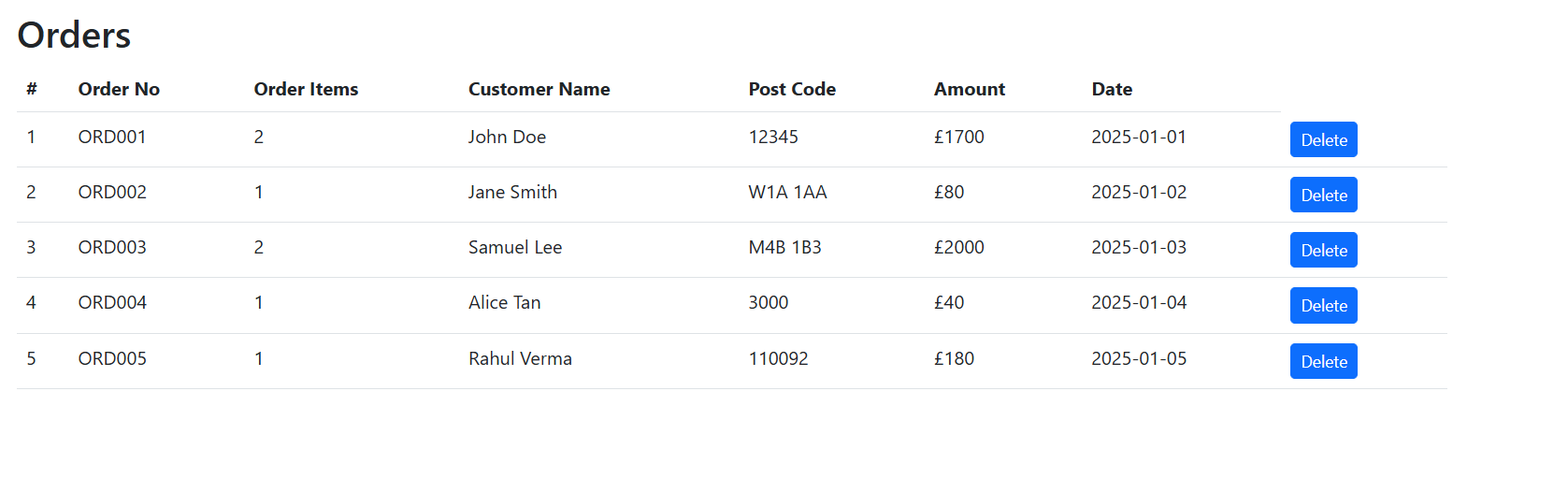
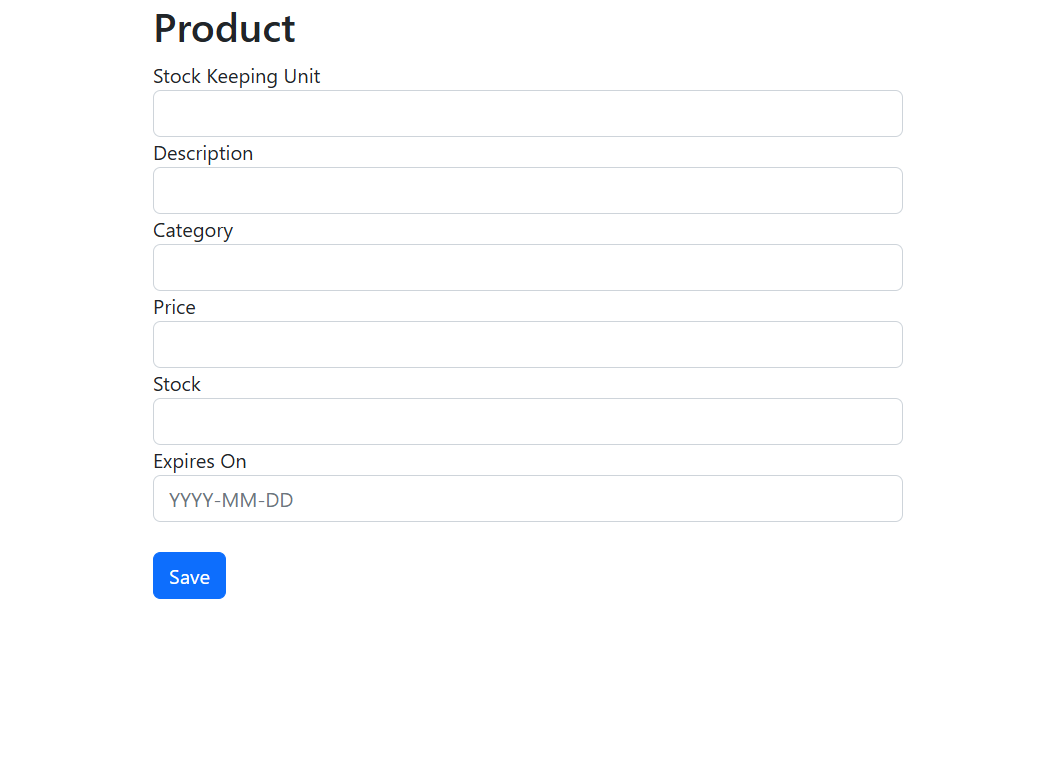
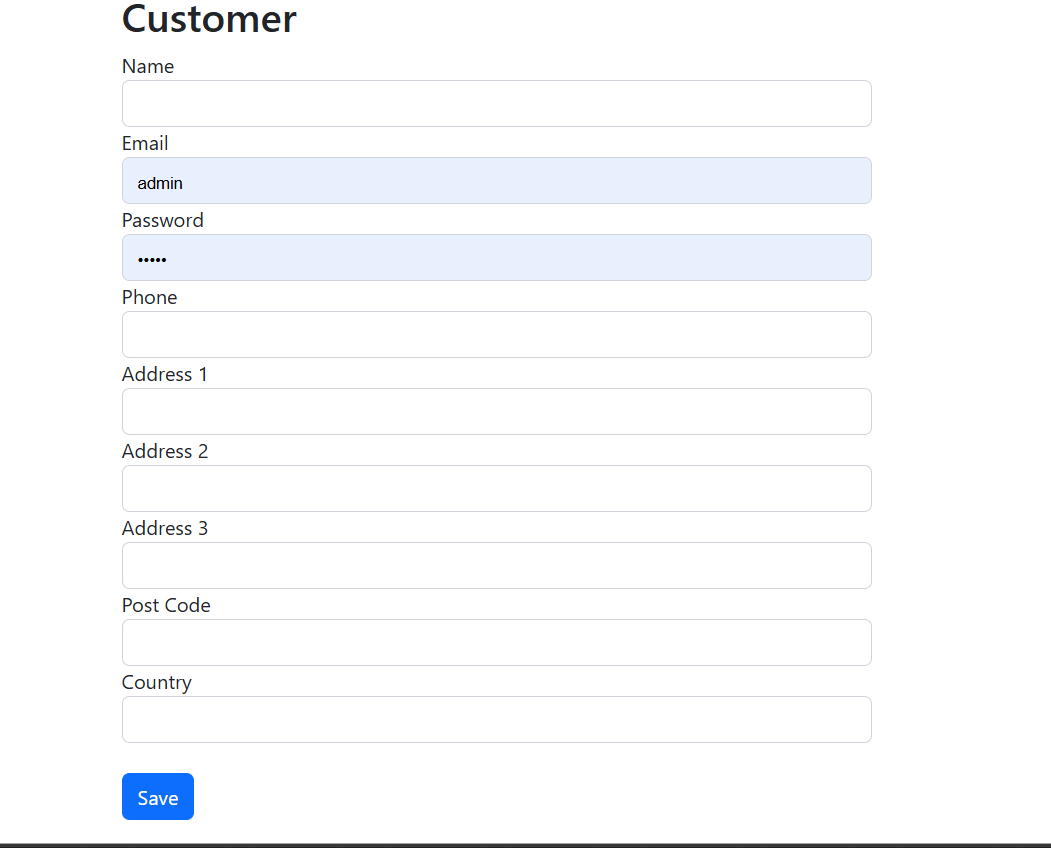
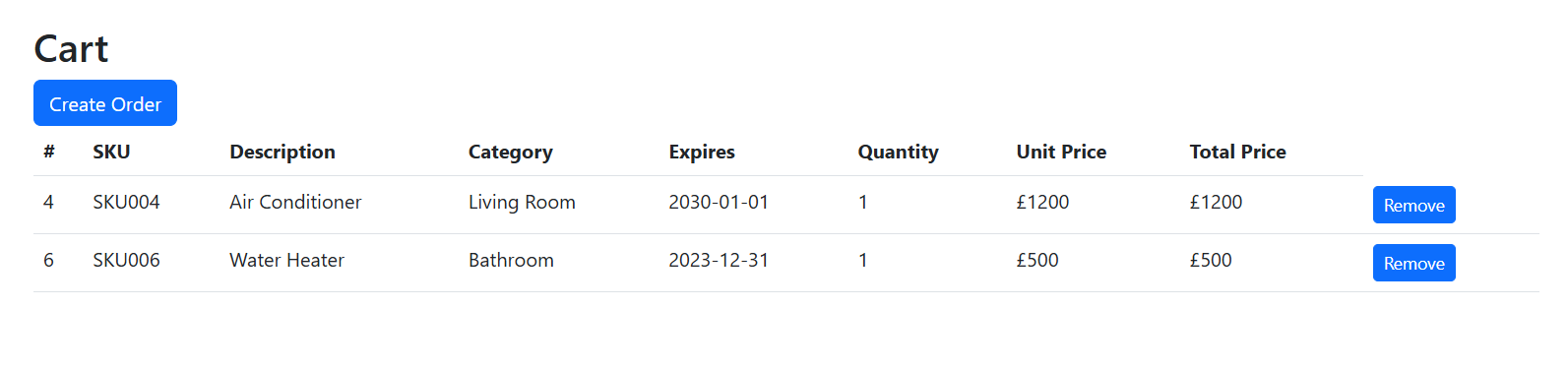
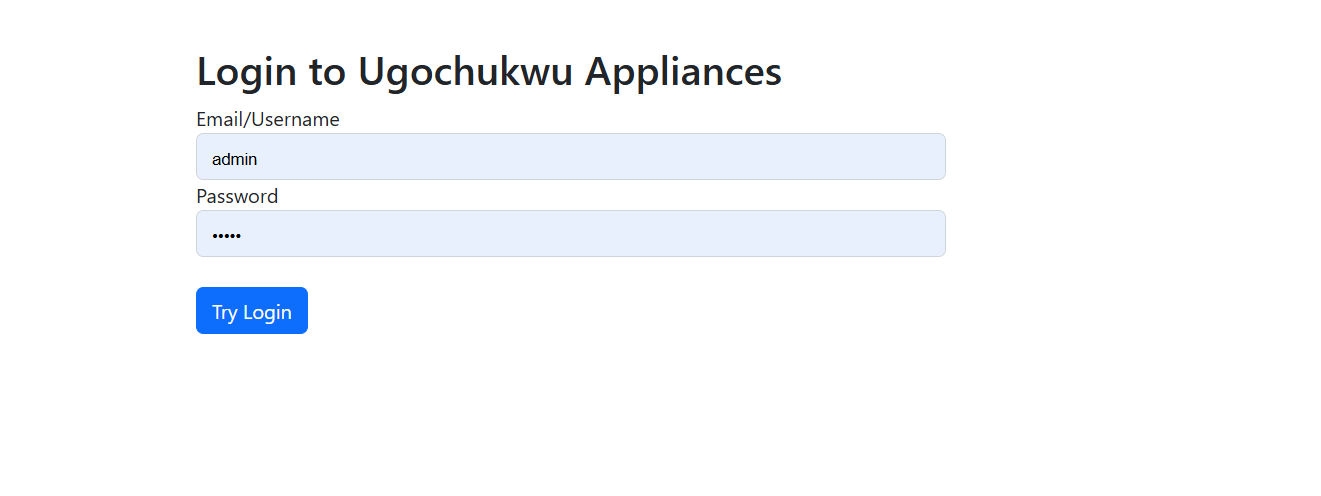
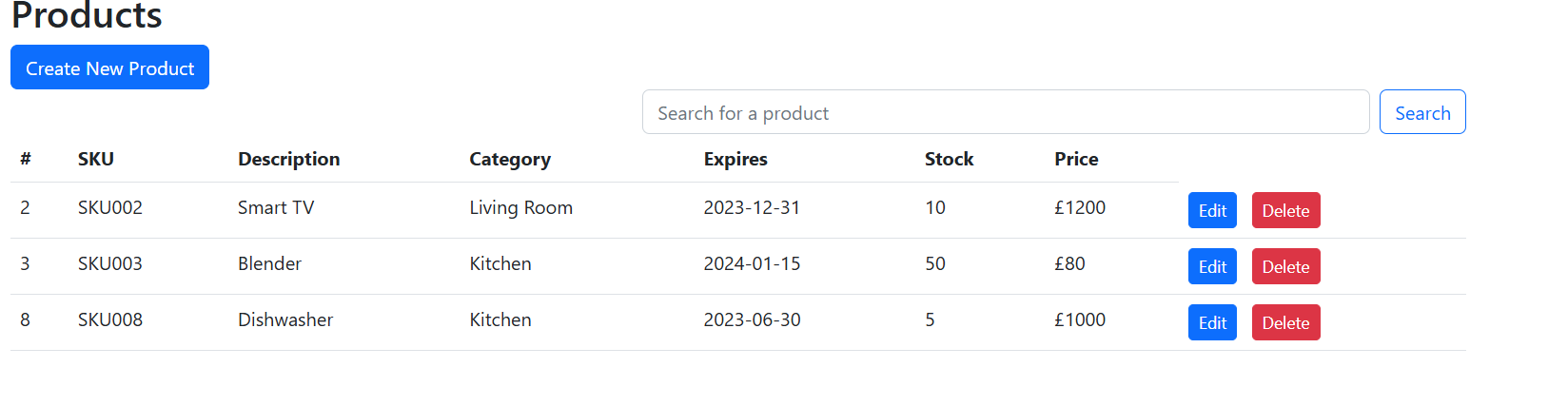
**Name: Ugochukwu Precious Onah  
Student ID: 24848777  
School and Dept: Computer Science, MSc, Manchester Metropolitan University.**

