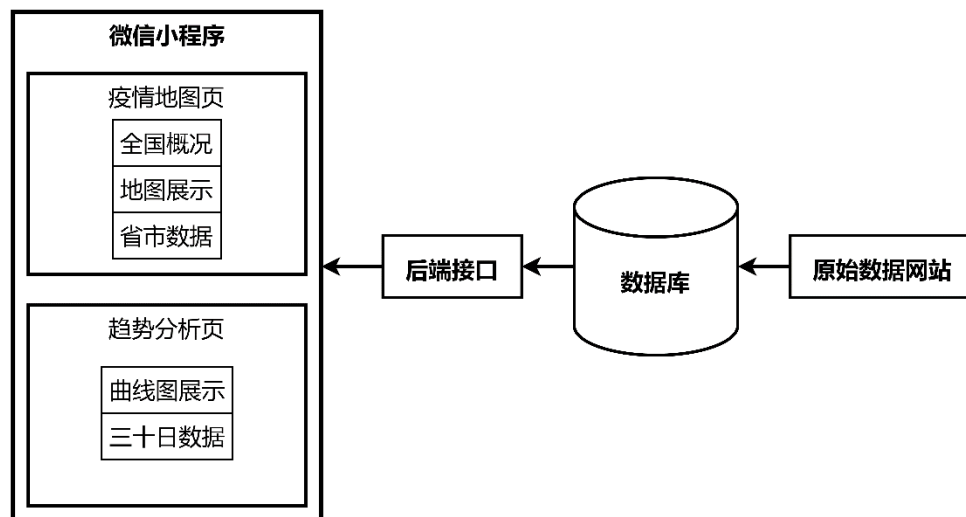


1、任务概述

1.1、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



1.2、系统（或用户）的特点

系统特点：

1. 前后端分离，数据存储于服务器
2. 展示疫情地图和折线图，可视化程度高

用户特点：关心疫情防控的用户，预期使用频度为一日一次

1.3、假定和约束

开发期限：

1. 需求分析：教学周 3-4 周
2. 概要设计：教学周 5-7 周

3. 详细设计：教学周 8-10 周
4. 编码调试：教学周 11-14 周

开发约束：

1. 微信小程序限制代码包在 2M 以内
2. 尽量将资源使用（如公网流量、数据库存储等）限制在云托管免费额度内

2、需求规定

2.1、软件功能说明

功能	输入量	处理	输出
获取每日疫情防控数据	原始数据网站	HTML 文本分析	以文件形式存储的疫情防控数据
数据列表展示	以文件形式存储的疫情防控数据	排序	小程序展示列表
存储历史数据	本日数据	存入数据库	历史数据库
（进阶）疫情地图展示	以文件形式存储的疫情防控数据	绘制地图	小程序展示地图
（进阶）折线图展示	三十天内疫情防控数据	绘制折线图	小程序展示折线图

产品容量：系统应支持的终端数、应支持的并行操作的用户数由微信云服务器承受能力所决定

2.2、对功能的一般性规定

1. 获取每日疫情防控数据：若原网站数据未更新或请求失败，则间隔一段时间后再次请求

2. 数据列表展示：按地区新增人数/累计人数从大到小排列
3. 疫情地图/折线图展示：可在 Python 中绘制，也可在前端实现（进阶）

2.3、用户界面

1. 设计主要采用蓝色、白色的主色调，扁平化的风格
2. 程序分为两个页面：疫情地图+今日数据、折线图+往日数据

2.4、对性能的一般性规定

2.4.1、精度

输入精度：字符串（HTML 报文）

输出精度：由字符串和整数组成的列表

2.4.2、时间特性要求

1. 初次运行前需要提前获取三十日的历史数据，以便折线图能够正常绘制
2. 每日爬取昨日数据并更新数据库
3. 展示获取最新数据的时间

2.4.3、灵活性

1. 获取每日疫情防控数据时，若原网站数据未更新或请求失败，则间隔一段时间后再次请求
2. 对原始 HTML 文本的分析需要具有普遍性
3. 程序无法正常运行时，在后台抛出错误提示以便修复

2.4.4、输入输出要求

1. 输入数据类型：字符串，即请求每日数据时产生的 HTML 报文
2. 输出数据类型：由字符串和整数组成的列表，地图与折线图等图例

2.5、数据管理能力要求

表一记录如下：

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
4 字节	4 字节	4 字节	4 字节	4 字节	4 字节	4 字节

每日新增记录大小约为 0.03KB

表二记录如下：

日期	省	市	现存确诊	累计确诊	累计死亡	累计治愈
4 字节	30 字节	30 字节	4 字节	4 字节	4 字节	4 字节

全国约有 333 个地级行政区，则每日新增记录大小约为 26KB

假设存储了 n 天的数据，则数据库的总存储量约为 (26.03*n) KB

2.6、故障处理要求

1. 文本分析失效
后果：当日数据无法及时更新
处理：检查匹配规则，重写分析程序
2. 无法请求原始数据网页
后果：当日数据无法及时更新
处理：隔一段时间后重新请求

2.7、其他专门要求

1. 可维护性：记录程序运行过程中发生的错误，并保存到后台
2. 可靠性：HTML 文本分析要具有普遍性
3. 运行环境可转换性：需要在多个设备上进行测试，保证能够适配大部分设备

3、运行环境规定

3.1、设备

安装有微信客户端的电脑或智能手机

3.2、支撑软件

操作系统：Android/IOS/Windows

编译（或汇编）程序：PyCharm/Visual Studio/微信开发者工具

数据库管理系统：MySQL

测试支持软件：微信

3.3、接口

程序的数据来源于 <https://lab.isaclin.cn/nCoV/>所提供的 API

数据通信协议：HTTPS

3.4、控制

运行方法：打开微信，搜索该小程序并运行

控制信号：用户的点击操作

4、尚需解决的问题

1. 采用什么框架实现后端接口
2. 如何针对成员所掌握的语言和技能进行合理分工