DASC 5333 Database Systems for Data Science
CSCI 4333 Design of Database Systems
Spring 2024
Homework #3 Database Modeling

UML Modeling

Download and install the *student* version of Astah's UML Editor: http://astah.net/download. Astah provides it free for students. This is one of the best free UML editors available. It has some restrictions but should be more than sufficient for our class. Another good alternative is to use Visual Paradigm community version. This is free but will leave a watermark when you export your model diagram to images or PDF. Instead, you may just take a screenshot of the model.

The homework assignment is to model a drastically simplified online team project management system named TeamRabbit (TR)

Multiplicities for attributes and associations should be as specific as possible. You do not need to explicitly specify the attribute multiplicity of 1..1, which is assumed to be the default. Attribute and association documentation are optional, but they are encouraged when not trivial. For example, the roles or names of associations may be displayed when appropriate. Operations for classes are not necessary. You may use the stereotypes <<PK>> and <<unique>> for attributes when appropriate. Create suitable data types for SQL if needed, e.g., SQL::Date, SQL::Time, SQL::DateTime.

Save your files as <<last-name>>_<<student_id>>_h3.asta (Astah's file format) and <<last-name>>_<<student_id>>_h3_documentation.docx (optional; for additional information if needed). Examples: bajaj_0007007_h3.asta. Include a PDF versions of your class diagram: <<last-name>>_<<student_id>>_h3.pdf, especially if you are not using Astah.

Submit your homework through Canvas, including .asta and/or .pdf files. If you are not using Astah, you must submit a PDF file.

TeamRabbit (TR)

Construct a data model using an UML class diagram to support a part of an online team project application: TeamRabbit (TR). This is a drastically simplified system that does not support many functions. Make reasonable assumptions.

The basic idea is to support teams to define, store, access, label, and update projects and sub-projects. Access to the system may be made through multiple types of devices such as computers, smartphones, web services, etc.

Users must be registered as TR members to use TR. The system should store the last name, first name, a unique screen name, and an optional working email address for every member. The email address does not need to be unique. A unique member id is assigned by the system and the start time of the member should be recorded. A member may be referred by another member to join TR, and the referrer should be recorded.

Each member must have a username and a password to access TR. However, an account may be used by a person other than a TR member, such as a TR staff or a TR system administrator. The management and administration module of the system is not modeled in this assignment.

A member can create any number of teams. The creator of a team is known as the owner of the team. Each team may also have managers who have heightened privileges relative to other team members. However, only the owner can change the managers of a project. The owner can also be a manager. A project can have any number of members. The time when a TR member joins a team should be recorded.

A member may have many roles in a team. Some of these role types are predefined by TR, for example, librarian, developer, team leader, supervisor, consultant, etc. They are standard role types. Some roles are team-specific and can be defined by a manager of the team. For example, a software team, S1, may define the team roles: modeler, programmer, technical writer, etc. When a team-specific role is defined, an optional description and a creation time should be stored together with the role name. The manager who defined the role should also be noted. Every role, standard or user-defined (team-specific), has a role level. The role level is an integer with a definition. Many roles can have the same role level.

A team has a unique id. The team's name should be stored with the team creation time and an optional description. A team has any number of projects. A project has a name and a description. A project may have any number of sub-projects. There is no limit on the number of levels of sub-projects. A project can be created by any team manager. The creator of a project should be noted. A project may have an expected completion time and an actual completion time. A project may be assigned to a team member. The creator and the assignee may or may not be the same person.

A project has a status. TR keeps track of status changes. A status has a name and a description. For example, a status history for a project may be:

1/17/2024 13:11:05: 'started'
1/18/2024 12:23:15: 'assigned'
1/22/2024 18:24:15: 'first draft completed'
1/23/2024 17:25:12: 'routed to audit'
2/10/2024 11:11:28: 'completed'