B2B Marketplace System Design Documentation

MEMBERS:

Mr Samkeliso Mlotsa– project manager and designer 202203737

Mr Sisekelo Dlamini – programmer and designer 202200948

Ms Nontokozo Djokoto – system architect and designer 202202834

Table of Contents

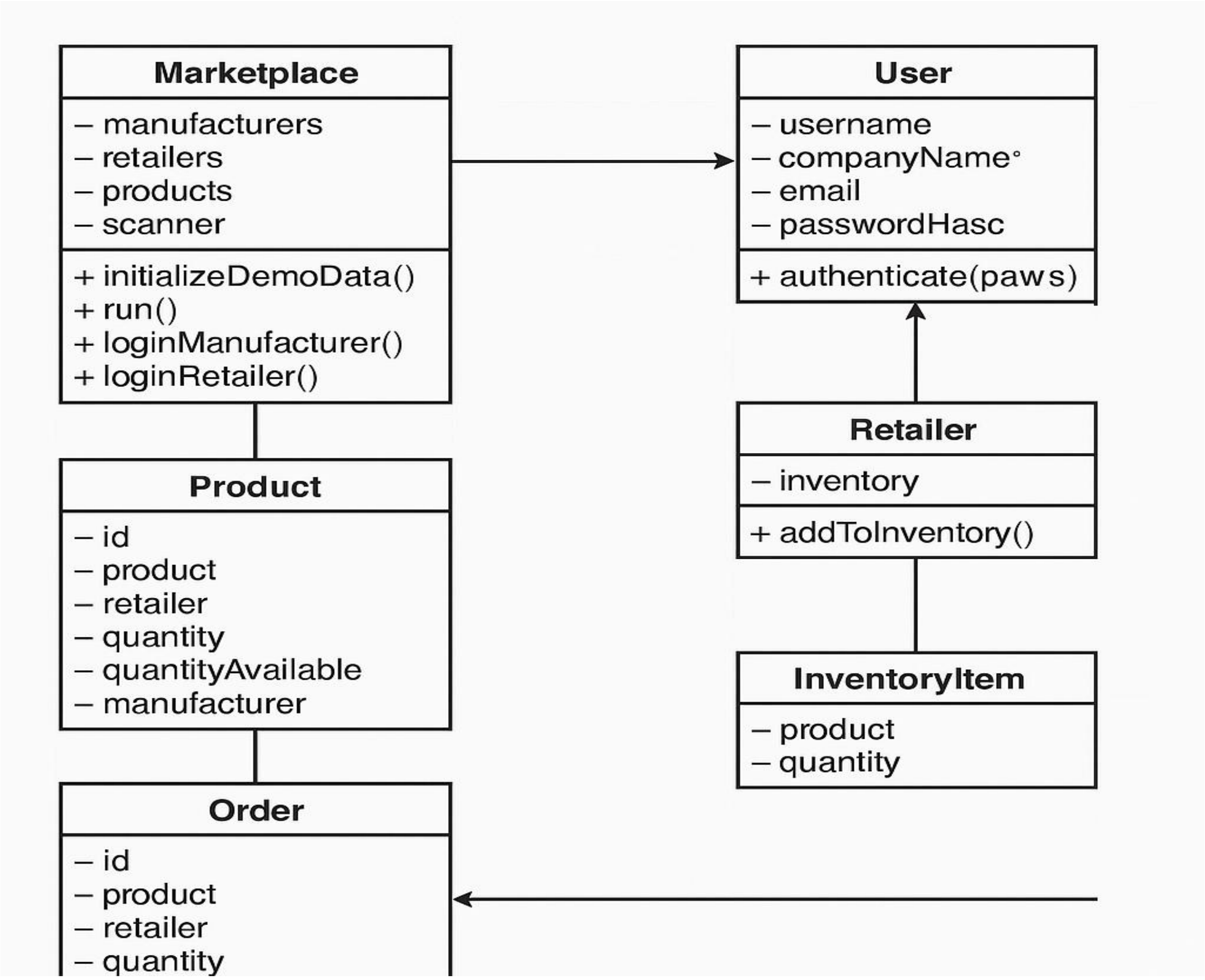
System Overview

The B2B Marketplace is a console-based application that connects manufacturers with retailers, facilitating the digitalization of supply chain operations. The system provides:

- Manufacturer capabilities:
  - Product listing and inventory management
  - Order fulfillment tracking
  - Production capacity monitoring

- Retailer capabilities:
  - Product browsing and ordering
  - Inventory tracking
  - Order history review

Class Diagram



Core Components

1. Marketplace Class

- Main controller class that orchestrates the application flow
- Manages all system data (users, products, orders)
- Provides the main menu and authentication routing

2. User Hierarchy

- **User (Abstract Base Class)**
  - Common properties for all users (username, company, email)
  - Basic authentication functionality
- **Manufacturer**
  - Specialized with production capacity tracking
  - Manages product listings
- **Retailer**
  - Maintains inventory of purchased products
  - Places orders with manufacturers

3. Product Management

- Tracks available products with quantity
- Maintains manufacturer relationship
- Supports inventory adjustments during ordering

4. Order Processing

- Tracks order lifecycle
- Calculates order totals

- Links products to retailers

Data Model

Entities and Relationships:

- Manufacturer (1) → (N) Product
- Retailer (1) → (N) Order
- Order (1) → (1) Product
- Retailer (1) → (N) InventoryItem
- InventoryItem(1) → (1) Product

Key Fields:

- Product: SKU, name, description, price, quantity
- Order: Timestamp, quantity, status, total amount
- User: Credentials, company information
- Manufacturer: Production capacity

Business Logic

Order Processing Flow:

1. Retailer selects product and quantity
2. System checks product availability
3. If available:
   - Order is created with PROCESSING status
   - Product quantity is reduced
   - Inventory item is added to retailer
4. Manufacturer can view and fulfill orders
5. Order status progresses through lifecycle

Inventory Management:

- Real-time quantity tracking
- Prevention of overselling (quantity checks)
- Automatic inventory updates for retailers

User Flows

Manufacturer Journey:

1. Log in with credentials
2. View pending orders
3. Check production capacity utilization
4. Fulfill orders by updating status
5. Log out

Retailer Journey:

1. Log in with credentials
2. Browse available products
3. Place order with desired quantity
4. View order history and status
5. Check current inventory
6. Log out

Security Considerations

1. Authentication:
   - Simple password hashing (demonstration only)

- Username/password verification

2. Data Protection:
   - All sensitive data stored in memory only
   - No persistence layer in current implementation

3. Access Control:
   - Strict separation of manufacturer/retailer views
   - Users can only see their own orders/inventory

Limitations and Future Enhancements

Current Limitations:

1. Console-based interface limits user experience
2. In-memory data storage (no persistence)
3. Simplified authentication (no password encryption)
4. Basic error handling

Recommended Enhancements:

1. Web Interface:
   - Convert to Spring Boot web application
   - Add HTML/JS frontend
2. Persistence Layer:
   - Integrate database (MySQL, PostgreSQL)
   - Implement proper data storage
3. Enhanced Security:
   - Add proper password hashing (BCrypt)

  - Implement session management

4. Additional Features:

  - Product search and filtering

  - Advanced reporting

  - Shipping/tracking integration

  - Payment processing

5. Scalability Improvements:

  - Microservices architecture

  - API endpoints for integration

Deployment Notes

The current implementation is designed to run as a standalone Java application. To execute:

1. Compile: `java B2BMarketplace.java`

2. Run: `java B2BMarketplace`

The system initializes with demo data for immediate testing:

- 2 manufacturers

- 2 retailers

- 3 sample products