**阿巴天气预测系统**

**系统设计文档**

**组长：于沛尧**

**组员：蔡子墨**

**李卓航**

**王乐言**

**于超**

**目 录**

[1 引 言 1](#_Toc45540912)

[1.1编写目的 1](#_Toc45540913)

[1.2背景 1](#_Toc45540914)

[1.3定义 1](#_Toc45540915)

[1.4参考资料 1](#_Toc45540916)

[2 系统设计 1](#_Toc45540917)

[2.1 系统整体架构设计 1](#_Toc45540918)

[2.2 整体功能设计 2](#_Toc45540919)

[2.3 用户功能模块设计 2](#_Toc45540920)

[2.2.1 root管理员功能设计 3](#_Toc45540921)

[2.2.2 管理级用户功能设计 3](#_Toc45540922)

[2.2.3 组长级用户功能设计 4](#_Toc45540923)

[2.2.4 普通用户功能设计 4](#_Toc45540924)

[3 数据库设计 5](#_Toc45540925)

[4 接口及过程设计 6](#_Toc45540926)

[4.1 数据处理服务器接口 6](#_Toc45540927)

[5 界面设计 7](#_Toc45540928)

[5.1登陆界面 7](#_Toc45540929)

[5.2 注册界面 7](#_Toc45540930)

[5.3 主功能界面 8](#_Toc45540931)

[6 小 结 11](#_Toc45540932)

# 1 引 言

## 1.1编写目的

本文档将从系统整体架构、各模块系统架构、数据库、接口、界面设计等方面进行综合描述，其中使用不同的视图来描述系统的每个方面。该文档用于记录与表述目前已有系统设计，层次分明，以便于开发者高效地开发与对系统的快速管理与完善。

## 1.2背景

阿巴天气预测系统是一个科学的预测系统，旨在为用户提供可靠的天气预测，方便用户外出旅行、居家备服。

天气预测系统具有强大的系统功能，包括用户注册、用户登录、数据展示、用户管理、部门管理、角色管理、权限管理等多种功能。

## 1.3定义

无

## 1.4参考资料

无

# 2 系统设计

## 2.1 系统整体架构设计

整体框架主要分为四个部分：用户客户端、网页后端服务器、数据处理后端、与相关数据库。

网页后端服务器基于http协议，主要承担了与客户端进行直接交互、账户管理、与数据处理后端进行交互的任务，并将得到的预测数据进行画面渲染，最终现实到用户端。用户信息储存在用户数据库中。

数据处理后端的主要任务有：使用ARIMA对气象数据进行气象预测，生成每个城市的未来7天的预测数据后，通过websocket长连接与网页后端服务器进行交互以json形式数据返回数据。

