# Unix Assignment 1 Report - Group 23

Members of the group:

Vulpe Dimitru, Aylin Ibryamov and Costin Grigore

We have decided to separate our project in different scripts joined by arguments, this decision was taken to avoid the creation of lots of unnecessary menus and to conform to UNIX standards.

We have encountered some problems doing this, the first one of them was creating a script to check in and check out the user's interaction with the files in the repository. The main problem was that the code did not run properly, therefore it would not work. We managed to solve this problem by taking another perspective for this functionality, by creating .log files to save the interactions of the user with the files in the repository.However there was still some little problems in the log script at first we did not know how to specify the file path for the log where it will be.

Another problem we have encountered was how to compare the differences between files and save them as a variable, as we were not sure on which approach to take when building the functionality for our project, we managed to implement this functionality by first creating a separate

script for our project that would manage the differences between the files and then implement it to the script that would allow the user to create new files, edit files previously created, delete files and show the contents of files. One of the things for us was to comprehend how to push and pull from a repository, after spending some time brainstorming we had a rough idea how to get there but, always seemed impossible.Moreover one of the most common issues we had is with git hub we had some sorta conflict that causing us issues. When we tried to do push or pull.

We have also had some trouble using variables from different scripts into other scripts, after some research online we managed to use the variables from one script into other scripts.

One of our major problems was that we realised our different scripts were not communicating as we expected them to communicate, unfortunately we did not manage to solve this main issue due to a lack of time. It was a bit unfortunate to find this out as the different scripts were working well by their own, as we have run some tests to make sure they were working before we tried to make them communicate.

We managed to create, edit and delete files using a UserAction.sh script but we did not have enough time to implement it into the project, because there was a little bit misunderstanding in between us. At the end what happened is one of us went with menus and the other with flags.

To conclude, the worst problem that we have encountered wasn’t necessarily the individual features of the program but it was the interscript communication. Bash simply is not built in a way to make these kind of features available, at least not comparatively to actual programing languages like C/C++ or Java. The way we tried to do all of the communication was through variables, but we were still hitting roadblocks like not being able to set environment variables which would persist throughout runs; for example one of the idea that we had was to first set the repository we would work with first (eg: “$ ./script.sh use testRepo”), then have “testRepo” set as the working repository for actions such as cloning, pulling and pushing.