DESIGN MODEL

**设计模型**

公共卫生危机应急管理系统

1851632 石稼晟

1751022 李翠琪

1852450 莫子毅

1852140 上官宇飞

1852035 史家豪

1856091 王子睿

##### 系统设计与分析 Object-Oriented Analysis and Design with UML

2020.6.26

目录 index

**[关于文档 3](#_Toc24216)**

[概述 4](#_Toc26894)

**[架构分析 5](#_Toc24864)**

[架构分析 5](#_Toc28091)

[总类图展示 12](#_Toc30695)

**[子系统和接口 13](#_Toc24091)**

[子系统与接口列表 13](#_Toc8008)

[子系统分类与说明 14](#_Toc14998)

[1. 用户管理子系统 14](#_Toc23963)

[2. 信息收集子系统 14](#_Toc8560)

[3. 信息清洗子系统 15](#_Toc27705)

[4. 信息分析子系统 16](#_Toc12851)

[5. 信息可视化子系统 17](#_Toc29107)

[6. 信息查询子系统 17](#_Toc12701)

[7. 信息发布子系统 17](#_Toc7827)

[8. 政府应急管理部门子系统 18](#_Toc6596)

[9. 预警子系统 18](#_Toc18790)

[10. 信息存储子系统 18](#_Toc24717)

[接口规范示例 20](#_Toc19428)

[信息子系统展示 22](#_Toc20137)

**[用例实现 23](#_Toc12903)**

[用例：注册 23](#_Toc21432)

[用例：上传信息 28](#_Toc6164)

**[关键设计机制 31](#_Toc26690)**

[设计机制 31](#_Toc7528)

[一、进程间通讯机制 31](#_Toc21659)

[二、持久性机制 32](#_Toc5121)

**[原型设计 34](#_Toc11265)**

[已实现原型设计： 34](#_Toc4573)

[期望效果： 35](#_Toc1381)

**[反馈 38](#_Toc28518)**

[未解决的问题 38](#_Toc1987)

[自我反思 38](#_Toc3842)

[成员贡献度 38](#_Toc3842)

# 

# 

# 关于文档

##### about the document

## 文档描述

**description**

本文档是基于移动互联网的公共卫生安全应急管理系统之分析模型说明，以文档化形式对项目模型进行分析与构建。本文档包含项目的简要概述、项目架构的分析、项目应用的分析机制与项目的分析模型。对于每个用例的分析模型，包括类图与时序图/交流图。本文档供项目监督者和开发者查阅。

根据前期对项目总体的把握和目标，对项目中对象的多种活动进行进一步分析。对于整个系统，我们进行了纵向分层的分析，将系统划分为用户层，应用层，业务层，管理层和控制层，通过分层分析对整个系统有了进一步的理解并且方便各种活动的实现。

# 概述

**general introduction**

## 项目概述

**general introduction**

目的

本文档的目的是提供公共卫生危机应急管理系统项目详细的初步分析模型，包括类图，时序图交流图，机制分析等等，还有本系统的功能和特点及其操作必须遵守的约束条件。本文档供系统的开发者和利益相关者阅读。

简要介绍

本项目专注于公共卫生安全应急管理，主要具备两个功能。第一个是是日常监测预防并预测突发性公共卫生危机的功能。系统能够随时监测医疗系统及科研机构的数据分析并能够对于疫情的爆发进行预测预警。第二个功能是疫情爆发后辅助政府部门联动其他系统进行决策、疫情防控和管理，为有关部门提供疫情防控相关的参考性建议。

本系统中主要有五种角色类型，分别是：市政府应急部门、医疗及科研机构、系统管理员、其他系统、普通用户。系统主要为政府部门系统提供宏观支持作为目标，通过各级别用户对疫情信息的共享和处理，帮助政府部门收集数据，对政府部门下达决策起到支持和辅助的作用。对于医疗及科研机构用户，可以从普通用户和社区卫生机构获取疫情的数据信息，生成疫情的相关报告；对于市政府应急部门，可以从预警报告中获得指导性意见，通过报告分析来发布相关决策。对于系统管理员，可以进行管理维护，审核并分类用户上报的疫情信息，并对其他管理员发起的管理请求进行互审，同时，系统管理员这个角色担任了公示信息的职责与任务。对于其他系统诸如地理信息系统，人力与物资管理系统，交通运输系统，商务系统等等其他辅助性系统，通过系统接受政府应急响应部门的任务。对于普通用户，可供用户上传自身信息，浏览系统公示信息及疫情实时数据信息。五类用户都可以进行注册、登录、上传并修改信息、查讯疫情的公开实时数据等用例。

本系统主要为政府部门系统提供宏观支持作为目标，通过各级别用户对疫情信息的共享和处理，对政府部门下达决策起到支持和辅助的作用，支持决策的全过程。项目基于已有的政府信息系统，从疫情数据的采集、汇集、存储和计算，最终为决策提供依据；通过将决策高效的分解、存储、下达、执行和跟踪反馈，来确保政府应对公共卫生危机决策的科学性和时效性。

平台与框架

该项目在Web平台，安卓app平台以及微信小程序平台上进行开发。并且前端，后端与数据库都部署于云服务器linux系统上。

前端采用vue框架开发搭建单页面应用。后端采用Spring框架为整个应用提供控制反转（IOC）和面向切面（AOP）功能。后端在视图层采用SpringMVC框架，以此完成对用户请求的转发和对用户的响应。采用Mybatis框架完成对数据库的增删改查等功能。

架构分析

**architectural analysis**

架构分析

**architectural analysis**

该系统分为十个主要的子系统（用户管理子系统, 信息收集子系统，信息清洗子系统，信息分析子系统，信息可视化子系统，信息查询子系统，信息发布子系统，政府应急管理部门子系统，预警子系统，信息存储子系统）以及其他辅助的小系统(如信息加密系统,可视化系统等),并且可与外部系统通信,如地理信息系统等。系统采用前后端分离技 术，并且通过API接口互相通信。前端为面向用户的移动应用程序以及Web网页界面，后端则部署于服务器上，由管理员运营维护。

该系统采用MVC架构模式（模型Model-视图View-控制器Controller）以降低耦合性，提高代码的可重用性。并且从体系结构的层次结构来看，该系统具有典型的五层结构(自上而下分别是表现层，业务逻辑层，数据访问层，数据库层，基础平台层，以及整个系统共用的公共工具包)：

* 表现层：包含了Web平台，安卓app平台以及微信小程序平台。以及和其他层交互的API（Application Programming Interface）
* 业务逻辑层：包含八个主要的子系统（登录子系统，注册子系统, 信息清洗子系统 信息处理子系统 预警子系统 信息收集子系统,任务子系统,报告子系统）以及地理信息子系统。
* 数据访问层: 包含对象关系映射（Object Relational Mapping）以及数据存取对象（Data Access Object）。其中ORM是通过使用描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系数据库中。而DAO位于业务逻辑和持久化数据之间，实现对持久化数据的访问。
* 数据库层:包含了人流分布地理信息数据库,患病人员数据库,政府信息数据库。
* 基础平台层: 包含了操作系统,数据库管理系统,GIS系统。
* 公共工具包: 包含了权限控制子系统,类型转换子系统,日志子系统,全文搜索子系统。

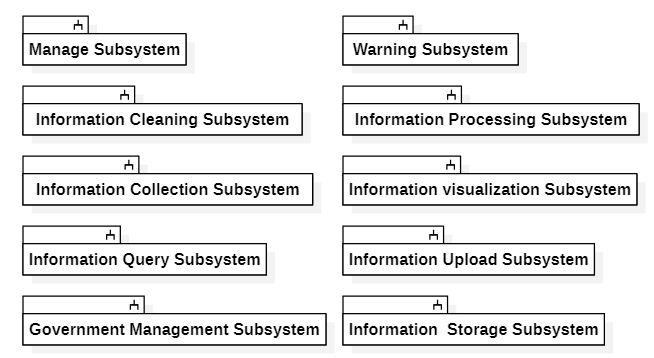
各层之间依赖关系与数据传递：表现层与业务逻辑层之间通过数据传输对象（DTO）进行交互的，它对业务数据进行数据封装。而在数据访问层与数据库层之间，采用数据访问接口DAO进行数据访问。并且表现层，业务逻辑层，数据访问层都需要依赖于公共工具包。

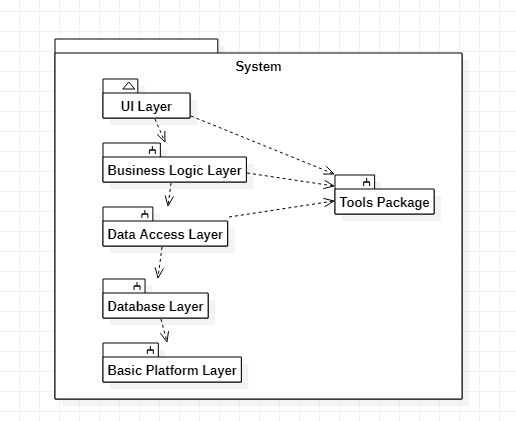
部署图：用户端可以是浏览器，安卓app，微信小程序等。为了缓解单台服务器压力，设置服务器集群。并且在用户端和Web服务器集群设置Nginx[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488" \t "_blank)服务器，以提供负载均衡服务。而为了缓解访问单一数据库速度瓶颈问题，设置了数据库集群。并且在服务器集群和数据库集群之间通过数据库中间件进行访问。

