Due Date: 30.12.2019 23:55

CENG211 – Programming Fundamentals
Homework #4

In this homework you are expected to implement an "**Online Marketing Application**" in Java. You should fulfill the concepts of:

- Object Oriented Design
- Forms of Inheritance
- Generics
- Collections
- UML Class Diagrams
- UML Sequence Diagrams

In the Online Marketing Application, you are expected to build an application that manages the online marketing operations. In this application, there are products and users which are explained in detail below. Also, you are expected to draw a sequence diagram in the given scenario. Additionally, you are required to update the given class diagram, if necessary.

Products

The online marketing application sells houseware, accessories, cosmetic and electronic products. These products are recorded in a file namely "products.csv".

- There is **{product category, product name, price, weight, quantity}** information at each line of this file. Also, each product holds its cargo price information. Cargo price is calculated as **weight * 2.0** for electronic and houseware products.
- The accessories are fragile products, which have a cargo price that is **4.5** times of an electronic or houseware product i.e., **weight * 2.0 * 4.5**. The cosmetic products are fast consumption products that have a cargo price **3.5** times of an electronic or houseware product i.e., **weight * 2.0 *3.5.**
- There should be a product inventory which holds the products and their corresponding quantity. In this product inventory, products should be added and removed in the given quantity. Also, the products should be found by their name. If a product is not existing in the inventory, a ProductNotFoundException should be thrown.

  Users

In this application, there are two types of user which are an admin user and a regular user. A regular user could be either a supplier or a customer. All users have a username, a password and an active balance. The username and the password are required for a user in order to sign in. Active balance of a user is zero when a user is first created in the system. Users can deposit money to raise their active balance.

- Regular users have contact information in order to sell or buy products. Contact information includes telephone, email and address.
- A regular user i.e. a customer or a supplier could have more than one address. These addresses should be found by their title in order to allow a customer to select his/her delivery address.
- Additionally, a customer can add products to the basket or remove products from the basket. Also, he/she can buy the products in the basket and make a payment. When the customer makes the payment, the total cost of the products which are total product price and the total cargo price should be decreased from the active balance. The decreased amount should be added to the active balance of the supplier.
- When the customer makes the payment, the products in the basket should be added to the customer's bought products inventory. This inventory includes the bought products and their corresponding quantities e.g. when a customer bought two identical pillows, bought products inventory includes <pillow,2>.
- If the total cost of the bought products by the customer is more than 2000 TL, the customer's new shopping will be free of cargo price. This 2000 TL includes the cost of the bought products prices without cargo prices. For each 2000 TL or more purchases, the customer gains one free of charge cargo.
- The basket of the customer should also hold a product inventory e.g. <candle,5>.
- A supplier, who has a shop should, register its shop with its name and tax number (16 digit unique number). The shop also has a product inventory e.g. <laptop,20>.
- Also, the sold products should be added to the supplier's old products inventory in order to calculate endorsement.
- The supplier can sell products after the admin's approval. After a customer makes the payment, the cost of the shopping, both the product's cost and the cargo cost should be added to the active balance of the supplier. The supplier is required to pay the %2 of this total cost (product price + cargo price) to the admin.
- Additionally, a supplier can accept the returning products. When the returning products' acceptance is completed, the products' prices without cargo price, should be returned to the customer. When a supplier accepts the returning products, remember to remove them from sold products inventory.
- The users of this application are recorded into the "users.csv" file. The user type 1 is the admin, type 2 is a customer and type 3 is a supplier.
- The products should be added to the shop inventories according to the category of the products and the category of the shops. For example, the cosmetic products should be in the cosmetic shop.
- Remember that a regular user i.e. a customer or a supplier could have more than one address. In this case, the other addresses' information is added at the end of the line in the correct order. Therefore, if a regular user adds a new address, the "users.csv" file should be updated accordingly.

In this homework you are given the UML Class diagram of the system in Figure 1. In order to implement this application, you need to follow this UML class diagram. You can add new classes, if necessary. Please update your class diagram and attach to the project zip, if you add any classes.

- This application's objects should be created automatically from the given files.
- You should decide the fields and methods of classes. Please design your classes wisely.
- **Draw a sequence diagram** that demonstrates a customer returns some of bought products.
- If a user interaction is required, please list the options to the user in order to make your application user-friendly. For example:

  ……

  Which operation do you want to perform?

