《农产品大数据分析系统》

项目概要设计

**V1.0**

**版 本 历 史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 作者 | 日期 | 备注 |
| 1.0 | 陈嘉、冯文豪、吴明羽、朱明航 | 2020-07-27 | 创建 |
|  |  |  |  |

目录

[《农产品大数据分析系统》 1](#_Toc47295136)

[项目概要设计 1](#_Toc47295137)

[第一部分 引言 4](#_Toc47295138)

[一、编写目的 4](#_Toc47295139)

[二、读者对象 4](#_Toc47295140)

[三、术语与缩写解释 4](#_Toc47295141)

[1、VUE 4](#_Toc47295142)

[2、SpringBoot 4](#_Toc47295143)

[3、大数据 5](#_Toc47295144)

[四、参考资料 5](#_Toc47295145)

[第二部分 项目概述 5](#_Toc47295146)

[一、项目描述 5](#_Toc47295147)

[二、项目功能描述 6](#_Toc47295148)

[1、价格查询 6](#_Toc47295149)

[2、数据监控 7](#_Toc47295150)

[第三部分 设计约束 7](#_Toc47295151)

[一、需求约束 7](#_Toc47295152)

[1、本系统应当遵循的技术标准 7](#_Toc47295153)

[2、软、硬件环境标准 7](#_Toc47295154)

[3、接口/协议标准 7](#_Toc47295155)

[4、用户界面标准 7](#_Toc47295156)

[5、软件质量 7](#_Toc47295157)

[二、隐含约束 10](#_Toc47295158)

[第四部分 农产品大数据分析系统方案设计 10](#_Toc47295159)

[一、农产品大数据分析系统相关业务流程 10](#_Toc47295160)

[1、价格查询及走势预测流程 10](#_Toc47295161)

[2、价格对比流程 11](#_Toc47295162)

[3、区域行情流程 12](#_Toc47295163)

[4、数据监控流程 13](#_Toc47295164)

[5、抓取量查询流程 14](#_Toc47295165)

[二、业务功能概要结构 14](#_Toc47295166)

[1、价格查询模块 15](#_Toc47295167)

[2、数据监控模块 17](#_Toc47295168)

[四、模块定义 17](#_Toc47295169)

[第五部分 E-R实体设计 18](#_Toc47295170)

[一、E-R实体结构图 18](#_Toc47295171)

[二、实体描述 19](#_Toc47295172)

[1、Result实体描述 19](#_Toc47295173)

[2、Market实体描述 19](#_Toc47295174)

[3、Varity实体描述 19](#_Toc47295175)

[4、Price实体描述 19](#_Toc47295176)

[5、Infocount实体描述 20](#_Toc47295177)

[第六部分 总体设计 20](#_Toc47295178)

[一、农产品大数据分析平台逻辑架构设计 20](#_Toc47295179)

[1、成熟的SpringBoot框架 21](#_Toc47295180)

[2、VUE前端设计框架 21](#_Toc47295181)

[3、专业的B&C/S结构 22](#_Toc47295182)

[4、NGNIX服务 22](#_Toc47295183)

[5、WEB网页形式 22](#_Toc47295184)

[二.物理架构设计 22](#_Toc47295185)

[三.技术架构设计 22](#_Toc47295186)

[1、Springboot模型设计 22](#_Toc47295187)

[2、MVVM框架 23](#_Toc47295188)

[3、Axios技术 24](#_Toc47295189)

[第七部分 用户界面设计 24](#_Toc47295190)

[一、主页面布局设计 24](#_Toc47295191)

[二、业务界面风格展示 25](#_Toc47295192)

[一、运行环境 26](#_Toc47295193)

[1、服务器环境 26](#_Toc47295194)

[2、客户机器环境 27](#_Toc47295195)

[3、开发环境要求 27](#_Toc47295196)

[二、系统性能要求 27](#_Toc47295197)

第一部分 引言

## 一、编写目的

编写本文的主要目的是把需求分析得到的用例模型转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。

本设计是指导详细设计和项目实施的重要指导性文件，也是进行系统集成测试和重要依据。

## 二、读者对象

该文档的读者为用户代表、软件分析人员、开发管理人员和测试人员。

## 三、术语与缩写解释

### 1、VUE

Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。

Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件 。

Vue.js 自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习，非常容易与其它库或已有项目整合。另一方面，在与相关工具和支持库一起使用时   ，Vue.js 也能完美地驱动复杂的单页应用。

### 2、SpringBoot

SpringBoot是由Pivotal团队在2013年开始研发、2014年4月发布第一个版本的全新开源的轻量级框架。它基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

### 3、大数据

大数据（Big data）通常用来形容一个公司创造的大量非结构化数据和半结构化数据，这些数据在下载到关系型数据库用于分析时会花费过多时间和金钱。大数据分析常和云计算联系到一起，因为实时的大型数据集分析需要像MapReduce一样的框架来向数十、数百或甚至数千的电脑分配工作。大数据需要特殊的技术，以有效地处理大量的容忍经过时间内的数据。适用于大数据的技术，包括大规模并行处理（MPP）数据库、数据挖掘、分布式文件系统、分布式数据库、云计算平台、互联网和可扩展的存储系统。大数据概念应用到IT操作工具产生的数据中，大数据可以使IT管理软件供应商解决大广泛的业务决策。IT系统、应用和技术基础设施每天每秒都在产生数据。大数据非结构化或者结构数据都代表了“所有用户的行为、服务级别、安全、风险、欺诈行为等更多操作”的绝对记录。