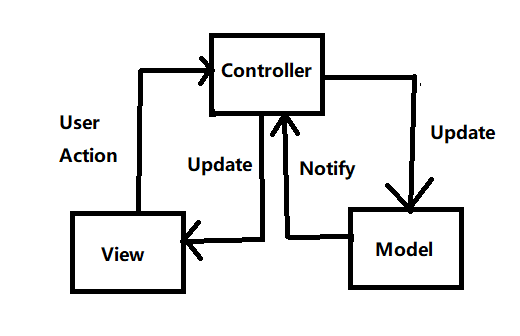
项目实施及使用技术初步方案：

* 前端浏览器端采用vue框架开发，用vue-cli脚手架搭建应用，vue-router实现路由配置，使用Element UI 控制样式。并且使用HTTP库axios与后端异步的交互json格式的数据。
* 后端使用Spring Boot框架实现快速搭建，用maven进行版本控制，项目构建，依赖管理等。并且在Web层整合SpringMVC框架，在持久层ORM使用Mybatis框架并且通过Druid连接池实现对数据库的访问。后端的服务器使用tomcat Web 应用服务器。数据库采用Mysql数据库，并且使用非关系型数据库Redis实现缓存功能。
* 上述架构中公共工具包的实现方案：使用SpringSecurity实现权限控制，搜索引擎框架ElasticSearch实现全文检索。使用Logback实现日志系统。
* 部署，前端，后端，数据库都分别部署于阿里云服务器上，操作系统环境为CentOS 7，64位linux系统。采用Docker容器管理linux服务器上的应用。并且使用Nginx服务器实现负载均衡以及反向代理功能。

