E-Commerce Platform - Comprehensive Documentation

1. User Documentation

Overview

The E-Commerce Platform is a console-based application designed to emulate an online marketplace. It allows users to register as buyers, sellers, or admins and provides role-based functionality for managing products and users.

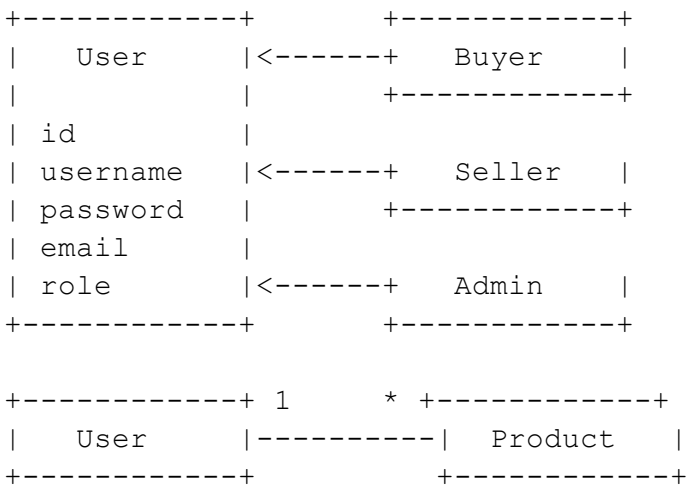
Application Classes

Main Classes

1. **App**
    - **Purpose:** Entry point of the application.
    - **Key Functions:**
        - Initializes the application.
        - Handles user login, registration, and navigation to role-specific menus.
        - Provides prompts based on user roles (buyer, seller, or admin).
2. **User**
    - **Purpose:** Represents a user in the system.
    - **Attributes:**
        - user_id: Unique identifier for the user.
        - username: User's unique name.
        - password: Encrypted password.
        - email: Contact email address.
        - role: Role of the user (buyer, seller, or admin).
    - **Derived Classes:**
        - Buyer: Subclass of User for buyer-specific functionality.
        - Seller: Subclass of User for seller-specific functionality.
        - Admin: Subclass of User for admin-specific functionality.
3. **Product**
    - **Purpose:** Represents a product listed by sellers.
    - **Attributes:**
        - product_id: Unique identifier for the product.

        - name: Name of the product.
        - price: Price of the product (NUMERIC).
        - quantity: Available quantity of the product.
        - seller_id: ID of the seller who listed the product.
4. **UserDAO and ProductDAO**
    - **Purpose**: Perform CRUD operations on users and products tables in the database.
    - **Key Functions**:
        - UserDAO: Handles user authentication, registration, and data retrieval.
        - ProductDAO: Manages product addition, updates, deletion, and retrieval.
5. **DBConnection**
    - **Purpose**: Manages database connectivity for the application.
    - **Key Features**:
        - Establishes connection to the PostgreSQL database.
        - Ensures proper resource cleanup.

Class Diagram



How to Start

1. Clone the repository:
   git clone https://github.com/your-repo/ecommerce-platform.git
2. Navigate to the project directory and build the application:
   mvn clean install
3. Start the application:
   java -cp target/ecommerce-platform-1.0-SNAPSHOT.jar com.ecommerce.App

---

2. Development Documentation

Directory Structure

The following structure represents an e-commerce platform:

```
ecommerce-platform
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── com
│   │   │   │   ├── ecommerce
│   │   │   │   │   ├── App.java          # Entry point of the application
│   │   │   │   │   ├── dao
│   │   │   │   │   │   ├── UserDAO.java  # Handles user-related database
│   │   │   │   │   │   ├── ProductDAO.java   # Handles product-related
database
│   │   │   │   │   ├── model
│   │   │   │   │   │   ├── User.java     # Represents a user
│   │   │   │   │   │   ├── Product.java  # Represents a product
│   │   │   │   │   ├── util
│   │   │   │   │   │   ├── DBConnection.java # Manages database
connections
│   ├── test
│   │   ├── java
│   │   │   ├── com
│   │   │   │   ├── ecommerce
│   │   │   │   │   ├── ProductDAOTest.java   # Unit tests for ProductDAO
│   │   │   │   │   ├── UserDAOTest.java      # Unit tests for UserDAO
├── pom.xml                                   # Maven configuration file
├── create_tables.sql             # SQL script to set up the database
tables
├── README.md                     # Documentation for the project
```

