|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| C:\Users\Administrator\Desktop\78D172AE6E29983F939CC0FCDA3_12AD0ED6_2FE0.gif78D172AE6E29983F939CC0FCDA3_12AD0ED6_2FE0 | | | |
| 项目功能论文 | | | |
| 基于疫情的疫情封控管理系统的  设计与实现 | | | |
|  | | | |
|  | 组 名 | 不会编程的小白组 |  |
|  | 学 院 | 计算机与信息技术学院 |  |
|  | 专 业 | 计算机科学与技术 |  |
|  | 班 级 | 计科1905 |  |
|  | | | |
| 二零二二 年 5 月 30 日 | | | |

摘 要

随着社会信息化的日益增强，人们对电子产品的需求也在不断增加，使得网络系统日趋流行。传统的手工统计管理方法大多是人工进行操作，效率低，使用不便，而在线统计管理则无需人工，基本实现了自动化，节约了大量的人力物力，大大提高了工作效率。基于对疫情封控管理的实际情况进行调研之后，设计了一种基于疫情的在线疫情封控管理系统。

本系统涉及Spring Boot、Vue框架等技术，Java作为系统的编程语言，JDK使用1.8.0\_25，IDE为IntelliJ IDEA，在疫情封控管理系统前端界面的实现上，主要依靠 vue、element-ui、vue-router、vuex、echarts等技术。前端和后端使用两个框架进行开发，便于后期的维护和扩充。本系统采用 MySQL作为数据库。在前端页面，管理者通过授权登录即可进行打卡、人员管理、查看物资和疫情概况的详细信息并进行增删改查操作的页面跳转。在后端系统中，管理员可以进行打卡管理、人员管理、查看物资和疫情概况的增删改查操作并连接MySQL数据库和Redis。在进行足够的需求分析后，系统设计实现了四个模块：日常防控管理、异常人员管理、防控物资管理、疫情概况。通过这四个业务模块实现了人员和物资的统计和管理，管理者通过登录页面授权登录即可进行人员和物资的详细信息并进行增删改查操作，基本满足了广大用户对疫情封控管理系统的各类需求。

疫情封控管理系统是针对当前的用户需求和社会发展需求而开发的，具有很好的实用价值。本系统具有功能齐全、使用简便、维护方便等特点。使用该疫情封控管理系统，可以极大地简化工作程序，提高工作效率；从而使物资和人员统计结果更具实用性、先进性、高效性。

关键词：疫情；vue；Spring Boot；新冠病毒

ABSTRACT

With the increasing informatization of society, people's demand for electronic products is also increasing, making network systems increasingly popular. The traditional manual statistical management methods are mostly manual operation, low efficiency, inconvenient to use, while online statistical management does not require labor, basically realize automation, save a lot of manpower and material resources, and greatly improve work efficiency. Based on the actual situation of epidemic containment management, an online epidemic containment management system based on the epidemic was designed.

This system involves Spring Boot, Vue framework and other technologies, Java as a programming language of the system, JDK using 1.8.0\_25, IDE for IntelliJ IDEA, in the implementation of the front-end interface of the epidemic containment management system, mainly rely on vue, element-ui, vue-router, vuex, echarts and other technologies. The front-end and back-end are developed using two frameworks for later maintenance and expansion. The system uses MySQL as the database. On the front-end page, managers can log in with authorization to perform punch cards, personnel management, view details of materials and epidemic situation profiles, and add, delete, modify and check the page jump. In the back-end system, administrators can add, delete, or modify punch card management, personnel management, view materials and epidemic profiles, and connect MySQL databases and Redis. After sufficient demand analysis, the system design implemented four modules: daily prevention and control management, abnormal personnel management, prevention and control material management, and epidemic situation overview. Through these four business modules, the statistics and management of personnel and materials are realized, and managers can authorize login through the login page to carry out detailed information of personnel and materials and carry out addition, deletion, modification and inspection operations, which basically meets the various needs of the majority of users for the epidemic containment management system.

The epidemic containment management system is developed for the current user needs and social development needs, and has good practical value. The system has the characteristics of complete functions, easy to use and convenient maintenance. The use of the epidemic containment management system can greatly simplify the work procedures and improve work efficiency; so that the statistical results of materials and personnel are more practical, advanced and efficient.

**Key words：**Pandemic; vue; Spring Boot; Coronavirus

目 录

[第1章 绪论 1](#_Toc105672499)

[1.1 项目背景及意义 1](#_Toc105672500)

[1.2 国内外研究现状 1](#_Toc105672501)

[1.3 主要研究内容 2](#_Toc105672502)

[1.4 论文组织结构 2](#_Toc105672503)

[第2章 关键技术介绍 3](#_Toc105672504)

[2.1 Spring Boot框架 3](#_Toc105672505)

[2.2 Vue框架 3](#_Toc105672506)

[2.3 Java开发语言 4](#_Toc105672507)

[2.4 MySQL数据库 4](#_Toc105672508)

[2.5 Redis 4](#_Toc105672509)

[2.6 ECharts 4](#_Toc105672510)

[2.7 MyBatis-Plus 5](#_Toc105672511)

[2.8 Element UI 5](#_Toc105672512)

[2.9 Vue Router 5](#_Toc105672513)

[2.10 Vuex 5](#_Toc105672514)

[2.11 Vue-cli 5](#_Toc105672515)

[2.12 本章小结 6](#_Toc105672516)

[第3章 系统需求分析 7](#_Toc105672517)

[3.1 需求概述 7](#_Toc105672518)

[3.2 功能需求分析 7](#_Toc105672519)

[3.2.1 日常防控管理 7](#_Toc105672520)

[3.2.2 异常人员管理 8](#_Toc105672521)

[3.2.3 防控物资管理 9](#_Toc105672522)

[3.2.4 疫情概况功能 9](#_Toc105672523)

[3.3 本章小结 10](#_Toc105672524)

[第4章 系统设计 11](#_Toc105672525)

[4.1 体系架构设计 11](#_Toc105672526)

[4.1.1 整体架构设计 11](#_Toc105672527)

[4.1.2 逻辑架构设计 12](#_Toc105672528)

[4.1.3 网络架构设计 12](#_Toc105672529)

[4.1.4 功能架构设计 13](#_Toc105672530)

[4.2 功能详细设计 14](#_Toc105672531)

[4.2.1 日常防控管理 14](#_Toc105672532)

[4.2.2 异常人员管理 15](#_Toc105672533)

[4.2.3 防控物资管理 18](#_Toc105672534)

[4.2.4 疫情概况 19](#_Toc105672535)

[4.3 数据库设计 21](#_Toc105672536)

[4.3.1 概念结构设计 21](#_Toc105672537)

[4.3.2 逻辑结构设计 25](#_Toc105672538)

[4.4 本章小结 27](#_Toc105672539)

[第5章 系统实现 28](#_Toc105672540)

[5.1 日常防控管理模块 28](#_Toc105672541)

[5.2 异常人员管理模块 32](#_Toc105672542)

[5.3 防控物资管理模块 39](#_Toc105672543)

[5.4 疫情概况模块 44](#_Toc105672544)

[5.5 本章小结 45](#_Toc105672545)

[第6章 系统测试 46](#_Toc105672546)

[6.1 测试原则 46](#_Toc105672547)

[6.2 功能测试 46](#_Toc105672548)

[6.2.1 日常防控管理功能测试 46](#_Toc105672549)

[6.2.2 异常人员管理功能测试 47](#_Toc105672550)

[6.2.3 防控物资管理功能测试 48](#_Toc105672551)

[6.2.4 疫情概况功能测试 49](#_Toc105672552)

[6.3 性能测试 49](#_Toc105672553)

[6.4 本章小结 50](#_Toc105672554)

[第7章 总结和展望 51](#_Toc105672555)

[7.1 总结 51](#_Toc105672556)

[7.2 展望 51](#_Toc105672557)

[参考文献 52](#_Toc105672558)

# 第1章 绪论

随着互联网的飞速发展，人们已经习惯了通过系统来管理信息，且随着疫情的飞速蔓延，这使得疫情封控管理已经被广泛地运用。传统的统计方式已不能适应现代社会发展的需要。下面作者将对疫情封控管理系统的项目背景及意义、国内外研究现状、主要研究内容以及论文组织结构进行论述。

## 1.1 项目背景及意义

随着计算机技术的飞速发展，特别是互联网的普及，使人们的日常生活中对信息的需求越来越大。在疫情当下的防疫生活中，我们时常接触一些统计行为，比如异常人员的统计、防疫物质的统计和人员健康打卡的数据统计等。传统的人工统计模式，效率低下，数据提交频繁，统计数据繁琐且费时耗力，统计的数据准确性依然无法得到保障，最主要的问题是工作效率低下，数据统计有误，已经满足不了当今社会的发展需求。开发一款符合当下用户需求的疫情封控管理系统，实现对异常人员的、防疫物质和人员健康打卡的数据统计的自动化管理，提高疫情封控的效率，会极大地减少资源的浪费，能有效提高效率。

基于上述背景，利用互联网等现代信息技术，开发出一种适用于广大群众、操作简便、实用的疫情封控管理系统，能够更快地提供更多的信息反馈。与传统的手工统计相比，在线疫情封控管理系统的优点是非常明显地，用户在家里通过网络即可提交自己的相关数据，而统计人员则可以使用数据库，通过计算机迅速地得出统计结果。这样既能节约大量的时间，又能节约大量的人力物力。

综上所述，疫情封控管理系统可以极大地简化工作程序，提高统计效率，满足当前的社会发展的需求，并且随着互联网技术的不断发展，[疫情封控管理系统的](http://checkonline.paperyy.com/api/write/true)作用也将会不断扩大。

## 1.2 国内外研究现状

二十一世纪，科学技术飞速发展，计算机科学也获得了前所未有的成就，互联网技术的发展与普及也越来越广泛。目前，国内外各高校、科研机构和个人疫情封控管理系统的研究已经取得了很大的进展。

在国内，各地区的企业部门也都有了防控管理功能，这不但增强了防控管理，而且对人员的健康安全也有很大的帮助。在安徽天恩科技有限公司中，已开发出较为成熟的疫情防控系统，他们的产品可以让管理员足不出户就可以及时掌握人员防疫信息、防疫物资采购和应用情况等数据，施工人员可以随时随地快速上报个人防疫信息。

自从疫情爆发以来，只有极少数的国家对疫情进行封控管理。而西方资本主义国家由于个人、自由主义观念盛行，而且更强调个人自由、个人利益至上，疫情初期拒绝戴口罩甚至游行抗议，不管不顾地继续聚会、旅游等，对疫情的防控放松，所以基本没什么研究。

## 1.3 主要研究内容

疫情封控管理系统的目的是创建一个基于疫情防控相关的数据管理页面，能直观的了解疫情封控的数据信息，[随时随地](http://checkonline.paperyy.com/api/write/true)都可以进行提交和管理数据，大大提高了数据的统计效率，减少了人力、物力资源的浪费。

本设计内容包括创日常防控管理、异常人员管理、防疫物质管理和疫情概况四个部分。日常防控管理中，用户可以自行对人员的姓名、性别、联系方式、体温、是否到达过高风险地区、健康状况、打卡时间和所属部门的打卡记录进行添加、修改和删除。异常人员管理中用户可以自行对疑似/确诊人员和隔离人员的性别、身份证号码、就诊（隔离）时间、就诊医院（隔离地点）、所属部门和联系方式等的数据进行添加、修改和删除。防疫物质管理中用户可以自行对防疫物资的数量、名称和负责人的数据信息进行添加、修改和删除。疫情概况中将上述数据的信息进行可视化处理。

## 1.4 论文组织结构

本论文主要是对疫情封控管理系统的设计与研究，并对其进行了编程和测试，从而实现用户通过前端页面授权登录就可对人员和物资进行管理。

全文共分七章，每一章的内容如下：

第1章 绪论：介绍了疫情封控管理系统现在的背景和意义，以及疫情封控管理系统的研究现状、主要内容以及本文的组织结构。

第2章 关键技术介绍：对基于疫情的疫情封控管理系统所使用的技术进行了简单地介绍。

第3章 系统需求分析：重点分析了基于疫情的疫情封控管理系统各个模块的需求，并结合用例图对其功能进行了详细地说明。

第4章 系统设计：根据第3章中系统的需求分析对本系统进行设计，并对其体系结构设计、功能详细设计、数据库设计等方面进行了详细地介绍。

第5章 系统实现：对疫情封控管理系统的各个模块进行了详细地介绍，并给出了各模块的运行效果图。

第6章 系统测试：对系统的功能和性能进行了测试，并给出了在测试过程中使用的一些测试方法。

第7章 总结和展望：对本系统的最终实现的效果进行了总结，并对未来的研究工作进行了展望。

# 第2章 关键技术介绍

本系统采用 IntelliJ IDEA开发工具和vscode开发者工具，利用 Java语言设计开发了基于疫情的疫情封控管理系统。本章主要介绍了Spring Boot框架，Vue框架，MySQL数据库等开发所需的技术。

## 2.1 Spring Boot框架

Spring Boot是Pivotal团队为新Spring软件设计的一个全新的框架。Spring Boot是以 Spring4.0为基础的，它不但保留了Spring框架的优良功能，同时也简化了 Spring的构建和开发过程。另外， Spring Boot在很多方面都做得很好，比如把很多框架结合起来，从而避免了软件的不同版本之间的冲突[2]。

Spring Boot框架有两个关键的战略：开箱即用和约定优于配置。开箱即用就是在开发期间，向 MAVEN项目的 pom文件中添加相关的依赖包，并且用对应的注释取代了冗长的 XML配置文件，从而管理对象的生命周期[3]。该特性使开发人员不必再进行复杂的配置工作，而是能将精力集中在业务逻辑上。约定优于配置，即 Spring Boot自身对目标结构进行配置，这样开发者就不用在配置和逻辑业务之间来回转换，而是将全部精力都放在了写逻辑业务上，从而大大提高了开发的效率[4]。这个功能在某种程度上降低了灵活性，增加了 BUG的位置，但它可以减少开发人员在代码编译、测试和打包等方面的自动化，同时也减少了 XML的配置[5]。

## 2.2 Vue框架

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue#libraries--plugins)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue.js 的目标是通过尽可能简单的 API 实现响应的[数据绑定](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%91%E5%AE%9A/2884939)和组合的视图组件。

Vue.js 自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习，非常容易与其它库或已有项目整合。另一方面，在与相关工具和支持库一起使用时 [2]  ，Vue.js 也能驱动复杂的单页应用。

## 2.3 Java开发语言

Java语言是一种典型的面向对象的静态编程语言，具有很强的实用性和易用性，它可以很好地实现面向对象理论，使编程人员能够很容易地进行复杂的编程[10]。

Java语言的语法简单明了，容易掌握，而且是纯面对对象的语言。Java是一种随着时代的飞速发展而发展起来的计算机编程语言。它能为互联网应用和多媒体访问提供支持，从而促进互联网和企业级的 Web应用[11]。

## 2.4 MySQL数据库

MySQL是一个安全、跨平台、高效的关系型数据库管理系统，它与PHP、Java等主流编程语言结合在一起[12]。MySQL使用非常方便，易于理解，功能强大，所以大多数用户选择使用它。而且随着技术的不断完善，MySQL将会带来更多的功能，更好的性能，以及更多的平台[13]。

