

【疫情管控系统】

——复工及人口子系统

总体设计说明书

组长：马梓睿

组员：程梦烨，沈乐明，谢煜宁

日期：2020 年 5 月 14 日

版本：1.0

目录

| | |
|-------------------|----|
| 1. 引言 | 5 |
| 1.1. 编写目的 | 5 |
| 1.2. 项目背景 | 5 |
| 1.3. 相关定义 | 6 |
| 1.4. 系统概述 | 6 |
| 1.5. 项目概述 | 7 |
| 2. 总体设计 | 8 |
| 2.1. 需求规定 | 8 |
| 2.1.1. 系统功能 | 8 |
| 2.1.2. 系统性能 | 9 |
| 2.1.3. 输入输出要求 | 9 |
| 2.2. 运行环境规定 | 9 |
| 2.2.1. 服务器端 | 9 |
| 2.2.2. 客户端 | 10 |
| 2.3. 基本设计概念和处理流程 | 10 |
| 2.4. 结构 | 11 |
| 3. 系统结构 | 13 |
| 3.1. 系统功能结构 | 13 |
| 3.2. 技术简介 | 15 |
| 3.3. 部署图 | 16 |
| 3.4. 类图 | 16 |
| 3.5. 内部接口图 | 17 |
| 3.6. 顺序图 | 17 |
| 3.6.1. 登录验证 | 17 |
| 3.6.2. 复工举措查询 | 18 |
| 3.6.3. 全国人口流动图查询 | 18 |
| 3.6.4. 复工通知发布 | 19 |
| 3.6.5. 全国人口流动情况发布 | 19 |
| 4. 执行概念 | 19 |
| 4.1. 登录状态查询 | 19 |
| 4.1.1. 模块概述 | 19 |
| 4.1.2. IPO 图 | 20 |
| 4.1.3. 功能 | 20 |
| 4.1.4. 输入项 | 20 |
| 4.1.5. 输出项 | 20 |
| 4.1.6. 设计方法（算法） | 20 |
| 4.1.7. 流程图 | 21 |
| 4.1.8. 测试计划 | 21 |
| 4.2. 复工相关举措查询 | 22 |
| 4.2.1. 模块概述 | 22 |
| 4.2.2. IPO 图 | 22 |
| 4.2.3. 功能 | 22 |

| | | |
|--------|---------------------|----|
| 4.2.4. | 输入项 | 22 |
| 4.2.5. | 输出项 | 22 |
| 4.2.6. | 设计方法（算法） | 23 |
| 4.2.7. | 流程图 | 23 |
| 4.2.8. | 测试计划 | 23 |
| 4.3. | 人口流动图查询 | 23 |
| 4.3.1. | 模块概述 | 23 |
| 4.3.2. | IPO 图 | 24 |
| 4.3.3. | 功能 | 24 |
| 4.3.4. | 输入项 | 24 |
| 4.3.5. | 输出项 | 24 |
| 4.3.6. | 设计方法（算法） | 24 |
| 4.3.7. | 流程图 | 25 |
| 4.3.8. | 测试计划 | 25 |
| 4.4. | 发布复工相关举措 | 25 |
| 4.4.1. | 模块概述 | 25 |
| 4.4.2. | IPO 图 | 26 |
| 4.4.3. | 功能 | 26 |
| 4.4.4. | 输入项 | 26 |
| 4.4.5. | 输出项 | 26 |
| 4.4.6. | 设计方法（算法） | 26 |
| 4.4.7. | 流程图 | 27 |
| 4.4.8. | 测试计划 | 27 |
| 4.5. | 发布人员流动相关举措 | 28 |
| 4.5.1. | 模块概述 | 28 |
| 4.5.2. | IPO 图 | 28 |
| 4.5.3. | 功能 | 28 |
| 4.5.4. | 输入项 | 28 |
| 4.5.5. | 输出项 | 28 |
| 4.5.6. | 设计方法（算法） | 29 |
| 4.5.7. | 流程图 | 30 |
| 4.5.8. | 测试计划 | 30 |
| 5. | 接口设计 | 31 |
| 5.1. | 界面设计 | 31 |
| 5.1.1. | 复工举措查询界面 | 31 |
| 5.1.2. | 人口流动图查询界面 | 32 |
| 5.1.3. | 管理员发布复工举措通知界面 | 32 |
| 5.1.4. | 管理员发布人口流动信息界面 | 33 |
| 5.2. | 外部接口 | 33 |
| 5.2.1. | 获取用户登录信息 | 33 |
| 5.2.2. | 复工复产信息更新 | 33 |
| 5.2.3. | 人口流动数据更新 | 33 |
| 5.2.4. | 连接数据库接口 | 34 |
| 5.3. | 内部接口 | 34 |

| | | |
|------|---------------|----|
| 6. | 数据库设计 | 35 |
| 6.1. | ER 图 | 35 |
| 6.2. | 逻辑结构设计 | 35 |
| 6.3. | 物理结构设计 | 35 |
| 7. | 运行设计 | 36 |
| 7.1. | 运行模块组合 | 36 |
| 7.2. | 运行控制 | 36 |
| 7.3. | 出错信息 | 37 |
| 7.4. | 补救措施 | 37 |
| 7.5. | 系统维护设计 | 38 |
| 8. | 需求回溯 | 38 |
| 8.1. | 功能性需求回溯 | 38 |
| 8.2. | 性能及安全需求 | 39 |

1. 引言

1.1. 编写目的

从本阶段开始，项目将进入正式开发阶段。这份总体设计报告的编写目的，是以本项目的需求分析报告为依据，从总体设计的角度，明确疫情管控系统——复工及人口流动子系统的总体架构、流程、数据结构、数据库设计。以期达到为编程人员提供依据、为系统的修改、维护提供条件，同时明确各模块外部接口、内部接口、用户接口，项目负责人将按计划说明书的要求布置和控制开发工作全过程。

本说明书的预期读者包括：

1. 客户
2. 项目经理
3. 本系统及其他子系统的软件开发人员
4. 软件质量分析人员
5. 软件测试人员
6. 用户文档编写者

1.2. 项目背景

软件系统名称 疫情管控系统-复工及人口流动子系统

任务提出者 浙江大学软件工程基础课程任课老师-王新宇

开发者 由浙江大学 2019-2020 学年夏学期软件工程基础课程部分学生组成的项目组

用户 网站管理员，数据更新员，个人用户

实现该软件的计算机网络 阿里云服务器作为服务端，测试电脑作为客户端

相关背景介绍 浙江大学计算机科学与技术学院软件工程基础课程的综合时实践部分，通过让学生开发一个实际的软件项目，使学生掌握软件开发的基本过程与方法，包括需求分析、总体设计、详细设计、测试维护等方面的基本知识、工具与方法，并在开发过程中实践基本的项目管理方法，包括项目计划、进度跟踪与控制、风险管理、软件度量、配置管理等，同

时培养良好的团队精神，学习如何在开发过程中与其它人的合作，共同完成任务，最终完成一个软件系统的开发，提交文档和最终可运行的程序。本次课程选取疫情管控系统作为综合性实验题目。

1.3. 相关定义

MySQL 一个小型关系型数据库管理系统。

JavaScript 一种面向对象的动态类型的区分大小写的客户端脚本语言。

AJAX 一种创建交互式网页应用的网页开发技术。

SQL 注入 通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。

React 一个用于构建用户界面的 JavaScript 库。

Django 一个开放源代码的 Web 应用框架,由 Python 写成。

1.4. 系统概述

疫情管控系统是用于快速发布疫情相关权威信息的现代化管理系统。各管理员可通过各自的管理员账户在相应子系统中实时发布相关信息，而用户则可在 Web 端实时查看疫情动态变化信息从而实现对疫情的快速管控和防控知识宣传。一个完整的疫情管控系统需要的模块有用户管理及疫情新闻发布、病例检测结果发布、同乘交通自查、复工及人口流动、物资申领。具体模块的功能要求如下：

(1) 用户管理及疫情新闻发布

超级管理员建立其余四个管理员账户；允许普通游客使用个人信息（如真实姓名，身份证号码和电子邮件地址）注册个人账户，并随时修改此信息以及个人密码；允许用户以游客身份进入。新闻管理员用户在后台实时发布最新的疫情新闻。新闻分为辟谣新闻、方法知识、抗疫进展三类，在前端界面按时间顺序进行排列，个人用户可在前端查看新闻具体内容并可对新闻内容进行评价、分享，而游客不能发表评论。

(2) 病例检测结果发布

各省管理员用户在后台发布各省每日确诊、境外输入、无证感染、治愈、死亡人数，并通过累计计算得出全国现存确诊总人数、境外输入总人数、现存无证感染总人数、累计确诊人数、累计死亡人数、累计治愈人数；前端除呈现出上述数据外还应根据上述信息形成国内疫情地图，全国治愈率/死亡率趋势图，国内湖北以及非湖北地区疫情各类人数(死亡、确诊、治愈)趋势图，湖北/非湖北地区新增确诊趋势图以及各省境外输入对

比图。

(3) 同乘交通自查

管理员在后台发布高危列车、航班信息（编号、起点、终点、时间）；个人用户在填写个人具体信息（真实姓名、身份证号、手机号、地址、邮箱）后可进行同乘自查。前端显示所有高危班次且支持相关信息的模糊检索。在匹配的班次后提供登记入口，用户进入后可登记个人信息并提醒居家隔离。管理员在后台可导出各班次的用户登记信息，且支持后台邮件群发功能，从而达到快速通知用户的目的。

(4) 复工及人口流动

各省管理员在后台发布各省市的复工情况，其中包括复工复产举措、社区管控举措、交通出行举措、医疗服务举措。同时可在后台发布其余各省人口流入本省的人口数量。在前端界面除展示复工复产举措、社区管控举措、交通出行举措、医疗服务举措外，还会展示全国人口流动图。

(5) 物资申领

个人用户可在前端填写个人信息（真实姓名、身份证号、手机号、地址、邮箱）申领各类政府发放物资。管理员可在后台发布各类物资的申领入口，且对各类物资进行截止时间设定。到达截止时间将自动关闭申领入口，系统随机自动摇号将结果自动发邮件通知各参与人。系统管理员在后台可导出中签人的个人信息列表。

