

1

**课程综合设计**

**高校疫情防控的分析与实现**

|  |  |
| --- | --- |
| **二级学院** | **医药信息工程学院** |
| **专 业** | **信息管理与信息系统** |
| **班 级** | **2019级** |
| **学生姓名** | **李慧婷** |
| **学 号** | **1920501123** |
| **指导教师** | **陈伟** |
|  |  |
|  |  |
| 2022年10月 | |

**高校疫情系统的分析与实现**

**【摘要】**随着互联网的快速发展，逐渐进入信息化、科技化社会，对互联网的依赖也越来越大。根据计算机专业知识，我们将了解软件开发概念和整个项目的流程，并参考爬取一些疫情数据来完成疫情管理系统。

本疫情管理系统主要分为用户功能模块和管理员功能模块，在设计上首先描述了疫情管理系统的开发，对该系统进行了更详细的需求分析，并研究了疫情管理系统的功能需求，业务流程，系统结构和数据库设计等问题。

该系统前台页面使用html，前端的框架主要是使用LayUI来进行完成，后端框架使用SpringBoot和MybatisPlus技术。SpringBoot基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

根据系统所需功能和使用习惯，决定以IntelliJIDEA为开发工具，基于SpringBoot框架，采用MySQL做后台数据库，本系统界面友好，而且操作更加简单，减少了用户学习操作的额外负担。

**【关键词】**疫情管理系统，SpringBoot，MySQL数据库，LayUI

目录

[**1 前言** 1](#_Toc118842109)

[**2绪论** 1](#_Toc118842110)

[**2.1研究背景** 1](#_Toc118842111)

[**2.2研究意义** 1](#_Toc118842112)

[**2.3研究内容** 2](#_Toc118842113)

[**3关键技术概述** 2](#_Toc118842114)

[**3.1前端开发技术** 2](#_Toc118842115)

[3.1.1 LayUI 2](#_Toc118842116)

[3.1.2 ECharts 2](#_Toc118842117)

[3.1.3 Ajax 3](#_Toc118842118)

[**3.2后端开发技术** 3](#_Toc118842119)

[3.2.1 SpringBoot 3](#_Toc118842120)

[3.2.2 MyBatis-Plus 3](#_Toc118842121)

[3.2.3 MySQL 3](#_Toc118842122)

[3.2.4 Shiro 4](#_Toc118842123)

[3.2.5 HttpClient爬虫 4](#_Toc118842124)

[**4需求分析** 4](#_Toc118842125)

[4.1设计内容 4](#_Toc118842126)

[4.2用户角色用例图 5](#_Toc118842127)

[4.2.1 总体用例图设计 5](#_Toc118842128)

[4.2.2 管理员用例图设计 6](#_Toc118842129)

[4.2.3 教师用例图设计 6](#_Toc118842130)

[4.2.4学生用例图设计 6](#_Toc118842131)

[4.3系统模块功能性需求 7](#_Toc118842132)

[4.3.1 普通用户模块 7](#_Toc118842133)

[4.3.2 疫情数据可视化模块 8](#_Toc118842134)

[4.3.3 疫情数据管理模块 8](#_Toc118842135)

[4.3.4 校园防疫管理模块 9](#_Toc118842136)

[4.3.5 系统管理模块 10](#_Toc118842137)

[4.4系统非功能性需求 11](#_Toc118842138)

[4.4.1 安全性 11](#_Toc118842139)

[4.4.2 高效性 11](#_Toc118842140)

[4.4.3 可扩展性 11](#_Toc118842141)

[4.4.4 易用性 12](#_Toc118842142)

[4.4.5 一致性 12](#_Toc118842143)

[4.4.6 时效性 12](#_Toc118842144)

[**5系统详细分析** 12](#_Toc118842145)

[5.1设计方法 12](#_Toc118842146)

[5.2系统模块详细设计 13](#_Toc118842147)

[5.2.1 用户模块 13](#_Toc118842148)

[5.2.2 疫情数据可视化模块 15](#_Toc118842149)

[5.2.3 疫情数据管理模块 16](#_Toc118842150)

[5.2.4 校园防疫管理模块 16](#_Toc118842151)

[5.2.5 系统管理模块 18](#_Toc118842152)

[5.3数据库设计 19](#_Toc118842153)

[5.3.1 概念模型设计 19](#_Toc118842154)

[5.3.2 数据库表设计 20](#_Toc118842155)

[**6系统实现** 25](#_Toc118842156)

[6.1配置文件代码 25](#_Toc118842157)

[6.2用户模块 26](#_Toc118842158)

[6.2.1 登录功能 26](#_Toc118842159)

[6.2.1 登录界面 28](#_Toc118842160)

[6.3疫情数据可视化模块 28](#_Toc118842161)

6.3.1 中国疫情数据可视化……………………………………28

[6.3.2 中国疫情数据可视化页面 30](#_Toc118842162)

[6.3疫情数据可视化模块 31](#_Toc118842163)

[6.3.1 查询疫情数据功能 31](#_Toc118842164)

[6.3.2 查询疫情数据页面 32](#_Toc118842165)

[6.3.3 导出Excel疫情数据功能 32](#_Toc118842166)

[6.4校园防疫管理 34](#_Toc118842167)

[6.4.1 健康打卡记录功能 34](#_Toc118842168)

[6.4.2 健康打卡记录页面 35](#_Toc118842169)

[6.5校园防疫管理 35](#_Toc118842170)

[6.8.1 分配后台功能权限功能 35](#_Toc118842171)

[6.8.2 分配后台功能权限页面 37](#_Toc118842172)

**7总结与展望** 38

**1 前言**

2020年新冠疫情席卷全国，在党中央的决策领导下全国人民战胜疫情，学生开始复课，但是新冠病毒还有传播的可能，校园疫情防控不可大意。因此开发一款疫情防控具有深远意义，将疫情防控信息数字化，信息化能够增加我们对疫情的防控能力，将危险遏制在萌芽之中。

**2绪论**

**2.1研究背景**

在2020这个让人期待的一年，发生了一件让人落泪的事，新的疫情席卷个了整个中国，紧随着疫情防控数据的管理系统就迫切需要建立。

防疫信息的发布和查询不受时间和地点的限制，即使用户注册，也可以进行搜索和管理操作，操作简单，对用户来说非常方便。 随着Internet产品的出现，与校园防灾相关的用户数量不足，因此，在当下疫情的常态化下，有关疫情管理的系统的出现是很有必要的。

该基于SpringBoot开发的疫情管理系统，同时使用MySQL数据库进行对数据的储存，整个项目界面美观，性能优越，能够很好的满足疫情人员管理的需求。

**2.2研究意义**

随着Internet的快速发展，我们已经进入了信息化，科学发展的社会，对Internet的依赖性日益增强。因此，我们在疫情期间开发了基于Java的疫情管理系统。通过在流行期间使用基于Java的疫情管理系统，用户能够实时了解到疫情的动态信息。管理员对系统具有完全控制权，这对用户非常方便。系统用户分工明确，充分保证了系统的正常运行。

该疫情管理系统具有以下实际意义。

1.用户可以随时查询。没有时间和地点限制。

2.系统管理员可以随时浏览以获得更直观的疫情数据。

3.改变了疫情预防体系的方法，更方便，更灵活。

4.我们实现了学校建设防疫情控制系统的信息化系统。

**2.3研究内容**

该高校疫情防控系统系统角色分别为学生、教师和管理员。参考了我国一些较知名的疫情数据查看系统，确定了本系统的主要框架和功能，完成需求分析工作。系统主要分为疫情数据可视化、疫情数据管理、校园防疫管理、系统管理四大模块，其中系统管理模块中包括菜单管理、角色管理和用户管理。根据需求分析完成对数据库表的设计。

**3关键技术概述**

在设计和开发该系统中，会涉及到很多的核心技术，包括前端和后端技术。

**3.1前端开发技术**

**3.1.1 LayUI**

LayUI 是一款采用自身模块规范的前端UI框架，遵循原生的HTML/CSS的书写与组织形式，属于轻量级框架，简单美观。主要是提供很多好看、方便的样式和Bookstrap有些相似，但该框架有个极大的好处就是定义了很多前后端交互的样式接口，如分页表格，只需在前端配置好接口，后端则按照定义好的接口规则返回数据，即可完成页面的展示，极大减少了后端人员的开发成本。

**3.1.2 ECharts**

ECharts是一款基于JavaScript的数据可视化的图表库，提供直观，生动，可交互，可个性化定制的数据可视化图表，它提供了常规的折线图、散点图、饼图、折线图，用于地理数据可视化的地图。能够流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），底层依赖矢量图形库 ZRender，提供直观，交互丰富，可高度个性化定制的数据可视化图表。

**3.1.3 Ajax**

Ajax即Asynchronous Javascript And XML（异步JavaScript和XML），使用Ajax技术网页能够快速地将增量更新在用户界面上，而不需要重载整个页面，这使得程序能够更快地回应用户的操作。Ajax不需要任何浏览器插件，但需要用户允许JavaScript在浏览器上执行。

**3.2后端开发技术**

#### 3.2.1 SpringBoot

SpringBoot基于Spring4.0设计，不仅继承了原有的优秀特性，SpringBoot整合了Spring 家族所有的框架，包括Spring-Web、SpringDataJPA 等优秀的框架，而且通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决，大大的减少了搭建环境的时间代价，节省了开发者的时间，使开发者有更多的时间和精力放到思考系统功能等更有价值的事情上来。

#### 3.2.2 MyBatis-Plus

MyBatis-Plus是一款非常有力量的 MyBatis加强版，它仅提供了加强，没有任何改动。零 SQL声明可以非常容易地实现基础的数据库运算，例如简单、批量和分页。MyBatis-Plus的加强版，实际上是对 MyBatis进行了自我包装和扩展，用户无需编写 XML文档，只要进行一些简单的设置即可。可以迅速完成 CRUD的基础运算，节省编程人员的大部分时间。

#### 3.2.3 MySQL

JDBC驱动程序不同于数据库，并且每个数据库中的应用程序都是单独打开的。当运行到程序中时，可以对大量数据执行多种查询和应用程序，对传统数据进行多种查询，而无需大量存储和理解，并且最重要的是，记录每个集合是主要操作无需合并层次结构，并且在使用精神上活跃且功能强大的MySQL数据库时，编写代码即可实现其他功能。如果您使用MySQL数据库，则只需创建一小段即可实现此功能。

#### Shiro

Shiro是Apache提供的一个安全框架，Shiro提供了认证、授权、加密和会话管理等功能，对于任何一个应用程序，Shiro 都可以提供全面的安全管理服务。Spring中有Spring Security，是一个权限框架，它和Spring依赖过于紧密，没有Shiro使用简单。 Shiro不依赖于Spring，Shiro不仅可以实现 Web应用的权限管理，还可以实现C/S系统，分布式系统权限管理，Shiro属于轻量框架，越来越多企业项目开始使用Shiro。

