

课程设计报告

课题名称：疫情数据可视化

课程名称：软件工程

学院：计算机与信息安全学院

专业：计算机科学与技术

成员：1800301037禤成伟

1800301038闫书明

1800301039袁嘉鸿

1800301040周志兆

指导老师：李优

报告日期：2021年4月

表一 团队任务分工表

|  |  |  |  |
| --- | --- | --- | --- |
| 题目 | 疫情数据可视化 | | |
| 负责人 | 禤成伟 | 指导老师 | 李优 |
| 序号 | 学号 | 姓名 | 个人任务描述 |
| 1 | 1800301037 | 禤成伟 | 负责项目主题部分的代码编写，指导完成项目任务，PPT制作 |
| 2 | 1800301038 | 闫书明 | 负责需求分析，总体设计，商讨软件架构，查找独立模块代码实现，PPT制作 |
| 3 | 1800301039 | 袁嘉鸿 | 负责详细设计，查找资料，查询实现独立功能模块 |
| 4 | 1800301040 | 周志兆 | 讨论实现可行性，进行软件测试，负责软件bug初步解决方案；进行项目总结，报告格式规整 |

关键词：数据、可视化

# 引言

2019年12月末，中国武汉发生新型冠状病毒（2019-nCoV) 感染的肺炎疫情，为遏制疫情蔓延，有效切断病毒传播途径，在中央政府指导下，各级政府部分采取了一系列防控措施： 2020年1 月23 日10时起对武汉“封城”，全国 31个省市也相继实施了严格的防控措施；全国各省市向武汉和湖北派遣医疗队参与救治工作；在全国范围内调配口罩、防护服、药品等急需的医疗资源支援武汉；指导和督促全国范围内拥有医疗物资生产资质的企业尽快恢复生产能力；定向拨付专项财政资金用于疾病防控；从其他省份调集物资保障武汉市民日常生活。

  值得一提的是，中国互联网企业在此次疫情防控中发挥了社会治理方面的重要作用。以腾讯为例，围绕应对疫情管控需求开发了十一款产品。其中疫情在线问诊功能，对于减少发热病人之间的相互交叉感染具有重要的作用，患者在家通过互联网向在线医生问诊，减少了病毒传播或感染的风险；谣言粉碎对于公众采取理性态度看待疫情的发展具有重要意义。

基于此背景，我们小组决定开发一个疫情可视化系统，以期更好的服务大众，锻炼提高。

# 1 可行性分析

如今是互联网的时代，这几年随着互联网技术的高速发展，涌现了许多成熟、 高效、稳定的互联网开发语言及工具，不仅如此，更是出现了能提高开发效率的 各种技术的框架，对于基本功能需求的管理系统的开发，技术已经十分成熟。使 用 Java-Web 作为后端主要技术和 Vue 作为主要前端技术，采用前后端分离开发8 的模式是当下主流，大大减少了开发过程中的冗余，代码耦合大大减少，且能让 开发人员分工明确，各司其职。不仅能让开发效率大大提高，而且更能让系统趋 于完善、易于维护。因此课程设计过程管理系统基于 Java 和 Vue 的开发，在技 术上完全可行。

# 2 需求分析

## 2.1系统目标及任务

为了完成本校学生的每日疫情基本情况的汇总以及查看，方便校方统计相关数据，并做出进一步的疫情应对教学计划，我们小组准备开发一个简单的疫情数据可视化系统。

## 2.2系统功能需求

疫情数据可视化系统主要包括四个模块：实时数据采集、数据预处理、数据可视化以及用户管理。系统各个模块的子功能如图2.1。具体如下：

1. 实时数据获取：爬取json数据，解析数据存入本地数据库
2. 数据预处理：统计全国整体疫情数据、统计各地区疫情数据
3. 数据可视化：实时动态地展示疫情地图、折线图、南丁格尔玫瑰图
4. 用户模块：注册与登录

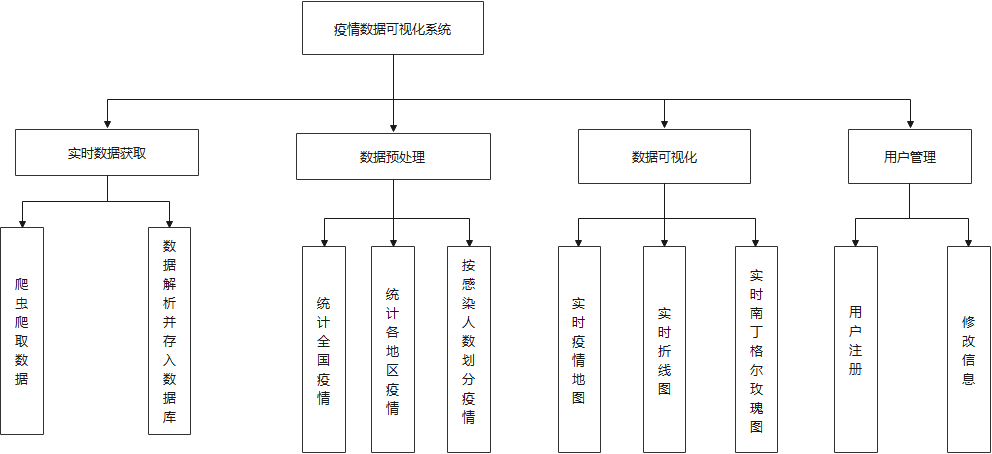
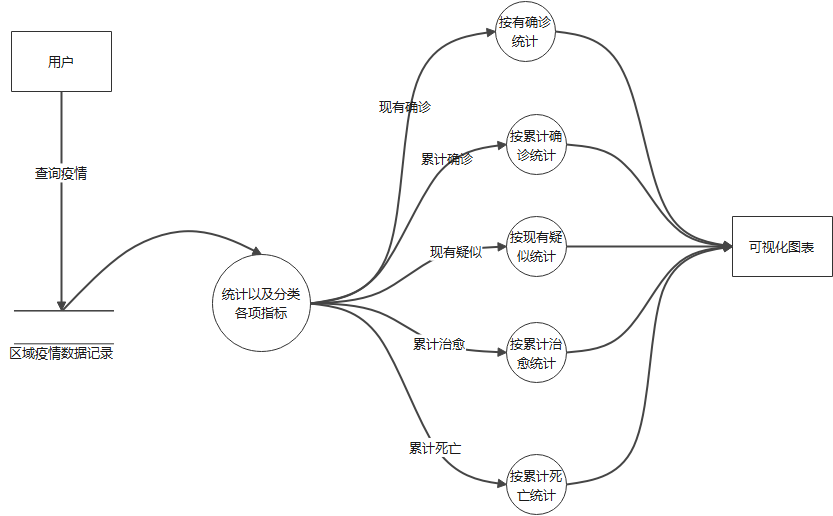


图2.1 系统功能图

按照上述系统所示功能设计，本系统的数据流图如图2.2所示：



2.2 系统数据流图

## 2.3性能需求

为了能够发挥本系统的实用效果，让用户切实体会到系统带来的效益，本系统需具备以下基本性能：

1. 系统必须准确显示实时疫情状况以便用户能够真实地了解疫情。
2. 本系统网页响应时间应在1s左右，从而确保用户体验。
3. 数据可视化的表现形式应该清晰明了并且具备较强的可读性。

## 2.4可靠性和可用性需求

各模块之间相互独立，一个模块出问题不影响其他模块工作。比如爬虫模块与可视化数据模块，这二者就是相互独立的：爬虫模块爬取数据到数据库，数据可视化模块从数据库中读取数据，二者独立工作，互不影响。爬虫模块失效后，数据可视化模块依然可以正常工作，读取数据库中的数据，只是读取出来的数据不保证是最新的。同样，数据可视化模块失效，爬虫模块依然可以正常工作，照常爬取数据到数据库中。

## 2.5出错处理需求

当用户请求异常时，我们会及时弹窗提示用户。

## 2.6接口需求

#### user-controller

##### 1）getUserInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **getUserInfo** | | | | |
| 接口描述 | getUserInfo | | | |
| URL | /user\_center/user/getUserInfo | | | |
| 请求方式 | get | | | |
| 请求类型 |  | | | |
| 返回类型 | \*/\* | | | |
| **参数名** | **数据类型** | **参数类型** | **是否必填** | **说明** |
| 1.token | string | header | N | 用户token |
| **状态码** | **描述** | | **说明** | |
| 200 | OK | |  | |
| 401 | Unauthorized | |  | |
| 403 | Forbidden | |  | |
| 404 | Not Found | |  | |
| **返回属性名** | **类型** | | **说明** | |
| 1.code | integer(int32) | |  | |
| 2.data | object | |  | |
| 3.message | string | |  | |
| 4.success | boolean | |  | |
| **示例** | | | | |
| **请求参数** | --header 'token:string' | | | |
| **返回值** | {"code":0,"data":{},"message":"string","success":true} | | | |