## 四、参考资料

《系统需求分析》

第二部分 项目概述

## 一、项目描述

农产品批发市场是我国大量农产品生产、销售和流通环节的非常重要的一环，同时也是农产品流通、价格形成、物流包装以及商品信息流动等环节的非常重要的载体。经过近40年的发展，农产品批发市场已经成为我国批发市场的重要组成部分，为我国大规模农产品销售提供了必不可少的销售平台。但是，随着经济社会的飞速发展，我国的农产品批发市场已进入发展新常态。为了不断推进农业经济的优化，实现可持续的产业发展和区域产业结构优化，进一步推动智慧农业的建设进程，需要全面及时掌握农业的发展动态，这需要依托农业大数据及相关大数据分析处理技术，建设一个农业大数据分析应用平台——农产品市场数据分析系统来支撑。

## 二、项目功能描述

用户业务主要包含价格查询和数据监控。

价格查询包含四个主要查询内容：区域行情、价格对比、价格走势、价格预测。

数据监控包含两个主要内容：数据更新时间和处理数据量。

### 1、价格查询

价格查询的各个内容介绍如下：

不同地区的市场情况不尽相同，为了直观对比不同地区农产品批发市场行情，将农产品价格显示到各地区，以地图的形式展示价格，便于用户对区域行情进行判断和分析。

由于不同地区或者不同时间，相同的农产品的价格往往存在差异性，针对不同地区的用户，其所关注的一般是本地的农产品价格，所以在使用过程，能先通过限制地区的方式初步筛选出所需要查看农产品的在当地的价格波动情况，然后再根据时间或者走势进行判断分析选择合适的价位。

除了不同产品之间的价格差别外，相同产品在不同时段和不同的市场，甚至相同的市场的价格都会有价格差异，因此我们通过表格的形式或柱状图等可视化的形式将各个价格进行展示，直观的反应出各个价格之间的差异进行对比。

为了给平台的用户提供更加具有价值的信息，我们根据农产品价格之前的波动情况和可能影响价格的因素进行分析和预测，最终为用户提供一个比较具有参考价值的农产品价格预测结果。

### 2、数据监控

为更好的向用户展示系统的数据规模，以及保持数据的实时性，系统每天定时爬取农产品数据并统计农产品市场总数、农产品品类总数、农产品品种总数以及数据量总数，并实时更新。

