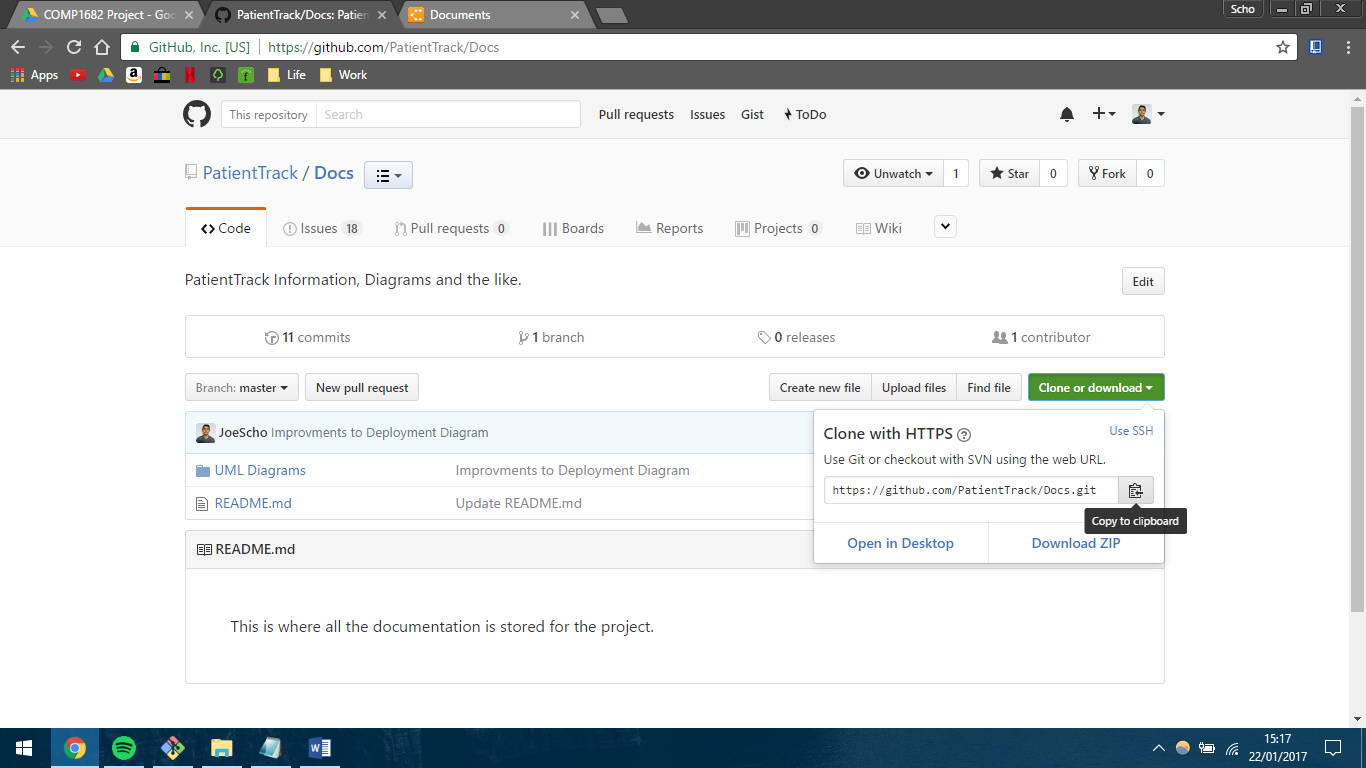
**GitHub Cheat Sheet**

**Commands to run are** highlighted in yellow**.**

**Cloning a repo:**

* Go to repo
* Copy clone link (see pic)
* Go to Git Bash
  + cd to desired directory (i.e. cd C:\users\Joe\documents\project)
  + git clone <link>
* Done!