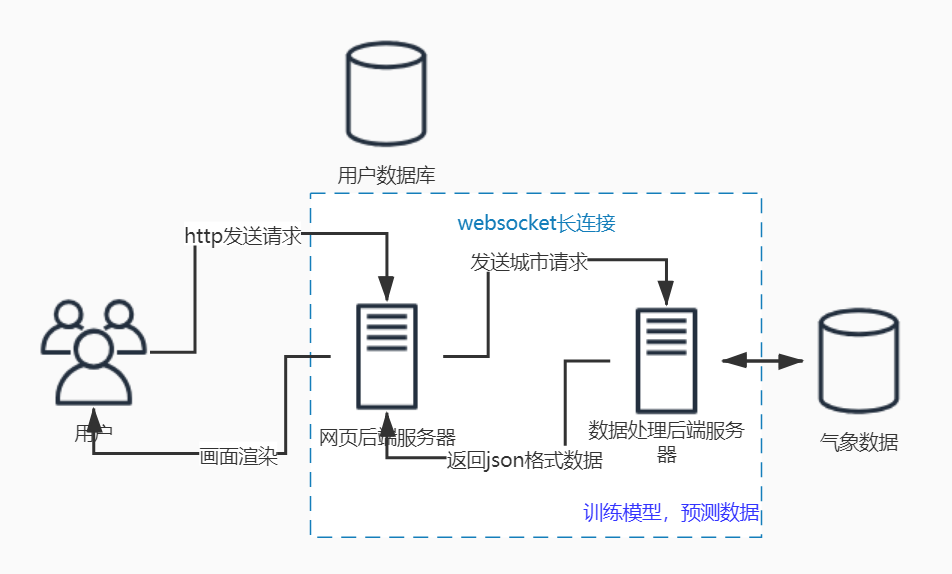


图 2.1 系统框架拓扑图

## 2.2 整体功能设计

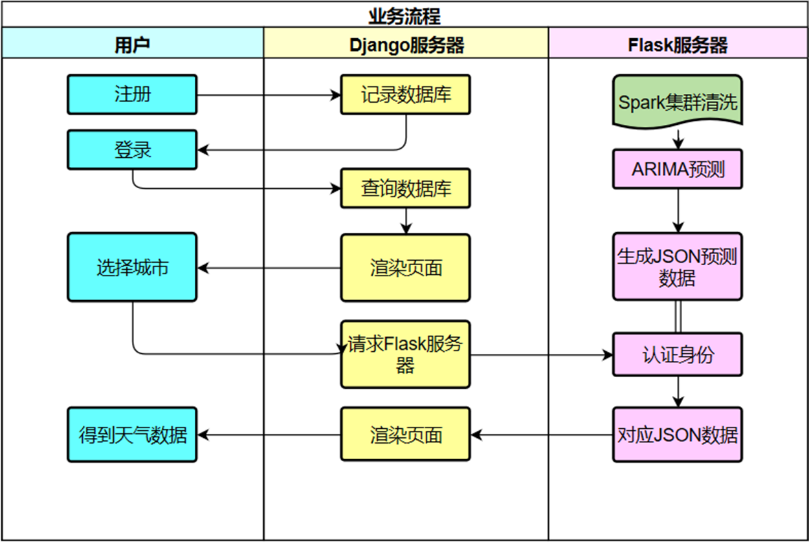


图 2.2 系统功能时序图

## 2.3 用户功能模块设计

根据RBAC模型的设计，我们将系统的角色主要设置为四种：root管理员、管理级用户、组长级用户、普通用户，根据用户的不同，画出对应的功能模块图。

### 2.2.1 root管理员功能设计

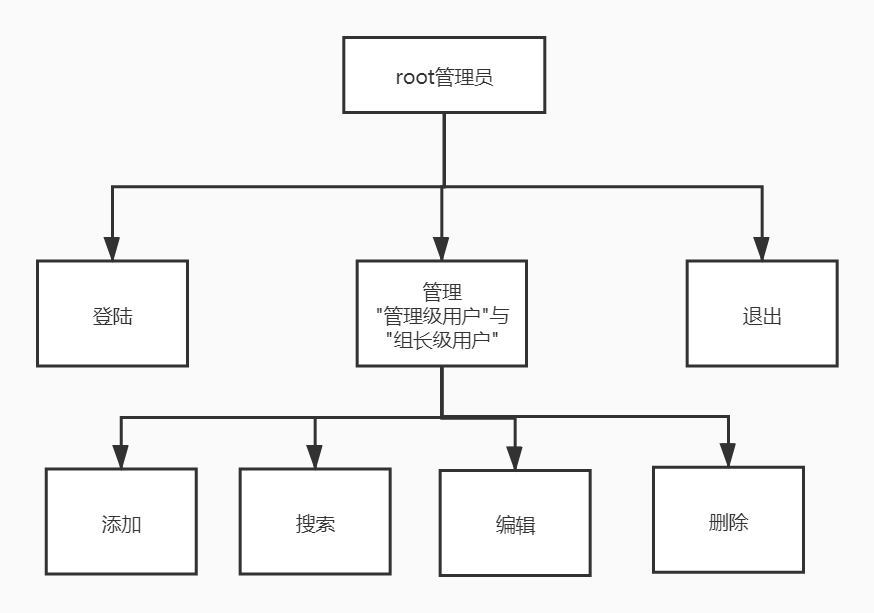


图 2.2.1 功能模块图

### 2.2.2 管理级用户功能设计

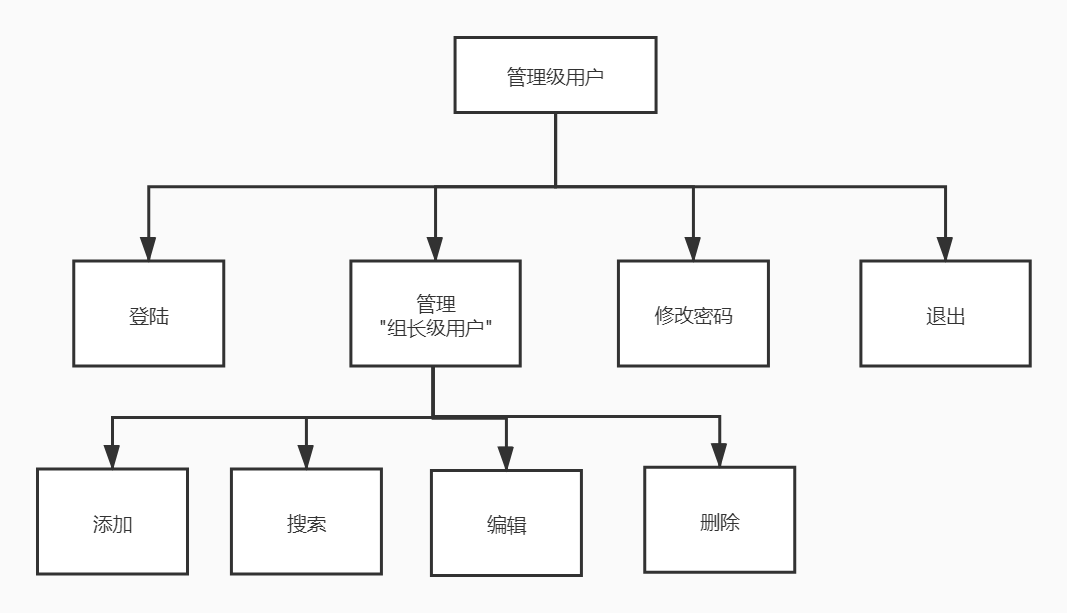


图 2.2.2 功能模块图