考虑到用户对农产品的具体价格数据只关心最近几天的，所以系统只对统计最近5天各个省份各个市场各个品类的抓取数据，若不够5天的，有几天数据展示几天。

第三部分 设计约束

## 一、需求约束

### 1、本系统应当遵循的技术标准

数据命名的规则遵循驼峰规则；

### 2、软、硬件环境标准

本系统采用B/S架构。java编写，数据库采用mysql。系统部署在centos8版本以上的设备里。

### 3、接口/协议标准

本系统通过SOAP协议实现Web终端和服务器之间的数据通信。

### 4、用户界面标准

使用VUE标准UI库，并进行界面全部美工优化。

### 5、软件质量

1）正确性

系统必须交易能够被正确处理；

2）健壮性

系统应能够7\*24小时无故障运行；

3）效率性

系统可以支持100个终端同时发起业务，处理业务的时间不超过10秒钟；

4）易用性

界面应采用图形化操作方式，便于业务人员操作；

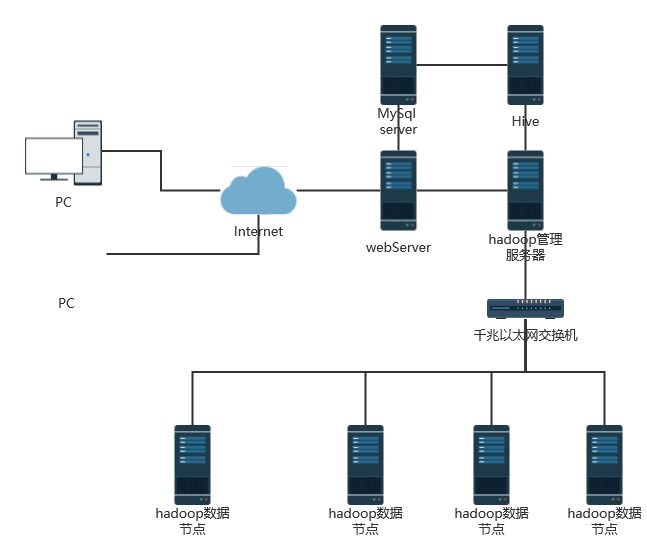
5）安全性

报文中的关键数据域以密文的方式传输；

6）可扩展性

应该充分考虑到将来交易的修改或增加，避免需求变更时大规模修改程序。

7）网络体系结构



A）PC

代表即各种终端设备。

B）Internet

即网络浏览器。

C）webserver

web服务器，互联网接入平台，通过开放对外的Web Service端口，用来访问网站。

D）Mysql Server

数据库管理服务器，用来存储所需的数据。

E）hadoop管理服务器

大数据主节点，用来控制各个数据节点。

F）hadoop数据节点

用来存储爬取的大数据。

G）Hive

用来分析，清洗hdfs种的数据，并将其数据通过sqoop传输给mysql服务器。

