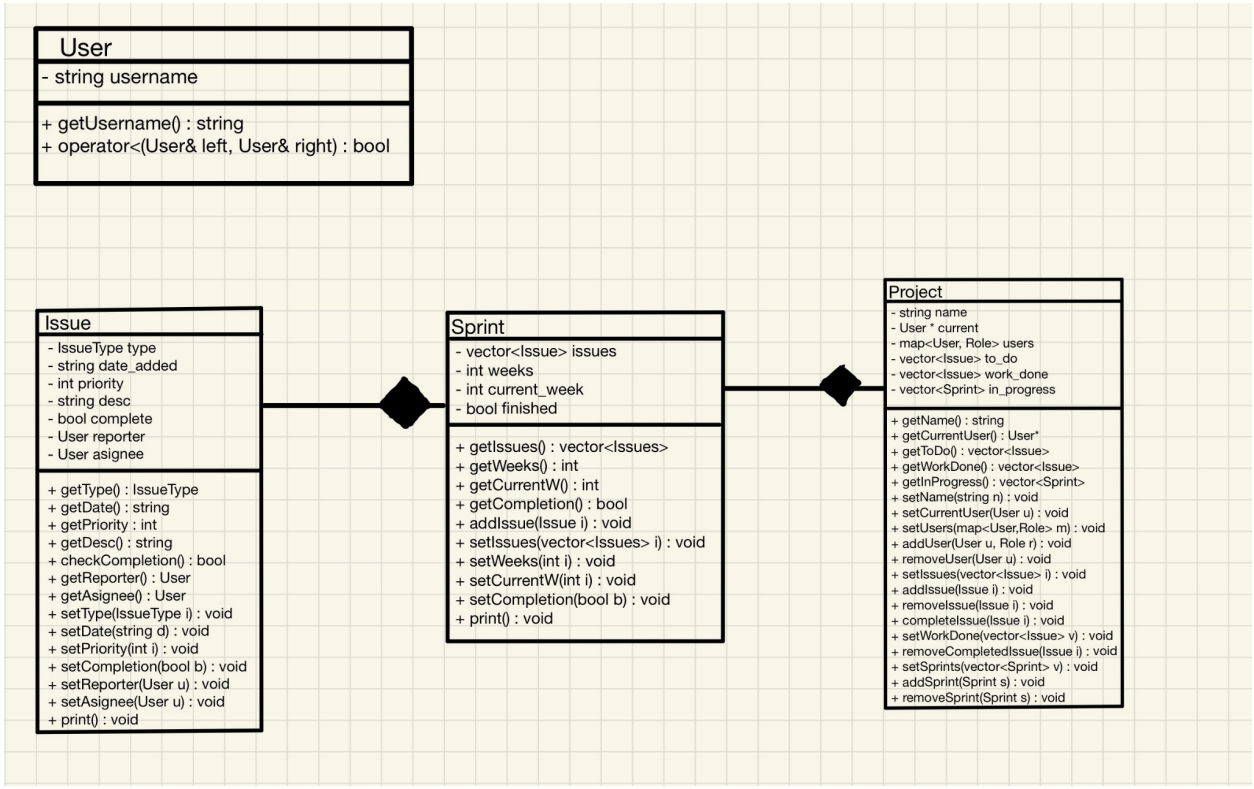
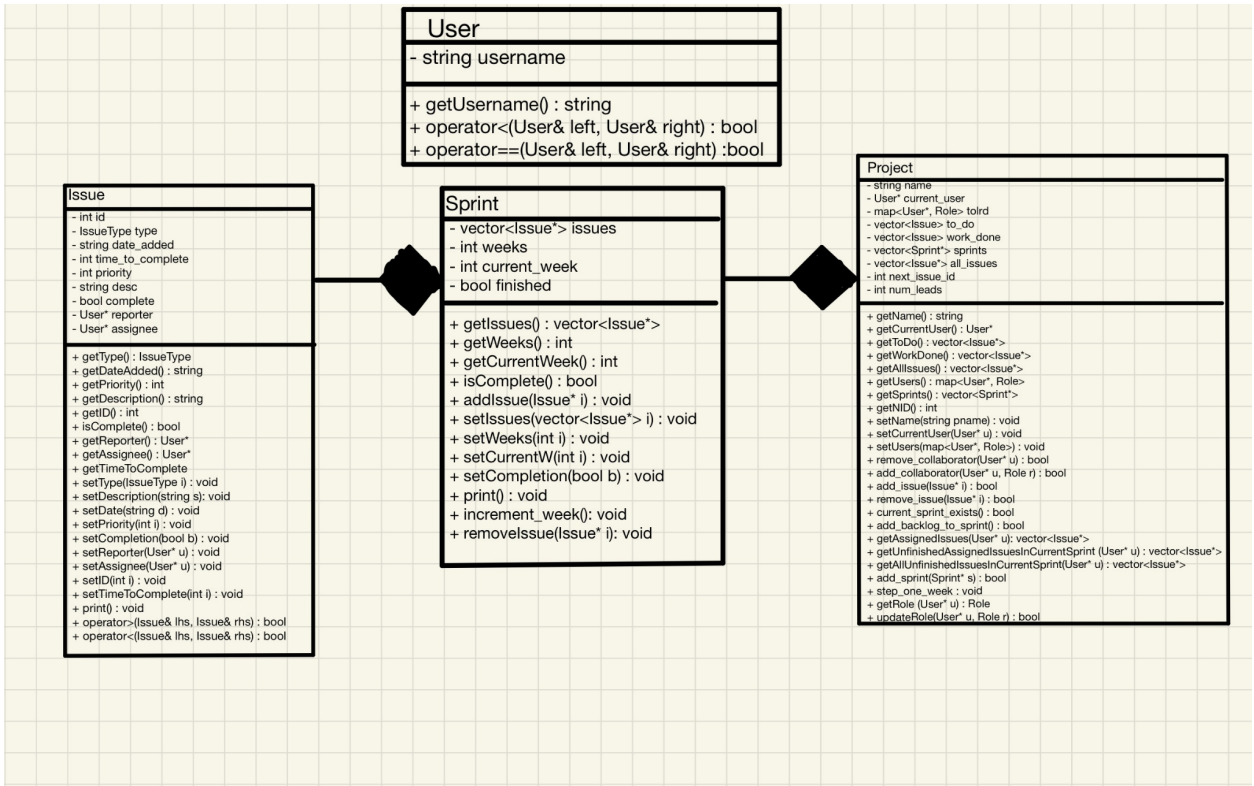
Initial UML



