If you want to learn more about GitHub go here ----> <http://git-scm.com/book>

**Foreword**

Before starting, I want to mention that in the interest of having one “owner” of the repository, I will be deleting your accounts from the WeatherApp repository. Equal collaboration would mean we all have read/write (full) access to the repo and that’s precisely what we’re trying to avoid. Rest assured, however, that you will all be contributing by writing code, but instead of merging directly to the repo, you will be working with your own Fork of the repo, pushing your changes to your Forked repo, and then making a request to merge your changes into the team’s repo.

**Setup**1. Prior to doing anything, make sure you have Git installed on your machine. From the main page on Github, you can search for my user name (eulloa), select ‘Users’ and pull up my page. There, you will find the ‘WeatherApp’ under ‘Popular Repositories’.

2. Once you have the WeatherApp on your screen, click on ‘Fork’ in the upper right corner of the page.

(Doing a Fork creates a copy of the repository and stores it in your very own Github section, completely separate from the group repostiroy)  
  
3. Now that you have the repo forked, look for the section ‘HTTPS clone URL’. Choose the manner in which you’re most comfortable getting this repository locally (onto your machine).

At this point, you should have the repository forked into your own section on Github and you should also have a local version on your computer. This local version can be named anything, although it makes sense to name it ‘Weather App’, or something similar. You can also store it anywhere you want on your machine.

4. Next, you want to set the URL from where you will occasionally be updating your repository.

(When one of your teammates makes a code change and it has been accepted and merged into the shared team repo, you need a way to grab those changes and see them in your section on Github as well as locally. This way you’re keeping your repo up to date with all the latest changes your teammates are making)

This is commonly referred to as the ‘upstream’. Again, it’s just the location of the shared, team repo. I’m sure there’s more than one way to do this but I use a command prompt (Git Bash). The command you want to run is:

git remote add upstream https://github.com/eulloa/WeatherApp.git

This will allow you to run:

git pull upstream

This command grabs the latest changes from the new repository and merges them into the current branch you’re on, keeping your forked repository on Github (and your local repo) up to date with your teammates changes.

The nice thing about this is that because you are no longer collaborators, you can only pull from the upstream, not push. This is important because it establishes one person as the “owner” of the repo, and only that person can merge code into the repo.

**Coding**

5. You are now ready to begin coding. This should be the workflow:

a) Make changes (these are local changes)

b) Stage and commit your changes (commits are the changes that will be included in your push to your own Github repo)

c) Once you are satisfied with the results of your changes and you have tested to make sure that your changes have not adversely affected the app, push your changes to your forked repo.

*This pushes up the changes you just made to your forked repo, not the team’s repo*.

6. Then, go to my account again and find the WeatherApp. If you recall, this is the team repo. On the right hand side of the page, click on ‘Pull Requests’.

7. A pull request is a request to merge the changes you just made in your forked repo to the team’s repo. Click on ‘New pull request’.

8. If you click on ‘compare across forks’ , you can have Github show you the differences in the repos (yours and the team’s) by setting the base fork to the team’s repo, and the head fork to your repo.

9. Once you’re satisfied that the changes you see are yours and that all looks good, click on ‘Create pull request’. Here you can add a title to your Request as well as a description. When done, hit ‘Create pull request’ one more time on the right hand side of the page.

10. That’s it! Now, when I log in as the “owner” of the repo, I can see that someone has made a pull request. I can review the changes you’ve made and I can accept them if they look good. Now, when your teammates pull from the upstream repo, they will get your latest changes that were just added to the team repo.