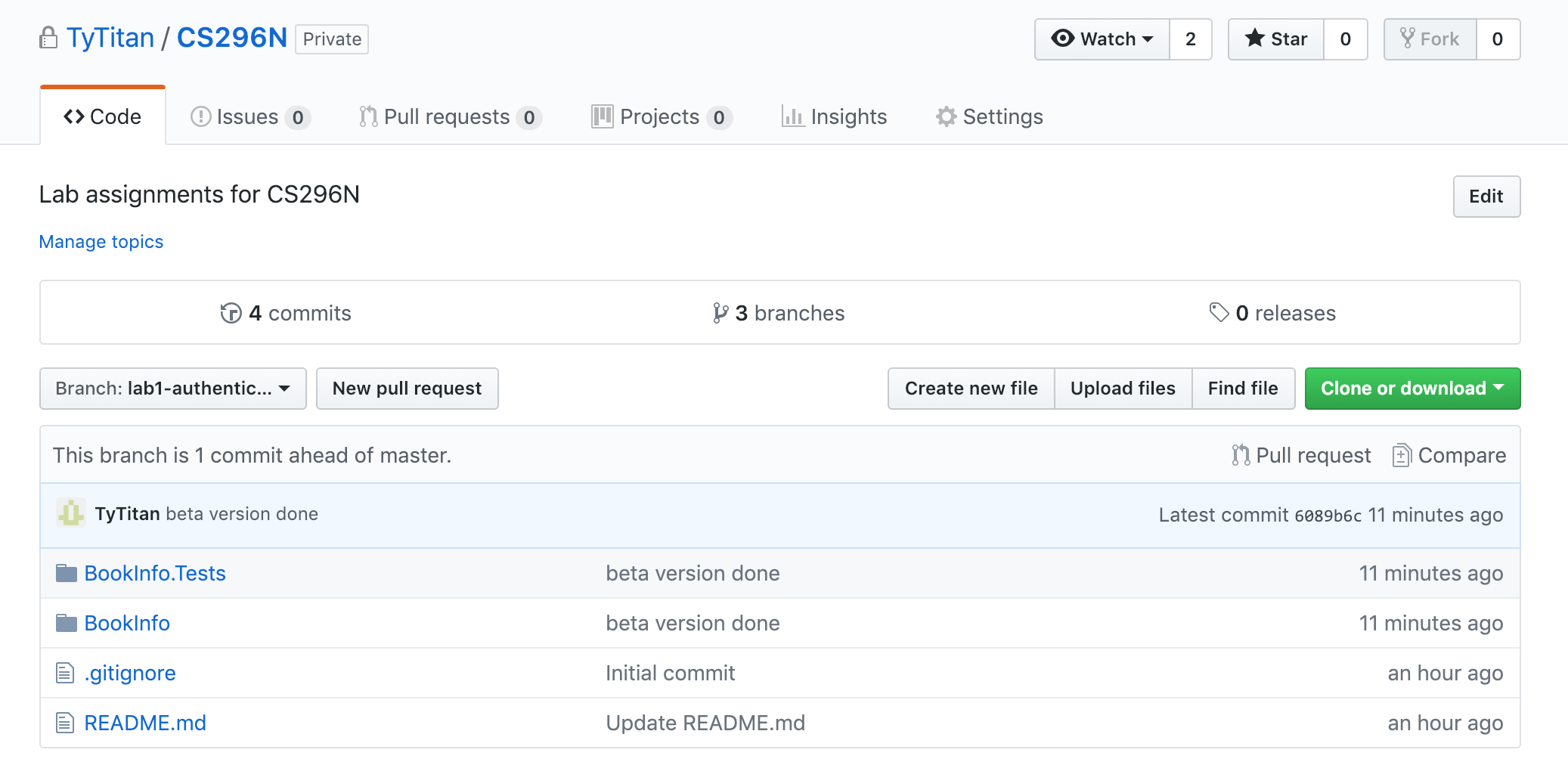
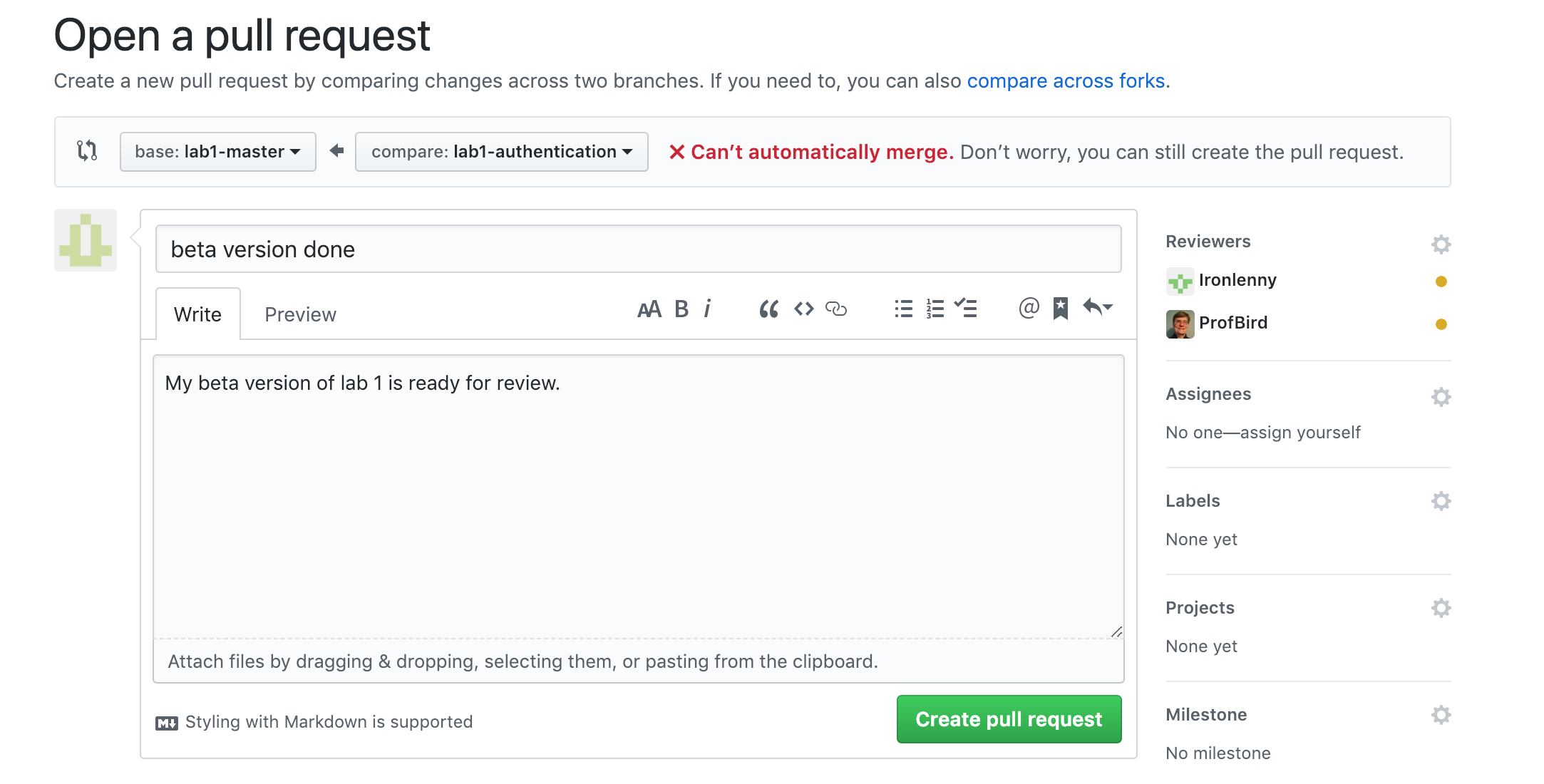
Imagine you are TyTitan and this is your repository. Your initial code commit (which could be from your last lab of last term) is in the *master* branch and the lab 1 code is in a branch named *lab1-validation*.

