

Java practical test challenge

Create a marketplace application with the help of Java language

There will be 3 different levels of task completion difficulty. Please choose the one, you are confident in.

Basic system requirements:

- System from the start has 3 products that can be sold. Each product contains:
 1. Id
 2. Name
 3. Price
- System from the start has 3 users that can buy products. Each user has:
 1. Id
 2. First name
 3. Last name
 4. Amount of money
- System has menu that supports next operations:
 1. Display list of all users
 2. Display list of all products
 3. User should be able to buy product, to do this operation we should enter: Id of user who want to buy product Id of product which user want to buy
 - If user doesn't have enough money to buy product, throw exception
 - If user successfully bought the product display message about successful purchase
 - When user is buying product, his amount of money decreases by product price
 - After successful purchase, information about user and his products has to be stored in collection best suited for that purpose
 4. Display list of user products by user id (If user didn't buy anything yet, don't show anything)
 5. Display list of users that bought product by product id (If nobody bought this product yet, don't show anything)

Medium system requirements:

- System starting without users and products, add next operations to menu:
 1. Ability to add new users to the system (with autogenerated id, not null fields validation and check for numbers in amount of money field)
 2. Ability to add new products to the system (with autogenerated id, not null fields validation and check for numbers in price field)
- Ability to delete products and users, if product deleted, it have to disappear from all users purchases
- If exception is thrown, don't crash application and show menu again, display error message to understand reason of error

Advanced system requirements:

- Add Spring Boot to application
- Add relational database and Hibernate to application for persistence storage
- Get rid of console menu and move all the logic to REST endpoints
- Implement global error handling and user friendly exception messages with correct HTTP statuses

When building the application try to create readable, self-documented code. If possible, follow clean code rules. It will be nice to use Dependency Injection principle and SOLID principles, and to divide your application by layers for better readability.