Computer Science

Loughborough University

21COA256: 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: keyboards and mice. The different types of keyboard that they sell are standard, flexible and gaming. The keyboards have the US layout or the UK layout. The different types of mouse that they sell are standard and gaming. The mice may have different number of buttons.

The users of the system will need to have unique user ID, unique username, name, 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 new 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 6-digit card number and 3-digit 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. 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 ascending by retail price.
  - 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 ascending by retail price.
  - Add items to their shopping basket.
  - View items in their shopping basket.
  - Pay for all items in the basket by choosing from the following payment methods:
    - PayPal in which the customers need to enter their PayPal email address.

    * Credit Card in which the customers need to enter a 6-digit card number and 3-digit security code.

  The pay function must update the stock, make the basket empty, and 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, and the delivery address is [full address]++".
    * For Credit Card, the message should be "amount+ paid using Credit Card, and the delivery address is [full address]++".

    \+ In the above messages, [amount] must be replaced by the numeric value of the total payment amount.

    ++ In the above messages, [full address] must be replaced by the user's (buyer's) full address.

- Cancel their shopping basket which empties out the entire content of the basket.
- Search/Filter the list of all available products to purchase by the following fields and view the results:
    * brand, and/or
    * mouse's button quantity

  (all product details must be displayed except original cost)

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 into stock 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 as your work will be assessed on Windows.

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), and
  - a text file called `Stock.txt` which should be used to read/write product details from/into (see Appendix B).
- An executable jar file `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

`CAS_<your initial>_<your last name>.zip`

and submitted electronically via Learn.

The hand-in date is Wednesday, 18th May 2022, 3:00pm.

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: 42%
  - Add a product to the stock: 7%
  - View the list of all products: 6%
  - Add items to the shopping basket: 3%
  - View items in their shopping basket: 4%
  - Pay for all items in the basket and showing proper message: 12%
  - Search: 8%
  - Cancel the shopping basket: 2%
  - 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: 8%

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 by double-clicking or by 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.
- The user interface cannot be a combination of CLI and GUI.

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.

Plagiarism

The coursework is individual and therefore should of course be your own work. Failure to do so would leave you open to prosecution for Academic Misconduct.

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, name, house number, postcode, city, role`.

```
101, user1, Daniel, 12, LE11 3TU, Loughborough, admin
102, user2, Fen, 14, E20 3BS, London, customer
103, user3, Emma, 100, BN1 3XP, Brighton, customer
104, user4, Bowen, 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 mice 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, flexible, 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, flexible, Microsoft, black, wired, 10, 26.50, 30.0, US
```