#### HttpClient爬虫

HttpClient 是 Apache 下的一个开源项目，用来提供高效的、最新的、功能丰富的支持 HTTP 协议的客户端编程工具包，并且它支持 HTTP 协议最新的版本和建议。它封装了java中URLConnection工具，并且将URLConnection的方法使用，都做了优化，更好的兼容了Http的访问和使用，更好的管理了缓存和Cookies，并且兼容了所有的Http访问的方法，使得，所有的网站，都可以使用这个工具来访问。

**4需求分析**

## 4.1设计内容

高校疫情防控系统主要分为管理员、教师和学生三种用户。管理员被赋予了系统的所有权限，可以是对疫情数据进行实时查看，对中国的疫情数据进行管理；教师具有对系统的增删改查功能，可以查看疫情数据，同时也可以对疫情数据进行管理；学生可以查看相关疫情信息，每天进行健康打卡。根据系统和业务的分析，将系统分为四大模块，具体模块划分如图4-1所示。

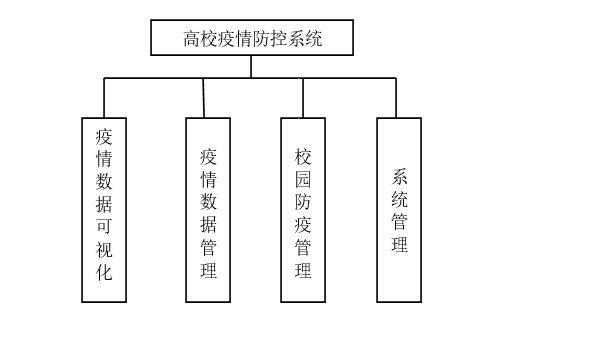


图4-1 功能模块图

## 4.2用户角色用例图

#### 4.2.1 总体用例图设计

总体用例图如图4-2所示

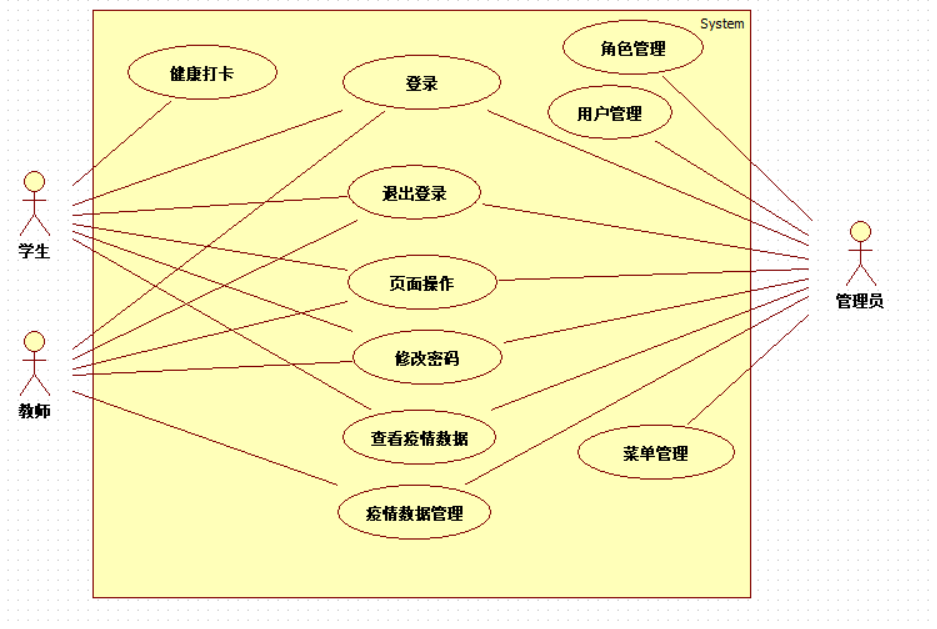


图4-2 总体用例图

#### 4.2.2 管理员用例图设计

管理员用例图如图4-3所示

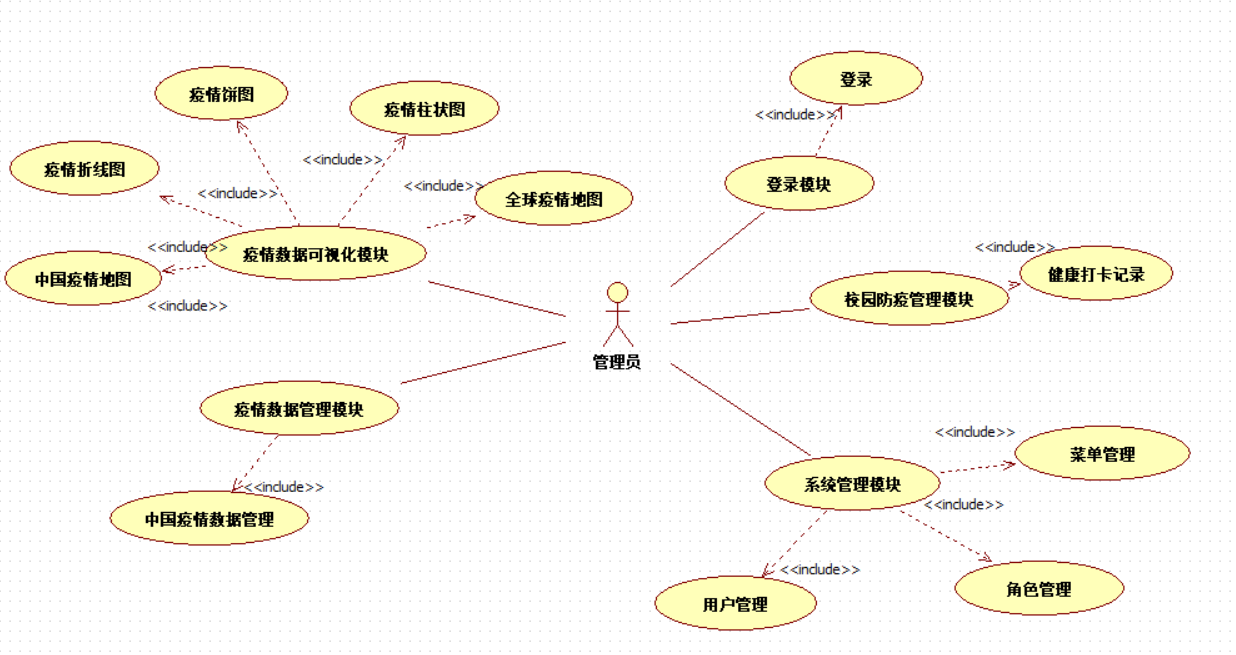
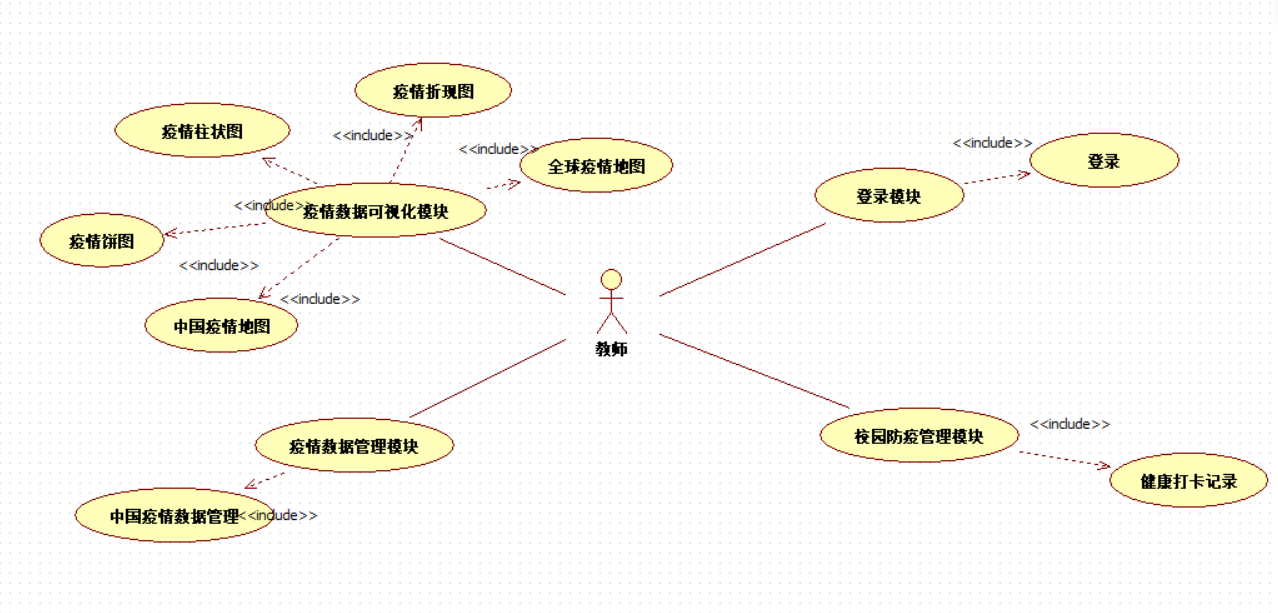


图4-3 管理员用例图

#### 教师用例图设计

教师用例图如图4-4所示

图4-4 教师用例图

#### 4.2.4学生用例图设计

学生用例图如图4-5所示

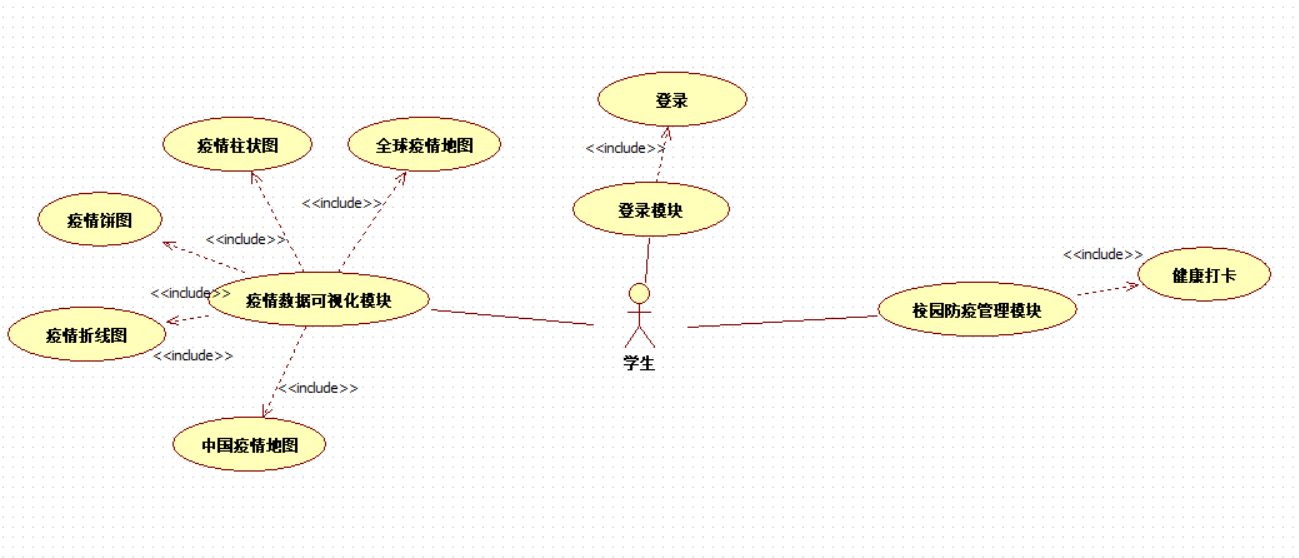


图4-5 学生用例图

## 4.3系统模块功能性需求

#### 4.3.1 普通用户模块

该模块主要实现用户在网站登录，系统统一设置登录密码为123456，在登录系统之后，用户可以修改密码，对网站中的页面进行操作，例如关闭当前页面，刷新当前，关闭全部页面，用户还可以查看中国疫情数据。用户模块的用例图如4-6所示

