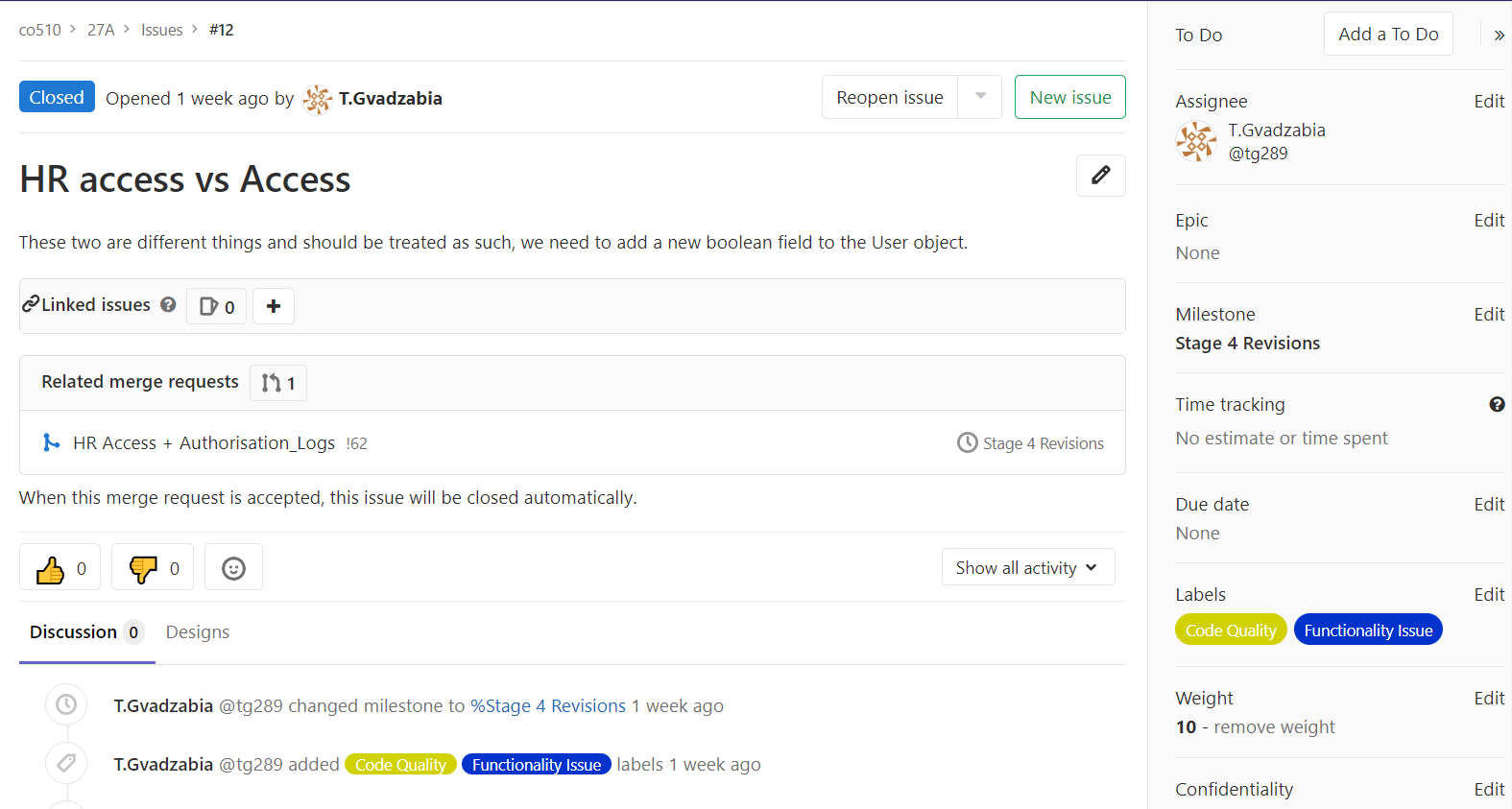
