

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 秦际州

学 院: 计算机科学与技术学院

系: 软件工程

专 业: 软件工程

学 号: 3220105929

指导教师: 胡晓军

2024 年 11 月 10 日

浙江大学实验报告

课程名称: B/S 体系软件设计 实验类型: 综合

实验名称: 多平台商品比价系统

学生姓名: 秦际州 专业: 软件工程 学号: 3220105929

同组学生姓名: 无 指导老师: 胡晓军

实验地点: 线上 实验日期: 2024 年 11 月 10 日

多平台商品比价系统 —— Price Sleuth 需求规格和系统设计说明书

目录

1 引言	4
1.1 项目背景	4
1.2 编写目的	4
1.3 文档描述	4
1.4 术语解释	5
2 总体描述	6
2.1 项目信息介绍	6
2.2 用户定义以及特征	6
2.3 设计实现的约束限制	7
2.4 假设和依赖	8
2.5 开发计划	9
3 需求分析	10

3.1 功能需求	10
3.2 需求优先级分析	13
3.3 非功能性需求	13
3.4 数据字典	16
4 系统设计	17
4.1 项目技术栈介绍	17
4.2 系统架构设计	20
4.3 数据库设计	21
4.4 系统接口设计	23
4.5 运行设计	25
5 系统界面原型	26
5.1 登录界面	27
5.2 用户注册	27
5.3 搜索界面（主界面）	28
5.4 购物车界面	28
5.5 商品固定界面（筛选排序）	29
5.6 商品固定详情页面	29
5.7 商品历史价格查询	30
5.8 手机端适配主页面	30
5.9 手机端登录界面	31
5.10 手机端注册界面	31
6 附录	32

1 引言

1.1 项目背景

本项目是浙江大学 2024-2025 学年秋冬学期 B/S 体系设计课程作业，即设计开发一个多平台商品比价系统。按照课程提出的要求，本项目需要实现一个具有用户登录注册功能、存储记录用户账号信息、允许使用多个平台进行商品搜索拆查询操作，购物车记录商品并显示历史价格等功能的同时适配于 PC 端和手机端的 Web 项目。Web 技术允许自行进行技术选型，要求独立完成、界面友好、提供必要的过程文档。最终使用 docker 将项目打包并提交使用。

1.2 编写目的

为了确保项目需求的明确性和可执行性，本文档旨在细化项目的具体要求，并针对项目开发过程中的关键方面进行详细规划。这包括技术选择、底层逻辑设计、系统架构搭建等。同时，本文档也将对项目的开发周期进行合理安排，并确保项目进度得到有效跟进。通过这种方式，我们能够确保项目从概念到实施的每一步都经过精心规划和执行。

1.3 文档描述

本文档旨在作为项目的正式需求规格和系统设计指南。文档内容被严格划分为六个核心部分：引言、总体描述、需求分析、系统设计、系统界面原型以及附录。

引言部分对本文档的编制背景、目的和结构进行了严谨的说明，并对文档中涉及的关键术语进行了定义，以便为潜在的读者提供清晰的阅读指导。

在总体描述部分，我们详细列出了项目的基本信息，包括项目名称、开发团队、目标用户群体等，并明确了项目在设计和实施过程中所面临的限制条件、假设和依赖关系。此外，还对项目的总体开发流程和关键里程碑进行了规划，以确保项目按计划推进。

需求分析部分深入探讨了项目的功能性和非功能性需求。功能性需求部分通过系统用例图和基于用例的详细描述，明确了项目的功能需求，并对其优先级进行了评估。非功能性需求部分则从性能、并发控制、用户界面的可视化、系统安全性和可维护性等多个维度提出了具体要求，并基于这些需求初步构建了数据字典，为后续的数据库设计和开发工作奠定基础。

系统设计部分对项目的技术选型、总体架构、数据库设计、系统接口和运行逻辑进行了全面构思。在技术选型确定后，对前端、后端及服务器相关的关键技术进行了概述。系统架构图详细展示了系统各模块之间的相互联系，并提出了前端、后端及运行环境的具体设计要求。数据库设计作为项目的重要组成部分，我们提出了详细的初步构想。同时，系统接口设计和运行设计也从前后端交互以及用户与前端交互的角度，将需求具体化，确保了项目的可操作性。

系统界面原型部分展示了项目前期开发的重要阶段性成果，包括系统的界面设计，为后续的详细设计和开发提供了参考。

附录部分则包含了其他必要的补充信息，以确保文档的完整性和项目的全面性，为项目的顺利实施提供额外的支持和指导。

1.4 术语解释

首先先对于项目本身使用过程中的术语进行解释。本项目是一个多平台的商品比价系统，提供给用户查询比价商品的功能，并能让用户获取到历史价格走向。于是在这里对几个相对重要的术语进行解释：

术语	术语解释
用户	本系统的使用者包括所有需要比较不同商家商品价格的人，能够通过本系统能够快速获取市场信息，做出明智的购物决策。
商品	商品是指在本比价系统中列出的各种待售物品，它们具有明确的市场价格和规格，供用户比较和选择，以便找到性价比最高的购买选项。
通知	通知，在此背景下特指降价提醒，是指系统自动向用户发送的实时更新信息，告知他们关注的商品价格变动情况，尤其是当商品

	价格下降到用户设定的预期值时，帮助用户把握最佳购买时机。
购物车	是指用户用于暂存和组织他们感兴趣的商品的虚拟空间。用户可以将多个商品添加到购物车中，方便比较、管理。

表 1 系统术语解释表

除此之外再解释几个在报告中出现的一些专业名词，方便读者在阅读报告是能理解这些术语表达的内容：

术语	术语解释
用例图	用例图是一种 UML（统一建模语言）图表，它描述了系统的功能如何与外部用户（参与者）交互，以满足特定的业务需求。用例图通过展示参与者和用例之间的关系，提供了系统功能和用户需求之间交互的高层次视图。
ER 图	ER 图（实体-关系图）是一种数据模型图，用于描述数据库的结构。它通过图形化的方式展示了实体（数据表）之间的关系，包括实体的属性、实体之间的联系以及实体集合之间的关联性。ER 图帮助设计者理解和组织复杂的数据关系，是数据库设计和系统分析的重要工具。

表 2 文档术语解释表

2 总体描述

2.1 项目信息介绍

项目名称	多平台商品比价系统 —— Price Sleuth
项目开发者	秦际州，3220105929
目标受众	用户
任务发起者	浙江大学 B/S 体系软件设计课程组

表 3 项目信息介绍表

2.2 用户定义以及特征

在第一部分中名词解释中提及到了用户，在这个部分将详细对用户这一目标

受众的特征进行分析解释，方便在后文使用“用户”一词来代指具有一下特征的人：

用户类	主要特征
用户	进行账号的注册和登陆操作
	对个人账号的个人信息进行修改调整
	使用多平台搜索功能进行商品搜索
	将有意向的商品添加进购物车中
	固定不同平台的不同商品进行实时价格比较
	查看个人账户下的购物车之中的商品、以及价格变化图
	能够接收到系统发送的降价提醒
	能够使用手机端访问本系统

表 4 用户特征表

2.3 设计实现的约束限制

本项目的运行开发过程需要建立在部分的约束条件下进行，为了保持项目的正常运行。

项目的成功部署和运行依赖于一个合适的服务器环境，这包括特定的操作系统、硬件配置和必需的软件依赖。我们必须确保服务器环境满足这些要求，以便项目能够顺利执行。简而言之，服务器必须具备项目运行所需的所有条件，以保证其稳定性和性能。

