CS202: IT Workshop I
Mid-Semester Lab examination
04 October 2023
Total marks: 60

Design and develop **a mini inventory management system** as described below.

Class hierarchy 1: "User" is class and it has two subclasses "SuperUser" and "NormalUser".

Class hierarchy 2: "Item" is a class and it has two subclasses "Consumables" and "NonConsumables". There are two more classes "Sell" and "Return".

The data members of the classes are as follows.

User (name, dateOfBirth, Address, PAN)

SuperUser (**SuperUserID, password**, dateOfJoining, salary, permissibleOperations)

NormalUser (userID, dateOfJoining, salary, dutyHourPerDay)

Item (itemCode, price, AvailableQty)

Consumables (dateOfExpiry)

NonConsumables (returnCount)

Sell (dateOfSell, itemCode, quantity, totalAmount)

Return (dateOfReturn, itemCode, quantity, returnAmount)

In order to distinguish between consumable items and non-consumable items, the item code for consumable items should be even numbers (0, 2, 4 ….) and item code for the non-consumable items should be odd numbers (1, 3, 5…..). Hence, make appropriate arrangements in the constructors of Consumables and NonConsumables to check if the item code is valid or not.

Make arraylists to contain the following types of objects in the systems 1) Item 2) Sell 3) Return

System supports following operations:

For SuperUser Class:
i) addNewStock (item): This will add a new item to the system [3]

ii) deleteStock (itemcode): This will delete an existing item (represented by itemcode) from the system. Before deleting, it should ensure that available quantity is nonzero. [5]

iii) modifyStock (itemcode): This will update an item details (it may be price or availableQty). Item is identified by itemcode. The new data should be taken from the user. [3]

For NormalUser Class:

i) sellItem (itemcode, quantity): This indicates selling of an item. This shall reduce the available quantity of the item by amount "quantity". [5]

ii) returnItem (itemcode, quantity): This indicates the return of an item. Only non-consumable items can be returned. The system should not accept a return in case of a consumable item. A return shall increase the available quantity of the item. [3]

iii) displayStock (itemcode): Displays the details of the item indicated by itemcode. [3]

iv) displayToBeExpiredItems(int m) : Display the consumable items that are going to be expired after m months from the current month. User input is m (as String) [5]

v) displaySell (startDate, endDate): Displays the details of the sold items within the date range. [3]

Define all the classes with appropriate variable names and method names. You need to use appropriate datatype for the fields. You may also use static fields wherever applicable. You may also assume additional fields as appropriate. [3+3+3=9]

The main menu (console) should ask for following options: [2]

1. Setup
2. SuperUser mode
3. NormalUser User mode
4. Exit

Note: If you select setup, you can create SuperUser object (exactly one) and a number of NormalUser user objects. Normal user objects can be stored in an array list. [3]

After setup (Option 1) is complete, you can ask the user again (by going back to the main menu) to enter through SuperUser mode or Normal user mode. If the user wants to choose SuperUser mode (Option 2) or Normal user mode (Option 3) before setup, don't allow → Print error msg and go back to the main menu. [3]

For SuperUser mode (Option 2), ask for SuperUserID and password from the user. If they don't match, exit from the SuperUser mode directly and don't allow the other functions to be executed i.e., go back to the main menu. Otherwise, give options to the SuperUser to execute other functions defined in the class. [3]

For NormalUser mode (Option 3), check whether the user exists by scanning the arraylist. If it exists, then give options to the user to execute the functions defined. Else, bring the control back to the main menu. [3]

Using Exit option (option 4), one can come out of the application. [2]

Good programming practice leads to mark [5]