## 平台整体框架设计

本文的气候可视化平台采用了B/S结构，B/S结构采取浏览器请求，服务器响应的工作模式。用户可以通过浏览器去访问Internet上由Web服务器产生的文本、数据、图片、动画、视频点播和声音等信息，而每一个Web服务器又可以通过各种方式与数据库服务器连接，大量的数据实际存放在数据库服务器中。B/S结构统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用

平台的前端使用React.js和Ant Design进行快速开发。系统后端采用Python与Flask框架快速开发，数据存储选择MySQL。平台整体框架设计如图4.1所示。

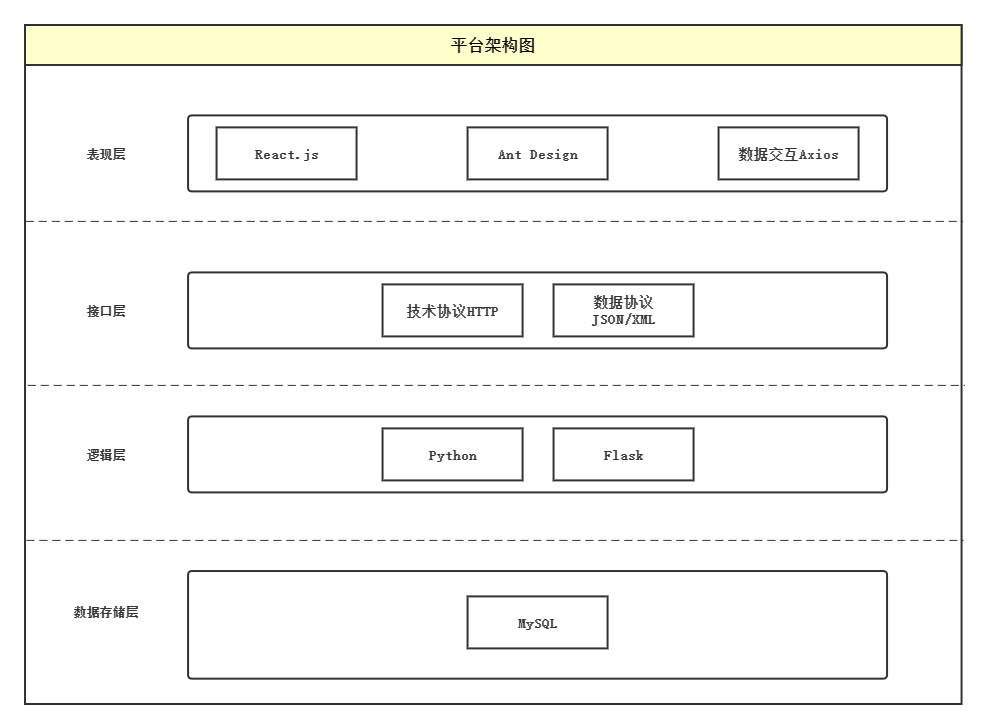


图 4.1 平台框架图

## 平台功能模块设计

本系统分为三大模块，分别是天气播报、气候可视化、历史天气。天气播报模块包含了天气即时显示、生活指数、气温预测等功能，气候可视化模块包括地图可视化、天气数据可视化、分钟级降水可视化、空气质量可视化、预警信息可视化等功能，历史天气包括历史年月气温图、下载天气历史数据等功能。整体功能结构图如图4.2。

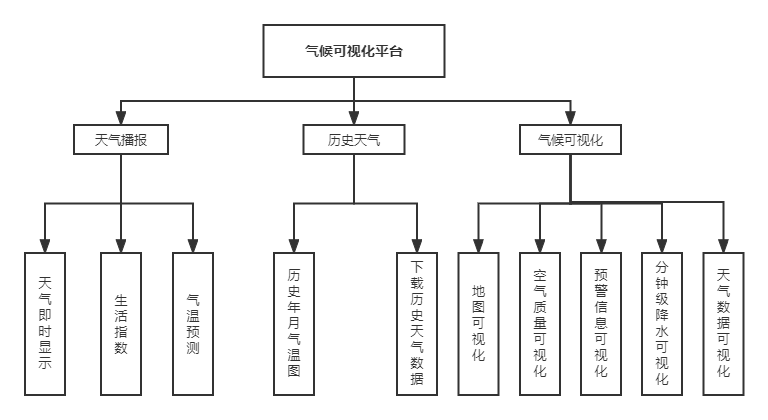
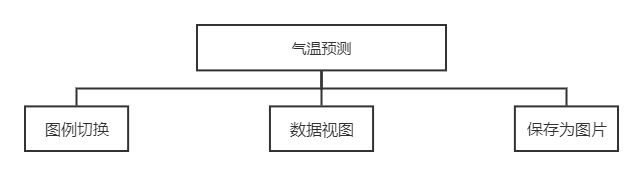