**Technical Documentation**

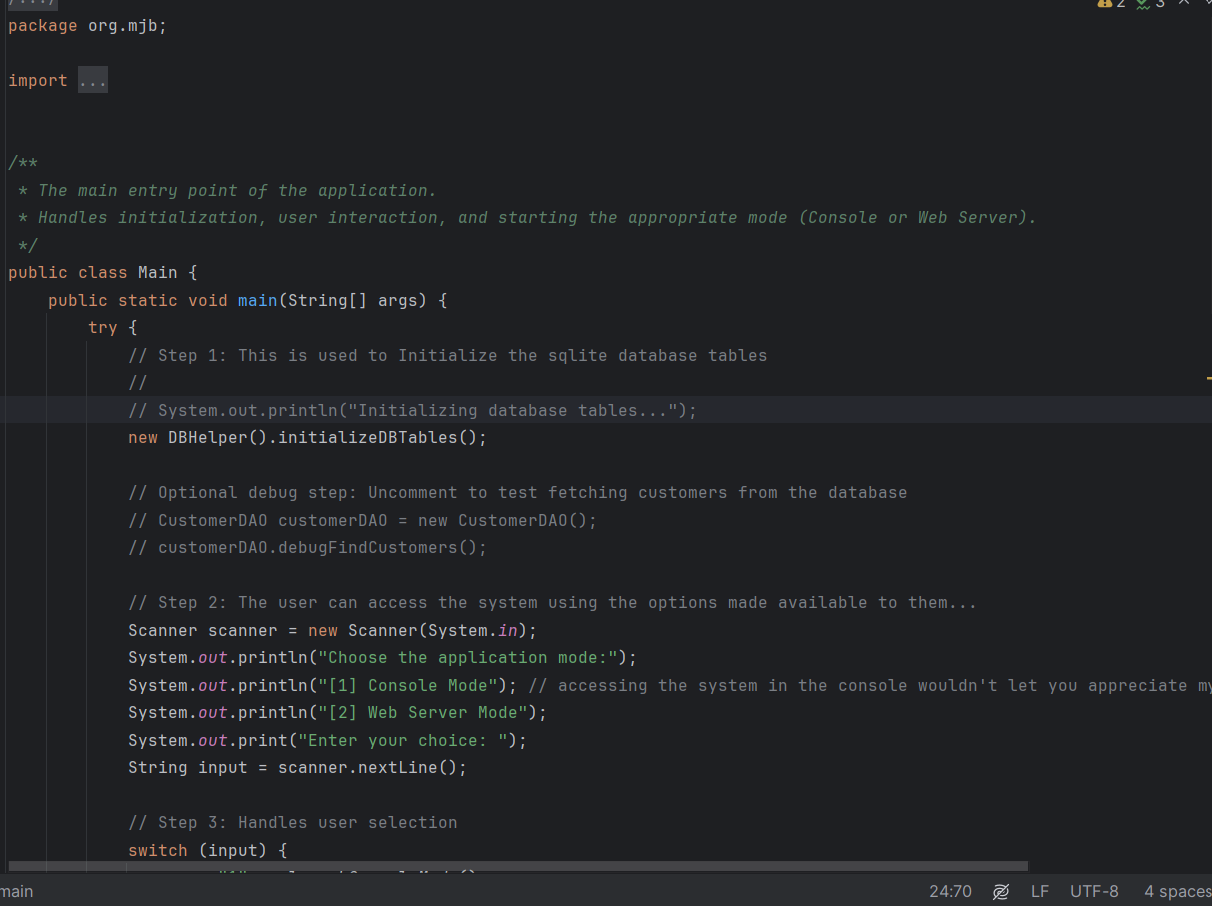
****

****

**Fig: Expired Products**

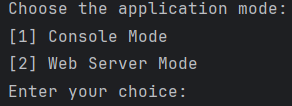
**Main Class**

The Main class serves as the entry point for the application. It handles initialization, user interaction, and mode selection, allowing the user to operate the system in either **Console Mode** or **Web Server Mode**. It encapsulates key logic for initializing database tables, interacting with users via the command line, and setting up the web server.

****

**Key Methods**

1. Initializes the database tables using DBHelper().initializeDBTables().
2. Prompts the user to choose between Console Mode or Web Server Mode.



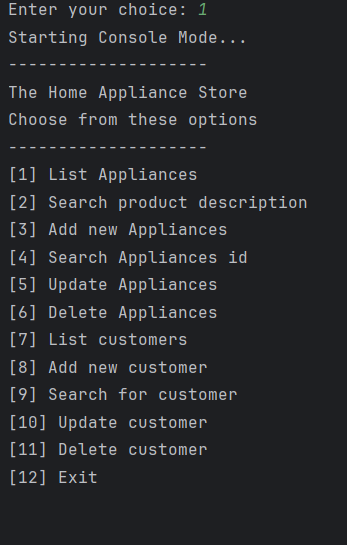
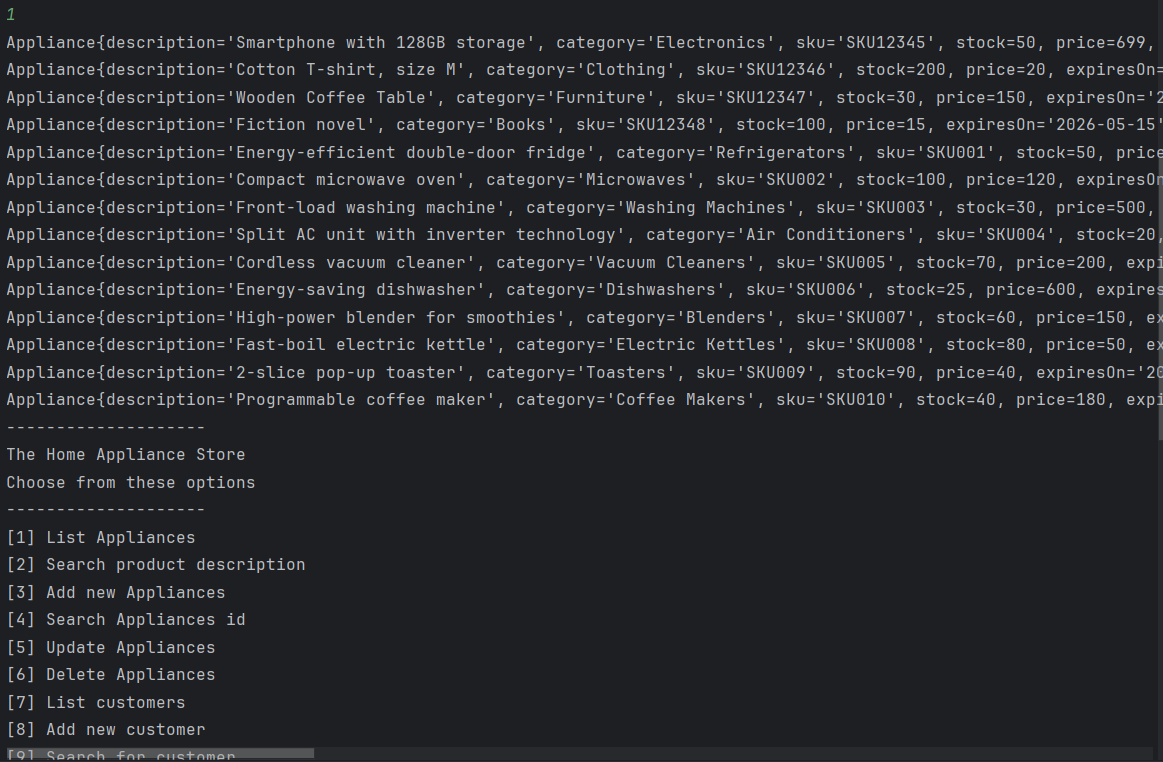
1. **launchConsoleMode()**
   * **Purpose**: Initializes and runs the system in Console Mode.
   * **Output**: Displays Console Mode options and processes user actions.
   * 
2. 