Build Process

1. Ensure Maven is installed.
2. Navigate to the project directory.
3. Run the build command:
   `mvn clean install`

Development Standards

- **Architecture**: Follow the MVC design for modularity and scalability.
- **Coding Standards**:

- Use meaningful class and method names.
- Include Javadoc comments for all public methods.
- Maintain consistent formatting and indentation.

- **Error Handling:**
  - Catch and log exceptions using SLF4J/Logback.
  - Ensure database resources are closed properly.

Setting Up the Database

1. **Create Database:**
   CREATE DATABASE ecommerce_db;

Create Tables:

```
-- Create the users table
CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,           -- Unique ID for each user
    username VARCHAR(50) NOT NULL UNIQUE, -- Unique username
    password VARCHAR(255) NOT NULL,    -- Hashed password
    email VARCHAR(100) NOT NULL UNIQUE, -- Email address
    role VARCHAR(10) NOT NULL          -- Role: buyer, seller, or admin
);

-- Create the products table
CREATE TABLE products (
    product_id SERIAL PRIMARY KEY,     -- Unique ID for each product
    name VARCHAR(100) NOT NULL,        -- Product name
    price DECIMAL(10, 2) NOT NULL,     -- Product price
    quantity INT NOT NULL,             -- Quantity available
    seller_id INT NOT NULL,            -- Foreign key linking to the
seller
    FOREIGN KEY (seller_id) REFERENCES users (user_id) ON DELETE CASCADE
);
```

Create Indexes:

```
Create indexes for performance
CREATE INDEX idx_users_username ON users (username);
CREATE INDEX idx_users_email ON users (email);
CREATE INDEX idx_products_name ON products (name);
```

Insert users data:

```
INSERT INTO users (username, password, email, role) VALUES
('John', 'John123', 'John123@gmail.com', 'buyer'),
('Tom', 'Tom123', 'Tom123@gmail.com', 'seller'),
('Robert', 'Robert123', 'Robert123@gmail.com', 'admin');
```

Insert products data:

```
INSERT INTO products (name, price, quantity, seller_id) VALUES
```

```
('Laptop', 999.99, 10, 2),  -- seller1's product
('Phone', 499.99, 20, 2),
('Headphones', 49.99, 100, 2)
```

2. **Configure Database Connection:**

```
Update DBConnection.java with your PostgreSQL credentials:
private static final String URL =
"jdbc:postgresql://localhost:5432/ecommerce_db";
private static final String USER = "postgres";
private static final String PASSWORD = "***********";
```

Getting the Source Code

- Clone the repository:
  `git clone https://github.com/your-repo/ecommerce-platform.git`

---

3. Deployment Documentation

Installation Steps

1. **Install Prerequisites:**
   - Java JDK 11 or higher.
   - Maven 3.8.1 or higher.
   - PostgreSQL database server.
2. **Set Up the Database:**
   - Follow the steps outlined in the development documentation.
3. **Build the Application:**
   - Navigate to the project directory and execute:
     `mvn clean install`
4. **Run the Application:**
   - Start the application using:
     `java -cp target/ecommerce-platform-1.0-SNAPSHOT.jar com.ecommerce.App`

Verifying Installation

1. **Test Registration:**
   - Register a user and verify that the users table is updated.
2. **Test Product Management:**

    - Log in as a seller and add products to ensure the products table is updated.
3. **Test Role-Specific Features**:
    - Confirm functionality for buyers, sellers, and admins through their respective menus.

Appendix

References

- Java Documentation
- Maven Documentation
- PostgreSQL Documentation