MySQL相对于其它关系型数据库具有自己的优越性。MySQL是 SQL数据库中最快速的一种。而且 MySQL数据库是一款完全免费的软件，用户不需要付费就能在网上下载 MySQL数据库[14]。基于以上作者选用MySQL数据库作为系统的数据库。

## 2.5 Redis

redis是一个key-value[存储系统](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E7%B3%BB%E7%BB%9F)。它支持存储的value类型相对更多，包括string(字符串)、list([链表](https://baike.baidu.com/item/%E9%93%BE%E8%A1%A8))、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些[数据类型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B)都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。为了保证效率，数据都是缓存在内存中。

Redis 是一个高性能的key-value数据库。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。 [1]

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。

## 2.6 ECharts

ECharts是一款基于[JavaScript](https://baike.baidu.com/item/JavaScript/321142)的[数据可视化](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%8F%AF%E8%A7%86%E5%8C%96/1252367)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。

ECharts 提供了常规的折线图、柱状图、散点图、饼图、K线图，用于统计的盒形图，用于地理数据可视化的地图、热力图、线图，用于关系数据可视化的关系图、treemap、旭日图，多维数据可视化的平行坐标，还有用于 BI 的漏斗图，仪表盘，并且支持图与图之间的混搭。

## 2.7 MyBatis-Plus

[MyBatis-Plus (opens new window)](https://github.com/baomidou/mybatis-plus)（简称 MP）是一个 [MyBatis (opens new window)](https://www.mybatis.org/mybatis-3/)的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。其操作都不需要通过SQL语句进行，就可以减少MyBatis写大量的xml文件或者注解的麻烦。

## 2.8 Element UI

Element UI 是基于 vue 实现的一套不依赖业务的 UI 组件库，提供了丰富的PC端组件，减少用户对常用组件的封装，降低了开发的难易程度。

Element UI是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，了解设计指南,帮助产品设计人员搭建逻辑清晰、结构合理且高效易用的产品。提供了配套设计资源，帮助你的网站快速成型。由饿了么公司前端团队开源。

## 2.9 Vue Router

vue-router是Vue.js官方的路由插件，它和vue.js是深度集成的，适合用于构建单页面应用。vue的单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来。传统的页面应用，是用一些超链接来实现页面切换和跳转的。在vue-router单页面应用中，则是路径之间的切换，也就是组件的切换。路由模块的本质 就是建立起url和页面之间的映射关系。

## 2.10 Vuex

Vuex 是一个专为 Vue.js 应用程序开发的[状态管理](http://www.javanx.cn/tag/%e7%8a%b6%e6%80%81%e7%ae%a1%e7%90%86/)模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。Vuex 也集成到 Vue 的官方调试工具 devtools extension，提供了诸如零配置的 time-travel 调试、状态快照导入导出等高级调试功能。简单的说：Vuex是[vue](http://www.javanx.cn/tag/vue/)框架中状态管理。

## 2.11 Vue-cli

vue-cli 是一个官方发布 vue.js 项目脚手架，一个专门为单页面应用快速搭建繁杂的脚手架，它可以轻松的创建新的应用程序而且可用于自动生成vue和webpack的项目模板。

vue-cli这个构建工具大大降低了webpack的使用难度，支持热更新，有webpack-dev-server的支持，相当于启动了一个请求服务器，给你搭建了一个测试环境，只关注开发就OK。

## 2.12 本章小结

本章主要介绍了本论文所研究的疫情封控管理系统使用的主要技术框架，包括Vue框架、Spring Boot框架。在分析了以上技术框架后，明确了该系统主要采用Spring Boot框架对疫情封控管理系统进行开发。

# 第3章 系统需求分析

系统需求分析是系统开发前期的铺垫工作，也是重要的基础工作之一，有效地进行分析，可以帮助我们避免或及早地排除早期的错误，从而降低开发成本，提高系统开发的成功率。本章主要讲述疫情封控管理系统的需求，根据本系统的所划分的四大模块进行分析，分别为日常防控管理、异常人员管理、防控物资管理、疫情概况。

## 3.1 需求概述

本系统分别从各个业务功能上进行相关的概述。

（1）日常防控管理功能：本模块是管理者对健康打卡的申报和打卡记录信息的查看，包括打卡记录的修改、搜索与删除，健康打卡申报的创建与重置的设置。

（2）异常人员管理功能：本模块是管理者对嫌疑/确诊人员登记和隔离人员登记的信息查看，包括嫌疑/确诊人员登记和隔离人员登记的新增记录、修改、搜索与删除设置。

（3）防控物资管理功能：本模块是管理者对防疫物资的新增和防疫物资信息的查看，包括防疫物资查看的修改、搜索与删除，新增防疫物资的创建与重置的设置。

（4）疫情概况功能：该模块的主要功能是数据可视化，包含了对物资情况的查看及其疫情趋势的查看和下载，让管理者能够更直接地查看疫情的趋势和物资的数目。

对上述四大功能进行详细阐述，绘制出各个功能的用例图，并结合文字说明，对各功能模块进行了需求分析，使得整个开发过程和思路更加明确。

## 3.2 功能需求分析

本系统的使用者只有：管理员。管理者进行防疫数据的查看；功能上划分为四个模块，分别是：日常防控管理功能、异常人员管理功能、防控物资管理功能、疫情概况功能。

### 3.2.1 日常防控管理

日常防控管理功能模块是本系统的一项基本业务，管理者通过授权登录后即可自行进行健康打卡的申报和打卡记录信息的查看，对打卡记录进行修改、搜索与删除，健康打卡申报进行创建与重置

日常防控管理功能用例图如图3-1所示。

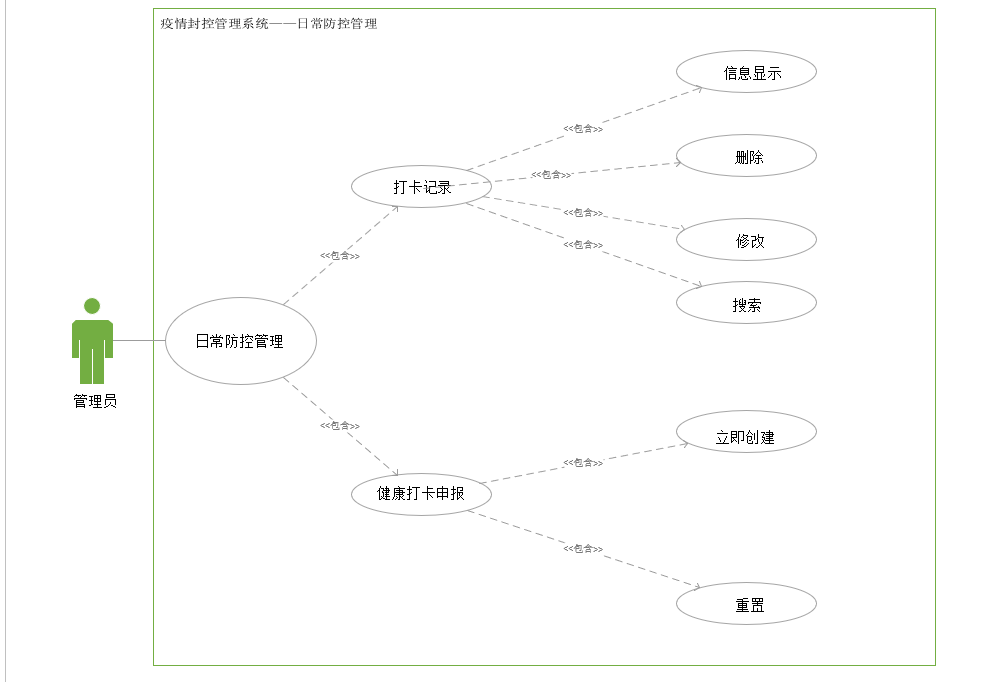


图3-1 日常防控管理功能用例图

### 3.2.2 异常人员管理

异常人员管理功能模块也是本系统的一项基本业务，管理者通过授权登录后即可自行进行嫌疑/确诊人员登记和隔离人员登记的信息查看，对嫌疑/确诊人员登记和隔离人员登记进行新增记录、修改、搜索与删除

异常人员管理功能用例图如图3-2所示。

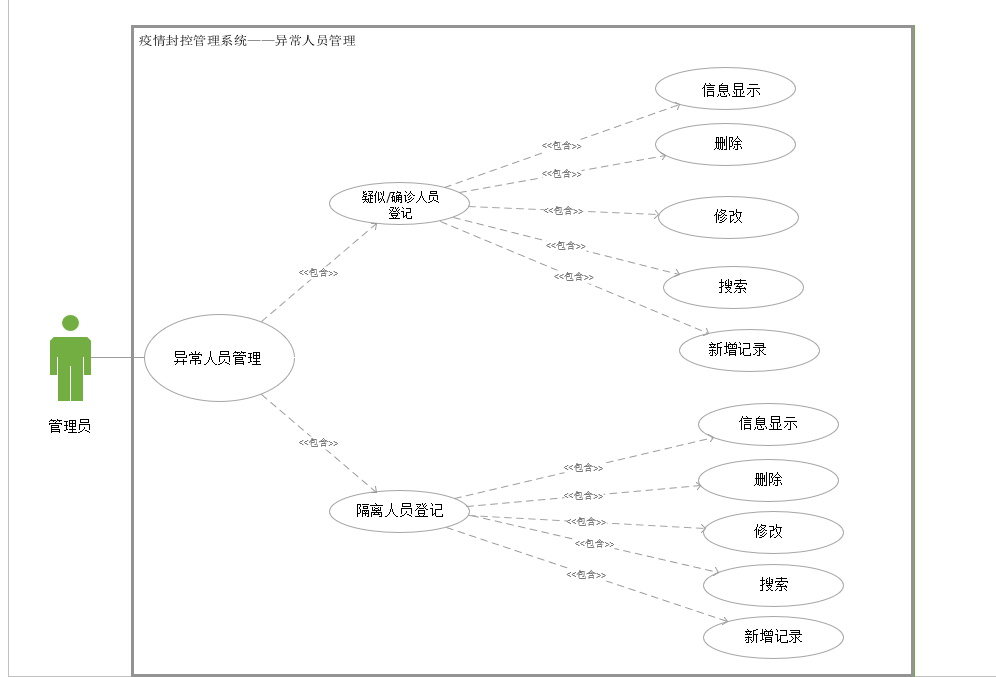


图3-2 异常人员管理功能用例图

### 3.2.3 防控物资管理

防控物资管理功能模块是本系统的一项基本业务，管理者通过授权登录后即可自行进行防疫物资的新增和防疫物资信息的查看，对防疫物资查看进行修改、搜索与删除，新增防疫物资进行创建与重置

防控物资管理功能用例图如图3-3所示。

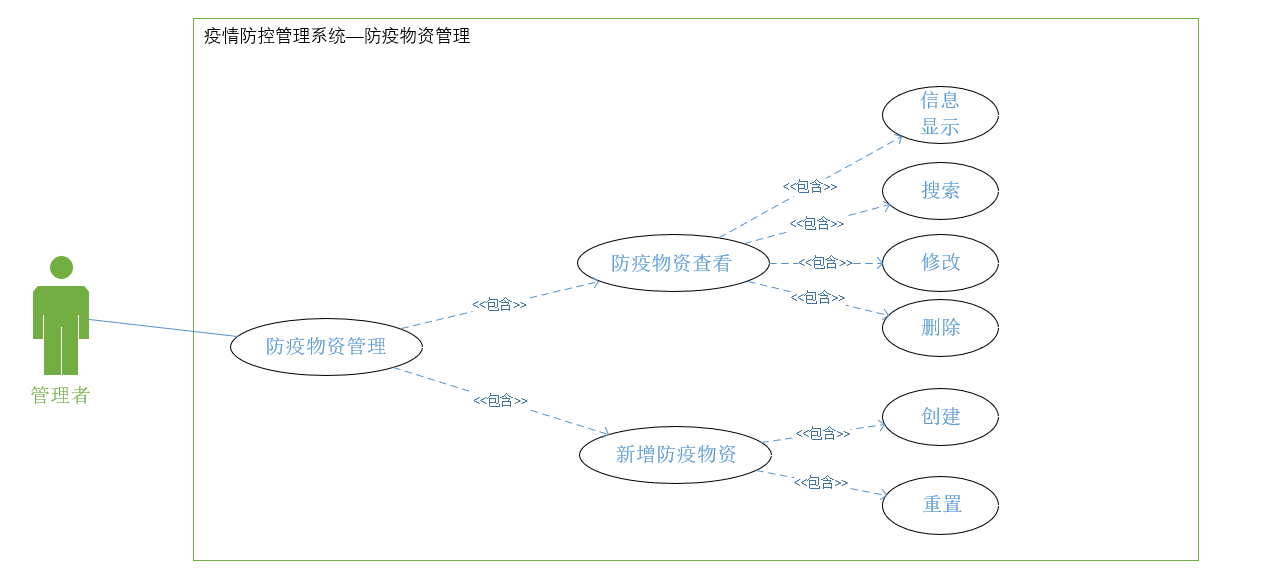


图3-3 防控物资管理功能用例图

### 3.2.4 疫情概况功能

为了更直观的查看防疫物资的结果和疫情趋势，系统对防疫数据进行了数据可视化功能。

疫情概况功能用例图如图3-5所示。

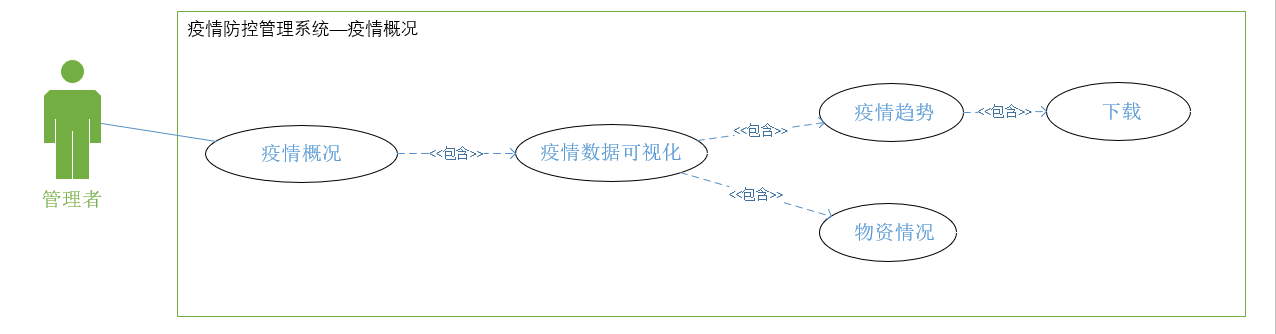


图3-5 疫情概况功能用例图

## 3.3 本章小结

本章主要从疫情封控管理系统的功能需求进行详细的分析，介绍了本系统初步具有的功能，从日常防控管理、异常人员管理、防控物资管理、疫情概况模块进行功能的需求分析，画出了系统四大功能模块的用例图。

# 第4章 系统设计

本章将对该系统进行详细的设计。通过流程图、时序图、文本说明等方法对系统进行了功能设计。体系架构设计与设计图相结合，从整体架构、逻辑架构、网络架构、功能架构四个方面进行了阐述。在此基础上，本文还将从概念结构和逻辑结构设计两个角度，分别用实体属性图、E-R图以及三线表加以说明。

