

# 程序设计综合实验概要设计

组长：莫繁滨 2020212018

组员：王鲁顺 2020212020

李皓廉 2020212074

# 1、系统概述

## 1.1、系统简介

基于微信小程序，实现新冠疫情防控的表格、地图、折线图等可视化展示。

术语：

序号	术语或缩略语	说明性定义
1	微信小程序	微信平台的内嵌式应用功能
2	疫情地图	用色块深度表示某地区该项指标人数多少的地图
3	Flask	基于 Python 的轻量级 Web 应用框架

## 1.2、系统目标

功能目标：

1. 爬取每日疫情防控数据并处理
2. 用微信小程序展示每日疫情防控数据
3. 提供三十天内的数据查询
4. 进阶目标：展示疫情地图
5. 进阶目标：以折线图形式展示疫情变化趋势

性能目标：

1. 在获取数据阶段，遇到网页无法访问的情况，程序需要具有多次尝试的能力。
2. 前后端分离，将数据存储在专用服务器，确保用户的访问体验。
3. 对本日疫情防控数据的查询要能够精确到市。

## 1.3、系统运行环境

硬件环境：

设备	设备要求
电脑	安装微信客户端
手机	安装微信客户端

软件环境：

环境	名称	版本
操作系统	Windows	10
操作系统	iOS/Android	/
应用平台	微信	/
数据库系统	MySQL	8.0
编程平台	PyCharm	Community Edition 2020
编程平台	微信开发者工具	1.05

网络协议:

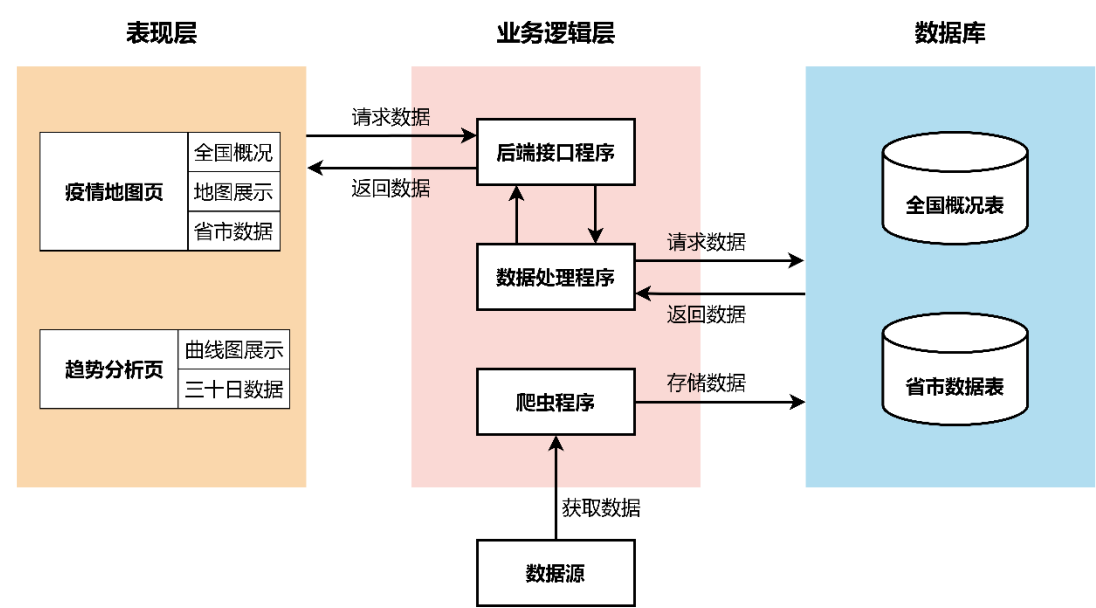
协议	概述
HTTP	应用层协议，指定了客户端可能发送给服务器什么样的消息，以及得到什么样的响应
TLS	安全传输层协议，用于在两个通信应用程序之间提供保密性和数据完整性
JSON	一种轻量级的数据交互格式

1.4、开发环境

环境	名称	版本
操作系统	Windows	10
数据库系统	MySQL	8.0
编程平台	PyCharm	Community Edition 2020
编程平台	微信开发者工具	1.05

