

《农产品批发市场大数据分析系统》

项 目 概 要 设 计

v1.0.4

**版 本 历 史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 日期 | 备注 |
| 1.0.0 | 刘瑞淇 | 第九组所有成员 | 2020-7-27 | 创建并完成最初版 |
| 1.0.1 | 魏振 |  | 2020-7-29 | 修改E-R实体图 |
| 1.0.2 | 徐佳俊 |  | 2020-7-29 | 完善逻辑架构 |
| 1.0.3 | 钟辉 |  | 2020-7-29 | 修改页面风格价绍 |
| 1.0.4 | 戢启瑞 |  | 2020-8-1 | 明确性能要求 |

**目 录**

第一部分 引言1

一、编写目的1

二、读者对象1

三、术语与缩写解释1

1、HTTP1

2、JSON2

3、B/S架构2

4、RESTFUL风格2

5、Vue2

6、JSoup3

四、参考资料3

第二部分 项目概述4

一、项目描述4

二、项目功能描述4

1、管理员登陆4

2、数据维护5

3、数据检索5

4、用户注册5

5、用户登录6

6、数据查询6

7、请求监控数据6

8、查询预测价格6

第三部分 约束分析8

一、需求约束8

1、本系统应当遵循的技术标准8

2、软、硬件环境标准8

3、接口/协议标准9

4、用户界面标准9

5、软件质量9

二、隐含约束11

第四部分 平台方案设计12

一、安全设计12

二、平台相关业务流程12

1、数据爬取存储流程12

2、注册登录流程13

3、数据监控流程14

4、数据查询流程15

5、价格预测流程16

6、用户管理流程17

三、业务功能概要结构18

1、认证模块18

1.1、用户注册18

1.2、用户/管理员登陆19

1.3、用户管理19

2、数据获取模块20

2.1、数据爬取20

2.2、数据存储20

3、数据监控模块21

3.1、抓取量查询21

3.2、实时数据监控21

4、数据查询模块22

4.1、抓取量查询22

4.2、实时数据监控22

4.3、实时数据监控23

5、价格预测模块23

四、模块定义24

第五部分 E-R实体文档25

一、E-R实体结构图25

二、实体描述25

1、User实体描述25

2、Administrator实体描述25

3、Market实体描述26

4、Produce实体描述26

5、PriceInfo实体描述26

6、PriceTrend实体描述27

7、NationalPriceComparsion实体描述27

8、LocalPriceComparsion实体描述27

9、PriceForecastResult实体描述27

第六部分 总体设计28

一、逻辑架构28

1、J2EE四层架构29

2、B/S架构29

3、服务器集群29

二、物理架构29

1、服务端29

2、客户端29

三、技术架构30

1、RESTFUL接口30

2、Hadoop30

3、Vue前端框架31

4、Flume31

第七部分 总体设计32

一、桌面布局设计32

1、登录界面风格32

2、主界面风格33

二、业务界面风格展示33

1、风格展示34

第八部分 总体设计35

一、运行环境35

1、浏览器环境35

2、客户机器环境35

3、开发环境要求35

二、系统性能要求36

**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、HTTP

HyperText Transfer Protocol，超文本传输协议。是因特网上应用最为广泛的一种网络传输协议，所有的WWW文件都必须遵守这个标准。HTTP是一个基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）的。

2、JSON

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。并且我们也可以使用JSON数据格式来实现前后端分离。

3、B/S架构

B/S架构即浏览器和服务器架构模式，是随着Internet技术的兴起，对C/S架构的一种变化或者改进的架构。在这种架构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端(Server)实现，形成所谓三层3-tier结构。B/S架构是WEB兴起后的一种网络架构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，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。

4、RESTFUL风格

REST，即Representational State Transfer的缩写。直接翻译的意思是"表现层状态转化"。

它是一种互联网应用程序的API设计理念：URL定位资源，用HTTP动词GET、POST、PUT、DELETE等方式对于服务器的资源进行操作。

5、Vue

Vue 是一套用于构建用户界面的渐进式JavaScript框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，方便与第三方库或既有项目整合。

6、Jsoup

Jsoup是一款Java的HTML解析器，可以直接解析某个URL地址，可以通过DOM、CSS以及类似于jQuery的操作方法来取出数据。

四、参考资料

《系统需求分析》

**第二部分 项目概述**

一、项目描述

近几年随着电子商务的兴起，农产品的网上销售也模式也逐渐发展起来，各种涉农信息的发布也改变了传统模式，直接利用网络资源进行宣传与销售。而原有的大型果蔬农产品批发市场，也逐步改变原有的管理销售模式，采用新的电子商务销售系统，对农产品交易和农资信息发布进行有效的整合，为人们在线进行农产品交易提供虚拟网络环境和保障农产品电子商务顺利运营的管理环境。 在此，为实现一个能为各地果蔬交易商提供及时有效的农产品信息，为了能够以更快更便捷的方式向大家展示农产品的各类信息，在保证信息真实可靠的前提下，决定利用各地各大果蔬农产品批发市场的交易系统所提供的交易信息，开发一款农产品交易市场的大数据分析系统。

二、项目功能描述

项目功能主要分为管理员功能和普通用户功能，管理员功能包括管理员登陆、数据维护、数据检索，而普通用户功能包括用户注册、用户登录、数据查询、请求监控数据、查询预测价格。

