



Web应用开发 技术报告

目录

[第一章 系统概述 4](#_Toc26968)

[第二章 系统框架 4](#_Toc7740)

[2.1 网络架构 5](#_Toc13715)

[2.2 系统架构 5](#_Toc30586)

[2.3 模块功能说明 7](#_Toc23604)

[第三章 系统需求分析 9](#_Toc10279)

[一、管理员登录模块 9](#_Toc10649)

[二、学院管理模块 9](#_Toc14926)

[三、专业管理模块 10](#_Toc14911)

[四、教师管理模块 11](#_Toc1369)

[五、学生管理模块 11](#_Toc3943)

[六、健康码管理模块 12](#_Toc2347)

[七、师生登录模块 13](#_Toc9094)

[八、健康码信息填报模块 13](#_Toc3884)

[九、健康码查询模块 14](#_Toc26377)

[第四章 系统详细设计 14](#_Toc28214)

[一、view层 14](#_Toc20175)

[二、controller层 2](#_Toc9047)1

[三、service层 3](#_Toc30898)2

[四、DAO层 3](#_Toc7676)8

第五章 数据库设计............................................................................................70

5.1数据库表概念结构设计................................................................................70

5.2数据库表概念关系模型................................................................................72

[第六章 系统实现功能 7](#_Toc9087)2

[6.1 管理员登录模块 7](#_Toc17029)2

[6.2 学院管理模块 7](#_Toc26549)4

[6.3 专业管理模块 7](#_Toc32399)5

[6.4 教师管理模块 7](#_Toc25716)6

[6.5 学生管理模块 7](#_Toc6949)7

[6.6 健康码管理模块 7](#_Toc12233)8

[6.7 师生登录模块 8](#_Toc804)0

[6.8 健康码信息填报模块 8](#_Toc25326)1

[6.9 健康码查询模块 8](#_Toc19700)3

[第七章 程序清单 8](#_Toc28129)4

1. **系统概述**

## 2019年底开始流行的新型冠状病毒（2019-nCoV）疫情给全世界人民带来了无法考量的灾难。在全国人民众志成城地抗疫下，我国的疫情于3月份逐渐得到了有效控制。浙江省人民政府新闻办公室于2020年4月3日下午举行第四十一场新冠肺炎疫情防控工作新闻发布会，浙江省教育厅公布了开学复课方案。为确保开学安全，浙江省各地教育主管部门和学校制定了一系列保障措施。一是在开学前，对即将返校的师生健康状况作了全面排摸，按“一人一档”要求建立师生健康信息档案。二是在开学后，师生继续凭“健康码+测体温”受控入校，健康状况有异常的不得进校，对因病请假缺课的学生要追踪病因。 本课程设计要求实现师生健康码管理系统的构建，具体包括管理员系统的构建以及师生系统的构建。管理员系统所面向的对象是本校的部分教师，管理员按照等级从高到低分别为：系统管理员、校级管理员、院级管理员。师生系统所面向的对象是本校的教师以及本校的学生。

**管理员系统：**

功能：设置学院、专业等信息并进行管理，可在后台导入教师数据和学生数据并进行管理，也可查询统计师生的健康码和每天的打卡情况。  
 操作：对学院、专业、教师、学生信息的添加、修改、查询，以及查询统计师生的健康码和每天的打卡情况。

**师生系统：**

功能：师生通过学校健康码页面填报信息，进行每日打卡，根据打卡信息进行判断，获得属于自己的二维码，并且可以查看自己的专属二维码作为进出校园的凭证。  
 操作：填报学校健康码页面的信息，获取、查看自己的二维码。修改自身电话号码，登录密码等信息。

1. **系统框架**

**2.1 网络架构**

本师生健康码管理系统使用客户端-服务器的网络架构(B/S 架构)，用户所浏览的的界面在浏览器上实现，并且绝大部分的事务逻辑都在服务器上完成处理。在前端，即客户端，以Structs 框架为基础，再加上HTML+JSP+JavaScript语言进行编写，负责系统的展示逻辑和少量的事务逻辑。在后端，即服务器端，通过Servlet技术接收和响应用户的请求，同时结合MySQL数据库，使用DAO设计完成数据库的连接、添加、修改、查询等操作，完成系统的绝大部分事务逻辑。

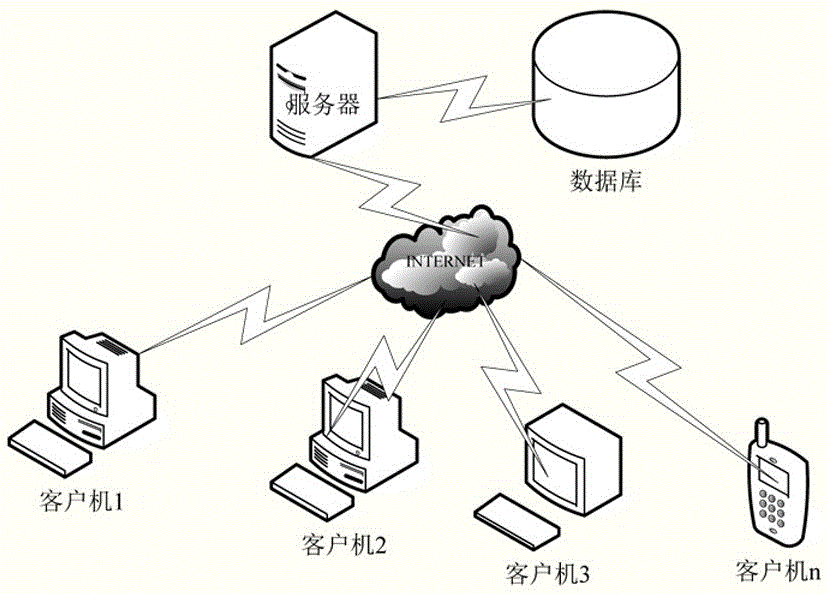


图1 B/S系统架构图

**2.2 系统架构**

本次课程设计完成的师生健康码管理系统共分为管理员系统、师生系统两个子系页。管理员系统有设置学院、专业等信息并进行管理，在后台导入教师数据和学生数据并进行管理，查询师生的健康码和每天的打卡情况等功能。其中管理员按照等级从高到低又分为系统管理员、校级管理员、院级管理员三个身份，不同级别的身份有不同的管理权限。

系统管理员可以设置全校的学院、专业等信息并进行管理，可在后台导入全校教师数据和学生数据并进行管理，也可查询全校师生的健康码和每天的打卡情况。

校级管理员可以设置全校的学院、专业、班级等信息并进行管理，也可查询全校师生的健康码和每天的打卡情况。

院级管理员可以设置本学院的学院、专业等信息并进行管理，可在后台导入本学院教师数据和学生数据并进行管理，也可查询本学院师生的健康码和每天的打卡情况。

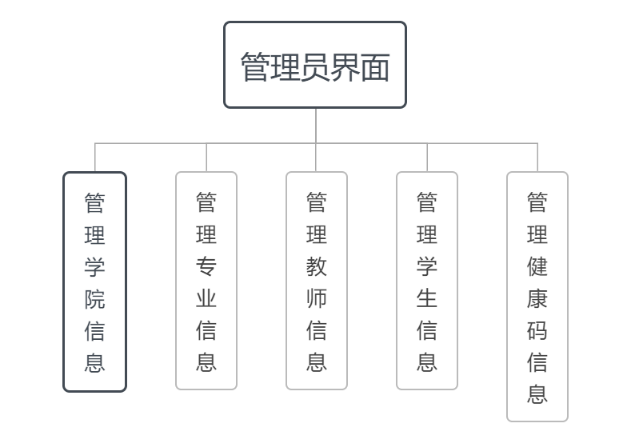


图2 管理员页面架构图

师生系统的功能有：一，通过学校健康码页面填报健康信息信息，获得师生的专属二维码。二，查看自己的专属二维码并且将其作为进出校园的凭证。三，修改个人基础信息。其中根据师生上报的信息所生成的健康码又分为绿码、黄码、红码三种。红色健康码的师生需连续打卡7天满足绿色健康码条件才可转为黄码；黄色健康码的师生需连续打卡7天满足绿色健康码条件才可转为绿码。

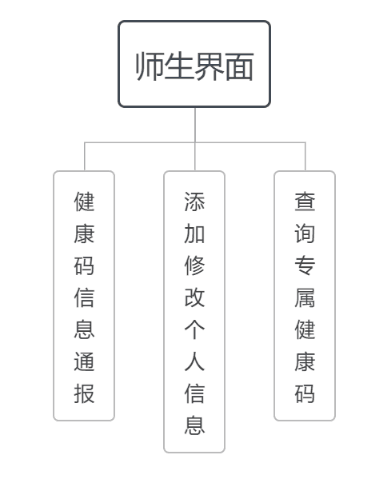


图3 师生界面架构图

除此之外，本师生健康码管理系统还与MySQL数据库建立的连接，能够做到将页面中信息的变化实时反映到MySQL数据库当中去。在MySQL数据库中，一共建立了7张数据表，分别为记录用户数据的user表，记录学院数据的college表，记录专业数据的major表，记录教师数据的teacher表，记录学生数据的student表，记录教师打卡信息的student\_record表以及记录学生打卡信息的teacher\_record表。

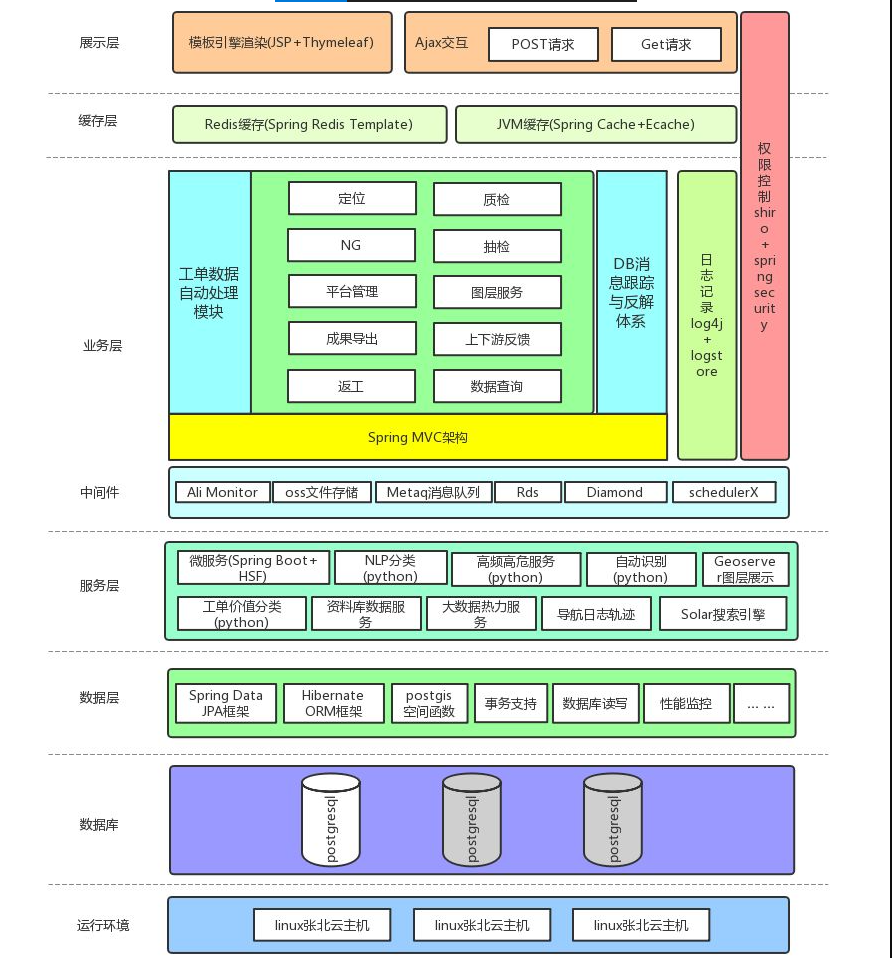


图4 MVC系统架构图

**2.3 模块功能说明**

本系统为多种类型，不同级别的用户提供不同的权限和操作，接下来按照管理员系统和师生系统对本系统的模块功能分别进行介绍。

**管理员系统：**

共分为管理员登录模块、学院管理模块、专业管理模块、教师管理模块、学生管理模块、健康码管理模块。

管理员登录模块提供用户登录的入口，在提交用户名和密码后进行账号密码验证，同时进行身份的验证，根据是系统管理员、校级管理员、院级管理员，跳转到对应的管理员系统。

学院管理模块有管理学院编号、学院名称、院长信息等功能，其具体操作有逐条增加学院信息、批量增加学院信息、查询所有的学院信息、按照学院名称模糊查询学院信息、按照学院编号模糊查询学院信息、修改学院信息以及删除学院信息。

专业管理模块用于管理专业编号、专业名称、专业所属学院信息，其具体操作有逐条增加专业信息、批量增加专业信息、查询所有的专业信息、按照专业名称模糊查询专业信息、按照专业编号模糊查询专业信息、修改专业信息以及删除专业信息。

教师管理模块用于管理教师工号、教师姓名、教师身份证号、教师所属学院、教师健康码状态、教师电话号码，其具体操作有逐条增加教师信息、批量增加教师信息、查询所有的教师信息、按照教师工号模糊查询教师信息、按照教师姓名模糊查询教师信息、修改教师信息以及删除教师信息。

学生管理模块用于管理学生学号、学生姓名、学生身份证号、学生所属学院、学生所属专业、学生电话号码、学生健康码状态，其具体操作有逐条增加学生信息、批量增加学生信息、查询所有的学生信息、按照学生学号模糊查询学生信息、按照学生姓名模糊查询学生信息、修改学生信息以及删除学生信息。

健康码管理模块用于查询师生打卡信息。其中师生打卡信息具体包括学号或工号、姓名、身份证号、手机号、身份、所属学院、打卡日期、打卡时间、健康码颜色、连续打卡绿码时间以及健康码目前状态信息。

**师生系统：**

共分为师生登录模块、健康码信息填报模块、健康码查询模块。

师生登录模块提供用户登录的入口，在提交后进行用户名与密码的验证，同时验证用户的身份，根据是教师还是学生，跳转到对应的用户系统。

健康码信息填报模块用于上传师生的基本信息，系统会根据师生所填写问卷选择的不同信息生成对应的健康码。并且若当前健康码为红色则需经过7天连续打卡且结果为绿色是方可转为黄色。持有黄色健康码的师生同样需经过7天连续打卡且结果为绿色是方可转为绿色

健康码查询模块用于查询师生打卡所生成的健康码，师生可以查看自己的专属二维码并且将其作为进出校园的凭证，绿码可以进出校园，黄码需居家观察7天不得进入校园，红码需居家医学观察或集中隔离14天不得进入校园。

1. **系统需求分析**

**一、管理员登录模块**

**1.功能描述**

输入账号和密码并选择角色后，按登录按钮，将信息发送到 Servlet 进行验证，若账号或密码或角色错误 则登录失败，需重新登录，若账号、密码、角色正确则跳转到对应的后续管理页面。

**2、业务处理流程及要求**

（1）输入的账号、密码，选择的角色均不能为空；

（2）若输入的账号、密码或选择的角色有错误，则提示错误信息，若输入的账号、密码信息为空，也会有相应的提示窗口进行对应信息的提示；

（3）若输入正确则按照管理员的类型进行相应的跳转，系统管理员跳转至系统管理员页面，校级管理员跳转至校级管理员页面，院级管理员跳转至院级管理员页面。

**3、输入信息**

账号、密码、角色（单选框）。

**4、输出信息**

若输入的账号、密码或选择的角色有错误，则提示错误信息，若输入的账号、密码信息为空，也会有相应的提示窗口进行对应信息的提示； 若输入正确则按照管理员的类型进行相应的跳转。

**二、学院管理模块**

**1.功能描述**

学院管理模块用于管理学院编号、学院名称信息，其具体操作有逐条增加学院信息、批量增加学院信息、查询所有的学院信息、按照学院名称模糊查询学院信息、修改学院信息以及删除学院信息，以上所有的操作都与MySQL数据库建立了联系。

1. **业务处理流程及要求**

（1）点击“查询所有学院信息”按钮，即可对所有的学院信息进行查询并显示。

（2）在文本框内输入学院的名称或是学院名称的部分文字，然后点击“查询”按钮，即可对该学院的信息进行模糊查询并显示。

（3）点击“新增学院信息”按钮，即可逐条进行学院信息的添加。

（4）点击“上传文件”文件按钮，选择对应的excel文件，即可完成对学院信息的批量导入。

**3、输入信息**

点击“查询”按钮，在文本框内输入学院的名称或是学院名称的部分文字，点击“新增学院信息”按钮，点击“上传文件”文件按钮，选择对应的excel文件。

**4、输出信息**

学院编号、学院名称信息。

**三、专业管理模块**

**1.功能描述**

专业管理模块用于管理专业编号、专业名称、专业所属学院信息，其具体操作有逐条增加专业信息、批量增加专业信息、查询所有的专业信息、按照专业名称模糊查询专业信息、按照学院编号模糊查询学院信息、修改专业信息以及删除专业信息。