2、总体结构设计

2.1、软件结构



2.2、设计思想

将软件分为表现层、业务逻辑层和数据库三个层级，使各层功能相对独立。

表现层：负责将数据可视化展示。采用微信小程序，将业务逻辑层传来的数据展示为疫情地图、曲线图和表格的形式。

业务逻辑层：负责获取数据、传递数据。使用爬虫程序，将数据源的数据保存到数据库中；表现层请求数据时，数据处理程序向数据库请求数据，并返回给表现层。

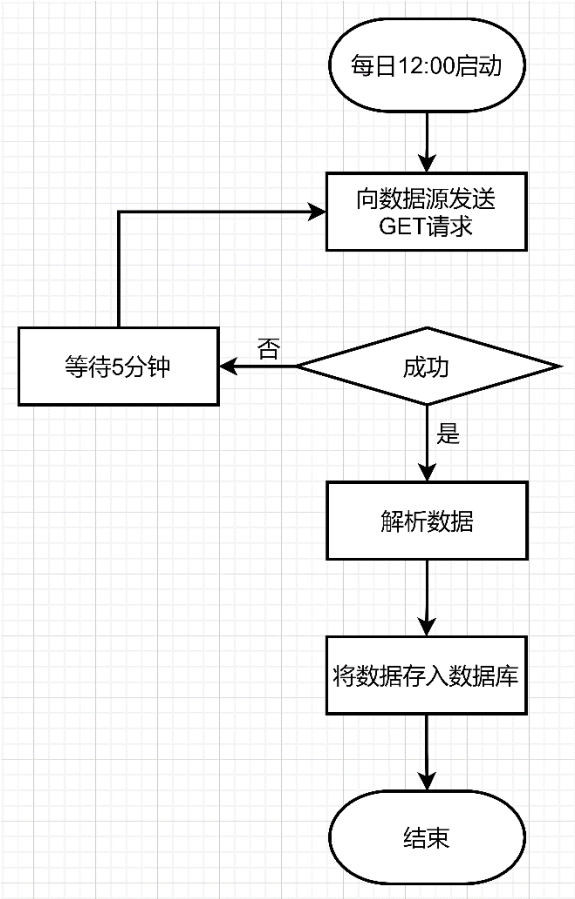
数据库：负责存储数据。包括全国概况表和省市数据表。

3、模块设计

模块名称	功能	接口
爬虫程序	从数据源获取数据	requests.get pymysql.connect
数据处理程序	接收后端接口程序请求，向数据库请求数据，处理后返回后端接口程序	parser getData
后端接口程序	接受表现层请求，向数据处理程序请求数据，处理后返回表现层	app.route
微信小程序	展示可视化数据	wx.request

3.1、爬虫程序

3.1.1、功能描述



3.1.2、接口描述

接口名称	输入信息	输出信息	异常处理
requests.get	数据源 URL	原始数据	请求失败，等待 5 分钟后重试
pymysql.connect	解析后数据	数据库记录	

### 3.1.3、数据结构描述

1. 原始数据：字符串形式的 JSON 文件
2. 解析后数据：字典形式的数据

### 3.1.4、实现思路

算法：

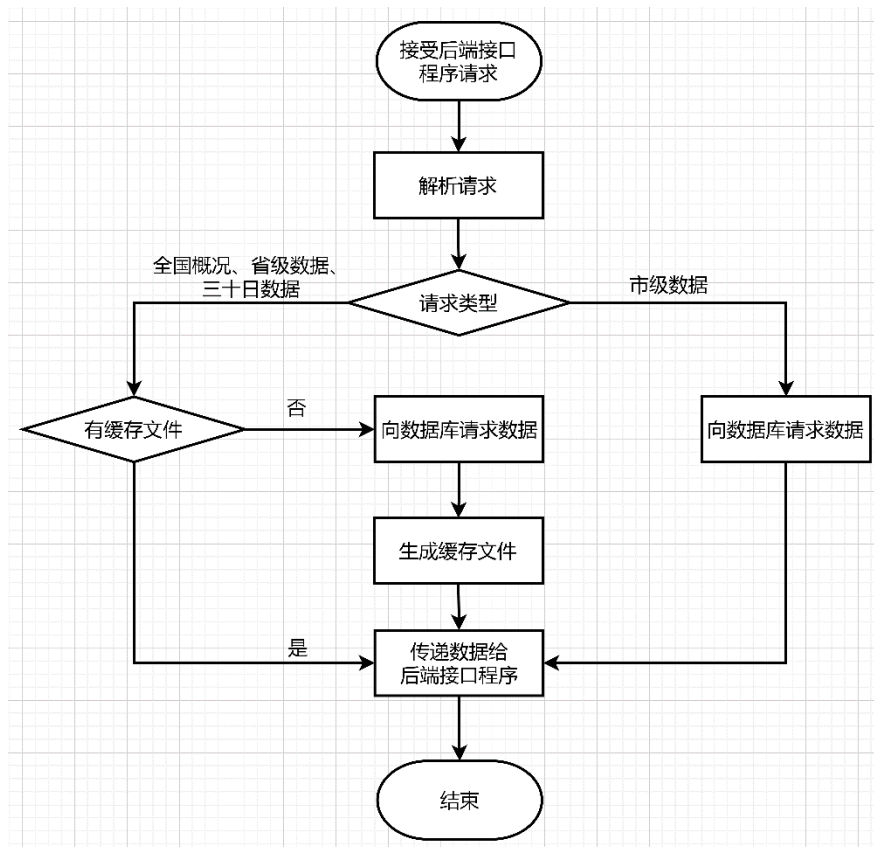
1. 通过 Python 的 requests 库，请求数据源的数据。
2. 解析数据后得到所需字段组成的字典。
3. 通过 pymysql 库连接数据库，存入数据。

