

**本 科 生 毕 业 设 计（论文）**

**（2018届）**

**信息工程学院**

**题 目： 产品流通的可视化表达与数据挖掘**

**学 号： 201505010315**

**姓 名： 倪畅**

**专业班级： 计算机科学与技术153班**

**指导教师： 陈文辉 职称： 讲师**

**2019年03月20日**



**本科生毕业设计（论文）诚信承诺书**

我谨在此承诺：本人所写的毕业设计（论文）《产品流通的可视化表达与数据挖掘》均系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了引用注释，如出现抄袭及侵犯他人知识产权的情况，后果由本人承担。

**承诺人（签名）：**

**2019年03月20日**

产品流通的可视化表达与数据挖掘

信息工程学院 计算机科学与技术153班 倪畅 指导教师：陈文辉

**摘要：**二十一世纪后，互联网行业的高速发展，正在慢慢改变人们的日常生活，衣食住行无不和互联网行业有着千丝万缕的联系。越来越多的互联网技术被运用在人们的日常生活中，诸如：人工智能、云计算、大数据、机器学习等等。为了加快提升人们的生活质量，我国正在大力推动智慧城市建设。旨在通过物联网和新一代互联网技术，结合新一代的创新思想，对社会的民生、治安、医疗、公共服务等作出智能的应答。基于目前已经相对成熟的智慧政务、智慧交通、智慧医疗和智慧旅游等项目的建设，本文主要研究以农产品为主要研究对象的产品流通和可视化及数据挖掘，从而搭建一个应用平台来便利民生。

通过市场的调研，了解了目前农产品的需求度和各个菜市场对农产品规范化的空缺，制定了合适的需求模型和设计方案：采用B/S架构；基于Spring、SpringMVC、Mybatis框架并利用Java8、Druid、Lombok、Logback、WebSocket、Maven等技术搭建服务端；基于React框架并利用Javascript、bootstrap、axios、Ant Design、 Echarts、npm等技术搭建用户端；基于ECharts进行大数据的可视化模块设计；基于二次指数平滑算法进行数据预测的实现；利用Tomcat和MySQL进行用户端和服务端的整合。

**关键词：**大数据；数据挖掘；预测分析；智慧城市；可视化平台

Visual representation and data mining of product circulation

**Abstract：**After the 21st century, the rapid development of the Internet industry is slowly changing people's daily lives, and the Internet, the industry, and the Internet industry are inextricably linked. More and more Internet technologies are being used in people's daily lives, such as artificial intelligence, cloud computing, big data, machine learning, and so on. In order to speed up the improvement of people's quality of life, China is vigorously promoting the construction of smart cities. It aims to respond intelligently to the people's livelihood, public security, medical care, and public services through the Internet of Things and next-generation Internet technologies, combined with a new generation of innovative ideas. Based on the current mature projects of smart government, smart transportation, smart medical care and smart tourism, this paper mainly studies product circulation and visualization and data mining with agricultural products as the main research object, thus building an application platform to facilitate people's livelihood.

Through market research, we have learned about the current demand for agricultural products and the vacancies in the standardization of agricultural products in various vegetable markets. We have developed a suitable demand model and design scheme: using B/S architecture; based on Spring, SpringMVC, Mybatis framework and using Java8, Druid , Lombok, Logback, WebSocket, Maven and other technologies to build the server; based on the React framework and use Javascript, bootstrap, axios, Ant Design, Echarts, npm and other technologies to build the user terminal; based on ECharts for large data visualization module design; based on the second Exponential smoothing algorithm for data prediction; using Tomcat and MySQL for client-side and server-side integration.

**KeyWords：**Big Data; Data mining; Predictive analysis; Smart City; Visualization platform

**目 录**

[1 绪论 3](#_Toc5638976)

[1.1 课题的背景 3](#_Toc5638977)

[1.2 现状的分析 3](#_Toc5638978)

[1.3 产品流通的可视化表达与数据挖掘应用平台的优势 3](#_Toc5638979)

[1.4 本论文主要结构 4](#_Toc5638980)

[2 开发环境与相关技术 5](#_Toc5638981)

[2.1 系统开发环境 5](#_Toc5638982)

[2.2 React 5](#_Toc5638983)

[2.3 ECharts 5](#_Toc5638984)

[2.4 npm 6](#_Toc5638985)

[2.5 axios 6](#_Toc5638986)

[2.6 SSM框架 6](#_Toc5638987)

[2.6.1 Spring 6](#_Toc5638988)

[2.6.2 SpringMVC 7](#_Toc5638989)

[2.6.1 Mybatis 8](#_Toc5638990)

[2.7 Java8 9](#_Toc5638991)

[2.8 Druid 10](#_Toc5638992)

[2.9 Lombok 10](#_Toc5638993)

[2.10 Logback 10](#_Toc5638994)

[2.11 Maven 10](#_Toc5638995)

[2.12 Websocket 10](#_Toc5638996)

[3 系统设计分析 12](#_Toc5638997)

[3.1 系统设计思想 12](#_Toc5638998)

[3.2 可行性分析 12](#_Toc5638999)

[3.2.1 技术可行性 12](#_Toc5639000)

[3.2.2 经济可行性 12](#_Toc5639001)

[3.2.3 操作可行性 13](#_Toc5639002)

[3.2.4 组织可行性 13](#_Toc5639003)

[3.3 业务流程 13](#_Toc5639004)

[3.3.1 系统流程图 13](#_Toc5639005)

[3.3.2 数据流程图 14](#_Toc5639006)

[3.4 用例分析 14](#_Toc5639007)

[4 系统设计 16](#_Toc5639008)

[4.1 非功能性需求 16](#_Toc5639009)

[4.2 功能性需求 16](#_Toc5639010)

[4.2.1 数据爬取 16](#_Toc5639011)

[4.2.2 产品未来价格预测分析 17](#_Toc5639012)

[4.2.3 大数据汇总 18](#_Toc5639013)

[4.2.4 商品查询 18](#_Toc5639014)

[4.2.5 登录和注册 18](#_Toc5639015)

[4.3 数据库概念结构设计 18](#_Toc5639016)

[4.4 数据库的逻辑结构设计 19](#_Toc5639017)

[5 系统实现 21](#_Toc5639018)

[5.1 系统流程图 21](#_Toc5639019)

[5.2 业务代码执行流程 21](#_Toc5639020)

[5.3 项目文件结构 22](#_Toc5639021)

[5.4 编号命名规范 23](#_Toc5639022)

[5.5 用户登录 23](#_Toc5639023)

[5.6 用户主界面 24](#_Toc5639024)

[5.6.1 留言板 24](#_Toc5639025)

[5.6.2 个人信息修改 25](#_Toc5639026)

[5.6.3 标题部分 25](#_Toc5639027)

[5.7 货运管理 25](#_Toc5639028)

[5.7.1 购销合同 26](#_Toc5639029)

[5.7.2 出货表 28](#_Toc5639030)

[5.7.3 出口报运 28](#_Toc5639031)

[5.7.4 装箱管理 29](#_Toc5639032)

[5.7.5 委托书 31](#_Toc5639033)

[5.8 统计分析 32](#_Toc5639034)

[5.8.1 生产厂家销售情况 33](#_Toc5639035)

[5.8.2 产品销售排行 33](#_Toc5639036)

[5.8.3 系统访问压力图 33](#_Toc5639037)

[5.8.4 财务统计 34](#_Toc5639038)

[5.9 基础信息 34](#_Toc5639039)

[5.9.1 数据字典 34](#_Toc5639040)

[5.9.2 厂家信息 34](#_Toc5639041)

[6 系统管理 36](#_Toc5639042)

[6.1 角色管理 36](#_Toc5639043)

[6.2 部门管理 38](#_Toc5639044)

[6.3 用户管理 38](#_Toc5639045)

[7 总结 40](#_Toc5639046)

[参考文献 41](#_Toc5639047)

[致谢 42](#_Toc5639048)

**1** **绪论**

**1.1 课题的背景**

“买菜”这一动作行为可以说是生活中我们每个人都会遇见的，每一个人都将会长大，承担家庭的责任。那么，为自己的家人做一顿可口的饭菜就免不了买菜。但是，随着生活水平的提高，反而越来越多的人对于买菜没有任何经验。他们吃惯了美味的外卖，就连水果，也是帮你切好，外卖到家，完全不知道现在蔬菜水果的价格行情。这就让有些卖菜商人们有机可乘：一味地抬高菜价，坐地起价，以次充好。这极大影响了广大市民们的生活水平和质量。

其次，一方面，生活在一线城市的人们的时间都是很宝贵的，他们工作压力大，为节省买菜时间，他们从不议价；另一方面，很多刚刚进入社会的小年轻们，从小享受着父母的宠爱，他们也不会议价。这就造成了目前市场鱼龙混杂，农产品价格不公开化透明化的问题。

于是，本文所描述的农产品的流通可视化应用平台也就应运而生。它不仅可以提供给市民们公正透明公开的菜场价格，还能让国家清楚的知道我国人民对各类农产品的需求，便于国家大局的规划；同时，他也能让菜农们了解现在市场最需要什么，让他们耕耘有方，不再滞销浪费。

