利用Java、VUE等语言实现的简易在线购物系统

作者：任奕圩（学院：计算机科学与技术，学号：PB24111587，分工：前端及前后端对接）

洪轶剑（学院：少年班学院，学号：PB24081582，分工：后端）

题目分析：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*

我们考虑使用java实现后端功能（如创建用户、添加商品、添加到购物车等功能），使用VUE实现前端功能（如图形用户界面、图形购物界面），使用json格式存储传输数据，使用spark和axios在前后端传输数据。

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，

实现过程：起初使用fastjson库来实现json文件读写，但由于未知原因无法在我们的项目里跑起来，于是换用Gson库，且成功运行。

实验结果：

个人总结：

任奕圩：

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。

源代码：

系统演示视频：