## 4.1 体系架构设计

本节主要从整体架构、逻辑架构、网络架构、功能架构四个部分来说明疫情封控管理系统体系架构的设计。

### 4.1.1 整体架构设计

本小节主要是描述疫情封控管理系统的工作方式及工作流程，可为具体的开发工作提供参考。整体架构设计如图4-1所示。

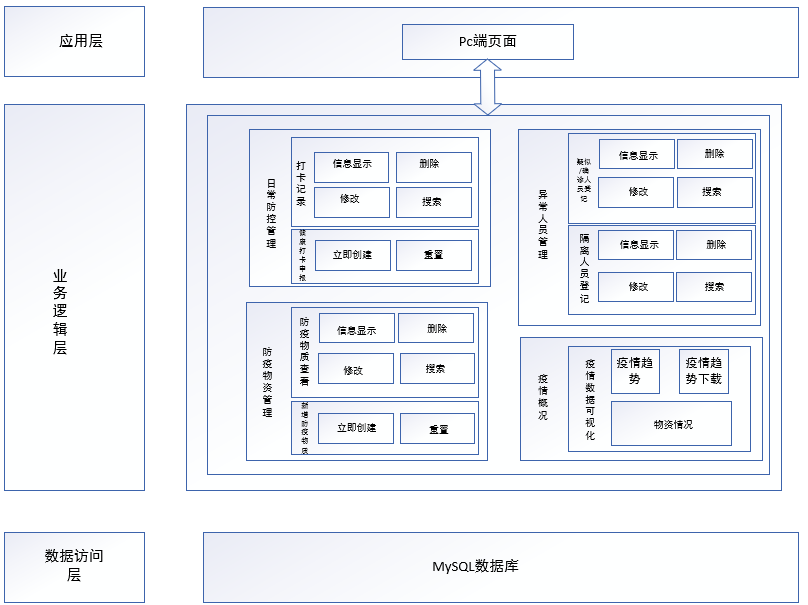


图4-1 整体架构设计图

### 4.1.2 逻辑架构设计

疫情封控管理系统的逻辑架构设计包含四部分：

使用者：管理者。

使用平台：PC端页面。

功能应用：主要为管理者通过前端页面登录进行打卡管理、人员管理、查看物资和疫情概况的信息查看并对其进行增删改查管理。

运行环境：服务器，数据库。

逻辑架构设计如图4-2所示。

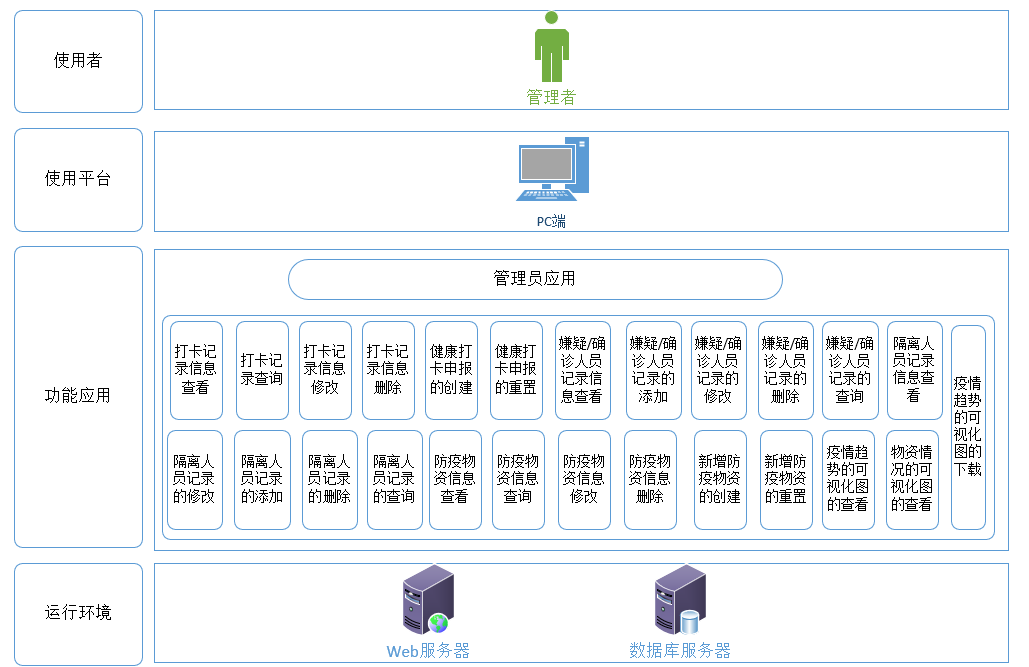


图4-2 逻辑架构设计图

### 4.1.3 网络架构设计

管理者通过浏览器访问登录页面，向服务器发送请求，服务器接收请求并做出响应，把用户需要的数据返回到前端页面。只要有一台装有浏览器的笔记本或台式电脑，就能进行操作

网络架构设计如图4-3所示。

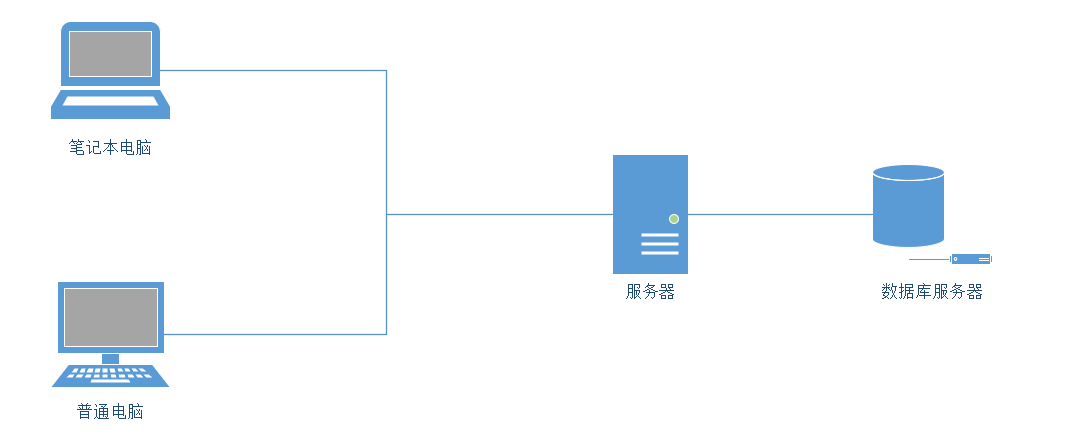


图4-3 网络架构设计图

### 4.1.4 功能架构设计

本节按照上一章的需求分析进行设计。该系统分为四个功能模块：日常防控管理、异常人员管理、防疫物资管理、疫情情况功能。

基于疫情的疫情封控管理系统的功能架构设计如图4-4所示。

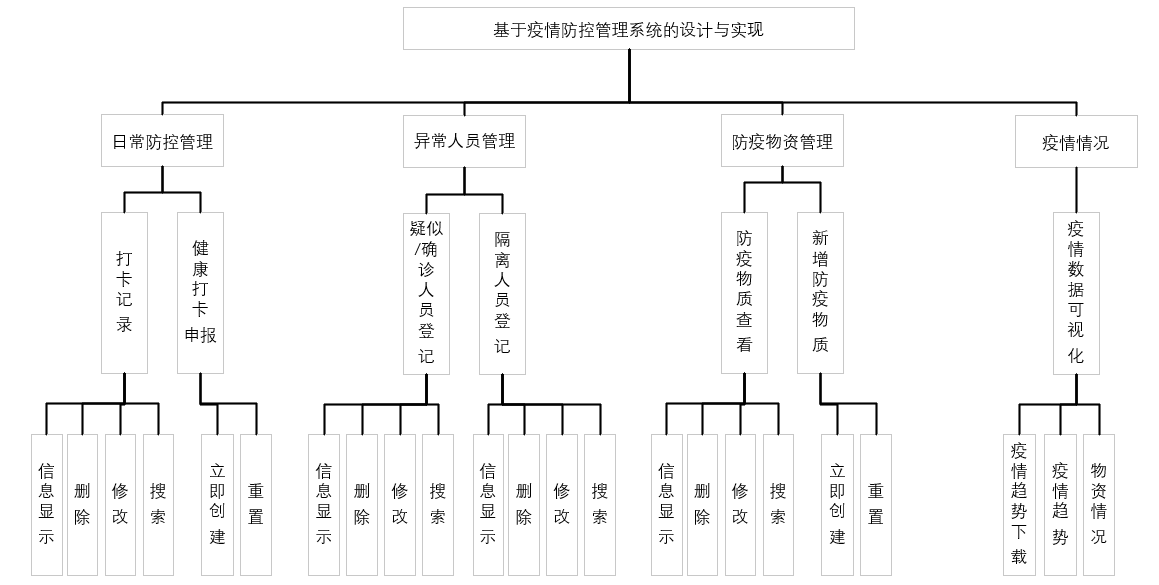


图4-4 功能架构设计图

## 4.2 功能详细设计

本节主要描述各个功能实现流程的详细设计。疫情封控管理系统在系统需求分析部分主要分为四大模块，分别为日常防控管理、异常人员管理、防控物资管理、疫情概况。下面将对该系统进行分析，得到具体的功能设计的操作流程。

### 4.2.1 日常防控管理

管理者先登录，登录成功后即可进行日常防控管理。管理者在日常防控管理主界面可以对打卡记录进行修改、搜索与删除，健康打卡申报进行创建与重置的设置。当管理者点击打卡记录时，系统会将数据库中数据显示到主页面中，当你查询时，系统会根据你输入的信息进行查找，并重新加载数据，当点击删改时，系统会将你输入的数据存储到数据库中，同时也会向管理者发送一个修改或删除成功的提示，修改或删除成功则重新加载数据。当管理者点击健康打卡申报时，显示打卡申报信息界面，填写完成当你点击创建时，系统会将你输入的数据存储到数据库中，同时也会向管理者发送一个创建成功的提示，创建成功则重新加载数据。

日常防控管理功能时序图如图4-5所示。

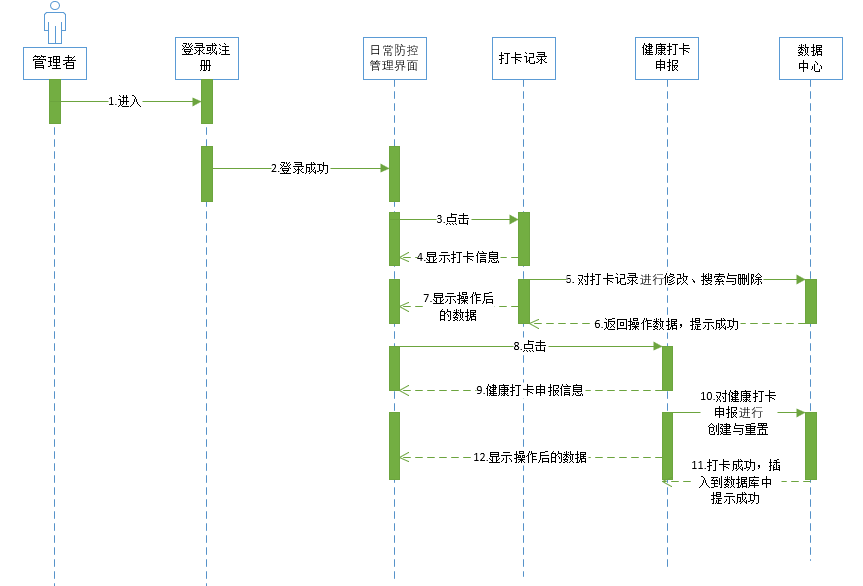


图4-5 日常防控管理功能时序图

日常防控管理模块的业务流程图如图4-6所示。

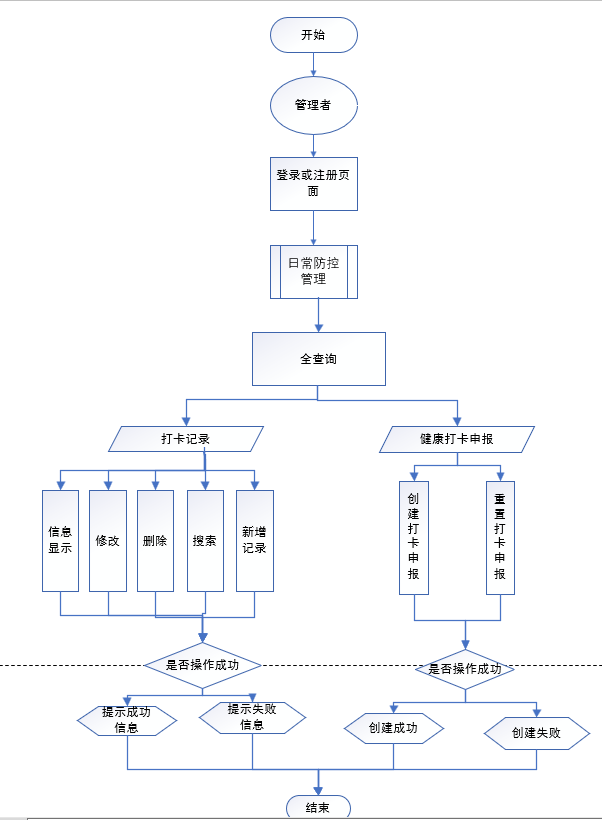


图4-6 日常防控管理功能业务流程图

### 4.2.2 异常人员管理

管理员进入登录注册首页，登录后进入首页点击异常人员管理，疑似/确诊人员的详情页显示疑似/确诊人员的详细信息。可以对疑似/确诊人员和隔离人员的数据进行操作。点击新增记录后弹出新增窗口将疑似/确诊人员或隔离人员的数据填写完后点击确定，就能将数据写入数据库。并在疑似/确诊人员和隔离人员的详情页显示疑似/确诊人员和隔离人员的信息。点击删除时会提示是否删除该数据的窗口，点击确定后数据库将删除该数据。点击修改时弹出修改窗口将疑似/确诊人员或隔离人员的数据填写完后点击确定，就能将数据库的该条数据进行修改，并在疑似/确诊人员和隔离人员的详情页显示疑似/确诊人员和隔离人员的信息。