**1.2 现状的分析**

“物价高”已经是我国的十大社会问题之一。价格上涨与食品安全和房价密切相关。令人惊讶的是物价高当然不仅仅是价格的上涨。当下媒体广泛报道的“卖菜难、买菜贵”也是价格上涨的问题。政府对房价调整的控制力不高，但对一般农副产品的监管力度相当强大，但由于它太强大，到最后，生活在底层的老百姓们就成了最大的受害者。一方面，农民烦恼“卖菜难”，另一方面，城市普通居民感到“买菜贵”。在价格上涨之后，买卖的不和谐声音必然是不可避免的问题。那么，我们为什么不去打造一个平台，让市场中的菜价透明化公正化规范化，每一种农副产品的价格都能追溯它的源头，真正的解决好买卖农副产品双方的主要矛盾，有利于提高市场自我管理的能力。同时，该平台既解决了菜农们农副产品的滞销，也有效解决了买菜市民对菜品价格和质量的抱怨。

其次，国家目前对智慧城市的建设如火如荼，其中在公共交通、公共资源、医疗、政务、旅游等行业已经逐渐成熟，但是在农产品这方面还未见其雏形，所以，这一行业的市场空缺很大，值得去研究开发一个智慧农产品平台——产品流通可视化和数据挖掘应用平台。

**1.3 产品流通的可视化表达与数据挖掘应用平台的优势**

柴米油盐酱醋茶，这些都与菜市场有着密不可分的关系。作为城市公共配套设施，菜市场关系着千家万户的菜篮子，是民生所系、政府之责、公共服务必需。买菜方不方便，菜价便不便宜，菜品安不安全，与每个人的日常生活息息相关，直接影响着人们的幸福指数。对于本平台的优势主要体现在以下五个方面：

（1）实时价格显示：减少了菜农和买菜市民之间还价时间，便于规范市场价格规范，避免了市民买菜贵的现象；

（2）单一农副产品价格查询：用户买菜前可以先在平台上了解菜品的历史价格，防止出现农副产品价格溢价的情况；

（3）所需农副产品未来七天价格预测：帮助买菜市民们了解未来的价格趋势，同时也能让菜农们知道未来该种植什么作物能带来更多的收益；

（4）推动我国的智慧城市建设发展，为我国智慧城市建设带来多元化发展；

（5）适当解决我国的某些社会民生问题，促进我国的社会经济的发展。

**1.4 本论文主要结构**

**2 开发环境与相关技术**

**2.1 系统开发环境**

系统开发环境详情见表2-1所示：

表2-1 系统开发环境表

|  |  |
| --- | --- |
| 类别 | 内容 |
| 系统 | Windows10  Windows7 |
| 编译器 | IntelliJ IDEA 2018.3.4  JetBrains WebStorm 2018.2.5 |
| 服务器 | Tomcat 7.0.92 |
| 数据库 | MySQL5.7 |
| 项目构建 | Maven  Npm |
| 框架 | Spring+SpringMVC+ MyBatis  React |
| 测试工具 | Postman |
| 浏览器 | Chrome |
| 设计语言 | Java8  JavaScript |
| 数据库建模 | PowerDesigner15.1  Navicat Premium 12 |

**2.2 React**

React的速度很快，与其他框架相比，React框架采取了一种特立独行的操作DOM的方式。它并不直接对DOM进行操作，它引入了一个“虚拟DOM”的概念，并把它安插在JavaScript逻辑和实现DOM之间；这一概念有效地提高了Web的性能。在渲染的过程中，React通过在虚拟DOM中的微操作来实现对实际DOM的局部更新。其次，这一虚拟DOM的概念还帮我们解决了跨浏览器的兼容问题，他为我们提供了标准的API接口。

React能进行模块化管理，使用React可以对要实现的每一个功能进行模块化隔离，模块之间互不影响，在开发过程中进行Bug修复、单元测试都十分简单高效，有效地提高了代码的可维护性。

**2.3 ECharts**

ECharts这项技术对于目前的大数据可视化可以说是十分地重要，它支持多种图表的格式显示，具有丰富的可视化图标类型，包括：折线图、柱状图、饼图、地图、热点图、散点图等等。用户都能通过配置图表的属性来定制自己所需要的报表。而且ECharts的多种数据格式无需转换可以直接使用，通过后台的JSON数据返回，既能做到定时刷新，也能做到实时更新数据，这一特性对于本文所描述的“产品流通可视化和数据挖掘应用平台”具有很好的可利用性：进行千万数据的前端渲染展现，对开发这个大数据应用平台有着很大的帮助，极大地简化了大数据可视化的开发步骤，缩短了开发所需的时间。

**2.4 npm**

npm的全称叫做Node Package Manager，它是一款Node的包管理工具，目前Node.js已经集成了npm。npm对我们开发者来说最大的优点就是方便我们引入各种所需要的包（npm install XXX），npm还提供了一个重用其它开发者代码的方式，npm也能够分享开发者自己的代码，同事也提供了一个便捷的方式来来管理不同版本的代码。

**2.5 axios**

axios就是基于promise用于浏览器和node.js的http客户端。它的主要特点如下：

（1）从浏览器生成xmlhttpRequests

（2）从node.js发出HTTP请求

（3）支持Promise API

（4）拦截请求和响应

（5）转换请求和响应数据

（6）取消请求

（7）JSON数据的自动转换

（8）客户端支持防止XSRF

之所以我选择使用axios而不使用ajax请求，是因为本文所描述的应用平台在系统设计上就采用前后端分离技术，没有使用传统的MVC编程，ajax不符合现在前端MVVM的潮流，而且JQuery的整个项目太大，单纯使用ajax就要把整个JQuery全部引入，这显得非常不合理。而axios就比较小巧灵活，使用npm install axios就能安装使用，十分方便。

**2.6 SSM框架**

SSM框架的全称是Spring、SpringMVC和Mybatis。SSM框架是继SSH（Struts、Spring和Hibernate）之后目前市面上比较主流的Java EE企业级开发框架，适用于搭建各种中大型的企业级应用系统。当然，目前市面上还有一种框架——Spring Boot，但是Spring Boot作为一个微服务框架，距离微服务的实现还是存在一些距离的。这个框架只是为了提高开发效率，为了提高开发人员的生产力，对于Restful格式的URL请求还需要开发人员自行结合实际进行规范化工作。就框架的配套插件来说，SSM框架还是远远强于Spring Boot框架的，所以，本课题所研究开发的大数据应用平台还是决定使用SSM框架。

2.6.1 Spring

Spring的优点有很多，它可以很好地解决企业级大型项目的复杂性。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。 简单来说，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。下面我就简单介绍几个Spring的特点：

（1）控制反转：

控制反转简称IOC，也叫依赖注入。在工厂模式下，对象由容器管理。您只需要在Spring配置文件中配置相应的bean，并设置相关的属性，这样Spring容器就可以生成类的实例对象和管理对象。当Spring容器启动时，Spring初始化配置文件中配置的所有bean，然后将那些已经初始化的bean分配给需要调用的类（假定类名为a），当需要调用它们时。分配方法是调用的setter方法来注入它们，而不是需要您在中更新这些bean。

（2）面向切面：

面向切面简称AOP，是Spring中另一个特点。首先，需要说明的是，AOP只是Spring的特征。与OOP一样，AOP是一种编程思想，而不是一种技术。AOP可以说是对OOP的补充和改进。OOP引入了封装、继承和多态的概念，建立了一个对象层次结构来模拟一组公共行为。当我们需要将公共行为引入分散的对象时，OOP似乎是无能为力的。也就是说，OOP允许您定义自上而下的关系，但它不适合定义从左到右的关系。例如，日志函数。日志代码倾向于水平地分布在所有对象级别，而不管其分布到的对象的核心功能是什么。在OOP设计中，会导致大量的代码重复，这不利于每个模块的重用。程序中的跨业务逻辑（如安全性、日志、事务等）被封装到一个方面，然后注入目标对象（特定的业务逻辑）。

2.6.2 SpringMVC

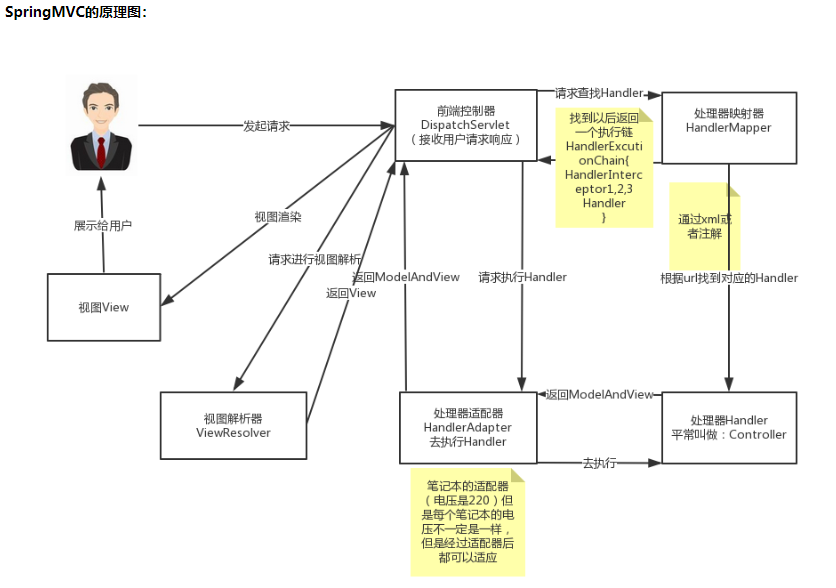
SpringMVC框架是一个MVC框架，它通过实现模型-视图-控制器模式来分离数据、业务和表示。从这个角度来看，SpringMVC与Struts和Struts2非常相似。SpringMVC是围绕DispatcherServlet设计的，它负责将请求分派给特定的处理程序。请求通过可配置的处理程序映射（handler mappings）、视图解析（view resolution）、区域设置（locale）和主题解析（theme resolution）进行处理，并传输到相应的视图。

