

# 浙江大学

## 本科实验报告

|       |            |
|-------|------------|
| 课程名称: | B/S 体系软件设计 |
| 题 目:  | 商品比价网站设计报告 |
| 姓 名:  | 吴语若        |
| 学 院:  | 计算机科学与技术学院 |
| 专 业:  | 计算机科学与技术   |
| 学 号:  | 3220104111 |
| 指导教师: | 胡晓军        |

2024 年 10 月 27 日

# 商品比价网站设计报告

3220104111 吴语若

2024-10-27

# 目 录

|                          |    |
|--------------------------|----|
| 1 引言 .....               | 1  |
| 1.1 编写目的 .....           | 1  |
| 1.2 项目背景 .....           | 1  |
| 2 系统需求分析 .....           | 1  |
| 2.1 实验要求 .....           | 1  |
| 2.2 功能需求分析 .....         | 2  |
| 2.2.1 用户模块需求分析 .....     | 2  |
| 2.2.2 商品查询模块需求分析 .....   | 2  |
| 2.2.3 商品信息管理模块需求分析 ..... | 3  |
| 2.3 非功能性需求分析 .....       | 3  |
| 2.3.1 性能需求 .....         | 3  |
| 2.3.2 输入输出需求 .....       | 3  |
| 2.3.3 数据管理需求 .....       | 4  |
| 2.3.4 可维护性需求 .....       | 5  |
| 2.3.5 硬件需求 .....         | 5  |
| 3 系统总体架构设计与技术选型 .....    | 5  |
| 3.1 系统功能模块设计 .....       | 5  |
| 3.2 系统总体架构设计 .....       | 6  |
| 3.3 技术栈选择 .....          | 6  |
| 3.3.1 前端技术栈 .....        | 7  |
| 3.3.2 后端技术栈 .....        | 7  |
| 3.3.3 数据库技术栈 .....       | 7  |
| 4 数据处理 .....             | 7  |
| 4.1 数据存储 .....           | 7  |
| 4.2 数据安全 .....           | 7  |
| 4.3 数据库设计及建表 .....       | 7  |
| 4.3.1 用户表 .....          | 8  |
| 4.3.2 商品表 .....          | 8  |
| 4.3.3 降价关系追踪表 .....      | 8  |
| 4.3.4 E-R 图 .....        | 9  |
| 5 系统接口设计 .....           | 9  |
| 5.1 用户信息接口 .....         | 9  |
| 5.1.1 用户注册 .....         | 9  |
| 5.1.2 用户登录 .....         | 10 |
| 5.1.3 用户注销 .....         | 10 |
| 5.1.4 退出登录 .....         | 10 |
| 5.1.5 验证用户名 .....        | 10 |
| 5.1.6 验证密码 .....         | 11 |
| 5.1.7 验证 email 格式 .....  | 11 |
| 5.1.8 修改用户名 .....        | 11 |
| 5.1.9 修改密码 .....         | 11 |

|                         |    |
|-------------------------|----|
| 5.1.10 修改 email .....   | 12 |
| 5.1.11 获取降价提醒商品列表 ..... | 12 |
| 5.2 商品信息接口 .....        | 12 |
| 5.2.1 获取商品信息 .....      | 12 |
| 5.2.2 添加商品信息 .....      | 12 |
| 5.2.3 更新商品基本信息 .....    | 13 |
| 5.2.4 更新商品价格 .....      | 14 |
| 5.2.5 删除商品信息 .....      | 14 |
| 5.2.6 商品查询 .....        | 14 |
| 5.2.7 商品是否降价 .....      | 14 |
| 5.2.8 商品降价提醒标记 .....    | 15 |
| 5.2.9 取消商品降价提醒 .....    | 15 |
| 5.2.10 发送降价提醒 .....     | 15 |
| 6 系统界面设计 .....          | 15 |
| 6.1 登录界面 .....          | 16 |
| 6.2 注册界面 .....          | 17 |
| 6.3 用户个人中心界面 .....      | 18 |
| 6.4 修改个人信息界面 .....      | 19 |
| 6.5 修改密码界面 .....        | 20 |
| 6.6 主界面（查询界面） .....     | 21 |
| 6.7 商品信息界面 .....        | 22 |
| 6.8 降价提醒推送框 .....       | 23 |
| 7 系统出错处理 .....          | 23 |
| 7.1 网络连接错误 .....        | 23 |
| 7.2 数据库错误 .....         | 23 |
| 7.3 应用程序错误 .....        | 24 |
| 7.4 安全漏洞 .....          | 24 |
| 7.5 第三方服务中断 .....       | 24 |
| 8 项目开发计划 .....          | 24 |

# 1 引言

## 1.1 编写目的

这个项目是 2024-2025 秋冬学期《B/S 体系软件设计》课程的大程，旨在开发一个成熟的商品价格比较网站。用户登录后，能够通过商品名称在主流电商平台上查询商品价格并支持同一商品在不同电商平台的价格比较，可以帮助用户更便捷地“货比三家”，买到高性价比的产品。这个网站不仅要用户友好，而且要适配手机移动端，确保在手机浏览器和内置于应用程序（如微信）的浏览器中都能够良好地展示。此外，项目还需要提供详尽的软件项目文档，以帮助开发者了解并掌握一整套 web 应用开发技术以及整个开发流程。