1. Click on “New Pull Request”

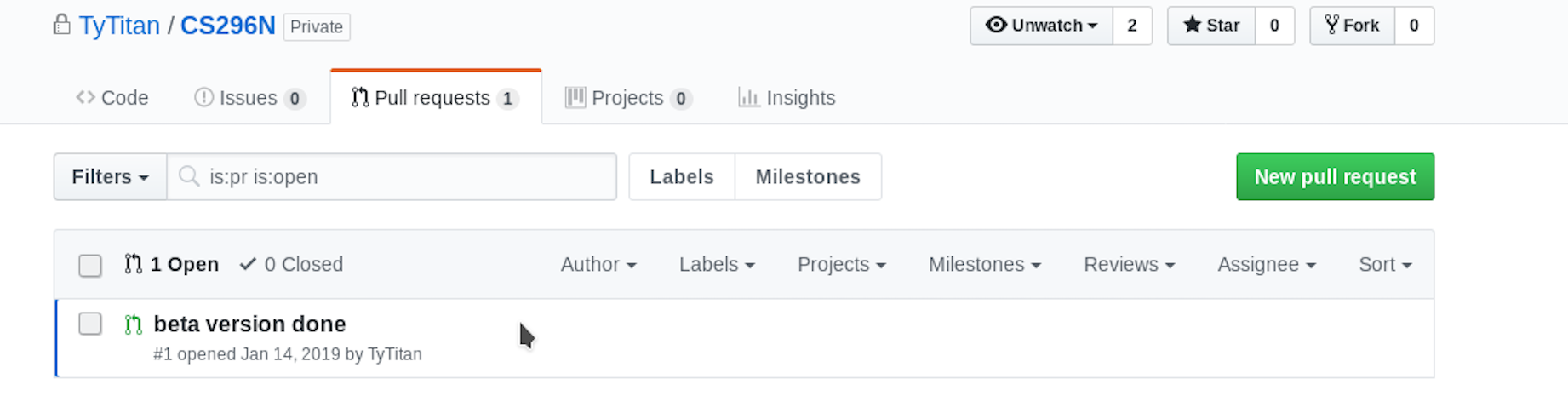


1. Choose the branch with your lab beta code as the “compare” branch.  
   Note that the *base* and *compare* branches must contain code that has differences.
2. Select reviewer: the student who going to review your code.
3. Add a comment.
4. Click on “Create Pull Request”.

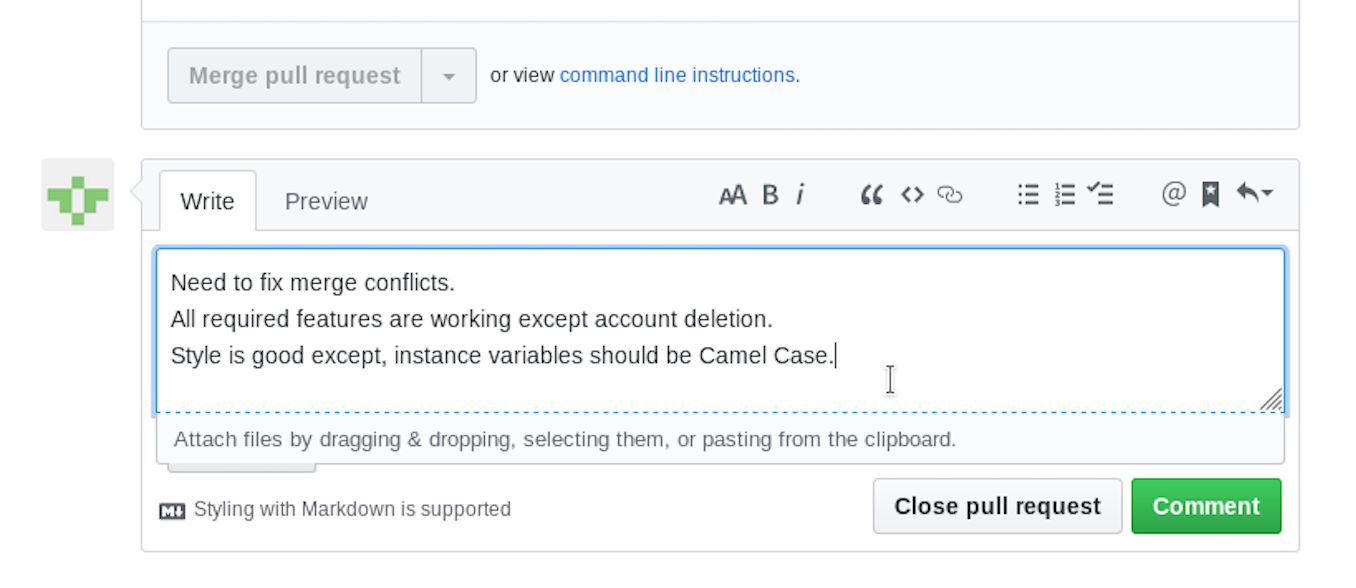
The pull request is now sent to your reviewer, ironlenny.

