

数据库系统课程设计报告

题目：微信扫码点名系统

小组成员：周璇、李明泽、李颖恒、卢子诚

1 小组分工

周璇 22307140088：E-R 图设计、系统部分功能实现、期末汇报 ppt 制作、期末报告撰写

李明泽 22307140089：数据库设计、期中汇报 ppt+ 讲稿、系统部分功能实现、期末报告撰写

李颖恒 23307110439：完善数据库测试数据、数据流图设计、期末报告完善

卢子诚 23307110086：确定项目程序框架、签到二维码发布和扫码等功能实现、部分后端接口和前端页面设计、期中与期末汇报

2 题目介绍和需求分析

2.1 题目介绍

移动互联网时代，传统课堂点名方式（如纸质签到、口头点名等）效率低、且难以统计。我们利用微信这一普及度极高的社交软件，结合数据库技术，开发了一套扫码点名系统，旨在提升教学管理效率，满足教学数字化需求。

2.2 需求分析

本小组旨在实现一款用于老师考勤学生签到的微信小程序。首先，我们从用户角度出发，对用户的需求进行挖掘，小程序的主要用户为学生、老师和管理员，各个用户的核心需求如表 1。

表 1: 用户需求表

用户	核心需求
学生	扫码签到、查询某一课程的签到情况、查看课表
教师	发布签到(设定开始时间、签到时长等)、查询学生出勤记录、登记学生请假情况、导出学生出勤记录、多维度数据分析
管理员	批量导入和导出学生、教师、课程、学生选课及用户信息

系统安全性需求主要有两方面：一是要保证学生的签到是合乎考勤条件的，即只有在规定时间内才能签到；二是要保证数据库中密码等敏感数据不会被非法读写，我们需要将密码进行哈希处理，数据库中不保存明码。

2.3 数据流图

讨论完用户的需求之后，我们作出数据流图如图 1。

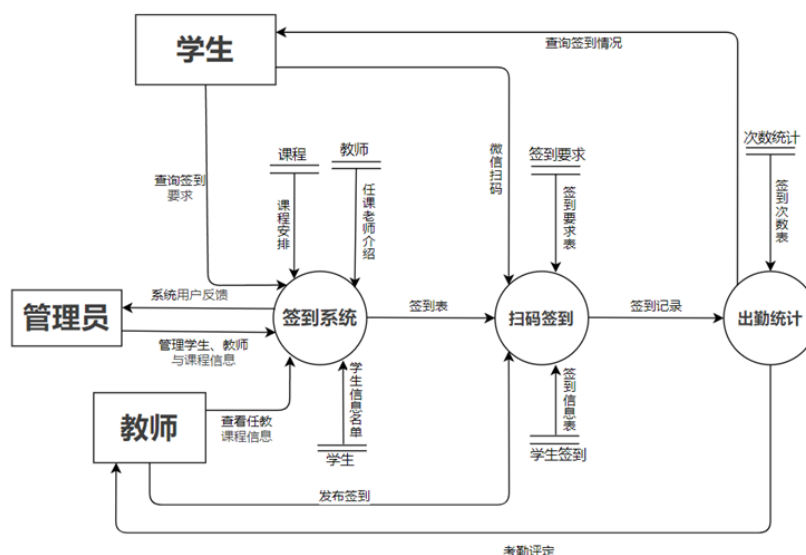


图 1: 数据流图

3 系统设计

3.1 开发平台及框架

3.1.1 Flask

Flask 是一个微型的 Python 开发的 Web 框架，没有默认使用的数据库、窗体验证工具。Flask 框架的核心构成比较简单，但具有很强的扩展性、兼容性，用户可以根据自己的需求实现功能，快速搭建一个网站或 Web 服务（摘自[百度百科](#)）。在我们小组的设计中,Flask 被用于搭建后端对 mysql 数据库进行管理。

3.1.2 微信开发者工具

微信开发者工具是微信官方提供的针对微信小程序的开发工具，集成了开发调试、代码编辑及程序发布等功能，帮助开发者简单和高效地开发微信小程序（摘自[CSDN](#)）。在我们小组的设计中，微信开发者工具被用于搭建前端的用户交互界面以及与后端进行交互。

3.1.3 MySQL

MySQL 是瑞典的 MySQL AB 公司开发的关系数据库系统，采用客户机/服务器架构，可用于 Unix、Linux、Windows 等各种流行操作系统平台，完全适用于网络，用其建造的数据库可在因特网上的任何地方访问，MySQL 具有功能强、使用简单、管理方便、运行速度快、可靠性高、安全保密性强等优点（摘自[CSDN](#)）。在我们小组的设计中，MySQL 是我们选用的数据库，用于存储数据以及数据处理。

3.2 功能规划

3.2.1 主体功能

1. 教师端核心功能:

发起签到任务：设置签到时间、地点、有效时长及定位要求，生成签到二维码。

管理考勤记录：查看历史签到数据、修改学生缺勤状态（如标记请假）、下载考勤报表。

多维度统计：按年级、性别、院系等维度分析学生出勤情况，生成可视化统计结果。

2. 学生端核心功能:

扫码签到：扫描教师生成的二维码，结合定位信息完成签到（距离限制 ≤ 100 米）。

查看考勤记录：查询个人在各课程中的出勤、请假、缺勤历史及统计数据。

3. 管理员端核心功能:

批量数据管理：导入 / 导出学生、教师、课程及考勤数据（如 Excel 格式）。

修改/批量删除信息：支持根据学生、教师、课程 id 搜索修改和删除相关信息，也支持批量删除信息。

3.3 算法设计

3.3.1 登录算法

用户在前端输入账号密码，经本地校验非空后调用 API(POST /login)；在后端，程序首先验证参数有效性，再查询数据库匹配账号，并进行密码加密比对 (BCrypt) 并确认用户身份，若匹配成功，则会返回成功状态和用户信息，若失败则会返回“账号或密码错误”；登录成功则会存储用户身份及其信息到 globalData，跳转首页，若失败则会显示错误提示。

