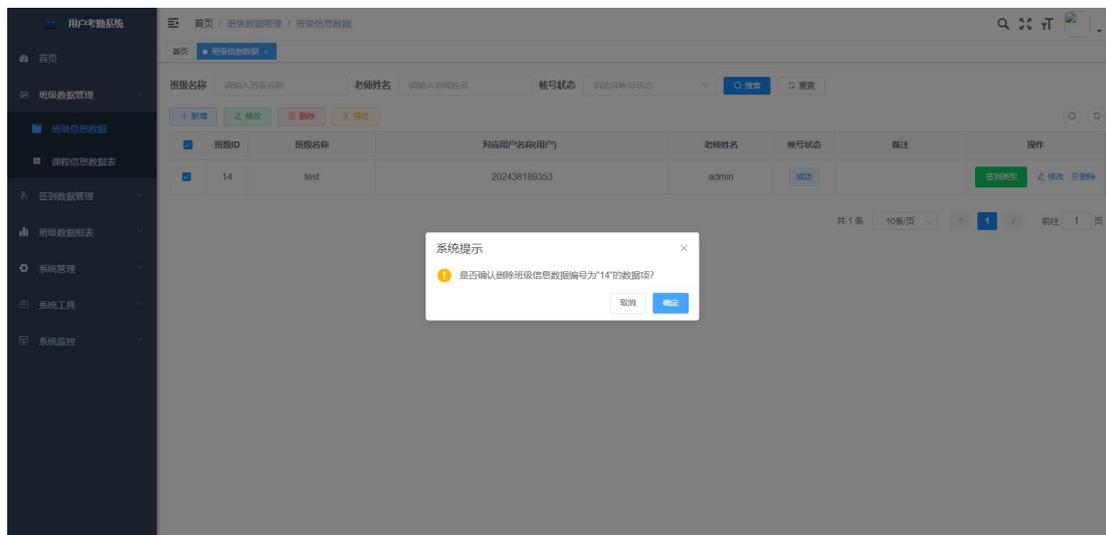


图 26 删除班级

(5) 增加课程:

前端发起增加课程请求，后端 Controller 控制层根据 url 进行接收，然后 WxCourseService.java 业务层进行业务处理，然后在持久层 (WxCourseMapper.java) 进行数据库修改，返回信息给前端，提示新建课程成

功。

如图 27 新增课程



图 27 新增课程

(6) 修改课程:

前端发起修改课程请求，后端 Controller 控制层根据 url 进行接收，然后 WxCourseService.java 业务层进行业务处理，然后在持久层 (WxCourseMapper.java) 进行数据库修改，返回信息给前端，提示修改课程成功。

如图 28 修改课程



图 28 修改课程

(7) 删除课程:

前端发起删除课程请求，后端 Controller 控制层根据 url 进行接收，然后 WxCourseService.java 业务层进行业务处理，然后在持久层

(WxCourseMapper.java) 进行数据库修改，返回信息给前端，提示删除课程成功。

如图 29 删除课程

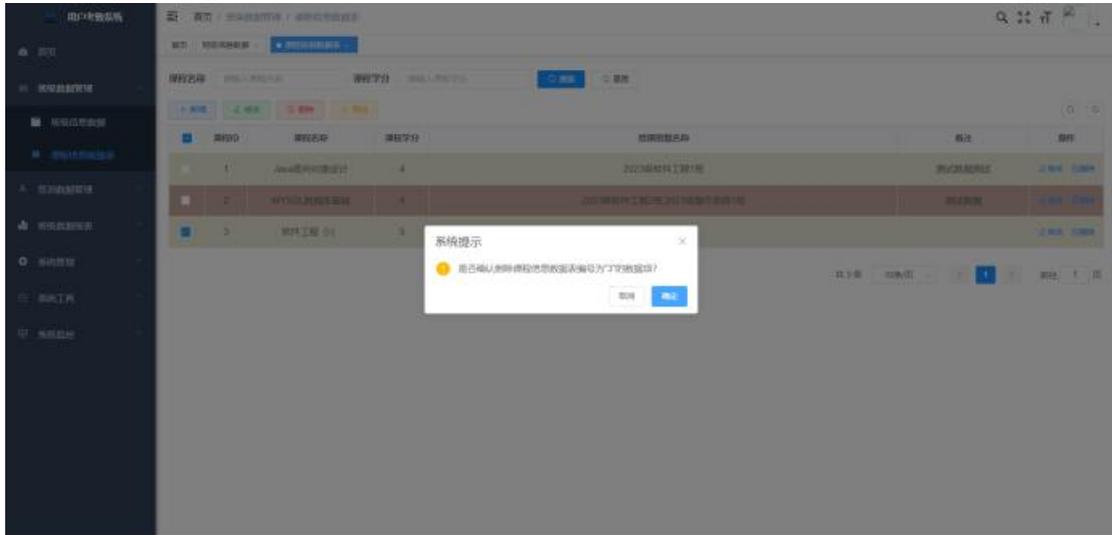


图 29 删除课程

(8) 发起签到:

前端发起新增签到请求，后端 Controller 控制层根据 url 进行接收，然后 WxSigninService.java 业务层进行业务处理，然后在持久层 (WxSigninMapper.java) 进行数据库修改，返回信息给前端，提示新建班级成功。

如图 30 新建签到

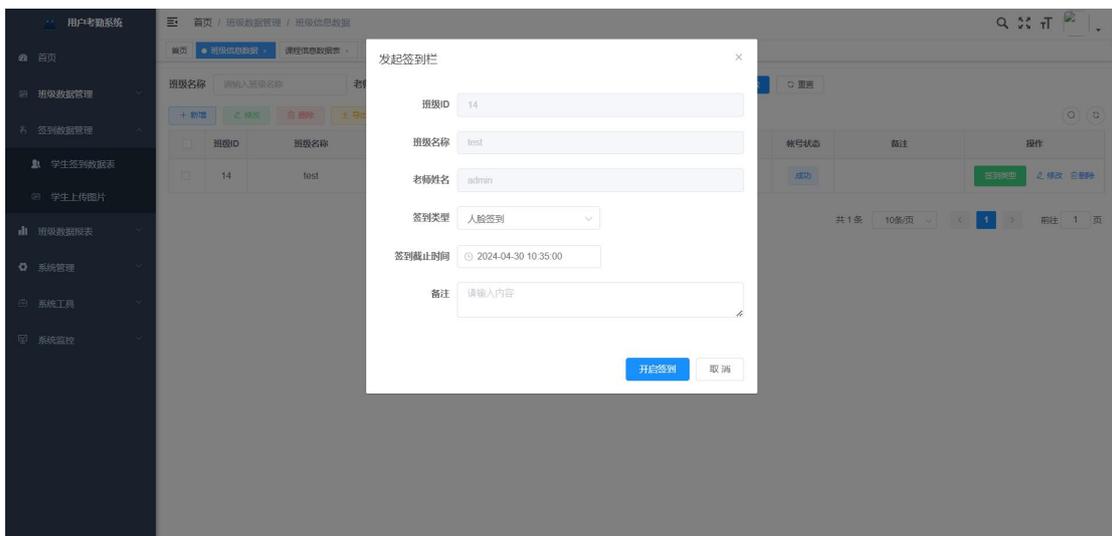


图 30 新建签到

(9) 修改签到:

前端发起修改签到请求，后端 Controller 控制层根据 url 进行接收，然后 WxSigninService.java 业务层进行业务处理，然后在持久层

(WxSigninMapper.java) 进行数据库修改, 返回信息给前端, 提示修改签到成功。

如图 31 修改签到

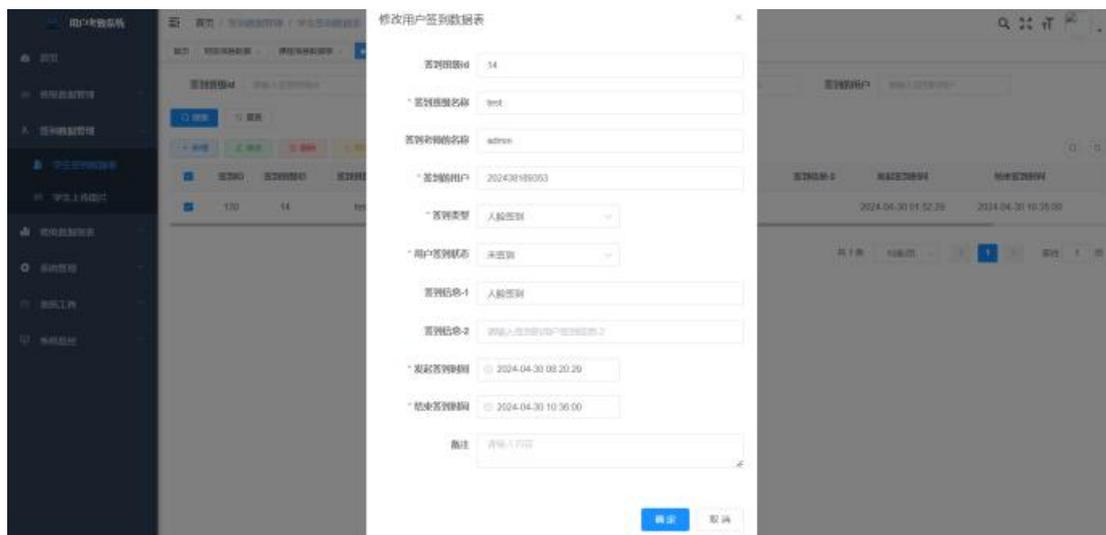


图 31 修改签到

(10) 删除签到:

前端发起新增签到请求, 后端 Controller 控制层根据 url 进行接收, 然后 WxSigninService.java 业务层进行业务处理, 然后在持久层 (WxSigninMapper.java) 进行数据库修改, 返回信息给前端, 提示删除签到成功。

如图 32 删除签到

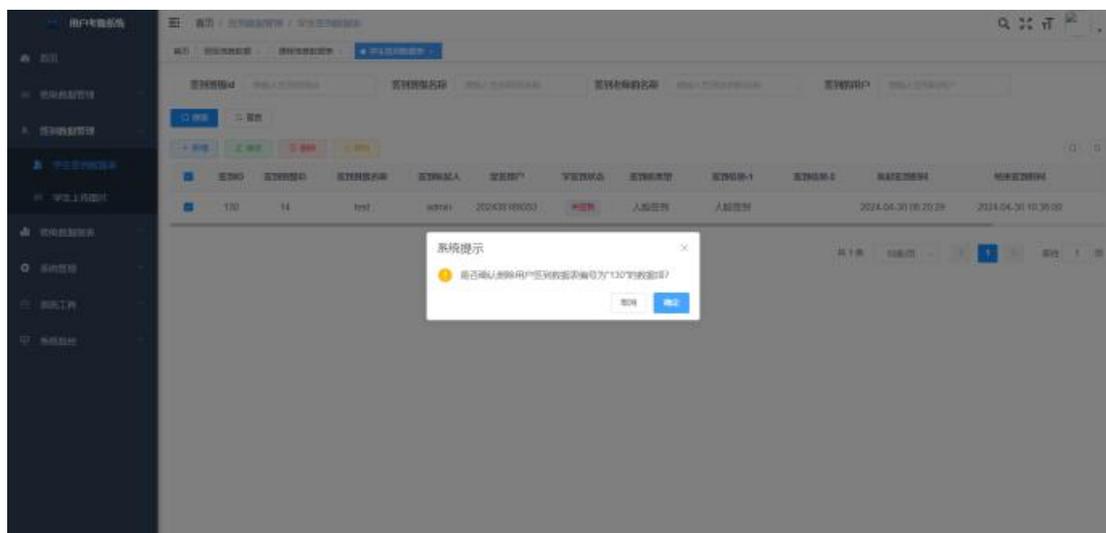


图 32 删除签到

(11) 新建用户:

前端发起新建用户请求, 后端 Controller 控制层根据 url 进行接收, 然后 ISysUserService.java 业务层进行业务处理, 然后在持久层 (ISysUserMapper.java) 进行数据库修改, 返回信息给前端, 提示新建用户成功。

如图 33 新建用户

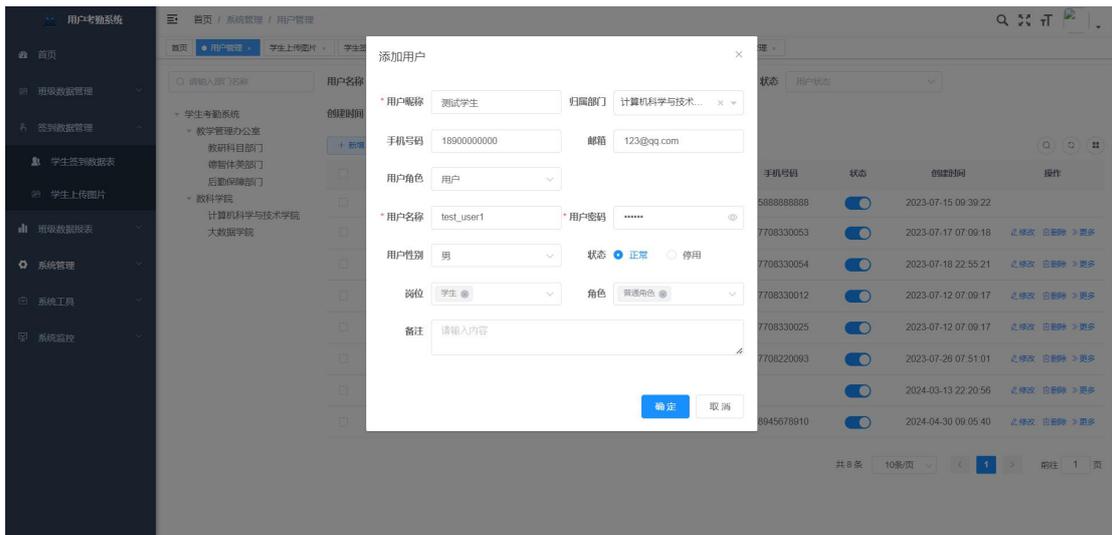


图 33 新建用户

(12) 修改用户:

前端发起修改用户请求，后端 Controller 控制层根据 url 进行接收，然后 ISysUserService.java 业务层进行业务处理，然后在持久层 (ISysUserMapper.java) 进行数据库修改，返回信息给前端，提示修改用户成功。

如图 34 修改用户



图 34 修改用户

(13) 删除用户:

前端发起删除用户请求，后端 Controller 控制层根据 url 进行接收，然后 ISysUserService.java 业务层进行业务处理，然后在持久层 (ISysUserMapper.java) 进行数据库修改，返回信息给前端，提示删除用户成功。

如图 35 删除用户

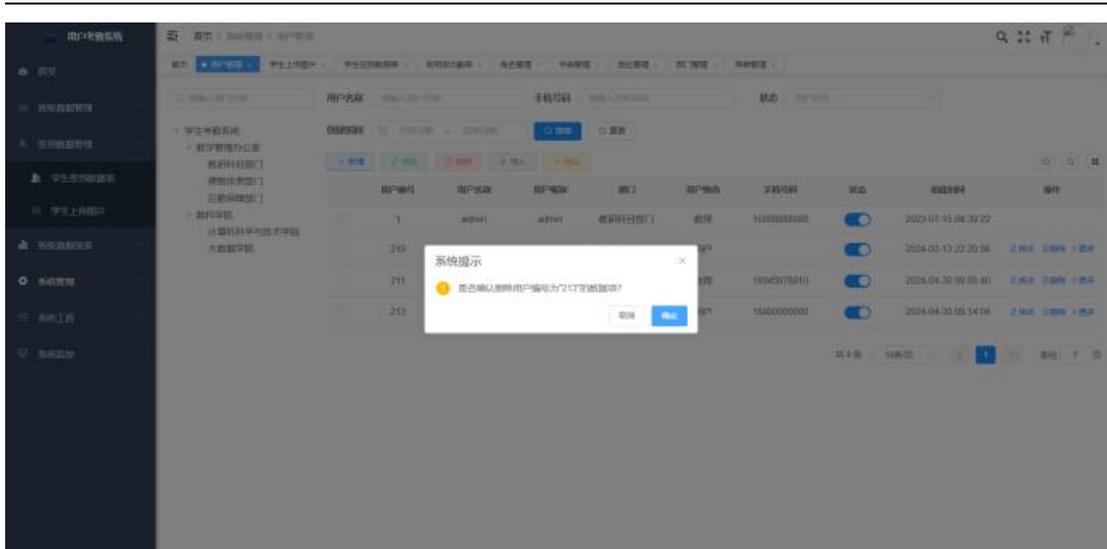


图 35 删除用户

(14) 后台新增人脸:

前端发起录入人脸图片请求，后端 Controller 控制层根据 url 进行接收，然后 WxStudentimagesService.java 业务层进行业务处理，然后在持久层 (WxStudentimagesMapper.java) 进行数据库修改，返回信息给前端，提示录入人脸成功成功。

如图 36 后台新增人脸



图 36 后台新增人脸

(15) 后台删除人脸:

前端发起删除人脸图片请求，后端 Controller 控制层根据 url 进行接收，然后 WxStudentimagesService.java 业务层进行业务处理，然后在持久层 (WxStudentimagesMapper.java) 进行数据库修改，返回信息给前端，提示删除人脸成功成功。

如图 37 后台删除人脸

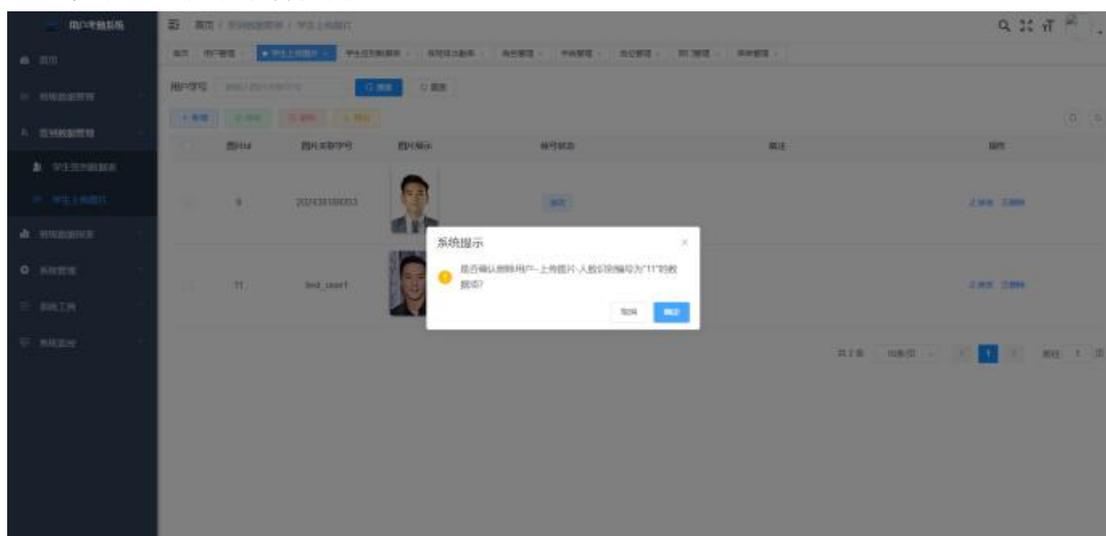


图 37 后台删除人脸

(16) 新建签到信息:

前端发起新建签到请求，后端 Controller 控制层根据 url 进行接收，然后 WxSigninService.java 业务层进行业务处理，然后在持久层 (WxSigninMapper.java) 进行数据库修改，返回信息给前端，提示新建签到成功。

如图 38 新建签到信息

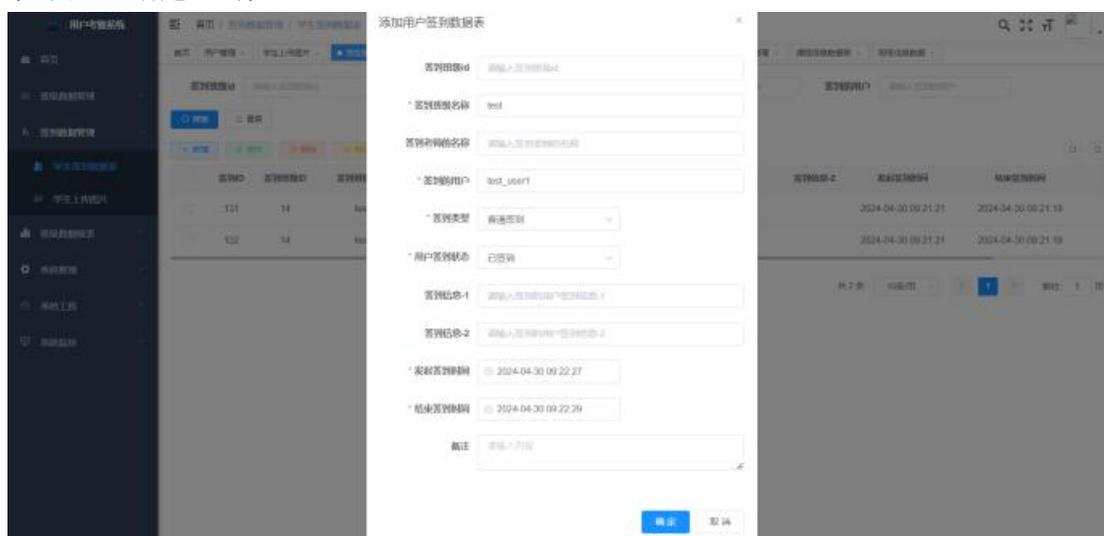


图 38 新建签到信息

(17) 修改签到信息:

前端发起修改签到请求，后端 Controller 控制层根据 url 进行接收，然后 WxSigninService.java 业务层进行业务处理，然后在持久层 (WxSigninMapper.java) 进行数据库修改，返回信息给前端，提示修改签到成功。

如图 39 修改签到信息

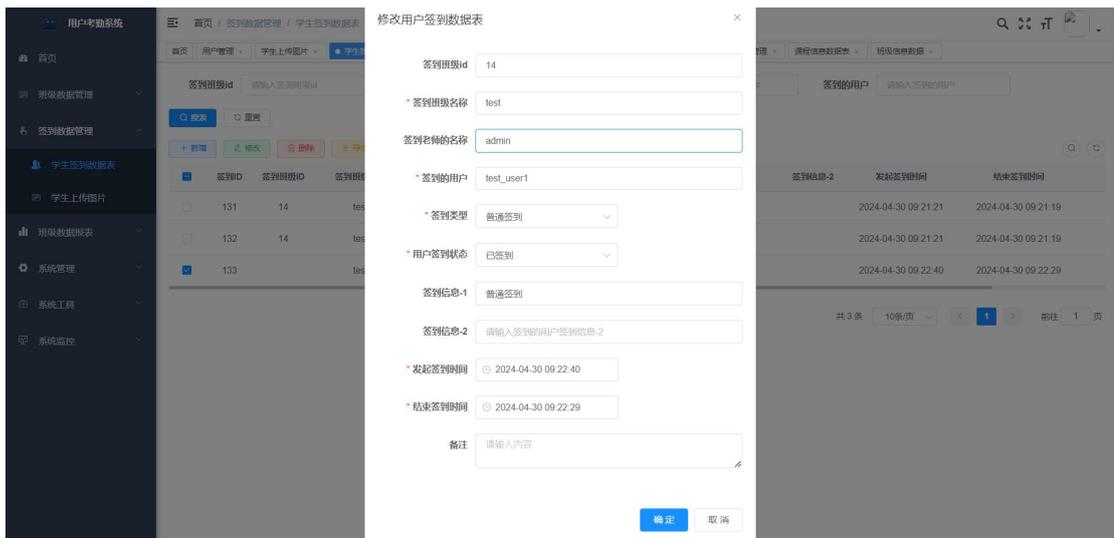


图 39 修改签到信息

(18) 删除签到信息:

前端发起删除签到请求，后端 Controller 控制层根据 url 进行接收，然后 WxSigninService.java 业务层进行业务处理，然后在持久层 (WxSigninMapper.java) 进行数据库修改，返回信息给前端，提示删除签到成功。

如图 40 删除签到信息

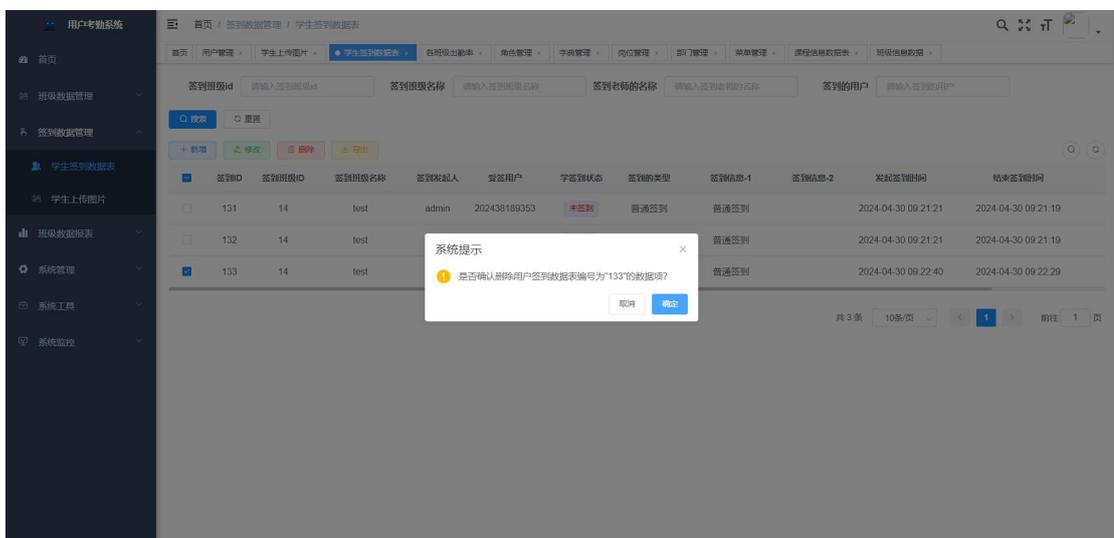


图 40 删除签到信息

(19) 查看各班出勤率:

1.ECharts.js 实现后台教师端的数据可视化功能，用于对考勤数据进行分析

和统计，
2.从后台数据库中获取考勤数据，包括学生的签到次数等统计信息，然后以 JSON 形势返回给前端。

3.后台教师端的页面中引入 ECharts.js 库，Vue 是通过 Node.js 管理的所以是

通过 NPM 下载的。

4. 根据需要创建相应的图表，如柱状图、折线图等，用于展示考勤数据的统计结果。可以使用 ECharts.js 提供的 API 进行图表的配置和数据的填充。

5. 将获取到的考勤数据填充到图表中，确保数据的准确性和完整性。

6. 将填充好数据的图表展示在后台教师端的界面上，可以根据需要进行样式和布局的调整，以便教师更直观地了解考勤情况。

如图 41 查看各班出勤率

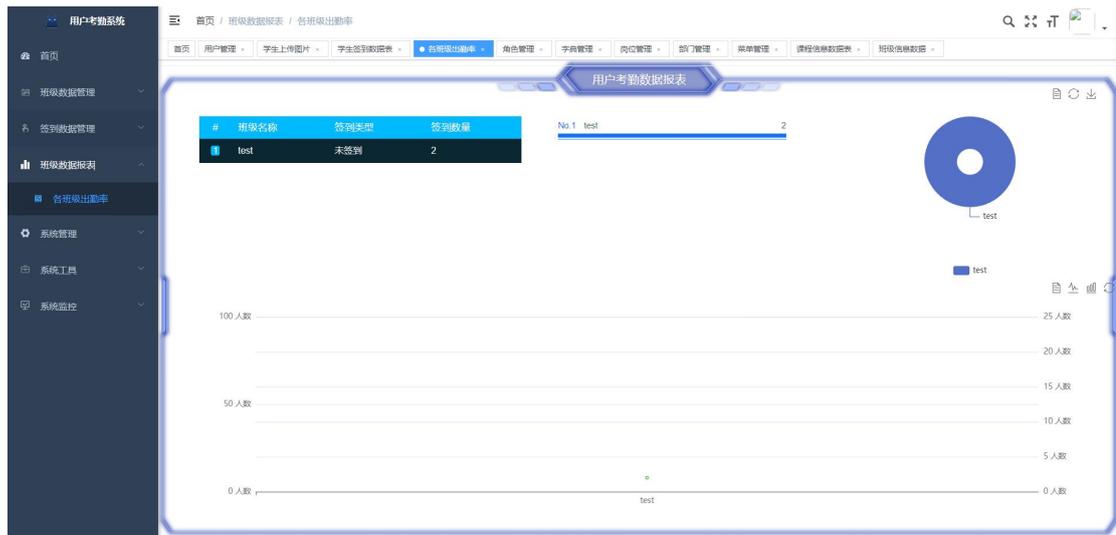


图 41 查看各班出勤率

5 系统展示

5.1 系统功能展示

(1) 小程序登录功能:

如图 42 小程序登录功能演示



图 42 小程序登录功能演示

(2) 修改个人信息:

如图 43 修改个人信息演示



图 43 修改个人信息演示

(3) 定位签到:

如图 44 定位签到演示



图 44 定位签到演示

(4) 人脸签到:

如图 45 人脸签到演示



图 45 人脸签到演示

(5) 人脸录入:

如图 46 人脸录入展示

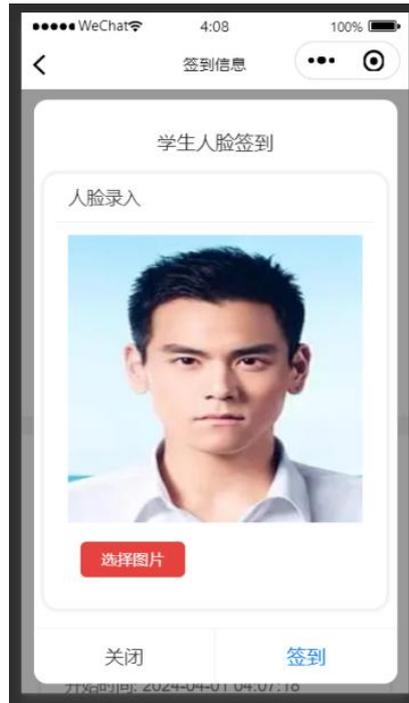


图 46 人脸录入展示

(6) 普通签到:

如图 47 普通签到演示

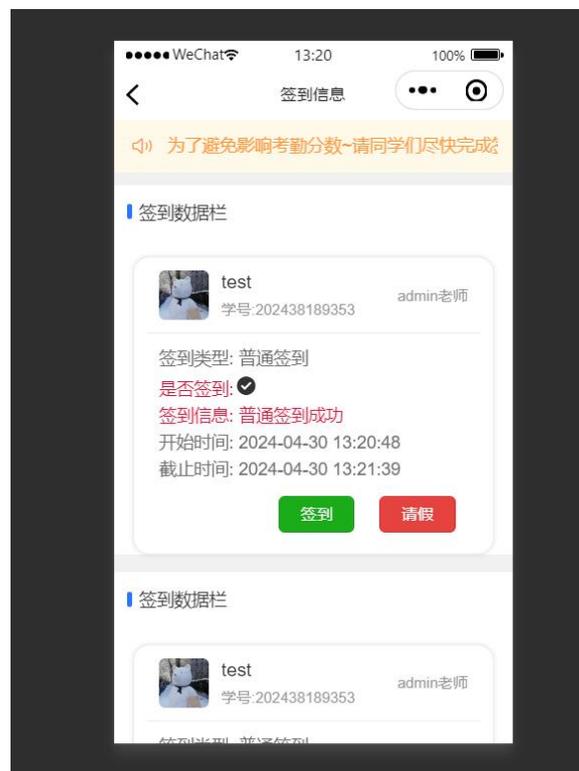


图 47 普通签到演示

(7) 签到请假:

如图 48 签到请假演示



图 48 签到请假演示

(8) 查看请假信息:

如图 49 查看请假信息

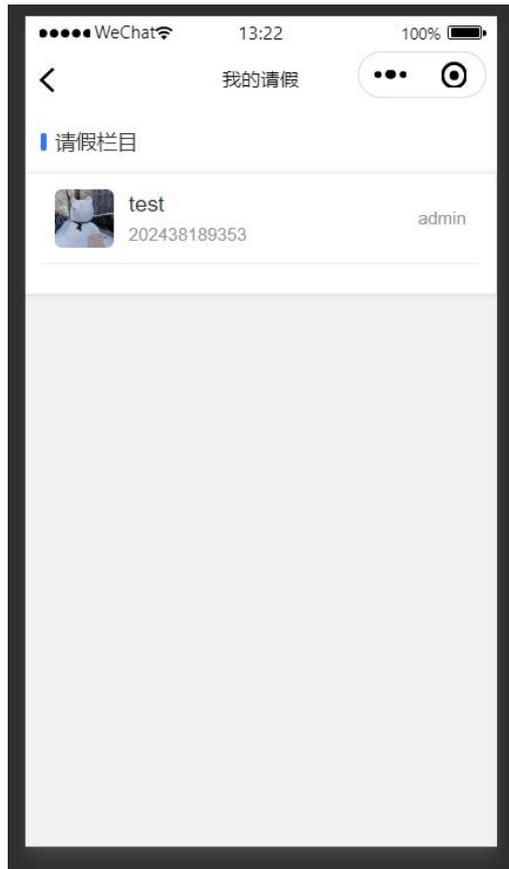


图 49 查看请假信息

(9) 后台登录:

如图 50 后台登陆演示

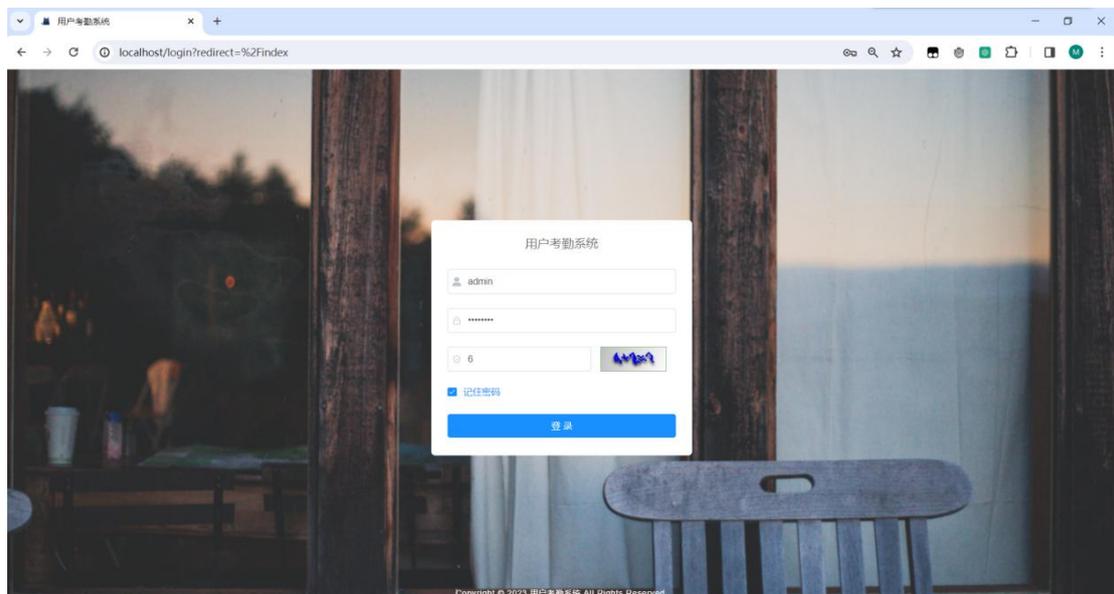


图 50 后台登陆演示

(10) 增加班级:

如图 51 新建班级信息演示

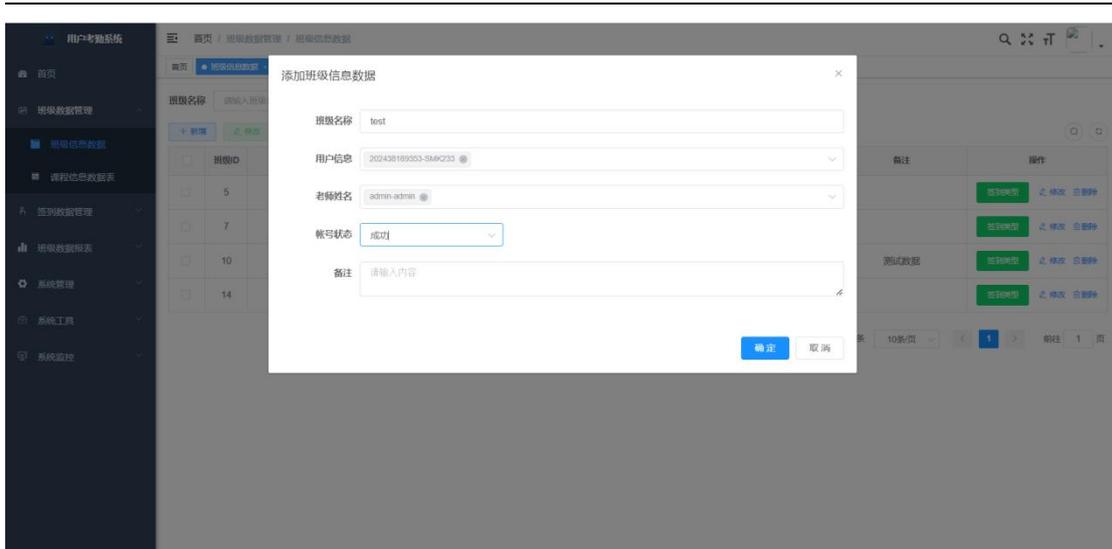


图 51 新建班级信息演示

(11) 修改班级信息:

如图 52 修改班级信息演示



图 52 修改班级信息演示

(12) 删除班级:

如图 53 删除班级信息演示

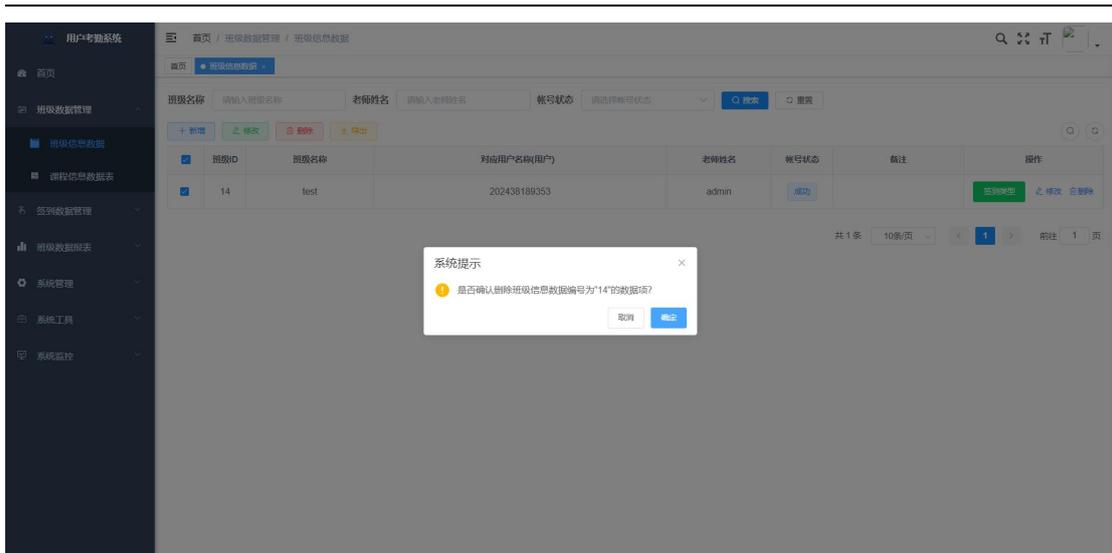


图 53 删除班级信息演示

(13) 增加课程:

如图 54 新建课程演示



图 54 新建课程演示

(14) 修改课程:

如图 55 修改课程演示

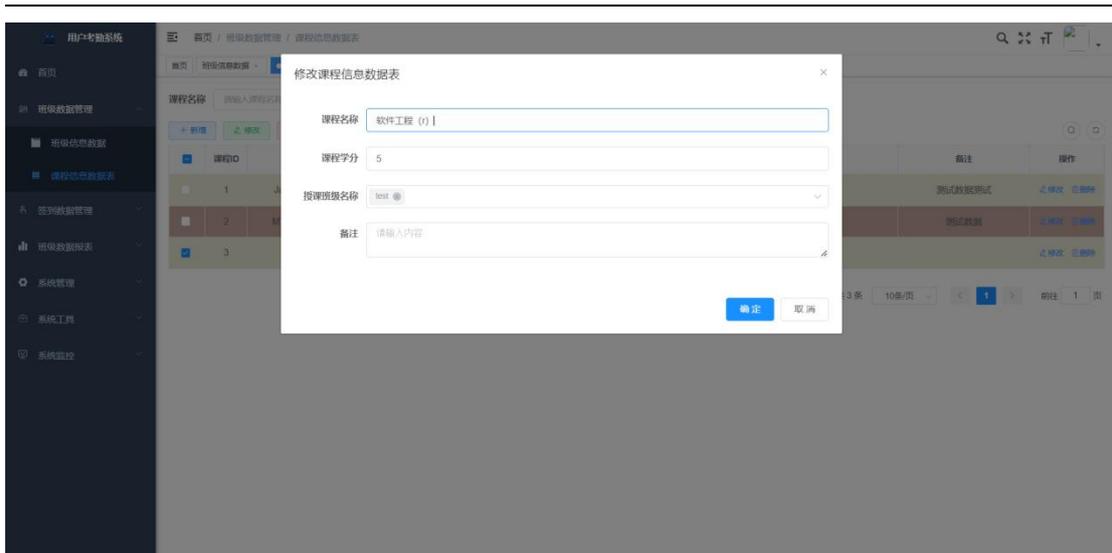


图 55 修改课程演示

(15) 删除课程:

如图 56 删除课程演示

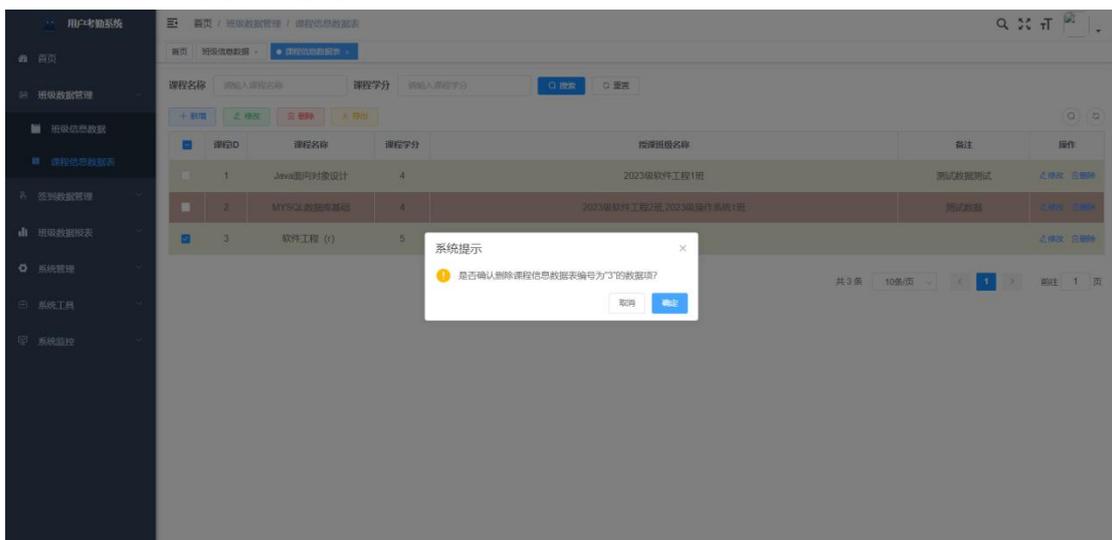


图 56 删除课程演示

(16) 发起签到

如图 57 发起签到演示

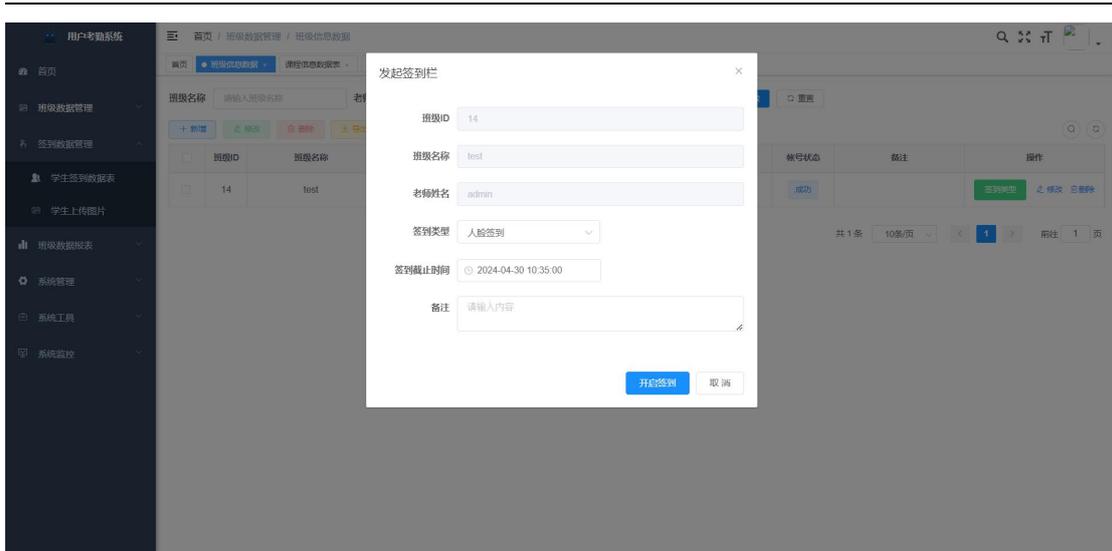


图 57 发起签到演示

(17) 修改签到:

如图 58 修改签到演示

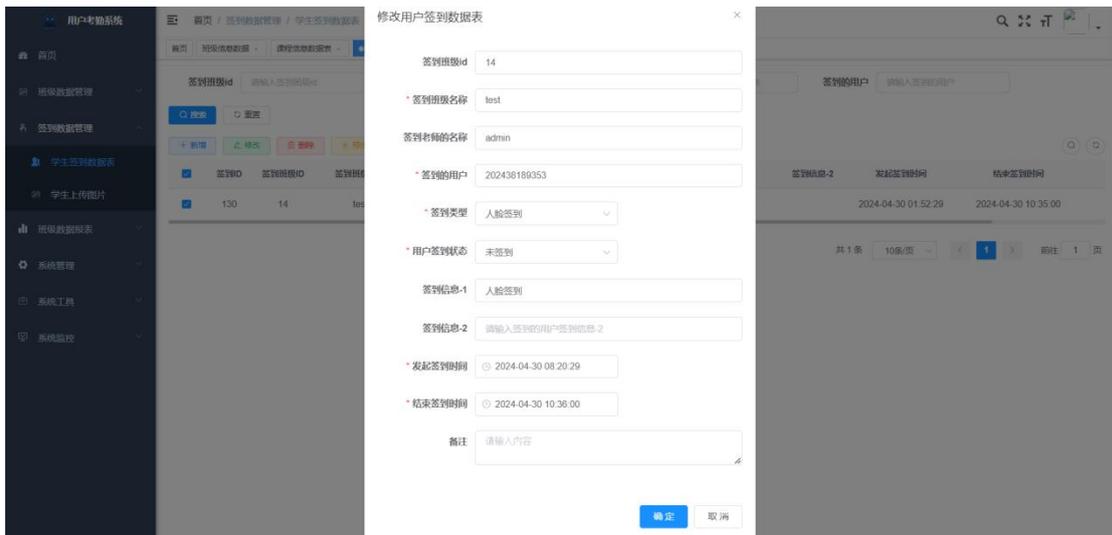


图 58 修改签到演示

(18) 删除签到:

如图 59 删除签到演示

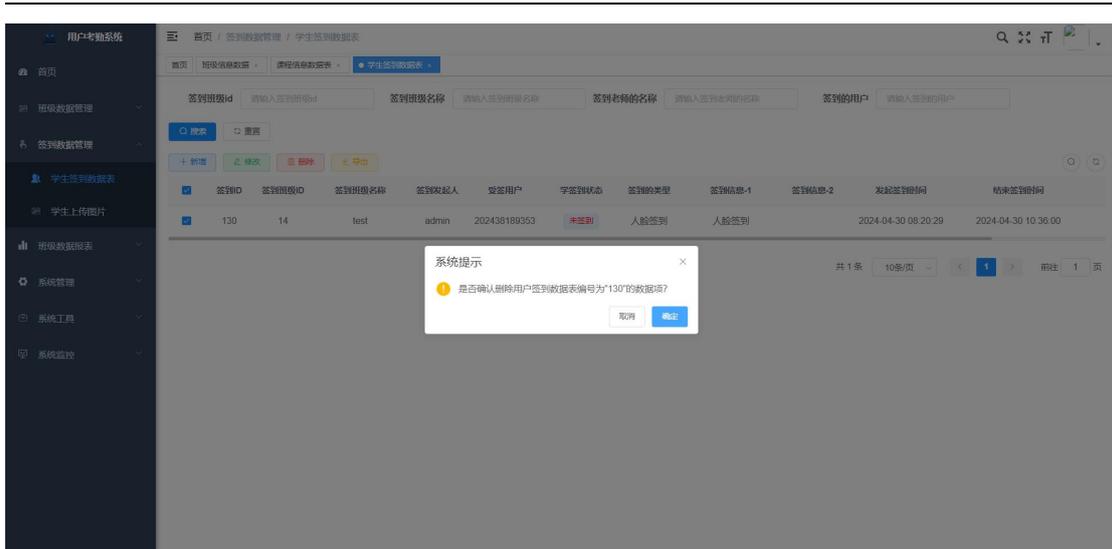


图 59 删除签到演示

(19) 新建用户:

如图 60 新建用户演示

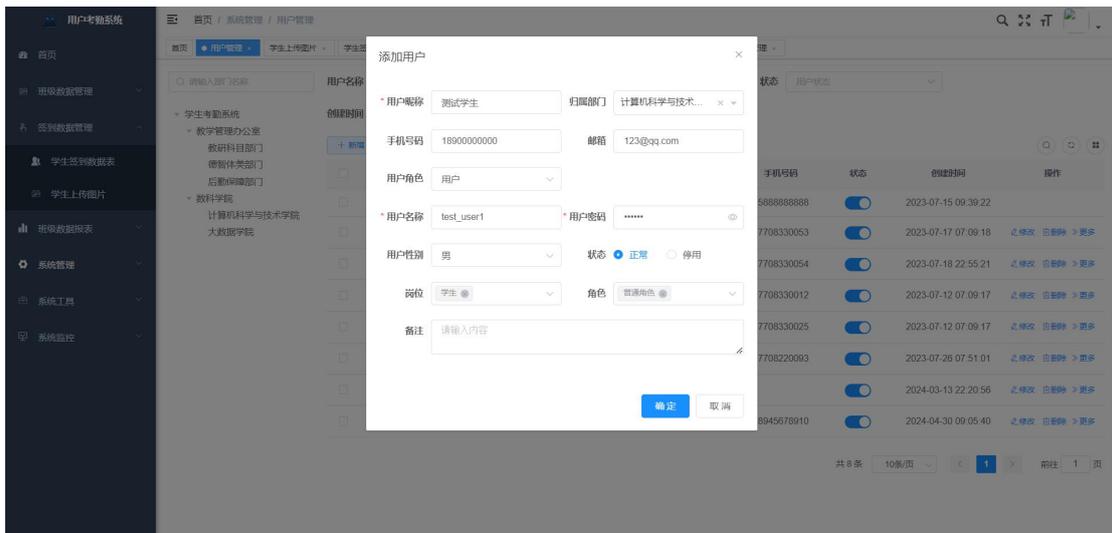


图 60 新建用户演示

(20) 修改用户:

如图 61 修改用户演示

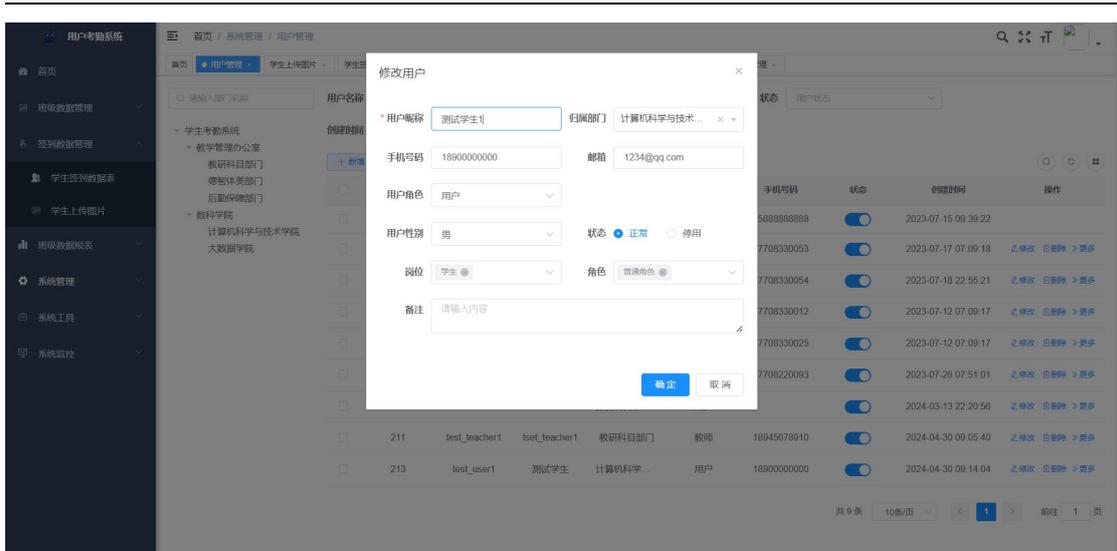


图 61 修改用户演示

(21) 删除用户:

如图 62 删除用户演示

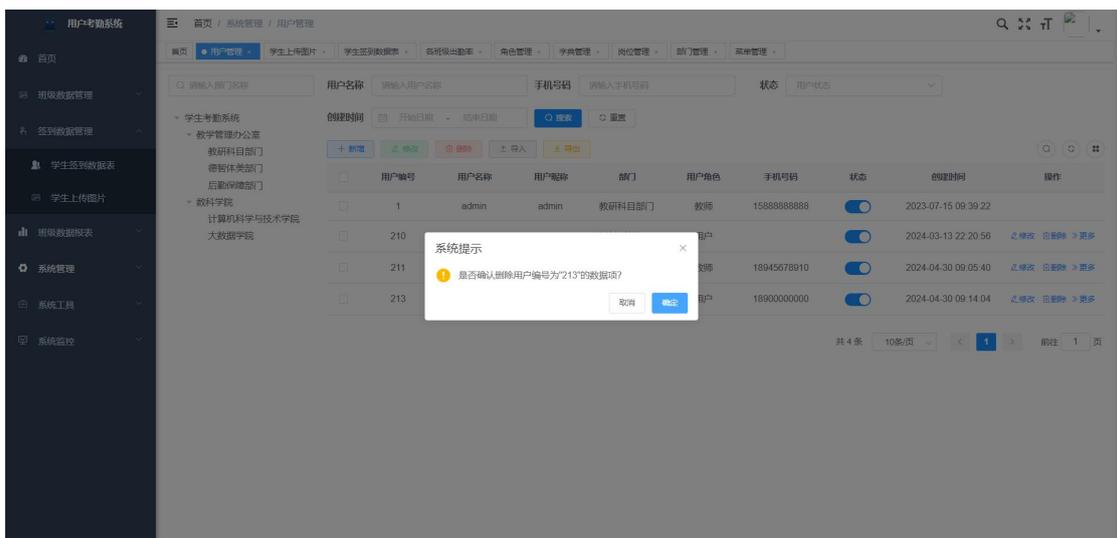


图 62 删除用户演示

(22) 后台新增人脸:

如图 63 后台新增人脸演示

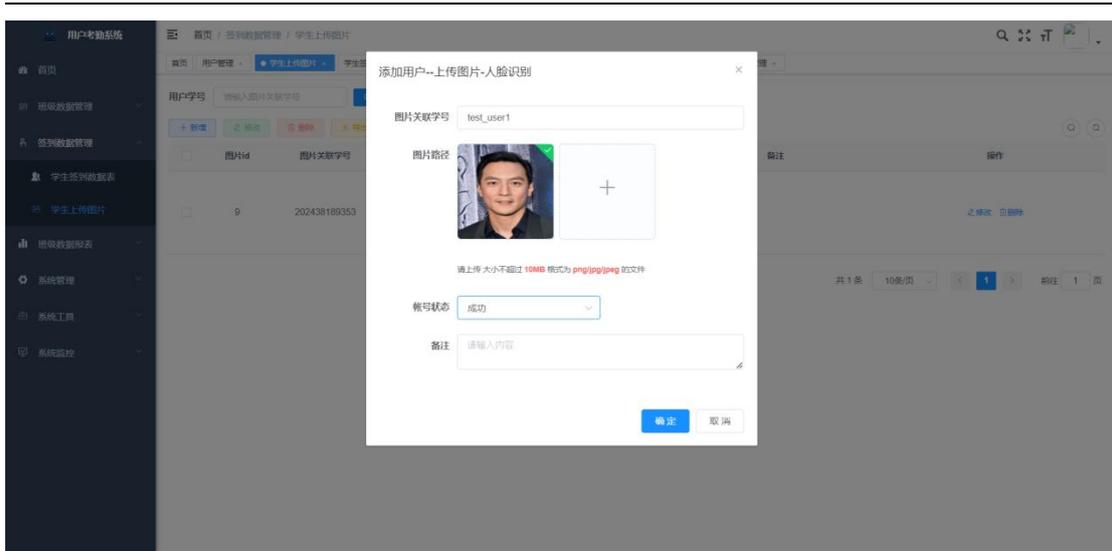


图 63 后台新增人脸演示

(23) 后台删除人脸:

如图 64 后台删除人脸演示

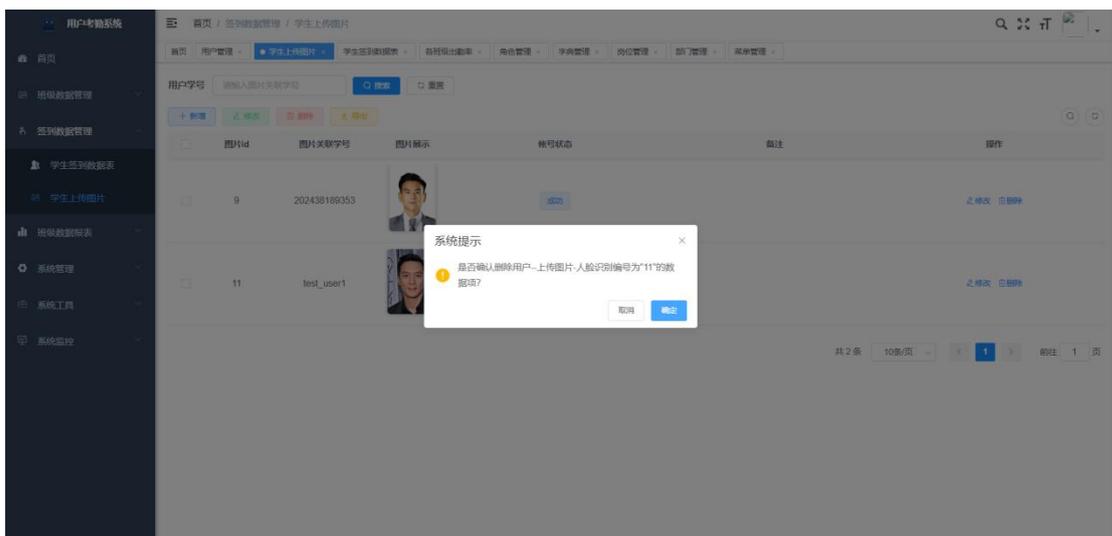


图 64 后台删除人脸演示

(24) 新建签到信息:

如图 65 新建签到信息演示

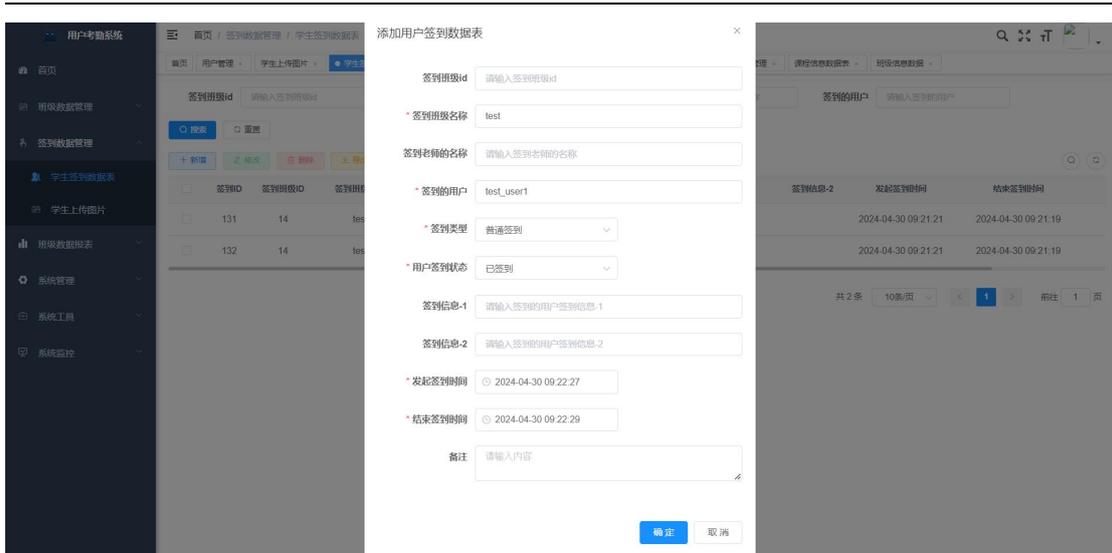


图 65 新建签到信息演示

(25) 修改签到信息:

如图 66 修改签到信息演示

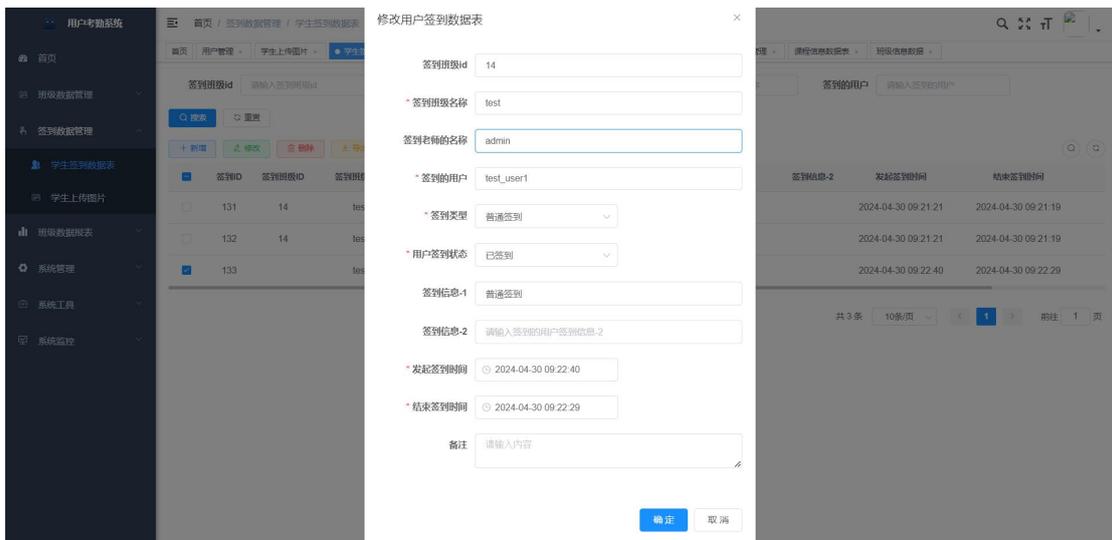


图 66 修改签到信息演示

(26) 删除签到信息:

