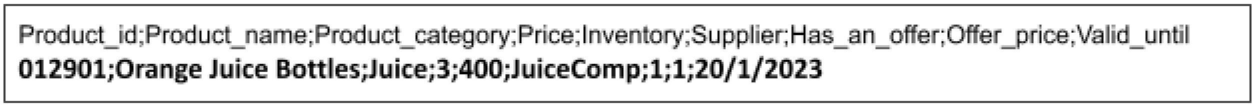
Birzeit University
Department of Electrical & Computer Engineering
Summer Semester, 2022/2023 ENCS3130
Linux Laboratory
Python Project – E-commerce System

Problem Overview:

E-commerce involves using online websites and computer tools to help businesses sell things and interact with customers over the Internet. The E-commerce Administrator is in charge of running the online store. On the flip side, the Shopper is the person who looks around and buys things from the online store. For this project, you're responsible for building an online shopping system that has these specific options:

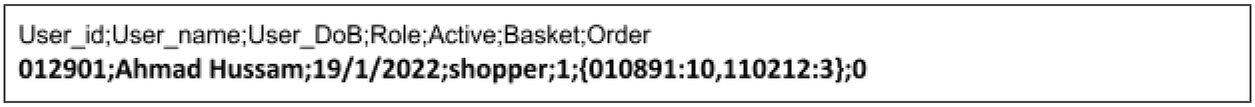
1. Add product (admin-only)
2. Place an item on sale (admin-only)
3. Update product (admin-only)
4. Add a new user (admin-only)
5. Update user (admin-only)
6. Display all users (admin-only)
7. List products (admin and shopper)
8. List shoppers (admin)
9. Add product to the basket (shopper-only)
10. Display basket (shopper-only)
11. Update basket (shopper-only)
12. Place order (shopper-only)
13. Execute order (admin-only)
14. Save products to a file (admin-only)
15. Save users to a text file (admin-only)
16. Exit

If a shopper attempts to access any of the admin options, a clear 'Access Denied' message should be displayed on the screen. Similarly, for the options that are intended for shoppers. At launch, the system should load a list of products from a text file (“**products.txt**”) saved in a file according to the following format:



```
Product_id;Product_name;Product_category;Price;Inventory;Supplier;Has_an_offer;Offer_price;Valid_until
012901;Orange Juice Bottles;Juice;3;400;JuiceComp;1;1;20/1/2023
```

The system should also load a list of users from a text file (“**users.txt**”) according to the following format:



```
User_id;User_name;User_DoB;Role;Active;Basket;Order
012901;Ahmad Hussam;19/1/2022;shopper;1;{010891:10,110212:3};0
```

The details of each field are described later. After that, the system must ask the user for the user_id to determine the role of the user (admin or shopper). If the user_id is not in the user's file, a clear 'Access Denied' message should be displayed on the screen. Also, the system should always validate all inputs from text files and users (date, int, string, …) and print a clear message on the screen for any invalid input.

The specifications of each option are as follows:

1. Add product (admin-only)

To add a product to the system, the **admin** must insert the following information using the standard input

- **Product_id** (unique integer): a 6-digit unique code for the product
- **Product_name** (String): item name such as orange juice bottles, apple juice bottles, banana, minced meat (1 kg), teeth brush, …, etc;
- **Product_category** (ENUM): the class of the product. In the system, products are classified into categories such as "clothes", "beauty", "shoes", …, etc)
- **Price** (Integer): the price of the product in dollars
- **Inventory** (Integer): number of items available for sale from the product
- **Supplier** (String): the company made the item or the provider how imported the item
- **Has_on_offer** (integer): one if the product is on sale otherwise zero.

2. Place an item on sale (admin-only)

After selecting a specific product, the admin can place an item on sale using this option. This requires the system to apply the following procedure:

1. Set the **Has_on_offer flag** of the product to one.
2. Add two attributes to the product:
   - **Offer_price** (Integer): the reduced price for the product.
   - **Valid_until** (Date): The date until which the discount on the item is valid.

3. Update product (admin-only)

The admin needs to select a product based on its code and has the option to update any field of the product other than the product code. The admin may use this option to add an offer on an existing product.

