

基于Django的家校通的设计与实现

【原文对照报告-大学生版】

报告编号: ad5fa3eae30b2a3f

检测时间: 2024-06-17 22:12:13

检测字符数: 25828

作者姓名: 殷海

所属单位: 东华理工大学长江学院(赣东学院)

检测结论: 全文总相似比 = 复写率 + 他引率 + 自引率 + 专业术语
9.25% = **9.17%** + **0.08%** + **0.0%** + **0.0%**

其他指标: 自写率: 90.75%

高频词: 用户, 管理, 系统, 功能, 家校通

典型相似文章: 无

疑似文字图片: 1

指标说明: 复写率: 相似或疑似重复内容占全文的比重

他引率: 引用他人的部分占全文的比重

自引率: 引用自己已发表部分占全文的比重

自写率: 原创内容占全文的比重

典型相似性: 相似或疑似重复内容占全文总相似比超过30% 专业术语: 公式定理、法律条文、行业用语等占全文的比重

相似片段: 总相似片段 50 期刊: 5 博硕: 16 综合: 0
外文: 0 自建库: 8 互联网: 21

检测范围: 中文科技期刊论文全文数据库 中文主要报纸全文数据库 中国专利特色数据库
博士/硕士学位论文全文数据库 中国主要会议论文特色数据库 港澳台文献资源
外文特色文献数据全库 维普优先出版论文全文数据库 互联网数据资源/互联网文档资源
高校自建资源库 图书资源 古籍文献资源
个人自建资源库 年鉴资源 IPUB原创作品

时间范围: 1989-01-01至2024-06-17

原文对照

颜色标注说明:

- 自写片段
- 复写片段（相似或疑似重复）
- 引用片段（引用）
- 专业术语（公式定理、法律条文、行业用语等）



東華理工大學長江學院
EAST CHINA INSTITUTE OF TECHNOLOGY, YANGTZE RIVER COLLEGE

本 科 生 毕 业 论 文

论文题目	基于Django的家校通的设计与实现
姓 名	殷海
学 号	20203031427
班 级	2030314
年 级	2020
专 业	计算机科学与技术
学 院	信息工程
指导教师	余恬
完成时间	2024年5月27日

作 者 声 明

本人以信誉郑重声明：所呈交的学位毕业论文，是本人在指导教师指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学或其他教育机构的学位或证书而使用过的材料。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业论文引起的法律结果完全由本人承担。

本毕业论文成果归东华理工大学所有。

特此声明。

毕业论文作者（签字）：

签字日期： 年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘要

本论文设计与实现了一个基于Django的家校通系统，目的是加强中小学家庭和学校之间的信息共享和学习管理。家校通后端采用了Django框架，成功地构建了一个功能丰富并且易于使用的系统；家校通系统前端使用Bootstrap框架，使界面设计更美观、响应更快、用户体验更好；采用数据库Sqlite使得家校通系统更轻量级、易于部署和跨平台；系统还采用了良好的软件工程实践，如模块化设计、代码重用，确保系统的可靠性和可维护性。家校通用户分为两种角色，家庭用户和班主任用户。通过需求分析，系统为他们提供了各种协作和管理的功能，其中家庭用户功能包括注册和登录、签到、查看通知、事项讨论、查看考核成绩，班主任用户在家庭用户功能基础上还可以发布通知、查看考勤统计记录、发布和评定考核记录、管理班级和成员。系统各项功能能够正确实现，系统操作简便，可以提高班级和家庭的协作效率和学习管理水平。未来可以通过用户反馈和数据分析来改进系统性能和用户体验。

关键词： 家校通；Django框架；Bootstrap；Sqlite

ABSTRACT

This thesis designs and implements a home-school communication system based on Django, which aims to enhance information sharing and learning management between primary and secondary school families and schools. The Django framework is used in the back-end of the home-school communication system, which successfully builds a feature-rich and easy-to-use system; the Bootstrap framework is used in the front-end, which makes the interface design of the home-school communication system more beautiful, responsive, and better user experience; the database is Sqlite, which makes the home-school communication system more lightweight, easy to deploy, and cross-platform; and the system adopts good software engineering practices such as modular design and code reuse to ensure the reliability and maintenance of the system. The system also adopts good software engineering practices, such as modular design and code reuse, to ensure the reliability and maintainability of the system. Users of Home-school Connect are categorised into two roles, family users and classroom teacher users. Through the requirement analysis, the system provides them with a variety of collaboration and management functions, in which family users can register and log in, sign in, view notices, discuss matters, and view assessment results, while classroom teacher users can release notices, view attendance statistics records, release and evaluate assessment records, and manage classes and members on the basis of the functions of family users. All functions of the system can be realised correctly, and the easy operation of the system can improve the efficiency of class and family collaboration and learning management. In the future, the system performance and user experience can be improved through user feedback and data analysis.

Key words: Home-School Communication System; Django framework; Bootstrap; Sqlite

目录

摘要	I
ABSTRACT	II
1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 研究内容	2
.....	

2 系统相关技术	3
2.1 前端开发技术	3
2.1.1 Bootstrap框架	3
2.1.2 ECHARTS	3
2.2 后端开发技术	3
2.2.1 MVT模式	3
2.2.2 Django框架	4
2.2.3 Sqlite	4
2.2.4 ORM	4
2.3 开发环境及开发工具	4
2.3.1 Pycharm	4
2.3.2 Navicat	5
2.3.3 MacOS 13.6	5
3 系统分析	6
3.1 可行性分析	6
3.1.1 技术可行性	6
3.1.2 经济可行性	6
3.1.3 社会可行性	6
3.2 需求分析	6
3.2.1 功能需求	6
3.2.2 性能需求	7
3.2.3 可扩展性需求	7
4 系统设计	9
4.1 系统模块设计	9
4.1.1 家庭用户模块	9
4.1.2 班主任模块	9
4.1.3 系统功能模块	9
4.1.4 核心功能的流程设计	10
4.2 数据库设计	12
.....	

4.2.1 数据库设计原则	12
4.2.2 数据库设计	12
4.2.3 数据库表设计	13
5 系统实现	16
5.1 模块开发	16
5.2 用户注册	16
5.3 用户登录	17
5.4 签到	18
5.5 考勤统计	20
5.6 通知栏	21
5.7 通知栏管理	22
5.8 考核管理	23
5.9 考核评定记录	25
5.10 事项	26
5.11 班级管理	28
5.12 成员管理	29
5.13 Django超级管理员后台	32
结 论	35
致 谢	36
参考文献	37

1 绪论

1.1 研究背景及意义

在当前的教育背景下，传统的家庭与学校之间的沟通模式面临着信息传递不及时、学生反馈不畅等问题，迫切需要一种能够实现实时参与、多方互动和自由沟通的沟通工具。家校通的产生提供了一个符合班级和家庭需求的学习工具，提高了学生的学习参与度和有效性。家校通的出现源于教学模式、技术进步和家庭需求的变化，为教育提供了更灵活、更个性化的学习模式，促进了教育的创新和进步。通过家庭和学校的融合可以实现中小学教育的发展和改善。

家校通利用互联网技术提供的数字化学习与合作平台，促进班主任、学生和家长之间的在线交流。其功能包括通知发布、成绩分析、讨论互动等，拓展交流学习方式和创新教学模式。家长、学生和班主任之间通过家校通的沟通更加顺畅，共同关注学生的成长。利用家校通改善教育领域的家校沟通现状，满足不断变化的教育需求。家校通是一个数字化的学习和协作平台，提供了便捷的在线交流功能，增强了教学的协作性和信息的透明度。推广家校通有助于推动新型教学模式的发展，鼓励家校之间的协作和探索性学习，提高沟通效率和学习效果。家校通还能够推动教育改革，提升教学质量、资源利用率，促进教育优化发展。作为一种创新的教学管理和沟通工具，家校通有助于教育领域的进步和改革，构建开放、透明、共享的教育环境，推动教育模式的创新和进步。

1.2 国内外研究现状

Swantje D等人在2019年《Antecedents and Outcomes of Parental Homework Involvement: How Do Family-School Partnerships Affect Parental Homework Involvement and Student Outcomes》中认为父母的参与可以提升孩子在学校的成绩和幸福感,研究表明,强有力的家校合作关系有助于提高孩子作业参与的质量,从而促进学生的成就和福祉^[1]。

