基于网络爬虫的跨平台商品比价系统设计与实现

张芷悠，侯卓轩，毕晓琳

（广东东软学院，广东 佛山 528225）

摘 要 在激烈的市场竞争中，消费者购物比价需求日益凸显，消费者费大量的时间、精力在各大平台自我进行性价比分析的问题也逐渐显露，有些零售平台挂虚假低价链接引流，现有的比价系统不足以鉴别出链接信息的真实性。本文介绍了应用Django、Vue3和Selenium三种框架技术实现的一个网购比价系统，该系统通过改善比价系统的信息筛选和甄别功能，排除虚假低价引流商品，进行精准比价，从而节省时间、提升购物满意度。

关键词 网络爬虫；跨平台比价；Django；Vue3；Selenium

中图分类号：TP3 **文献标志码**：A **文章编号**：2097-3365(2024)08-0013-03

随着信息技术的迅猛发展，电子商务行业在全球范围内取得了长足进步，线上购物已成为消费者日常生活的重要组成部分。在激烈的市场竞争中，电商平台种类繁多，对于同一种产品，由于不同的电子商务平台抽取商家的佣金、提供给商家的补贴和促销力度不一致，导致销售价格不一致，消费者往往是不清楚的，需要花费时间和精力手动跨平台比价。消费者购物比价需求日益凸显，他们渴望能够快速、准确地获取商品信息，以做出更明智的购买决策。然而，现有的比价系统存在诸多不足，比如有些零售平台挂虚假低价链接引流，现有的比价系统不足以鉴别出链接信息的真实性。本系统通过优化比价系统的设计与功能，旨在提高消费者的购物体验。改善比价系统的信息筛选和甄别功能，排除虚假低价引流商品，使消费者能够快速获取真正需要的商品信息，进行精准比价，从而节省时间、提升购物满意度。

1 技术实现

1.1 Django后端框架

Django是一个高级Python Web框架，它鼓励快速开发和干净、务实的设计。Django具有强大的数据库抽象层，即Django ORM（对象关系映射），它使得数据库操作变得简单直观，减少了开发者编写SQL语句的繁琐工作[1]。Django内置了强大的模板系统，可以快速构建复杂的Web页面，并支持自定义模板标签和过滤器，提高了代码的可重用性和可维护性。

1.2 Vue3前端框架

Vue3是一个构建用户界面的渐进式框架，采用了响应式的数据绑定机制，使得数据变化能够自动驱动视图更新，减少了手动操作DOM的繁琐性，Vue3还提供了丰富的组件化开发模式，使得开发者能够将页面拆分成多个独立的组件，提高了代码的可维护性和复用性[2]。此外，Vue3还优化了性能表现，通过虚拟DOM技术和更高效的渲染算法，提升了页面的渲染速度和用户体验。

1.3 Selenium爬虫框架

Selenium是一个用于自动化Web浏览器操作的工具，它通过驱动浏览器、模拟用户的操作来访问网页并获取所需的信息。Selenium的工作原理主要基于Web Driver接口，通过浏览器驱动与浏览器进行通信，实现自动化操作。自动化爬虫技术广泛应用于数据分析、信息采集、搜索引擎等领域[3]。在电商平台比价系统中，爬虫技术被用于抓取各个电商平台的商品价格、用户评价等信息，为比价系统提供数据支持。

2 需求分析与功能设计

2.1 需求分析

本比价系统的目标用户群体主要定位于学生党、上班族等具有比价需求的消费人群。他们通常希望能够在购物前快速、准确地获取到在不同电商平台中同类商品的价格、性能等信息，以便做出明智的购买决策。

功能需求方面，本比价系统致力于提供一站式服务。首先，系统需要能够抓取各个电商平台的商品信息，确保用户能够实时获取最新的价格数据。其次，系统需对抓取到的信息进行筛选和过滤，确保信息的真实性和有效性，避免用户受到虚假信息的误导。非功能性需求方面，本比价系统的响应时间应不超过300毫秒，以确保用户操作时能够获得流畅且及时的反馈。