**1、管理员登陆**

用户以管理员身份登入系统，拥有管理员功能。

登录所需提交的表单包括

（1）用户名

（2）密码

点击登陆后进入系统管理员主页，实现系统管理。

**2、数据维护**

数据维护是管理员向服务器发出请求，对数据库中的一些新的、过期或者有损数据进行新增、更新、删除。

请求表单中包括

（1）商品名

（2）地区

（3）地区批发市场

（4）时间范围

提交增删改查请求后系统返回增删改查结果并提示。

**3、数据检索**

数据检索是管理员向服务器提交查询请求，输入要查询的商品名以及查询的详细条件可以得到一个系统返回结果。

查询的数据所需的内容包括：

（1）商品名

（2）地区

（3）地区批发市场

（4）时间范围

点击提交后服务器返回所检索的商品的详细价格表格信息。

**4、用户注册**

用户注册是游客用户对网站申请账号的功能

需要提交的内容有：

（1）手机号或者邮箱号

（2）密码

（3）确认密码

注册成功则提示注册成功并返回登陆界面，注册失败则提示注册失败具体原因，比如手机号邮箱号被注册等等。

**5、用户登录**

用户以用户名以及密码登入系统，拥有普通用户的功能。

登录所需提交的表单包括

（1）用户名

（2）密码

点击登陆后进入网站主页。

**6、数据查询**

数据查询是用户向服务器提交查询请求，输入要查询的商品名以及查询的详细条件可以得到一个系统返回结果，其中包括商品价格、商品价格走势、商品同期价格对比。

查询的数据所需的内容包括：

（1）商品名

（2）地区

（3）地区批发市场

（4）时间范围

点击提交后服务器返回所检索的商品的详细价格信息折线图。

**7、请求监控数据**

请求监控数据功能是用户查看今日的商品数据爬取情况，包括市场总数、商品总数等等。

输入的请求表单内容有：

（1）地区

（2）地区市场

（3）品类

系统返回数据抓取量和实时数据，以各种图表为载体展示。

**8、查询预测价格**

查询预测价格是用户提交商品名以及详细条件后系统对未来七日的价格做出预测并展示，对用户估价起到参考作用

查询的数据所需的内容包括：

（1）商品名

（2）地区

（3）地区批发市场

系统返回预测的未来七日价格折线图。

**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

数据命名规范：均采用相应官方文档中所推荐的命名方案。如数据库库表采用下划线命名法，字段采用驼峰命名法；Java变量采用驼峰命名法；Vue中组件名应为多个单词，中间用‘-’来连接等等。

2、软、硬件环境标准

本系统采用B/S架构。前端采用Vue框架，后端使用SpringBoot框架，数据库使用Mysql，使用Hadoop大数据框架、Hive做数据仓库、Flume提取日志文件、Sqoop进行数据迁移。

服务器均采用Centos操作系统，Web服务器和负载均衡服务器配置均为2核4G，后端业务服务器、数据库服务器、Hadoop节点服务器均为4核16G。

服务器上软件版本如下：

* + - JDK 1.8
    - MySQL 8.0.19
    - Hadoop 3.2.1
    - Hive  3.1.2
    - Sqoop 2
    - Flume 1.9.0
    - Nginx 1.16.1
    - Zeppelin 0.8.1
    - Azkaban

3、接口/协议标准

本系统通过Restful API形式实现浏览器和服务器之间的数据通信。

4、用户界面标准

前端使用Vue框架的Element 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10000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；

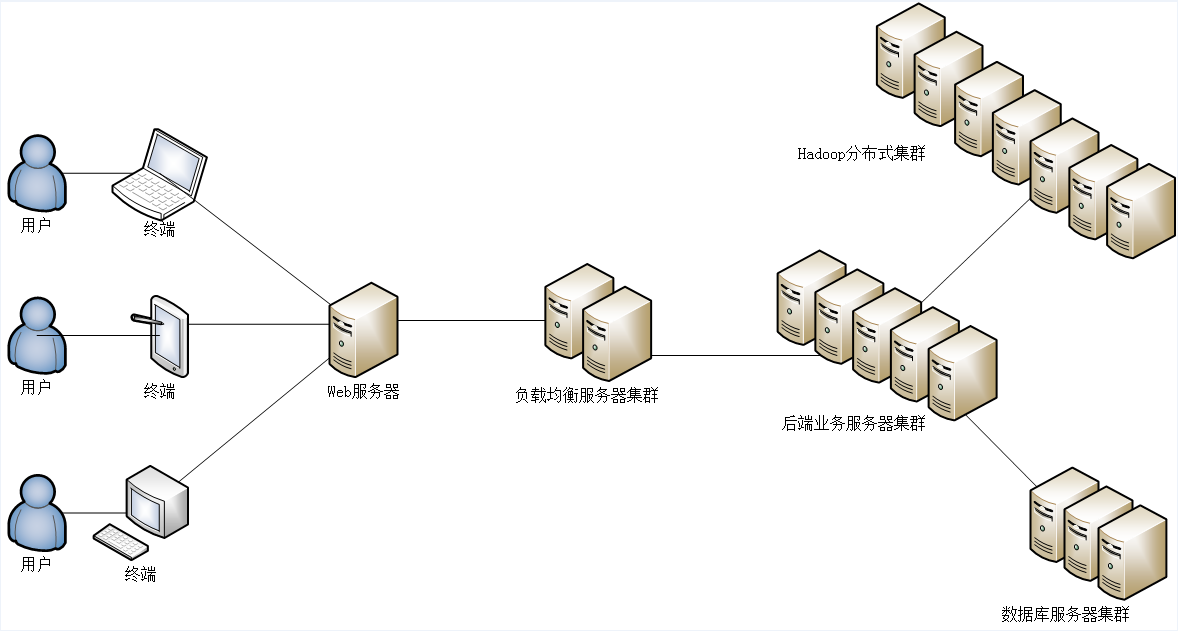
5）安全性

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



注：网络结构说明

A）终端

表示各类能够上网的设备，如PC机、笔记本、手机、iPad

B）用户

操作终端的使用用户

C）Web服务器

指部署前端项目的服务器

D）负载均衡服务器集群

几台负载均衡服务器组成的集群，部署Nginx反向代理进行业务的分流

E）后端业务服务器集群

部署后台代码，接受前端发来的请求，访问数据库或数据仓库进行一定的处理，返回给前端相应的结果

F）数据库服务器集群

主要将用户信息以及数据分析的结果信息存储在Mysql数据库里，集群实现主从复制和读写分离，提高了数据库的高可用性

G）Hadoop分布式集群

这里采用了7个节点的Hadoop分布式架构，可以用来存储大量爬取产生的数据。同时多节点的结构也可以通过一些配置来实现Hadoop的高可用

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）网页兼容多款不同的浏览器如Chrome、Firefox、IE、Edge、Safari；