1.5. 项目概述

复工及人口流动模块是疫情管控系统的子系统，与其他子系统有着密切的联系。复工及人口流动子系统将涉及各省复工复产等举措的发布以及流动情况的公示，在整个系统中扮演着举足轻重的地位。

本模块设置两种身份——用户和管理员，并为其提供不同的功能。用户能够查询全国各省的复工复产、社区管控、交通出行、医疗服务等举措和全国各地流动情况，而管理员则负责及时发布更新各省复工复产等情况的通知。

为便于整个系统的整合和对接，该模块需要用到用户管理子系统的接口。

2. 总体设计

2.1. 需求规定

2.1.1. 系统功能

➤ **为用户提供复工复产举措查询**

疫情管控系统将向用户提供全国各省复工复产最新通知, 用户可根据各通知标题及大纲内容确定自己是否对此通知感兴趣并需要点击通知进入详细阅读。为了优化用户的体验, 疫情管控系统将为用户提供区域搜索栏进行通知筛选。

➤ **为用户提供社区管控举措查询**

疫情管控系统将向用户提供全国各省社区管控最新通知, 用户可根据各通知标题及大纲内容确定自己是否对此通知感兴趣并需要点击通知进入详细阅读。为了优化用户的体验, 疫情管控系统将为用户提供区域搜索栏进行通知筛选。

➤ **为用户提供交通出行举措查询**

疫情管控系统将向用户提供全国各省交通出行最新通知, 用户可根据各通知标题及大纲内容确定自己是否对此通知感兴趣并需要点击通知进入详细阅读。为了优化用户的体验, 疫情管控系统将为用户提供区域搜索栏进行通知筛选。

➤ **为用户提供医疗服务举措查询**

疫情管控系统将向用户提供全国各省医疗服务最新通知, 用户可根据各通知标题及大纲内容确定自己是否对此通知感兴趣并需要点击通知进入详细阅读。为了优化用户的体验, 疫情管控系统将为用户提供区域搜索栏进行通知筛选。

➤ **为用户提供全国各省人口流动图**

疫情管控系统将向用户提供全国各省人口流动图, 用户可点击中国地图各区域以动态了解其余各省流入本省的人口数据; 同时疫情管控系统将提供全国各省 2020 年人口流动情况柱状图。

➤ **为各省管理员提供发布通知权限**

各省管理员通过右上角“管理员登录”按钮登录疫情管控系统后, 将拥有发布复工复产、社区管控、交通出行、医疗服务通知的权限; 通知模板包含标题栏、大纲栏及内容栏。

➤ **为各省管理员提供人口流入本省情况发布权限**

各省管理员通过右上角“管理员登录”按钮登录疫情管控系统后, 将拥有发布其余各省人

口流入本省情况的权限；人口流入数据模板包含流入日期、其余各省流入人口填写栏。

2.1.2. 系统性能

对于疫情管控系统而言，拥有友好的交互环境和简单易懂的数据可视化功能对于用户体验而言是至关重要的。为此，设计者在设计账户管理时，应该尽可能地站在使用者的角度去考虑问题，真正使用户感到一致、简洁、方便、快速、有效。因此，我们对我们的性能提出如下要求：

- 界面设计应简洁直观，布局合理，清晰地呈现信息，突出重点内容。操作方便，用户容易上手。
- 系统具有良好的反应速度，给用户良好的使用体验。我们要求在良好的网络情况下，系统应具有以下时间特性要求：
 - (1) 单个用户在线时：Web 响应时间小于 1s，信息搜索响应时间小于 2s。
 - (2) 500 个用户同时在线时：Web 响应时间小于 2s。信息搜索响应时间小于 5s。
- 访问容量：该系统至少在同一时间内支持 500 个用户并发访问。
- 服务器配置最低要求：CPU2.6G，内存 2.0G，硬盘 7200 转。
- 可用性：该子系统应实现在大多数流行的 Web 浏览器中正确显示和执行，包括 Firefox、Chrome、Edge、IE 等。

2.1.3. 输入输出要求

客户端通过网页展现给用户一个友好的界面，用户可以通过提交表单或者点击超链接向服务器提供数据与命令。服务器后台处理后将结果显示到用户的网页界面上。

API 则为其他子系统和前端提供清晰、简洁的接口，子系统通过 API 向服务器发送请求，服务器后台处理后返回格式化的结果；若子系统进行非法操作，服务器能够进行判断并返回错误信息，避免发送的请求影响后端稳定性。

2.2. 运行环境规定

2.2.1. 服务器端

出于对安全性、可用性和成本的考虑，我们选择使用云服务器。

设备要求：

- CPU：不小于 2.0 GHz
- 内存：不小于 2.0 GB
- 带宽：不小于 5 M

软件依赖：

- 操作系统：Windows Vista/7/8/8.1/10, Mac OS, Linux
- 数据库平台：MySQL
- MySQL 管理软件：MySQL WorkBench 等
- 开发工具：能支持网页开发的工具均可（如 IDEA）
- 测试工具：能支持测试的工具均可（如 JEST）
- 浏览器：Chrome、Edge 浏览器
- 前端框架：Vue.js
- 后端框架：Django

2.2.2. 客户端

外围设备：

- 键盘鼠标：可正常使用
- 显示器：可正常使用
- 硬盘：不小于 100GB
- 硬盘转速：不小于 7200 rpm

通讯设备：

- 网线：正常联通且数据传输能力良好
- 网卡：100M

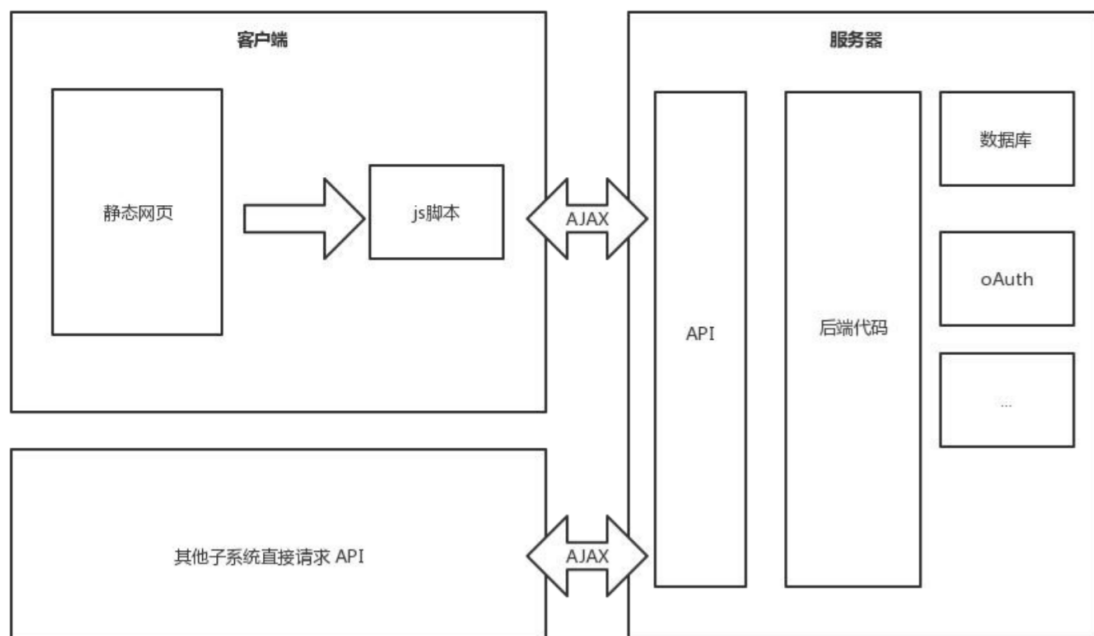
2.3. 基本设计概念和处理流程

本子系统是一个横跨前后端的大模块，主要负责复工复产举措发布及查询、全国人口流

动情况发布及查看。

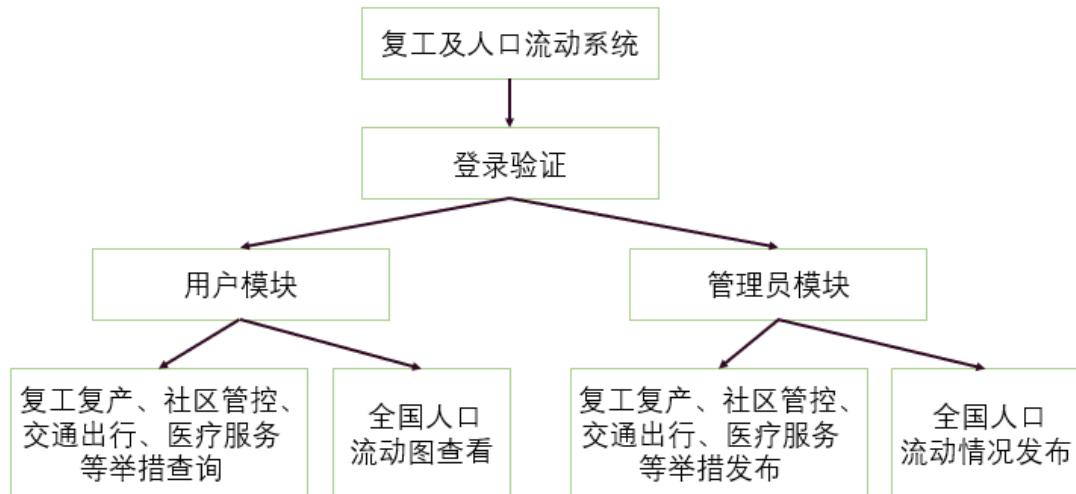
- 服务器端：基于 Python 的 Django 框架
- 数据库：数据库采用 MySQL，通过 promise 来实现异步查询
- 身份验证框架：采用 Oauth
- 客户端：考虑到本系统所使用的框架较新，推荐使用 Chrome、Firefox、Safari 或 Microsoft Edge 浏览器。不推荐使用任何一个版本的 IE 浏览器，即使使用，版本须在 11 及以上。

