

CSCI 5333.3 DBMS HW #1 Grading notes

Some issues and observations:

- Missing multiplicity.
- Incorrect multiplicity.
- Do not have too many use cases in a top level use case diagram.
- Many actors can share the same use cases, such as write comments, read survey results etc. (They may have different privileges which may not need to be captured in a top level diagram.)
- Document at least your actors, use cases and classes.
- Not all functions need to be captured in a top level use case diagram, only major ones. The focus should be on data, not operations. Operations are not as important as data in this phase of data model.
- Use cases are actions (sentences). Examples of inappropriate use case names:
 - User voting system
 - Display of voting results and choices
 - Administrator system
 - Administrator info
 - New Administrator
 - User information
 - Survey question
 - Database
 - Other utilities
 - Search, vote, comment, read, Ranking(too generic)
- When you model implementation classes (such as Java or C#), the focus is on the operations (as most data members should be private and hidden. On the other hand, when you perform data modeling for databases, the focus is on the data members required. Operations are still useful, but they are used to support data requirement.
- Many modeled implementation classes: classes that use database access to get data to implement certain applications (such as registration, search, display, view, graphics, etc.) They are needed in the later phases of modeling. Focusing too much on them obscure data requirements.

Examples of problematic modeling:

- No class for Question. No class for vote.
- Missing associations, eg. Between member and vote, between member and comments.
- Include classes that are not needed, e.g. search, Output Page.
- Missing attributes 'isOpen', 'Timestamp'