3）系统对于管理员的操作以及用户的操作保有日志记录

**第四部分 移动平台方案设计**

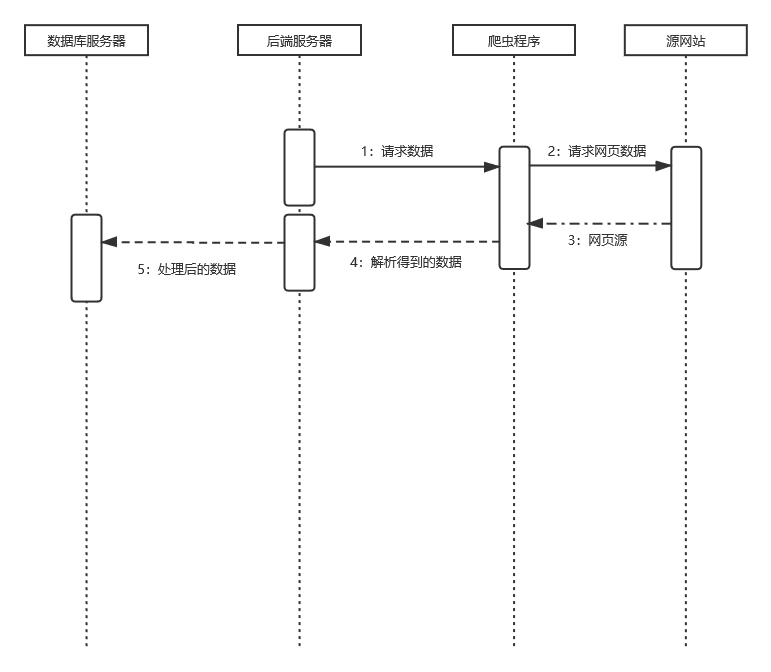
一、安全设计

本平台采用用户数据库的方法管理用户信息，用户名和密码的校验在后台完成，基于此区分用户的登陆状态。

同时添加管理员作为数据维护人员，对过期数据进行删除或更新。

二、平台相关业务流程

1、数据爬取存储流程

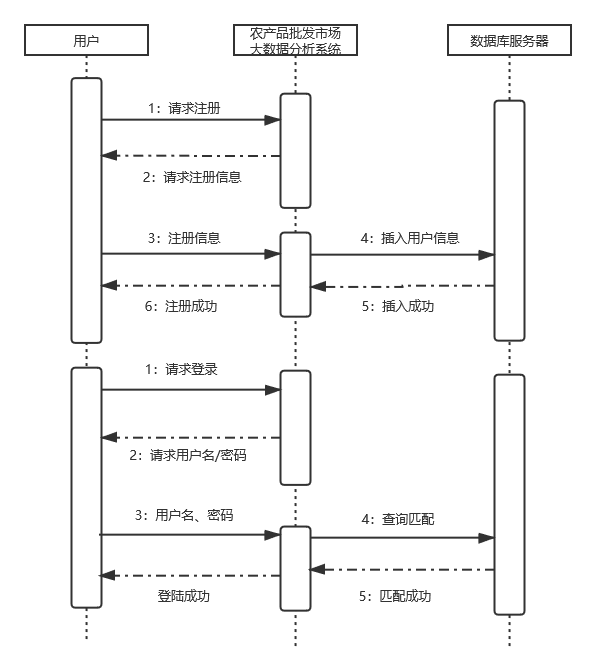


**说明：**

数据爬取存储流程包括：

* + - 开发人员通过后端服务器请求某网站数据
    - 爬虫程序通过代理访问源网页
    - 爬取网页源保存用于分析
    - 爬虫程序通过JSoup等解析网页源
    - 解析得到的数据通过数据统计、清洗等得到需要保存的数据
    - 保存有效数据到数据库服务器中

