

项目需求分析报告

指导老师：李宇



2020-7-13

组长：王磊

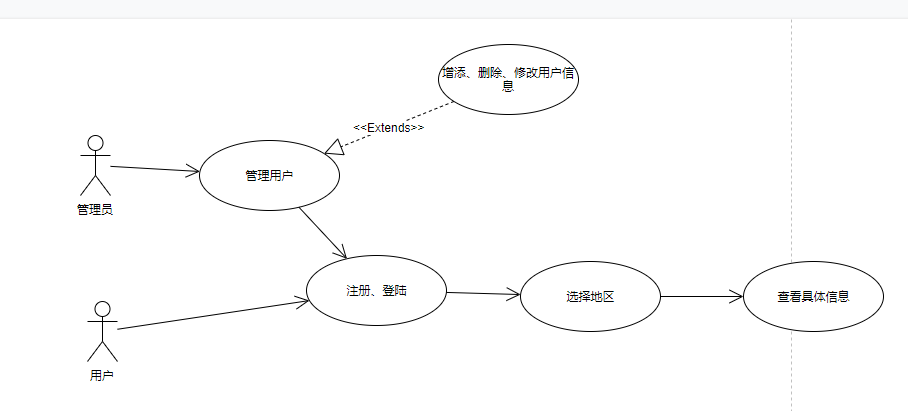
组员：申淳元、燕怡楠、韦永剑、吴明昊

# 项目介绍

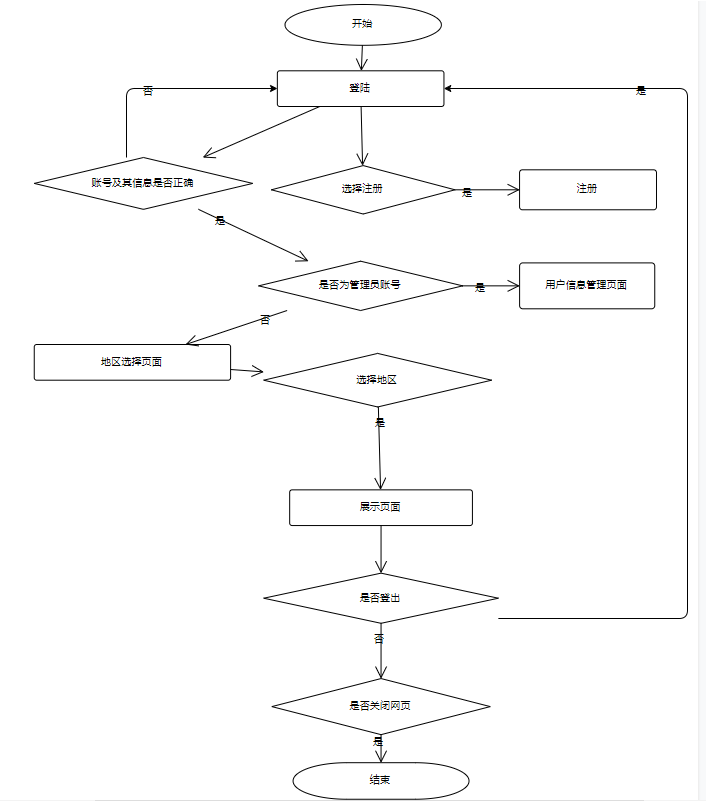
## 1.1 项目背景

该项目是基于时间序列模型对历史气象数据进行分析，数据采集 NCDC 气象官方网站。主要功能是对北京地区 1980 年至 2020 年的气象数据进行 ETL 清洗后，筛选出某年中每一天的最低气温，最高气温，然后使用 Python 对数据进行时间序列分析，预测一周的气温走势。最后，使用 web 终端对分析的结果进行可视化展示。

## 1.2项目用例图



## 1.3项目流程图



# 2.需求分析

## 2.1 功能性需求

### 2.1.1 用户登录注册

1 介绍

进入气温预测系统，可以登录、注册。进入注册界面，输入用户名、密码、确认密码进行注册；进入登录界面，输入用户名、密码及验证码进行登录。

2输入

用户名、密码及验证码

3 处理

注册：读取数据库信息，若已存在玩家的用户名与用户输入的用户名一致，则显示用户名已存在请重新输入；若不存在则帮助用户注册。

登录：读取数据库信息，跟用户输入进行比对，比对成功则登陆成功，进入游戏选择界面。

4 输出

若注册成功则返回登录界面；

若登录成功则进入选择城市界面

### 2.1.2 数据库存储

1 介绍

将mysql数据库与以flask作为框架的服务器进行连接，并通过flask\_sqlalchemy对数据库进行对象化操作，使后端可以通过对象化的操作对数据库进行增删改查等操作。服务器将注册的用户数据保存至数据库，对用户信息进行操作和存储。数据库test\_db中user表存储用户信息，表头信息包括username,password,kind,分别是用户的用户名，密码和用户角色。用户角色分为user,vip,admin,root四种，分别对应不同的权限。