##### 2）login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **login** | | | | |
| 接口描述 | login | | | |
| URL | /user\_center/user/login | | | |
| 请求方式 | post | | | |
| 请求类型 | application/json | | | |
| 返回类型 | \*/\* | | | |
| **参数名** | **数据类型** | **参数类型** | **是否必填** | **说明** |
| 1.token | string | header | N | 用户token |
| 2.user | object:User | body | Y | user |
| 2.1.age | integer(int32) |  | N |  |
| 2.2.avatar | string |  | N |  |
| 2.3.gmtCreate | string(date-time) |  | N |  |
| 2.4.gmtModified | string(date-time) |  | N |  |
| 2.5.id | string |  | N |  |
| 2.6.isDeleted | boolean |  | N |  |
| 2.7.isDisabled | boolean |  | N |  |
| 2.8.mail | string |  | N |  |
| 2.9.mobile | string |  | N |  |
| 2.10.nickname | string |  | N |  |
| 2.11.password | string |  | N |  |
| 2.12.sex | integer(int32) |  | N |  |
| **状态码** | **描述** | | **说明** | |
| 200 | OK | |  | |
| 201 | Created | |  | |
| 401 | Unauthorized | |  | |
| 403 | Forbidden | |  | |
| 404 | Not Found | |  | |
| **返回属性名** | **类型** | | **说明** | |
| 1.code | integer(int32) | |  | |
| 2.data | object | |  | |
| 3.message | string | |  | |
| 4.success | boolean | |  | |
| **示例** | | | | |
| **请求参数** | --header 'token:string' -d '{"age":0,"avatar":"string","gmtCreate":"2020/01/01 00:00:00","gmtModified":"2020/01/01 00:00:00","id":"string","isDeleted":true,"isDisabled":true,"mail":"string","mobile":"string","nickname":"string","password":"string","sex":0}' | | | |
| **返回值** | {"code":0,"data":{},"message":"string","success":true} | | | |

##### 3）registerUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **registerUser** | | | | |
| 接口描述 | registerUser | | | |
| URL | /user\_center/user/register | | | |
| 请求方式 | post | | | |
| 请求类型 | application/json | | | |
| 返回类型 | \*/\* | | | |
| **参数名** | **数据类型** | **参数类型** | **是否必填** | **说明** |
| 1.registerVo | object:RegisterVo | body | Y | registerVo |
| 1.1.code | string |  | N |  |
| 1.2.mail | string |  | N |  |
| 1.3.mobile | string |  | N |  |
| 1.4.nickname | string |  | N |  |
| 1.5.password | string |  | N |  |
| 2.token | string | header | N | 用户token |
| **状态码** | **描述** | | **说明** | |
| 200 | OK | |  | |
| 201 | Created | |  | |
| 401 | Unauthorized | |  | |
| 403 | Forbidden | |  | |
| 404 | Not Found | |  | |
| **返回属性名** | **类型** | | **说明** | |
| 1.code | integer(int32) | |  | |
| 2.data | object | |  | |
| 3.message | string | |  | |
| 4.success | boolean | |  | |
| **示例** | | | | |
| **请求参数** | --header 'token:string' -d '{"code":"string","mail":"string","mobile":"string","nickname":"string","password":"string"}' | | | |
| **返回值** | {"code":0,"data":{},"message":"string","success":true} | | | |

#### area-controller

##### 1）queryAWeek

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **queryAWeek** | | | | |
| 接口描述 | queryAWeek | | | |
| URL | /visualization\_data/area/queryAWeek | | | |
| 请求方式 | get | | | |
| 请求类型 |  | | | |
| 返回类型 | \*/\* | | | |
| **参数名** | **数据类型** | **参数类型** | **是否必填** | **说明** |
| 1.token | string | header | N | 用户token |
| **状态码** | **描述** | | **说明** | |
| 200 | OK | |  | |
| 401 | Unauthorized | |  | |
| 403 | Forbidden | |  | |
| 404 | Not Found | |  | |
| **返回属性名** | **类型** | | **说明** | |
| 1.code | integer(int32) | |  | |
| 2.data | object | |  | |
| 3.message | string | |  | |
| 4.success | boolean | |  | |
| **示例** | | | | |
| **请求参数** | --header 'token:string' | | | |
| **返回值** | {"code":0,"data":{},"message":"string","success":true} | | | |

#### code-controller

##### 1）sendMsm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sendMsm** | | | | |
| 接口描述 | sendMsm | | | |
| URL | /msm/code/send/{phone} | | | |
| 请求方式 | get | | | |
| 请求类型 |  | | | |
| 返回类型 | \*/\* | | | |
| **参数名** | **数据类型** | **参数类型** | **是否必填** | **说明** |
| 1.phone | string | path | Y | phone |
| 2.token | string | header | N | 用户token |
| **状态码** | **描述** | | **说明** | |
| 200 | OK | |  | |
| 401 | Unauthorized | |  | |
| 403 | Forbidden | |  | |
| 404 | Not Found | |  | |
| **返回属性名** | **类型** | | **说明** | |
| 1.code | integer(int32) | |  | |
| 2.data | object | |  | |
| 3.message | string | |  | |
| 4.success | boolean | |  | |
| **示例** | | | | |
| **请求参数** | --header 'token:string' | | | |
| **返回值** | {"code":0,"data":{},"message":"string","success":true} | | | |

## 2.7 约束

本系统需在细节上进行打磨，因此本系统的功能约束有以下几个方面：

1. 疫情地图能随着鼠标落点在不同区域间的变化而显示不同的高亮色块。
2. 可视化的图表动态显示，从而带来不一样的视觉效果。
3. 不同程度的疫情区域显示不同程度的渐变色。

# 3 总体设计

## 3.1 软件架构设计

疫情可视化系统的软件架构大致可以包含以下七个层次，展示层、网关、控制层、业务逻辑层、数据访问层、数据库和运行环境层。软件架构设计如图 3.1 所示：

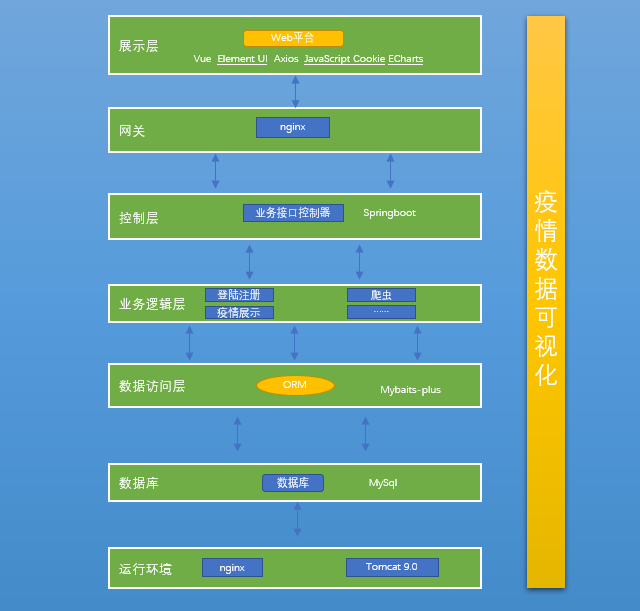


图3.1 软件架构设计图

## 3.2 数据库设计

### 3.2.1数据库关系分析

根据需求调研，基本可以对数据库的关系大致抽象如下描述:

用户信息独立存在，暂时只作为登陆网站用户的信息收集和网站访问限制保护。一个国家可以有多个省份，每个地区都有自己的上一级地域，国家作为最大的地区其上一级地域设置为国家自身。

权限方面，各个普通账户对数据只有查看权限，没有编辑权限。

### 3.2.2概念结构与逻辑结构的设计

根据上述抽象的关系，可设计出 E-R 图，如图 3.2 所示:

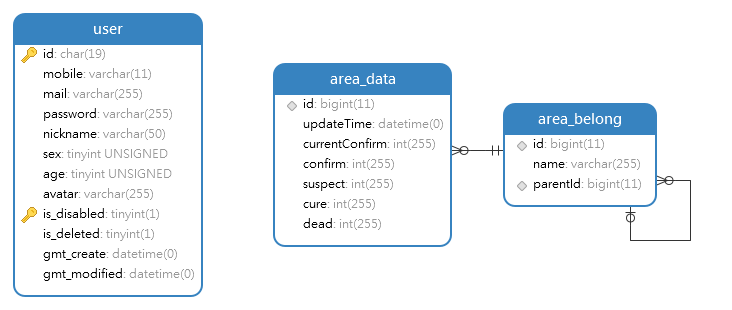
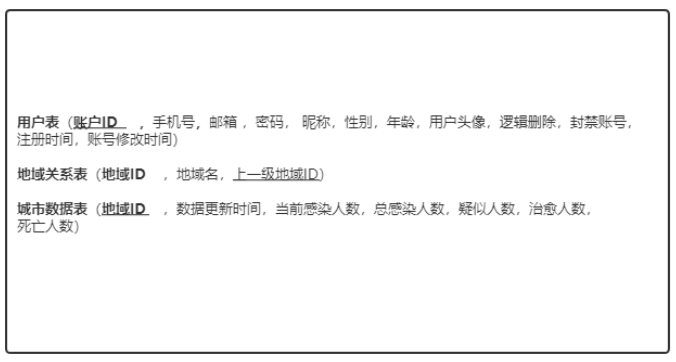


