Advanced Programming Java

SD10 S3 Sprint 1

**Java E-Commerce Platform**

**System Documentation**

Document Version: 1.0.0

Date: 08/09/2024

**Contents**

[1 DOCUMENT MANAGEMENT 3](#_Toc14176)

[1.1 Contributors 3](#_Toc14177)

[1.2 Version Control 3](#_Toc14178)

[2 USER DOCUMENTATION 4](#_Toc14179)

[2.1 Classes & Functionality 4](#_Toc14180)

[2.2 UML Diagrams 5](#_Toc14181)

[2.3 How to Start/Access the Application 5](#_Toc14182)

[3 DEVELOPMENT DOCUMENTATION 5](#_Toc14183)

[3.1 Javadoc’s 5](#_Toc14184)

[3.2 Source Code Structure 5](#_Toc14185)

[3.3 Build Process 6](#_Toc14186)

[3.4 Complier Time Dependencies 6](#_Toc14187)

[3.5 Development Standards 6](#_Toc14188)

[3.6 Setting up the Database for Development 6](#_Toc14189)

[3.7 Getting the Source Code from the Repository 7](#_Toc14190)

[4 DEPLOYMENT DOCUMENTATION 7](#_Toc14191)

# Documentation Management

This project is a Java application designed to manage users (Admin, Buyer, Seller) and computer parts within an e-commerce platform. The system provides functionalities for user management, role-based access, and detailed management of computer parts (items) including adding, editing, and removing parts. The platform is built using Java and PostgreSQL and is console-based.

## Contributors

|  |  |  |
| --- | --- | --- |
| **Role** | **Unit** | **Name** |
| Software Development Lead |  | Ethan Murphy |
| Technical Writer/Tester |  | Brenda Armstrong |
|  |  |  |
|  |  |  |
| *Other document contributors* |  |  |
|  |  |  |

## 

## Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
| 09/08/2024 | 1.0.0 | Brenda Armstrong |  | Initial Version |
| 10/08/2024 | 1.1.0 | Brenda Armstrong | All | Updated with improved functionality and code structure |
|  |  |  |  |  |
|  |  |  |  |  |

# User Documentation

## The Computer Parts E-Commerce Platform is a Java-based application designed to manage various computer parts available for purchase. The platform supports different user roles such as Admin, Buyer, and Seller, each with specific capabilities. Users can manage parts (items) by adding new parts, updating existing parts, or removing parts from the inventory. Buyers can browse through available parts, add them to their cart, and make purchases.

## 2.1 Classes and Functionality:

## User: The base class for all user roles in the system. It contains common attributes such as username, password, and email that are shared among all user types (Admin, Buyer, Seller).

## Attributes: username, password, email, role, cart

## Methods:

* + getRole(): Returns the role of the user.
  + getUsername(), getPassword(), getEmail(): Getter methods for user attributes.
  + setUsername(String), setPassword(String), setEmail(String): Setter methods for user attributes.
  + addToCart(Items item): Adds an item to the user's shopping cart.
  + viewCart(): Displays the user's shopping cart and total price.

**Admin**: Represents an administrator with capabilities to manage the platform's overall settings and users.

* Attributes: Inherits from User
* **Methods****:**
  + getRole(): Returns "admin" as the user's role.

**Buyer:** Represents a user who can browse, select, and purchase computer parts from the platform. Buyers can add items to their cart, view their cart, and make purchases.

* **Attributes:** Inherits from User
* **Methods:**
  + getRole(): Returns "buyer" as the user's role.

**Seller:** Represents a user who can list computer parts for sale on the platform. Sellers can manage the items they have listed, including adding new parts, updating existing parts, or removing them from sale.

* **Attributes:** Inherits from User
* **Methods:**
  + getRole(): Returns "seller" as the user's role.

**Items:** Represents a computer part available in the inventory. This class includes attributes for the item’s category, description, name, price, and SKU (Stock Keeping Unit), and provides methods to manage these details.

* **Attributes:** category, description, name, price, sku
* **Methods:**
  + getItemCat(): Returns the item’s category.
  + getItemDesc(): Returns the item’s description.
  + getItemName(): Returns the item’s name.
  + getItemPrice(): Returns the item’s price.
  + getItemSku(): Returns the item’s SKU.
  + setItemCat(String category): Sets the item’s category.
  + setItemDesc(String description): Sets the item’s description.
  + setItemName(String name): Sets the item’s name.
  + setItemPrice(double price): Sets the item’s price.
  + setItemSku(String sku): Sets the item’s SKU.

## UserDAO: Data Access Object for managing Users in the database.

* Methods:
  + getAllUsers(): Retrieves all users from the database.
  + addUser(User user): Adds a new user to the database.
  + deleteUser(String username): Removes a user from the database.