前端控制器（DisatcherServlet）:接收请求，响应结果，返回可以是json,String等数据类型，也可以是页面（Model）。

处理器映射器（HandlerMapping）:根据URL去查找处理器，一般通过xml配置或者注解进行查找。

处理器（Handler）：就是我们常说的controller控制器啦，由程序员编写。

处理器适配器（HandlerAdapter）:可以将处理器包装成适配器，这样就可以支持多种类型的处理器。

视图解析器（ViewResovler）:进行视图解析，返回view对象（常见的有JSP,FreeMark等）。

下面我就简单的描述一下SpringMVC的主要流程：

（1）用户发送请求到前端控制器（DispatcherServlet）。

（2）前端控制器请求处理器映射器（HandlerMapping）去查找处理器（Handler）。

（3）找到以后处理器映射器（HandlerMappering）向前端控制器返回执行链（HandlerExecutionChain）。

（4）前端控制器（DispatcherServlet）调用处理器适配器（HandlerAdapter）去执行处理器（Handler）。

（5）处理器适配器去执行Handler。

（6）处理器执行完给处理器适配器返回ModelAndView。

（7）处理器适配器向前端控制器返

回ModelAndView。

（8）前端控制器请求视图解析器（ViewResolver）去进行视图解析。

（9）视图解析器向前端控制器返回View。

（10）前端控制器对视图进行渲染。

（11）前端控制器向用户响应结果。

2.6.1 Mybatis

Mybatis相对于传统的数据库连接方式JDBC来说，具有很大的优越性和先进性。首先，Mybatis可以做到对SQL语句的统一管理，有专门的xml文件来配置数据库和SQL语句，根据查询所需要的参数进行动态的拼接SQL语句；而我们使用JDBC对数据库进行操作时，SQL查询语句分布在若干个Java类中，这样可读性和可维护性都比较差，代码不够规范，缺少模块化。其次，Mybatis可以对结果进行映射，你可以返回任意类型的数据，这个类型包括但不限于java类、list、map、tree等等。但是JDBC就表现的很不起眼，它仅仅返回一个ResultSet结果集，然后开发人员还要从这个ResultSet结果集中取出我们想要的结果并封装成所需要的类型。

**2.7 Java8**

对于本研究的课题，本人考虑了后期数据量会数百上千万的情况，为了让用户能有更好的软件使用体验和更优的软件流畅及稳定性。再加上目前市场Java8的环境逐渐成熟，出于综合考虑，本人决定使用Java8作为开发语言；下面，本人将阐述Java8的新特性在系统中的使用。

（1）Stream流

Java8中的流式处理在本应用系统平台中已经无处不在了，Java8的流式处理代替了此前版本的for循环，用简洁的语法，就能对集合进行操作。身为一个程序开发者，我们都知道编码中对集合进行操作处理是非常麻烦的：传统操作不仅需要大量的编码，操作效率还十分低下。但在Java8中我们可以通过流，它就像是一个SQL语句，你只要告诉它你需要什么，他就会返回给你什么结果，这极大提高了开发效率和开发质量。

（2）Lambda表达式

身为一个开发者，对于匿名内部类大家可能都不陌生，编码的简洁让越来越多的人喜欢上了它。千万不要以为这已经是最简单的，Java8就对匿名内部类进行了优化升级，甚至可以说是一次翻天地府的改版。无中生有，从无到有可以说是对Java8中的Lambda表达式的最简概括。

Lambda表达式格式：

(parameters) -> expression或(parameters) ->{statements;}

不管你写的方法需不需要参数或者这个方法需不需要返回值，用Lambda表达式编写你都能很好的解决，并且带来很好的可读性和可维护性，维护人员也不用再抓着头皮反复阅读他人编写的冗长代码。

（3）函数式编程

函数编程的最直接的表现，莫过于将函数作为数据自由传递，结合泛型推导能力，使代码表达能力获得飞一般的提升。Java8的函数式编程主要体现在以下三个方面：函数接口(Function)、流(Stream)和聚合器(Collector)。这几个方面在本文中我均已阐述，当然如果大家对这方面感兴趣，可以阅读我在文末的参考文献Java8实战一书，相信你会因此爱上Java8。

（4）Java8中的线程池

从Java线程创建、销毁和减去线程对于计算机资源来说是一项相对昂贵的任务。如果我们需要使用多线程来处理任务，频繁地创建和销毁线程，就会造成计算机资源的浪费，从而出现线程池技术。Java8中的ThreadPoolExecutor方法完美地提供了解决Java中多线程中的方法。本系统就是使用了这种方法，在该部分功能的具体实现中我会详细介绍，在此不再敖述。

**2.8 Druid**

数据库的连接建立和关闭是十分消耗服务器集群的内存和资源的，尤其是在大型项目中，数据库带来的资源浪费和消耗尤为明显。合理的控制数据库请求次数和数据库连接池的最大连接数对于开发者来说简直苦不堪言。这时候，使用阿里巴巴开源的Druid连接池就不存在这种问题，所有的设置只需要建立一个javax.sql.DataSource的实例就完全搞定。相比原来的JDBC，使用Druid不仅可以减少大量开发时间来配置连接池的相关设置，还能保证数据库的安全性，阿里巴巴Druid连接池仅支持通过特殊算法加密后的加密密码进行连接，对于软件安全的提升有很大帮助。

**2.9 Lombok**

Lombok可以说是另一个给程序员带来的福音，只要在IDE和项目中配置好Lombok，所有Javabean不用再编写构造函数方法、GetterAndSetter方法，也不用重写toString和hashcode的方法，节省开发时间用在业务逻辑层上，具体用法就是在Javabean中加入注解：@Setter、@Getter、@Log4j、@NoArgsConstructor和@AllArgsConstructor等，在项目编译的时候，自动在.class文件进行自动生成上述方法，极大简化了Javabean的代码量。但是要注意：使用此方法一定要保证Javabean中的所有变量全都符合CamelCase命名法，否则在编译时会发生错乱。

**2.10 Logback**

每一个开发工程师都希望自己设计的系统在无论多么恶劣的环境下，它的日志记录系统都能正常运行，以便于后期维护升级或者遇到Bug的时候能够有迹可循。Logback就可以很好的胜任这个工作，它相比较log4j有着更快的执行速度，还能对想要的日志进行过滤操作，节省了很大一部分的物理内存。本文所描述的应用平台详细地记录了每一次工作时，该平台出现的的所有异常和所有数据插入记录；根据日志，可以很快的知道哪一天因为应用平台的某种异常情况我们没有爬取新的数据，以便后期进行数据补充处理。

**2.11 Maven**

身为一个后端开发者，我已经不止一次地遇到这种情况：因为某些引入包的版本冲突所导致项目不能按照我们预期的结果进行运行。在没有使用Maven之前，我经常犯这样的错，包与包之间版本不一致导致报错或者同一类型的包被重复引用不同的版本导师编译失败。使用了Maven之后，这种问题完全不存在，完美适配目前这个提倡自动化配置、自动化发布的互联网时代。

**2.12 Websocket**

想要实现一个系统的前后端实时通信，离不开Websocket。在本文所描述的应用平台，本人使用Spring自带的Websocket模块，SockJS。前端可以直接通过“HTTP://”为头的URL进行监听后端服务端口，使用Logback的过滤机制，把所需要的信息进行过滤操作，实时和前端进行交互。使用Websocket的优点主要有下：

（1）节省带宽。持续轮询服务器端数据的方法是使用HTTP协议。头部信息量大，有效数据比例低。然而，使用websocket，头部信息非常小，有效数据的比例很高。

（2）没有浪费。轮询方法可以轮询10次，只会遇到服务器数据更新，那么前9次都是白色轮询，因为没有更改任何数据。WebSocket由服务器主动发回，它所来自的只是新数据。

（3）实时性，考虑到服务器的压力，不可能用轮询方法进行短时间的间隔，否则服务器的压力太大，所以轮询时间间隔比较长，几秒钟，设置超过10秒。WebSocket由服务器推送，实时性能最高。

**3 系统设计分析**

**3.1 系统设计思想**

本系统的功能模块分为横向功能和纵向功能。横向功能包括：用户管理、大屏展示、产品搜索、产品分析和数据获取等等大类功能；纵向功能细分为依次细分为：用户注册、用户登录，上月产品汇总数据、上月全国产品流通、上月热门产品汇总，所有产品查询、按类产品查询，单一产品搜索查询、单一产品数据挖掘，网络爬虫等模块功能。具体如图3-1所示。



图3-1 系统功能模块结构图

本大数据产品流通平台为基于SSM框架的Java Web项目，分成前端和后端两部分，并采用前后端分离编码、模块化编码。其中前端主要采用React、Javascript、Bootstrap、Axios、Ant Design、 Echarts、NPM等编码技术，后端主要采用SSM框架进行模块开发，主要开发技术包括：SSM、Java8、Druid、Lombok、Logback、WebSocket、Maven等等。本大数据应用平台开发严格按照GIT代码管理和阿里巴巴Java编码规范手册，具有非常强的可移植性、可扩展性和可维护性。