图 3.2  E-R

根据上述 E-R 图的设计以及 E-R 图转化为关系模型的规范, 作关系模型的转化，关系的码将用下划线表示，具体如表 3.1 所示:

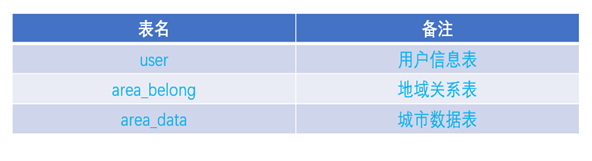
表 3.1 关系模型的转化



### 3.2.3物理结构的设计与实现

依据上述设计的关系模型，使用 MySQL 数据库建立了 3 张关系表，它们的表名以及对应的名称如表 3.2所示:

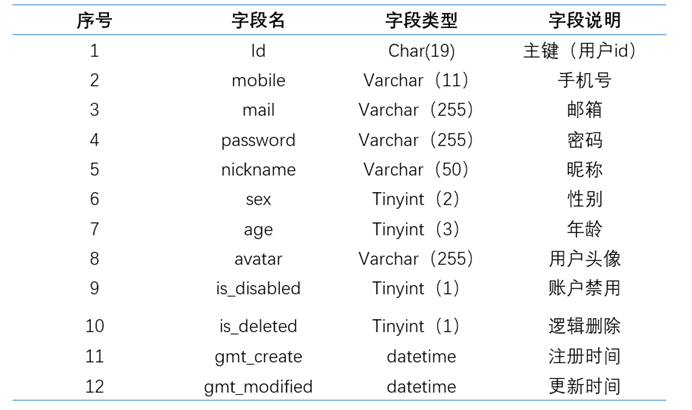
表 3.2



#### 3.2.3.1用户信息表

用户信息表主要用于存储用户的各种信息，主要包括账号，密码，手机号，邮箱，昵称，性别，年龄，头像，账号注册时间，账号信息修改时间 以及对账户标记是否进行逻辑删除，标记账户是否被禁用。使用账号 ID 作为该表的唯一主键，能够有效的保证账号的唯一性。该表如表 3.3 所示:

表 3.3 用户信息表



#### 3.2.3.2 地区关系表

地区关系表主要用于存储地区间的关系信息，记录城市和上一级地区省区的关系，可以由该明确其从属关系。主要信息字段为，地域id，地名，上一级地域id。表中的上一级地域id作为外键关联本表中的地域id，上一级的id必须在id中存在。该表如表3.4 所示:

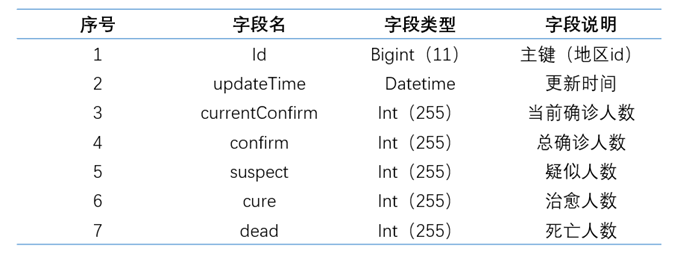
表 3.4 地区关系表



#### 3.2.3.3 城市数据表

城市数据表主要展示当前获取的城市疫情数据，包括城市id，数据更新时间，当前感染人数，地区感染过的总人数，当前疑似的，治愈的人数，死亡人数。城市的id作为外键关联地区关系表中的地域id。该表如表3.5 所示:

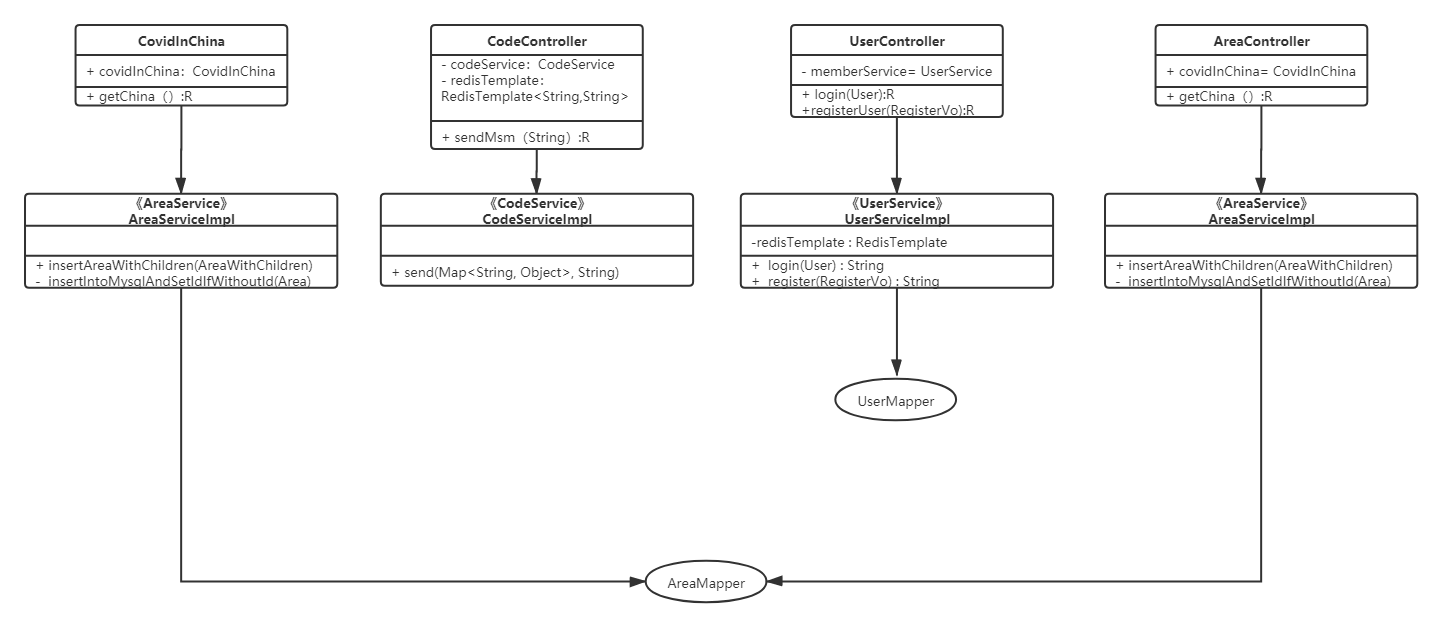
表 3.5 城市数据表



## 3.3类图设计

对于整个系统来说，为保证整个项目的运行需要的类较多，在这里只将系统中业务处理 中的类进行分析设计。系统的业务类主要由爬虫相关的CovidInChina，短信服务相关的CodeController系列类，账户注册登录UserController系列类，前端页面展示AreaController系列类。每一个功能的业务类有对应的负责分发请求的控制器，具体实现类，及对应的操作数据的 Mapper 类。该图如图 3.3 所示:

图 3.3 系统类图



# 4 详细设计

根据需求分析以及总体设计，疫情数据可视化主要有三个模块：注册模块、登录模块、数据可视化模块。

## 4.1注册模块的设计与实现

用户使用疫情数据可视化系统前需先注册一个账号，注册成功后才可登录进入主界面，用户的注册过程为：

（1）用户请求注册。

（2）系统返回注册界面。

（3）用户输入手机号码、邮箱以及密码，并获取验证码。

（4）系统生成验证码并根据用户手机号码发送验证码。

（5）用户输入收到的短信验证码并点击注册。

（6）系统根据用户输入的验证码进行校验。

（7）如果系统对验证码校验成功，则将用户信息写入数据库，并跳转到登录界面。

（8）如果验证码校验失败，则返回登录失败的原因及相关提示。

该过程的活动图如图4.1所示。

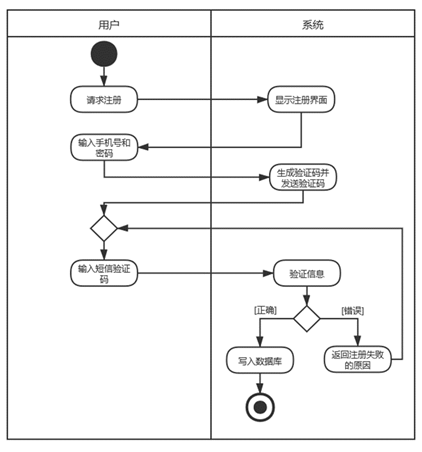


图4.1 用户注册活动图

实现效果如图4.2所示。

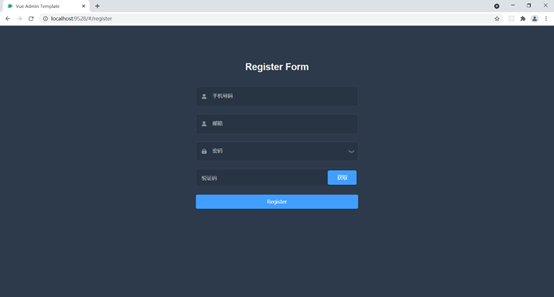


图4.2用户注册界面

## 4.2登录模块的设计与实现

用户完成注册后，即可使用手机号码和密码进行登录，登录的过程如下：

（1）系统先判断本地是否存在cookie，若存在进入用户界面，否则进入登录页面。

（2）若进入登录页面，用户输入手机号码以及密码，系统判断账户是否存在，若已存在则进行数据验证，若数据验证成功可以进入用户界面，若验证失败则退回登录页面。

（3）如果系统判定账户不存在则进入到注册页面进行注册。

该过程的流程图如图4.3所示。

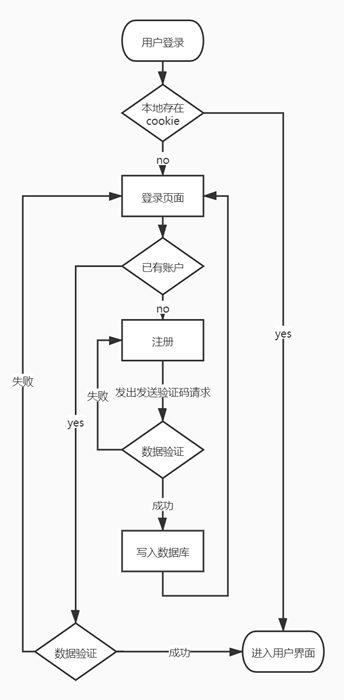


图4.3 流程图

实现效果如图4.4所示。

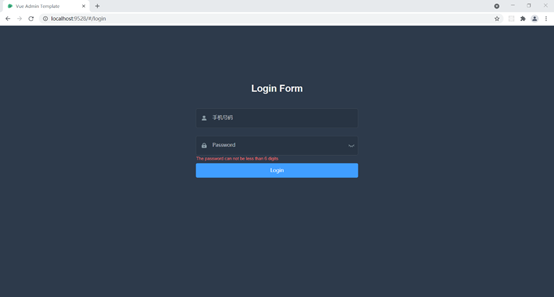


图4.4 用户登录界面

## 4.3数据可视化模块的设计与实现

系统爬取每日疫情数据，包括各省、市的现有确诊、累计确诊、累计治愈和累计死亡等信息，并将这些收集到的数据解析然后写入数据库，当用户成功登录后，系统将查询数据库并将查询到的结果以可视化的方式显示给用户。

该过程的活动图如图4.5所示。

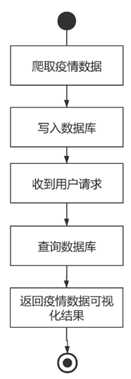


图4.5 活动图

实现效果如图4.6所示。

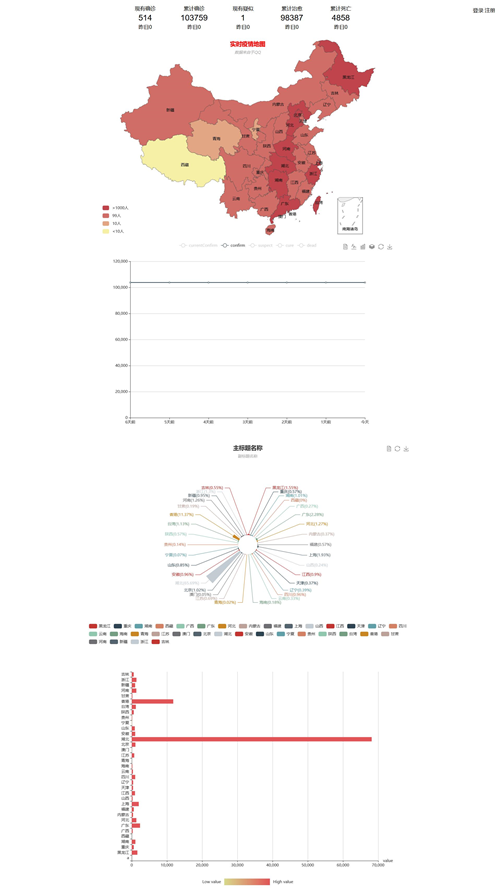


  图4.6

# 5 软件测试

本章节为疫情数据可视化系统功能的测试,编写的目的在于总结测试阶段的测试和分析测试结果，得到对软件质量的评价，描述软件是否符合预期需求。同时分析软件存在的缺陷，为修复Bug和预防后续开发可能会引发更多缺陷提供建议。

## 5.1功能测试

### 5.1.1注册功能测试

注册界面如图 5.1 所示:

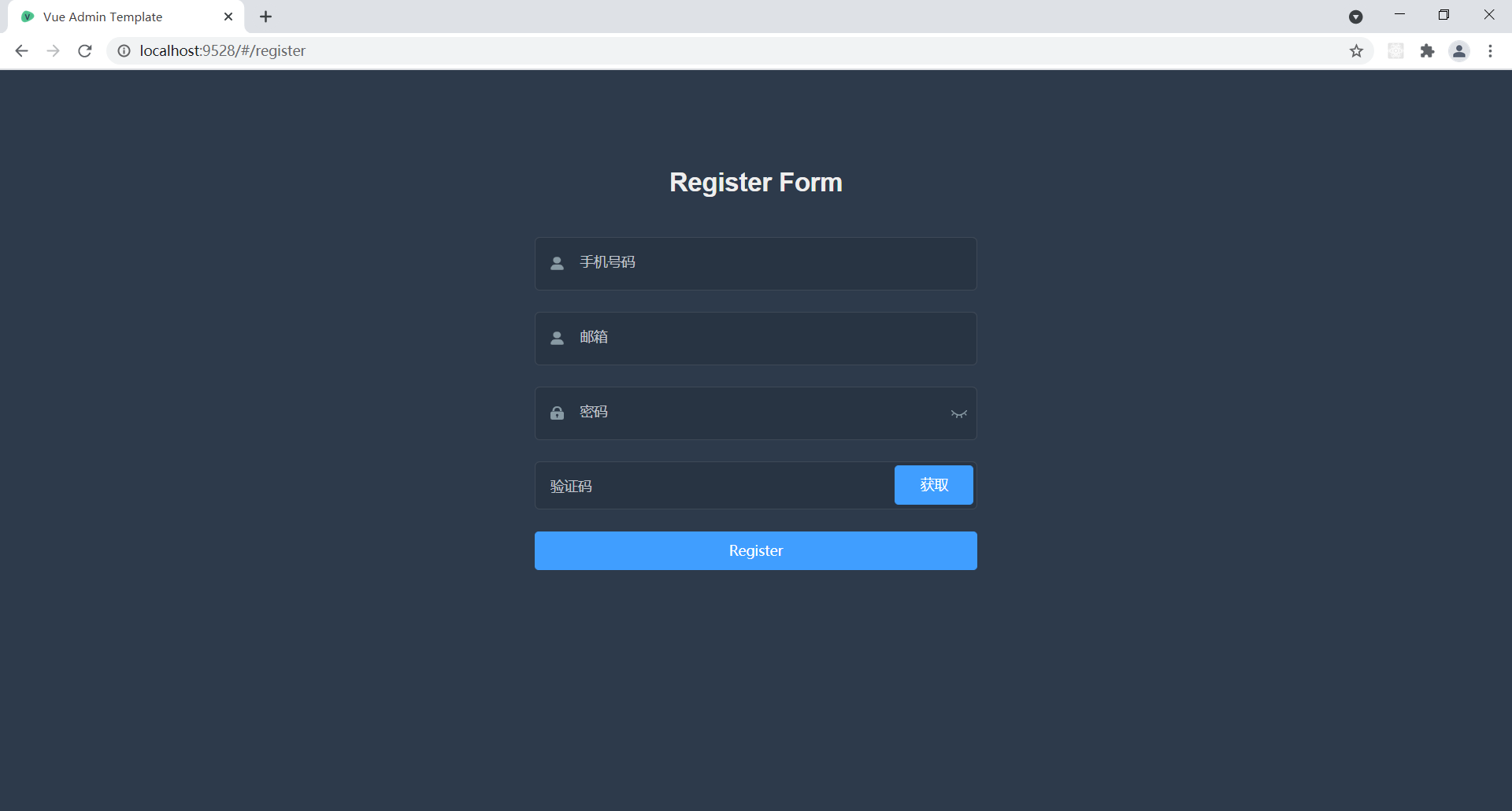


图5.1 用户注册界面

注册要求输入手机号码、邮箱、密码以及验证码等信息。手机号码要求输入11位纯数字，若输入位数不够11位或者输入非数字，则会提示输入有误，要求重新输入。输入的邮箱必须是正确的邮箱格式，比如xxx@qq.com、xxx@163.com等格式，否则会有错误提示，要求重新输入。密码的格式要求是大于6位的数字、字母或者字符的组合。输入完以上三项，点击获取，稍后将会收到验证码，输入正确的验证码，点击Register按钮注册，即可成功注册。

