

# 程序设计综合实验详细设计

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

**基本情况：**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。

**背景：**2020 年起，新冠肺炎的爆发给我们的生活带来了极大的影响。人们出行前需要查看某个地区是否受到疫情影响，而将疫情数据可视化展示，可以为人们的生活提供便利。

## 1.2、术语表

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

## 1.3、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、数据结构说明

### 2.1、常量

名称	功能说明	具体常量说明
start_url	需要爬取数据的 URL	https://ncov.dxy.cn/ncovh5/view/pneumonia
mapping	省级行政区的中文名和拼音缩写的对应关系	如“北京: beijing”

### 2.2、变量

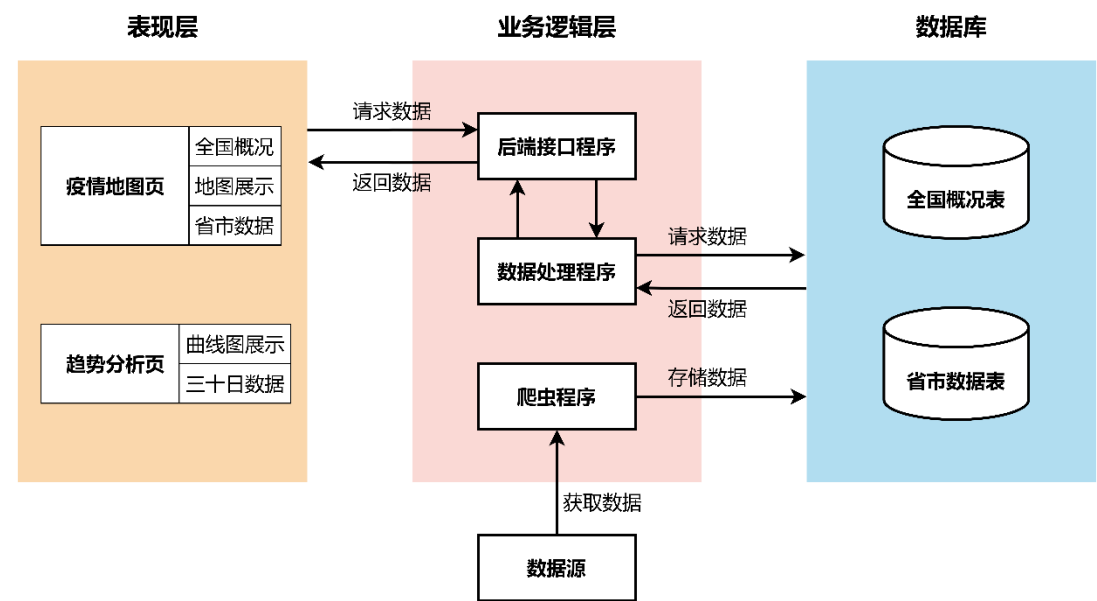
名称	功能说明	具体变量说明
overall	缓存的 JSON 文件	全国概况数据和省级数据
trend	缓存的 JSON 文件	三十日数据

### 2.3、数据结构

名称	功能说明	取值
json_data	后端接口程序向微信小程序发送的数据	用户请求的数据

## 3、模块设计

### 3.1、软件结构



### 3.2、功能设计说明

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。

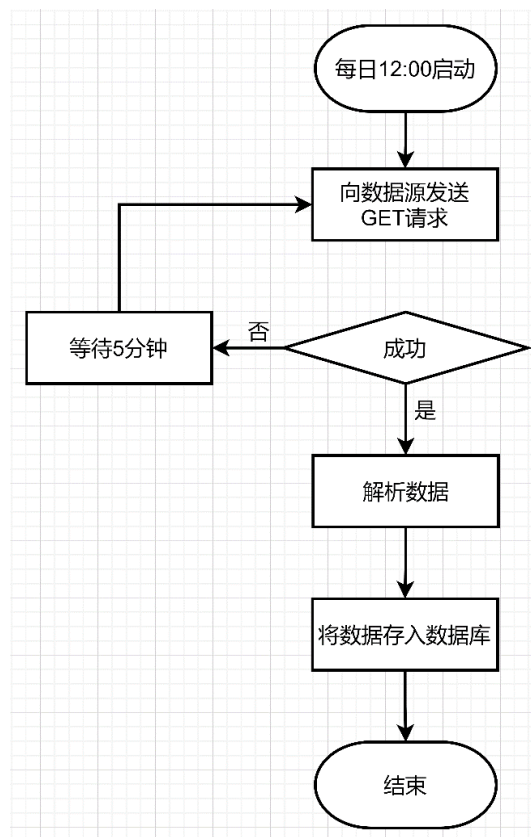
表现层：负责将数据可视化展示。采用微信小程序，将业务逻辑层传来的数据展示为疫情地图、曲线图和表格的形式。