### 2.2.3 组长级用户功能设计

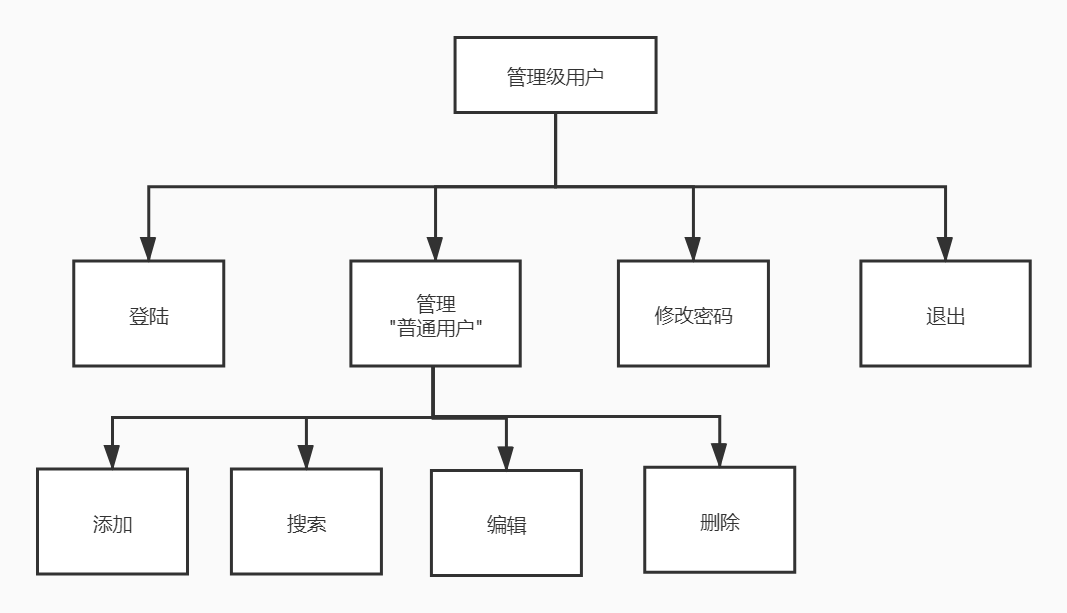


图 2.2.3 功能模块图

### 2.2.4 普通用户功能设计

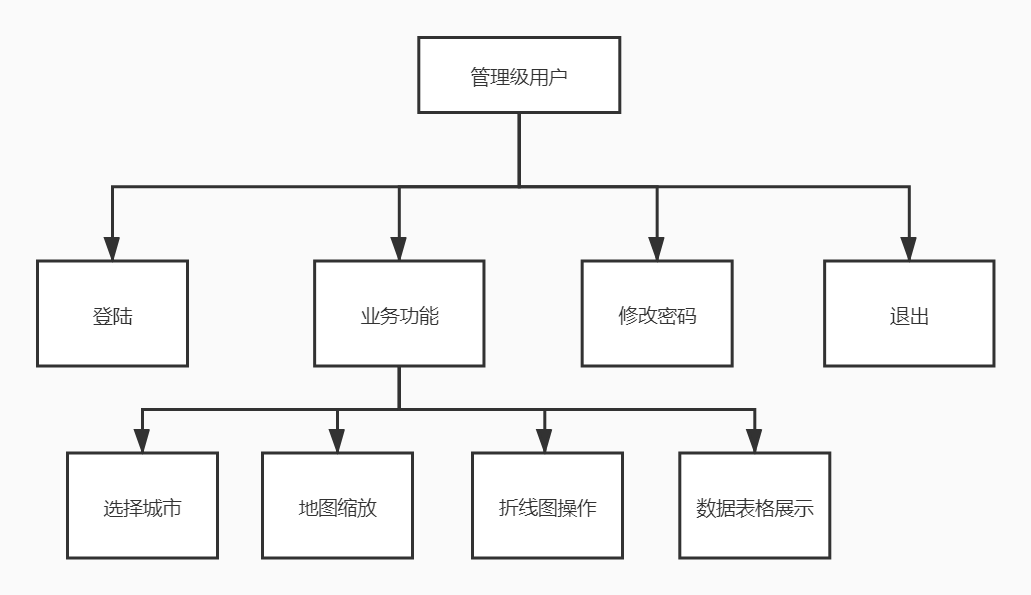


图 2.2.4 功能模块图

# 3 数据库设计

系统中数据库主要是用户管理数据库，本章以表格的形式列出系统内每一个数据库表的表结构，即字段名、类型、长度、小数位数、是否主键、索引、是否允许为空、说明等。

包括如下各表如表4.1所示。

表4.1 系统数据库表

|  |  |  |
| --- | --- | --- |
| 序号 | 表名(英文) | 说明 |
| 1 | login\_user | 普通用户信息 |
| 2 | auth\_user | 管理员用户信息 |
|  |  |  |

（1）普通用户基本信息表(login\_user)

用来存储普通用户信息的基本情况，如表3.2(1)所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性(英文) | 类型 | 主键 | 外键 | 空值 | 属性(中文) |
| id | int | Yes |  |  | 记录ID |
| name | varchar(128) | No | No | No | 用户姓名 |
| password | varchar(256) |  |  |  | 用户密码 |
| email | varchar(254) |  |  |  | 邮箱 |
| c\_time  city | varchar(datetime)  varchar(32) |  |  |  | 注册日期  所在城市 |