这份文档是项目的系统设计文档，包含了系统的需求分析、总体架构设计与技术选型、数据库设计、系统接口以及界面设计等内容。它详细描述了商品价格比较网站的设计细节。这个项目的范围涵盖了完整的 web 前后端开发以及相关项目文档的编写，而且整个项目将由单一开发者独立完成。

## 1.2 项目背景

在现代电子商务的蓬勃发展，消费者面临着来自不同平台的海量商品信息和价格选择。随着电商平台的多样化，商品价格的波动性和差异性日益显著，消费者在购物时常常需要耗费大量时间和精力来进行价格比较和决策。这种情况催生了对高效、便捷的价格比较工具的需求。

商品价格比较网站的开发正是为了应对这一挑战。通过整合淘宝、京东等主流电商平台的商品信息，该网站能够为用户提供实时的价格查询和比较服务。用户不仅可以查看商品的当前价格，还可以通过历史数据分析价格趋势，帮助用户在价格最低时进行购买。此外，网站提供降价提醒功能，用户可以针对特定商品设置提醒，当商品价格下降时，系统会通过邮件或 App 通知用户。

项目的背景是为了提升消费者的购物体验，满足其对价格透明度和购物便捷性的需求。通过提供全面的价格信息和智能化的价格监控服务，商品价格比较网站旨在帮助消费者做出更明智的购物决策，节省时间和成本。这一项目不仅顺应了电子商务发展的趋势，也为消费者带来了实实在在的便利和价值。

# 2 系统需求分析

## 2.1 实验要求

需要实现的基本功能如下：

1. 实现用户注册、登录功能，用户注册时需要填写必要的信息并验证：如用户名、密码要求在 6 字节以上，email 的格式验证，并保证用户名和 email 在系统中唯一，用户登录后可以进行以下操作：
2. 通过商品名称在主流电商平台上查询该商品实时价格

- i. 商品名称建议分词处理优化查询;
  - ii. 查询多个结果的处理;
  - iii. 很多平台需要平台用户登录验证后才可以进行查询。(直接使用开发者账号获取数据)
3. 支持至少 2 个以上平台查询价格进行比较 (淘宝、京东等)。
  4. 建立商品库, 将商品信息和商品价格保存在数据库中。商品信息包含名称、多级品类、规格、条码、图片等, 方便后续查询。
  5. 提供商品查询界面能显示商品信息, 把历史价格用图表形式显示 (如价格走势图中)。
  6. 支持设置降价提醒, 针对指定商品定时查询最新价格, 如有降价发送提醒, 可以通过邮件, App 推送等方式实现。
  7. 样式适配手机, 开发手机 App 或能够在手机浏览器/微信等应用内置的浏览器中友好显示。

增强功能:

8. 如开发手机端, 可以用相机拍摄商品图片或扫码商品条码进行商品查询。

## 2.2 功能需求分析

该项目主要是一个 B/S 架构的 web 应用, 对于每个用户, 它提供了以下功能:

- 用户注册与登录
- 用户基本信息管理与修改
- 商品多平台实时价格查询
- 商品信息及历史价格查看
- 商品降价提醒

### 2.2.1 用户模块需求分析

1. 用户注册: 用户通过注册界面填入注册信息并验证 (用户名、密码要求在 6 字节以上, email 的格式验证, 并保证用户名和 email 在系统中唯一), 验证通过后将注册信息添加入用户相关的数据库并返回用户登录主界面, 同时数据库内的用户密码需要加密以防信息泄露。
2. 用户登录: 用户通过输入用户名或邮箱以及对应的密码进行登录, 验证失败后输出错误信息, 验证成功后跳转至商品查询界面。
3. 用户信息修改: 用户可以在登录后进入个人主页进行用户名、密码和邮箱等信息的修改, 在修改后信息通过验证后保存至相关数据库。
4. 退出登录: 用户点击退出登录按钮后退出登录状态, 回到用户登录主界面。
5. 用户注销: 用户可以在登录后进入个人主页进行用户的注销, 并将注销的用户数据在相关数据库删除, 注销后自动退出登录。

### 2.2.2 商品查询模块需求分析

1. 查询权限：使用开发者电商平台账号完成电商平台的登录，确保各平台都有查询价格的权限。
2. 商品查询：对输入的商品名称进行分词处理优化查询，并显示处理后的多个查询结果（在最前面提供最低价的商品信息）；点击相应查询结果可以显示商品的详细信息（包括历史价格走势、名称、多级品类、规格、条码、图片等）。
  - 支持多样化查询方式，例如手机端可以相机拍摄图片查询、扫描商品条码查询。
3. 降价追踪：用户通过在商品信息卡片上标记“降价提醒我”可以启用降价提醒功能，定时查询最新的商品价格更新商品数据库，如果有降价则通过邮件（用户为登出状态）、app 推送（用户为登录状态）发送提醒；通过取消勾选“降价提醒我”标记取消该商品的降价提醒。

### 2.2.3 商品信息管理模块需求分析

1. 商品数据库的建立：建立商品库，将商品信息和商品价格保存在数据库中。商品信息包含名称、多级品类、规格、条码、图片。
2. 商品数据更新：定期查询商品是否存在、基本信息是否更新以及最新商品价格，维护历史价格表单并判断用户是否追踪该商品的降价情况，如有降价则通知该商品对应的追踪降价情况的所有用户。