邱睿在2019年《基于微信小程序的家校通系统开发与实现》写到早期的家校通信平台是由三大通信运营商为代表开发的,其主要功能是与用户的电话号码绑定,仅限于允许运营商使用该电话号码,家长接收和回复短信,实现家庭和学校之间的初步沟通^[2]。他认为这个平台在一定程度上促进了家庭和学校之间的交流。汪时冲和张鸽在2018年《大学生手机日常使用情况调查与分析——以某市属高校为例》谈了阿里集团在2014年与首都教改委共同推出了“阿里师生”项目,并在北京市的小学一年级进行试点,该项目具备通知、通讯录、班级群等功能^[3]。2016年1月,其被北京市教改委更名为“师生家校”,在市内各小学推行。随后,一所来自武汉市洪山区的小学率先试用了家校交流APP系统,并在全区推广使用。

Django在2005年发布至今共迭代了21个版本,目前技术已经非常成熟并且在Web开发中的领先地位。Xiya Y等人在2023年《Design and Deployment of Django-based Housing Information Management System Django》中认为Django是Python比较流行的web框架,可以简化开发过程,使开发人员能够快速构建的web应用程序^[4]。Bootstrap开发于2011,从Bootstrap 1版本更新迭代到Bootstrap 5成为前端最受欢迎的框架同时保持了它在前端开发中的领先地位。赵丙秀在2023年《Bootstrap 5的定制化研究》认为它是简单高效而备受喜爱前端框架,使开发人员的前端页面更具吸引力更易于使用^[5]。

1.3 研究内容

本文主要研究如何开发家校通并实现其基本功能。在研究相关文献后,本文选择基于Python开发语言的Django框架,以便实现跨平台的使用。为了确保数据安全,在家校通的开发中,将使用SHA1加密算法对所有用户密码进行加密和存储。在数据库的选择上采用了Sqlite,因为它具有良好的兼容性,可以在多个平台上使用,从而确保系统的稳定性。在后端开发方面将充分利用Django框架的优势,包括其丰富的内置功能以提高开发效率,例如ORM、表单处理和身份验证系统。在前端开发方面将利用Bootstrap的内置样式,因为它提供了丰富的UI组件和样式库,从而大大减少工作量并加快开发进度。家校通将实现的功能包括用户注册和登录、签到、考勤统计、通知管理、考核管理、事项、班级管理以及成员管理等。

2 系统相关技术

2.1 前端开发技术

2.1.1 Bootstrap框架

Bootstrap是一个基于HTML、CSS和JavaScript的简单高效而备受喜爱前端框架。提供许多样式和组件,但有时需要自定义。Bootstrap 5引入了CSS和Sass变量使定制变得更容易,可以通过预编译版或源码版进行定制^[5]。Bootstrap具有布局响应快、样式丰富、内置JavaScript插件、兼容性强、文档资源丰富等优点。通过Bootstrap,开发人员可以快速构建各种项目规模的网站和web应用程序,使开发人员的前端页面更具吸引力更易于使用。

2.1.2 ECHARTS

ECharts是由百度开发的开源的可视化图表库,兼容很多浏览器能够在电脑和移动设备上流畅运行。它提供直观、可交互的数据可视化图表,用户只需提供基本数据就能通过适当的动画展示数据的变化^[6]。Echarts广泛应用于数据分析、商业报告、可视化大屏和地理信息系统等领域。使用Echarts让考勤统计分析页面把数据直观地显示给班主任用户。

2.2 后端开发技术

2.2.1 MVT模式

目前基于Python的Web开发框架大多数都使用MVT设计模式。Django作为全栈框架采用MTV模式,把动态逻辑和静态页面分离展示。它还使用ORM技术将数据库表抽象成Python的类。使数据库表的操作转换为Python类的操作,避免了编写复杂的SQL语句。如图2-1。Django框架的核心是MVT模式,业务逻辑、数据处理和页面呈现通过模型、视图和模板三个组件实现了的分离。这使代码模块化、维护更简便。同时Django提供了丰富的功能和工具,简化了开发并且提高了效率^[7]。



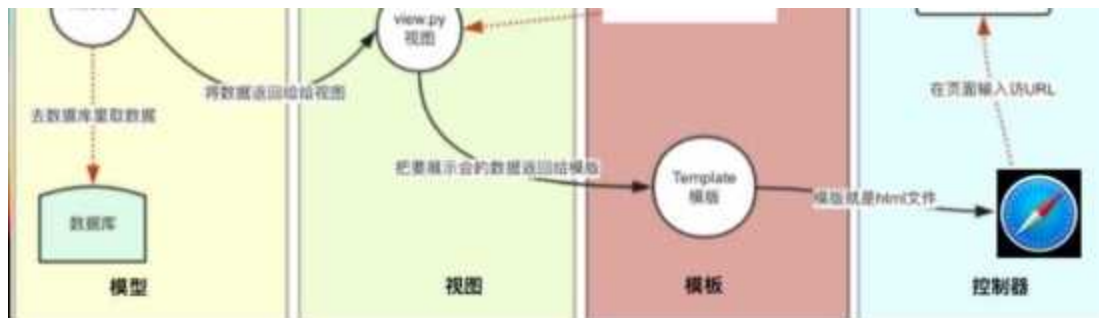


图2-1 MVT模式图表

2.2.2 Django框架

Django是Python比较流行的web框架，因为它可以简化开发过程。它核心概念是可插拔应用程序使项目结构清楚，让项目更加容易管理。Django的模块化开发让代码更加可读和可维护。举个例子，电子商务网站将产品、用户、订单等分离为独立的应用扩大了灵活性。Django能加快开发因为它有广泛的特性和工具。数据库交互操作可以通过ORM简化，数据验证和处理可以用表单系统简化。Django还有管理员界面和内置的用户身份验证系统使开发人员能够快速构建的web应用程序^[8]。

2.2.3 Sqlite

Sqlite是D. Richard Shipp在2000年发明的强大的嵌入式关系数据库管理系统。SQL引擎不需要配置，支持系统出故障后恢复数据。每个文件可保存一个完整数据库，不同计算机间可以共享^[9]。

2.2.4 ORM

ORM是一种非常好的技术将Python类与数据库进行连接，有简单数据库API可供用户使用。使用ORM框架开发人员无需编写SQL语句操纵数据库。原理是将数据库表映射成类，行映射成实例，字段映射成属性。在Django中ORM转化类、实例和属性操作为SQL。这设计使得数据库的操作更便捷，提升代码可读性和可维护性^[10]。

2.3开发环境及开发工具

2.3.1 Pycharm

JetBrains公司为软件创建了智能化、提高生产效率的开发工具，开发人员简化其复杂的需求并自动工作。JetBrains 广泛的开发人员生产力工具包括最流行Python 开发工具：PyCharm^[11]。

2.3.2 Navicat

从资料了解到Navicat是一个数据库软件产品，这个软件提供了一个集中的界面可以连接许多数据库并进行管理操作。可以方便查看家校通数据读写是否成功，还可以使用Navicat方便地进行批量操作，如导入导出数据、创建查询工具、修改表字段、进行数据同步和备份等^[12]。

2.3.3 MacOS 13.6

MacOS是Apple公司开发的基于Unix的界面友好的操作系统，支持大部分Linux代码，具有稳定性、直观用户界面和丰富开发工具，为开发者提供良好的编程环境。系统自带Python为开发者提供了便利，使得项目开发更轻松稳定，确保项目顺利进行和成功交付。

3 系统分析

2.4 可行性分析

3.1.1 技术可行性

Bootstrap有响应迅速、好看的界面设计工具，而Django是一个强大的Python web框架，Sqlite是一个轻量级的文件数据库。目前这些技术拥有广泛的社区支持和文档资源，可以帮助快速构建高效和可维护的web程序，并已有一些成功案例。总体而言，使用Bootstrap、Django和Sqlite可以有良好的体验、加速开发过程和可靠的数据存储，能够开发本文所需的家校通平台。因此，本研究在技术上是可行的。

3.1.2 经济可行性

引入家校通系统可以大大减少学校和家庭的通讯成本。传统的纸质通知需要耗费大量的人力、纸张、油墨和邮寄等费用。家校通采用的是开源的技术，降低了开发成本和后期方便定制功能。家校通系统所有数据以文本的形式存储，学校不必购买昂贵的高性能服务器、带宽以及大存储空间就能正常运行家校通的服务。因此家校通在经济上是可行的。

3.1.3 社会可行性

家校通的签到和签退功能提高了管理效率和学生安全，通知功能方便了学校和家长之间的沟通，讨论功能促进了沟通和解决问题，分数发布和历史查看功能有助于了解学习进度，分数登记功能支持个性化学习。因此，家校通系统的功能设计能够满足教学管理和家校沟通的需要，能够提高教育质量和学生学习质量，具有社会可行性。

2.5 需求分析

2.5.1 功能需求

通过线下走访中小学，本研究把家校通的用户角色分为两类：班主任用户和家庭用户。家庭用户以学生为基础，家长和学生可以共享同一个帐户；班主任用户拥有所有功能权限，而家庭用户只需要基本功能。