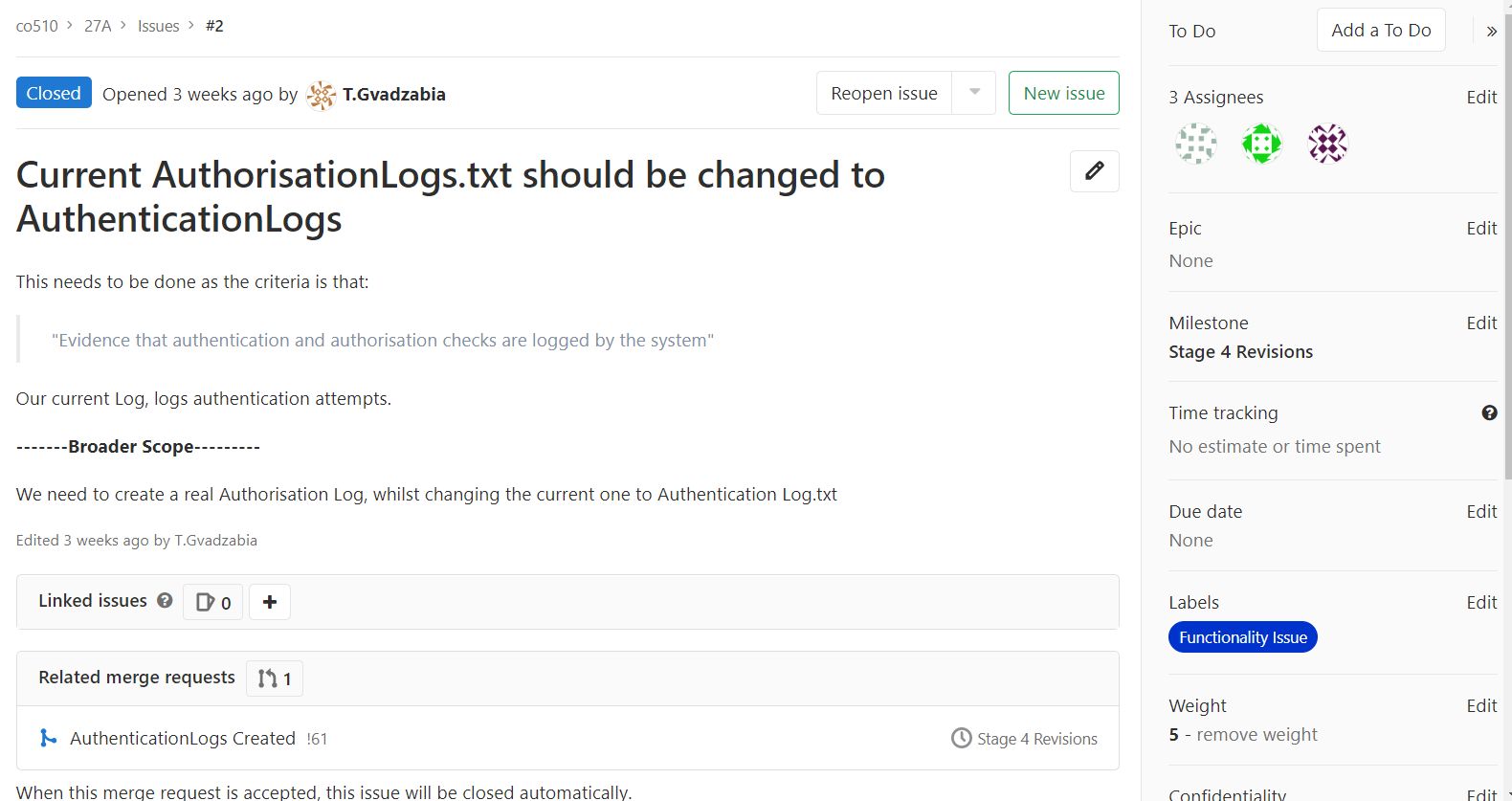
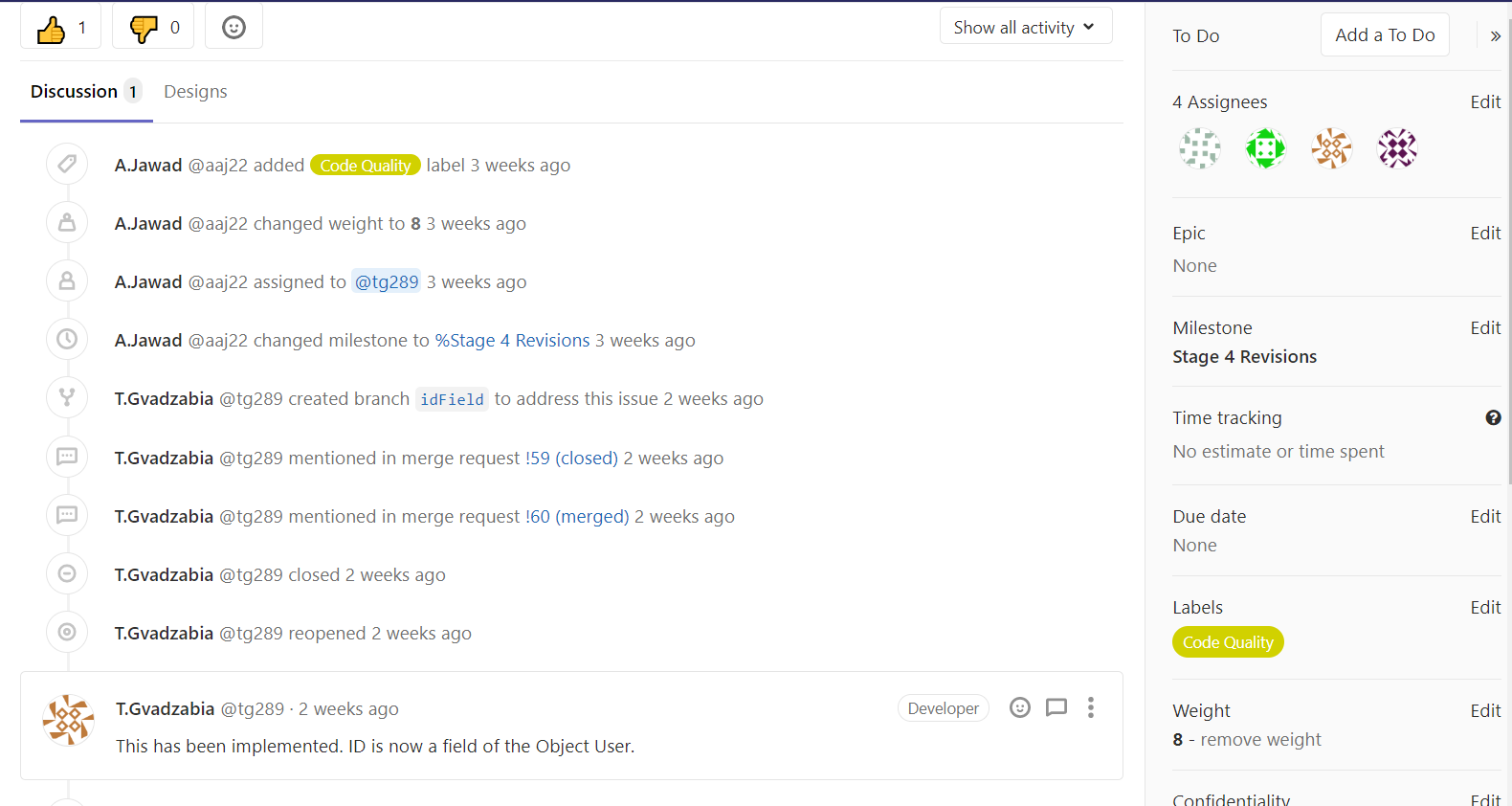