2输入

用户信息，包括用户名，密码和用户角色

3 处理

用户注册时，前端将用户信息以json的数据格式发送到服务器，服务器在数据库进行搜索，如果用户名已注册过向前端返回“exist”信号表示用户名已存在，同时前端进行提示。如果用户名未注册过，用户权限设置为user后添加到数据库user表中。

用户登录时，前端将用户信息以json传输到后端服务器，服务器在数据库中进行检索，如果用户名存在且和密码相匹配,将用户角色返回给前端。

4 输出

根据用户角色和操作返回对应的用户信息，与前端进行信息传递。

### 2.1.3 用户的增、删、改

1 介绍

在RBAC2模型的基础上对用户信息进行管理。用户角色为admin和root的用户可以对用户信息进行增删改查的操作，用户角色为vip的用户只可以查看角色为user的用户的账号信息。

2输入

用户的操作请求，被操作用户的信息

3 处理

首先根据用户的角色以及相应的权限来划分提供给用户的其他用户信息类型，以及是否可以进行操作。用户进行操作后，将修改后的用户信息发送到服务器，服务器在数据库中进行增删改查等相应的操作。

4 输出

操作成功和失败向前端发送相应的信息。

### 2.1.4 数据可视化展示

1 介绍

用户选择省份，网页将对应省会从用户选择日期开始未来七天温度信息以表格和折线图的形式进行展示。

2输入

用户在地图上点击省份，并在跳转页面中选择日期。

3 处理

前端网页获取用户选择地区和日期，将结果以json格式传给后端服务器，服务器读取根据模型进行预测得到的数据，并将其以json的形式传给前端。前端用js中的echarts根据数据绘制图像

4 输出

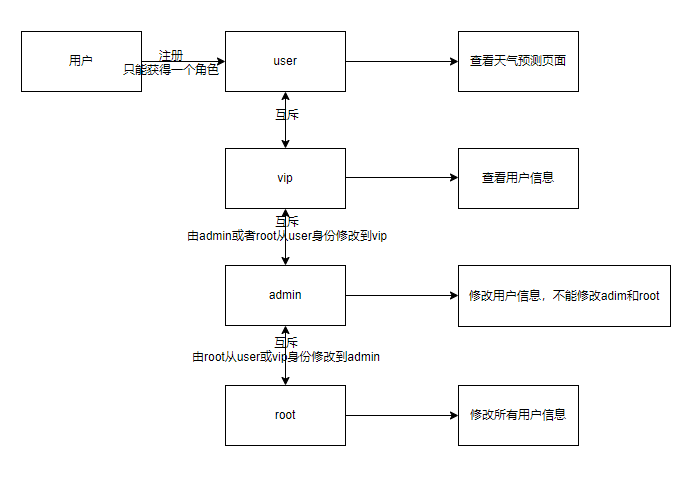
以表格和折线图的形式展示未来七天最高最低气温的预测。

### 2.1.5java调用flask框架

JavaSpring 作为前端利用websocket网络与后端flask进行信息的相互传递，实现网页间的页面跳转，登录注册，查看天气等功能。

### 2.1.6 RBAC模型

我们组采用RBAC模型类型为RBAC-2模型，下图为建模



### 2.1.7 网页部署

1 介绍

通过网页可以直接使用全部功能，保证全天候的查询、预测服务。

2输入

通过域名[http://123.56.45.139](http://123.56.45.139/):8080/java访问，进行相关操作。

3 处理

通过tomcat运行前端，并与同样在服务器上的后端交换数据，后端处理后由前端输出。

4 输出

可访问的内容，实现相关功能·。

### 2.1.8气温预测

1 介绍

根据往年的最高最低气温预测未来一年的数据。

2输入

从气象网站NOAA获取各省份省会的天气数据集。并对数据集进行数据清洗，将无效及异常的数据清洗去除，提取出每天的最高和最低气温。

3 处理

将清洗后数据导入模型中，调整模型参数进行拟合，根据输入训练集进行预测，得到未来一年的当地最高气温和最低气温。

4 输出

将未来一年每天的最高气温和最低气温分别输出保存起来，并以对应的省份名字作为文件名以进行区分，便于服务器调用。

## 2.2 非功能性需求

## 2.2.1 界面友好性需求

系统界面简洁美观，预测动画流畅，各个界面主题风格一致，界面布局简单，功能按钮分布合理。

## 2.2.2 易用性需求

系统提供统一的操作界面和方式，操作界面美观大方，布局合理，功能完善，易于操作，对于新用户容易上手，并提供适当的帮助信息，使产品提高符合用户习惯的能力以及其对使用的期望。

## 2.3需求分级

|  |  |
| --- | --- |
| **需求名称** | **需求分级** |
| 天气预测 | A |
| 数据可视化展示 | A |
| 数据库储存 | A |
| Java调用flask框架 | A |
| RBAC模型 | B |
| 用户的增、删、改 | B |
| 网页部署 | C |

A.必须的：绝对基本的特性；如果这种需求不满足，整个项目不能正常交付使用。

B.重要的：不是基本的特性，但如果这些需求不满足会使整体项目工作的价值下降，影响产品的生存能力。

C.最好有的：期望的特性；但省略一个或多个这样的特性不会影响产品的生存能力。