Final UML



Explanation

The biggest changes from our initial design came from the project and issue classes. Many of the changes in the issue class came after finding features we overlooked in the rubric and adapting the design to take in fields such as a unique ID and estimated time to complete. These fields came with their own getters and setters. We also realized that in order to sort the list of issues by priority, we would need to overload some operators to compare the priority values of two issues. In the User class we added a new operator to compare the equality of the usernames of two users. In the Project class we made many changes to accommodate for roadblocks we hit while creating the application. We created a new field to hold all issues in a project, as well as a field to track the last issue ID (for incrementing to keep IDs unique) and a field that keeps track of the amount of leads in a project. These fields were used throughout the application and we found that it was more convenient and saved time and lines of code by adding these into the project class. The bulk of the methods in the project class are getters and setters which allow the user to view or update the private data. A few specific getters were added to get issues assigned to Users depending on the sprint or whether the issue or sprint is marked as complete. The step method steps one week into the future, and updates all of the necessary information in the sprints to account for the week advancing. We also added a method that adds the backlog into the current sprint, allowing for the sprint to be populated with issues sorted by priority. Any issues that remain once the sprint is complete get moved into the next sprint.

The most challenging part of the project was definitely the initial design, which took about two hours of just discussion and drawing to finalize in a way that made sense. I believe that by tackling the design first and coming out with a fairly in-depth design of our project saved us hours of work and stress and allowed us to split the work easier since one of us could work on

methods that were still unimplemented while the other worked in main. The second largest hurdle was storing and importing data from files to populate the projects. This task gave me a deeper appreciation of database systems like MySQL which can take in more data and process it faster than we could. The system we used involves reading a file one line at a time and taking in delimiters that tell the code when to move on to the next field. This presented us with many unexpected errors which took hours to fully iron out but resulted in an application that can successfully save and load data. The final challenge was the actual scope of the project. Even with the design and data aspects complete after a couple hours of work, the sheer amount of code that had to be written and tested took more time than either of us expected. In total, the two of us worked close to twenty hours each on this project, the majority of which was spent on the GUI and helper functions.