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。

### 3.3、爬虫程序

#### 3.3.1、设计图



#### 3.3.2、功能描述

定时获取网站的疫情防控数据，解析处理后将数据存入数据库，并将数据库中的过期数据删除。

#### 3.3.3、输入数据

该模块没有输入数据。

#### 3.3.4、输出数据

疫情防控数据：数据库中的数据表，包括全国数据与地区数据。

### 3.3.5、数据设计

局部数据结构：

名称	数据结构	定义	取值
start_url	字符串	需要爬取数据的 URL	https://ncov.dxy.cn/ncovh5/view/pneumonia
overall	scrapy 框架下的类	用于保存爬取到全国数据	
area	scrapy 框架下的类	用于保存爬取到的地区数据	

相关数据库表：overall 表和 area 表。

### 3.3.6、算法和流程

(1) 定时启动爬虫程序

(2) 解析网页数据

通过函数 parse() 提取疫情防控数据，并将数据传给函数 process()。

(3) 将数据存入数据库

将当日数据存入数据库，并删除过期数据；若获取网站数据失败，则使用昨日数据更新数据库。

### 3.3.7、函数说明

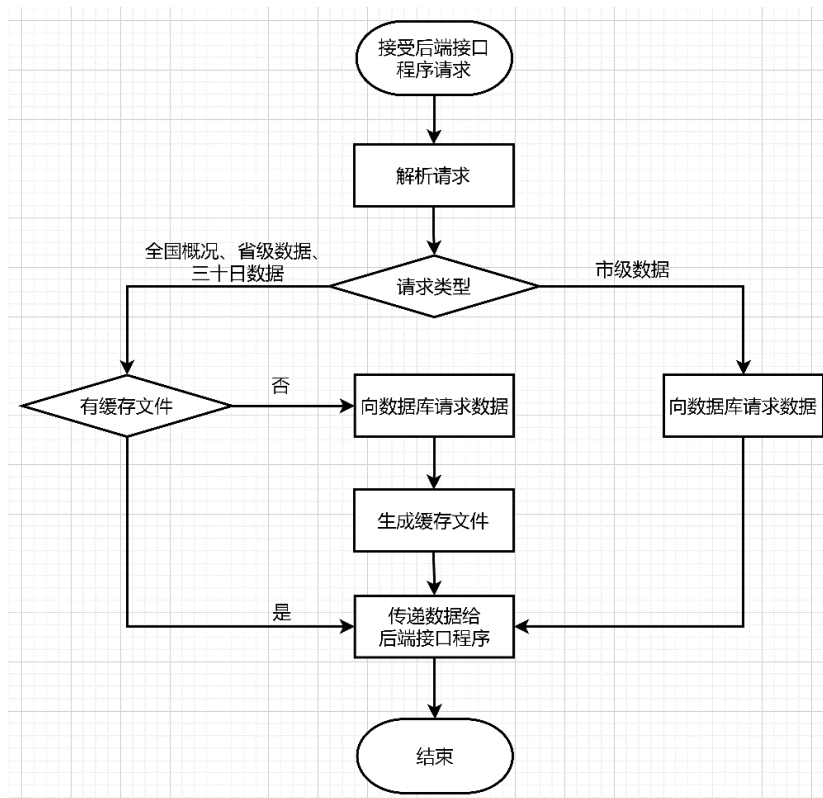
函数名称	功能	参数	返回值
start_scrapy()	定时启动爬虫程序		
parse()	解析网页数据	start_url	
process()	对爬取的数据进行处理，并存入数据库		