Doing a Code Review

Now imagine you are TyTitan’s lab partner, ironlenny. You just got an e-mail with the PR, when you follow the link in the e-mail you will be taken to TyTitan’s repository and you’ll see this:

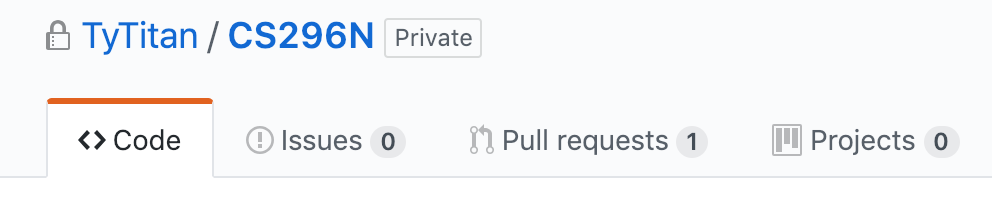


If you haven’t already cloned TyTitan’s repository, do that now. Now you can pull the branch, look at the code, run it, and do a code review. If the code review form is in MarkDown format, you can copy and paste the completed review into the comments window, otherwise you can attach it to the comment.



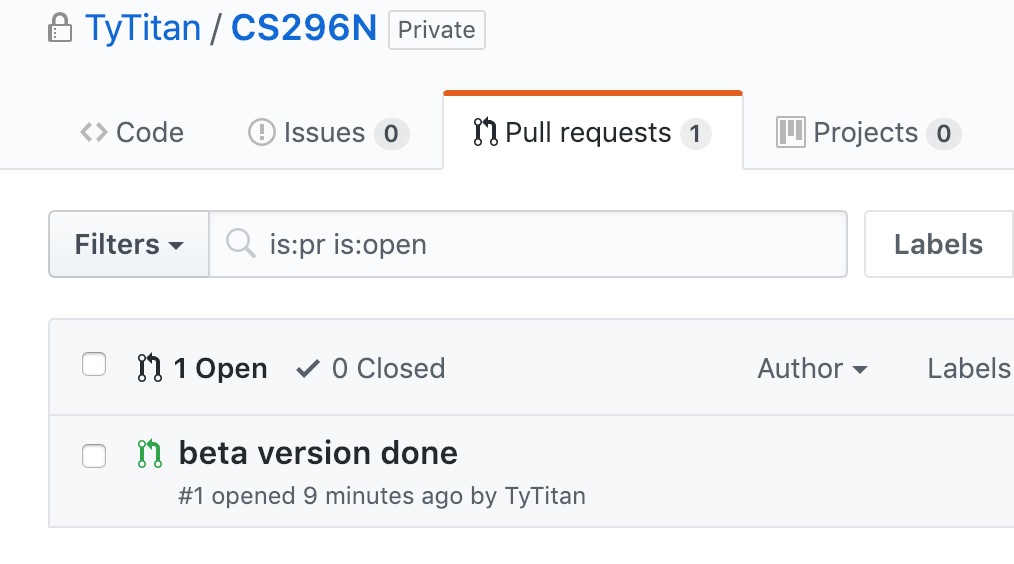
Reading the code review from your lab partner

Now imagine you are TyTitan again. After your lab partner has done the code review, you can read their comments by first clicking on “Pull requests”

