FINAL REPORT by NICOLE ROSAS

In our initial design we have a Project class with properties and functions to it. The main change to it is that for any vectors, we have a vector of pointers, not vectors of classes. This is to make sure that memory is being allocated correctly and no copies have been made. We have included more functions to Project class as well, mainly to run a sprint and check if a username is already inside a project. We also add a new property to Project, which is a deadline represented by an int.

The two enumerations: StatusType and IssueType, have not changed and remained the same and are used for the Issue Class.

Our Issue class has the same properties as mentioned in the diagram and it has more functions included, such as gets and sets functions for each property and printing an issue function which prints out to the command line what the issue entails. There is one big change for issue, in the diagram I have an arrow from Issue class to Sprint class, and this is a mistake. It should be the opposite, as the Issue class is not a child of Sprint, they are two different things. However the sprint class is composed of the issue class which is why it should be an arrow from sprint class to issue class.

For our User class, we have made the project and vectors also be list of pointers to better allocate memory. Other than that it’s properties have not been touched but there are more functions for User. We have added a print function to print out their desired issue or sprint as well as project. It is the parent class of the two children classes: TeamMember and ProjectLead.

For both, in the diagram there is a limited set of functions and a property. However, when implementing it in CPP we have added more functions. This is because, after looking at the rubric I had figured which functions were specific to which child class. Such as modifying issues and projects is strictly for ProjectLead while the TeamMember can only add a comment to an issue or assign themselves an issue. Also there is no distinct string property for both TeamMember and ProjectLead

One major thing that has not been shown in the diagram is our way of saving and loading the project, users, issues, and sprints. When we save our objects that we have made, we will go through every user, every sprint, and every issue and call their output function and inside that func will open a specific file and write its properties inside to store it. The main() will also have to call the output function in order to load the saved objects. This was not in the diagram, because I had not thought of a way to save and load the objects for this project, but Michael has written the code to save and load objects.

We have not updated the diagram, so the first uml diagram is what we have been working with.