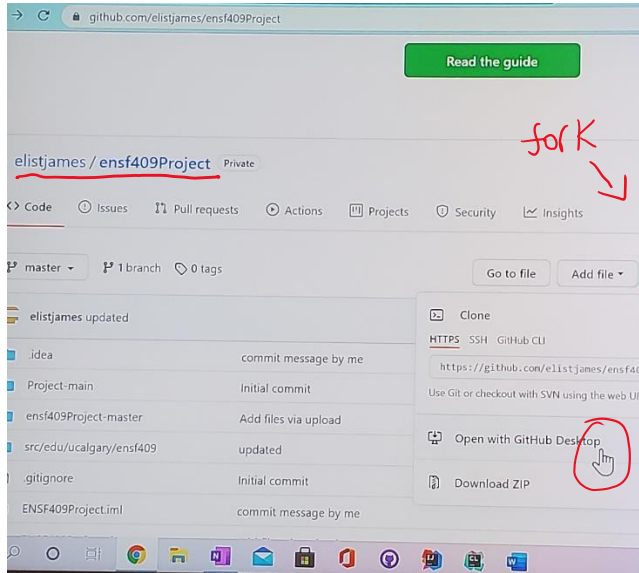
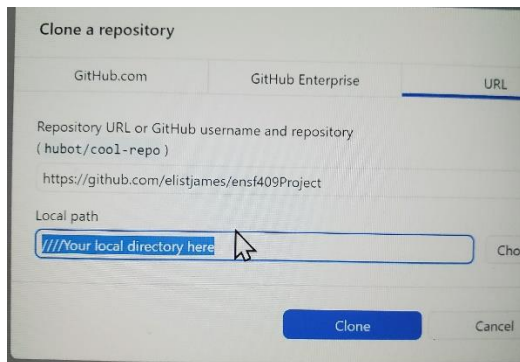


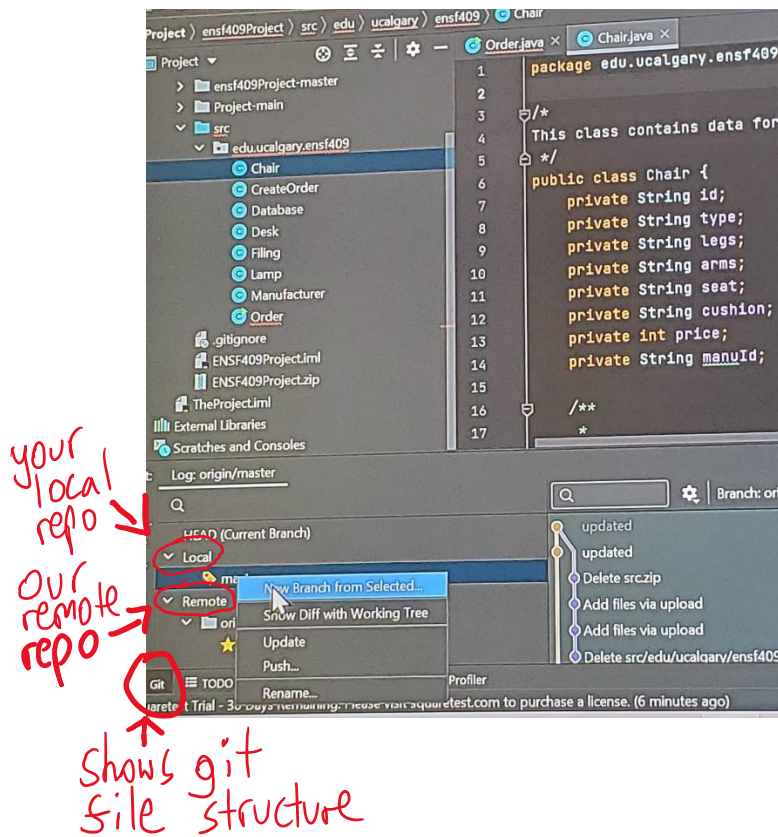
1. Fork the repo off Eli's main page where our repo is (far right button). This will put a copy on your github page, in case something happens, we all have a backup.
2. Install GitHub desktop (or use command line but this is easier)
3. Clone our remote repo off of Eli's page (this will allow us to work on our own local repo while sending our work to the remote repo on Eli's page):



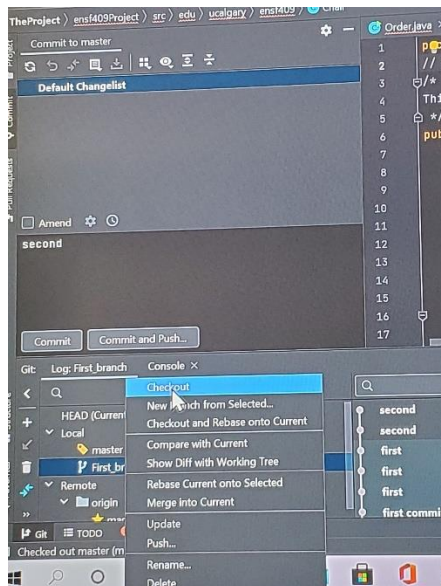
4. Choose where you want to store the local repo; a directory somewhere in your local files:



