CSE 757 AU04, Assignment 3

Due Oct 18 at 9:30am

Note: These are simple problems. Do not overanalyze them.

Problem 1 [2pts]

Consider a company that sells some product over the phone; for simplicity, assume that this is the only product sold by the company. A customer calls and talks with a customer service representative (CSR) to place an order for certain quantity of the product. The CSR uses a software system to place the order.

Consider the PlaceOrder use case. The main success scenario looks like this (oversimplified on purpose)

1. CSR starts a new order
2. CSR enters quantity
3. System presents price to the CSR
4. CSR enters credit card info
5. System obtains verification of credit card info from an external credit authorization system. Assume that the credit card is not charged - this will happen in the future when the items are shipped.
6. System presents a summary of the order to the CSR

Show the system sequence diagram (SSD) for this scenario. Make sure you represent the information that is exchanged between the system and the actors.

Problem 2 [2pts]

Consider a software system for a video store, and a RentVideos use case for that system. Assume we have cash-only transactions, and the store is stand-alone, without any communications with external accounting, inventory, etc. Each renter has a membership and some unique corresponding membership id. No rentals are allowed if a renter has unpaid late fees. The main success scenario for RentVideos may look like this:

1. Customer arrives with several videos
2. Cashier enters membership id
3. System presents late fee status
4. Cashier enters video id
5. System presents due date, rental price, video info, running total. Repeat 4-5 until done.
6. Cashier enters payment amount
7. System presents change and receipt
8. System logs rental transaction and updates availability info

The running total includes taxes. The system maintains info about which videos are currently available in the store (similar to an inventory in a retail store), and updates this info in step 8.

Think how this success scenario can “go wrong” and write the corresponding alternative scenarios. When writing down these scenarios, use the format described in class (see the lecture notes). Define at least three alternative scenarios.
**Problem 3** [3pts]

1. Consider a software system for storing information about customers and service providers. A customer can use many providers at the same time. There can be many customers that use the same service provider. Show a domain model for this problem.

2. Consider the system from above, and suppose that if a customer decides to use a provider, she signs a service contract at some *level of service* (e.g. “basic”, “plus”, “premium”, and many others). For any customer X and any provider Y, at any point of time there can be at most one contract between X and Y. Show a domain model for this problem.

**Problem 4** [3pt]

Consider the video store system described in Problem 2.

1. Use case **AddCustomer** for the system is as follows:

   Cashier enters the name of a new customer (for simplicity, we do not consider customer address, phone, etc.). The system assigns a unique member ID to the customer and records the information about the new customer.

   For simplicity, consider only the main success scenario, and ignore possible alternative scenarios. Assume that in the future we will not need to know which cashier entered which customer into the system.

   A. Which concepts are relevant for this use case? Show the corresponding conceptual classes using UML, and the relevant attributes (with their types) inside each class.

   B. Which associations between the classes from A. are relevant for this use case (e.g., they are examined, created, or destroyed during the use case)? Show them in the class diagram, with the appropriate names and multiplicities.

2. Suppose the store rents out only VHS tapes, DVDs, and games. The domain model has a class **RentableItem** with subclasses Tape, DVD, and Game. Is **RentableItem** an abstract class? Why?

3. For simplicity, suppose the store rents out only DVDs. Each physical DVD has a unique integer ID, different from the IDs of all other DVDs in the store. Each physical DVD contains exactly one movie. There could be multiple DVDs (with different IDs) that contain the same movie. Each movie has a title (e.g., “Blade Runner”), year of release (e.g., 1982), and a rating (e.g., “R”). Show the relevant part of the domain model (using UML), with information about conceptual classes, attributes, and associations, including types of attributes and names/multiplicities of associations.