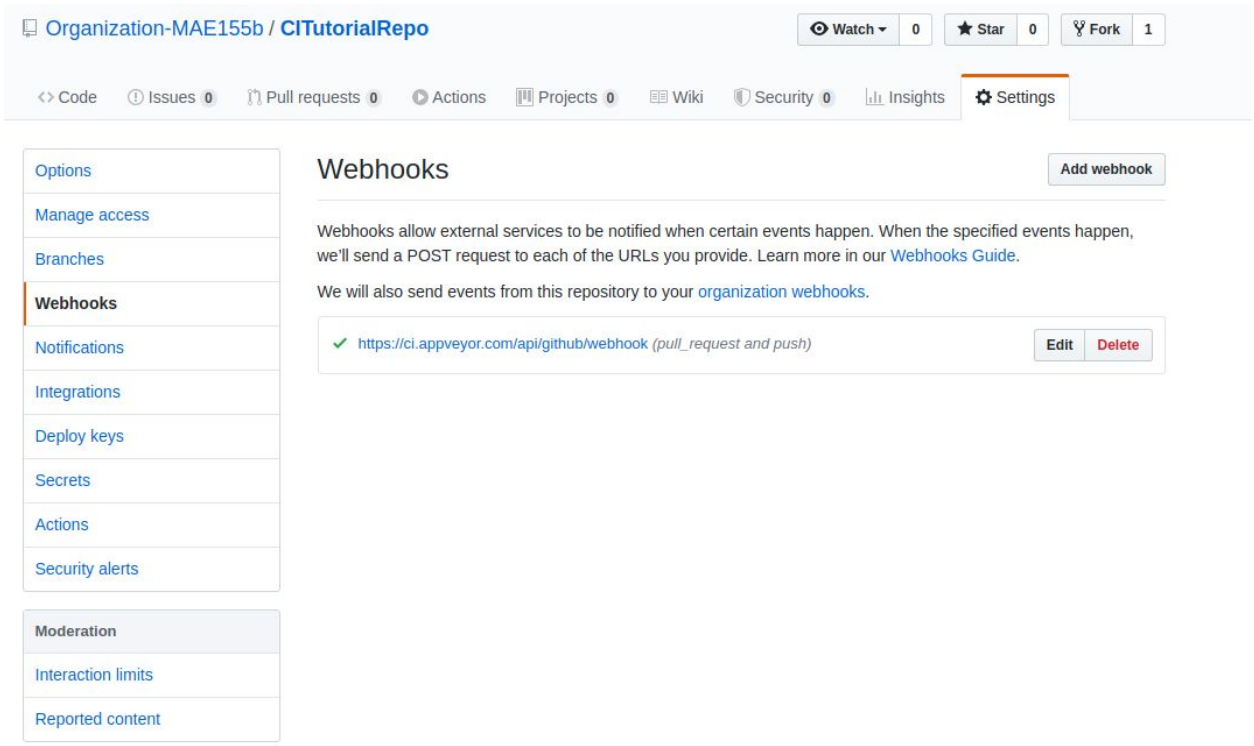
Appveyor Setup

This is to provide you guys with a step by step guide on how to get appveyor working on your github organization.

Steps:

1. Each team member creates their own github account.
2. One of the team members creates the organization for the team, and adds the other the teammates to the organization.
3. Create a repository in the organization, and make sure that the repository is PUBLIC, and initialize it with a README file.
4. Each team member should fork the repository.
5. At this point, each team member can start doing work on their respective repository, but I will outline the steps taken in the video to get the tests working for everyone.
6. Each member should clone their repository onto their local machine using the command: git clone “repository https”
7. Download the files from Canvas, that are all the files from the tutorial. The uploaded files are the ones that have no errors in them.
8. Then copy these files into the repository on your computer. The repository is the folder that is named the same thing as the forked repository. In the tutorial, it was CITutorialRepo. If you want to see what I mean visually, go to A1 in the Appendix in the bottom of the document and make sure to scroll down, because there are some images that provide context.
9. Once you have added the files to the repository on your computer. You can make sure that git sees the new files by using the command: git status. All of the new files will be shown in red.
10. You can go through the files and decide which ones you want to add using the command: git add “filename”. If you want to add all the files, use the command in the quotes: “git add .” . For the sake of this example, you want to add everything so use the “git add .” command.
11. Once you have added everything use the: git commit -am “A message about you're committing” command. The -a means everything, so in this case you’re committing everything. The -m means message, so you have to attach a message.
12. Next, use the command: “git remote -v” . This command will list the places you can send your commit. You should see the url for your own github accounts repository listed as origin, not the repository of the organization.
13. Use the command: “git push origin master”
14. Refresh your github account, and you should see all of the files added under your repository. It should also say that your repository is 1 commit ahead of the organization.
15. The leader of the organization should go to the appveyor.com, and give appveyor OAuth on the account.
16. Once authorized, add a project. Make sure to add the organization as a project.