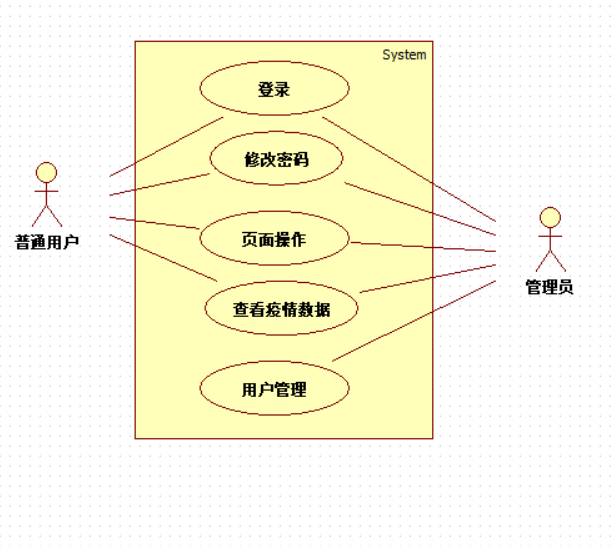


图4-6 用户模块用例图

#### 4.3.2 疫情数据可视化模块

在高校疫情管理系统中，将中国疫情地图的数据显示在后台首页，任何登录都可以进行查看，任何角色的用户登录系统都可以查看疫情数据的显示，不同形式的疫情数据的数据具有一致性，其中全球疫情数据是连接数据库进行查询，没有实时更新。疫情数据可视化模块的用例图如4-7所示。

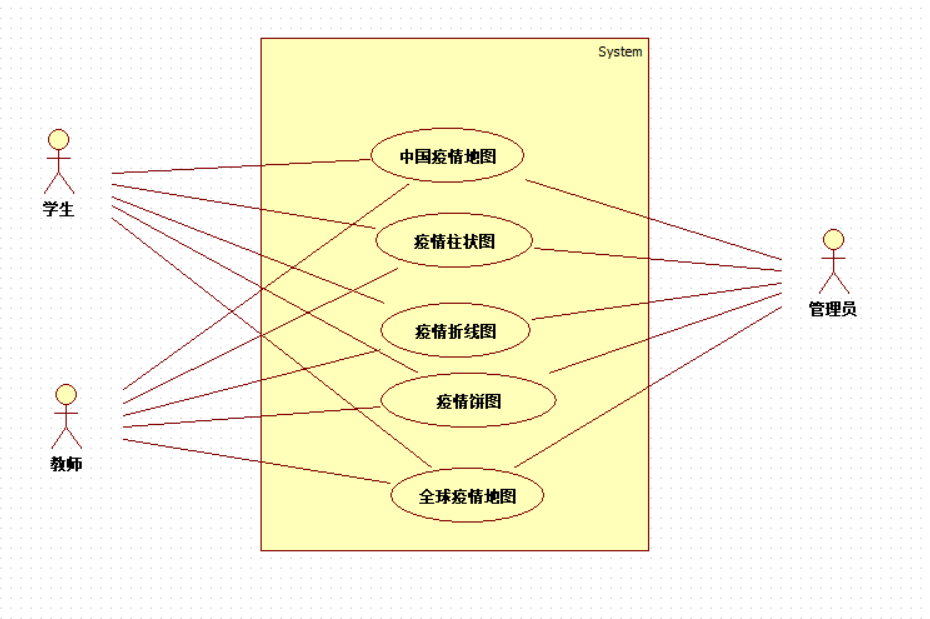


图4-7 疫情数据可视化模块用例图

#### 4.3.3 疫情数据管理模块

在高校疫情防控系统中，教师和管理员都拥有疫情数据管理模块的所有权限。教师和管理员可以对疫情数据进行增删改查，同时也可以将数据进行上传和下载，还可以打印当前页的疫情数据。疫情数据管理模块用例图如4-8所示。

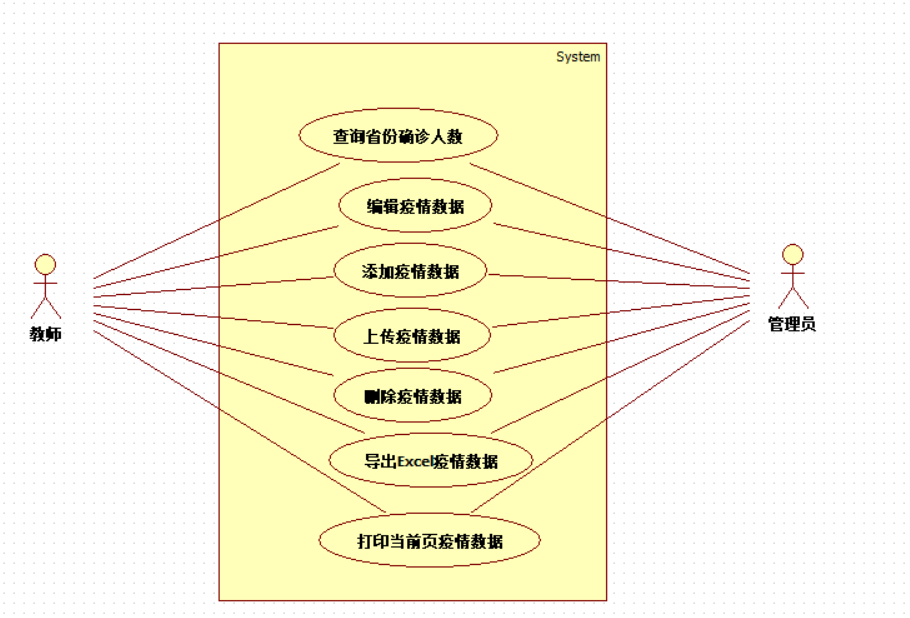


图4-8 疫情数据管理模块用例图

#### 4.3.4 校园防疫管理模块

在本系统校园防疫管理模块中，学生可以每天登录系统进行健康打卡，教师和管理员可以登录系统查看学生打卡情况，用户可以对健康打卡记录进行增删改查，也可以将数据进行导出或者打印。校园防疫管理模块用例图如4-9所示。

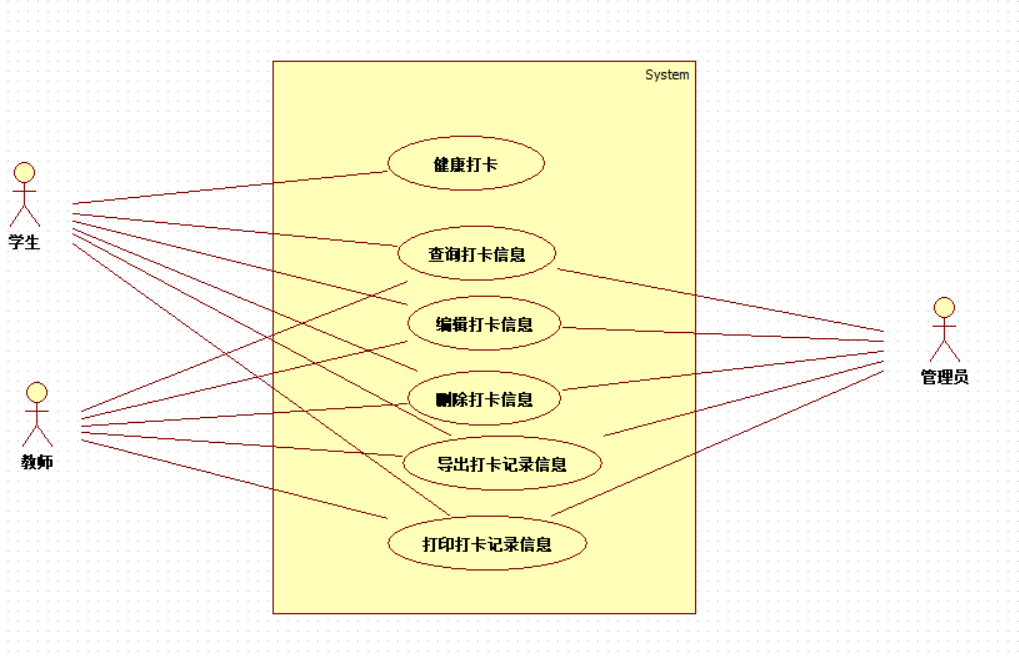


表4-9 校园防疫管理模块用例图

#### 4.3.5 系统管理模块

在校园疫情防控系统的系统管理模块，主要是实现了对不同用户的权限管理。角色主要分为管理员、教师和学生，不同的角色会赋予不同的菜单权限，用户管理中，可以对一个用户赋予多个角色，这样就能够实现对用户赋予不同的权限。系统管理模块用例图如图所示。

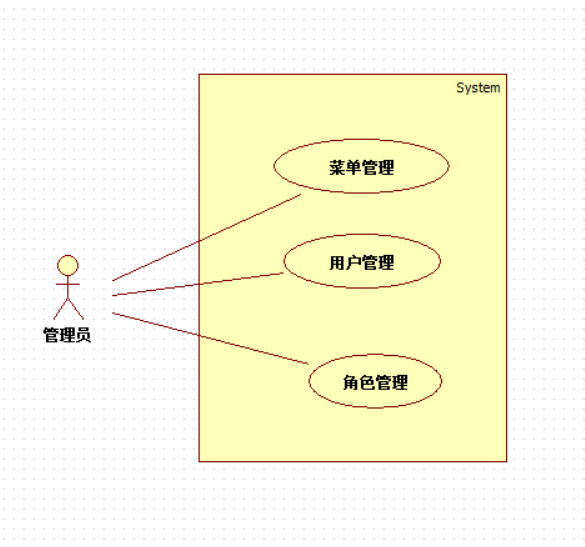


图4-10 系统管理模块用例图

## 4.4系统非功能性需求

在完成了一些基本功能需求之外，还需满足一些肺功能性的需求，来保证用户能够正常使用该系统，具体如下：

#### 4.4.1 安全性

系统中实现了对疫情数据的增删改查，以及用户的一些个人信息，为了保证系统的正常使用以及对疫情实况的了解，需要保证系统具有一定的安全性。

#### 4.4.2 高效性

校园疫情防控系统中有实现了对疫情数据的上传和下载，大量的疫情数据对系统的高效性提出了要求，因此在开发中应该提升系统对疫情数据的访问效率，所以校园疫情防控系统应该具备较高的高效性。