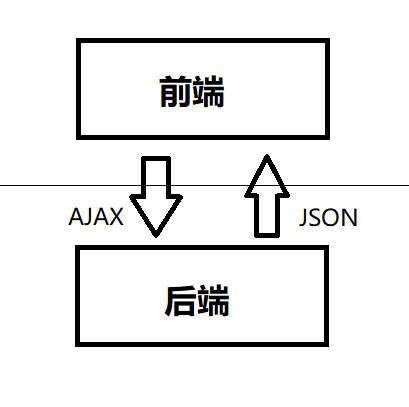
十个子系统：

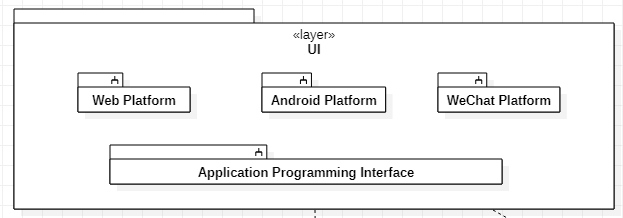
系统五层机构：

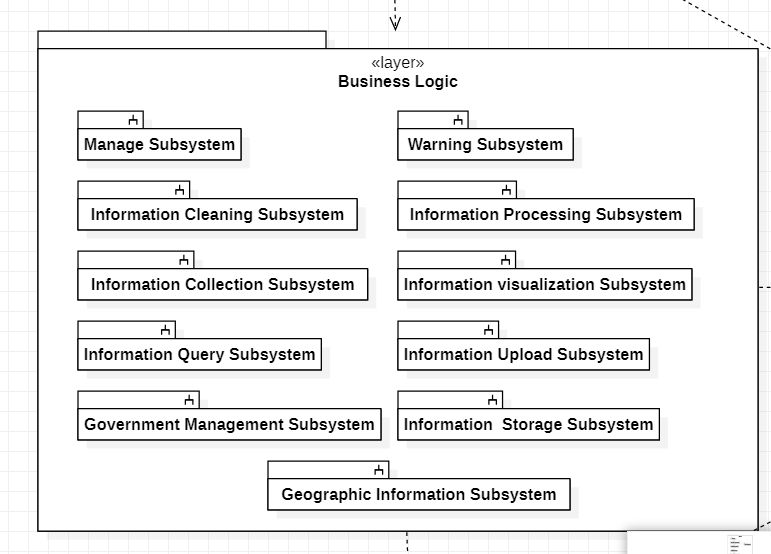
MVC架构模型：



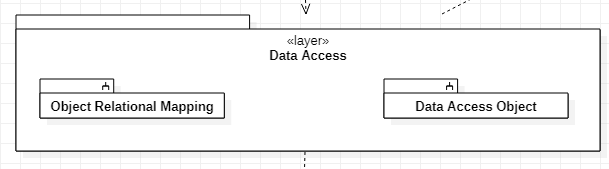
前后端分离：

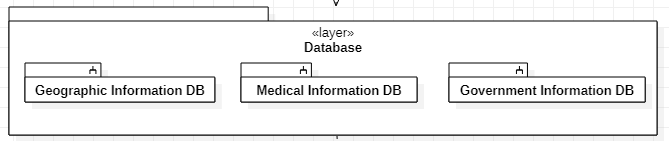
表现层：

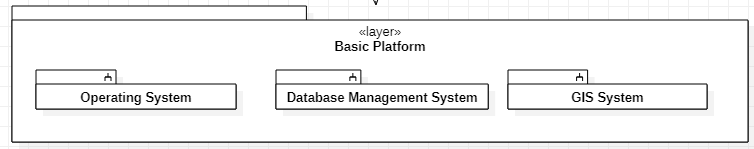
业务逻辑层：

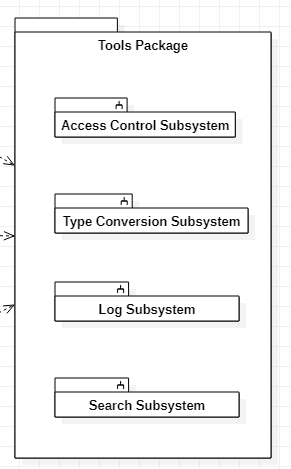


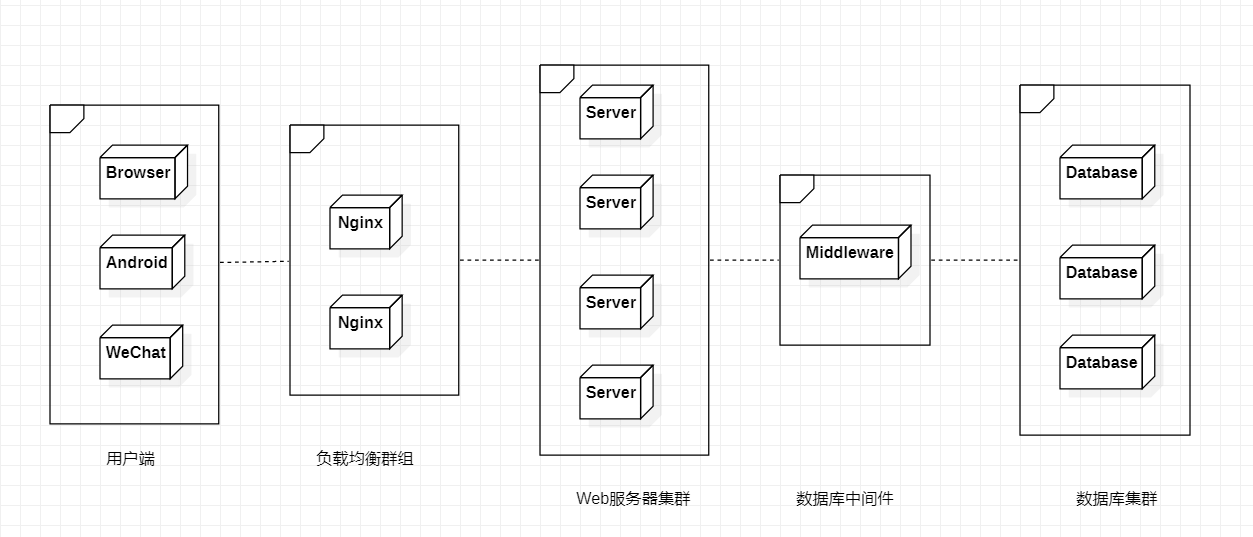
数据访问层：

****数据库层：

****基础设施层：

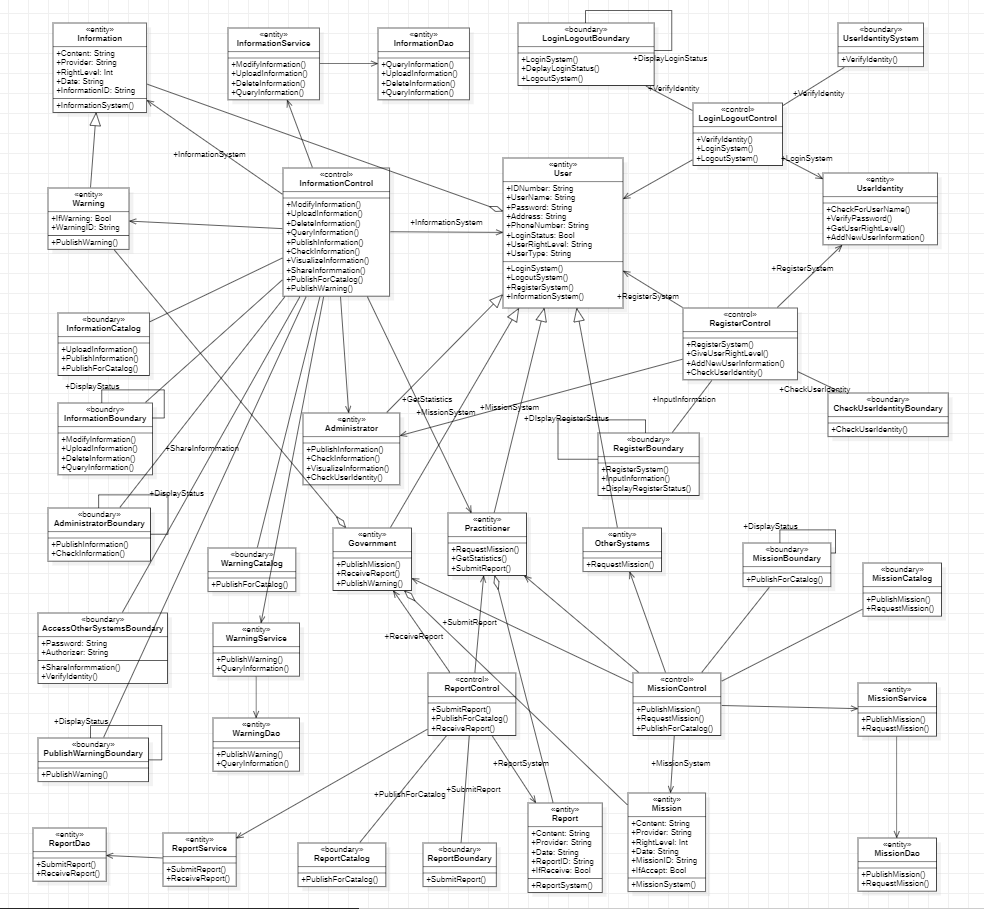
公共工具包：

部署图：



总类图展示

**Class diagram display**



子系统和接口

**Subsystems and interfaces**

子系统与接口列表

**list of subsystem and interfaces**

|  |  |
| --- | --- |
| 用户管理子系统 | IUserSignIn |
| IUserSignUp |
| IUserFindBackPassword |
| 信息收集子系统 | IUploadInformation |
| IAlterInformation |
| IMonitorEnvironment |
| IStoreInformation |
| 信息清洗子系统 | IRequestRawInformation |
| IStoreCleanInformation |
| 信息分析子系统 | IRequestCleanInformation |
| ILinkModelLibary |
| IStoreIndexInformation |
| 信息可视化子系统 | IRequestGeographicData |
| IQueryMathSubsystem |
| IStoreVisuaInformation |
| 信息查询子系统 | IQueryInformatoin |
| 信息发布子系统 | IPublishInformation |
| 政府应急管理部门子系统 | ICheckAssignment |
| IPublishAssignment |
| 预警子系统 | ISendWarningMessage |
| 信息存储子系统 | IStoreInformation |
| ISendInformation |

子系统分类与说明

1. 用户管理子系统
   1. 功能：本系统用来管理所有的用户的信息，用户可以实现注册和登录，以及当忘记密码之后的找回
   2. 接口：
      1. IUserSignIn（使用者：用户 功能：用户登录系统）

接口函数：

SetUserInformation(String Username,String Password):bool

Signin():bool

* + 1. IUserSignUp（使用者：用户 功能：用户注册系统）

接口函数：

SetUserSignUpInformation(String Username , String Password):bool

SignUp():bool

* + 1. IUserFindBackPassword （使用者：用户和管理员 功能；找回密码）

接口函数：

SetUserAuthenticationInformation(String Username , String Authentication):bool

GetUserPassword():String

1. 信息收集子系统
   1. 功能：收集与本次疫情相关的所有信息，有三种主要的来源，使用本系统的用户上传的信息，获取医院的患病的人数的数据，获取药店的药物的出售的信息。
   2. 接口：
      1. IUploadInformation（使用者：所有注册的用户 功能：用户可以使用通过接口向系统中上传他们的信息 ）

接口函数：

SetUploadInformation(User userInfo , Info information):bool

UploadInformation():bool

* + 1. IAlterInformation（使用者：所有注册的用户 功能：用户可以修改自己上传过的信息）

接口函数：

SetAlteredInformation(User userInfo , Infor information , Info oldInfo,Enum alertType):bool

AlertInformation():bool

* + 1. IMonitorEnvironment（功能：监测医院的生病人数和生病种类，监测药店相关药品的出售情况）

接口函数:

MonitorHospital():JSON

SetHospitalInformation(Address addr,Time time,Info information):Info

SendHospitalInformation():bool

* + 1. IStoreInformation（功能：连接存储信息子系统将所有收集到的信息进行存储）

ConnectDatabase(Stirng user, String password): bool

SetInformation(Info information , String remark): bool

SendInformation():void

1. 信息清洗子系统
   1. 功能：由于收集到的所有的信息十分繁杂，来源多种。彼此之间的信息有重复也有错误，因此需要使用本子系统来对收集到的所有数据进行清洗，来获得可以分析处理的数据子系统中主要是根据多种算法和标准来实现了对数据的清洗
   2. 接口：
      1. IRequestRawInformation（使用者：信息存储子系统 连接信息存储子系统，获得要清洗的信息）

接口函数：

RequestRawInformation(Administrator admi ,Type requestType):String

SetRawInformation(Info[ ] information):bool

* + 1. IStoreCleanInformation（使用者：信息存储子系统 连接信息存储子系统，将清洗过的信息进行存储）

SendCleanInformation():CleanInfo

1. 信息分析子系统
   1. 功能：本系统的主要的针对目标是为了对公共卫生危机进行很好的解决，因此需要对所有收集到的信息进行分析和处理，来得到对于疫情的信息的关键指数。由于这样一个分析处理需要长时间不断地进行，且数据量巨大，因此需要设置单独地子系统来处理维护。
   2. 接口:
      1. IRequestCleanInformation  
         RequestCleamInformation(Administrator admi ,Type requestType):String  
         UpdateCleanInformation(Info[ ] information):bool
      2. ILinkModelLibary  
         QueryModel (String name):Model  
         ApplyModel(Model model):void
      3. IStoreIndexInformation

ConnectDatabase(Stirng user, String password): bool  
SendIndex(Index index):bool

1. 信息可视化子系统
   1. 功能：完成对信息的可视化，连接使用其他系统（如地理信息系统等）来辅助进行可视化
   2. 接口
      1. IRequestGeographicData（使用者：地理信息子系统 功能：获取地理信息子系统的相关信息，来完成信息的可视化）

RequestGeographicData(UserInfo user , String name):JSON

* + 1. IQueryMathSubsystem（使用者：数学模型子系统 功能：连接数学模型子系统，获得数学模型，辅助完成对信息的图表的可视化信息的创建）

QueryMathTools(UserInfo user , String name):JSON

* + 1. IStoreVisuaInformation（使用者：信息存储子系统 连接信息存储子系统，将可视化的信息进行存储）

ConnectDatabase(Stirng user, String password): bool

SendVisualInformation(VisualInfo):bool

1. 信息查询子系统
   1. 功能：使得所有用户可以查询有关疫情的所有信息
   2. 接口
      1. IQueryInformatoin（使用者：所有用户 功能：普通用户可以通过网站或应用程序来访问本系统并查询到需要的信息）

SetUserInfo(User user , Authorization au):bool

SendInformation():Info

1. 信息发布子系统
   1. 功能：本系统的主要功能是负责信息的发布，即将所有经过清洗之后的信息、所有经过可视化之后的信息发布在我们的所有平台上，并给用户发送已经送达的信息提醒用户查看信息。
   2. 接口：
      1. IPublishInformation(功能：发布清理之后和可视化之后的信息)  
         接口函数：

PublishInformation(Info[ ] infolisi):bool

1. 政府应急管理部门子系统
   1. 功能：提供所有辅助政府应急部门的功能，如政府部门之间的发布、接收任务。辅助指挥等等。
   2. 接口
      1. ICheckAssignment  
         接口函数：

GetAssignment(User userInfo):Assignment

* + 1. IPublishAssignment

接口函数：

SendAssignment(Assignement assignment , String publisher):bool

1. 预警子系统
   1. 功能：实时检测当前数据，当超过阈值时就进行预警
   2. 接口：
      1. ISendWarningMessage

SendWarningMessage(Warning warn):void

* + 1. IUpdateRealTimeIndex

RequestIndex(Address a ,Time t):bool

UpdateIndex(Index index):bool

1. 信息存储子系统
   1. 持久化存储所有的信息，使用Oracle等数据库，然后提供存储的接口
   2. 接口：
      1. IStoreInformation（使用者：所有产生数据的子系统 功能：对其产生的所有数据进行存储）

SetUser(User userInfo):bool

UpdateInformation(Info information):bool

StoreInformation(Info information):bool

* + 1. ISendInformation（使用者：所有需要信息的子系统 功能：子系统描述需要的信息，然后数据库从中进行选择并返回信息）

UpdateUserInfo(User user):bool

SendInformation():Info

接口规范示例

服务名称：

请求地理信息接口

HTTP访问方式：

POST

功能描述：

获取地理信息子系统中储存的地理信息，

请求URL：

<http://xxxx.com/api/geograph>

适用对象：

系统管理员使用，只有当确定身份之后才可以使用

调用参数：

系统参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| username | String | Y | 系统验证的用户名 |
| password | String | Y | 系统验证的密码 |

应用级参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| province | String | Y | 指出获得哪一个省的信息 |
| city | String | N | 详细指出哪一个城市 |

返回字段：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| id | int | 表示省份或者城市的编号 |
| Parentid | Int | 若是城市，则表示其省份的id |
| Name | String | 表示省份或者城市的姓名 |
| Level | Int | 用来表示是省份还是城市 |
| Lng | Double | 表示经度 |
| Lat | Double | 表示纬度 |

返回示例：

{

    "error\_code":0,

    "data":[

        {

            "id":"1",

            "parentid":"0",

            "name":"北京市",

            "level":"1",

            "lng":"114.23300",

            "lat":"29.579858"

        },

        {

            "id":"1234",

            "parentid":"13",

            "name":"郑州市",

            "level":"2",

            "lng":"114.23300",

            "lat":"29.579858"

        }

        .........