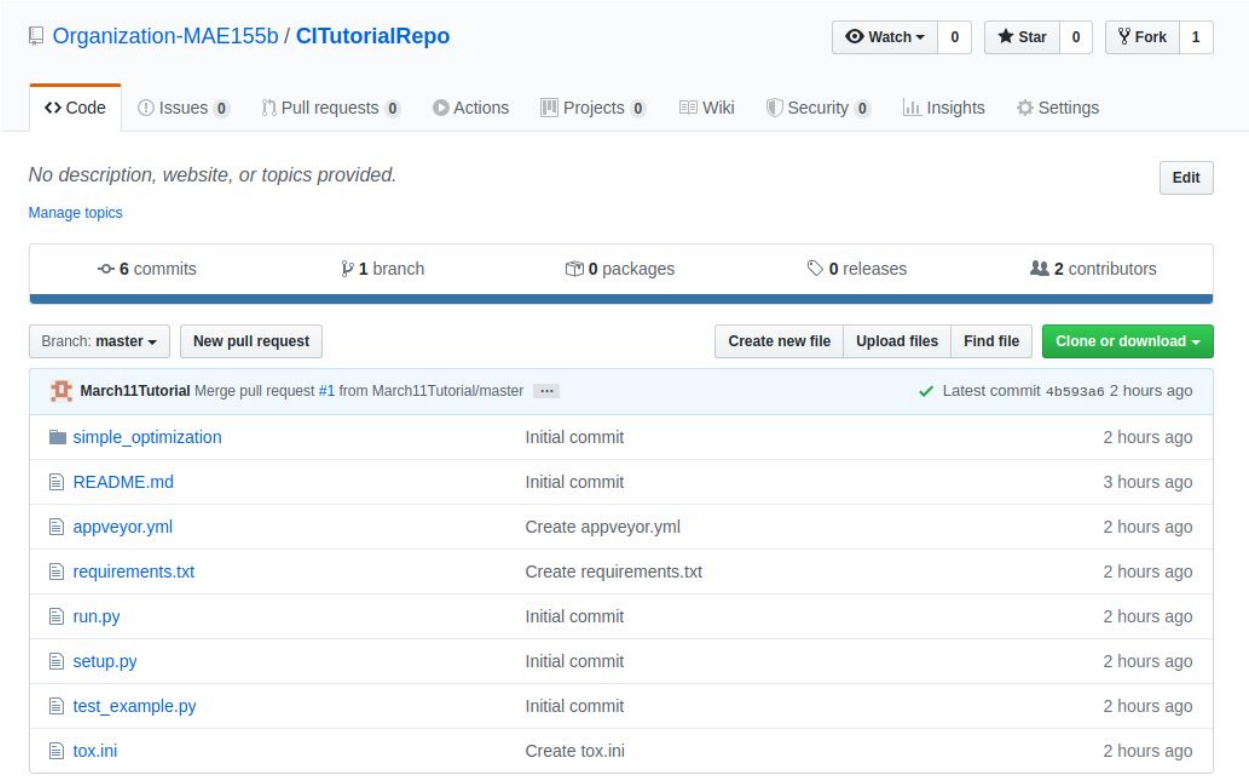
17. To make sure that everything is set up properly, you should go to the organization, and click on settings. Under settings there is a tab called “webhooks” . You should see something like this:



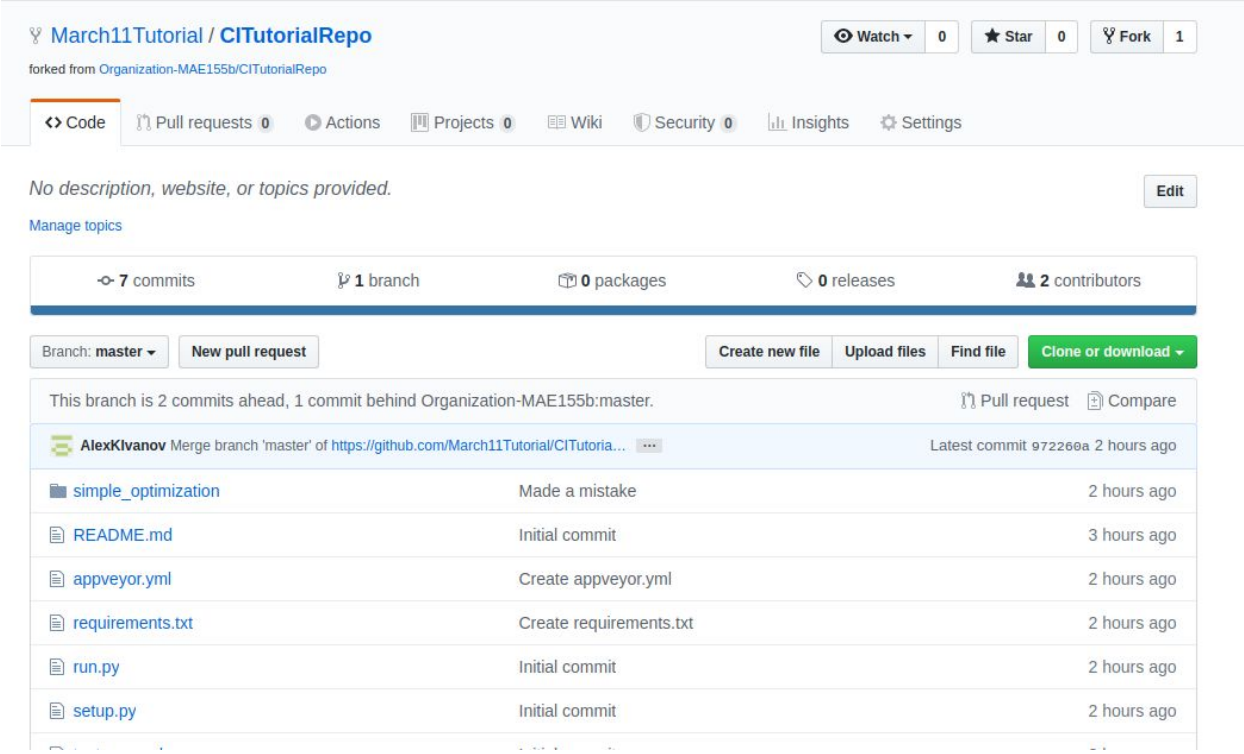
18. If you see this, then everything is good to go. Try to issue a pull request.
19. If the pull request shows that appveyor is checking the tests, and they pass then you have done everything correctly.
20. Try to make a mistake on purpose on your computer, and push to your repository, and issue a pull request with the error.

If you have any questions, or something is unclear, please let me know and send me an email at: alivanov@eng.ucsd.edu