为了确保代码的统一性、清晰度和易于维护，必须遵循一套明确的编码标准，这包括命名规则、缩进风格、注释要求以及函数和类的设计原则。遵循这些规范有助于团队成员之间的有效沟通，同时提升代码的整体质量和维护效率。简而言之，统一的代码规范是提高团队协作和代码质量的关键。

在工具版本选择上，我们必须依据项目依赖的软件组件或框架的要求以及兼容性考虑，来确定并使用特定的开发工具、框架或库版本。这些版本限制确保了

项目能够与依赖的组件无缝集成，并且能够满足特定的功能需求。简而言之，遵循这些版本规范是为了确保项目的兼容性和依赖关系得到妥善处理。

于是经过考虑，决定使用如下的硬件配置，软件技术栈以及版本控制工具作为文本项目的约束：

操作系统	Windows 10 或者以上版本等主流操作系统
硬件配置	具备 2 核 CPU，4GB 内存以上的主流设备
前端框架	使用 Vue.js 3.0 作为前端框架
后端框架	使用 Django 作为后端框架
数据库	使用 MySQL5.7 以上版本作为数据库
爬虫工具	使用 selenium 作为爬虫工具
版本控制工具	Git

表 5 框架依赖表

2.4 假设和依赖

本项目的可实行性建立在以下的假设和依赖的基础上：

编号	假设与依赖描述
AS-1	假设系统运行的操作系统上的所有最新安全更新和补丁都已经安装。
AS-2	项目依赖于特定版本的第三方库，以确保功能的正确实现和性能的最优化。
AS-3	假设服务器将提供足够的硬件支持以满足项目运行的性能要求。
AS-4	假设有一个稳定的网络连接，带宽至少为 100Mbps
AS-5	假设数据库服务器将 24/7 全天候运行，并且有专业的 DBA 团队进行维护。
AS-6	假设无不可控的突发意外情况

表 6 假设依赖表

2.5 开发计划

本项目的截止日期设定在 2023-2024 学年冬季学期的第 8 周。为了按时完成，我们规划了以下时间表：

在冬季学期的第 2 周结束前，我们将根据项目的整体需求，建立前端框架，完成前端静态页面的初步开发，并实现大部分系统界面原型。在此过程中，我们将充分考虑技术限制，确保项目在满足所有需求的同时，保持其可行性和可操作性，并对需求进行细化，制定设计方案，编制项目需求规格和系统设计文档。

在冬季学期的第 4 周结束前，我们将完成前端开发工作，并向后台团队提出接口需求。

在冬季学期的第 5 周结束前，我们将初步完成数据库和后端的开发工作，并实现前端所需的接口。若遇到技术上难以实现的接口，我们将根据实际情况调整前端逻辑。

在冬季学期的第 6 周结束前，我们将完成服务器的搭建工作，并完成后台开发，随后对整个项目进行集成测试。

在冬季学期的第 8 周结束前，我们将整理完成所有项目相关文档。

3 需求分析

3.1 功能需求

3.1.1 系统用例图

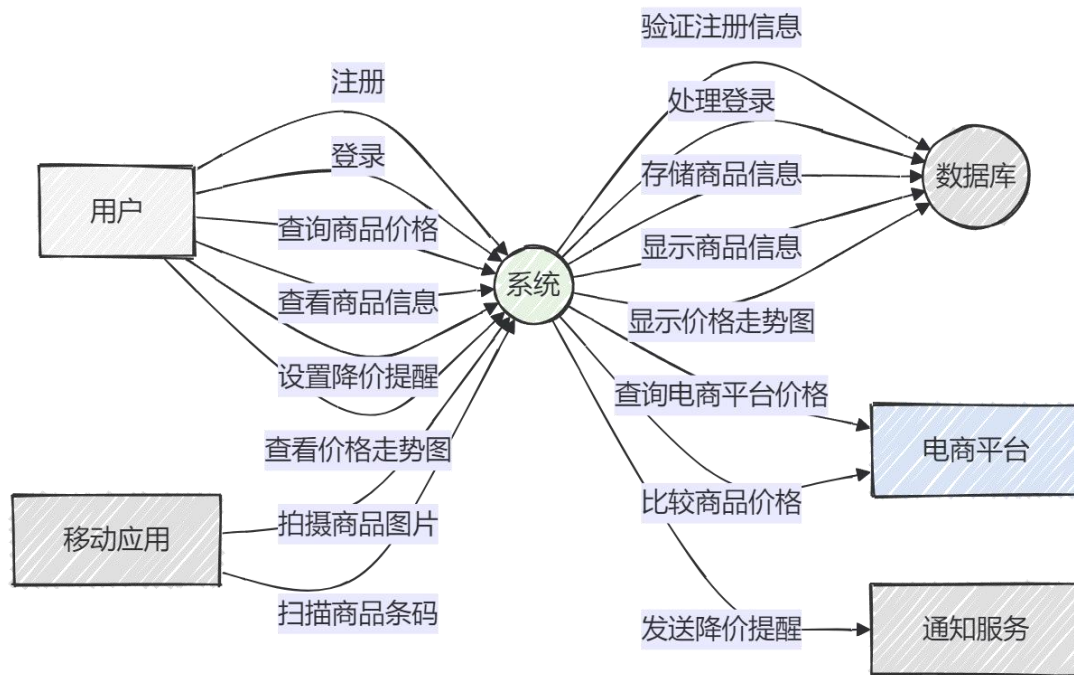


图 1 系统总用例图

3.1.2 用户账户注册

用户注册功能允许新用户创建账户并加入系统。在注册过程中，用户需要提供一系列必要的个人信息，包括但不限于用户名、密码（至少 6 个字符）和电子邮箱地址。系统将对这些信息进行验证，确保密码符合安全要求，电子邮箱格式正确，并且用户名和电子邮箱地址在系统中是唯一的，以避免重复注册。注册成功后，用户将收到确认信息，并可以使用新创建的账户登录系统。

3.1.3 用户登录功能

用户登录功能允许已注册用户通过提供他们的用户名和密码来访问系统。系统将验证这些凭据是否与数据库中存储的信息匹配（数据库中的密码信息均使用

加密后的密文存储，避免攻击）。如果验证成功，用户将被授权进入系统，并能够访问他们权限范围内的功能和资源。如果登录失败，系统将提供错误提示，要求用户重新输入正确的凭据。

3.1.4 用户查看个人信息

用户可以查看自己的个人信息，包括但不限于用户名、注册邮箱、联系方式等。这个功能允许用户在个人中心查看和管理与自己账户相关的基本信息，增加了用户对自己数据的控制和透明度。

3.1.5 用户修改个人信息

用户可以修改自己的个人信息，如密码更新、邮箱更改、联系方式更新等。系统将提供安全的修改流程，确保用户信息的安全性和准确性。在修改敏感信息如密码或邮箱时，系统可能需要额外的验证步骤，重复输入原密码等操作。

3.1.6 商品价格查询

系统提供基于商品名称的实时价格查询服务，用户可以输入商品名称，系统将自动在主流电商平台上搜索并返回价格信息。为了提高搜索的准确性和效率，系统将对商品名称进行分词处理。此外，系统能够处理和展示来自多个电商平台的查询结果，即使某些平台要求用户登录后才能访问价格信息。