#### 4.4.3 可扩展性

由于疫情具有不稳定因素，疫情的不断变化要求我们实现的需求也是未知的，因此这个系统应当具备灵活性，可以适时拓展新的功能以满足当下需要。

#### 4.4.4 易用性

系统设计应该是简单易学的，设计的各种功能应该简单明了，不需要努力学习和培训，缩短用户熟悉系统的进程。

#### 4.4.5 一致性

由于系统中实现了疫情数据的不同展示形式，所以要求疫情数据的不同展示方式具备一致性，数据应当准确，同时数据还要可以独立保存，不能删除数据时把还需要的数据都删除掉。

#### 4.4.6 时效性

由于疫情数据需要不定时的更新，才能实现我们对于疫情的正常防控，所以要求系统显示的数据具有失效性。

**5系统详细分析**

## 5.1设计方法

本系统基于B/S开发模式，前后端跨域问题主要是使用Ajax和HttpClient完成。前端页面使用Jquery进行搭建，LayUI进行页面的美化，ECharts用于展示中国疫情地、世界疫情地图，以及使用柱形图、饼状图、折线图显示中国疫情数据，后端通过控制层调用业务层，业务层实现方法在通过持久层和Mapper接口层来查询MySQL数据库，查询到数据控制层通过JSON数据形式返回给前端，前端通过Ajax请求到数据，然后显示给前端视图层。

该系统的特定模块通过Spring Boot来实现，并利用MyBatisPlus来继承了BaseMapper接口中一系列CRUD操作，可以直接使用，并且内置的分页插件完成了系统数据的分页查询功能。本系统使用Shiro作为系统的安全框架来实现用户的认证和授权。除了一些静态资源可以正常放行之外，系统的其他功能均会被拦截，直到完成了用户的登录。数据使用MySql来实现存储，可以实现数据的批量的增删改查。

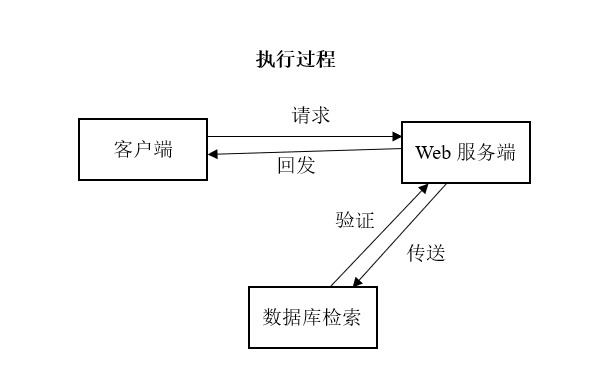


图5-1系统原理图

## 5.2系统模块详细设计

#### 5.2.1 用户模块

1. 面对用户角色

用户模块对用户角色主要实现以下功能：

(1) 网站登录页面进行登录，主要实现普通在网络登录页面进行登录，登录要求输入正确的用户名、密码、验证码即可完成登录。

(2) 登录成功之后，用户就可以在后台首页查看中国疫情数据。

(3) 用户登陆该站点后可以通过该网页进行自己的密码进行更改，系统的初始密码为123456。用户模块对应用户角色实现的功能序列图如5-2所示。

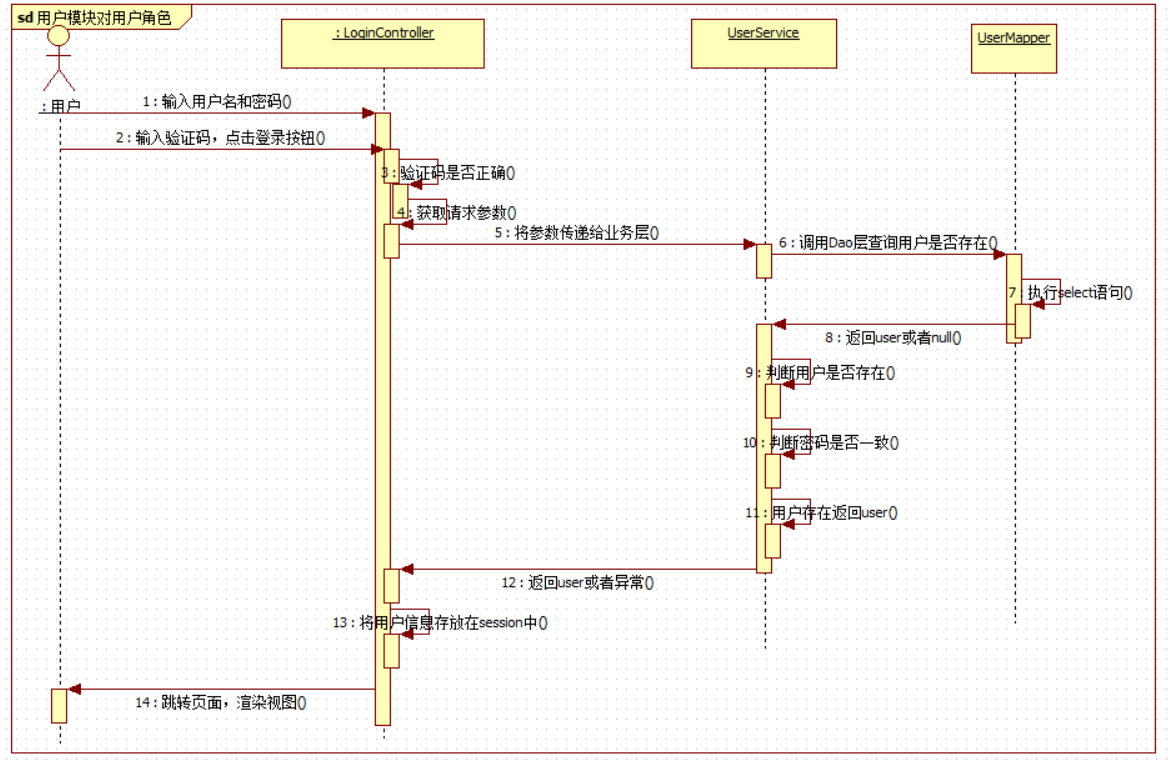


图5-2 用户实现的功能序列图

1. 面对管理员角色

用户模块对管理员角色主要实现以下功能：

(1) 网站登录页面进行登录，管理员登录要求输入正确的用户名、密码、验证码即可完成登录。

(2) 登录成功之后，用户就可以在后台首页查看中国疫情数据。

(3) 用户登陆该站点后可以通过该网页进行自己的密码进行更改，系统的初始密码为123456。管理员也可以对普通用户的密码进行初始化或者管理。用户模块对应管理员角色实现的功能序列图如5-3所示。