### 3.3.8 全局数据结构与该模块的关系

该模块未调用全局数据结构。

## 3.4、数据处理程序

### 3.3.1、设计图



### 3.4.2、功能描述

接受后端接口程序传来的请求，解析出需要请求的数据类型，向数据库请求相应数据或读取缓存文件后回传给后端接口程序。

### 3.4.3、输入数据

请求类型参数：字符串形式。“overall”表示请求全国概况和省级数据，“trend”表示用于表现趋势的三十日数据，“beijing”表示市级数据。

### 3.4.4、输出数据

缓存文件：JSON 文件。包括全国概况、省级数据文件，和三十日数据文件。

回传数据：字典形式。

### 3.4.5、数据设计

局部数据结构：

名称	数据结构	定义	取值
param	字符串	后端接口程序传递来的请求类型参数	“overall”、“trend”和各省级行政区拼音缩写
sql	字符串	由 param 解析出的 sql 语句	“select * from overall where XX”
raw_data	字典	从数据库中取得的原始数据	

json_data	字典	处理 data 后得到的数据	
-----------	----	----------------	--

相关数据库表：overall 表和 area 表。

数据存储设计：缓存 JSON 文件，记录今日的全国概况、省级数据或三十日数据。

### 3.4.6、算法和流程

#### （1）解析请求类型

根据 param 解析请求类型。

如果请求的是全国概况、省级数据或三十日数据，查找是否有今日数据的缓存文件。若有则读取，否则生成相应 SQL 语句。

如果请求的是市级数据（避免占用过多存储空间不生成缓存文件），直接生成相应 SQL 语句。

#### （2）向数据库请求数据

连接数据库，执行 SQL 语句得到 raw\_data。若今日数据未入库则使用昨日数据。

#### （3）处理数据

将 raw\_data 数据处理成微信小程序需要的 json\_data。若为全国概况、省级数据或三十日数据则生成缓存文件。

#### （4）回传数据

将 JSON 格式的数据传递给后端接口程序。

### 3.4.7、函数说明

函数名称	功能	参数	返回值
get_json()	由 param 获得 JSON 格式的数据	param	json_data
spelling()	拼音转换为地区名	地区名对应拼音	地区名

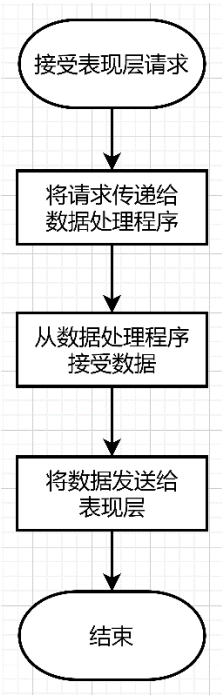
### 3.4.8 全局数据结构与该模块的关系

使用了 overall、trend 缓存文件生成 json\_data，使用了 province 文件确定省级行政区的完整中文名。

## 3.5、后端接口程序

详细描述各功能模块的功能、数据结构、具体算法和流程等。

3.5.1、设计图



3.5.2、功能描述

接受微信小程序客户端的请求，取出请求 URL 的参数后传递给数据处理程序，得到回传数据后发送给客户端。

3.5.3、输入数据

客户端请求的 URL：字符串形式。包含与请求数据类型有关的参数。

3.5.4、输出数据

回传数据：JSON 形式。微信小程序客户端渲染界面需要的数据。

3.5.5、数据设计

局部数据结构：

名称	数 据 结构	定义	取值
req_url	字 符 串	客户端请求的 URL	形如“https://域名/请求数据类型”
param	字 符 串	由 req_url 提取出来的请求数据类型参数	“overall”、“trend”和各省级行政区拼音缩写
json_data	字典	传递给微信小程序的数据	