Fig: Console Appliance search

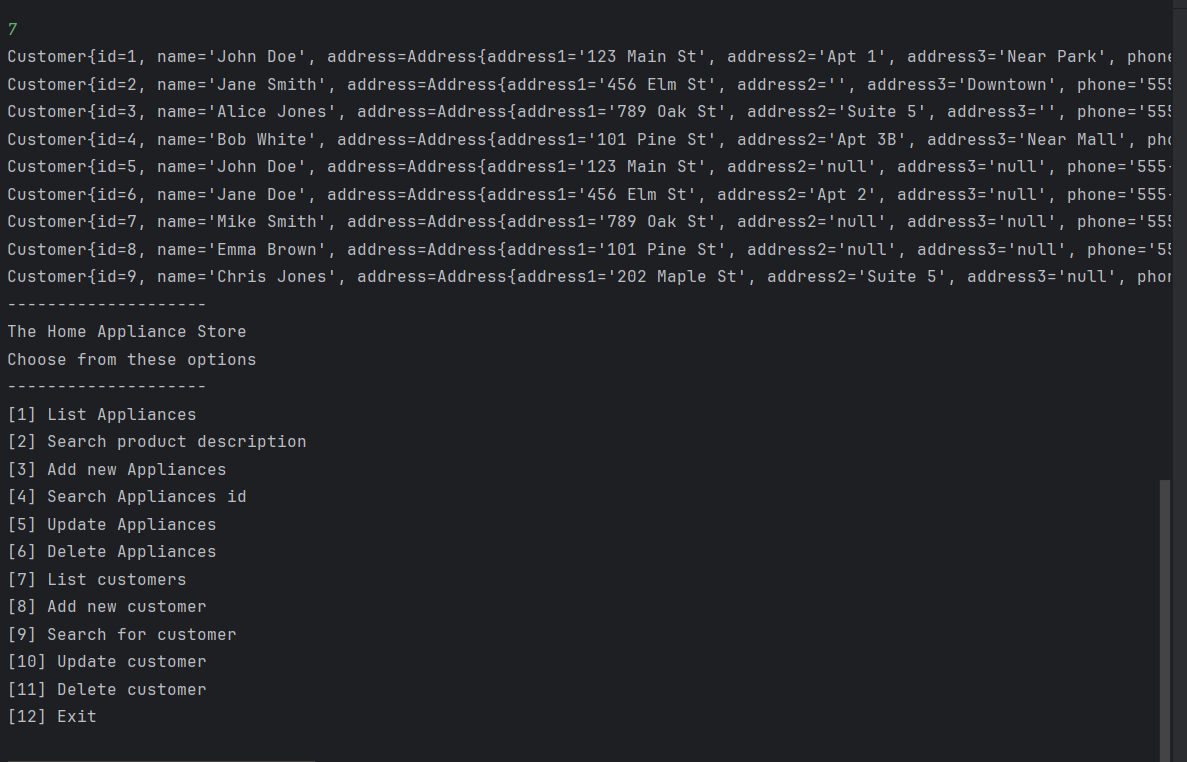
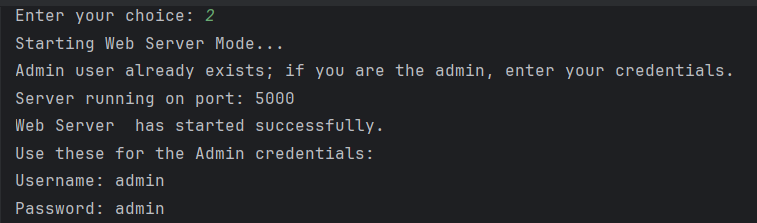


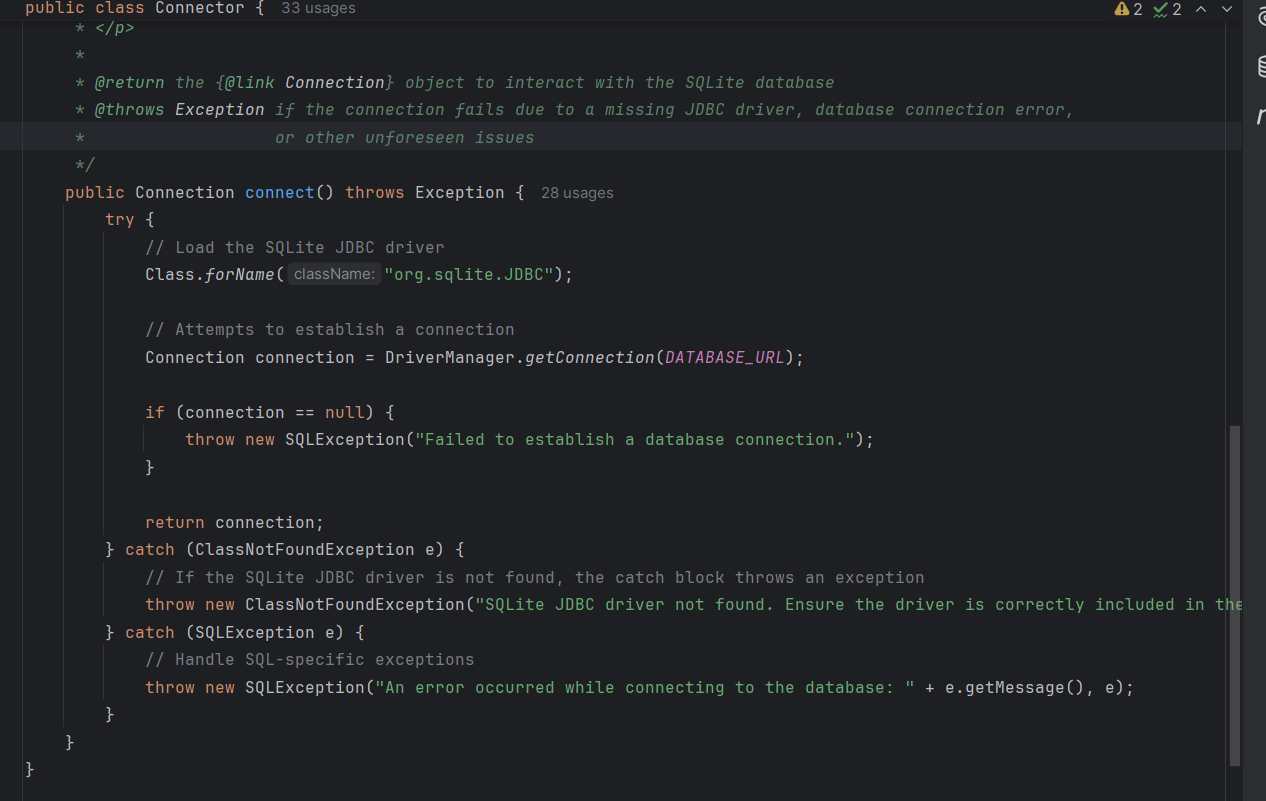
Fig: Console Customer search

1. **launchWebServerMode()**
   * **Purpose**: Initializes and runs the system in Web Server Mode, port 5000.