### 5.1.2登录功能测试

登录界面如图 5.2 所示:

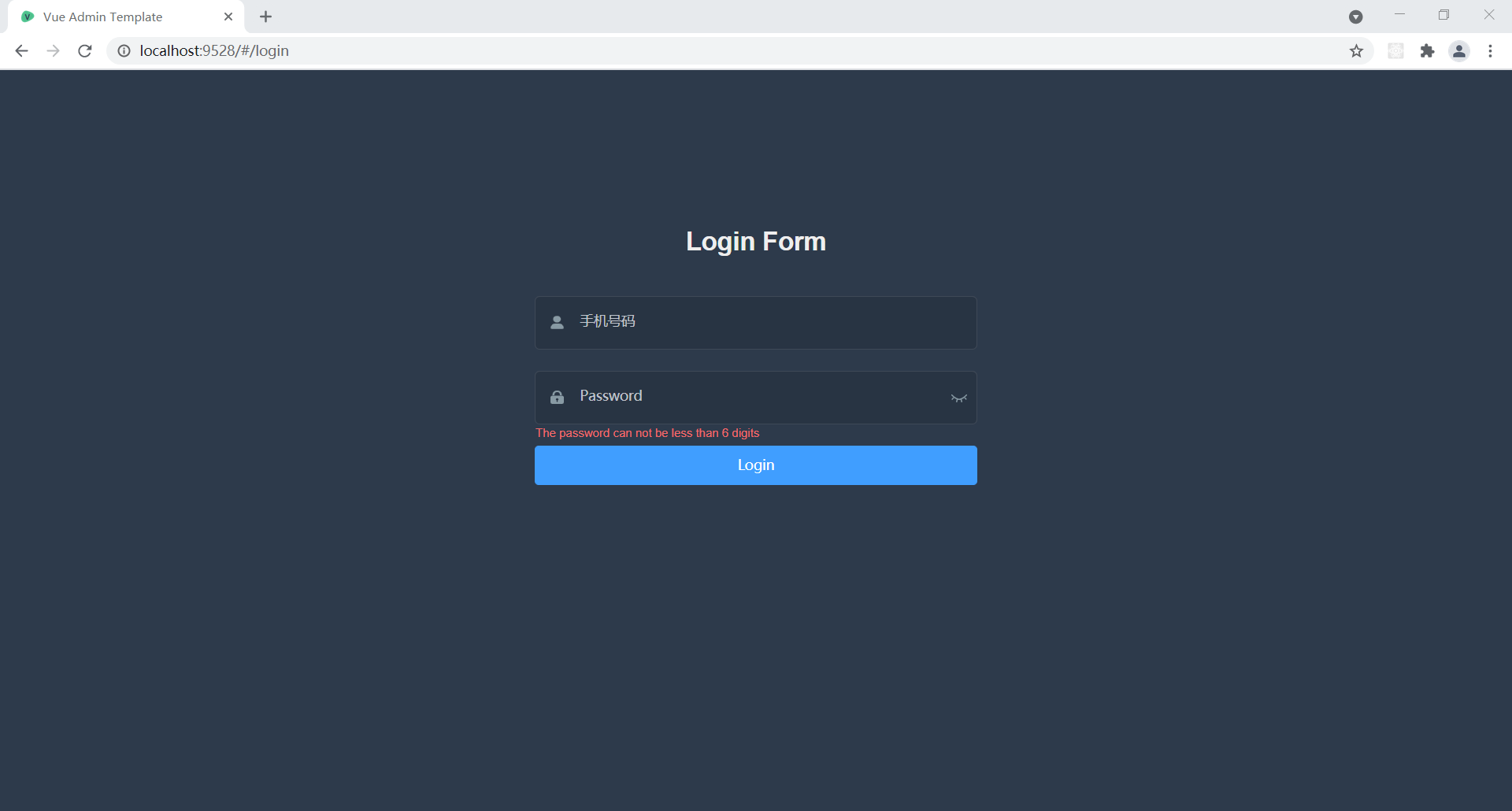


图5.2 用户登录界面

用户输入注册时的手机号码以及设置的密码，然后点击Login按钮，数据将进行验证。若验证成功，可进入到用户界面。否则提示输入有误，要求重新输入。

## 5.2用户界面测试

用户登录成功后，跳转至用户界面。如图5.3所示。用户界面分为现有确诊、累计确诊、现有疑似、累计治愈和累计死亡五大模块。

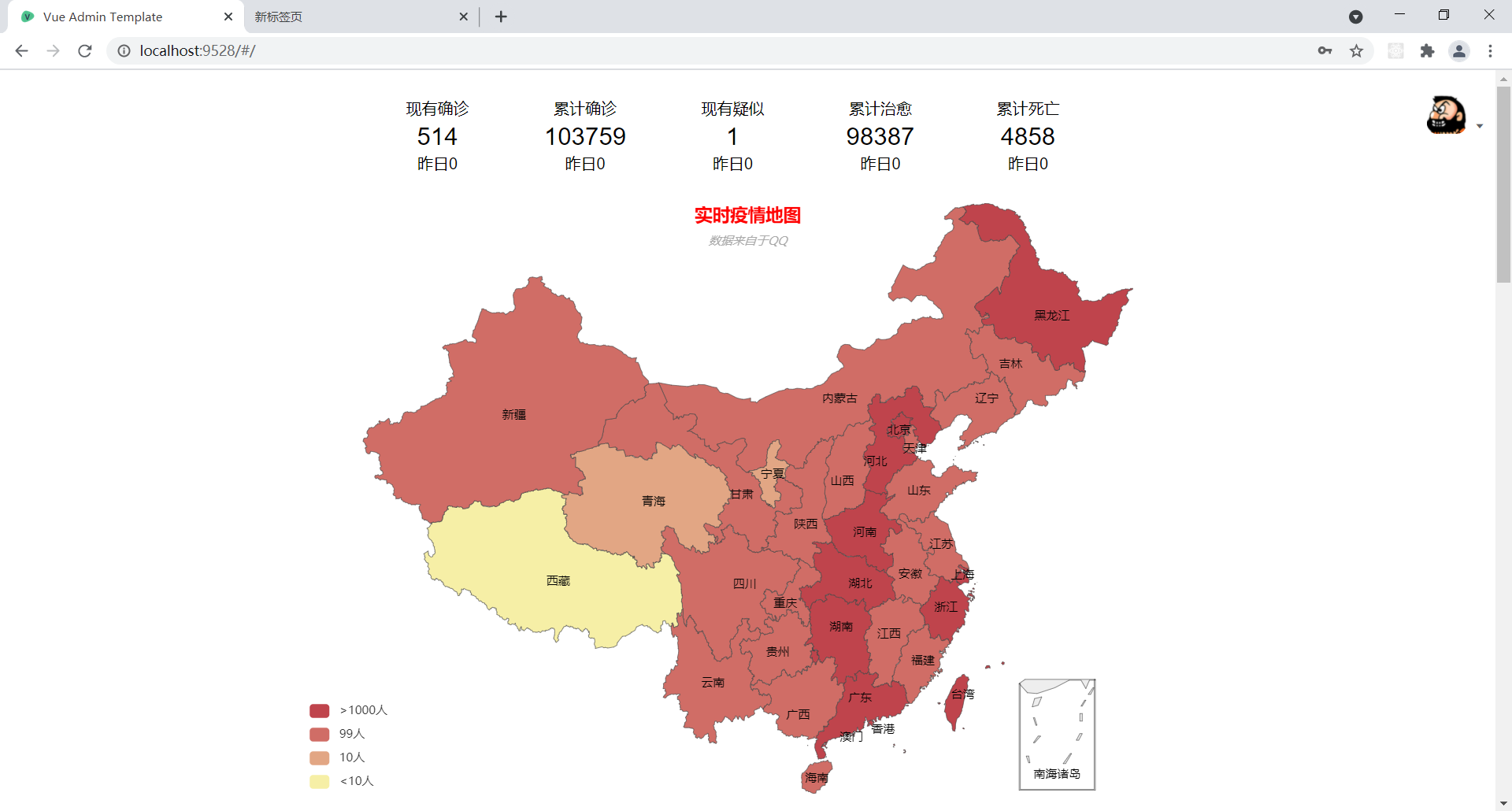


图5.3 用户界面

当用户点击某一模块的按钮时，地图上展示的就是当前模块的疫情数据。同时，折线图、南丁格尔 玫瑰图以及条形图也会展示对应的疫情数据。如图5.4、图5.5和图5.6所示。