    ]

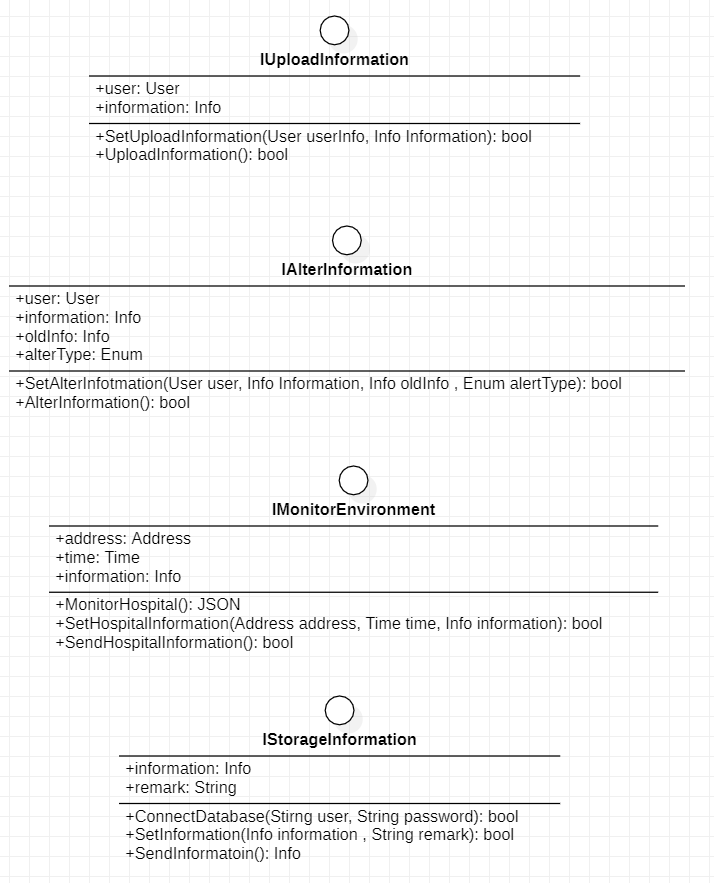
}

返回参数备注：

Error\_code表示请求是否成功，若是为0则返回成功，其余情况分别对应不同的错误。

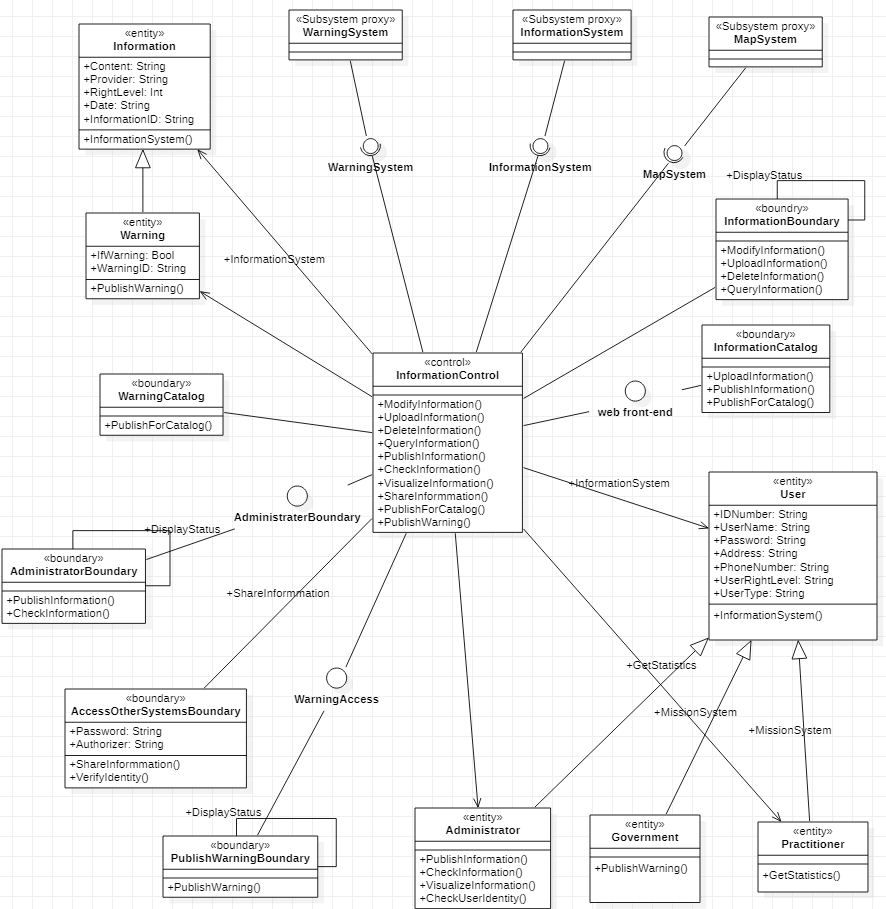
信息收集子系统展示

**Information collection subsystem display**



信息总子系统类图展示

**Information subsystem class diagram display**



用例实现

**use case model**

用例：注册

用例注册过程：

1、用户向系统发送注册请求，进入录入信息页面并录入基本信息，系统请求系统管理员检查用户类别，根据验证信息返回值。

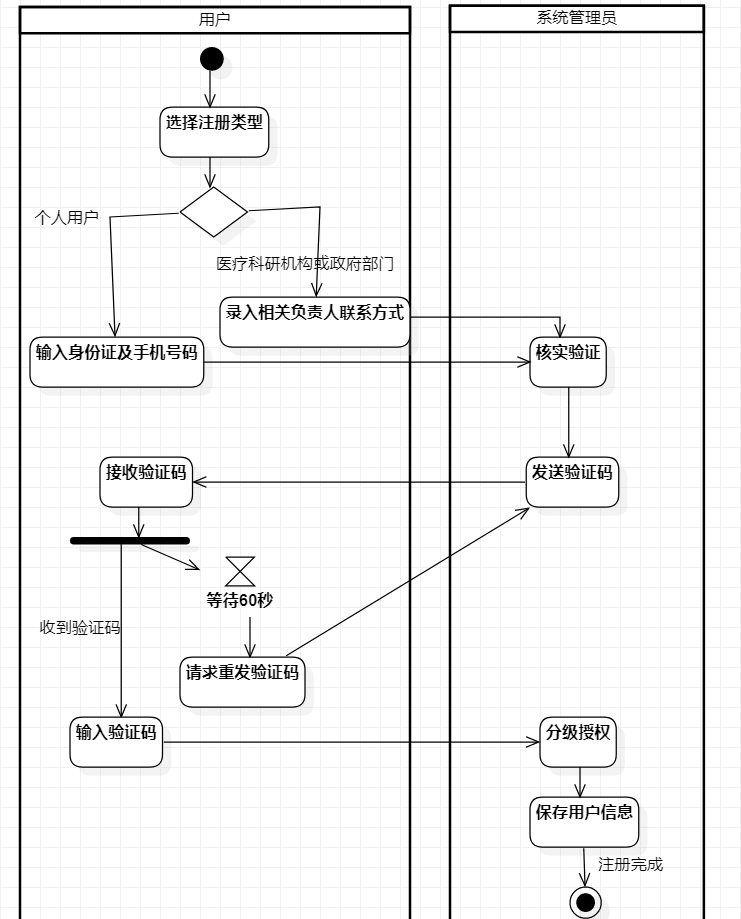
2、系统根据用户类别授予该用户权限，并向关系数据库发送增加用户信息的请求，在数据库中存储信息后，在用户所见页面展示注册状态并跳转至登录页面。