**The Connector Class**

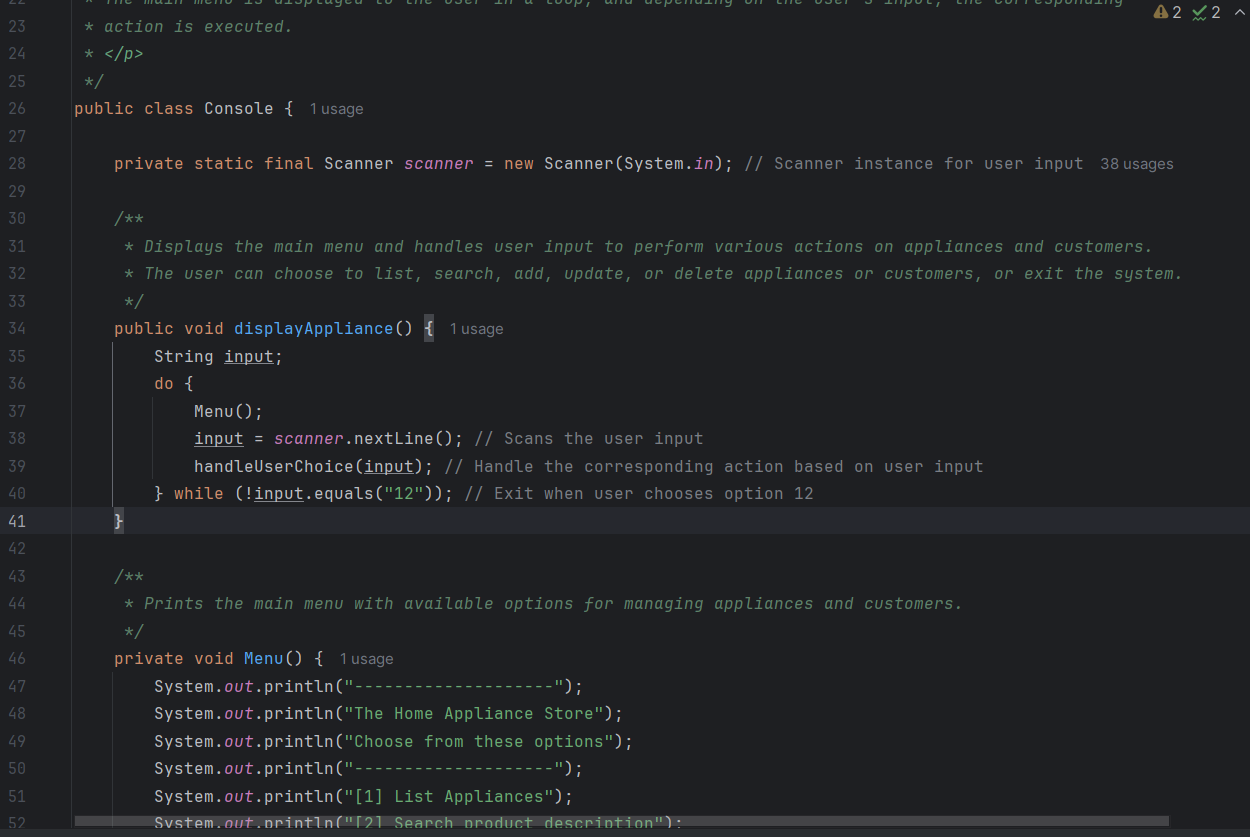
**Class :** The Connector class is responsible for establishing a connection to the SQLite database. It encapsulates the logic required to interact with the database file and ensures that the database connection is properly established or raises relevant exceptions if the process fails.

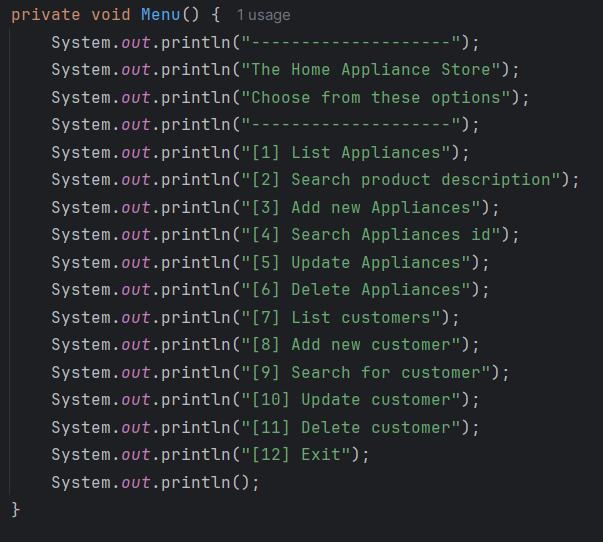


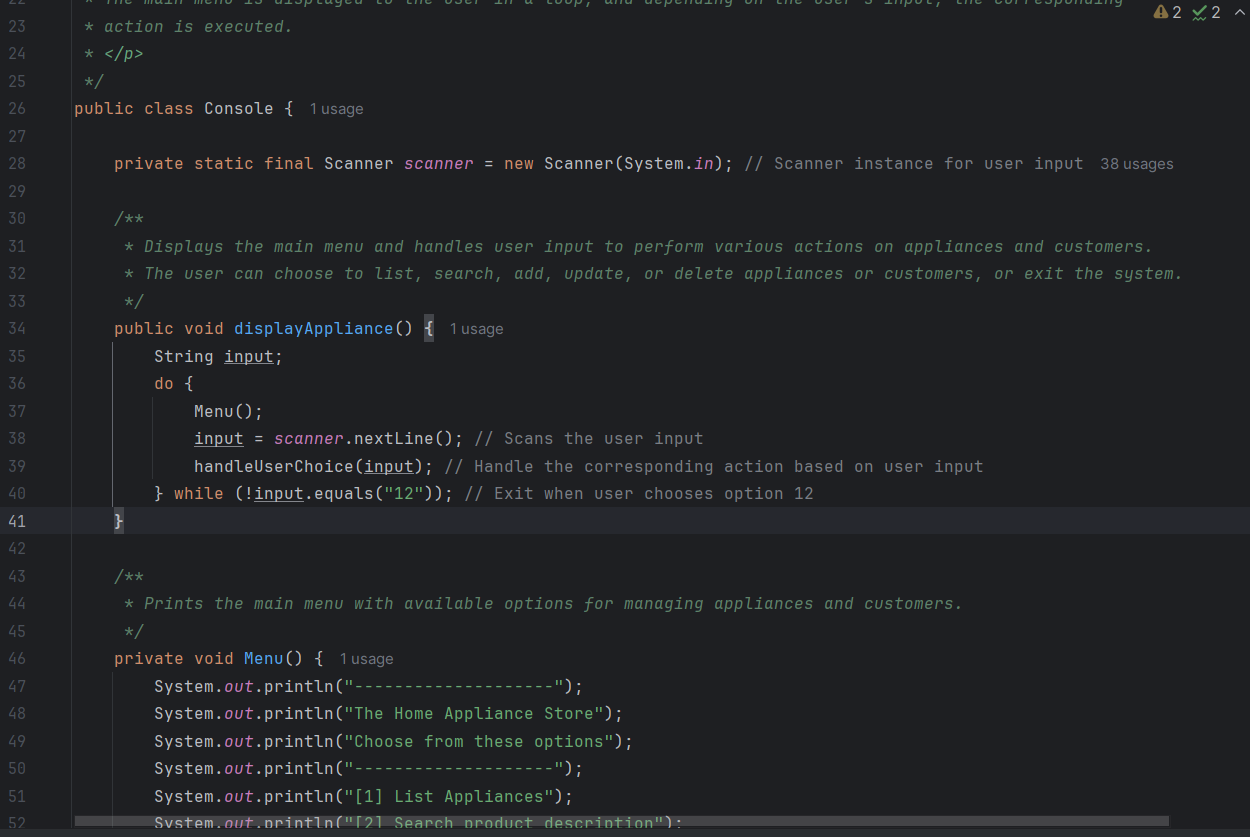
**The Console Class**

The Console class serves as the main interface for managing the appliance store’s operations. It provides an interactive console-based interface for managing appliances and customers, with functionality for listing, searching, adding, updating, and deleting records

## **Class Diagram**







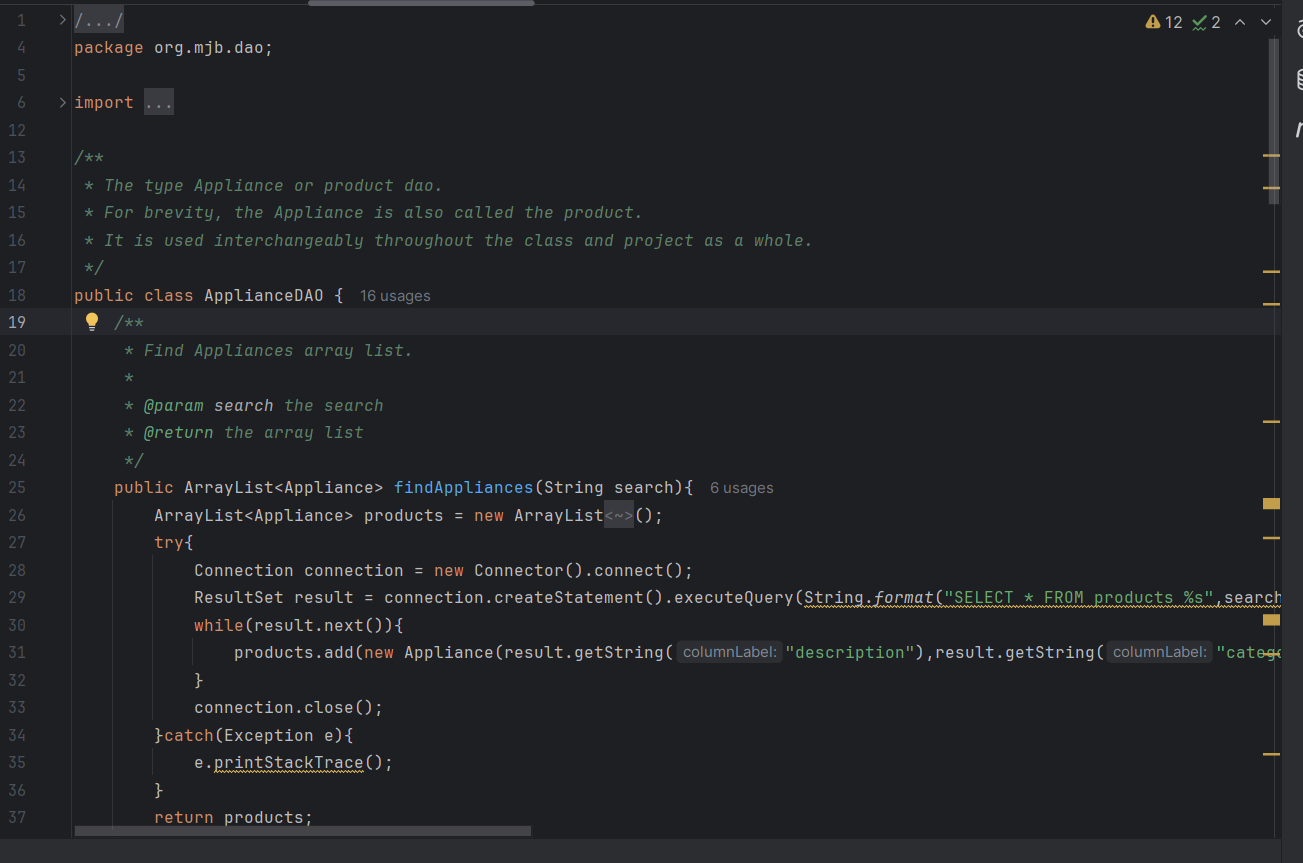
**The DBHelper Class**

The DBHelper class in the org.mjb package provides methods for managing SQLite database tables, including creating and deleting them.