图 4.2 系统详细功能图

1. 天气播报

天气播报主要用途是显示即时天气信息、生活指数、气温预测，用户可以参考天气播报的功能与建议决定一天的出行情况。

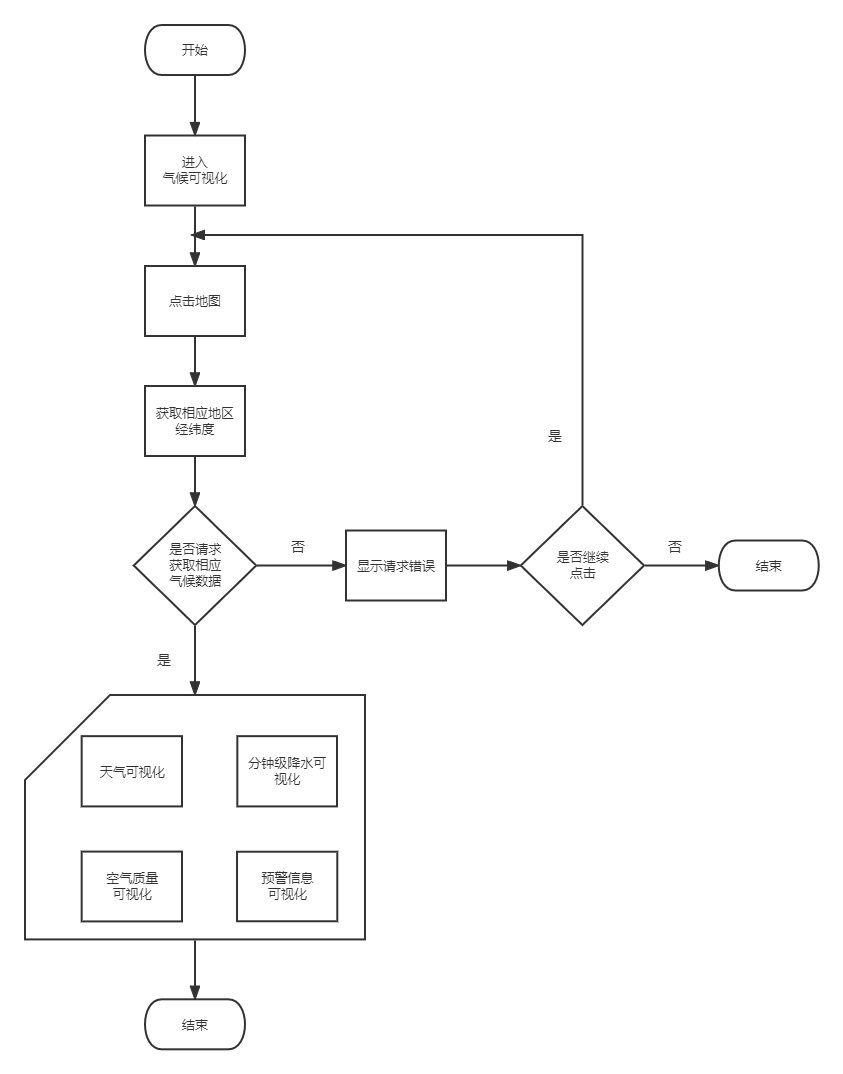
气温预测功能采用图表的形式进行展示，图表提供图例的切换，例如折线图、柱状图，数据视图，保存为图片等功能。气温预测功能结构如图4.3所示。



