



微信点名管理系统 项目推进情况汇报

小组成员：洪家权 盖烈森 马静 陈皓阳

时间：2025年4月16日



目 录

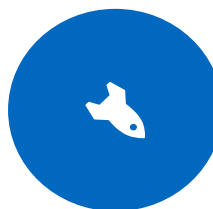
C o n t e n t s



PART 01

需求分析

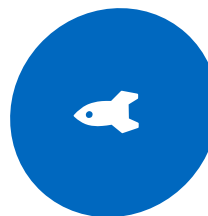
系统设计目标、核心功能板块



PART 02

概念模型

关系模式及其范式、关系种类、完整性约束



PART 03

模块与功能

系统功能模块与技术实现思路

01

PART 01

需求分析

系统设计目标

本系统利用微信平台的用户基础，及其扫码接口作为前端签到入口；核心数据，包括学生、课程、教师、考勤及请假记录等，将由后端应用程序依赖健壮的关系数据库进行高效、安全的存储与管理；同时，系统提供 Web 前端界面，支持各类用户（学生、教师、管理员）进行信息查询、管理操作（增删改）等交互，并能直观地展示所需的考勤统计结果。通过整合这些技术，期望能显著提升考勤效率，减少人为错误，并为教学管理提供有效的数据支持。

点名效率低？

实时化

扫码签到快速完成
出勤数据自动同步

数据难追溯？

无纸化

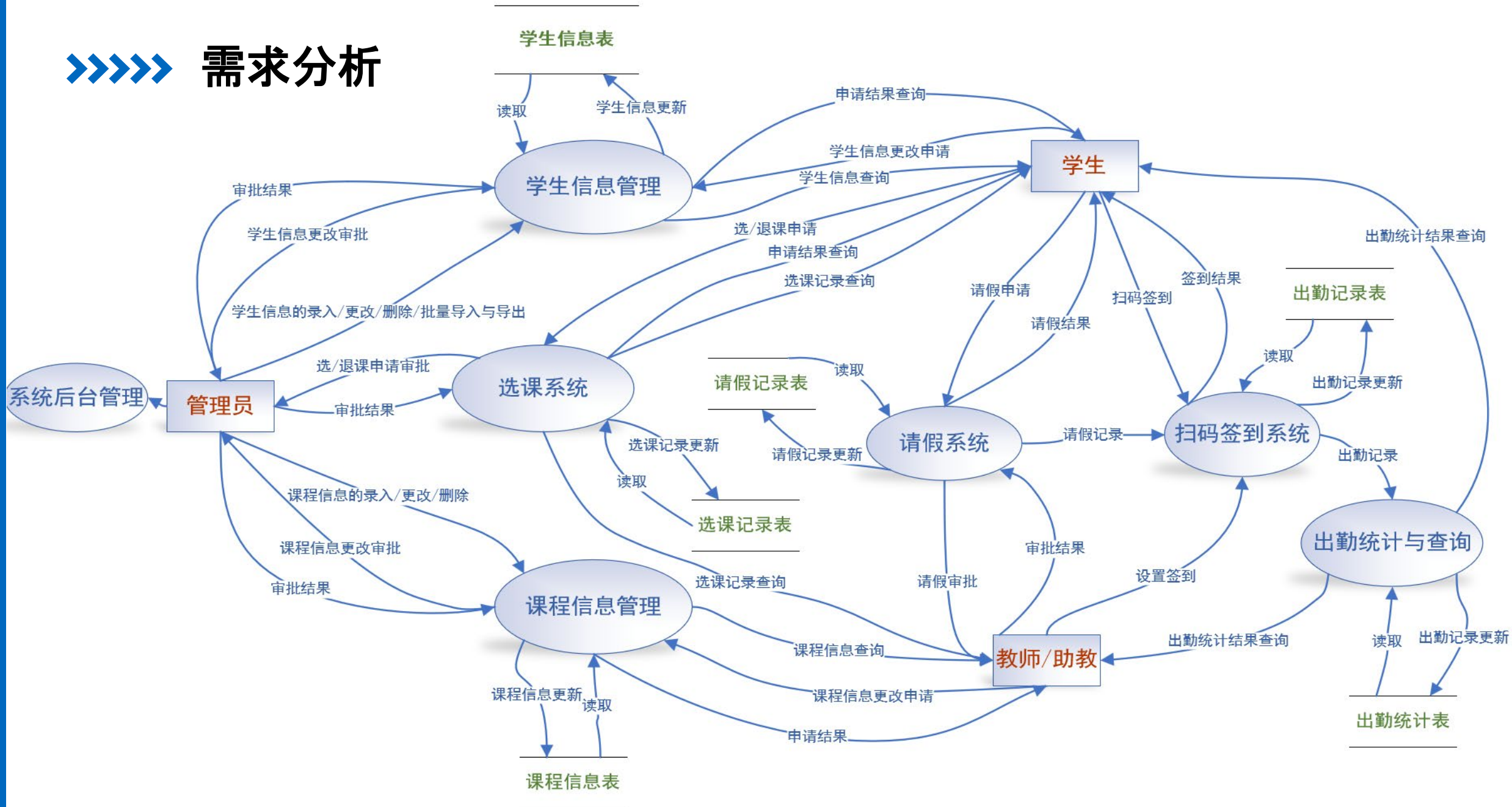
请假、选课线上审批
历史记录可追溯（增删改）

统计数据难？

可视化

多维统计报表一键生成
辅助教学决策

需求分析



核心功能模块

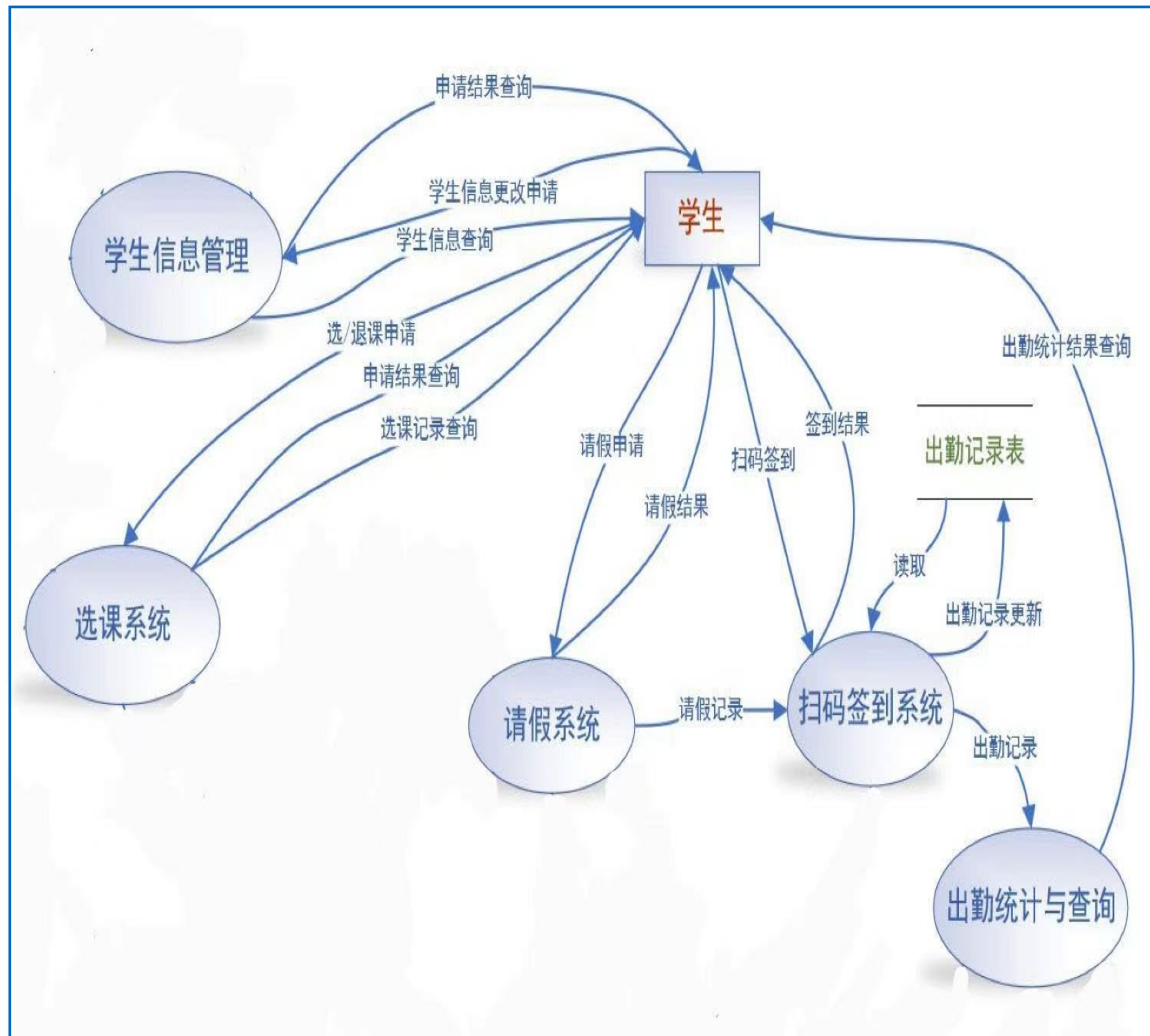
学生端核心场景

微信签到 在教室现场进行单次课程的微信扫码点名签到

请假-选退课闭环 web端发起请假/选退课申请
→教师审批→结果实时查询

课程安排 微web平台实时查询课程安排变动
(与教师课程安排变更/电子请假/选退课联动)

