Exercise 4.9: Team GEEKS Branching Strategy

We have elected to use 4 different branches in the development of this project: master, development, feature, and hot fix.

Master: A branch for production-ready code. Each “release” is pushed to master from development.

Development: A branch for the current build of the product. Each finished task, once tested, is pushed to master. This branch should be the default branch of the repo.

Feature: Each leaf level requirement that we’re working on, listed in the GradebookRequirements document, will have a feature branch. Leaf level requirements are the sub-requirements of the two main requirements (manage courses and manage sections), such as ‘add a course’ and ‘report on courses’. Once completed, the feature branch will be pushed to development where it will be tested. Currently, we are planning to implement 11 leaf level requirements, which will result in 11 individual feature branches. Once a feature branch is merged into development, it will be removed. NOTE: Each specific change to a feature should branch of the main feature branch. I.E. If a team member creates a new function for the “add course” feature, the function should be created and worked on in a sub branch of the “add course” feature with an appropriate name (e.g ‘add-course-function-3'). Once the function development is completed, a pull request to merge back in will be made so that the changes may be reviewed and approved by the team.

Hot fix: This type of branch will be used to fix minor bugs that pop up later within sections of the project that have been pushed into the master branch. Once fixed, the branch will be merged back into both the development branch and the master branch. By merging back into both, both the original branch where the issue came from (master) as well as the current branch being worked on (development) will be fixed and up to date.

Testing: Although helpful in larger scale projects, it will not be used in our project repo. The inclusion of this branch would be redundant since the team will be testing the code as it is being developed. The action of testing itself will be done in the development branch.

Merge Days: We have agreed as a team that Fridays would be set aside for final review of code before any merges take place. Reviews of pull requests can take place as they pop up, but we would like to have a meeting on Fridays to ensure that all members of the team are up to speed on what is going on within the project and allow us to bring up any issues we may be facing. This does not negate using Teams and GitHub to discuss changes during the week, but rather sets a “soft” deadline for work.