处理流程图如下：



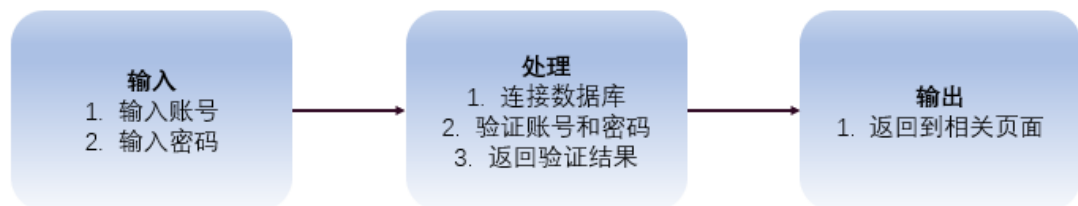
2.4. 结构

2.4.1 复工及人口流动系统层次图

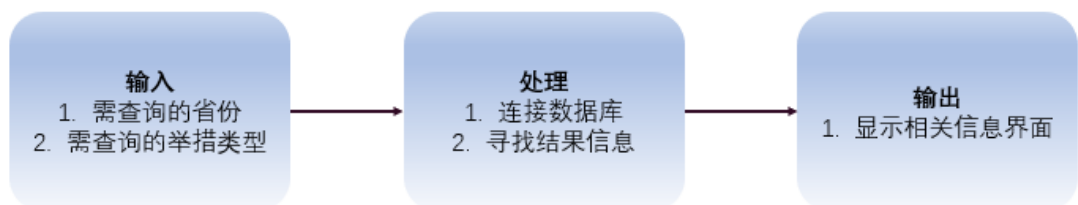


2.4.2 功能 IPO 图

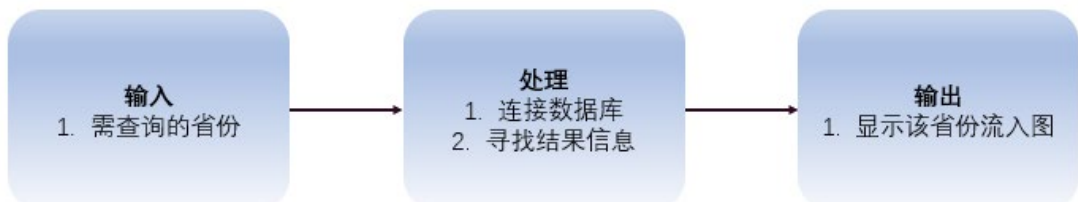
2.4.2.1 登录验证



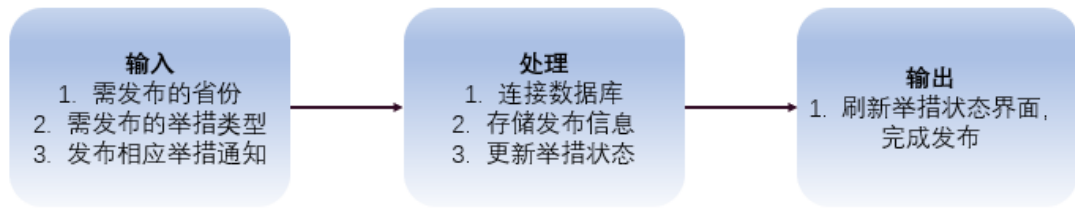
2.4.2.2 复工复产等举措查询



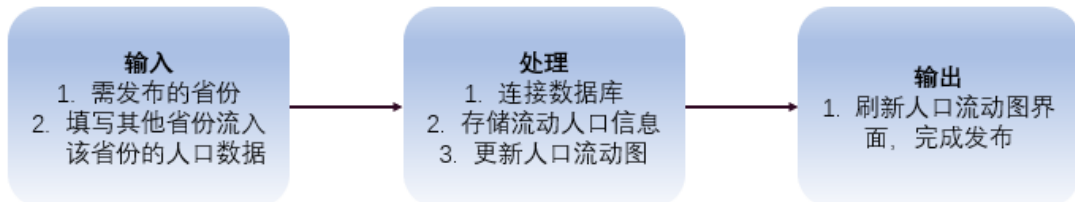
2.4.2.3 全国流动图查询



2.4.2.4 复工复产等举措发布



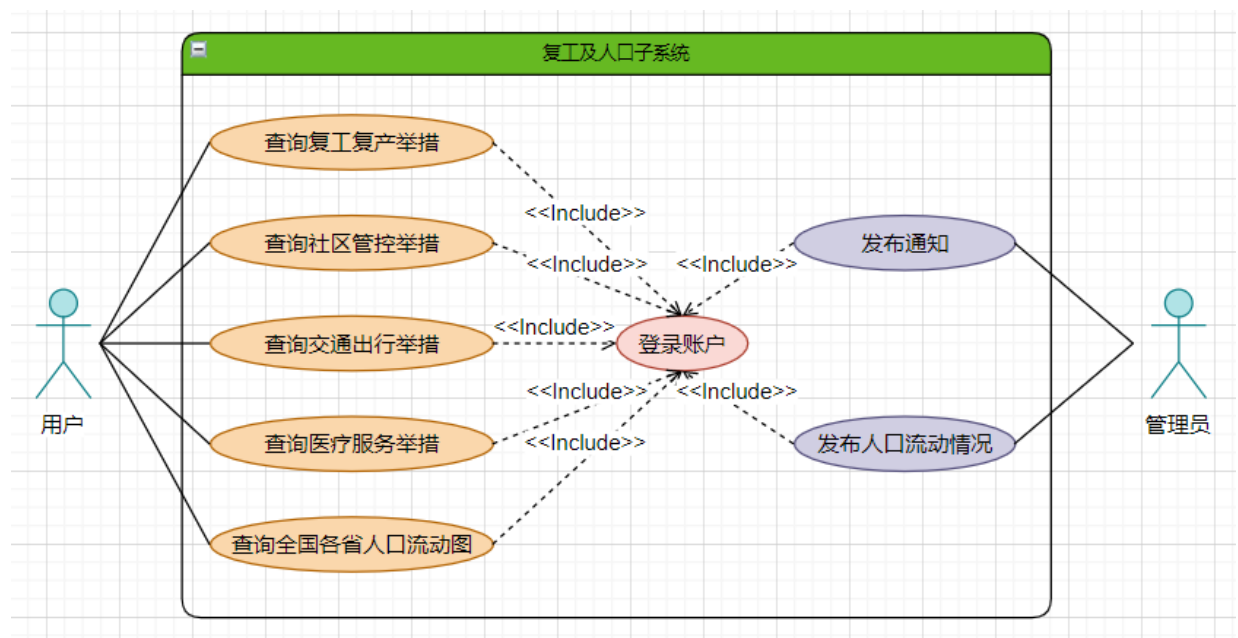
2.4.2.5 全国流动情况发布



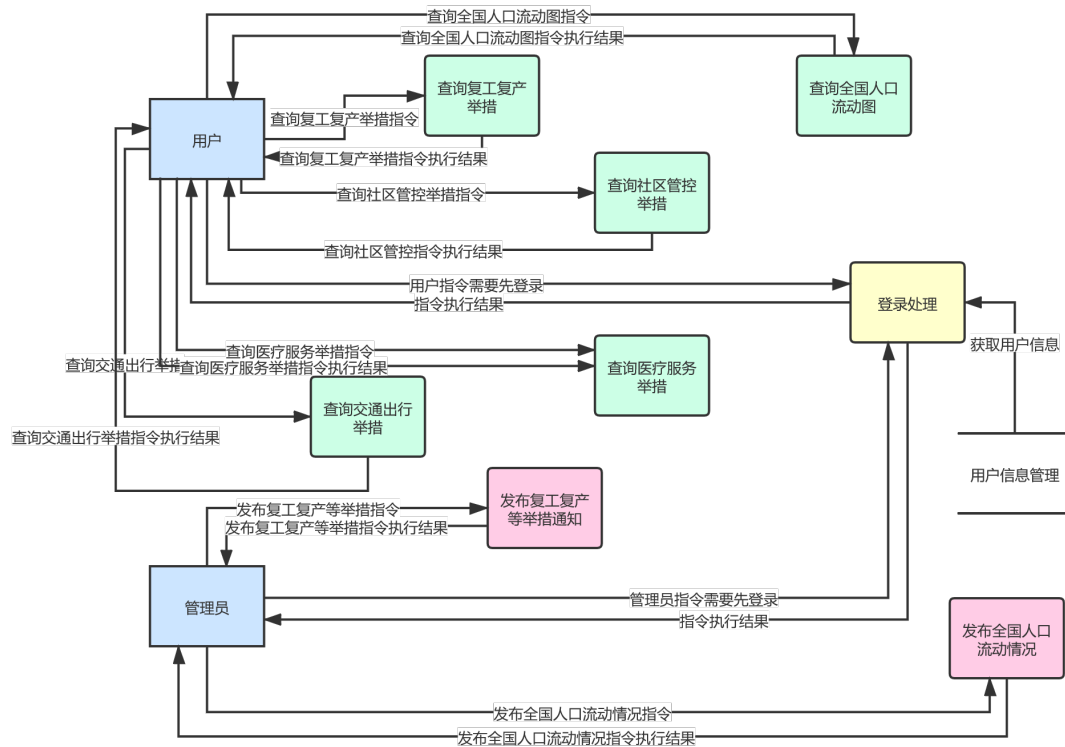
3. 系统结构

3.1. 系统功能结构

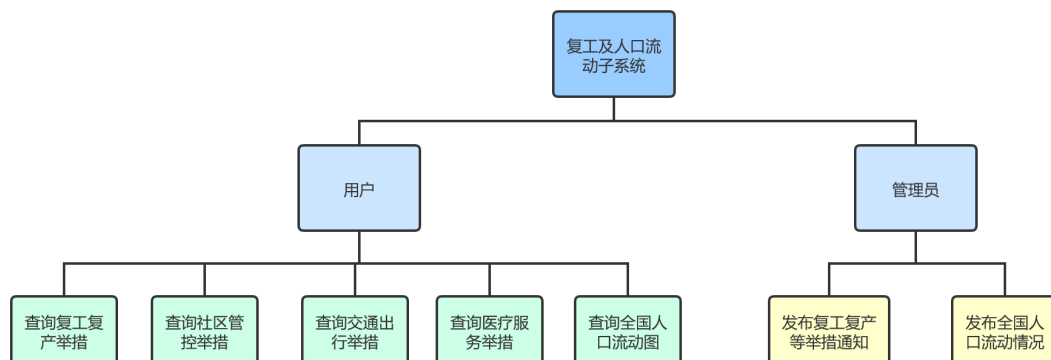
通过需求分析，已经完成对系统的用例分析，具体如下所示：



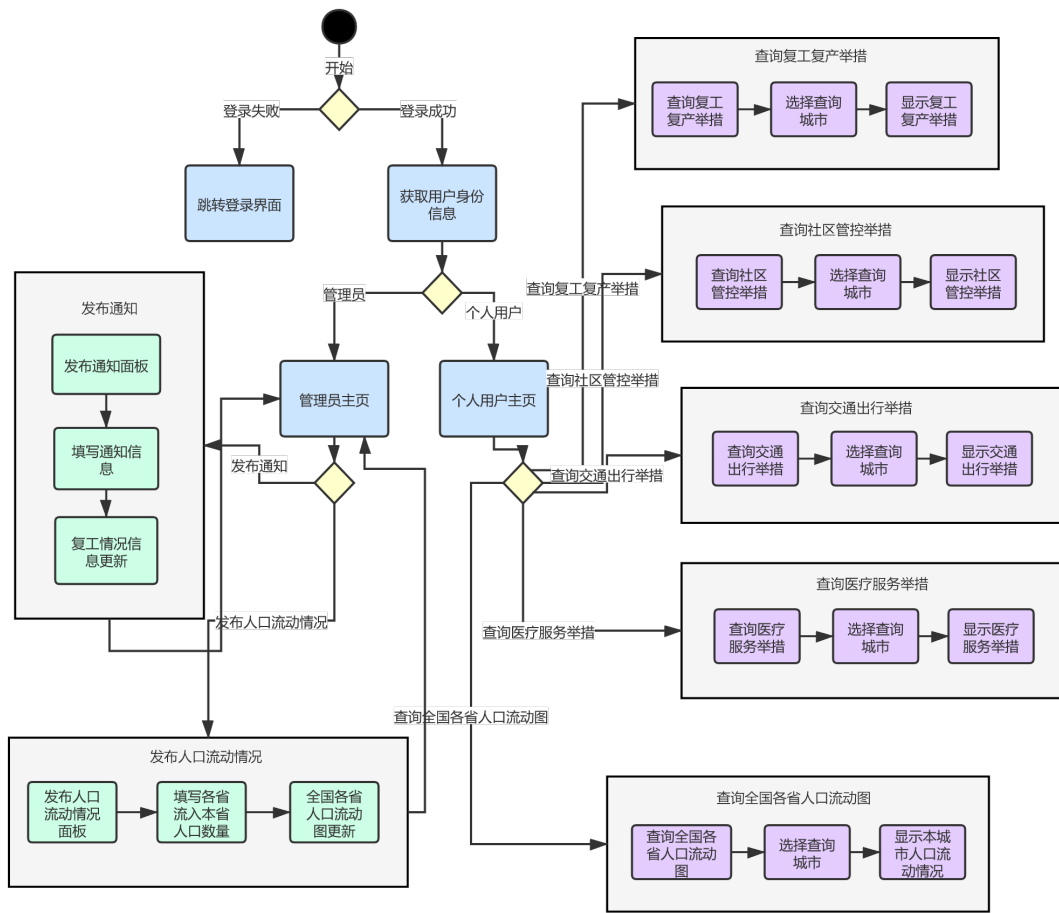
需求分析中，已经完成了对数据流相应的分析，具体如下图所示：



通过对系统需求的分析，系统的总体结构设计变得明晰了。系统的系统层次图如下所示：



