Java Final Sprint (SD11)
Java Console-Based E-Commerce Platform

Students: Chelsea Slade (Mayne), Ashley Legge, Allison Boone
Due: December 11, 2024

Table of Contents

User Documentation

1.1 E-Commerce Platform Overview/Menu Options Explanation

The E-Commerce Platform is a console-based application simulating an online marketplace. It offers a user-friendly menu system, allowing individuals to interact within a realistic market environment. Users can register under three distinct roles: buyer, seller, or administrator, each with tailored privileges. The role-based system ensures a secure and efficient marketplace experience.

Key Features:
- **User Registration & Authentication:** Users can register with the platform, selecting their desired role. Securing passwords hashing (using BCrypt) ensures data protection.
- **Role-Based Control:** The system employs a role-based access control, granting each user (buyer, seller, administrator) access to their relevant functionalities and features.
- **Product Management:**
  - **Buyers:** Can browse the available products, search & view information.
  - **Sellers:** Can list their products for sale, including details like name, price, quantities. Additionally, they can manage their listings by updating or deleting items.
  - **Administrators:** Can view all products in the system including the seller's information.
    - **User Management:**
      - Administrators can also view all users including their details and have the authority to delete accounts.

Menu Options:
- **Buyer Menu:**
  - Browse Products
  - Search Products
  - View Product Detail
- **Seller Menu:**
  - Add Products
  - Manage Products
  - View All Listed Products
- **Admin Menu:**
  - View Users
  - Delete Users
  - View Products

User Class and UserDAO

The User class serves as the base for all user accounts within the platform, containing essential attributes such as the user ID, username, password, email address, and role (e.g., buyer, seller, or admin).

- The UserDAO class is responsible for handling all database interactions related to user data. It manages CRUD operations for user accounts, such as registering new users, retrieving account information, updating profiles, and deleting user records.

Buyer Class

The Buyer class extends from the User class, inheriting its attributes and methods. It adds functionality specific to buyers, including the ability to browse available products, search for items, and view detailed product information.

Seller Class

The Seller class also inherits from the User class, providing additional methods tailored to sellers. Sellers can use this class to add new products to the marketplace, modify existing product listings, remove products, and view a list of all products they have listed for sale.

Admin Class

The Admin class, as a subclass of User, grants the highest level of access within the platform. Admins can view all registered users, manage user accounts (including deleting users), and oversee the entire product catalog, including information about each seller and their products.

Product Class and ProductDAO

The Product class represents individual items available for purchase in the marketplace. It contains attributes such as product ID, name, price, quantity, and the associated seller's ID. This class encapsulates the details that define a product within the platform.

- The ProductDAO class manages all interactions with the database related to product information. It handles CRUD operations for products, including adding new products to the database, retrieving product details, updating product listings, and removing products from the marketplace.

ProductService

The ProductService class serves as an intermediary between the logic layer and the data access layer. It interacts with the ProductDAO to perform various product-related tasks such as adding new products, updating existing ones, and managing product inventories.

UserService

The UserService class contains the core business logic for managing user accounts. It is responsible for handling user registration, authenticating login credentials, managing roles (assigning roles such as buyer, seller, or admin to users), and updating user profiles. This class ensures that user-related processes are carried out efficiently and securely.
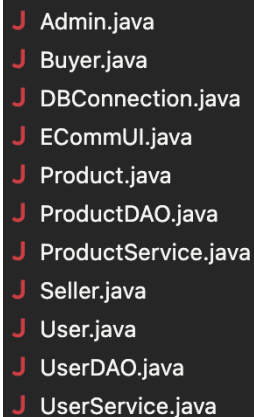
1.3 Class Diagram
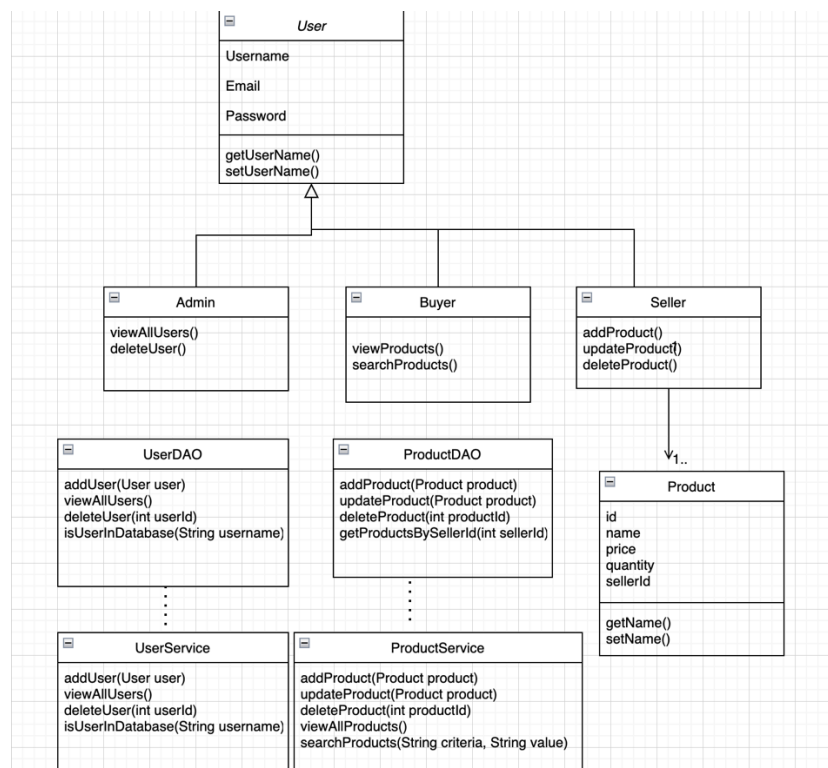
Development Documentation

2.1 Source Code Description

The E-Commerce platform's source code is organized into a well-defined structure. Classes like *Buyer, Seller, Admin* represent different user roles with specified functionalities. The *Product* class models product information, while *ProductDAO* and *ProductService* handles data access and logistics for products. Similarly, *User, UserDAO,* and *UserService* manages user-related operations. This clear separation of concerns enhances the codes maintainability and promotes a simple structure.

```
J  Admin.java
J  Buyer.java
J  DBConnection.java
J  ECommUI.java
J  Product.java
J  ProductDAO.java
J  ProductService.java
J  Seller.java
J  User.java
J  UserDAO.java
J  UserService.java
```

2.2 Development Standards

- Version control through GitHub with detailed commit messages and branching.
- Classes named according to PascalCase.
- Methods and variables named according to camelCase.
- Code clearly commented throughout for readability.

2.3 Database Theory

The platform utilizes a PostgreSQL database to be efficient in storing and managing user & product data. The database allows several different benefits:
- Ensures that user and product information is preserved, preventing data loss between sessions
- Enforces data consistency through constraints and relationships.
- Handles a growing number of users and products, supporting the platform's expansion.
- Incorporates security measures, including user authentication and access controls, to protect sensitive user and product information.

Deployment Documentation

3.1 Installation Details

Source code for the program can be found at: https://github.com/chelseaslade/Java-Sprint-1/tree/main

To download the code, select the green "code" button, and then copy the link under HTTPS (the repository URL). Enter your chosen IDE (ex. Visual Studio Code), open the terminal, and enter:

<git clone *insert URL here*>

This will clone the repository to your local system and allow you to run and edit the code.

Detailed instructions to clone from GitHub located here:
https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository

Once the files are accessible in the IDE, navigate to the ECommUI.java file and run it. This will display the main menu of the program in the IDE terminal and prompt the user to make inputs to choose various options.