**Working on a repo:**

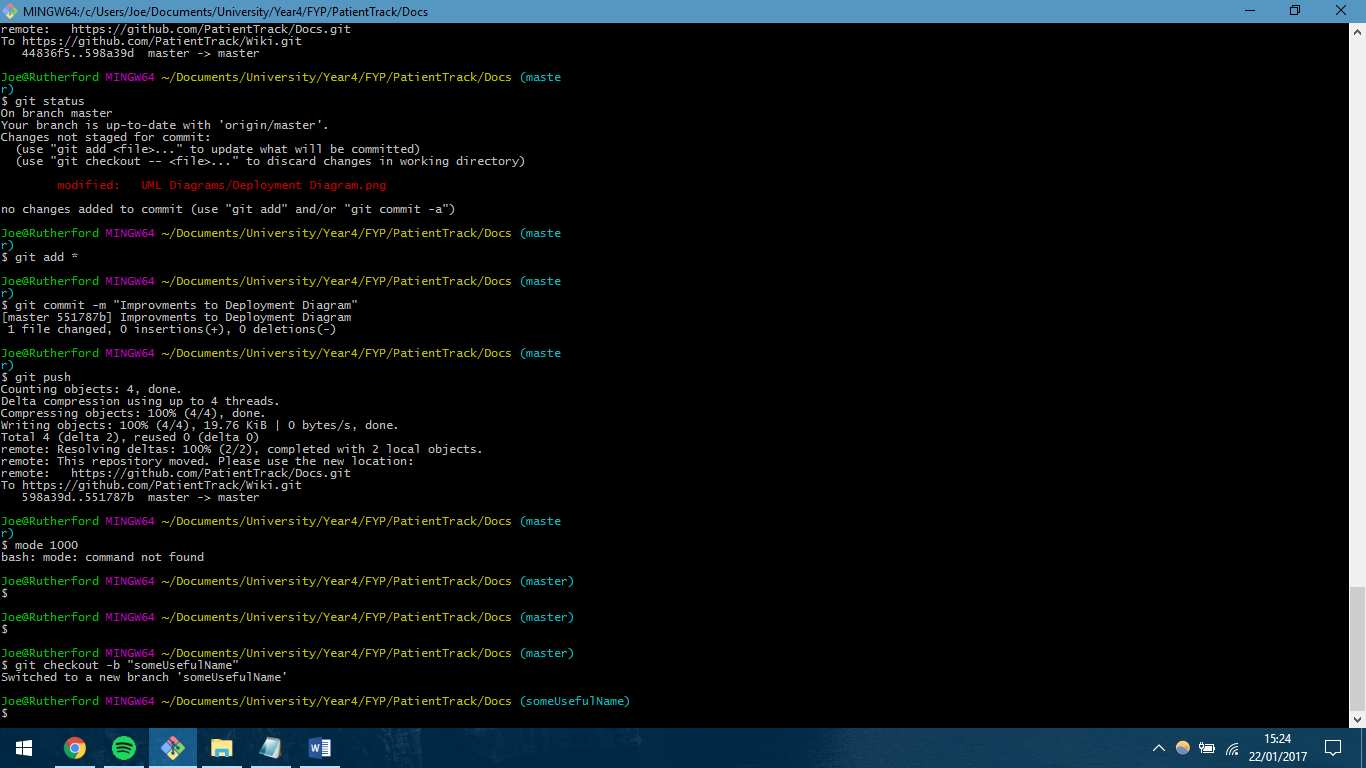
* Once cloned, the repo is stored locally. This is just a copy, changes made do not affect the online repo.. yet.
* So open the code in whichever IDE / text editor you like and do what you need to do!

**Delivering code:**

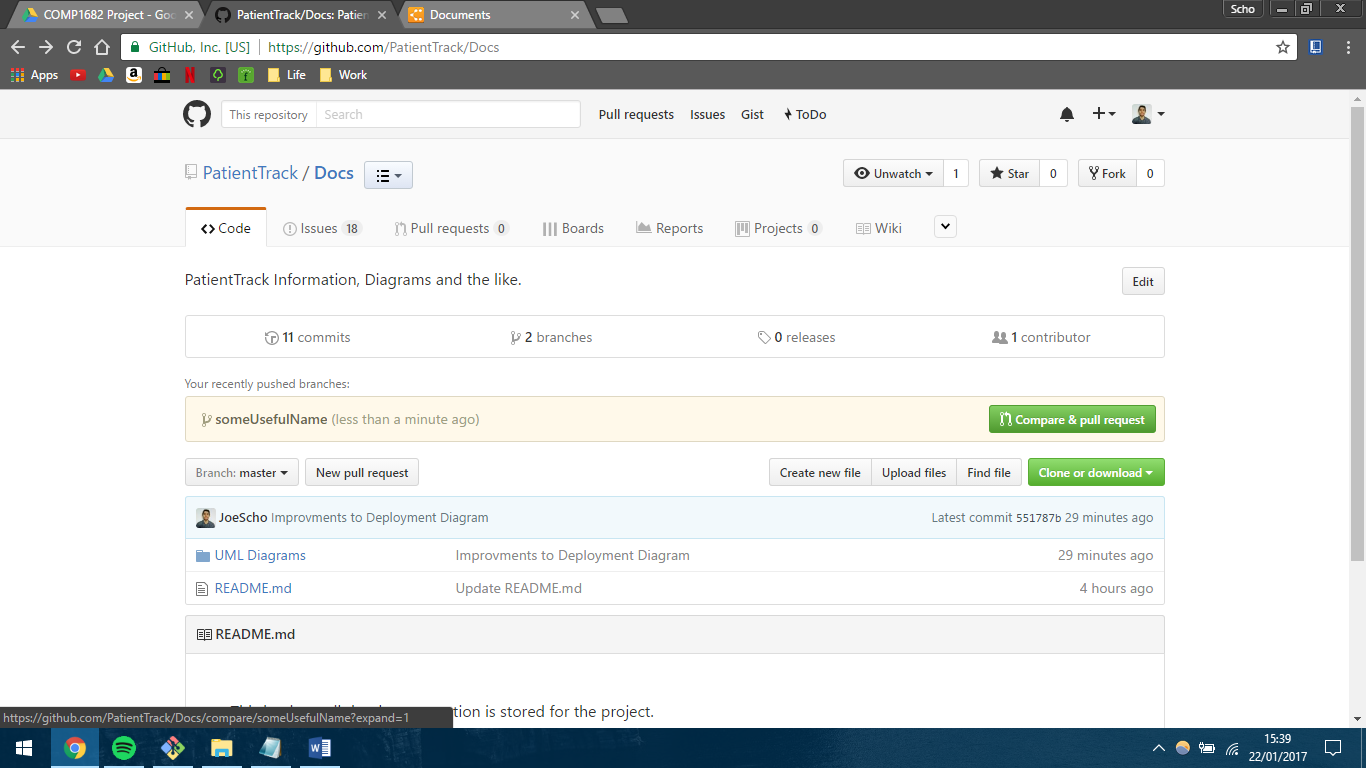
* You shouldn’t deliver code to the master branch, so make your own – I’ll explain branches below.
* Anyway, once you’ve finished working, deliver your code **straight away**. Otherwise you might lose stuff.
* Run git status – this will show you all the files you have added / changed / deleted. Check this is as expected.
* Run git add <files you want to deliver> (alternatively run git add \* if you want to deliver all changes)
* Run git commit – m “<some explanatory message>”. This is the commit message that will appear on GitHub.
* Run git push. This will upload the changes.