学期统计 web平台实时查询某门课程至今考勤情况统计



核心功能模块

教师端管理中枢

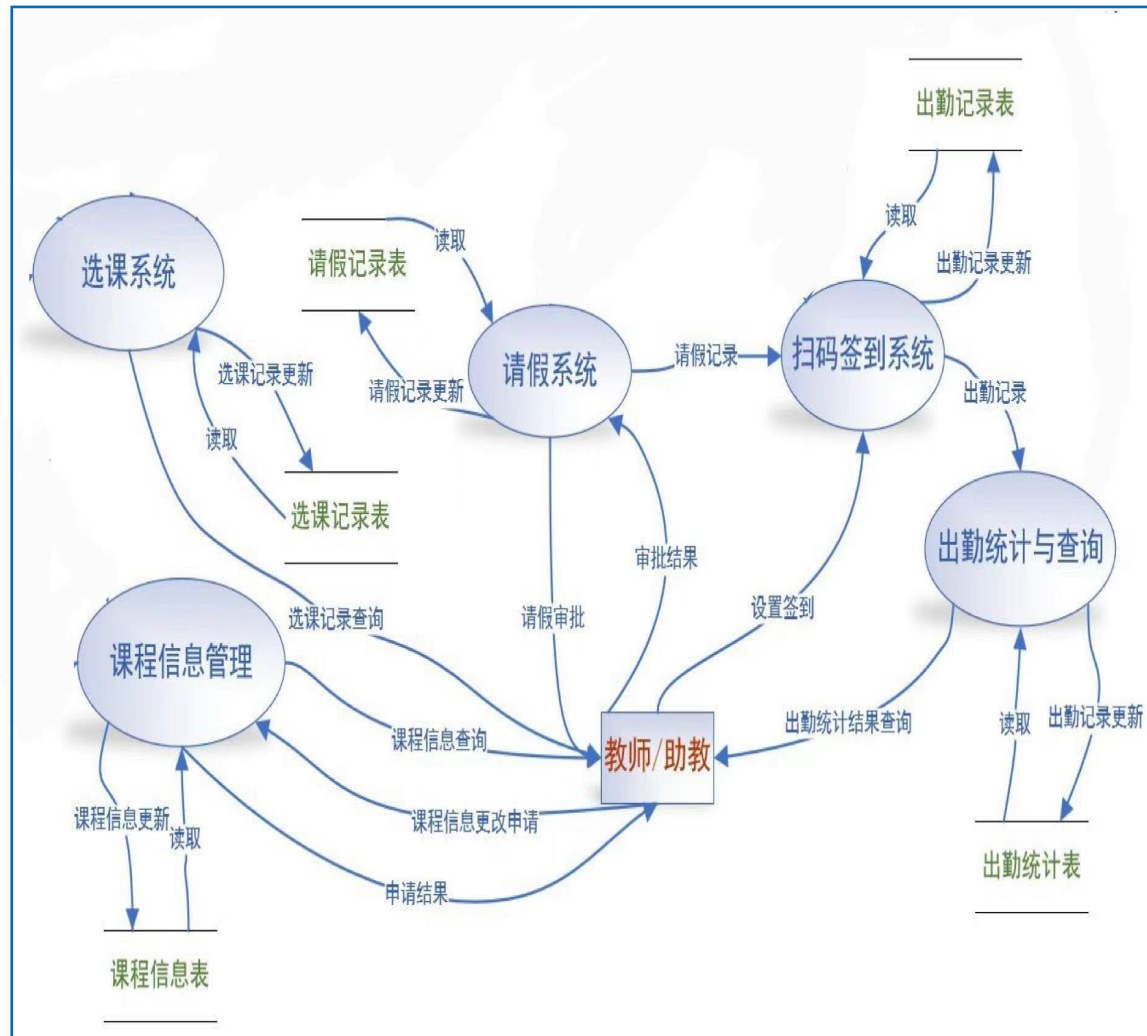
签到控制 教师为指定课程课次发起考勤事件，设定扫码时段状态，并监控学生签到情况

请假审批 批准或驳回，可添加审批备注

课程安排调整 向管理员申请/自行在系统中调整课程安排，包括课程时间地点等

学生信息查询 查询课程参与列表的学生信息及考勤统计，关注每一位学生的课程参与情况

#数据看板 可按课程、时间范围、学生等维度筛选查看考勤信息统计图表，方便管理



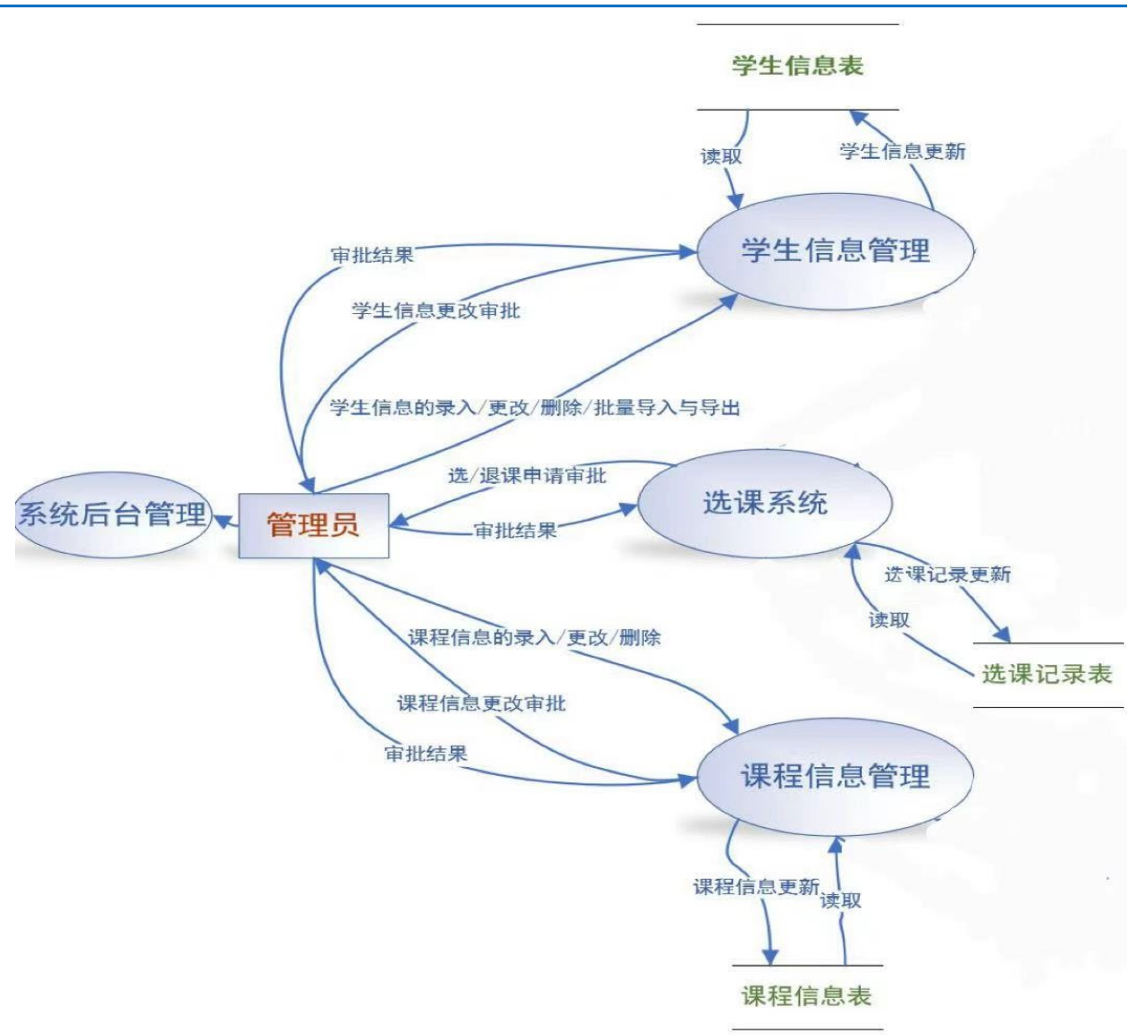
核心功能模块

管理员后台引擎

系统配置 维护（增/删/改/查）院系、专业、课程、教师、学生、管理员等核心基础信息表；支持基本信息的批量导入

教学流程管理 审批包括学生选退课申请，审批教师教学安排变更申请等，并据此修改参与课程人员列表与教学安排时间表

数据监控与分析 查询并再必要时修正考勤记录或请假记录；查看全院、特定课程或教师的考勤统计报表；确保关键数据的一致性

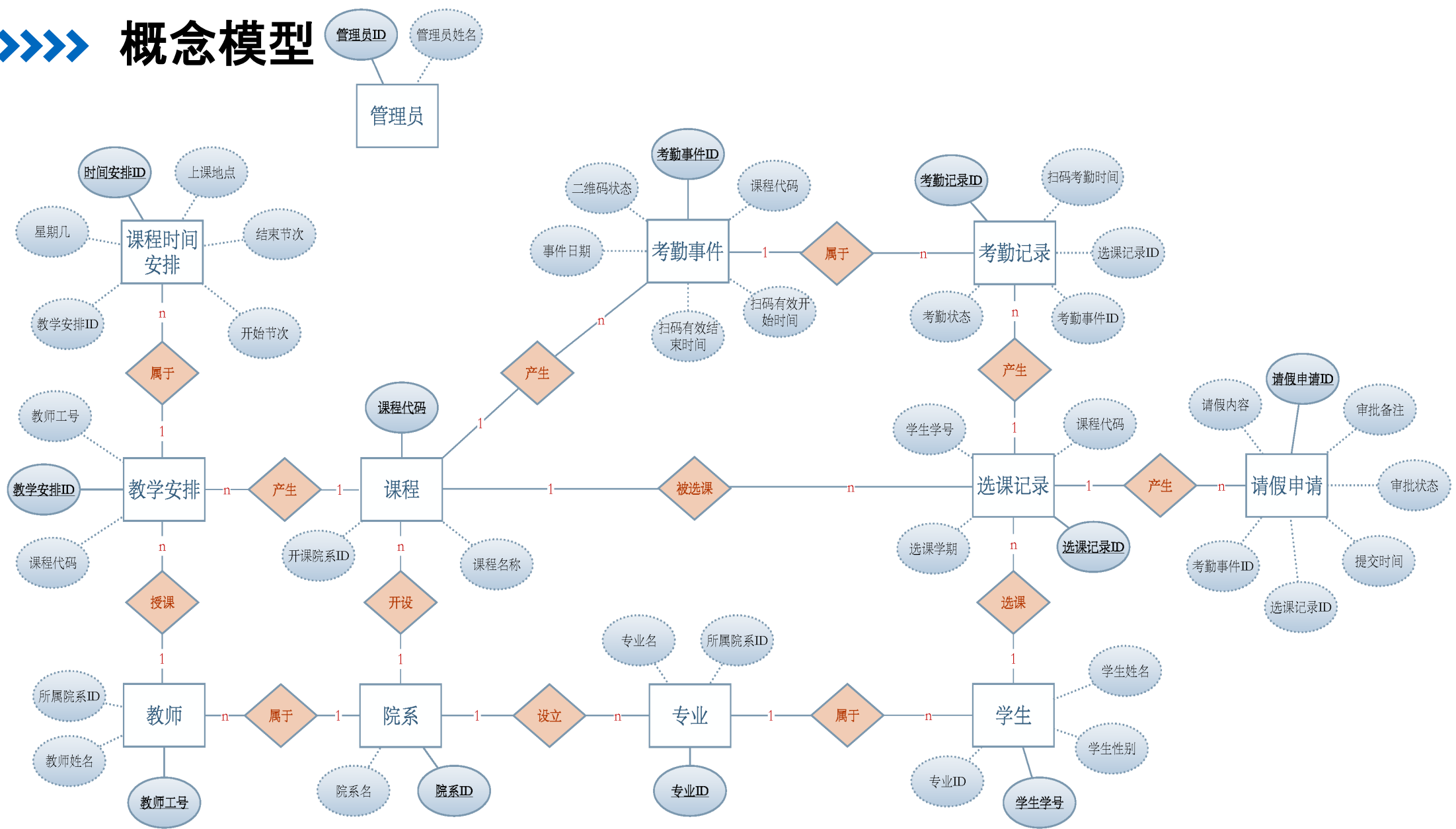


02

PART 02

概念模型

概念模型



概念模型 共包含12个核心实体及相关联系

基础信息层

涵盖院系、专业、学生、教师、课程及管理员的基本数据

教学计划层

通过教学安排（连接教师与课程）和课程时间安排（定义具体时间地点）来明确授课计划

核心业务层

- * 以选课记录（连接学生、课程、学期）为关键，衍生学生应有的考勤记录（记载出勤/缺勤/请假状态）和请假申请（记录请假及审批流程）
