# Java Ecommerce Platform

Developed by Elliott Butt

Submitted 2024-08-09

## User Documentation

### Introduction

This program is a console-based ecommerce platform written in Java and using PostgreSQL as a database.

### Program Features

Many features of this program are based on the role of the logged-in user.

#### Main Menu

1. Login to the platform using existing login credentials.
2. Register a new credential to login to the platform.

#### Admin Features

1. View a list of all users (admins, buyers, and sellers).
2. View a list of all sellers and their associated information.
3. View a list of products available in the store, including seller information for each product.
4. Delete users based on various search parameters (by username, ID, or role).

#### Buyer Features

1. Browse products available in the store, either by category or all products at once.
2. Search for a specific product by product name, product ID, or seller name.

#### Seller Features

1. Add a product to the store belonging to the logged-in seller.
2. Update a product belonging to the logged-in seller.
3. Delete a product belonging to the logged-in seller.
4. View a list of all products belonging to the logged-in seller.

### Classes

#### Users

These classes all represent the various types of users that can login to the platform. The three user classes are:

1. Admin
2. Buyer
3. Seller

These classes all inherit from the base User class. The User class as five fields: ID, username, password (stored as a hashed password in the database using BCrypt), email, and role.

Only the Seller class has additional fields, based on their 'seller information'. Seller information is stored in a separate table in the database. These additional fields are: sellerId, storeName, storeDescription, contactNumber, storeEmail, url, and address.

#### Products

All products on the platform are represented by the Product class. It holds all relevant information about a product: productId, name, description, category, categoryId, quantity, price, sellerId, and sellerName.

#### Menus

These classes hold all the prompts for the console-based user-interface. They allow the logged-in user to interact with the program, and display the correct options based on the user's role.

The menus call functions from the UserServices and ProductServices classes, allowing the user to interact with the program and database directly.

#### UserServices/UserDAO

This is where all the logic regarding user management is housed. The UserServices class is the program-side logic, and the UserDOA is concerned only with database integration and logic.

In these classes, you can find the bulk of the program's user-based functionality:

1. Add/delete users from the database
2. Login/Authorize user logins
3. Register new users and add them to the database
4. Search for specific users based on search criteria
5. Retrieve all users from database.

#### ProductServices/ProductDAO

This is where all the logic regarding product management is housed. Just like with the user-related classes, the ProductServices class is the program-side logic, and the ProductDAO is concerned only with database integration and logic.

In these classes, you can find the bulk of the program's user-based functionality:

1. Add/update/delete products from the database
2. Search for specific products based on search criteria
3. Retrieve all products from database

### Class Diagram

A screenshot of a computer

Description automatically generated

## Development Documentation

### Javadocs

The javadocs can be found in the \_Documentation\_ directory.

### Directory Structure

This is a representation of the directory structure.

Database/

SQL/

00-ALL.sql

01-CREATE-categories.sql

02-CREATE-users.sql

03-CREATE-sellers.sql

04-CREATE-products-sql

05-INSERT-categories.sql

06-INSERT-users.sql

07-INSERT-sellers.sql

08-INSERT-products.sql

DatabaseConnection.java

Documentation/

images/

Class-Diagram.png

javadocs/

Documentation.md

Documentation.docx

lib/

jBCrypt-0.4.1.jar

postgresql-42.6.0.jar

Menus/

AdminMenu.java

BuyerMenu.java

MainMenu.java

MenuService.java

SellerMenu.java

Products/

Product.java

ProductDAO.java

ProductService.java

Users/

Admin.java

Buyer.java

Seller.java

User.java

UserDAO.java

UserService.java

### Dependencies

There are 2 dependencies for this program:

1. jBCrypt
2. PostgreSQL

You can download jBCrypt from https://www.mindrot.org/projects/jBCrypt/.

You can get PostgreSQL from https://jdbc.postgresql.org/download/.

### Database Setup

This project uses PostgreSQL for its database. You will need then postgresql.jar file (see the dependencies section).

You'll need to initialize a new database using a tool like pgAdmin or the psql CLI. You can then run the file '00-ALL.sql' found in the \_Database\_ directory to create all the necessary tables in insert mock data.

### Source Code from Repo

Repository link: https://github.com/sweetboymusik/final-sprint-java-elliott-butt

You can download the source code directly through this github page or clone the repository to your local machine.

## Deployment Documentation

Find below instructions on how to install and run the program:

1. Download all files off the GitHub Repository

Go to https://github.com/sweetboymusik/final-sprint-java-elliott-butt, create a directory anywhere on the device, and download all the files from the repository into that directory.

Alternatively, see the 'Source Code from Repo' section of this document.

2. Open Menu.java in VSCode

Navigate into the Menus directory and right click on MainMenu.java and press 'open with', then find and click on Visual Studio Code.

3. Start the program

On the top right of your screen you should see a play button, click that and a terminal will open on the bottom of the screen with the program running in it.