**2、业务处理流程及要求**

（1）在文本框内输入专业的名称，然后点击“查询”按钮，即可对该专业的信息进行模糊查询并显示。

（2）点击“查询”按钮，即可对所有的专业信息进行查询并显示。

（3）点击“新增专业信息”按钮，即可逐条进行专业信息的添加。

（4）点击“上传文件”文件按钮，选择对应的excel文件，即可完成对专业信息的批量导入。

**3、输入信息**

点击“查询”按钮，在文本框内输入专业的名称或是专业名称的部分文字，点击“新增专业信息”按钮，点击“上传文件”文件按钮，选择对应的excel文件。

**4、输出信息**

专业编号、专业名称、专业所属学院信息。

**四、教师管理模块**

**1.功能描述**

教师管理模块用于管理教师工号、教师姓名、教师身份证号、教师所属学院、教师健康码状态、教师电话号码，其具体操作有逐条增加教师信息、批量增加教师信息、查询所有的教师信息、按照教师工号模糊查询教师信息、按照教师姓名模糊查询教师信息、修改教师信息以及删除教师信息。

**2、业务处理流程及要求**

（1）在文本框内输入教师的姓名，然后点击“查询”按钮，即可对该教师的信息进行模糊查询并显示。

（2）点击“查询”按钮，即可对所有的教师信息进行查询并显示。

（3）点击“新增教师信息”按钮，即可逐条进行教师信息的添加。

（4）点击“上传文件”文件按钮，选择对应的excel文件，即可完成对教师信息的批量导入。

**3、输入信息**

点击“查询”按钮，在文本框内输入教师的姓名，点击“新增教师信息”按钮，点击“选择”文件按钮，选择对应的excel文件，然后点击“批量导入数据”。

**4、输出信息**

教师工号、教师姓名、教师身份证号、教师所属学院、教师健康码状态、教师电话号码

**五、学生管理模块**

**1.功能描述**

学生管理模块用于管理学生学号、学生姓名、学生身份证号、学生所属学院、学生所属专业、学生电话号码、学生健康码状态，其具体操作有逐条增加学生信息、批量增加学生信息、查询所有的学生信息、按照学生学号模糊查询学生信息、按照学生姓名模糊查询学生信息、修改学生信息以及删除学生信息。

**2、业务处理流程及要求**

（1）在文本框内输入学生的姓名，然后点击“查询”按钮，即可对该学生的信息进行模糊查询并显示。

（2）点击“查询”按钮，即可对所有的学生信息进行查询并显示。

（3）点击“新增学生信息”按钮，即可逐条进行学生信息的添加。

（4）点击“上传文件”文件按钮，选择对应的excel文件，即可完成对学生信息的批量导入。

**3、输入信息**

点击“查询”按钮，在文本框内输入学生的姓名，点击“新增学生信息”按钮，点击“选择”文件按钮，选择对应的excel文件，然后点击“批量导入数据”。

**4、输出信息**

学生学号、学生姓名、学生身份证号、学生所属学院、学生所属专业、学生电话号码

**六、健康码管理模块**

**1.功能描述**

健康码管理模块用于查询师生打卡信息。其中师生打卡信息具体包括学号或工号、姓名、身份证号、手机号、身份、所属学院、打卡日期、打卡时间、健康码颜色、连续打卡绿码时间以及健康码目前状态信息。

**2、业务处理流程及要求**

（1）在文本框内输入教师的姓名或学生的姓名，然后点击“查询”按钮，即可对该教师或学生的信息进行模糊查询并显示。

（2）点击“查询”按钮，即可对所有的教师/学生信息进行查询并显示。

**3、输入信息**

点击“查询”按钮。

**4、输出信息**

学号或工号、姓名、身份证号、手机号、身份、所属学院、打卡日期、打卡时间、健康码颜色、连续打卡绿码时间、健康码目前状态信息。

**七、师生登录模块**

**1.功能描述**

师生登录模块提供用户登录的入口，在提交后进行相关信息的验证，同时验证用户的身份，根据是教师还是学生，跳转到对应的用户系统。

**2、业务处理流程及要求**

（1）初始密码为学号加身份证后8位，进入系统后可更改密码，选择的角色均不能为空；

（2）若输入的用户名，密码或选择的角色有错误或是输入的用户名，密码信息为空，则会跳出提示窗口进行对应信息的提示；

（3）若输入正确则按照角色的类型进行相应的跳转，教师跳转至教师页面，学生跳转至学生页面。

**3、输入信息**

用户名、密码、角色。

**4、输出信息**

若输入的用户名、密码和选择的角色正确则按照角色的类型进行相应的跳转。

**八、健康码信息填报模块**

**1.功能描述**

健康码信息填报模块用于上传师生的基本信息，系统会根据师生所填写问卷选择的不同信息生成对应的健康码。并且若当前健康码为红色则需经过7天连续打卡且结果为绿色是方可转为黄色。持有黄色健康码的师生同样需经过7天连续打卡且结果为绿色是方可转为绿色

**2、业务处理流程及要求**

（1）输入手机号不能为空；

（2）健康码填报的信息均不能为空，具体填报信息如下：

* 个人信息，包括姓名、身份证号、性别、手机号，其中（其中姓名、身份证号、工号或学号为系统默认值，不可修改）。
* 本人近期（14天内）是否去过重点疫区？
* 本人近期（14天内）是否去过国外？
* 本人近期（14天内）是否接触过新冠确诊病人或疑似病人？
* 本人是否被卫生部门确认为新冠肺炎确诊病例或疑似病例？
* 当前健康状况？无异常、发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻等。
* 本人郑重承诺：填报信息真实，愿意承担相应的法律责任。

（3）若输入信息完整则跳转页面，显示打卡成功，以及根据填报信息对应生成的健康码。

**3、输入信息**

手机号、健康码填报信息。

**4、输出信息**

以及根据填报信息对应生成的健康码。

**九、健康码查询模块**

**1.功能描述**

健康码查询模块用于查询师生打卡所生成的健康码，师生可以查看自己的专属二维码并且将其作为进出校园的凭证，绿码可以进出校园，黄码需居家观察7天不得进入校园，红码需居家医学观察或集中隔离14天不得进入校园。

**2、业务处理流程及要求**

（1）点击“教师/学生打卡情况查询”，即可查看自己的专属二维码作为进出校园的凭证，以及打卡的日期、时间以及姓名及身份。

**3、输入信息**

点击“教师/学生打卡情况查询”。

**4、输出信息**

查看自己的专属二维码作为进出校园的凭证，以及打卡的日期、时间，以及姓名及身份。

1. **系统详细设计**

**一、view层**

html页面1：welcome.html

|  |  |
| --- | --- |
| 描述 | 系统公告页面 |
| 所在工程目录 | /WebContent/ |
| 主要用途 | 用于展现系统公告，通知信息等 |

html页面2：**teacher\_punch\_query.html**

|  |  |
| --- | --- |
| 描述 | 教师打卡信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 数据接口 | record/queryTeacherRecordInfo |  |
| 主要用途 | 教师完成打卡上报后，其打卡信息经查询后会展现在该页 |
| 包含字段 | 编号、工号、姓名、性别、身份证号、学院、级别、打卡日期、打卡时间、健康码状态 |

html页面3：**teacher\_info\_query.html**

|  |  |
| --- | --- |
| 描述 | 教师信息查询页面 |
| 所在工程目录 | /WebContent/ |
|  |  |
| 主要用途 | 教师的基本注册信息查询后会展现在该页 |
| 包含字段 | 工号、姓名、性别、身份证号、手机号、学院 |

html页面4：**student\_punch\_query.html**

|  |  |
| --- | --- |
| 描述 | 学生打卡信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 数据接口 | record/queryStudentRecordInfo |
| 主要用途 | 学生完成打卡上报后，其打卡信息经查询后会展现在该页 |
| 包含字段 | 编号、学号、姓名、性别、身份证号、学院、专业、级别、打卡日期、打卡时间、健康码状态 |

html页面5：**student\_info\_query.html**

|  |  |
| --- | --- |
| 描述 | 学生信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 数据接口 | student/queryStudentInfo |
| 主要用途 | 学生的基本注册信息查询后会展现在该页 |
| 包含字段 | 学号、姓名、性别、身份证号、手机号、学院、专业 |

html页面6：college**\_info\_query.html**

|  |  |
| --- | --- |
| 描述 | 学院信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 数据接口 | college/queryCollegeInfo |
| 主要用途 | 查询学院信息，专业编号与名称对应 |
| 包含字段 | 学院编号、学院名称、院长 |

jsp页面1：login.jsp

|  |  |
| --- | --- |
| 描述 | 登录页面 |
| 所在工程目录 | /WebContent/ |
| 用途 | 登陆界面，为所有用户公用，设置权限分离，登陆后执行不同的跳转 |
| 表单提交地址 | user/login |
| 重点处理 | 捕获并展示后端传来的tips信息,方便用户知道发生的事件 |

jsp页面2：background\_system.jsp

|  |  |
| --- | --- |
| 描述 | 管理系统主页面 |
| 所在工程目录 | /WebContent/ |
| 用途 | 后台主界面，也是其他页面的容器，可提供给用户操作界面的工具、引导用户使用健康码系统 |
| 重点处理 | 通过替换iframe的src属性来替换展现不同的用户页 |
| 根据不同的登录角色来设计展现不同的页面，实现权限分离 |

jsp页面3：**teacher\_info\_add.jsp**

|  |  |
| --- | --- |
| 描述 | 教师添加页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | teacher/add |
| 主要用途 | 新增教师 |
| 包含字段 | 工号、姓名、性别、身份证号、手机号、学院、角色 |

jsp页面4：**student\_info\_add.jsp**

|  |  |
| --- | --- |
| 描述 | 学生添加页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | student/add |
| 主要用途 | 新增学生 |
| 包含字段 | 学号、姓名、性别、身份证号、手机号、学院、专业 |

jsp页面5：**person\_punch.jsp**

|  |  |
| --- | --- |
| 描述 | 个人打卡页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | record/punch |
| 主要用途 | 用户填写信息完成打卡上报，上报的信息将作为健康码状态的参考依据 |
| 重点处理 | 对于访问权限错误的用户，系统将不允许其访问，使其弹回登陆页面，并显示权限错误的提示信息 |
| 加入一些javascript代码用以控制填报打卡信息的逻辑 |

jsp页面6：**person\_info\_query.jsp**

|  |  |
| --- | --- |
| 描述 | 个人信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 主要用途 | 师生权限用户查询个人的基本信息，根据查询的用户的身份不同，我们分为教师和学生两种，两者展现的字段略有差别 |
| 包含字段 | 学号（学生）、工号（教师）、姓名、性别、身份证号、手机号、学院、专业（学生） |

jsp页面7：**person\_info\_modify.jsp**

|  |  |
| --- | --- |
| 描述 | 个人信息修改页面 |
| 所在工程目录 | /WebContent/ |
| 主要用途 | 结合实际情况，考虑到学校系统主要信息多为固定，不允许轻易更改，此功能仅可以修改自己的手机号字段，此字段也服务于后面的打卡信息展示 |
| 包含字段 | 手机号 |

jsp页面8：**passwordModify.jsp**

|  |  |
| --- | --- |
| 描述 | 修改密码页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | user/modifyPass |
| 主要用途 | 所有用户均有修改密码的功能需求，所以我们将其嵌入主界面的组件中，使得用户在任何界面都可以直接访问到修改密码的页面，十分的方便快捷 |
| 重点处理 | 修改密码需要提供原密码，并且保证两次输入的密码一致，对密码的强度有要求，要求长度在6-12位 |

jsp页面9：**major\_info\_query.jsp**

|  |  |
| --- | --- |
| 描述 | 专业信息查询页面 |
| 所在工程目录 | /WebContent/ |
| 数据接口 | major/queryMajorInfo |
| 主要用途 | 查询专业信息，专业编号与名称对应 |
| 包含字段 | 专业编号、专业名称、开设学院 |

jsp页面10：**major\_info\_add.jsp**

|  |  |
| --- | --- |
| 描述 | 专业添加页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | major/add |
| 主要用途 | 添加新专业，对应实际情况中的开设新专业 |
| 包含字段 | 专业编号、专业名称、开设学院 |

jsp页面11：**health\_code.jsp**

|  |  |
| --- | --- |
| 描述 | 健康码信息页面 |
| 所在工程目录 | /WebContent/ |
| 主要用途 | 添加新专业，对应实际情况中的开设新专业 |
| 包含字段 | 用户身份、用户姓名、打卡日期、打卡时间、健康码 |
| 重点处理 | 通过直接访问该页面是不可取的，必须优先登录，所以增加了直接访问跳转登录界面的功能 |
| 后端生产随机健康码base64数据串，前端使用base64解码展示 |

jsp页面12：college**\_info\_add.jsp**

|  |  |
| --- | --- |
| 描述 | 新增学院页面 |
| 所在工程目录 | /WebContent/ |
| 表单提交地址 | college/add |
| 主要用途 | 新增学院信息，新增的学院不可与之前存在的学院冲突 |
| 包含字段 | 学院编号、学院名称、院长 |

**二、controller层**

类1：CollegeAction

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | CollegeAction | | |
| 继承或实现 | 继承ActionSupport类 | | |
| 描述 | 学院服务接口类 | | |
| 类别 | Java | | |
| 包 | uhcms.Controller | | |
| 与其它类调用关系 | 调用了CollegeService，UserService，College，User类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| CollegeAction | Public | 构造函数 |
| getCollege | Public | 返回学院信息 |
| getInputStream | Public | 导入参数 |
| setInputStream | Public | 返回参数 |
| setCname | Public | 返回学院名 |
| getCname | Public | 导入学院名 |
| setCollege | Public | 导入学院信息 |
| add | Public | 添加学院信息在前后端之间的传输 |
| queryCollegeInfo | Public | 修改学院信息在前后端之间的传输 |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加学院信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 添加学院信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryCollegeInfo |
| 描述 | 修改学院信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 修改学院信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法3的说明

|  |  |
| --- | --- |
| 方法名称 | setCollege |
| 描述 | 导入学院信息 |
| 输入参数 | 无 |
| 步骤 | College college |
| 输出参数 | void |
| 注释 | 无 |

方法4的说明

|  |  |
| --- | --- |
| 方法名称 | setCname |
| 描述 | 导入学院名 |
| 输入参数 | String cname |
| 步骤 | 导入学院名 |
| 输出参数 | void |
| 注释 | 无 |

方法5的说明

|  |  |
| --- | --- |
| 方法名称 | getCollege |
| 描述 | 返回学院信息 |
| 输入参数 | 无 |
| 步骤 | 返回学院信息 |
| 输出参数 | String |
| 注释 | 无 |

方法6的说明

|  |  |
| --- | --- |
| 方法名称 | getCname |
| 描述 | 返回学院名 |
| 输入参数 | 无 |
| 步骤 | 返回学院名 |
| 输出参数 | String |
| 注释 | 无 |

方法7的说明

|  |  |
| --- | --- |
| 方法名称 | getInputStream |
| 描述 | 返回参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | InputStream |
| 注释 | 无 |

方法8的说明

|  |  |
| --- | --- |
| 方法名称 | setInputStream |
| 描述 | 导入参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | void |
| 注释 | 无 |

方法9的说明

|  |  |
| --- | --- |
| 方法名称 | CollegeAction |
| 描述 | 构造函数 |
| 输入参数 | 无 |
| 步骤 | 构造函数 |
| 输出参数 | 无 |
| 注释 | 无 |

类1：StudentAction

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | StudentAction | | |
| 继承或实现 | 继承ActionSupport类 | | |
| 描述 | 学生服务接口类 | | |
| 类别 | Java | | |
| 包 | uhcms.Controller | | |
| 与其它类调用关系 | 调用了StudentService，UserService，Student，User类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| StudentAction | Public | 构造函数 |
| getStudent | Public | 返回学生信息 |
| getInputStream | Public | 导入参数 |
| setInputStream | Public | 返回参数 |
| setname | Public | 返回学生名 |
| getname | Public | 导入学生名 |
| setStudent | Public | 导入学生信息 |
| add | Public | 添加学生信息在前后端之间的传输 |
| queryStudentInfo | Public | 修改学生信息在前后端之间的传输 |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加学生信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 添加学生信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryStudentInfo |
| 描述 | 修改学生信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 修改学生信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法3的说明

|  |  |
| --- | --- |
| 方法名称 | setStudent |
| 描述 | 导入学生信息 |
| 输入参数 | 无 |
| 步骤 | Student Student |
| 输出参数 | void |
| 注释 | 无 |

