Algorithms

Data Retrieval

1. **Initialization:**
    1. Start the process.
    2. Initialize system databases (Product Database, User Database, Admin Database, Store Database, and Flyer Database).
    3. Retrieve the user's location (for nearby store detection) and logged-in status (User/Admin).
    4. Request user input for product search (keywords, categories, store names, or flyer search).

2. **Fetch Product Details (User Request):**
    1. User searches for a product or selects an item from a grocery list.
    2. Query the **Product Database** based on the search keywords or selected item.
        - **Check:** If the product exists in the database.
            - **Yes:** Proceed to the next step.
            - **No:** Display an error message that the product is unavailable.
    3. Retrieve real-time product prices from the **Store Database** for all stores near the user's location.
        - Compare prices across stores.
        - Display the list of stores with corresponding prices.

3. **Fetch Inventory Details:**
    1. Once the product is identified, query the **Store Database** for inventory availability.
        - Check: If the product is available in the inventory of the selected stores.
            - **Yes:** Display available quantity for each store.
            - **No:** Notify the user of stock unavailability.
    2. If inventory exists, fetch additional details such as **product quantity offered for price matching**.

4. **Fetch Flyer Details:**
    1. Query the **Flyer Database** for the current flyer information related to the product or store.
        - Retrieve details such as:
            - Flyer starts and end dates.
            - Flyer-specific prices or discounts.
    2. Display flyer details, including price validity period (start and end dates) and promotional pricing (if applicable).

5. **Admin-Specific Data Fetch (Admin Request):**
    1. If logged in as **Admin**, query additional admin-specific data such as:

- Current flyer status for the selected store.
- Existing product categories for updating or deleting.
- Price matching rules across stores.

2. Retrieve current products and store information for **adding, updating, or deleting** items.
    - Allow admins to modify price matching details, product quantities, and store records.

6. **User-Specific Data Retrieval:**
    1. the **User Database** for logged-in users to fetch:
        - User account details.
        - Personalized grocery list.
        - Past searches or purchase history.
    2. Fetch **notifications** related to product status (availability, price drop, flyer start date, etc.).
    3. Display personalized recommendations based on current flyer data, inventory, and user preferences.

7. **End Process:**
    1. Finalize and display all retrieved information (prices, inventory, flyers, personalized notifications).
    2. End.

Cart Management and Checkout

1. **Add Item to Cart**

    1. Begin
    2. Search for the item by name
    3. If the item exists in the inventory, then
        - If the item already exists in the cart, then
            - Update the quantity or cancel
        - If the item does not exist in the cart, then
            - Add the item to the cart with the specified quantity
    4. Recalculate the total amount in the cart every time an item is added
    5. End

2. **Update Quantity in Cart**

    1. Begin
    2. Select the item, then
        - Update the item quantity or cancel
    3. Recalculate the total amount in the cart once the item is updated
    4. End

3. **Delete Item from Cart**

    1. Begin
    2. Click on the “trash” button
    3. Remove the item from the cart.
    4. Recalculate the total amount in the cart once the item is deleted
    5. End

4. **Checkout Process**

    1. Begin
    2. Ask the user to confirm the order
    3. If the user confirms, then
        - Proceed to the order gateway
        - Complete the transaction
    4. Send a confirmation to the user
    5. Clear the cart.
    6. End

Login Authentication Process

User Login Authentication

1. Start
2. For user input ask user to enter email and password.
3. Validation of email format:
    a. If the email format is invalid (e.g., missing "@" or domain):
        i. Display an error message: "Invalid email format."
        ii. Return to step 2 to prompt for email and password again.
4. Writing a query in database
    a. Now we will execute the query from the database so that we can retrieve the user details using the provided email.
        i. User= database.query(“select * from users where email = email”)
5. Checking if User exists or not
    a. If user is NULL (user not found):
        i. Display an error message: "User not found."
        ii. Return to step 2 to prompt for email and password again.
6. Verify Password:
    a. Compare the provided password with the stored hashed password:
        i. If user.password is equal to the hash of the provided password:
            1. Set the user session as authenticated (e.g., session.setAuthenticated(user)).
            2. Redirect to the user dashboard (e.g., redirect to dashboard).
        ii. Else
            1. Display an error message: "Invalid password."
            2. Return to step 2 to prompt for email and password again.
7. End

Budget Management

Budget Check Feature

1. Start the Budget Check Process
   - Triggered when the user adds or updates an item in the grocery list.

2. Calculate the Total Cost of the List
   - Initialize a variable total_cost = 0.
   - For each item in the grocery list:
     Multiply the quantity by the price per unit to get the item's total cost.
     Add the item's cost to total_cost.

3. Compare Total Cost with Budget
   - If total_cost > budget_limit:
     Proceed to Step 4 (Notify User).
   - Else:
     Skip to Step 7 (Save List).

4. Notify the User
   - Display a warning message: "Budget exceeded. Please adjust your list."

5. Provide Editing Options
   - Allow the user to:
     Edit item quantity
     Remove items
   - If the user makes changes, proceed to Step 6.

6. Recalculate the Total Cost
   - Recompute the total cost using the updated list.
   - Return to Step 3 (Check if the new total is within the budget).

7. Save the List
   - If the total cost is within the budget, save the grocery list.

8. End the Budget Check Process
   - Display a success message: "List saved successfully."