

**组长：王子昂**

**组员：王子昂**

**陈灯月**

**王迪**

**黎佩瑜**

**童路勤**

天气预测

系统设计文档

系统设计文档

[1．引言 2](#_Toc45550812)

[1.1编写目的 2](#_Toc45550813)

[1.2项目背景 2](#_Toc45550814)

[2．任务概述 2](#_Toc45550815)

[2.1目标 2](#_Toc45550816)

[2.2运行环境 2](#_Toc45550817)

[2.3需求概述 3](#_Toc45550818)

[2.4条件与限制 3](#_Toc45550819)

[3．总体设计 4](#_Toc45550820)

[3.1处理流程 4](#_Toc45550821)

[3.2总体结构和模块外部设计 5](#_Toc45550822)

[3.3功能分配 7](#_Toc45550823)

[3.3接口设计 9](#_Toc45550824)

[3.3.1用户输入输出接口设计 9](#_Toc45550825)

[3.3.2管理员输入输出接口设计 11](#_Toc45550826)

# 1．引言

## 1.1编写目的

由前面的需求分析，得出了软件的基本需求，和基本的数据流图，要实现整个项目需要对用户的需求进行设计，概要设计主要是利用比较抽象的语言对整个需求进行概括，确定软件的体系结构，主要描述“设计什么”和“如何设计”。根据需求规格说明书所列出的功能和性能需求，要明确说明本项目的设计思路、整体结构和主要有哪些模块组成，同时还要描述清楚各模块的结构及模块间信息传递的接口。为设计人员提供详细设计依据；为研发管理人员提供管理依据；为系统测试人员提供测试依据；为文档人员提供书写各种说明书的依据。

本报告的预期读者是参与开发的设计、开发、测试、文档等人员。

## 1.2项目背景

1、软件的名称：天气预测网页

2、开发工具与语言：MySql，Navicat，Eclipse，PyCharm，Spark，Java，Python，JavaScripts，Vscode，Tomcat，shiro等

3、项目的任务提出者：北京交通大学软件学院。

4、开发者：王子昂，王迪，黎佩瑜，陈灯月，童路勤。

5、用户：全体。

6、实现软件的单位：软件学院小学期实践课小组。

# 2．任务概述

## 2.1目标

对项目进行更细化的系统软件设计，要明确说明本项目的设计思路、整体结构和主要有哪些模块组成，使项目执行更高效。

## 2.2运行环境

操作系统：Microsoft Windows 2000 Advanced Server

支持环境：IIS 5.0

数 据 库：MySql Server 2019

## 2.3需求概述

天气预报（测）或气象预报（测）是使用现代科学技术对未来某一地点地球大气层的状态进行预测。从史前人类就已经开始对天气进行预测来相应地安排其工作与生活（比如农业生产、军事行动等等）。今天的天气预报主要是使用收集大量的数据（气温、湿度、风向和风速、气压等等），然后使用目前对大气过程的认识（气象学）来确定未来空气变化。

本项目主要通过对过去的天气数据集进行清洗，建立模型对清洗后的数据进行训练并预测出未来的天气情况，将这些数据在网页端以图表的形式显示给用户，让用户更加直观观察未来的天气情况。天气预测系统采用网页形式访问的形式查询全国各地重要城市的未来天气情况。用户需要方便地查询全国各地重要城市的未来天气情况。为了使用此功能，用户可以通过在网页上进行注册并登录，进入天气查询，查询各地的未来天气。对于管理员，天气预测系统采用网页形式访问的形式使管理员登录后可以方便地进行用户的增添，删除，修改，查询等操作。