**图 4.3 气温预测功能结构图**

1. 气候可视化

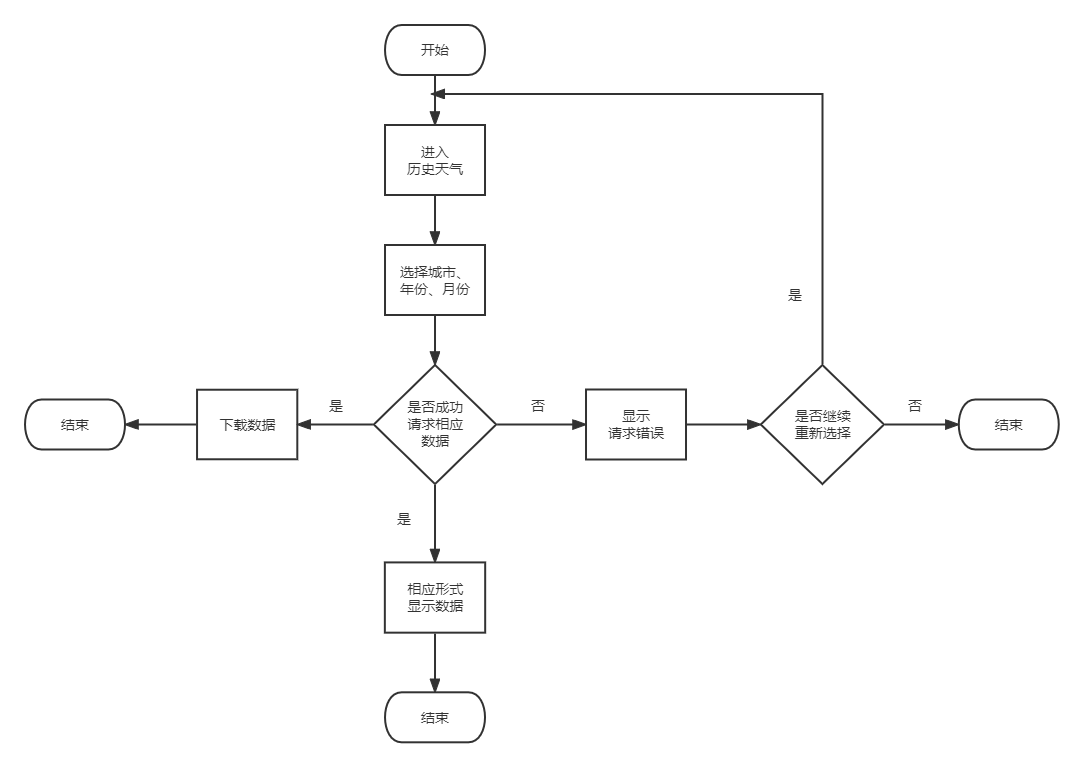
气候可视化主要用途是将相关的气候数据使用图表、地图的形式将数据进行可视化显示，利用图表的形式使得数据生动而形象。气候可视化功能流程图如图4.4所示。



**图 4.4 气候可视化功能流程图**

1. 历史天气

历史天气主要功能是用户可以根据特定的城市、年份、月份来查询相应的历史天气数据，提供文件下载功能，显示相应城市历史年月的平均最高温度和最低温度。历史天气功能流程图如图4.5所示。



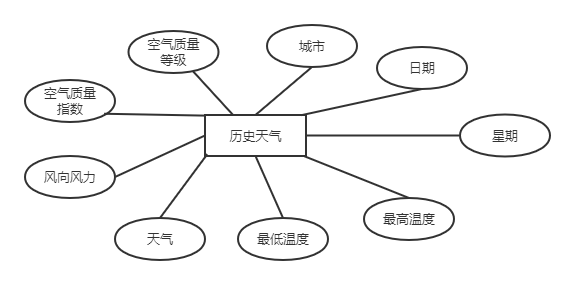
**图 4.5 历史天气功能流程图**

## 数据库设计

无论是应用程序，还是数据库如何变化，数据始终是最重要的部分。数据库是现代各种计算机应用系统的核心，数据库设计是数据库应用系统设计与开发的关键性作用[22]。良好的数据库设计，不仅可以节省数据的存储空间，能够保证数据的完整性，方便进行数据库应用系统的开发，同时拥有健壮、可靠的数据也有助于开发和测试等。

本平台数据库设计主要是为历史天气功能提供服务，通过使用数据库可以节约存储空间、方便快捷查询历史天气信息。本平台数据库选取MySQL数据库，支持多线程，充分利用CPU资源，提供用于管理、检查、优化数据库操作的管理工具。

本系统需要存储在数据库中的数据设计如表4.1所示，历史天气实体图如图4.6所示。



**图 4.6 历史天气实体图**

**表 4.1 历史天气数表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 允许空值 | 主键 | 说明 |
| 1 | city | varchar | 255 | Y | N | 城市 |
| 2 | date | varchar | 255 | Y | N | 日期 |
| 3 | week | varchar | 255 | Y | N | 星期 |
| 4 | max\_temp | varchar | 255 | Y | N | 最高温度 |
| 5 | min\_temp | varchar | 255 | Y | N | 最低温度 |
| 6 | weather | varchar | 255 | Y | N | 天气 |
| 7 | wind\_power | varchar | 255 | Y | N | 风向风力 |
| 8 | air\_quality\_index | varchar | 255 | Y | N | 空气质量指数 |
| 9 | air\_quality\_class | varchar | 255 | Y | N | 空气质量等级 |

天气播报时序图如图6.4所示。