- * 由特定课程出发，创建每次课的课堂考勤事件，学生对考勤事件的响应则记录在中，与相应的学生考勤记录和学生请假申请相关联

关系模式

专业

专业（专业ID，专业名，所属院系ID（FK））

院系

院系（院系ID，院系名）

学生

学生（学生学号，学生姓名，学生性别，学生专业ID（FK））

教师

教师（教师工号，教师姓名，所属院系ID（FK））

关系模式

管理员

管理员（管理员ID，管理员姓名）

课程

课程（课程代码，课程名称，开课院系ID（FK））

教学安排

教学安排（教学安排ID，教师工号（FK），课程代码（FK））

课程时间安排

课程时间安排（时间安排ID，教学安排ID（FK），星期几，开始节次，结束节次，上课地点）

关系模式

选课记录

选课记录（选课记录ID，学生学号（FK），课程代码（FK），选课学期）

考勤事件

考勤事件（考勤事件ID，课程代码（FK），扫码有效开始时间，扫码有效结束时间，事件日期，二维码状态）

考勤记录

考勤记录（考勤记录ID，选课记录ID（FK），考勤事件ID（FK），扫码考勤时间，考勤状态）

请假申请

请假申请（请假申请ID，选课记录ID（FK），考勤事件ID（FK），请假内容，提交时间，审批状态，审批备注）

非主属性对候选键无传递依赖

满足3NF

关系种类

课程 (1) ^{被选课} <--> 选课记录 (n)

院系 (1) ^{开设} <--> 课程 (n)

教师 (n) ^{属于} <--> 院系 (1)

院系 (1) ^{设立} <--> 专业 (n)