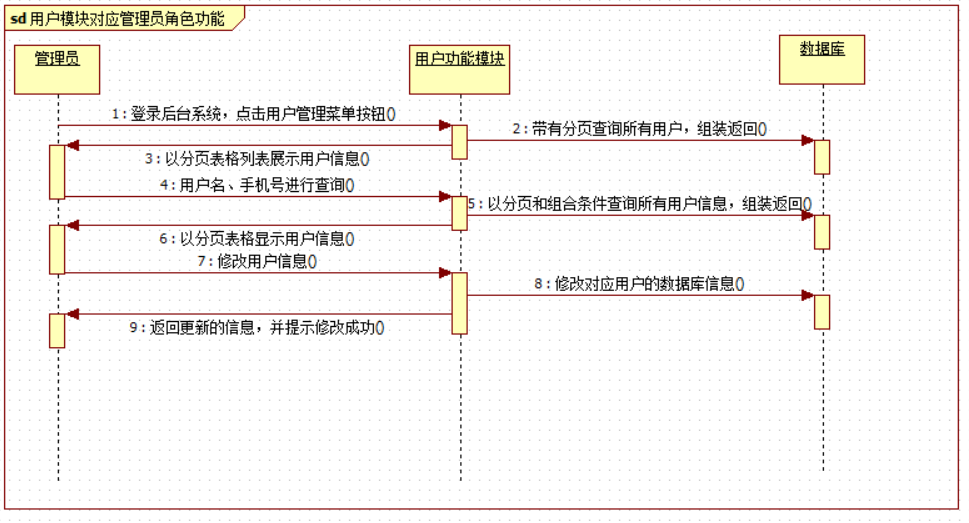


图5-3 管理员角色功能序列图

#### 5.2.2 疫情数据可视化模块

疫情数据可视化对应登录用户主要实现以下功能：

(1) 系统用户正常登录系统

(2) 登录成功之后，用户点击疫情数据可视化菜单，进行查看疫情信息

疫情数据可视化模块对应登录用户实现的功能序列图如5-4所示。

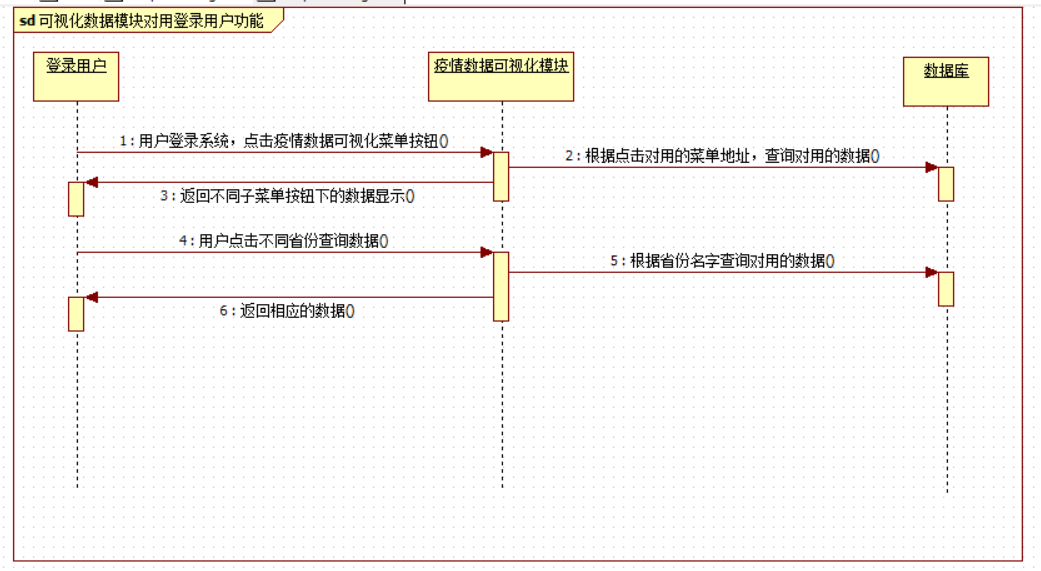


图5-4 可视化数据模块对用登录用户功能序列图

#### 5.2.3 疫情数据管理模块

疫情数据管理模块对应教师或者管理员主要实现以下功能：

(1) 教师或者管理员可以查看到所有省份的疫情数据，并且可以按照省份名称进行条件搜索。

(2) 教师或者管理员可以对疫情数据进行基本的管理操作，包括增加数据，编辑和删除数据，同时也可以对数据进行上传和下载。

疫情数据管理模块对应管理员和教师实现的功能序列图如5-5所示。

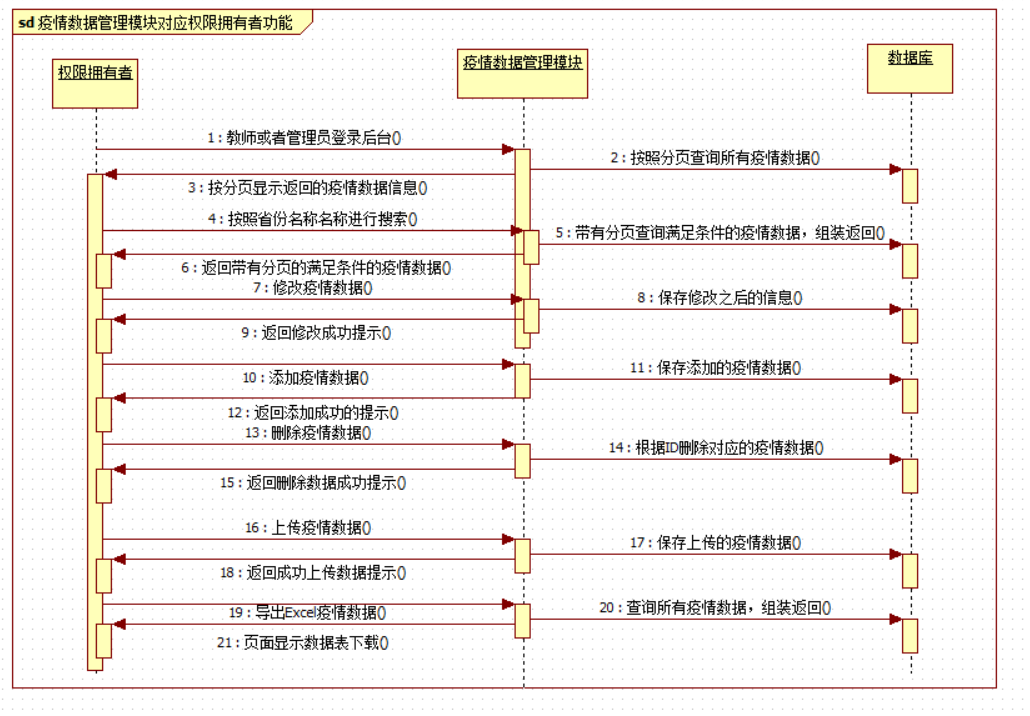


图5-5 疫情数据管理模块对应管理员或教师功能序列图

#### 5.2.4 校园防疫管理模块

1．面对学生用户角色

校园防疫管理系统对于学生用户主要实现以下功能：

(1) 学生可以每天登录系统进行健康打卡。

(2) 学生可以对打卡记录进行操作，包括编辑和删除记录，根据姓名和手机号进行搜索打卡记录。同时也可以对当前页的打卡记录进行导出。

校园防疫管理模块对应学生用户实现的功能序列图如5-6所示。

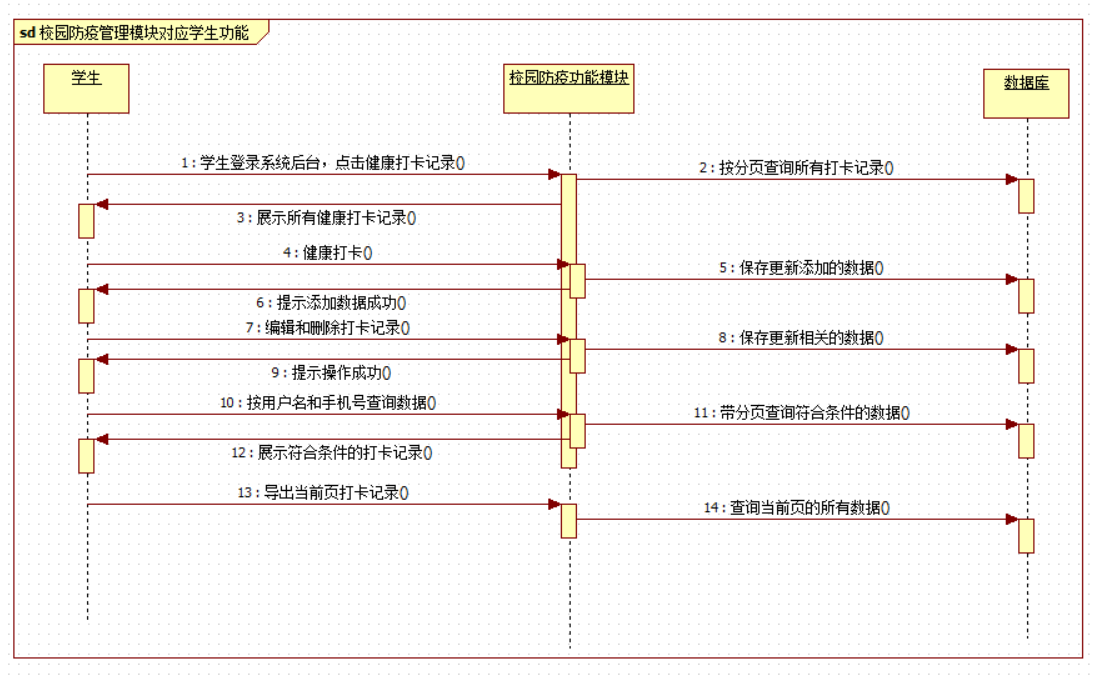


图5-6 校园防疫管理模块对应学生功能序列图

2.面对管理员和教师角色

校园防疫管理系统对于教师和管理员用户主要实现以下功能：

(1) 教师和管理员可以登录系统查看学生打卡情况。

(2) 教师和管理员可以对打卡记录进行操作，包括编辑和删除记录，根据姓名和手机号进行搜索打卡记录。同时也可以对当前页的打卡记录进行导出。

校园防疫管理模块对应教师和管理员用户实现的功能序列图如5-7所示。

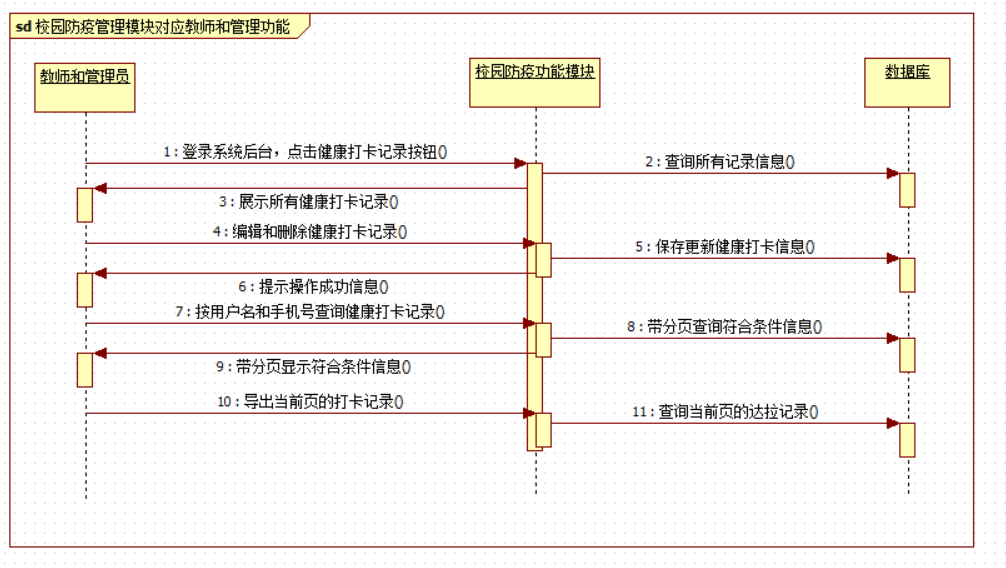


图5-7 校园防疫管理模块对应教师和管理员功能序列图

#### 5.2.5 系统管理模块

菜单管理模块对应管理员主要实现以下功能：

(1) 管理员可以登录系统，进入菜单管理页面，可以对菜单进行增删改查操作，同时可以设置菜单的地址，对菜单显示表格内容进行筛选，导出和打印。

(2) 管理员登录系统后台，进入角色管理页面，可以对角色进行基本的增删改查操作，以及给对应的角色分配权限。

(3) 管理员登录系统后台，进入用户管理页面，可以对用户进行基本的增删改查操作，以及对用户进行重置密码，分配一个或者多个角色。

系统管理模块对应管理员实现的功能序列图如5-8所示。

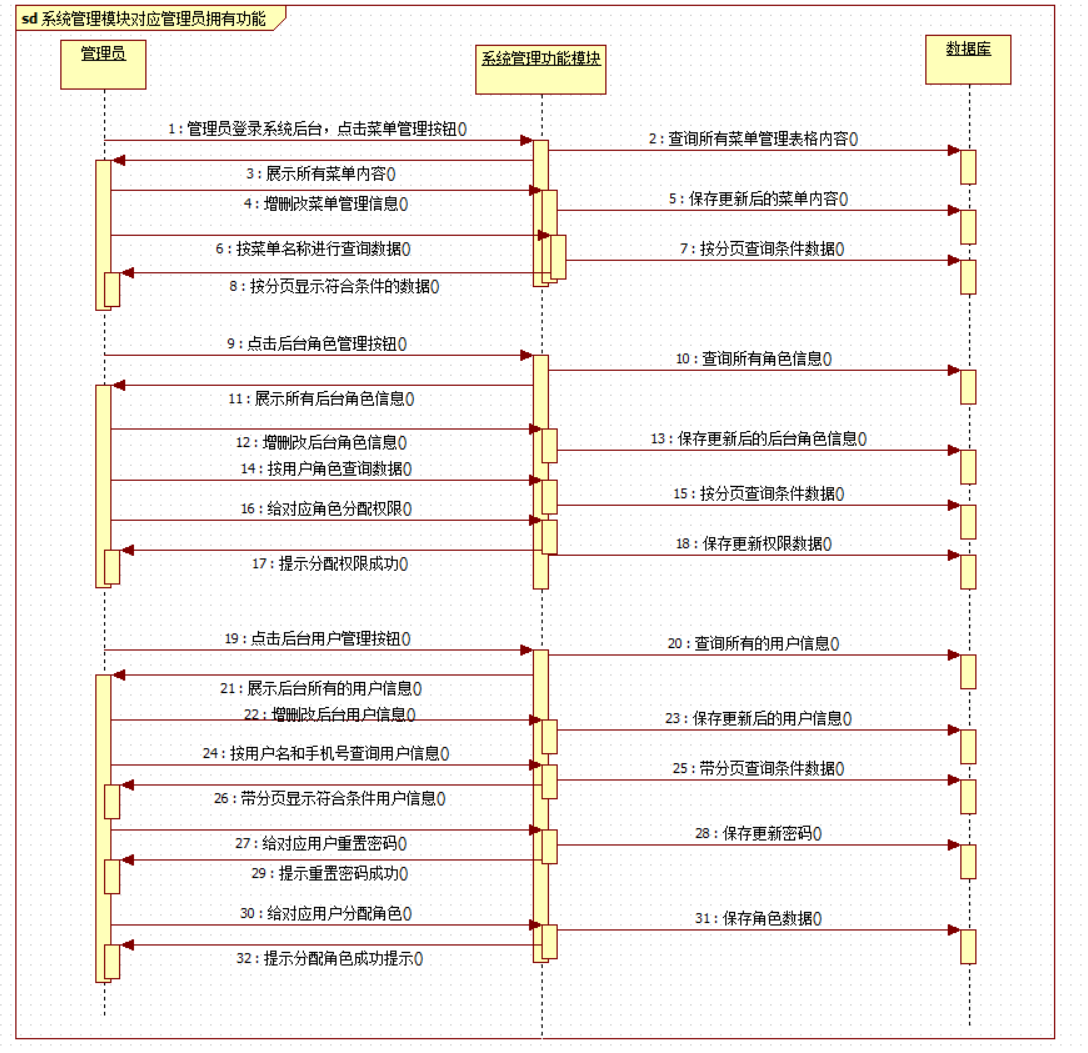


图5-8 系统管理模块对应管理员功能序列图

## 5.3数据库设计

高校疫情系统的主要实现是方便用户即使查看疫情数据，以及实时进行校园防控管理，这要求该系统可以对大量的数据进行存储和处理，因此需要应用到数据库。

#### 5.3.1 概念模型设计

概念模型设计主要是基于数据分析从上而下设计整个系统的数据库概念结构。从用户的角度开发视图，并分析集成视图以获取最终结果。根据系统分析，高校疫情系统的总体E-R模型如图5-10所示。

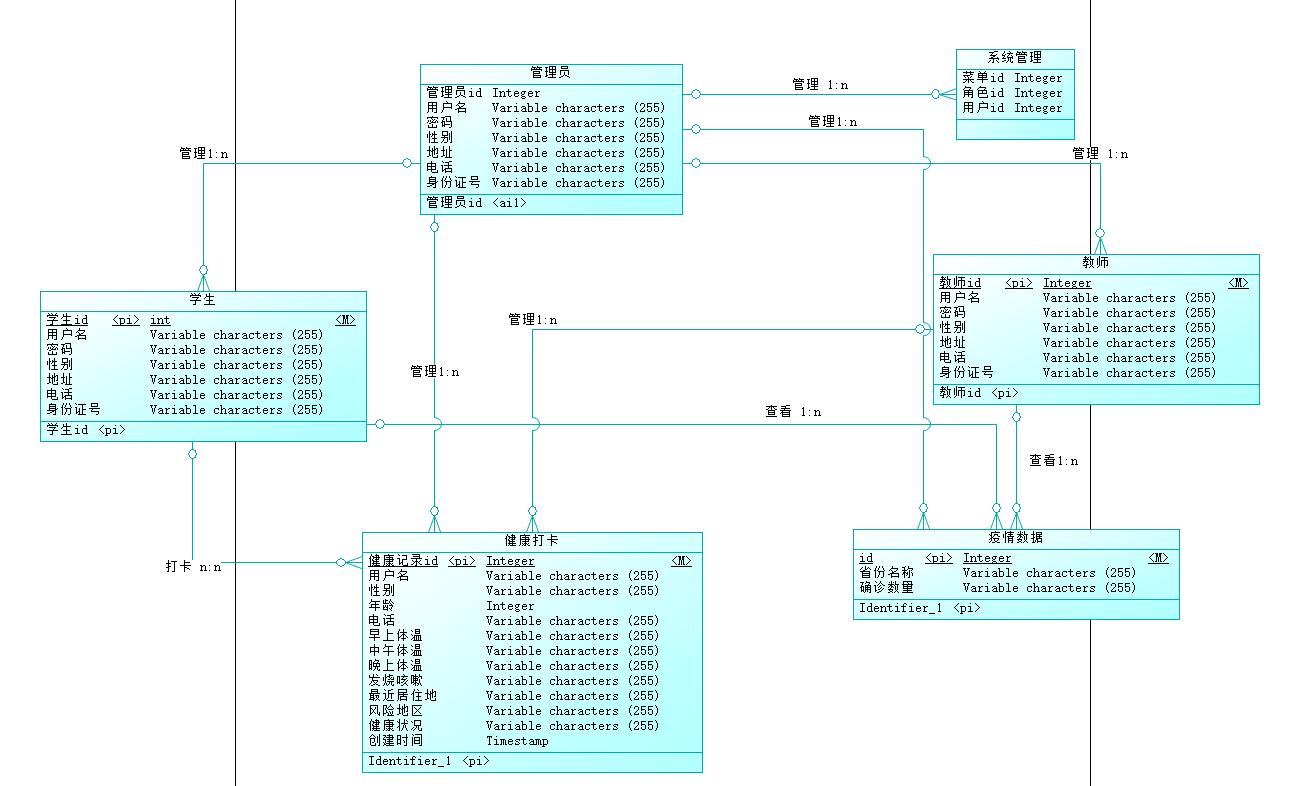


图5-10 系统总体E-R模型图

#### 5.3.2 数据库表设计

数据库设计概念是基于数据分析从下而上设计整个系统21的数据库概念结构。从用户角度开发视图，并包括数据项、记录与记录之间的关联、一致性和安全性限制，分析集成视图以获取最终结果。

校园疫情防控系统的数据库表设计如下：

1. 中国疫情数据表(nocv\_data)，设计了中国疫情数据ID、省份名称、确诊人数、更新时间字段。具体设计如下表5-1所示。

表5-1 中国疫情数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 疫情数据ID |
| name | varchar | 255 | 无 | 省份名称 |
| value | int | 11 | 无 | 确诊数量 |
| update\_time | datetime | 0 | 无 | 更新时间 |

1. 全球疫情数据表(nocv\_global\_data)，设计了全球疫情数据ID、国家名称、确诊数量字段，具体设计如下表5-2所示。

表5-2 全球疫情数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 疫情数据ID |
| name | varchar | 255 | 无 | 国家名称 |
| value | int | 11 | 无 | 确诊数量 |

1. 疫情折线图(line\_trend)，设计了疫情数据ID、确诊人数、隔离人数、治愈人数、死亡人数、疑似人数、创建时间字段，具体设计如下表5-3所示。