如图 67 删除签到信息演示

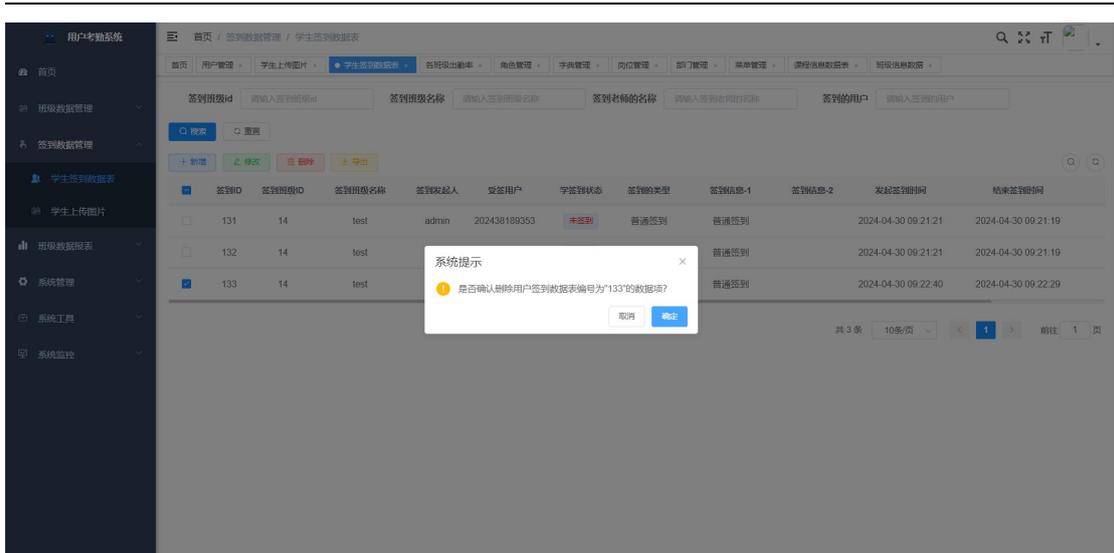


图 67 删除签到信息演示

(27) 查看各班出勤率:

如图 68 查看各班出勤率演示



图 68 查看各班出勤率演示

6 总结与展望

6.1 结论

本论文旨在设计和实现一款基于人脸识别技术的考勤管理系统，以解决学校考勤管理过程中存在的问题，提高考勤的准确性和效率。在以下几个方面都取得了一定的成果。

(1) 识别度更高：

本项目采用识别学生的面部特征进行打卡记录，由于每个人的面部特征都是独一无二的，人脸识别系统能够更准确地确认打卡人员的身份，从而大大减少了误判的可能性。达到了提高了考勤管理的准确性和可靠性的目的。

(2) 防止代打卡：

本项目采用人脸识别进行考勤打卡，只会识别已注册人员的面部特征，不是本人的面部无法通过验证进行打卡，因此可以有效地防止代打卡现象的发生，达到了保障了考勤数据的真实性和可信度的目的。

(3) 无需携带卡片：

本项目采用人脸识别方式进行考勤打卡，无需学生携带任何卡片，只需通过面部识别即可完成打卡。这大大减少了学生因忘记携带卡片或卡片丢失而无法打卡的困扰，提高了打卡的便捷性和灵活性。达到了无需携带卡片的目的。

(4) 可追溯性：

本项目可以记录每位学生的打卡记录，包括打卡时间、地点等信息。这为管理人员提供了更加详细和准确的考勤数据，便于进行后续的考勤管理和数据分析。达到了具有可追溯性的目的。

(5) 节约时间：

本项目采用人脸识别进行打卡，节约了大量的排队等待打卡的时间。由于无需接触和使用打卡机打卡，学生只需简单地通过面部识别即可完成打卡，大大提高了打卡的效率和便捷性。达到了节约时间的目的。

(6) 节约成本：

本项目无需额外的硬件设备例如打卡机和打卡卡片，也不需要额外的人力去维护打卡机等设备，因此能够节省大量的硬件成本和管理成本。达到了节约成本的目的。

综上所述，本项目设计的基于人脸识别技术的学生考勤系统成功实现了提高

识别度、防止代打卡、无需携带卡片、可追溯性、节省时间和节约成本等方面的目标，为学校考勤管理带来了改进。

参考文献

- [1] 方磊. 基于外耳结构的人耳特征提取研究[D]. 天津大学,2007
- [2] 李斯含,张柏洋,张亚楠,吕昕航,许君. 基于新技术领域的服装智慧门店应用分析与展望[J]. 毛纺科技,2021,049(7):90-94
- [3] 韩捷. 全矢谱技术及设备故障诊断应用研究[D]. 同济大学,同济大学机械工程学院,2005
- [4] 阚宝. 智能仪器设备管理系统的设计与实现[D]. 合肥工业大学,2021
- [5] 赵慧玲. 在设备管理系统中如何实现多条件复杂查询[J]. 长春工程学院学报(自然科学版),2008,009(2):65-68
- [6] 黄睿. 跨终端线下商品促销信息推荐系统的设计与实现[D]. 东南大学,2016
- [7] 鲍晓宇. 绿色建筑运营管理能耗监测和控制方法的研究与分析[J]. 科技资讯,2015,013(35):48-49
- [8] 周艳红,文冬,朱欣欣. 基于 Web 的高校教学平台设计与实现[J]. 福建电脑,2007,000(4):137139
- [9] 史程. 教务网络服务平台的设计与规划[J]. 天津职业院校联合学报,2012,014(8):83-87
- [10] 漆成. 长沙市高中新课改校级管理系统的设计与实现[D]. 山东大学,2011
- [11] 彭涛. 基于微信小程序的影视信息查看小工具[J]. 信息通信,2019,000(10):258-259
- [12] 刘会森. 面向用户数据安全的云存储系统研究与设计[D]. 西安电子科技大学,2017
- [13] 刘汉英,周剑勋. 基于 Web 的机械设备设计、制造及管理系统信息化控制软件的设计与实现[J]. 信息系统工程,2010,000(11):123-125
- [14] 叶诗韵,黄志成. 基于人脸识别的考生身份识别应用研究[J]. 软件,2018,039(12):37-39
- [15] 龚潇雨. 基于数字化校园的即时通讯平台的设计与实现[D]. 湖南大学,2017
- [16] 蔡立志,章伟,陈敏刚,王乃琪. 基于深度学习的细粒度皮肤癌图像分类研究[J]. 计算机应用与软件,2023,40(6):140-146
- [17] 王浩旻,张忠能. 集团化酒店中央预订系统的设计框架[J]. 微型电脑应用,2005,021(5):24-26
- [18] 乔心路. 多 AGV 仓储系统设计与路径规划研究[D]. 南京理工大学,2019
- [19] 叶培春. 基于 IT-Kid 的访问控制模型研究[D]. 东北大学,2006
- [20] 张恒铭. 基于 SpringBoot+MyBatis 框架的高校人事信息发布系统设计与实现[J]. 电脑知识与技术,2020,016(18):68-70
- [21] 马泽. 对象存储服务对空间非结构化 r 数据存储的探讨 r——以阿里云 OSS 为例[J]. 江西科学,2018,036(2):347-352
- [22] 王龙. 机器视觉在奶牛体况评定中的应用[D]. 东华大学,2014
- [23] 方磊. 基于外耳结构的人耳特征提取研究[D]. 天津大学,2007
- [24] A Flexible Multi-Layer Map Model Designed for Lane-Level Route Planning in Autonomous Vehicles [J]. Kun Jiang, Diange Yang, Chaoran Liu 2019
- [25] Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP]. Yu, D.,Deng, L..IEEE Signal Processing Magazine,2011
- [26] Radon and discrete cosine transforms based feature extraction and dimensionality reduction

approach for face recognition. Dattatray V. Jadhav;;Raghunath S. Holambe.Signal Processing,2008

- [27] 基于 OpenCV 的织物疵点自动检测技术研究 [J]. 时峰 ,张凤生 ,王者胜 . 青岛大学学报 (工程技术版) . 2014,第 002 期
- [28] 基于深度学习和多传感器信息融合的全场景自动泊车技术研究和应用 [J]. 卢振兴 ,曾云 . 环渤海经济瞭望 . 2019,第 007 期
- [29] 基于 OpenCV 的人脸识别课堂考勤系统[J]. 金维香;邢晨.浙江水利水电学院学报,2019(06)
- [30] 基于 OpenCV 的人脸识别智能考勤系统的实现[J]. 姜旭涛;黄承宁;陆洋.电脑知识与技术,2020(29)

致谢

大学的学习时间总是过得飞快，转眼就到了要毕业的季节。在这四年里，我收获了知识、友情和成长，也遇到了挑战、困难和磨炼。在此，我想对所有帮助过我的人表示衷心的感谢。