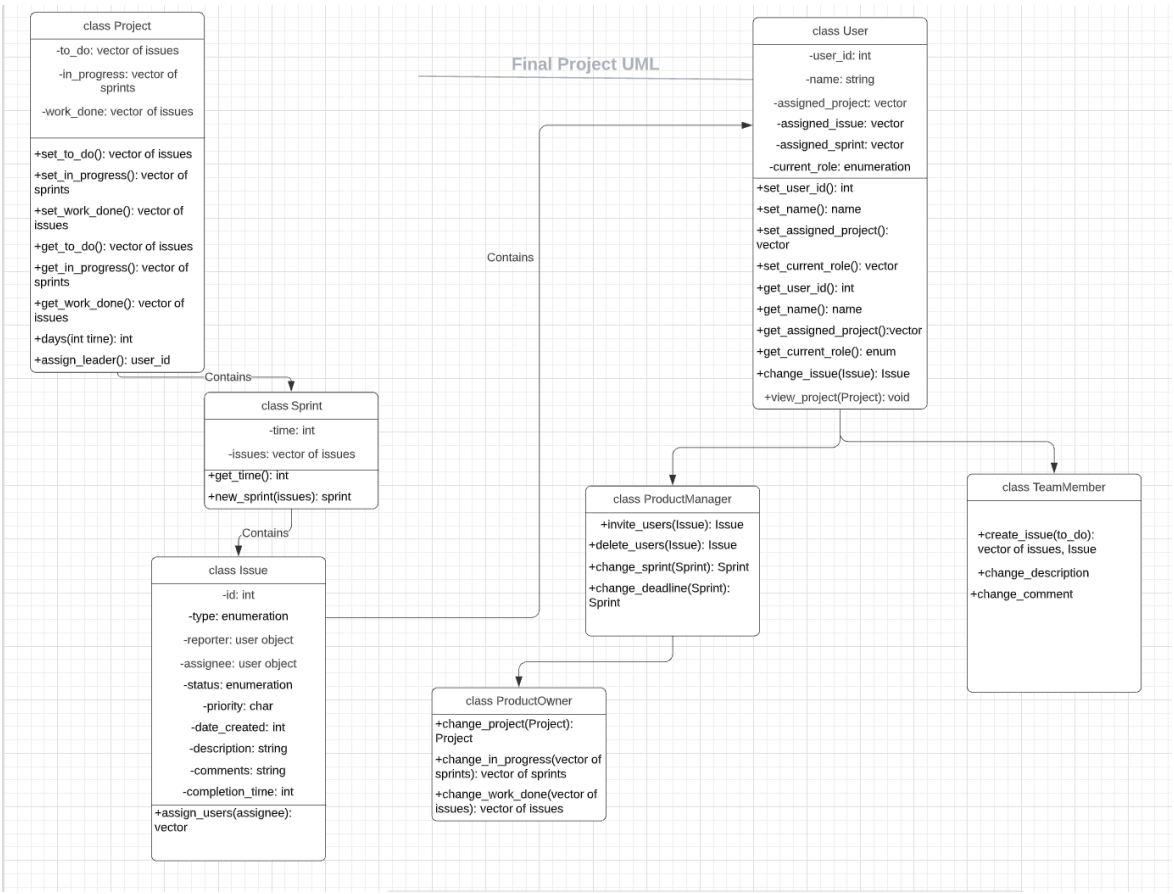
Final Project Report

By Emily Wang and Xinyu Wu

GitHub Repository:
https://github.ccs.neu.edu/thexinyu/cs3520-Fall-2021/tree/main/finalproject