**Branches**

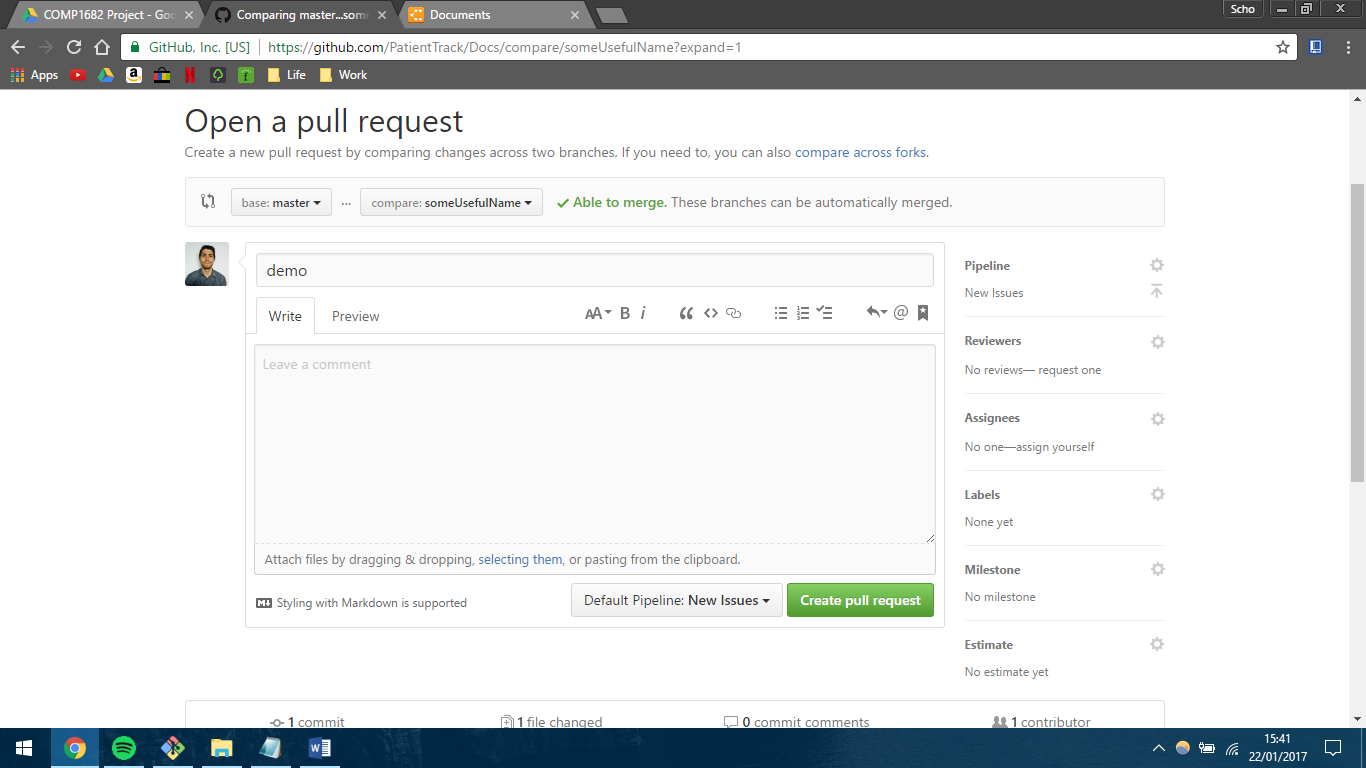
* When delivering code, you shouldn’t deliver straight to the master branch in case you accidentally write code that breaks everything.
* This is where Branches come in handy.
* A branch is essentially a copy of the master branch which you can see online and compare the two. The idea is that you deliver your code to this, submit a ‘pull request’, someone else checks the code and merges it with the master branch.
* To make a branch, run git checkout -b “someUsefulName” in git bash. The branch will be created and you will automatically be in it.