2、注册登录流程

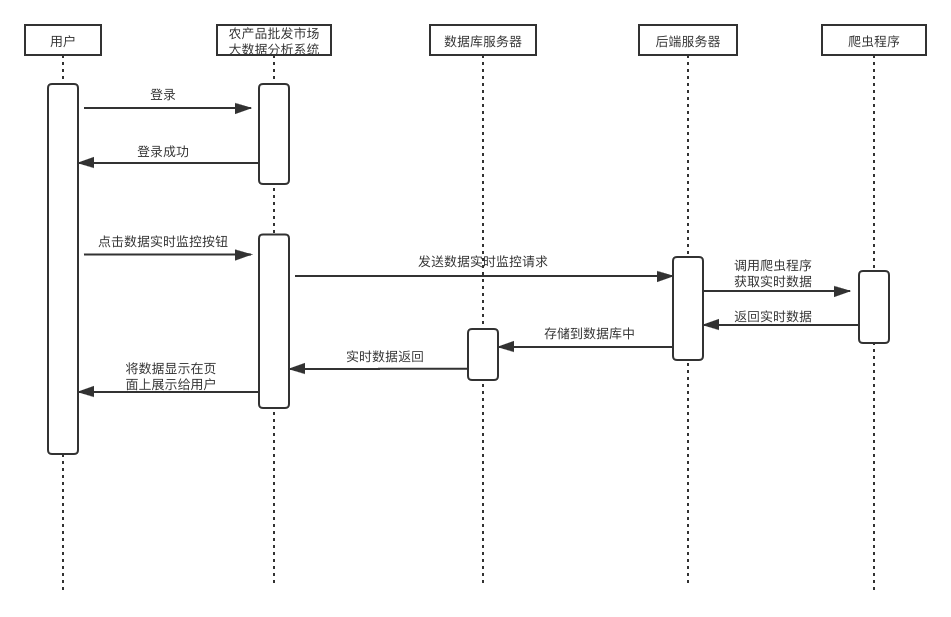


**说明：**

注册登录的主要业务流程包括：

* + 用户通过点击登陆界面上的注册按钮请求注册
  + 系统通过输入框请求注册信息
  + 用户输入注册信息
  + 系统递交注册信息到数据库，执行插入
  + 返回插入结果至系统
  + 系统通过前端界面返回注册结果
  + 用户通过登陆界面请求登录
  + 系统通过输入框请求用户名和密码
  + 系统递交用户名和密码到数据库，执行查询操作
  + 若查询到，则用户名和密码匹配，返回成功，反之返回失败
  + 系统通过前端界面返回登陆结果，页面跳转或出错报警

3、数据监控流程

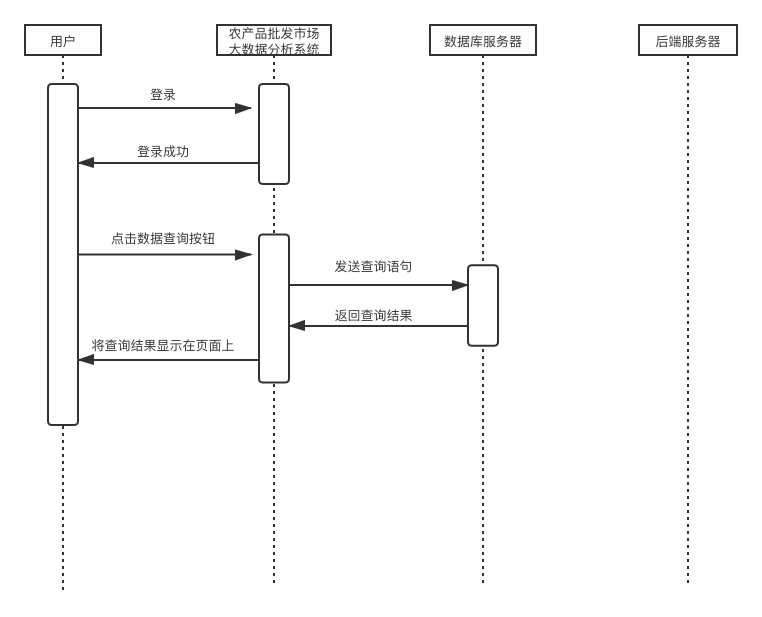


**说明：**

数据监控主要流程包括。

* + 用户登录
  + 用户点击数据实时监控按钮
  + 系统向后端服务器发送数据实时监控请求
  + 后端服务器调用爬虫程序获取实时数据
  + 爬虫程序返回实时数据
  + 后端服务器将实时数据存储到数据库服务器中
  + 数据库服务器返回实时数据给系统
  + 系统将实时数据展示给用户