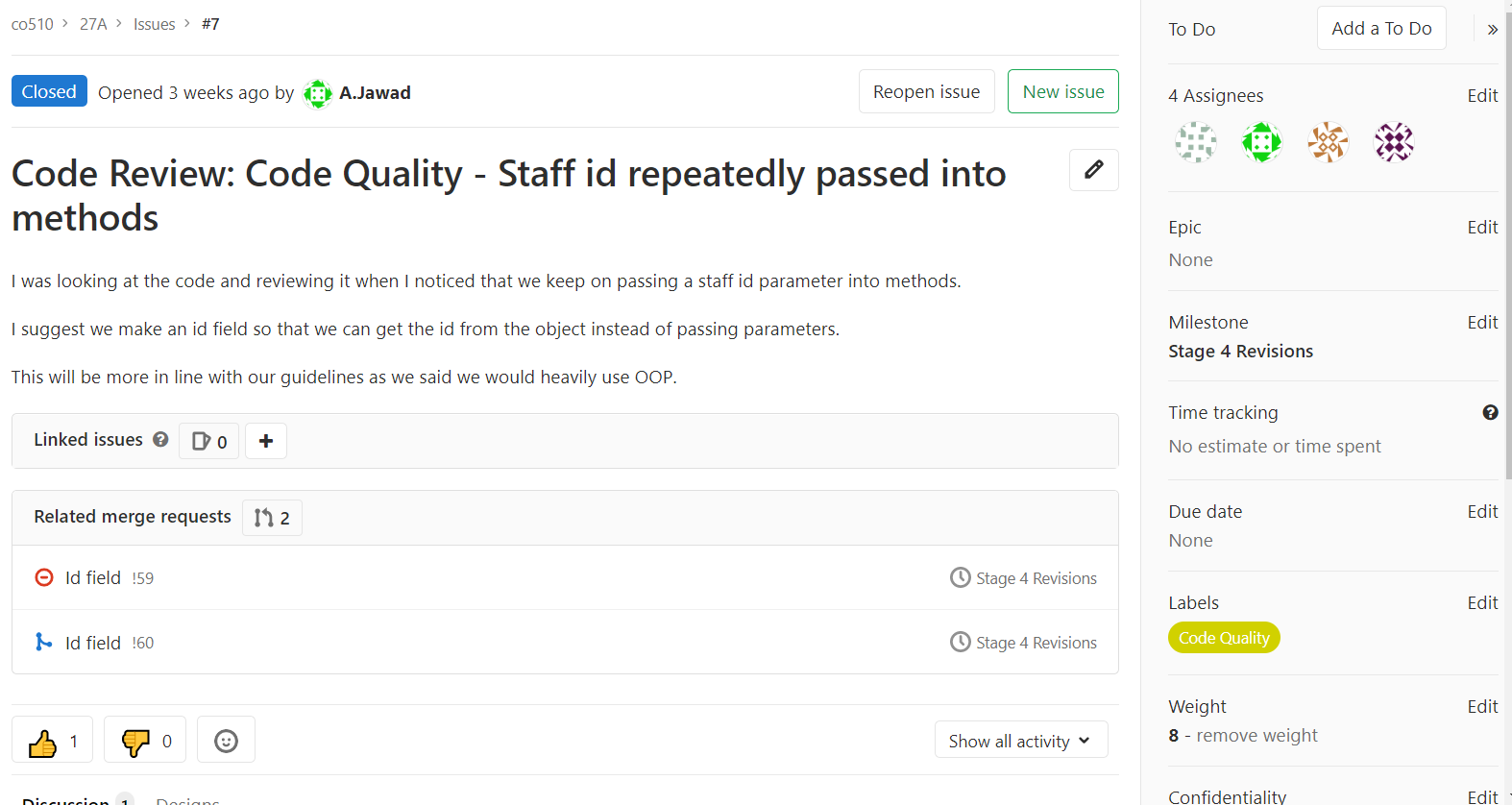
Revisions 