用户直接在搜索框中输入需要搜索商品的名称或者种类等关键词，点击搜索按钮之后就可以进行搜索。

3.1.7 多平台价格比较

用户可以比较至少两个以上主流电商平台（例如京东、苏宁、唯品会等）的商品价格。系统将从这些平台获取价格信息，并提供一个直观的比较界面，帮助用户做出购买决策。

用户在各个平台搜索结果显示的商品中能选择固定某些商品，可以打开比价页面进行商品比价，全面横向比较各商品价格等特征。

3.1.8 商品收藏

商品收藏功能允许用户将他们感兴趣的商品保存到个人账户中，以便日后查看或购买。这个功能增强了用户的购物体验，使用户能够快速访问他们之前浏览过的商品，同时也方便用户进行商品比较和决策。

用户在浏览商品信息时，可以选择将商品添加到他们的收藏列表中。系统将提供明显的“收藏”按钮或链接，用户只需一键点击即可收藏商品。

用户可以在个人中心或通过特定的收藏页面查看他们收藏的所有商品。这个列表将展示商品的详细信息，包括名称、图片、当前价格等，并保持更新。

用户可以管理他们的收藏列表，包括移除不再感兴趣的商品或查看商品的实时价格变动。系统将提供直观的界面，让用户轻松管理收藏的商品。

3.1.9 商品库管理

系统将建立一个商品库，用于存储和管理商品的详细信息，包括名称、多级品类、规格、条码和图片等。这个商品库将支持高效的商品信息检索，方便用户快速找到他们感兴趣的商品。

凡是用户搜索后添加到购物车中的商品，均会自动添加到商品库中。方便后续的价格查询与跟进。

3.1.10 商品信息展示

系统将提供一个用户界面，用于展示商品的详细信息，并以图表形式展示商品的历史价格走势。这将帮助用户了解商品价格的变动趋势，并做出更明智的购买选择。

3.1.11 降价提醒功能

用户可以为特定商品设置降价提醒，系统将定时查询这些商品的最新价格，并在检测到价格下降时通过邮件或 App 推送等方式通知用户。

3.1.12 移动端适配

系统将优化其界面和功能，以确保在手机浏览器和微信等应用内置浏览器中提供良好的用户体验。

3.2 需求优先级分析

依据风险价值四象限矩阵，对各需求用例做出以下评级：

编号	用例名称	风险	价值	优先级
SE-01	用户注册	中	高	中
SE-02	用户登录	中	高	高
SE-03	用户修改个人信息	中	低	低
SE-04	商品价格查询	高	高	高
SE-05	多平台比价	低	高	高
SE-06	商品收藏	中	高	高
SE-07	商品信息展示	中	中	中
SE-08	降价提示	中	中	中

表 7 需求优先级表

3.3 非功能性需求

3.3.1 性能需求

系统的设计注重稳定性和可靠性，确保在一周内系统维护和重启的次数不超过一次，以最小化对用户的影响。系统将集成实时监控功能，能够迅速检测出任何非正常情况，比如与外部设备的通信中断或数据库连接失败，并及时向用户反馈，避免用户长时间等待。此外，系统将实现对主流 Web 浏览器的全面支持，包括 Firefox、Chrome、Edge 等，确保无论用户使用何种浏览器，都能获得一致且正确的显示和执行效果。

在响应速度方面，系统将保证对用户操作的即时反应，Web 页面的响应时间在正常条件下不超过 1 秒，涉及数据库的操作不超过 2 秒。即使在用户访问量激增的高峰时段，系统也能保证 Web 响应时间在 2 秒以内，数据库操作在 5 秒以内，以维持良好的用户体验。

同时，系统将具备高并发处理能力，至少能够支持 100 人的并发访问，确保在用户量激增时系统性能不受影响，保持流畅运行。这些性能需求的设定旨在为用户提供一个高效、稳定且响应迅速的网络服务环境。

3.3.2 安全性需求

为了确保系统的安全性，我们将采取一系列措施来保护用户数据和系统资源。首先，我们将实施严格的保密性措施，包括对用户名和密码进行加密存储，以及在敏感数据交换时使用加密技术，以防止未经授权的访问和管理员的越权操作。此外，我们将防止 SQL 注入、密码破解和会话劫持等安全威胁，确保用户登录过程的安全。

在数据完整性方面，我们将通过数据库约束和关键操作的限制来防止非法用户对数据的无意或恶意修改、插入和删除，确保数据的完整性和防止数据丢失。

为了防止攻击，系统将采取安全措施，以抵御 SQL 注入、密码破解和会话劫持等攻击。

最后，个人账户信息将存储在安全的数据库中，并采取适当的加密措施，以保护数据的机密性和完整性。同时，我们将对用户输入的数据进行严格验证和过滤，防止恶意用户通过注入攻击执行恶意数据库查询。通过这些综合性的安全措施，我们将确保系统的安全性和用户数据的保护。

3.3.3 可视化需求

系统的可视化设计旨在提供一个清晰、直观的用户体验。用户界面将引导用户轻松完成注册和登录流程，通过明确的指示和即时反馈来确保用户输入的信息符合要求。在用户注册时，界面会明确标识必填字段，并在输入不符合规范时提供实时反馈。

商品查询界面将展示商品的详细信息，并通过图表直观地呈现商品的历史价格走势，使用户能够迅速把握价格变化，辅助购买决策。收藏功能则通过一个易于管理的列表，让用户便捷地查看、添加或删除收藏商品，同时保持商品信息的同步更新。

针对移动端，界面设计考虑到手机屏幕的尺寸和操作习惯，确保在手机浏览器和各种应用内置浏览器中都能提供良好的用户体验。这意味着界面元素设计得足够大，便于触摸操作，布局自适应不同设备的屏幕尺寸。

3.3.4 可维护需求

在开发一个成熟的系统时，确保其可维护性是至关重要的。为此，系统模块的设计应遵循高内聚、低耦合的原则，这意味着每个模块应有明确的职责和功能，同时尽量减少模块间的依赖。这样的设计有助于简化系统的理解和后续的维护工作。

文档的完备性、清晰性和可读性对软件的可维护性起着决定性作用。因此，系统开发过程中应产生一系列相关文档，包括简明一致的书写风格，以提高系统的可读性和可修改性。这些文档将帮助操作人员快速查阅功能，同时使维护人员能够系统地掌握架构。交付时，文档应齐全、详尽，并符合相关标准。

编程风格也是保证系统可维护性的关键因素。程序应包含详细的注释和统一的编程格式，确保结构清晰、注释明确，以便于调试和测试人员快速定位错误。具体要求包括避免使用含糊的代码，采用有意义的数据和过程命名，以及适当的注释。此外，应采用模块化、结构化的设计方法，并确保文档的正确性、一致性和完整性。

单元测试是确保系统质量的另一个重要方面。核心模块应配备单元测试，以保证在交互时各子模块和系统整体的正常运作。可测试性的要求包括模块化和良好的结构，以及系统的可理解性和可靠性。系统应能显示任意的中间结果，并以清晰的描述方式说明输出，根据要求显示所有输入。