学生 (n) ^{属于} <--> 专业 (1)

关系种类

考勤记录 (n) $\xrightarrow{\text{属于}}$ 考勤事件 (1)

选课记录 (1) $\xrightarrow{\text{产生}}$ 考勤记录 (n)

选课记录 (1) $\xrightarrow{\text{产生}}$ 请假申请 (n)

学生 (1) $\xrightarrow{\text{选课}}$ 选课记录 (n)

关系种类

课程时间安排 (n) $\xrightarrow{\text{属于}}$ 教学安排 (1)

教师 (1) $\xrightarrow{\text{授课}}$ 教学安排 (n)

课程 (1) $\xrightarrow{\text{产生}}$ 教学安排 (n)

课程 (1) $\xrightarrow{\text{产生}}$ 考勤事件 (n)

完整约束

实体完整性

主键必须唯一且非空

根据实体特性，混合使用了自增整数作为代理主键和具有业务含义的自然主键

参照完整性

外键必须引用有效的主键值

不同的删除和更新规则

ON DELETE RESTRICT: 应用于基础数据表的主键被引用时

ON DELETE CASCADE: 应用于关联/从属关系

ON DELETE SET NULL: 应用于允许关联丢失但记录本身仍有意义的情况

ON UPDATE CASCADE: 普遍应用于所有外键

用户定义的约束

考勤事件的时间有效性

CHECK: 开始时间早于结束时间

课程时间合理性

CHECK: 课程开始节次早于结束节次

枚举类型限制

取值范围为【待审批、已通过、已拒绝】

仅允许“男”或“女”