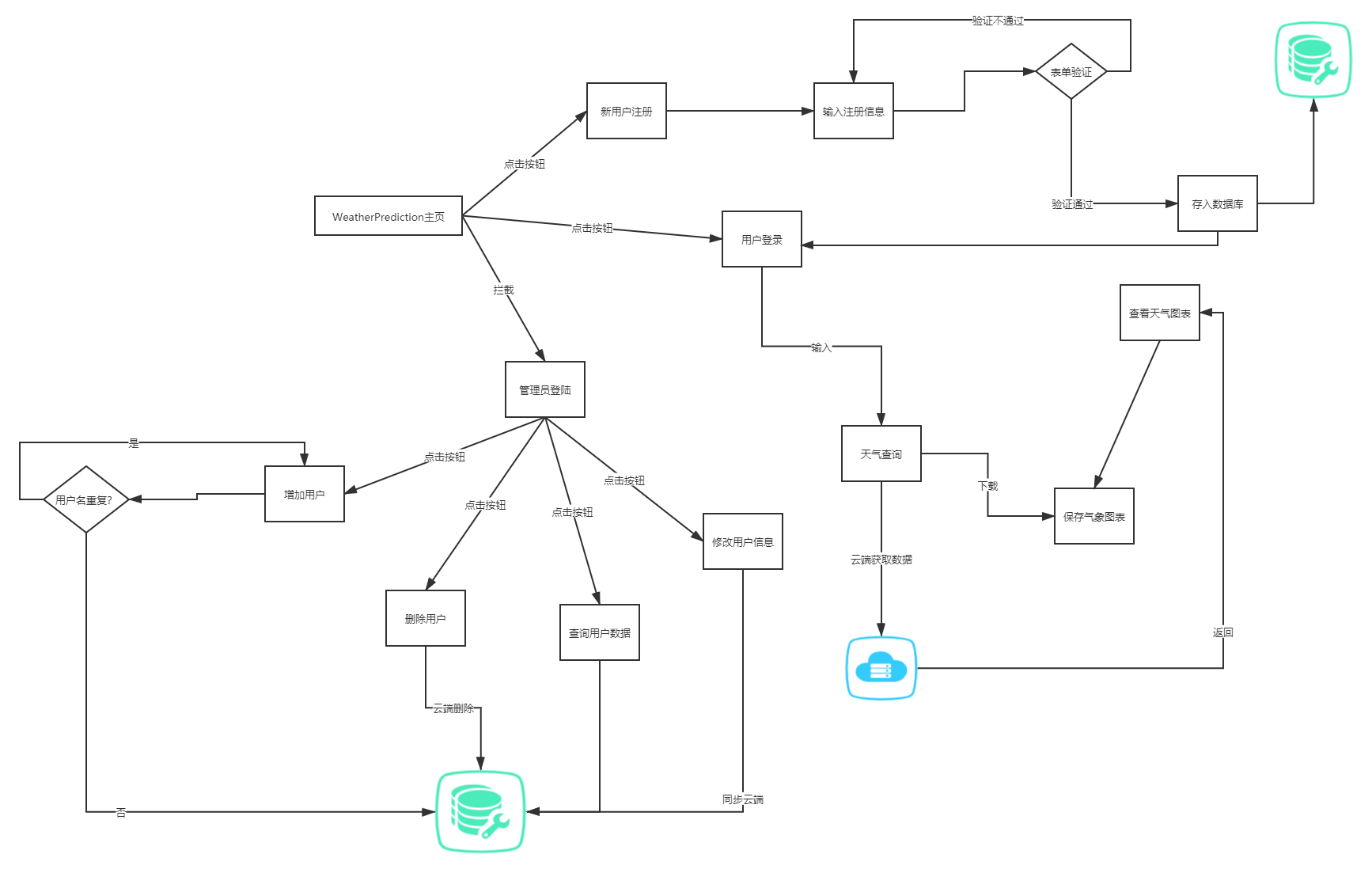
## 2.4条件与限制

1、项目应具备条件：应提供相应的硬件配置，相应的软件及一定数量的技术人员和管理人员等，以及相应的开发环境。

2、工作及完成期限：应按照老师的计划完成任务和工作，在7月12日前完成项目整体开发。

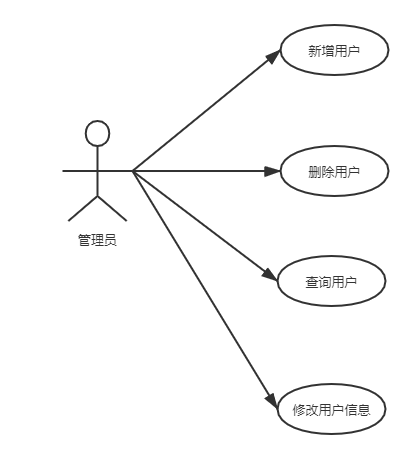
# 3．总体设计

## 3.1处理流程

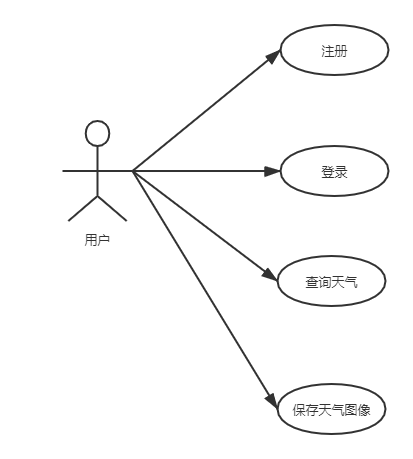


**3.1 处理流程**

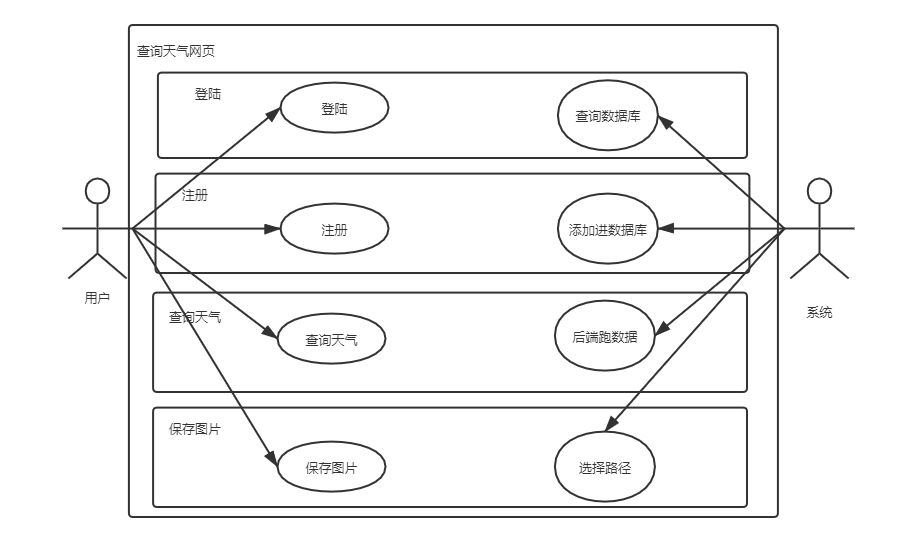
## 3.2总体结构和模块外部设计



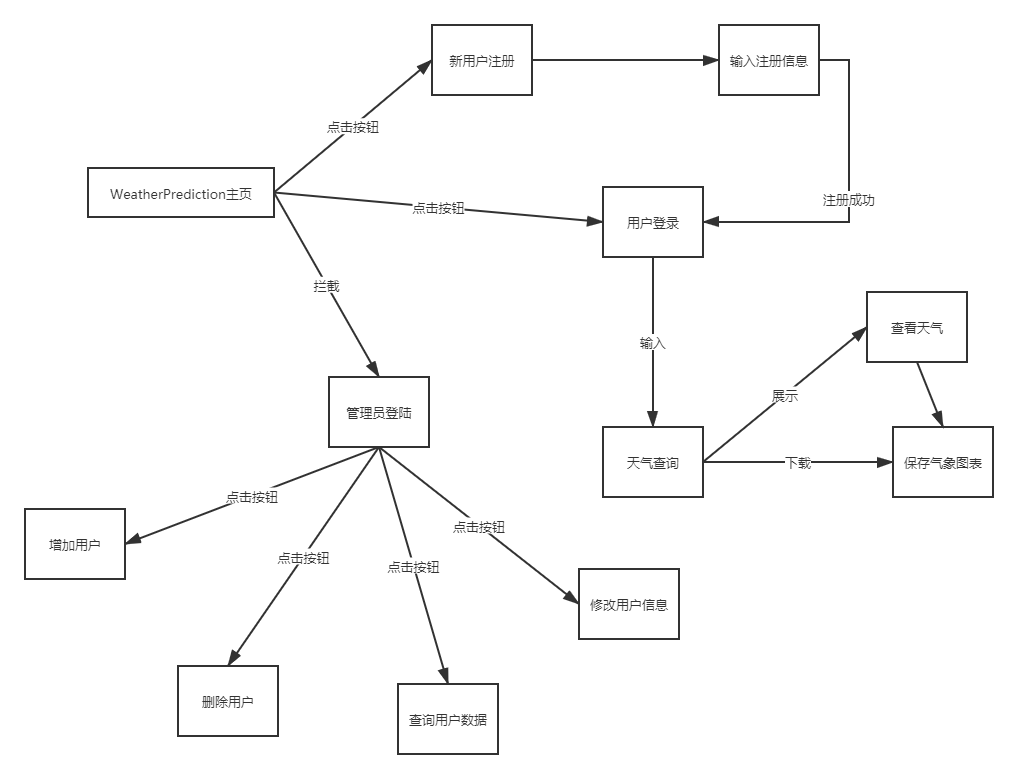
**3.2-1 管理员模块**



**3.2-2 普通用户模块**

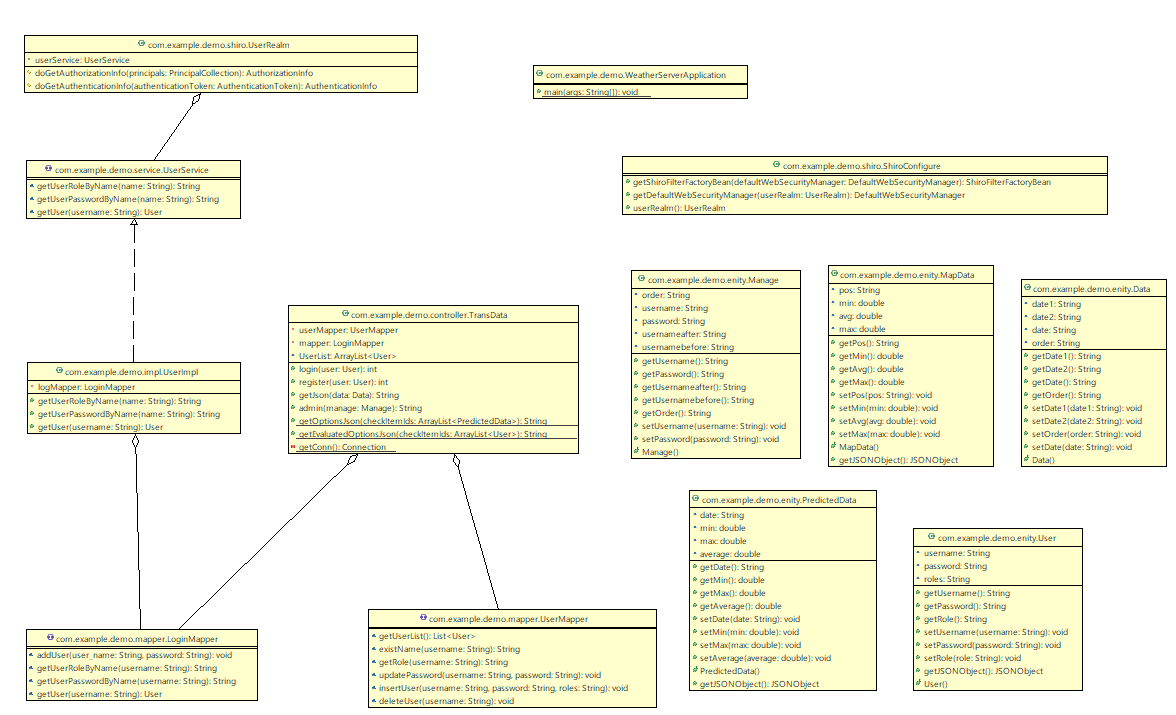
****

**3.2-3 用户和系统**



**3.2-4 总体交互结构**

## 3.3功能分配



com.example.demo. WeatherServerApplication.java

springbooot启动类

com.example.demo.controller.Transdata.java

public String login(@RequestBody User user)