## 2.3 非功能性需求分析

### 2.3.1 性能需求

1. 响应时间：
  - 用户在进行商品查询时，系统应在 3 秒内返回结果，以确保快速响应。
  - 页面加载时间应控制在 2 秒以内，提升用户体验。
2. 并发用户支持：
  - 系统应支持至少 1000 个并发用户访问，确保在高峰期也能稳定运行。
  - 需采用负载均衡技术，分配服务器资源以处理大量请求。
3. 数据处理效率：
  - 系统应能够在 1 分钟内完成对至少 1000 条商品价格信息的更新。
  - 使用高效的数据库查询和缓存机制，减少数据处理延迟。
4. 可扩展性：
  - 系统架构需具备良好的可扩展性，以便在用户增长时能够方便地增加服务器和数据库容量。
  - 支持模块化设计，便于后续功能扩展和性能优化。
5. 可靠性：
  - 系统应具备 99.9% 的可用性，确保服务的持续稳定。
  - 需配置自动化监控和故障恢复机制，快速响应并解决潜在问题。

### 2.3.2 输入输出需求

#### 1. 输入需求:

- 用户注册信息: 包括用户名、密码、电子邮件等, 需进行格式验证和唯一性检查。
- 商品查询关键词: 支持模糊搜索和关键词优化, 提升查询准确性。
- 用户设置的降价提醒条件: 包括目标价格和通知方式 (如邮件或 App 推送)。
- 确保不能被用户的非法输入所破坏, 例如 SQL 注入攻击。

#### 2. 输出需求:

- 商品价格信息: 实时显示来自不同电商平台的商品价格, 支持图表形式展示价格趋势。
- 查询结果: 快速返回符合用户输入条件的商品列表, 包含商品名称、图片、价格等信息。
- 降价提醒通知: 当商品价格下降到用户设定的目标时, 系统通过邮件或 App 推送通知用户。

### 2.3.3 数据管理需求

#### 1. 数据准确性:

- 保证所有展示的商品信息准确无误, 包括历史价格走势图、名称、多级品类、规格、条码、图片等。定期与商家同步数据; 也可以尝试引入第三方数据校验服务; 设置用户反馈渠道以纠正错误信息。

#### 2. 数据完整性:

- 确保数据库中每个商品条目的关键字段 (如 ID、名称、价格) 都必须填写完整。建立严格的输入规则, 对于必填项不允许为空; 采用预填充技术减少人工录入错误。

#### 3. 数据一致性:

- 实现数据同步机制, 确保多源数据更新时能够及时反映最新状态; 使用事务处理来保证操作的原子性, 防止部分成功导致的数据不一致。

#### 4. 数据安全性:

- 保护用户个人信息及交易记录等敏感数据不受未授权访问。采用加密技术存储敏感信息; 设置权限控制, 限制对敏感数据的访问; 定期进行安全审计和漏洞扫描。

#### 5. 数据备份与恢复:

- 确保即使发生意外情况也能快速恢复系统运行。制定详细的备份策略, 包括备份频率、备份内容以及备份方式; 建立灾难恢复计划, 测试并优化恢复流程。

#### 6. 数据可扩展性:

- 采用分布式架构设计, 便于水平扩展; 优化数据库索引, 提高查询效率; 考虑使用云服务以获得弹性计算能力。

#### 7. 数据隐私合规:



- 遵守相关法律法规关于个人隐私保护的规定。明确告知用户哪些信息会被收集及其用途；提供方便的选项让用户选择是否同意数据收集；遵循 GDPR 等国际标准要求。

#### 2.3.4 可维护性需求

1. 模块化设计：
  - 通过将系统划分为多个独立的模块，降低各部分之间的耦合度，使得单个模块的修改不会影响到其他部分。
2. 代码质量：
  - 保持代码的整洁性和可读性，便于后续的修改和扩展。
3. 自动化测试：
  - 构建单元测试、集成测试和端到端测试框架，通过自动化测试提高软件的质量。
4. 错误监控与日志记录：
  - 实时监控系統运行状态，快速定位并解决出现的问题。合理配置日志级别，记录关键操作和异常信息。
5. 版本控制：
  - 使用 Git 等版本控制系统，有效管理代码版本，确保每次提交都有详细的变更说明，支持回滚。
6. 用户反馈循环：
  - 建立有效的用户反馈机制，持续改进产品。

#### 2.3.5 硬件需求

- 操作系统：Windows 11、Linux 等
- 硬件配置：具备 4 核 CPU、8GB 内存以上的配置，以支持系统的基本运行

### 3 系统总体架构设计与技术选型

#### 3.1 系统功能模块设计

- 用户模块
  - 用户注册
  - 用户登录
- 用户个人中心模块
  - 用户信息修改
  - 用户注销
  - 用户登出
  - 追踪降价商品更改
- 商品模块
  - 商品查询