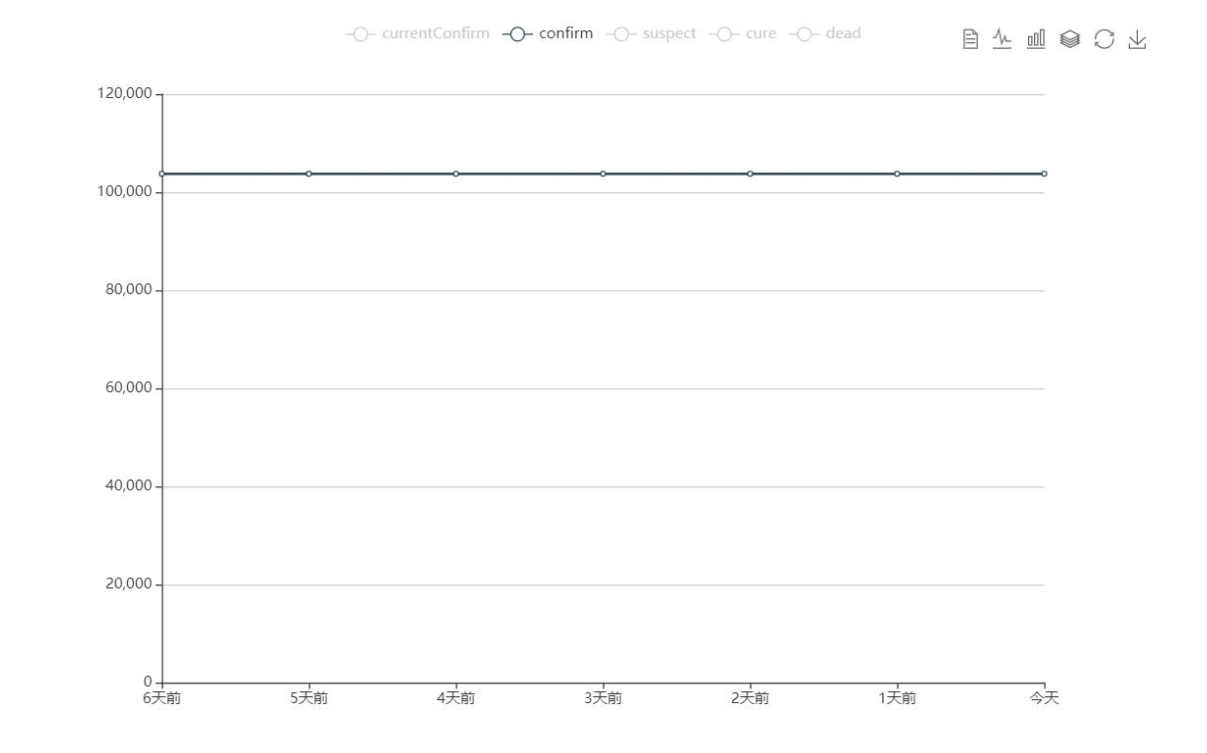


图5.4 折线图

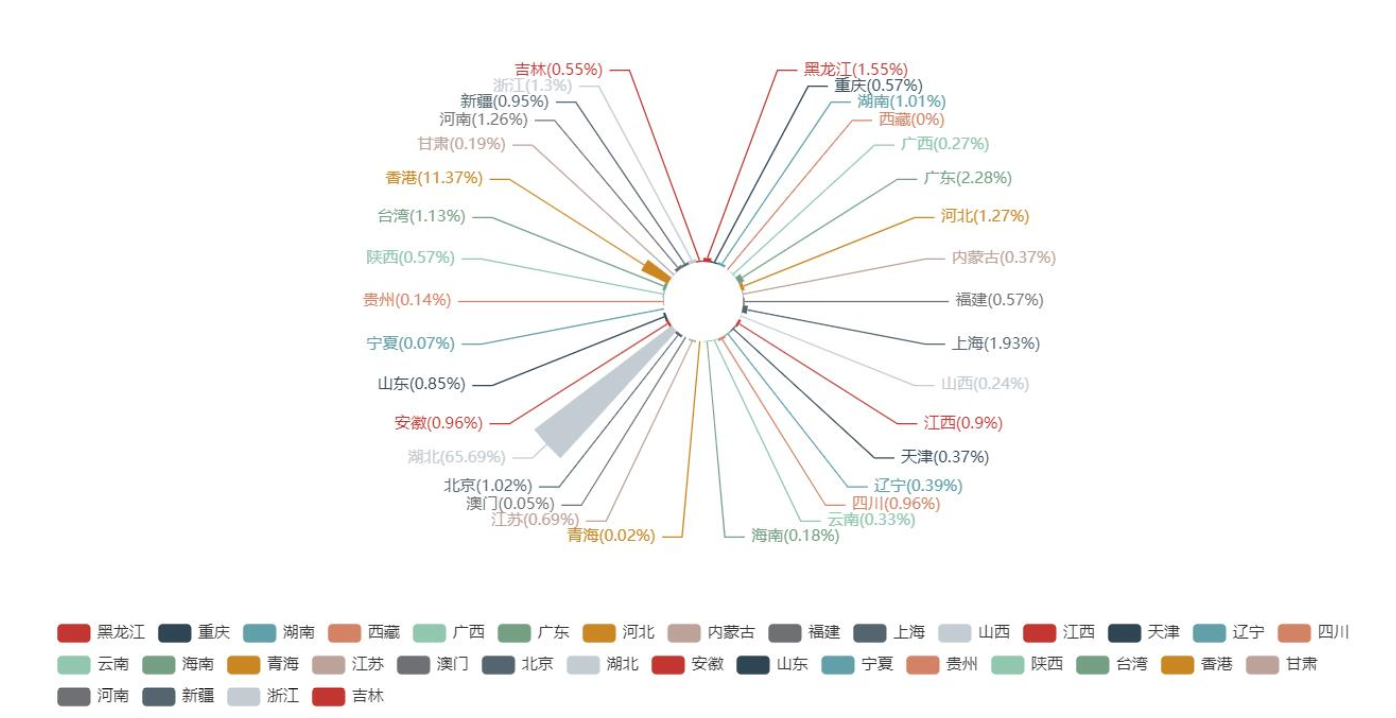


图5.5 南丁格尔 玫瑰图

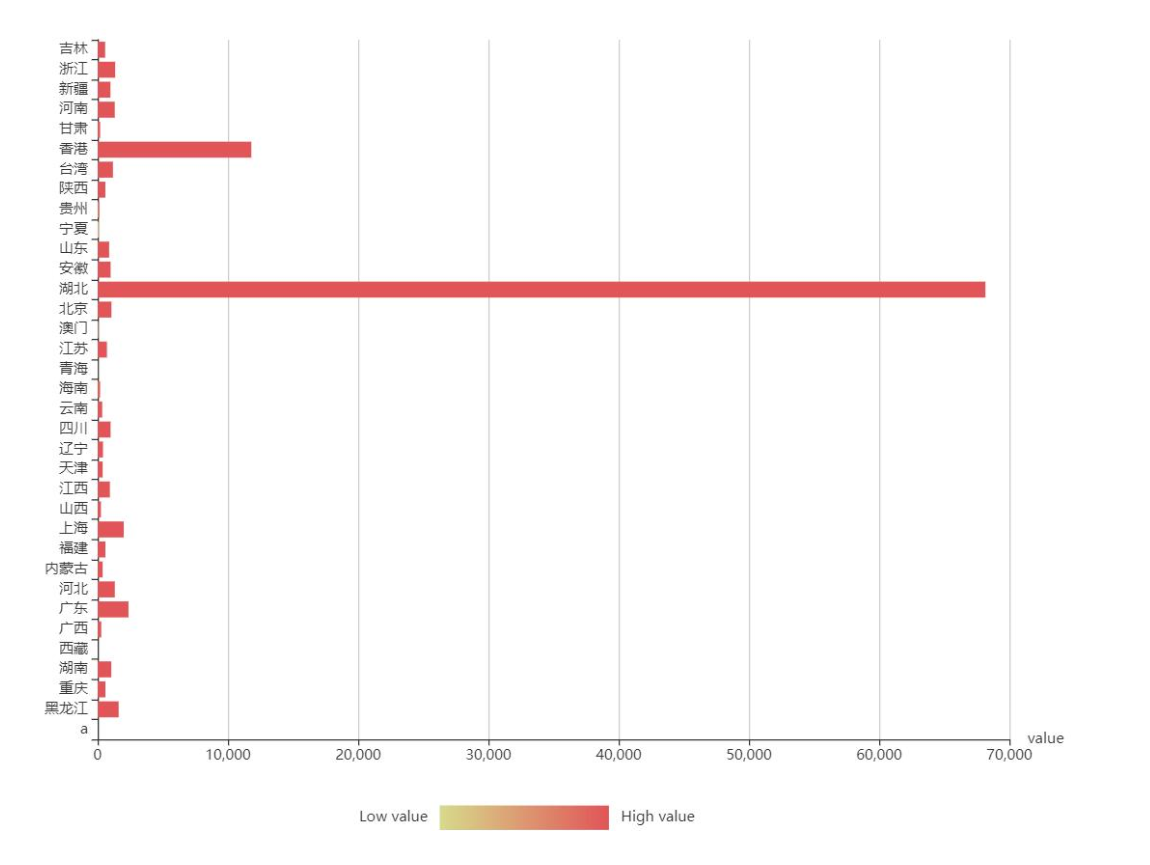


图5.6 条形图

若用户点击地图上的某一个省份，则地图、折线图、南丁格尔 玫瑰图以及条形图展示该省份相应的疫情实时数据。

表5.1是对用户界面的功能进行测试的结果。

表 5.1 用户界面功能性测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 测试步骤 | 预期结果 | 实际结果 | 测试结果 |
| 现有确诊 | 点击现有确诊按钮 | 地图、折线图、南丁格尔 玫瑰图以及条形图展示现有确诊的数据 | 同预期 | 通过 |
| 累计确诊 | 点击累计确诊按钮 | 地图、折线图、南丁格尔 玫瑰图以及条形图展示累计确诊的数据 | 同预期 | 通过 |
| 现有疑似 | 点击现有疑似按钮 | 地图、折线图、南丁格尔 玫瑰图以及条形图展示现有疑似的数据 | 同预期 | 通过 |
| 累计治愈 | 点击累计治愈按钮 | 地图、折线图、南丁格尔 玫瑰图以及条形图展示累计治愈的数据 | 同预期 | 通过 |
| 累计死亡 | 点击累计死亡按钮 | 地图、折线图、南丁格尔 玫瑰图以及条形图展示累计死亡的数据 | 同预期 | 通过 |