需求分析中完成了系统的状态图设计，这里引用状态图作为系统运行的整体流程参考。具体如下图所示：



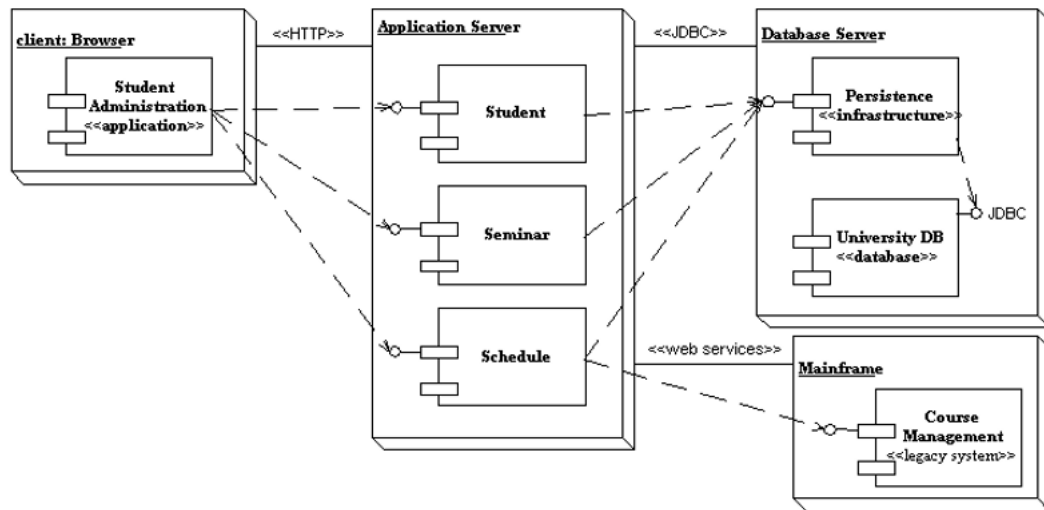
3.2. 技术简介

在系统设计中,我们将采用基于python的Django框架来进行Web页面的构建。Django是高水准的Python编程语言驱动的一个开源模型。视图,控制器风格的Web应用程序框架,它起源于开源社区。使用这种架构,程序员可以方便、快捷地创建高品质、易维护、数据库驱动的应用程序。另外,在Django框架中,还包含许多功能强大的第三方插件,使得Django具有较强的可扩展性。

此外,我们将采用MVC模型对整个项目进行设计和实现。“MVC模式(三层架构模式)(Model-View-Controller)是软件工程中的一种软件架构模式,把软件系统分为三个基本部分:模型(Model)、视图(View)和控制器(Controller)。”在这种模式下,程序的开发人员能够将模型、视图、控制器这三个模块独立开来,从而能够对各个模块进行更为方便的修改和完善。在这个模式中,“模型”不依赖“视图”和“控制器”,它用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法,能够对数据库中的数据进行直接访问。模型中数据的变化一般会通过一种刷新机制被公布,而为了实现该机制,相应的视图必须事先注册,以了解相应的数据改变。“视图”的任务则是实现数据有目的的显示。它需要访问相应的数据模型从而实现上面我们提到过的刷新功能。“控制器”则负责处理事件并作出响应。