通过满足这些需求，本比价系统将能够帮助用户快速、准确地获取商品信息，提高购物效率，满足用户的比价需求。

2.2 功能设计

用户使用系统流程如图1。

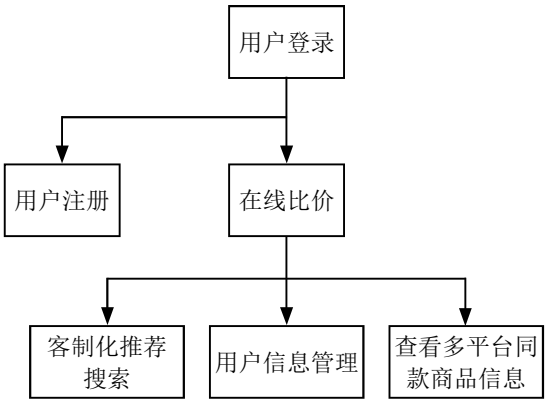

图1 用户流程图

用户注册账户后登录账号，输入搜索关键词，勾选多个目标平台，搜索目标网站的同类商品，比价后点击心仪商品跳转到目标网站进行购物。用户登录后，可以编辑系统在这个过程中存储的该用户历史搜索记录，用于客制化搜索推荐。

用户通过用户界面与系统交互，用户首先需要通过提供个人信息并设置登录凭证进行账户注册。注册成功后，用户可以使用注册的凭证登录到系统中。登录后，用户通过输入关键词进行搜索。系统接收到这些关键词后，调用爬虫程序，用户可以通过勾选操作选择多个目标平台，系统会根据用户的勾选情况，在多个平台上执行搜索任务。为了排除虚假低价引流商品，系统还提供了信息筛选功能。搜索完成后，系统会返回搜索结果，展示给用户同类商品的信息。

当用户找到心仪的商品后，可以点击商品链接，系统会进行跳转操作，将用户重定向到目标网站的购物页面，从而完成购物流程的引导。

此外，在用户登录状态下，系统还会记录用户的搜索历史。这些历史记录会被存储在数据库中，以便后续分析。用户可以通过编辑功能，对搜索历史进行查看和删除等操作，以便更好地管理自己的搜索记录。

3 系统设计与实现

3.1 系统架构设计与实现

3.1.1 后端数据处理与存储

在本系统的运作流程中，后端一旦接收到前端传递的数据，首要任务是进行数据预处理与清洗。这一过程旨在剔除冗余或不必要的信息，并对数据进行必要的格式调整以适应系统要求。随后，经过清洗的数据将被妥善地存储至数据库中，以保证数据的持久保存。为了实现高效且可靠的数据管理，本系统采用了Django框架[4]提供的ORM（对象关系映射）机制，这一机制极大地简化了数据操作的复杂性，确保了数据的完整性与可靠性。通过这一系列操作，本系统不仅提高了数据处理效率，还为用户提供了更为安全稳定的数据服务。

3.1.2 后端API设计与数据传输

本系统后端设计并开发了符合RESTful风格的API接口，定义了清晰的URL路由和请求方法，提供对数据的增删改查等操作，再通过API接口将数据传输给前端，使用JSON格式进行数据的传输，确保数据的易读性和可解析性[5]。后端实现对API的权限控制，确保只有经过身份验证和授权的用户才能访问和操作数据，提高系统的安全性。

3.1.3 前后端交互与页面展示

在系统的运作中，前端通过AJAX技术与后端进行深度的信息交换[6]。这一过程不仅涵盖了请求的发送，还涉及后端响应数据的接收，从而确保了前后端之间数据传递的高效与准确。一旦前端成功接收到后端发送的数据，便会借助先进的前端框架对这些数据进行处理，进一步将其转化为用户易于理解和操作的页面内容，从而以更加直观和友好的方式将信息展示给用户。这样的设计极大地提升了用户体验，并增强了系统的整体交互性。

3.1.4 爬虫自动化搜索与数据抓取

如图2，本系统通过配置与初始化Selenium Web Driver的Chrome浏览器实例，建立了一个能够连接到在指定端口上运行的Chrome调试协议的会话。该会话的创建使得系统操控浏览器行为，进行自动化操作。基于用户输入的关键词和平台名称，系统通过遍历当前所有打开的浏览器窗口句柄，在多个浏览器标签页中查找并切换到包含目标平台名称的页面。

一旦切换至目标页面，系统便利用BeautifulSoup库对HTML页面进行深度解析。通过遍历网页文档中的标签树，系统能够根据标签的属性、层级关系以及特定结构来精准定位所需信息的位置。同时，结合元素的CLASS_NAME和XPath选择器，系统能够精准地定位页面上的特定元素，模拟用户行为，实现自动化搜索功能。

本系统还集成了Selenium的自动滚动和等待机制。通过模拟用户滚动页面的行为，系统能够确保页面中的所有动态内容得以加载和展示。这一特性保证了系统能够实时处理和抓取页面中的动态内容，有效避免