## 5.3兼容性测试

在Microsoft Edge、Chrome和Firefox浏览器上进行了完整流程的兼容性测试，如图5.7、图5.8和5.9所示。

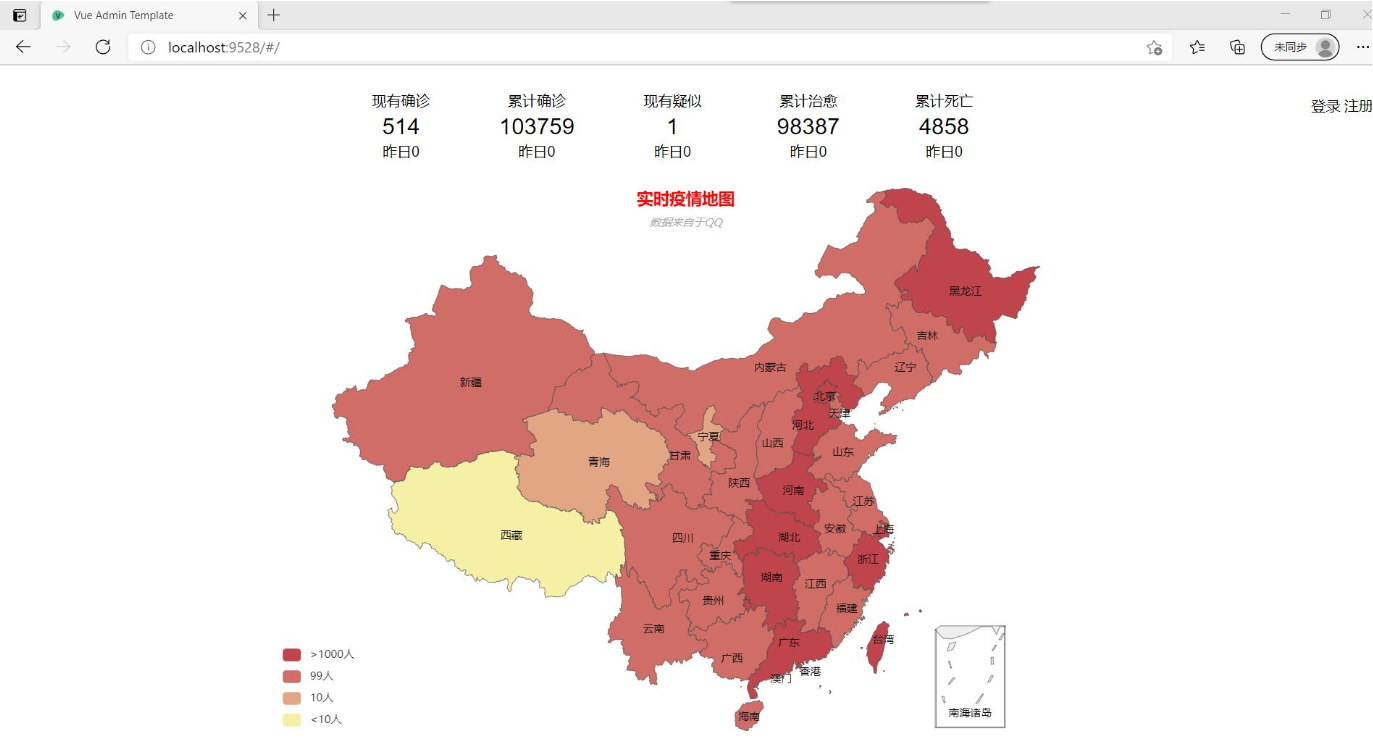


图5.7 Microsoft Edge测试

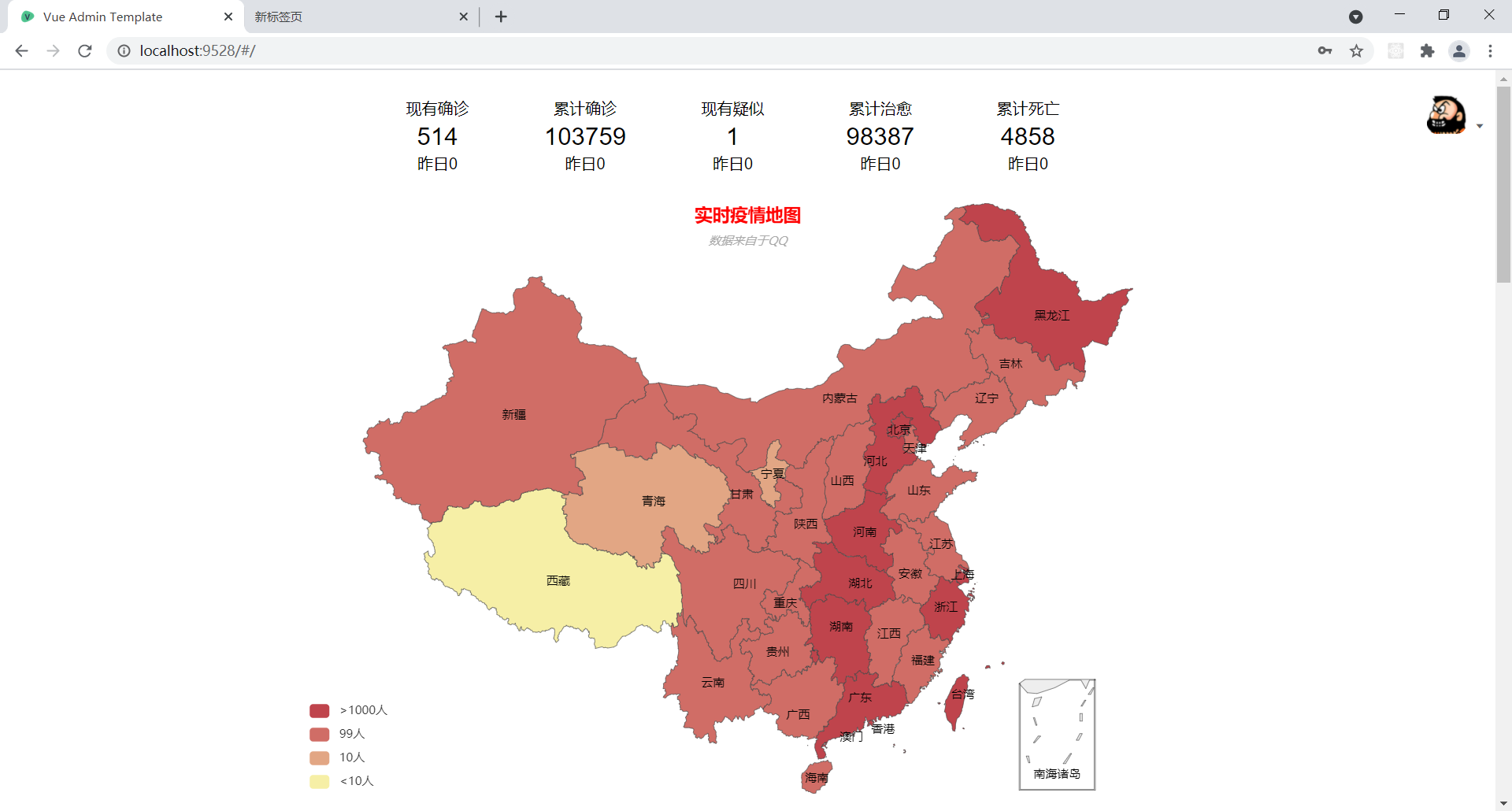


图5.8 Chrome测试

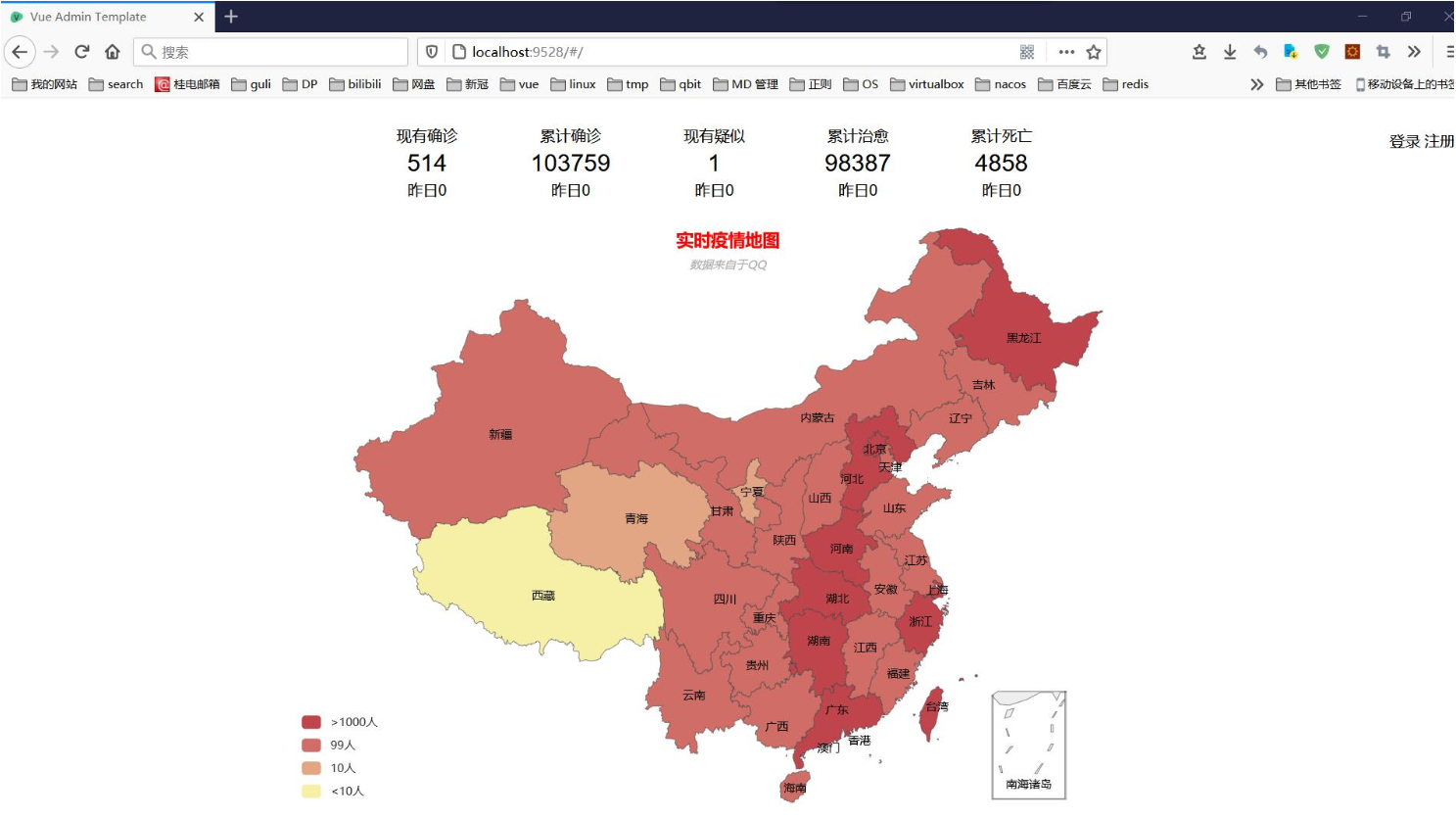


图5.9 Firefox测试

经测试，系统在不同的浏览器上运行流畅，风格没有变化，字体和图片大小合适，对界面未产生变化，系统兼容性很好。

## 5.4 测试总结

通过测试，系统包含的主要功能点如下:

（1）用户注册：未注册想查看疫情数据的用户可以进行注册。

（2）用户登录：已注册的用户可以进行登录，登录系统查看疫情的实时数据。

（3）查看现有确诊、累计确诊、现有疑似、累计治愈以及累计死亡等疫情实时数据。

该系统大部分功能达到预期效果，核心功能满足预期要求，但仍存在一些问题，例如部分功能尚未开发完成、有些错误信息没有提示等。

# 6 关键技术总结

## 6.1 crawler

爬虫模块。简单来说就是爬取https://view.inews.qq.com/g2/getOnsInfo?name=disease\_h5 的数据，进行数据清洗后插入数据库。这里 httpclient 来进行数据爬取，并用 fastjson 解析，最后用 mybatis 插入 MySQL 数据库 covid 中的 area 表。

## 6.2 visualization\_data

可视化数据模块。其功能是将 area 表中的数据读取出来并返回给前端。

## 6.3 user

### 6.3.1 登录

这里使用 jwt 生成的 token 来验证用户身份。用户发出登录请求，请求成功后会返回一个 token 给用户，之后用户根据 token 获取需要的信息。

