```python
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import json
from collections import Counter

# The ShoppingList class manages shopping lists.
class ShoppingList:
    def __init__(self):
        self.shopping_list = []  # Current shopping list
        self.previous_shopping_lists = []  # Previous shopping lists

    # Method to add a product to the shopping list
    def add_product(self, product):
        self.shopping_list.append(product)
        print(f"{product} added to the shopping list.")

    # Method to remove a product from the shopping list
    def remove_product(self, product):
        if product in self.shopping_list:
            self.shopping_list.remove(product)
            print(f"{product} removed from the shopping list.")
        else:
            print("Product is not in the list.")

    # Method to complete shopping, adds the current list to previous lists and clears the current list
    def complete_shopping(self):
        completed_shopping = self.shopping_list.copy()
        self.previous_shopping_lists.append(completed_shopping)
        self.shopping_list.clear()

    # Method to view previous shopping lists
    def view_previous_lists(self):
        if not self.previous_shopping_lists:
            print("No previous shopping lists found.")
        else:
            for i, shopping_list in enumerate(self.previous_shopping_lists):
                print(f"{i+1}. Shopping List:")
                for product in shopping_list:
                    print(f"- {product}")
                print()

    # Method to clear previous shopping lists
    def clear_previous_lists(self):
        self.previous_shopping_lists.clear()

    # Method to analyze based on previous shopping lists
    def analyze(self):
        if not self.previous_shopping_lists:
            print("No previous shopping lists available for analysis.")
            return
        products = [product for shopping_list in self.previous_shopping_lists for product in shopping_list]
        analysis = Counter(products)
        most_purchased_product = max(analysis, key=analysis.get)
        for product, quantity in analysis.items():
            print(f"{product}: {quantity} units")
        print(f"Most frequently purchased product: {most_purchased_product}. Buying this product in bulk would save you both money and time.")

    # Method to view discounts (not implemented yet)
    def view_discounts(self):
        pass

# The FamilyMembers class manages family members and their shopping lists.
class FamilyMembers:
    def __init__(self):
        self.family_members = {}

    # Method to add a new family member
    def add_member(self, name, password):
        self.family_members[name] = {'password': password, 'list': ShoppingList()}

    # Method to view shopping lists of family members
    def view_members_list(self):
        for name, data in self.family_members.items():
            print(f"{name}'s Shopping List:")
            data['list'].view_previous_lists()
            print()

    # Method to delete a family member's shopping list
    def delete_member_list(self):
        name = input("Enter the username to delete: ")
        password = input("Enter the password: ")
        if name in self.family_members:
            if self.family_members[name]['password'] == password:
                del self.family_members[name]
                print(f"{name} user successfully deleted.")
            else:
                print("Incorrect password.")
        else:
            print("User not found.")

    # Method to access a family member's shopping list
    def access(self, name, password):
        if name in self.family_members:
            if self.family_members[name]['password'] == password:
                return self.family_members[name]['list']
            else:
                print("Incorrect password.")
        else:
```

```
            print("User not found.")

    # Method to view existing family members
    def view_existing_members(self):
        if not self.family_members:
            print("No users exist yet.")
        else:
            print("Existing Users:")
            for name in self.family_members.keys():
                print(f"- {name}")

# Function to display the main menu
def main_menu():
    print("1. Create a new user")
    print("2. User Login")
    print("3. Show existing users")
    print("4. Delete User")
    print("5. Exit")

# Function to create a new shopping list
def create_new_list(family):
    name = input("Enter the username for the new shopping list: ")
    password = input("Set the user password: ")
    family.add_member(name, password)
    print(f"A new shopping list has been created for {name}.")

# Function to access an existing shopping list
def access_existing_list(family):
    name = input("Enter the username you want to access: ")
    password = input("Enter the password: ")
    user = family.access(name, password)
    if user:
        list_menu(user)

# Function representing the shopping list menu
def list_menu(user):
    while True:
        print("\nShopping List Menu:")
        print("1. Add Product")
        print("2. Remove Product")
        print("3. Complete Shopping")
        print("4. View Previous Shopping Lists")
        print("5. Clear Previous Shopping Lists")
        print("6. Analyze")
        print("7. Back to Main Menu")

        choice = input("Select an operation: ")

        if choice == "1":
            product = input("Enter the product you want to add: ")
            user.add_product(product)
        elif choice == "2":
            product = input("Enter the product you want to remove: ")
            user.remove_product(product)
        elif choice == "3":
            user.complete_shopping()
            print("Shopping completed.")
        elif choice == "4":
            user.view_previous_lists()
        elif choice == "5":
            user.clear_previous_lists()
            print("Previous shopping lists cleared.")
        elif choice == "6":
            user.analyze()
        elif choice == "7":
            break
        else:
            print("Invalid choice. Please try again.")

# Main function of the program
def main():
    family = FamilyMembers()
    while True:
        main_menu()
        choice = input("Select an option: ")

        if choice == "1":
            create_new_list(family)
        elif choice == "2":
            access_existing_list(family)
        elif choice == "3":
            family.view_existing_members()
        elif choice == "4":
            family.delete_member_list()
        elif choice == "5":
            print("Exiting the program...")
            break
        else:
            print("Invalid choice. Please try again.")

# Entry point of the program
if __name__ == "__main__":
    main()


# In[ ]:
```