异常人员管理功能时序图如图4-7所示。

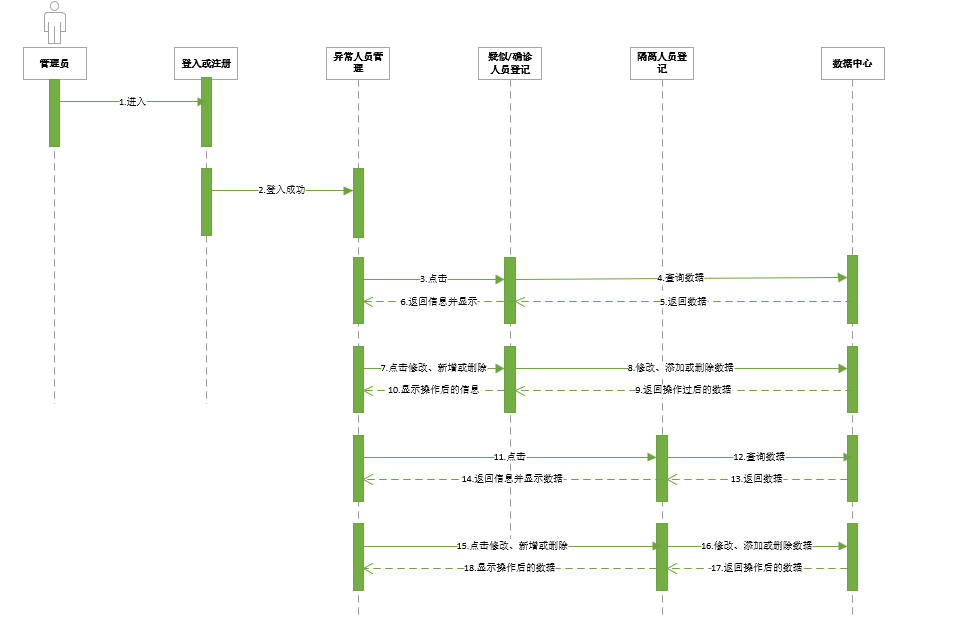


图4-7 异常人员管理功能时序图

异常人员管理模块的业务流程图如图4-8所示。

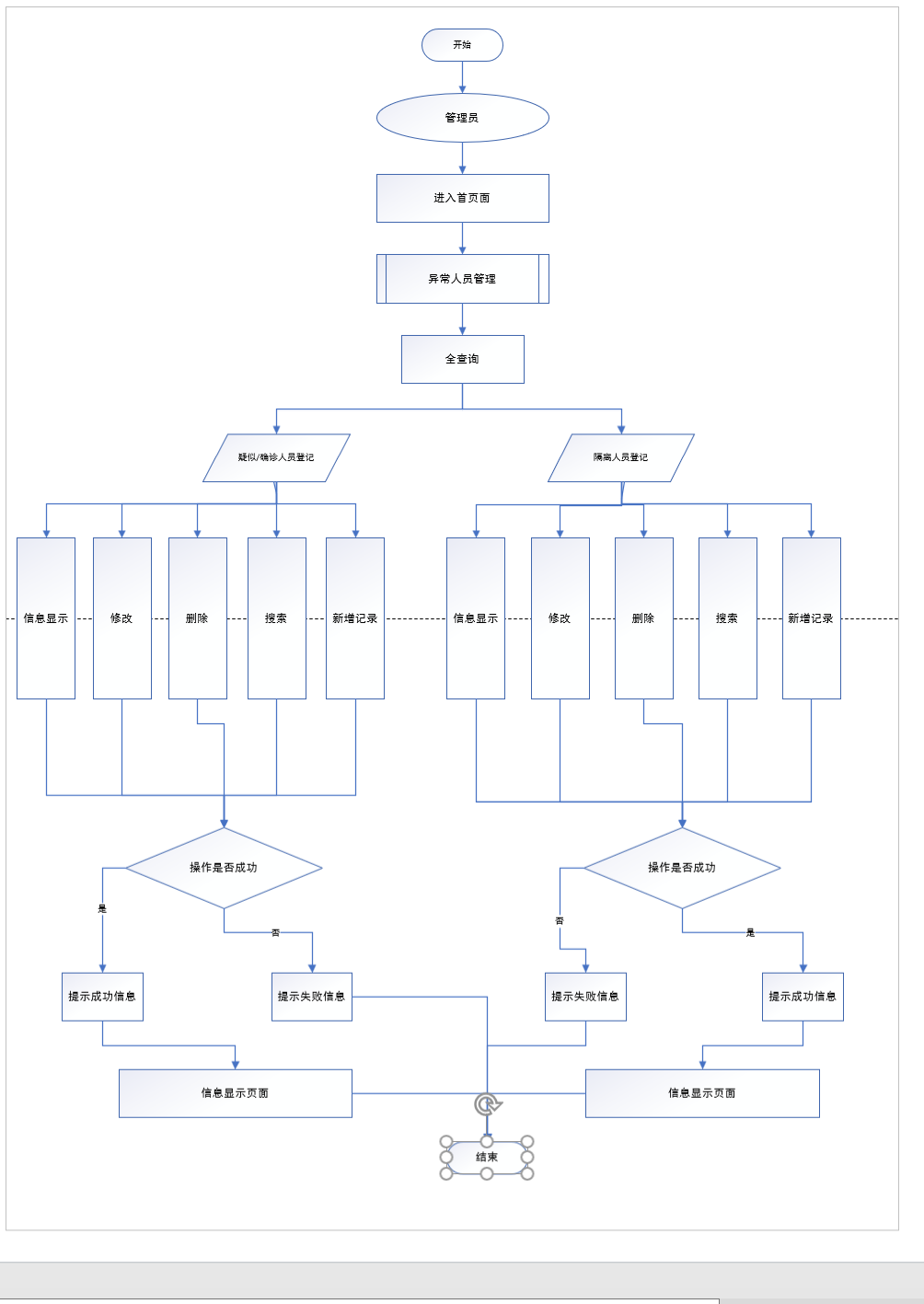


图4-8 异常人员管理功能业务流程图

### 4.2.3 防控物资管理

管理者先登录，登录成功后即可进行日常防控管理。管理者在防控物资管理主界面可以对防疫物资查看的修改、搜索与删除，新增防疫物资的创建与重置。当管理者点击防疫物资查看时，系统会将数据库中数据显示到主页面中，当你查询时，系统会根据你输入的信息进行查找，并重新加载数据，当点击删改时，系统会将你输入的数据存储到数据库中，同时也会向管理者发送一个修改或删除成功的提示，修改或删除成功则重新加载数据。当管理者点击新增防疫物资时，显示思新增信息界面，填写完成当你点击创建时，系统会将你输入的数据存储到数据库中，同时也会向管理者发送一个创建成功的提示，创建成功则重新加载数据。

防控物资管理功能时序图如图4-9所示。

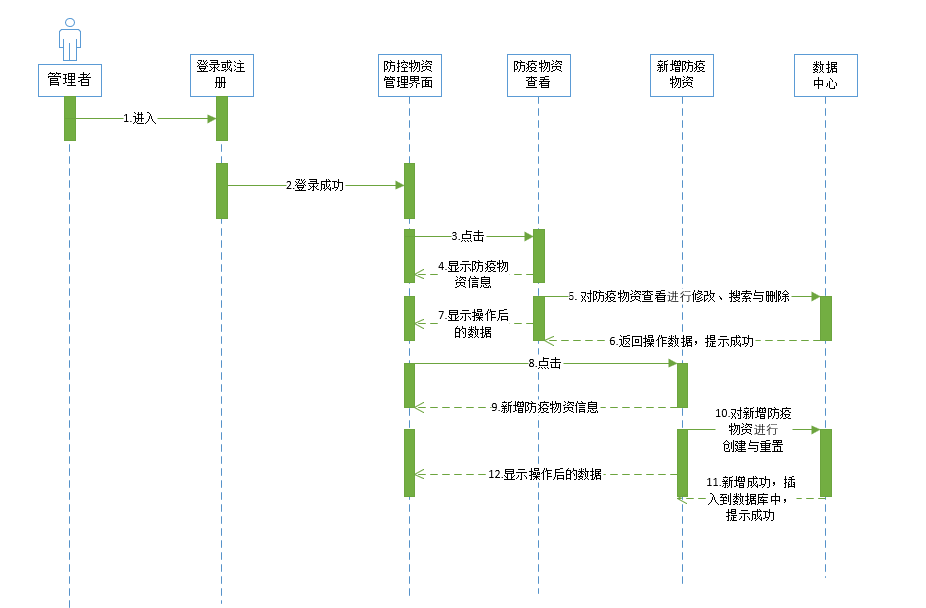


图4-9 防控物资管理功能时序图

防控物资管理模块的业务流程图如图4-10所示。

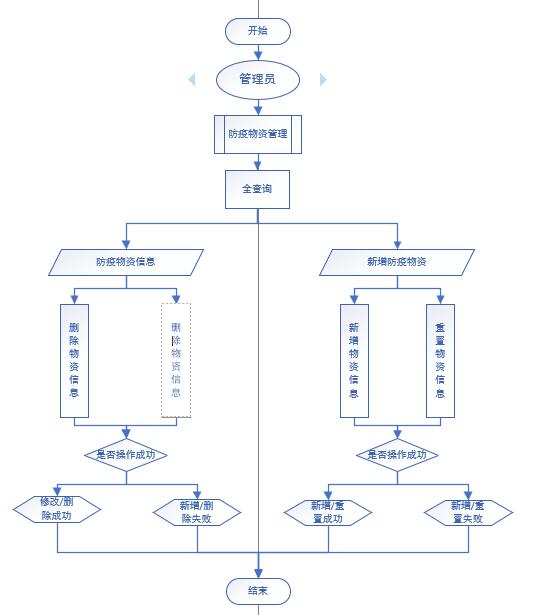


图4-10 防控物资管理功能业务流程图

### 4.2.4 疫情概况

疫情概况模块是为了让管理者能够更直接地查看疫情的趋势和物资的数目，及可以显示数据，也可以下载数据。

疫情概况功能时序图如图4-13所示。

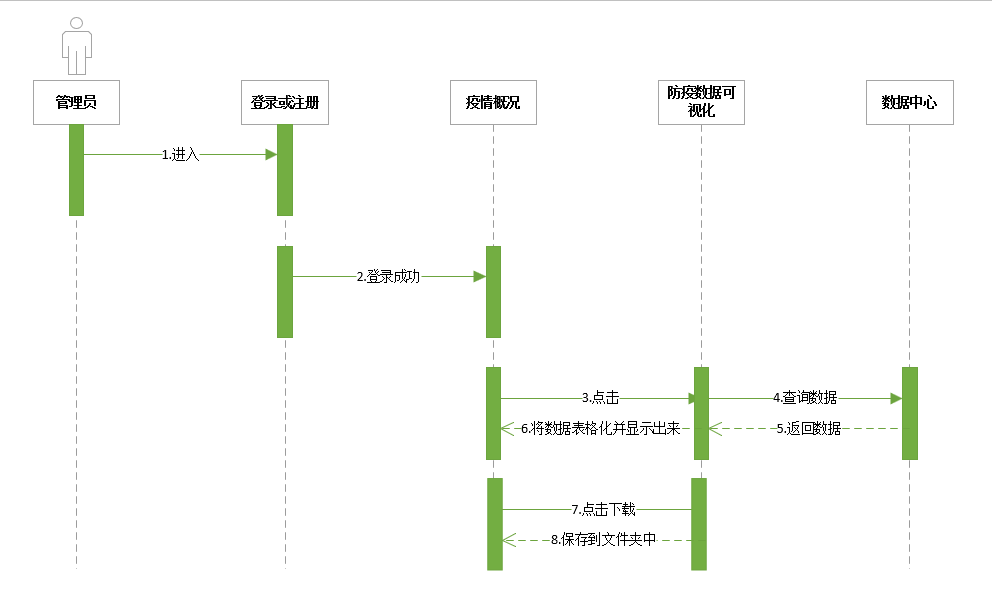


图4-13 疫情概况功能时序图

疫情概况模块的业务流程图如图4-14所示。

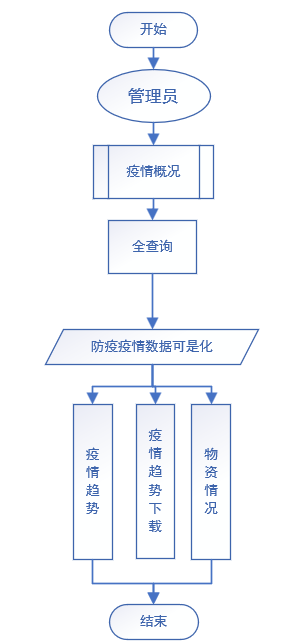


图4-14 疫情概况功能业务流程图

## 4.3 数据库设计

本小节主要介绍了数据库的设计，根据对疫情封控管理系统的分析，设计所需的数据表，并且通过数据的交互，找到各个表的相互关系，从而得到整个数据库的结构。下面将从数据库的概念结构设计和逻辑结构设计两个部分，结合实体属性图、E-R图和库表结构对数据库进行介绍。

### 4.3.1 概念结构设计

根据需求分析，疫情封控管理系统设计到以下实体：打卡记录实体、嫌疑/确诊人员登记实体、隔离人员登记实体、防疫物资管理实体、登录注册实体、编码信息实体。

（1）打卡记录信息表emp\_health，该表可储存打卡人员的基本信息，打卡记录的基本信息包括：编号id，名称name，性别sex，电话号码phonenum，体温temp，是否经过高风险地区risk，健康状况health，备注content，打卡时间createTime，所属部门depart。

打卡记录信息表emp\_health的实体属性图如图4-15所示。

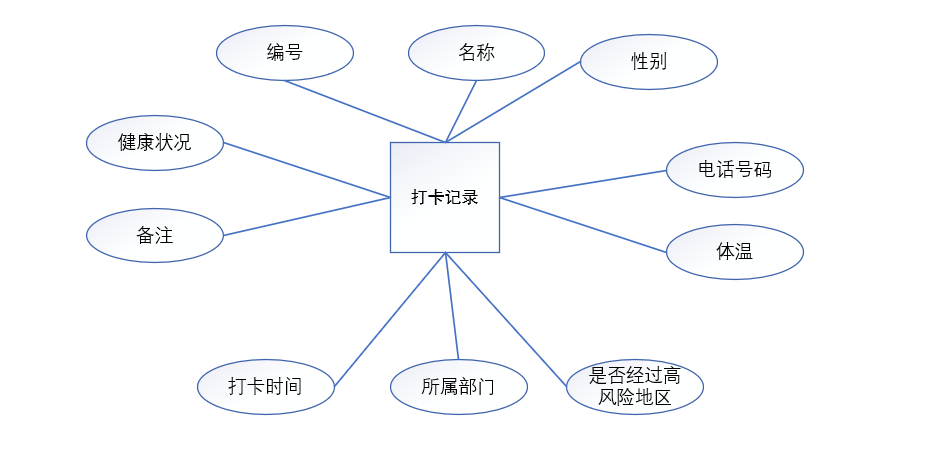


图4-15 打卡记录信息实体属性图

（2）疑似/确诊人员登记信息表emp\_iden，该表可以存储疑似/确诊人员的基本信息，疑似/确诊人员的基本信息包括：编号id，名称name，性别sex，身份证号码idcard，就诊日期idate，就诊医院place，手机号码phonenum，登记时间register，病历status，所属部门depart。

疑似/确诊人员登记信息表emp\_iden的实体属性图如图4-16所示。

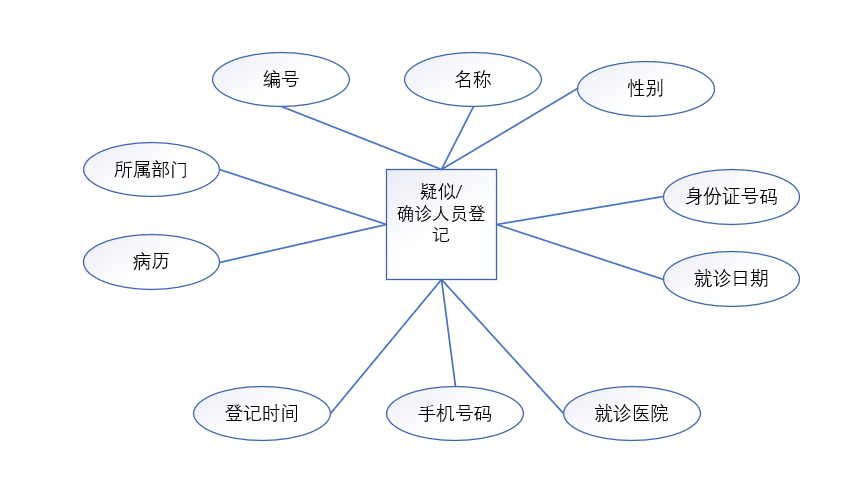


图4-16 疑似/确诊人员登记信息实体属性图

（3）隔离人员登记信息表emp\_is，该表可以存储隔离人员的基本信息, 隔离人员的基本信息包括：编号id，名称name，性别sex，手机号码phone，体温temp，隔离方式type，隔离地点place，开始隔离时间begin，结束隔离时间end，出发地leaved，途径地arrived，备注content，所属部门depart。