方法4的说明

|  |  |
| --- | --- |
| 方法名称 | setname |
| 描述 | 导入学生名 |
| 输入参数 | String name |
| 步骤 | 导入学生名 |
| 输出参数 | void |
| 注释 | 无 |

方法5的说明

|  |  |
| --- | --- |
| 方法名称 | getStudent |
| 描述 | 返回学生信息 |
| 输入参数 | 无 |
| 步骤 | 返回学生信息 |
| 输出参数 | String |
| 注释 | 无 |

方法6的说明

|  |  |
| --- | --- |
| 方法名称 | getname |
| 描述 | 返回学生名 |
| 输入参数 | 无 |
| 步骤 | 返回学生名 |
| 输出参数 | String |
| 注释 | 无 |

方法7的说明

|  |  |
| --- | --- |
| 方法名称 | getInputStream |
| 描述 | 返回参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | InputStream |
| 注释 | 无 |

方法8的说明

|  |  |
| --- | --- |
| 方法名称 | setInputStream |
| 描述 | 导入参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | void |
| 注释 | 无 |

方法9的说明

|  |  |
| --- | --- |
| 方法名称 | StudentAction |
| 描述 | 构造函数 |
| 输入参数 | 无 |
| 步骤 | 构造函数 |
| 输出参数 | 无 |
| 注释 | 无 |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | TeacherAction | | |
| 继承或实现 | 继承ActionSupport类 | | |
| 描述 | 教师服务接口类 | | |
| 类别 | Java | | |
| 包 | uhcms.Controller | | |
| 与其它类调用关系 | 调用了TeacherService，UserService，Teacher，User类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| TeacherAction | Public | 构造函数 |
| getTeacher | Public | 返回教师信息 |
| getInputStream | Public | 导入参数 |
| setInputStream | Public | 返回参数 |
| setname | Public | 返回教师名 |
| getname | Public | 导入教师名 |
| setTeacher | Public | 导入教师信息 |
| add | Public | 添加教师信息在前后端之间的传输 |
| queryTeacherInfo | Public | 修改教师信息在前后端之间的传输 |

类1：MajorAction

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | MajorAction | | |
| 继承或实现 | 继承ActionSupport类 | | |
| 描述 | 专业服务接口类 | | |
| 类别 | Java | | |
| 包 | uhcms.Controller | | |
| 与其它类调用关系 | 调用了MajorService，UserService，Major，User类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| MajorAction | Public | 构造函数 |
| getMajor | Public | 返回专业信息 |
| getInputStream | Public | 导入参数 |
| setInputStream | Public | 返回参数 |
| setname | Public | 返回专业名 |
| getname | Public | 导入专业名 |
| setMajor | Public | 导入专业信息 |
| add | Public | 添加专业信息在前后端之间的传输 |
| queryMajorInfo | Public | 修改专业信息在前后端之间的传输 |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加专业信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 添加专业信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryMajorInfo |
| 描述 | 修改专业信息在前后端之间的传输 |
| 输入参数 | 无 |
| 步骤 | 修改专业信息在前后端之间的传输 |
| 输出参数 | String |
| 注释 | 无 |

方法3的说明

|  |  |
| --- | --- |
| 方法名称 | setMajor |
| 描述 | 导入专业信息 |
| 输入参数 | 无 |
| 步骤 | Major Major |
| 输出参数 | void |
| 注释 | 无 |

方法4的说明

|  |  |
| --- | --- |
| 方法名称 | setname |
| 描述 | 导入专业名 |
| 输入参数 | String name |
| 步骤 | 导入专业名 |
| 输出参数 | void |
| 注释 | 无 |

方法5的说明

|  |  |
| --- | --- |
| 方法名称 | getMajor |
| 描述 | 返回专业信息 |
| 输入参数 | 无 |
| 步骤 | 返回专业信息 |
| 输出参数 | String |
| 注释 | 无 |

方法6的说明

|  |  |
| --- | --- |
| 方法名称 | getname |
| 描述 | 返回专业名 |
| 输入参数 | 无 |
| 步骤 | 返回专业名 |
| 输出参数 | String |
| 注释 | 无 |

方法7的说明

|  |  |
| --- | --- |
| 方法名称 | getInputStream |
| 描述 | 返回参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | InputStream |
| 注释 | 无 |

方法8的说明

|  |  |
| --- | --- |
| 方法名称 | setInputStream |
| 描述 | 导入参数 |
| 输入参数 | 无 |
| 步骤 | 返回参数 |
| 输出参数 | void |
| 注释 | 无 |

方法9的说明

|  |  |
| --- | --- |
| 方法名称 | MajorAction |
| 描述 | 构造函数 |
| 输入参数 | 无 |
| 步骤 | 构造函数 |
| 输出参数 | 无 |
| 注释 | 无 |

**三、service层**

类1：CollegeService

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | CollegeService | | |
| 继承或实现 |  | | |
| 描述 | 学院服务类 | | |
| 类别 | Java | | |
| 包 | uhcms.service | | |
| 与其它类调用关系 | 调用了CollegeDao类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| Add | Public | 添加学院信息 |
| queryCollegeInfo | Public | 修改学院信息 |
|  |  |  |
|  |  |  |
|  |  |  |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加学院信息 |
| 输入参数 | College college |
| 步骤 | 添加学院信息 |
| 输出参数 | boolean |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryCollegeInfo |
| 描述 | 修改学院信息 |
| 输入参数 | String name |
| 步骤 | 修改学院信息 |
| 输出参数 | ArrayList<College> |
| 注释 | 无 |

类2：TeacherService

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | TeacherService | | |
| 继承或实现 |  | | |
| 描述 | 学院服务类 | | |
| 类别 | Java | | |
| 包 | uhcms.service | | |
| 与其它类调用关系 | 调用了TeacherDao类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| Add | Public | 添加教师信息 |
| queryTeacherInfo | Public | 修改教师信息 |
|  |  |  |
|  |  |  |
|  |  |  |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加教师信息 |
| 输入参数 | Teacher teacher |
| 步骤 | 添加教师信息 |
| 输出参数 | boolean |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryTeacherInfo |
| 描述 | 修改教师信息 |
| 输入参数 | String name |
| 步骤 | 修改教师信息 |
| 输出参数 | ArrayList<teacher> |
| 注释 | 无 |

类3：StudentService

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | StudentService | | |
| 继承或实现 |  | | |
| 描述 | 学生服务类 | | |
| 类别 | Java | | |
| 包 | uhcms.service | | |
| 与其它类调用关系 | 调用了StudentDao类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| Add | Public | 添加学生信息 |
| queryStudentInfo | Public | 修改学生信息 |
|  |  |  |
|  |  |  |
|  |  |  |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加学生信息 |
| 输入参数 | Student Student |
| 步骤 | 添加学生信息 |
| 输出参数 | boolean |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryStudentInfo |
| 描述 | 修改学生信息 |
| 输入参数 | String name |
| 步骤 | 修改学生信息 |
| 输出参数 | ArrayList<Student> |
| 注释 | 无 |

类4：MajorService

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | MajorService | | |
| 继承或实现 |  | | |
| 描述 | 专业服务类 | | |
| 类别 | Java | | |
| 包 | uhcms.service | | |
| 与其它类调用关系 | 调用了MajorDao类 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| Add | Public | 添加专业信息 |
| queryMajorInfo | Public | 修改专业信息 |
|  |  |  |
|  |  |  |
|  |  |  |

方法1的说明

|  |  |
| --- | --- |
| 方法名称 | Add |
| 描述 | 添加专业信息 |
| 输入参数 | Major Major |
| 步骤 | 添加专业信息 |
| 输出参数 | boolean |
| 注释 | 无 |

方法2的说明

|  |  |
| --- | --- |
| 方法名称 | queryMajorInfo |
| 描述 | 修改专业信息 |
| 输入参数 | String name |
| 步骤 | 修改专业信息 |
| 输出参数 | ArrayList<Major> |
| 注释 | 无 |

1. **DAO层**

BaseDAO类

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | BaseDAO | | |
| 继承或实现 | 无 | | |
| 描述 | DAO设计基类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| BaseDAO () |  | 连接数据库基础设置 |
| getConnection() |  | 获取连接 |
|  |  |  |
|  |  |  |
|  |  |  |

BaseDAO()

|  |  |
| --- | --- |
| 方法名称 | BaseDAO () |
| 描述 | 连接数据库基础设置 |
| 输入参数 | 无 |
| 步骤 | 设置账号密码数据库类型 |
| 输出参数 | 无 |
|  |  |
| 方法名称 | getConnection() |
| 描述 | 返回连接 |
| 输入参数 | 无 |
| 步骤 | 获取连接并返回 |
| 输出参数 | Connection |
| 注释 |  |

CollegeDAO类

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | CollegeDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 学院DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| insertCollege | boolean | 插入学院信息 |
| SelectByFuzzyName | ArrayList<College> | 根据学院名模糊查找 |
| selectAllCollege() | ArrayList<College> | 查找所有学院 |
| selectCollegeByTno | College | 根据学院老师工号查找 |
| modifyCollegeByCollege | boolean | 修改学院信息 |
| deleteCollegeByCno | boolean | 删除学院信息 |

insertCollege

|  |  |
| --- | --- |
| 方法名称 | insertCollege |
| 描述 | 插入学院信息 |
| 输入参数 | College |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

SelectByFuzzyName

|  |  |
| --- | --- |
| 方法名称 | SelectByFuzzyName |
| 描述 | 根据学院名模糊查找 |
| 输入参数 | String name |
| 步骤 |  |
| 输出参数 | ArrayList<College> |
| 注释 |  |

selectAllCollege()

|  |  |
| --- | --- |
| 方法名称 | selectAllCollege() |
| 描述 | 查找所有学院 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<College> |
| 注释 |  |

selectCollegeByTno

|  |  |
| --- | --- |
| 方法名称 | selectCollegeByTno |
| 描述 | 根据学院老师工号查找 |
| 输入参数 | String Tno |
| 步骤 |  |
| 输出参数 | College |
| 注释 |  |

modifyCollegeByCollege

|  |  |
| --- | --- |
| 方法名称 | modifyCollegeByCollege |
| 描述 | 修改学院信息 |
| 输入参数 | College |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

deleteCollegeByCno

|  |  |
| --- | --- |
| 方法名称 | deleteCollegeByCno |
| 描述 | 删除学院信息 |
| 输入参数 | String cno |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

MajorDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | MajorDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 专业DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| insertMajor | boolean | 插入专业信息 |
| selectByFuzzyMname | ArrayList<Major> | 根据专业名模糊查找 |
| selectByFuzzyTnoMname | ArrayList<Major> | 根据专业名模糊查找 |
| selectAllMajor | ArrayList<Major> | 查找所有专业 |
| selectMajorByTno | ArrayList<Major> | 查找学院信息按照教师工号 |
| modifyMajorByMajor | boolean | 修改专业信息 |
| deleteMajorByMno | boolean | 删除专业 |

insertMajor

|  |  |
| --- | --- |
| 方法名称 | insertMajor |
| 描述 | 插入专业信息 |
| 输入参数 | Major |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

selectByFuzzyMname

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzyMname |
| 描述 | 根据专业名模糊查找 |
| 输入参数 | String mname |
| 步骤 |  |
| 输出参数 | ArrayList<Major> |
| 注释 |  |

selectAllMajor

|  |  |
| --- | --- |
| 方法名称 | selectAllMajor |
| 描述 | 查找所有专业 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<Major> |
| 注释 |  |

modifyMajorByMajor

|  |  |
| --- | --- |
| 方法名称 | modifyMajorByMajor |
| 描述 | 修改专业信息 |
| 输入参数 | Major |
| 步骤 |  |
| 输出参数 | Boolean |
| 注释 |  |

deleteMajorByMno

|  |  |
| --- | --- |
| 方法名称 | deleteMajorByMno |
| 描述 | 删除专业信息 |
| 输入参数 | String Mno |
| 步骤 |  |
| 输出参数 | Boolean |
| 注释 |  |

RecordDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | RecordDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 打卡信息DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| insertStudentRecord | boolean | 通过学生类插入 |
| insertTeacherRecord | boolean | 通过教师类插入 |
| selectStudentRecordBySno | ArrayList<StudentRecord> | 按照学号查询打卡记录，并按照时间降序排列 |
| selectTeacherRecordByTno | ArrayList<TeacherRecord> | 根据教师号查找打卡记录 |
| selectAllStudentRecord() | ArrayList<StudentRecord> | 查询所有学生打卡记录降序排列 |
| selectAllTeacherRecord() | ArrayList<TeacherRecord> | 查找所有打卡记录 |
| selectStudentRecordByFuzzySname | ArrayList<StudentRecord> | 按照学生姓名模糊查询打卡记录，降序排列 |
| selectTeacherRecordByFuzzyTname | ArrayList<TeacherRecord> | 根据教师姓名模糊查询 |
| selectAllStudentRecordByRdate | ArrayList<StudentRecord> | 按照时间查询打卡记录 |
| selectAllTeacherRecordByRdate | ArrayList<TeacherRecord> | 根据日期查找打卡记录 |
| checkIfStudentColorChange | boolean | 检测学生是否达到标准 |
| checkIfTeacherColorChange | boolean | 检测教师是否达到标准 |
| selectTeacherRecordByFuzzyTnameAndCollege | List<TeacherRecord> | 根据老师准确学院和模糊姓名查询 |
| selectStudentRecordByFuzzySnameAndCollege | List<StudentRecord> | 根据学生准确学院和模糊姓名查询 |
|  |  |  |

insertStudentRecord

|  |  |
| --- | --- |
| 方法名称 | insertStudentRecord |
| 描述 | 通过学生类插入 |
| 输入参数 | StudentRecord |
| 步骤 |  |
| 输出参数 | Boolean |
| 注释 |  |

insertTeacherRecord

|  |  |
| --- | --- |
| 方法名称 | insertTeacherRecord |
| 描述 | 通过学生类插入 |
| 输入参数 | TeacherRecord |
| 步骤 |  |
| 输出参数 | Boolean |
| 注释 |  |

selectStudentRecordBySno

|  |  |
| --- | --- |
| 方法名称 | selectStudentRecordBySno |
| 描述 | 按照学号查询打卡记录，并按照时间降序排列 |
| 输入参数 | String sno |
| 步骤 |  |
| 输出参数 | ArrayList<StudentRecord> |
| 注释 |  |

selectTeacherRecordBySno

|  |  |
| --- | --- |
| 方法名称 | selectTeacherRecordBySno |
| 描述 | 按照工号查询打卡记录，并按照时间降序排列 |
| 输入参数 | String sno |
| 步骤 |  |
| 输出参数 | ArrayList<TeacherRecord> |
| 注释 |  |

selectAllStudentRecord()

|  |  |
| --- | --- |
| 方法名称 | selectAllStudentRecord() |
| 描述 | 查询所有打卡记录 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<StudentRecord> |
| 注释 |  |

selectAllTeacherRecord()

|  |  |
| --- | --- |
| 方法名称 | selectAllTeacherRecord() |
| 描述 | 查询所有打卡记录 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<TeacherRecord> |
| 注释 |  |

selectStudentRecordByFuzzySname

|  |  |
| --- | --- |
| 方法名称 | selectStudentRecordByFuzzySname |
| 描述 | 按照学生姓名模糊查询打卡记录，并按照时间降序排列 |
| 输入参数 | String sname |
| 步骤 |  |
| 输出参数 | ArrayList<StudentRecord> |
| 注释 |  |

selectTeacherRecordByFuzzySname

|  |  |
| --- | --- |
| 方法名称 | selectTeacherRecordByFuzzySname |
| 描述 | 按照姓名模糊查询打卡记录，并按照时间降序排列 |
| 输入参数 | String tname |
| 步骤 |  |
| 输出参数 | ArrayList<TeacherRecord> |
| 注释 |  |

selectAllStudentRecordByRdate()

|  |  |
| --- | --- |
| 方法名称 | selectAllStudentRecordByRdate() |
| 描述 | 根据日期查找打卡记录 |
| 输入参数 | String s\_time1 |
| 步骤 |  |
| 输出参数 | ArrayList<StudentRecord> |
| 注释 |  |

selectAllTeacherRecordByRdate()

|  |  |
| --- | --- |
| 方法名称 | selectAllTeacherRecordByRdate() |
| 描述 | 根据日期查找打卡记录 |
| 输入参数 | String s\_time1 |
| 步骤 |  |
| 输出参数 | ArrayList<TeacherRecord> |
| 注释 |  |

checkIfStudentColorChange()

|  |  |
| --- | --- |
| 方法名称 | checkIfStudentColorChange() |
| 描述 | 检测学生是否达到标准 |
| 输入参数 | String sno, String color |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

checkIfTeacherColorChange

|  |  |
| --- | --- |
| 方法名称 | checkIfTeacherColorChange |
| 描述 | 检测教师是否达到标准 |
| 输入参数 | String sno, String color |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

selectTeacherRecordByFuzzyTnameAndCollege()

|  |  |
| --- | --- |
| 方法名称 | selectTeacherRecordByFuzzyTnameAndCollege() |
| 描述 | 根据老师准确学院和模糊姓名查询 |
| 输入参数 | String name, String college |
| 步骤 |  |
| 输出参数 | List<TeacherRecord> |
| 注释 |  |

selectStudentRecordByFuzzySnameAndCollege

|  |  |
| --- | --- |
| 方法名称 | selectStudentRecordByFuzzySnameAndCollege |
| 描述 | 根据学生准确学院和模糊姓名查询 |
| 输入参数 | String name, String college |
| 步骤 |  |
| 输出参数 | List<StudentRecord> |
| 注释 |  |

StudentDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | StudentDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 学生类DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| insertStudent | boolean | 添加学生 |
| selectByFuzzySname | ArrayList<Student> | 按照名字模糊查询 |
| selectByFuzzySnameAndCollege | ArrayList<Student> | 按照名字模糊查询某一学院的学生信息 |
| selectByFuzzyTnoSno | ArrayList<Student> | 查询与教师号同学院的按学号进行模糊查询的学生 |
| selectAllStudent() | ArrayList<Student> | 查询所有学生 |
| selectCollegestudentByTno | ArrayList<Student> | 查询与教师工号同学院的老师 |
| selectStudentBySno | Student | 根据学号查找学生 |
| modifyStudentByStudent | boolean | 修改学生信息 |
| deleteStudentBySno | boolean | 删除学生 |

insertStudent()

|  |  |
| --- | --- |
| 方法名称 | insertStudent() |
| 描述 | 添加学生 |
| 输入参数 | Student student |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

selectByFuzzySname()

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzySname() |
| 描述 | 按照名字模糊查询 |
| 输入参数 | String sname |
| 步骤 |  |
| 输出参数 | ArrayList<Student> |
| 注释 |  |

selectByFuzzySnameAndCollege()

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzySnameAndCollege() |
| 描述 | 按照名字模糊查询某一学院的学生信息 |
| 输入参数 | String sname, String college |
| 步骤 |  |
| 输出参数 | ArrayList<Student> |
| 注释 |  |

selectByFuzzyTnoSno()

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzyTnoSno() |
| 描述 | 按照名字模糊查询某一学院的学生信息 |
| 输入参数 | String number1, String number2 |
| 步骤 |  |
| 输出参数 | ArrayList<Student> |
| 注释 |  |

selectAllStudent()

|  |  |
| --- | --- |
| 方法名称 | selectAllStudent() |
| 描述 | 查询所有学生 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<Student> |
| 注释 |  |

selectCollegestudentByTno()

|  |  |
| --- | --- |
| 方法名称 | selectCollegestudentByTno() |
| 描述 | 查询与教师工号同学院的老师 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | ArrayList<Student> |
| 注释 |  |

selectStudentBySno

|  |  |
| --- | --- |
| 方法名称 | selectStudentBySno |
| 描述 | 根据学号查找学生 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | Student |
| 注释 |  |

modifyStudentByStudent()

|  |  |
| --- | --- |
| 方法名称 | modifyStudentByStudent() |
| 描述 | 修改学生信息 |
| 输入参数 | Student student |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

deleteStudentBySno()

|  |  |
| --- | --- |
| 方法名称 | deleteStudentBySno() |
| 描述 | 修改学生信息 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

TeacherDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | TeacherDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 教师类DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| insertTeacher | boolean | 添加教师 |
| selectByFuzzyTname | ArrayList<Teacher> | 按照名字模糊查询 |
| selectByFuzzyTcollegeTname | ArrayList<Teacher> | 根据学院和姓名模糊查询教师 |
| selectAllTeacher | ArrayList<Teacher> | 查询与教师号同学院的按学号进行模糊查询的学生 |
| selectTeacherByTno | Teacher | 查找所有教师 |
| selectCollegeTeacherByTcollege | ArrayList<Teacher> | 根据学院查找教师 |
| modifyTeacherByTeacher | boolean | 修改教师信息 |
| deleteTeacherByTno | boolean | 删除教师 |

insertTeacher()

|  |  |
| --- | --- |
| 方法名称 | insertTeacher() |
| 描述 | 添加教师 |
| 输入参数 | Teacher teacher |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

selectByFuzzyTname()

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzyTname() |
| 描述 | 根据姓名模糊查询 |
| 输入参数 | String tname |
| 步骤 |  |
| 输出参数 | ArrayList<Teacher> |
| 注释 |  |

selectByFuzzyTcollegeTname()

|  |  |
| --- | --- |
| 方法名称 | selectByFuzzyTcollegeTname() |
| 描述 | 根据学院和姓名模糊查询教师 |
| 输入参数 | String college,String tname |
| 步骤 |  |
| 输出参数 | ArrayList<Teacher> |
| 注释 |  |

selectAllTeacher()

|  |  |
| --- | --- |
| 方法名称 | selectAllTeacher() |
| 描述 | 查找所有教师 |
| 输入参数 |  |
| 步骤 |  |
| 输出参数 | ArrayList<Teacher> |
| 注释 |  |

selectCollegeTeacherByTcollege()

|  |  |
| --- | --- |
| 方法名称 | selectCollegeTeacherByTcollege() |
| 描述 | 根据学院查找教师 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | ArrayList<Teacher> |
| 注释 |  |

selectTeacherByTno()

|  |  |
| --- | --- |
| 方法名称 | selectTeacherByTno() |
| 描述 | 根据工号查找教师 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | Teacher |
| 注释 |  |

modifyTeacherByTeacher()

|  |  |
| --- | --- |
| 方法名称 | modifyTeacherByTeacher() |
| 描述 | 修改教师信息 |
| 输入参数 | Teacher teacher |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

deleteTeacherByTno()

|  |  |
| --- | --- |
| 方法名称 | deleteTeacherByTno() |
| 描述 | 删除教师 |
| 输入参数 | String number |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

UserDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | TeacherDAO | | |
| 继承或实现 | BaseDAO类 | | |
| 描述 | 用户类DAO设计类 | | |
| 类别 |  | | |
| 包 | Uhcms.dao | | |
| 与其它类的调用关系 | 是 | | |
| 主要方法 | 名称 | 属性 | 描述 |
| selectPasswordByUsernameAndRole | String | 通过用户名与角色查找密码 |
| selectPasswordByUsername | String | 通过用户名查找密码 |
| updatePasswordByUsername | boolean | 更新密码 |
| insertUserInfo | boolean | 添加用户信息 |

selectPasswordByUsernameAndRole()

|  |  |
| --- | --- |
| 方法名称 | selectPasswordByUsernameAndRole() |
| 描述 | 通过用户名与角色查找密码 |
| 输入参数 | String username,String role |
| 步骤 |  |
| 输出参数 | String |
| 注释 |  |

selectPasswordByUsername()

|  |  |
| --- | --- |
| 方法名称 | selectPasswordByUsername() |
| 描述 | 通过用户名查找密码 |
| 输入参数 | String username |
| 步骤 |  |
| 输出参数 | String |
| 注释 |  |

updatePasswordByUsername()

|  |  |
| --- | --- |
| 方法名称 | updatePasswordByUsername() |
| 描述 | 更新密码 |
| 输入参数 | String username,String password |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

insertUserInfo()

|  |  |
| --- | --- |
| 方法名称 | insertUserInfo() |
| 描述 | 添加用户信息 |
| 输入参数 | String username, String password, String role |
| 步骤 |  |
| 输出参数 | boolean |
| 注释 |  |

Major

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | Major | | |
| 继承或实现 | 无 | | |
| 描述 | 专业信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.MajorDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| mname | String | 专业名 |
| mno | String | 专业号 |
| mcollege\_no | String | 所属学院号 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 主要方法 | 名称 | 属性 | 描述 |
| getMname |  | 返回专业名 |
| setMname |  | 设置专业名 |
| getMno |  | 返回专业号 |
| setMno |  | 设置专业号 |
| getMcollege\_no |  | 返回学院号 |
| setMcollege\_no |  | 设置学院号 |
|  |  |  |  |
| 名称 | College | | |
| 继承或实现 | 无 | | |
| 描述 | 学院信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.CollegeDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| Cname | String | 学院名 |
| Cno | String | 学院号 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 主要方法 | 名称 | 属性 | 描述 |
| getCName |  | 返回学院名 |
| setCName |  | 设置学院名 |
| getCno |  | 返回学院号 |
| setCno |  | 设置学院号 |
|  |  |  |

Student

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | Student | | |
| 继承或实现 | 无 | | |
| 描述 | 学生信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.StudentDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| Sno | String | 学号 |
| name | String | 姓名 |
| sex | String | 性别 |
| id | String | 身份证号 |
| number | String | 电话号码 |
| college | String | 学院 |
| major | String | 专业 |
| state | String | 健康码状态 |
| 主要方法 | 名称 | 属性 | 描述 |
| getSno/setSno | String/void | 设置/返回学号 |
| get/setName | String/void | 设置/返回姓名 |
| get/setSex | String/void | 设置/返回性别 |
| get/setId | String/void | 设置/返回身份证号 |
| get/setNumber | string/void | 设置/返回电话号码 |
| get/setCollege | String/void | 设置/返回学院 |
| get/setMajor | String/void | 设置/返回专业 |
| get/setState | String/void | 设置/返回健康码状态 |

Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | Teacher | | |
| 继承或实现 | 无 | | |
| 描述 | 教师信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.TeacherDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| tno | String | 工号 |
| tname | String | 姓名 |
| tsex | String | 性别 |
| tid | String | 身份证号 |
| tnumber | String | 电话号码 |
| tcollege | String | 学院 |
| tstate | String | 健康码状态 |
| 主要方法 | 名称 | 属性 | 描述 |
| getSno/setTno | String/void | 设置/返回工号 |
| get/setTname | String/void | 设置/返回姓名 |
| get/setTsex | String/void | 设置/返回性别 |
| get/setTid | String/void | 设置/返回身份证号 |
| get/setTnumber | string/void | 设置/返回电话号码 |
| get/setTcollege | String/void | 设置/返回学院 |
| get/setTstate | String/void | 设置/返回健康码状态 |

User

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | User | | |
| 继承或实现 | 无 | | |
| 描述 | 用户信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.UserDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| username | String | 用户名 |
| password | String | 密码 |
| role | String | 级别 |
| 主要方法 | 名称 | 属性 | 描述 |
| get/setUsername | String/void | 设置/返回用户名 |
| get/setPassword | String/void | 设置/返回密码 |
| get/setRole | String/void | 设置/返回级别 |

StudentRecord

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | StudentRecord | | |
| 继承或实现 | 无 | | |
| 描述 | 学生打卡信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.StudentRecordDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| rno | String | 打卡序号 |
| sno | String | 学号 |
| snumber | String | 电话号码 |
| judgement | String | 判断结果 |
| current\_state | String | 当前状态 |
| rtime | String | 打卡时间 |
| rdate | String | 打卡日期 |
| 主要方法 | 名称 | 属性 | 描述 |
| get/setRno | String/void | 设置/返回序号 |
| get/setSno | String/void | 设置/返回学号 |
| get/setSnumber | String/void | 设置/返回电话号码 |
| get/setJudgement | String/void | 设置/返回判断结果 |
| get/setCurrent\_state | String/void | 设置/返回当前状态 |
| get/setRtime | String/void | 设置/返回打卡时间 |
| get/setRdate | String/void | 设置/返回打卡日期 |

TeacherRecord

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | TeacherRecord | | |
| 继承或实现 | 无 | | |
| 描述 | 学生打卡信息类 | | |
| 类别 |  | | |
| 包 | Uhcms.model | | |
| 与其它类的调用关系 | 被uhcms.dao.TeacherRecordDAO调用 | | |
| 属性 | 名称 | 类型 | 描述 |
| rno | String | 打卡序号 |
| tno | String | 学号 |
| tnumber | String | 电话号码 |
| judgement | String | 判断结果 |
| current\_state | String | 当前状态 |
| rtime | String | 打卡时间 |
| rdate | String | 打卡日期 |
| 主要方法 | 名称 | 属性 | 描述 |
| get/setRno | String/void | 设置/返回序号 |
| get/setTno | String/void | 设置/返回学号 |
| get/setTnumber | String/void | 设置/返回电话号码 |
| get/setJudgement | String/void | 设置/返回判断结果 |
| get/setCurrent\_state | String/void | 设置/返回当前状态 |
| get/setRtime | String/void | 设置/返回打卡时间 |
| get/setRdate | String/void | 设置/返回打卡日期 |

**第五章 数据库设计**

5.1数据库表概念结构设计

* 实体1：学院(College)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| cno | 学院编号 | VARCHAR | 长度<=20 |
| cname | 学院名 | VARCHAR | 长度<=20 |
| cdean\_no | 院长 | VARCHAR | 长度<=6 |

* 实体2：专业(Major)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| mno | 专业编号 | VARCHAR | 长度<=20 |
| mname | 专业名 | VARCHAR | 长度<=20 |
| mcollege\_no | 所属学院 | VARCHAR | 长度<=20 |

* 实体3：学生(Student)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| sno | 学号 | VARCHAR | 长度<=20 |
| sname | 姓名 | VARCHAR | 长度<=20 |
| ssex | 性别 | ENUM('男', '女') |  |
| sid | 身份证号 | VARCHAR | 长度=18 |
| snumber | 电话号码 | VARCHAR | 长度<=20 |
| scollege | 所属学院 | VARCHAR | 长度<=20 |
| smajor | 所属专业 | VARCHAR | 长度<=20 |
| sstate | 当前状态 | VARCHAR | ENUM('绿', '黄', '红', '蓝') |

* 实体4：教师(Teacher)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| tno | 工号 | VARCHAR | 长度<=20 |
| tname | 姓名 | VARCHAR | 长度<=20 |
| tsex | 性别 | ENUM | ('男', '女') |
| tid | 身份证号 | VARCHAR | 长度=18 |
| tnumber | 电话号码 | VARCHAR | 长度<=20 |
| tcollege | 所属学院 | VARCHAR | 长度<=20 |
| tstate | 当前状态 | ENUM | ('绿', '黄', '红', '蓝') |

* 实体5：用户(User)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| role | 用户级别 | ENUM | ('admin', 'school', 'college', 'personal') |
| username | 用户名 | VARCHAR | 长度<=20 |
| password | 用户密码 | VARCHAR | 长度<=50 |

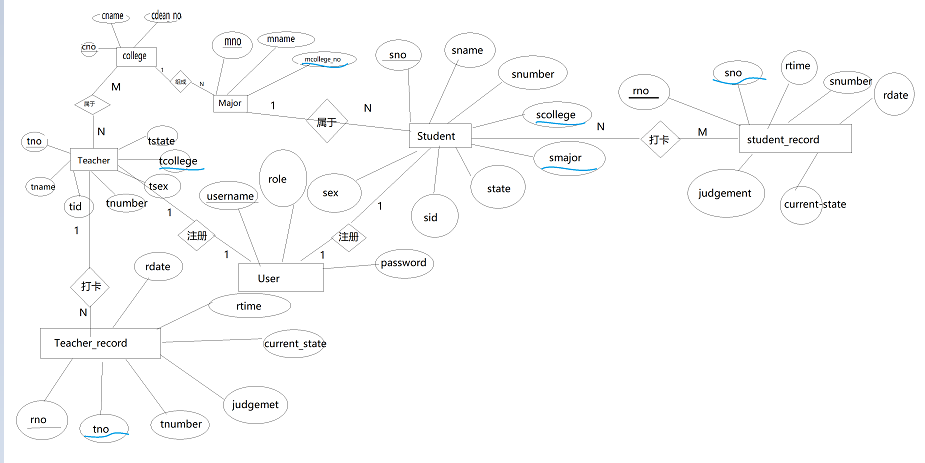
* 实体6：学生打卡信息(Teacher)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| rno | 序号 | INT | 自动增长 |
| sno | 学名 | VARCHAR | 长度<=20 |
| snumber | 电话号码 | VARCHAR | 长度<=20 |
| judgement | 判断结果 | ENUM | ('绿', '黄', '红', '蓝') |
| current\_state | 当前状态 | ENUM | ('绿', '黄', '红', '蓝') |
| rtime | 打卡时间 | VARCHAR | 长度<=20 |
| rdate | 打卡日期 | VARCHAR | 长度<=20 |

