**Unified Modeling Languages (UML)**

**Description**

You are contracting with a client who runs a small hotel chain. This client wants you to design a piece of software to handle the day-to-day operations of the business. You have asked for information about how the business is run and receive this description.

We own several hotels and need to keep track of pricing, availability, and reservations in each building. Customers may call ahead to place reservations; they may also simply walk in to get a room (which is pretty much like making a reservation that starts immediately). The customer’s reservation will specify how long the room is reserved. Once the customer finishes the stay, we charge payment for the room. We need to make sure that appointments never overlap, of course.

We also provide services during the customer’s stay: pay-per-view movies, delivery service, and so on. When we make a bill for the customer, we must keep track of the cost of the stay, the duration of the stay (as customers may leave before or after their reservation is complete), and any services that the customer consumed. Different hotels offer different services, so we need to keep track of the offerings at each hotel. Note that prices may change over time, so we need some way of keeping track of what the customer paid during their stay regardless of what prices are now.

We also charge for overstaying a reservation. If a customer is still staying in a room past the end of a reservation, we charge the regular rate for the room plus a ten percent fee. Customers are allowed to extend their reservations, though, so this fee only applies if the customer overstays the reservation without asking us to extend it. In rare cases, we will also charge fees for missing or damaged items – some customers seem to have a fondness for our towels – and we bill these fees in the same way that we bill services.

We also have a customer rewards program to encourage customer loyalty. Customers may have a rewards account which allows them to accumulate rewards points. One point is earned for each dollar the customer spends. Customers may pay their bills using either points or money but not both. Rooms and services have prices in terms of points as well as in terms of money. Points do not last forever; they expire two years after they are earned if they have not been spent. This means that we need to know not only how many points each customer has but also when they were earned. Remember that different amounts of points can be earned at different times.

Ignoring the user interface for the moment, you’ll be designing a piece of software which can accommodate the above requirements.

***Question 1: Domain model***

**(5 points)** Draw a UML class diagram which illustrates a domain model for the above application. Make sure to include appropriate relationships on the diagram, notation indicating the multiplicity of these relationships (“1..\*”, etc.), and a few methods and attributes for key entities.

***Question 2: Use Case Model***

**(5 points)**Draw a Use case diagram which illustrates the use cases and the actors for the above application.

***Question 3: Sequence Model***

**(5 points)**Draw a sequence diagram which illustrates the sequence of messages for the above application.

***Question 4: Reservation states***

**(5 points)** Draw a UML state diagram which indicates the states that a reservation can be in. You will need to infer the appropriate states from the description provided to you by the client; examples include scheduled, missed, overdue, and complete. Use “event[guard]/action” syntax where appropriate to indicate what causes state transitions in your diagram: the event describes what causes the state transition, the guard describes the condition that must hold, and the action describes some step which occurs as a result of the transition. You may assume that reservations cannot be reopened.

***Question 5: Reservation activity***

**(4 points)** Draw a UML activity diagram that shows the process of a customer creating and keeping (or missing) a reservation. Remember to include cases for ordering services, canceling the reservation, and so forth. As above, assume that reservations are not reopened even if they are canceled.