非空约束与默认值

根据业务逻辑设置必须存在的字段

设置默认值，提供初始状态

实体名	实体描述	属性名	属性描述	数据类型	主键	外键	非空	唯一	自增	外键参照	约束补充说明
Department	院系信息	dept_id	院系ID	TINYINT UNSIGNED	是	否	是	是	是		
		dept_name	院系名称	VARCHAR(50)	否	否	是	是	否		
Major	专业信息	major_id	专业ID	TINYINT UNSIGNED	是	否	是	是	是		
		major_name	专业名称	TINYINT UNSIGNED	否	否	是	是	否		
		dept_id	院系ID	TINYINT UNSIGNED	否	是	是	是	否	Department (dept_id), 限制删除 (若有专业属于该院系则不能删), 更新时级联	
Student	学生信息	stu_id	学生学号	CHAR(11)	是	否	是	是	否		
		stu_name	学生姓名	VARCHAR(50)	否	否	是	否	否		
		stu_sex	学生性别	ENUM('男', '女')	否	否	是	是	否		
		major_id	专业ID	TINYINT UNSIGNED	否	是	否	否	否	Major (major_id), 当专业删除时学生专业设为NULL, 更新时级联	
Course	课程信息	course_id	课程代码	CHAR(12)	是	否	是	是	否		
		course_name	课程名称	ARCHAR(50)	否	否	是	否	否		
		dept_id	院系ID	TINYINT UNSIGNED	否	是	是	否	否	Department (dept_id), 限制删除 (若课程属于该院系则不能删), 更新时级联	
Teacher	教师信息	teacher_id	教师工号	CHAR(5)	是	否	是	是	否		
		teacher_nam	教师姓名	VARCHAR(50)	否	否	是	是	否		
		dept_id	院系ID	TINYINT UNSIGNED	否	是	是	是	否	Department (dept_id), 限制删除 (若教师属于该院系则不能删), 更新时级联	



AttendanceEvent	课堂考勤事件	event_id	考勤事件ID	INT	是	否	是	是	是		
		course_id	关联的课程代码	CHAR(12)	否	是	是	否	否	Course(course_id) 课程删除则考勤事件删除	
		event_date	本次考勤事件发生的日期	DATE	否	否	是	否	否		
		scan_start_time	允许扫码的最早时间(时:分:秒)	TIME	否	否	是	否	否		
		scan_end_time	允许扫码的最早时间(时:分:秒)	TIME	否	否	是	否	否		
		event_statuses	二维码/事件状态	ENUM('有效','无效') DEFAULT '有效'	否	否	是	否	否		
Admin	管理员	admin_id	管理员工号	CHAR(5)	是	否	是	是	否		
		admin_name	管理员姓名	VARCHAR(50)	否	否	是	是	否		
TeachingAssignment	教学安排	assign_id	教学安排ID	INT	是	否	是	是	是		
		teacher_id	教师工号	CHAR(5)	否	是	是	是	否	Teacher(teacher_id)	确保同一教师与同一课程的安排不重复(同一课程的不同时间体现在Schedule中)
		course_id	课程代码	CHAR(12)	否	是	是		否	Course(course_id)	
ClassSchedule	课程时间安排	schedule_id	时间安排ID	INT	是	否	是	是	是		
		assign_id	关联的教学安排ID	INT UNSIGNED	否	是	是	否	否	TeachingAssignment(assign_id) 当教学安排删除时, 对应的时间安排也删除	
		day_of_week	星期几(例如, 1 代表周一, 2 代表周二)	TINYINT UNSIGNED	否	否	是	否	否		确保星期范围正确(1, 7)
		start_period	开始节次(例如, 1 代表第1节)	TINYINT UNSIGNED	否	否	是	否	否		确保课程节次在范围内(假设一天最多13节课)
		end_period	结束节次(例如, 2 代表到第2节结束)	TINYINT UNSIGNED	否	否	是	否	否		确保结束节次不小于开始节次且在范围内
		location	上课地点	VARCHAR(50)	否	否	否	否	否		



实体 AttendanceEvent (课堂考勤事件)

event_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,

考勤事件ID, 无符号整数, 自动增长, 主键

course_id CHAR(12) NOT NULL,

关联的课程代码, 外键, 关联 Course 表

event_date DATE NOT NULL,

本次考勤事件发生的日期, 日期格式, 非空

scan_start_time TIME NOT NULL,

允许扫码的最早时间 (时:分:秒) , 时间格式, 非空

scan_end_time TIME NOT NULL,

允许扫码的最晚时间 (时:分:秒) , 时间格式, 非空

event_status ENUM('有效', '无效') NOT NULL DEFAULT '有效',

二维码/事件状态, 枚举类型, 非空, 默认为 '有效'

FOREIGN KEY (course_id) REFERENCES Course(course_id) ON DELETE CASCADE
ON UPDATE CASCADE