3.3.2 课程加载算法

系统提供两个课程加载接口。

1. 完整课表查询接口：当接收到学生提交的学生 ID 后，系统会从数据库中提取该学生的选课记录，然后将选课记录与课程信息、教师信息进行关联查询，从而获取到该学生所选课程的详细信息，并返回查询成功的状态。

2. 今日课程查询接口：首先系统会获取当前的日期，确定今天是星期几，然后根据学生 ID 在选课记录中筛选出当天的课程。如果没有找到课程，返回失败状态；如果找到课程，就将课程名称、教师 ID、上课时间和上课地点等信息整理成 JSON 格式的数据，并返回查询成功的状态。

3.3.3 课程考勤查看算法

当用户选中一门课并进入其历史签到记录选项，前端应将用户 id 传入后端，后端返回通过执行 SQL 语句的记录数据，数据存于前端的局部变量中用于展示：用户为老师时，汇总展示学生的历史出勤情况；用户为学生时，展示学生本人的出勤情况。此外我们还在前端对数据做了进一步处理：老师端界面可选课程代码和考勤日期查询相关学生出勤情况，学生端则会有一个自己整体出勤情况的统计，点击课程名称可查看该门课的所有历史考勤信息。

3.3.4 教师发起签到算法

教师设定签到参数（课程、签到时间、地点、时长），前端生成包含课程信息的二维码（一次课只能发布一次签到），采用 QR Code 编码标准，二维码有效期与 endTime 绑定。后端将前端传回的签到信息更新至签到要求表，并在后端设置定时任务与自动更新逻辑：学生在规定时间内签到，则在考勤记录表中记录为出勤；若时间截止，系统将触发定时任务，遍历课程学生表，对未签到的学生在考勤记录中标记为缺勤，同时更新考勤次数表。前端显示二维码，供学生扫码完成签到操作。

3.3.5 学生签到算法

学生签到接口在接收到包含学生 ID、课程 ID、签到日期、开始 / 结束时间等必填字段的 POST 请求后，先校验字段完整性，缺失则返回“签到码无效”；接着解析日期时间参数，查询数据库验证该课程当天是否存在签到要求，若不存在则返回“签到要求不存在”，同时检查学生是否已签到，重复则返回“已经签到过了”；然后通过当前时间与签到时间段对比验证时间有效性，早于开始时间或晚于结束时间分别返回“签到时间未到”或“签到时间已过”；全部验证通过后，根据时间和位置状态设置签到状态（成功为 checked，失败为 absent）。最后将签到日志存入数据库并更新学生的签到统计数据（成功则实际签到次数加 1，首次签到则创建统计记录），若过程中出现异常则回滚事务并返回“签到失败”，最终返回包含签到结果、状态描述及状态码的响应。

3.3.6 缺勤查询与修改算法

教师在登记请假界面，根据课程 ID 和日期，查询该课程某天出勤状态为“缺勤”的记录，若未找到符合条件的记录，前端显示“没有找到缺勤学生”；若找到记录，则可以选择（单选或多选）缺勤的学生，将学生的出勤状态由“缺勤”改为“请假”。若过程中出现异常则回滚事务，返回包含错误详情的失败响应

3.3.7 多维度统计算法

教师在统计界面选择查询维度（年级/性别/院系），系统会向后端传入课程代码以及维度，后端及数据库处理后返回相应维度的统计结果并将之以柱状图的形式可视化。

3.4 数据库设计

3.4.1 实体及联系

实体：

1. 学生属性：学号、姓名、性别、院系、专业、年级（入学年份）、出生年月
2. 课程属性：课程代码、课程名称、学分、开课院系、上课地点、课程时间
3. 出勤记录属性：出勤记录编号、学生学号、课程代码、出勤日期、出勤情况、签到时间
4. 教师属性：教师编号、教师姓名、性别、所属院系、出生年月、职称
5. 管理员属性：管理员编号、用户名、密码

关系：

1. 学生与课程：多对多关系（n:m）。
2. 学生与出勤记录：一对多关系（1:n）。
3. 课程与出勤记录：一对多关系（1:n）。
4. 教师与课程：多对多关系（n:m）。
5. 管理员与系统：一对多关系（1:n）。