隔离人员登记信息表emp\_is的实体属性图如图4-17所示。

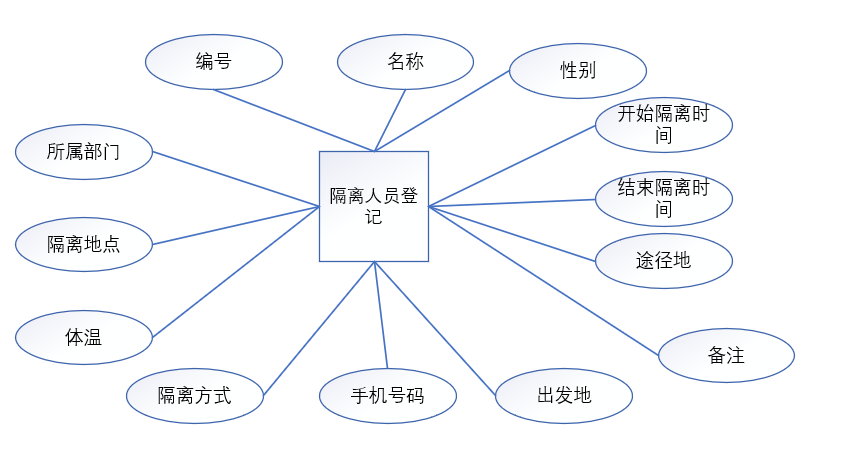


图4-17 隔离人员登记信息实体属性图

（4）防疫物资管理信息表material\_manage，该表可以存储防疫物资的基本信息，防疫物资的基本信息包括：物资编号id，名称name，数量count，计量单位type，是否为重点物资isImp，负责人charge，负责人电话cnum，更新时间updateTime。

防疫物资管理信息表material\_manage的实体属性图如图4-18所示。

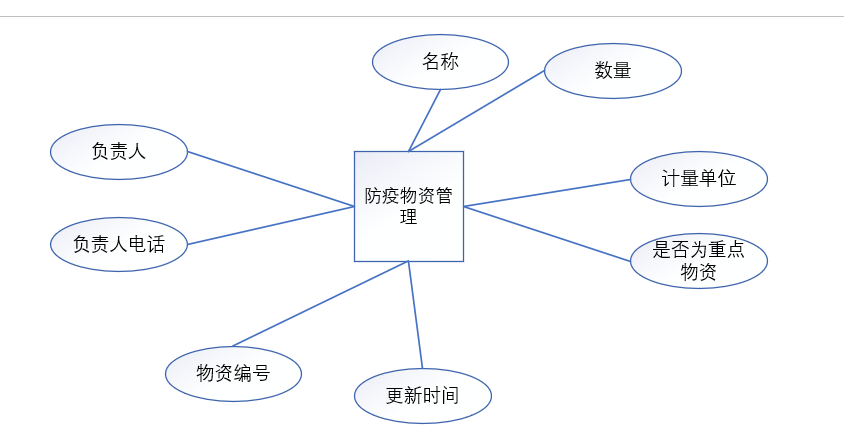


图4-18 防疫物资管理信息实体属性图

（5）管理员登录信息表user，该表可以存储管理员登录的账号和密码的详细信息，管理员登录的详细信息包括：编号id，用户名username，密码password，所属部门depart。

管理员登录信息表user的实体属性图如图4-19所示。

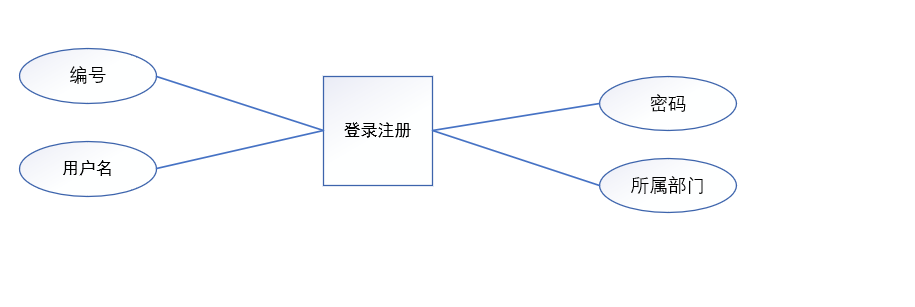


图4-19 管理员登录信息实体属性图

（6）部门信息表department，该表可以存储部门的基本信息，部门的基本信息包括：部门编号id，部门名称name，负责人charge。

部门信息表department的实体属性图如图4-20所示。

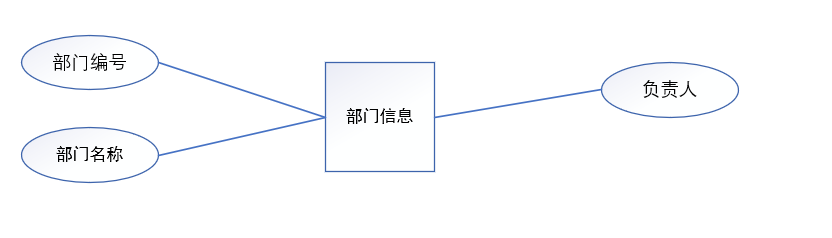


图4-20 部门信息实体属性图

（7）数据库总体结构E-R模型图如图4-24所示。

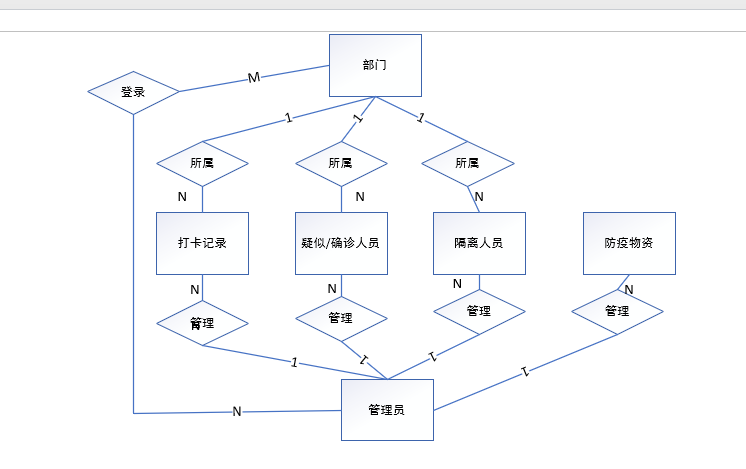


图4-24 数据库总体E-R图

### 4.3.2 逻辑结构设计

数据库的逻辑结构设计主要是把E-R图转化为数据库所需的表，从而实现E-R模型向关系模型的转化，具体表结构如下。

（1）打卡记录信息表emp\_health，用来存储打卡人员的基本信息，其相应的字段信息如表4-1所示。

表4-1 打卡记录信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | id | 编号 | int(10) | 是 |
| 2 | name | 名称 | varchar(20) | 否 |
| 3 | sex | 性别 | int(1) | 否 |
| 4 | phonenum | 电话号码 | bigint(20) | 否 |
| 5 | temp | 体温 | float(4,1) | 否 |
| 6 | risk | 是否经过高风险地区 | varchar(30) | 否 |
| 7 | health | 健康状况 | varchar(30) | 否 |
| 8 | content | 备注 | varchar(255) | 否 |
| 9 | createTime | 打卡时间 | datetime | 否 |

（2）疑似/确诊人员登记信息表emp\_iden，用来存储疑似/确诊人员的基本信息，其相应字段信息如表4-2所示。

表4-2 疑似/确诊人员登记信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | id | 编号 | bigint(20) | 是 |
| 2 | name | 名称 | varchar(10) | 否 |
| 3 | sex | 性别 | int(2) | 否 |
| 4 | idcard | 身份证号码 | varchar(20) | 否 |
| 5 | idate | 就诊日期 | date | 否 |
| 6 | place | 就诊医院 | varchar(100) | 否 |
| 7 | phonenum | 手机号码 | bigint(100) | 否 |
| 8 | register | 登记时间 | datetime | 否 |
| 9 | status | 病历 | varchar(10) | 否 |

续表4-2 疑似/确诊人员登记信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 10 | depart | 所属部门 | varchar(20) | 否 |

（3）隔离人员登记信息表emp\_is，用来存储隔离人员的基本信息，其相应字段信息如表4-3所示。

表4-3 隔离人员登记信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | id | 编号 | bigint(20) | 是 |
| 2 | name | 名称 | varchar(20) | 否 |
| 3 | sex | 性别 | int(3) | 否 |
| 4 | phone | 手机号码 | bigint(20) | 否 |
| 5 | temp | 体温 | float(10,2) | 否 |
| 6 | type | 隔离方式 | varchar(10) | 否 |
| 7 | place | 隔离地点 | varchar(100) | 否 |
| 8 | begin | 开始隔离时间 | datetime | 否 |
| 9 | end | 结束隔离时间 | datetime | 否 |
| 10 | leaved | 出发地 | varchar(100) | 否 |
| 11 | arrived | 途径地 | varchar(100) | 否 |
| 12 | content | 备注 | varchar(255) | 否 |
| 13 | depart | 所属部门 | varchar(20) | 否 |

（4）防疫物资管理信息表material\_manage，用来存储防疫物资的基本信息， 其相应字段信息如表4-4所示。

表4-4 防疫物资管理信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | id | 物资编号 | int(20) | 是 |
| 2 | name | 名称 | varchar(255) | 否 |
| 3 | count | 数量 | int(11) | 否 |

续表4-4 防疫物资管理信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 4 | type | 计量单位 | varchar(255) | 否 |
| 5 | isImp | 是否为重点物资 | int(11) | 否 |
| 6 | charge | 负责人 | varchar(255) | 否 |
| 7 | cnum | 负责人电话 | bigint(20) | 否 |
| 8 | updateTime | 更新时间 | datetime | 否 |

（5）管理员登录信息表user，用来存储管理员登录的账号和密码的详细信息，其相应字段信息如表4-5所示。

表4-5 管理员登录信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | id | id号 | int(11) | 是 |
| 2 | username | 用户名 | varchar(20) | 否 |
| 3 | password | 密码 | varchar(255) | 否 |
| 4 | depart | 所属部门 | varchar(20) | 否 |

（6）部门信息表department，用来存储部门的基本信息，其相应字段信息如表4-6所示。

表4-6 部门信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 描述 | 类型和长度 | 主键 |
| 1 | Id | 部门编号 | Int(11) | 是 |
| 2 | name | 部门名称 | varchar(20) | 否 |
| 3 | charge | 负责人 | varchar(20) | 否 |

## 4.4 本章小结

本章通过大量的图和表，对疫情封控管理系统的体系架构设计、功能详细设计和数据库设计进行了详细的说明。

# 第5章 系统实现

通过对疫情封控管理系统的深入的分析和设计，本系统分为日常防控管理功能模块、异常人员管理功能模块、防控物资管理功能模块以及疫情概况功能模块，本章将详细地介绍各个功能模块的实现。

## 5.1 日常防控管理模块

日常防控管理模块当管理者进入到前端登录注册页面后，登录后就可以进行健康打卡的申报和打卡记录信息的查看，包括打卡记录的修改、搜索与删除，健康打卡申报的创建与重置从而进行信息更新和查看。

日常防控管理的流程图如图5-1所示。

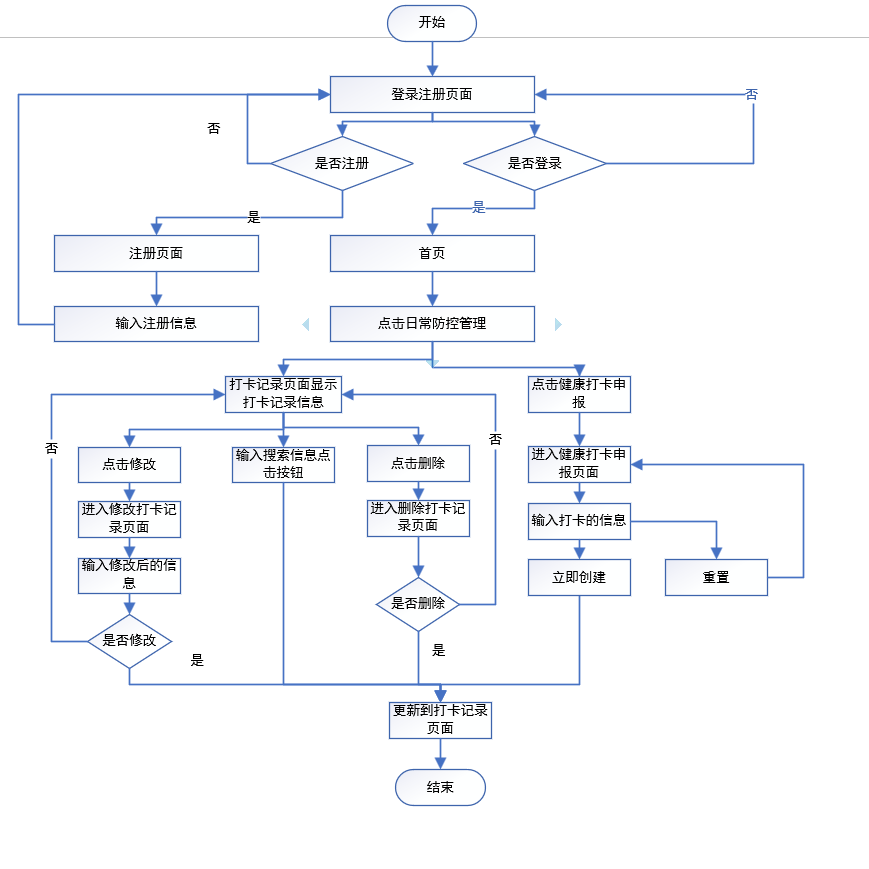


图5-1 日常防控管理模块流程图

打卡记录的核心代码如下：

@GetMapping("/findAll/{page}/{size}") //显示  
public Page<EmpHealth> findAll(@PathVariable("page") Integer page, @PathVariable("size") Integer size){  
 Object cacheObject = redisCache.getCacheObject(USER\_LOGIN\_STATE);  
 if (cacheObject!=null){  
 QueryWrapper<EmpHealth> wrapper=new QueryWrapper<>();  
 wrapper.orderByDesc("createTime");  
 Page<EmpHealth> page1= new Page<>(page,size);  
 Page<EmpHealth> result=mapper.selectPage(page1,wrapper).addOrder();  
 return result;  
 }  
 return null;  
}

@GetMapping("/findById/{id}") //id查找  
public EmpHealth findById(@PathVariable("id") Integer id){  
 return mapper.selectById(id);  
}  
@PutMapping("/update") //修改  
public String update(@RequestBody EmpHealth emp){  
 int result=mapper.updateById(emp);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@DeleteMapping("/deleteById/{id}") //删除  
public void deleteById(@PathVariable("id")Integer id){  
 mapper.deleteById(id);  
}  
@GetMapping("/search/{searchkey}/{stext}") //查询  
public List<EmpHealth> search(@PathVariable("searchkey")String searchkey, @PathVariable("stext")String stext){  
 QueryWrapper<EmpHealth> userQueryWrapper = Wrappers.query();  
 userQueryWrapper.like(searchkey,stext);  
 return mapper.selectList(userQueryWrapper);  
}