(1) 家庭用户需求

家庭用户需要在家校通上注册和登录，一旦注册成功，用户将成为家庭用户角色。家庭用户可以签到签退，例如学生离开家时签到，学生回家时签退。根据相应时间，班主任能了解学生的上学时间，确认学生安全到家。

在通知栏中家庭用户可以查看通知，在事件模块中进行请假、讨论、提问等等。家庭用户还可以查看所有考核记录，比如考试成绩以及所有历史**考试记录**，**考试成绩根据分数分为优秀、良好、及格和不及格，不同成绩等级会以不同的颜色显示**，家庭用户能更直观地了解学生在考核中的表现。

(2) 班主任需求

家校通为班主任提供了丰富的功能，包括登录系统、查看学生的签到签退记录、发布和查看通知、**事项管理、考核记录、管理用户等**。

签到管理包括查看自己班级的学生签到、签退情况以及时长统计，方便记录用户的活动情况。通过柱状图展示的签到总时长，班主任能够清晰了解签到情况。

通知栏可以进行通知管理，包括发布通知和设置通知级别。

事项管理模块提供了多功能管理，包括请假管理、讨论管理、**提问管理等所有事项**。

考核记录和考核管理模块可以查看和管理考核情况，包括发布考核、评分和撰写评语。

班主任可以通过班级管理功能和成员管理功能对班级信息和成员信息进行管理和修改。

3.2.2 性能需求

有必要根据用户需求建立一个高性能的家校通系统，家校通应该让用户体验更好、网页数据响应更快。以下是需要考虑的一些因素：

(1) 响应时间：家校通应该能够快速响应用户请求，不要让用户等待太久。使用高效的算法和数据结构优化代码以及合理配置服务器来实现。

(2) 数据库性能：为了使系统能够更快地读取和写入数据，以及更有效地查询数据，有必要优化数据库设计和查询方法。

3.2.3 可扩展性需求

可扩展性是确保平台能够满足未来需求和功能的重要因素。以下是一些考虑因素：

(1) 模块化设计：把不同的功能做成独立的模块。如果以后需要增加新的功能就可以很容易地添加新的模块，不会对整个系统产生太大的影响。

(2) 松耦合架构：尽量让家校通中的各个部该尽量独立，彼此之间的联系要尽量少。**如果要添加新功能或者修改现有功能就不会对其他部分造成太大的影响**，也能更容易地扩展系统。

4 系统设计

2.6 系统模块设计

4.1.1 家庭用户模块

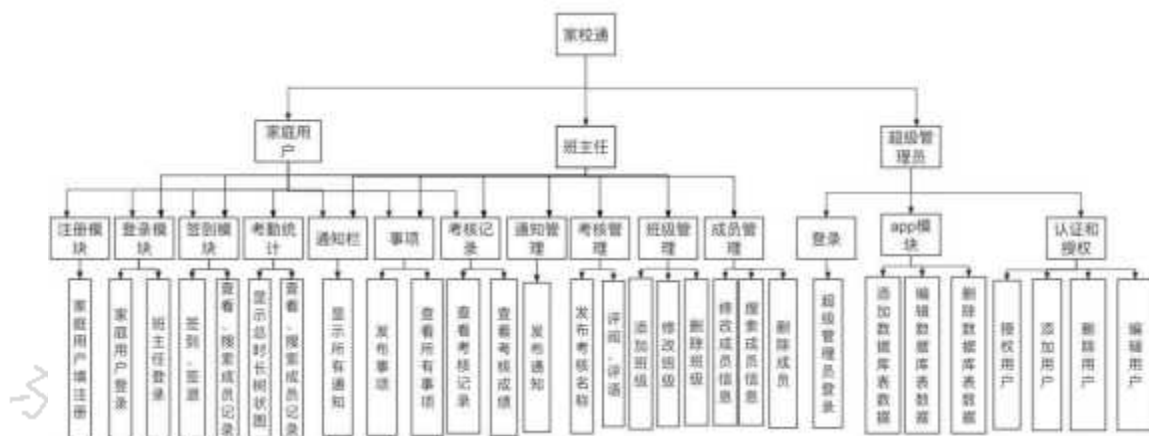
平台提供家庭用户可以使用的功能，注册模块允许用户注册帐户并登录，需要提供必要的信息，包括学生ID、用户名、密码、电话号码、电子邮件和班主任提供的班级代码；在登录模块中验证用户身份并分配权限；考勤功能包括签到和签退；通知栏显示已发布的通知，根据通知级别显示不同的颜色；事件管理模块用于请假、讨论、提问等；考核记录模块显示所有的考核内容和等级，不同的等级用不同的颜色分类。

4.1.2 班主任模块

班主任比家庭用户多了额外的签到统计功能来显示总签到时间；考核管理模块发布评估、评分和撰写评论；班级管理模块对班级信息进行管理；成员管理模块可以对成员进行修改、删除、查询、设置成员类别。

4.1.3 系统功能模块

本系统设计了前台和超级后台两个模块，前台供班主任和家庭用户访问。超级管理功能是Django框架自带的，



4.1.4 核心功能的流程设计

(1) 签到功能

点击签到按钮时，提示是否确认签到，选择“是”签到成功。流程图如图4-2。

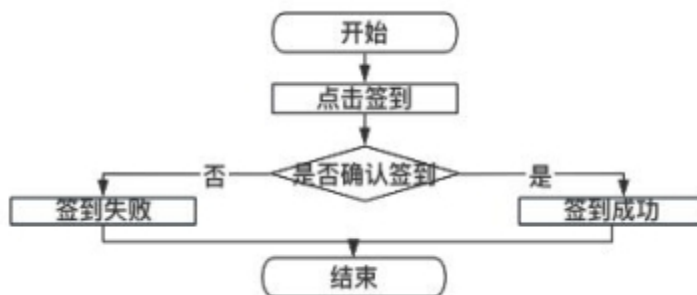


图4-2 签到流程图

(2) 通知管理功能

输入通知内容，系统判断是否有内容，有则发布成功。流程如图4-3。

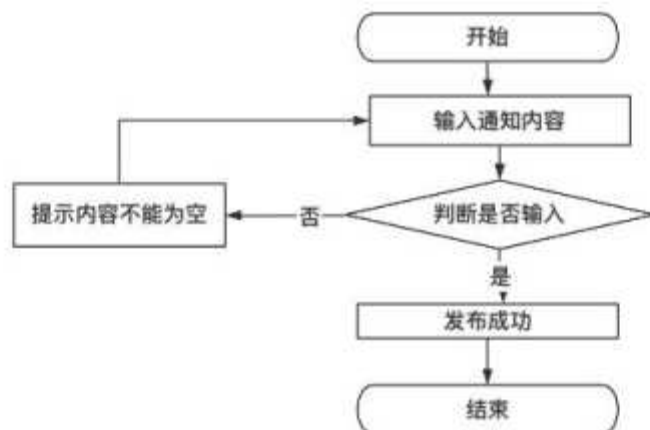
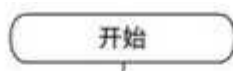