3.4.2 E-R 图

结合需求分析和我们确定的实体，绘制 E-R 图如下：

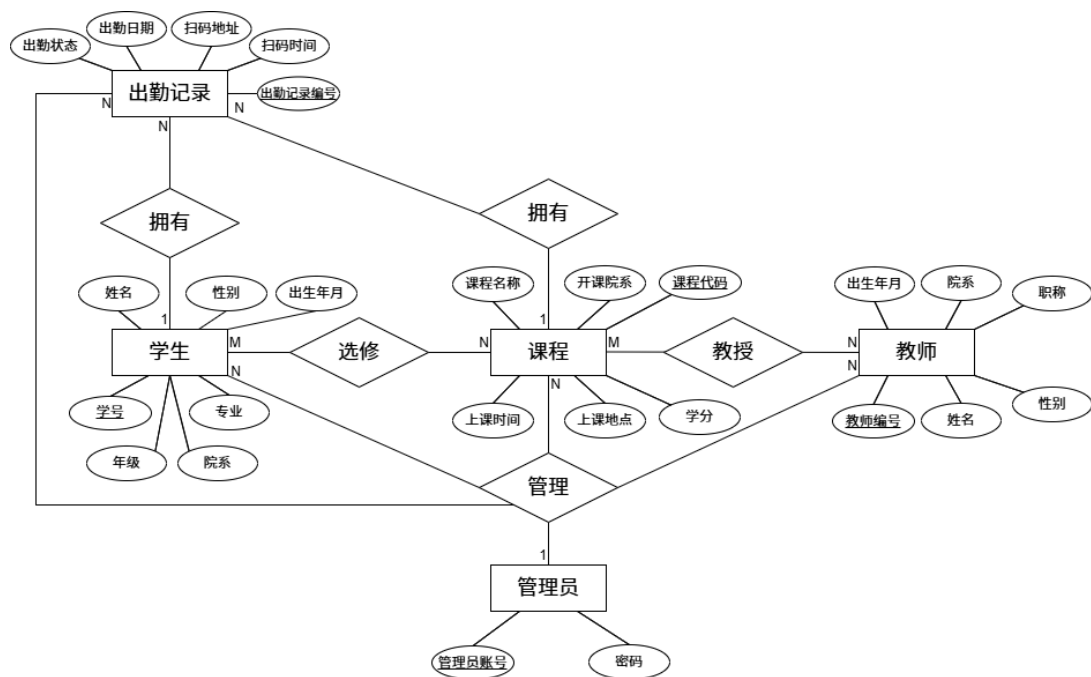


图 2: E-R 图

3.4.3 关系模式

根据 E-R 图，我们建立的关系模式如下。

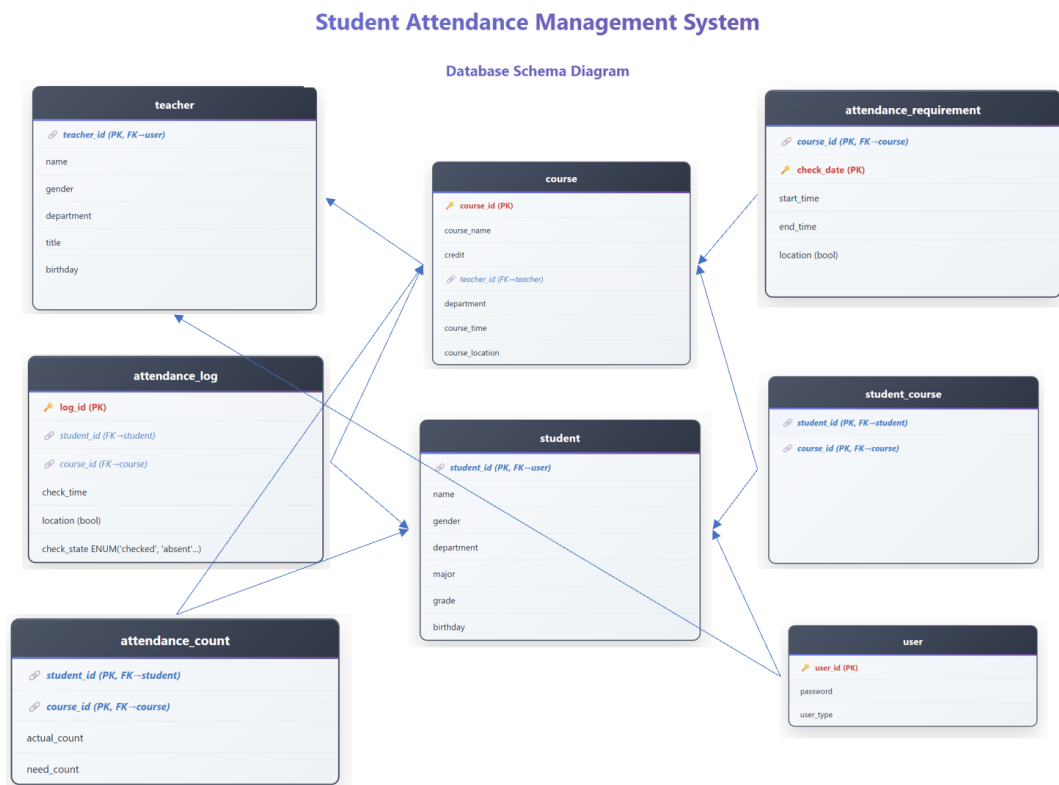


图 3: 数据库模型图

3.4.4 数据库表的建立

1. 学生表

列名	数据类型	说明
学号 (student_id)	varchar	主键, 外键连接用户表
姓名 (name)	varchar	Not null
性别 (gender)	char	Not null
院系 (department)	varchar	Not null, Index
专业 (major)	varchar	Not null
年级 (grade)	varchar	Not null, Index(实际为入学年份)
出生年月 (birthday)	Date	

2. 课程表

列名	数据类型	说明
课程代码 (course_id)	varchar	主键
课程名称 (course_name)	varchar	Not null
学分 (credit)	int	Not null default 0
教师号 (teacher_id)	varchar	Not null, 外键连接教师表
院系 (department)	varchar	Not null
上课时间 (time)	datetime	Not null
上课地址 (Course_location)	varchar	Not null

3. 教师表

列名	数据类型	说明
教师号 (teacher_id)	varchar	主键, 外键连接用户表
教师姓名 (name)	varchar	Not null
出生年月 (birthday)	Date	
性别 (gender)	varchar	Not null
院系 (department)	varchar	Not null
职称 (title)	varchar	Not null

4. 学生选课表

列名	数据类型	说明
课程号 (course_id)	varchar	联合主键, 外键连接课程表, Index
学生号 (student_id)	varchar	联合主键, 外键连接学生表

5. 学生考勤统计表

列名	数据类型	说明
学号 (student_id)	varchar	联合主键, 外键连接学生表
课程代码 (course_id)	varchar	联合主键, 外键连接课程表
实际签到次数 (actual_count)	int	Not null default 0
应当签到次数 (need_count)	int	Not null default 0