## 二、隐含约束

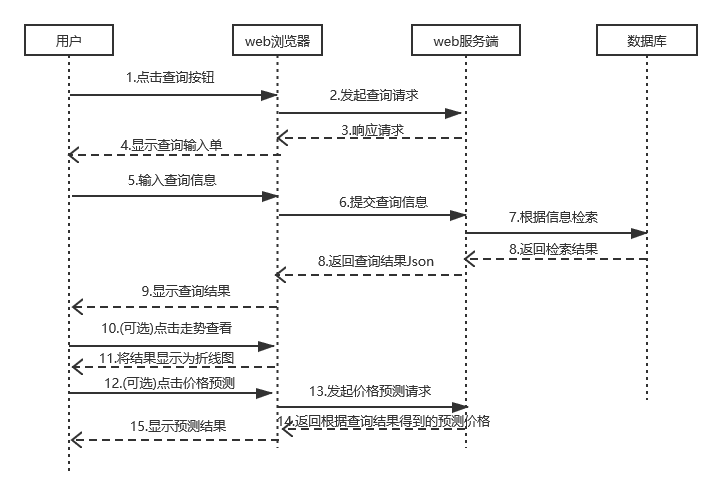
1）用户具有基本的业务技能和基本的电脑知识，对我们提供的操作界面应保证他们经过简单培训后无障碍的操作；

2）应该把有可能变动的参数存放到配置文件或数据库中，保证修改参数的灵活性。

第四部分 农产品大数据分析系统方案设计

## 一、农产品大数据分析系统相关业务流程

### 1、价格查询及走势预测流程

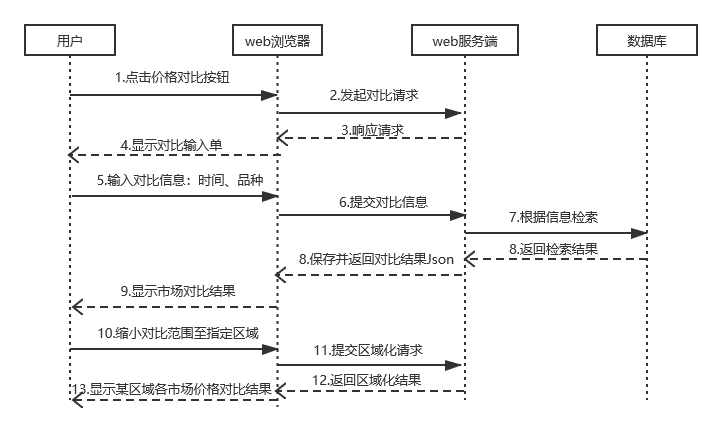


**说明：**

价格查询主要流程包括：

* + 用户通过web浏览器向服务器发起查询请求
  + 服务器返回一个查询输入表单
  + 用户通过浏览器提交查询信息包括区域、市场、品类、品种
  + 服务端在Mysql数据库进行相应的查询，得到查询结果
  + 服务端将查询结果通过json的格式返回给浏览器
  + 浏览器显示查询结果给用户
  + 用户点击走势查看，浏览器将查询结果通过时间显示为折线图
  + 用户发起价格预测请求，服务端会根据查询的价格生成一个预测价格并返回

### 2、价格对比流程

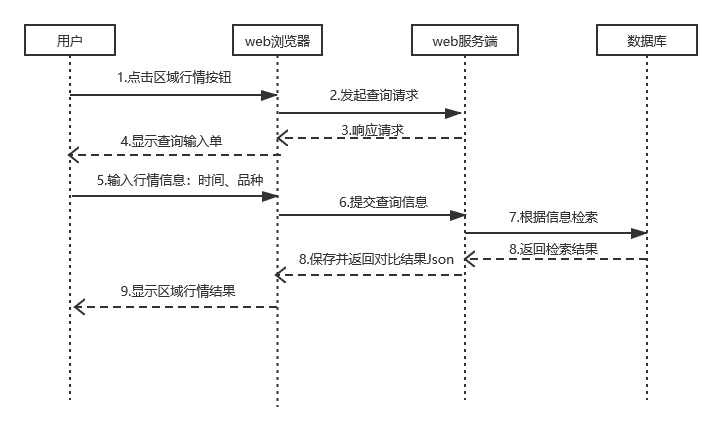


**说明：**

价格对比主要业务流程包括：

* + 用户通过浏览器向服务端发起价格对比请求
  + 服务器返回一个价格对比输入表单
  + 用户通过浏览器提交对比信息如时间、品种给服务器
  + 服务端在Mysql数据库进行对比信息检索，并得到检索结果
  + 服务端返回Json格式的返回结果给浏览器
  + 浏览器显示对比结果
  + 用户可以选择将对比结果区域化，即查看某区域各市场的价格对比
  + 服务端返回区域化结果，浏览器进行显示