* 实体7：教师打卡信息(Teacher)

|  |  |  |  |
| --- | --- | --- | --- |
| 标识符 | 意义 | 类型 | 值域 |
| rno | 序号 | INT | 自动增长 |
| tno | 工名 | VARCHAR | 长度<=20 |
| tnumber | 电话号码 | VARCHAR | 长度<=20 |
| judgement | 判断结果 | ENUM | ('绿', '黄', '红', '蓝') |
| current\_state | 当前状态 | ENUM | ('绿', '黄', '红', '蓝') |
| rtime | 打卡时间 | VARCHAR | 长度<=20 |
| rdate | 打卡日期 | VARCHAR | 长度<=20 |

5.2数据库表概念关系模型

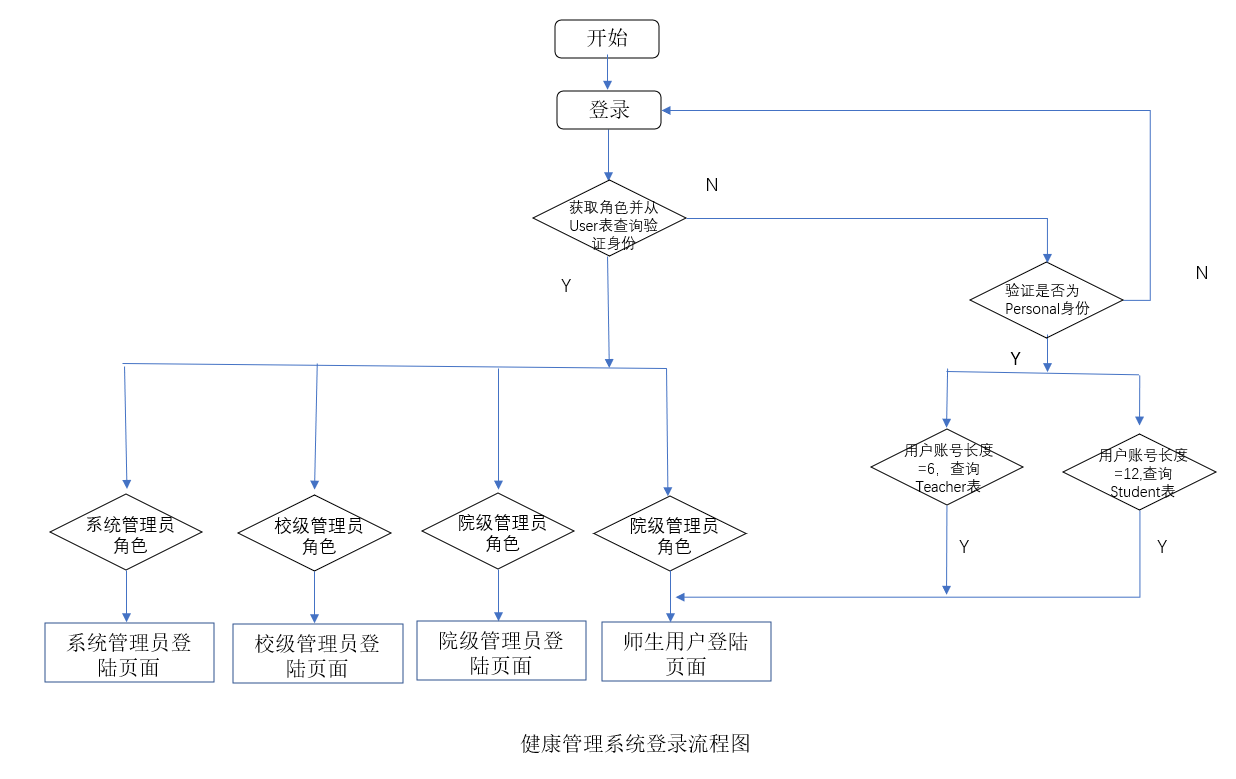
****

**第六章 系统实现功能**

**6.1 管理员登录模块**

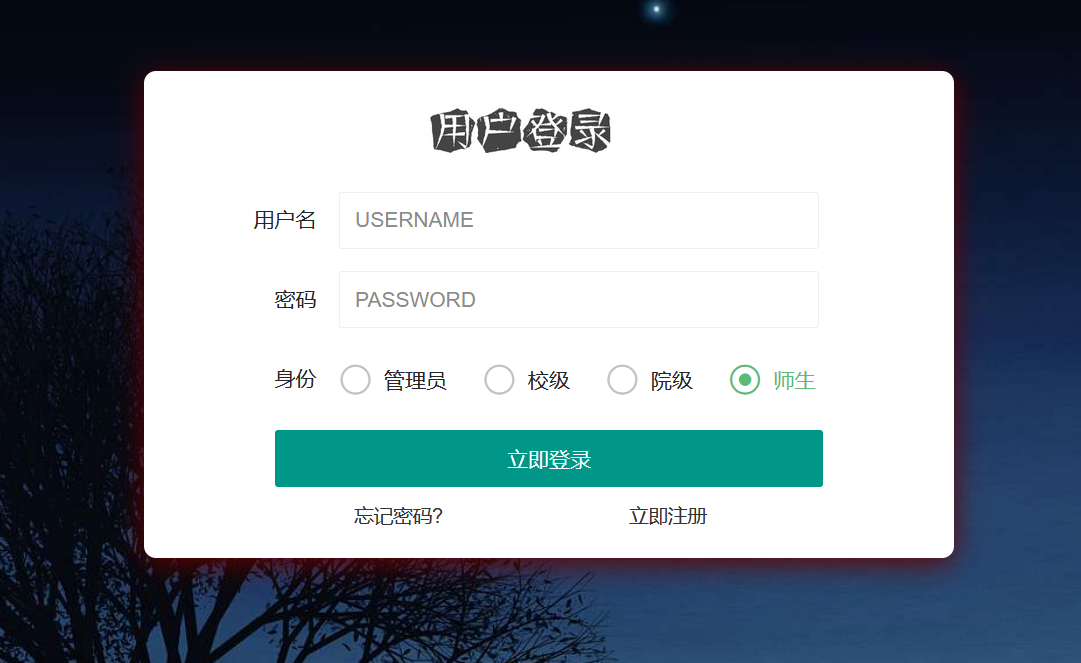
管理员登录模块提供用户登录的入口，在提交用户名和密码后进行账号密码验证，同时进行身份的验证，根据是系统管理员、校级管理员、院级管理员，跳转到对应的管理员系统。

**模块流程图：**

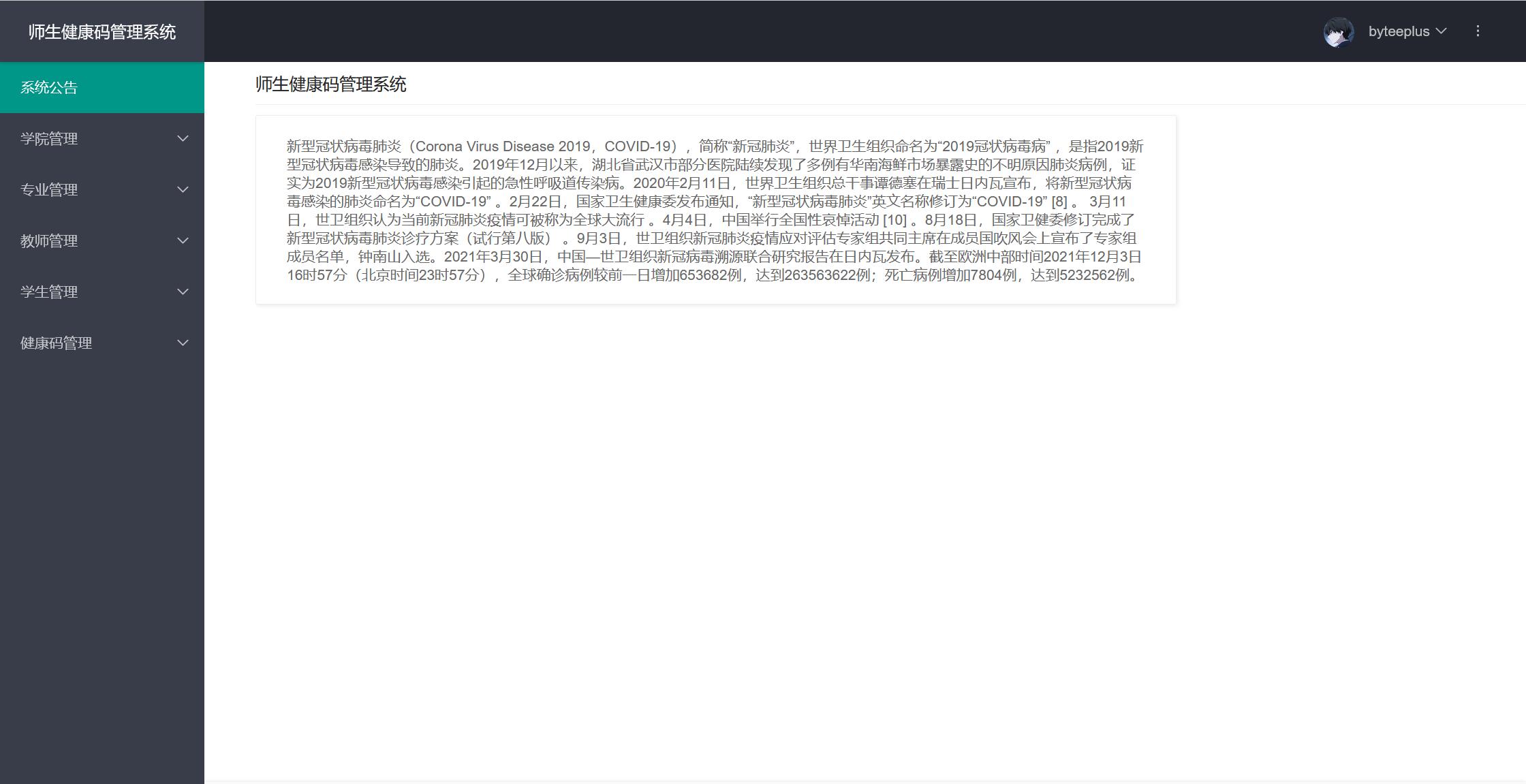


**演示截图**

主登录界面图下图所示：



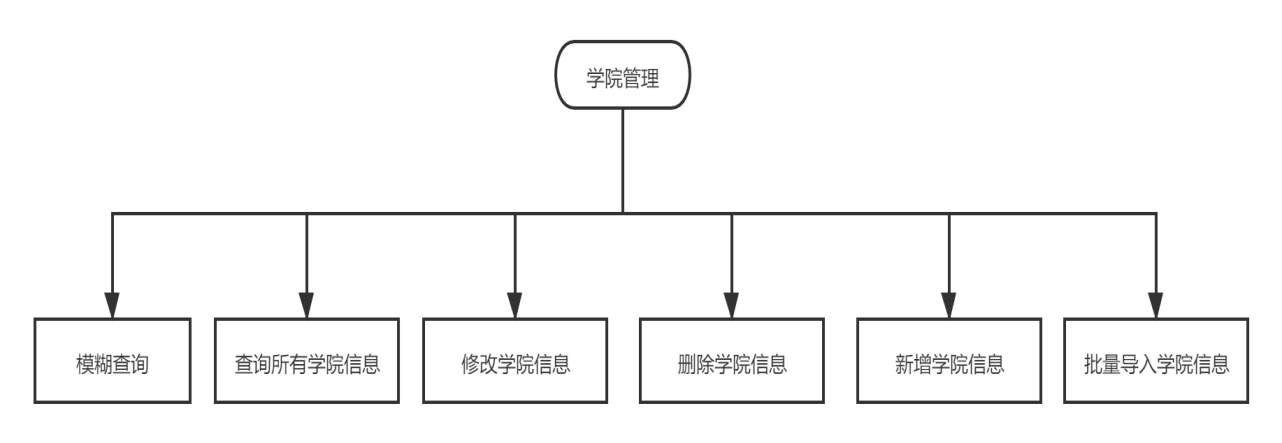
师生登录后界面如下图所示：



**6.2 学院管理模块**

学院管理模块用于管理学院编号、学院名称信息，其具体操作有逐条增加学院信息、批量增加学院信息、查询所有的学院信息、按照学院名称模糊查询学院信息、修改学院信息以及删除学院信息，以上所有的操作都与MySQL数据库建立了联系。

**模块流程图：**

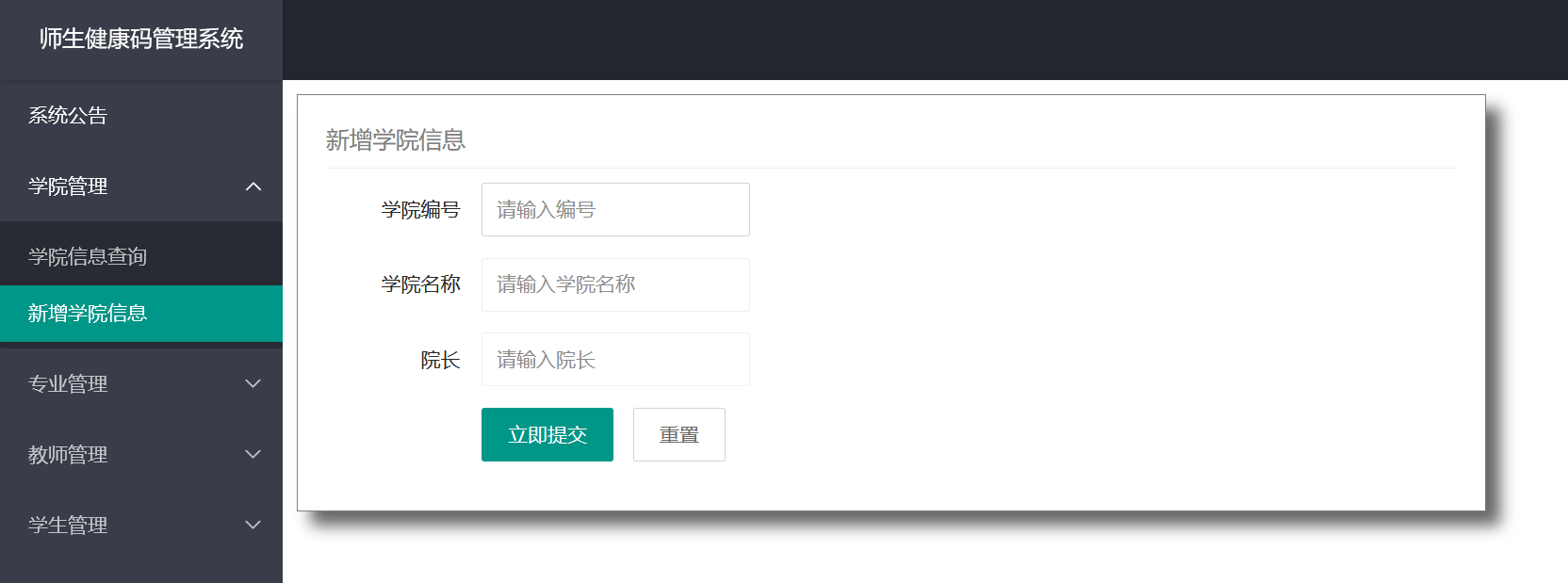
****

**演示截图**

学院基本信息表如下：



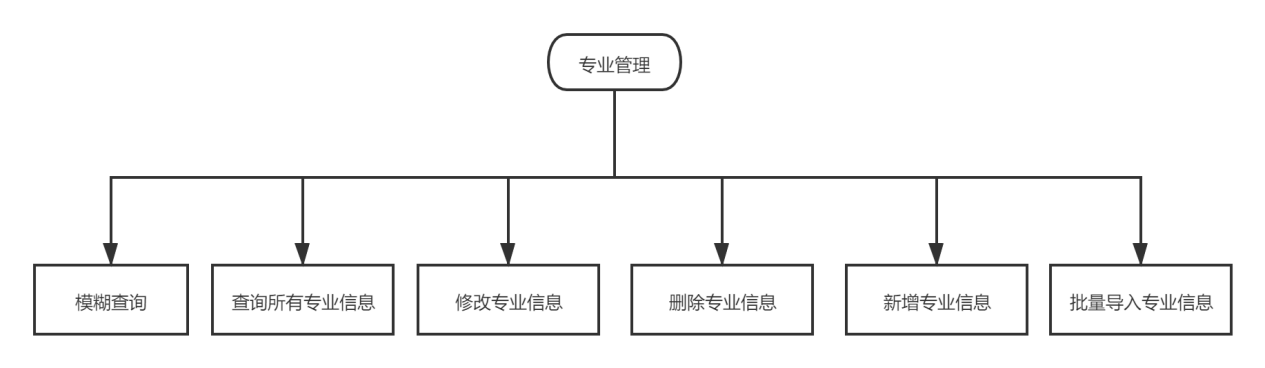
新增学院信息：



**6.3 专业管理模块**

专业管理模块用于管理专业编号、专业名称、专业所属学院信息，其具体操作有逐条增加专业信息、批量增加专业信息、查询所有的专业信息、按照专业名称模糊查询专业信息、按照学院编号模糊查询学院信息、修改专业信息以及删除专业信息。