Table 3.2(1) login\_user

（2）管理员用户基本信息表(auth\_user)

用来存储管理员用户信息的基本情况，如表3.2所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性(英文) | 类型 | 主键 | 外键 | 空值 | 属性(中文) |
| id | int | Yes |  |  | 记录ID |
| username | varchar(150) |  |  |  | 用户姓名 |
| first\_name | varchar(30) |  |  |  | 用户姓氏 |
| last\_name | varchar(150) |  |  |  | 用户名字 |
| password | varchar(256) |  |  |  | 用户密码 |
| email | varchar(254) |  |  |  | 邮箱 |
| date\_joined | datetime |  |  |  | 加入日期 |
| last\_login  is\_superuser  is\_staff | datetime  bool  bool |  |  |  | 上次登陆时间  是否是超级管理员  是否是员工 |

Table 3.2(2) auth\_user

# 4 接口及过程设计

## 4.1 数据处理服务器接口

在网页后端与数据处理后端进行交互时，采用的是websocket长连接，网页后端首先发送验证信息与账户的用户名给数据处理后端，以便于在数据处理服务器中并发时对每个线程的管理。连接成功之后，用户可以发送请求信息，即一个带有城市首字母大写的拼音的字符串，例：“BeiJing”；数据处理服务接收到字符串之后，将相应城市的json数据传输给网页服务端。

从数据处理后端回传给网页后端的json数据格式如下：



图 4.1 数据处理后端返回的json格式数据信息

# 5 界面设计

## 5.1登陆界面

主要分为一个登陆标题，三个提示标签，三个输入框，输入用户名、密码与验证码，下方有“新用户注册”与“登陆”两个按钮，输入正确的用户名、密码与验证码后，点击登陆即可进入主界面；若未进行注册，则返回提示信息。



图 5.1 普通用户登陆界面

## 5.2 注册界面

注册界面主要信息有5项：用户名、密码、确认密码、邮箱与城市。界面设计元素类似于登陆界面，设计原则简洁为主，能够使用户更加直观地完成注册的任务。



图 5.2 普通用户注册界面

## 5.3 主功能界面

主功能界面主要采用Django的python web进行网页开发，采用了MTV的框架模式（模型M，视图V和模版T），基于Echarts的图表绘制以及中国地图实现，JavaScript配合Django进行HTML设计，Django重量级自带数据库进行信息保存，显示预测图表信息。

