Term Project: Online ordering and delivery system

□ Overview

You are required to design and implement a software system that acts as a centralized platform that performs all required operations related to online ordering and delivery service. This system serves customers such that they can reach and order from different restaurants and marketplaces located nearby more smoothly and efficiently via their mobiles or PCs. On the other hand, the system enables vendors to promote their products for a wide range of people with attractive options like popularity, offers, and online review. Our system will be customized and simple such that it offers basic functions with restrictions.

□ Database

The system maintains a database for all involved customers, providers, orders, and offers. Hence, your system must include an I/O file for each entity, so we will have four files in total. These files must be **permanent** (remain after program halt) and accessed by users to achieve their desired functionalities and be updated continuously throughout the running time of the system. Every **customer** has the following information: name (unique), password, and cash credit. Whereas every **provider** has the following information: name (unique), category, location, review, delivery rate, income, and list of items. For every **order**, the system records the following information: ID (unique), customer name, provider name, status, list of ordered items, and total cost. Finally, the **offer** record has the following information: ID (unique), provider name, item ID, quantity in the offer, and offer price.

As a prerequisite, your implementation program must initially load the database with a set of customers and providers **automatically** at the beginning of execution just before doing anything else. The following data must be loaded as is upon running your program:

[**CUSTOMERS**]

Name	Password	Cash credit
ali	11	100 JD
omar	22	75 JD
maha	33	50 JD
anas	44	80 JD
reem	55	120 JD

[**PROVIDERS**]

Name	Category	Location	Review	Delivery rate	List of items	Income
Hot and Cold	Food	Area 1	68%	5%	I1,I2,I5	100
My Bread	Food	Area 1	75%	7%	I2,I3,I4,I5	200
Good Recipe	Food	Area 2	90%	12%	I1,I2,I3,I4,I5	300
Tasty	Food	Area 3	73%	6%	I1,I2,I3,I5	400
Stop Here	Market	Area 1	88%	10%	I6,I7,I8,I9,I10	500
Good Mart	Market	Area 2	95%	15%	I6,I7,I8,I9,I10	600

WMs	Market	Area 2	82%	9%	I6,I8,I9,I10	700
24Hours	Market	Area 3	79%	8%	I6,I7,I8,I9	800
C-Market	Market	Area 1	92%	14%	I6,I7,I8,I9,I10	900

[**ITEMS**]

ID	Description	Price/unit	Quantity
I1	Juice	4	10
I2	Sandwich	8	30
I3	Sweet	15	20
I4	Steak	20	15
I5	Salad	5	25
I6	Hand wash	3	50
I7	Spices	6	35
I8	Tissues	10	70
I9	Oil	7	30
I10	Sanitizer	2	100

Note that the table **ITEMS** above is not a file in our database, instead, it is a raw data used to fill the items list of each provider. So, it is used just to not duplicate item information for each provider.

□ Requirement Specification

The providers (or the vendors) in our system are divided into two categories: food and market. Moreover, every provider serves customers in specific area. In this regard, our system covers three areas: **Area 1**, **Area 2**, and **Area 3**. Hence, the customer must determine both his location (area) and the preferred category upon placing the order.

The order has three status values: ^{1}**not paid**, which is used once the order is placed just before paying, ^{2}**paid**, which is used once the customer pays the total cost of the placed order, or ^{3}**delivered**, which is used once the order is delivered to the customer just after paying.

Providers may present offers to customers in order to motivate them to buy more quantities. In this regard, the provider specifies the quantity of items inside the offer package and the offer price. For example, the WMs provider may create an offer that has **quantity=4** of the item “**ID=I6**”:”Hand wash” and the **offer price=10**.

When the customer chooses a provider to serve his order, the system must show the offers related to that provider letting the customer to buy it.

Just after the order is placed, the customer is required to pay the total cost of it in order to prepare it for delivery. Hence, the total cost of the order is the cost of the ordered items in addition to the delivery cost. **Delivery cost= delivery rate*the cost of the ordered items**.

Once the order is delivered to the customer, he must post a review (evaluation) for the provider that delivers the order to build a trust metric for upcoming orders. The **review** field in the provider file is calculated as the average of the current **review** value and the **newly entered review** value. For example, if the current review value for a provider in the file=**80%** and the customer enters a review value of **90%**, then the review value = Average (**80%**,**90%**) = **85%**. As a result, the review value in the provider file must be updated to be **85%** instead of **80%**.

The system has **six** users: **five** customers and **one** administrator. They all must log in using their names and passwords. The administrator logs in using the following credential:

username: **admin** password: **00**

Refer to **page1** to get the name and password of each of the five customers that will be used for logging in the system.