- 商品价格对比
- 商品信息显示
- 商品历史价格图表
- 商品降价追踪标记
- 商品降价提醒

## 3.2 系统总体架构设计

系统的总体架构如图 1 所示：

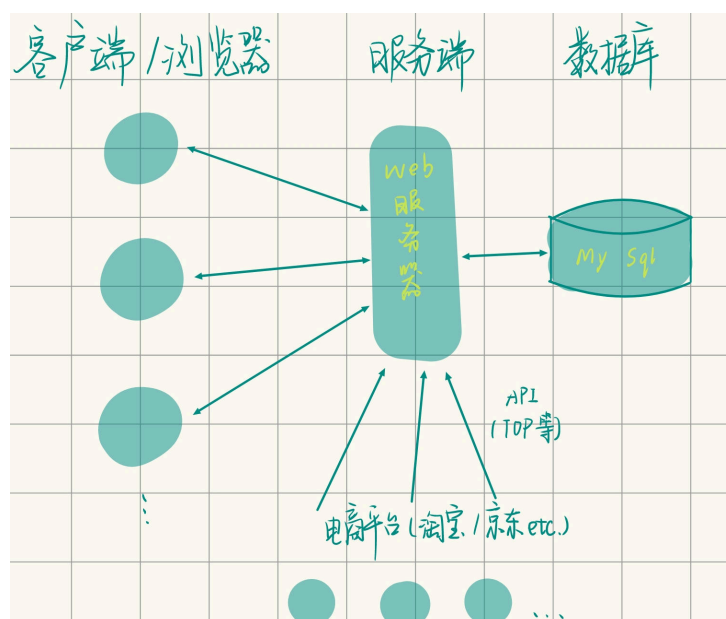


图 1 系统总体架构设计

服务端接收用户在网页上发出的请求，并对其进行处理，这是用户与系统交互的主要接口之一，负责处理用户的请求并提供相应的数据和功能。

而对于获取各个电商平台的价格和商品信息的功能，我们需要通过后端服务器调用各大电商平台的 API 实现，例如淘宝平台的 TOP 等。

而通过后端和数据库的交互，我们将前端获取的用户信息和通过 API 获取的商品信息由服务器写入数据库并由数据库进行储存管理，随时给服务器提供所需数据并再由服务器提供给使用前端的用户。

## 3.3 技术栈选择

- 前端技术栈
  - VUE3
  - ECharts
  - Element Plus
- 后端技术栈
  - SpringBoot
  - SpringMVC

- 电商平台 API 例如 TOP
- 数据库技术栈
  - MySQL

### 3.3.1 前端技术栈

前端整体是使用了 VUE3 框架并用 JavaScript 语言编写代码。而在开发过程中使用了 Element Plus 这一基于 VUE3 的高质量组件库，使用 ECharts 这一数据可视化图表库完成了历史价格图表绘制这一功能。

### 3.3.2 后端技术栈

后端整体是使用了 Spring Boot 框架并用 Java 语言编写代码，同时使用 Spring MVC 框架构建了后端接口。而对于获取电商平台商品信息的操作，我们则通过调用电商平台的 API 例如淘宝的 TOP 实现。

### 3.3.3 数据库技术栈

数据库方面选用了 MySQL 数据库，使用 SQL 语言编写数据库代码，使用 MySQL 构建数据库并实现数据的存储和管理。

## 4 数据处理

### 4.1 数据存储

本项目使用 MySQL 数据库存储数据，进行数据的管理。

### 4.2 数据安全

为了确保商品价格比较网站的数据安全，我们采取了一系列严格的技术和管理措施，旨在保护用户的个人信息、交易记录以及平台上的其他敏感数据。以下是我们在数据安全方面的主要措施：

1. 用户需要注册账号登录后才能使用网站的所有功能。只有用户登录后才能查看自己的历史降价追踪记录、修改个人信息。
2. 所有用户数据在传输过程中均使用 HTTPS 协议，确保数据在客户端与服务器之间传输时的安全性。
3. 用户密码在数据库中以加密形式存储，即使数据库泄露，也无法直接获取明文密码。

### 4.3 数据库设计及建表

本项目数据库一共设计了 3 张表，分别是用户表、商品表和降价关系追踪表。其中，用户表用于存储用户的个人信息，商品表用于存储商品的基本信息和历史价格，而降价关系追踪表则用于存储哪些用户在关注哪些商品是否降价。

### 4.3.1 用户表

用户表用于存储用户的个人信息，包括用户名、密码、邮箱等。其中，用户名和邮箱是唯一的，这意味着用户名和邮箱在系统中不能重复。这可以确保用户的唯一性，防止用户重复注册。此外，用户的密码是经过加密的，这可以确保用户的密码不会被泄露。

| 字段名      | 类型      | 长度  | 是否为空 | 是否唯一 | 备注       |
|----------|---------|-----|------|------|----------|
| uid      | int     | 16  | 否    | 是    | 主键，用户 ID |
| email    | varchar | 255 | 否    | 是    | 邮箱       |
| username | varchar | 255 | 否    | 是    | 用户名      |
| password | varchar | 255 | 否    | 否    | 密码       |