### 3.1.5、其他

模块表现形式为 py 脚本，需调用 requests、pymysql 库。

## 3.2、数据处理程序

### 3.2.1、功能描述



### 3.2.2、接口描述

接口名称	输入信息	输出信息	异常处理
parser	后端接口程序请求	用户请求的数据	
getData	请求类型	用户请求的数据	

### 3.2.3、数据结构描述

1. 后端接口程序请求：字符串形式
2. 用户请求的数据：字符串形式的 JSON 文件

#### 3.2.4、实现思路

##### 算法：

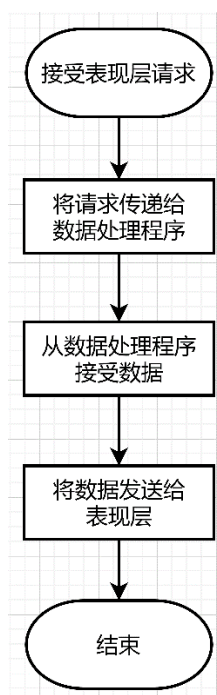
1. 接收后端接口程序请求并分析。
2. 若请求的是全国概况、省级数据或三十日数据等通用数据，则查看是否存在缓存，若有缓存则直接传递缓存文件，若无缓存则请求数据库生成 JSON 格式字符串并缓存。若请求的是市级数据的个性化数据，则请求数据库生成 JSON 格式字符串。
3. 将所得数据发送到后端接口程序。

#### 3.2.5、其他

模块表现形式为 py 脚本，需调用 pymysql 库。

### 3.3、后端接口程序

#### 3.3.1、功能描述



#### 3.3.2、接口描述

接口名称	输入信息	输出信息	异常处理
app.route	表现层请求（URL）	用户请求的数据	

#### 3.3.3、数据结构描述

1. 表现层请求：字符串形式的 URL
2. 待发送数据：字符串形式的 JSON 文件

#### 3.3.4、实现思路

#### 算法:

1. 接受微信小程序的请求, URL 形如 “https://域名/请求类型”。通过 Flask 框架将请求类型取出, 作为数据处理程序 parser 接口的参数。