**模块流程图：**

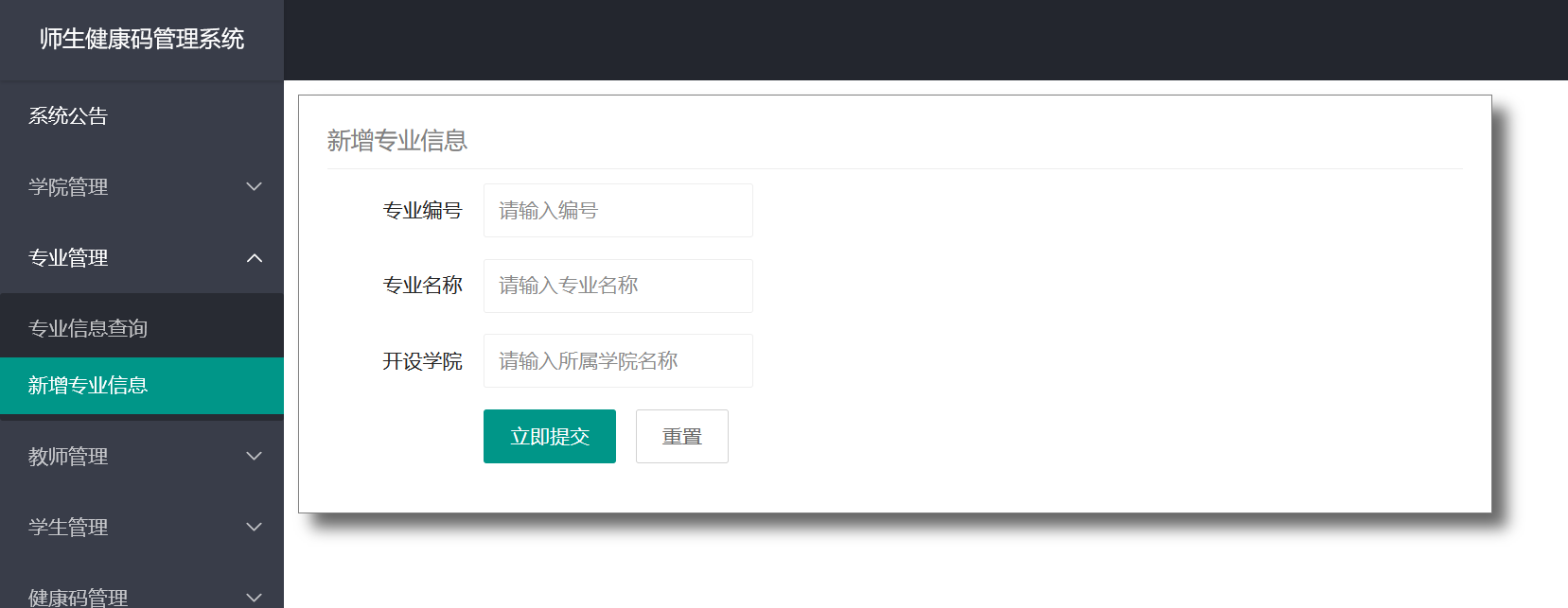


**演示截图**

专业基本信息表如下：



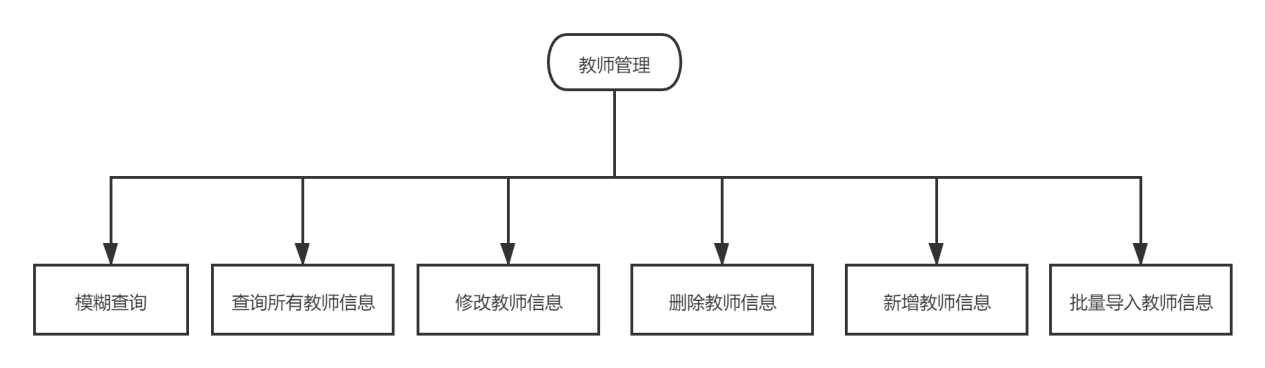
新增专业信息：



**6.4教师管理模块**

教师管理模块用于管理教师工号、教师姓名、教师身份证号、教师所属学院、教师健康码状态、教师电话号码，其具体操作有逐条增加教师信息、批量增加教师信息、查询所有的教师信息、按照教师工号模糊查询教师信息、按照教师姓名模糊查询教师信息、修改教师信息以及删除教师信息。

**模块流程图：**

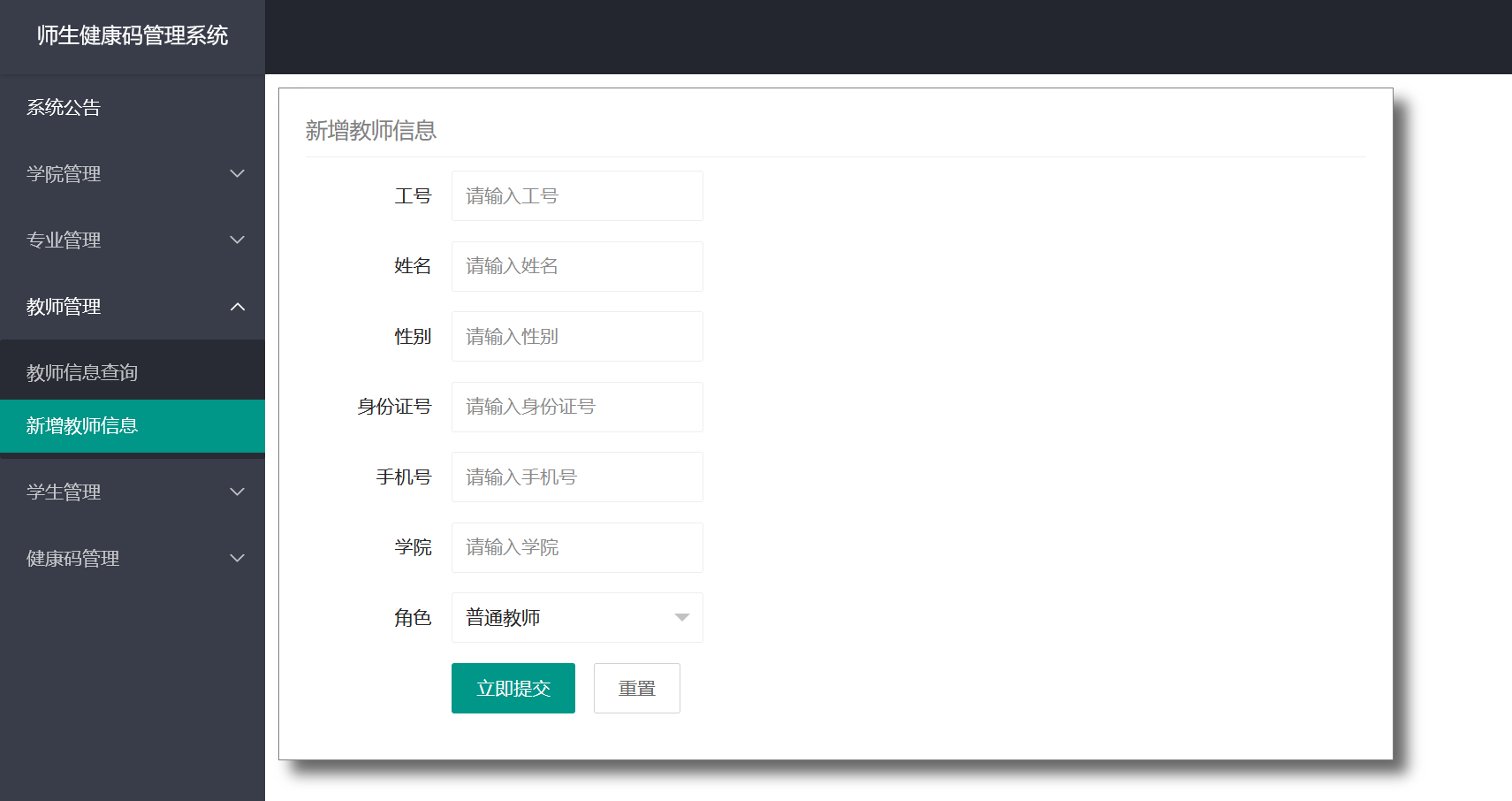


**演示截图**

教师基本信息表如下：



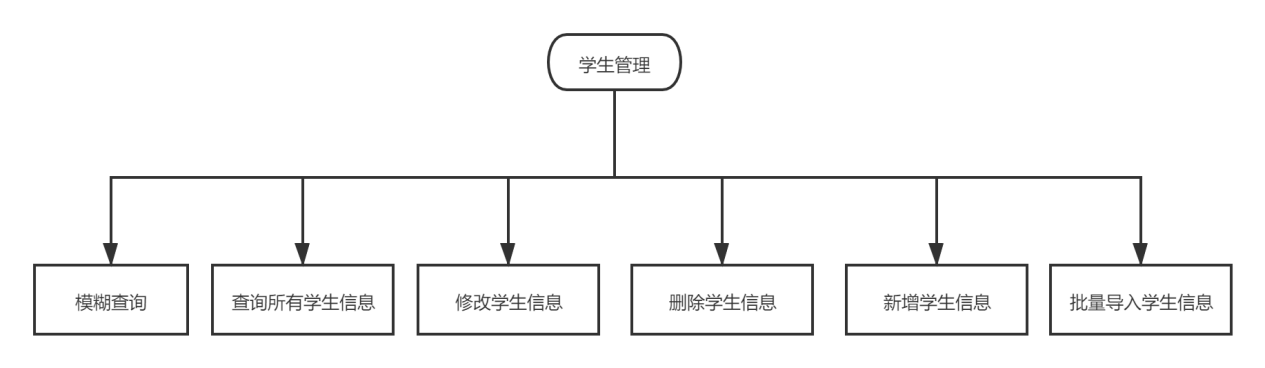
新增教师信息：



**6.5 学生管理模块**

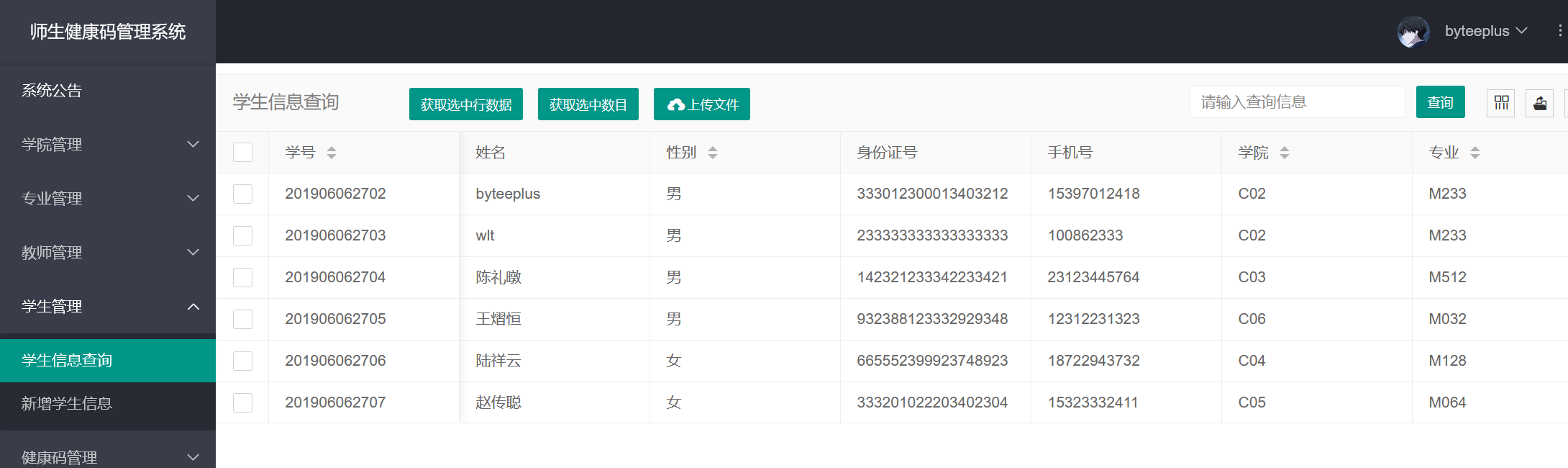
学生管理模块用于管理学生学号、学生姓名、学生身份证号、学生所属学院、学生所属专业、学生电话号码、学生健康码状态，其具体操作有逐条增加学生信息、批量增加学生信息、查询所有的学生信息、按照学生学号模糊查询学生信息、按照学生姓名模糊查询学生信息、修改学生信息以及删除学生信息。

**模块流程图：**

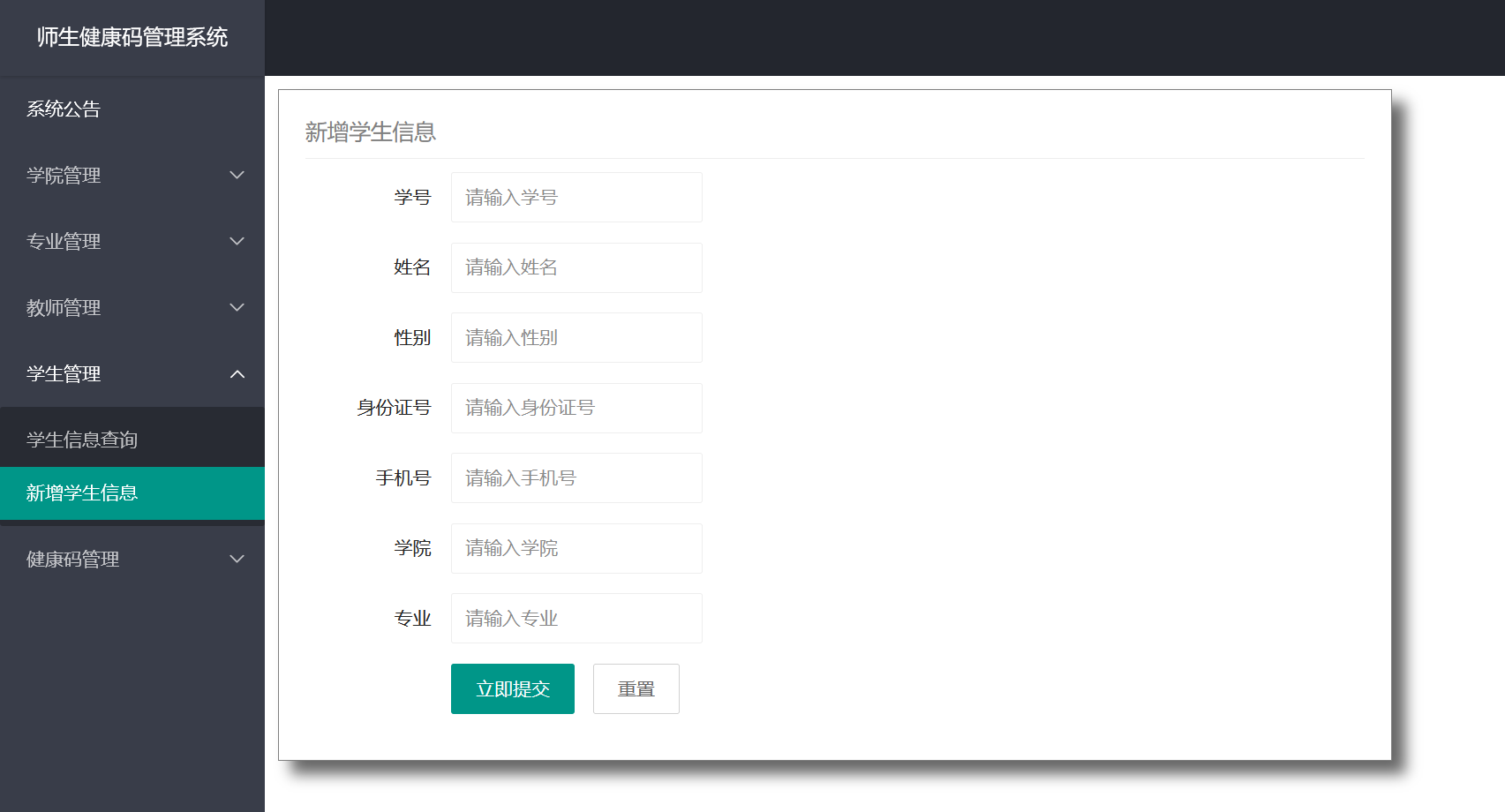
****

**演示截图**

学生基本信息表如下：



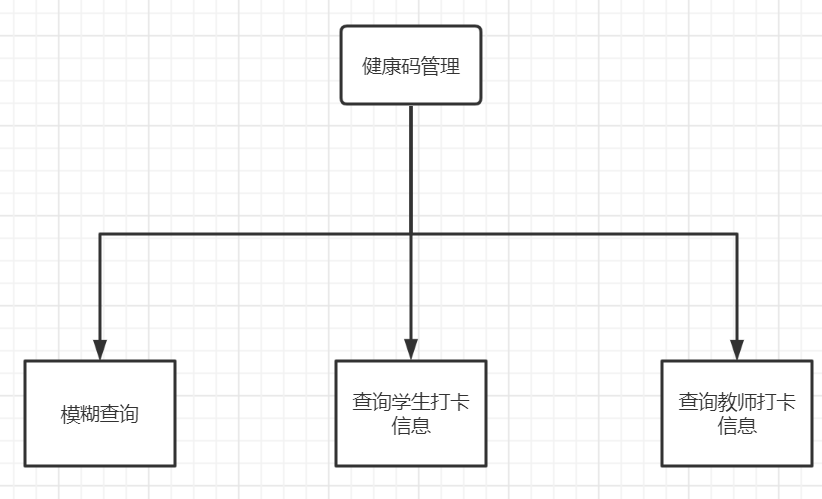
新增学生信息：



**6.6 健康码管理模块**

健康码管理模块用于查询师生打卡信息。其中师生打卡信息具体包括学号或工号、姓名、身份证号、手机号、身份、所属学院、打卡日期、打卡时间、健康码颜色、连续打卡绿码时间以及健康码目前状态信息。