### 4.3.2 商品表

商品表用于存储商品的基本信息，包括商品名称、商品所属电商平台、商品现在的价格、商品多级品类、规格、条码、图片和近十周的历史价格数组。其中，商品 ID 是唯一的，这意味着商品 ID 在系统中不能重复。这可以确保商品的唯一性，防止商品重复获取。

| 字段名               | 类型      | 长度     | 是否为空 | 是否唯一 | 备注  |
|-------------------|---------|--------|------|------|---|
| pid               | int     | 16     | 否    | 是    | 主键，商品 ID  |
| productname       | varchar | 255    | 否    | 否    | 商品名称  |
| platform          | varchar | 100    | 否    | 否    | 商品所属电商平台  |
| current_price     | decimal | (10,2) | 否    | 否    | 商品当前的价格   |
| category          | varchar | 255    | 否    | 否    | 商品的多级品类(以逗号分隔，如“电子 > 手机 > 智能手机”)  |
| specification     | text    | 256    | 是    | 否    | 商品的规格描述   |
| barcode           | varchar | 255    | 是    | 否    | 商品的条码   |
| image_url         | varchar | 255    | 是    | 否    | 商品的图片 URL   |
| image_url         | varchar | 255    | 是    | 否    | 商品的链接 URL   |
| historical_prices | JSON    | 256    | 是    | 否    | 商品近十周的历史价格数组（格式：[{"week": 1, "price": 100}, {"week": 2, "price": 95}, ...]） |

### 4.3.3 降价关系追踪表

降价关系追踪表则用于存储哪些用户在关注哪些商品是否降价，需要发送相关的降价提醒，是商品和用户多对多关系的实现。故降价关系追踪表需要包括商品的 id 和用户的 id，并与用户表和商品表关联，级联删除。

| 字段名 | 类型  | 长度 | 是否为空 | 是否唯一 | 备注                      |
|-----|-----|----|------|------|-------------------------|
| uid | int | 16 | 否    | 否    | 外键, 用户 id, 与用户表关联, 级联删除 |
| pid | int | 16 | 否    | 否    | 外键, 商品 id, 与商品表关联, 级联删除 |

#### 4.3.4 E-R 图

该项目各表之间关系的 E-R 图如下：

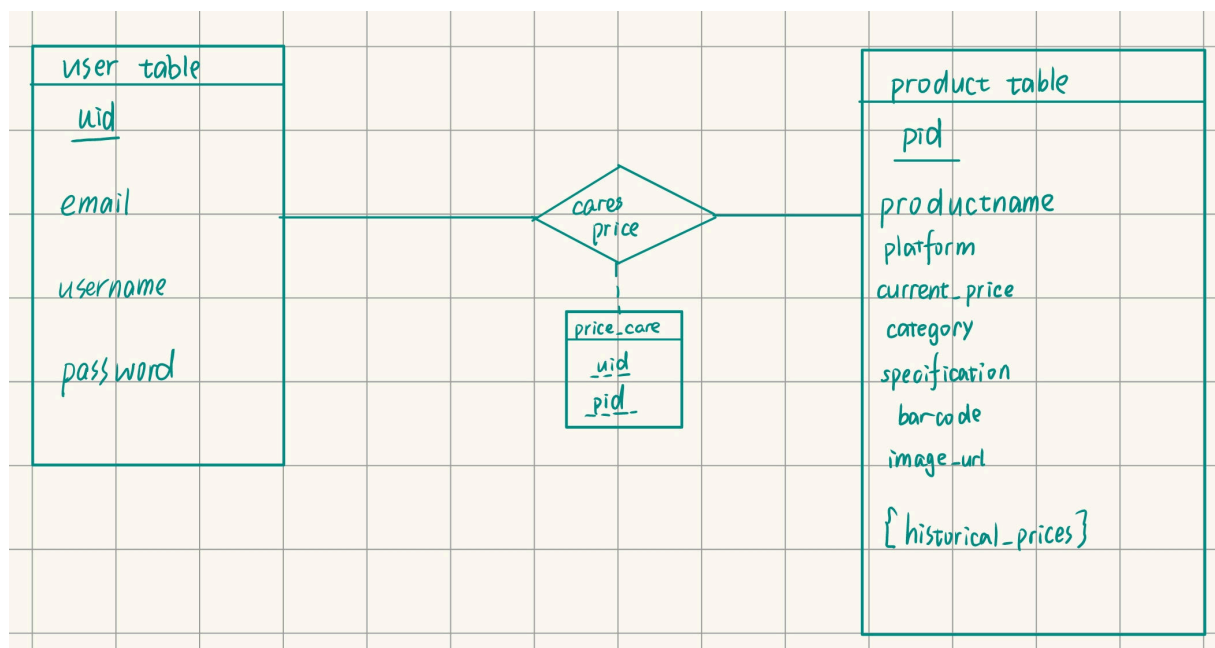


图 2 E-R 图

## 5 系统接口设计

### 5.1 用户信息接口

#### 5.1.1 用户注册

- 函数名: user\_register()
- 请求方式: POST
- 输入参数:
  - 用户名
  - 密码
  - 确认的密码
  - email 地址
- 返回值:
  - success - 注册成功
  - 错误信息 - 注册失败, 返回具体出错信息 (格式不对、用户名被占用等)
- 修改数据内容:
  - 在用户表项中添加对应的用户实例

### 5.1.2 用户登录

- 函数名: `user_login()`
- 请求方式: POST
- 输入参数:
  - 用户名
  - 密码
- 返回值:
  - `success` - 登录成功
  - `error` - 登录失败
- 修改数据内容:
  - 记录当前登录的用户 `id` (未登录状态下是-1)

### 5.1.3 用户注销

- 函数名: `user_cancel()`
- 请求方式: POST
- 输入参数:
  - 当前用户 `id`
  - 密码
- 返回值:
  - `success` - 注销成功
  - 错误信息 - 注销失败, 返回具体出错信息 (`id` 不存在、密码错误)
- 修改数据内容:
  - 在用户表项中删除对应的用户实例