3、用户输入账号密码发送登录请求，系统在关系数据库中查找用户名并验证密码，验证成功则进入系统主页面，否则返回登陆失败信息。

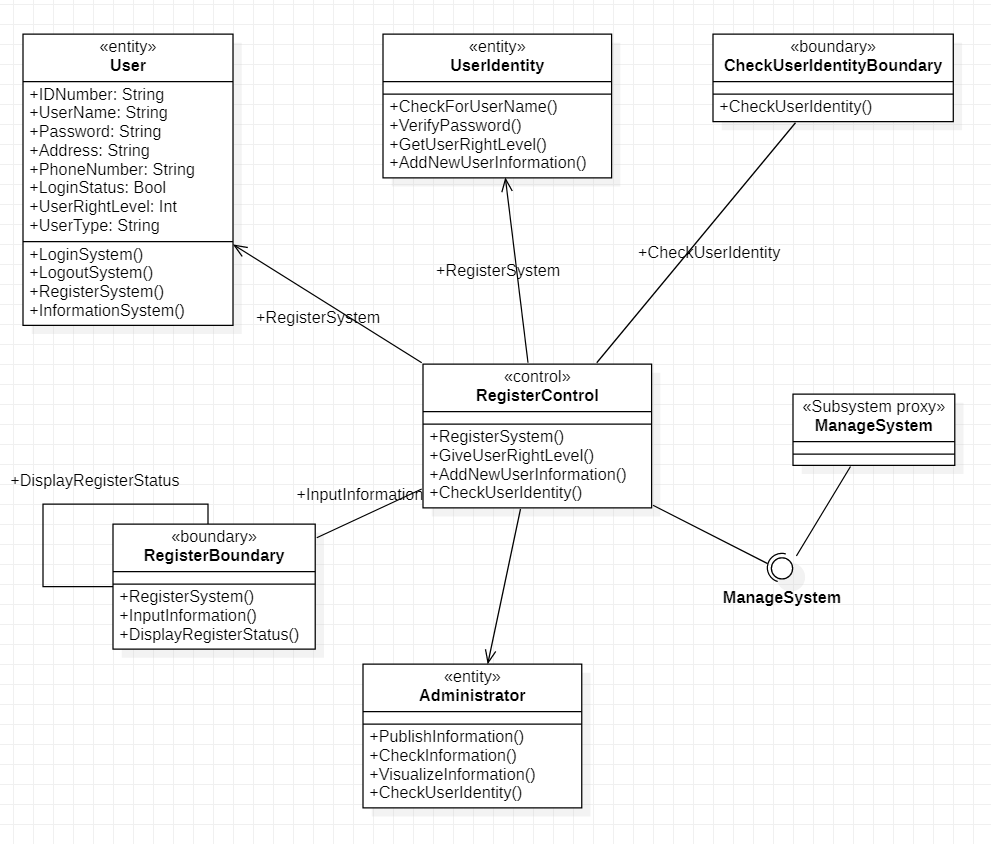
用例规约：

|  |  |
| --- | --- |
| 用例:注册 | |
| 编号 | UC01 |
| 功能 | 用户注册账号，管理员根据信息授予不同用户以不同级别的权限。 |
| 参与者列表 | 所有用户（普通用户、医疗及科研机构、市政府应急部门）、系统管理员 |
| 前置条件 | 用户第一次进入该系统的登录界面。 |
| 基本事件流 | 1、用户从登录界面进入注册界面。  2、用户进行基本信息的录入。  2.1、用户选择注册用户类型。  2.2、根据用户所选的用户类型进入信息录入界面。  2.3、普通用户录入身份证号、手机号等个人基本信息。  2.4、医疗及科研机构录入医院地址、联系方式（公邮等）等信息。  2.5、市政府录入相关负责人的联系方式等信息。  3、用户进行身份验证。  3.1、普通用户使用身份证及手机短信验证码验证身份。  3.2、医疗及科研机构、市政府由系统所发验证码进行验证。  4、系统管理员验证用户身份。  5、系统根据用户录入的用户类别授予权限。  5.1、普通用户授予三级权限（浏览系统公示信息及疫情实时数据信息、上传自身信息）。  5.2、医疗及科研机构授予二级权限（浏览系统所有信息、上传疫情实施数据信息）。  5.3、市政府应急响应部门授予一级权限（选择公开信息上传、发布命令）。  6、用户成功完成注册并返回登录界面。 |
| 代替事件流 | 1、用户未完成录入则退出注册界面时出现未完成注册的的提醒。  2、用户身份验证错误。  2.1、普通用户身份证验证失败时系统提示重新录入。  2.2、普通用户未收到短信验证时，60s后可以要求重新发送短信验证。  2.3、普通用户手机短信验证错误系统提示重新输入。  2.4、医疗及科研机构、市政府验证失败时60s后可以要求重新发送验证码，若超过三次错误则禁止该用户进行身份验证。 |
| 后置条件 | 用户得到自己的账号及所授权限。 |

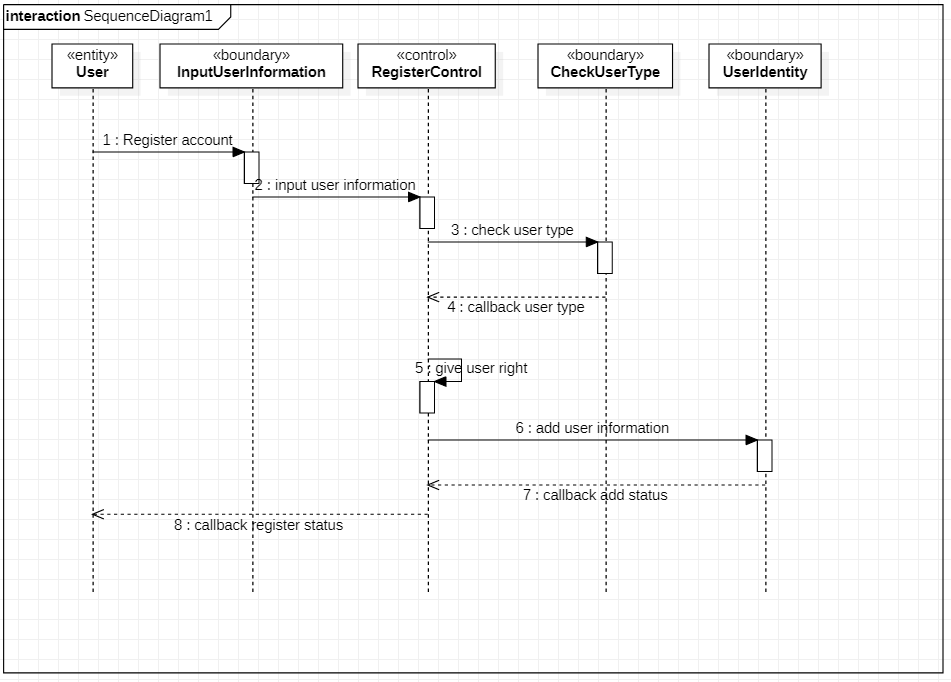
活动图：



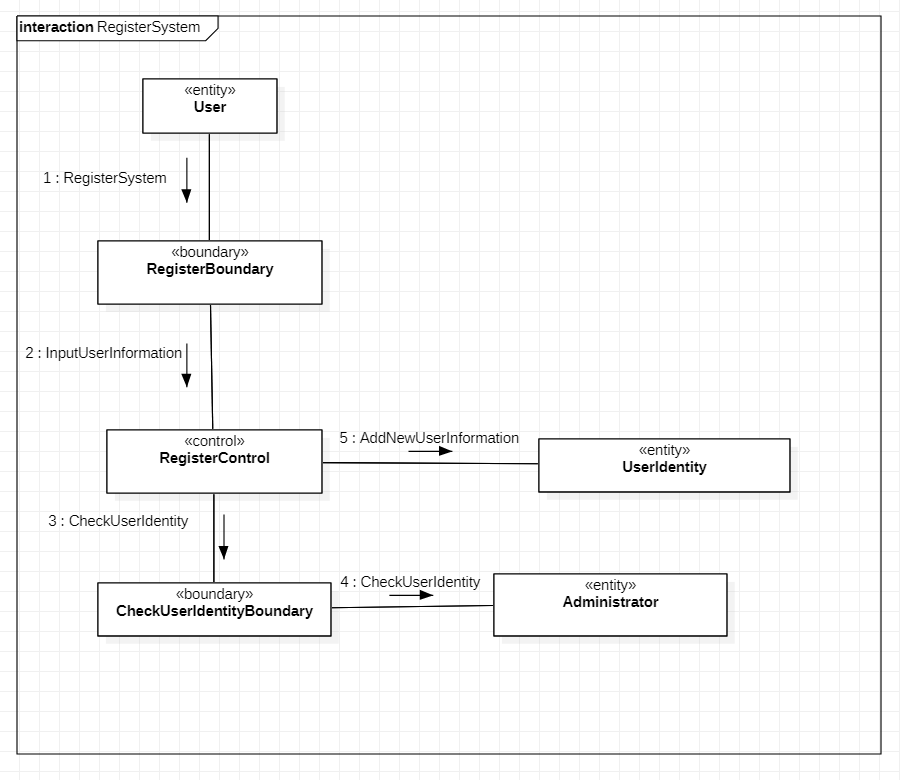
类图：



时序图：



通讯图：



用例：上传信息

用例上传信息过程：

1、用户进入上传信息页面，将需要上传的信息以文本形式上传至页面内，点击保存或者上传。系统反馈保存或上传是否成功信息反馈。

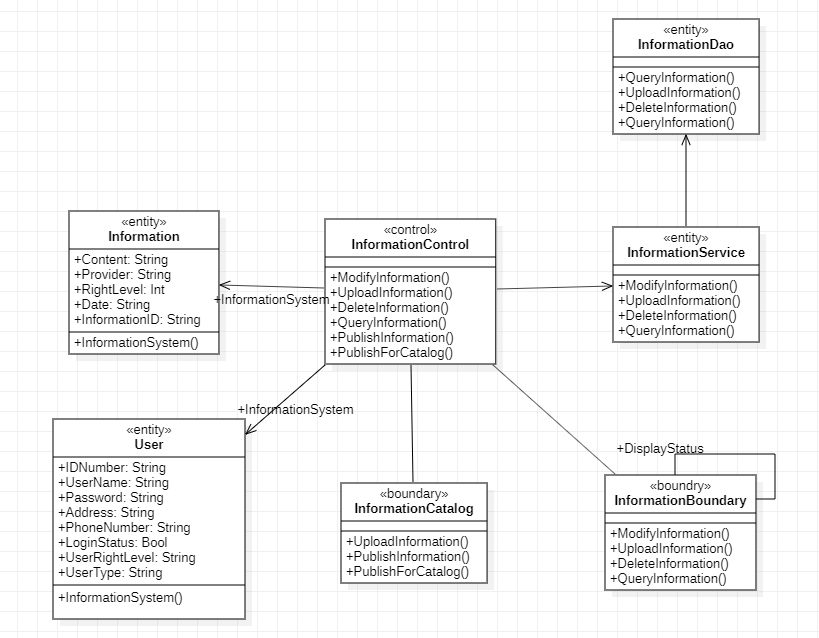
2、若提交成功，信息进入信息数据库，管理员对于数据库进行访问，调取信息，对于上传的信息进行标记审核通过。