3.4 数据字典

完成需求分析之后，为了文档的可读性，以及方便后续继续的数据库设计，在此对本系统需要使用维护的变量数据进行整理设计展示。在本次设计中主要会使用到以下的一些数据项：

编号	数据名称	类型	长度	备注
DI-01	用户 ID	整数	6	自动生成
DI-02	用户名	字符串	大于 6	用户设定
DI-03	密码	字符串	大于 6	用户设定，密文存储
DI-04	电子邮箱	字符串	非空	需要满足邮箱的格式
DI-05	手机号	字符串	11	用户设定，非必要
DI-06	商品 ID	整数	6	自动生成
DI-07	商品编码	字符串	非空	电商平台的商品唯一编码
DI-08	商品平台	字符串	2	商品来源平台名称
DI-09	商品名称	字符串	非空	从电商平台获取
DI-10	商品链接	字符串	非空	从电商平台获取
DI-11	商品图片	字符串	非空	从电商平台获取
DI-12	商品价格	字符串	非空	从电商平台获取
DI-13	商品历史价格 1	字符串	非空	从电商平台获取
DI-14	商品历史价格 2	字符串	非空	从电商平台获取
DI-15	商品历史价格 3	字符串	非空	从电商平台获取
DI-16	商品历史价格 4	字符串	非空	从电商平台获取
DI-17	价格队列编号	整数	6	记录此时数组中实时价格编号

表 8 数据字典表

4 系统设计

4.1 项目技术栈介绍

4.1.1 技术栈选择

项目采用前后端分离的 Web 开发技术，在后端编写一系列接口供前端调用，实现前后端的协作。在这个模块明确一下具体的技术栈。

前端：使用 Vue 3 作为主要框架，npm 作为包管理工具，Element Plus 作为组件库，结合 axios、echarts、pinia、vue-router 等工具实现前后端交互等功能。

后端：使用 Django 作为主要框架，并调用 selenium 等库实现网页爬虫功能。

数据库：使用 MySQL 数据库。

部署：使用 docker 部署



图 2 技术栈图

4.1.2 技术栈介绍

Vue 3 是 Vue.js 的下一代版本，这是一个广受欢迎的渐进式 JavaScript 框架，用于构建用户界面。Vue 3 引入了许多新特性和改进，包括 Composition API、响应式系统的重写、性能优化、更好的类型推断以及对 TypeScript 的更好支持。它旨在提供更小的打包体积、更快的虚拟 DOM 重写、更灵活的组件组合以及更强大的工具和生态系统支持，使得开发者能够更高效地构建现代化的 Web 应用。

npm (Node Package Manager) 是一个广泛使用的 JavaScript 包管理器，它允许开发者共享和管理代码，自动化执行常见的开发任务，并且是 Node.js 平台的默认包管理工具，提供了一个庞大的开源库生态系统，使得开发者可以轻松地安装、共享和管理项目依赖。

Element Plus 是一个专为 Vue 3 设计的桌面端组件库，它是 Element UI 的升级版本，提供了丰富的组件和强大的功能。这个库继承了 Element UI 的设计理念，保持了组件库的易用性和丰富性，同时进行了现代化的升级，支持 Vue 3 的 Composition API，使得组件更加灵活和易用。Element Plus 提供了现代的 API 设计，丰富的组件库，自定义能力强，一致性设计，良好的文档支持，以及强大的社区支持。它完全基于 Vue 3 构建，支持 Composition API 和 TypeScript，提供了大量的桌面端组件，涵盖了几乎所有的交互需求，并且高度可定制，支持多语言环境。Element Plus 还以其优秀的文档和示例代码著称，帮助开发者快速上手和使用组件。通过 Element Plus，开发者可以快速构建出界面原型，减少开发时间，同时确保应用的响应性和交互质量。

Axios 是一个基于 Promise 的 HTTP 客户端，用于浏览器和 node.js 环境，它提供了一个简单易用的 API，用于向 HTTP 服务器发送请求，并处理返回的响应，支持请求和响应的拦截、转换数据格式、客户端支持防御 XSRF 等特性，是现代 Web 开发中常用的数据交互工具。

ECharts 是一个由百度开源的数据可视化库，它提供了丰富的图表类型和灵活的配置选项，使得开发者能够轻松地在 Web 页面上展示复杂的数据可视化效果，包括但不限于折线图、柱状图、饼图、散点图、雷达图、地图、K 线图等，同时支持数据的动态交互和实时更新，是构建数据驱动应用的强大工具。

Pinia 是 Vue.js 的官方状态管理库，专为 Vue 3 设计，同时兼容 Vue 2。它提供了一种简单直观的方式来管理应用程序的状态，允许跨组件或页面共享状态。Pinia 的 API 设计灵感来源于 Vuex，但对于熟悉 Vuex 的开发者来说，Pinia 带来了一些改进，例如去掉了 Mutations，直接在 Actions 中修改状态，无论是同步还是异步操作都更加直观和灵活。Pinia 还提供了插件系统，用于扩展和增强其功能，如添加中间件、持久化存储、调试工具等。它支持 TypeScript，提供了类型安全的 API 和开发体验，使得状态管理更加健壮和易于维护。此外，Pinia 支持服务端渲染，并且与 Vue DevTools 集成，方便开发者调试状态变化，提高开发效率。

Vue Router 是 Vue.js 官方的路由管理器，它与 Vue.js 的核心深度集成，允

许开发者在构建单页面应用（SPA）时，通过声明式配置和组件化的方式，轻松实现页面路由的控制和导航，支持嵌套路由、路由参数、查询、通配符等高级功能，并提供了丰富的 API 和钩子函数，以实现复杂的路由逻辑和页面间的交互，是构建现代 Web 应用中不可或缺的组件。

Django 是一个高级的 Python Web 框架，它鼓励快速开发和干净、实用的设计。Django 自带了许多内置功能，如用户认证、内容管理、站点地图和缓存，以及一个强大的 ORM（对象关系映射），使得开发者可以专注于编写代码而不是重新发明轮子。它遵循“不要重复自己”（DRY）的原则，提供了一个高层次的抽象，以提高开发效率和减少错误，同时保持代码的可维护性和可扩展性。Django 以其“电池包含”哲学而闻名，旨在帮助开发者快速启动和运行项目，同时提供了足够的灵活性来定制和扩展以满足各种项目需求。

Selenium 是一个开源的自动化测试工具，专门用于 Web 应用程序的测试，支持多种编程语言绑定，并能够模拟用户在浏览器中的行为，如点击、输入、导航等。它提供了一套完整的 API 来编写测试脚本，能够实现跨浏览器和跨平台的自动化测试，支持 Chrome、Firefox、Internet Explorer 等多种浏览器，是进行 Web 自动化测试和爬虫开发中广泛使用的工具。

MySQL 是一个流行的开源关系型数据库管理系统（RDBMS），使用结构化查询语言（SQL）进行数据库管理，以其高性能、可靠性和易用性而闻名。它支持多种操作系统，提供了广泛的应用程序接口（API），允许开发者轻松地集成 MySQL 到各种应用程序中，适用于各种规模的应用，从小型应用到大型企业级应用。MySQL 以其强大的功能和灵活性，成为许多网站和 Web 应用后端数据库的首选。

Docker 是一个开源的应用容器引擎，它允许开发者打包他们的应用以及应用的依赖环境到一个轻量级、可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。Docker 容器可以运行在多种平台上，包括物理机、虚拟机、数据中心和云平台，提供了快速的交付和部署应用的能力，同时保证了应用的一致性和安全性。