**模块流程图：**

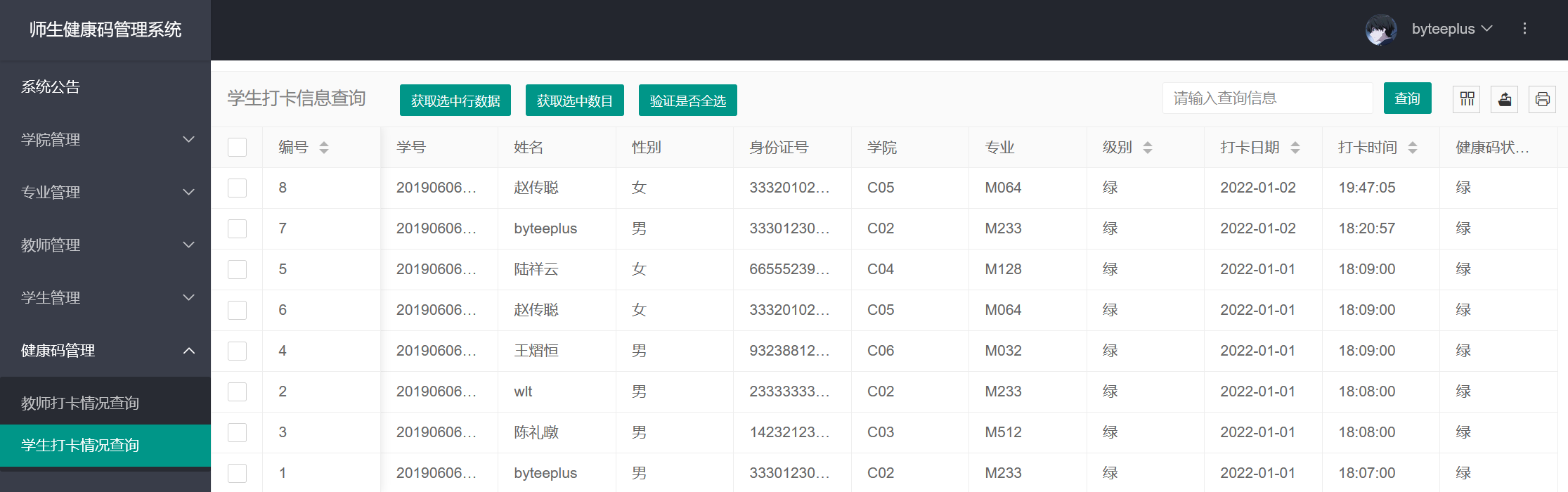


**演示截图**

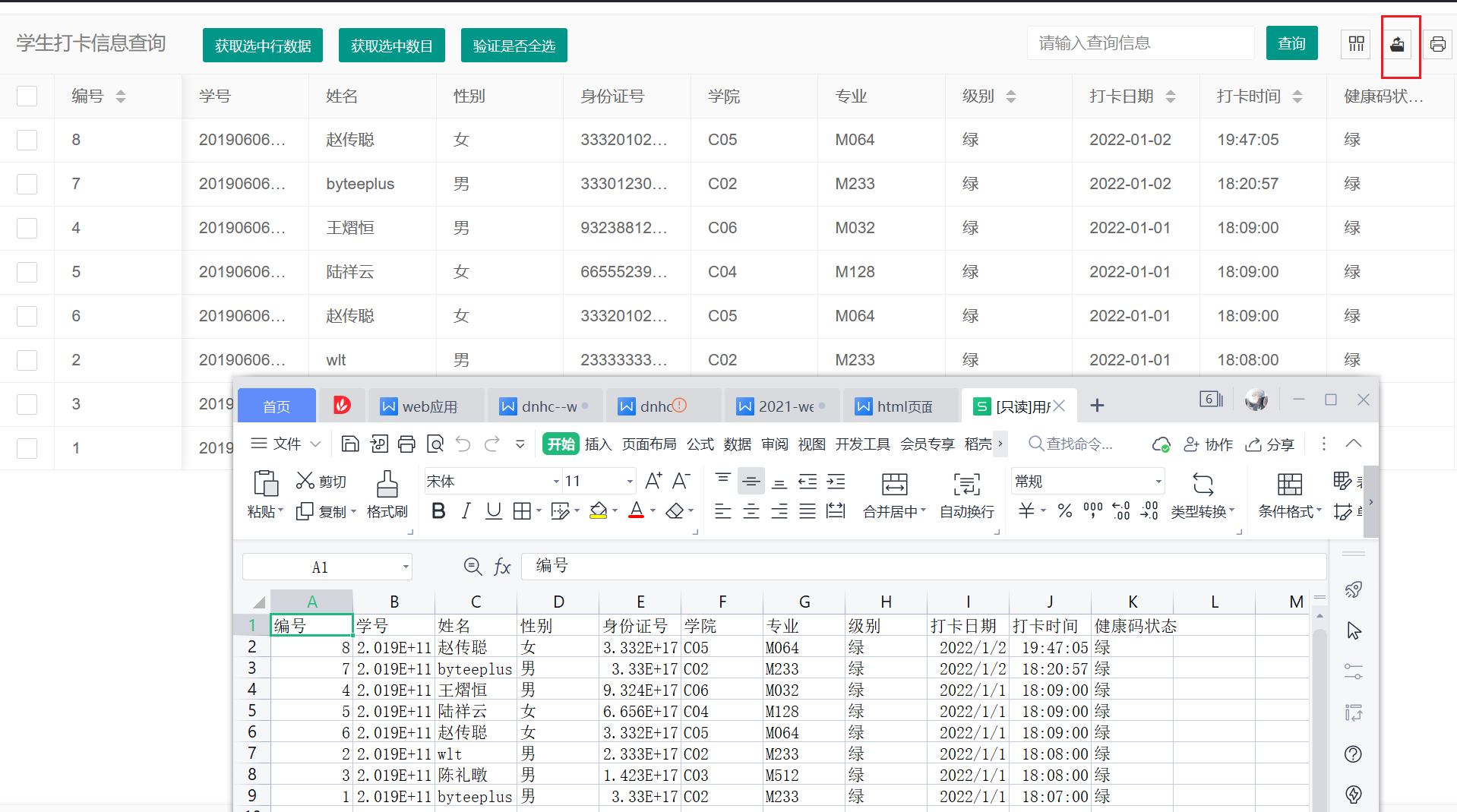
教师打卡基本信息表如下：



学生打卡基本信息表如下：



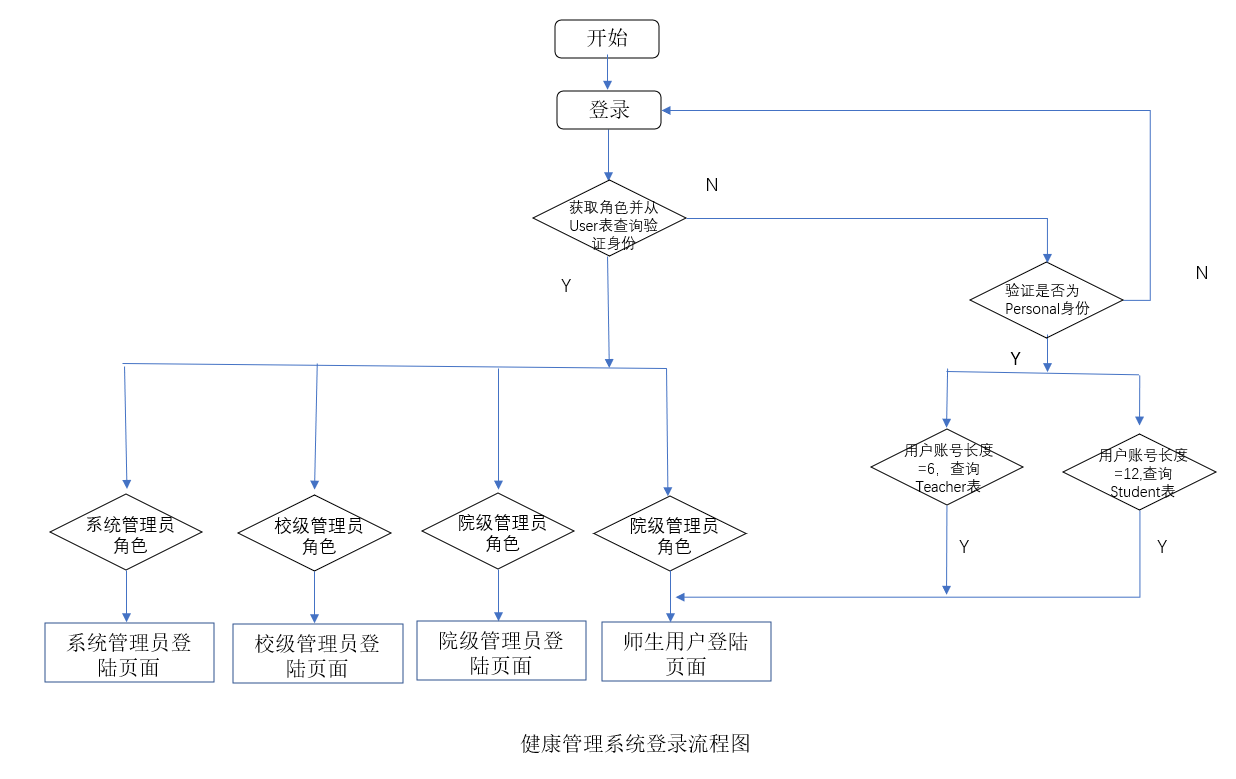
导出数据：



**6.7 师生登录模块**

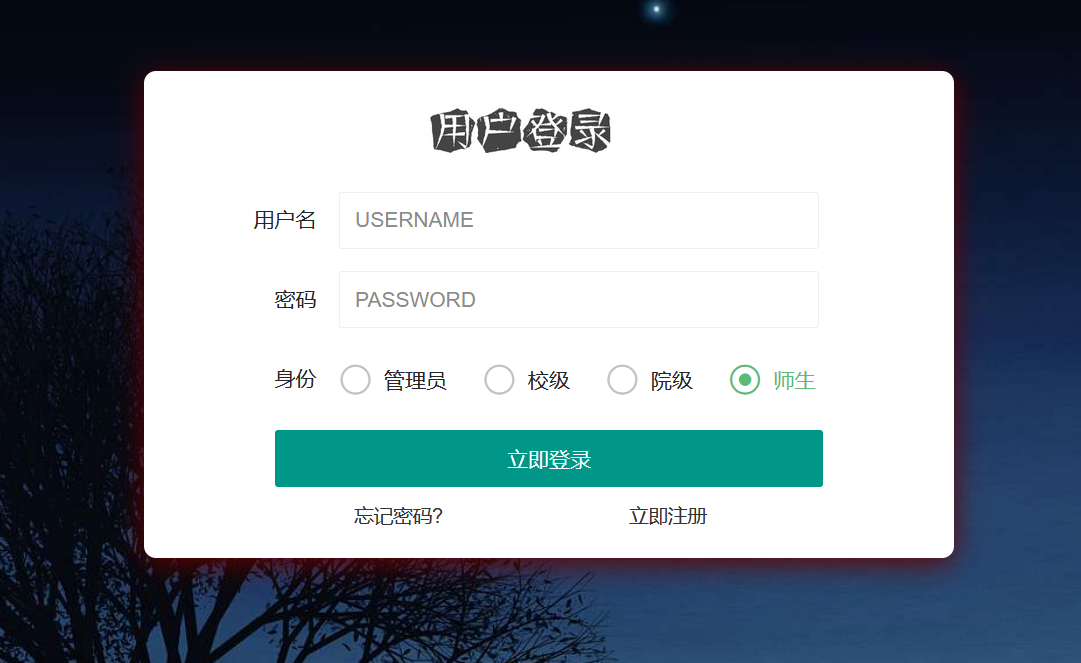
师生登录模块提供用户登录的入口，在提交后进行相关信息的验证，同时验证用户的身份，根据是教师还是学生，跳转到对应的用户系统。

**模块流程图：**

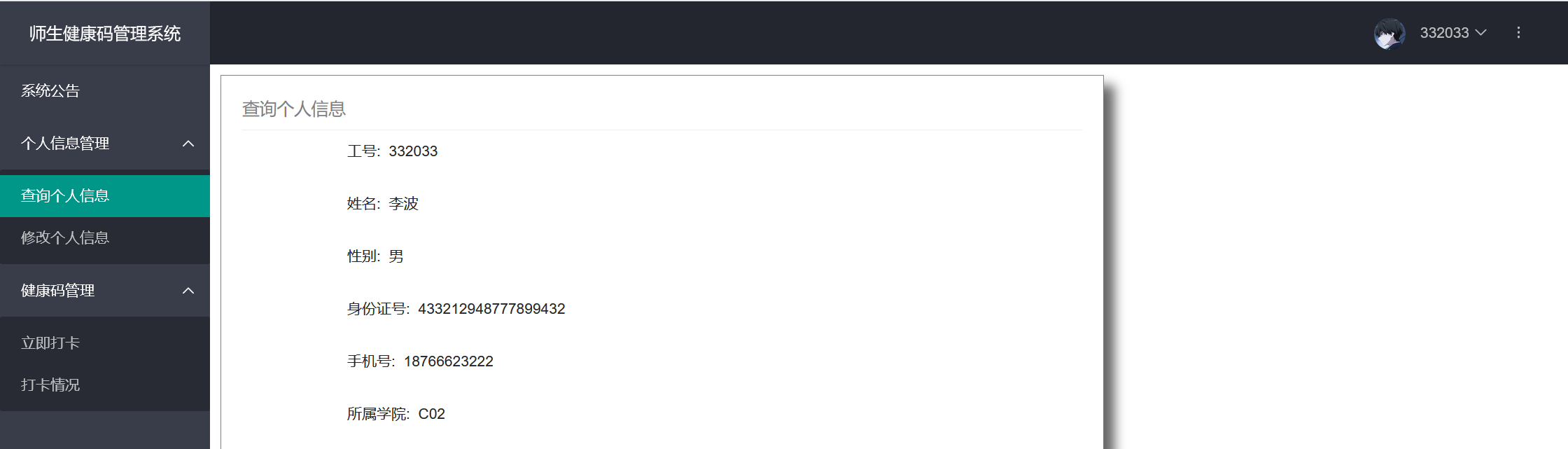


**演示截图**

主登录界面图下图所示：



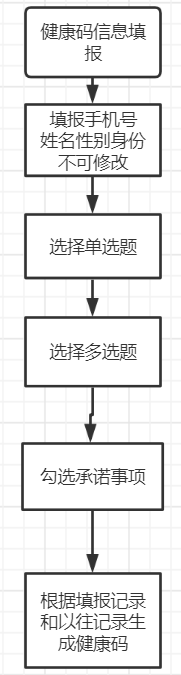
师生登录后界面如下图所示：



**6.8 健康码信息填报模块**

健康码管理模块用于查询师生打卡信息。其中师生打卡信息具体包括学号或工号、姓名、身份证号、手机号、身份、所属学院、打卡日期、打卡时间、健康码颜色、连续打卡绿码时间以及健康码目前状态信息。