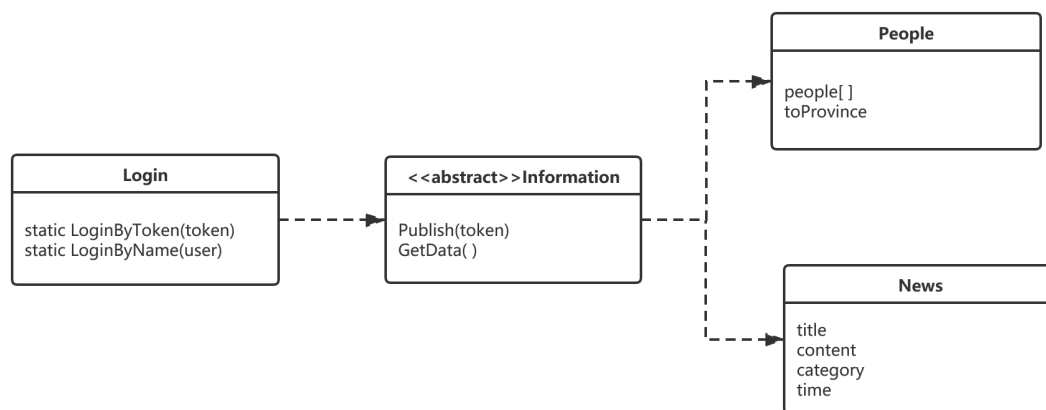
3.3. 部署图

现在，我们将就系统运行时的结构以及相应的硬件配置及其软件部署进行一个设计。这里我们将采用 UML 中的部署图对整个系统作一个规划。具体设计如下图所示：

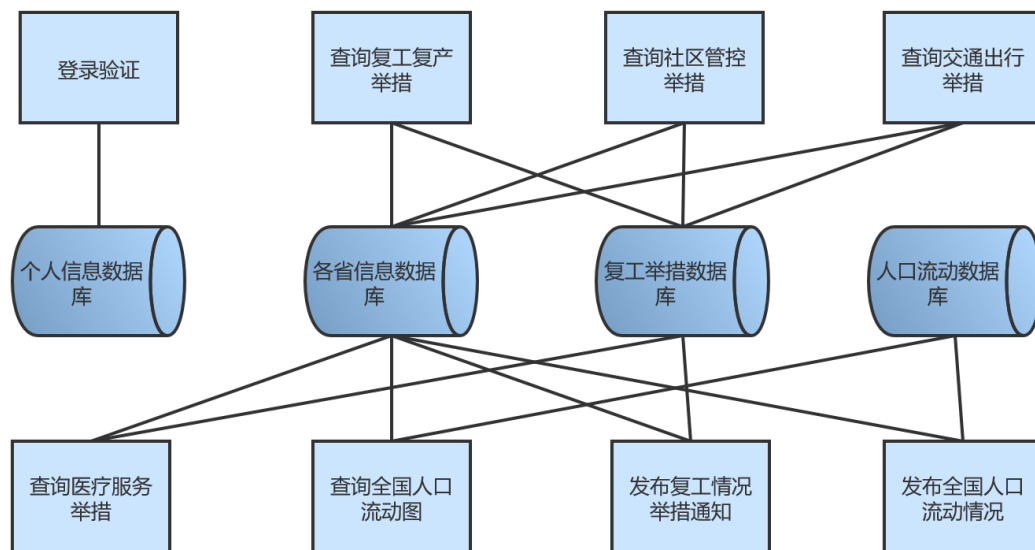


上述图是典型的的 WEB 应用的 UML 部署图，我们的应用也以上述图为基础，分为用户，服务器，数据库等核心部分。

3.4. 类图

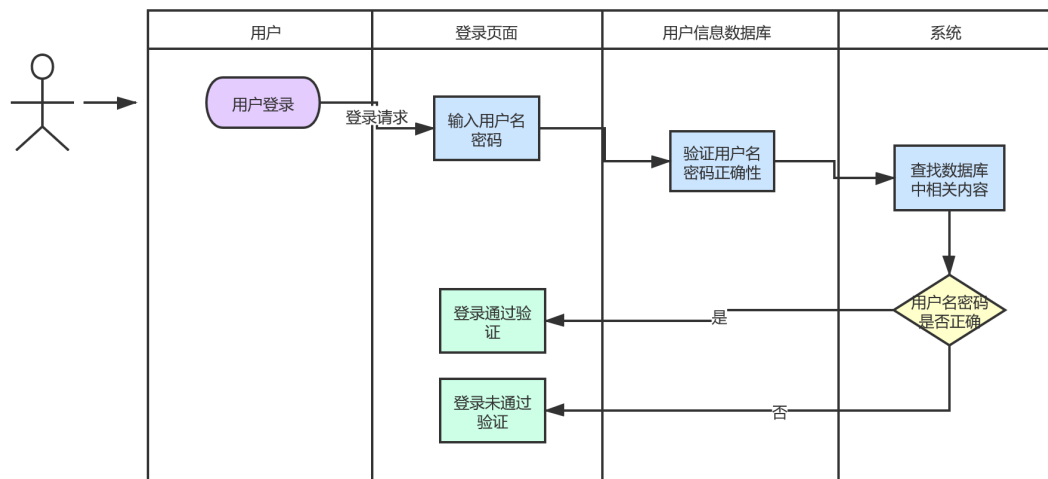


3.5. 内部接口图

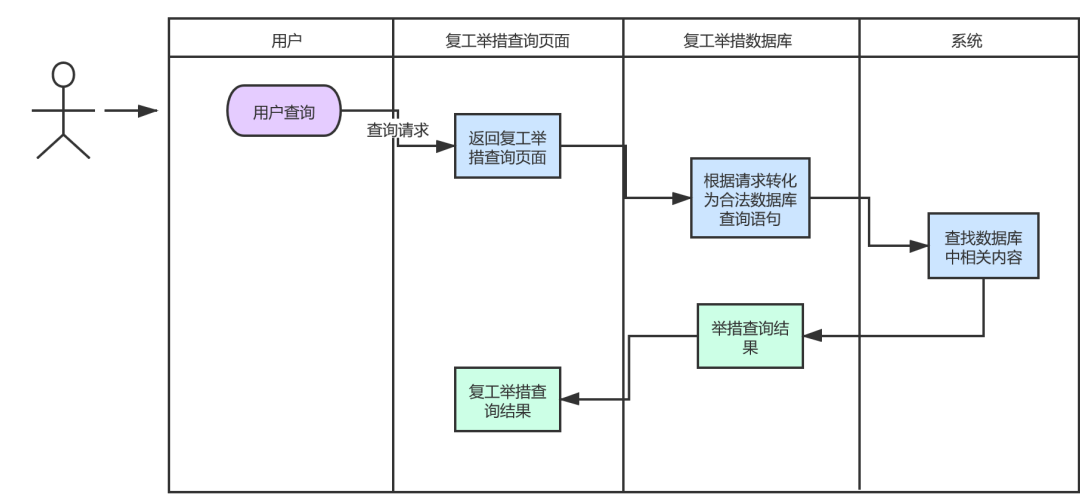


3.6. 顺序图

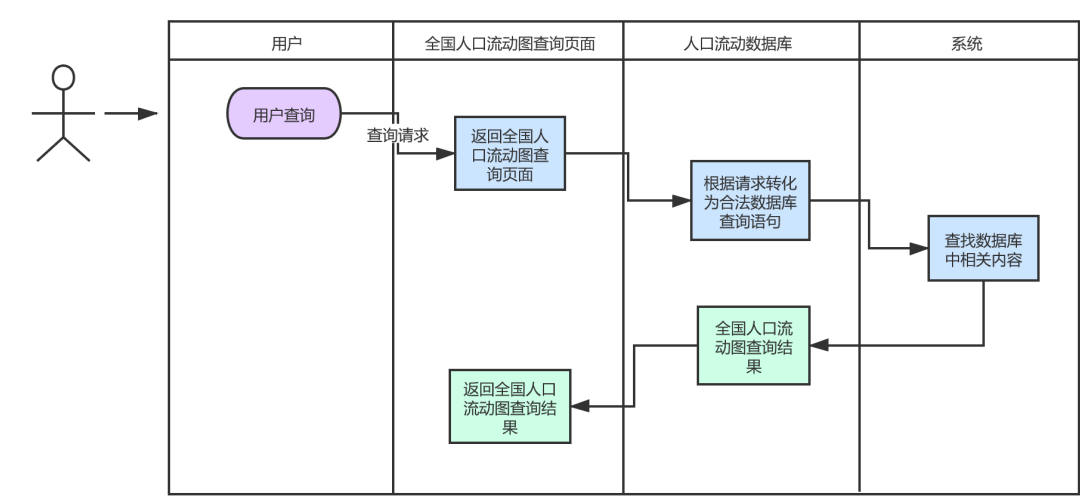
3.6.1. 登录验证



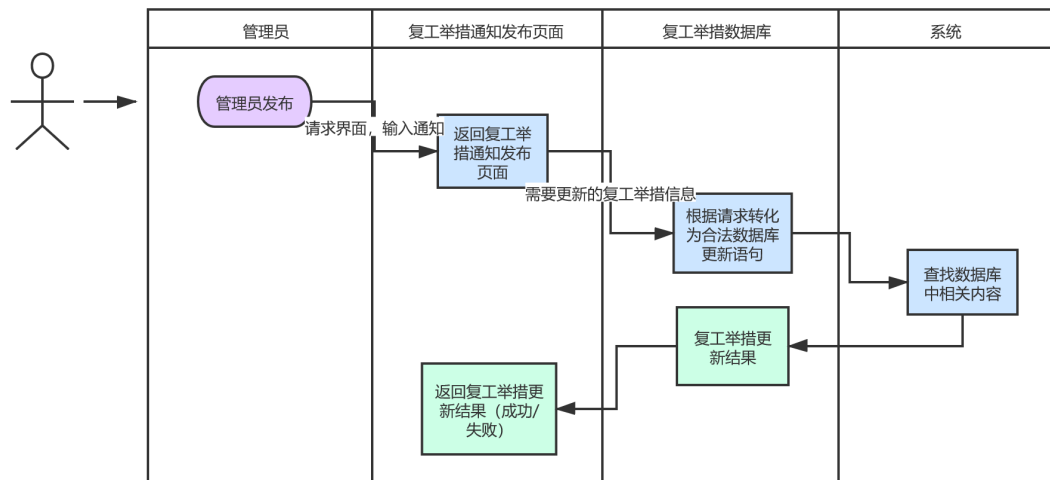
3.6.2. 复工举措查询



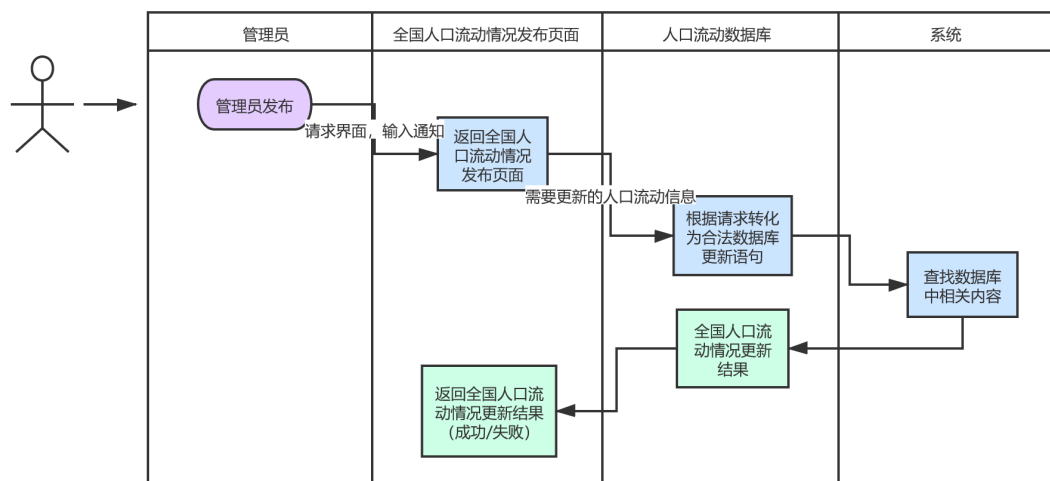
3.6.3. 全国人口流动图查询



3.6.4. 复工通知发布



3.6.5. 全国人口流动情况发布



4. 执行概念

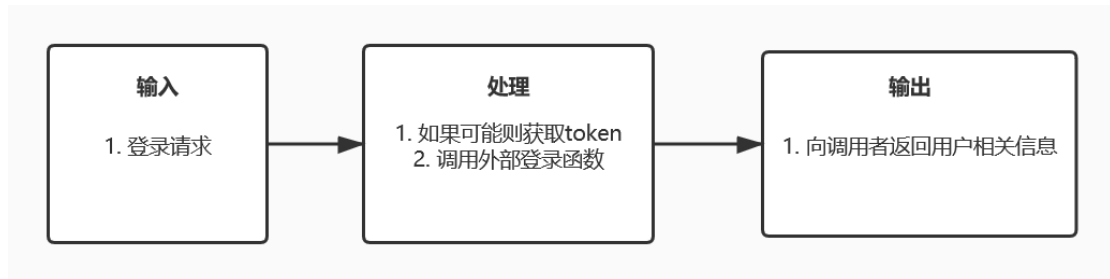
4.1. 登录状态查询

4.1.1. 模块概述

本模块通过调用用户管理子系统的登录方法查询系统的登陆状态。由于服务器是无状态设计的，因此每次进行服务器请求都需要调用该模块。本模块主要采用 token 的方式进行验

