```
import java.util.ArrayList;
import java.util.List;

abstract class Product {
    protected String name;
    protected double price;
    protected int quantity;

    public Product(String name, double price, int quantity) {
        this.name = name;
        this.price = price;
        this.quantity = quantity;
    }

    public abstract double calculateTotalPrice();

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public int getQuantity() {
        return quantity;
    }
}

class Electronics extends Product {
    private String brand;

    public Electronics(String name, double price, int quantity, String brand) {
        super(name, price, quantity);
        this.brand = brand;
    }

    @Override
    public double calculateTotalPrice() {
        return price * quantity;
    }

    public String getBrand() {
        return brand;
    }
}

class Clothing extends Product {
    private String size;

    public Clothing(String name, double price, int quantity, String size) {
        super(name, price, quantity);
```

```
        this.size = size;
    }

    @Override
    public double calculateTotalPrice() {
        return price * quantity;
    }

    public String getSize() {
        return size;
    }
}

class Book extends Product {
    private String author;

    public Book(String name, double price, int quantity, String author) {
        super(name, price, quantity);
        this.author = author;
    }

    @Override
    public double calculateTotalPrice() {
        return price * quantity;
    }

    public String getAuthor() {
        return author;
    }
}

class ShoppingCart {
    private List<Product> products;

    public ShoppingCart() {
        products = new ArrayList<>();
    }

    public void addProduct(Product product) throws IllegalArgumentException {
        if (product == null || product.getQuantity() <= 0) {
            throw new IllegalArgumentException("Invalid product or quantity.");
        }
        products.add(product);
    }

    public double calculateTotal() {
        double total = 0;
        for (Product product : products) {
            total += product.calculateTotalPrice();
        }
        return total;
    }
```

```
    public void viewCart() {
        if (products.isEmpty()) {
            System.out.println("Your shopping cart is empty.");
            return;
        }
        System.out.println("Products in your cart:");
        for (Product product : products) {
            System.out.println(product.getName() + " - " + product.getQuantity() + " x $" +
product.getPrice() + " = $" + product.calculateTotalPrice());
        }
    }

    public List<Product> getProducts() {
        return products;
    }
}

class User {
    private String username;
    private String email;
    private ShoppingCart shoppingCart;

    public User(String username, String email) {
        this.username = username;
        this.email = email;
        this.shoppingCart = new ShoppingCart();
    }

    public String getUsername() {
        return username;
    }

    public String getEmail() {
        return email;
    }

    public ShoppingCart getShoppingCart() {
        return shoppingCart;
    }

    public void viewCart() {
        System.out.println("User: " + username);
        shoppingCart.viewCart();
    }

    public void finalizeCart() {
        double total = shoppingCart.calculateTotal();
        System.out.println("Finalizing Cart for " + username);
        System.out.println("Total cost: $" + total);
    }
}
```

```
public class OnlineStoreSystem {
    public static void main(String[] args) {
        try {
            Electronics phone = new Electronics("Smartphone", 699.99, 2, "BrandX");
            Clothing shirt = new Clothing("T-shirt", 19.99, 3, "M");
            Book book = new Book("Java Programming", 49.99, 1, "John Doe");

            User user = new User("johndoe", "johndoe@example.com");

            user.getShoppingCart().addProduct(phone);
            user.getShoppingCart().addProduct(shirt);
            user.getShoppingCart().addProduct(book);

            user.viewCart();
            user.finalizeCart();

        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
s
```