### 5.1.4 退出登录

- 函数名: `user_logout()`
- 请求方式: POST
- 输入参数:
  - 当前用户 `id`
- 返回值:
  - `success` - 退出登录成功
  - `error` - 退出登录失败
- 修改数据内容:
  - 将当前用户 `id` 置-1 (未登录状态)

### 5.1.5 验证用户名

- 函数名: `legal_username()`
- 请求方式: GET
- 输入参数:

- 用户名
- 返回值:
  - success - 用户名合法
  - error - 用户名不合法

#### 5.1.6 验证密码

- 函数名: legal\_password()
- 请求方式: GET
- 输入参数:
  - 密码
- 返回值:
  - success - 密码合法
  - error - 密码不合法

#### 5.1.7 验证 email 格式

- 函数名: legal\_email()
- 请求方式: GET
- 输入参数:
  - 邮箱
- 返回值:
  - success - 邮箱合法
  - error - 邮箱不合法

#### 5.1.8 修改用户名

- 函数名: update\_username()
- 请求方式: POST
- 输入参数:
  - 当前用户 id
  - 新用户名
- 返回值:
  - success - 修改成功
  - 错误信息 - 修改失败, 返回具体出错信息(用户名不合法、用户名被占用等)
- 修改数据内容:
  - 在用户表项中修改对应的用户实例的用户名

#### 5.1.9 修改密码

- 函数名: update\_password()
- 请求方式: POST
- 输入参数:

- 当前用户 id
- 旧密码
- 新密码
- 返回值：
  - `success` - 修改成功
  - 错误信息 - 修改失败，返回具体出错信息（旧密码不正确、新密码不合法等）
- 修改数据内容：
  - 在用户表项中修改对应的用户实例的密码

#### 5.1.10 修改 email

- 函数名： `update_email()`
- 请求方式： POST
- 输入参数：
  - 当前用户 id
  - 新邮箱
- 返回值：
  - `success` - 修改成功
  - 错误信息 - 修改失败，返回具体出错信息（新邮箱不合法等）
- 修改数据内容：
  - 在用户表项中修改对应的用户实例的邮箱

#### 5.1.11 获取降价提醒商品列表

- 函数名： `user_getproductlist()`
- 请求方式： GET
- 输入参数：
  - 当前用户 id
- 返回值：
  - 当前用户设置的降价提醒的商品 id 列表

### 5.2 商品信息接口

#### 5.2.1 获取商品信息

- 函数名： `product_getinfo()`
- 请求方式： GET
- 输入参数：
  - 商品 id
- 返回值：
  - 对应的商品信息结构体

#### 5.2.2 添加商品信息



- 函数名: `product_add()`
- 请求方式: POST
- 输入参数:
  - 商品名称
  - 商品所属电商平台
  - 商品当前的价格
  - 商品的多级品类
  - 商品的规格描述
  - 商品的条码
  - 商品的图片 URL
  - 商品的购买链接
  - 商品的历史价格数据
- 返回值:
  - `success` - 添加成功
  - 错误信息 - 添加失败, 返回具体出错信息 (商品价格出错等)
- 修改数据内容:
  - 在商品表项中添加对应的商品实例

### 5.2.3 更新商品基本信息

- 函数名: `product_update()`
- 请求方式: POST
- 输入参数:
  - 修改的商品 id
  - 修改的项目序号:
    - 1 - 商品名称
    - 2 - 商品所属电商平台
    - 3 - 商品当前的价格
    - 4 - 商品的多级品类
    - 5 - 商品的规格描述
    - 6 - 商品的条码
    - 7 - 商品的图片 URL
    - 8 - 商品的购买链接
  - 修改的新的内容
- 返回值:
  - `success` - 修改成功
  - 错误信息 - 修改失败, 返回具体出错信息 (找不到商品、商品信息出错等)
- 修改数据内容:
  - 在商品表项中修改对应的商品实例的内容

#### 5.2.4 更新商品价格

- 函数名: `product_priceupdate()`
- 请求方式: POST
- 输入参数:
  - 修改的商品 id
  - 最新价格
- 返回值:
  - `success` - 修改成功
  - 错误信息 - 修改失败, 返回具体出错信息 (找不到商品、商品价格出错等)
- 修改数据内容:
  - 在商品表项中修改对应的商品实例的价格, 并每隔一周管理历史价格表 (将之前的价格填入历史价格第一位, 所有历史价格后移一位)

#### 5.2.5 删除商品信息

- 函数名: `product_delete()`
- 请求方式: POST
- 输入参数:
  - 删除的商品 id
- 返回值:
  - `success` - 删除成功
  - 错误信息 - 删除失败, 返回具体出错信息 (找不到商品等)
- 修改数据内容:
  - 在商品表项中删除对应的商品实例 (注意级联删除降价关系追踪表的实例)

#### 5.2.6 商品查询

- 函数名: `product_search()`
- 请求方式: GET
- 输入参数:
  - 查询的内容
- 返回值:
  - 通过优化搜索找到的经过价格从低到高排序的商品 id 列表