证，token 作为唯一的通行凭证进行授权。

4.1.2. IPO 图



4.1.3. 功能

用户电脑的 cookie 上存储 token，通过将 token 发送至用户管理子系统的登录功能进行登录验证，若用户没有 token 则跳转执行新的登录。获取到 token 后该模块向调用该模块的对象返回用户的信息，包括用户名、用户权限、包含扩展信息的 JSON 三项基本信息。

4.1.4. 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|-------|-------|--------|------|
| token | token | string | 函数传参 |

4.1.5. 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------------|----------|----------------------------------|-----------|
| 登录结果 | success | bool | 返回值对象中的属性 |
| 用户名 | userName | string | 返回值对象中的属性 |
| 用户权限 | type | int (0: 普通用户, 1: 省管理员, 2: 超级管理员) | 返回值对象中的属性 |
| 扩展信息的 JSON | extend | string (JSON) | 返回值对象中的属性 |

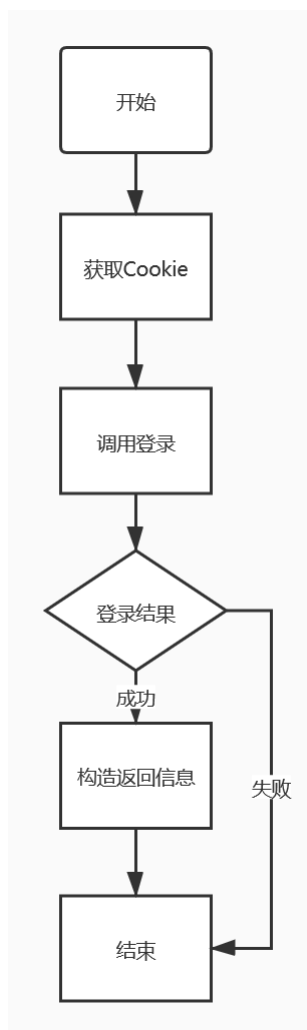
4.1.6. 设计方法（算法）

```

1. token = cookie["token"]
2. user = UserModel.Login(token)
3. if user is null:
4.     successful = false
  
```

```
5. else:
6.     successful = true
7. return {
8.     succ: successful,
9.     userName: user.name,
10.    type: user.type,
11.    extend: user
12. }
```

4.1.7. 流程图



4.1.8. 测试计划

| 输入数据 | 预期结果 |
|-----------|----------|
| 正确的 token | 返回对应用户信息 |
| 错误的 token | 登录失败 |

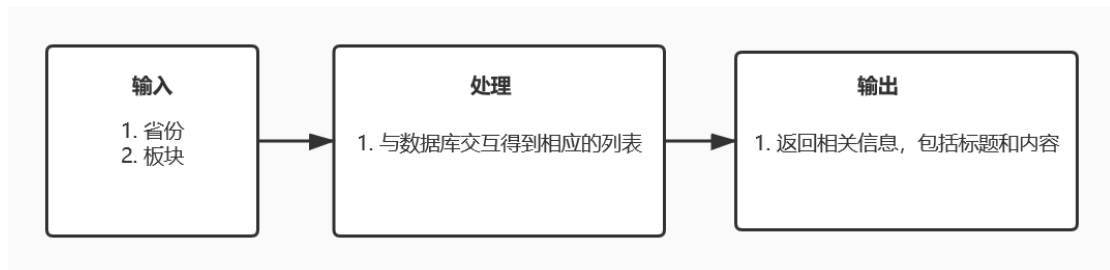
| | |
|---------|------|
| 无 token | 登录失败 |
|---------|------|

4.2. 复工相关举措查询

4.2.1. 模块概述

本模块通过数据库交互显示各省当前发布的复工相关举措，分为复工复产举措、社区管控举措、交通出行举措、医疗服务举措四个板块。

4.2.2. IPO 图



4.2.3. 功能

用户选定省份和板块后前端处理模块将请求发往后端，后端通过查询数据库得到相关信息，并返回包含标题、内容、省份、板块、发布日期时间的信息。后端部分直接按照接口格式返回信息，前端部分通过信息流的展示模式进行展示。

4.2.4. 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|----|----------|--------|---------|
| 省份 | province | string | HTTP 请求 |
| 板块 | category | string | HTTP 请求 |

4.2.5. 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|----------|---------------|----------|
| 新闻列表 | / | string (JSON) | 返回值对象 |
| 标题 | title | string | 列表中对象的属性 |
| 内容 | content | string | 列表中对象的属性 |
| 板块 | category | string | 列表中对象的属性 |

| | | | |
|------|----------|-------------------|----------|
| 省份 | province | string | 列表中对象的属性 |
| 发布时间 | time | string (标准日期时间格式) | 列表中对象的属性 |
| 扩展信息 | extend | string (JSON) | 列表中对象的属性 |

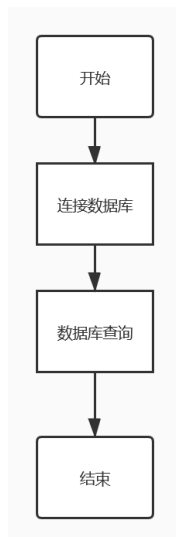
4.2.6. 设计方法（算法）

```

1. database.connect();
2. list = database.query(province, category);
3. return list;

```

4.2.7. 流程图



4.2.8. 测试计划

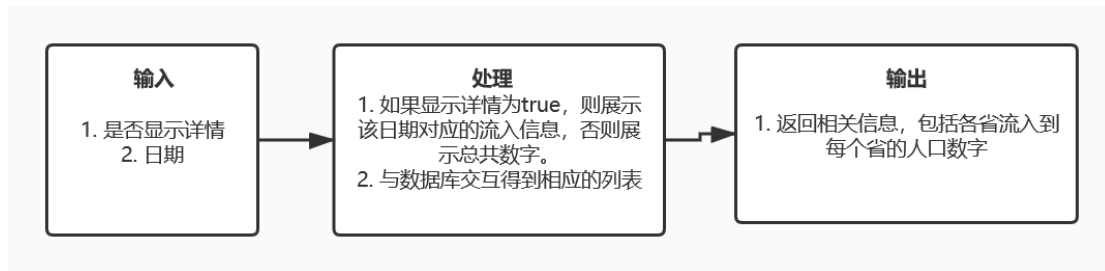
| 输入数据 | 预期结果 |
|----------|-----------|
| 正确的省份及板块 | 返回正确的信息列表 |
| 错误的省份或板块 | 返回空列表 |

4.3. 人口流动图查询

4.3.1. 模块概述

本模块的前端部分通过一个地图展示流入到一个省份的所有省份人口数据, 为了加快展示速度, 这里采用预先返回的方式, 也就是直接返回所有数据。

4.3.2. IPO 图



4.3.3. 功能

用户可以指定一个日期，如果指定了，那么展示当天的人口流动数据，包括每个省从各种地点流入的人口；如果没有指定，那么返回截至当天的总共数据。

4.3.4. 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|----------|--------|-----------------|---------|
| 是否展示当天数据 | detail | bool | HTTP 请求 |
| 指定日期 | date | string (标准日期格式) | HTTP 请求 |

4.3.5. 输出项

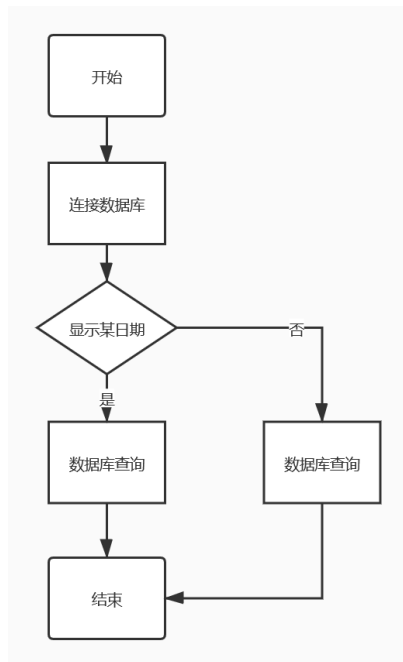
| 名称 | 标识 | 类型和格式 | 输出方式 |
|----------|--------|----------------------|-----------|
| 流入人口对象 | / | string (JSON object) | 返回值对象 |
| 流入目标省份 | (各省名字) | string (JSON object) | 上一项中对象的属性 |
| 流出地点流出人口 | (各省名字) | int (JSON object) | 上一项中对象的属性 |

4.3.6. 设计方法（算法）

```

1. database.connect();
2. if (detail):
3.     return database.query(all)
4. else:
5.     return database.query(date)
  
```


4.3.7. 流程图



4.3.8. 测试计划

| 输入数据 | 预期结果 |
|------------|---------------|
| 显示详情，正确的日期 | 返回正确的信息对象 |
| 显示所有，任意的日期 | 返回正确的信息对象 |
| 显示详情，错误的日期 | 空对象 |
| 详情参数错误 | 应返回与显示所有相同的对象 |

4.4. 发布复工相关举措

4.4.1. 模块概述

本模块作为后台模块实现，拥有相关权限的用户可以发布复工相关举措信息。

4.4.2. IPO 图



4.4.3. 功能

具有发布权限的用户通过 token 进行登录, 在验证了发布权限后可以直接把相关信息进行发布。

4.4.4. 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|-------|----------|--------|---------|
| token | token | string | HTTP 请求 |
| 省份 | province | string | HTTP 请求 |
| 板块 | category | string | HTTP 请求 |
| 标题 | title | string | HTTP 请求 |
| 内容 | content | string | HTTP 请求 |