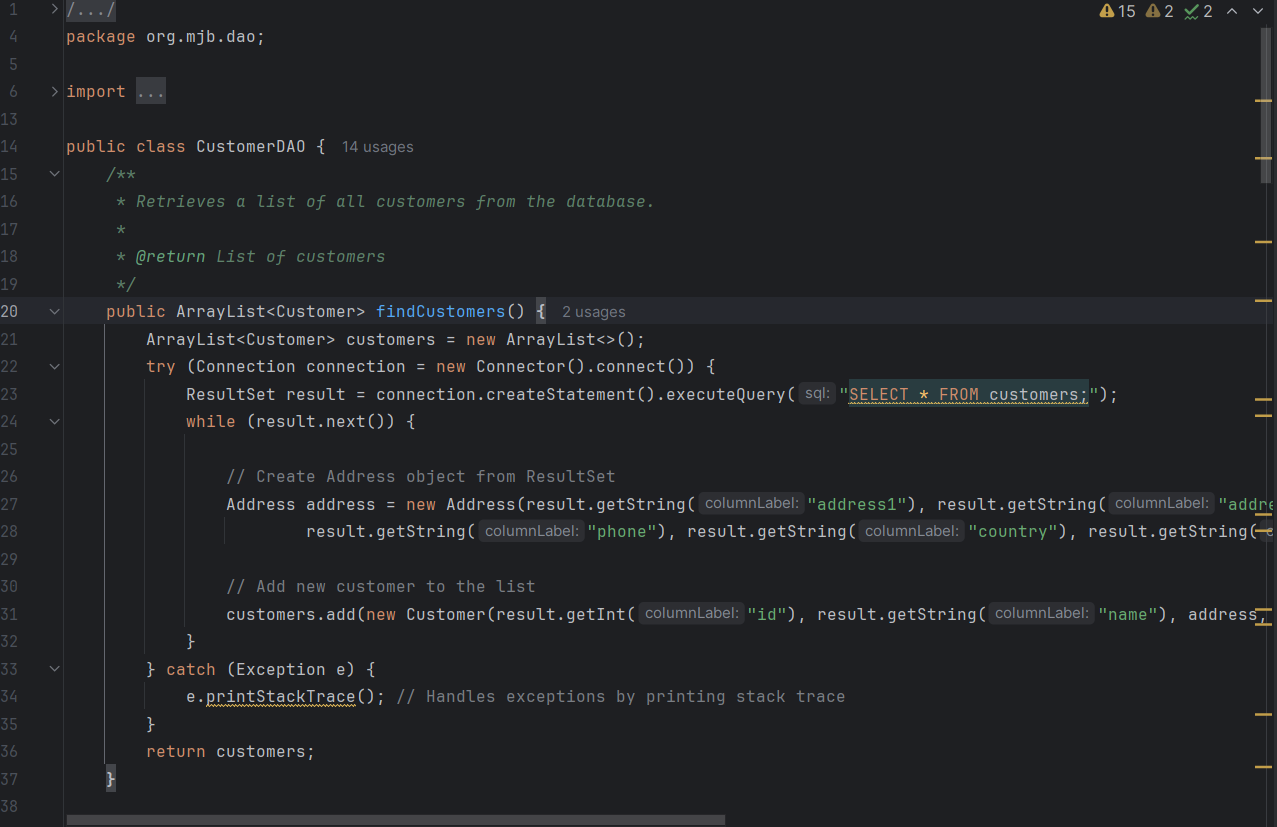
**The ApplianceDAO Class**

The ApplianceDAO class provides a comprehensive Data Access Object (DAO) implementation for managing Appliance objects in the database.



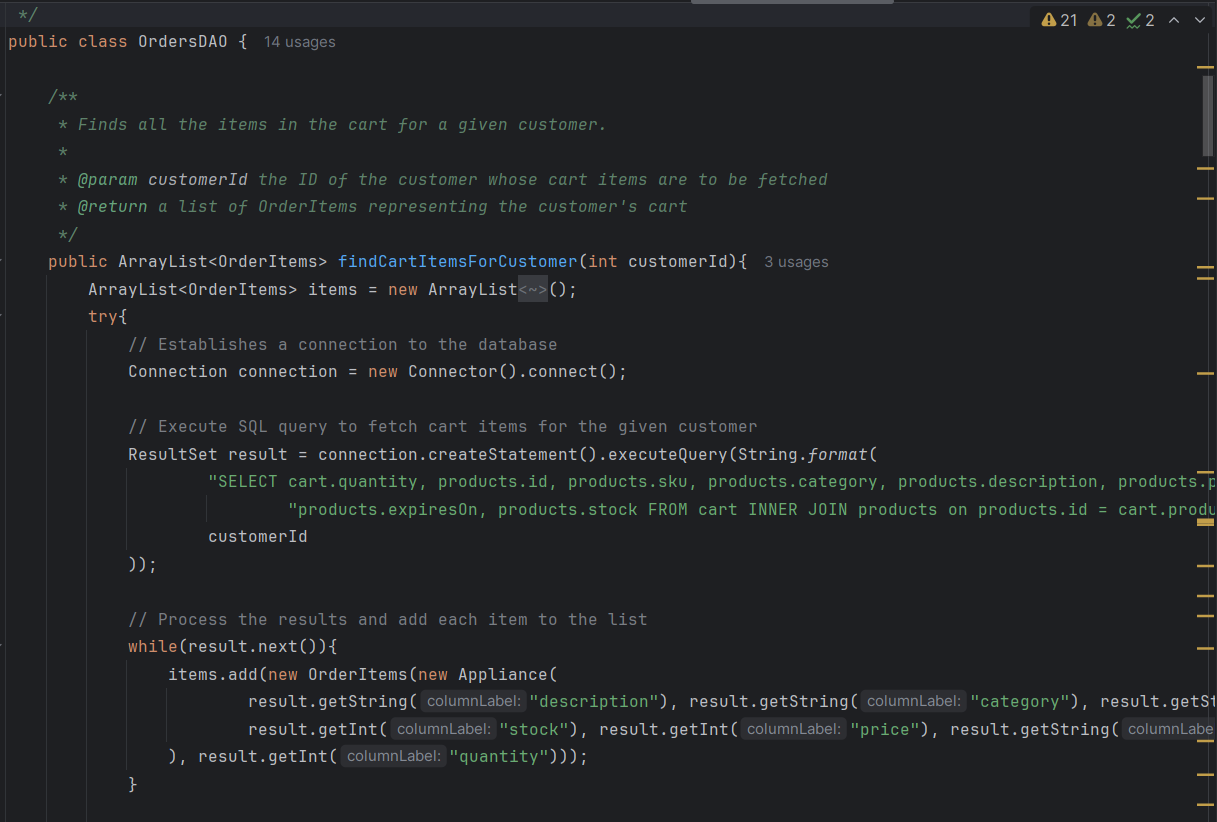
**CustomersDAO**

The CustomerDAO implementation you provided is well-structured and covers common CRUD operations along with an authentication mechanism (logInAsCustomer).



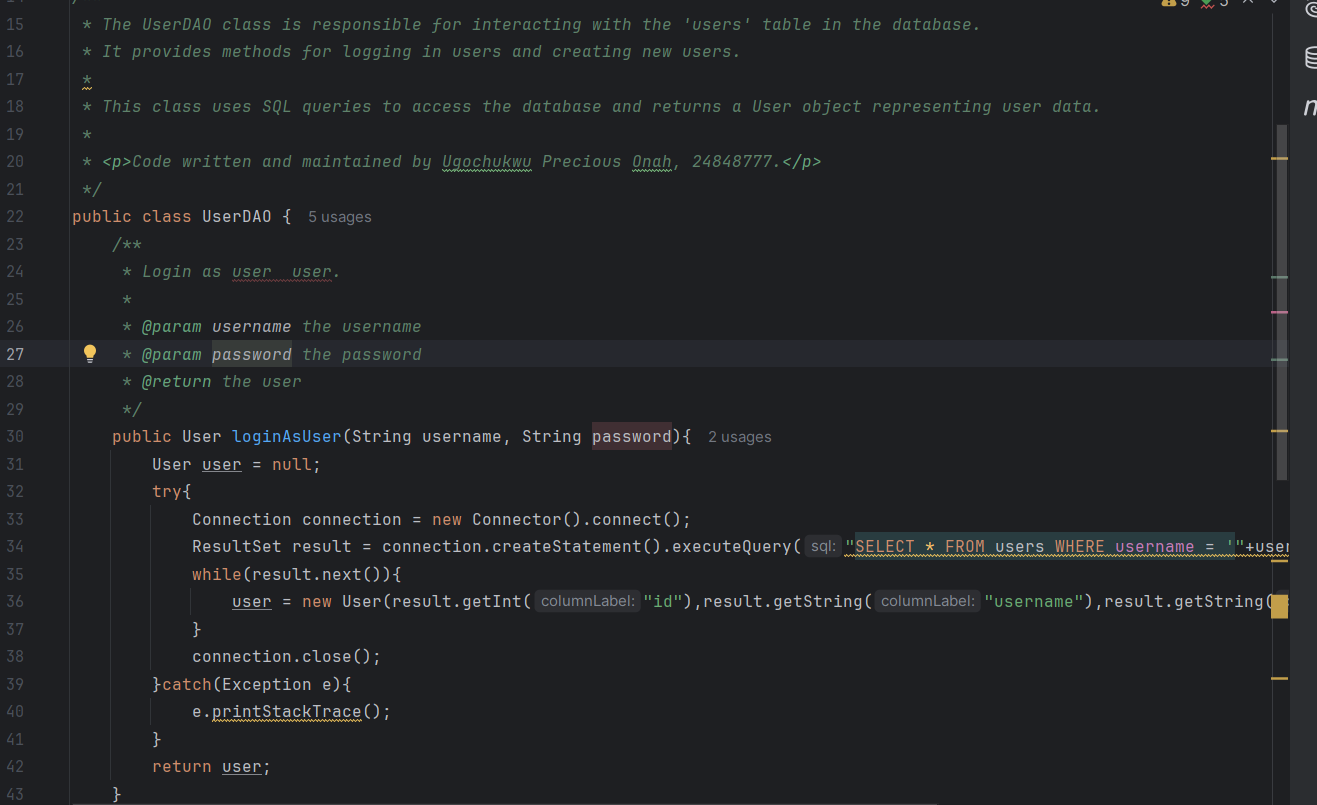
# **OrdersDAO**

## The OrdersDAO class is a Data Access Object (DAO) that provides methods to interact with the "Orders" table in a SQLite database. It encapsulates CRUD (Create, Read, Update, Delete) operations and abstracts database queries to manage orders efficiently.



