Computer Science

Loughborough University

19COA256: Object-oriented Programming Coursework Assignment

Dr Hossein Nevisi Semester 2

Task

You are to develop a system in Java for a Computer Accessories Shop (CAS) to help the business to handle many of the shop's activities in an easy and practical manner. Each product the shop stocks has a barcode (a unique 6 digit number), brand, colour, connectivity (either wired or wireless), quantity in stock, original cost and retail price. The shop sells two types of product: mouses and keyboards. The different types of keyboard that they sell are standard, internet, gaming and flexible. The keyboards have the US layout or the UK layout. The different types of mouse that they sell are standard and gaming. The mouses may have different number of buttons.

The users of the system will need to have unique user ID, unique username, surname, address (consisting of house number, postcode and city). A user can be an admin of the system or a customer. The admin user has the right to add a product to the system and view all products with all their attributes. A customer in addition to the above users' attributes will need to have a shopping basket. Customers should be able to add items to their shopping basket and pay for all items in the basket (by choosing a payment method which can be either PayPal or Credit Card). The Credit Card payments consist of a 16 digits card number and 3 digits security code, and for PayPal payments, an email address is required. Furthermore, customers need to be able to cancel all items in the shopping basket and mark all as save for later. A customer should be also able to view all product attributes except original cost.

The program should include at least the products listed in Appendix B.

Functionality Two types of roles need to be created: "Admin" and "Customer". The system works with the list of users provided in a file (see Appendix A). Once the user logs into the system by choosing from the list of available usernames, they should be able to access the below functionalities according their roles.

- **Admin**
  - View all products with all their attributes sorted descending by quantity in stock.
  - Add a product to the current product list (stock) with these parameters: *barcode, brand, colour, connectivity, quantity in stock, original cost, retail price*. If the product being added is a keyboard then the following parameters need to be included as well: *keyboard type, keyboard layout*, and if the product is a mouse then the following parameters need to be included: *mouse type, number of buttons*.
- **Customer**
  - View all available products with all their attributes (except original cost) sorted descending by quantity in stock.
  - Add an item to their shopping basket.
  - Pay for all items in the basket by choosing from the following payment methods (see how the system must log this activity in the Activity Logging paragraph):
    - PayPal in which the customers need to enter their PayPal email address
    - Credit Card in which the customers need to enter a 16 digits card number and 3 digits security code

The pay function will display a different message on the screen based on the chosen payment method as follows:

  * For PayPal, the message should be "amount+ paid using PayPal".
  * For Credit Card, the message should be "amount+ paid using Credit Card".
  + In the above messages, "amount" must be replaced by the numeric value of the total payment amount. This transaction must make the basket empty as well.

- Cancel their shopping basket which empties out the entire content of the basket (see how the system must log this activity in the Activity Logging paragraph).
- Save for later which empties out the entire content of the basket (see how the system must log this activity in the Activity Logging paragraph).
- Search ... Displays a list of all available products to purchase on the screen can be filtered by
  * your chosen brand, and/or
  * keyboards with UK layout

  (all product details must be displayed except original cost)

The system must log these customer activities: pay, cancel and save for later, in a file as follows:

- **Activity Logging**
  * After the completion of the above activities (pay, cancel and save for later), for each item previously in the basket, a new line must be added to the log file (see Appendix C) with these parameters: user ID, postcode, barcode, retail price, quantity of item, payment method (only for pay activity), status and date. The status is logged based on pay, cancel and save for later as "purchased", "cancelled" or "saved". The date format is dd-mm-yyyy, e.g. 13-03-2020 for the 13th March 2020, and it should show when the record is logged. The records in the log file must be in descending date order.

The program should be able to avoid and detect errors. Examples: (i) The customers should be notified if they want to buy an item that is out of stock. (ii) Products with a same barcode must not be added more than once.

User Interface You should provide a graphical user interface (GUI) or command-line interface (CLI). A GUI scores more points than a CLI. A good interface is intuitive, quick and easy to use.

Design and Implementation

Both Design and implementation parts must be based on the Object Oriented methodology. Before you start programming, design your software and structure it in an object-oriented way. Identify classes, their attributes and methods, think about how they interact with each other, and design the user interface before you produce any code.

You should use Eclipse to develop the program. Your code must compile and run in Eclipse on Windows operating system.

Note that this is an individual exercise and that you must not share any code.

Writing Understandable Code

Your code must be understandable for readers. For this purpose, you should use

- meaningful names for your classes, objects, methods and variables
- comments properly in your code
- indentation appropriately

Documentation and Submission

You are expected to submit

- all source files (i.e. java files) as well as all other files needed to compile and run the program in Eclipse including
  - a text file called `UserAccounts.txt` which should be used to read user details from (see Appendix A),
  - a text file called `Stock.txt` which should be used to read/write product details from/into (see Appendix B), and
  - a text file called `ActivityLog.txt` which should be used to log user activities (see Appendix C)
- a jar file `cas.jar` that is executed by the command `java -jar cas.jar`.
- a class diagram which describes your object-oriented approach by showing all classes, their attributes, methods, relationships, etc. The file must be called `ClassDiagram` and it must be either in PDF or JPG format.
- a PDF document called `NotCompleted` describing which functionality is not completed or not working properly in your application

All files need to be zipped into a single zip file called

`CAS2020_<your first name>_<your last name>.zip`

and submitted electronically via Learn.