6. 课程签到要求表

列名	数据类型	说明
课程代码 (course_id)	varchar	联合主键, 外键连接课程表, 复合索引
日期 (check_date)	date	联合主键, 复合索引
开始时间 (start_time)	time	Not null
结束时间 (end_time)	time	Not null
需要定位 (need_loc)	bool	Default null

7. 学生考勤记录表

列名	数据类型	说明
出勤记录编号 (log_id)	int	自增主键
课程代码 (course_id)	varchar	Not null, 外键连接课程表, 学生课程复合索引
学号 (student_id)	varchar	Not null, 外键连接学生表, 学生课程复合索引
签到时间 (check_time)	datetime	Not null
地址/定位 (loc)	Boolean	Default false
出勤状态 (check_state)	Enum('checked', 'absent', 'leave')	Not null

8. 用户表

列名	数据类型	说明
用户号 (user_id)	varchar	主键
密码 (password)	char	Not null
用户类型 (user_type)	ENUM('student', 'teacher', 'admin')	Not null

3.4.5 数据库索引的建立

由于考勤系统涉及许多查询操作, 所以我们在相关的表上建立了索引来加速操作。

表 2: 考勤系统数据库索引设计

表名	索引名	字段组合	核心作用
student	idx_student_dept	department	加速查询, 如统计各院系学生分布
	idx_student_grade	grade	加速查询, 如统计各年级学生分布
student_course	idx_sc_course	course_id	加速查询, 如查询某课程的选课学生列表
attendance_log	idx_log_student_course	student_id, course_id	加速学生-课程维度的考勤记录查询, 如查看学生某课程的所有考勤情况
	idx_log_composite	course_id, check_time, student_id	支持课程 + 时间 + 学生的组合查询, 如统计某课程某时间段内的学生出勤情况
attendance_requirement	idx_req_course	course_id, check_date	加速课程签到规则查询, 如获取某课程某天的签到时间要求

3.5 前端设计

3.5.1 技术栈与开发工具

微信开发者工具: 微信官方提供的集成开发环境, 支持小程序的开发、调试、预览及发布, 涵盖代码编辑、界面预览、网络请求调试等功能。

基础技术栈:

1.WXML (WeiXin Markup Language): 类似 HTML 的标签语言, 用于构建小程序页面的结构, 通过组件化方式搭建界面元素。

2.WXSS (WeiXin Style Sheets): 类似 CSS 的样式语言, 用于定义页面样式, 支持 rpx 响应式单位, 适配不同尺寸的移动设备屏幕。

3.JavaScript: 实现页面的交互逻辑、数据处理及与后端的接口通信, 通过微信小程序提供的 API 实现扫码、定位等功能。

3.5.2 页面结构设计

1. 登录页面:

功能: 提供账号密码输入框, 支持学生、教师、管理员三类用户登录, 具备表单验证功能 (如账号格式、密码长度校验)。

交互: 输入账号密码后点击登录按钮, 前端调用后端登录接口, 根据返回的用户类型跳转到对应主页。

2. 管理员端页面:

主页: 提供批量导入、导出考勤、查询与修改信息等功能入口, 采用功能模块平铺布局。

信息导入页面: 下拉菜单选择导入类型 (学生 / 教师 / 课程等), 文件上传控件选择 Excel 文件, 显示导入进度与结果提示。

考勤导出页面: 多条件筛选 (教师 / 课程 / 日期), 点击导出按钮生成 Excel 文件并提供下载链接。

修改信息页面: 根据用户输入的 id 查找学生、教师或课程信息, 可进行修改提交或删除相关信息。

批量删除页面: 可选择删除考勤记录、某届学生等。

设置页面: 支持密码修改。

3. 教师端页面:

主页: 显示今日授课课程、发布签到入口、考勤统计入口, 课程卡片包含课程名称、时间、地点及签到状态快捷入口。

发布签到页: 提供课程选择、开始时间设置、签到时长配置、定位需求开关等表单控件, 点击发布后生成对应二维码。

统计页面: 支持按课程、日期查询考勤记录, 提供导出 Excel、登记请假功能, 多维度统计模块 (年级 / 性别 / 院系) 采用下拉菜单选择维度。

个人信息页: 展示教师基本信息 (工号、姓名、院系、职称等), 支持密码修改。

4. 学生端页面:

主页: 显示今日课程列表、扫码签到入口、课表与考勤查询入口, 采用卡片式布局展示课程信息 (课程名称、时间、地点)。

课表页面: 以周为单位的日历式布局, 显示每周各时段课程, 点击课程可查看详情, 支持左右滑动切换周次。

考勤页面: 按课程分类展示考勤记录, 显示应签 / 实签次数、签到状态 (出勤 / 请假 / 缺勤), 点击课程可查看具体签到历史。

个人信息页: 展示学生基本信息 (学号、姓名、院系、专业等), 支持信息查看与密码修改。

3.6 后端设计

3.6.1 架构与技术栈

框架选择: 采用 Flask 微框架搭建后端服务, 具备轻量级、高扩展性特点, 支持通过插件 (如 Flask-SQLAlchemy) 扩展数据库操作、路由管理等功能。

架构模式: 遵循 MVC (模型 - 视图 - 控制器) 架构

1. 模型 (Models): 使用 SQLAlchemy ORM 工具定义数据库表结构与对象映射, 如学生、课程、考勤记录等实体类。

2. 视图 (Views): 通过路由函数处理前端请求, 返回 JSON 格式数据, 不直接操作 HTML 模板 (前端负责页面渲染)。

3. 控制器 (Controllers): 在路由函数中实现业务逻辑控制, 调用模型层进行数据操作, 处理请求参数与返回结果。

4. 数据库: 采用 MySQL 关系型数据库, 存储系统核心数据 (用户信息、课程信息、考勤记录等), 通过 SQLAlchemy 实现 ORM 映射, 简化数据库操作。