图4-3 发布通知流程图

(3) 考核功能

输入考核主题，然后进行成绩批阅。如果分数格式不对就提示不为十进制数，否则评阅成功。考核流程图如图4-4。



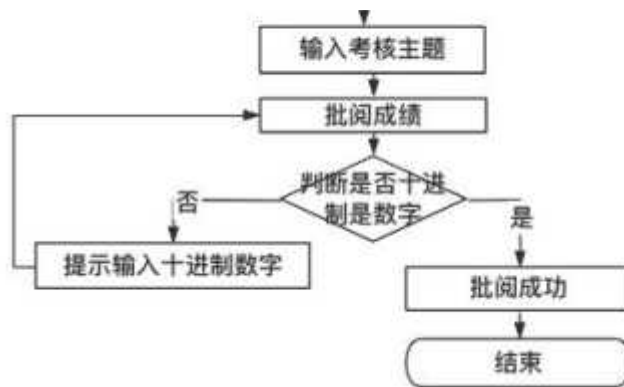


图4-4 考核流程图

(4) 事项功能

输入事项内容，如果事项有内容则可以发布成功。流程如图4-5。

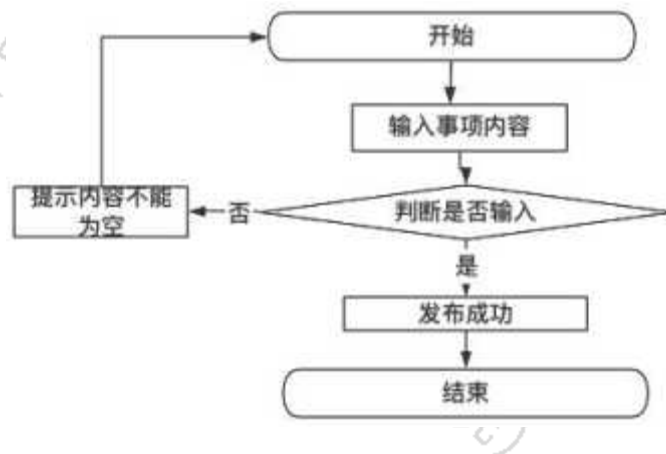


图4-5 事项流程图

(5) 班级管理功能

进行班级管理时，输入班级名称，确认则添加成功。编辑班级名称时，保存则编辑成功。删除时，确认删除班级则删除成功。流程如图4-6。

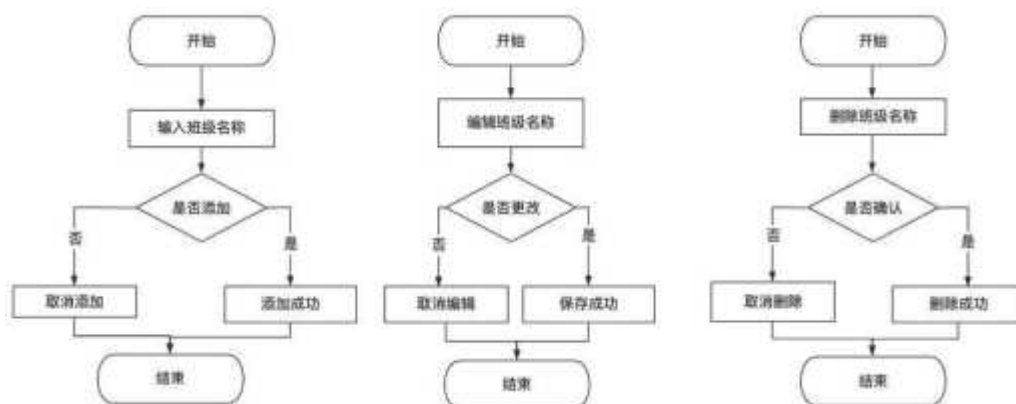


图4-6 班级管理流程图

(6) 成员管理功能

首先搜索成员，对已存在的成员可以进行信息编辑和删除操作。流程图如图4-7。



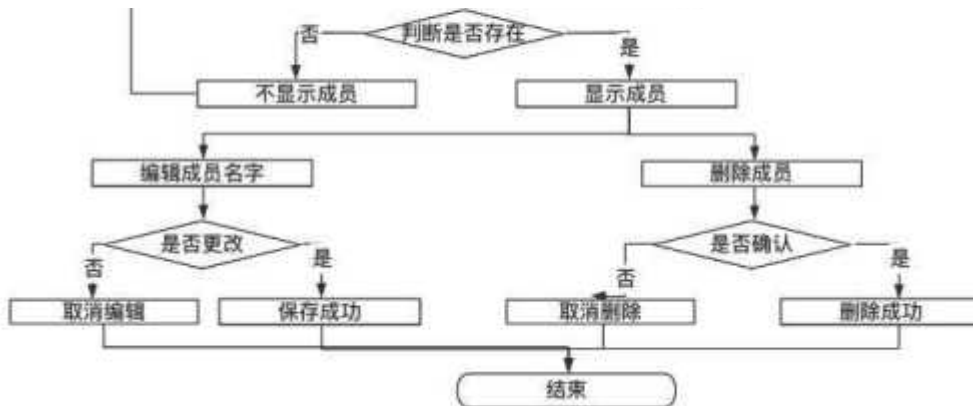


图4-7 成员管理流程图

2.7 数据库设计

2.7.1 数据库设计原则

数据库设计的原则是保证数据的安全，数据要完整准确不能出错。数据随时满足用户需求，数据数量可扩大。数据库需要确保数据的完整性不可分割，一致性是准确和相互关联的，安全性防止未经授权的访问或损坏，可扩展性应对数据量的增长和系统变化。

2.7.2 数据库设计

家校通的E-R图如图4-8所示。

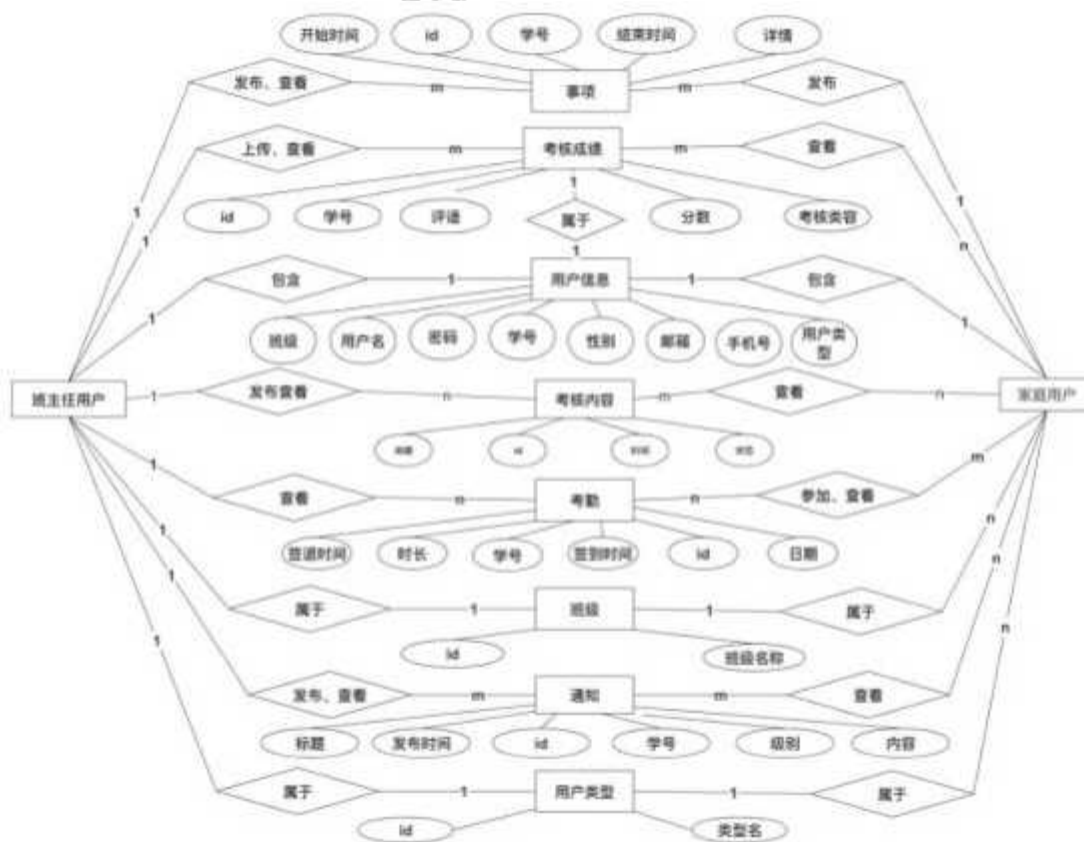


图4-8 系统E-R图

2.7.3 数据库表设计

(1) 用户类型表(app_usertype表)：

该表如表4-1，存储用户类型，包括序号id和用户类型caption。其中，序号id是该表的主键，保证了每个用户

类型的唯一性，当系统登录时，通过此表进行用户类型校验，检查用户的类型，获得不同的功能权限。

表4-1 用户类型表

字段名	数据类型	长度	是否可为空	是否主键	说明
id	INTEGER	4	否	是	序号自动增
caption	varchar	10	否	否	用户类型

(2) 班级表(app_classinfo):

该表如表4-2，存储班级信息，包括班级代码id和班级名name。其中，班级代码id是该表的主键，保证了每个班级的唯一性。

表4-2 班级表

字段名	数据类型	长度	是否可为空	是否主键	说明
id	INTEGER	4	否	是	班级代码
name	varchar	20	否	否	班级名

(3) 用户信息表(app_userinfo):

该表如表4-3，存储用户信息，包括学号studentNum、密码password、用户名username、班级cid、用户类型user_type、手机号phone、邮箱email和性别gender。其中，学号studentNum是该表的主键，保证了每个用户的唯一性，班级cid是表ClassInfo的外键，用户类型user_type是表UserType的外键，当系统登录时，通过此表进行登录校验，检查邮箱和密码是否对应。

表4-3 用户信息表

字段名	数据类型	长度	是否可为空	是否主键	说明
studentNum	varchar	15	否	是	学号
password	varchar	64	否	否	密码
username	varchar	15	否	否	用户名
cid	INTEGER	4	否	否	班级
user_type	INTEGER	4	否	否	用户类型
phone	varchar	11	否	否	手机号
email	varchar	20	允许空	否	邮箱
gender	INTEGER	4	否	否	性别