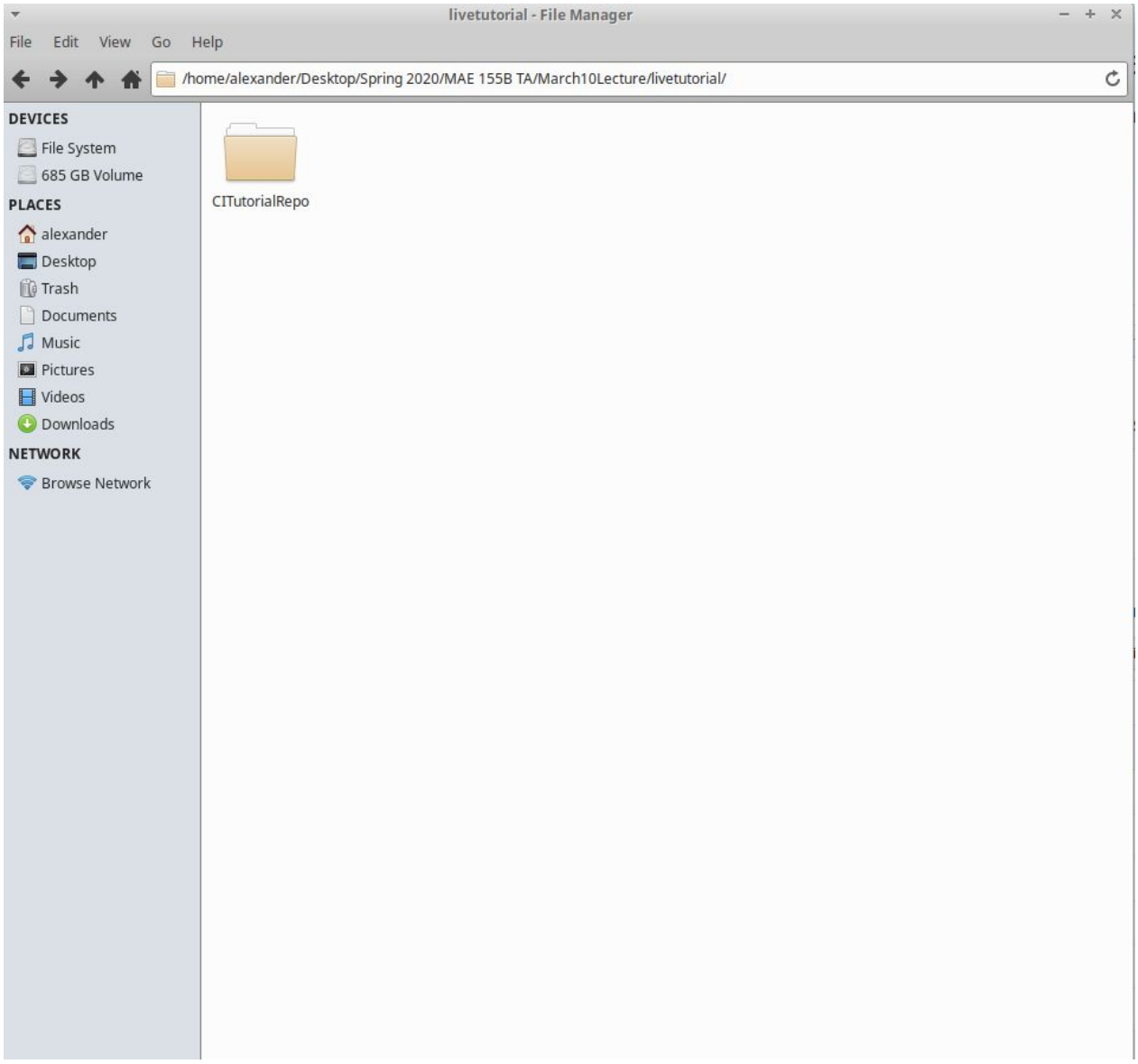
Appendix:
A1: Visualization of the Repository



In the image above, this is a screenshot of the organization. Here you can see that the repository created in the organization is called: CITutorialRepo, and all of the files are saved under that repository. When you first do the forking of the repository, you will most likely only have a readme.md file in this repository.



In the image above, this is my account's repository. You can see that it is titled the same thing as the organization repository, because I forked this repository from the organization. Forking a repository is essentially copying a repository and all of the files that are currently in it. So we can see here that the repository on my account is titled the same thing as the repository in the organization.



In the image above, this is the folder that is created once I use the command: git clone “repository url”. I cloned this from MY fork, you should not clone this from the organization. Once this is cloned, you can see that it created a folder titled the same thing as my repository, and the same thing as the organization's repository. Inside this folder is where you can use the git commands.