3、信息目录界面对于审核成功的信息进行标题展示，同时，建立和数据库之间的联系。

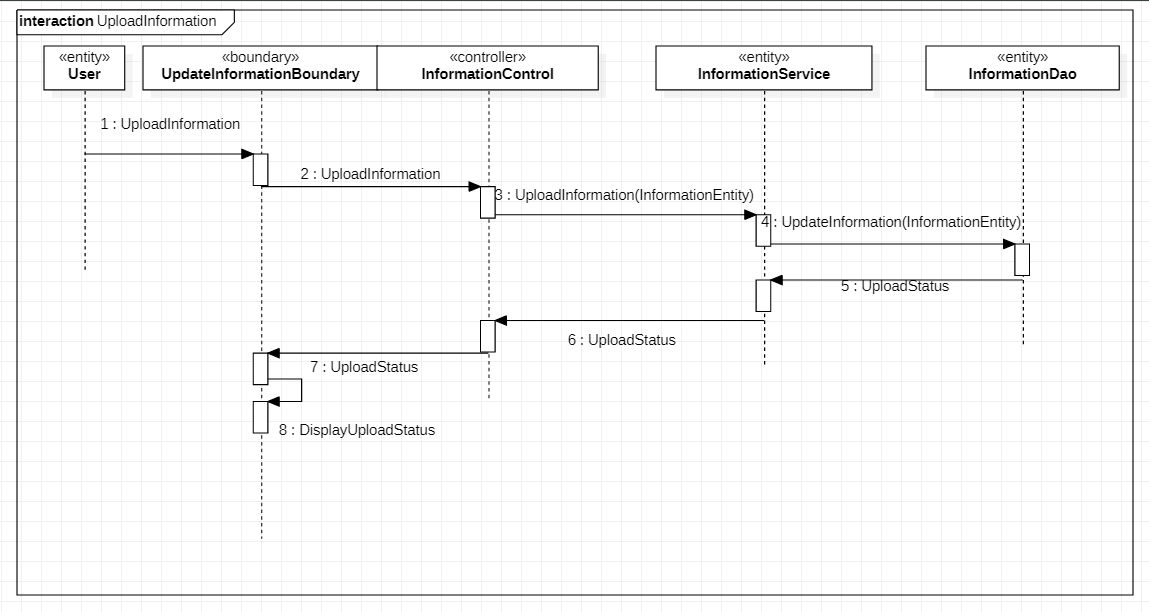
用例规约：

|  |  |
| --- | --- |
| 用例: 上传信息 | |
| 编号 | UC03 |
| 功能 | 使用户能顺利使用该系统上传信息。 |
| 参与者列表 | 普通用户，医疗及科研机构，市政府应急响应部门 |
| 前置条件 | 用户已注册登录。 |
| 基本事件流 | 1、用户在客户端点击用户信息界面。  2、用户点击上传信息按钮。  2.1、根据用户被授予的权限，用户只能选择不同的上传信息模块。  2.2、三级权限（普通用户）只能选择自身信息上传模块。  2.3、二级权限（医疗及科研机构）可选择疫情实时数据上传模块。  2.4、一级权限（市政府应急响应部门）可选择公开信息上传模块。  3、输入相关信息。  4、点击保存按钮。 |
| 代替事件流 | 1、用户退出系统  2、用户未输入要上传的信息。  2.1、用户收到提示“信息不能为空”。  2.2、用户输入信息。  3、用户填写信息后点击返回按钮。  3.1、用户收到提示“确认上传信息”。  3.2、如果用户点击确认按钮，则保存并且上传信息，返回上一界面。  3.3、否则，直接返回上一界面，不上传信息。 |
| 后置条件 | 用户成功上传相关信息。 |