### 6.3.2 注册

用户注册需要短信验证码，短信验证码使用阿里云短信服务+redis 实现。当用户按下获取短信按钮后，后端将会随机生成四位验证码，并存入redis，设置五分钟有效期。之后根据用户填写的手机号码调用阿里云短信服务，将四位验证码发送给用户。用户收到短信后输入验证码，按下注册按钮，后端会取出 redis 中的验证码，与用户传递过来的验证码进行比较，如果相等，则插入数据库，用户跳转到登录页面，否则提示用户验证码错误。

## 6.4 网关

这里我们使用 nginx 作为反向代理，对外暴露一个统一的端口

## 6.5 部署

centOS7+docker部署。相信在平时学习工作中大家没少吃环境搭建的苦，为了解决环境搭建问题，我们特意选择了 docker 进行部署。使用 docker 部署十分简单，只需编写好 Dockerfile，然后再 docker run ，程序就运行起来了，无需再费心搭建环境！

# 7 总结和展望

总体来看，本次课设还算很顺利。通过这次课程学习，使我们对前面所学知识有了进一步的理解和认识与掌握，使我们进一步认识到课本所学知识与实际应用其实是不一样的，在实际应用中需要具体问题具体分析，灵活的变通处理，而不是照着课本上上的知识生搬硬套。同时，我们深感只有通过具体的项目实践，才能更好的掌握所学知识，并进一步的融会贯通。这次课程设计使我们深刻认识到了一个项目的实现最重要的还是需求分析而不是代码的实现。一个系统实现的关键不是代码编写，而是分析设计，只有分析设计合理了，在后面代码实现的过程中才不会遇到问题。本次课程设计使我们再次认识到了团队协作的重要性，众人拾柴火焰高。一个人的能力毕竟是有限的，而大家的力量是无穷的，有时候一个很小的问题，自己怎么也看不出来，叫别人来帮着看一下可能马上就能解决。在这期间，体会到了大家同心协力去做一件事的快乐，也意识到从事软件开发的人要有着广博的学识，对各个行业都应有所了解。期间也遇到了其他一些难题，不过都在同学的帮助下，顺利解决了，在此向他们一并表示感谢。

当然，通过本次课程设计我们也认识到了自己对相关专业知识的掌握理解还不够透彻，对业务的需求理解不是很到位。总体而言，本次课程设计使我们对软件开发有了进一步的认识，学到了很多知识，我们感到受益匪浅。