**3.2 可行性分析**

3.2.1 技术可行性

就目前现阶段来说，大数据开发技术已经不断成熟，很多公司都已经基于Hadoop和Impala成立专门部门进行大数据分析实践；至于B/S架构的Java Web网站也已在十几年前逐步发展成熟，所以，从技术层面上来说，这个大数据应用平台是完全可以实现的。

3.2.2 经济可行性

21世纪互联网时代，信息技术逐渐平民化，开发成本逐渐在降低；本文所描述的大数据应用平台所使用的技术都是开源技术，不需要支付任何费用，开发成本比较低。最重要的一点是，本问所描述的应用平台研发成功后能给带来巨大回报，能很好地解决“卖菜难，买菜贵”等社会民生问题，极大便利了我国广大人民的生活方式，加快了我国智慧城市化建设。

3.2.3 操作可行性

用户动态页面交互只有若干个简明按钮和特色图片，同时考虑了不识字和识字人群，操作简单，功能强大，界面美观，符合用户与人机之间的交互体验性。

3.2.4 组织可行性

本项目是一个基于SSM框架的前后端分离的Java Web网站，前端使用WebStorm开发，后端使用IDEA开发，数据库使用MySQL，服务器为Tomcat。基本架构合理，网站部署简单高效，接口联调规范，各接口之间的方法互不干扰影响，组织架构明了，后期可扩展性和可维护性较高，模块简单清晰明了。

**3.3 业务流程**

3.3.1 系统流程图



图3-2 系统流程图

对于本大数据应用平台的完整操作流程大致如下：首先用户应该注册账号进行登录，然后本平台会自动识别用户权限，如果是普通用户将会看到上个月产品流通和各类热销产品的汇总数据，进入详情页可以查询各类产品的历史价格行情；如果平台识别出该用户为会员用户，除了普通用户的基本功能之外，还可以进行产品查询搜索进行单一产品的历史价格查询和对该产品的未来数据分析预测；每种类型的用户都可以打印各类产品的价格图表方便今后直接查看。除此之外。管理员用户，负责每天的数据维护和更新还有用户权限的分配；但是因为目前该应用平台的推广和正处在成熟优化阶段，目前所有用户都免费享有会员用户的所有权限，所有历史数据也未在云端同步，各个用户也同时享有管理员权限，按天可以同步爬去所有数据。

3.3.2 数据流程图

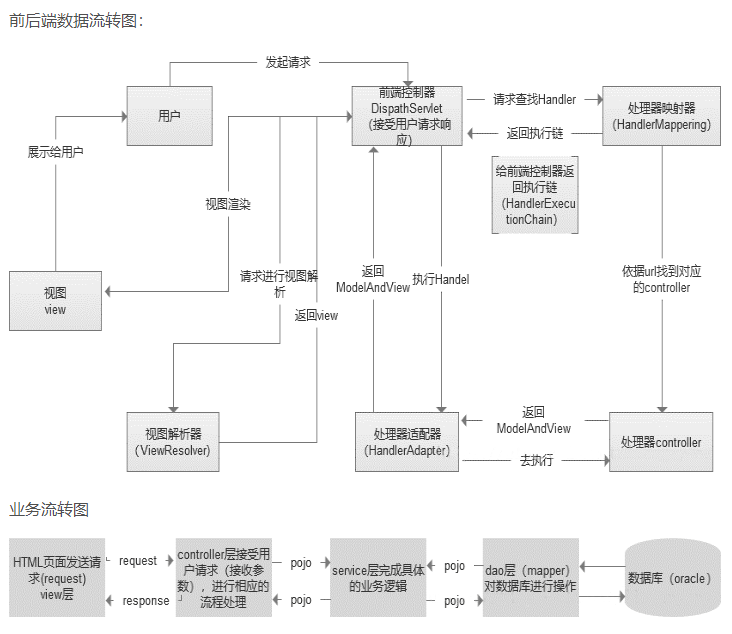


图3-3 数据流程图

基于SSM框架搭建的B/S架构Java Web网站，其最大的优势就是用户所需的配置尽可能地简单，对用户的隐私有着合法的保护；用户无需进行复杂繁琐的配置，只需要安装一个浏览器就能安全使用这个大数据应用平台，本网站的工作原理和前后端程序流转及业务流转都在上图3-3中具体体现出。

**3.4 用例分析**

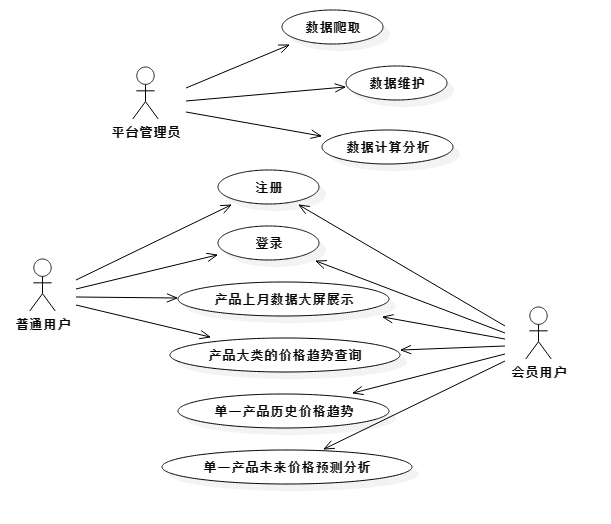


图3-4 用例分析

本应用平台的用例图如图3-4所示。平台管理员主要负责系统的数据维护，数据爬取以及数据的计算分析和权限管理等工作。普通用户只有注册、登录、查看产品上月数据大屏展示和产品大类的价格趋势查询等；会员用户除了拥有普通用户的功能以外，还拥有单一产品历史价格趋势分析功能和单一产品未来价格预测分析等功能。本应用平台的主要核心功能为：产品未来价格预测和数据爬取等。

**4 系统设计**

**4.1 非功能性需求**

一个系统或者应用平台仅仅有着强大的功能和美观的界面并不足以称作是一个优秀的系统或应用平台。没有优良的用户人机交互体验、流畅的用户操作和系统或应用平台平稳的运行环境作为基础，系统和应用平台的优越性将会毫无体验，用户自然也不会对你的产品产生兴趣。目前软件免费化已是市场趋势，想要让自己开发的软件走入平常百姓家，系统和应用平台的优越性自然少不了，本应用平台就实现了以下非功能性需求：

（1）所有功能模块响应时间极短。虽然系统本身的数据量比较大，但是经过大数据技术本身的优越性和对数据库进行了缓存读写，查数时间可以缩短到3秒以内。

（2）视图层和逻辑层代码模块分离，互不干扰、互不影响，对后期的系统功能扩展和平台维护提供便利。

（3）数据分页操作，极大方便了用户查询的定位。

（4）Spring注解自动注入，简化配置，高效开发。

（5）数据单独存放，定时备份，避免了数据视图不存在。

（6）系统界面科技感强，操作简便，适合各个年龄段人群使用。。

**4.2 功能性需求**

农产品流通和数据挖掘应用平台的核心模块为数据爬取模块和产品未来价格预测分析模块，除此之外还有登录模块，注册模块，大数据汇总模块，商品查询模块等。详细分析如下：

4.2.1 数据爬取

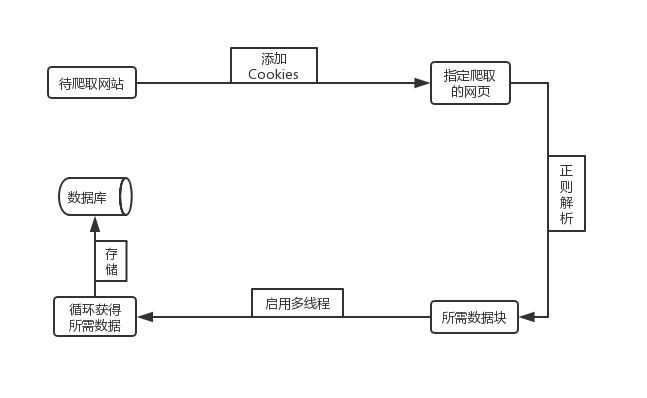
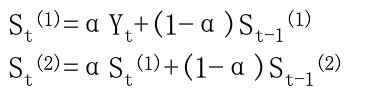


图4-1 用例分析

本系统所拥有的所有数据都是通过互联网对特定网站进行爬取获得，所使用的技术是比较基础的Java常用爬虫技术，具体原理如上图4-1所示。首先需要前期调研进行爬取网站选取，本文所描述系统的网站是由我的导师所提供的。确定了要爬取的网站之后就要先注册该网站的账号，为了防止该网站的反爬虫机制，需要将登录账号后的Cookie信息添加到HTTP请求的头部。接着就要对该网站的url进行不断请求解析，然后进行对所爬取html页面的正则匹配。只有符合我们要求的有效规范数据才会被存储在数据库中。因为数据量比较大，所以在这里使用了多线程并发存储数据。这样不仅保证了爬虫获取数据效率的高效，同时还能有效防止所登录Cookie信息的过期而导致数据获取失败。