1- SIGN UP
2- SIGN IN
3- SIGN OUT

User Input:1

…….

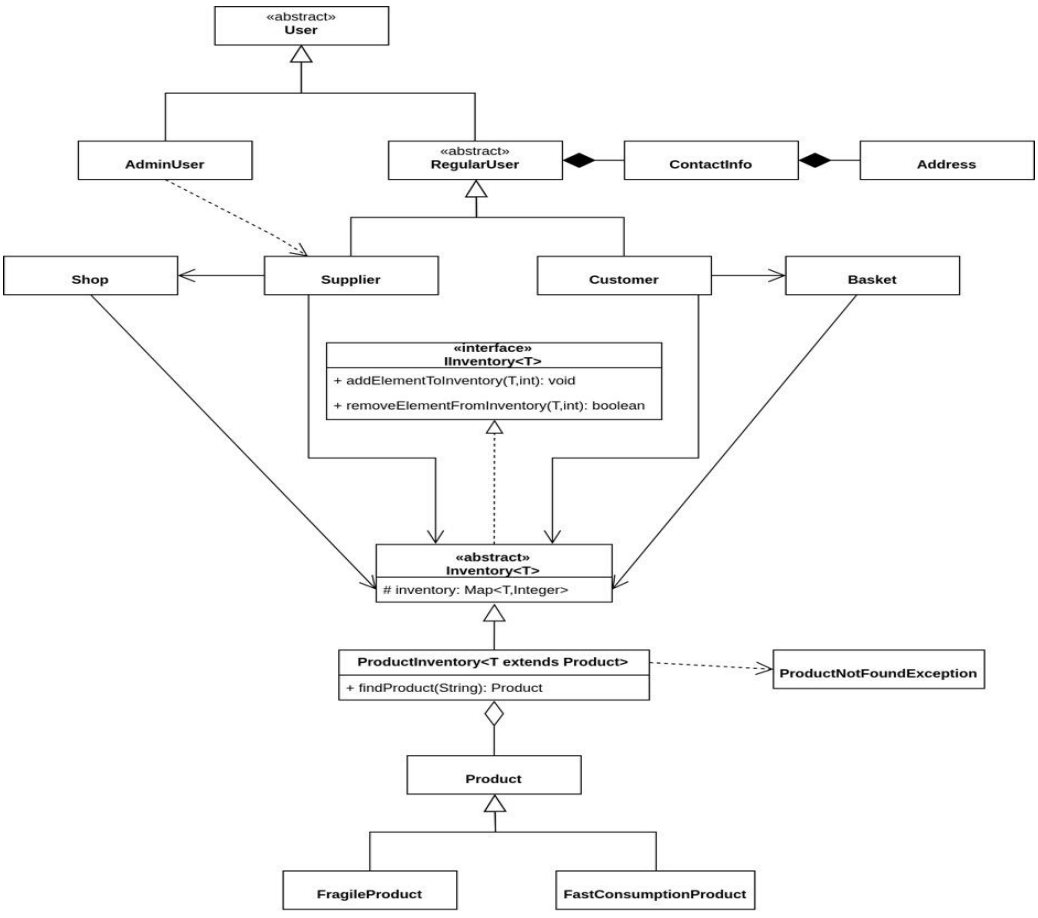

Figure 1. UML Class Diagram

Important Notes:

1. You are expected to write clean, readable, and tester-friendly code (e.g. make the selections with numbers). Please try to maximize reusability and prevent from redundancy in your methods.

2. You are expected to implement all classes, interfaces and exceptions in UML class diagram.
3. You should create an instance of each product. Hold the quantity of a product in the inventory.
4. You can update the class diagram using draw.io [1] by using the attached XML file .

Assignment Rules:

1. In this lecture's homework, there is no cheating allowed. If any cheating has been detected, they will be **graded as 0** and there will be no further discussion on this.

2. You are expected to submit your homework in groups. Therefore, **only one of you** will be sufficient to submit your homework.
3. Make sure you export your homework as an **Eclipse project**. You can use other IDEs as well, however, you must test if it supported by Eclipse.
4. Submit your homework through **CMS**.
5. Name and export your Java Project with your assigned **group ID** (which will be announced on CMS) as the given format below:

   G25_CENG211_HW3.zip

6. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.

References:

[1] https://www.draw.io/