学校人员管理详细设计

第七组

\*已上传至github：https://github.com/Achenganggyel/School-People-Manage/upload/main

|  |  |  |  |
| --- | --- | --- | --- |
| **变更历史** | | | |
| 版本 | 内容 | 修改人 | 日期 |
| 0.0.5 | 报告初步 | 赖俊涛 | 2022-12-13 |
| 0.0.8 | 接口，页面设计，数据库补充 | 赵阳 | 2022-12-15 |
| 1.0.0 | 完善报告 | 赖俊涛 | 2022-12-20 |

# 项目结构总结

SCHOOLPEOPLE\_MANAGE/

|-- screenshots 界面预览

|-- src

|    |-- api 前端调用本地文件和后端数据的接口

|    |-- assets 资产，包括图片和数据

|    |-- calc 数据相关的计算函数

|    |-- components 工作台组件的设计

|    |-- router 管理系统的各页面路由

|    |-- server node.js和后端数据库

|    |-- store 权限设置，动态菜单，tags标签页相关方法

|    |-- utils 公共工具，包括axios

|    |-- views 各前端页面的具体设计

|    |-- App.vue 全局css样式设置

|    |-- main.ts 项目配置

|    |-- vite-env.d.ts

|-- 文字资料 开发中记录的文件资料，包括数据库定义

|-- Readme.md 项目概述

|-- vite.config.ts

|-- index.html  web网页配置，包含标签名称

# 需求描述

|  |  |
| --- | --- |
| **需求名称** | **需求具体含义** |
| **用户登录** | 设计登录逻辑，设置权限，若用户无账号则登录失败 |
| **管理人员** | 查看信息，修改状态，统计出勤率 |
| **数据统计** | 查看各类人员及出勤率等的统计图 |
| **表单上传** | 图片上传板块，用户上传监控图片，主要应用于监控设备传送至系统的图片缺失的场景 |
| **用户信息** | 登录用户的基本信息，包括ID、简介等，包括密码修改 |
| **消息通知** | 查看各部门发布的消息 |
| **权限限制** | 将用户权限分为普通用户和管理员，普通用户拥有查看数据、上传表单等权限，管理员则增加限制普通用户权限、发布消息的特殊权限 |

# 功能拆分

## 登录功能

1. 一个登录页面制作，界面
2. 登录功能，一个登录接口
3. 单点登录需要分布式缓存，集成redis或本地缓存
4. 登录校验

## 工作台首页

1. 一个工作台首页制作，界面
2. 菜单功能，路由跳转
3. tags保存打开页面信息
4. header, sidebar, main经典三段式布局
5. 统一的主题

## 人员管理子页面

1. 一个人员管理页面制作，界面
2. 功能：查看学生，教师等人员信息
3. 连接数据库，即使查看和更新

## 消息通知

1. 界面：消息通知查看界面
2. 功能：接收系统的相关消息
3. 连接后端代码和数据库，由管理员发送通知，或者接收异常报错信息

## 数据统计

1. 界面：数据统计页面，有扇面图，柱状图，折线图
2. 功能：直观观察人员管理等信息
3. 连接数据库，使用数据可视化绘制统计图

## 权限管理

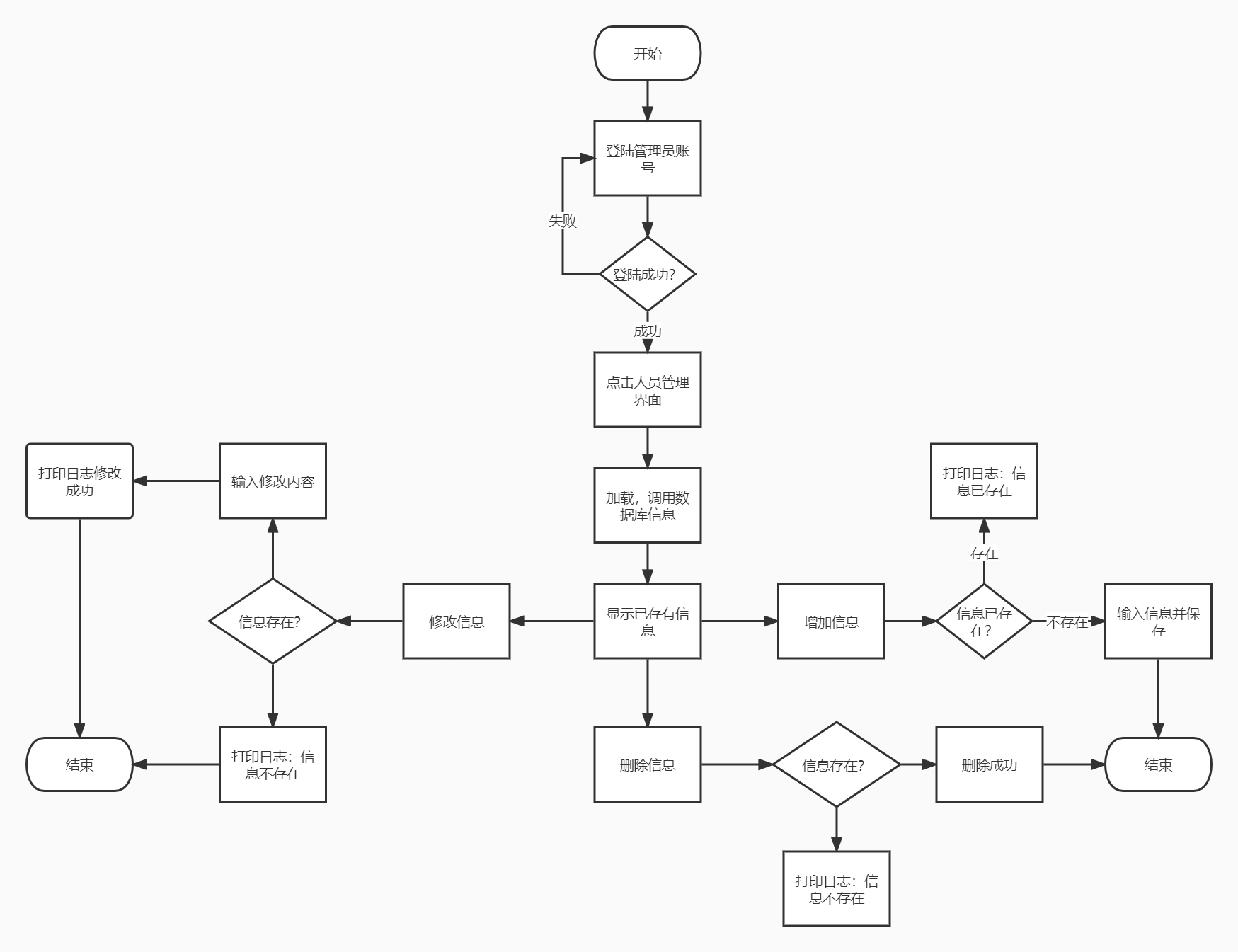
1. 界面：权限管理界面
2. 功能：调整用户和管理员的权限，如只读，读写等等
3. 使用数据库用户主键控制权限