### 3、区域行情流程

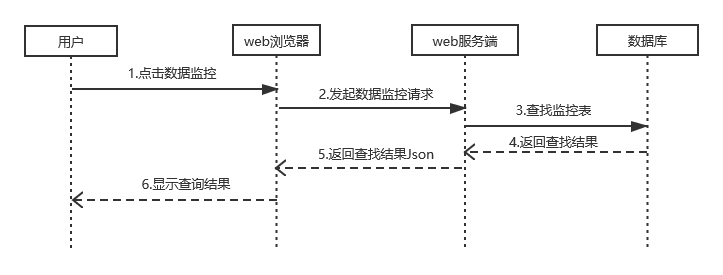


**说明：**

区域行情主要流程包括。

* + 用户通过浏览器向服务端发起区域行情查询请求
  + 服务器返回一个查询输入表单
  + 用户通过浏览器提交对比信息如时间、品种给服务器
  + 服务端在Mysql数据库进行对比信息检索，并得到检索结果
  + 服务端返回Json格式的返回结果给浏览器
  + 浏览器显示区域行情结果，通过地图显示

### 4、数据监控流程

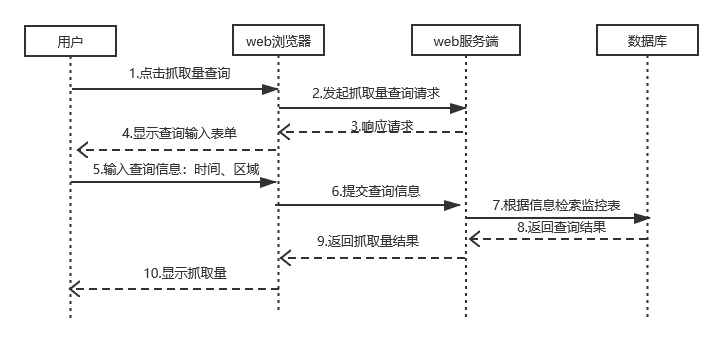


**说明：**

数据监控模块主要是对市场总数、品类总数、品种总数等数据规模进行监控，其主要流程如下：

* + 用户通过web浏览器发起数据监控查询请求
  + 服务端查找数据库的数据监控信息
  + 服务器返回json格式数据，浏览器进行显示

### 5、抓取量查询流程



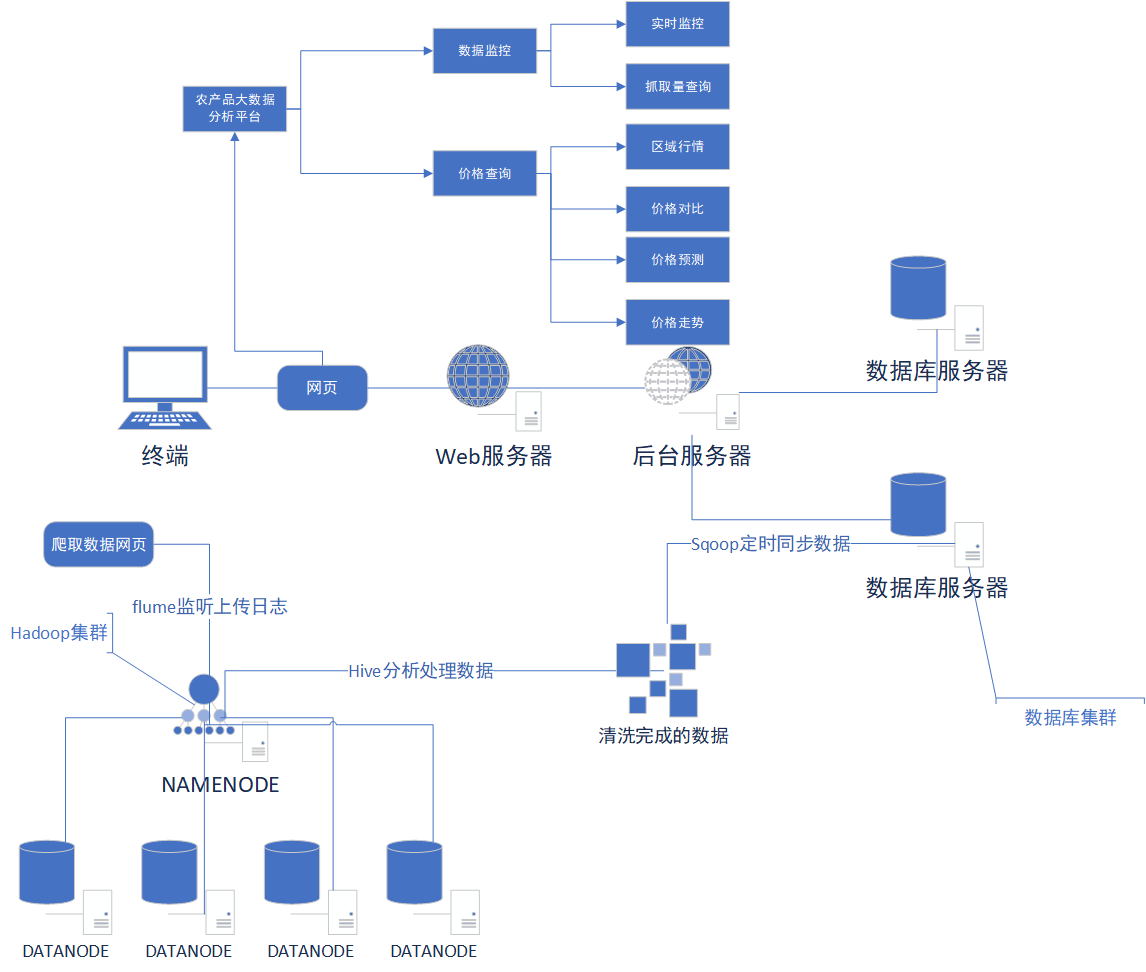
**说明：**

抓取量查询是对某一区域的数据规模进行查询，主要流程如下：

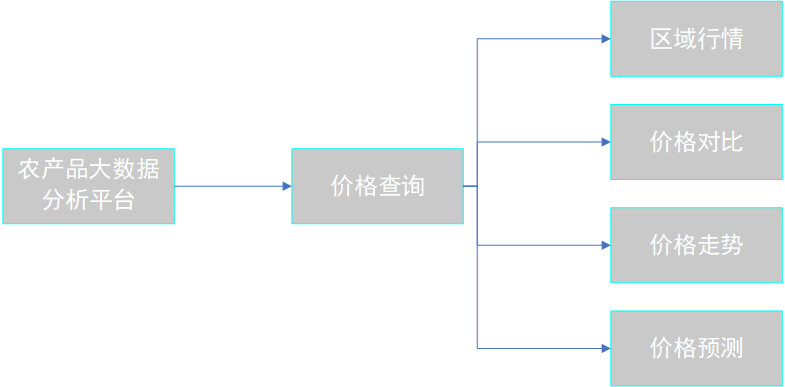
* + 用户通过浏览器发起查询请求
  + 服务端响应请求，浏览器显示输入表单
  + 用户输入查询信息，服务器根据信息检索数据库对应监控表
  + 服务器返回结果，浏览器进行显示

## 二、业务功能概要结构

逻辑结构图如下：



### 1、价格查询模块



#### 1.1、区域行情

1、系统界面开始初始化。

2、在系统初始化的时候，首先判定是否设置区域参数，如果设置，则直接获取所选区域的相关信息，如农产品种类，批发市场，价格等；如果没有设置，则使用默认的区域的相关信息进行显示。