## UserService: Main class that handles user interactions and business logic.

* Methods:
  + main(String[] args): Entry point of the application.
  + login(): Handles user login.
  + signUp(): Handles user registration.
  + handleUserSession(User user): Manages user sessions based on role.
  + handleAdminSession(Admin admin): Manages admin-specific operations.
  + handleBuyerSession(Buyer buyer): Manages buyer-specific operations.
  + handleSellerSession(Seller seller): Manages seller-specific operations.

UML Class Diagram

A screenshot of a computer

Description automatically generated

## How to Start/Access the Application

1. **Install JDK:** Ensure that JDK 11 or higher is installed on your system.
2. **Clone Repository:** Clone the project repository using Git.
3. **Install Dependencies:** Download and include the PostgreSQL JDBC driver and jBCrypt in your project.
   * **pgJDBC:** [Download here](https://jdbc.postgresql.org/download/postgresql-42.7.3.jar)
   * **jBCrypt:** Available in the dependencies directory of the project.
4. **Compile and Run:** Use VS Code or your preferred IDE to compile the project:
   * Open the terminal in VS Code (Terminal > New Terminal).
   * Compile and Run: To compile your Java code, you can use the VS Code integrated terminal. Open the terminal within VS Code by selecting "Terminal" > "New Terminal" from the menu. Then, navigate to the directory containing your Java files and use the javac command to compile them. You can also use the built-in "Run" and "Debug" features of VS Code to compile and execute your Java code menu.
   * Run the UserService class to start the application from UserService.java

# Development Documentation

## JavaDoc:

JavaDoc comments are provided for all classes and methods to document their functionality and usage.

## Source Code Structure:

All Java source files are organized in a single directory. Each class is defined in its own .java file.

* User.java
* Admin.java
* Buyer.java
* Seller.java
* Items.java
* UserDAO.java
* UserService.java

## Build Process:

Compile the project using a Java compiler (e.g., javac \*.java). You can also use the built-in "Run" and "Debug" features of VS Code to compile and execute the main method of the Demo class.

## Compiler Time Dependencies:

Add these dependencies to your project to compile successfully. The project depends on the following libraries:

* **pgJDBC:** PostgreSQL JDBC driver.
* **jBCrypt:** A library for hashing passwords securely.

Ensure these JAR files are in your classpath when compiling and running the application.

## 

## Development Standards:

* + Classes start with an uppercase letter and use CamelCase (e.g., UserService).
  + Methods and variables start with a lowercase letter and use camelCase (e.g., addToCart).
  + Constants are in all uppercase with underscores (e.g., JDBC\_URL).
  + Use meaningful and descriptive names for classes, methods, and variables.
  + Include comments for complex logic or non-obvious code sections.
  + Use proper indentation and formatting for improved readability., and structuring code.

## Setting up a Database for Development:

This application uses PostgreSQL for persistent data storage. The PostgreSQL database is used to store information about users, roles, and computer parts.

Follow these steps to set up the database:

1. Install PostgreSQL on your development machine.
2. Create a new database named "Java".
3. Use the provided SQL scripts to create the necessary tables:
   * Run usersCreate.sql to create the users table.
   * Run prodsCreate.sql to create the products table.
   * Run prodsInsert.sql to populate the products table with sample data.

**Database Data Schema**

* Users Table
  + username (text, primary key)
  + password (text)
  + email (text)
  + role (text)
* Products Table
  + itemname (text)
  + itemsku (text, primary key)
  + itemprice (double precision)
  + itemdesc (text)
  + itemcat (text)
  + userListed (text)

## Getting the Source Code from the Repository:

git clone https://github.com/emurphy04/SD10-S3-SPRINT-2-JAVA

Navigate to the JAVA directory: cd SD10-S3-SPRINT-2/JAVA

# Deployment Documentation

#### **4.1 Installation Manual**

To deploy the application:

1. Java Runtime Environment (JRE): Ensure that JRE 11 or higher is installed on the target system.
2. PostgreSQL: Install and configure PostgreSQL on the target system.
3. Database Configuration:
   * Create a new database named "Java".
   * Import the SQL schema using the provided SQL scripts (usersCreate.sql and prodsCreate.sql).
4. Application Files:
   * Copy all compiled .class files to the deployment directory.
   * Include the required JAR files (PostgreSQL JDBC driver and jBCrypt) in the deployment directory.
5. Configuration:
   * Update the database connection details in ItemDAO.java and UserDAO.java if necessary.

***4.2 To Demonstrate the Project:***

To run the application:

1. Open a terminal or command prompt.
2. Navigate to the directory containing the compiled .class files and JAR dependencies.
3. Run the command: java -cp .:postgresql-42.7.3.jar:jbcrypt-0.4.jar UserService (Use ; instead of : as a separator on Windows)
4. Follow the on-screen prompts to interact with the system.
5. Use the following steps to demonstrate key features:
   * Register a new user (buyer or seller)
   * Log in as an admin to manage users and view all products
   * Log in as a seller to add new product listings
   * Log in as a buyer to search for products, add items to cart, and checkout