4、数据查询流程

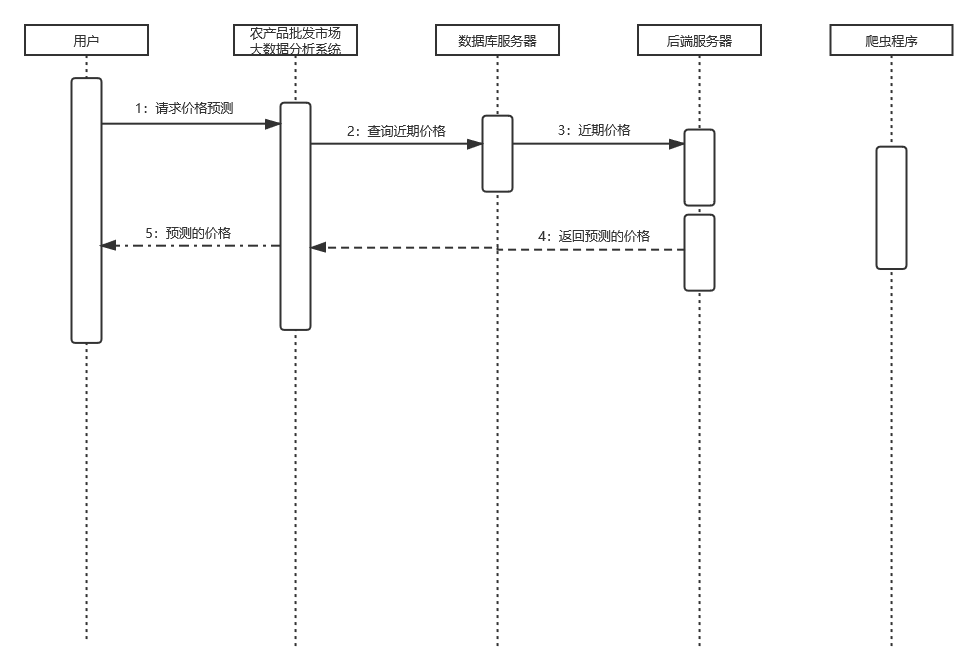


**说明：**

数据查询流程主要流程如下：

* + 用户登录
  + 用户点击数据查询按钮
  + 系统向数据库服务器发送查询语句
  + 数据库服务器返回查询结果给系统
  + 系统将查询结果展示给用户

5、价格预测流程

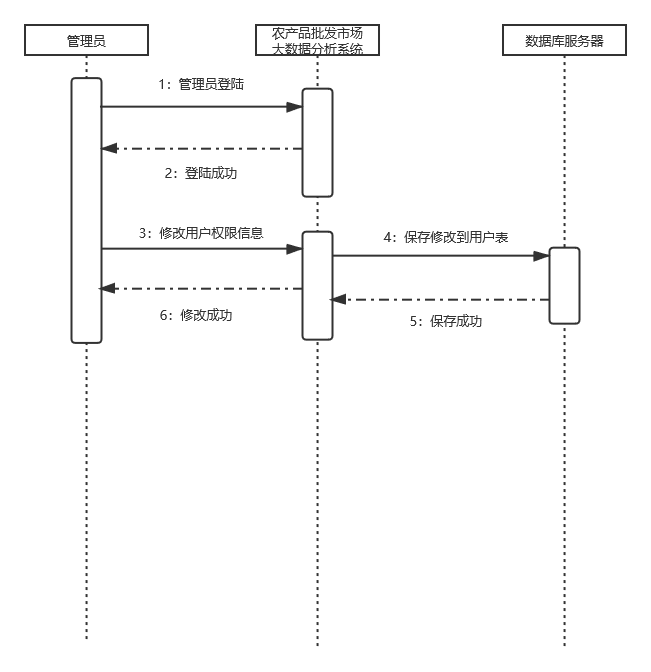


**说明：**

价格查询流程主要流程如下：

* + 用户通过前端页面请求价格预测
  + 系统向数据库请求近段时间价格
  + 数据库提交查询到的价格信息到后端服务器分析
  + 后端服务器返回分析得到的预测结果到系统
  + 系统将预测结果展示给用户

6、用户管理流程



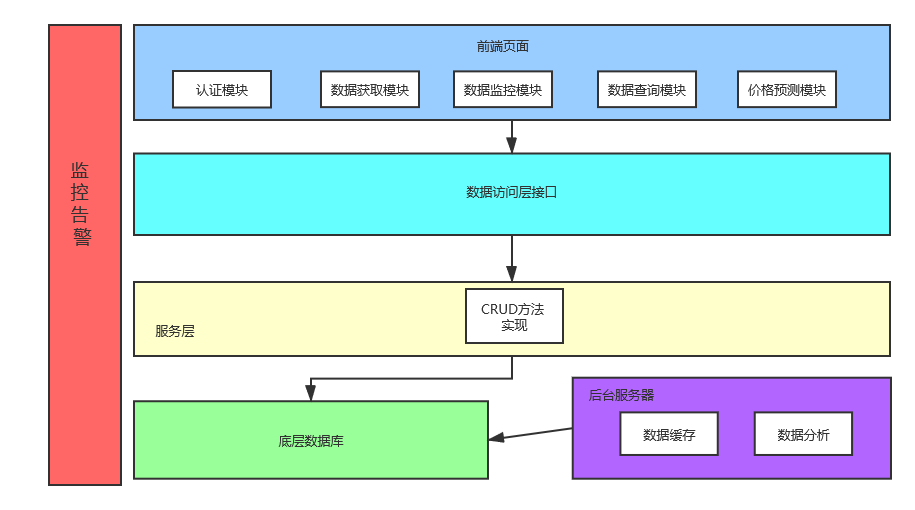
**说明：**

用户管理流程主要流程如下：

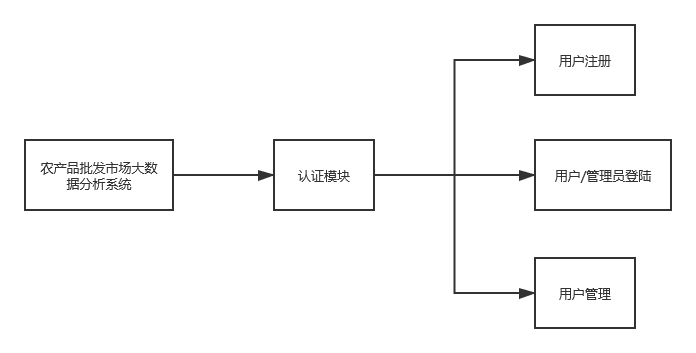
* + 管理员登陆
  + 管理员通过前端界面提出修改用户权限请求
  + 系统将修改的信息提交到数据库
  + 数据库保存修改后返回保存成功至系统
  + 系统将修改成功返回至管理员