外键约束: 关联课程代码, 课程删除则考勤事件删除

Enrollment	选课记录	enroll_id	选课记录ID	INT UNSIGNED	是	否	是	是	是		
		stu_id	学生学号	CHAR(11)	否	是	是	是	否	Student(stu_id), 学生删除则选课记录删除, 更新时级联	确保同一学生同一学期不能重复选同一门课
		course_id	课程代码	CHAR(12)	否	是	是		否	Course(course_id), 课程删除则选课记录删除, 更新时级联	
		semester	选课学期	CHAR(6)	否	否	是		否		
Attendance	考勤记录	attend_id	考勤记录ID	INT UNSIGNED	是	否	是	是	是		
		enroll_id	选课记录ID	INT UNSIGNED	否	是	是	是	否	Enrollment(enroll_id), 选课记录删除则考勤记录删除, 更新时级联	确保一条选课记录在同一次考勤事件中只有一条考勤记录
		event_id	考勤事件ID	INT UNSIGNED	否	是	是		否	AttendanceEvent(event_id), 考勤事件删除则考勤记录删除, 更新时级联	
		scan_time	扫码考勤时间	DATETIME	否	否	否	否	否		
		status	考勤状态	ENUM('出勤', '缺勤', '请假')	否	否	是	是	否		
		notes	备注	TEXT	否	否	否	否	否		
LeaveRequest	请假申请	leave_request_id	请假申请ID	INT UNSIGNED	是	否	是	是	是		
		enroll_id	申请请假的选课记录ID	INT UNSIGNED	否	是	是	是	否	enroll_id(enroll_id), 选课记录删除则请假申请删除	确保学生对同一次考勤事件只能提交一份请假申请
		event_id	申请请假的考勤事件ID	INT UNSIGNED	否	是	是		否	AttendanceEvent(event_id), 考勤事件删除则请假申请删除	
		reason	请假内容(原因)	TEXT	否	否	是	否	否		
		submit_time	提交时间	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	否	否		否	否		
		approval_status	审批状态	ENUM('待审批', '已批准', '已驳回') NULL DEFAULT '待审批'	否	否	是	否	否		
		approver_notes	审批备注	TEXT	否	否	否	否	否		
		approver_teacher_id	审批教师工号	CHAR(5)	否	是	否	否	否	Teacher(teacher_id), 教师信息删除则审核教师工号设为NULL, 级联更新	

实体 Enrollment (选课记录)

选课记录ID, 无符号整数, 自动增长, 主键
学生学号, 外键, 关联 Student 表, 非空
课程代码, 外键, 关联 Course 表, 非空
选课学期, 字符串, 定长6字符 ('2024-1'), 非空
外键约束: 关联学生学号, 学生删除则选课记录删除, 更新时级联
外键约束: 关联课程代码, 课程删除则选课记录删除, 更新时级联
唯一约束: 确保同一学生同一学期不能重复选同一门课

实体 Attendance (考勤记录)

考勤记录ID, 无符号整数, 自动增长, 主键
对应的选课记录ID, 外键, 关联 Enrollment 表, 非空
对应的考勤事件ID, 外键, 关联 AttendanceEvent 表, 非空
扫码考勤时间, 年月日时分秒, 允许为空
考勤状态, 枚举类型, 非空
备注, 文本类型, 可为空
外键约束: 关联选课记录ID, 选课记录删除则考勤记录删除, 更新时级联
外键约束: 关联考勤事件ID, 考勤事件删除则考勤记录删除, 更新时级联
唯一约束: 确保一条选课记录在同一次考勤事件中只有一条考勤记录

实体 LeaveRequest (请假申请)

请假申请ID, 无符号整数, 自动增长, 主键
申请请假的选课记录ID, 外键, 关联 Enrollment 表, 非空
申请请假的考勤事件ID, 外键, 关联 AttendanceEvent 表, 非空
请假内容 (原因), 长文本类型, 非空
提交时间, 时间戳, 默认为当前时间
审批状态, 枚举类型, 不允许为空, 默认为 '待审批'
审批备注, 长文本类型, 允许为空
审批教师工号, 外键, 关联 Teacher 表, 允许为空
外键约束: 关联选课记录ID, 选课记录删除则请假申请删除
外键约束: 关联考勤事件ID, 考勤事件删除则请假申请删除
外键约束: 关联教师工号, 教师信息删除则审核教师工号设为 NULL, 级联更新
唯一约束: 确保学生对同一次考勤事件只能提交一份请假申请

03

PART 03

模型与功能

模块划分

概述

本项目预计采用 Python Django 全栈框架开发，整合后端逻辑与前端页面渲染

数据库选用 MySQL (InnoDB引擎, utf8mb4字符集)，通过 Django ORM 进行数据交互

前端利用 Django Templates 生成 HTML 页面

嵌入 JavaScript 调用微信 JS-SDK 实现核心的扫码功能

系统为学生、教师、管理员三类角色提供服务

Django ORM (对象关系映射)

Django 内建的强大工具，让我们能用 Python 代码（操作类和对象）来间接、安全地操作数据库表

微信 JS-SDK (JavaScript 开发工具包)

微信官方提供的、用于网页开发的 JavaScript 工具库。它允许在微信内置浏览器中运行的网页调用微信 App 的原生功能，例如**摄像头扫码** (wx.scanQRCode)、获取用户信息、微信支付等。

