**Git Cheat Sheet**

\*Note: Okay guys this is a quick reference guide for you guys on using Git. If anything is not clear to you, please ask me (Alex) ahead of time before something goes wrong it really is no bother.

**Branching Model**

Our branching model is a simple one we will be doing most of our branching from develop always but here is a quick example.

Master – we will only merge when we complete a new develop

Develop – main place we will work and branch from

* Branches from Master
* Merges to Master upon version completion

Feature – name it something relevant to what it will be implementing

* Branches from develop
* Merges to develop upon feature completion

Hotfix – used when a bug is discovered, and we need to fix it

* Branches from develop
* Merges to develop upon bug squashing

Document – used when you are making a change to any of our document files tedious I know but it’s the only way.

* Branches from develop
* Merges to develop

**Flow for Pushing**

* git status (check to make sure what you changed is showing)
* git add - - all (if you are sure you want to add all)
  + git add <filename> (if you want to add specific files)
* git commit -m “your message” (leave a relevant message plz)
* git push (this will push to current branch)
  + git push origin <your branch> (push to specific branch)

**Flow for Pulling**

* git status origin <branch> (to get the status of the branch)
* if pulling from a branch and you have done work locally
  + git stash (store your changes)
  + git pull (pulls from current branch use origin <branch> for specific)
  + git stash pop (bring back your changes after pulling)
* git pull (pulls from current branch)
  + git pull origin <branch> (pulls from specific branch)

**Creating Branches**

* ensure you are on the right branch to create your new branch from
* we will usually be branching from develop
* switch to develop branch
  + git checkout develop
* name the branch something relevant to the feature the branch is for
  + ex: email, login, database
* git branch <branch name> (creates a new branch using the current branch)
* git checkout <branch name> (switch to your newly created branch)

**Merging Branches**

* after your feature is completed and tested we will want to merge it back to develop (consult with Alex)
* git checkout <original branch> (switch to the branch that we will merge into in our case usually develop)
* git merge <branch name> (merge the given branch into the current branch)
* delete the branch you created because it is no longer needed
  + git branch -d <branch name> (deletes the given branch)
* you can now create a new branch for your next feature 😊

**Conflicts**

* hopefully we will avoid all conflicts by ensuring that none of us are ever stepping on each other’s toes. However, I do understand that sometimes it is unavoidable if such blasphemous acts occur please consult your local SCRUMLORD
* That is all

**Branching and Merging Example:**

1. Do work on a project.

2. Create a branch for a new feature you’re working on.

3. Do some work in that branch.

At this stage, you’ll receive a call that another issue is critical, and you need a

hotfix. You’ll do the following:

1. Switch to your production branch (develop).

2. Create a branch to add the hotfix.

3. After it’s tested, merge the hotfix branch, and push to production.

4. Switch back to your original story and continue working.

5. Finish said feature and merge back to develop when completed