**City University of Seattle**

**CS 504 - Software Engineering Summer 2021**

**Independent Project 1**

**Anil Erturk**

For the independent project, a Configuration Management System (CMS) will be implemented. The CMS will add structure and process to the project, as well as support teamwork. The blazor app will be used to demonstrate the software engineering concepts.

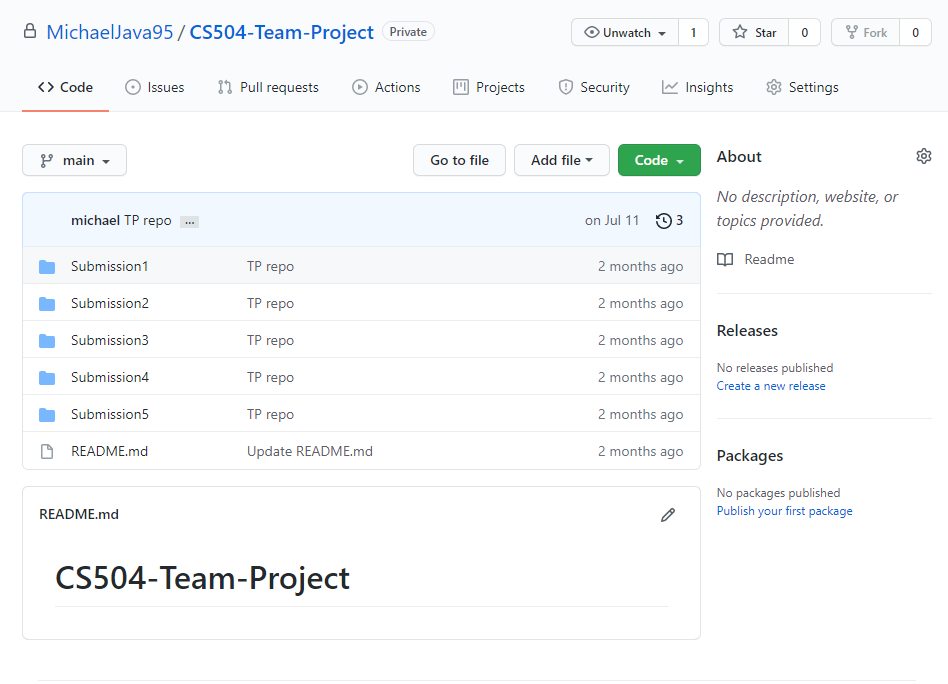
Some of the features of CMS to be demonstrated:

* Code check out
* Code check in
* Code snapshot (or release point)
* Code in development by two users, first one checks out code, second checks out code, makes changes and checks in, first makes changes and needs to check in.
* Code changes need to be rolled back to a previous level.
* One developer is working on a new version, one is fixing bugs in a previous release.

**-Code check out**

Git is not necessarily an online CMS. A folder can be selected as the repo and the repo in the directory can be created using the command “git init”.

Github serves as a remote repo. Here is the repo created for the project.

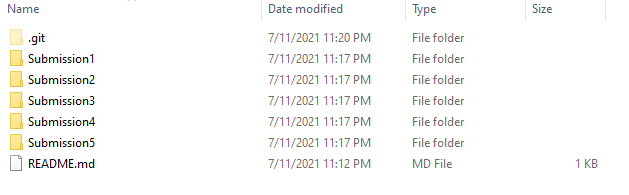


Another way to get started with a repo is cloning a remote repo using the command