3.5.6、算法和流程

(1) 等待客户端请求



### (2) 提取请求数据类型参数

通过 flask 框架的@app.route 装饰器装饰的 req() 函数提取 req\_url 中的 param。

### (3) 请求数据并回传

通过数据处理程序定义的 get\_json() 函数，获取数据后返回给微信小程序客户端。

#### 3.5.7、函数说明

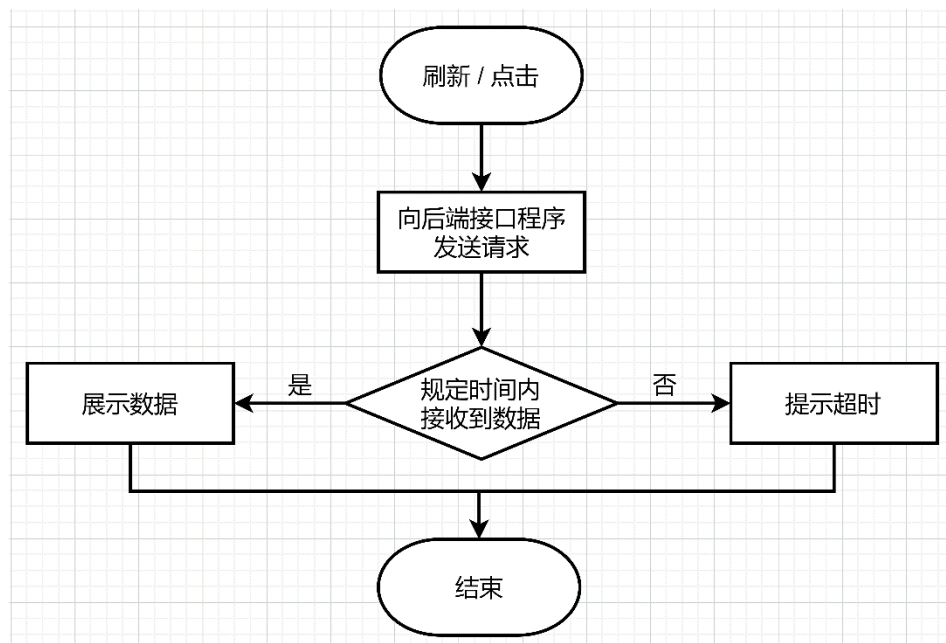
函数名称	功能	参数	返回值
data_request()	提取请求数据类型参数，调用 get_json() 后回传数据	req_url	json_data

#### 3.5.8 全局数据结构与该模块的关系

未调用全局数据结构。

## 3.6、微信小程序

### 3.6.1、设计图



### 3.6.2、功能描述

向数据处理程序请求疫情数据，并将数据以图表的形式展示在客户端上。

### 3.6.3、输入数据

中国地图的数据：JSON 形式。微信小程序客户端渲染图表需要的数据。

### 3.6.4、输出数据

客户端请求的 URL：字符串形式。包含与请求数据类型有关的参数。

3.6.5、数据设计

局部数据结构：

名称	数据结构	定义	取值
json_data	字典	接受自后端接口程序的数据	
myMap	Echarts 实例	保存渲染全国疫情分析地图相关的数据	包括 width 、 height 、 devicePixelRatio 等

3.6.6、算法和流程

- (1) 向数据处理程序发送请求
- (2) 等待数据处理程序回传
- (3) 提取回传数据类型参数

获取数据处理程序的数据 json\_data，将数据传给函数 initChartMap() 和函数 initChartLine() 渲染图表并显示。

3.5.7、函数说明

函数名称	功能	参数	返回值
initChartMap()	渲染全国疫情分布地图	json_data	
initChartLine()	渲染各省市疫情折线趋势图	json_data	

3.6.8 全局数据结构与该模块的关系

使用了 province 文件确定省级行政区的拼音缩写。

4、接口设计

4.1、用户接口

程序无向用户提供的街口。

4.2、外部接口

4.2.1 全国概况数据

URL: <https://ncov.dxy.cn/ncovh5/view/pneumonia>

方法: GET

请求参数: 无

返回参数（部分）:

名称	类型	说明
currentConfirmed	Number	现存确诊人数
confirmed	Number	累计确诊人数
overseas	Number	境外输入人数
cured	Number	治愈人数
dead	Number	死亡人数
asymptomatic	Number	现存无症状人数
update	Number	数据最后变动时间

#### 4.2.2 省市数据

**URL:** <https://ncov.dxy.cn/ncovh5/view/pneumonia>

**方法:** GET

**请求参数:** 无

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
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

### 4.3、 内部接口

#### 4.3.1、 接口说明

微信小程序通过 URL 向后端接口程序发出请求，后端接口程序通过 Flask 的 app.route 修饰器判断请求的数据类型，取得数据后回传给微信小程序。

#### 4.3.2、 调用方式

请求的数据类型有以下三种：

URL	数据类型
<a href="https://域名/overall">https://域名/overall</a>	用于渲染地图页的数据
<a href="https://域名/trend">https://域名/trend</a>	用于渲染折线图页的数据
<a href="https://域名/beijing">https://域名/beijing</a>	用于渲染地图页中市级数据子页的数据

## 5、数据库设计

数据库系统：MySQL

数据库设计：建立名为 nCoV 的数据库，由全国概况表 overall 和省市数据表 area 组成。

数据表设计：

全国概况表：

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

省市数据表：

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

## 6、系统出错处理

### 6.1、 出错信息

错误	输出信息形式	输出信息含义	处理方式
数据源无法访问	字符串	提示数据源无法访问	等待 5 分钟后重新访问
小程序请求数据超时	微信小程序弹框	提示请求超时	用户选择是否重新请求

### 6.2、 补救措施

再启动技术：数据源无法访问时，等待五分钟后再次启动爬虫程序，对数据源进行访问。

## 7、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。