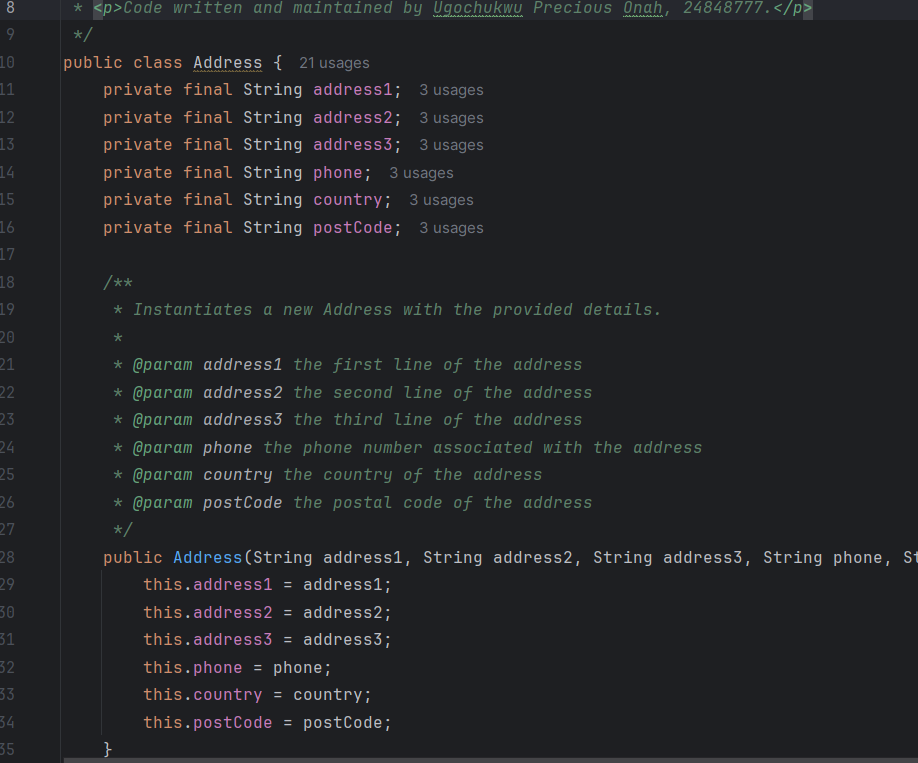
### ****UserDAO Class****

The UserDAO (Data Access Object) class provides methods for interacting with the users table in the database. It includes functionality for logging in a user and creating a new user.



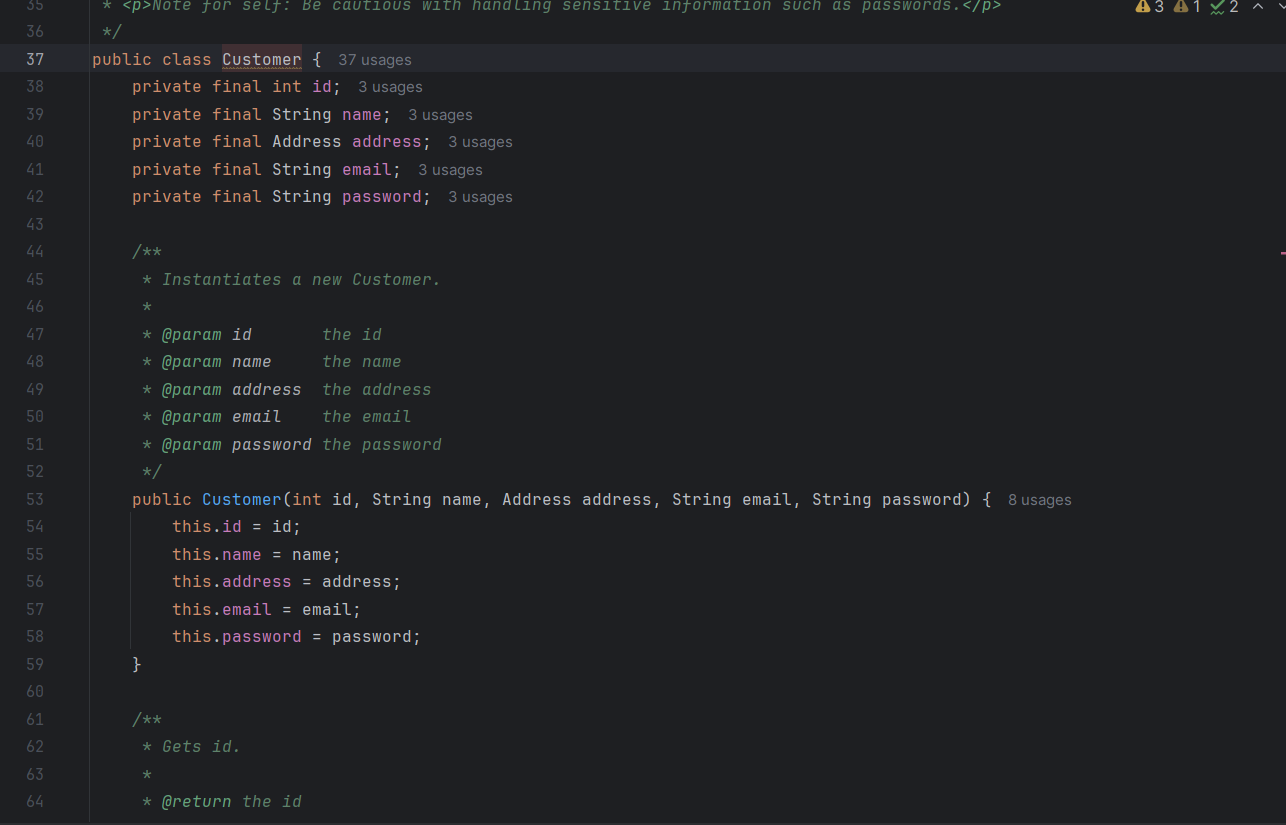
### ****Address Class****

The Address class represents a physical address with multiple components, including street address lines, phone number, country, and postal code.



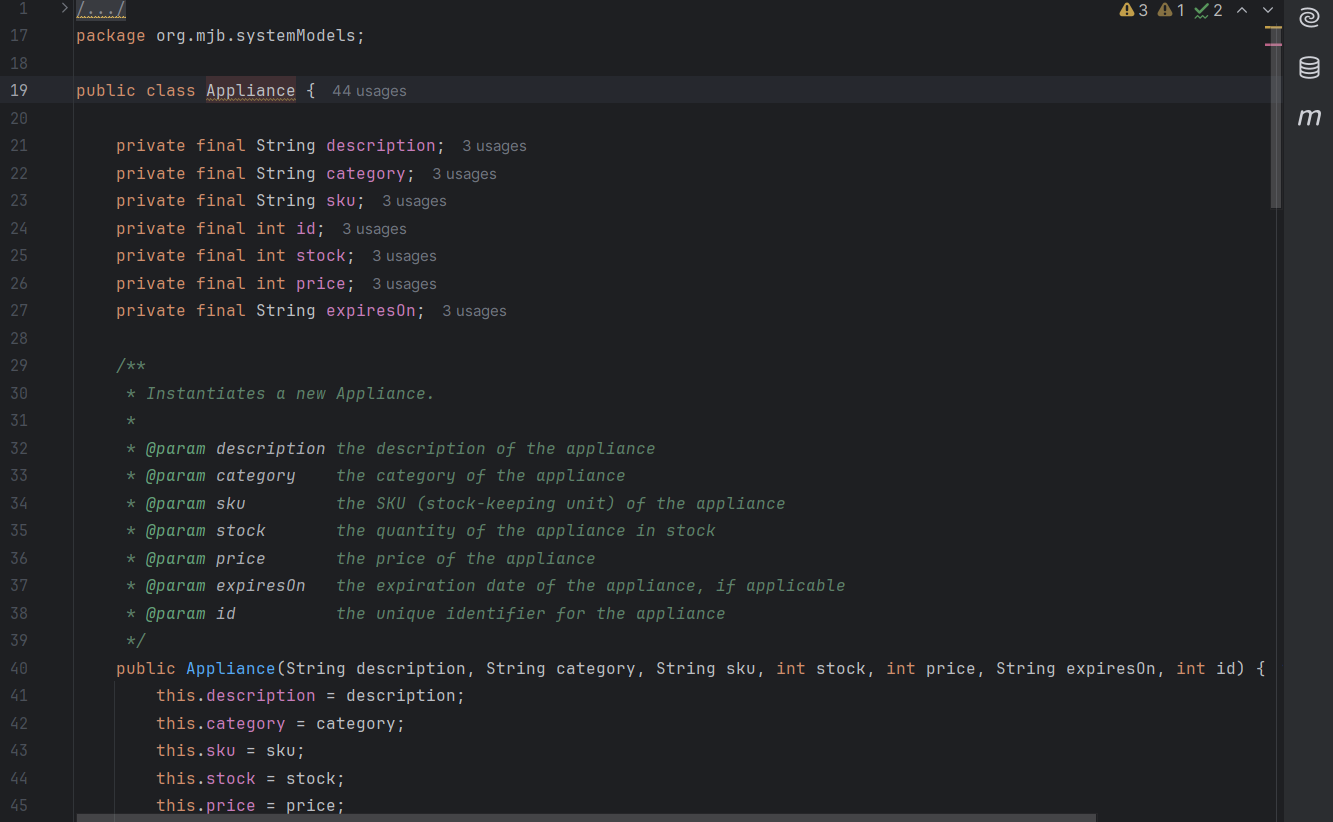
### ****Customer Class****

The Customer class represents a customer entity in the system. It encapsulates information about a customer, including their personal details (name, email, and password), as well as their associated address. This class is part of the org.mjb.systemModels package and is used to model a customer with basic attributes.