三、业务功能概要结构

农产品批发市场大数据分析系统的逻辑结构图如下：



1、认证模块



1.1、用户注册

1、初始化登陆界面，界面初始化完后。

2、用户点击注册按钮，前往注册界面，初始化注册界面。

3、用户在输入区输入注册用户名、密码和验证码完毕。

4、系统获取用户输入。

5、用户将注册信息推送至数据库，执行插入操作。

6、若插入成功则通过界面返回注册成功，否则返回注册失败提醒。

1.2、用户/管理员登录

1、登录界面初始化完。

2、用户/管理员在用户名输入区输入用户名完毕。

3、用户/管理员在密码输入区输入密码完毕。

4、系统获取用户/管理员输入的用户名与密码。

5、传递用户名和密码参数至数据库，执行查询操作。

6、若查询到了一条数据，则证明用户名和密码匹配，返回查询成功至系统，否则返回失败。

7、若系统接收成功信息则界面跳转至主界面，否则返回登录失败提醒。

1.3、用户管理

1、管理员登陆后管理员界面初始化完毕。

2、管理员通过前端页面对用户权限作出修改。

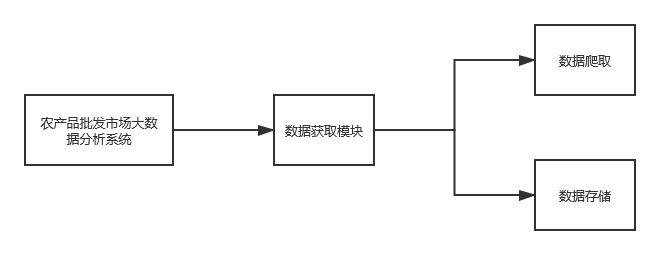
3、系统将管理员做出的修改消息提交到数据库。

4、数据库根据管理员选择的修改操作对用户表作出修改。

5、数据库修改成功后返回修改成功至系统，若出错则返回出错信息。

6、管理员通过前端界面获取修改成功或失败的信息。

2、数据获取模块



2.1、数据爬取

1、项目后台需要每天定时爬取源网站的数据。

2、大数据分析系统通过后端服务器请求源网站数据。

3、爬虫程序通过代理访问源网站，爬取网页源并保存。

4、使用JSoup解析网页源，从中提取需要的数据。

2.2、数据存储

1、使用爬虫抓取的数据分为两部分进行存储。

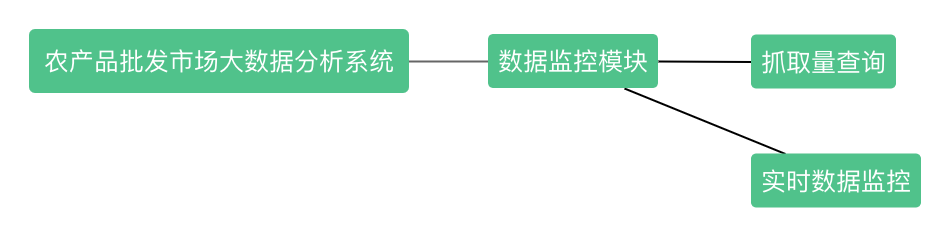
2、一部分存储在分布式系统HDFS中。

3、另一部分存储在Hive中，通过Hive查询将结果导入MySQL数据库。

4、通过MapReduce对数据进行清洗，满足不同需求的数据采取不同的规则清洗。

5、由此得到规范清晰可使用的数据。

3、数据监控模块



3.1、抓取量查询

1、初始化抓取量查询界面

2、系统向数据库服务器请求批发市场目录，以便用户进行选择

3、用户选择特定的批发市场，并点击查询按钮

4、系统向数据库服务器发起抓取量查询的请求

5、显示抓取量查询的结果

3.2、实时数据监控

1、初始化实时数据监控界面

2、系统向数据库服务器请求批发市场目录以及蔬菜品种目录，以便用户进行选择

3、用户选择特定的批发市场或者特定的蔬菜品种，并点击监控按钮

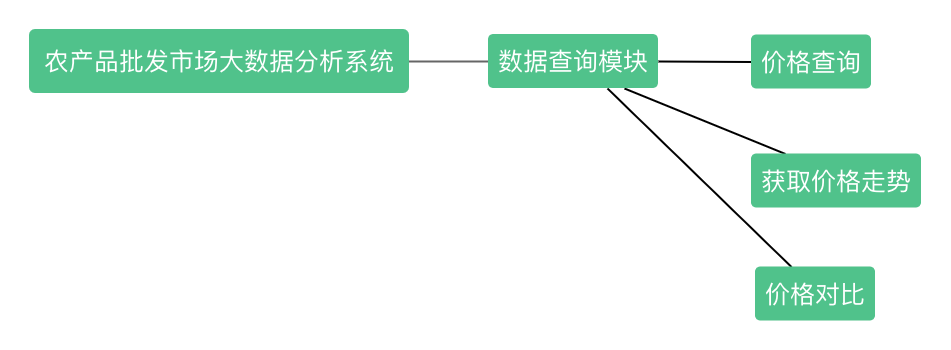
4、系统向服务器发送实时数据监控请求

5、服务器启动爬虫程序，爬取指定批发市场或者蔬菜品种当日最新的价格数据

6、数据保存到hdfs中，然后导入到MySQL中，最后返回给系统

7、显示实时数据监控的结果

4、数据查询模块



4.1、价格查询

1、初始化价格查询界面

2、系统向数据库服务器请求批发市场目录、蔬菜品种目录，以便用户选择

3、用户选择特定的批发市场以及蔬菜品种，并点击查询按钮

4、系统向数据库服务器发起价格查询数据请求

5、显示价格查询的结果

4.2、获取价格走势

1、初始化获取价格走势界面

2、系统向数据库服务器请求蔬菜品种目录，以便用户选择

3、用户选择特定的蔬菜品种，并点击获取按钮

4、系统向数据库服务器发起获取价格走势数据请求

5、显示获取价格走势的结果

4.3、价格对比

1、初始化价格对比界面

2、系统向数据库服务器请求蔬菜品种目录，以便用户选择

3、用户选择特定的蔬菜品种，并点击对比按钮

4、系统向数据库服务器发起价格对比数据请求

5、显示各批发市场该农产品的价格对比图

5、价格预测模块

1、用户在主界面上点击主菜单中的价格预测。

2、初始化价格预测界面。

3、用户选择特定的蔬菜品种，并点击预测按钮。

4、系统向数据库服务器查询近段时间该蔬菜品种的平均价格。

5、后端服务器通过近段时间的价格预测出下一天的价格。

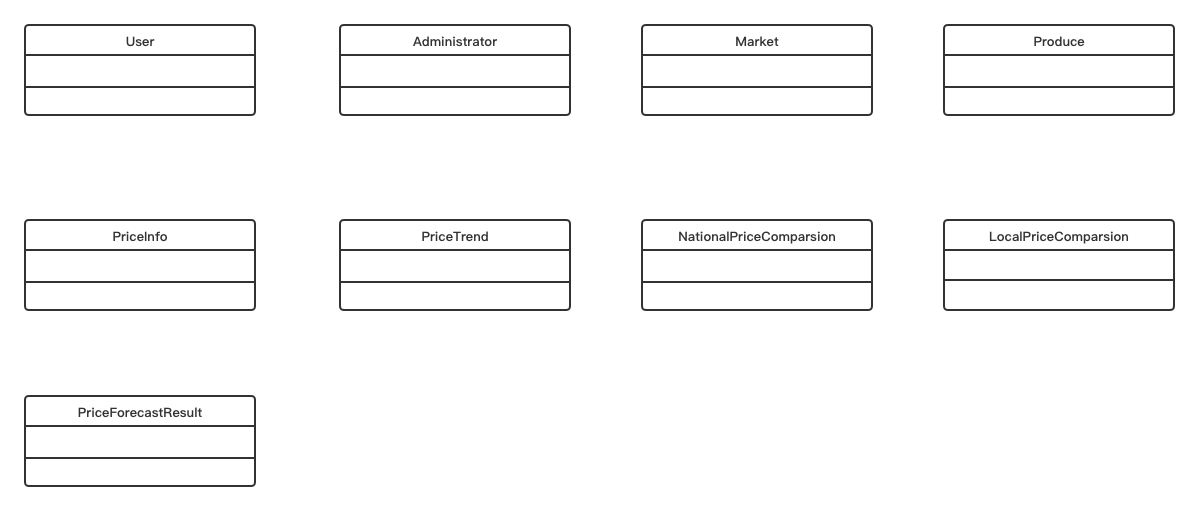
6、前端界面显示近段时间价格变化图和预测的价格。

四、模块定义

|  |  |  |
| --- | --- | --- |
| 模块 | 组建 | 规格/型号 |
| 认证模块 | 用户注册 | 根据用户输入注册信息完成注册 |
| 用户/管理员登陆 | 根据用户名密码是否存在及匹配完成登录 |
| 用户管理 | 管理员为用户修改权限 |
| 数据获取模块 | 数据爬取 | 爬取源网站上需要的数据 |