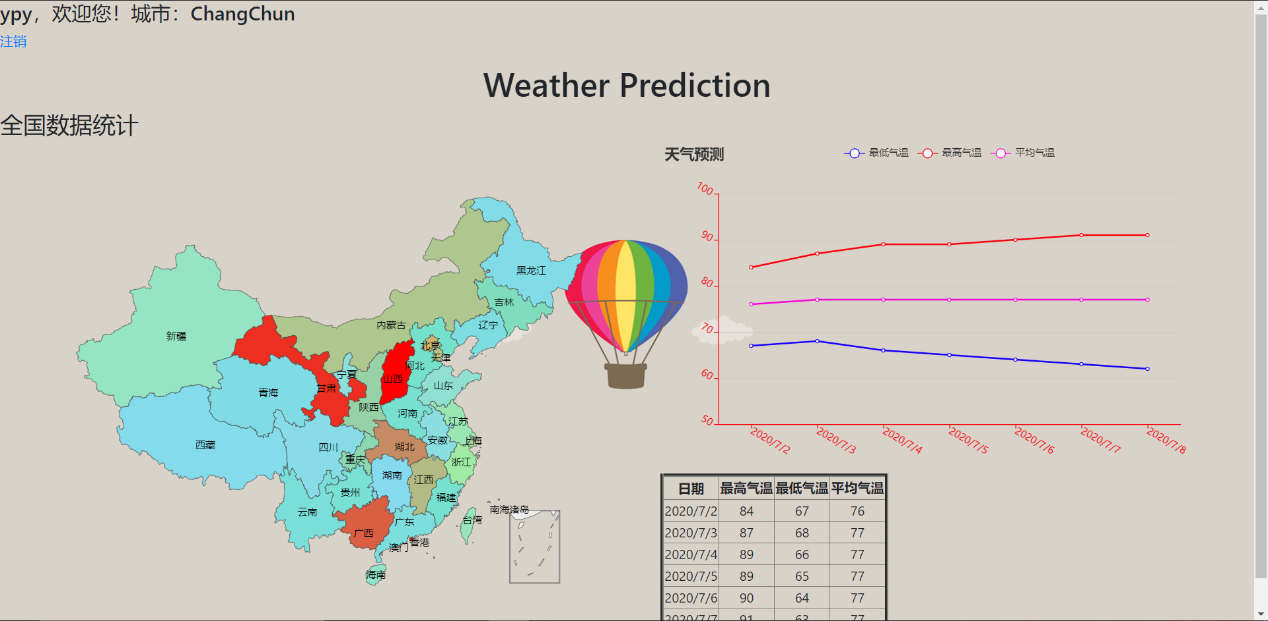


图 5.3.1 主界面设计

用户可以点击地图上的省级行政区，显示其省会未来七天的天气情况（如图4-10）。

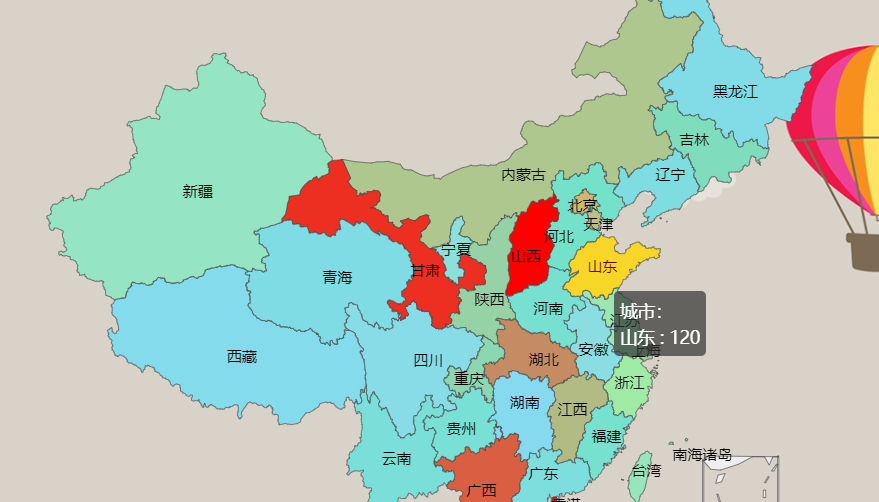


图 5.3.2 点击“山东”

