

1 综述

1.1 背景：

新冠肺炎疫情是一次全球性公共卫生事件，在受疫情的影响下，全世界数以亿计的民众迫切想要了解当前的疫情数据。在几个月前，阿里、百度、腾讯等企业都实现了自己的疫情数据网站，人们动动手指就能知晓疫情形势，从而更科学地做好自身防护工作。

1.2 应用领域与应用对象：

所有想了解疫情基本情况的个人

1.3 应用需求：

由于新冠肺炎疫情的全球化，人们越来越关心世界各地的疫情传播情况以及数据，因此一款集合国内外疫情数据、新闻、防范知识的可视化平台是迫切所需的。

1.4 同类产品对比：

对比同类的疫情数据可视化系统，本系统的优势在于，不仅拥有诸多图表以及地图区域疫情数据展示，还拥有传播模拟，谣言、新闻搜索，防范知识等额外功能，能够极大丰富用户体验。

2.系统综述：

2.1 系统功能简介

ShowMeData 系统是一套基于网站的可视化系统，为所有关注疫情的人提供历史以及当前疫情传播的数据可视化，利用可信的网络信息来源以及团队自建的数据库系统，提供实时疫情数据的可视化；疫情相关新闻的滚动展示，可靠的辟谣搜索系统，还包含疫情新闻实时热度分析。

2.2 系统技术阐述

该系统主要引用以下技术：Vue, E-charts, Datav, AntDesign, Flask, MySQL, Webpack,

Web Crawler

Vue 是一套用于构建用户界面的渐进式前端 **JavaScript** 框架，其被设计为可以自底向上逐层应用。它的核心库只关注视图层，易于上手且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以及各种支持类库结合使用时，该技术也完全能够为复杂的单页应用提供驱动。

E-Charts 是一个使用 **JavaScript** 实现的开源可视化库，可流畅运行在 **PC** 和移动设备上，兼容当前绝大部分浏览器，底层依赖矢量图形库 **ZRender**，用于提供直观，交互丰富，可高度个性化定制的数据可视化图表。

Datav 组件库基于 **Vue** 框架，主要用于构建大屏数据展示页面即数据可视化，拥有多种类型组件可供使用，能够有效展示多种类型图表

AntDesign 作为一个完备的 **UI** 组件库，其使用 **TypeScript** 开发，提供完整的类型定义文件，提炼自企业级中后台产品的交互语言和视觉风格，拥有全链路开发和设计工具体系。

Flask 是一个使用 **Python** 编写的轻量级后端 **Web** 应用框架，其基于 **Werkzeug** 工具箱和 **Jinja2** 模板引擎，并能够通过 **SQLAlchemy** 模块便捷地链接关系型数据库，进行数据存储以及操作。

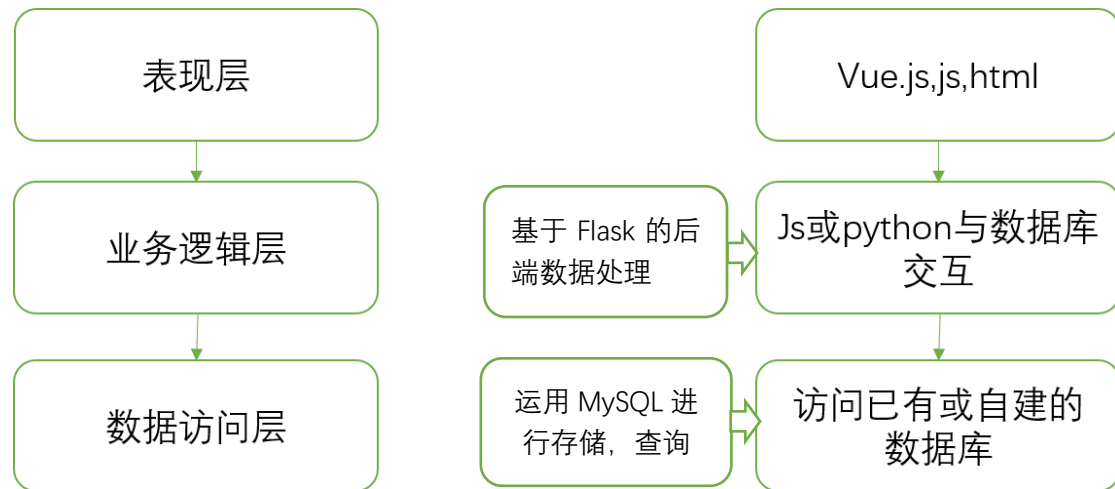
MySQL 是最流行的关系型数据库管理系统之一，在 **WEB** 应用方面，**MySQL** 是最好的 **RDBMS** (**Relational Database Management System**, 关系数据库管理系统) 应用软件之一，有着高水平、可扩展、安全、可靠和无故障运行的特点。