从前端收到用户名username和密码password，通过shiro认证密码进行登陆。

public String getJson(@RequestBody Data data)

从前端收到日期与指令，指令chart返回前端该日期往后七天（含该日期）北京的气温预测数据，其他指令返回全国各城市该日期的气温预测数据。

public String admin(@RequestBody Manage manage)

从前端接收一个java Bean实体类，根据all/delete/modify/add相关指令，对数据库进行对应的增删查改，并返回前端相应的信息（操作成功/失败原因）。

public static String getOptionsJson(ArrayList<PredictedData> checkItemIds)

将预测数据arraylist转成json串

public static String getEvaluatedOptionsJson(ArrayList<User> checkItemIds)

将用户数据测数据arraylist转成json串

private static Connection getConn()

连接数据库并返回相应的字符串

com.example.demo.enity

Data.java : pickdate页面获取前端请求的Java bean

Manage.java: admin页面获取前端请求的Java bean

MapData.java: 为了给前端发送json串。自定义类MapData存一个地方的温度数据，方便转化。

PredictedData.java: 为了给前端发送json串。自定义类PredictedData存北京某天开始的七天温度数据，方便转化。

User.java: 为了给前端发送json串.自定义类User存一个人的全部信息，转化后传给前端显示在用户管理页面。

com.example.demo.impl. UserImpl.java

利用LoginMapper对象实现UserService端口方法。

public String getUserRoleByName(String name)

通过用户名获得他的角色

public String getUserPasswordByName(String name)

通过用户名获得他的密码

public User getUser(String username)

通过用户名获得该User类

com.example.demo.mapper.\*

LoginMapper.java: 从数据库中获取名字，密码和User对象。实现增加新的用户到数据库

UserMapper.java: 获取全部用户List, 获取指定用户名，获取角色，更新密码，增加新的用户，删除用户

com.example.demo.service.UserService.java

端口，包括通过用户名获得他的角色，通过用户名获得他的密码，通过用户名获得该User类。

com.example.demo.shiro.\*

ShiroConfigure.java:

PublicShiroFilterFactoryBean getShiroFilterFactoryBean(@Qualifier("getDefaultWebSecurityManager") DefaultWebSecurityManager defaultWebSecurityManager)

设置安全管理器，添加Shiro的内置过滤器

@Bean(name="getDefaultWebSecurityManager")

public DefaultWebSecurityManager getDefaultWebSecurityManager(@Qualifier("userRealm") UserRealm userRealm)

@Bean(name="userRealm")

public UserRealm userRealm()

shiro常规要写的内容。

UserRealm.java:

protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals)

获取用户身份做权限认证

protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authenticationToken)