4.2.2 产品未来价格预测分析

本文所描述的大数据应用平台的产品未来价格预测分析模块所应用的算法是目前市场上比较成熟的二次指数平滑算法，下面就是该算法的基本公式。



式中：

St(2)——第t周期的二次指数平滑值；

St(1)——第t周期的一次指数平滑值；

St-1(2)——第t-1周期的二次指数平滑值；

α——加权系数（也称为平滑系数）。

指数平滑法对实际序列具有平滑效果。加权系数（平滑系数）a越小，平滑效果越强，但对实际数据变化的响应越迟缓。在实时序列的线性变化部分，指数平滑值序列的滞后偏差程度随加权系数（平滑系数）a的增大而减小，但当时间序列的变化呈线性趋势时，使用第一个指数平滑方法进行预测时，仍然存在明显的滞后偏差。因此，也需要修改。修正后的方法也是在第一次指数平滑和第二次指数平滑的基础上，利用滞后偏差规律找出曲线的发展方向和趋势，建立线性趋势预测模型，使生成的预测数据更准确、更客观。下图4-2就是对二次指数平滑算法的原理描述。

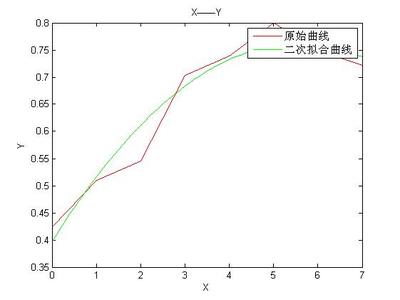


图4-2 二次指数平滑算法原理图

4.2.3 大数据汇总

本文所描述的大数据应用平台的大部分功能都是对大数据的图表展示，这就离不开百度的开源项目技术——ECharts。百度开发的ECharts（Enterprise Charts）拥有诸多特性：它提供了丰富的可视化图标类型：柱状图、折线图、散点图、饼图等自定义个性化图表和混合图表；能满足用户动态数据实时传输；前端渲染的效果可以自定义展示；最主要的是兼容当前所有主流浏览器，这一特性将很好地满足本系统在前期运营的大力推广，用户不用额外安装任何软件，只需要一个域名就可以在任何地方访问，方便用户在买菜时实时比对商品价格，对本系统无疑是锦上添花。

4.2.4 商品查询

对于本系统的查询功能，在刚开始的需求设计仅仅考虑了每个查询接口的具体设计，但在实际开发中发现这样操作的话最后的接口数目会超过100个。为了避免代码的冗余和功能的重复，本人在系统完成后期进行了代码重构，将所有查询接口整合到一个查询接口，所有入参被写成一个类，这样就能很好地对接口进行复用，优化代码效率。并且，当查询单一产品的时候用Mybatis和Java进行重构产品索引方法，使得产品搜索索引能在极短时间内进行响应。最终，商品查询的接口方法可以完美兼容任何个数和类型输入参数的查询请求。

Mybatis实现树状索引方案：通过ResultMap可以实现将数据的查询结果映射为复杂类型的pojo，在这里我将这个结果转换成了树的数据类型。

Java实现树状索引方案：Java实现主要通过Java8的Stream流式编程中的map和collect方法，具体原理不再敖述，实例如下：

nameValue.stream()

.map(Product::getSizeType)

.map(sizeType -> new MergeEnumProduct(ProductConstants.SIZE\_TYPE, sizeType))

.collect(Collectors.toCollection(LinkedList::new));

4.2.5 登录和注册

像普通系统的设计一样，本系统也拥有注册和登录的界面，注册和登录的合法性判定也和其他系统相类似。但是本系统追求以动态界面吸引刺激用户注册登录使用，所以主要从这两个界面的简约、动态特效来设计。另外，为了大数据展示效果达到最佳，本系统的导航栏也进行了可伸缩滑动设计。

## 4.3 数据库概念结构设计

本系统的数据库部分采用PowerDesigner和Navicat Premium进行建模，在建立数据模型之前绘制了数据ER图。

数据ER图:基于本人的思维方式，我在数据库建表前进行了思维可视化，其数据关系具体如图4-2所示。

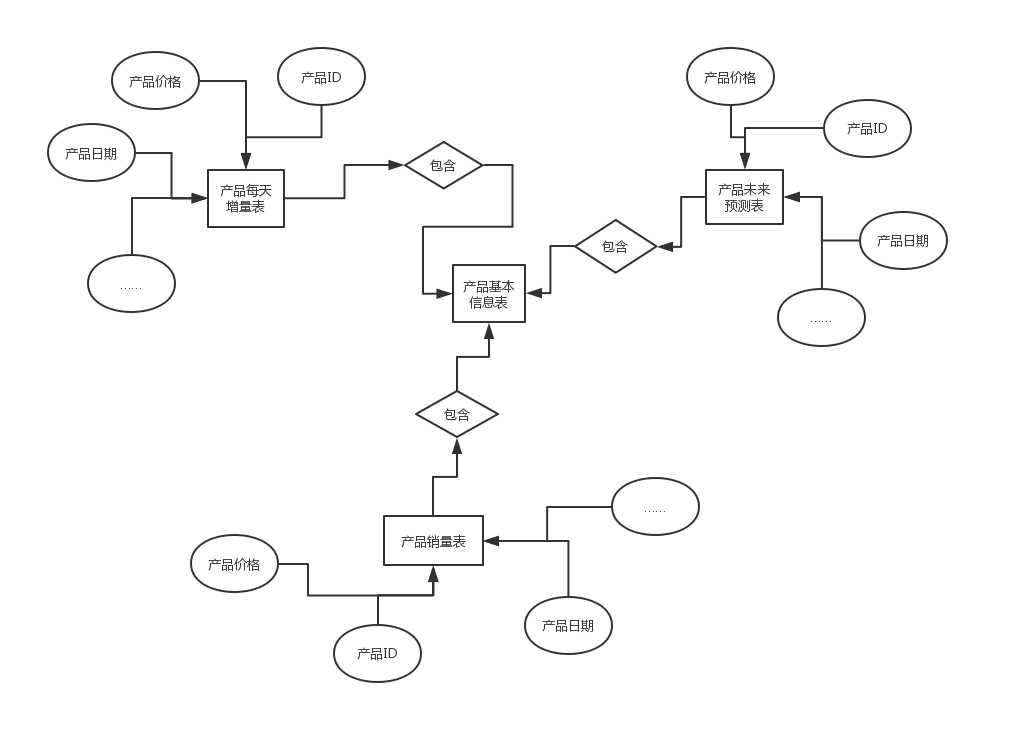


图4-2 数据ER图

## 4.4 数据库的逻辑结构设计

由于本系统的业务性并不十分复杂，主要建立了5张数据表，表之间相互独立，表内有唯一的主键，具体表内字段属性值和字段类型如图4-5所示。

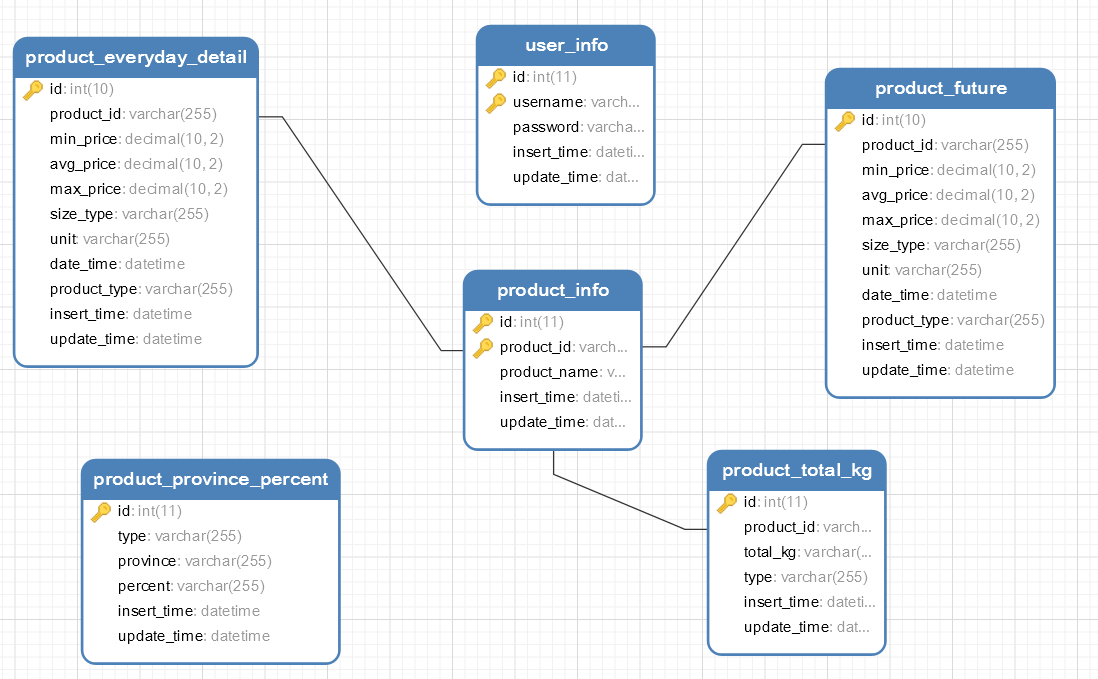


图4-5 数据库逻辑结构图

逻辑结构图中的数据库表如表4-1所示。

表4-1 数据库表

|  |  |
| --- | --- |
| 表名 | 备注 |
| PRODUCT\_EVERYDAY\_DETAIL | 主要存储产品每天价格行情明细 |
| PRODUCT\_FUTURE | 主要存储产品未来七天的价格行情明细 |
| PRODUCT\_INFO | 主要存储产品的基本信息 |
| USER\_INFO | 主要存储用户的信息明细和权限配置 |
| PRODUCT\_PROVINCE\_PERCENT | 主要存储各类产品在各个省市的流通明细 |
| PRODUCT\_TOTAL\_KG | 主要存储各类热门产品信息明细 |