表5-3 疫情折线图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 疫情数据ID |
| confirm | int | 11 | 无 | 确诊人数 |
| isolation | int | 11 | 无 | 隔离人数 |
| cure | int | 11 | 无 | 治愈人数 |
| dead | int | 11 | 无 | 死亡人数 |
| similar | int | 11 | 无 | 疑似人数 |
| create\_time | datetime | 0 | 无 | 创建时间 |

1. 健康打卡数据表(health\_clock)，设计了健康打卡ID、用户名、性别、年龄、电话、早上体温、中午体温、晚上体温、发烧咳嗽、近期居住地、风险地区、近期接触地、健康状况、创建时间字段，具体设计如下表5-4所示。

表5-4 健康打卡数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 健康打卡ID |
| username | varchar | 255 | 无 | 用户名 |
| sex | varchar | 255 | 无 | 性别 |
| age | int | 11 | 无 | 年龄 |
| phone | varchar | 255 | 无 | 手机号 |
| morning\_temp | varchar | 255 | 无 | 早上体温 |
| afternoon\_temp | varchar | 255 | 无 | 中午体温 |
| night\_temp | varchar | 255 | 无 | 晚上体温 |
| fever\_and\_cough | varchar | 255 | 无 | 发烧咳嗽 |
| recent\_zone | varchar | 255 | 无 | 近期居住地 |
| risk\_zone | varchar | 255 | 无 | 风险居住地 |
| health\_status | varchar | 255 | 无 | 健康状况 |
| create\_time | timestamp | 0 | 时间默认为当前时间 | 创建时间 |

1. 菜单表(menu)，设计了菜单ID，父级菜单ID、类型、菜单名称、权限、菜单图标、菜单地址、是否展开、排序码、是否可用字段，具体设计如下表5-5所示。

表5-5 菜单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 菜单ID |
| pid | int | 11 | 无 | 父级菜单ID |
| type | varchar | 255 | 无 | 类型 |
| permission | varchar | 255 | 无 | 权限 |
| icon | varchar | 255 | 无 | 菜单图标 |
| href | varchar | 255 | 无 | 菜单地址 |
| open | int | 11 | 无 | 是否展开 |
| ordernum | int | 11 | 无 | 排序码 |
| available | int | 11 | 无 | 是否展开 |

1. 角色表(role)，设计了角色ID、角色名称、角色备注字段，具体的设计如表5-6所示。

表5-6 角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 角色ID |
| name | varchar | 255 | 无 | 角色名称 |
| remark | varchar | 255 | 无 | 角色备注 |

1. 菜单角色表(role\_menu)，设计了角色id、菜单id字段，具体的设计如表5-7所示。

表5-7 菜单角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| rid | int | 11 | 无 | 角色ID |
| mid | int | 11 | 无 | 菜单ID |

1. 用户表(user)，设计了用户ID、用户名、密码、性别、年龄、地址、电话、身份证、班级ID、学院ID、教师ID、盐字段，具体设计如表5-8所示。

表5-8用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 用户ID |
| username | varchar | 255 | 无 | 用户名 |
| password | varchar | 255 | 无 | 密码 |
| sex | varchar | 255 | 无 | 性别 |
| age | int | 11 | 无 | 年龄 |
| address | varchar | 255 | 无 | 地址 |
| phone | varchar | 255 | 无 | 电话 |
| card | varchar | 255 | 无 | 身份证号 |
| ban\_ji\_id | int | 11 | 无 | 班级ID |
| xue\_yuan-id | int | 11 | 无 | 学院ID |
| teacher\_id | int | 11 | 无 | 教师ID |
| salt | varchar | 255 | 无 | 盐 |