Stage 4 Revisions:

* We had to revise the HR access functionality so that they become two different entities instead of being the same element. We did this by adding a new Boolean field to the User object. In the screenshot below, we are logging an issue in the Stage 4 revisions milestone to track this and added a label of code quality and functionality issue to define what type of issue it is. We also used the weighting function to acknowledge the importance of the issue on the whole project. This issue received a rating of 10 as it was a very pressing issue which needed to be solved in order for the project to move ahead.
* We didn’t have an authorisation log for our Yuconz system application. We thought authentication would be enough. We later found out that authorisation is a requirement and started creating functionality for it.



We put a weight of 5 on this as it was an important feature that was required but it didn’t stop development in other areas of the application. This is why it was lower than the Access error. A functionality issue label was given as this was functionality that needed to be added for it to be solved.

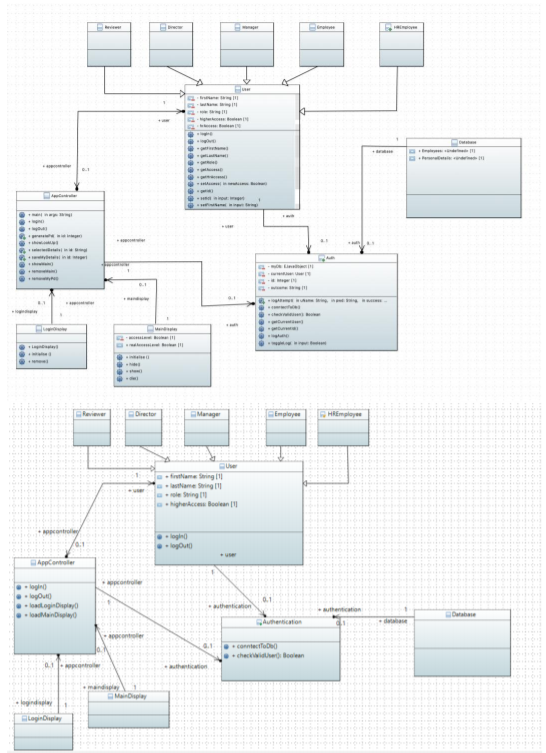
* After doing our code review, we found out a code quality issue for Stage 4. We logged an issue which stated that the staff id was being repeatedly passed into methods. There was a suggestion made that we could implement a field. This would be more in line with our developer guidelines as we explained how we would heavily use Object Orientated Programming.