**5 系统实现**

**5.1 系统流程图**

实际所开发的大数据应用平台如图5-1所示，本应用平台除了正常系统应有的用户注册登录以及CURD模块之外，还有数据爬虫、数据预测分析和权限后台管理模块（一期平台为了推广使用，暂未开放），并支持产品的精准搜索查询和图表的打印。



图5-1 系统流程图

**5.2 业务代码执行流程**

当用户在前端界面点击本应用平台的某个功能按钮，如用户点击了“产品价格预测”模块，具体如图5-2所示。首先SSM框架进行前端的所有请求拦截，这一步主要在Controller中实现，然后业务流程跳转到业务逻辑层，进行各种逻辑判断和工具类方法的调用，最后通过Mybatis进行与数据库的交互，进行数据分析预测。具体详见本文3.3.2的数据流程图部分。

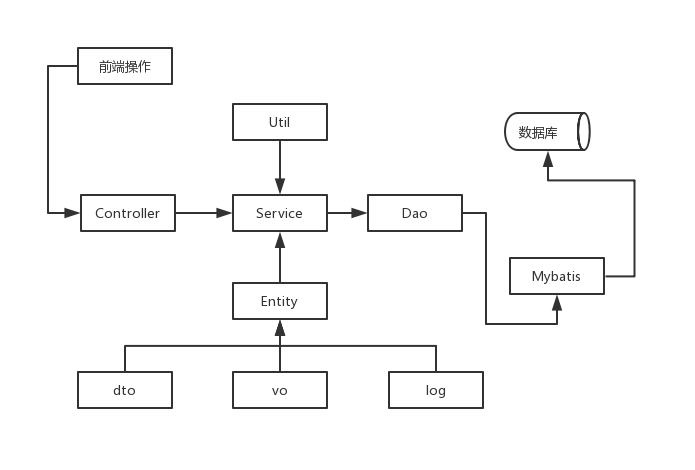


图5-2 代码执行流程图

**5.3 项目文件结构**

本项目采用前后端分离的技术，后端基于Maven进行自动化搭建，其具体的项目文件结构如图5-3所示。前端基于NPM进行React框架的搭建，其具体的项目文件结构如图5-4所示。

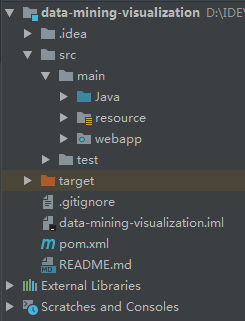
****

图5-3 项目文件结构图（后端）

本系统后端的所有主要核心代码都在Java文件夹中，资源配置在resource文件夹中，所有的测试用例和方法在test文件夹中。使用Maven最大的好处就是不需要进行jar包的单独下载和引入，本系统所有的jar包都在pom.xml文件中，管理规范，简便开发步骤和提高开发时效。

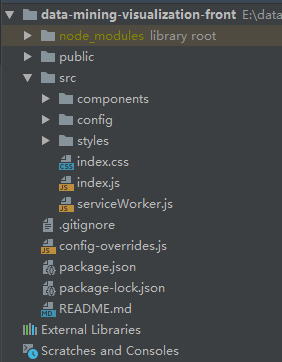


图5-4 项目文件结构图（前端）

本系统前端的主要核心代码在components文件夹中，所有的依赖包都通过NPM进行安装引入在node\_modules文件夹中，package.json文件中存储了各种依赖包的版本，统一管理方便后期版本升级等。

Java文件夹主要存放了所编写的后端核心代码，com.zafu.nichang为本人的包名，config中编写了主要的全局配置文件，比如跨域请求和Websocket；constants文件夹主要存放本项目的一些常量的申明；controller中编写了本项目的所有接口，并进行不同URL请求的拦截；entity中是一些实体对象的一些扩展，在这里统一规范了返回前端请求数据的主要结构；enums进行全局的一些枚举值的申明；exception对本项目的所有异常进行了统一规范的处理；filter主要进行前后端日志格式的交互规范和本地后端控制台日志的保存存储；handler中的主要模块是使用Spring中的Websocket功能，搭配后端日志进行实时前端推送消息；mapper主要编写Mybatis和数据库交互的接口方法；model申明了本项目中的所有实体类、抽象类；service是本项目的主要逻辑层，包含各种接口方法和其实例函数；util主要进行数据爬取模块的编写，解析网页、多线程高并发进行爬取等操作。本项目后端核心代码的文件结构如图5-5所示。

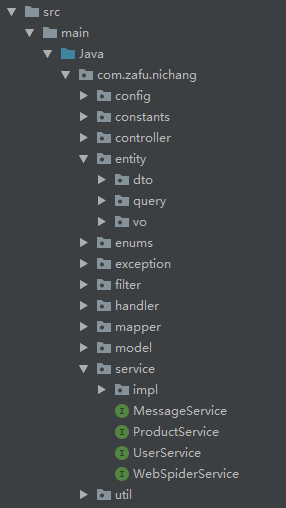


图5-5 核心代码目录结构图（后端）

本项目的前端核心代码目录结构主要如图5-6所示，主要在components文件夹中进行编写各个界面；在config文件夹中主要进行全局的IP和URL配置和后端返回data数据格式的处理。

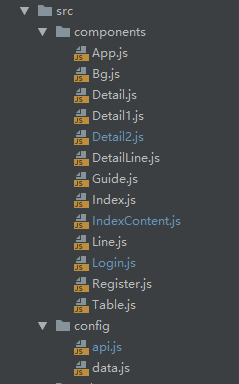


图5-6 核心代码目录结构图（前端）

**5.4 项目命名规范**

本项目后端代码参考《阿里巴巴Java开发手册》进行标准化格式规范编码。前端采用标准化前端编码规范：HTML规范、CSS规范、JavaScript规范以及各类文件命名规范。

**5.5 用户模块**

本应用平台的主界面主要由三部分组成，标题部分、动态背景以及主要主体。其中动态背景是通过构图函数和时间延时函数进行组合所构建出来的。最终形成一个从左下方往右上方在一定范围内随机速度运行的动态背景壁纸。

5.6.1 登录界面

系统的登录界面的主体是两个输入框、保持登录按钮、忘记密码链接和登录注册按钮。输入正确的用户名和密码，点击“登录”按钮后，后台根据登录用户的用户名密码进行登录权限认证，认证成功后跳转到该用户所具有的功能模块，否则登录失败，显示登录失败具体信息。目前，为了应用平台的推广和使用，后台数据库默认添加所有新注册用户均为管理员用户。具体如图5-7所示。

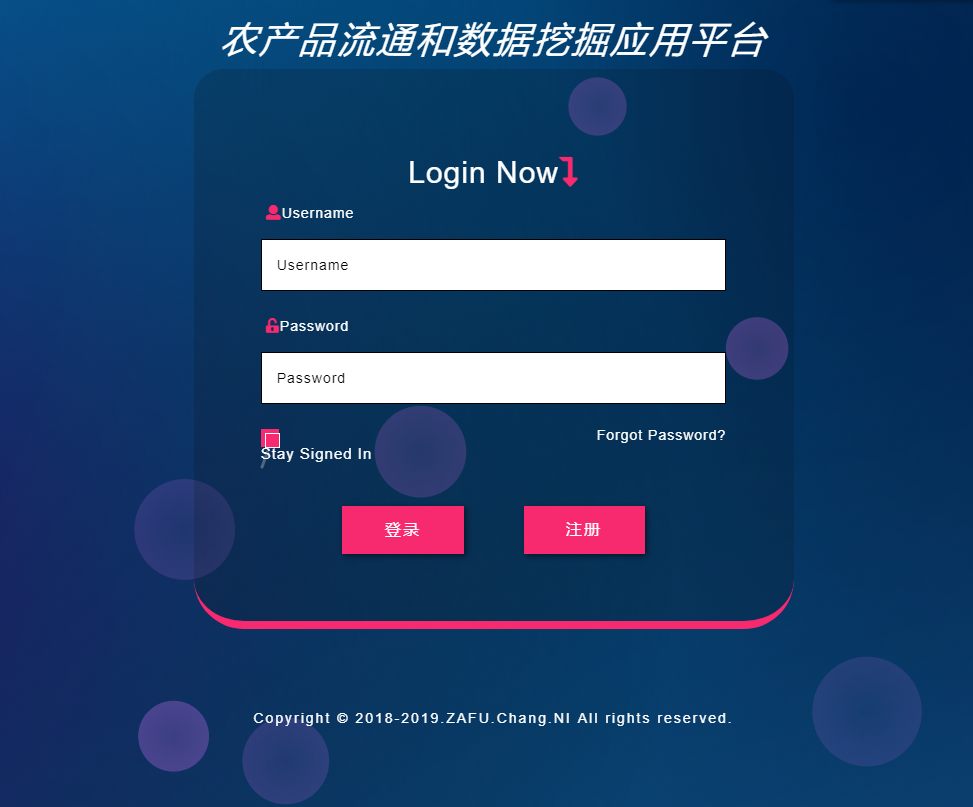
****

图5-7 用户登录界面图

5.6.2 注册界面

在登录界面点击“注册”按钮即可跳转到新用户注册界面。用户输入自己设定的用户名密码后进行前端的非法字符判定，判定为合法后请求到后端进行用户名的合法性判定，这里主要进行用户名是否被占用等。验证通过后跳转到登录界面，否则让该用户重新注册。具体界面详情见图5-8所示。