4.4.5. 输出项

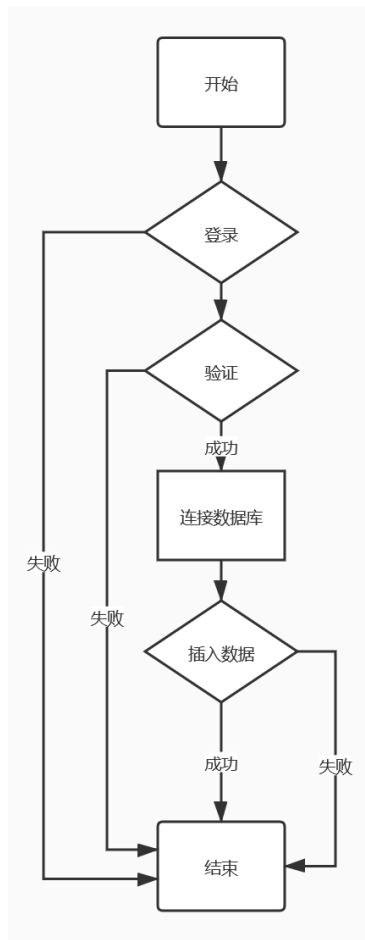
| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|---------|-------|-----------|
| 发布结果 | success | bool | 返回值对象中的属性 |

4.4.6. 设计方法（算法）

```

1. if user = login(token) failed:
2.     return false;
3. if authentication(user) failed:
4.     return false;
5. database.connect();
6. if database.insert(province, category, title, content) failed:
7.     return false;
8. return true
  
```

4.4.7. 流程图



4.4.8. 测试计划

| 输入数据 | 预期结果 |
|-----------------|------|
| 登录失败 | 失败 |
| 登录成功，验证失败 | 失败 |
| 登录和验证成功，插入不规范数据 | 失败 |
| 登录和验证成功，插入规范数据 | 成功 |

4.5. 发布人员流动相关举措

4.5.1. 模块概述

本模块也作为后台模块实现，提供人员流动发布的功能。

4.5.2. IPO 图



4.5.3. 功能

具有发布权限的用户首先通过 token 进行登录，然后可以插入一系列（流入省份，流出省份，人数）的数据列表。也就是一次调用可以插入多个数据，只要有一个数据插入失败那么整个调用即失败，不会插入任何数据。

4.5.4. 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|-------|-------|--------|-----------|
| token | token | string | HTTP 请求 |
| 流出省份 | from | string | 输入列表的一项属性 |
| 流入省份 | to | string | 输入列表的一项属性 |
| 人数 | value | int | 输入列表的一项属性 |

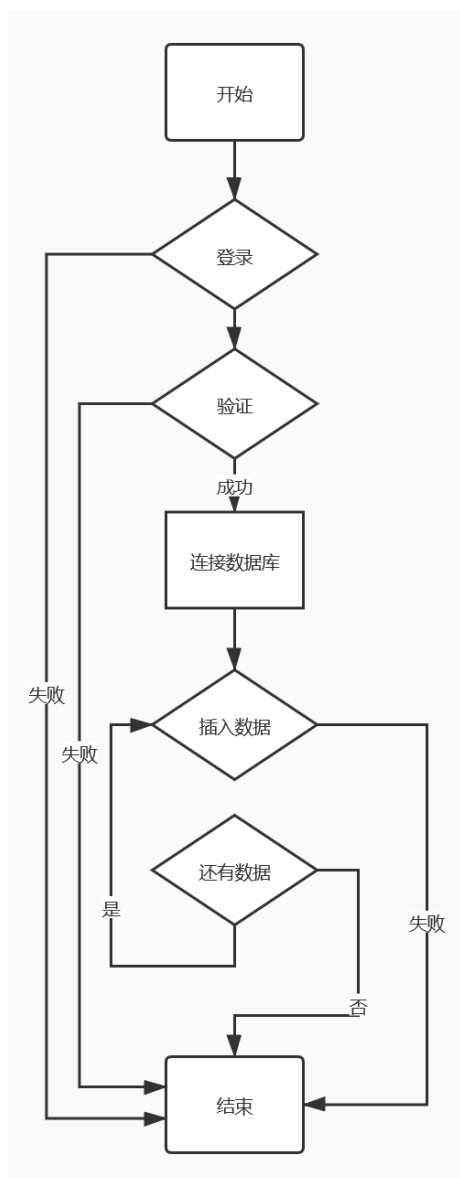
4.5.5. 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|---------|-------|-----------|
| 发布结果 | success | bool | 返回值对象中的属性 |

4.5.6. 设计方法（算法）

```
1.  if user = login(token) failed:
2.      return false;
3.  if authentication(user) failed:
4.      return false;
5.  database.connect();
6.  foreach i in list:
7.      if database.insert(i) failed:
8.          return false;
9.  return true
```

4.5.7. 流程图



4.5.8. 测试计划

| 输入数据 | 预期结果 |
|--------------------|----------|
| 登录失败 | 失败 |
| 登录成功，验证失败 | 失败 |
| 登录和验证成功，插入数据任何一个错误 | 失败，数据库不变 |
| 登录和验证成功，插入规范数据 | 成功 |

5. 接口设计

5.1. 界面设计

5.1.1. 复工举措查询界面

复工及人口流动子系统

湖北 全国其他地区 全国人口流动 [log in](#)

导航栏，分别能跳转列湖北复工情况、全国复工情况、全国人口流动图查询

全国各地复工复学情况一览

数据来源：综合各官方网站及媒体报道整理 [Learn more](#)

湖北省内四类复工举措查询

湖北省内复工复学情况一览

复工复产 社区管控 交通出行 医疗服务

按照城市展开查询

| 城市 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
|----|-----------|----------|---------|-------|
| 武汉 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 孝感 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 黄冈 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 荆州 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 鄂州 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 随州 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 襄阳 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |

全国其他地区四类复工举措查询

全国其他地区复工复学情况一览

复工复产 社区管控 交通出行 医疗服务

按照省份展开查询

| 省份 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
|----|-----------|----------|---------|-------|
| 北京 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 上海 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 广东 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 浙江 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 江苏 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 天津 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |
| 湖南 | 有序开放23家景区 | 高校社区严格封控 | 网约车恢复运营 | ***** |

5.1.2. 人口流动图查询界面



5.1.3. 管理员发布复工举措通知界面

复工及人口流动子系统

个人中心 退出登录

发布复工相关举措

标题 这里写标题ABC123

内容 Duis porttitor vulputate arcu, at hendrerit eros cursus accumsan. Donec a dui vitae velit feugiat vulputate. Aliquam erat volutpat. In quis leo nec urna iaculis luctus. Mauris ut lorem at odio volutpat maximus ut nec erat. Donec vulputate urna eu magna venenatis, a fermentum nisi lacinia. Nam tempor luctus nisl. Nulla nunc metus, imperdiet id luctus at, feugiat ac nibh. Donec non tempus ipsum. Nunc enim nulla, placerat a ornare vitae, pretium convallis tellus. Proin posuere aliquet sem ac laoreet. Suspendisse non turpis pellentesque, auctor turpis et, laoreet nisi. Nullam euismod ligula urna, eu porttitor odio

省份 北京

板块 复工复产

发布

导航栏: 具有可以跳转到其他各模块的菜单, 以及用户信息的折叠菜单

内容输入区域: 用户在该区域可以发布相关信息

5.2.4. 连接数据库接口

使用 node.js 框架的 mysql 模块与数据库进行连接，一个简单的例子如下：

```

1. var mysql          = require('mysql');
2. var connection     = mysql.createConnection(
3.   {
4.     host      : 'localhost',
5.     user      : 'dbuser',
6.     password: '*****'
7.   }
8. );
9.
10. connection.connect();
11.
12. connection.query('SELECT 1 + 1 AS solution', function(err, rows, fields)
13. {
14.   if (err) throw err;
15.   console.log('The solution is: ', rows[0].solution);
16. });
17.
18. connection.end();

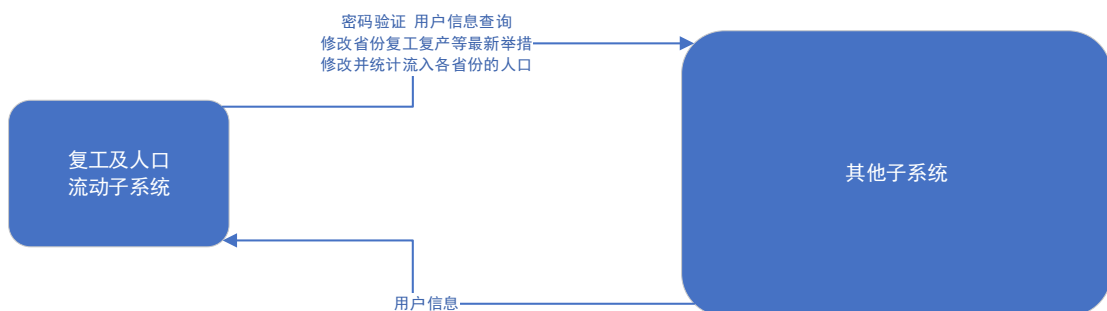
```

5.3. 内部接口

本系统的数据库与其他几个子系统的数据库是相互独立的，本系统需要其他子系统给出的接口，并为其他子系统提供接口。系统间的数据交互表现为数据库中的数据传输，并通过Session中保存的状态完成模块间的通信。