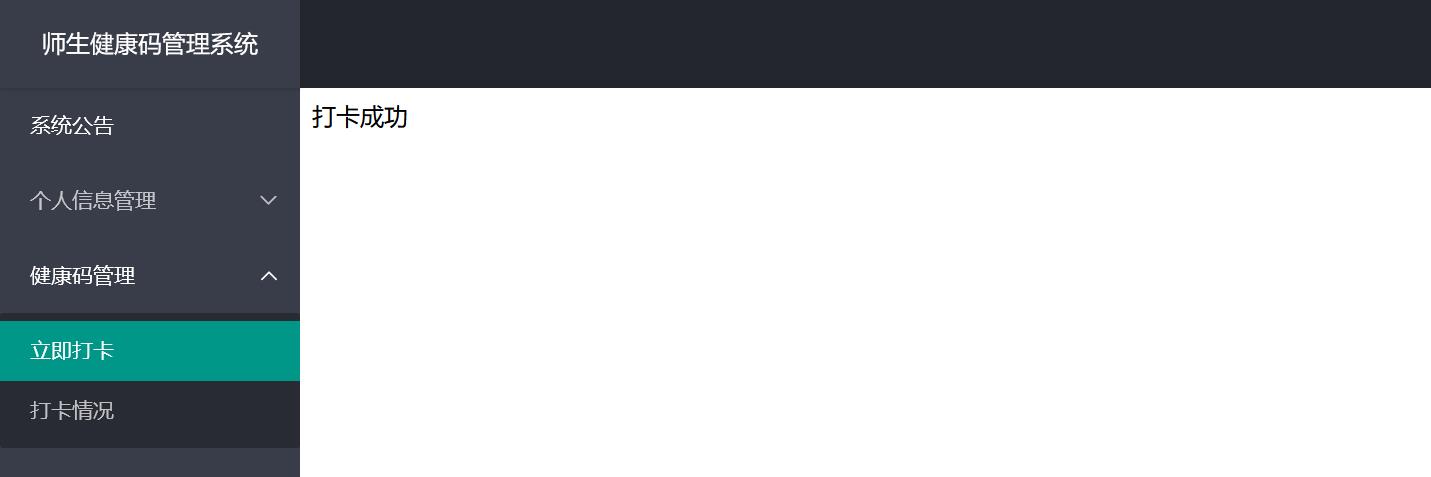
**模块流程图：**



**演示截图**

健康码信息填报页面如下图所示：





**6.9 健康码查询模块**

健康码查询模块用于查询师生打卡所生成的健康码，师生可以查看自己的专属二维码并且将其作为进出校园的凭证，绿码可以进出校园，黄码需居家观察7天不得进入校园，红码需居家医学观察或集中隔离14天不得进入校园。

**演示截图**

健康码展示页面如下图所示：

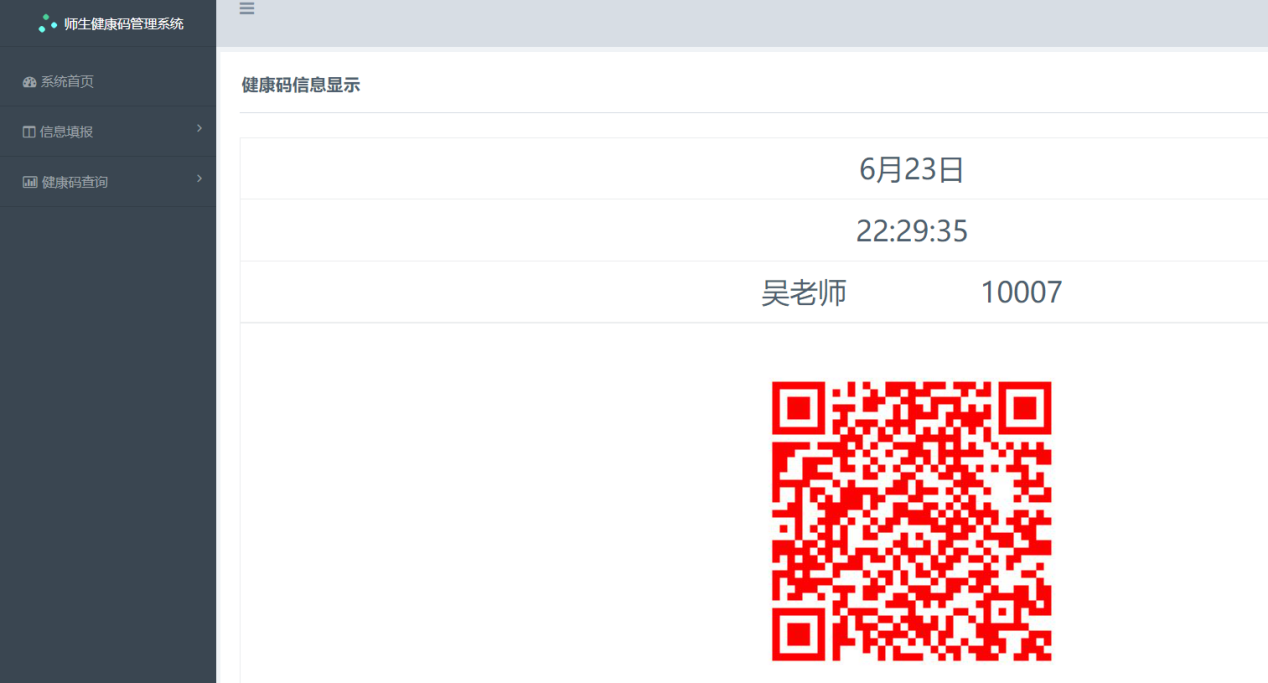
绿码：



黄码：



红码：



**第七章 程序清单**

index.jsp：管理员登录页面。

login.jsp：师生登录页面。

class\_insert.jsp：系统管理员新增班级信息的页面。

class\_management.jsp：系统管理员查看全部班级信息的页面。

class\_query.jsp：系统管理员模糊查询班级信息的页面。

class\_query\_insert.jsp：校级管理员新增班级信息的页面。

col\_class.jsp：院级管理员管理班级信息的页面。

col\_class\_query.jsp：院级管理员查看全部班级信息的页面。

col\_class\_query\_insert.jsp：院级管理员新增班级信息的页面。

col\_college\_query.jsp：院级管理员查看全部学院信息的页面。

col\_echarts.jsp：院级管理员管理echarts图表信息的页面。

col\_homepage.jsp：院级管理员主页面。

col\_major.jsp：院级管理员管理专业信息的页面。

col\_major\_query.jsp：院级管理员查看所有专业信息的页面。

col\_major\_query\_insert.jsp：院级管理员新增专业信息的页面。

col\_s\_healthcode.jsp：院级管理员查看本学院学生打卡信息的页面。

col\_s\_query.jsp：院级管理员查看本学院全部学生信息的页面。

col\_s\_query\_insert.jsp：院级管理员新增学生信息的页面。

col\_showStudentInformation.jsp：院级管理员模糊查询本院学生打卡信息的页面。

col\_showTeacherInformation.jsp：院级管理员模糊查询本院教师打卡信息的页面。

col\_student.jsp：院级管理员管理学生信息的页面。

col\_t\_healthcode.jsp：院级管理员查看本院教师打卡信息的页面。

col\_t\_query\_insert.jsp：院级管理员新增教师信息的页面。

col\_t\_query.jsp：院级管理员查询本学院全部教师的页面。

col\_teacher.jsp：院级管理员管理本学院教师信息的页面。

college\_insert.jsp：系统管理员新增学院信息页面。

college\_management.jsp：系统管理员查看全部学院信息的页面。

college\_query.jsp：校级管理员查询学习全部学院信息的页面。

college\_query\_insert.jsp：校级管理员新增学院信息的页面。

major\_insert.jsp：系统管理员新增专业信息的页面。

major\_management.jsp：系统管理员查询全校所有专业信息的页面。

major\_query.jsp：校级管理员查询全校所有专业信息的页面。

major\_query\_insert.jsp：校级管理员新增专业信息的页面。

s\_healthcode.jsp：系统管理员查询学生打卡信息的页面。

s\_information.jsp：学生健康码信息填报页面。

s\_information\_display.jsp：学生健康码信息展示页面。

s\_insert.jsp：系统管理员新增学生信息页面。

s\_management.jsp：系统管理员查询全校所有学生信息的页面。

s\_query.jsp：校级管理员查询全校所有学生信息的页面。

s\_query\_insert.jsp：校级管理员新增学生信息的页面。

sch\_class.jsp：校级管理员管理班级信息的页面。

sch\_college.jsp：校级管理员管理学院信息的页面。

sch\_echarts.jsp校级管理员管理echarts图表信息的页面。

sch\_homepage.jsp：校级管理员主页面。

sch\_major.jsp：校级管理员管理专业信息的页面。

sch\_s\_healthcode.jsp：校级管理员查询学生打卡信息的页面。

sch\_showecharts.jsp：校级管理员查询各个学院echarts图表信息的页面。

sch\_showStudentInformation.jsp：校级管理员查询学生打卡信息页面。

sch\_showTeacherInformation.jsp：校级管理员查询教师打卡信息页面。

sch\_student.jsp：校级管理员管理学生信息页面。

sch\_t\_healthcode.jsp：校级管理员查询教师打卡信息的页面。

sch\_teacher.jsp：校级管理员管理教师信息页面。

show\_class\_query.jsp：校级管理员查询班级信息页面。

show\_col\_class\_query.jsp：院级管理员查询班级信息页面。

show\_col\_major\_query.jsp：院级管理员查询专业信息页面。

show\_col\_s\_query.jsp：院级管理员查询学生信息页面。

show\_col\_t\_query.jsp：院级管理员查询教师信息页面。

show\_major\_query.jsp：校级管理员查询专业信息页面。

show\_college\_query.jsp：校级管理员查询专业信息页面。

show\_s\_query.jsp：校级管理员查询学生信息页面。

show\_t\_query.jsp：校级管理员查询教师信息页面。

showClassQuery.jsp：系统管理员查询班级信息页面。

showCollegeQuery.jsp：系统管理员查询学院信息页面。

showMajorQuery.jsp：系统管理员查询专业信息页面。

showStudentInformation.jsp：系统管理员查询学生打卡信息页面。

showStudentQuery.jsp：系统管理员查询学生信息页面。

showTeacherInformation.jsp：系统管理员查询教师打卡信息页面。

showTeacherQuery.jsp：管理员查询教师信息页面。

stu\_homepage.jsp：学生登录主页面。

sys\_class.jsp：系统管理员管理班级信息页面。

sys\_college.jsp：系统管理员管理学院信息页面。

sys\_echarts.jsp：系统管理员管理echarts图表信息页面。

sys\_homepage.jsp：系统管理员主页面。

sys\_major.jsp：系统管理员管理专业信息页面。

sys\_showecharts.jsp：系统管理员查询echarts图表信息页面。

sys\_student.jsp：系统管理员管理学生信息页面。

sys\_teacher.jsp：系统管理员管理教师信息页面。

t\_healthcode.jsp：系统管理员查询教师打卡信息页面。

t\_information.jsp：教师健康码信息填报页面。

t\_information\_display.jsp：教师健康码信息展示页面。

t\_insert.jsp：系统管理员新增教师页面。

t\_management.jsp：系统管理员查询所有教师信息页面。

t\_query.jsp：校级管理员查询所有教师信息页面。

t\_query\_insert.jsp：校级管理员新增教师页面。

tea\_homepage.jsp：教师登录主页面。

col\_college\_query\_deleteServlet：删除学院信息。

col\_college\_query\_updateServlet：更新学院信息。

college\_deleteServlet：删除学院信息。

college\_insertServlet：新增学院信息。

college\_query\_deleteServlet：删除学院信息。

college\_query\_insertServlet：新增学院信息。

college\_query\_updateServlet：更新学院信息。

college\_queryServlet：查询学院信息。

college\_updateServlet：更新学院信息。

sch\_college\_import：批量导入学院信息。

sch\_college\_queryServlet：查询学院信息。

sys\_college\_import：批量导入学院信息。

col\_major\_import：批量导入专业信息。

major\_insertServlet：新增专业信息。

major\_query\_updateServlet：更新专业信息。

col\_major\_queryServlet：查询专业信息。

col\_major\_query\_deleteServlet：删除专业信息。

major\_queryServlet：查询专业信息。

col\_major\_query\_updateServlet：更新专业信息。

major\_query\_insertServlet：新增专业信息。

major\_deleteServlet：删除专业信息。

col\_major\_query\_insertServlet：新增专业信息。

major\_query\_deleteServlet：删除专业信息。

major\_updateServlet：更新专业信息。

sch\_major\_import：批量导入专业信息。

sch\_major\_queryServlet：查询专业信息。

sys\_major\_import：批量导入专业信息。

loginServlet：管理员登录控制。

s\_informationServlet：学生健康码生成处理。

sch\_echartsServlet：echarts图表信息处理。

sys\_echartsServlet：echarts图表信息处理。

t\_informationServlet：教师健康码生成处理。

ts\_loginServlet：师生登录控制。

class\_deleteServlet：删除班级信息。

class\_insertServlet：新增班级信息。

class\_query\_deleteServlet：删除班级信息。

class\_query\_insertServlet：新增班级信息。

class\_query\_updateServlet：更新班级信息。

class\_queryServlet：查询班级信息。

class\_updateServlet：更新班级信息。

col\_class\_import：批量导入班级信息。

col\_class\_query\_deleteServlet：删除班级信息。

col\_class\_query\_insertServlet：新增班级信息。

col\_class\_query\_updateServlet：更新班级信息。

col\_class\_queryServlet：查询班级信息。

sch\_class\_import：批量导入班级信息。

sch\_class\_queryServlet：查询班级信息。

sys\_class\_import：批量导入班级信息。

s\_query\_deleteServlet：删除学生信息。

col\_s\_query\_deleteServlet：删除学生信息。

s\_insertServlet：新增学生信息。

s\_deleteServlet：删除学生信息。

col\_s\_queryServlet：查询学生信息。

col\_s\_healthcodeServlet：处理学生打卡信息。

s\_healthcodeServlet：处理学生打卡信息。

s\_query\_updateServlet：更新学生信息。

col\_s\_query\_updateServlet：更新学生信息。

s\_updateServlet：更新学生信息。

s\_query\_insertServlet：新增学生信息。

col\_student\_import：批量导入学生信息。

s\_queryServlet：查询学生信息。

sch\_s\_healthcodeServlet：处理学生打卡信息。

col\_s\_insertServlet：新增学生信息。

sch\_s\_queryServlet：查询学生信息。

sch\_student\_import：批量导入学生信息。

sys\_student\_import：批量导入学生信息。

t\_healthcodeServlet：处理教师打卡信息。

col\_t\_healthcodeServlet：处理教师打卡信息。

t\_query\_deleteServlet：删除教师信息。

sch\_t\_healthcodeServlet：处理教师打卡信息。

col\_t\_queryServlet：查询教师信息。

sch\_teacher\_import：批量导入教师信息。

t\_deleteServlet：删除教师信息。

col\_t\_query\_deleteServlet：删除教师信息。

sys\_teacher\_import：批量导入教师信息。

t\_insertServlet：新增教师信息。

col\_teacher\_import：批量导入教师信息。

col\_t\_query\_updateServlet：更新教师信息。

sch\_t\_queryServlet：查询教师信息。

col\_t\_query\_insertServlet：新增教师信息。

t\_query\_insertServlet：新增教师信息。

t\_queryServlet：查询教师信息。

t\_query\_updateServlet：更新教师信息。

t\_updateServlet：更新教师信息。

BaseDao：基础的Dao类，提供数据源返回和连接返回方法。

ClassDao：提供从数据库获取班级信息的方法。

CollegeDao：提供从数据库获取学院信息的方法。

MajorDao：提供从数据库获取专业信息的方法。

S\_informationDao：提供从数据库获取学生打卡信息的方法。

StudentDao：提供从数据库获取学生信息的方法。

T\_informationDao：提供从数据库获取教师打卡信息的方法。

TeacherDao：提供从数据库获取教师信息的方法。

College：存储班级信息的JavaBean。

Major：存储专业信息的JavaBean。

S\_information：存储学生打卡信息的JavaBean。

sClass：存储班级信息的JavaBean。

Student：存储学生信息的JavaBean。

T\_information：存储教师打卡信息的JavaBean。