4. Add a new user (admin-only)

To add a user to the system, the **admin** must insert the following information using the standard input

- **User_id** (unique integer): a 6-digit unique code for the item
- **User name** (String)
- **User date of birth** (Date)
- **Role** (Integer): the role of the user admin or shopper
- **Active** (Integer): one for active users otherwise not active

For every added user, the system should define two additional attributes:

- **Basket** (dictionary {product_id: number of items}): a dictionary that contains products and the number of items from each product selected for purchase by the user. The basket will be empty if the user does not select any items.
- **Order** (integer): one if the user finished adding items to the basket and wants to make an order, otherwise zero.

5. Update user (admin-only)

The admin needs to select a user based on its code and has the option to update any field other than the user code.

6. Display all users (admin-only)

Using this option, the admin can display the information of all users on the screen.

7. List products (admin and shopper)

Using this option, the admin and shopper can display all the information of products in the E-commerce system on the screen based on specific criteria as follows:

- **All:** all products
- **Offers:** products that have offers/discount
- **Category:** products belonging to a specific category. The user must input the name of the category.
- **Name:** products with the name entered by the user. It's important to note that a single product might be present in the system but offered by various suppliers.

8. List shoppers (admin)

Using this option, the admin can display all the information of shoppers in the E-commerce system on the screen based on specific criteria as follows:

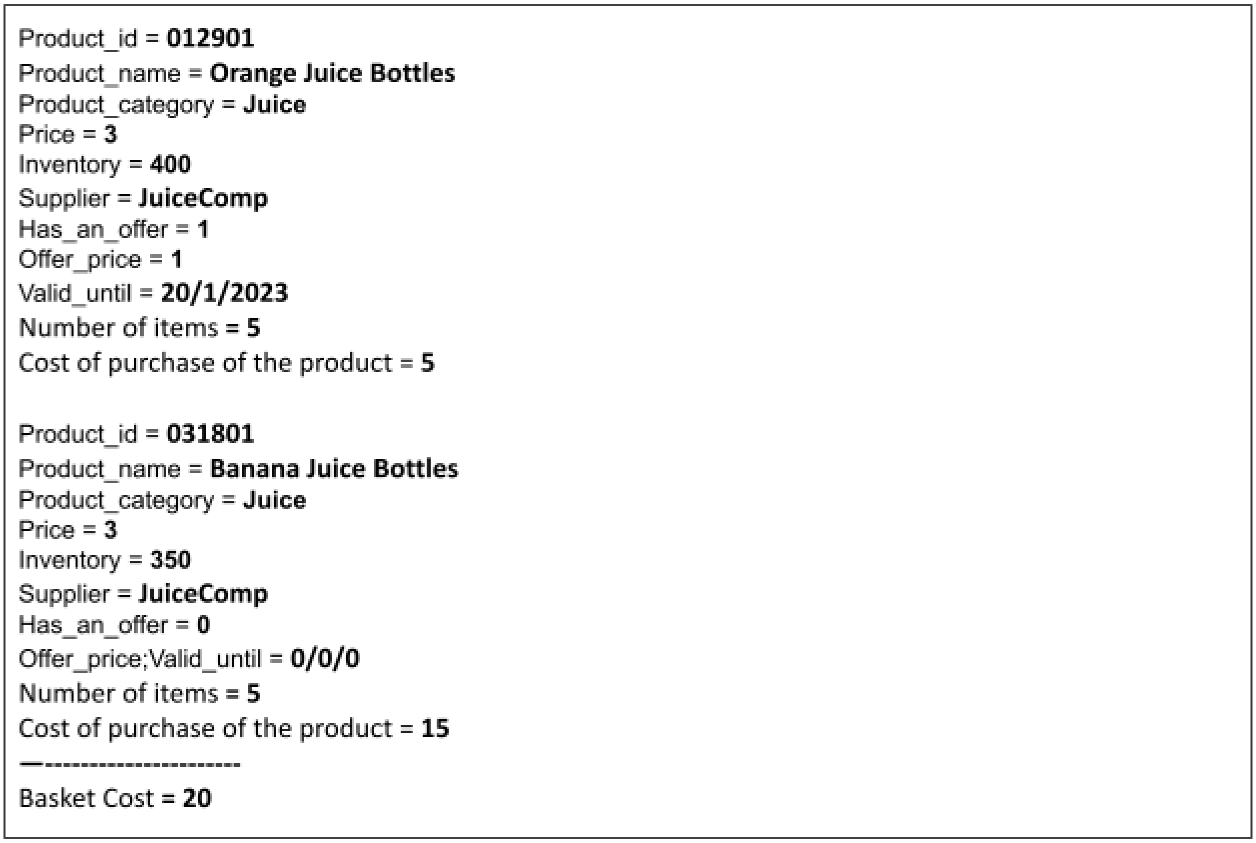
- **All:** all shoppers
- **With items in the basket:** all shoppers that have added products for purchase to the basket.
- **Has unprocessed orders:** all shoppers that have made an order and the order is still not processed by the admin.
- **Requested an order**: al shoppers that have an order that needs to be processed.