#### 5.2.7 商品是否降价

- 函数名: `product_checkprice()`
- 请求方式: GET
- 输入参数:
  - 商品 id
- 返回值:
  - 1 - 商品现在的最新价格低于历史价格

- 0 - 商品现在的最新价格未低于历史价格

### 5.2.8 商品降价提醒标记

- 函数名: `product_pricehint()`
- 请求方式: POST
- 输入参数:
  - 当前用户 id
  - 商品 id
- 返回值:
  - `success` - 对应商品降价提醒设置成功
  - 错误信息 - 设置失败, 返回具体出错信息(找不到商品、已经设置降价提醒等)
- 修改数据内容:
  - 在降价关系追踪表中添加对应实例

### 5.2.9 取消商品降价提醒

- 函数名: `product_pricehintdelete()`
- 请求方式: POST
- 输入参数:
  - 当前用户 id
  - 商品 id
- 返回值:
  - `success` - 对应商品降价提醒取消设置成功
  - 错误信息 - 设置失败, 返回具体出错信息(找不到商品、未设置降价提醒等)
- 修改数据内容:
  - 在降价关系追踪表中删除对应实例

### 5.2.10 发送降价提醒

- 函数名: `product_pricesend()`
- 请求方式: GET
- 输入参数:
  - 商品 id
- 返回值:
  - 商品历史价格
  - 商品最新价格
  - 关注该商品的用户 id 列表
  - 商品名称

## 6 系统界面设计

## 6.1 登录界面

**BEST PRODUCT** BUY!

# 商品比价网站

username:

password:

[forget password](#)

**LOGIN**

[Register NOW!](#)

DESIGN By Wyr, Zhejiang University

图3 登录界面设计

## 6.2 注册界面



The registration interface features a central logo with the text "BEST PRODUCT" and a "buy!" button. Below the logo is the title "商品比价网站". The form includes four input fields: "email:", "username:", "password:", and "password again:", each with a toggle icon. A large "REGISTER" button is positioned below the fields. At the bottom, it says "DESIGN By Wyr, Zhejiang University".

**BEST PRODUCT** buy!

商品比价网站

email:

username:

password: 

password again: 

**REGISTER**

DESIGN By Wyr, Zhejiang University


图4 注册界面设计

### 6.3 用户个人中心界面



图5 用户个人中心界面设计

## 6.4 修改个人信息界面



# 个人中心

商品比价网站

登出

注销

Username: 

Save

Email : 

Save

---

关注的商品:

| 序号  | 商品名称         | 价格     |                 |                 |
|---|--------------|--------|-----------------|-----------------|
| 1   | SMG 棒球帽      | 199.99 | <div>详细信息</div> | <div>取消关注</div> |
| 2   | 《除了禁止入内》实体专辑 | 299.99 | <div>详细信息</div> | <div>取消关注</div> |
| <div> <div>1</div> <div>2</div> <div>3</div> </div> |              |        |                 |                 |