#### 1.2、价格走势

1、系统界面初始化完。

2、用户在农产品种类和区域设置参数完毕，点击确认。

3、后台向数据库搜索相关信息。

4、后台将得到的价格，时间等信息包装成一个json返回数据返回前端。

5、前端将得到的Json数据解析并显示在折线图中

#### 1.3、价格对比

1、用户选择一个农产品进行价格查询完毕。

2、在已加载完成的该界面，设置右侧需要对比的参数，点击确认。

3、前端向后台发送对比参数，由种类，时间，区域包装的实体信息。

4、后台向数据库搜索该实体对应的价格信息返回前端。

5、前端将得到的数据与已有的数据进行图表显示对比。

#### 1.4、价格预测

1、用户进行价格查询。

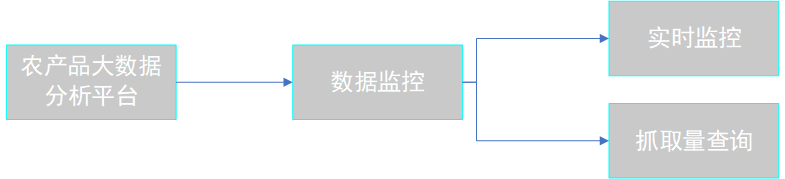
2、前台向后端发送查询实体，（种类，区域，批发市场，时间）。

3、服务器返回查询结果。

4、前端将返回结果显示为一个列表，并提供视图切换功能，并将得到的查询结果发送到后端的价格预测模块。

5、后端进行价格预测处理并返回预测结果，前端将得到的预测结果与之前查询得到的已有数据一起显示为一个折线图。

### 2、数据监控模块



#### 2.1、实时监控模块

1、进入数据监控界面，等待界面加载。

2、加载时前端向后端发送数据查询请求。

3、后端向数据库查询实时监控表。

4、后端返回查询结果。

5、前端将所得到的市场总数，品类总数，品种总数，数据总数显示。

#### 2.2、抓取量查询模块

1、数据监控界面加载完成。

2、用户输入所要查询的时间和区域。

3、系统根据查询参数，在数据监控表中查询抓取量并返回前端。

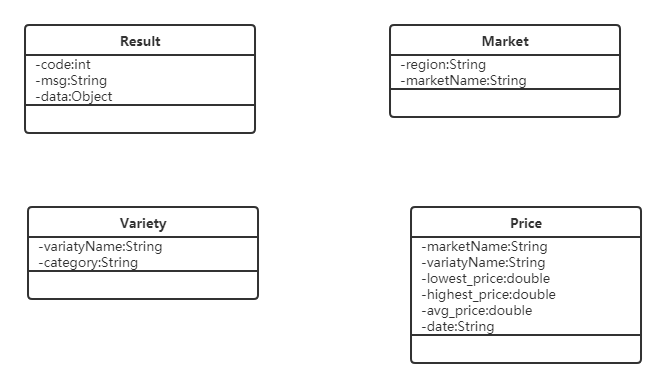
4、前端显示抓取量。

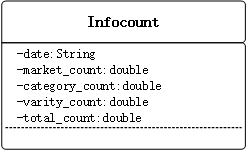
## 四、模块定义

|  |  |  |
| --- | --- | --- |
| 模块 | 组建 | 规格/型号 |
| 价格查询 | 区域行情 | 显示一个区域的总体价格情况 |
| 价格走势 | 显示农产品的价格变化图 |
| 价格对比 | 对比农产品在不同市场和时间以及与其他产品的价格 |
| 价格预测 | 根据模型预测农产品价格 |
| 数据监控 | 实时监控 | 显示数据的更新时间 |
| 抓取量查询 | 显示处理的数据条数 |

第五部分 E-R实体设计

## 一、E-R实体结构图





## 二、实体描述

### 1、Result实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | code | 状态码 | Int |
| 2 | msg | 返回信息 | String |
| 3 | data | 返回数据 | Object |

### 2、Market实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | region | 所属区域 | String |
| 2 | marketName | 市场名 | String |

### 3、Varity实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | category | 品类 | String |
| 2 | varityName | 品种名 | String |

### 4、Price实体描述

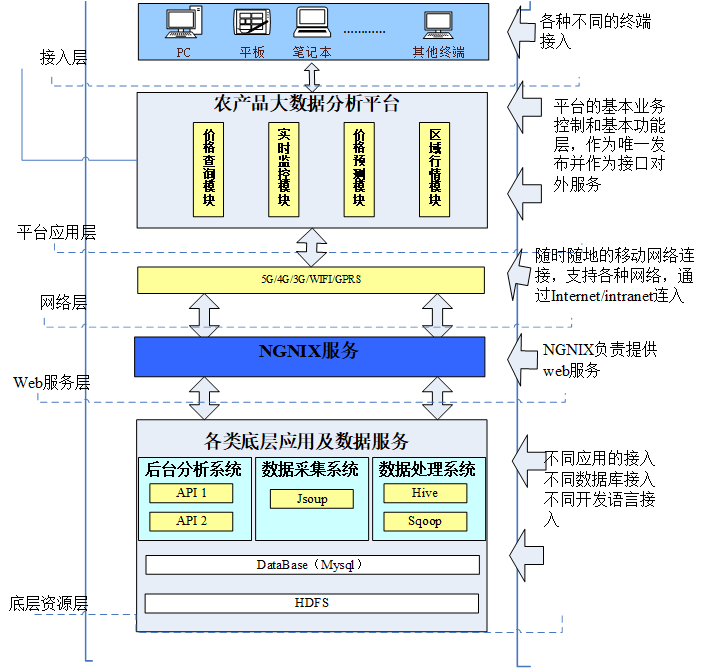
|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | marketName | 市场名 | String |
| 2 | varityName | 品种名 | String |
| 3 | lowest\_price | 最低价格 | double |
| 4 | highest\_price | 最高价格 | double |
| 5 | avg\_price | 平均价格 | double |
| 6 | date | 日期 | String |

### 5、Infocount实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | date | 日期 | String |
| 2 | market\_count | 市场总数 | double |
| 3 | category\_count | 品类总数 | double |
| 4 | varity\_count | 品种总数 | double |
| 5 | total\_count | 数据总数 | double |

第六部分 总体设计

## 一、农产品大数据分析平台逻辑架构设计



### 1、成熟的SpringBoot框架

从前后端分离的角度出发，采用成熟的springboot框架，采用特定的方式进行配置，使用”习惯优于配置“的理念，定义模板化的个性配置。提供自带的代码生成器以及接口生成文档，为前端提供标准的接口文档。

### 2、VUE前端设计框架

vue是渐进式JavaScript框架，“渐进式框架”和“自底向上增量开发的设计”是Vue开发的两个概念

vue可以在任意其他类型的项目中使用，使用成本较低，更灵活，主张较弱，在Vue的项目中也可以轻松融汇其他的技术来开发。

易用（使用成本低），灵活（生态系统完善，适用于任何规模的项目），高效（体积小，优化好，性能好）

### 3、专业的B&C/S结构

考虑到B/S在信息采集和信息展示方面的灵活性的特点，C/S在安全性、事务处理和交互性方面的特点。本工程采用B&C/S结构，充分满足客户的业务习惯和需求。

### 4、NGNIX服务

轻量级，同样起web 服务，比apache 占用更少的内存及资源，抗并发，nginx 处理请求是异步非阻塞的，而apache 则是阻塞型的，在高并发下nginx 保持低资源低消耗高性能，高度模块化的设计，编写模块相对简单，支持7层负载均衡