3.6.2 核心模块设计

1. 用户认证模块:

登录认证: 接收前端登录请求, 验证账号密码 (密码通过 Bcrypt 哈希算法加密存储), 确认用户身份信息, 返回给前端用于后续请求鉴权。

权限控制: 根据 JWT 中的用户角色 (学生 / 教师 / 管理员), 通过 Flask 钩子函数 (before_request) 拦截请求, 校验用户是否具备操作权限 (如管理员才能执行批量导入功能)。

2. 签到考勤模块:

发布签到: 教师端提交签到参数 (课程、时间、时长、定位需求), 后端生成唯一签到二维码标识, 存储签到规则 (开始时间、结束时间 = 开始时间 + 时长)。

扫码签到: 接收学生扫码请求, 校验签到时间 (是否在有效时间段内)、定位信息 (若启用定位), 更新考勤记录状态 (出勤), 防止重复签到 (同一学生同一课程在有效时间内仅允许签到一次)。

3. 数据统计模块:

考勤查询: 支持按课程、日期、学生等条件查询考勤记录, 返回结构化数据 (如学生出勤列表、缺勤原因统计)。

多维度统计: 根据教师选择的维度 (年级、性别、院系), 对考勤数据进行分组聚合, 计算出勤率、缺勤率等指标, 为后续可视化展示提供数据基础。

4. 数据导入导出模块:

批量导入: 解析 Excel 文件数据 (如学生信息、课程表), 验证数据格式 (如学号唯一性、课程时间有效性), 批量插入或更新数据库记录, 返回导入结果 (成功 / 失败条数及原因)。

导出考勤: 根据筛选条件 (教师、课程、日期) 查询考勤数据, 格式化为 Excel 表格, 包含学生信息、签到时间、状态等字段。

3.6.3 接口设计

接口规范:

1. 协议: 基于 HTTP/HTTPS 协议, 使用 RESTful 风格设计接口, 资源通过 URL 定位 (如/api/students 表示学生资源)。

2. 请求方法: 遵循 REST 规范

POST: 创建和获取资源 (如发布签到、查询信息)。

GET：获取资源（如导出 Excel 文件）。

3. 响应格式：统一返回 JSON 格式。

3.6.4 安全与性能优化

1. 安全措施：

密码加密：用户密码采用 Bcrypt 算法加盐后存储，避免明文泄露。

SQL 注入防护：通过 SQLAlchemy ORM 工具执行数据库操作，自动参数化查询，防止 SQL 注入攻击。

输入检查：严格检查所有输入框的输入格式，防止非法输入。

2. 性能优化：

数据库索引：在高频查询字段（如学号、课程代码、签到时间）创建索引，提升查询效率。

缓存机制：对不常变更的数据（如课程表、教师信息）使用 Redis 缓存，减少数据库查询压力。

异步任务：将耗时操作（如 Excel 导出、批量导入）放入后台任务队列（如 Celery），避免阻塞主接口响应。

4 功能介绍

4.1 管理员端

管理员端主要实现了批量导入信息和导出特定考勤信息的功能。管理员可选批量导入，进入导入信息页面，首先选择导入类型，包含学生信息、教师信息、课程信息、学生课程表和用户信息，之后点击选择 Excel 文件，选择想要导入的表格，点击开始导入，会提示导入成功或失败，失败会提示缺少信息或者其他报错（注意：导入格式应参考模板文件，对于存在依赖的表格需要正确的导入顺序）。管理员可选导出考勤，进入页面后可选择对应的教师、课程和日期，默认为所有，选择好后点击导出考勤数据即可导出表格。管理员可选修改信息，跳转页面，输入课程、教师或学生 id，可查找出对应的相关信息，之后可对相关信息进行修改并提交，也可删除相关信息。管理员点击批量删除，跳转页面，输入 id，可选择删除考勤记录、某届学生等。管理员端也实现了修改密码的功能。

4.2 教师端

教师端主要实现了教师发布签到、查看签到记录、登记请假信息、导出考勤信息、多维度统计考勤信息等功能。教师点击发布签到即可进入发布签到页面，选择好相应的课程、开始时间、签到时间及是否需要定位，点击发布签到即可。教师点击统计即可进入统计页面，选择相应的课程及日期，点击查询即可查询该课程当日的出勤情况；点击导出 excel 即可导出表格文件；点击登记请假即可选择缺勤学生，修改其缺勤状态为请

假；点击统计即可对该课程的所有考勤信息进行多维度考勤统计，可选维度包括年级、性别和院系。教师端也实现了查看教师本人的身份信息和修改密码的功能。

4.3 学生端

学生端主要实现了扫码签到、查看课表、查看学生本人考勤信息等功能。学生点击扫码签到即可选择对应的二维码进行扫描签到，点击课表即可查看本人课表，点击考勤即可查看本人所选课程对应的考勤情况。学生端也实现了查看学生本人的身份信息和修改密码的功能。

5 部署说明

5.1 python 部署

python3:

创建虚拟环境:

```
1 cd /...      #进入目录
2 python -m venv .flask_env # 创建虚拟环境flask_env
```

激活虚拟环境:

在 FD 考勤目录下:

```
1 source flask_env/bin/activate      #Linux系统激活环境
2 flask_env\Scripts\activate         #Windows系统激活环境
```

安装所需库:

```
1 pip install -r requirements.txt
```

python 环境安装完毕。

5.2 构建并配置数据库

构建数据库见 sql 文件 ‘database.sql’，创建数据库 attendance_system, 在 MySQL workbench 直接运行即可。

配置数据库连接 URI 见 ‘backend/init.py’，需修改数据库密码。

```
1 # MySQL数据库连接URI，格式：mysql+连接器://用户名:密码@主机:端口/数据库名,这里需要替换密码和数据库名
2 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:密码@localhost:3306/attendance_system'
```