git clone https://github.com/MichaelJava95/CS504-Team-Project.git

We can also specify a branch to clone using the command

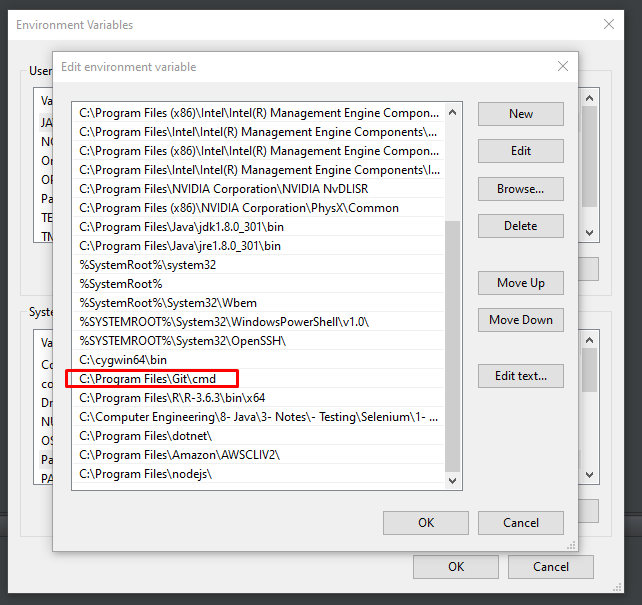
git clone –b branchName https://github.com/MichaelJava95/CS504-Team-Project.git



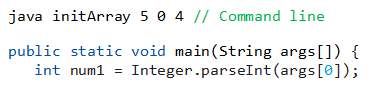
The local repo can be updated with the latest remote repo using the command

git pull https://github.com/MichaelJava95/CS504-Team-Project.git

In order for these commands to work from the command prompt, git needs to be added to the environment variables. The directories added to the environment variables are searched for the executable when a command is executed. An executable is a compiled version of a source code. Environment variables allow us to invoke the executable even when we are not at its directory. Notice that when we call the executable’s main method, we are also passing command-line arguments (main method arguments). In the case of cloning, clone and the URL are the two arguments we pass to the main method of the git executable.



The main method of the git executable can access these parameters and use them. Here is an example of how we call an executable from the command line (first line), and how the main method can access them through a String array parameter called args (second and third lines).



**-Code check in**

The initial version of the project can be moved to the Submission1 folder and the project folder can be added to the next commit using the command

git add BlazorApp

The following command will show all the changes to the previous version of the repo.

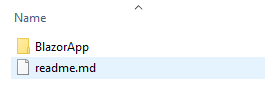
git status

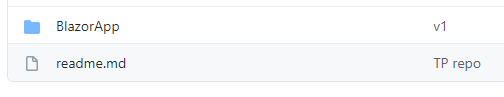
The commit is given an information message using the command

git commit –m “v1”

The commit is pushed onto the remote repo using the command

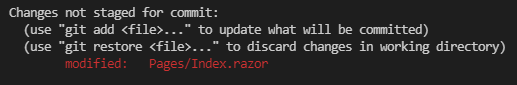
git push origin master



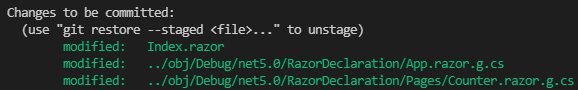


Now a change can be made to the file. Such as changing the header message in the index.razor file from “Hello, world!” To “Hello user!”.

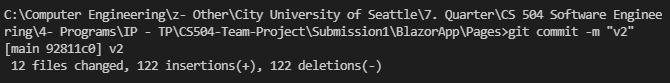
“git status” command will show us the changes not staged for commit.



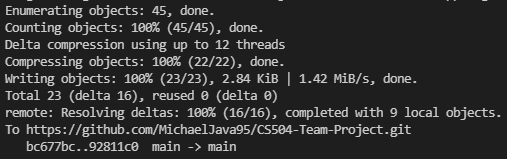
If we use the command “git add .”, the changes will be added to a commit.



“git commit –m “v2” command adds a message to the commit and shows us the details of the commit.



