Sample Problems and Solutions for Modeling Quiz: Use Case Diagrams

CS 2720

Sample Problems

1. Answer the following questions regarding the diagram (and underlying model) given below.

(a) What operations (use cases) can a center manager perform?
(b) Describe the relationship between paying a bill and viewing payment history. Give the UML term for the relationship and an English description of the relationship’s meaning in the model.
(c) Describe the relationship between viewing payment history, viewing progress reports, and selecting a date. Give the UML term for the relationship and an English description of the relationship’s meaning in the model.
(d) Do you agree with having PayPal Web Service be an actor? Why or why not?
2. Identify two “diagram smells” in the diagram below and state why they are considered “smells”.

3. Create a use case diagram for the model summarized below.

Create a use case diagram for a simplified GitLab application. In this simplified version, users are classified as either guests, developers, or administrators. Guests can view the project repository and issues. A guest can directly view the issues, or they can view the issues when viewing the repository.

Developers can do everything a guest can do, as well as modify the repository and add issues. A comment must be entered whenever the repository is modified or an issue is added.

Administrators can do everything a developer can do, as well as edit user information.
1. Answer the following questions regarding the diagram (and underlying model) given below.

(a) A center manager can add customers and edit customers. Because a center manager is also a customer, a center manager can also pay bills, view payment history, and view progress reports.

(b) “View Payment History” is an extension (the “extend” relationship) of “Pay Bill”. This means that while a user is paying a bill, they may also move to viewing the payment history. Both paying bills and viewing payment histories may occur independently as well.

(c) “View Payment History” and “View Progress Report” both include “Select Date”. This means the steps defined in “Select Date” are used both when viewing payment history and when viewing progress reports. Selecting a date by itself is not important for a user, but doing so helps viewing history or progress reports.

(d) As with most of my “agree or disagree” questions, you could make a case either way. What I would be looking for in a question like this is that you can give valid criteria for making the decision – in this case, criteria for when to include something as an actor. I give two versions of answers below.

   **Agree:** I agree with having PayPal as an actor. One rule of thumb we discussed in class was to have an actor be anyone that initiates an interaction with the system. A customer might send payment information to PayPal, and then PayPal might automatically pay the customer’s bill every month, without prompting from the customer. If this is the case, then PayPal should be an actor.

   **Disagree:** I do not think PayPal Web Service should be an actor. We mentioned in class that an actor should initiate an interaction with the system independently, not just be something used to satisfy an interaction or use from another user. In this case, the Customer is the one that will pay the bill. PayPal may only be used as a way for the customer to send money while completing one of the pay bill use cases. Since PayPal does not initiate the interaction without a customer, it should not be an actor.
2. Identify two “diagram smells” in the diagram below. Below are some of the possible smells you may have identified.

- Login should not be a use case. Use cases should represent complete interactions – things that a user might do exclusively. It is very doubtful a user would ever with to just log in.

- The Authentication System should not be an actor. Actors should initiate interactions with the system independently. In this case, I’m guessing the system checks with the authentication system each time a member tries to log in, meaning it is not an external actor but rather an entity used to complete the use case.

- The “Select Date” use case (almost surely) has the arrow in the wrong direction. It is likely that viewing announcements may involve the member selecting the date range, not the other way around. As given, the diagram is saying that selecting a date includes viewing announcements, which does not make sense.

- The “Select Date” use case should not be an included use case as it has only one related use case. An included use case is meant to eliminate redundancy by giving a single spot to define behavior done by multiple use cases. If selecting a date is only done by one use case, there is no need for the include relationship.

- The “Edit Roster” use case is not part of an include relationship and is not connected to an actor. An extends relationship is meant to eliminate redundancy by showing how a user can move to another separate use case while participating in another. In this case, nobody can ever edit the roster by itself, they must first view the roster. This means editing a roster is really just a part of viewing a roster.

3. Create a use case diagram for the model summarized below.