FRUIT ONLINE SHOP APPLICATION

I. Overview

Fruit Shop management system in java is basically developed for manage the Fruit Shop. In the Fruit Shop, product and Shopping management is very important. By making system is computerized it make possible to reduce effort, work is efficient and increase their revenue opportunities for shop owner. The program provides shop owners tools to run their business effectively.

II. Functional Requirements

Main Screen as below:

For Customer (Default)	For Admin	For Sale
FRUIT SHOP SYSTEM *1. Shopping* *2. Login* *3. Exit* *Choose a menu item (1-3):*	*FRUIT SHOP SYSTEM* *1. Manage users* *2. Manage fruits* *3. View orders* *4. Shopping* *5. Logout* *Choose a menu item (1-5):*	*FRUIT SHOP SYSTEM* *1. Manage fruits* *2. View orders* *3. Shopping* *4. Logout* *Choose a menu item (1-4):*

The Admin or Sale's Main Screen is shown for the logged-in use with relevant role; On the application start or after admin/sale chooses to logout, the default screen is displayed

Function details:

1. Manage Users:

- Each User has following attributes: userId, userName, password, userType
  - userId is an auto increased integer (start from 1)
  - userName has >=5 chars, unique & must start with a letter character
  - Password must include >=6 chars, including only letters or numbers
  - userType has value: 1 if the user is admin, else the user is a sale
- On selecting "Manage users" menu from "Main Screen", the below screen is shown

  USERS MANAGEMENT
  List of current users:
  | ++ Id ++ | ++ User Name ++ | ++ Password ++ | ++ Role ++ |

1	*kiennt*	*Vietnam*	*Admin*
3	*tuanta*	*Tuan134*	*Sale*
4	*thuypx*	*14France*	*Sale*

  Choose your actions:
  1. Add new user
  2. Edit/update a user
  3. Delete a user
  4. Main screen
  (Please choose 1, 2 or 3 to add, edit or delete a user respectively; choose 4 to go back)
- After each user is created, updated or deleted, the program returns to the Users Management screen with the users list updated.

2. Manage Fruits:

- Each Fruit has following attributes: Fruit Id, Fruit Name, Price, Quantity and Origin.

    - The Id, Quantity is positive integer; The Price is a positive real number
    - Fruit name & origin are string, include only letters, numbers, or space characters
- On selecting "Manage fruits" menu from "Main Screen", the below screen is shown

FRUITS MANAGEMENT
List of current fruits:

Id	Fruit Name	Origin	Price	Quantity
1	Coconut	Vietnam	2$	6
3	Orange	US	5$	14
4	Apple	Thailand	4$	9
5	Grape	France	9$	7

Choose your actions:
1. Add new fruit
2. Edit/update a fruit
3. Delete a fruit
4. Main screen
(Please choose 1, 2 or 3 to add, edit or delete a fruit respectively; choose 4 to go back)

- After each Fruit is created, updated or deleted, the program returns to the Fruit Management screen with the fruits list updated.

3. View orders

- To view the current orders list, in the form as below

Customer: Marry Carie
Product | Quantity | Price | Amount
1. Apple 3 1$ 3$
2. Mango 2 2$ 4$
Total: 7$

Customer: John Smith
Product | Quantity | Price | Amount
1. JackFruit 3 3$ 9$
2. Mango 2 2$ 4$
Total: 13$

4. Shopping:

On selecting the Shopping menu item, the program displays all *available* fruits. For example:

List of current available fruits:

Id	Fruit Name	Origin	Price	Quantity
1	Coconut	Vietnam	2$	6
3	Orange	US	5$	14
4	Apple	Thailand	4$	9
5	Grape	France	9$	7

To order, customer selects fruit by inputting the fruit id, for example: when customer selects item 1, the program shows:

You selected: Coconut
Please input quantity:

After customer inputs quantity of fruit:

- If the inputted quantity is greater than fruit's available quantity, an error is shown and the customer then has to re-input the lower quantity.

- If the inputted quantity is valid, the program shows message: Do you want to order now (Y/N). If customer selects N, the program returns to List of Fruit to continue ordering. If select Y, the program displays:

  Product. | Quantity | Price | Amount
  Coconut 3 2$ 6$
  Total: 6$
  Input your name:

  Customer inputs his/her name to finish ordering

On finishing the order, the fruit's available quantity is updated accordingly. The user then is redirected back to the main screen.

III. Other Requirements

1. Application design & structure

The team is required to follow the object-oriented analysing & designing approach to design the application components (MVC pattern). For this, with each of the data entities, you need:

- One Model/Entity class to model the entity (define entity attributes & their entry)
- One or more View classes to handle the inputting & outputting the entity information
- One or more Controller classes to include the business logics handling, data storing, calculating, etc.

The software components (classes) are named consistently (refer Oracle Java Coding conventions as mention in the next part) and distributed into the relevant packages, classified by either the data entity (named with the entity names) or the class types (model, view, controller).

2. Data storage requirements

All the application data are stored permanently in the files.
The user data are stored in the user.dat file (object reading/writing), each of other data are stored in one text file (readable outside, text reading/writing)

3. Other requirements

- Strictly follow the Oracle Java coding conventions & practices
- View classes are not allowed to call Controller classes & vice versa
- Both the View & the Controller classes can refer/use the entity classes
- A special class with the **main** method would co-ordinate all other classes
- One or more common classes might be needed to include the shared methods or constants. Those classes can use the entity classes, but are not allowed to user other classes