This was then revised, and a field variable was implemented of the Object User. A weighting of 8 was given as it was a very big code quality issue and was going against our developer guidelines. The code quality label was given to mark it as a code quality issue.

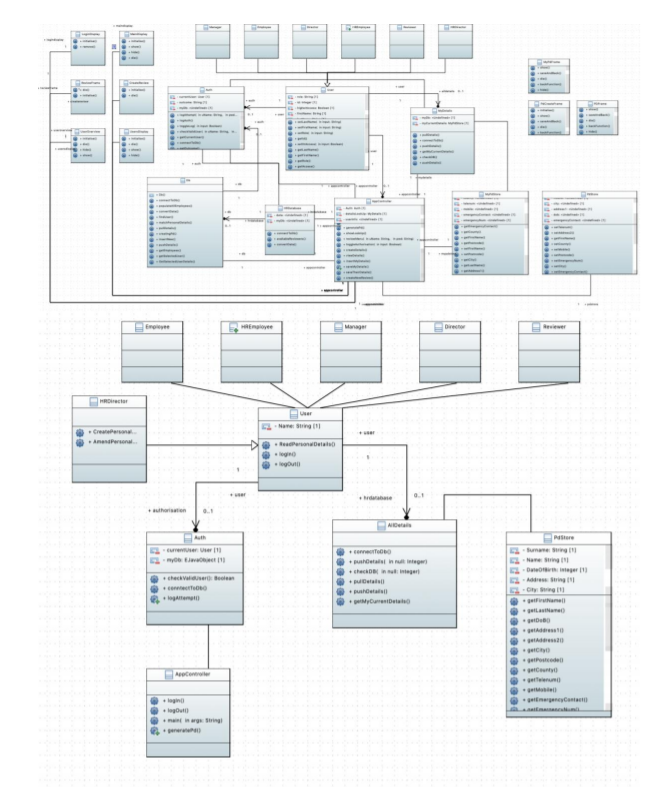
Stage 5 Revisions:

* We went through multiple revisions for our Papyrus UML models. After each revision, we expanded by creating system design revisions which followed the UML model requirements. We implemented TDD as we wrote some of the tests before implementing the code. This helped us test the behaviour of our revisions continuously to deliver robust code. Using Pipelines was a big part of being test driven as tests were continuously being run when pushing to master branch after each revision. This made our software development workflow very efficient and seamless.

The first revision is as below:

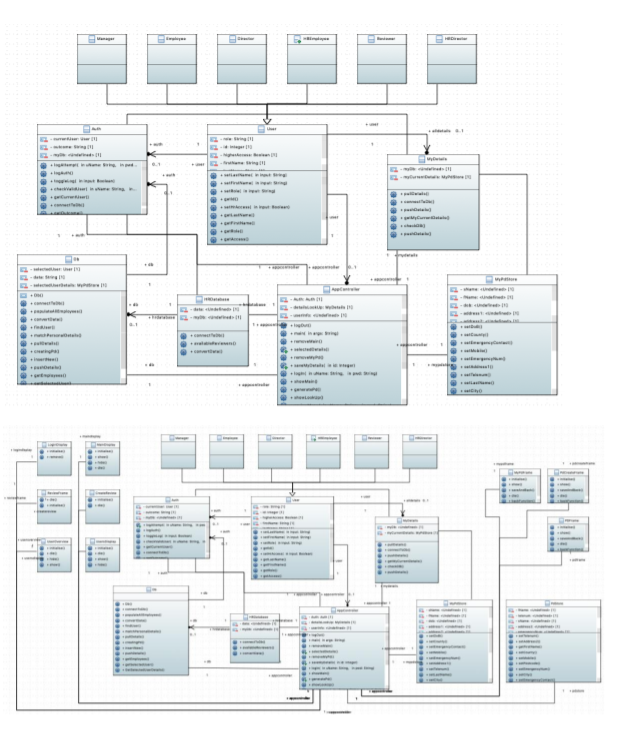


This included generalisation of the Yuconz system.