(4) 考勤表(app_attendance):

该表如表4-4，存储考勤信息，序号id、签到时间start_time、签退时间end_time、时长duration、日期date和学号stu。其中，序号id是该表的主键，保证了每个签到数据的唯一性，学号stu是表UserInfo的外键，当用户点击签到时，通过此表进行签到记录。

表4-4 考勤表

字段名	数据类型	长度	是否可为空	是否主键	说明
id	INTEGER	4	否	是	序号自动增
start_time	datetime	8	否	否	签到时间
end_time	datetime	8	否	否	签退时间
duration	decimal	10	否	否	时长
date	date	3	否	否	日期
stu	varchar	15	否	否	学号

(5) 考核成绩表(app_exam):

该表如表4-5, 存储考核成绩信息, 包括序号id、分数point、评语detail、考核内容content_id和学号user。其中, 序号id是该表的主键, 保证了每个考核成绩的唯一性, 学号user是表UserInfo的外键, 考核内容content是表ExamContent的外键, 系统查看成绩时, 通过此表查看所有考核内容及其分数; 系统提交成绩时, 通过此表储存成绩信息。

表4-5 考核成绩表

字段名	数据类型	长度	是否可为空	是否主键	说明
Id	INTEGER	4	否	是	序号自动增
point	decimal	10	否	否	分数
detail	TEXT	200	否	否	评语
content	INTEGER	4	否	否	考核内容
user	varchar	15	否	否	学号

(6) 考核内容表(app_examcontent):

该表如表4-6, 存储考核内容信息, 包括序号id、标题title、时间date和状态state。其中, 序号id是该表的主键, 保证了每个考核内容的唯一性, 当系统发布考核时, 通过此表进行存储成考核内容。

表4-6 考核内容表

字段名	数据类型	长度	是否可为空	是否主键	说明
Id	INTEGER	4	否	是	序号自动增
title	TEXT	200	否	否	标题
date	date	3	否	否	时间
state	bool	2	否	否	状态

(7) 事项表(app_leave):

该表如表4-7, 存储事项信息, 包括开始时间start_time、结束时间end_time、详情explain、用户名字user和序号id。其中, 序号id是该表的主键, 保证了每个事项的唯一性, 用户user是表UserInfo中username的外键, 当系统发布事项时, 通过此表存储事项和界面显示所有事项。

表4-7 事项表

字段名	数据类型	长度	是否可为空	是否主键	说明
start_time	date	3	否	否	开始时间
end_time	date	3	否	否	结束时间
explain	TEXT	500	否	否	详情
user	varchar	15	否	否	用户
Id	INTEGER	4	否	是	序号自动增

(8) 通知表(app_notice):

该表如表4-8, 该表存储通知信息, 包括序号id、发布时间post_time、标题head、内容content、级别level和用户姓名author。其中, 序号id是该表的主键, 保证了每个通知的唯一性, 用户名author是表UserInfo中username的外键。发布通知时, 通过此表进行通知信息存储等。

表4-8 通知表

字段名	数据类型	长度	是否可为空	是否主键	说明
Id	INTEGER	4	否	是	序号自动增

post_time	datetime	8	否	否	发布时间
head	TEXT	200	否	否	标题
content	TEXT	500	否	否	内容
level	INTEGER	4	否	否	级别
author	varchar	15	否	否	用户名

5 系统实现

5.1 模块开发

模块化开发将整个软件系统分解为独立且功能清晰的模块，使代码结构更清晰，更易于维护。团队成员可以在各自负责的模块中独立工作，从而提高协作效率。模块化设计促进了代码复用，减少了重复代码编写的工作量，提高了开发效率，降低了系统修改或扩展的风险。因为修改的范围更清晰，对整个系统的影响也更小。模块化设计提高了系统的稳定性和可靠性，因为模块之间的耦合度较低，一个模块的问题不会影响其他模块的正常运行。家校通选择模块化开发，旨在构建一个更易于维护、扩展和稳定的软件系统，以满足日益复杂和不断变化的需求。

5.2 用户注册

用户注册界面如图5-1所示，当用户点击“确认注册”按钮时，前端系统调用Ajax向服务器发送注册请求，并将学号、用户名、密码、手机号码、Email发送到后端系统。使用hashlib库的sha1算法对密码进行加密，将加密后的密码和其他信息一起创建一个新的UserInfo对象并设置用户类型为家庭用户，将新创建的对象保存到数据库中。

家校通注册

成员信息注册

学号:

用户名:

密码:

确认密码:

手机号码:

Email:

Class_id:

图5-1 用户注册界面

后端用户注册View核心代码如下：

```
def register_verify(request):
    if request.method == 'POST':
        print('验证成功')
        username = request.POST.get('username')
        email = request.POST.get('email')
        stu_num = request.POST.get('stu_num')
        pwd = request.POST.get('password')
        cid_id = request.POST.get('class')
        m1 = hashlib.sha1()
        m1.update(pwd.encode('utf8'))
        pwd = m1.hexdigest()
        phone = request.POST.get('phone')
        a=UserInfo.objects.create(username=username, email=email,cid_id=cid_id,studentNum=stu_num,
password=pwd,phone=phone, user_type_id=2)
```

a. save()

return HttpResponse('OK')

5.3 用户登录

用户登录界面如图5-2所示，当用户输入邮箱和密码进行用户登录时，如果是POST请求，**获取用户输入的邮箱和密码，对密码进行加密，然后调用check_login函数验证邮箱和密码**。如果验证成功，创建一个重定向对象response，将用户的邮箱和密码分别存储在名为'qwer'和'asdf'的cookie中，并设置这两个cookie的过期时间为3600秒。如果验证失败，将错误信息传递给模板并渲染出登录页面。如果是GET请求，调用check_cookie函数**检查是否存在有效的登录cookie**。如果存在有效的登录cookie，重定向到首页。如果不存在有效的登录cookie，渲染出登录页面。



图5-2 用户登陆界面

后端用户登录的View核心代码如下：

```
def login(request):
    if request.method == 'POST':
        email = request.POST['email']
        password = request.POST['password']
        m1 = hashlib.shal()
        m1.update(password.encode('utf8'))
        password = m1.hexdigest()
        if check_login(email, password):
            response = redirect('/index/')
            response.set_cookie('qwer', email, 3600)
            response.set_cookie('asdf', password, 3600)
        return response
    else:
        return render(request, 'account/page-login.html', {'error_msg': '账号或密码错误请重新输入'})
    else:
        (flag, rank) = check_cookie(request)
        print('flag', flag)
        if flag:
            return redirect('/index/')
            return render(request, 'account/page-login.html', {'error_msg': ''})
```

5.4 签到

签到界面如图5-3所示，当用户签到时，前端设置flag的值，判断变量 flag 是否为真，如果为真则进入下一层条件判断。**判断请求的方法是否为 POST 方法，如果是 POST 方法，则获取请求中的 'sign' 参数的值，如果 sign_flag 的值为 'True'，则创建一个Attendance对象，其中stu属性为 user，start_time 属性为当前时间；如果**

sign_flag的值为'False'，则获取当前用户最后一次签到记录中end_time为空的记录。获取当前时间，并计算出当前时间与最后一次签到记录的开始时间之间的时长（以小时为单位），并保留一位小数。更新最后一次签到记录的end_time 属性为当前时间，并将时长保存到 duration 属性中。



图5-3 签到模块

后端用户签到View核心代码如下：

```
def check(request):
    (flag, rank) = check_cookie(request)
    user = rank
    if flag:
        if request.method == 'POST':
            sign_flag = request.POST.get('sign')
            print('sign_flag', type(sign_flag), sign_flag)
            if sign_flag == 'True':
                Attendance.objects.create(stu=user, start_time=datetime.datetime.now())
            elif sign_flag == 'False':
                cur_attendent = Attendance.objects.filter(stu=user, end_time= None)
                tmp_time = datetime.datetime.now()
                duration = round((tmp_time - cur_attendent.last().start_time).seconds / 3600, 1)
                cur_attendent.update(end_time=tmp_time, duration=duration)
            return HttpResponse(request, '操作成功')
        else:
            pre_att = Attendance.objects.filter(stu=user).order_by('id').last()
            if pre_att:
                if (datetime.datetime.now() - pre_att.start_time.replace(tzinfo= None)).seconds / 3600 > 6 and
                pre_att.end_time == None:
                    pre_att.delete()
                sign_flag = True
                elif (datetime.datetime.now() - pre_att.start_time.replace(tzinfo= None)).seconds / 3600 < 6
                and pre_att.end_time == None:
                    sign_flag = False
            else:
                sign_flag = True
            else:
                sign_flag = False
```

```
sign_flag = True
if user.user_type.caption == '班主任':
    att_list = Attendance.objects.filter(stu__cid=user.cid). order_by('-id')
else:
    att_list = Attendance.objects.filter(stu=user).order_by('-id')
    return render(request, 'attendance/check.html', locals())
    return render(request, 'account/page-login.html', {'error_msg': ''})
```