| 数据存储 | 将爬取的数据经处理后存储到HDFS或数据库 |
| 数据监控模块 | 抓取量查询 | 查询某特定批发市场抓取到的数据总量信息 |
| 实时数据监控 | 实时抓取特定批发市场或蔬菜品种的最新价格记录 |
| 数据查询模块 | 价格查询 | 查询特定批发市场特定蔬菜品种的价格记录 |
| 获取价格走势 | 以图表形式展示一段时间内某蔬菜的价格 |
| 价格对比 | 以图表形式展示各批发市场某蔬菜价格对比图 |
| 价格预测模块 |  | 查询近段时间价格并基于此预测价格并显示 |

**第五部分 E-R实体设计**

一、E-R实体结构图



二、实体描述

1、User实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | username | 用户id | String |
| 2 | password | 用户密码 | String |
| 3 | name | 用户昵称 | String |
| 4 | sex | 用户性别 | char |
| 5 | phoneNum | 用户注册手机号 | String |
| 6 | emailAddr | 用户邮箱地址 | String |

2、Administrator实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | username | 管理员id | String |
| 2 | password | 管理员密码 | String |
| 3 | name | 管理员姓名 | String |
| 4 | phoneNum | 管理员手机号 | String |
| 5 | emailAddr | 管理员邮箱地址 | String |

3、Market实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | marketId | 批发市场id | String |
| 2 | marketName | 批发市场名字 | String |
| 3 | province | 批发市场所在省份 | String |
| 4 | city | 批发市场所在城市 | String |

4、Produce实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produceId | 农产品编号 | String |
| 2 | produceName | 农产品姓名 | String |
| 3 | produceType | 农产品种类 | String |

5、PriceInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produce | 农产品 | Produce |
| 2 | market | 批发市场 | Market |
| 3 | date | 日期 | Date |
| 4 | price | 在日期date，maker市场中produce的价格 | double |

6、PriceTrend实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produce | 农产品 | Produce |
| 2 | startDate | 开始日期 | Date |
| 3 | endDate | 结束日期 | Date |
| 4 | maxPriceArray | 农产品从开始日期到结束日期每天的最高价数组 | ArrayList<Double> |
| 5 | minPriceArray | 农产品从开始日期到结束日期每天的最低价数组 | ArrayList<Double> |
| 6 | averagePriceArray | 农产品从开始日期到结束日期每天的平均价格数组 | ArrayList<Double> |

7、NationalPriceComparsion实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produce | 农产品 | Produce |
| 2 | priceMap | 用于存放该农产品在不同省份的不同平均价格 | Map<String,Double> |

8、LocalPriceComparsion实体描述

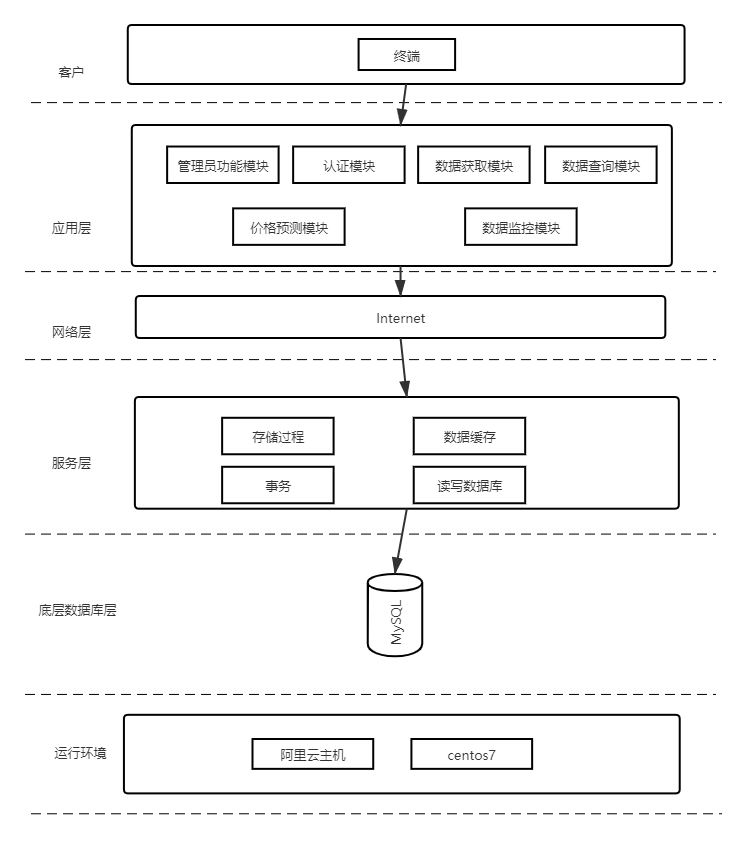
|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produce | 农产品 | Produce |
| 2 | priceMap | 用于存放该农产品在不同批发市场的不同价格 | Map<Market,Double> |