### 5、WEB网页形式

采用web网页接入的形式，使得终端不受限制，只要由浏览器和网络，任何的终端设备均可访问该农产品大数据分析平台

## 二.物理架构设计

1）服务端

系统服务端可以部署到云平台上，系统管理员通过IE实现业务管理操作；

2）具体应用

系统服务端可以部署到云平台上，作为实际数据的支持者和提供者，并接入Web终端的访问。

3）客户端

因为系统采用B&C/S模式，所以PC客户端只要求浏览器和网络支持；

4）智能终端

通过不同平台，不同语言开发的C/S模式系统，通过Webservice实现消息通信。

## 三.技术架构设计

### 1、Springboot模型设计

系统实现java的SpringBoot系统架构，使用java SDK1.8标准库，使用内嵌的tomcat构建服务器，使用Mybatis Plus对数据库进行操作，使用mysql数据库进行数据的持久化。

1）Controller层

Controller层负责与Web端进行交互，接收Web端发送过来的请求，调用Service层的方法处理请求，最后将后台处理的结果打包发送给Web端。

2）Service层

Service层负责服务数据的处理，实现请求的处理逻辑，调用Mapper层的方法对数据库进行数据的增删改查操作，并将结果返回给Controller层。

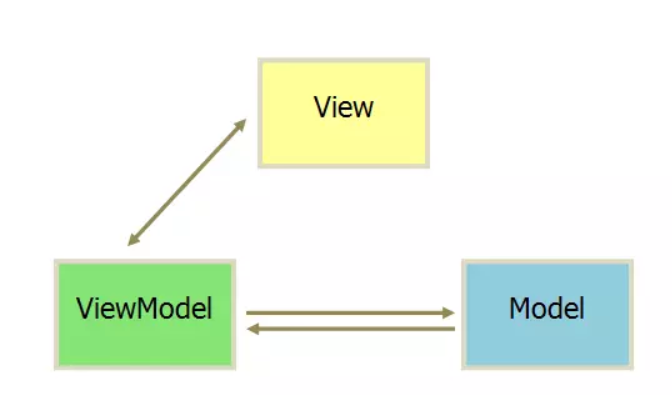
3）Entity层

Entity层负责数据的定义，Entity层中的类规定了数据库中对应的表的结构。

4）Mapper层

Mapper层负责实现与数据库进行交互的方法，对数据库进行操作，将结果返回给Service层。

### 2、MVVM框架



MVVM 由 Model,View,ViewModel 三部分构成，Model 层代表数据模型，也可以在Model中定义数据修改和操作的业务逻辑；View 代表UI 组件，它负责将数据模型转化成UI 展现出来，ViewModel 是一个同步View 和 Model的对象。

在MVVM架构下，View 和 Model 之间并没有直接的联系，而是通过ViewModel进行交互，Model 和 ViewModel 之间的交互是双向的， 因此View 数据的变化会同步到Model中，而Model 数据的变化也会立即反应到View 上。

ViewModel 通过双向数据绑定把 View 层和 Model 层连接了起来，而View 和 Model 之间的同步工作完全是自动的，无需人为干涉，因此开发者只需关注业务逻辑，不需要手动操作DOM, 不需要关注数据状态的同步问题，复杂的数据状态维护完全由 MVVM 来统一管理。

### 3、Axios技术

axios主要是用于向后台发起请求的，还有在请求中做更多是可控功能。

axios 特点：

* 从浏览器中创建 XMLHttpRequests
* 从 node.js 创建 http请求
* 支持Promise API
* 拦截请求和响应 （就是有interceptor）
* 转换请求数据和响应数据
* 取消请求
* 自动转换 JSON 数据
* 客户端支持防御

