6.170 Software Studio
Fall 2014
TeamMotivate: Problem Analysis

Motivation:
TeamMotivate is an online app with the intent of providing a user friendly, and simple interface for members of a large-scale project to monitor and track progress of the project as a whole. It allows for members to create projects, break them down into tasks, and assign them to other members.

Purpose:

Provide a means for different departments of a company to keep track of progress of a project. In the case of large scale projects, it is essential to be able communicate between departments to efficiently gauge and monitor the progress across disciplines.

Provide a easy way to see workload of the day/week. Often, when members are asked for availability for more tasks, they will not have a full gauge on their responsibilities for a certain timespan. TeamMotivate will provide a easy way to see if a member is available.

Context Diagram:
The scope of this project would be a large company setting, with multiple departments that need to communicate effectively to accomplish a task.

Concept:

Project: This is the key concept our application is based on. Our application runs on the existence and completion of large scale projects.

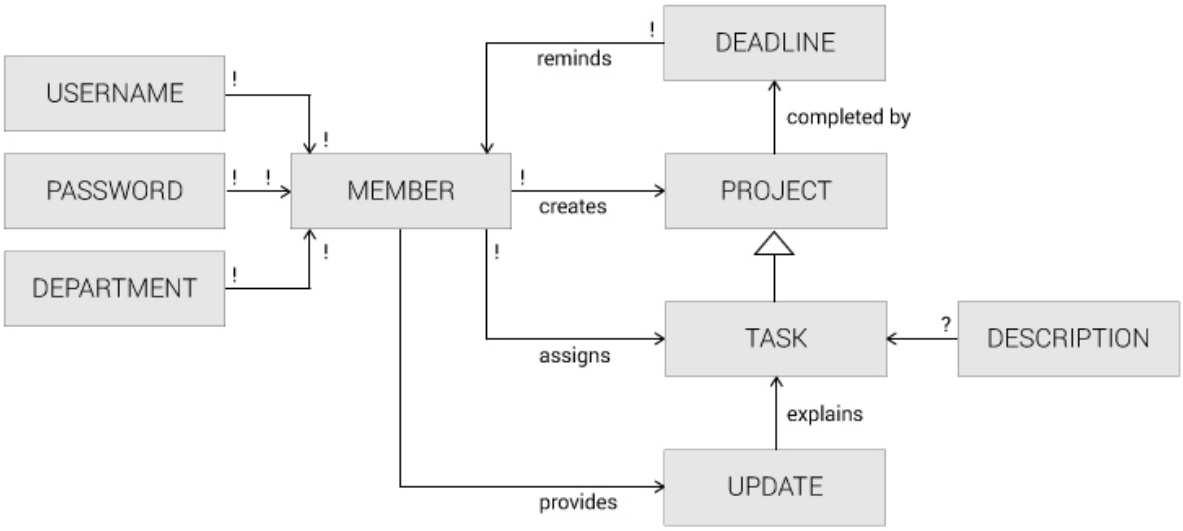
Tasks: Each project can be split up in to multiple tasks that may be assigned and kept track of individually.

Deadline: A deadline may simply be thought as an attribute of a project or task. However, in the case of large scale projects, deadline is a pivotal concept that

determines the success of a project. We utilize the idea of a strict deadline: if a project is not completed by this deadline, repercussion will be enforced.

Updates: As a member is completing his tasks, he should make comments and provide updates in a paragraph format to explain his progress. This will help others understand the issues encountered and provide feedback throughout the process. This will also let others know the effort and time put into completing a task.

Data Model:



Design Challenges

What if a member drops out of a project?
We remove him from the database, and we must make sure that the tasks do not disappear with him. The tasks must be reassigned to another member without breaking our application.

Potential Solutions:
One way to prevent this is when a member decides to drop out, the project manager reassigns the tasks to people with proper qualifications and are least busy before actually removing the member from the database.

Another less feasible way is to prevent a member from dropping out before completing all his assigned tasks. Doing this will conveniently result in not needing to change anything within our application, but there are probably legal reasons why we cannot implement this.

What happens if a project is complete?

Once a project is done, there are several ways we can proceed. One thought is to close the project and archive it. Do we want it to remain in our dashboard page or remove it entirely? What if there are modifications that we determine are needed to be made? Do we remark the project as incomplete and proceed appropriately?

Potential Solutions:
Once a project is complete, we archive it and provide access to a webpage that shows a list of completed projects. We can provide an edit ability to the project that will bring it back to the dashboard.

We can decide to close completed projects entirely, and if users decide that there are modifications to be made, they can create a new project with an appropriate title reflecting the changes.

Data Design Justification:
Most of the key points of my data design model have been explained in my concepts section.

This is subtle, but we included a department attribute in a member, which is a key focus for the problem we are addressing. The problem that large scale projects often have is the lack of communication between departments, so the attribute is here to set as a reminder of our purpose.