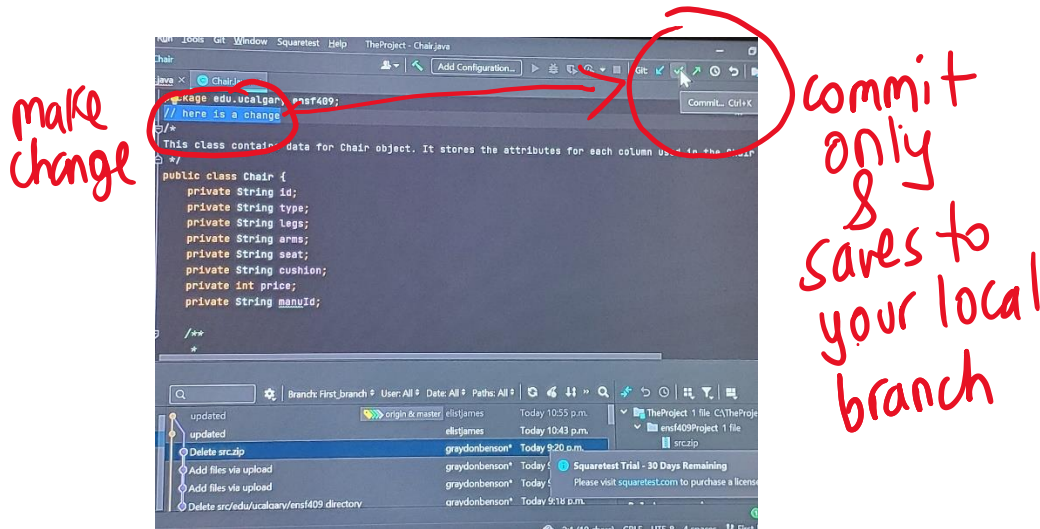
5. Open a new project with your chosen directory in IntelliJ where the repo was stored when you cloned it.
6. This is where you can physically see the local and remote files in IntelliJ and can select where you want to make a branch for your local repo where you can work:



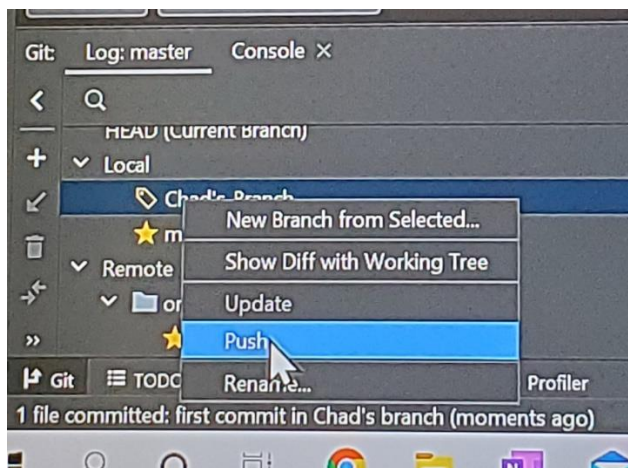
- Once you made a branch, make sure your selecting the branch you want to commit to. This can be done by using “checkout” branch. This allows you to move to different branches:



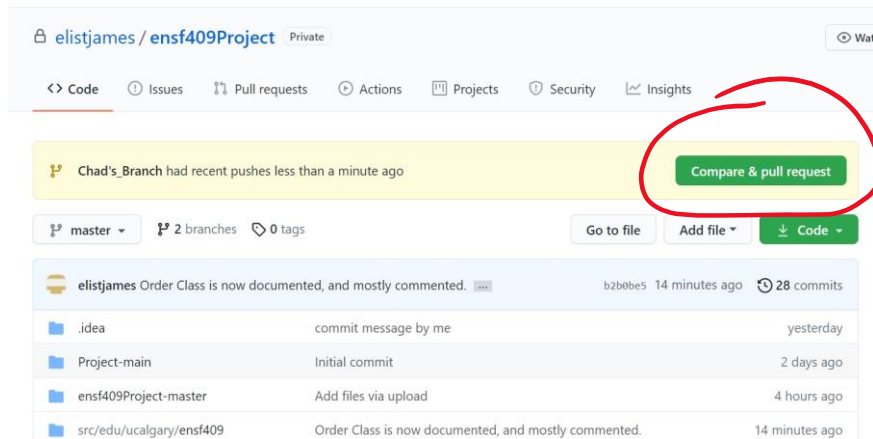
8. Now your work is stored in your local branch and you can come back to it later. You can commit as many times as you want and keep saving it to the same branch, make a new one, merge them if you want all in your local workspace.



9. When you want to send your branch that you've worked on to the remote repo. Then, you push the branch. This will create a compare and pull request option where we can review the changes to all files being made on Eli's github page:



10. Now the branch is on the remote repo. Here is the compare and pull request button:



11. Now you click the button and make a pull request. This allows everyone to view the change under pull requests tab before merging to the master branch in the repo on Eli's github page.

P.S I'm assuming Graydon is familiar with IntelliJ but the concepts should be the same at least if your using Eclipse instead.