The hand-in date is Wednesday, 13th May 2020, 2:00pm.

Marking

Your software will be marked with respect to functionality as well as object-oriented implementation and design. The following is a very rough guide how the assignment will be marked.

- Object-oriented design and high-level definition (coding) of the classes: 35%
  - Class Diagram: 15%
  - High-level definition (coding) of the classes based on the Object-Oriented concepts and features: 20%
- User interface: 15%
- Functionality: 44%
  - Add a product to the stock: 6%
  - View the list of all products : 6%
  - Add items to the shopping basket: 4%
  - Pay for all items in the basket: 10%
  - Search: 6%
  - Cancel all, save for later, and log activities: 12%
  - If any of the above functions works fine but not implemented using Object-Oriented methodology, half of the mark will be reduced, and if the code is not understandable, one-fifth of the mark will be reduced.
- Validating data and detecting errors: 6%

The following requirements are absolutely essential. If any of them is not fulfilled, you will lose the main part of the mark.

- The code must compile and run.

- The jar file containing the program must run calling `java -jar cas.jar`.
- The implemented functionality is usable via an interface.
- The interface is easy to work for the users work with the system for the first time.
- You must use object oriented programming approach to develop the system.
- The implementation part must be based on your class diagram.
- You should only use text files (.txt) to read/write data from/into.

Please be aware:

- The source files (i.e. java files) and all other files necessary for compiling and running the program must be provided. The code must compile and run in Eclipse on Windows operating system.
- If the code is not submitted or does not compile and if there is no working jar file, the program cannot be tested and will therefore not pass.

A User Accounts

The initial list of valid users of the system (which must be stored in UserAccounts.txt) should be as follows, where each user is given as a comma-separated list in the form: `user ID, username, surname, house number, postcode, city, role`.

```
101, user1, Smith, 12, LE11 3TU, Loughborough, admin
102, user2, Williams, 14, E20 3BS, London, customer
103, user3, Taylor, 100, BN1 3XP, Brighton, customer
104, user4, Lee, 57, PA3 2SW, Glasgow, customer
```

B Stock Data

The initial list of products in the stock (which must be stored in Stock.txt) should be as follows, where each product is given as a comma-separated list in the form: *barcode, device name, device type, brand, colour, connectivity, quantity in stock, original cost, retail price, additional information*. The additional information for keyboards is their layout (either UK or US) and for mouses is number of their buttons.

```
112233, mouse, gaming, Logitech, black, wireless, 15, 7.50, 9.50, 3
123456, keyboard, gaming, Corsair, black, wired, 2, 30.0, 39.99, UK
124455, keyboard, standard, Advent, white, wired, 10, 3.50, 5.99, UK
124566, mouse, standard, Advent, grey, wired, 15, 2.50, 4.99, 2
125567, keyboard, internet, Logitech, black, wireless, 4, 25.99, 30.0, US
221101, mouse, standard, Logitech, black, wired, 3, 3.0, 6.99, 3
221122, mouse, gaming, Razer, black, wired, 1, 28.0, 40.99, 7
223044, mouse, gaming, Anker, blue, wired, 3, 16.0, 18.50, 10
234555, keyboard, standard, Apple, white, wireless, 10, 75.50, 85.50, US
235066, keyboard, flexible, Microsoft, black, wireless, 5, 45.50, 60.0, UK
236677, mouse, standard, Asus, blue, wireless, 8, 10.0, 16.99, 5
237700, mouse, gaming, Anker, black, wired, 1, 8.0, 11.99, 7
237788, keyboard, standard, Logitech, grey, wired, 15, 6.50, 8.99, UK
```

```
237799, mouse, standard, Logitech, grey, wireless, 13, 7.0, 8.0, 2
238800, keyboard, internet, Microsoft, black, wired, 10, 26.50, 30.0, US
```

C Activity Logs

After the completion of pay, cancel and save for later, for each item previously in the basket, a new line must be added to ActivityLog.txt as a comma-separated list in the form: user ID, postcode, barcode, retail price, quantity of item, status, payment method (only for pay activity) and date. The records in the log file must be in descending date order. See below an example of how the information is logged into ActivityLog.txt:

```
104, PA3 2SW, 123456, 39.99, 1, purchased, PayPal, 20-02-2020
104, PA3 2SW, 124566, 4.99, 2, purchased, PayPal, 20-02-2020
103, BN1 3XP, 124566, 4.99, 2, saved, , 18-02-2020
104, PA3 2SW, 237788, 8.99, 5, cancelled, , 15-01-2020
104, PA3 2SW, 112233, 9.50, 1, cancelled, , 15-01-2020
102, E20 3BS, 234555, 85.50, 2, purchased, Credit Card, 10-01-2020
```