9. Add product to the basket (shopper-only)

Using this option, the shopper can add products to the basket. Products are added based on product id. The shopper must decide the product quantity (number of items from the product) to be added.

10. Display basket (shopper-only)

Using this option, the shopper can display the products in the basket. In addition to the product details, the system must print on screen the cost of purchase of the product (the price of the product x number of items) and the total cost of the items in the basket.



```
Product_id = 012901
Product_name = Orange Juice Bottles
Product_category = Juice
Price = 3
Inventory = 400
Supplier = JuiceComp
Has_an_offer = 1
Offer_price = 1
Valid_until = 20/1/2023
Number of items = 5
Cost of purchase of the product = 5

Product_id = 031801
Product_name = Banana Juice Bottles
Product_category = Juice
Price = 3
Inventory = 350
Supplier = JuiceComp
Has_an_offer = 0
Offer_price;Valid_until = 0/0/0
Number of items = 5
Cost of purchase of the product = 15
—----------------------
Basket Cost = 20
```

11. Update basket (shopper-only)

Using this option, the shopper can update the basket as follows:

- **Clear:** Remove all products from the basket
- **Remove:** Remove a specific product from the basket based on product id
- **Update:** Change the number of items of a particular product in the basket based on product id

12. Place order (shopper-only)

Using this option, the shopper can request the purchase of items by changing the value of the order field as described before.

13. Execute order (admin-only)

Using this option, the admin can execute an order. This requires the system to apply the following procedure:

3. Deduct the selected items from the product's inventory. For instance, if a product initially has 400 items in inventory and a shopper basket contains 5 items from this product, the inventory will be adjusted to 395 after the order is executed.
4. Clear the items from the shopper basket.

14. Save products to a file (admin-only)

Save the products including all fields described earlier to a text file. The system should ask the admin to input the name of the text file.

15. Save users to a text file (admin-only)

Save the users including all fields described earlier to a text file. The system should ask the admin to input the name of the text file.

16. Exit (admin and shopper)

The execution of this command will terminate the software. The system should ask the user to save products and users if the user hasn't done that already.

Submissions:

- You need to submit the code in .py format.
- You also need to submit a "users.txt" file and at least three "products.txt" files.
- Write a report with
  - Screenshots for the main menu and any other submenus and the description of each of them;
  - Screenshots for the output of the options in the software mentioned above.

Important notes:

- Write the code for the Python program to satisfy the requirements described above.
- You must use functions (at least one function for each option).
- **You must use OOP concepts (classes, inheritance …).**
- **You must organize your project in modules. I. e. have each class and the main function in separate modules (Python scripts).**
- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your script has enough comments inserted to add clarity.
- Work in groups of at most two students (it is ok to make groups from different sections)
- Deadline: Saturday, 9 September 2023 at 11:59 pm. Please submit your project through Ritaj as a reply to this message.
- This project is per group effort: instances of cheating will result in you failing the lab.

Grading Policy:

Item	**Weight**
System launch - load products from a text file	5
System launch - load users from a text file	5
System launch - validate user id and print access denied messages	5
Add product (admin-only)	5
Place an item on sale (admin-only)	5
Add a new user (admin-only)	5
Update user (admin-only)	5
Display all users (admin-only)	5
List products (admin and shopper)	5
List shoppers (admin)	5
Add product to the basket (shopper-only)	5
Display basket (shopper-only)	5
Update basket (shopper-only)	5
Place order (shopper-only)	5
Execute order (admin-only)	5
Save products to a file (admin-only)	5
Save users to a text file (admin-only)	5
Exit (admin and shopper)	5
Using OOP	10
Use modules	10
Code Structure	10
Discussion	30
Total	**150**