4.2 系统架构设计

4.2.1 前端架构图

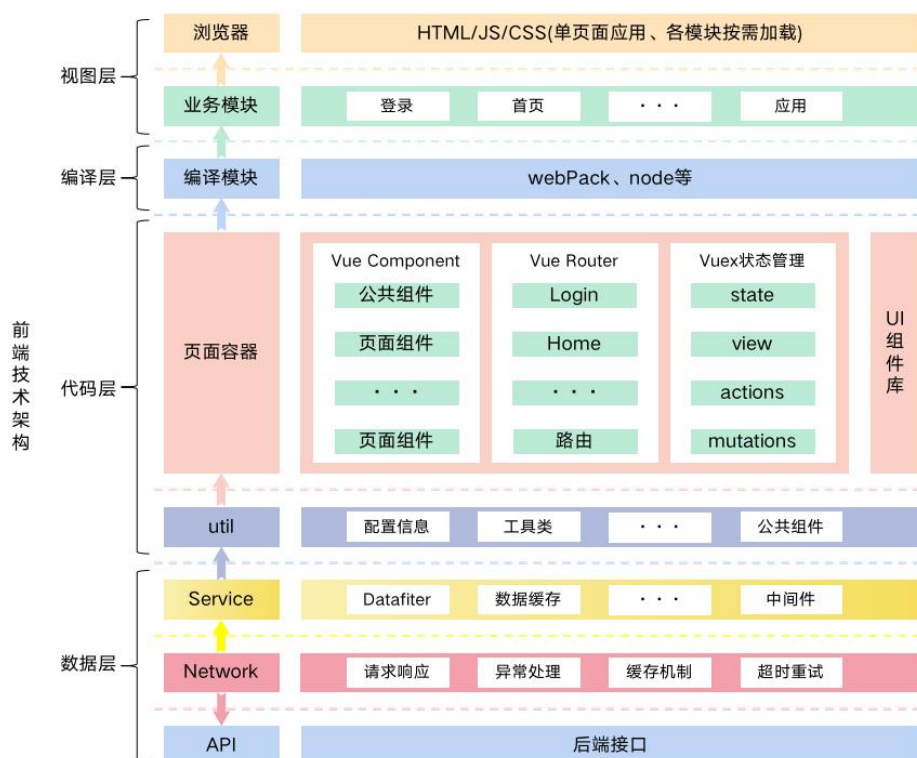


图 3 前端架构图

4.2.2 后端架构图

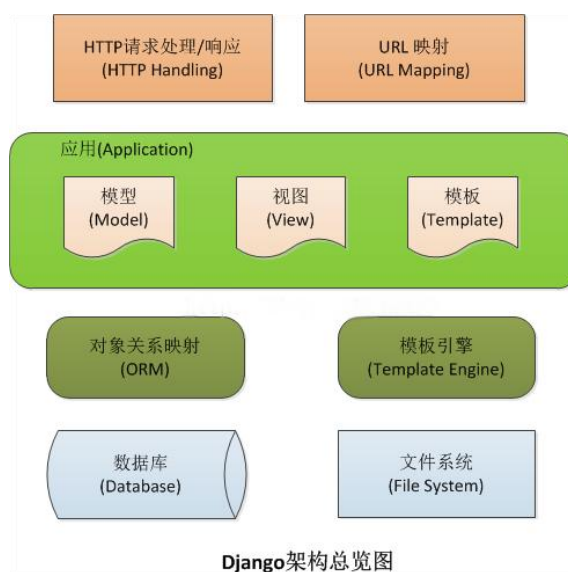


图 4 后端架构图

4.2.3 系统运行环境

在之前的软硬件部分提及到了此部分的相关内容，在这里将详细展示系统运行时的软件与硬件环境：

操作系统	Windows 10 或者以上版本等主流操作系统
网站服务器	Nginx
数据库服务器	Linux Socket
浏览器	Edge、chrome 等主流浏览器
处理器	具备 2 核及以上 CPU 内核
内存	4GB 以上空间内存
网卡	传输速率在 1Mbps 以上

表 9 运行环境表

4.3 数据库设计

4.3.1 数据表设计

本项目的数据库一共设计了三个数据表，分别用于存储用户个人信息，商品库中商品信息，用户收藏商品信息，具体设计如下：

User			
字段名	类型	描述	备注
id	int AI PK	用户 ID	主键，自增
password	varchar(128)	用户密码	加密存储
last_login	datetime(6)	最后登录时间	
username	varchar(150)	用户名	非空，唯一
email	varchar(254)	电子邮件地址	非空，唯一
phone_number	varchar(128)	手机号码	

表 9 用户信息表

Goods			
字段名	类型	描述	备注
id	int AI PK	商品 ID	主键，自增
good_id	varchar(30)	电商平台商品 ID	非空，唯一
good_from	varchar(2)	商品来源	JD, A, SN
good_title	varchar(200)	商品标题	非空
good_link	varchar(200)	商品链接	非空
good_img	varchar(200)	商品图片链接	非空
price_queue_id	int	价格队列 ID	非空
good_price0	varchar(30)	商品价格 0	
good_price1	varchar(30)	商品价格 1	
good_price2	varchar(30)	商品价格 2	
good_price3	varchar(30)	商品价格 3	
good_price4	varchar(30)	商品价格 4	

表 10 商品信息表表

Car			
字段名	类型	描述	备注
id	bigint AI PK	主键 ID	主键，自增
user_id_id	int	用户 ID	外键，关联用户表
good_id_id	int	商品 ID	外键，关联商品表

表 11 用户购物车表

4.3.2 ER 图

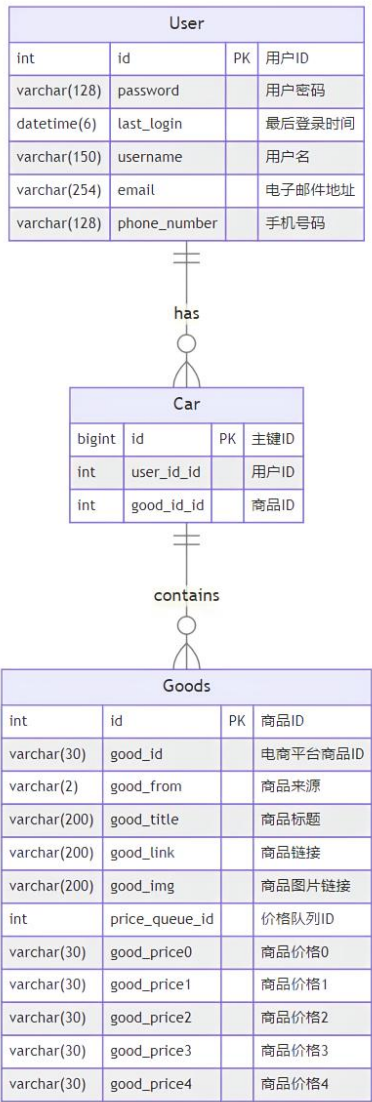


图 5 数据库 ER 图

4.4 系统接口设计

本系统使用 axios 库进行前后端的交互，为了实现需求部分列举的功能，本系统定义并使用了多种 API，在此部分将进行详细介绍：

用户注册	
URL	/login/register

参数值	account,email,password
返回值	Success(if success),Error Message(if fail)

表 12 用户注册表

用户登录	
URL	/login
参数值	account,password
返回值	Success(if success),Error Message(if fail)