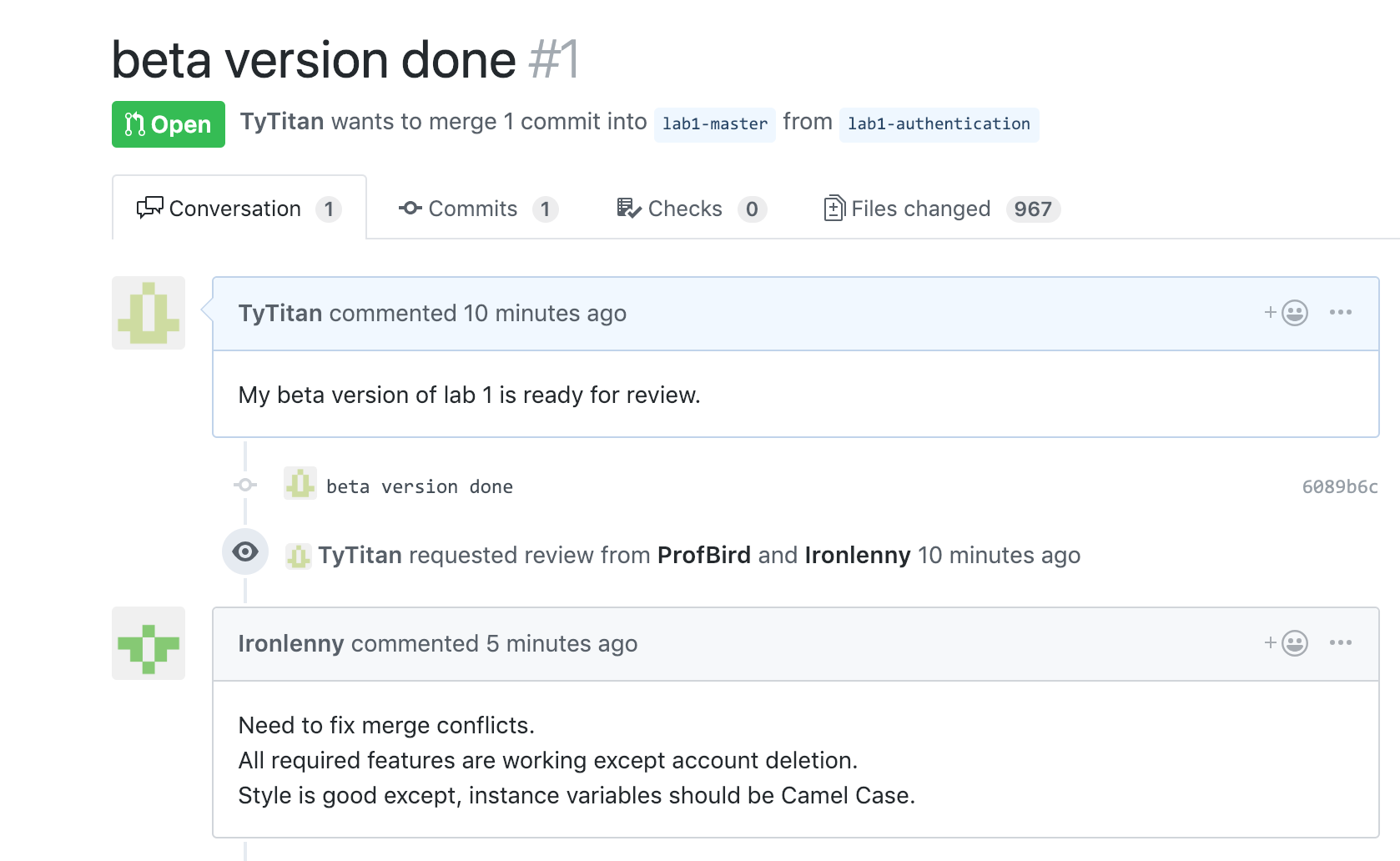


Then select the current pull request

(see next page)



Then read your lab partner’s comments.



Your lab partner, ironlenny, doesn’t need to do the merge or close the PR, you can do that yourself once you are satisfied that the code is production-worthy. Your merge will end up back in master, but keep the Lab1-Validation branch to show what you did for that lab.

It’s fine to make multiple changes to the beta branch and get feedback from your lab partner multiple times if your lab partner is willing.