1. 用户角色表(user\_role)，设计了用户ID、角色ID字段，具体的设计如表5-9所示。

表5-9 用户角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| uid | int | 11 | 无 | 用户ID |
| rid | int | 11 | 无 | 角色ID |

1. 学院表(xue\_yuan)，设计了学院ID、学院名字，具体的设计如表5-10所示。

表5-10 学院表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 学院ID |
| name | varchar | 255 | 无 | 学院名字 |

1. 班级表(ban\_ji)，设计了班级ID、班级名字、学院ID字段，具体设计如表5-11所示。

表5-11 班级表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 班级ID |
| name | varchar | 255 | 无 | 班级名字 |
| xue\_yuan\_id | int | 11 | 无 | 学院ID |

1. 中国疫情数据表(china\_total)，设计了疫情数据ID、累计确诊数量、累计输入病例、重症病例、累计治愈病例、累计死亡人数、疑似病例、更新时间字段，具体设计如表5-12所示。

表5-12 中国疫情数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 数据长度 | 约束 | 注释 |
| id | int | 11 | 主键，非空，自增 | 疫情数据ID |
| confirm | int | 11 | 无 | 累计确诊数量 |
| input | int | 11 | 无 | 累计输入病例 |
| severe | int | 11 | 无 | 重症病例 |
| heal | int | 11 | 无 | 累计治愈病例 |
| dead | int | 11 | 无 | 累计死亡人数 |
| suspect | varchar | 11 | 无 | 疑似病例 |
| update\_time | datetime | 0 | 无 | 更新时间 |

**6系统实现**

## 6.1配置文件代码

校园疫情系统后端使用Spring Boot自动构建，使用Spring MVC构建项目，持久层使用MyBatisPlus实现带有分页的查询功能，使用Shiro完成登录验证功能，所以相关的配置文件主要是Shiro配置文件和一些有关数据库的配置文件。

Shiro的配置文件如下参考源码中的Config包下的ShiroConfig文件。

其他模块的配置文件如下表6-1所示。

|  |
| --- |
| spring:  datasource:  username: root  password: 123456  url: jdbc:mysql://localhost:3306/nocv?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone = GMT  driver-class-name: com.mysql.cj.jdbc.Driver  #shiro的配置(不进行拦截的路径）  shiro:  anon-urls:  - /toLogin\*  - /login.html\*  - /login/login  - /login/getCode  - /css/\*\*  - /echarts/\*\*  - /images/\*\*  - /layui/\*\*  - /layui\_ext/\*\*  - /js/\*\*  login-url: /index.html  log-out-url: /login/logout\*  authc-ulrs:  - /\*\* |

## 6.2用户模块

#### 6.2.1 登录功能

用户在登录网站的时候，需要填入图片上的验证码，所以后端需要实现获取到验证码以及从前台获取到用户填入的验证码进行比对，主要的业务流程代码如下表6-2所示。

表6-2 后端发送验证码和登录功能代码

|  |
| --- |
| //验证码逻辑  LineCaptcha captcha = CaptchaUtil.createLineCaptcha(116, 36, 4, 10);  session.setAttribute("code",captcha.getCode());  ServletOutputStream outputStream = response.getOutputStream();  captcha.write(outputStream);  outputStream.close();  //登录功能逻辑  if(code!=null && sessionCode.equals(code)){  Subject subject = SecurityUtils.getSubject();  UsernamePasswordToken token = new UsernamePasswordToken(username, password);  subject.login(token);  User user = (User) subject.getPrincipal();  if (user!=null){  dataView.setCode(200);  dataView.setMsg("登录成功！");  session.setAttribute("user",user);  return dataView;  }else {  dataView.setCode(100);  dataView.setMsg("用户名或者密码错误，登录失败！");  }  }  dataView.setCode(100);  dataView.setMsg("验证码错误！");  return dataView; |

#### 6.2.1 登录界面

系统的前端登录界面如图6-1所示。



图6-1 登录界面

#### 6.3疫情数据可视化模块 6.3.1 中国疫情数据可视化

中国疫情数据可视化主要是利用HttpClient网页爬虫技术进行爬取网页接口的疫情数据，能够对数据进行实时更新，然后对爬取的数据进行解析，最后前端发送请求获取显示数据在前台。主要的业务流程代码如下表6-3所示。

表6-3 疫情数据对接网易的后端实现代码

|  |
| --- |
| //发送网络请求的工具  RequestConfig requestConfig = RequestConfig.custom()  .setSocketTimeout(10000)  .setConnectTimeout(10000)  .setConnectionRequestTimeout(10000)  .build();  CloseableHttpClient httpClient = null;  HttpGet request = null;  CloseableHttpResponse response = null;  try {  httpClient = HttpClients.createDefault();  request = new HttpGet("https://c.m.163.com/ug/api/wuhan/app/data/list-total");  request.setConfig(requestConfig);  response = httpClient.execute(request);  int statusCode = response.getStatusLine().getStatusCode();  if(statusCode==200){  HttpEntity entity = response.getEntity();  String String = EntityUtils.toString(entity, "utf-8");  return String;  }  }finally {  if (response!=null){  response.close();  }  if (request!=null){  request.releaseConnection();  }  if (httpClient!=null){  httpClient.close();  }  }  return null;  //发送查询数据请求  @RequestMapping("/query")  @ResponseBody  public List<NocvData> queryData() throws ParseException {  //倒叙查询最后34条数据  List<NocvData> list = indexService.listOrderByLimit34();  return list;  } |

#### 6.3.2 中国疫情数据可视化页面

中国疫情可视化模块中的中国疫情地图前端界面如图6-2所示。

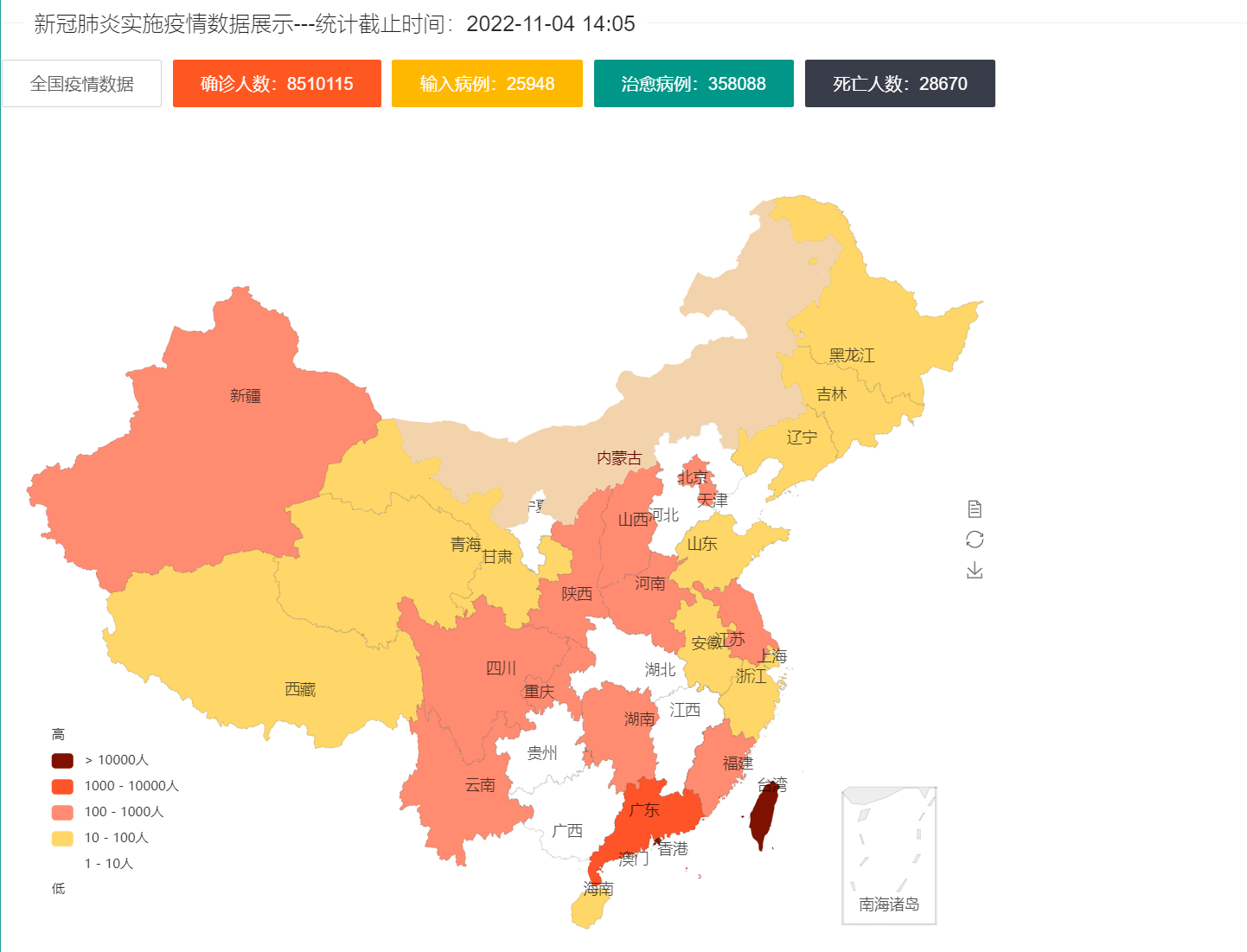


图6-2 中国疫情地图

## 6.3疫情数据可视化模块

### 6.3.1 查询疫情数据功能

由于查询数据库时会插入大量的数据，所以会经常使用到查询功能，使用MyBatisPlus的带有分页的查询功能能够更好的显示疫情数据，后端实现这一功能的业务流程代码主要如下表6-4所示。

|  |
| --- |
| IPage<NocvData> page = new Page<>(nocvDataVo.getPage(), nocvDataVo.getLimit());  //2.创建模糊查询条件  QueryWrapper<NocvData> queryWrapper = new QueryWrapper<>();  queryWrapper.like(!(nocvDataVo.getName()==null),"name",nocvDataVo.getName());  //3.疫情确诊数据最多的排在最上面  queryWrapper.orderByDesc("value");  //4.查询数据库  indexService.page(page,queryWrapper);  //5.返回数据封装  DataView dataView = new DataView(page.getTotal(),page.getRecords());  return dataView; |