其主要界面如图5-2，5-3，5-4，5-5所示。



图5-2 打卡记录界面

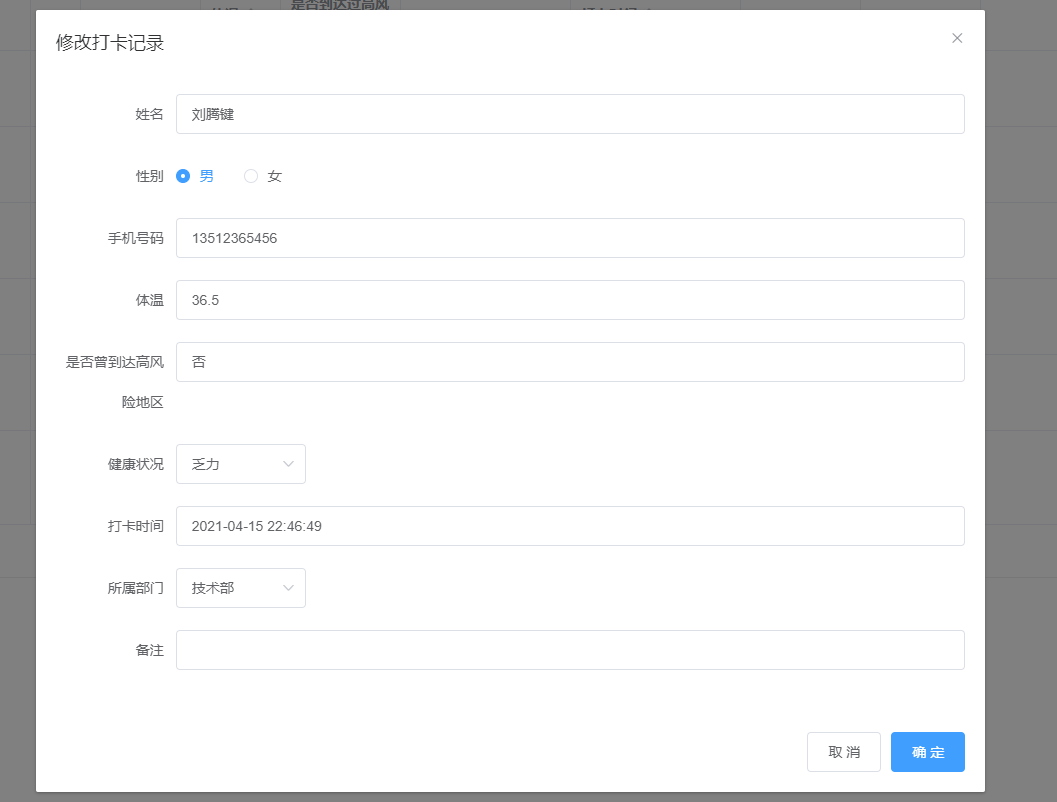


图5-3 打卡记录修改界面

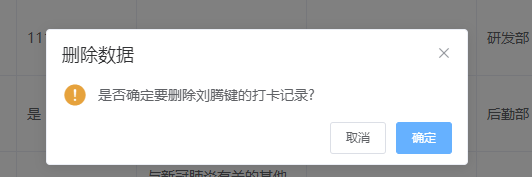


图5-4 打卡记录删除界面



图5-5 打卡记录查询界面

健康打卡申报的核心代码如下：

@PostMapping("/save")  
public String save(@RequestBody EmpHealth emp){  
 int result = mapper.insert(emp);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}

其主要界面如图5-6所示。

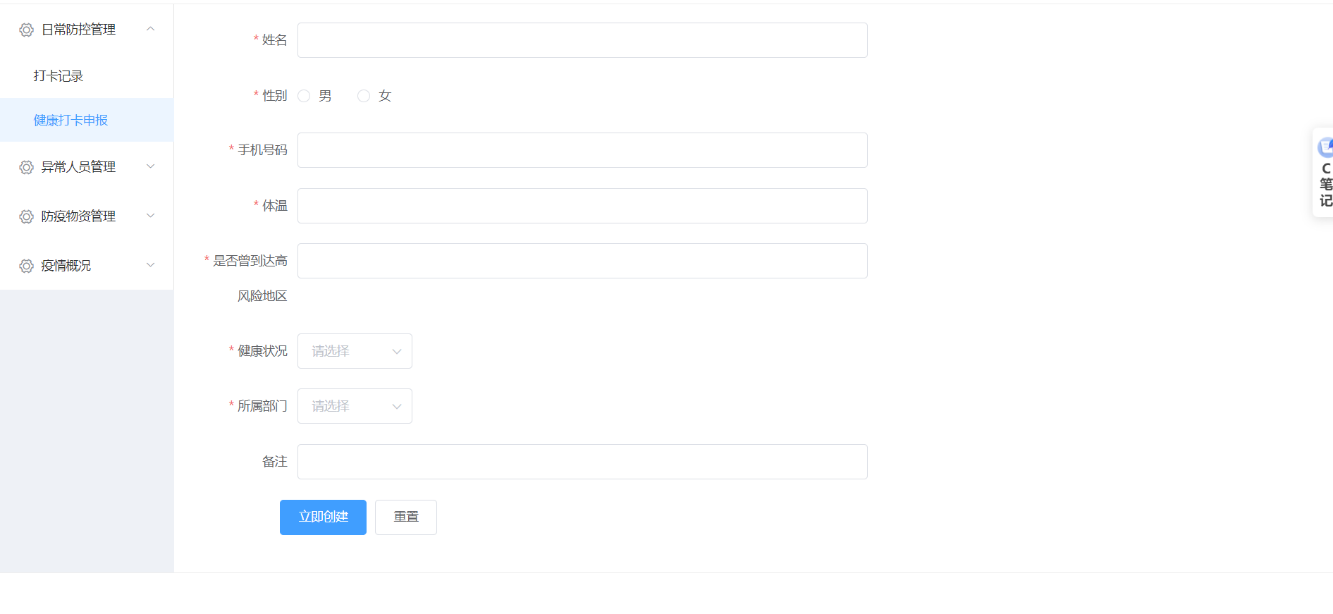


图5-6 健康打卡申报界面

## 5.2 异常人员管理模块

异常人员管理模块当管理员进入前端首页，点击异常人员管理，管理员就可以进行嫌疑/确诊人员登记和隔离人员登记的信息查看，包括嫌疑/确诊人员登记和隔离人员登记的新增记录、修改、搜索与删除从而进行信息更新和查看。

异常人员管理的流程图如图5-7所示。

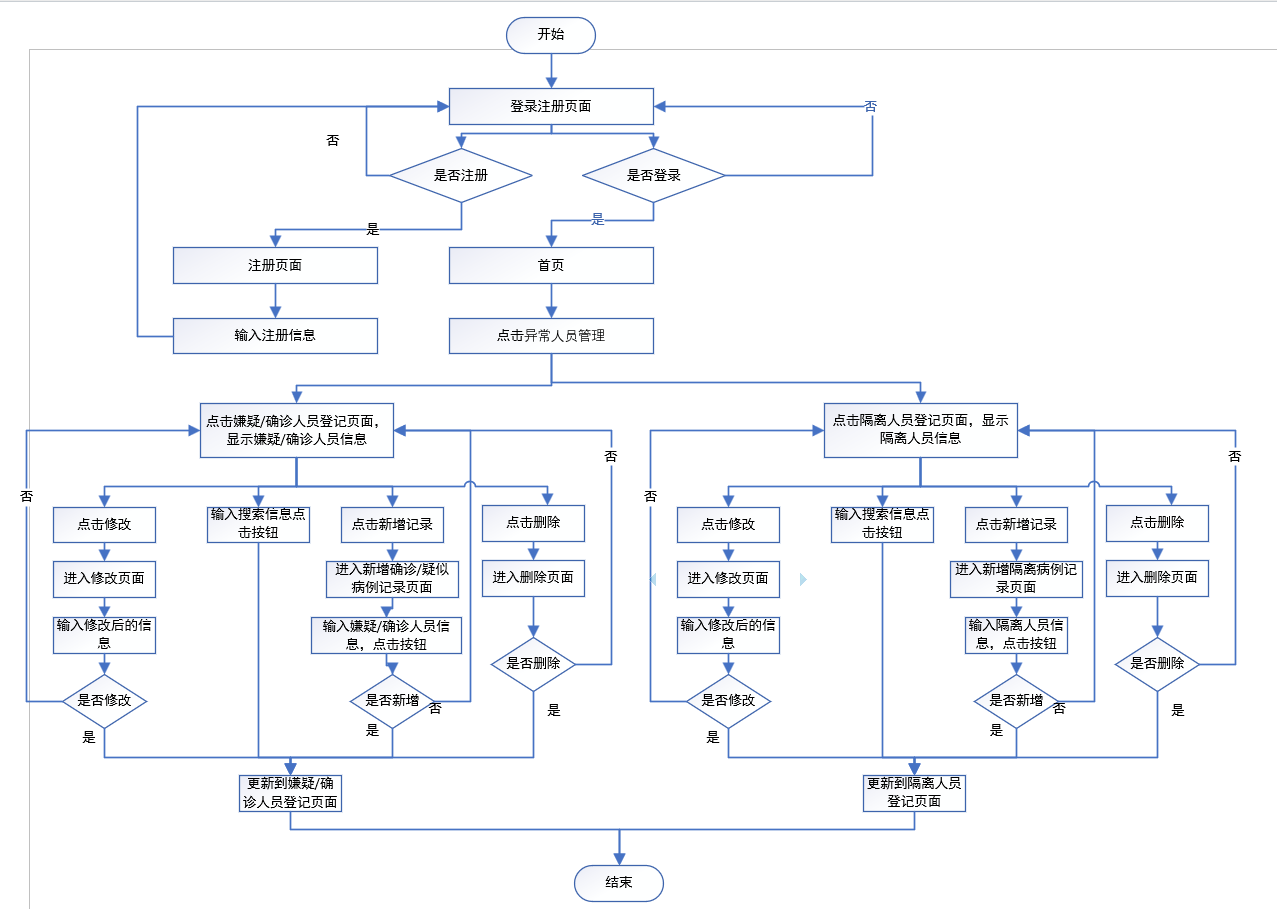


图5-7 异常人员管理模块流程图

嫌疑/确诊人员登记的核心代码如下：

@GetMapping("/findAll/{page}/{size}") //信息显示  
public Page<EmpIden> findAll(@PathVariable("page") Integer page, @PathVariable("size") Integer size){  
 Object cacheObject = redisCache.getCacheObject(USER\_LOGIN\_STATE);  
 if (cacheObject!=null){  
 Page<EmpIden> page1= new Page<>(page,size);  
 Page<EmpIden> result=mapper.selectPage(page1,null);  
 return result;  
 }  
 return null;  
}

@PostMapping("/save") //新增  
public String save(@RequestBody EmpIden empIden){  
 int result = mapper.insert(empIden);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@GetMapping("/findById/{id}") //id查询  
public EmpIden findById(@PathVariable("id") Integer id){  
 return mapper.selectById(id);  
}  
@PutMapping("/update") //修改  
public String update(@RequestBody EmpIden empIden){  
 int result=mapper.updateById(empIden);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@DeleteMapping("/deleteById/{id}") //删除  
public void deleteById(@PathVariable("id")Integer id){  
 mapper.deleteById(id);  
}  
@GetMapping("/search/{searchkey}/{stext}") //查询  
public List<EmpIden> search(@PathVariable("searchkey")String searchkey, @PathVariable("stext")String stext){  
 QueryWrapper<EmpIden> userQueryWrapper = Wrappers.query();  
 userQueryWrapper.like(searchkey,stext);  
 return mapper.selectList(userQueryWrapper);  
}

其主要界面如图5-8，5-9，5-10，5-11，5-12所示。



图5-8 嫌疑/确诊人员登记界面

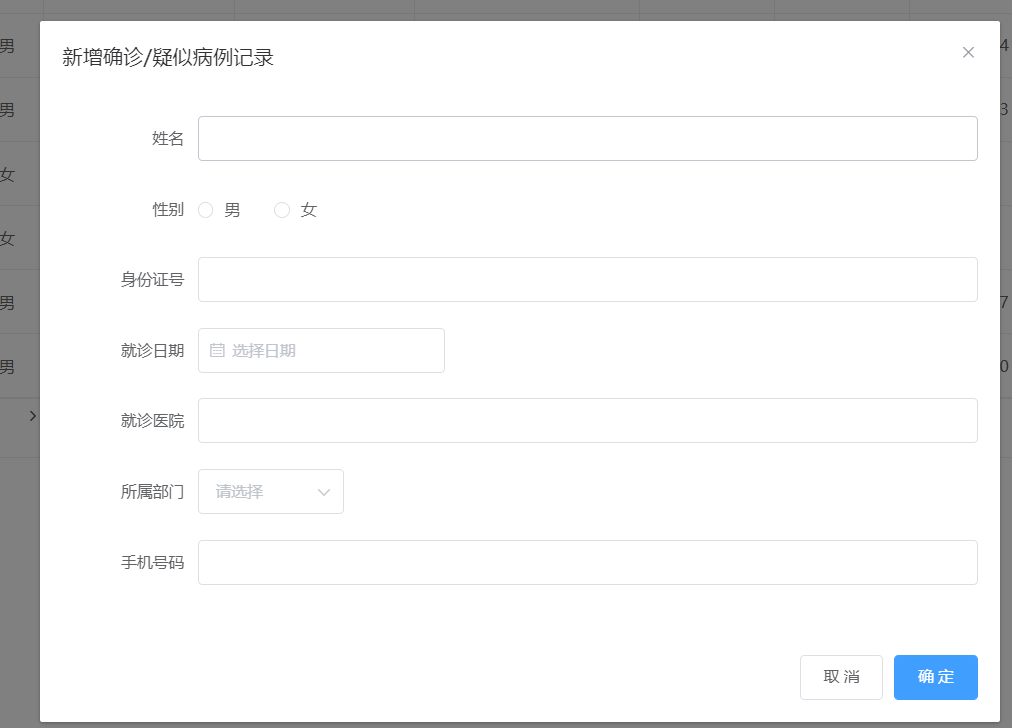


图5-9 嫌疑/确诊人员登记新增界面



图5-10 嫌疑/确诊人员登记修改界面

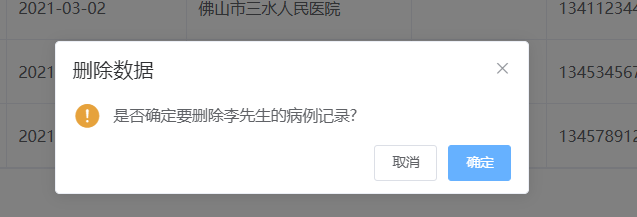


图5-11 嫌疑/确诊人员登记删除界面



图5-12 嫌疑/确诊人员登记查询界面

隔离人员登记的核心代码如下：

@GetMapping("/findAll/{page}/{size}") //信息显示  
public Page<EmpIs> findAll(@PathVariable("page") Integer page, @PathVariable("size") Integer size){  
 Object cacheObject = redisCache.getCacheObject(USER\_LOGIN\_STATE);  
 if (cacheObject!=null){  
 Page<EmpIs> page1= new Page<>(page,size);  
 Page<EmpIs> result=mapper.selectPage(page1,null);  
 return result;  
 }  
 return null;  
}  
@PostMapping("/save") //新增  
public String save(@RequestBody EmpIs empis){  
 Format f = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar c = Calendar.getInstance();  
 c.setTime(empis.getBegin());  
 c.add(Calendar.DAY\_OF\_MONTH, 14);  
 Date end = c.getTime();  
 empis.setEnd(end);  
 int result = mapper.insert(empis);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@GetMapping("/findById/{id}") //id查询  
public EmpIs findById(@PathVariable("id") Integer id){  
 return mapper.selectById(id);  
}  
@PutMapping("/update") //修改  
public String update(@RequestBody EmpIs empis){  
 Format f = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar c = Calendar.getInstance();  
 c.setTime(empis.getBegin());  
 c.add(Calendar.DAY\_OF\_MONTH, 14);  
 Date end = c.getTime();  
 empis.setEnd(end);  
 int result=mapper.updateById(empis);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@DeleteMapping("/deleteById/{id}") //删除  
public void deleteById(@PathVariable("id")Integer id){  
 mapper.deleteById(id);  
}  
@GetMapping("/search/{searchkey}/{stext}") //查询  
public List<EmpIs> search(@PathVariable("searchkey")String searchkey, @PathVariable("stext")String stext){  
 QueryWrapper<EmpIs> userQueryWrapper = Wrappers.query();  
 userQueryWrapper.like(searchkey,stext);  
 return mapper.selectList(userQueryWrapper);  
}