5.3 运行小程序

运行‘app.py’文件，启动后端程序。安装微信开发者工具，前端在微信开发者工具应用内导入‘frontend’目录，进行编译测试。

5.4 系统使用说明

5.4.1 进入小程序

环境配置好后，执行 5.3 运行小程序的步骤即可启动小程序，输入账号和密码即可进入小程序（账号 100~199 为学生账号，200~209 为教师账号，300 为管理员账号，默认密码统一为 123456）。



图 4: 登录界面

5.4.2 管理员登录与相关功能

输入管理员账号和密码即可登录为管理员。



图 5: 管理员端初始页面

点击批量导入即可进入导入信息页面，首先选择导入类型，包含学生信息、教师信息、课程信息、学生课程表和用户信息，之后点击选择 Excel 文件，选择想要导入的表格，最后点击开始导入，会提示导入成功或失败，失败会提示缺少信息或者其他报错（注意：导入格式应参考模板文件，若导入的信息之间存在依赖需要确保正确的文件导入顺序）。

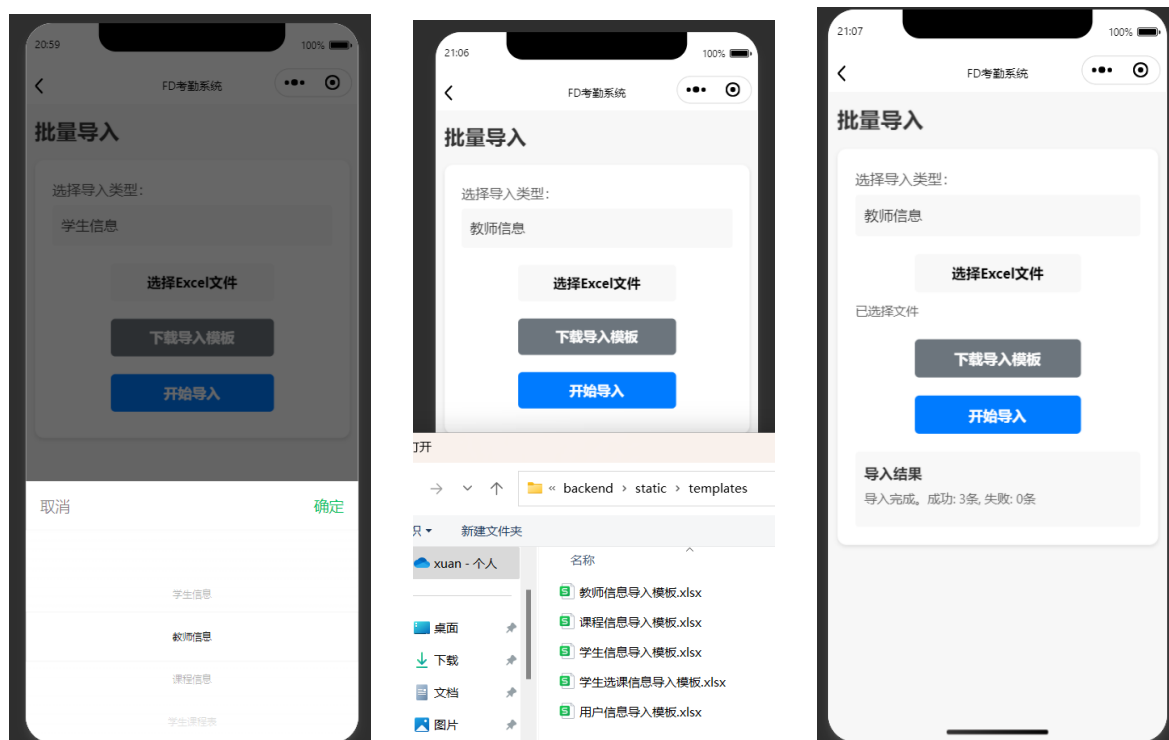


图 6: 管理员端批量导入功能视图

点击导出考勤进入页面可选择对应的教师、课程和日期，默认为所有，选择好后点击导出考勤数据即可导出表格。

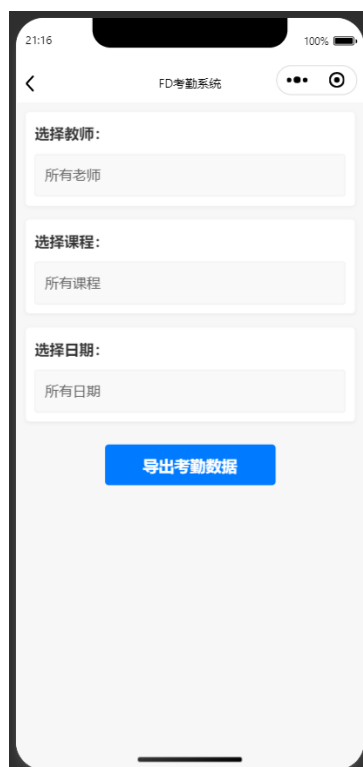


图 7: 管理员端导出考勤

	A	B	C	D	E	F	G	H	I	J
	记录ID	学号	学生姓名	课程编号	课程名称	教师工号	教师姓名	签到时间	签到状态	签到地点
1	1	100	赵同学	1	大数据基础	200	张老师	2023-09-04 10:05:00	出勤	是
2	2	100	赵同学	1	大数据基础	200	张老师	2023-09-11 10:10:00	出勤	是
3	3	100	赵同学	1	大数据基础	200	张老师	2023-09-18 10:03:00	出勤	是
4	4	100	赵同学	1	大数据基础	200	张老师	2023-09-25 10:15:00	出勤	是
5	5	100	赵同学	1	大数据基础	200	张老师	2023-10-09 10:08:00	出勤	是
6	6	100	赵同学	1	大数据基础	200	张老师	2023-10-16 10:30:00	请假	否
7	7	100	赵同学	1	大数据基础	200	张老师	2023-10-23 10:07:00	出勤	是
8	8	100	赵同学	1	大数据基础	200	张老师	2023-10-30 10:12:00	出勤	是
9	9	100	赵同学	1	大数据基础	200	张老师	2023-11-06 10:05:00	出勤	是
10	10	100	赵同学	1	大数据基础	200	张老师	2023-11-13 10:00:00	出勤	是
11	11	100	赵同学	1	大数据基础	200	张老师	2023-11-20 10:20:00	请假	是
12	12	100	赵同学	1	大数据基础	200	张老师	2023-11-27 10:09:00	出勤	是
13	13	100	赵同学	1	大数据基础	200	张老师	2023-12-04 10:11:00	出勤	是
14	14	100	赵同学	1	大数据基础	200	张老师	2023-12-11 10:06:00	出勤	是
15	15	100	赵同学	1	大数据基础	200	张老师	2023-12-18 10:04:00	出勤	是