# 流程图

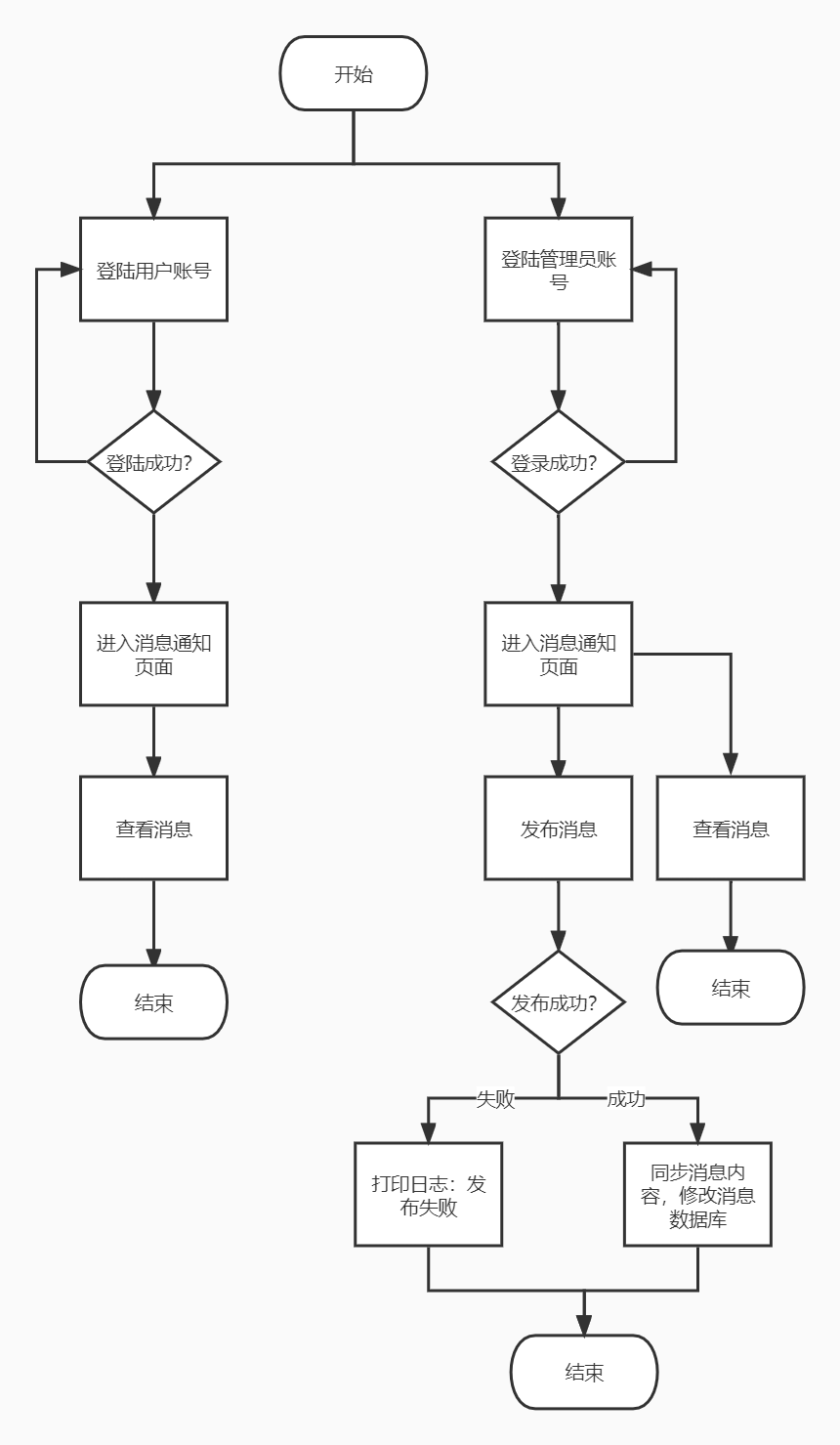
## 登录功能



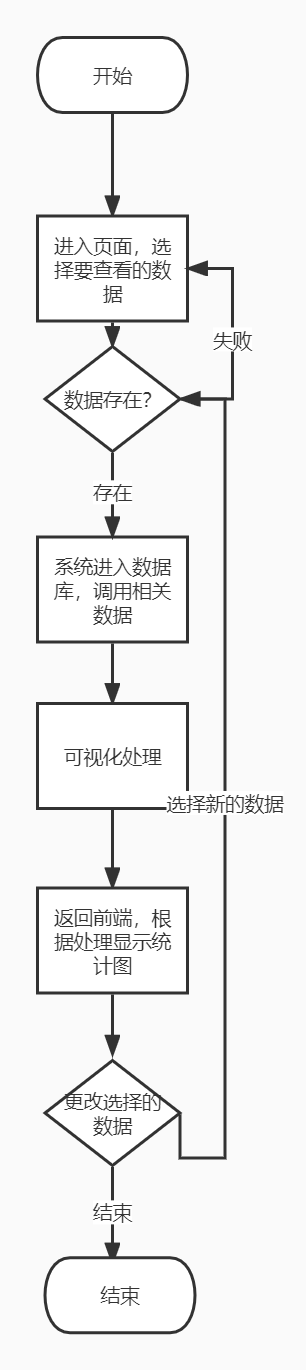
## 人员管理子页面



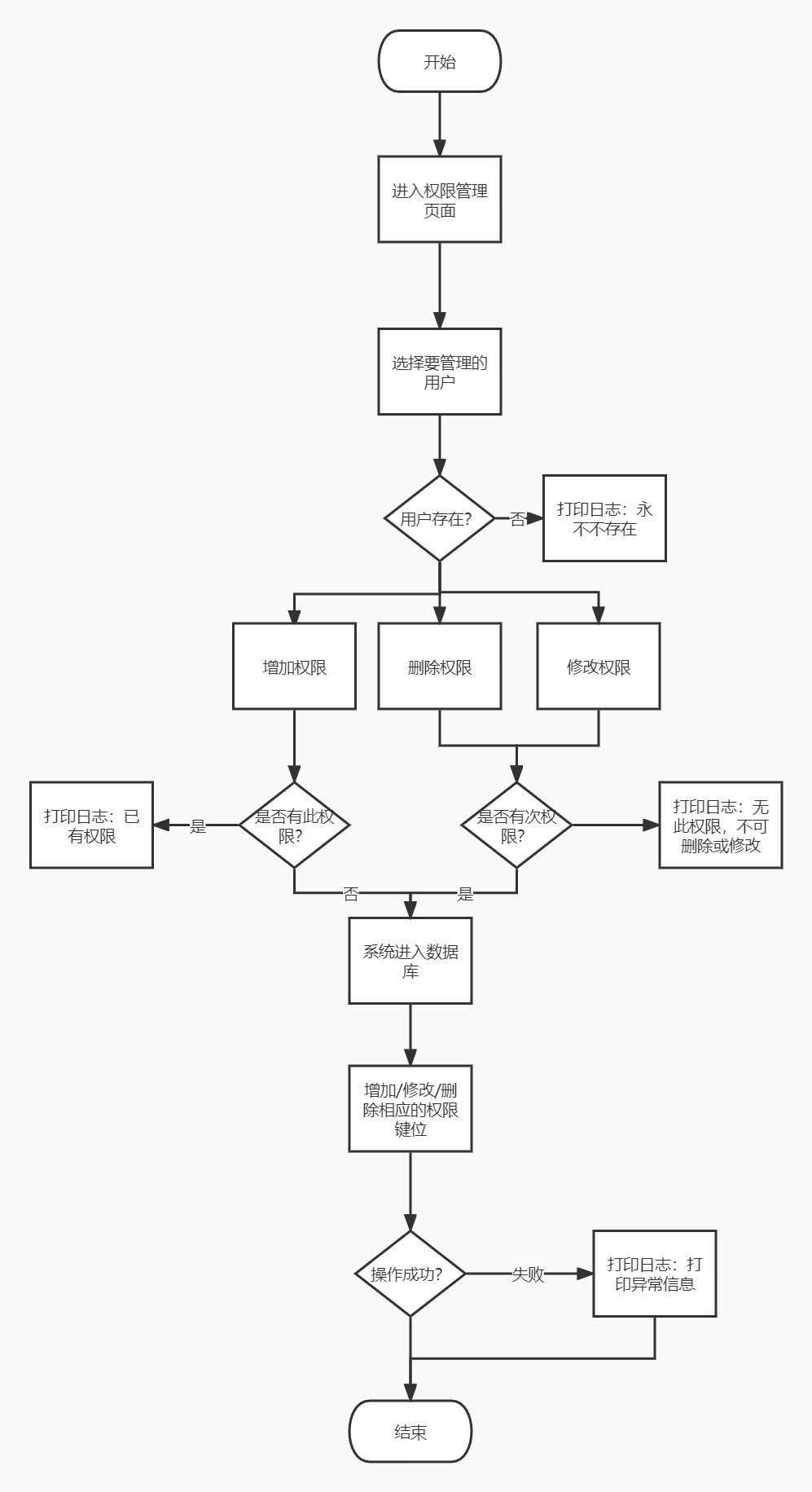
## 消息通知



## 数据统计



## 权限管理



# 接口设计

## 登录接口

### 接口信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 登录跳转 | | | |
| 地址 | /login | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| loginName | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| id | 用户ID | String |  |
| loginName | 用户名 | String |  |
| name | 昵称 | String |  |
| token | 登录凭证 | String | 所有请求都带此token |

### 处理逻辑

1. 登录校验，根据loginName获取用户信息
   1. 获取不到，打日志：用户名不存在；报错返回：用户名不存在或密码错误
   2. 获取到了，比较参数密码与数据库密码
      1. 相等，登录校验成功
      2. 不相等，打日志：密码不对；报错返回：用户名不存在或密码错误
2. 校验通过后，加载资源权限
3. 组装用户登录信息LoginUserDto
4. 生成唯一token，作为用户登录标识
5. 以token为key，LoginUserDto为value，存入redis中
6. 返回登录结果

## 老师数据接口

### 接口信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 获取json格式的数据 | | | |
| 地址 | /teachermanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 上传数据 | | | |
| 地址 | /teachermanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

### 处理逻辑

1. 连接people\_manage数据库
2. 获取目标地址的数据或上传数据

## 学生数据接口

### 接口信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 获取json格式的数据 | | | |
| 地址 | /studentmanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 上传数据 | | | |
| 地址 | /studentmanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

### 处理逻辑

1. 连接people\_manage数据库
2. 获取目标地址的数据或上传数据

## 其他职工数据接口

### 接口信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 获取json格式的数据 | | | |
| 地址 | /stuffmanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 上传数据 | | | |
| 地址 | /stuffmanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

### 处理逻辑

1. 连接people\_manage数据库
2. 获取目标地址的数据或上传数据

## 流动人员数据接口

### 接口信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 获取json格式的数据 | | | |
| 地址 | /mobilemanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 上传数据 | | | |
| 地址 | /mobilemanage | | | |
| 请求方式 | GET | | | |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| url | 地址 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| data | 数据 | json | json格式 |