表 13 用户登录表

商品搜索	
URL	/search
参数值	search
返回值	商品信息(if success),Error Message(if fail)

表 14 商品搜索表

添加购物车（收藏商品）	
URL	/add
参数值	good_id,good_from,good_title,good_link,good_img,account,good_price
返回值	Success(if success),Error Message(if fail)

表 15 添加购物车表

搜索购物车	
URL	/scar
参数值	account
返回值	购物车中商品信息(if success),Error Message(if fail)

表 16 搜索购物车表

删除购物车商品

URL	/delete
参数值	account,good_id
返回值	Success(if success),Error Message(if fail)

表 17 删除购物车表

修改个人信息	
URL	/change
参数值	account,new_email,new_phone_num
返回值	Success(if success),Error Message(if fail)

表 18 修改个人信息表

4.5 运行设计

操作	行为	预期结果
注册	在登录界面点击注册按钮跳到注册界面，输入基本信息并点击注册按钮	注册一个新账号
登录	在登录界面输入用户名，并输入密码，点击登录按钮	登录成功，跳转到主页面
搜索商品	登录完成之后，在搜索输入框中输入需要搜索的信息，点击 search 按钮开始搜索	主界面显示搜索成功，并加载出搜索出的所有商品
查看个人信息	登陆完成之后，在菜单点击个人信息页面查看个人信息	显示基本的个人信息
修改个人信息	在个人信息界面，点击修改个人信息按钮，填写新的个人信息并点击确认修改	提示修改成功，显示立刻变更
退出登录	在个人信息界面点击退出登录按钮	提示退出成功，并跳转到登录界面
查看商品详情	在搜索完成之后点击商品卡片的下部的查看详情按钮	弹出新的窗口其中显示商品的详细信息，跳转到电商平台界面

添加购物车	在搜索完成之后点击商品卡片的下部的添加到购物车按钮	将商品添加到了购物车中，并提示添加成功
固定商品	在搜索完成之后点击商品卡片的下部的固定按钮	在固定框中显示相关产品的信息
查看固定产品	在搜索界面点击查看比较商品按钮	弹出新的界面，每一种固定商品的对比
查看购物车	登陆完成之后，在菜单点击购物车按钮查看购物车	切换界面，显示购物车中商品信息
从购物车中删除	点击商品下方的删除按钮	提示删除成功，并不显示这个商品
查看历史价格	点击商品下方的查看历史价格的按钮	弹出新的弹窗，显示出历史价格折线统计图
排序商品	点击上方的排序按钮，此时按照价钱进行排序	显示出按照商品价格进行排序的排序结果
筛选商品	在上方筛选框中输入筛选文字信息	界面中只显示筛选过后的产品