## **Appliance Class**

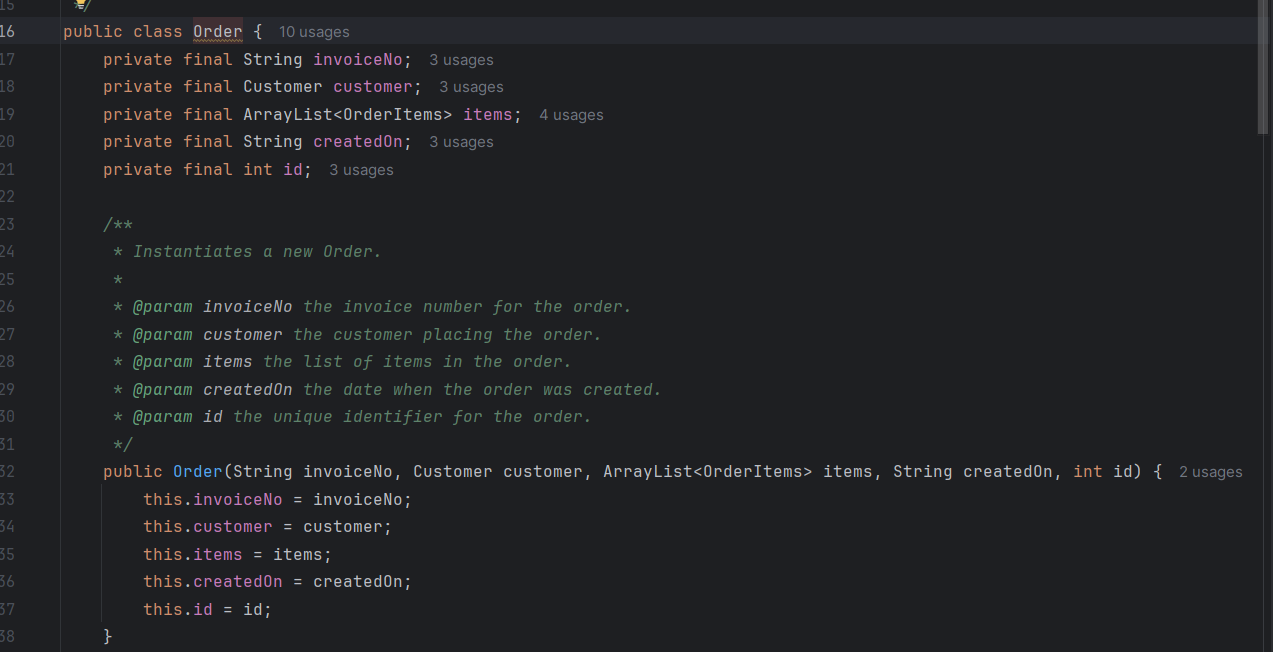
The Appliance class represents an appliance product in the system. It contains details about the appliance, such as its, category, SKU (stock-keeping unit), stock quantity, price, expiration date, and a unique identifier (id). It is used to manage the properties of an appliance product and is also referred to as "product" in the project.



## Order Class

### :

The Order class represents an order in the system, including the customer making the order, the items in the order, and the total cost.



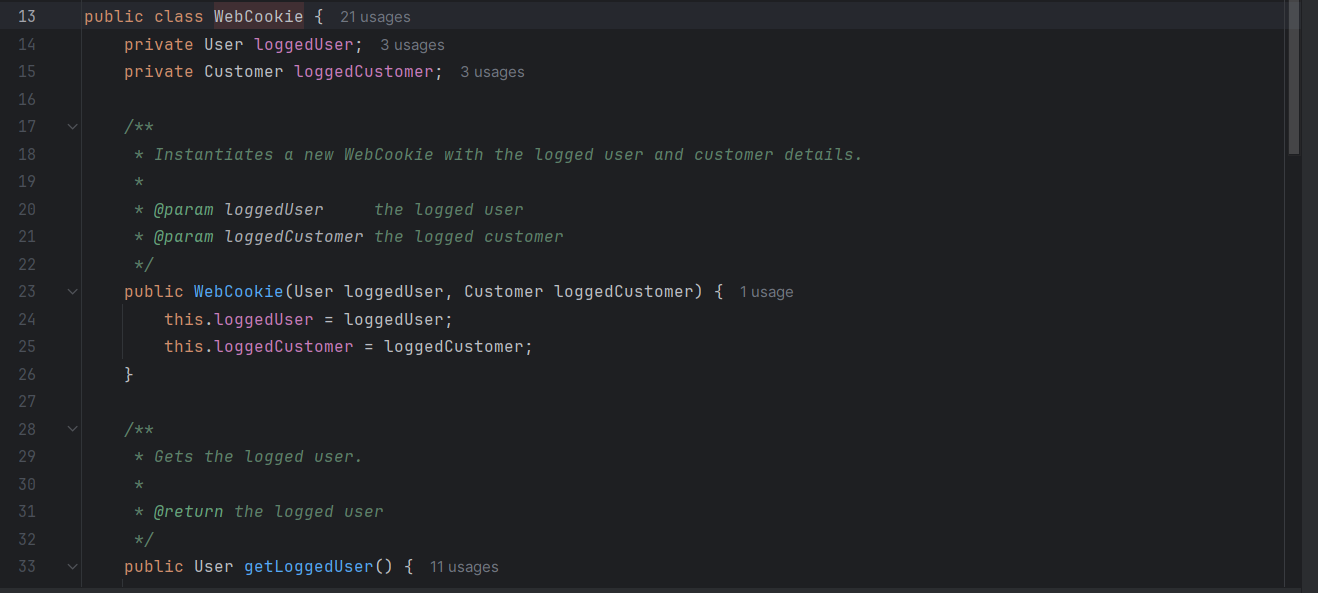
### ****WebCart Class Documentation****

The WebCart class is a part of the web server package that handles HTTP requests related to the shopping cart functionality in an online store. It manages customer cart items by allowing users to view, add, and remove products from the cart. The cart page is displayed dynamically with data retrieved from the backend SQLite database.



## WebCookie Class

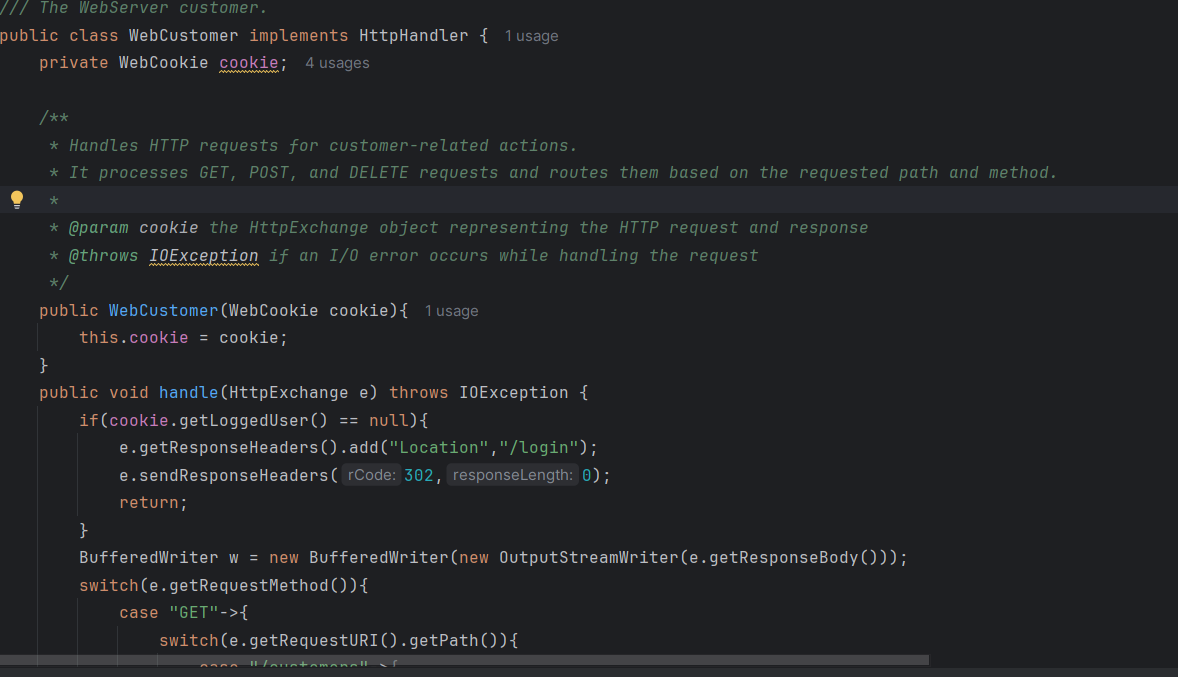
The WebCookie class is responsible for managing the session information for users and customers in the web application. It stores the details of the logged-in user and customer, allowing other parts of the application to access this information for personalized interaction and session management.



### WebCustomer ****Class****

#### ****:****

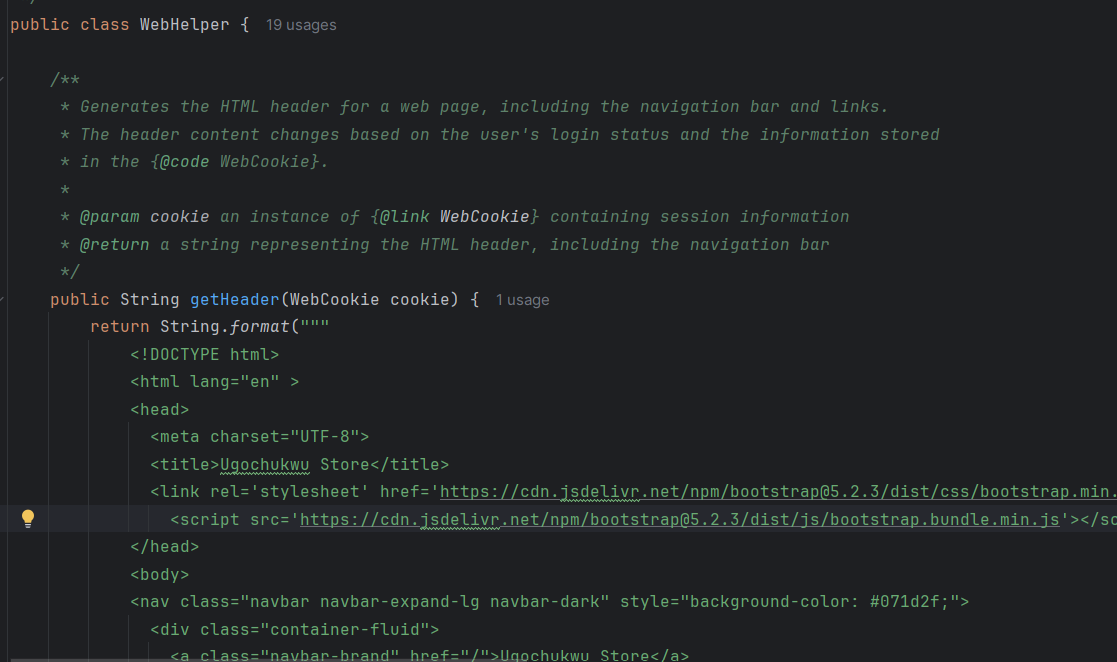
The WebCustomer class handles HTTP requests related to customer operations within a web server. This class implements the HttpHandler interface, enabling it to process HTTP methods such as GET, POST, and DELETE for managing customer records. It allows actions such as viewing a list of customers, creating new customers, editing customer details, and deleting customers.