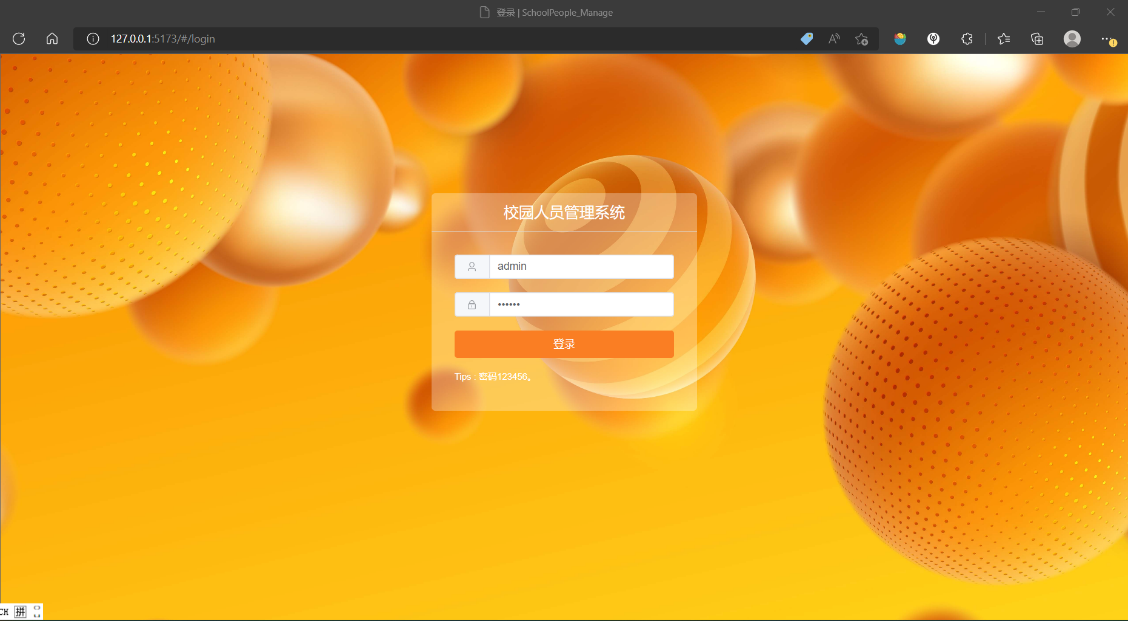
### 处理逻辑

1. 连接people\_manage数据库
2. 获取目标地址的数据或上传数据

# 页面设计

## 登录页面

### 页面展示



### 处理逻辑

1. 用户名、密码、非空校验
2. 点击登录按钮时，调用登录接口
   1. 登录成功，跳转到欢迎页面
   2. 登录失败，弹出失败提示

## 工作台首页

### 页面展示

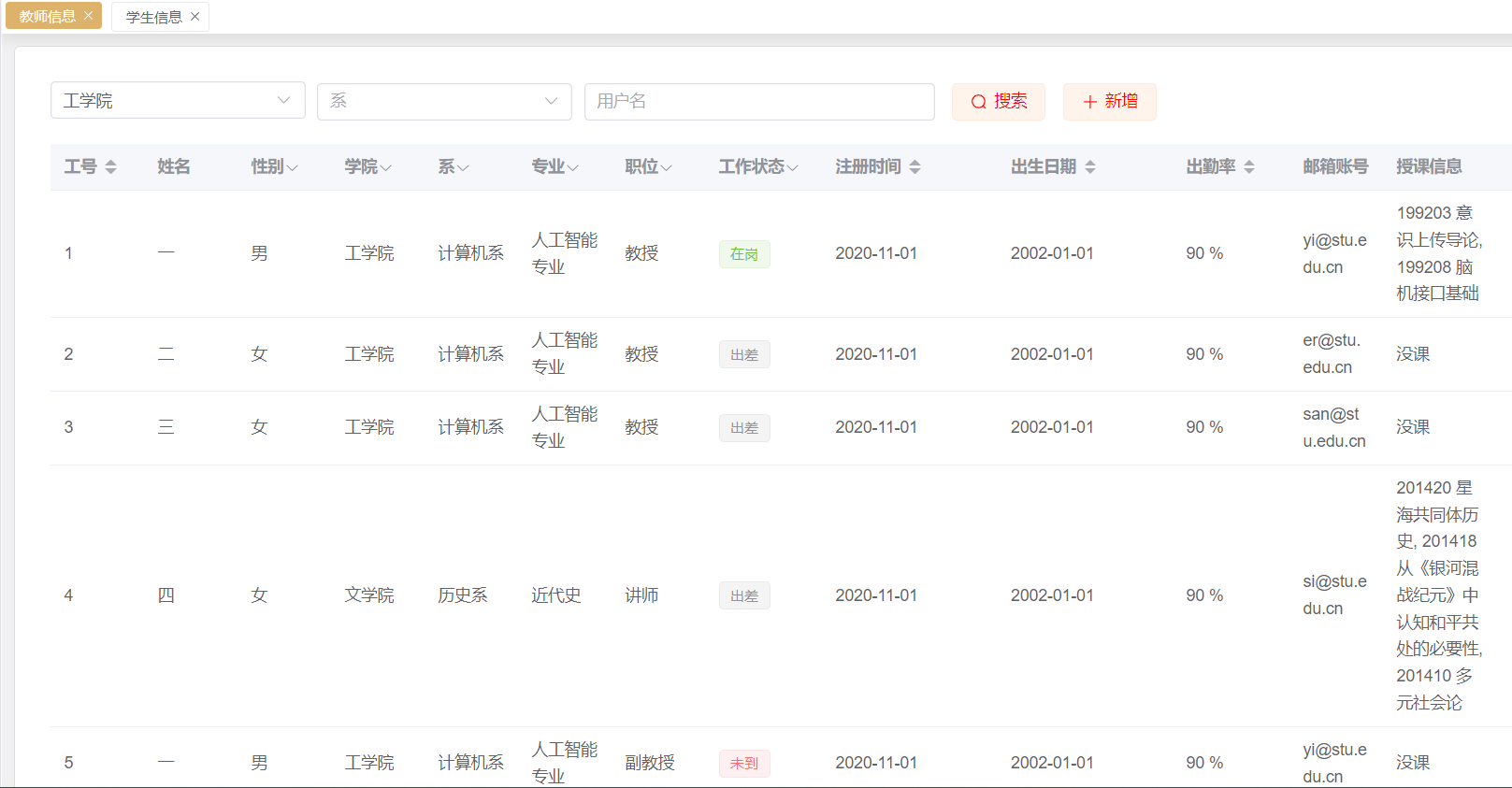


