**Algorithm for the Stationery System**

**1. Start**

1. Display the welcome screen with system details.

**2. Main Menu**

1. Prompt the user to choose their role:
   * 1 for Staff.
   * 2 for Customer.
   * 3 to End Program.
2. If the user chooses 1 (Staff):
   * Go to **Staff Login**.
3. If the user chooses 2 (Customer):
   * Go to **Customer Menu**.
4. If the user chooses 3:
   * Exit the program.

**3. Staff Login**

1. Prompt for username and password.
2. Convert the username to lowercase for uniformity.
3. Validate credentials:
   * If valid, display a welcome message and proceed to the **Staff Menu**.
   * If invalid, display an error message and prompt again.

**4. Staff Menu**

1. Display options:
   * 1 to View Stock.
   * 2 to Update Stock.
   * 3 to Exit.
2. If the user chooses 1:
   * Display the current stock (names, prices, and quantities).
3. If the user chooses 2 (Update Stock):
   * Prompt to choose between:
     + Update Quantity.
     + Update Price.
   * For updating quantity:
     + Display the list of items.
     + Prompt the user to select an item.
     + Ask for the new quantity and update the stock.
   * For updating price:
     + Display the list of items.
     + Prompt the user to select an item.
     + Ask for the new price and update the stock.
4. If the user chooses 3:
   * Return to the main menu.

**5. Customer Menu**

1. Prompt for the customer's name.
2. Display the available stock (names, prices, and quantities).
3. Allow the customer to:
   * Select items to buy.
   * Enter the quantity for each item.
   * Update the total amount.
4. Calculate the total bill with any applicable discounts.
5. Display the invoice:
   * Items bought.
   * Quantities.
   * Prices.
   * Total amount.
   * Discount (if any).
6. Log the purchase details using the logCustomerPurchase function.
7. Update stock quantities based on items purchased.

**6. Save and Log Operations**

1. Save the updated stock details to a file using the saveStockToFile function.
2. Log customer purchases for future reference using the logCustomerPurchase function.

**7. End Program**

1. Display a thank-you message.
2. Terminate the program.

**Note**

This algorithm is modular and ensures the following:

* **Separation of Concerns:** Separate logic for staff and customers.
* **User-Friendly Design:** Clear prompts for input.
* **Data Persistence:** Updated stock is saved to a file for future sessions.
* **Scalability:** Easy to add new functionality (e.g., additional staff roles or customer discounts).

Would you like me to refine or elaborate on any specific part of this algorithm?