16	16	100	赵同学	1	大数据基础	200	张老师	2023-12-25 10:00:00	出勤	是
17	17	101	钱同学	1	大数据基础	200	张老师	2023-09-04 10:02:00	出勤	是
18	18	101	钱同学	1	大数据基础	200	张老师	2023-09-11 10:08:00	出勤	是
19	19	101	钱同学	1	大数据基础	200	张老师	2023-09-18 10:05:00	出勤	是
20	20	101	钱同学	1	大数据基础	200	张老师	2023-09-25 10:12:00	出勤	是
21	21	101	钱同学	1	大数据基础	200	张老师	2023-10-09 10:10:00	出勤	是
22	22	101	钱同学	1	大数据基础	200	张老师	2023-10-16 10:00:00	出勤	是
23	23	101	钱同学	1	大数据基础	200	张老师	2023-10-23 10:15:00	出勤	是
24	24	101	钱同学	1	大数据基础	200	张老师	2023-10-30 10:07:00	出勤	是
25	25	101	钱同学	1	大数据基础	200	张老师	2023-11-06 10:03:00	出勤	是
26	26	101	钱同学	1	大数据基础	200	张老师	2023-11-13 10:09:00	出勤	是
27	27	101	钱同学	1	大数据基础	200	张老师	2023-11-20 10:00:00	出勤	是
28	28	101	钱同学	1	大数据基础	200	张老师	2023-11-27 10:11:00	出勤	是
29	29	101	钱同学	1	大数据基础	200	张老师	2023-12-04 10:06:00	出勤	是
30	30	101	钱同学	1	大数据基础	200	张老师	2023-12-11 10:04:00	出勤	是
31	31	101	钱同学	1	大数据基础	200	张老师	2023-12-18 10:00:00	出勤	是

图 8: 管理员端导出表格示例

管理员点击修改信息，跳转页面，输入课程、教师或学生 id，可查找出对应的相关信息，之后可对相关信息进行修改并提交，也可删除相关信息。

11:09 76%

FD 考勤系统

100 搜索

当前信息:

学生姓名: 赵同学

性别: 男

学院: 大数据学院

专业: 大数据

年级: 2022

生日: 2003-01-01

提交修改

删除相关数据

图 9: 管理员修改信息页面

管理员点击批量删除，跳转页面，输入 id，可选择删除考勤记录、某届学生等。

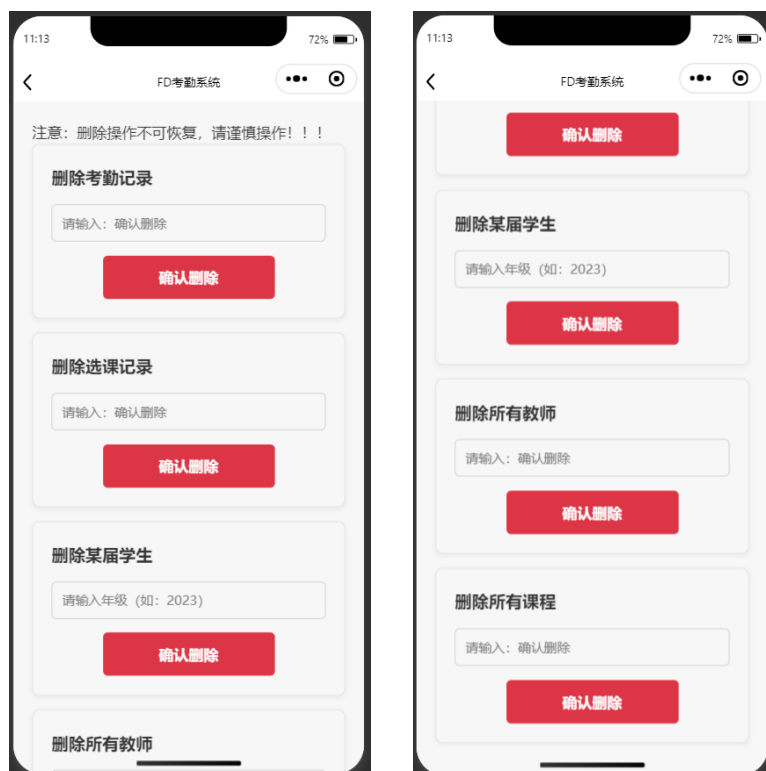


图 10: 管理员端批量删除功能视图

管理员点击设置即可修改密码和退出登录。



图 11: 管理员端设置页面

5.4.3 教师登录与相关功能

输入教师账号和密码即可登录为教师。



图 12: 教师端初始页面

教师点击发布签到即可进入发布签到页面，选择好相应的课程、开始时间、签到时间及是否需要定位，点击发布签到即可。



图 13: 教师端发布签到页面

教师点击统计即可进入统计页面，选择相应的课程及日期，点击查询即可查询该课程当日的出勤情况；点击导出 excel 即可导出表格文件；点击登记请假即可选择缺勤学生，修改其缺勤状态为请假；点击统计即可对该课程的所有考勤信息进行多维度考勤统计，可选维度包括年级、性别和院系。