### 6.3.2 查询疫情数据页面

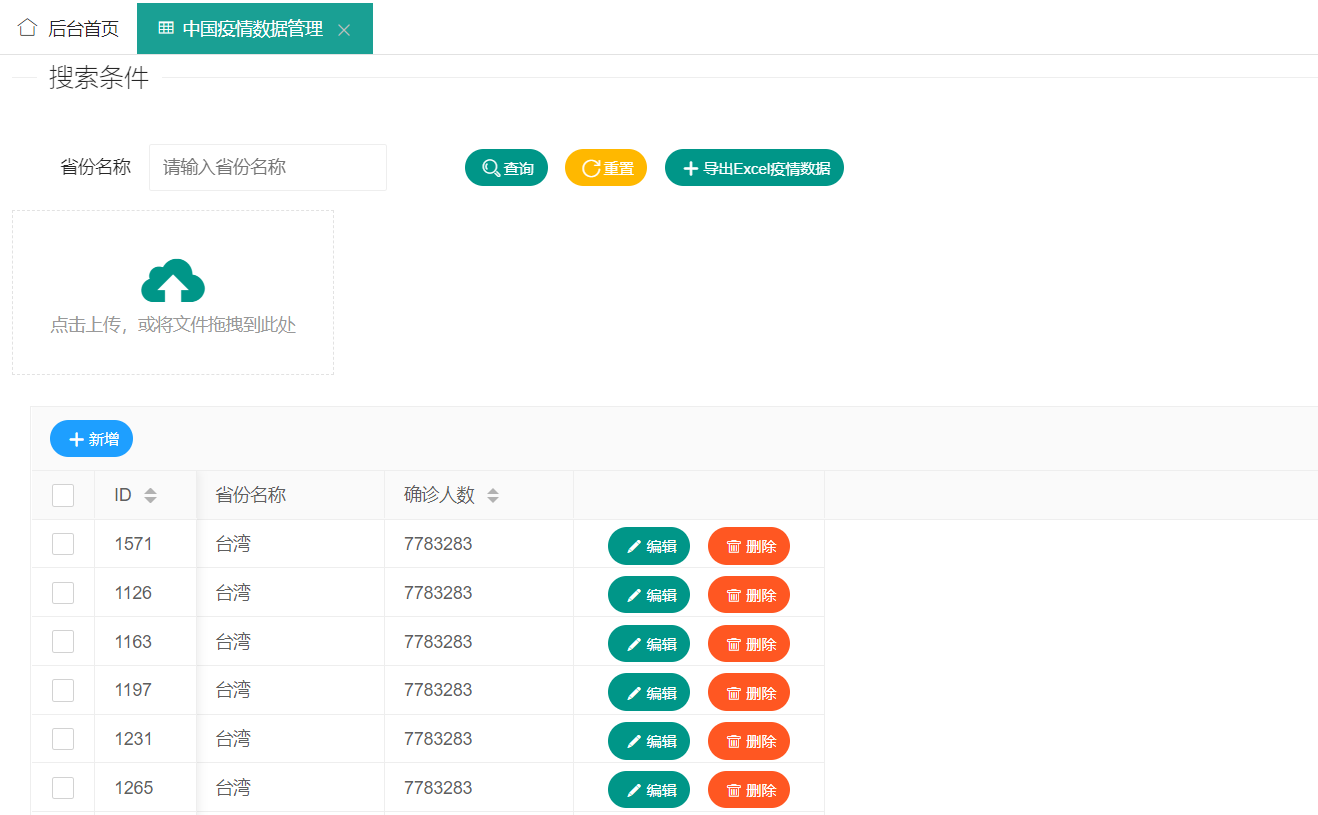


图6-3 查询疫情数据页面

### 6.3.3 导出Excel疫情数据功能

查询到的疫情数据可以导出为excel表格进行存储，可以方便传读和查看。后台的主要业务流程代码如下表6-5。

表6-5 导出为Excel疫情数据代码

|  |
| --- |
| List<NocvData> list = indexService.list();  //2.建立Excel对象 封装数据  response.setCharacterEncoding("UTF-8");  //2.1创建Excel对象  HSSFWorkbook wb = new HSSFWorkbook();  //2.2 创建sheet对象  HSSFSheet sheet = wb.createSheet("中国疫情数据sheet1");  //2.3创建表头名字  HSSFRow hssfRow = sheet.createRow(0);  hssfRow.createCell(0).setCellValue("城市名称");  hssfRow.createCell(1).setCellValue("确诊数量");  //3.遍历数据，封装Excel对象  for (NocvData data : list) {  HSSFRow dataRow = sheet.createRow(sheet.getLastRowNum() + 1);  dataRow.createCell(0).setCellValue(data.getName());//创建单元格，获取城市名字  dataRow.createCell(1).setCellValue(data.getValue());//创建单元格，获取对应的取值  }  //4.建立输出流 输出浏览器文件  OutputStream os = null;  //4.1.设置Excel的名称 输出类型编码  try {  response.setContentType("application/octet-stream;charset=utf-8");  response.setHeader("Content-Disposition", "attachment;filename=" + new String("中国疫情数据表".getBytes(), "iso-8859-1") + ".xls");  //4.2用输出流写到excel  os = response.getOutputStream();  wb.write(os);  os.flush();  } catch (Exception e) {  e.printStackTrace();  } finally {  try {  if (os != null) {  os.close();  }  } catch (Exception e) {  e.printStackTrace();  }  } |

## 6.4校园防疫管理

### 6.4.1 健康打卡记录功能

在系统后台，学生登录系统之后可以进行健康打卡，包括早中晚的体温，居住地，风险地区，近期接触地和健康状况等，在后端实现主要业务流程代码如下表6-6。

表6-6 健康打卡记录的主要代码

|  |
| --- |
| public DataView addOrUpdateHealthClock(HealthClock healthClock){  boolean save = healthClockService.saveOrUpdate(healthClock);  DataView dataView = new DataView();  if (save){  dataView.setCode(200);  dataView.setMsg("添加成功！");  return dataView;  }  dataView.setCode(100);  dataView.setMsg("添加失败！");  return dataView; |

### 6.4.2 健康打卡记录页面



图6-4 健康打卡前端页面显示

## 6.5校园防疫管理

#### 6.8.1 分配后台功能权限功能

该系统通过菜单、角色和用户三表查询达到分配权限的功能。管理员能够通过给不同的角色赋予不同的权限，一个用户可以分配一个或者多个角色，从而达到对用户分配不同权限的目的。后端的实现的该功能的业务流程代码如下表6-7所示。

表6-7 分配权限功能后端代码

|  |
| --- |
| @RequestMapping("/initPermissionByRoleId")  @ResponseBody  public DataView initPermissionByRoleId(Integer roleId) {  //1.把所有的菜单权限查出来  QueryWrapper<Menu> queryWrapper = new QueryWrapper<>();  List<Menu> allPermissions = menuService.list();  //2.根据角色rid查询所有的菜单mid  List<Integer> currentRolePermissions = roleService.queryAllPermissionByRid(roleId);  //3.rid mid 所有的ID，去查询菜单和角色的数据  List<Menu> menus = null;  //4.查询到mid rid  if (currentRolePermissions.size() > 0) {  queryWrapper.in("id", currentRolePermissions);  menus = menuService.list(queryWrapper);  } else {  menus = new ArrayList<>();  }  //5.返回前台的格式，带有层级关系 treenode  List<TreeNode> nodes = new ArrayList<>();  for (Menu allPermission : allPermissions) {  String checkArr = "0";//标志符  for (Menu currentPermission : menus) {  if (allPermission.getId().equals(currentPermission.getId())) {  checkArr = "1";  break;  }  }  Boolean spread = (allPermission.getOpen() == null || allPermission.getOpen() == 1) ? true : false;  nodes.add(new TreeNode(allPermission.getId(), allPermission.getPid(), allPermission.getTitle(), spread, checkArr));  }  return new DataView(nodes);  }  @RequestMapping("/saveRolePermission")  @ResponseBody  public DataView saveRolePermission(Integer rid,Integer[] mids) {  //1.保存分配权限的时候，需要根据rid删除之前所有mid的权限  roleService.deleteRoleByRid(rid);  //2.保存权限  if (mids != null && mids.length > 0) {  for (Integer mid : mids) {  roleService.saveRoleMenu(rid, mid);  }  }  DataView dataView = new DataView();  dataView.setCode(200);  dataView.setMsg("菜单权限分配成功！");  return dataView;  }  } |

#### 6.8.2 分配后台功能权限页面

在后台系统的角色管理功能的给角色分配后台功能权限的前端界面如图6-8所示。

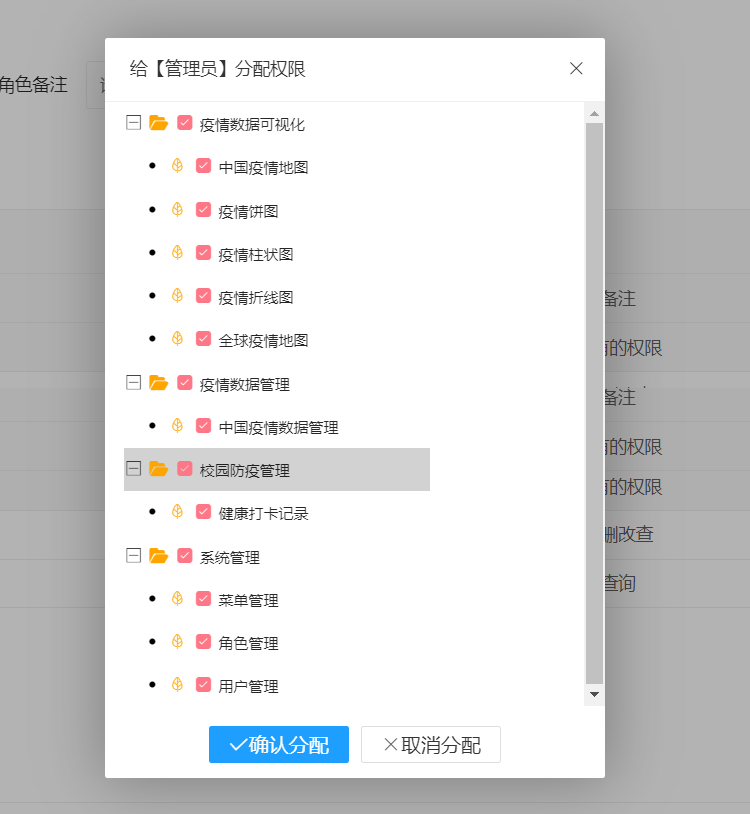


图6-5 分配权限页面

# 7总结与展望

本系统是一个基于Java的疫情管理系统，系统可以对相关信息进行收集、编辑和查询。

在整个项目的开发过程中遇到了很多困难。所谓万事开头难，在系统开发初期就出现中国疫情地图不能正常显示的问题，非常感谢老师大晚上帮助我解决了这个问题，才使我的项目得以继续下去。在这期间也不断解决了很多问题，感谢身边的同学的热心帮助！不过伴随着项目开发，我慢慢的积累了许多经验，提高了解决问题的能力，从刚开始一遇到bug的手忙脚乱，到后面的淡定从容，也是一种学习和成长。同时我也深知书本中的知识大多偏向于理论，必须应用于实践才行。

由于个人能力的有限和时间比较仓促，该系统基本功能都做了实现，但有些功能还有待后续完善，有些功能设计实现的还不太完美，界面设计也不太漂亮。后续我会不断继续完善本项目，争取更好。