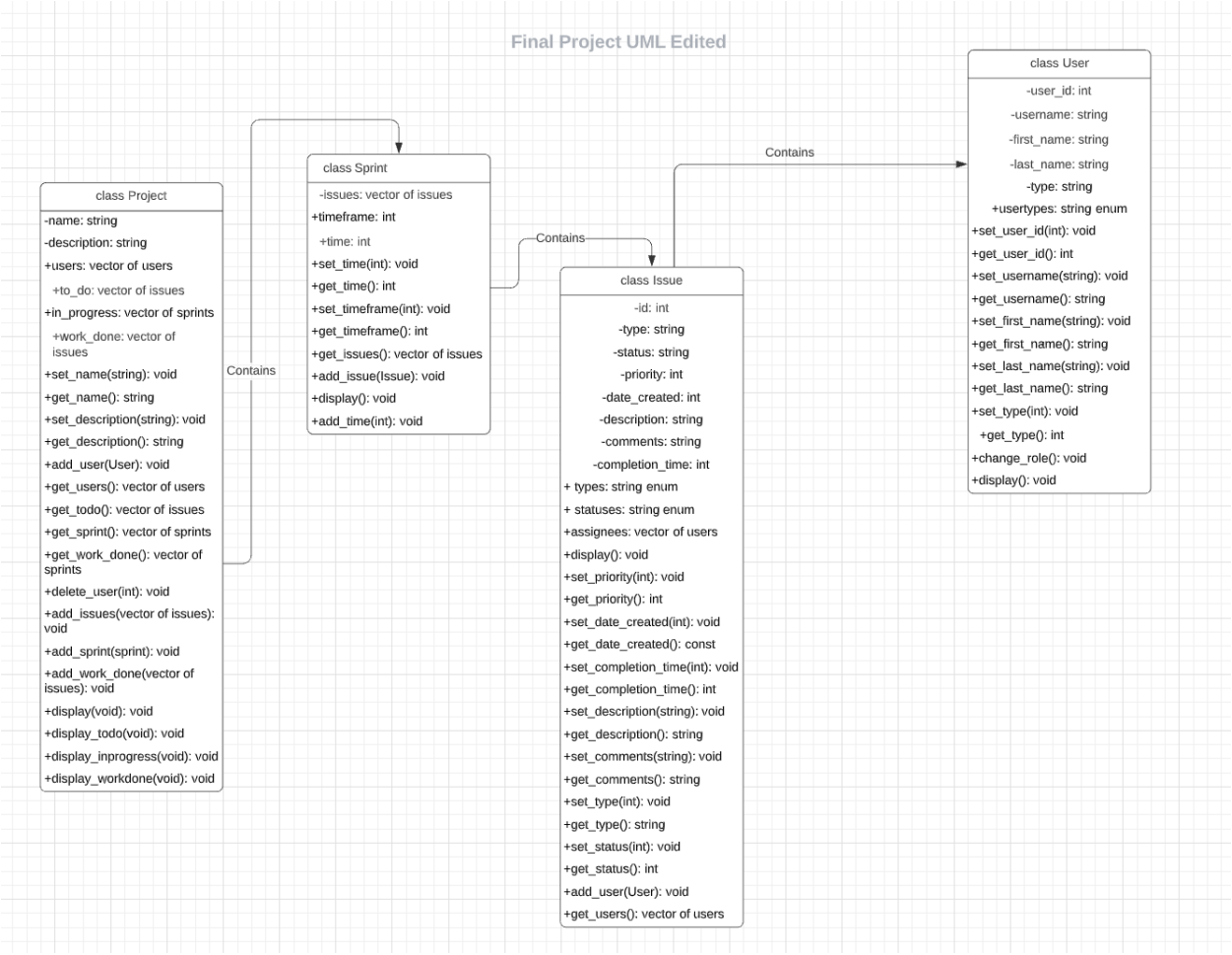
I. Initial Approach

When first approaching this project, we started by creating a basic UML diagram depicting the different classes we thought we would implement. This included: Project, Sprint, Issue, User, Lead (derived from User), and Member (derived from User). This turned out to be similar but not exactly the same as our final program. Before starting programming our application, we started with a base User class and two derived role classes (Lead and Member). This seemed logical since the roles have different functionalities and access to issues. At the beginning, we also decided to create three different vectors in the Project class, representing issues that are queued in the to-do section, issues as part of sprints in the 'in progress' section, and also issues that were finished. In terms of the Sprint class, we started with two data members to implement: timeframe and a vector of issues. This represented how long the sprint would last, and what issues were active in the sprint to be completed.



II. Design Changes

There were a good number of changes to our methodology as we went from start to finish. This included adding many more data members to each class (i.e. a time tracker called 'time' in the Sprint class) that allowed us to better simulate the program. A major change we made to our program's design is omitting the Lead and Member classes that were originally derived from the User class. We found that using enumerations for the Lead and Member roles made more sense since the functions that differentiate the two classes are not defined in the classes. Rather, they are defined in other parts of the program. Other changes in the program are primarily on how we called the program and how we represented the Sprint. We used an enumeration to change the status of issues as time was added to the Sprint simulations.



III. Program Testing

In order to test our program, we initialized 5 different users and 6 different issues along with 2 sprints. This allowed us to go through the program, make changes to all of

the different moving parts, and see whether our program ran as expected. These initializations are created in the initial.cpp file. Users can also test the program by creating their own users and implementing different functionalities depending on the role of their user.

IV. Discussion

From beginning to end, we ran into a good number of challenges including trouble with the derived User classes (we ended up making a last minute change and define user role as an enumeration), deciding where to assign users to issues, and initializing the program with data. While there are still some minor portions that we were unable to finish, this project represents our best work and was a steep learning curve. It allowed us to apply our knowledge from this past semester, and challenged us to become better coders, particularly in the C++ language.