了因内容加载不全而导致的数据丢失或错误。

此外，本系统会对爬取到的原始数据进行一系列的数据清洗和格式化处理。这一步骤包括去除无关字符、处理特殊格式、转换数据类型，最终将清洗后的数据存储到字典中，便于后续处理和存储。

3.1.5 数据筛选与离群值剔除

本研究旨在精准剔除那些误导消费者的虚假低价引流商品。为实现这一目标，我们创新性地运用了一种基于统计学的离群值检测方法。首先，我们计算了同类商品价格的算术平均数与标准差，进而设定了一个阈值，即超过平均价格正负三倍标准差的数据将被判定为离群值。紧接着，我们利用这些统计信息，识别并剔除了那些异常高昂或异常低廉的价格数据，确保数据的真实性和可靠性。在数据处理过程中，我们借助了 NumPy 这一强大的科学计算库，高效地完成了数据筛选工作，最终将符合正常价格区间的商品精准地呈现给前端用户。

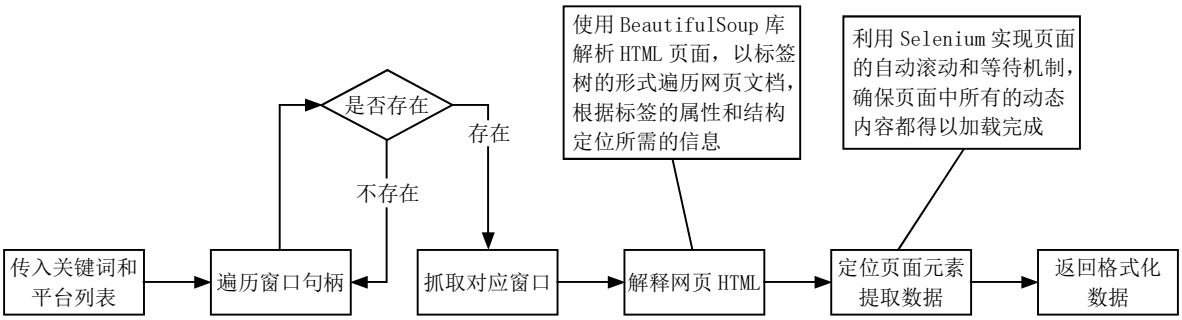

图 2 selenium 爬虫工作流程

3.2 数据库设计

根据主要业务，本小节对商品比价系统中的两个主要数据表结构进行介绍。

3.2.1 商品信息表

如表 1，该表主要介绍商品基础信息。

表 1 商品基础信息

字段名	字段描述	数据类型
product_id	商品 ID	int
product_description	商品描述	varchar
product_name	商品名称	varchar
product_url	商品图片链接	varchar

3.2.2 平台信息表

如表 2，该表主要介绍平台信息。

表 2 平台信息

字段名	字段描述	数据类型
platform_id	平台 ID	int
platform_name	平台名称	varchar
platform_url	平台链接	varchar

4 结论

本研究成功实现了一个搜索、比价和剔除虚假引流商品等功能的比价系统。用户能够方便地进行账号管理，并通过搜索功能快速获取商品信息，相比传统的比价系统，多了信息筛选的功能。通过应用 Selenium 框架，系统能准确抓取电商平台的数据，为用户提供了实时、准确的商品信息，确保了比价的可靠性。系统经过多次测试，表现出良好的性能稳定性。综上所述，本研究成功开发了一个功能完善、操作便捷的网络购物比价系统，并充分利用了 Django、Vue3 以及 Selenium 的技术优势。系统实现成果显著，技术应用效果良好，为网络购物比价提供了有效的技术支持。未来，我们将进一步优化系统性能，提升用户体验，推动网络购物比价系统的进一步发展。

参考文献：

[1] 张同硕，廖明军，张荣华，等．基于 Django 的交通事故数据可视分析系统设计与实现 [J]. 软件导刊，2023,22(07):112-117.

[2] 顾鲍尔 .Vue.js+Django 高性能全栈论道 [M]. 北京：清华大学出版社，2022.

[3] 王帅 . 基于 Selenium 框架的反爬虫程序设计与实现 [J]. 信息记录材料，2023,24(06):86-88.

[4] 同 [3].

[5] 陈锐，何华军，王辰 . 基于 RESTful 接口的基础数据对接设计 [J]. 电脑编程技巧与维护，2022(09):122-124,137.

[6] 杨兴海，杨林鹏，杨兴荣，等 . 一种针对 AJAX 技术动态网络数据进行数据获取的方法 :CN202211483800.4[P]. [2024-06-15].