## Java软件开发基础课程实验报告

# 基于JavaWeb的简易在线购物系统

#### 小组成员

1. 任奕圩

学院：计算机科学与技术，学号：PB24111587

工作内容：基于Vue的前端构建；前后端api的构建

1. 洪轶剑

学院：少年班学院，学号：PB24081582

工作内容：使用Java编写网页后端程序

#### 题目分析

**题目：线上超市购物系统要求实现的功能如下：**

a) 注册与注销账号

b) 区分管理员或用户登录系统

c) 所有操作均在图形用户界面实现

d) 实现商品信息的增加、删除、查找、修改等操作，区分管理员和用户的

权限

e) 用户可以使用购物车功能，也可以直接购买商品

f) 管理员可以管理所有订单，包括查看订单状态、修改订单等

g) 所有商品信息保存在文件中，可以通过修改文件来修改界面信息

h) 所有商品应该属于商品类，每个商品都应具有名称、类别、库存数量、生

产日期、保质期、价格等基本属性和其他可补充的属性，系统中应不少

于 20条商品信息

本项目为“**基于 JavaWeb 的简易在线购物系统**”，核心目标是实现一个具有用户注册登录、商品浏览、购物车、订单管理等功能的简易电子商务平台。系统应区分普通用户与管理员，具有图形化用户界面，并支持商品信息与购物车信息的增删改查操作。

核心问题主要包括以下几点：

1. 用户身份管理与权限控制  
   系统需支持用户的注册、登录与注销，并能够识别并区分管理员与普通用户的登录行为，从而实现权限隔离。例如，只有管理员可以进行商品管理和订单管理，而用户只能浏览、添加商品至购物车和提交订单。
2. 前后端交互与数据同步  
   由于本系统采用前后端分离的架构，前端由 Vue 构建图形化界面，后端使用 Java 语言编写逻辑处理代码，因此前后端的数据通信需设计合理的 API 接口，并采用统一的数据格式（如 JSON）进行传输。前后端数据同步是系统稳定运行的关键。
3. 商品与订单的增删改查功能设计  
   系统应支持商品的添加、删除、查询与修改，且要求商品至少包含名称、类别、库存、生产日期、保质期、价格等基本信息，并保留扩展属性的设计能力。同时，系统需支持用户下单与管理员查看、修改订单状态等操作。
4. 数据持久化与文件读写  
   所有商品信息和订单信息需要持久化存储。考虑到简易系统的特性，我们采用文件存储方式替代数据库，通过 JSON 格式对商品与订单进行结构化存储与读取。系统运行时加载数据，并在操作后保存更新，确保数据一致性。
5. 用户交互友好性与界面逻辑设计  
   系统所有操作需通过图形用户界面完成。Vue 框架提供了现代化的前端开发能力，可以快速构建动态响应式界面，提升用户体验。前端需与后端良好协作，确保操作结果实时反馈、状态可视化、界面友好。

解决方法概述如下：

1. 后端实现： 使用 Java 语言结合 Spark 框架编写后端服务，负责处理业务逻辑，如用户认证、商品管理、订单处理等，并提供 RESTful 风格的接口供前端调用。
2. 前端实现： 使用 Vue 框架构建图形用户界面，实现注册登录、商品浏览、购物车、下单等用户界面功能，通过 Axios 与后端 API 通信。
3. 数据传输与存储： 使用 JSON 作为前后端通信格式，同时用于在本地文件中进行商品与订单信息的持久化存储。
4. 模块划分与协作开发： 小组成员按模块划分任务，前后端并行开发，通过接口文档进行联调，保证项目协作效率与功能完整性。

通过上述设计思路，本实验不仅极大的提升了我们对 JavaWeb 技术体系的理解和应用能力，也提升了前后端协同开发与系统设计的实践水平。

#### 题目设计实现

在后端中，考虑使用User类和Product类处理，并以json格式保存数据，便于前后端数据传输

|  |
| --- |
| Class org.example.model.User |
| -id: String  -name: String  -email: String  -password: String  -group: Group  -cart: ArrayList<String> |
| //Standard getter and setter  +addCartItem(item: CartItem): void  +removeCartItem(item: CartItem): void |

|  |
| --- |
| Class org.example.model.Product |
| -id: String  -image: String  -title: String  -priceInteger: int  -priceDecimal: int  -category: Category  -description: String |
| //Standard getter and setter  +toString(): String |

其中Group和Category为枚举类：

|  |
| --- |
| Enum org.example.model.Group |
| ADMIN  CUSTOMER |

|  |
| --- |
| Enum org.example.model.Category |
| VIDEOCARD  CPU  MOTHERBOARD  RAM  STORAGE  POWERSUPPLY  COOLING  CASE  PERIPHERALS  PAD  MONITOR  LAPTOP  GRAPHIC\_TABLET |

