Byte Me! Food Ordering System - Assignment 4

Overview

This project enhances the **Byte Me!** food ordering system by incorporating a **Graphical User Interface (GUI)**, **I/O Stream Management** for data persistence, and **JUnit Testing** to ensure the system behaves as expected.

Features

1. GUI Enhancements

- **GUI Class (Swing-based):** The GUI is built using **Swing** and provides a user-friendly interface for browsing the menu and viewing pending orders.
    - **Browse Menu:** Displays a list of items available in the canteen, including item names, prices, and availability.
    - **Pending Orders:** Displays orders placed by users, including order number, items ordered, and their status (e.g., preparing, out for delivery).
    - **Navigation Buttons:** Users can navigate between the menu page and the pending orders page using buttons. Both pages feature a "Return to Main Menu" button for seamless navigation.
    - The GUI is designed to display information only. All updates, such as changing order statuses or adding items to the cart, are handled through the **CLI** (Command Line Interface).

2. I/O Stream Management

- **Order History (customerOrders.txt):**
    - The system saves order histories for each customer in a file called `customerOrders.txt`.
    - Every time an order is placed, the order details (food items, quantity, price) are appended to this file. This ensures that order data is retained even after the program is closed.
- **User Management (users.txt):**
    - The system manages user data (username, password, role) using a file called `users.txt`.

    - New user data is appended to this file when a new user registers via the **Signup** feature. This ensures that the user credentials are available for login even after the program is restarted.

3. JUnit Testing

- **CustomerTest:** This test case ensures the system prevents customers from ordering out-of-stock items. It simulates placing an order for an unavailable item and verifies that the order is not processed.
- **LoginTest:** This test case verifies the login functionality by simulating invalid login attempts. It tests for:
    - Invalid usernames (non-existent usernames).
    - Incorrect passwords (wrong password for an existing username).
    - Incorrect roles (attempting to log in with the wrong role).

How to Run

1. Running the CLI Program

- First, the **CLI** program will be run. This allows users to browse the menu, place orders, view pending orders, and manage users.
- You can start the CLI interface by running the `App.java` class.

2. Running the GUI Program

- After the CLI program is executed, the **GUI** program can be run to browse the menu and view pending orders.
- The GUI program is implemented in the `GUI.java` class and can be executed directly to launch the graphical interface.

3. Test the System

- You can run the JUnit tests by executing the **JUnit test classes** (`CustomerTest.java` and `LoginTest.java`) using your IDE.
- The tests will verify:
    - Out-of-stock item handling (CustomerTest).
    - Invalid login attempts (LoginTest).