表 19 运行设计表

5 系统界面原型

这里提供初期的开发成果，功能并不完善，后期将在此基础之上继续开发完善，实现所有需求功能。

5.1 登录界面

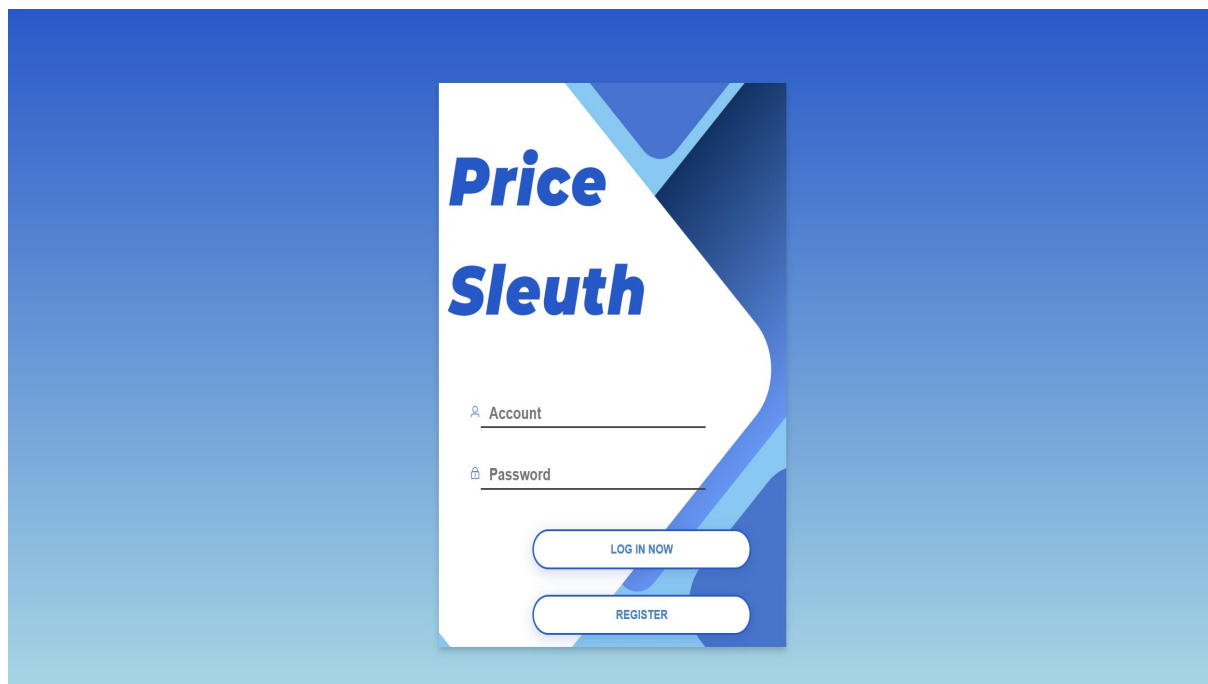


图 6 用户登录界面原型

5.2 用户注册

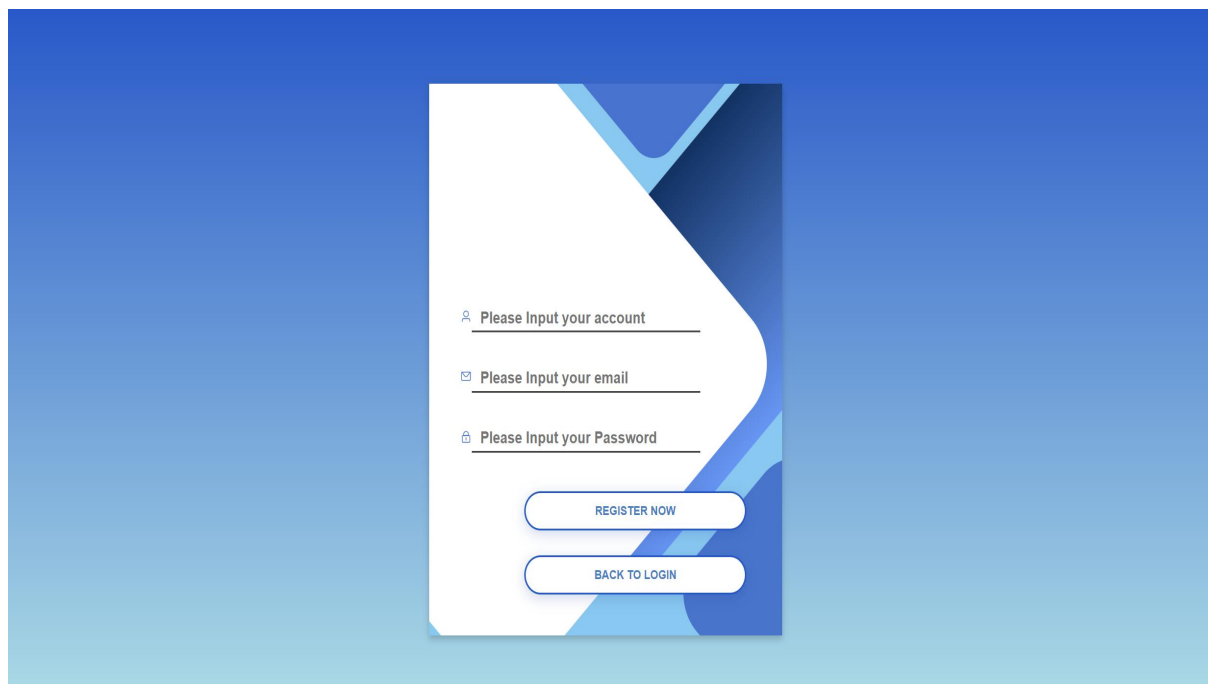


图 7 用户注册界面原型

5.3 搜索界面（主界面）

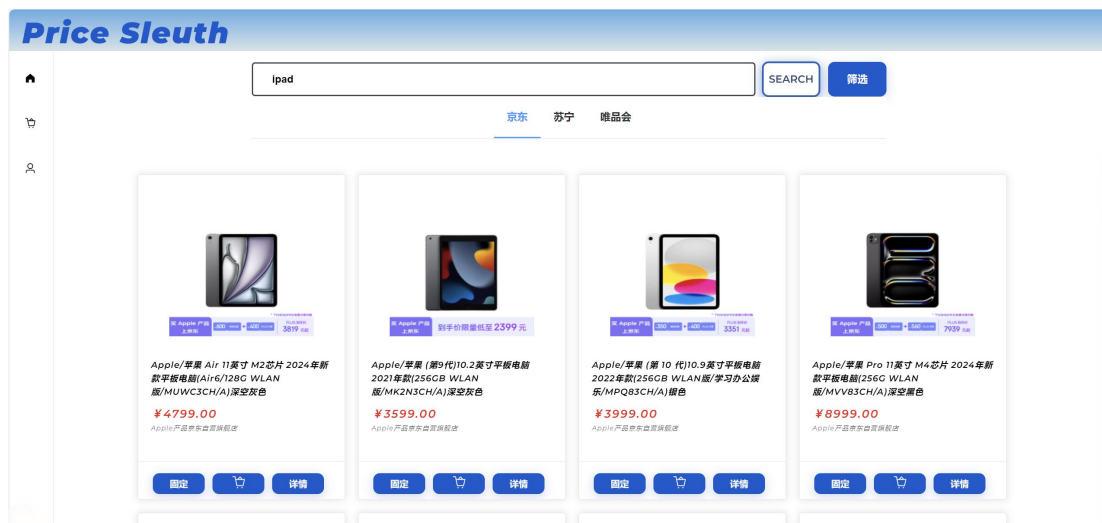


图 8 用户搜索界面原型

5.4 购物车界面

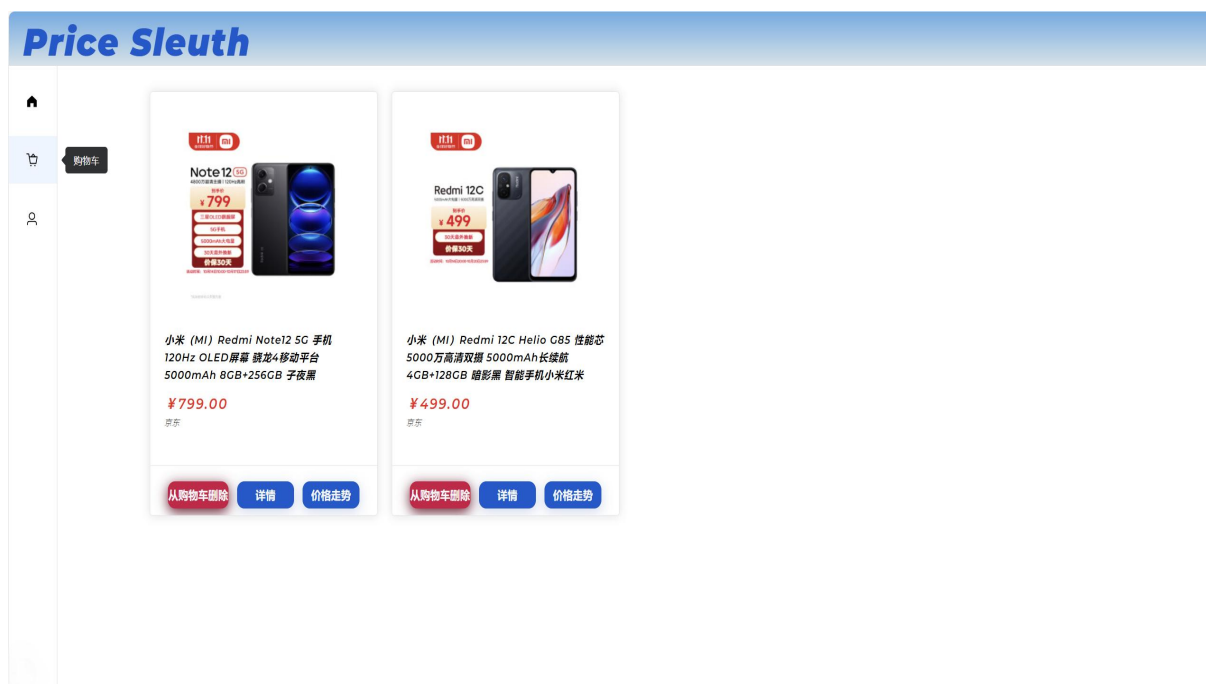


图 9 用户购物车界面原型

5.5 商品固定界面（筛选排序）



图 10 用户固定界面原型

5.6 商品固定详情页面



图 11 用户固定详情界面原型