用户在地图上悬停鼠标，通过滚轮上下滚动来放缩地图比例。

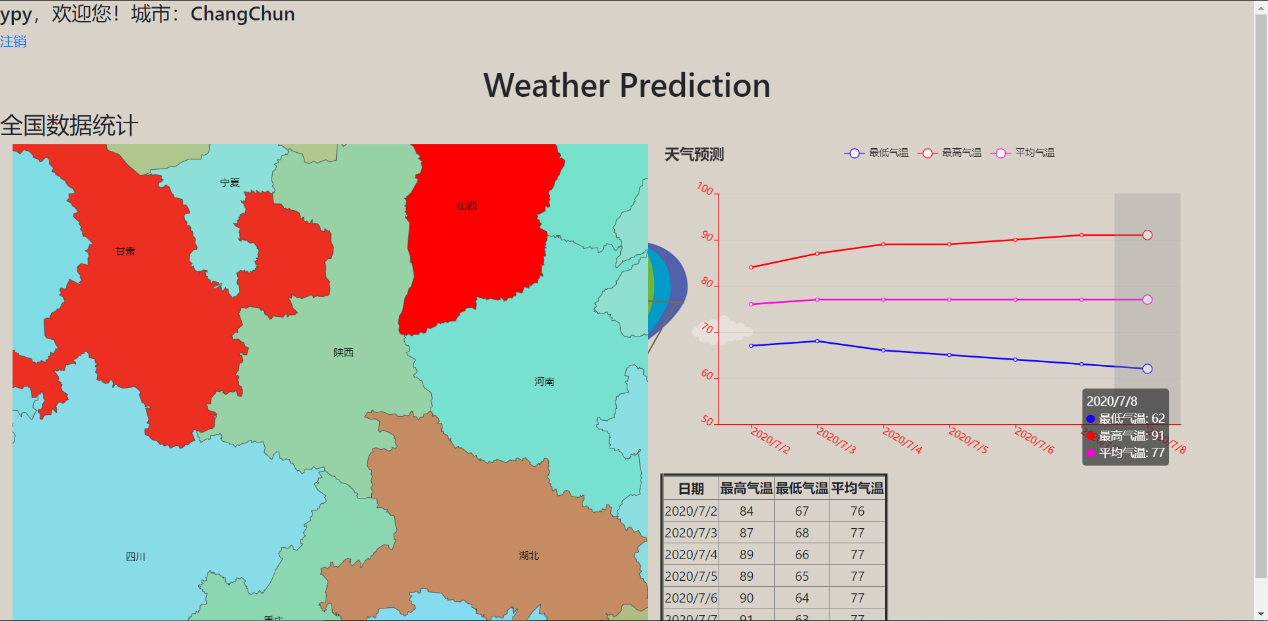


图5.3.3 缩放地图

用户可以点击折线图上的三个可以点击的按钮，来隐藏/显示折线。鼠标悬停在某一天上时，可以显示当天的天气情况。

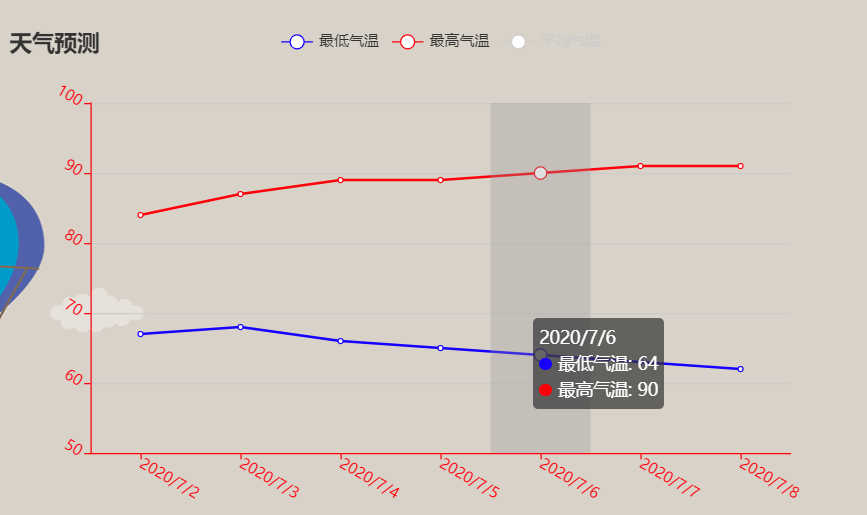


图 5.3.4 曲线设计

# 6 小 结

该文档分别从系统设计、数据库设计、接口以及过程设计、界面设计四大模块分析。系统设计主要是本气象数据分析系统的整体架构设计图，以及整体业务功能的设计，并且在用户管理层面上，不同用户所担当的角色不同，所对应的功能也不同。数据库主要是用于储存各用户信息的数据，主要表单有普通用户与管理员用户两种。接口设计主要表述了在网页服务端与数据服务端使用websocket交互时相互传输的协议以及相应格式。界面设计主要从用户登陆、注册界面与主功能界面进行展示，并且列出了使用的技术与优势。