### 处理逻辑

1. 从数据库或本地缓存读取数据
2. 显示数据

### 页面展示

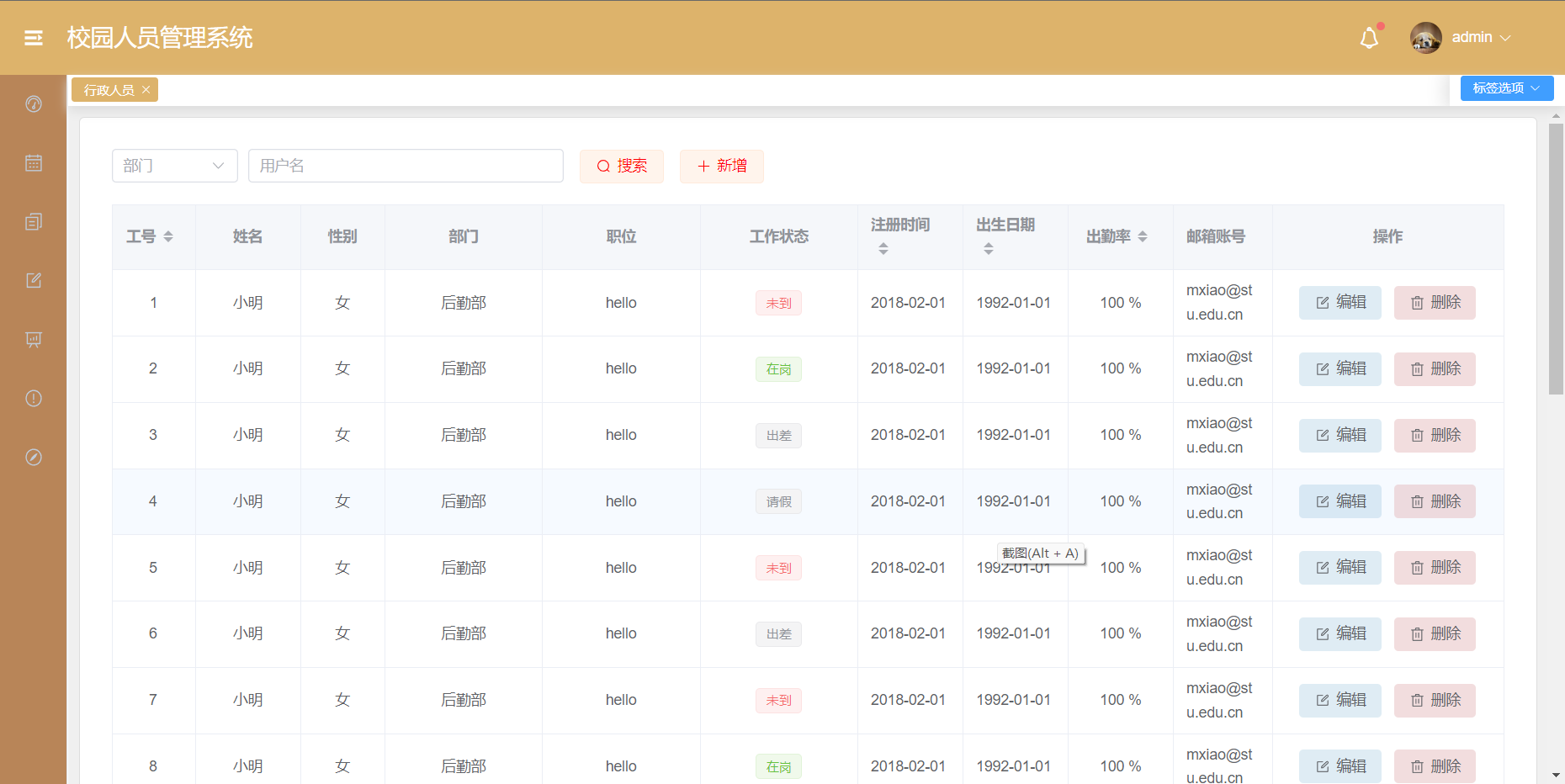
老师



学生



其他职工



流动人员



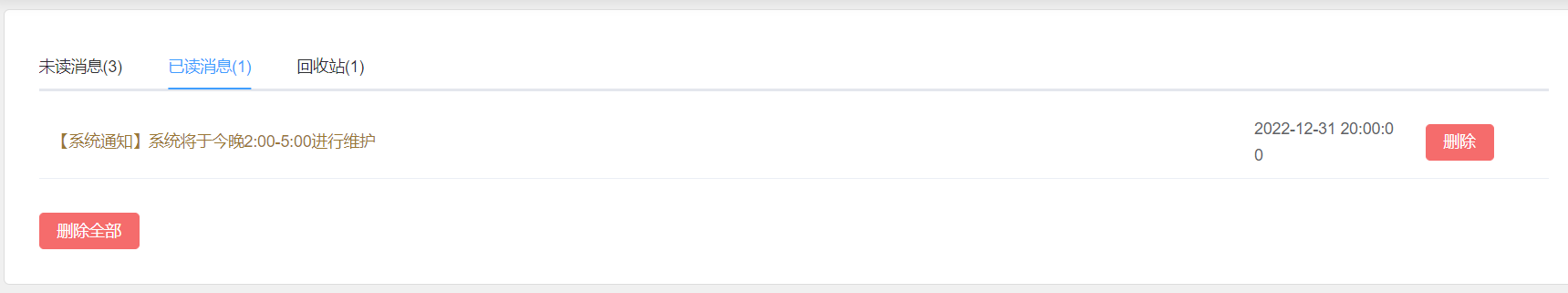
### 处理逻辑..TODO

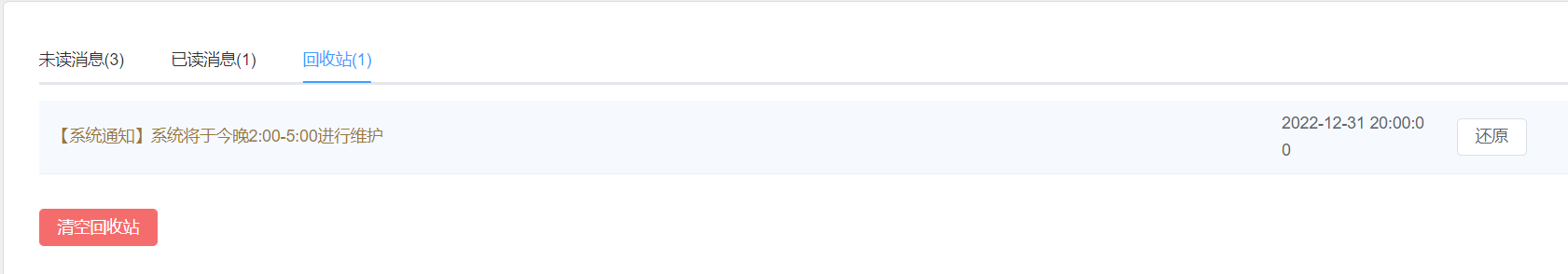
1. 调用数据库接口
   1. 调用成功，显示数据库数据
   2. 调用失败，显示本地缓存数据