5.7 商品历史价格查询



图 12 商品历史价格查询界面原型

5.8 手机端适配主页面

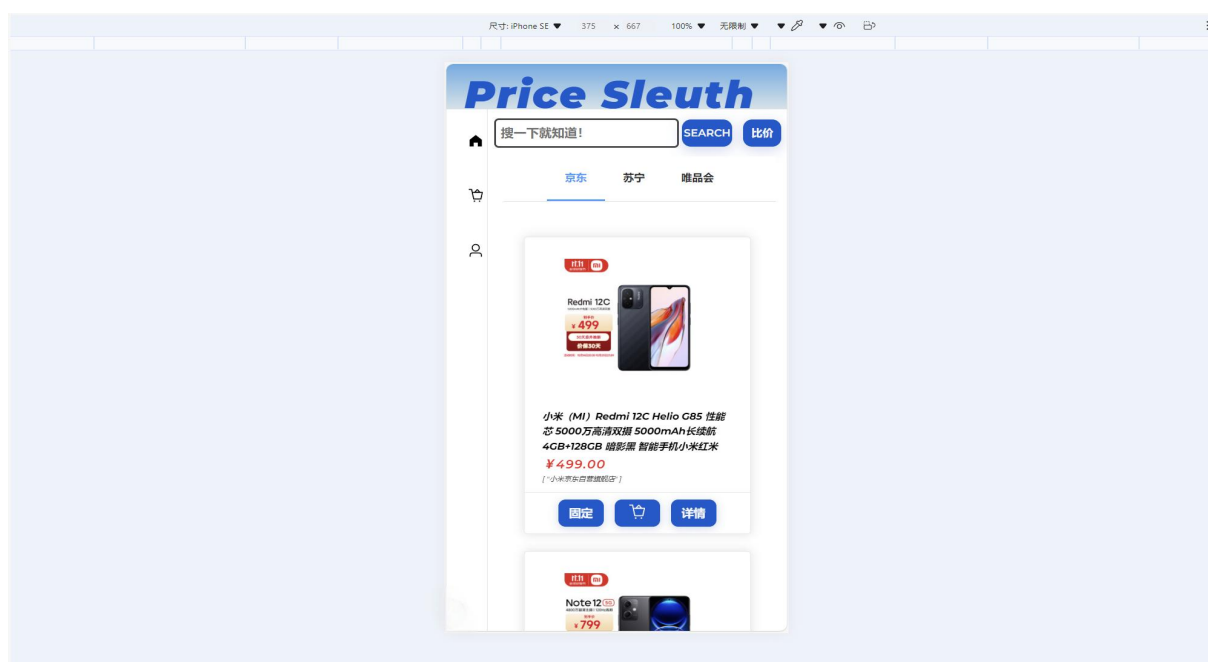


图 13 手机端主界面原型

5.9 手机端登录界面

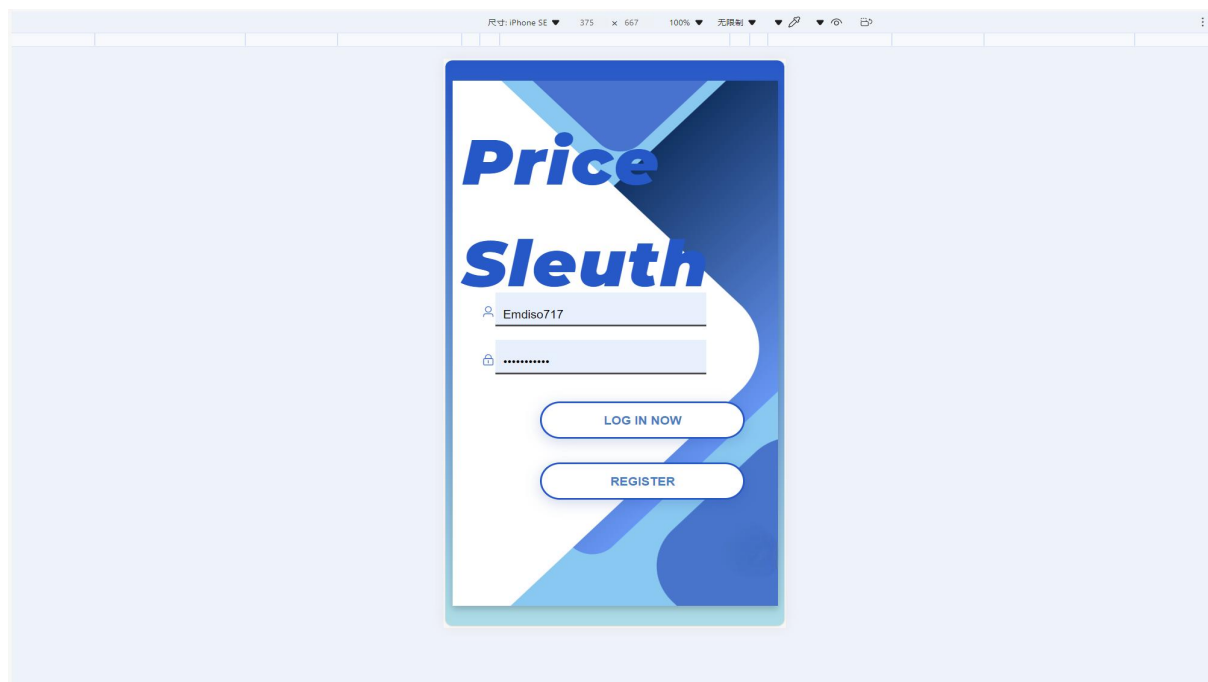


图 14 手机端登录界面原型

5.10 手机端注册界面

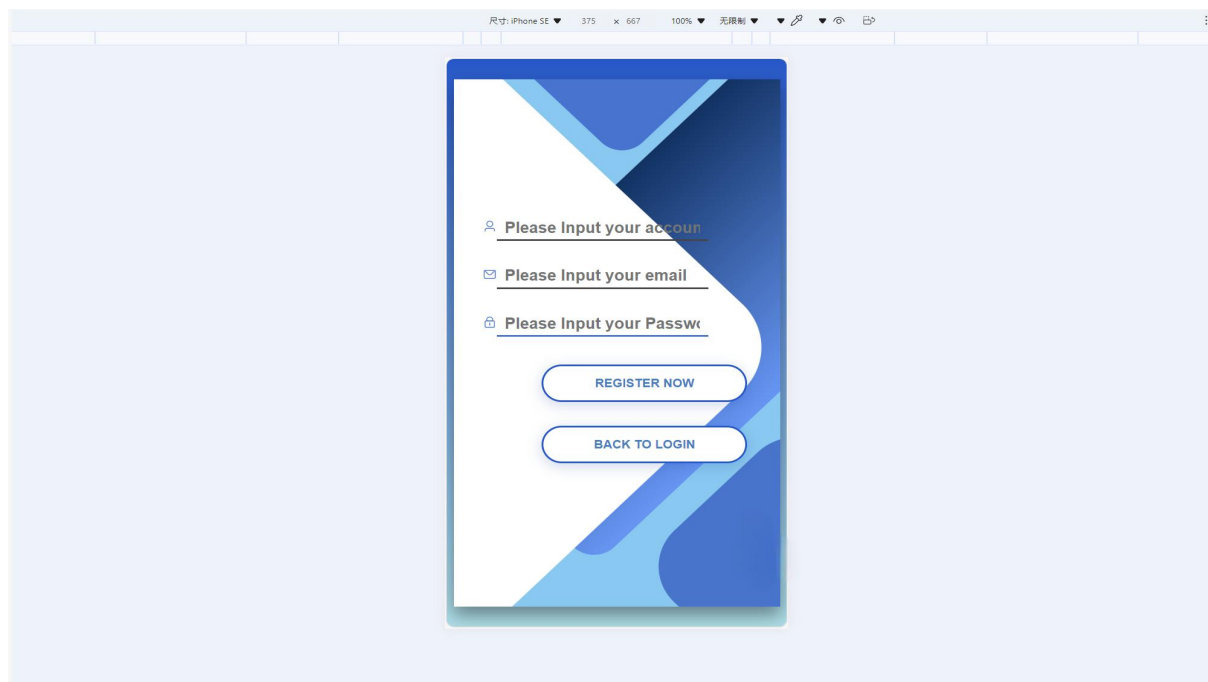


图 14 手机端注册界面原型

6 附录

本文档作为设计初期的设计规划，仅代表设计初期的想法和设想，仅起到规划参考作用。考虑到在实际实现过程中会遇到各种的实际问题，所以最终实现的细节可能会和报告中存在偏差和出入，同时在界面展示部分展示的界面也只是部分设计较为成熟的界面，还有部分界面和详细功能没有完全实现将在后续的开发过程中逐步完善，补充全面。