其主要界面如图5-13，5-14，5-15，5-16，5-17所示。



图5-13 隔离人员登记界面



图5-14 隔离人员登记新增界面

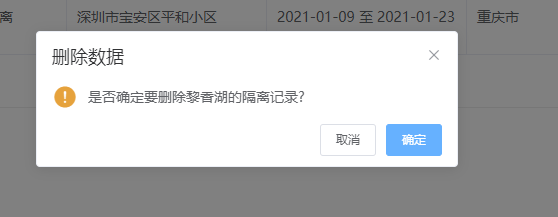


图5-15 隔离人员登记删除界面



图5-16 隔离人员登记删除界面



图5-17 隔离人员登记查询界面

## 5.3 防控物资管理模块

防控物资管理当管理员进入前端首页，点击防控物资管理，管理员就可以进行防疫物资的新增和防疫物资信息的查看，包括防疫物资查看的修改、搜索与删除，新增防疫物资的创建与重置从而进行信息更新和查看。

防控物资管理的流程图如图5-18所示。

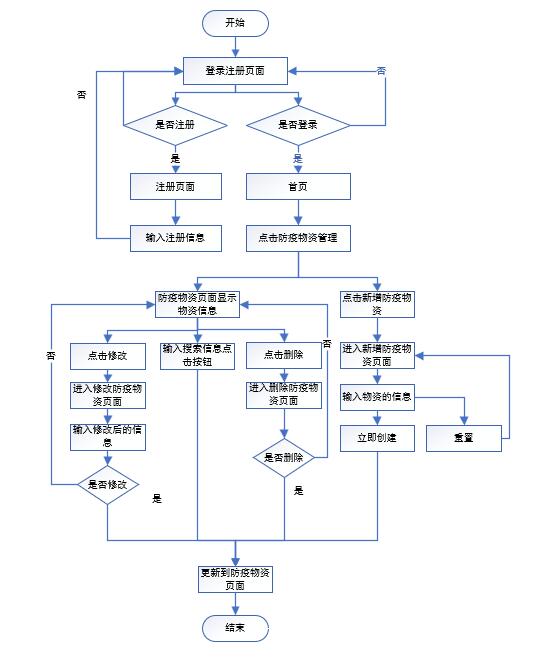


图5-18 防控物资管理模块流程图

防疫物资查看的核心代码如下：

@GetMapping("/findAll/{page}/{size}") //信息显示  
public Page<MaterialManage> findAll(@PathVariable("page") Integer page, @PathVariable("size") Integer size){  
 Object cacheObject = redisCache.getCacheObject(USER\_LOGIN\_STATE);  
 if (cacheObject!=null){  
 Page<MaterialManage> page1= new Page<>(page,size);  
 Page<MaterialManage> result=mapper.selectPage(page1,null);  
 return result;  
 }  
 return null;  
}  
@GetMapping("/findById/{id}") //id查询  
public MaterialManage findById(@PathVariable("id") Integer id){  
 return mapper.selectById(id);  
}  
@PutMapping("/update") //修改  
public String update(@RequestBody MaterialManage material){  
 int result=mapper.updateById(material);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}  
@DeleteMapping("/deleteById/{id}") //删除  
public void deleteById(@PathVariable("id")Integer id){  
 mapper.deleteById(id);  
}  
@GetMapping("/search/{searchkey}/{stext}") //查询  
public List<MaterialManage> search(@PathVariable("searchkey")String searchkey, @PathVariable("stext")String stext){  
 QueryWrapper<MaterialManage> userQueryWrapper = Wrappers.query();  
 userQueryWrapper.like(searchkey,stext);  
 return mapper.selectList(userQueryWrapper);  
}

其主要界面如图5-19，5-20，5-21，5-22所示。



图5-19 防疫物资查看界面

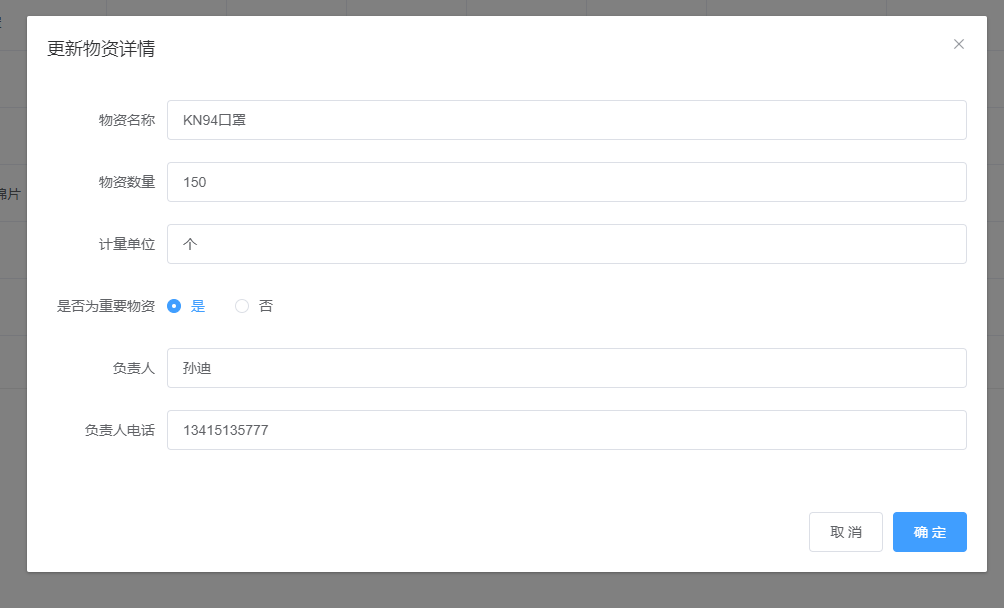


图5-20 防疫物资查看修改界面

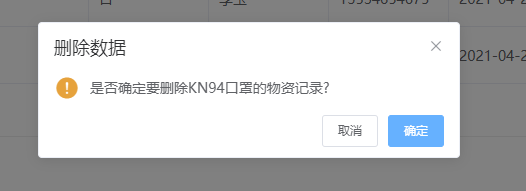


图5-21 防疫物资查看删除界面



图5-22 防疫物资查看查询界面

新增防疫物资的核心代码如下：

@PostMapping("/save") //新增  
public String save(@RequestBody MaterialManage material){  
 int result = mapper.insert(material);  
 if (result==1){  
 return "success";  
 }else {  
 return "error";  
 }  
}

其主要界面如图5-23所示。

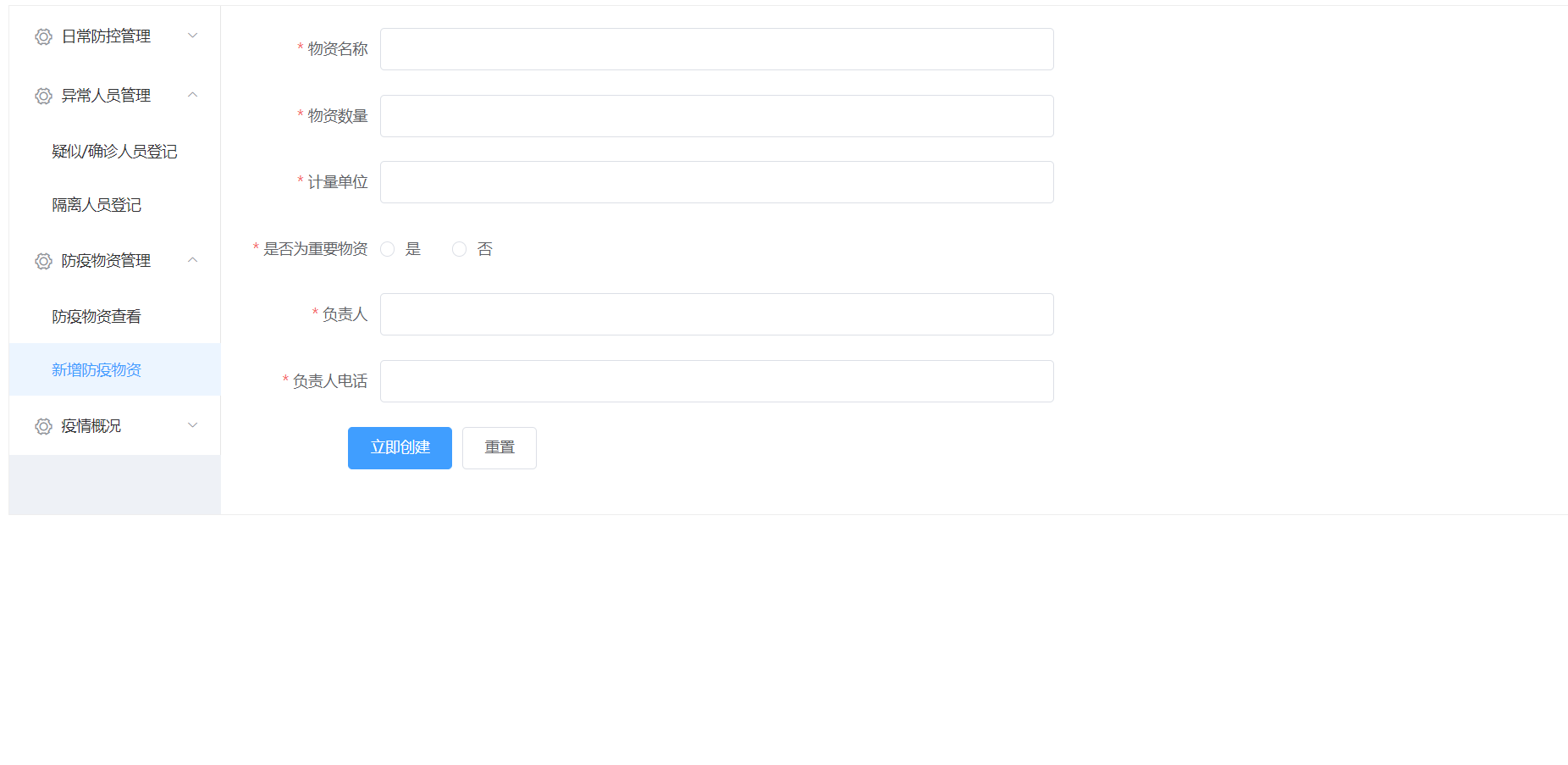


图5-23 防疫物资查看界面

## 5.4 疫情概况模块

疫情概况模块用来让管理者能够更直接地查看疫情的趋势和物资的数目，及可以显示数据，也可以下载数据。

疫情概况的流程图如图5-24所示。

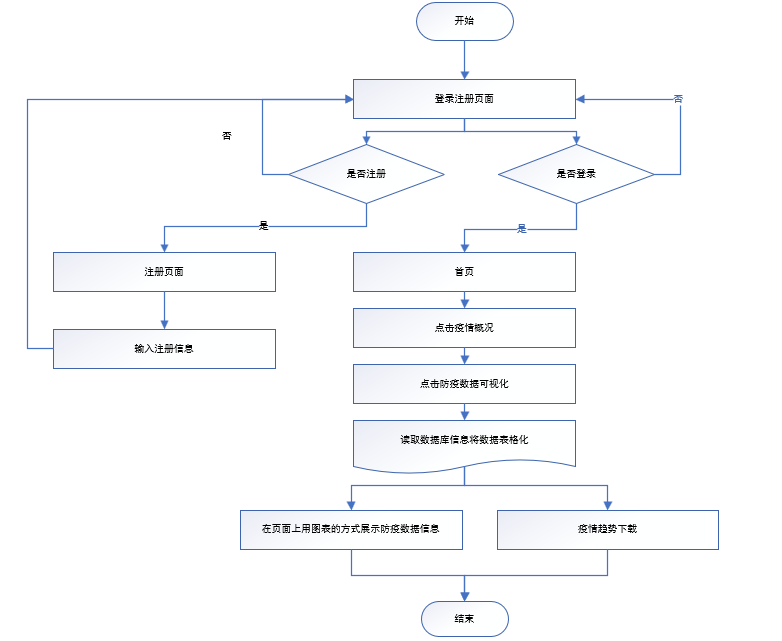


图5-24 疫情概况模块流程图

防疫数据可视化的核心代码如下：

@GetMapping("/LineVO") //疫情趋势   
public LineVO getLineVO(){  
 return this.chartService.lineVOList();  
}  
@GetMapping("/PieVO") //物资情况  
public List<PieVo> getPieVO(){  
 return this.chartService.pieVOMap();  
}