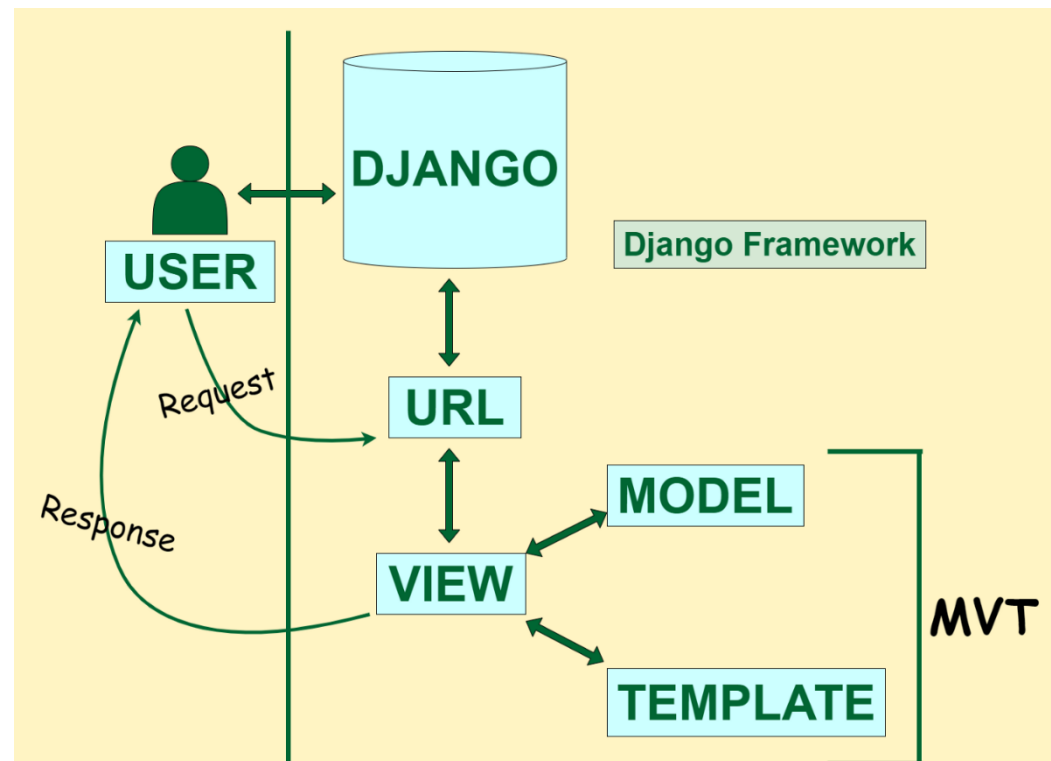
本项目核心的扫码签到功能预计通过前端网页嵌入 JS-SDK，调用 wx.scanQRCode 接口来实现

模块划分

可能的代码文件结构框架

WeChat_QR_Scanning_Attendance_System/

```
├── attendance_system/
│   ├── settings.py # 核心配置 (数据库, App 注册等)
│   ├── urls.py # 主 URL 路由
│   └── ...
├── apps/
│   ├── users/ # 用户与基础信息 App
│   │   ├── models.py # Department, Major, Student, Teacher, Admin 模型
│   │   ├── views.py # 用户/组织信息管理视图, 微信登录与绑定视图
│   │   └── ...
│   ├── courses/ # 课程与教学计划 App
│   │   ├── models.py # Course, TeachingAssignment, ClassSchedule 模型
│   │   ├── views.py # 课表查询视图, 后台课程管理视图
│   │   └── ...
│   ├── attendance/ # 考勤与请假核心 App
│   │   ├── models.py # Enrollment, AttendanceEvent, Attendance, LeaveRequest 模型
│   │   ├── views.py # 考勤事件/签到/请假/审批/查询/统计等核心视图
│   │   └── services.py # 复杂业务逻辑 (微信交互, 状态更新, 统计计算)
│   └── ...
├── templates/ # 全局 HTML 模板目录
│   └── base.html
├── static/ # 全局 CSS, JS 目录
│   ├── attendance/
│   ├── checkin.js # 调用微信扫码及提交签到请求的 JS
│   └── qr_generator.js # 教师端生成二维码的 JS
├── Document/ # 文档目录
├── manage.py # Django 管理脚本
└── requirements.txt # Python 依赖列表
```



模块划分

学生界面

聚焦于个人课表查看、参与扫码签到、查询个人考勤记录、提交和查询请假申请等功能

教师界面

聚焦于管理所授课程、查看学生名单、发起和管理考勤事件、查看考勤结果、审批请假申请等功能

管理员界面

聚焦于后台管理，提供对系统所有基础数据的增删改查功能，以及查看全局考勤统计报表等

e.g. 学生核心流程：微信绑定与扫码签到

阶段一：首次登录与微信绑定(一次性)

访问入口 (微信内网页) → 触发登录 → 微信授权 → 用户确认 → 获取授权码 (Code) → 后端换取 OpenID → 验证/绑定学号 (stu_id) → 建立会话 (Session) → 登录完成

阶段二：日常扫码签到(每次)

教师准备 (创建事件/展示含 event_id 的 QR) → 学生访问页面 (已登录) → 点击扫码按钮 → 前端调用 wx.scanQRCode → 微信扫码返回 event_id → 前端自动提交 (event_id + Session) → 后端验证 (身份/事件/时间/选课/是否重复) → 数据库记录考勤 → 前端显示结果

二维码 代表考勤事件 id / 是一个指向系统的 URL，且包含了考勤事件 ID



THANKS