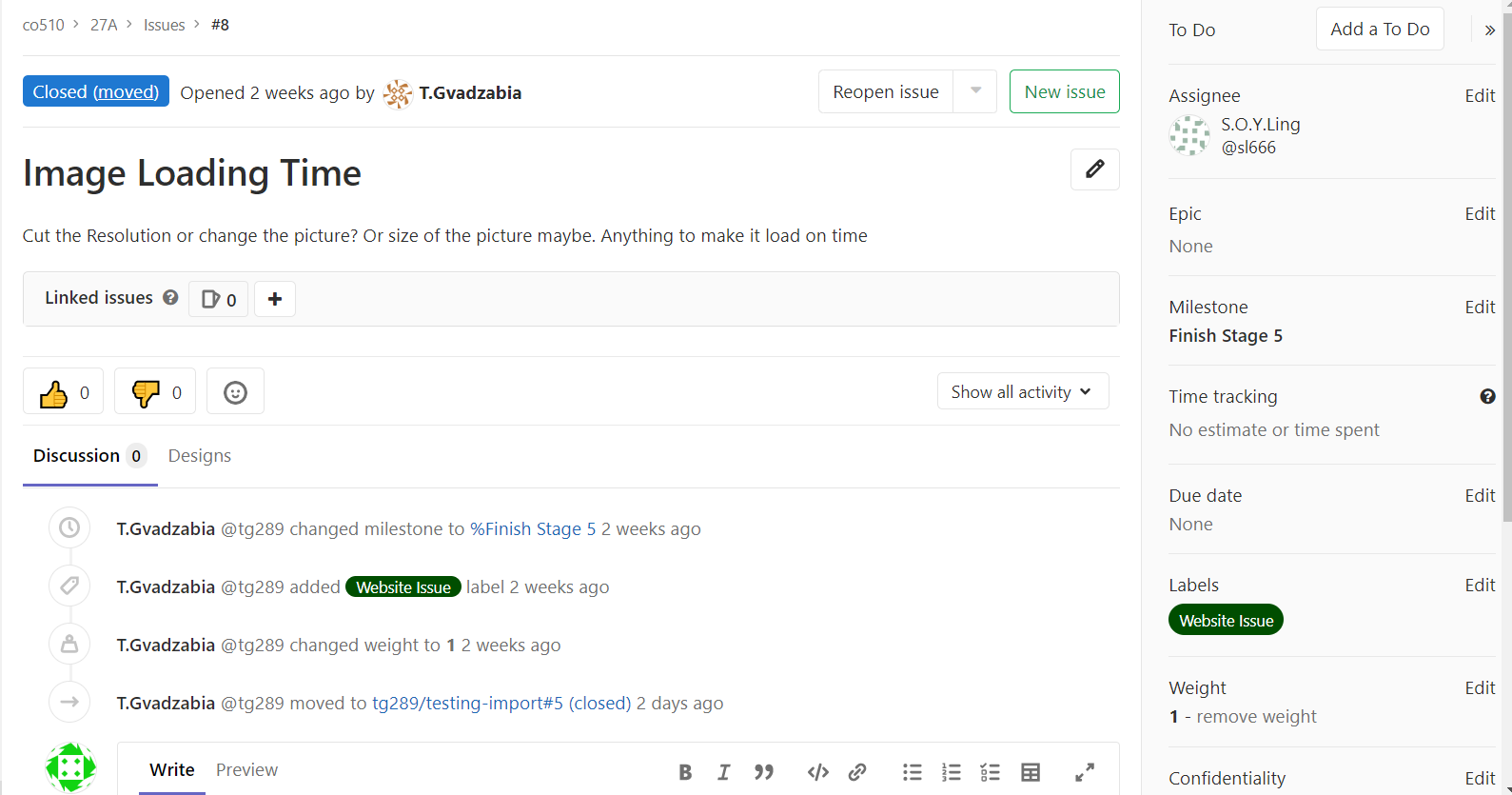
The second revision is as below:

Merged three separate diagrams into one big one and removed duplicate classes. This led us on a journey of refactoring our code so that it corresponds with the UML model.

The third revision is as below:



For this revision, we added new methods and GUI frames as classes. We also fixed some associations. These were then implemented in code as we needed these new elements to make a complete working application.

* We had very large image loading times when navigating through the website. We believed it was due to the file size of the picture we were using on the website. This was given a weighting of 1 as it could be fixed at any time and was a quick fix. There was a specified website issue given so we could track this.

To solve the issue, we cut the resolution of the image. This resulted in very fast loading times which gave users a good experience.