## 消息通知

### 页面展示





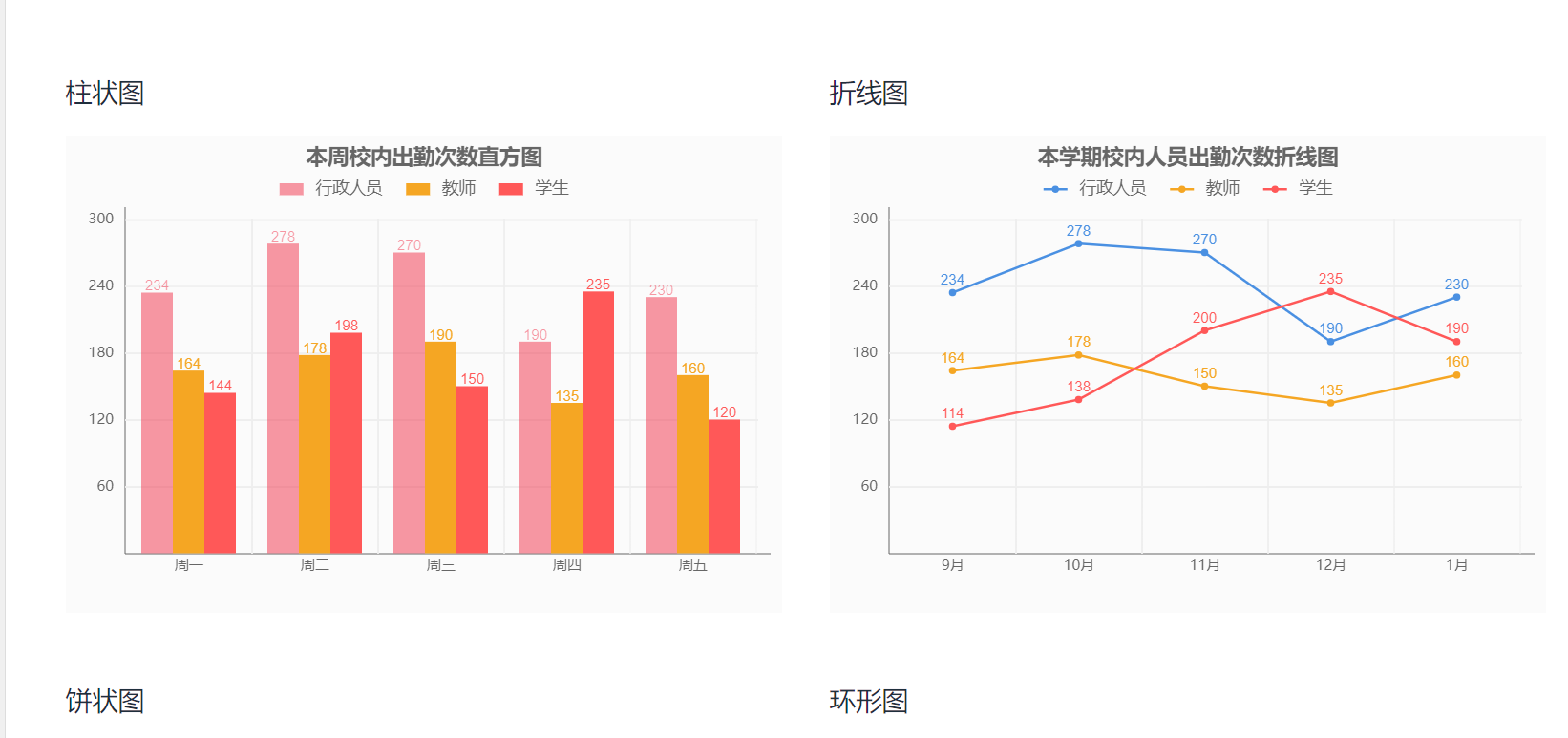


### 处理逻辑

1. 从本地数据取出对应数据
2. 点击已读时，将其从未读数组中剪切到已读，已读删除则剪切到回收站数组

## 数据统计

### 页面展示



### 处理逻辑

1. 调用缓存中的数据，计算对应的数据
2. 调用scharts函数展示

## 权限管理

### 页面展示



### 处理逻辑

1. 修改本地设置
2. 下次进入时，侧栏读取设置

## 通知发送

### 页面展示



### 处理逻辑

1. 使用户填写必写的“通知名称”，选择发送单位、日期等，并且设置优先级，选择性地填充通知详情
2. 点击表单提交，将表单数据上传至数据库
3. 重置表单则将表单状态恢复至默认

## 个人中心

### 页面展示



### 处理逻辑

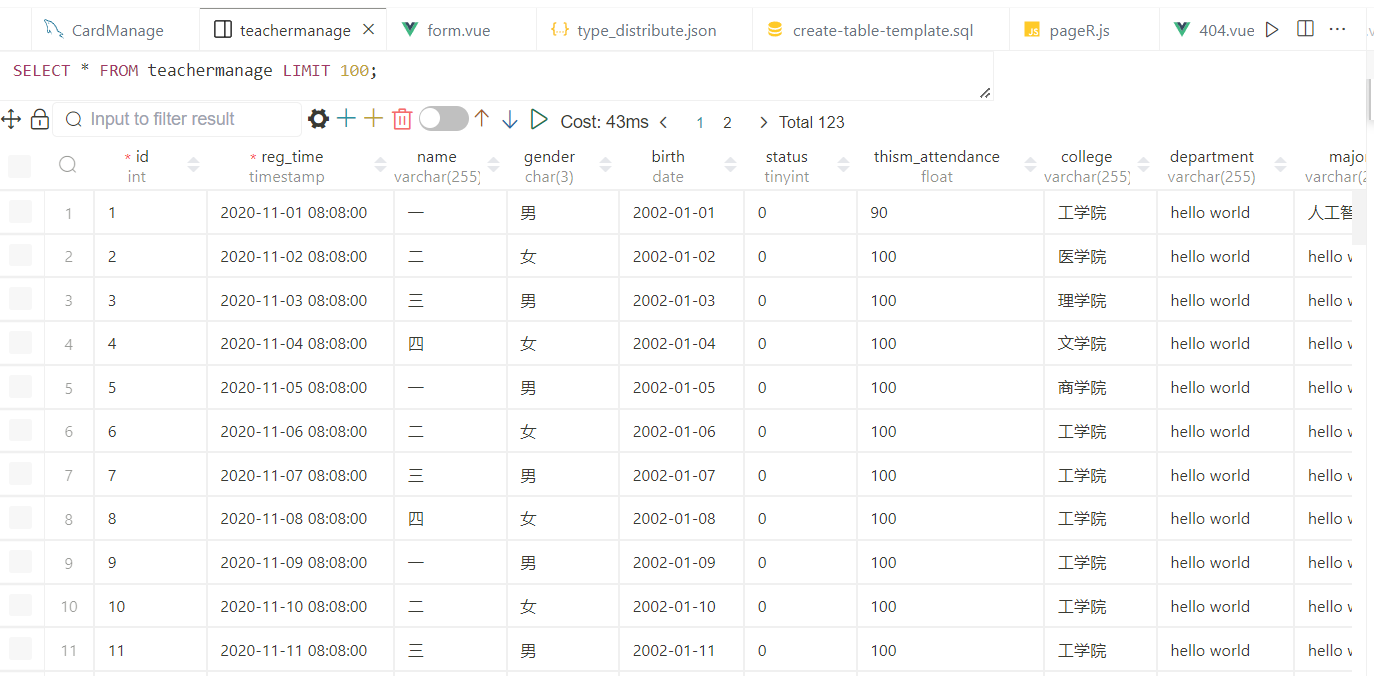
1. 首先获取本地缓存的信息，只有缓存没有时才会取数据库中的
2. 显示头像，简介和用户名
3. 更改密码功能
   1. 旧密码和缓存中一致时，可继续
   2. 点击保存按钮后，更新本地缓存密码
   3. 上传至数据库