5.5 考勤统计

考勤分析界面如图5-4所示，当班主任登录时，后端获取当前日期和星期几，计算本周的第一天和最后一天，查询Attendance表中日期在本周范围内的记录，并按照学号、用户名和班级名称进行分组，计算总时长。将查询结果转换为JSON格式，使用`json.dumps()`方法将查询结果列表转换为JSON字符串。使用`HttpResponse()`函数将JSON字符串作为响应内容返回给客户端。渲染名为'total.html'的模板，使用`render()`函数将模板和变量传递给模板引擎进行渲染。使用`HttpResponse()`函数将渲染后的页面作为响应内容返回给客户端。



图5-4 考勤统计模块

后端用户考勤统计View核心代码如下：

```
def total(request):
    (flag, user) = check_cookie(request)
    (flag, rank) = check_cookie(request)
    user = rank
    if rank.user_type.caption == '班主任':
    if request.method == 'POST':
        nowdate = datetime.datetime.now()
        weekDay = datetime.datetime.weekday(nowdate)
        firstDay = nowdate - datetime.timedelta(days=weekDay)
        lastDay = nowdate + datetime.timedelta(days=6 - weekDay)
        info_list = Attendance.objects.filter(date__gte=firstDay, date__lte= lastDay, stu__cid=user.cid).
        values( '\stu', 'stu__username', 'stu__cid__name', ).annotate(total_time=Sum('duration')).order_by()
```

```

info_list = json.dumps(list(info_list), cls=DecimalEncoder)
return HttpResponse(info_list)
else:
nowdate = datetime.datetime.now()
weekDay = datetime.datetime.weekday(nowdate)
firstDay = nowdate - datetime.timedelta(days=weekDay)
lastDay = nowdate + datetime.timedelta(days=6 - weekDay)
info_list = Attendance.objects.filter(date__gte=firstDay, date__lte =lastDay,stu__cid=user.cid).
values('stu', 'stu_username', 'stu_cid_name').annotate(
total_time=Sum('duration')).order_by()
return render(request, 'attendance/total.html', locals())
else:
return render(request, 'denied.html')

```

5.6 通知栏

用户通知栏界面如图5-5所示，当用户查看通知时，该函数的功能是获取所有的Notice对象，并按照发布日期的倒序进行排序，然后将结果存储在info_list变量中。最后，使用render函数将请求、模板路径和局部变量作为参数，返回一个渲染后的HTML页面给用户。



家校通G到勤分析统计粒透知管理可极评定记录考核理事项口理上，成器管理o出营录系备2024年2月26日11:092024月学费收取每人50元整，你为费来直型得用2024年2月25日11:072024.28开字阳标：2024年春季学开学通知教的各位学生，家长和教职员工：我们学地您面布，2024年春季学精的开学日期已，我们湖在个满力和成长约学与相架，共用新的学习挑线阳机造。开学日期和关事境下：1. “开学日期：“2024年2月8日(超五)2. “上课时间：“核各班级和读程的安持、具体上课时间可能合有用不同，请保强前了解的课程时间表，并孩时参加读堂，3.**校历提醒：“请注意校质中的重要日期，如作安排，考试时间等，我们将在学校站上发布校历，供尽参考，4. “胶园安全：“字校将续加强校安全排施，确保生，教职工工和访客的，请守校园安全双，配合学校工作人，5课程安“门为学生供来富5样程择，中们配发展自己的学术和个人力，生门美注程安用盈年学院科：生及关注学，开关活，们在这个中一学生分力，取博优异的。数取员工将金力为学生提供优质的教育资源和服务，助力他们实现自己的学业和职业日标，最后，让我们共网期特着一个满成长，收肤和悦的春李学期!祝源大家在新的学刚中取得成功!需与学检长办公室电上名)日期：2024年2月26日来自4主任的

图5-5 通知栏

后端用户通知栏View核心代码如下：

```

@is_login
def notice(request):
(flag, user) = check_cookie(request)
user_class = user.cid

```



```
info_list=Notice.objects.filter(author__cid=user_class).order_by('-post_date')
return render(request, 'notice/notice.html', locals())
```

5.7 通知栏管理

通知发布界面如图5-6所示，当用户发布通知时，检查用户是否已经登录，如果没有登录则会跳转到登录页面。检查用户是否为班主任，如果是，则会根据请求方法创建通知或者显示通知管理页面；如果不是班主任，则会显示无权访问通知管理页面，如图5-7。

图5-6 发布通知



图5-7 学生用户在此界面无权限

后端用户通知栏管理View核心代码如下：

```
@is_login
def noticeManage(request):
    (flag, user) = check_cookie(request)
    if user.user_type.caption == '班主任':
        if request.method == 'POST':
            title = request.POST.get('title')
            content = request.POST.get('content')
            level = request.POST.get('selectLevel')
            Notice.objects.create(head=title, content=content, level=level, author=user)
            return render(request, 'notice/notice_manage.html')
        else:
```

```
return render(request, 'notice/notice_manage.html')
else:
return render(request, 'denied.html')
```

5.8 考核管理

考核管理界面如图5-8、图5-9所示，当用户发布考核主题和考核批阅时，代码判断用户类型是否为班主任，如果是，则继续执行相应的操作；如果不是，则返回相应的无权访问页面如图5-10。

对于考核管理，根据请求方法创建考核内容或者显示考核管理页面。如果请求方法为 POST，则会获取请求中的 title 参数，并根据 title 创建一个 ExamContent 对象。如果 title 为空，则获取所有用户的数量，并获取请求中的 exam_id 参数。然后，使用一个循环来获取每个用户的得分、学生ID 和详情，并根据这些信息创建一个 Exam 对象。最后，通过 ExamContent 的 filter 方法将对应的考核内容的状态更新为 True。最后将考核内容列表和用户列表传递给相应的模板进行渲染。

图5-8 发布考核

图5-9 考核打分和写评语





图5-10 学生无权限操作

后端用户考核管理View核心代码如下：

```
def exam_manage(request):
    (flag, user) = check_cookie(request)
    if user.user_type.caption == '班主任':
        if request.method == 'POST':
            title = request.POST.get('title')
            if title:
                ExamContent.objects.create(title=title)
            else:
                count = UserInfo.objects.filter(cid=user.cid).count()
                content_id = request.POST.get('exam_id')
                for i in range(count):
                    point = request.POST.get('point{}'.format(i))
                    stuID = request.POST.get('stu{}'.format(i))
                    detail = request.POST.get('detail{}'.format(i))
                    Exam.objects.create(point=point, content_id=content_id, user_id=stuID, detail=detail)
                    ExamContent.objects.filter(id=content_id).update(state=True)
                    check_list = ExamContent.objects.filter(state=False).order_by('-id')
                    user_list = UserInfo.objects.filter(cid=user.cid).order_by('studentNum')
                    return render(request, 'exam/exam_manage.html', locals())
                else:
                    return render(request, 'denied.html')
```

5.9 考核评定记录

用户所有考核名称如图5-11所示，当用户查看时，根据请求的查询参数 exam_id 来查询相关的考试内容和考试结果，如图5-12，并根据最高分（班主任设置默认为满分）计算每个结果的分数比例，然后将查询到的数据和定义好的分数范围列表传递给模板进行渲染。

考核评定记录 总分	
在此输入院系考核的名称...	
考核名称	日期
2023下半年五年级四班数学期末考试成绩	2024年2月26日
2023下半年五年级四班语文期末考试成绩	2024年2月26日
2023下半年五年级四班英语期末考试成绩	2024年2月26日
2023下半年五年级四班科学期末考试成绩	2024年2月27日

图5-11 所有考核主题

考核评定记录 总分100	
100%	100%

姓名	学号	考核名称	成绩	评语
殷海	202003031427	2023下半年五年級四班数学期末考试成绩	96	优秀
良好(80%-89%)				
姓名	学号	考核名称	成绩	评语
李臣	202003031424	2023下半年五年級四班数学期末考试成绩	87	有进步
杜金辉	202003031426	2023下半年五年級四班数学期末考试成绩	80	
及格(60%-79%)				
姓名	学号	考核名称	成绩	评语
曹国栋	202003031413	2023下半年五年級四班数学期末考试成绩	79	
石玉行	202003031425	2023下半年五年級四班数学期末考试成绩	68	退步严重
不及格(0-59%)				
姓名	学号	考核名称	成绩	评语

