

Version Control

On GitHub, we need to stick to a certain standards so that the teams could collaborate better.

There are two **protected branches**: Whenever there's a push or merge request, it requires the approve of scrum master and it requires more than one reviewer. And it will automatically run the tests.

- **main** This branch is a **protected** branch. After the version on **develop** branch is released and there's no bug on it, the **develop** branch can be merged into main. **Only develop branch** can merge into this branch. This guarantees that the **main branch** is always pure and lack of bugs. After releasing, we merge **develop** branch into this **main** branch.
- **develop** This branch is the **default** branch. When using `merge` command, the default target branch is this one. At the end of each sprint, we release the version on this branch. Every merge into this branch should be runnable and tested in the local environment. The codes on this branch is updated in anytime. **Create** a new branch from this one every time you start a new task. This ensures your developing branches is always updated. After a while of launching, if there are not many problems on the current release, we can merge this branch into the **main** branch

We use feature branches to code in our local environment. It's recommended that you create a new feature branch each time when you develop a new feature(task). If a few of features are mixed in one branch, it will be difficult when you want to launch one of them. For the purpose of readability, it's recommended that all the members use the format below to name your developing branch.

- **feat_XXX**
 - An example: `feat_homepageNavigator`
 - Create branches from **develop** branch *only*
 - When developing your feature, please don't fix others' bugs or modify codes which are not wrote by you. Otherwise it's easy to get conflicts if someone is also modifying the same file, and fixing conflicts is really time-consuming. If you find there's something wrong, create a task on Trello with some description and assign it to your team member if you know who's in charge of it. **Note**: please don't use code formatter for *the whole project*.
 - Only raise a merge request after your feature has completed and fully tested in your local environment. Commit and push them into your own branch if you're in the halfway.
 - After the current branch is merged into the **develop** branch, it's highly suggested that you create a new branch from **develop** branch, other than using the same old one.
- **hotfix_XXX**
 - We may not use this kind of branch much since we only have two sprints. Usually after releasing, if there's any problem online, we create this branch to fix a small bug and merge it into **develop** branch. Please don't use this to develop a feature or fix larger bugs.



