Assignment 4

Online Discussion Forums UML

Posted Fri, Mar 29. Due Wed, Apr 10 by 11 PM IN SAKAI (NOT Bitbucket).

Worth 35 points (3.5% of course grade.)

In this assignment you and your partner will develop an object-oriented design and draw the UML, for the discussion forums hosted on a website, as described below.

Online Discussion Forums

There is a website that hosts discussion forums. A person first needs to be a registered user on the website before they can participate in one or more forums. A user does not necessarily have to be a member of any forum.

- The website hosts several forums.
- Each forum has one or more members.
- Members of a forum can:
  1. post messages to the forum
  2. read messages posted by other members to the forum, filtered and/or sorted by message author, thread, or date
  3. leave a forum
- Each message is part of a discussion thread. It can be the initiating message for a new thread, or a response to a previous message in a thread
- Users on the website can:
  1. create new forums
  2. join an existing forum as member
- Every forum has a single owner, who is initially the creator of the forum. The creator of a forum automatically becomes its first member.
- Each forum has one moderator, who is initially the creator of the forum. Only a forum member can be its moderator.
- Owners can:
  1. invite users to join their forum by sending them an email invitation
  2. invite a member to be the moderator - if the invitation is accepted, the new moderator replaces the current moderator
  3. invite a member to take over ownership - if the invitation is accepted, the ownership is transferred to that member
  4. delete the forum
- Moderators can:
  1. deny or pass through messages posted to the forum by members
  2. resign as moderator by sending an email to the owner, at which point the owner becomes the moderator

UML

Choose the most appropriate representation for each entity in the model. (Entity here refers to class or interface.) Use interfaces and abstract classes as needed. In each class, list the key attributes and methods you determine are needed to fulfill the specification. Not all details are explicitly mentioned in the specification above, but must be filled in by inference. Such details include choice of attributes and their data types, choice of operations with parameters and return values, and choice of access levels (public/protected/package/private) for attributes and operations.

Pay special attention to the relationships between entities. Remember, super-sub classes, and interface implementations are the strongest connections between entities, followed by associations, followed by dependencies. Make sure you are as specific as you can be with the multiplicities in associations, and whether an association is a plain old link or an aggregation or a composition, whether it is bidirectional, unidirectional or reflexive. Note: Every association is by definition bidirectional unless you explicitly show unidirectionality. Make sure you show multiplicities for all associations.

Supplement to the UML

For each association, list the data structure(s) (full type definition in Java) that will be used to implement the data that arises out of the association, and name the class(es) to which it/they will belong. (Refer to Slide 7--Relationship data--of the lecture notes in mar12-uml-3.pdf for an example.) You are not required to write the class outline. Just draw a table with two columns, one for each data structure type definition, and the other for the class in which it will be defined. If the class is not either of the end points of the association, but a new class altogether, then make this fact explicit in the table.

Grading

Your UML and supplement will be graded on completeness, appropriate choices in class/interface/abstract class for entities, in relationships between entities, and in the data structures used to implement association-specific data. "Appropriateness" speaks to the best applicable/most precise design choices based on all the object-oriented design issues we have covered in class, flexibility (how easy it is to use your design for a range of functionality), extensibility (how easily your design can be extended to add more capabilities), maintainability (how easy it is to fix problems that may be arise in implementation). Not all of these may apply with equal importance (every model is different) but you want to use this as a checklist against your design to spot, and correct, obvious flaws.

Submission Instructions

Submit a single PDF file, called `forums_uml.pdf`, to Sakai. (Whatever software you use to draw the diagrams/write text, you MUST convert to PDF.) The PDF document should have the UML as well as the supplementary table of data structures. If you have trouble fitting the UML in one page, draw the UML with only the class names and relationships between them on one page, and then on other pages, draw the details for each entity. Hand-drawn/hand-written submissions will NOT be accepted.

Submit ONE PDF for your team, but write the names of both members in the document.