图 14: 教师端页面

自动保存

关

sUh4htUpjJA5c9a33344887bac4ebe25f22e4814e

文件

开始

OfficePLUS

插入

绘图

页面布局

公式

数据

审阅

视图

PDF

A1

:

fx

学号

	A	B	C	D	E	F
1	学号	姓名	状态	签到时间		
2	100	赵同学	出勤	2023-12-18 10:04:00		
3	101	钱同学	出勤	2023-12-18 10:08:00		
4	102	孙同学	出勤	2023-12-18 10:12:00		
5						
6						
7						

图 15: 教师端导出考勤数据示例



图 16: 教师端登记请假功能视图



图 17: 教师端多维度统计功能视图

教师点击信息即可查看本人的身份信息。



图 18: 教师端信息页面

教师点击设置即可修改密码和退出登录。

5.4.4 学生登录与相关功能

输入学生账号和密码即可登录为学生。



图 19: 学生端初始页面

点击扫码签到即可选择对应的二维码进行扫描签到 (教师发布的二维码会默认保存在 'backend/static/qr_codes' 文件夹下)。

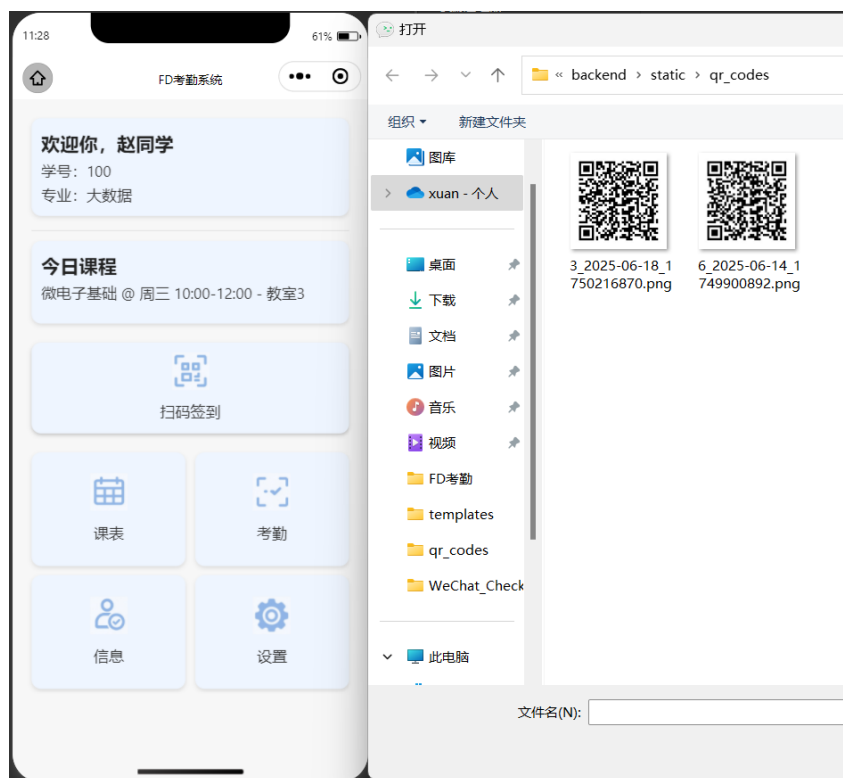


图 20: 学生端扫码签到

学生点击课表即可查看本人课表，点击某课程即可查看该课程相关信息。



图 21: 学生端课表页面

学生点击考勤即可查看本人所选课程对应的考勤情况，点击某课程即可查看该课程相关的本人的考勤记录。



图 22: 学生端考勤页面

学生点击信息即可查看本人的身份信息。



图 23: 学生端信息页面.

学生点击设置即可修改密码和退出登录。

6 未来展望

6.1 功能扩展

1. 智能考勤分析：引入数据分析算法，基于学生考勤记录（出勤频率、缺勤模式）生成个性化学习预警，如频繁缺勤的学生自动推送提醒通知。
2. 可视化报表：为教师和管理员提供图表化考勤统计（如出勤率趋势图、各院系缺勤率对比柱状图），支持导出 PDF 报表，提升数据可读性。
3. 移动推送功能：集成微信服务号模板消息，实现签到提醒（课前 10 分钟推送签到通知）、缺勤预警（未签到时推送提醒），提高签到率。
4. 跨平台支持：扩展为 H5 网页版和 APP 版，满足非微信环境下的考勤需求，如企业培训、会议签到等场景。

6.2 技术升级

1. 引入 AI 技术：引入人脸识别技术作为辅助签到方式，解决代扫码签到问题；利用 NLP 技术分析学生请假理由，自动归类缺勤原因（如病假、事假）。
2. 分布式数据库：当数据量增大时，采用 MySQL 主从复制或分库分表策略，提升数据库读写性能，保障高并发场景下的系统稳定性。
3. 云服务部署：将系统迁移至阿里云、腾讯云等云平台，利用云服务器的弹性扩展能力应对访问高峰，同时提供自动备份与容灾方案。

7 代码下载地址及代码说明

项目代码：[GitHub Repo](#)

代码说明：[README.md](#)