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 (the menu items are shown according to the logged-in user's role)

For Customer (Default)	For Admin	For Sale
FRUIT SHOP SYSTEM *1. Shopping* *2. Log In* *3. Exit* *Choose a menu item (1-3):*	*FRUIT SHOP SYSTEM* *1. Manage users* *2. Manage fruits* *3. View orders* *4. Shopping* *5. Log Out* *6. Exit* *Choose a menu item (1-6):*	*FRUIT SHOP SYSTEM* *1. Manage fruits* *2. View orders* *3. Shopping* *4. Log Out* *5. Exit* *Choose a menu item (1-6):*

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 log out, the default screen is displayed

Function details:

1. Log In:

- The user can log in when the user status is active and with valid email and password
- In case the user forget his/her password, he can choose to reset password with his email
- After logging in, the relevant Main Screen would be shown to the user (Admin or Sale)

2. Manage Users:

- Each user has following attributes: User ID, Email, Full Name, Password, User Type
  - User Id is an auto increased integer (start from 1)
  - Email: valid email address, unique & must start with a letter character
  - Full name: valid name (no numeric, no special character), <= 30 character length
  - Password must include >=6 chars, including only letters or numbers
  - User Type 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 | Email                  | Full Name            | Role    | Status
  1    kienntvn@gmail.com       Nguyen Trung Kien      Admin    Active
  3    tuanta@yahoo.com         Nguyen Anh Tuan        Sale     Active
  4    thuypx@fpt.edu.vn        Pham Xuan Thuy         Sale     Inactive
Choose your actions:
    1. Add new user
    2. Update a user
    3. Delete a user
    4. Main screen
    5. Exit
(Please choose 1, 2 or 3 to add, edit or delete a user respectively; choose 4 to go back)
```

- After adding new user, an email will be sent to the new user email address for informing
- After one user is created, updated or deleted, the program returns to the Users Management screen with the users list updated.

3. 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, from that to allow user to add new fruit(s), update or delete a specific fruit

```
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. 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)
```

- The user can only delete the fruit(s) which there has not been any customer order yet.
- After each Fruit is created, updated or deleted, the program returns to the Fruit Management screen with the fruits list updated.

4. View orders

- To view the current orders list, in the form as below (the list is shorted by the Order Id)

```
Order: 15, Pending | Customer: Marry Carie (0912568368, kien@fpt.com.vn)
     Product | Quantity | Price | Amount
      Apple       3        1$      3$
     Mango        2        2$      4$
Total: 7$

Order: 17, Paid | Customer: John Smith (0384136861, tuananh@gmail.com)
   Product | Quantity | Price | Amount
   JackFruit      3       3$      9$
     Mango        2       2$      4$
Total: 13$
```

- From this order list screen, the user can choose to
    - Filter the list by the status (Pending, Paid, Canceled) or view all orders
    - Change the status of a specific order from Pending to Paid or Canceled
- When a pending order is changed to paid
    - The relevant fruit(s)' available quantities are updated accordingly.
    - In case available quantity of any fruit in the order is less than the ordered quantity, then a warning message is shown to the user, from that allowing him/her to cancel the order or change the ordered quantity to make the order valid.
- After changing the order information, an updating email would be sent to the customer via his/her registered email address.

5. 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 full name and email to finish ordering. After the order is submitted, a confirmation email would be send to his/her inputted email address

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 analyzing & 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 logic 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 a binary file named user.dat (unreadable outside)
- Each of other data are stored in one text file (readable outside)

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