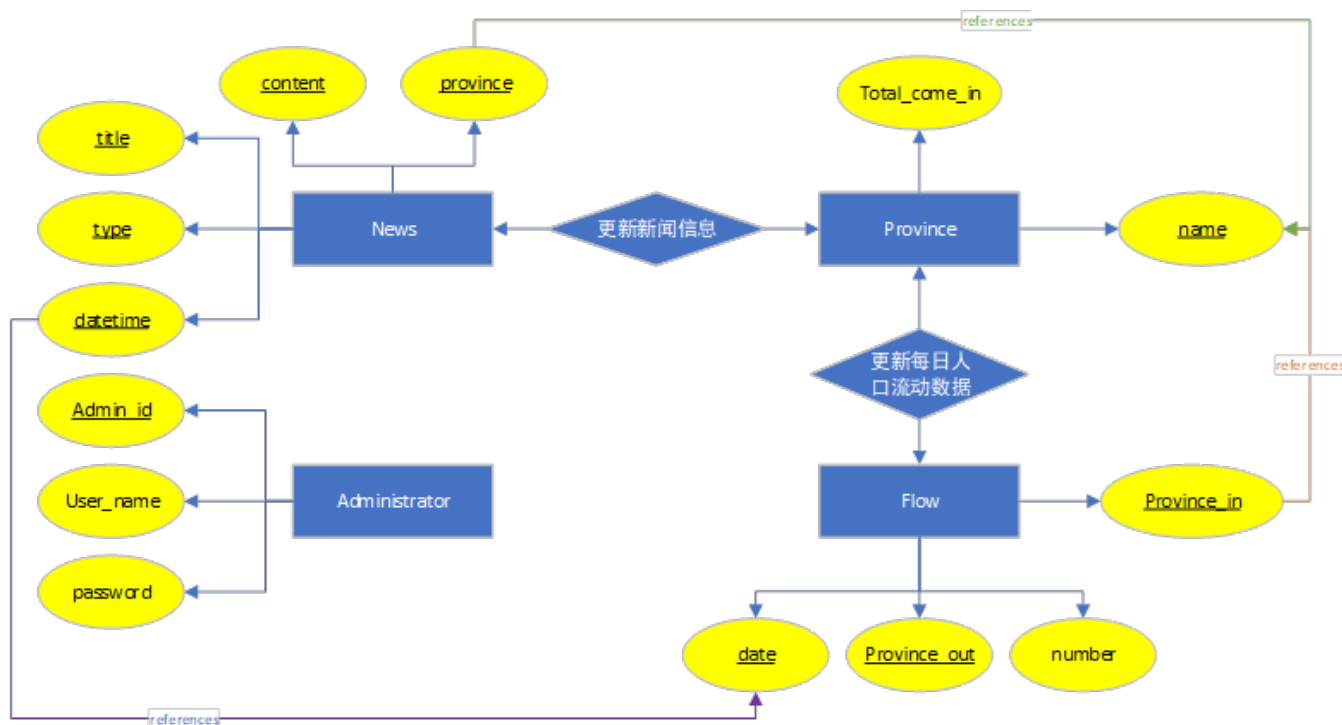
本系统给出的接口是密码验证、用户信息查询、修改省份复工复产等最新举措、修改并统计流入各省份的人口。

本系统需要的外部接口是用户信息。



6. 数据库设计

6.1. ER 图



6.2. 逻辑结构设计

1. 最新新闻/复工复产举措
News (datetime, type, Province, title, content)
2. 省份信息
Province (name, total_come_in)
3. 人口流动记录
Flow (date, province_in, province_out, number)

6.3. 物理结构设计

1. 最新新闻/复工复产举措

| Attributes | Type | Is NULL? | Primary key? | Foreign key? | Remark |
|------------|-------------|----------|--------------|--------------|--------|
| datetime | datetime | N | Y | N | 日期 |
| Province | Varchar(20) | N | Y | 参照表 Province | 身份名称 |

| | | | | | |
|---------|-----------------|---|---|---|------|
| type | Varchar(15) | N | Y | N | 新闻类型 |
| Title | Varchar(100) | N | Y | N | 新闻标题 |
| content | Varchar(500000) | Y | Y | N | 内容 |

2. 省份信息

| Attributes | Type | Is NULL? | Primary key? | Foreign key? | Remark |
|---------------|---------------|----------|--------------|--------------|--------|
| Name | Varchar(20) | N | Y | N | 省份名称 |
| Total_come_in | numeric(5, 0) | N | N | N | 总流入人数 |

3. 人口流动记录

| Attributes | Type | Is NULL? | Primary key? | Foreign key? | Remark |
|--------------|---------------|----------|--------------|-----------------|----------|
| Date | date | N | Y | N | 日期 |
| Province_in | Varchar(20) | N | Y | 参照表 Province | 人口流入省份名称 |
| Province_out | Varchar(20) | N | Y | N | 人口流出省份名称 |
| number | Numeric(5, 0) | N | N | N | 流动人口数 |

7. 运行设计

7.1. 运行模块组合

本子系统按照功能划分模块，每个模块又按流程划分为客户端界面，客户端脚本，服务器后台程序。功能模块之间会共享部分界面（导航栏等）。

7.2. 运行控制

用户通过处于主界面的导航栏选择功能：用户登录模块（将在后续调试中使用第一组的模块）：用户在表单中输入正确的账户密码，从而登录系统。根据“用户”和“管理员”两种不同的身份，将拥有不同的模块功能。

➤ 若登录者为用户

查询复工复产、社区管控、交通出行、医疗服务等举措

查询复工复产、社区管控、交通出行、医疗服务等举措，详细信息包括省份名称、举措标题、举措简介、举措详细介绍以及附带的物资领取等专属链接等。

查询全国人口流动图

查询人口流动图，详细信息包括省份名称、其他省份流入该省份人口数据。

➤ 若登录者为管理员

发布复工复产、社区管控、交通出行、医疗服务等举措

发布复工复产、社区管控、交通出行、医疗服务等举措，详细信息包括省份名称、举措标题、举措简介、举措详细介绍以及附带的物资领取等专属链接等。

发布全国人口流动情况

发布人口流动图，详细信息包括省份名称、其他省份流入该省份人口数据。

7.3. 出错信息

| 系统输出信息的形式 | 含义 | 处理方法 |
|-----------|--------------------------------|--------------------------------------|
| 数据库无法连接 | ①数据库配置出错 ②数据库连接数超过上限 | ①修改数据库配置 ②限制并发访问量 |
| 用户名或密码错误 | ①用户忘记密码 ②被他人恶意篡改密码 | 通过登录界面的“找回密码”进行密码修改 |
| 服务器暂时无法访问 | ①服务器正在维护中 ②短时间内有大量流量导致服务器瘫痪 | 对于②的情况，联系系统管理员进行紧急处理 |
| 无法读取磁盘内容 | 磁盘受损 | 对磁盘和数据库进行周期性的备份 |
| 非法访问 | 部分用户企图访问管理员界面或后台程序，窃取网站 | 限制普通用户越权访问，通过各种手段保护后台数据 |
| 数据库执行出错 | 部分用户企图恶意实施 SQL 注入，导致正常查询出错 | 使用参数绑定的方法进行 SQL 语言构建。对用户填入的信息进行字符串过滤 |

7.4. 补救措施

7.4.1 系统恢复

系统崩溃后，根据系统运行日志恢复系统和数据，并重新启动系统。

7.4.2 定时备份

- (1) 周期性地备份数据库中的数据，将其存储在更稳定的介质上，并及时进行校对、更新，以避免数据损失之后难以找回。
- (2) 将工程代码通过 GitHub 进行完善的版本管理。

7.4.3 人工操作

当出现紧急情况时，数据库管理员人工地对数据库中的数据进行修改，并做相应的记录。

7.5. 系统维护设计

- (1) 用户在该系统执行操作时应该留下痕迹，以方便检查系统是否被恶意篡改。同时系统管理员定时查看系统日志，统计非法攻击来源和次数，并针对相应攻击加强安全措施。
- (2) 系统管理员登录不仅需要账户密码，还需要识别 IP，禁止非白名单 IP 任何访问。
- (3) 系统维护人员及时更新技术漏洞，通过各种手段防止各种对系统的攻击，增强代码的可靠性。
- (4) 定期维护数据库，涉及到检查数据库表、日志文件等，确保数据的正确性。
- (5) 在可能出错的地方使用 try-catch 语句捕获异常，并输出相应的出错信息和可能的处理方法提示。

8. 需求回溯

8.1. 功能性需求回溯

| 需求 ID | 需求简述 | 对应的设计模块 | 实现方式 |
|-------|-----------|---------|--|
| A01 | 用户身份有效性核验 | 身份验证模块 | 页面加载时，服务器验证 Session 是否过期，如果过期则重定向到所有子系统统一的登录界面。否则用户将根据 Session 中存储的身份信息，拥有不同的模块功能。 |

| | | | |
|-----|-----------------------------|--------------|--|
| A02 | 为用户提供能够显示全国各省复工复产等举措的界面 | 复工复产等举措查询模块 | 服务端验证 Session 后确认用户身份后，调用函数显示用户所需界面。 |
| A03 | 为用户提供能够显示全国人口流动图的界面 | 全国人口流动图查看模块 | 服务端验证 Session 后确认用户身份后，调用函数显示用户所需界面。 |
| A04 | 为管理员提供能够填写并发布全国各省复工复产等举措的界面 | 复工复产等举措发布模块 | 服务端验证 Session 后确认管理员身份后，调用函数显示管理员所需界面。 |
| A05 | 为管理员提供能够填写并发布全国各省人口流动情况的界面 | 全国人口流动情况发布模块 | 服务端验证 Session 后确认管理员身份后，调用函数显示管理员所需界面。 |

8.2. 性能及安全需求

| 需求 ID | 需求简述 | 实现方式 |
|-------|--------------------------------------|---|
| B01 | 【安全性】用户账户安全 | 使用 OAuth2.0 标准协议 |
| B02 | 【安全性】数据库数据保护，防止用户恶意访问数据库或者破坏数据库 | ①使用占位符进行查询函数绑定 ②数据库根据不同的最小需要的操作权限 |
| B03 | 【性能】能够应对数据库访问量过大的情况以及通过大量访问量来实现攻击的手段 | 对于频繁访问数据库的操作，需要建立索引，使用 redis 缓存来优化 |
| B04 | 【性能】客户端通过网页展现给用户一个友好、易于操作的界面 | 通过 Bootstrap 和外部 CSS 样式表实现扁平化样式与栅格排版，使用 JavaScript 和 JQuery 实现数据的动态显示和交互。 |