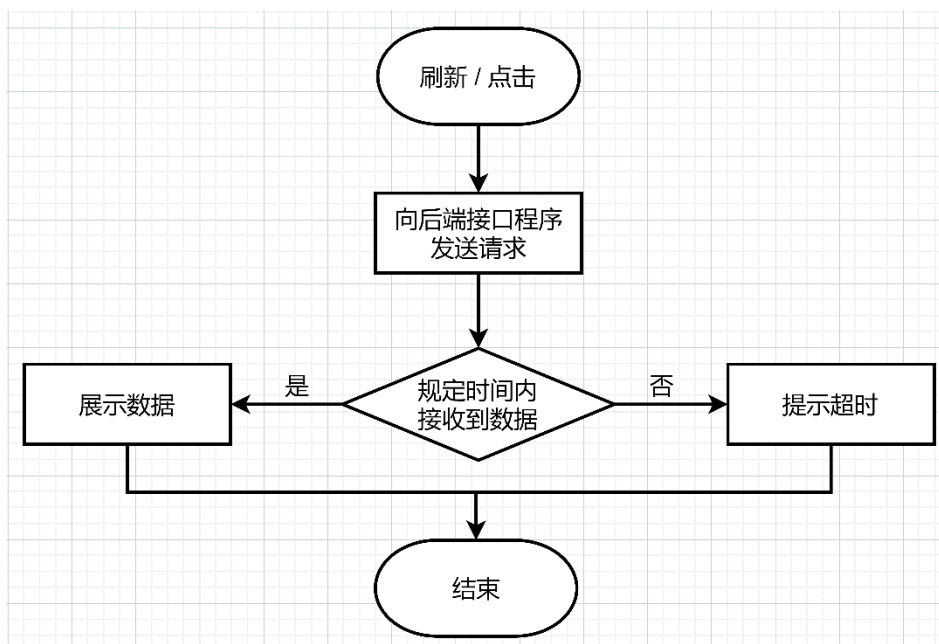
2. 接收到数据处理程序生成的 JSON 文件后, 将其发送到表现层。

#### 3.3.5、其他

模块表现形式为 py 脚本, 需调用 Flask 库。

### 3.4、微信小程序

#### 3.4.1、功能描述



#### 3.4.2、接口描述

接口名称	输入信息	输出信息	异常处理
wx.request	数据请求 (URL)	用户请求的数据	若超时则发出提示

#### 3.4.3、数据结构描述

1. 从后端接口程序接收的数据: 字符串形式的 JSON 文件
2. 页面所需数据: JavaScript 对象

#### 3.4.4、实现思路

**内部结构:** 包括疫情地图页和趋势分析页两个页面。其中疫情地图页包括全国概况、地图展示和省市数据, 趋势分析页包括曲线图展示和三十日数据。

**算法:** 当初次进入页面或刷新页面时, 向后端接口程序请求本页所需数据, 接受到数据后渲染页面。当点击疫情地图页表格中的省份时, 向后端接口程序请求该省份数据, 接受到数据后创建新页面展示。

3.4.5、其他

**交互界面类型：**微信小程序

**设计原则：**主色调为蓝色和白色。疫情地图页和趋势分析页切换通过微信小程序原生 TabBar 实现，疫情地图和曲线图用 Echarts 库实现。

4、数据库与数据结构设计

4.1、 数据库及数据表

**数据库系统：**MySQL

**数据库设计：**由全国概况表和省市数据表组成

**数据表设计：**

**全国概况表字段：**日期、现存确诊、境外输入、现存无症状、累计确诊、累计死亡、累计治愈

**省市数据表字段：**日期、省、市、现存确诊、累计确诊、累计死亡、累计治愈

4.2、 数据结构设计

**全国概况表：**

名称	日期	现存确诊	境外输入	现存无症状	累计确诊	累计死亡	累计治愈
标识符	date	currentConfirmed	imported	asymptomatic	confirmed	dead	cured
长度	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节

**省市数据表：**

名称	日期	省	市	现存确诊	累计确诊	累计死亡	累计治愈
标识符	date	province	city	currentConfirmed	confirmed	dead	cured
长度	4 字节	30 字节	30 字节	4 字节	4 字节	4 字节	4 字节

4.3、 数据存储设计

**存储要求：**存储 30 天内的数据，以便绘制曲线图。30 天过后数据需进行删除，减少数据库负担。

**访问方法：**通过 Python 调用 pymysql 库，连接数据库，进行增删改查。



索引：全国概况表为日期创建唯一索引；省市数据表为（日期，省）创建唯一索引。

## 5、接口设计

### 5.1、外部接口

#### 5.1.1 全国概况数据

URL: <https://lab.isaacclin.cn/nCoV/api/overall>

方法: GET

请求参数: 无

返回参数 (部分):

名称	类型	说明
currentConfirmedCount	Number	现存确诊人数
confirmedCount	Number	累计确诊人数
suspectedCount	Number	疑似感染人数
curedCount	Number	治愈人数
deadCount	Number	死亡人数
seriousCount	Number	重症病例人数
updateTime	Number	数据最后变动时间

#### 5.1.2 省市数据

URL: <https://lab.isaacclin.cn/nCoV/api/area>

方法: GET

请求参数:

名称	类型	说明
province	String	省级行政区名称

返回参数 (部分):

名称	类型	说明
provinceShortName	String	省份、地区或直辖市简称
currentConfirmedCount	Number	现存确诊人数
confirmedCount	Number	累计确诊人数
suspectedCount	Number	疑似感染人数
curedCount	Number	治愈人数
deadCount	Number	死亡人数
cities	Array	下属城市数据
updateTime	Number	数据最后变动时间

### 5.2、内部接口

接口名称	所属模块	请求参数	返回参数
requests.get	爬虫程序	数据源 URL	原始数据
pymysql.connect	爬虫程序	解析后数据	数据库记录
parser	数据处理程序	后端接口程序请求	用户请求的数据
getData	数据处理程序	请求类型	用户请求的数据
app.route	后端接口程序	表现层请求（URL）	用户请求的数据
wx.request	微信小程序	数据请求（URL）	用户请求的数据

## 6、其他设计

**安全设计：**采用前后端分离的模式。后端接口只允许 GET 请求，确保数据不会被篡改。

**内存管理：**市级详细数据通过点击操作访问，减少对用户端内存的占用。