图5-12 考核分数详情

后端考核记录View核心代码如下：

```
def exam(request):
    exam_list = ExamContent.objects.all()
    exam_id = request.GET.get('exam_id')
    if exam_id:
        user_list = Exam.objects.filter(content_id=exam_id).all()
        total_point = max(item.point for item in user_list)
        ratio_list = [item.point / total_point for item in user_list]
        user_list_with_ratio = list(zip(user_list, ratio_list))
        return render(request, 'exam/exam.html', locals())
```

5.10 事项

用户事项界面如图5-13所示，当用户查看或发布事项时，获取所有的事项记录，如果请求方法是POST，表示用户提交了事项表单，执行以下操作：获取开始时间和结束时间，并打印开始时间。根据开始时间和结束时间计算事项天数。获取事项说明。创建一个新的事项记录，并将开始时间、结束时间、用户和说明保存到数据库中。最后，渲染leave.html模板，并将请求对象和局部变量传递给模板进行渲染。

请假/申请/问题/回复/记录

选择日期: 2024-03-19

结束时间: 2024-03-20

原因事由: (如果不差请保持默认时间, 请前请正确的时间)

批复回应:

提交

请假/申请/问题/回复/记录 记录 (对于批复, 管理員应同意颜色显示绿色, 不同意显示红色)

用户: 李臣
时间: 2024年2月28日—2024年2月28日
原因/回应: 车程下午到半程

用户: 石玉行
时间: 2024年3月3日—2024年3月6日
原因/回应:



图5-13 事项

后端事项View核心代码如下：

```
def leave(request):
    (flag, user) = check_cookie(request)
    user_class = user.cid
    leave_list = Leave.objects.filter(user__cid=user_class)
    if request.method == 'POST':
        starttime = request.POST.get('starttime')
        endtime = request.POST.get('endtime')
        print(starttime)
        a = int(datetime.datetime.strptime(starttime, '%Y-%m-%d').day - datetime.datetime.strptime(endtime, '%Y-%m-%d').day)+1
        explain = request.POST.get('explain')
        Leave.objects.create(start_time=starttime, end_time=endtime, user=user, explain=explain)
        return render(request, 'attendance/leave.html', locals())
```

5.11 班级管理

班级界面如图5-14所示，当用户查看班级时后端根据cookie验证的结果和用户的权限，渲染不同的模板页面。如果验证通过且用户权限为班主任，则显示所有的ClassInfo对象；如果验证通过但用户权限不为班主任，则显示一个拒绝访问的页面如图5-15；如果验证未通过则显示登录页面。



图5-14 所有班级信息



图5-15 非班主任用户类型无权限操作本模块

当用户编辑班级时，根据请求中的参数add_class_name，判断是否已经存在同名班级。如果存在则返回‘班级已存在’；如果不存在则将该班级添加到数据库中，并返回‘添加成功’。

后端编辑班级View核心代码如下：

```
def edit_class(request):
    (flag, rank) = check_cookie(request)
    print('flag', flag)
    if flag:
        if rank.user_type.caption == '班主任':
            if request.method == 'POST':
                pre_edit_id = request.POST.get('edit_id')
                class_name = request.POST.get('edit_class_name')
                temp_flag = ClassInfo.objects.filter(name=class_name)
            print('pre_edit_id', pre_edit_id)
            pre_obj = ClassInfo.objects.get(id=pre_edit_id)
            if not temp_flag and class_name:
                pre_obj.name = class_name
                pre_obj.save()
            return HttpResponseRedirect('班级修改成功')
            class_list = ClassInfo.objects.all()
            return render(request, 'class/classManage.html', {'class_list': class_list})
        # return HttpResponseRedirect('编辑班级')
    else:
        return render(request, 'denied.html')
    else:
        return render(request, 'account/page-login.html', {'error_msg': ''})
```

5.12 成员管理

成员管理界面如图5-16、图5-17所示，当用户删除成员时，后端处理成员信息的删除请求，只有班主任才有限执行删除操作。当用户查看成员时，系统调用Ajax向服务器发送请求，并将学号、用户名、密码、手机号码、Email发送到前端系统。如果验证通过但用户权限不为班主任，则显示一个拒绝访问的页面如图5-18。

管理认证成功

成员管理

通过学号或名字搜索

查询

学号	班级	姓名	操作
1	五年级4班	4班主任	<div>编辑</div> <div>删除</div>
20203031425	五年级4班	石玉行	<div>编辑</div> <div>删除</div>
20203031427	五年级4班	殷海	<div>编辑</div> <div>删除</div>
20203031424	五年级4班	李臣	<div>编辑</div> <div>删除</div>
20203031426	五年级4班	杜金辉	<div>编辑</div> <div>删除</div>
20203031413	五年级4班	曹国栋	<div>编辑</div> <div>删除</div>

1

图5-16 成员信息管理

成员管理

学号

20203031427

姓名

殷海

Email

20203031427@stu.com

密码

7c4a8d09ca3762af61e59520943dc26494f8941b

班级

五年级4班

用户类型

家庭用户

性别

☒ 男
 ☐ 女

联系方式

15355905540

确定

返回

图5-17 成员编辑





图5-18非班主任用户类型无权限操作本模块

当用户编辑成员时，通过检查用户的cookie来验证用户是否登录。如果验证通过，则进一步判断用户的身份是否为班主任；如果是班主任并且请求方法是POST，则获取表单中的学生编号、用户名、电子邮件、性别、班级、用户类型和电话等信息。

根据学生编号在数据库中查找相应的用户对象，并将获取到的信息更新到该对象中。获取所有用户信息的列表，并将其传递给名为“member_list”的变量。重定向到“/memberManage/”页面，并将“member_list”作为参数传递。如果请求方法不是POST，则获取名为“edit_sno”的参数，该参数表示要编辑的学生编号。获取所有用户类型和班级的列表。

根据学生编号在数据库中查找相应的用户对象，并将其赋值给变量“edit_stu_obj”。渲染名为“edit_member.html”的模板，并传递当前的请求对象和局部变量。如果用户的身份不是班主任，则渲染名为“member_manage_denied.html”的模板。如果用户的cookie验证未通过，则渲染名为“page-login.html”的模板，并传递一个空的错误消息。

后端编辑成员View核心代码如下：

```
def edit_member(request):
    (flag, rank) = check_cookie(request)
    if flag:
        if rank.user_type.caption == '班主任':
            if request.method == 'POST':
                student_num = request.POST.get('student_num')
                username = request.POST.get('username')
                email = request.POST.get('email')
                gender = int(request.POST.get('gender'))
                cls = ClassInfo.objects.get(name=request.POST.get('cls'))
                usertype = UserType.objects.get(caption=request.POST.get('user_type'))
                phone = request.POST.get('phone')
                password = request.POST.get('password')
                edit_obj = UserInfo.objects.filter(studentNum=student_num)
                passq = edit_obj.values('password').first()
                if password==passq['password']:
                    edit_obj.update(studentNum=student_num, username= username, email=email, cid=cls,
user_type=usertype, gender=gender, phone=phone,)
                else:
                    a = hashlib.sh1(password.encode("utf-8")).hexdigest()
                    edit_obj.update(studentNum=student_num, username=user name, email=email, password=a, cid=cls,
user_type=use rtype, gender=gender, phone=phone,)
                    member_list = UserInfo.objects.all()
                    return redirect('/memberManage/', {'member_list':member_li st})
            else:
                edit_member_id = request.GET.get('edit_sno')
                stu_type_list = UserType.objects.all()
                cls_list = ClassInfo.objects.all()
                edit_stu_obj = UserInfo.objects.get(studentNum= edit_member_id)
```

```

return render(request, 'member/edit_member.html', locals())
else:
return render(request, 'denied.html')
else:
return render(request, 'account/page-login.html', {'error_msg': ''})