# 数据库实施

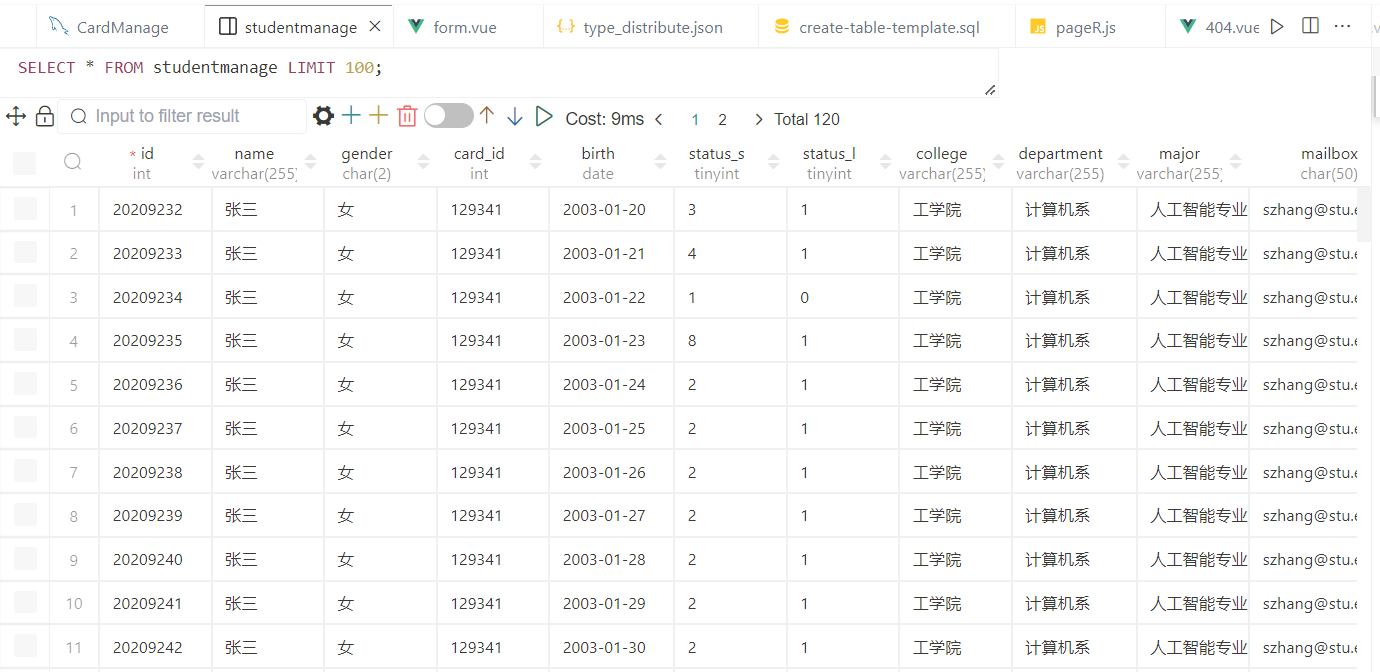
如设定，数据库people\_manage有四张表：teachermanage, studentmanage, stuffmanage, mobilemanage。第一是老师管理，二是学生，三是其他职工，最后是流动人员。