9、PriceForecastResult实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | produce | 农产品 | Produce |
| 2 | forecastResult | 农产品预测结果 | double |

**第六部分 总体设计**

一、逻辑架构



1、J2EE四层架构

从构件化与集成化的角度出发，采用J2EE四层架构的技术方案、构件化的开发模式，应用功能的实现在支撑构件的基础上完成。实现了和OS的无关性，保证系统的跨平台、扩展性和安全性。

2、B/S架构

B/S架构采取浏览器请求，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；从Web服务器上下载程序到本地来执行，在下载过程中若遇到与数据库有关的指令，由Web服务器交给数据库服务器来解释执行，并返回给Web服务器，Web服务器又返回给用户。

3、服务器集群

通过高性能网络或局域网互联的服务器集群分而治之，可以获得很高的整体性能。采用多台阿里云服务器可以承受网络负载中大量的独立任务，分布式的结构可以大大减少服务器的压力。

二、物理架构

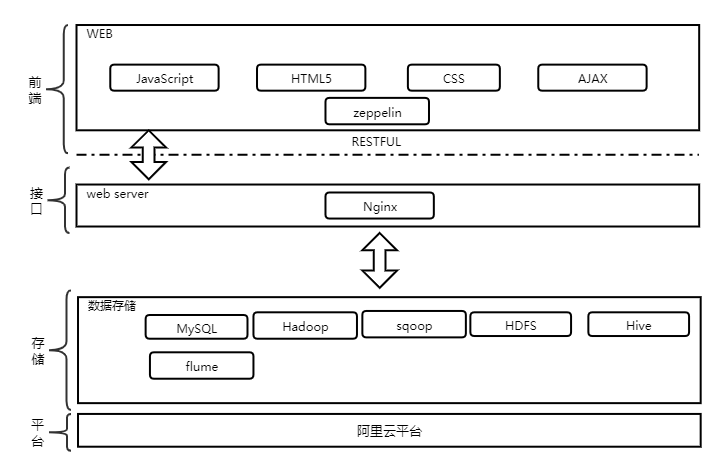
1、服务端

系统服务端部署到阿里云云平台上，系统管理员通过浏览器实现业务管理操作；

2、客户端

因为系统采用B/S模式，所有终端只要求浏览器支持；

三、技术架构



1、RESTFUL接口

RESTful风格，就是目前最流行的一种互联网软件架构。它结构清晰、符合标准、易于理解、扩展方便，所以正得到越来越多网站的采用。

2、Hadoop在分布式环境中存储和处理大数据

Hadoop具有按位存储和处理数据能力的高可靠性。通过可用的计算机集群分配数据，完成存储和计算任务，这些集群可以方便地扩展到数以千计的节点中，具有高扩展性。Hadoop能够在节点之间进行动态地移动数据，并保证各个节点的动态平衡，处理速度非常快，具有高效性。

3、vue前端框架

vue的双向数据绑定特性以及技术的成形，实现了项目的热加载，改完页面代码能立即在浏览器方面显示效果，提高开发效率

4、Flume

Flume是一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据，将大批量的不同数据源的日志数据收集、聚合、移动到数据中心（HDFS）进行存储的系统。即是日志采集和汇总的工具同时，Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

第七部分 用户界面设计

一、桌面布局设计

1、登录界面风格



1. 主桌面风格



1.点击导航列表按钮后进入相应的内容页面

2.点击右上角的人物图标可以进入相应的个人资料设置，可以更改个人信息以及推出登录。

3.左边的菜单栏中，将每一个功能都列了出来，将来有新增功能时会在左边进行添加，功能过多时，会设置多级菜单，尽量保持页面风格简洁。

二、业务界面风格展示

1、风格展示



1.点击导航列表按钮后进入相应的内容页面

2.点击右上角的人物图标可以进入相应的个人资料设置，可以更改个人信息以及推出登录。

3.左边的菜单栏中，将每一个功能都列了出来，将来有新增功能时会在左边进行添加，功能过多时，会设置多级菜单，尽量保持页面风格简洁。

**第八部分 运行环境和部署**

一、运行环境

1、浏览器环境

1) Edge

2) IE

3) FireFox

4) Chrome

5) Safari

2、客户机器环境

1）Windows

2）MacOs

3）Linux

3、开发环境要求

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 版本 |
| 开发平台 | Mac OS | 10.7 |
|  | Windows | 10 |
|  | CentOS | 7,8 |
| 开发工具 | IntelliJ IDEA | 2020.1.3 |
|  | IntelliJ WebStorm | 2020.1.3 |
| 代码管理工具 | Gogs | 0.11.91 |
| 开发环境 | SpringBoot |  |
|  | Vue |  |
|  | MySQL | 8.0.19 |

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B/S模式，支持所有浏览器的访问。 |
| 2 | 面向对象开发语言与框架 | A | 采用Java语言，多方框架进行开发 |
| 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 | 采用目前体验最好、最流行的iPhone与iPad移动终端。 |
| 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 | WebService服务器端 | A | 支持协议定制，支持安全过滤，支持消息队列，多种模式重发，支持成功检测。消息延迟<=2S，丢包<=0.001%。 |
| 16 | MAP平台端 | A | 消息延迟<=2S，丢包<=0.001%。 |
| 17 | 集成部署环境 | 服务器 | A | CentOS7及以上 |
| 18 | 数据库 | A | Mysql8及以上 |
| 19 | 手持机 | A | Android/IOS |
| 20 | PC电脑 | A | Win/MacOs/Linux |
| 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.） | | | | |