Webpack 是一个现代 **JavaScript** 应用程序的静态模块打包器 (**module bundler**)。当 **webpack** 处理应用程序时，它会递归地构建一个依赖关系图，其中包含应用程序需要的每个模块，然后将所有这些模块打包成一个或多个 **bundle**。

Web Crawler

Web Crawler 即网络爬虫，是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。本项目主要使用基于 **Python** 语言中 **Request** 库，**Beautiful Soup** 库在 **Robots** 协议范围内进行少量的数据爬取。

2.3 系统架构



2.4 运行环境要求

前端基础框架: Vue

后端基础框架: Flask

IE8 以上或 Chrome, Firefox, Safari

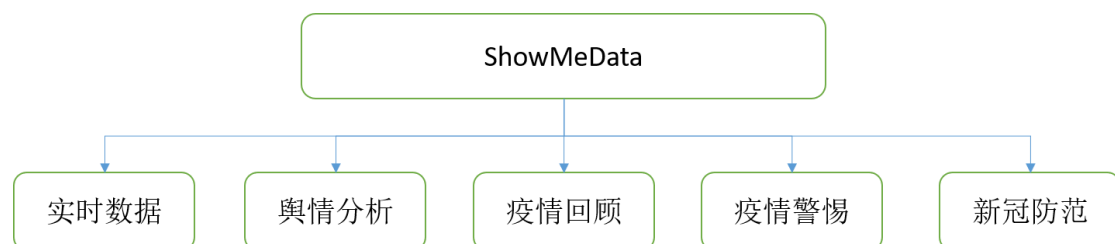
数据库: MySQL

2.5 设计方法

本系统采用传统的系统开发生命周期的方法, 采用自顶向下, 逐步求精的结构化的系统设计方法, 经历了问题定义, 可行性研究, 需求分析, 概要设计, 详细设计, 软件实现, 软件测试, 运行维护等阶段。

3.功能列表

3.1 功能结构:



本系统主要通过 WEB 可视化的方式提供新冠疫情相关信息以及提示

实时数据：主要展示当前以及历史疫情数据

舆情分析：提供实时的新闻推送和搜索热点分析

疫情回顾：主要展示疫情发展过程

疫情警惕：展示疫情模拟和谣言提醒

新冠防范：提供新冠知识的提醒和相关从业人员的提醒

3.2 具体页面的展示与说明

1.实时数据：

1. 感染人数统计：使用轮播图实现的全球累计感染人数排名的滚动展示，可通过不断滚动展示不同国家的历史疫情数据
2. 国内疫情新增展示：基于中国地图的中国疫情今日新增分布图和国内新增输入病例的玫瑰图，可通过鼠标点击或悬停的方式突出某个地区的数据。
3. 世界累计展示：基于世界地图的世界累计感染者分布图，可通过鼠标点击或悬停的方式突出某个地区的数据。
4. 疫情新闻展示：疫情相关新闻的滚动展示与内容来源分析，可通过点击链接或查看简要内容获取新闻资讯。



数据来源：

1. 全球及国内新增数据：https://interface.sina.cn/news/wap/fymap2020_data.d.json
2. 疫情相关新闻数据：<https://api.yonyoucloud.com/apis/dst/ncov/query>
3. 疫情全球排位：<https://corona.lmao.ninja/v2/countries?yesterday&sort=cases>
4. 历史谣言，新闻数据：<https://github.com/BlankerL/DXY-COVID-19-Data>

II. 舆情分析：

1. 新闻热度图：通过爬取百度指数数据，构建有关新冠肺炎关键词的热词排行表。通过查看热度图可知与疫情相关的热度词语。
2. 新闻搜索：利用网络的已有新闻数据仓库，建立 MySQL 数据库与 Flask 后端的连接，完成实时的新闻搜索功能。通过查询的方式可提供新闻的查看与检索。
3. 疫情新闻展示：疫情相关新闻的滚动展示与网页来源的链接，当用户未执行新闻搜索操作时，会默认提供当前的实时新闻。