对于用户添加到购物车中的商品，使用CartItem类

|  |
| --- |
| Class org.example.model.CartItem |
| -id: String  -image: String  -title: String  -priceInteger: int  -priceDecimal: int  -quantity: int  -selected: boolean |
| //Standard getter and setter  +toString(): String |

CartController类实现用户对购物车的操作（添加商品、修改商品数量、选中商品等）

|  |
| --- |
| Class org.example.api.CartController |
| +getCart(req: Request, res: Response): Object  +addToCart(req: Request, res: Response): Object  +updateCartItem(req: Request, res: Response): Object  +deleteCartItem(req: Request, res: Response): Object  +bathcDeleteCartItem(req: Request, res: Response): Object  +clearCart(req: Request, res: Response): Object  +selectAllCartItems(req: Request, res: Response): Object |

ProductController类实现管理员对商品的添加或修改：

|  |
| --- |
| Class org.example.api.ProductController |
| +getAllProducts(req: Request, res: Response): Object  +getProductById(req: Request, res: Response): Object  +addToProducts(req: Request, res: Response): Object  +updateProduct(req: Request, res: Response): Object |

AppConfig类

|  |
| --- |
| Class org.example.config.Appconfig |
| +PRODUCTS\_FILE: String  +CART\_FILE: String  +USERS\_FILE: String  +DATA\_DIR: String |
| +getPoint(): int  +isDevelopment(): boolean |

JsonIO类实现json格式文件的读写

|  |
| --- |
| Class org.example.repository.JsonIO |
| -cartFileLock: ReadWriteLock  -GSON: Gson  -DATA\_DIR: String |
| +getDataFile(filename: String): File  +readJsonList(filename: String, typeToken: Type): List<T>  +writeJsonList(filename: String, list: List<T>, typeToken: Type): void  +readCartItems(filename: String): List<CartItem>  +writeCartItems(filename: String, cartItems: List<CartItem>): void  +readUsers(filename: String): List<User>  +writeUsers(filename: String, users: List<User>): void  +readProducts(filename: String): List<Product>  +writeProducts(filename: String, products: List<Product>): void  +readKeywords(keywordFile: String): Map<String,List<String>>  +testReadProducts(filename: String): Product[] |

对于前后端对接，

Router类

|  |
| --- |
|  |
|  |

App类

|  |
| --- |
| Class org.example.App |
| -GSON: Gson |
| +main(args: String[]): void  +getGson(): Gson  -setupStaticFileService(): void  -setupCors(): void |

在前端中，使用Vue来编写页面内容，Axios来进行http传输。通过前后端的有关逻辑，实现了美观的交互界面和便捷人性化的交互方式。

#### 实验结果

本项目非常完整的实现了一个电子商务购物平台的绝大多数功能，包括

1. 在首页展示商品卡片
2. 商品分类与搜索
3. 用户登录、注册、退出登录、注销账号
4. 购物车的查看，管理购物车内商品，包括修改数量，删除商品，全选等
5. 发起订单，查看订单状态
6. 清晰便捷的导航栏
7. 用户个人中心，可以管理个人资料、收货地址、购物车、修改密码、注销账号等
8. 管理员账号的登录、注册、安全退出，以及管理员权限验证等安全措施
9. 系统控制台页面，可以查看当前商品总数，订单状态
10. 商品的增加、编辑、删除
11. 美观的UI设计

具体内容在讲解视频中有详细介绍。

#### 个人总结

1. 任奕圩：在本项目中负责前端和前后端的api。通过本次项目的实践，我的工程能力得到了巨大的提升。对于从未接触过JavaWeb项目的新手来说，我们不仅学会了从0开始组织起项目结构，还完整实现了从前端到后端的全栈开发，这样的成果还是相当令我满意的。在与同伴一起推进项目内容的过程中，也确实让我感受到了前后端并行开发的优越性。在今后，我对这个项目的构想是，使用爬虫等技术实时从各大电商平台上获取商品价格，为用户筛选出最优价，并部署到网络，让这个项目具有更大的实用价值。

总而言之，这次项目让我受益良多，这门课也是我整个大一下学期里我认为的收获最大的课程。

1. 洪轶剑：我在此次项目中主要负责后端搭建，这次的工程能正常跑起来，网页也能用，确实很有成就感，同时还在与同伴共同搭建项目时学会了使用github。但是现在回头看这些后端的代码确实还有很多不足：（1）数据结构能够进一步优化，例如我们为了保存用户的购物车，单独创建了CartItem类，后来才发现这个类是不必要的，可以为User添加一个Map<String,Int>来实现；（2）在JsonIO类中，我们为了User类、Product类、CartItem类都单独写了read方法和write方法，但这应该是可以用泛型来统一成一个read和write方法的。总之这个项目虽然能运行但还有非常多亟须改进的地方。

#### 附件

源代码：见附件1

系统演示视频：见附件2