## 数据库物理设计

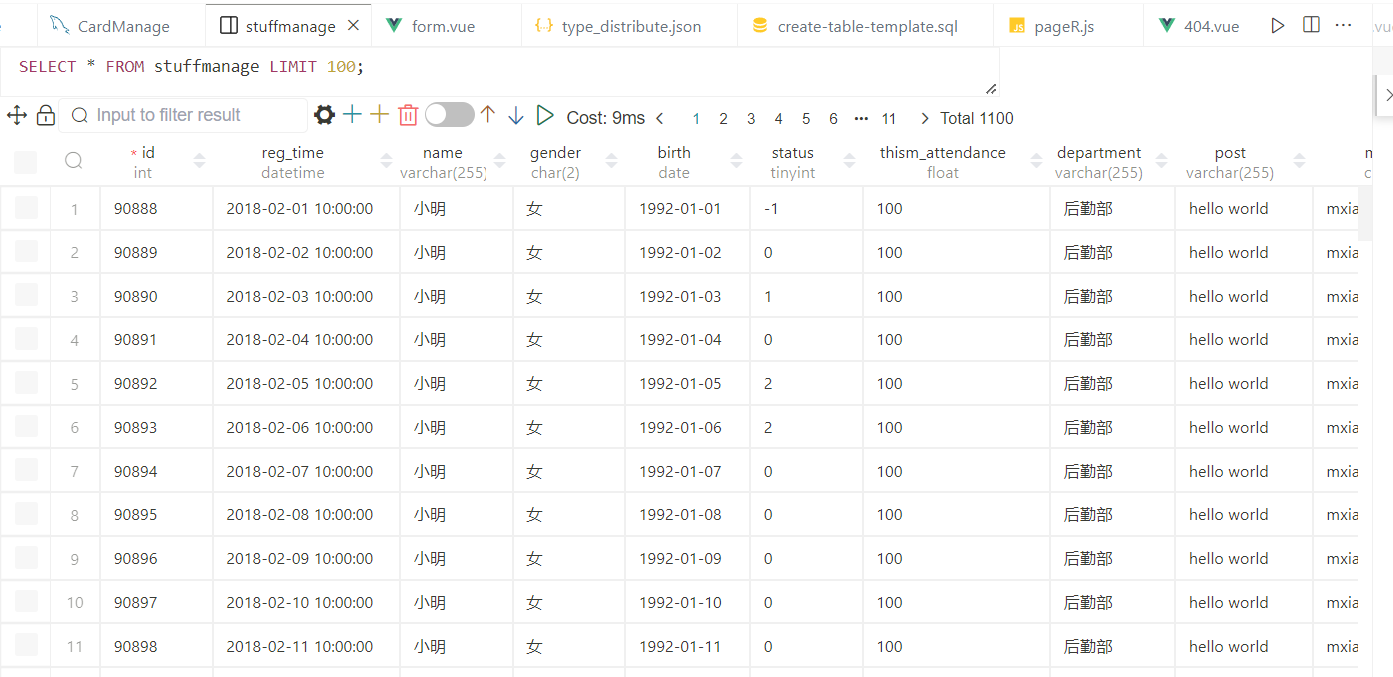
老师



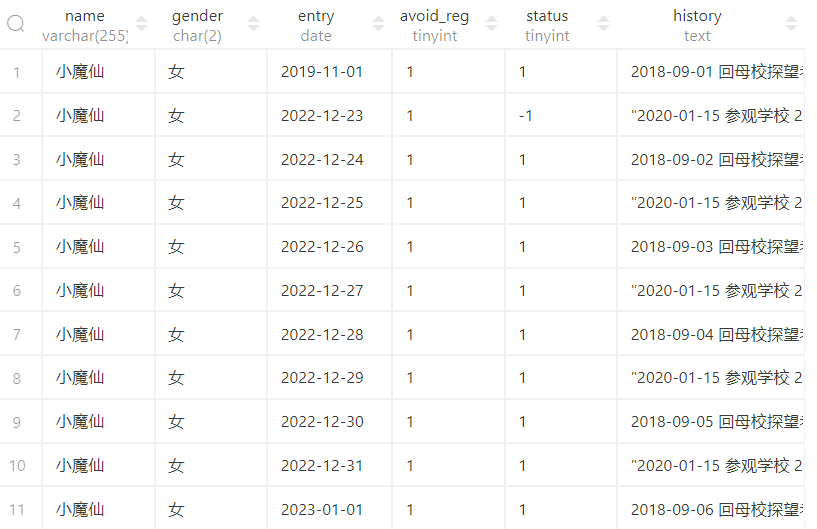
学生



其他职工

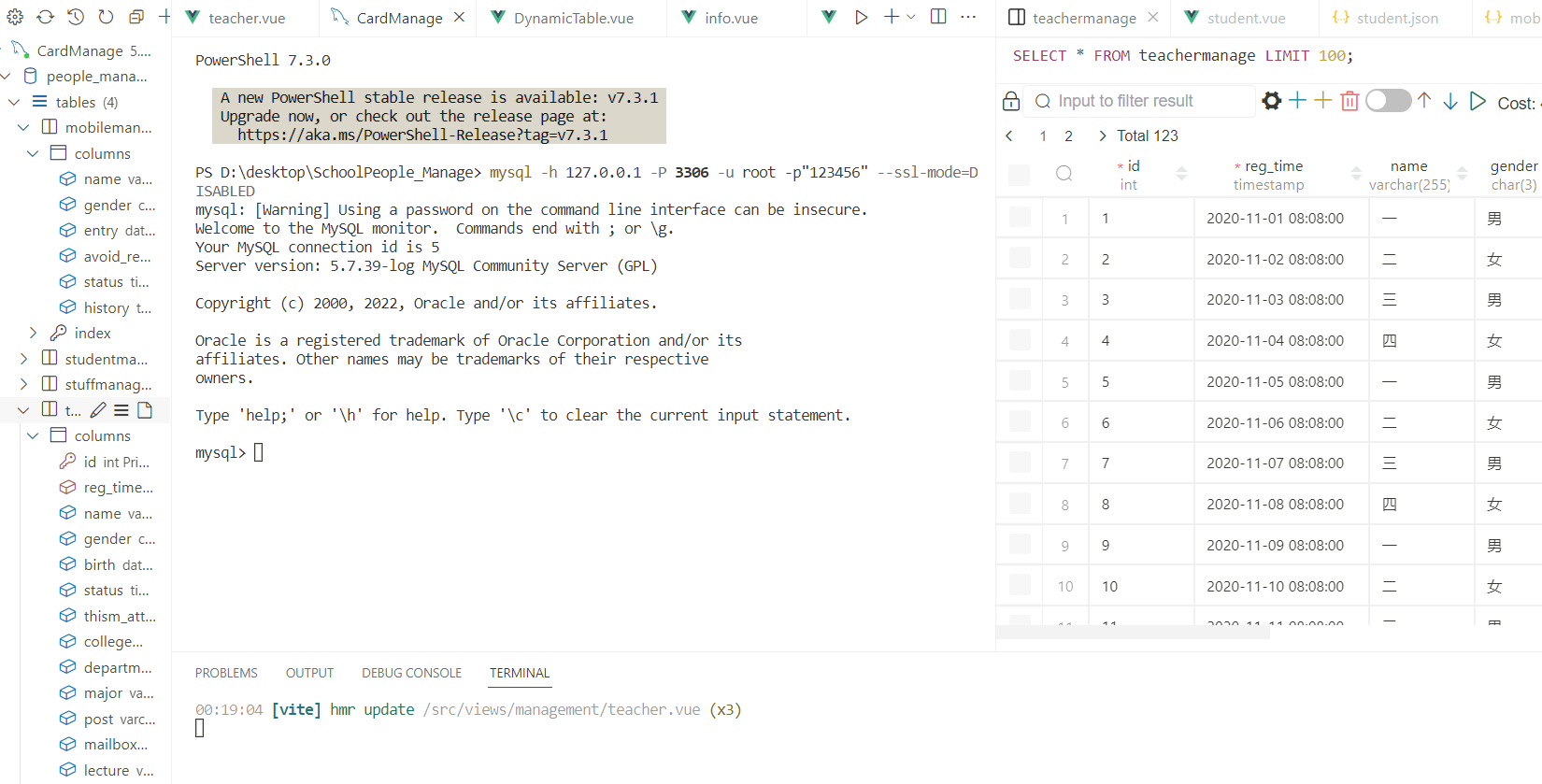


流动人员



## 数据库预览

左侧时数据库结构，中间是终端，右侧是数据库中的数据



## 数据库配置

以root登入mysql数据库端口127.0.0.1，连接people\_manage数据库



## 数据库接口

### node连接数据库

以下是node后端获取数据库值的方法，将数据库结果转为了json格式





### 前端连接node后端

封装axios请求接口，前端通过接口跨域请求后端数据