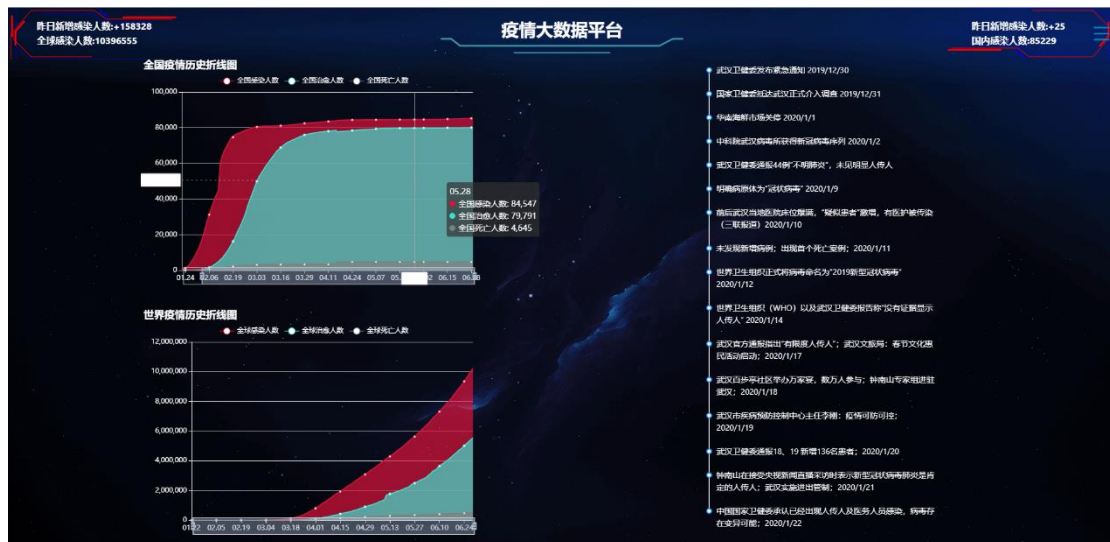


数据来源：

1. 全球及国内新增数据：https://interface.sina.cn/news/wap/fymap2020_data.d.json
2. 疫情相关新闻数据：<https://api.yonyoucloud.com/apis/dst/ncov/query>
3. 新闻查询数据：通过网络挖掘自建的新闻 Mysql 数据库
4. 实时热点：通过爬虫爬取到的百度指数接口

III. 疫情回顾：

1. 疫情折线图：利用 E-Charts 实现全国以及世界的可回顾溯源的疫情历史折线图。可通过鼠标拉拽实现不同时间段内的疫情数据查看。
2. 疫情大事件：展示新冠肺炎大事件的相关时间线。可通过标注的时间与时间内容知晓疫情初期发展过程。



数据来源:

1. 历史数据: https://interface.sina.cn/news/wap/fymap2020_data.d.json
2. 时间线信息: 通过网络检索可知

IV.疫情警惕:

1. 模拟传播: 利用 E-Charts 实现基于 JS 的模拟传染病传播过程演示模块, 能够给用户展示基础简易的模拟病毒传播过程。用户可通过设置自己的数据来模拟新冠在人群中发展的过程
2. 谣言搜索: 利用网络的已有谣言数据仓库, 建立 MySQL 数据库与 Flask 后端的连接, 完成实时的谣言搜索功能。通过查询的方式可提供谣言的查看与检索。
3. 疫情新闻展示: 疫情相关新闻的滚动展示与网页来源的链接, 当用户未执行谣言的搜索操作时, 会默认提供当前的实时新闻。



数据来源:

1. 谣言数据: 通过网络挖掘自建的谣言 Mysql 数据库

V.新冠防范：

1. 疫情防范：通过滚动栏的形式，展示对抗新冠疫情的方式方法。可通过不断滚动展示不同国家的历史疫情数据
2. 疫情工作危险指数：通过滚动的形式，展示疫情之下危险行业的危险指数，提醒相关从业人员注意防范。



数据来源：_

1. 新闻查询数据：通过网络挖掘自建的新闻 Mysql 数据库
2. 危险指数排名：网络数据

4.未来改进计划：

1. 展示平台多样化。基于更多平台（APP, 微信公众号等）的展示。
2. 画面架构合理化。优化画面元素的构建，搭建出更加合理的画面安排。
3. 展示数据深度化。搭建更加强大的爬虫系统，分析网络舆情。
4. 网站的云端化。由于本系统是基于 Flask + Vue 的前后端框架，因此可以将数据库以及后端应用程序便捷地部署在网络云端服务器上，例如阿里云等。

5. Reference：

<https://cn.vuejs.org/>

<https://echarts.apache.org/zh/index.html>

<https://www.webpackjs.com/>

<https://ant.design/index-cn>

<http://datav.jiaminghi.com/>

https://interface.sina.cn/news/wap/fymap2020_data.d.json

<https://api.yonyoucloud.com/apis/dst/ncov/query>

<https://corona.lmao.ninja/v2/countries?yesterday&sort=cases>

<https://github.com/BlankerL/DXY-COVID-19-Data>