获取输入的用户名和密码，交给shiro判断密码正确与否，做身份认证。

## 3.3接口设计

### 3.3.1用户输入输出接口设计

|  |  |  |  |
| --- | --- | --- | --- |
| **注册输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **用户名** | **字符串** | **无** | **键盘输入** |
| **密码** | **字符串** | **无** | **键盘输入** |
| **验证码** | **字符串** | **数字** | **键盘输入** |

|  |  |  |  |
| --- | --- | --- | --- |
| **登录输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **用户名** | **字符串** | **无** | **键盘输入** |
| **密码** | **字符串** | **无** | **键盘输入** |
| **验证码** | **字符串** | **数字** | **键盘输入** |

|  |  |  |  |
| --- | --- | --- | --- |
| **天气查询输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **起始日期** | **字符串** | **格式‘yy-mm-dd’** | **鼠标单击响应，智能选择** |
| **结束日期** | **字符串** | **格式‘yy-mm-dd’** | **鼠标单击响应，智能选择** |
| **单个日期** | **字符串** | **格式‘yy-mm-dd’** | **鼠标单击响应，智能选择** |

|  |  |  |
| --- | --- | --- |
| **登录输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **登陆成功** | **弹窗** | **登陆成功，正在跳转中。。。。。** |
| **登录失败** | **弹窗** | **登陆失败！** |

|  |  |  |
| --- | --- | --- |
| **注册输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **注册成功** | **弹窗** | **注册成功，请前往登陆。。。** |
| **注册失败** | **弹窗** | **注册失败** |

|  |  |  |
| --- | --- | --- |
| **图表输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **Echart** | **Ve-line** |  |
| **Echart** | **Ve-map** |  |

输入操作：点击注册按钮。

输出效果：进入注册页面

输入操作通过界面进行。

输入操作：键盘输入用户信息，点击注册。

输出效果：注册成功后跳转登录界面，并弹窗。

输入操作通过界面进行。

输入操作：点击登录按钮。

输出效果：进入登录页面

输入操作通过界面进行。

输入操作：键盘输入用户信息，点击登录。

输出效果：登录成功后跳转界面，并弹窗。

输入操作通过界面进行。

输入操作：选择起始日期和终止日期，点击搜索按钮。

输出效果：显示未来北京的天气折线图

输入操作通过界面进行。

输入操作：选择单个日期，点击搜索按钮。

输出效果：显示未来全国某一天的天气图

输入操作通过界面进行。

### 3.3.2管理员输入输出接口设计

|  |  |  |  |
| --- | --- | --- | --- |
| **管理员查询用户输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **用户名** | **字符串** | **无** | **键盘输入** |

|  |  |  |  |
| --- | --- | --- | --- |
| **管理员删除用户输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **删除** | **按钮** | **无** | **鼠标点击响应** |

|  |  |  |  |
| --- | --- | --- | --- |
| **管理员修改用户输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **用户名** | **字符串** | **不重复** | **键盘输入** |
| **密码** | **字符串** | **无** | **键盘输入** |

|  |  |  |  |
| --- | --- | --- | --- |
| **管理员增加用户输入** | | | |
| **数据名称** | **数据类型** | **约束条件** | **输入方式** |
| **用户名** | **字符串** | **不重复** | **键盘输入** |
| **密码** | **字符串** | **无** | **键盘输入** |

|  |  |  |
| --- | --- | --- |
| **管理员删除用户输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **删除成功** | **弹窗** | **Wow, 删除成功了！** |
| **删除失败** | **弹窗** | **删除失败！** |

|  |  |  |
| --- | --- | --- |
| **管理员增加用户输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **增加成功** | **弹窗** | **新用户增加成功。** |
| **增加失败** | **弹窗** | **用户名重复** |

|  |  |  |
| --- | --- | --- |
| **管理员修改用户信息输出** | | |
| **数据名称** | **数据类型** | **备注** |
| **修改成功** | **弹窗** | **编辑成功！** |
| **修改失败** | **弹窗** | **用户名重复** |

输入操作：点击增加按钮。

输出效果：增加新用户成功后弹窗

输入操作通过界面进行。

输入操作：点击删除按钮。

输出效果：删除用户成功后弹窗

输入操作通过界面进行。

输入操作：点击修改按钮。

输出效果：修改用户信息成功后弹窗

输入操作通过界面进行。

输入操作：键盘输入用户信息。

输出效果：自动搜出用户

输入操作通过界面进行。