* The deliver code to the branch as normal. Once you are happy that your work is complete then you can submit a pull request. You do this using the GitHub website.



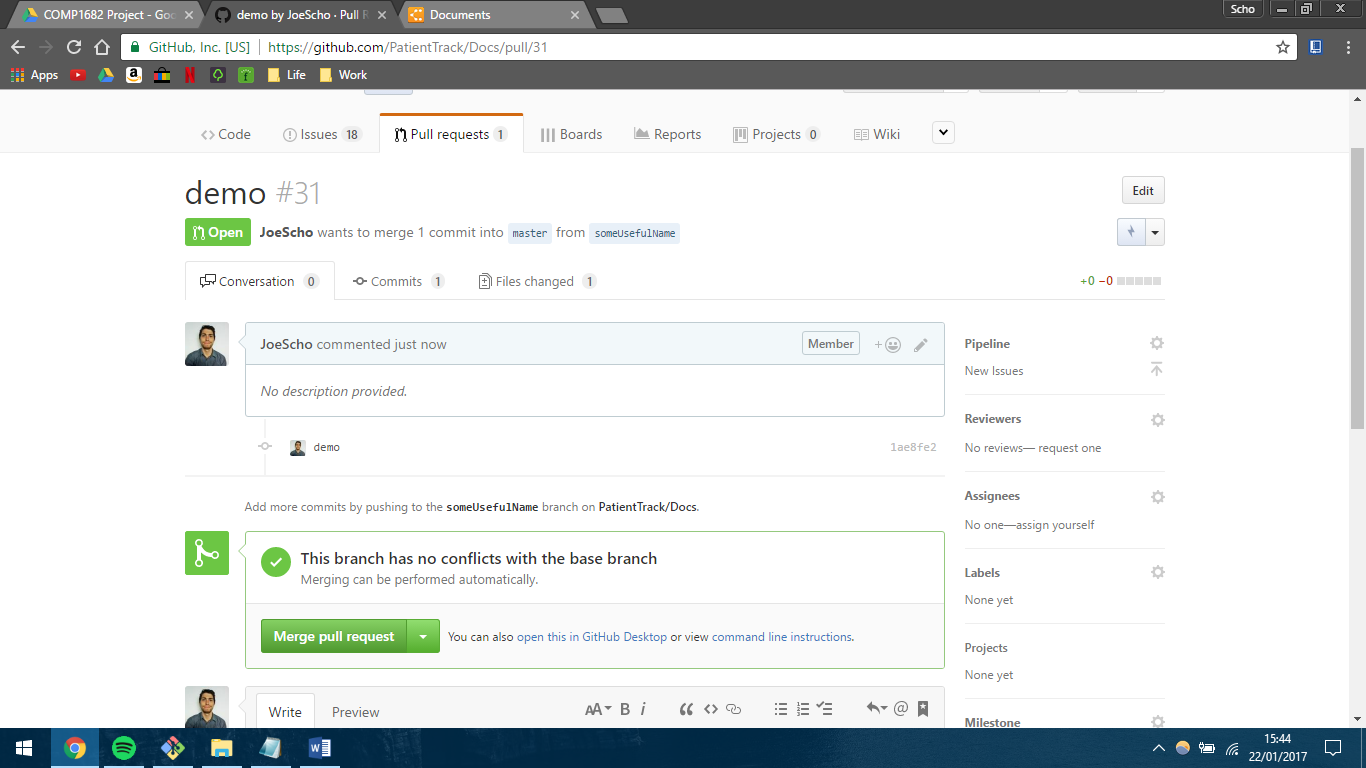
* Once you have created the pull request, assign it to someone to review and merge your code. You *can* merge it yourself, but please don’t.



* Once the PR is merged, delete the branch you were working on by running

git branch -d “yourBranchName”

**Merging a PR**

* If there are no conflicts, then it’s as easy as clicking a button.
* If there are then you have to compare the code and put it all together yourself… But that’s not likely to happen unless 2 people are working on the same part of the same file at the same time.