```

5.13 Django超级管理员后台

Django超级管理员后台登录界面如图5-19所示，Django超级管理员后台管理界面如图5-20所示。



图5-19 Django超级管理员后台登录

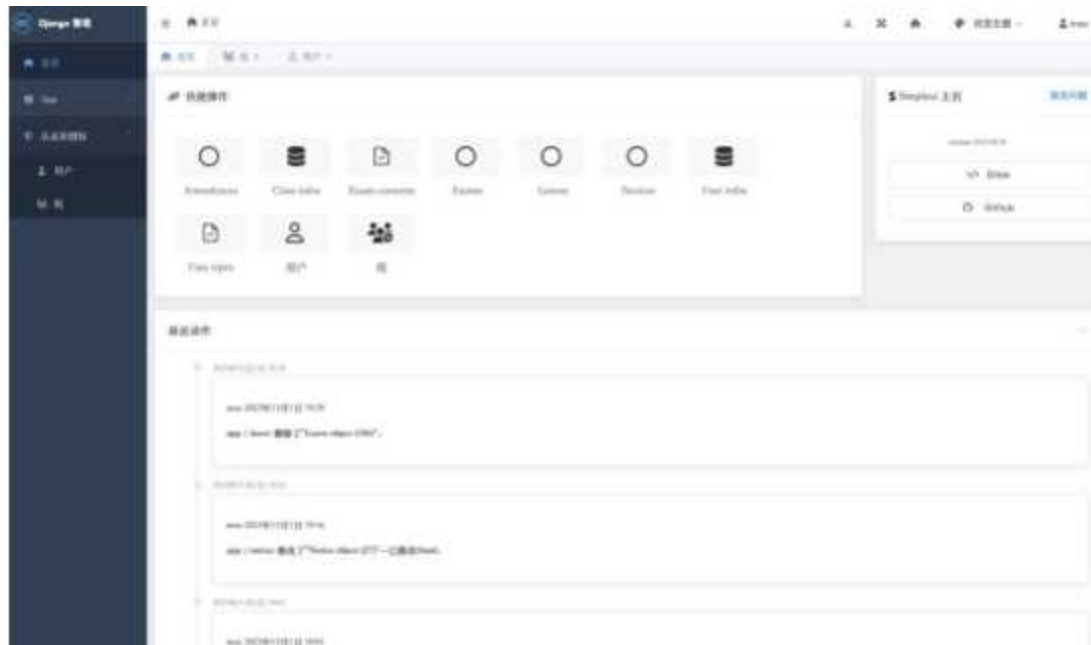


图5-20 Django超级管理员后台主界面

可以通过运行命令`python manage.py createsuperuser`来创建超级管理员。超级管理员具有最高权限，可以管理其他用户、数据、组等^[11]。

后端配置注册admin核心代码如下：

```
from django.contrib import admin

from .models import UserInfo, UserType, ClassInfo, Attendance, Notice, Leave, Exam, ExamContent

class UserInfoAdmin(admin.ModelAdmin):
    list_display = ['studentNum', 'username', 'cid', 'password', 'gender', 'phone', 'email',]

class UserTypeAdmin(admin.ModelAdmin):
    list_display = ['id', 'caption']

class ClassInfoAdmin(admin.ModelAdmin):
    list_display = ['id', 'name']

class MajorInfoAdmin(admin.ModelAdmin):
    list_display = ['id', 'name',]

class AttendanceAdmin(admin.ModelAdmin):
    list_display = ['id', 'stu', 'date', 'start_time', 'end_time', 'is_leave', 'duration', 'detail']

class NoticeAdmin(admin.ModelAdmin):
    list_display = ['id', 'author', 'head', 'content', 'level']

class LeaveAdmin(admin.ModelAdmin):
    list_display = ['id', 'start_time', 'end_time', 'explain']

class ExamAdmin(admin.ModelAdmin):
    list_display = ['id', 'user', 'content', 'point', 'detail']

class ExamContentAdmin(admin.ModelAdmin):
    list_display = ['id', 'title', 'date', 'state']

admin.site.register(UserType, UserTypeAdmin)
admin.site.register(UserInfo, UserInfoAdmin)
admin.site.register(ClassInfo, ClassInfoAdmin)
admin.site.register(Attendance, AttendanceAdmin)
admin.site.register(Notice, NoticeAdmin)
admin.site.register(Leave, LeaveAdmin)
admin.site.register(ExamContent, ExamContentAdmin)
admin.site.register(Exam, ExamAdmin)
```

结 论

本文旨在设计和实现一个基于Django的家校通系统。系统以Django框架为后端，Bootstrap框架为前端，Sqlite为数据库，成功构建了一个功能强大、易用、美观的家校通，为家庭用户提供了丰富的功能，为班主任提供了教育沟通和管理工具。

本研究验证了使用Django框架构建家校通的可行性和优势。Django提供了一种快速高效的构建web应用程序的方法，具有强大的安全性和可扩展性，非常适合家庭-学校沟通的平台。本研究充分利用了Django的模型、视图、模板、表单等功能和特性确保平台的可靠性和可维护性，通过优化前端搜索，提高了响应速度，减轻了服务器负担。

家庭用户在家校通中可以注册、登录和注销登录，还可以查看班级发布的通知、讨论事项、学生成绩。班主任用户具有考勤统计功能，设计的签到签退功能方便班主任了解学生的上学和放学情况、是否安全到家，用柱状图显示学生的考勤时长情况；还可以发布通知，参加事项讨论，管理班级和成员，发布考试和成绩登记等。这些功能有助于信息共享和协作效率以及学习管理，促进学生的参与和成长。

综上，本文开发的家校通提供了一个综合平台，家庭用户可以更方便地参与讨论和查看成绩，班主任可以更好地管理学习任务并评估学生的表现，将有助于提高课堂的整体学习效率和动力。

当然，本系统还有一些局限性，比如设计的数据库目前只能支持一个学校使用，未来还将考虑支持多个学校；还将考虑成绩由Excel文件导入，把通知发送到手机短信上等等。

致 谢

在完成这篇家校通论文的过程中，我要向许多人表示最深切的感谢。他们的支持、鼓励和帮助是我成功完成这项研究工作的重要动力和资源。

我要衷心感谢我的指导老师余恬。他的专业知识、丰富的经验和深入的见解对我的研究起到了至关重要的推动

作用。他耐心地回答我的问题，引导我思考更深层次的问题，帮助我克服困难。在每一次讨论中，他都给了我宝贵的建议，帮助我不断提高研究水平。没有他的支持，我无法完成这篇论文。

我要感谢我的同学们，特别是那些和我一起参与研究的同学们。我们一起工作，合作交流，相互激励和监督。他们的智慧、勇气和友谊让我感到无比幸运和自豪。我们相互支持、鼓励、共同成长，形成深厚的团队精神。

我要对参与访谈和研究的学生们和老师们表示感谢。他们的意见为我提供了有理的支持。他们的参加让我的研究更加符合社会。

感谢我的家人和朋友。他们的支持是我前进的动力，他们的陪伴和鼓励让我信心十足。

参考文献

- [1]Swantje D ,Sittipan Y ,Kathrin J .Antecedents and Outcomes of Parental Homework Involvement: How Do Family-School Partnerships Affect Parental Homework Involvement and Student Outcomes?[J]. Frontiers in psychology, 2019, 101048.
- [2]邱睿. 基于微信小程序的家校通系统开发与实现[D]. 上海师范大学, 2019.
- [3]汪时冲, 张鸽. 大学生手机日常使用情况调查与分析——以某市属高校为例[J]. 教育现代化, 2018, 5 (53) :324-327.
- [4]Xiya Y ,Xianhe L ,Changping W , et al.Design and Deployment of Django-based Housing Information Management System[J]. Journal of Physics: Conference Series, 2023, 2425 (1).
- [5]赵丙秀. Bootstrap5的定制化研究[J]. 电脑编程技巧与维护, 2023, (09).
- [6]夏小翔. 基于Echarts学生成绩管理系统设计[J]. 鄂州大学学报, 2023, 30 (05).
- [7]郭鹤楠. 基于Django和Python技术的网站设计与实现[J]. 数字通信世界, 2023, (06)..
- [8]邱红丽, 张舒雅. 基于Django框架的web项目开发研究[J]. 科学技术创新, 2021, (27).
- [9]Junlong S .SQLite Encryption Method for Embedded Databases Based on Chaos Algorithm[J]. Journal of Applied Mathematics, 2023, 2023.
- [10] 蹇常林. ORM在Django操作数据库中的应用[J]. 技术与市场, 2020, 27 (01).
- [11] Anaconda Inc. ; Anaconda and JetBrains Join Forces to Launch 'PyCharm for Anaconda' [J]. Journal of Engineering, 2019.
- [12] 梁国达. 基于VMware+Navicat实现空管自动化的基础数据迁移[J]. 科学技术创新, 2020, (08).

相似片段说明

相似片段中“综合”包括：《中文主要报纸全文数据库》《中国专利特色数据库》《中国主要会议论文特色数据库》《港澳台文献资源》《图书资源》《维普优先出版论文全文数据库》《年鉴资源》《古籍文献资源》《IPUB原创作品》

须知

- 1、报告编号系送检论文检测报告在本系统中的唯一编号。
- 2、本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

客服热线：400-607-5550、客服QQ：4006075550、客服邮箱：vpcs@fanyu.com

唯一官方网站：<https://vpcs.fanyu.com>



关注微信公众号