**Documentation Group 12 Java Final**

**Sprint**

**User Documentation:**

This application is a Java console-based E-Commerce platform. It allows users to register as buyers, sellers, or admins, and perform various role-specific actions such as browsing products, adding products, and managing users.

**Classes and Their Working**

1. **User (Abstract Class):**
   * Represents a generic user with attributes like user\_id, username, password, email, role, and role\_id.
   * Provides getters, setters, and a toString() method for a string representation.
2. **Buyer (Class):**
   * Inherits from User.
   * Represents a buyer with all attributes from the User class.
   * Overrides toString() for a string representation.
3. **Seller (Class):**
   * Inherits from User.
   * Represents a seller with all attributes from the User class.
   * Overrides toString() for a string representation.
4. **Admin (Class):**
   * Inherits from User.
   * Represents an admin with all attributes from the User class.
   * Overrides toString() for a string representation.
5. **Product (Class):**
   * Represents a product with attributes like id, name, price, quantity, and sellerId.
   * Provides getters, setters, and a toString() method for a string representation.
6. **UserService (Class):**
   * Handles user-related operations such as registration, authentication, and user management.
   * Methods include registerUser, authenticateUser, getUserByUsername, getAllUsers, and deleteUser.
7. **ProductService (Class):**
   * Handles product-related operations such as adding, updating, and deleting products.
   * Methods include addProduct, updateProduct, deleteProduct, getProductById, getAllProducts, searchProductsByName, and getProductsBySeller.
8. **Menu (Class):**
   * Provides the console-based user interface for interacting with the application.
   * Methods include displayMenu, registerUser, loginUser, displayRoleBasedMenu, displayBuyerMenu, displaySellerMenu, displayAdminMenu, browseProducts, searchProduct, viewProductInfo, addProduct, updateProduct, deleteProduct, viewMyProducts, viewAllUsers, deleteUser, and viewAllProducts.

**Use the following command in the terminal to run the project:**  
mvn exec:java -Dexec.mainClass="com.ecommerce.cli.Menu"

**Development Documentation:**

To view the JavaDocs for this project open the index.html file located in target/site/apidocs.

**Directory Structure**

src/

main/

java/

com/

ecommerce/

cli/

Menu.java

test/

java/

com/

* **src/**: The root directory for the source code.
  + **main/**: Contains the main application source code.
    - **java/**: The root directory for Java source files.
      * **com/**: The base package for the project.
        + **ecommerce/**: The package for the e-commerce platform.

**cli/**: Contains classes related to the command-line interface.

**Menu.java**: Provides the console-based user interface for interacting with the application.

* + **test/**: Contains the test source code.
    - **java/**: The root directory for Java test files.
      * **com/**: The base package for the test code.

This structure follows the standard Maven project layout.

**Build Process**

**Compile the Project:**

* + Navigate to the root directory of the project.
  + Use Maven to compile the project: **mvn clean compile**

**Compiler Time Dependencies**

The project uses Maven for dependency management. Dependencies are defined in the [pom.xml](vscode-file://vscode-app/Applications/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) file. Common dependencies might include:

* + JUnit for testing
  + Log4j for logging
  + MySQL Connector for database connectivity

**Setting Up a Database for Development**

1. Install MySQL:
   * On macOS, you can use Homebrew: **brew install mysql**
2. Start MySQL Service: **brew services start mysql**
3. Create a Database:

Log in to MySQL

**mysql -u root -p**

Create a new database:

**CREATE DATABASE ecommerce\_dev;**

1. Configure Database Connection:

Update the application.properties or application.yml file with your database connection details:

**spring.datasource.url=jdbc:mysql://localhost:3306/ecommerce\_dev**

**spring.datasource.username=ecommerce\_user**

**spring.datasource.password=ecommerce\_password**

**Getting the Source Code from the Repository**

1. Clone the Repository:
   * Use Git to clone the repository:

**git clone https://github.com/BrianJackman/Java-Final-Sprint**

1. Navigate to the Project Directory:

**cd ecommerce-project**

1. Install Dependencies:
   * Use Maven to install the project dependencies:

**mvn clean install**

This setup should help you get started with the development process for your project.

**Deployment Documentation:**

This document provides the steps needed to install and run the e-commerce application.

**Prerequisites**

1. Java Development Kit (JDK): Ensure JDK 11 or higher is installed.
2. Apache Maven: Ensure Maven is installed.
3. MySQL: Ensure MySQL is installed and running.

**Steps to Deploy the Application**

Clone the Repository - Open a terminal and run:

**git clone**[**https://github.com/BrianJackman/Java-FinalSprint**](https://github.com/BrianJackman/Java-FinalSprint)

Navigate to the project directory: **cd ecommerce-project**

Set Up the Database - Start MySQL service: **brew services start mysql**

Log in to MySQL:

**mysql -u root -p**

Create a new database:

**CREATE DATABASE ecommerce\_dev;**

Create a new user and grant privileges:

**CREATE USER 'ecommerce\_user'@'localhost' IDENTIFIED BY 'ecommerce\_password';**

**GRANT ALL PRIVILEGES ON ecommerce\_dev.\* TO 'ecommerce\_user'@'localhost';**

**FLUSH PRIVILEGES;**

Configure Application Properties – Update the application.properties or application.yml file with your database connection details:

**spring.datasource.url=jdbc:mysql://localhost:3306/ecommerce\_dev**

**spring.datasource.username=ecommerce\_user**

**spring.datasource.password=ecommerce\_password**

Build the Project - Use Maven to build the project:

**mvn clean install**

Run the Application - Use Maven to run the application:

**mvn exec:java -Dexec.mainClass="com.ecommerce.cli.Menu"**

Interact with the Application - The application will start in the terminal, and you can interact with it using the CLI.

**Group Members: Brian Jackman, Luke Metcalfe, Morgan Browne.**

**Contributions:**

**Brian Jackman:**

-Set up initial project structure

-Set up SQL database

-Created CLI menu and compiled all the code  
-Handled debugging

-Documentation

**Luke Metcalfe:**

-User.java

-Buyer.java

-Seller.java

-Admin.java

-UserDAO.java

-UserService.java

-Documentation

**Morgan Browne:**

-Product.java

-ProductDAO.java

-ProductService.java

-Worked on CLI menu

-Documentation