◀

▶

DESIGN By Wyr, Zhejiang University

图6 修改个人中心界面设计

## 6.5 修改密码界面



商品比价网站

修改密码

☐  
☐  
☐

UPDATE

DESIGN By Wyr, Zhejiang University

图 7 修改密码界面设计



## 6.6 主界面（查询界面）

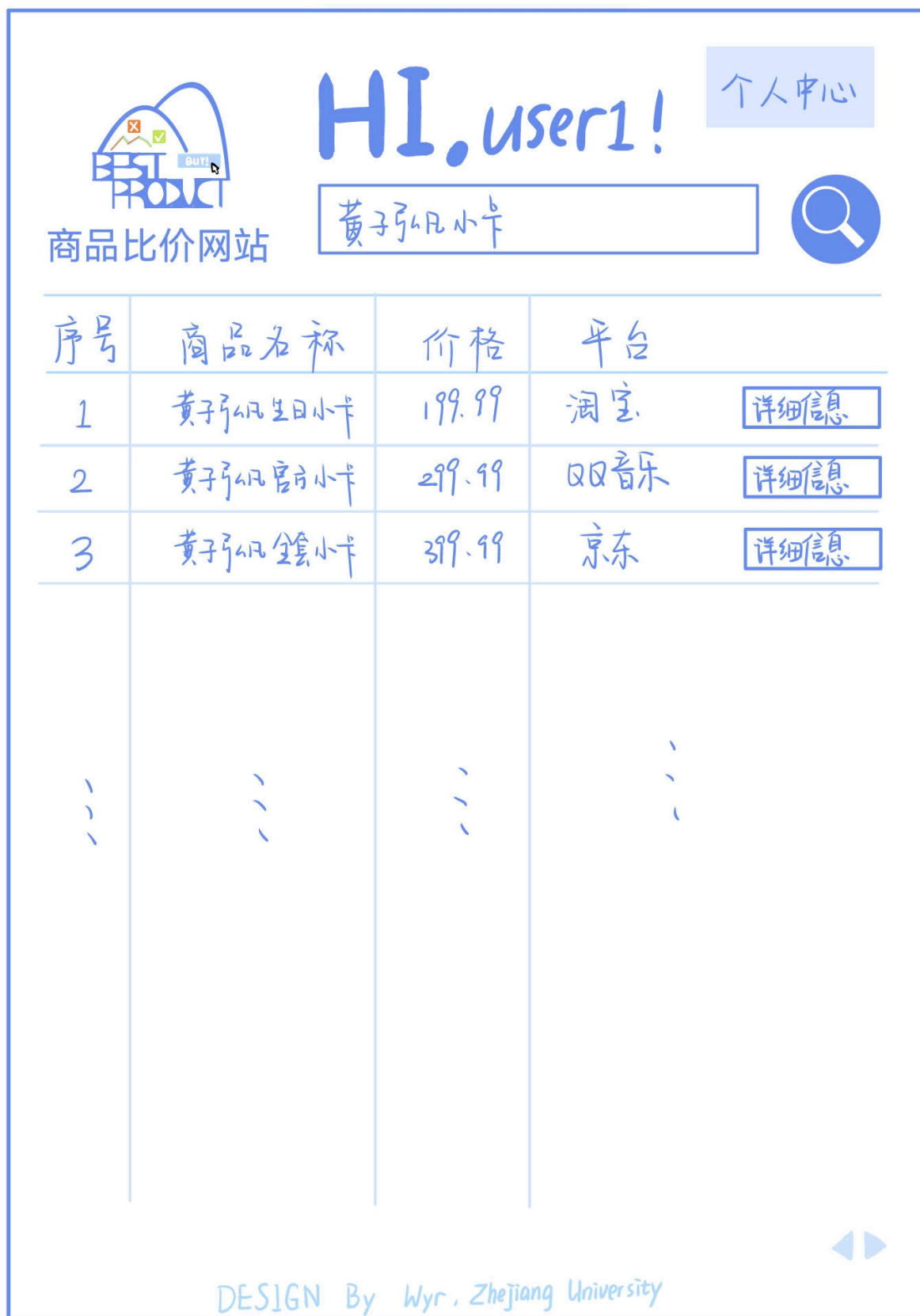


图8 主界面设计

## 6.7 商品信息界面



图9 商品信息界面设计

## 6.8 降价提醒推送框

点开即跳转至对应商品信息界面。



图 10 降价提醒推送框设计

## 7 系统出错处理

### 7.1 网络连接错误

- 现象：用户无法访问网站或请求超时。
- 原因：网络故障、服务器宕机、DNS 解析问题等。
- 防范措施：
  - 使用负载均衡器分散流量，避免单点故障。
  - 部署多区域、多云环境，提高系统的可用性。
  - 定期检查网络设备和线路，确保网络稳定。
- 应急预案：
  - 启用备用服务器和网络路径。
  - 监控系统实时告警，快速定位故障点。
  - 发布公告告知用户当前状况，并提供临时解决方案。

### 7.2 数据库错误

- 现象：数据查询失败、数据丢失、数据不一致等。
- 原因：数据库服务器故障、磁盘空间不足、SQL 注入攻击等。
- 防范措施：
  - 定期备份数据库，确保数据安全。
  - 使用主从复制和读写分离技术，提高数据库的可用性和性能。
  - 实施 SQL 注入防护，使用参数化查询。
- 应急预案：
  - 迅速切换到备用数据库服务器。
  - 恢复最近的数据库备份。

- 检查日志文件，定位并修复数据库错误。

### 7.3 应用程序错误

- 现象：页面显示异常、功能失效、系统崩溃等。
- 原因：代码 bug、内存泄漏、第三方服务不可用等。
- 防范措施：
  - 采用自动化测试，确保代码质量。
  - 实施代码审查，及时发现和修复潜在问题。
  - 设置资源限制，防止资源耗尽。
- 应急预案：
  - 重启应用程序服务。
  - 回滚到上一个稳定的版本。
  - 分析日志文件，定位并修复错误。

### 7.4 安全漏洞

- 现象：数据泄露、未经授权的访问、恶意攻击等。
- 原因：系统漏洞、弱密码、缺乏安全防护等。
- 防范措施：
  - 定期进行安全审计和漏洞扫描。
  - 使用防火墙和入侵检测系统。
  - 强制使用复杂密码，定期更换密码。
- 应急预案：
  - 立即切断受攻击的服务器与互联网的连接。
  - 启动安全应急响应团队，分析攻击来源和手段。
  - 修复安全漏洞，加强防护措施。

### 7.5 第三方服务中断

- 现象：依赖的第三方服务（如支付网关、物流查询等）不可用。
- 原因：第三方服务提供商故障、网络问题等。
- 防范措施：
  - 选择多个可靠的第三方服务提供商，实现冗余。
  - 设计容错机制，当第三方服务不可用时，提供替代方案或提示用户稍后再试。
- 应急预案：
  - 通知第三方服务提供商，请求技术支持。
  - 启用备用服务，确保核心功能正常运行。
  - 发布公告告知用户当前状况，并提供临时解决方案。

## 8 项目开发计划

| 开发时间          | 开发内容                |
|---------------|---------------------|
| 11.02 前       | 系统需求分析、总体架构设计、技术选型  |
| 11.03 - 11.10 | 前后端开发知识预备，撰写设计报告    |
| 11.11 - 11.24 | 前端界面 UI 设计与开发       |
| 11.25 - 12.08 | 数据库及后端接口设计与开发       |
| 12.09 - 12.15 | 完成前端与后端的集成测试        |
| 12.16 - 12.20 | 完成系统的部署与测试          |
| 12.21 - 12.23 | 完成系统测试文档、用户手册等文档的编写 |