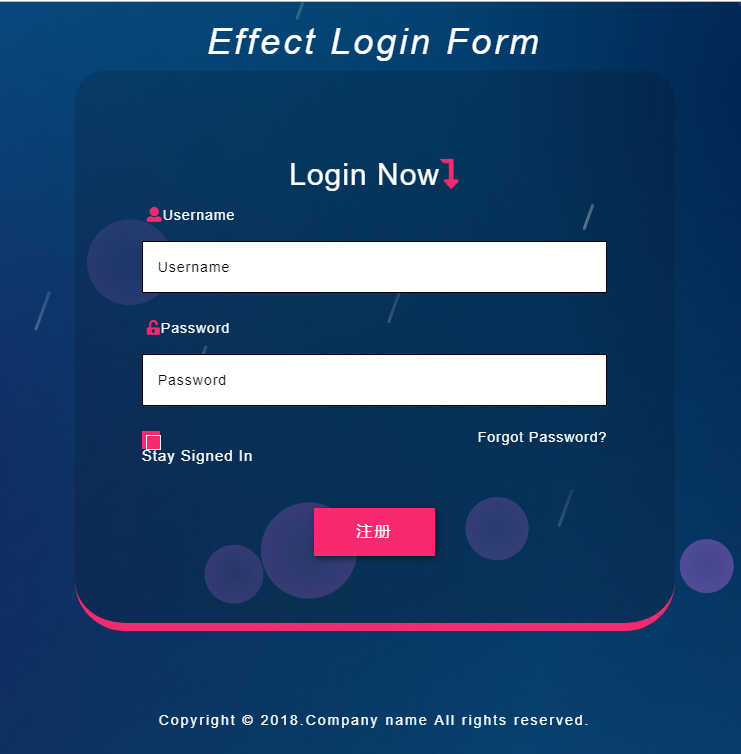


图5-8 用户注册界面图

5.6.3 导航栏

在用户成功登录系统后，在本系统的左侧有一个可以伸缩的导航栏。这个可伸缩导航栏既能满足全屏首页界面的美观，又能满足用户正常随意跳转本系统的任一功能模块。具体界面详情见图5-9所示。



图5-8 应用平台导航栏界面图

**5.7 首页模块**

本模块页面如图5-12所示。

****

图5-12 货运管理首页

5.7.1 上月统计

点击“购销合同”功能按钮，出现下图，此页面提供对合同的查看编辑功能，上报至报运部门，打印输出合同表，合同归档等功能。这些操作按钮由权限控制显示，只有拥有对应权限的才会显示。页面如图5-13所示。



图5-13 购销合同首页

5.7.2 农产品分类统计

点击左侧“出货表”功能按钮，前台会展示一个新页面，如图5-18所示。点击“打印”，传递给后台一个船期，后台根据此日期查询数据库，然后打印输出表格。

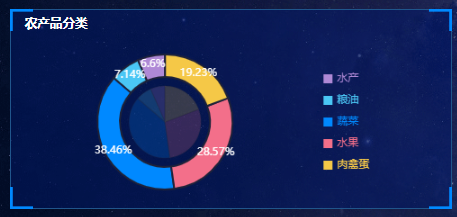


图5-18 出货表首页图

5.7.3 农产品明细统计

该模块提供了关于报运单的查看功能，修改功能，删除功能以及上报至装箱部门。

点击“出口报运”功能按钮，前台页面会改变。如图5-19所示。



图5-19 出口报运首页

首先根据ID，更新报运单的状态为“已装箱”，然后将相关的合同ID置于页面上，用户填写相关信息后即可完成装箱。

**5.8 产品价格详情模块**

本模块主要是一些统计信息，包括生产厂家的销售情况、产品的销售排行记录、系统访问压力图、以及简单的财务统计。页面如图5-28所示。

****

图5-28 统计分析首页

5.8.1 图片按钮分类

此页面采用了amCharts的饼形图来显示销售情况，其数据来源于数据库的查询，然后通过预置flash动画读取xml配置文件，然后在支持Adobe Flash Player的浏览器中可以显示出来，如图5-29所示。

****

图5-29 厂家销售情况图

5.8.2 价格比较图

此页面采用amCharts的柱形图来直观的展示不同产品的销售情况，该柱形图也属于flash动图数据来源于contract\_product\_c数据表，图5-30展示了两种货物的销售排行情况。

