**Project Milestone #1 - Proposal**

|  |  |  |
| --- | --- | --- |
| **Victor Reginato**  Hamilton, Canada  reginavp@mcmaster.ca | **Erica Cheyne**  Hamilton, Canada  cheyneem@mcmaster.ca | **Don Pham**  Hamilton, Canada  phamd@mcmaster.ca |

**ABSTRACT**

This paper proposes Git to be the design project topic. The proposal explains the reasons behind choosing Git and outlines the tasks required to complete the project.

**INTRODUCTION**

Git is a source control management solution that has many convenient features. It is often used by developers to maintain source code for applications. It can be easy to start using Git through the GitHub desktop GUI (Graphical User Interface). Git has had various GUI’s developed with different purposes. Some implementations are aimed at making the version control easier to use for newcomers. Other GUIs attempt to show detailed commit and branching history of a repository. There are a wide range of users of Git and its associated GUIs, from beginners to professionals. There are GUI solutions available for many use cases.

**CURRENT ISSUES**

Some of the things you may want to do with Git are rather specific. Many of the GUI based solutions hide too many details from the user. The oversimplification of Git commands can often result in undesired results in one’s repository which can be disastrous.

For example in the GitHub desktop GUI provided by GitHub, you can “Sync” your staged changes to the remote repository. The Sync button in the GUI pulls changes from the master branch and then pushes those changes to the remote. In the event that you want to disregard the state of the remote repo, and push only your local repo to the current HEAD, you are unable to specify this through the GUI.

Another obvious issue with GitHub desktop GUI is that you do not have the option of stashing your changes if you quickly want to test something on another branch. The GUI does not let you use the features of Git to its fullest extent.

**SCOPE**

Our final product will not necessarily be fully functional,

however, we will strive for completeness of the interface. The bare minimum for our project will be the shell of the user interface. We will have buttons with their uses / equivalent commands clear. As the project progresses, especially if we have success with the user interface portion, we plan on adding in functional features. Given the project timelines, we do not expect to be able to finish a complete solution, but would be pleased to have a set of features implemented.

**DIVISION OF LABOUR**

We plan on splitting up the tasks in each milestone as evenly as possible:

1. Milestone 1
   1. Choosing an idea - Group meeting to discuss and decide
   2. Writing out the proposal - work spread as evenly between all 3 members as possible
2. Milestone 2
   1. Reviewing Four applications - The whole group reviews one application together; each group member reviews one individually
3. Milestone 3
   1. Each group member makes one persona
   2. The entire group will collaborate on design iterations, individually making suggestions and jointly deciding whether to accept them
4. Milestone 4
   1. Requirements not yet given, expectation is to share the load evenly with individual ability affecting how much each member can contribute

**CONCLUSION**

The GUI that will be developed as a result of this project should make it simple for people to know which tools are available to them as well as possible use cases for those tools. The GUI also needs to clearly and unambiguously display the current state of the repository. The overall goal of the design project is to make a GUI that allows for the user to take full advantage of the features that Git offers, while remaining as uncluttered as possible.

Tweather