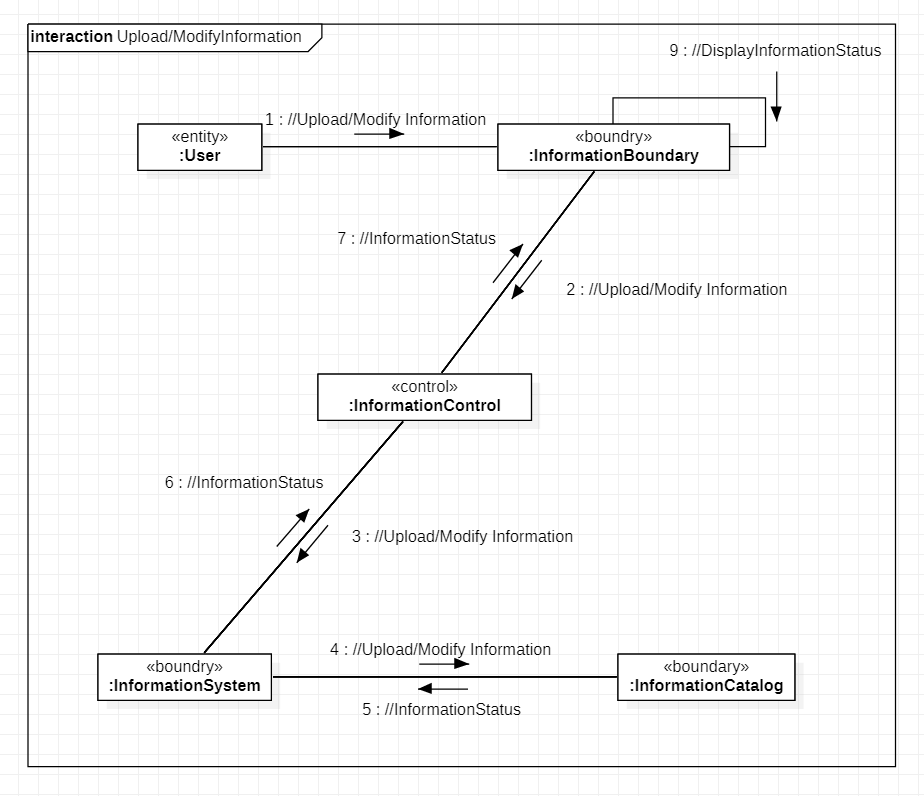
类图：



时序图：



通讯图：



关键设计机制

**critical design mechanisms**

设计机制

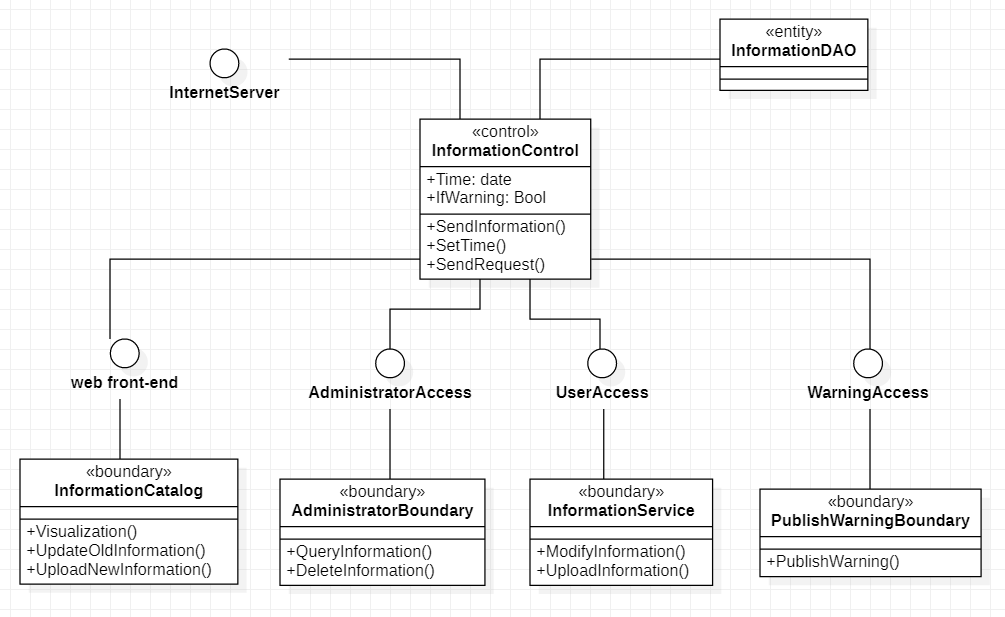
**design mechanisms**

在项目分析阶段，分析机制通常被用来减低分析的复杂性，通过对复杂行为的简化，为软件设计师提高一致性，使之后的环节更为方便。

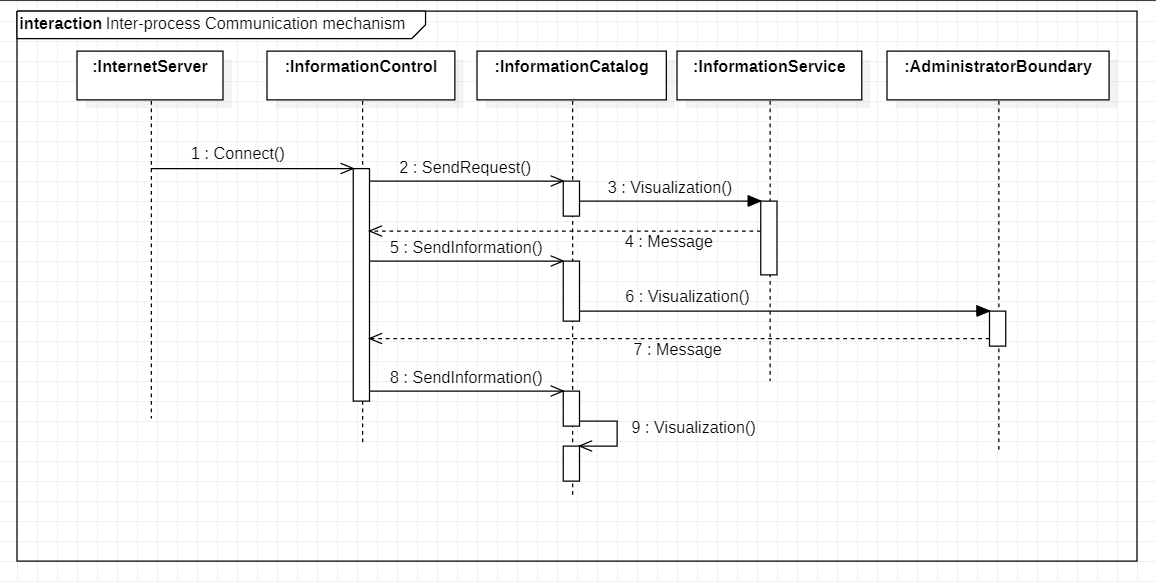
对本系统进行分析时，我们采用四种分析机制，即持久性机制（Persistency mechanism）、进程间通讯机制（Inter-process Communication mechanism）、遗留接口机制（Legacy interface mechanism）和安全机制（Security mechanism）。对于四种机制，我们将重点展示持久性机制和进程间通讯机制：

一、进程间通讯机制

用户端（information service）、管理员（administrator boundary）、政府预警（publish warning boundary）通过接口和信息控制类（information control）关联，该类已经与网络服务器（internet server）连接，最终通过web前端接口呈现在信息目录（information catalog）里，可实现消息的传递。同时，通过数据存取对象（Data Access Object），可以持久化数据进行访问。



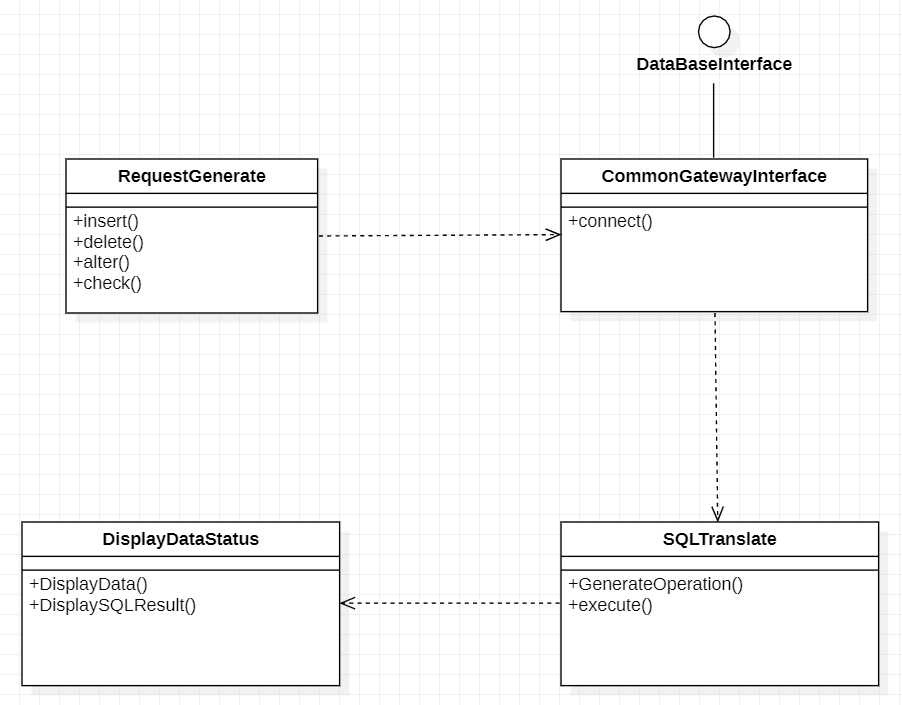
首先，InformationControl与网络服务器InternetServer连接，确保信息联网传递。此后，InformationCatalog利用web实现可视化，用户通过InformationService上传信息到InformationControl，在经过管理员端审核后，信息最终在InformationCatalog上可见。



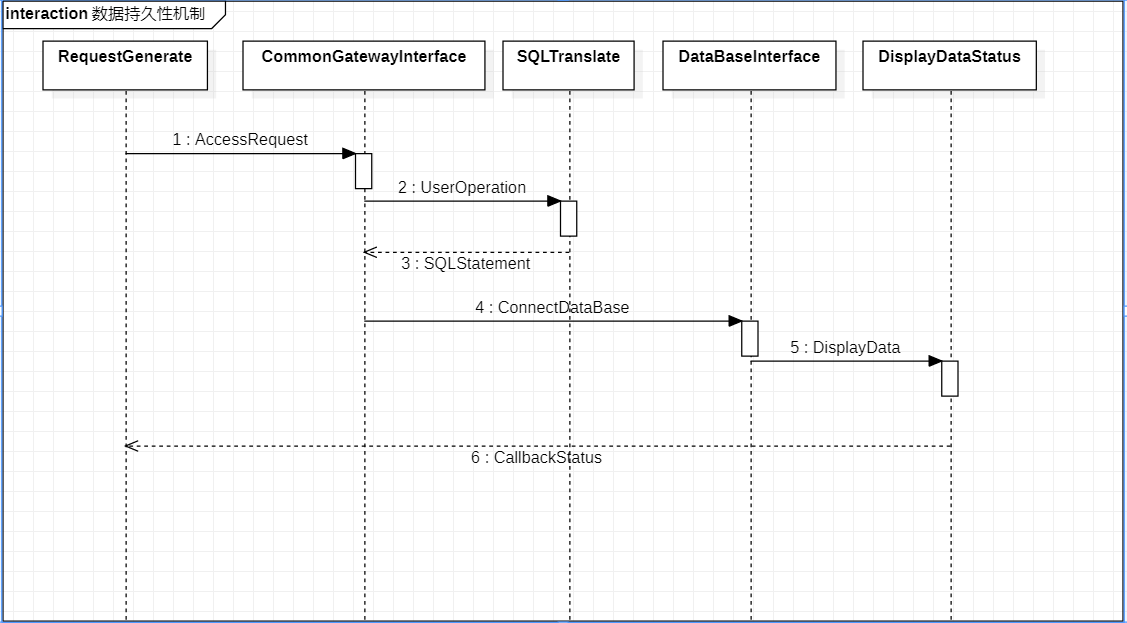
二、持久性机制

在该系统中的注册用例、上传信息等用例中，普通民众、政府、医院等用户都有存储数据的需求，并且部分数据需要持久性保留，例如用户在注册时需要将账户、密码及个人信息存储在数据库中；医疗人员需要上传实时的疫情情况时需要存储数据，这时就需要该机制来保障数据的持久性保留。

RequestGenerate：对用户的响应生成请求，并将该请求送至Web服务器。  
CommonGatewayInterface：即网关界面，通过一个数据库接口，建立Web页面与数据库之间的联系。  
DataBaseInterface：一个数据库接口。  
SQLTranslate：将用户的请求生成相应的SQL语句并执行。  
DisplayDataStatus：展示数据或者展示写数据库操作的结果。