****

图5-30 产品销售排行图

5.8.3 分页表格

此页面的折线图的数据来源于login\_log\_p数据表，用户每次登陆的时候，都会往此表写入数据，用户记录不同时刻的访问量。如图5-31所示。

****

图5-31 系统访问压力图

**5.9 未来数据预测模块**

此模块是记录了本系统的一些基础信息，包括数据字典以及厂家信息，其中厂家信息支持报表导出。页面如图5-33所示。

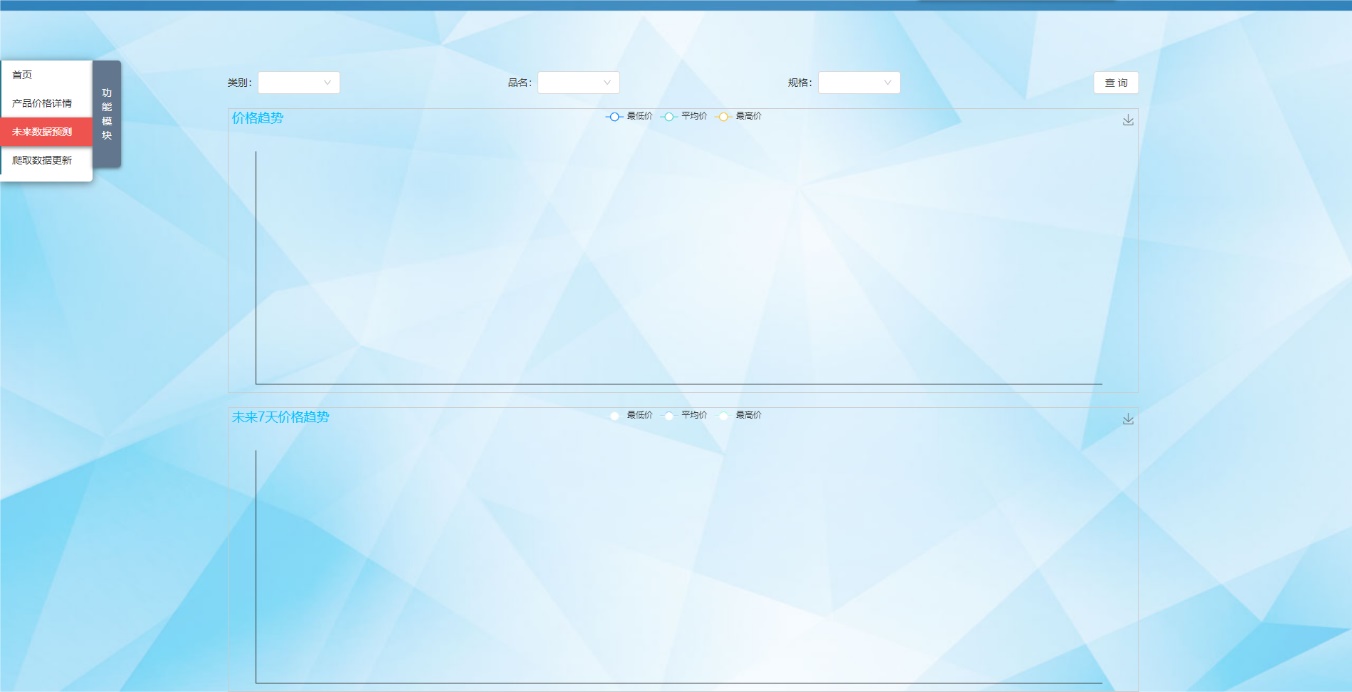
****

图5-33 基础信息首页

5.9.1 树状索引搜索查询

此页面展示了系统中的数据字典，数据字典的作用是限定了系统中某些字段的选择，如人员级别、附件类型等，具体如图5-34所示。

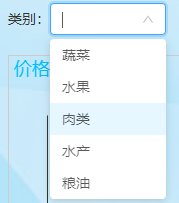
****

图5-34 数据字典树

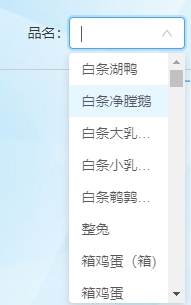
****

图5-34 数据字典树

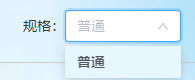
****

图5-34 数据字典树

5.9.2 历史价格走势与未来趋势预测

本页面记录了所有与本公司合作过的厂家的基本信息，支持报表导出通讯录。页面如图5-35所示。

****

图5-35 生产厂家首页

本处使用了开源项目pageHelper进行分页。需在pom文件中添加如下依赖。

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

<version>5.1.2</version>

</dependency>

后台默认分页以5条记录为一页，在select框提供了“5”，“10”，“20”三种类型。



图5-36 分页选择

页面中图5-36部分做了验证处理，当没有前一页或者是已经是第一页的时候，图片变成灰色，并且点击无跳转效果。

**5.10 爬取数据更新模块**

此模块是记录了本系统的一些基础信息，包括数据字典以及厂家信息，其中厂家信息支持报表导出。页面如图5-33所示。

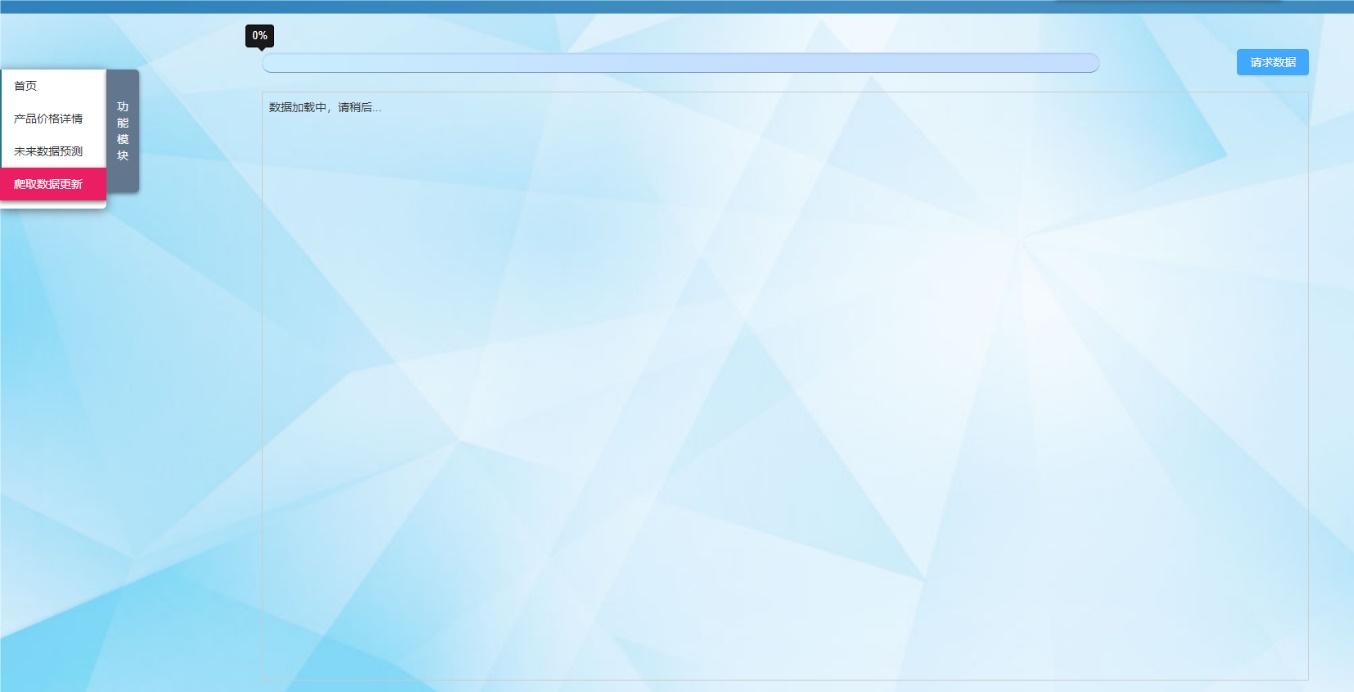
****

图5-33 基础信息首页

5.10.1 爬取进度条

此页面展示了系统中的数据字典，数据字典的作用是限定了系统中某些字段的选择，如人员级别、附件类型等，具体如图5-34所示。

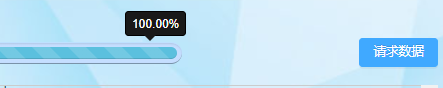


图5-34 数据字典树

5.10.2 爬取信息交互

本页面记录了所有与本公司合作过的厂家的基本信息，支持报表导出通讯录。页面如图5-35所示。

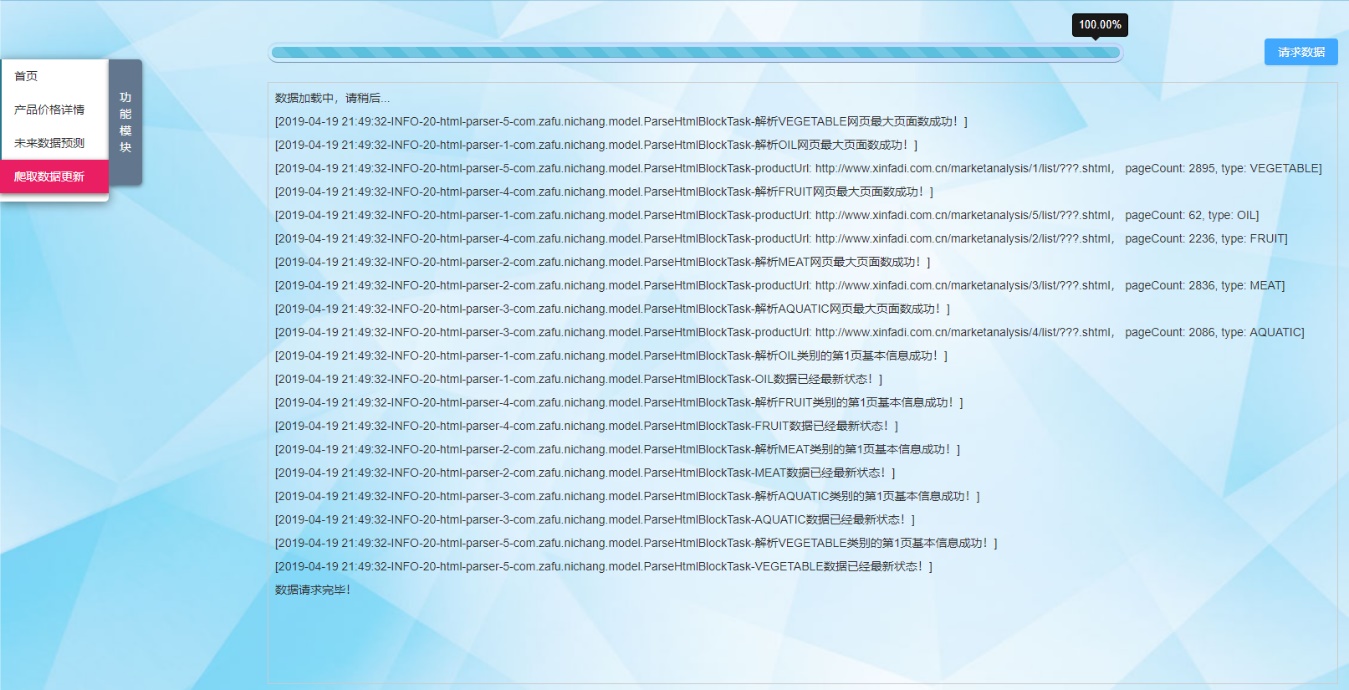
****

图5-35 生产厂家首页

本处使用了开源项目pageHelper进行分页。需在pom文件中添加如下依赖。

。

**6 总结**

。

**参考文献**

[1] Craig Walls.Spring实战[M]. 北京：人民邮电出版社，2015：123-245.

[2] Simplifying Enterprise Java Applications with the Spring Framework[M].Apress: 2008-06:10-45.

[3] Leslie M. Fine. Selling and sales management[J]. Business Horizons, 2007,50(3):2-8.

[4] H.Sebastian Heese,Jayashankar M. Swaminathan. Inventory and sales effort management under unobservable lost sales[J]. European Journal of Operational Research, 2010,207(3):12-20.

[5] 申和平. 设计模式在企业管理系统中的研究与应用[D]. 大连理工大学，2007.

[6] 冯金辉. J2EE设计模式比较及应用实现研究[D]. 浙江大学，2004.

[7] 郑希彬. Web应用系统设计模式的研究与应用[D]. 山东科技大学，2005.

[8] 石明辉. 基于JAVA的多数据库中间件的设计与实现[D]. 上海交通大学，2011.

[9] 余冬梅, 张秋余, 张涛, 王海军.JAVA在WEB数据库中间件中的应用[J]. 计算机工程与应

用，2002(17):208-210.

[10] 徐雯, 高建华. 基于Spring MVC及MyBatis的Web应用框架研究[J]. 微型电脑应用， 2012，28(07):1-4-30.

[11] 李霞. MVC设计模式的原理与实现[D]. 吉林大学，2004.

[12] 刘增杰. MySQL5.5从零开始学[M].北京：清华大学出版社，2012-03：23-45.

[13] 许晓斌. Maven实战[M].北京：机械工业出版社，2011-04：123-205.

[14] 李刚. Spring+Mybatis企业应用实战[M].北京：电子工业出版社，2017-07：78-103.

[15] 许令波. 深入分析Java Web技术内幕[M].北京：电子工业出版社，2014-09：203-245.

[16] 陈静红. 服装企业销售管理信息系统的分析与设计[D].云南大学，2013.

[17] 杨丽辉. 基于J2EE架构MVC模式的房地产销售管理信息系统的设计与实现[D].吉林学，2004.

**致谢**

。