其主要界面如图5-25所示。

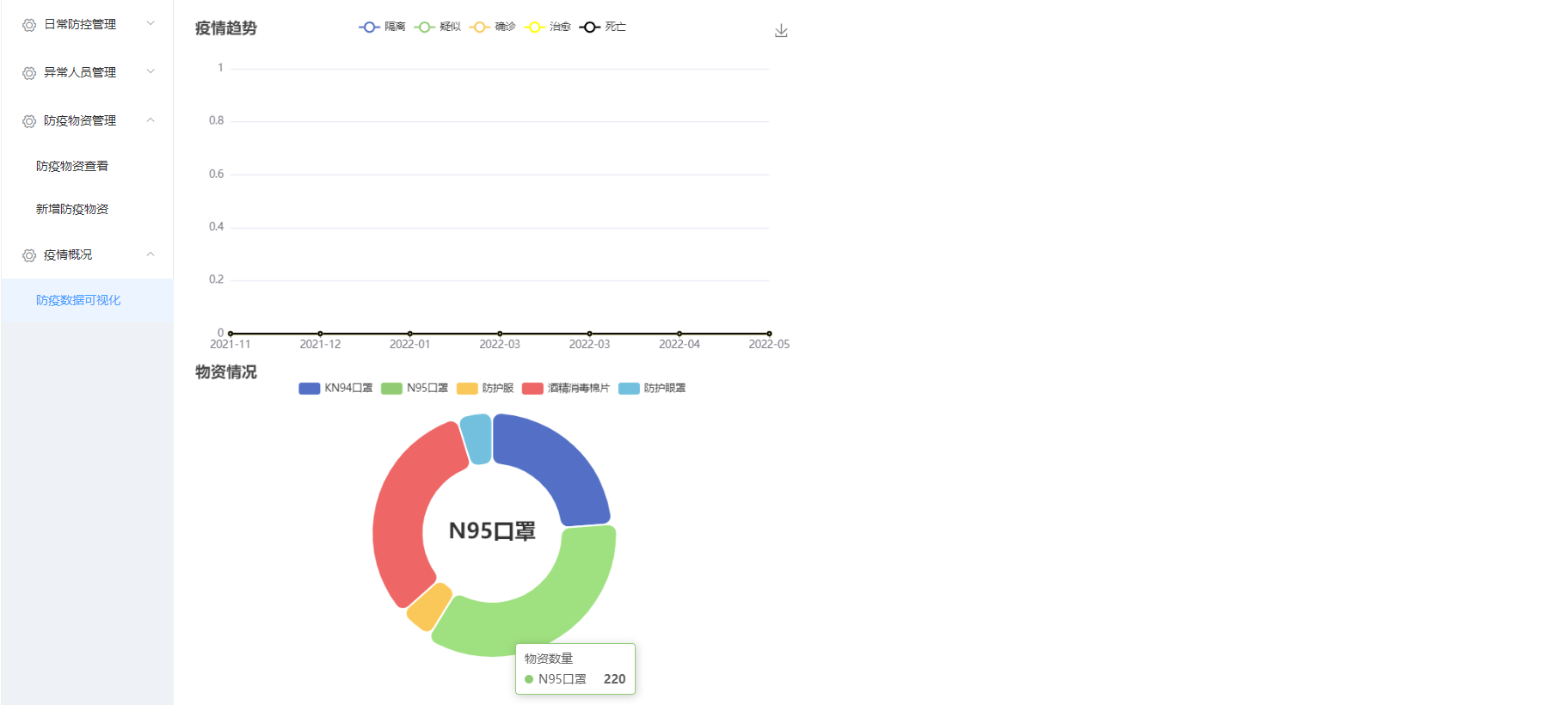


图5-25 疫情概况计界面

## 5.5 本章小结

本章重点介绍了疫情封控管理系统各功能模块的实现方法。以流程图及核心代码为基础，明确表达了该系统已完成的所有功能，并给出了最终效果图。

# 第6章 系统测试

本章主要对基于疫情的疫情封控管系统测试的内容和结果进行阐述。介绍了系统所要求的测试环境和方法，测试的重点是功能和系统性能。通过本章节，可以测试系统的功能是否达到使用要求。

## 6.1 测试原则

根据疫情封控管系统的开发需求，设计了一个操作简便，界面友好，可扩充性强的前端页面项目。本系统主要面向管理人员。在进行测试时，需要对各功能单独进行测试。对疫情封控管系统进行合理地测试，对于保证系统的实际使用至关重要。全面测试整个系统，发现关键问题，确保系统的整体运行正常。

## 6.2 功能测试

功能测试包含日常防控管理功能模块测试、异常人员管理功能模块测试、防控物资管理功能模块测试以及疫情概况功能模块测试，因而对系统各模块进行单独测试。

### 6.2.1 日常防控管理功能测试

管理者登录前端页面后，对健康打卡的申报和打卡记录信息的查看，并对各个模块下的各项功能进行测试，主要是测试打卡记录信息的查询、编辑和删除等功能，以及健康打卡申报的新增和重置。该模块各功能的测试用例表如表6-1所示。

表6-1 日常防控管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试子功能 | 测试详细 | 预期结果 | 实测结果 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日常防控管理 | 打卡记录信息显示 | 点击打卡记录，可以显示打卡记录信息 | 准确实现 | 准确实现 |
| 修改打卡记录信息 | 点击修改按钮，对此条打卡记录进行修改保存 | 准确实现 | 准确实现 |
| 搜索打卡记录信息 | 点击搜索按钮，寻找目标记录 | 准确实现 | 准确实现 |
| 重置健康打卡申报 | 点击健康打卡申报的重置按钮，可以将新输入的将  康打卡申报信息清零 | 准确实现 | 准确实现 |
| 删除打卡记录信息 | 点击删除按钮，此条打卡记录删除 | 准确实现 | 准确实现 |
| 创建健康打卡申报 | 点击健康打卡申报的立即创建按钮，可以添加新的  打卡记录 | 准确实现 | 准确实现 |

通过上述测试，打卡记录信息显示正确并且符合预期的结果，该健康打卡申报的添加可以顺利更新并在打卡记录页面显示，事实证明日常防控管理功能可以正常使用。

### 6.2.2 异常人员管理功能测试

管理者登录前端页面后，对嫌疑/确诊人员登记和隔离人员登记的信息查看，并对各个模块下的各项功能进行测试，主要是测试嫌疑/确诊人员登记和隔离人员登记信息的查询、新增、修改和删除等功能。该模块各功能的测试用例表如表6-2所示。

表6-2 异常人员管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试子功能 | 测试详细 | 预期结果 | 实测结果 |
| 异常人员管理 | 嫌疑/确诊人员登记信息显示 | 点击嫌疑/确诊人员登记，可以显示疑似/确诊人员登记的信息 | 准确实现 | 准确实现 |
| 嫌疑/确诊人员信息删除 | 点击删除按钮，删除此条疑似/确诊人员登记信息 | 准确实现 | 准确实现 |
| 嫌疑/确诊人员信息修改 | 点击修改按钮，对此条疑似/确诊人员登记信息修改 | 准确实现 | 准确实现 |
| 嫌疑/确诊人员信息搜索 | 点击搜索按钮，显示目标疑似/确诊人员登记信息 | 准确实现 | 准确实现 |
| 嫌疑/确诊人员信息新增记录 | 点击新增记录按钮，可以添加新的疑似/确诊人员登记信息 | 准确实现 | 准确实现 |
|  | 隔离人员登记信息显示 | 点击隔离人员登记，可以显示隔离人员的登记信息 | 准确实现 | 准确实现 |
|  | 隔离人员信息删除 | 点击删除按钮，删除此条隔离人员登记信息 | 准确实现 | 准确实现 |
|  | 隔离人员信息修改 | 点击修改按钮，对此条隔离人员登记信息修改保存 | 准确实现 | 准确实现 |
|  | 隔离人员信息搜索 | 点击搜索按钮，显示目标隔离人员登记信息 | 准确实现 | 准确实现 |
|  | 隔离人员信息新增记录 | 点击新增记录按钮，可以添加新的隔离人员登记信息 | 准确实现 | 准确实现 |

通过上述测试，嫌疑/确诊人员登记和隔离人员登记信息的显示是正确的，而且管理者可以对嫌疑/确诊人员登记和隔离人员登记信息进行查询、新增、修改和删除，达到了预期的效果。结果显示异常人员管理功能是有效的。

### 6.2.3 防控物资管理功能测试

管理者登录前端页面后，对防疫物资的新增和防疫物资信息的查看，并对各个模块下的各项功能进行测试，主要是测试防疫物资查看信息的查询、编辑和删除等功能，以及新增防疫物资的新增和重置。该模块各功能的测试用例表如表6-3所示。

表6-3 防控物资管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试子功能 | 测试详细 | 预期结果 | 实测结果 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 防控物资管理 | 防疫物资查看信息显示 | 点击防疫物资查看，查看防疫物资信息 | 准确实现 | 准确实现 |
| 防疫物资查看信息修改 | 点击修改按钮，对防疫物资信息进行修改保存 | 准确实现 | 准确实现 |
| 防疫物资查看信息搜索 | 点击搜索按钮，寻找目标防疫物资信息 | 准确实现 | 准确实现 |
| 新增防疫物资重置 | 点击新增防疫物资的重置按钮，可以将新输入的防疫物资信息清零 | 准确实现 | 准确实现 |
| 防疫物资查看信息删除 | 点击删除按钮，删除防疫物资信息 | 准确实现 | 准确实现 |
| 新增防疫物资创建 | 点击新增防疫物资的立即  创建按钮，可以添加新的防疫物资信息 | 准确实现 | 准确实现 |

通过上述测试，防疫物资查看信息显示正确并且符合预期的结果，该新增防疫物资的添加可以顺利更新并在打卡记录页面显示，事实证明防控物资管理功能可以正常使用。

### 6.2.4 疫情概况功能测试

使用管理员的账号登录后，选择疫情概况，并对各个模块下的各项功能进行测试，主要是测试疫情数据可视化的疫情趋势和物资情况的可视化图和疫情趋势的可视化图下载。该模块的功能测试用例表如表6-4所示。

表6-4 疫情概况测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试子功能 | 测试详细 | 预期结果 | 实测结果 |
|  | 疫情趋势可视化图的下载 | 点击防疫数据可视化的下载按钮，可将疫情趋势有关的信息下载下来 | 准确实现 | 准确实现 |
| 疫情概况 | 疫情趋势可视化图的显示 | 点击防疫数据可视化可以看见疫情趋势的可视化图 | 准确实现 | 准确实现 |
|  | 物资情况可视化图的显示 | 点击防疫数据可视化可以看见物资情况的可视化图 | 准确实现 | 准确实现 |

通过上述测试，物资情况页和疫情趋势页中所显示的相应的防疫物资查看和异常人员管理及日常防控管理所提供的图表信息都是正确的，与期望的结果一致，这表明了疫情概况功能是正确的。

## 6.3 性能测试

利用Apache JMeter性能测试工具，对本系统的性能进行测试，并将实测数据与性能指标进行比较，以确定该系统达到标准。

（1）性能指标预测是检测系统性能的对比参考材料，性能指标如表6-5所示。

表6-5 性能指标

|  |  |
| --- | --- |
| 考察项 | 目标值 |
| 响应时间 | <=3秒 |
| 业务成功率 | >98% |
| 并发数 | 20 |
| CPU使用率 | <75% |
| 内存使用率 | <75% |

（2）测试工具：Apache JMeter。

（3）测试环境：使用Windows 11操作系统，使用Google Chrome浏览器。

（4）测试结果：本次测试结果如表6-6所示。

表6-6 测试结果表

|  |  |  |  |
| --- | --- | --- | --- |
| 考察值 | 目标值 | 实际值 | 是否通过 |
| 响应时间 | <=3秒 | 1.15 | 是 |
| 业务成功率 | >98% | 100% | 是 |
| 并发数 | 24 | 24 | 是 |
| CPU使用率 | <75% | 37.25% | 是 |
| 内存使用率 | <75% | 46.71% | 是 |

测试结果表明，疫情封控管理系统的性能达到了预期的效果，尽管存在着一些潜在的问题，但测试数据良好。如果可以改进数据库，那么系统的性能可能得到改善。

## 6.4 本章小结

本章首先对疫情封控管理系统的各主要功能模块进行了具体的测试和分析，并采用相应的测试工具对其进行了性能测试。测试结果显示，系统各功能模块均符合设计要求。同时也表明，基于疫情的疫情封控管理系统能够很好的满足使用者的基本需要。

# 第7章 总结和展望

本章重点对该系统的设计成果进行了总结，阐述了在开发中碰到的问题和有待改进的地方，并对其今后的发展趋势作了简短的说明。

## 7.1 总结

本论文对基于疫情防控管理系统的设计与实现作了详尽的叙述与说明。在技术方面，对整个系统进行了整体的设计与实施，并就其测试原理和性能等问题进行了论述。本文的研究内容如下：

（1）该系统实现了日常防控管理、异常人员管理、防空物资管理等功能。创建者可以检查异常人员信息。

（2）该系统已在浏览器端完成了管理员登录、用户管理、角色管理、菜单管理和字典管理等系统的基本功能。

（3）根据日常防控管理里面的数据，管理员可以在数据统计界面中查看疫情状况和异常人员活动状态来参考分析，查看这段区间内的疫情概况。

（4）管理员可以在后台管理系统中查看所有检查的人员信息，而且可以对程序中的人员信息进行实时变更。

## 7.2 展望

本系统设计的功能虽然基本都实现了，但因为实际使用情况和本小组的技术水平的原因，系统中的一些细节还有待进一步的研究和改进。

（1）系统在日常防控管理的设计上，现有功能是只能通过管理员来添加人员信息和情况更新，造成管理者的工作量较大。后续希望可以使普通用户可以进行信息填写，由管理者和系统进行审核信息内容。

（2）在线疫情防控管理系统的各项功能均已完成，但在具体使用方面还存在一些需要改进的地方，不同的病原种类其疫情概况也不尽相同，所以面对不同的病原种类其防空管理还需要完善。

# 参考文献

|  |  |
| --- | --- |
| [1] | 熊柏祥.基于Springboot和Vue框架的考试资源服务平台的设计与实现[J].信息与电脑(理论版),2022,34(01):97-99+103. |
| [2] | 田海晴. 基于SpringBoot和Vue框架的共享运营管理平台的设计与实现[D].山东大学,2020.DOI:10.27272/d.cnki.gshdu.2020.004528. |
| [3] | 石科. Research and Development of Movie Recommendation System Based on Hybrid Recommendation Algorithm[D].首都经济贸易大学,2020.DOI:10.27338/d.cnki.gsjmu.2020.001161. |
| [4] | 魏聪媛. 基于Spring Boot的云南省森林火险预测系统的研建[D].北京林业大学,2020.DOI:10.26949/d.cnki.gblyu.2020.000129. |
| [5] | 陶文杰. 基于Mpvue和Spring Boot的线上选房平台的设计与实现[D].北京交通大学,2020.DOI:10.26944/d.cnki.gbfju.2020.003716. |
| [6] | Menezes Gabriel,Cafeo Bruno,Hora Andre. How are framework code samples maintained and used by developers? The case of Android and Spring Boot[J]. The Journal of Systems & Software,2022,185. |
| [7] | 常凯琦. 基于Android的会计学习平台的设计与实现[D].北京交通大学,2020.DOI:10.26944/d.cnki.gbfju.2020.002776. |
| [8] | 李军,朱雪琴, 抗击新冠肺炎疫情COVID-19　南京首例新冠肺炎康复者成功捐献血浆. 丁森,李胜 主编,南京卫生健康年鉴,广陵书社,2021,18,年鉴.DOI:10.40959/y.cnki.ynjws.2022.000151. |
| [9] | 金孟勇. 基于Spring-Boot技术的VR流媒体后台系统的研究与实现[D].北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.002007. |
| [10] | 张杰. 基于React+Spring的教学系统设计与实现[D].山东师范大学,2019. |
| [11] | 魏博. 基于Spring Cloud的云平台管理系统的研究及实现[D].华北电力大学(北京),2019.DOI:10.27140/d.cnki.ghbbu.2019.000295. |
| [12] | Huang Liming,Lee Man Ying,Chen Xiaojie,Tseng Hsien Wei,Yang Cheng Fu,Lee Shun Fa. Using Microservice Architecture as a Load Prediction Strategy for Management System of University Public Service[J]. SENSORS AND MATERIALS,2021,33(2). |
| [13] | 新冠肺炎疫情防控　对新冠肺炎密切接触者集中医学观察点进行终末消毒消杀. 赵丹 主编,安阳县年鉴,中原出版传媒集团中原传媒股份公司中州古籍出版社,2021,8,年鉴.DOI:10.38420/y.cnki.yanxn.2022.000013. |
| [14] | 金明俐. 基于Spring Boot与Vue框架的叫号系统的设计与实现[D].中国地质大学(北京),2021.DOI:10.27493/d.cnki.gzdzy.2021.001349. |
| [15] | 疫情防控　“非典”疫情监测报告. 姚志彬 主编,广东卫生年鉴,广东省出版集团广东人民出版社,2006,10,年鉴. |