The system is required to satisfy the main functional requirements for all users (customers and administrator). This necessitate to explore high-level use cases then decomposing them into sub use cases to draw a full image of the involved implementation steps. Consequently, our system has the following high-level use cases for the **administrator**:

1. Add new item for a specific provider.
2. Present new offer.
3. View all providers.
4. View all customers.
5. View all orders.
6. View all offers.
7. Deliver all paid orders.
8. Cancel offer.

and the following high-level use cases for the **customer**:

1. Place new order.
2. View my orders.
3. Pay for order.
4. Post a review for a provider.

YOUR PROGRAM MUST SHOW ONLY THE FUNCTIONS RELATED TO THE CURRENTLY LOGGED IN USER. THEREFORE, UPON RUNNING YOUR PROGRAM, THE SYSTEM MUST DISPLAY TWO LOGIN OPTIONS (LOGIN SCREEN): AS ^{1}ADMINISTRATOR, OR AS ^{2}CUSTOMER. THEN BASED ON THE SELECTED USER TYPE, THE SYSTEM MUST DISPLAY THE OPTIONS RELATED TO THE LOGGED-IN USER ONLY.

THE OPTIONS MENU FOR BOTH CUSTOMER AND ADMINISTOR MUST PROVIDE LOGOUT OPTION TO RETURN TO THE LOGIN SCREEN TO ENABLE YOU TO LOG IN WITH DIFFERENT USER TYPE.

In details, below are the involved steps for each use case:

Use case 1: Add new item for a specific provider

- The system asks the administrator to enter all information of the item to be added.
- The system asks the administrator to enter the provider name.

- The system adds the new item to the items list of the provider and updates the database accordingly.

Use case 2: Present new offer

- The system asks the administrator to enter provider name who wants to present the offer.
- The system displays all information of all items related to the provider.
- The system asks the administrator to enter all required information for the new offer (unique ID, item ID, quantity of items in the offer, and offer price).
- A new offer is added to the offer file in the database.

Use case 3: View all providers

- The system displays full information of all providers saved in the database in formatted manner.

Use case 4: View all customers

- The system displays full information of all customers saved in the database in formatted manner.

Use case 5: View all orders

- The system displays full information of all orders (with all status values) for all customers that are saved in the database in formatted manner.

Use case 6: View all offers

- The system displays full information of all offers for all providers that are saved in the database in formatted manner.

Use case 7: Deliver all paid orders

- The system checks all orders for all customers that are saved in the database and changes all orders with “**paid**” status to “**delivered**” and reflects these changes in the database.

Use case 8: Cancel offer

- The system displays full information of all offers of all providers that are saved in the database in formatted manner.
- The system asks administrator to enter the ID of the offer to be cancelled.
- The system removes the information of the offer from the database.

Use case 9: Place new order

- The system asks customer to enter his current location (Area 1, Area 2, or Area 3).
- The system asks customer to enter the category of the provider.

- The system displays full information of all matched providers.
- The system asks customer to enter **provider name**.
- The system displays full information of all items related to the provider in addition to all offers presented by the same provider.
- The system asks customer to enter IDs of the desired items besides the quantity of each item, also he can enter the IDs of the offers he wants to buy.
- Once the customer has finished selecting the **items/offers** that he wants to add to his order, the system calculates the **total cost** of the order (items cost+delivery cost).
- The system asks customer to enter a unique **ID** for the order.
- The system sets the **status** of the new order to “not paid”.
- The system sets the **customer name** of the order to the currently logged-in customer.
- The system adds the new order to the database.
- The system updates quantity field of all involved items in the items list of the involved provider and reflects these changes in the database.

Use case 10: View my orders

- The system displays full information of all orders (with all status values) related only to the currently logged-in customer saved in the database.

Use case 11: Pay for order

- The system displays full information of all not-paid orders (order status=”**not paid**”) related only to the currently logged-in customer saved in the database.
- The system asks customer to enter the ID of the order to be paid.
- The system changes the status of the order from “**not paid**” to “**paid**” and reflects this change in the database.
- The system updates the cash credit of the currently logged-in customer and reflects this change in the database.
- The system updates the income of the provider that serves the order and reflects this change in the database.

Use case 12: Post a review for a provider

- The system displays full information of all delivered orders (order status=”**delivered**”) related only to the currently logged-in customer saved in the database.
- The system asks customer to enter the ID of the order.
- The system asks customer to enter a review value for the provider that delivered the order.
- The system updates the review value of the provider and reflects this change in the database.

You are required to satisfy and implement every step above, else your grade will be affected accordingly. Please don’t assume anything even if it makes sense. In case you do not understand something, please feel free to ask about it.