时序图：



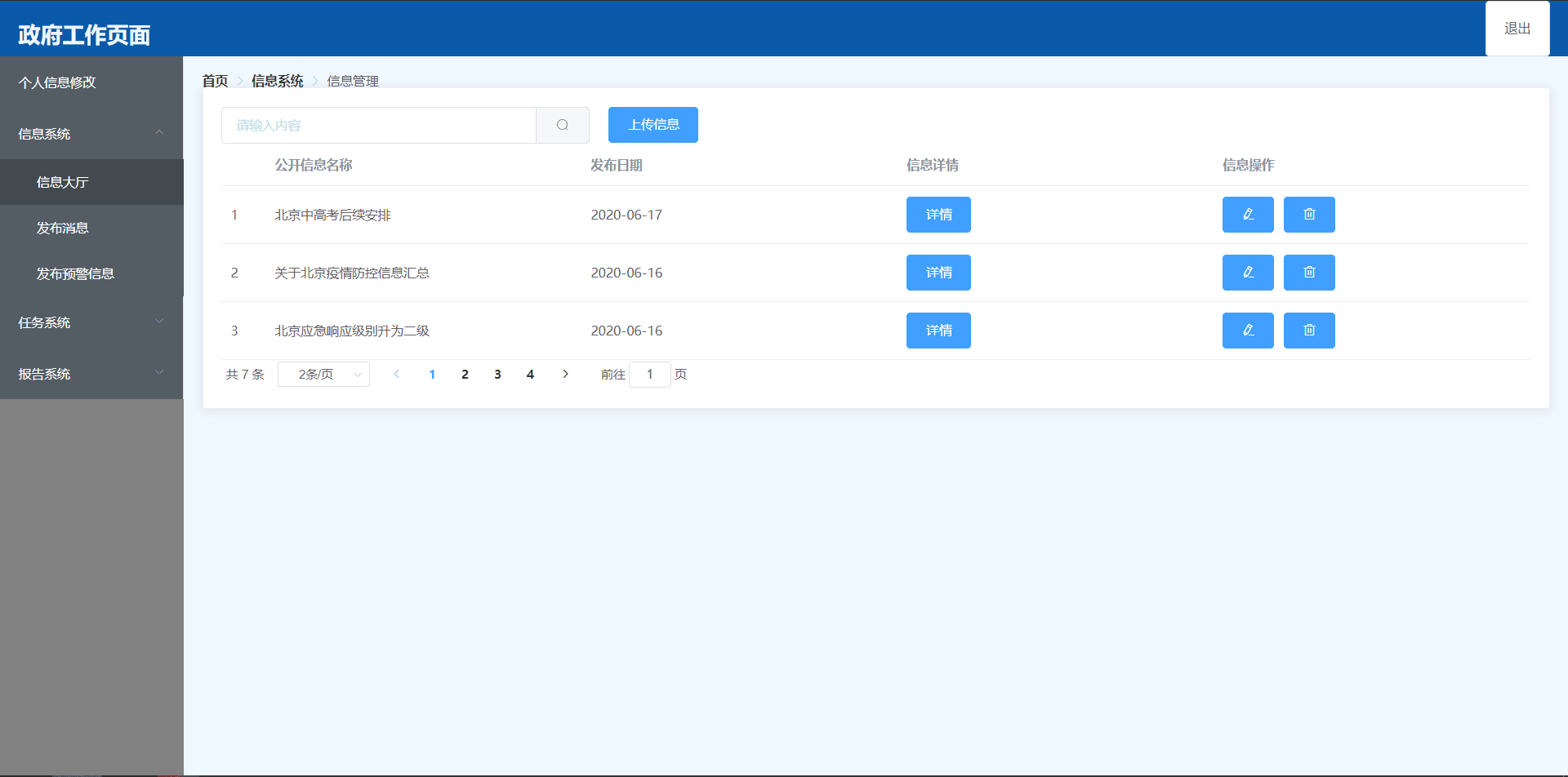
原型设计

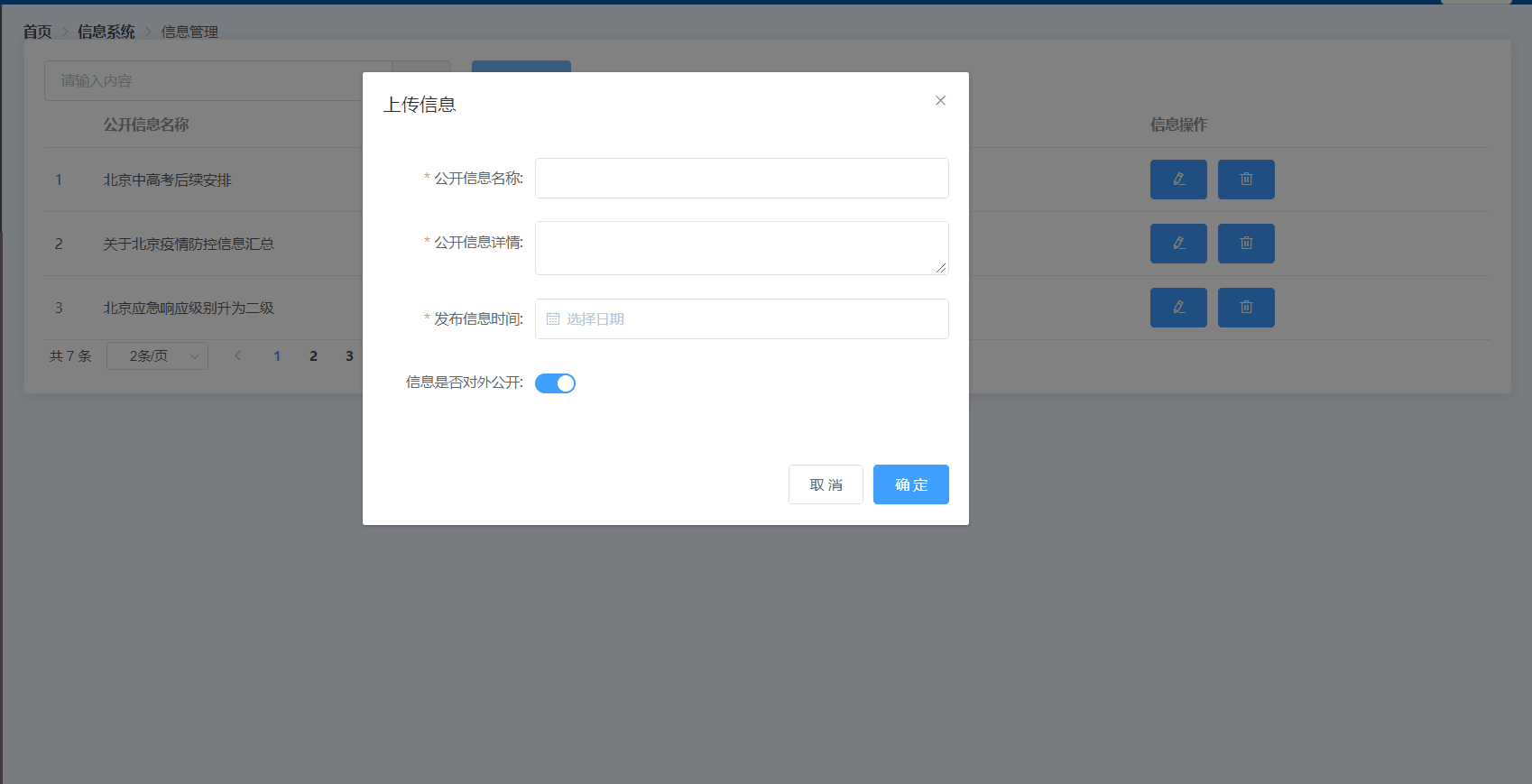
**prototyping**

已实现原型设计：

已实现只有政府工作人员登录后才能进入系统工作页面系统，否则将跳转到政府工作人员登录页面。 在信息系统中，政府工作人员能通过信息大厅按钮查看政府之前已发布的所有信息的概况，也可以查看相应信息的详情或者对其进行修改和删除操作。在发布消息按钮中，可以发布新的信息，发布的信息的内容包括公开信息名称，公开信息详情，公开信息时间，以及是否对外公开等等。政府工作人员也可以通过发布预警信息按钮发布预警信息。

前端浏览器端采用vue框架开发，用vue-cli脚手架搭建应用，vue-router实现路由配置，使用Element UI 控制样式。并且使用HTTP库axios与后端异步的交互json格式的数据。数据库采用Mysql数据库，并且使用非关系型数据库Redis实现缓存功能（图片可放大查看）。





期望效果：

**首页界面**

首页界面呈现的是公共卫生应急管理系统的首页，右上角分别有注册，登录等用例；中间为首页，疫情公布，个人求助，人力与物资调配，个人信息，数据库，地理信息系统，邮箱，问题反馈，资源共享，信息中心；主界面有最新疫情通报，人力与物资调配系统，物理信息系统，资料共享等等。

****

**注册登录界面**

注册登录界面展示的是登陆界面，在统一身份认证下分别是普通用户，系统管理员，高级用户（医疗及科研机构，政府相关部门，其他系统的工作人员）三种不同的登陆方式。下侧有注册和找回密码等用例按钮。



**普通用户上传个人信息界面**

对应着的是邮箱。普通用户可以上传信息，可以设置草稿，信息查询，待办事项，已发送等相关按钮辅助上传信息等用例。



**结合可视化信息的地理信息系统界面**

该页面对应着的是地理信息系统的公开信息内容，以可视化形式展开来，获得当前情况下的疫情最新进展。



**工作人员接受任务，上传信息，接受信息界面**

工作人员包括政府工作人员，医疗系统工作人员等等可以上传信息，接受信息，接受任务等等，可以设置草稿，信息查询，待办事项，已发送等相关按钮辅助上传信息等用例。可以浏览到已公布任务等信息。



# 反馈

**Feedback**

未解决的问题

**unsolved problem**

未解决的问题在于由于时间紧，任务重，可能存在有一些细节的地方没有很好的完成对接。同时存在某些子系统和接口考虑不严谨的情况。

原型设计完成了前端的设计，但未完成服务器的部署问题，需要进一步实践。

自我反思

**self reflection**

1、前期分工安排较为不合理，导致进度变慢，同时由于是线上交流，不如真实情况中交流效果好，相互之间交流没有达到最优效果。

2、本次项目自主查找了很多资料，包括架构分析，架构风格等很多曾经没有接触过的信息，但是查找资料的效率可以再提高一些。

3、本项目由于时间紧迫，由于组内很少有项目经验的通讯，原型设计只能进行到前端的设计和api的选定，没有进入下一阶段的研究。

## 

## 成员贡献度

**Member contributions**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **文档描述** | **架构分析** | **子系统及接口** | **子系统示例** | **用例实现展示** | | **关键设计机制** | | **文档更新** | **原型设计** |
| 石稼晟 | 李翠琪  石稼晟 | 史家豪  石稼晟 | 史家豪 | **注册** | **上传信息** | **即时通讯** | **持久性** | 王子睿 | 石稼晟  李翠琪 |
| 上官宇飞 | 石稼晟 | 上官宇飞 | 莫子毅 |