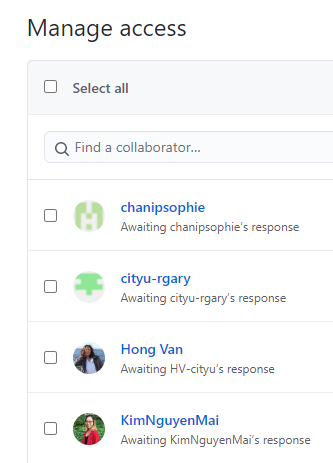
“git push origin master” pushes the commit to the remote repo.





Every commit we make gets assigned some metadata and this commit + metadata becomes a snapshot. The repo consists of series of snapshots.

**-Multiple users**

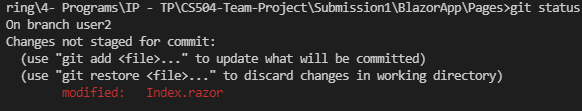


Multiple people can work on a single project. This is generally done through branches.

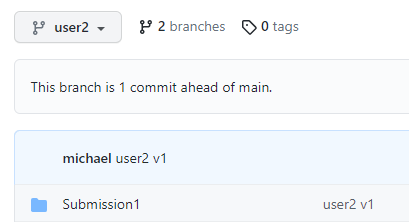
The second user can change the header to “Hello, user2!”. Then the branch can be changed to user2 using the command “git checkout –b user2”.



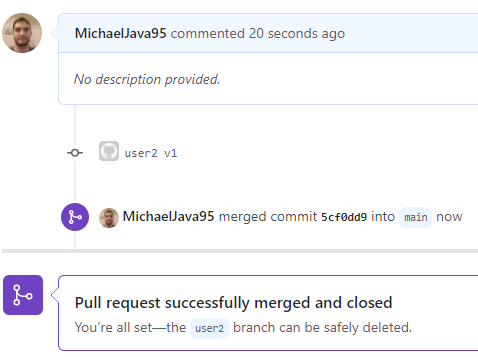
“git status” command shows the changes on branch user2.



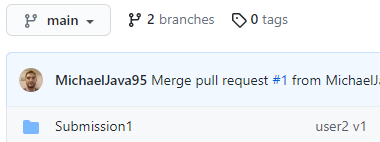
After add, commit, and push commands, we can see the new branch on GitHub. The branch name “user2” can be seen on the top left. The person who pushed this commit can be seen as “michael”. And the commit message can be seen as “user2 v1”.



A pull request to the master branch can be created. And this pull request can be confirmed.

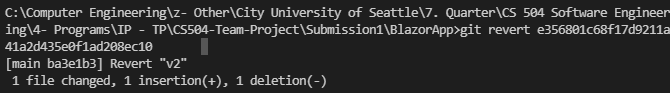


Now the master branch also has the latest version.



**-Roll back**

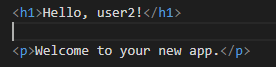
Going to a previous commit on our local machine is possible.

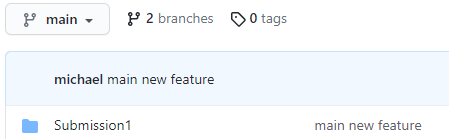


**-One user working on a change while another user fixing a bug**

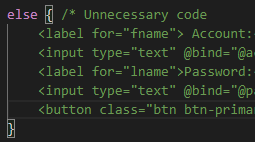
Because of branches, separate users can work on separate parts of the software. And these changes can be merged. main will only add a new feature while the user2 is changing another file. This doesn’t cause a problem.

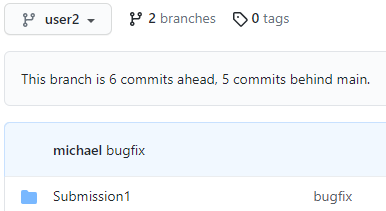
The main branch will add a new paragraph to the index.razor file. Then this new feature will be pushed to the main branch.





The user2 branch will fix a problem with the signup.razor file. Then this fix will be pushed to the user2 branch.





These separate changes can be merged using a pull request.