第七部分 用户界面设计

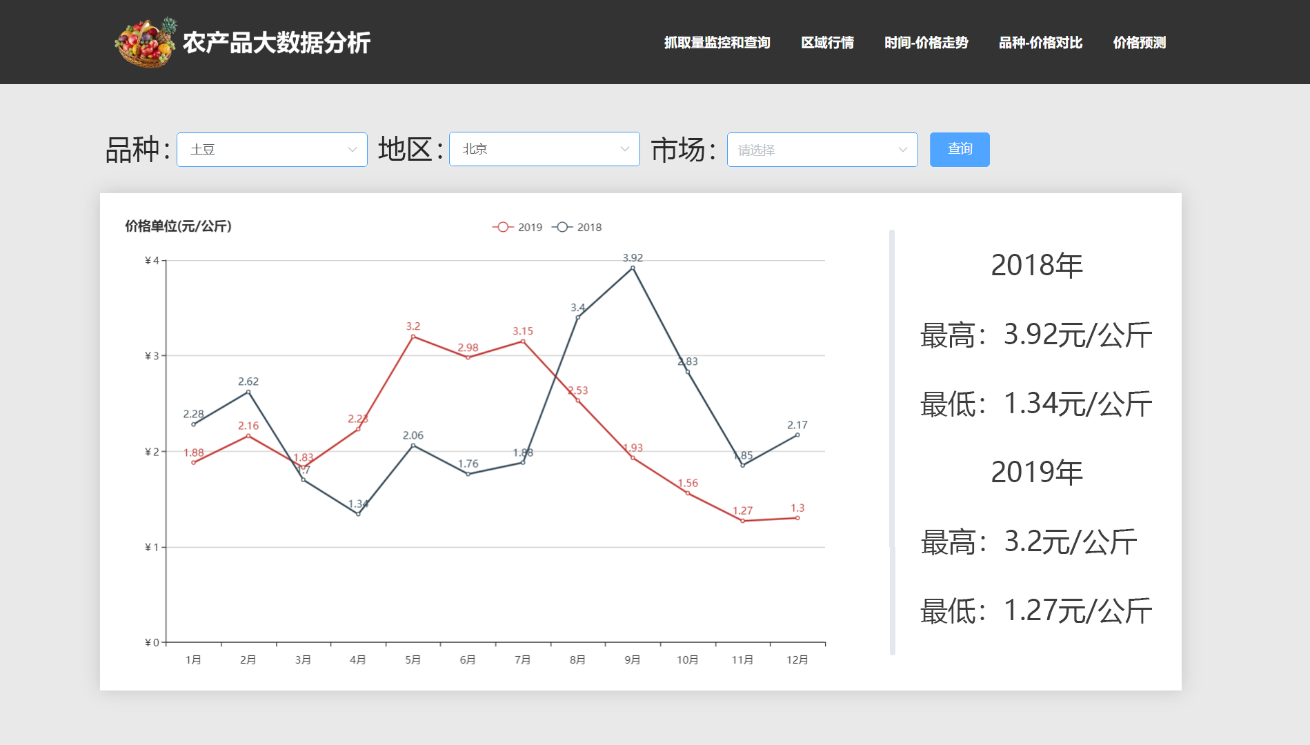
## 一、主页面布局设计



1. 主页包含了我们大数据分析系统的主要功能的链接
2. 导航栏中链接可以直接跳转至相应功能页面
3. 部分功能以一个简单的窗口展示在主页

## 二、业务界面风格展示





统一风格采用顶部的导航条操作风格。理论上大体分成四区：按钮导航区、条件查询输入区、详细信息显示区、信息列表摘要显示区。

根据业务不同，每个区域根据需要可以不要。

**第八部分 运行环境和部署**

## 一、运行环境

### 1、服务器环境

1）OS：Centos 8

2）8GB以上内存，建议使用16GB内存；

3）300G以上的硬盘空间，不包括数据存储空间

4）JDK：jdk1.8.2

5）DB：MySql 8.0.19

6）Application Server：Niginx 1.16.1

7）Hadoop: 3.2.1

8）Hive: 3.1.2

9）Flume: 1.9.0

10）Scala: 2.12.8

11）服务器数量：7台 （包括1台前端服务器，2个后端节点，3台大数据节点服务器，1台数据库节点）

### 2、客户机器环境

1）512MB以上内存，建议使用1024MB内存；

2）WEB：要求IE7以上版本，最好IE8以上版本。

### 3、开发环境要求

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 版本 |
| 开发平台 | Windows | 10 |
| 开发工具 | IDEA | IntelliJ IDEA 2019.2.1 x64 |
| 代码管理工具 | git | 2.27.0 |
| 开发环境 | JAVA SDK | 1.8 |
| SCALA\_SDK | 2.12.8 |

## 二、系统性能要求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| # | 项目 | 模块 | 级别 | 技术参数 |
| 1 | 设计实现技术指标 | 系统架构 | A | 采用C-B/S模式三层架构。 |
| 2 | 面向对象开发语言与框架 | A | 采用java开发语言和mysql等技术。 |
| 3 | 注释和文档 | A | 符合CMMI软件开发过程标准文档（至少提供：需求、概要、详细设计、测试报告、部署和环境、用户手册），代码注释量>=30%。 |
| 4 | 模块化和适合实训 | A | SOA设计、模块化，保证系统各模块单元较强的独立性适合实训教学。 |
| 5 | 测试覆盖率 | A | 功能覆盖率>=100%，业务覆盖率>=100%，语言覆盖率>=100%，逻辑覆盖率>=80%。 |
| 6 | 资源利用率要求 | CPU占用率 | B | <=50%利用率（附近标准配置） |
| 7 | 内存使用率 | B | <=75%利用率（附近标准配置） |
| 8 | 响应时间要求 | 服务器 | B | <=100ms（附近标准配置） |
| 9 | 网络 | B | <=100ms（附近标准配置） |
| 10 | 客户端 | B | <=5s（附近标准配置） |
| 11 | 系统稳定性要求 | 成熟性 | A | 真实的用户，成功使用本系统 |
| 12 | 稳定性 | B | 无故障运行时间>=365天，系统恢复时间<=2小时。 |
| 13 | 先进性 | A | 采用大数据分析。 |
| 14 | 典型意义 | A | 案例项目要有典型意义，有推广价值。 |
| 15 | Web服务接口要求 | Ngnix服务器端 | A | 消息延迟<=2S，丢包<=0.001%。 |
| 16 |
| 17 |
| 18 | 集成部署环境 | 服务器 | A | Centos 8 |
| 19 | 数据库 | A | Mysql 8.0 及以上 |
| 20 | 浏览器 | A | IE内核9以及以上 |
| 21 | PC电脑 | A | Microsoft Windows 7 及以上 |
| 说明：级别（A:表示非常重要必须达到的技术性能要求,B:表示重要推荐达到的技术性能要求,C：表示非重要可以弱化的技术性能要求.） | | | | |