### WebHelper ****Class****

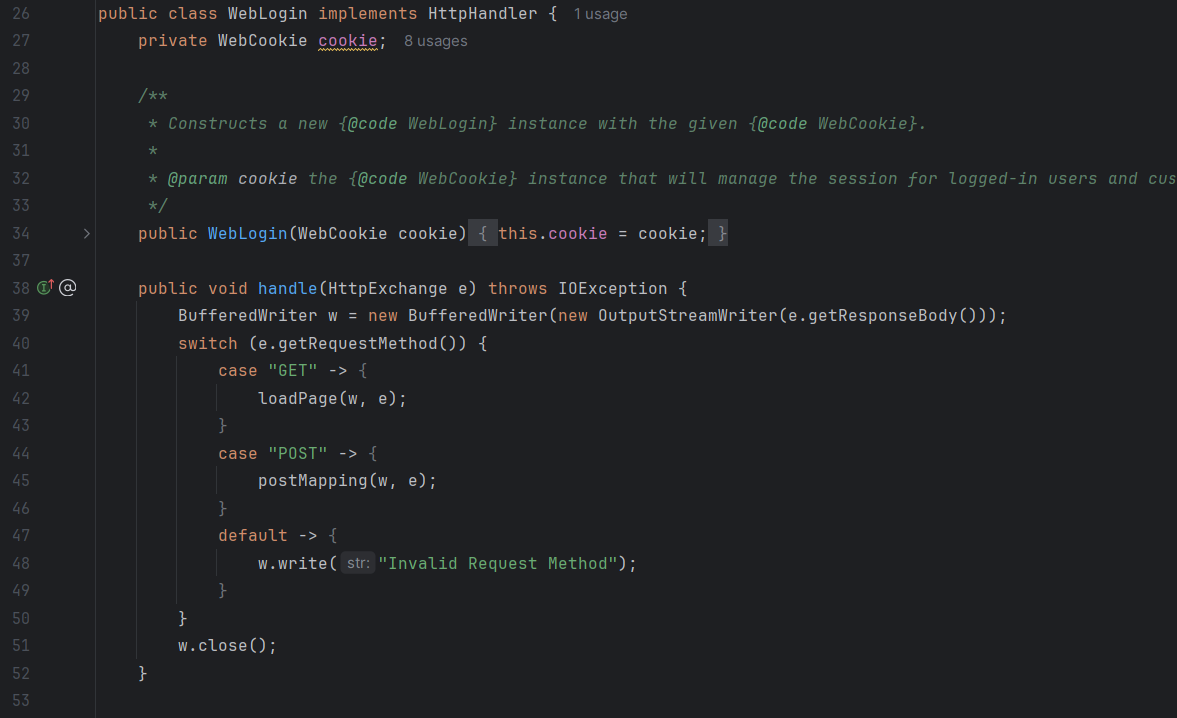
#### ****:****

The WebHelper class provides utility functions to assist with web server operations, specifically handling HTTP requests and generating dynamic HTML content. It includes methods for generating HTML headers and footers, building web pages, processing HTTP query strings, and formatting request parameters into a map for easy access. The class is used to help manage customer interactions, including login, product display, and cart management on the server.



## WebLogin Class

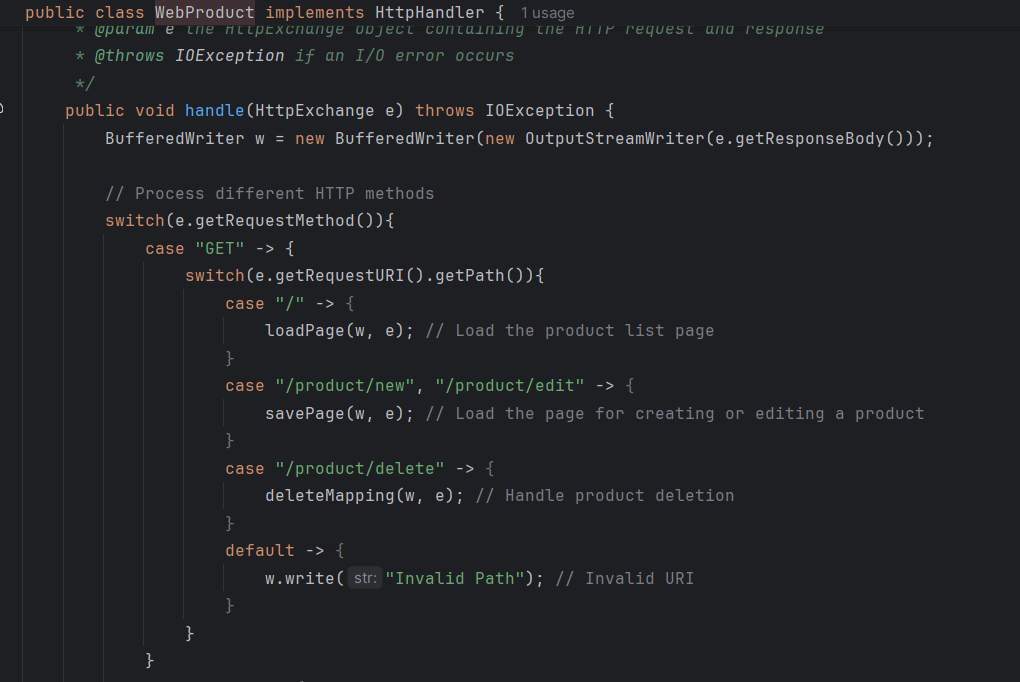
The WebLogin class implements the HttpHandler interface from the com.sun.net.httpserver package, providing a mechanism to handle HTTP requests related to the login functionality on the website. The class interacts with the WebCookie class to manage session data (logged-in user and customer), and it processes both GET and POST HTTP requests to render the login page and authenticate users, respectively.



### WebProduct Class

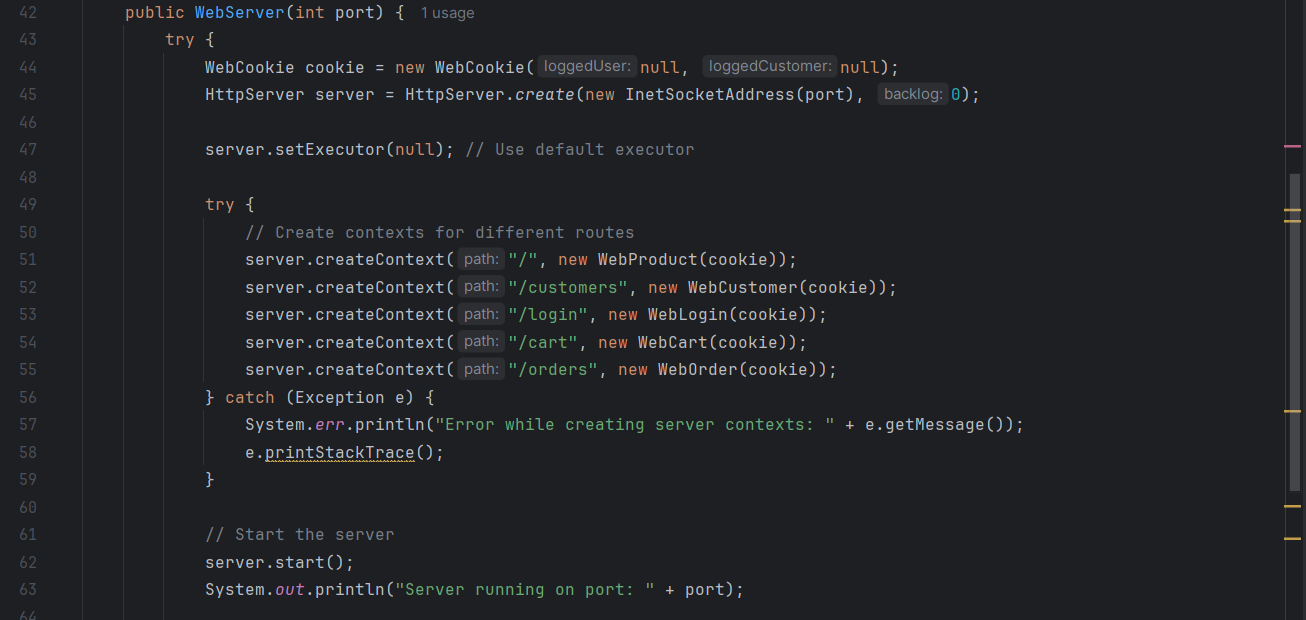
#### :

The WebProduct class is part of the web server's handling module for managing appliance-related HTTP requests. It implements the HttpHandler interface from the com.sun.net.httpserver package to handle various HTTP requests (GET, POST, DELETE). This class interacts with the appliance database through ApplianceDAO and renders HTML pages for displaying, adding, editing, or deleting appliances based on the request's URI and HTTP method. The class includes logic to differentiate the views and functionality depending on user authentication and the specific request paths (e.g., /product/new, /product/edit, /product/delete).



### WebServer Class

The WebServer class is responsible for setting up and starting an HTTP server using Java's com.sun.net.httpserver.HttpServer class. It listens for incoming HTTP requests and routes them to different handlers based on the request paths. The class is designed to serve as the entry point for initializing the web server and binding request handlers to specific URL contexts.



Design pattern-

A screenshot of a computer program

Description automatically generated

Fig 1: singleton design patter, ensuring only one database connection instance is used

**A screenshot of a computer program

Description automatically generated**

Fig 2: DAO design pattern , separating data logic from business logic

A computer screen with text and images

Description automatically generated

Fig 3: factory design pattern for SQL