File Stalker

Collaborative Team Project

Stage III – User Interface Design

CSC260

Dr. Pulimood

3 April 2012

Team C.H.I.M.P

Alexa Cain

Danny DeChiara

Derek Duchesne

Tim Lee

Glen Oakley

Steve Schwarcz

**User Interface Overview:**

*Start Screen:*

Once the user starts up the program, they will be taken to a command line window displaying the title of the program and some of the more useful commands for searching through File Stalker’s log system. These commands will include:

* Changing the currently monitored directory
* Filtering the log for creations to a specified directory
* Filtering the log for removals from a specified directory
* Filtering the log for modifications from a specified file or directory
* Printing output from the log
* A help command for additional, miscellaneous tasks

The naming convention for each command involving files and directories will be consistent with the basic file manipulation commands found on Unix systems. (Example: “cd Desktop” will be used to switch File Stalker’s monitored directory. Below the list of common commands are two lines displaying:

* The currently monitored directory
* The current filter

*The Selection Screen*

The selection screen displays the same information that the start screen does with the exception of no longer displaying the File Stalker logo. At the end of each input from the user, the selection screen is refreshed with new information on the currently monitored directory and current filters.

*The Filtering Screen*

When the user is entering commands to be able to filter output, they can specify certain additions to the filter, such as one asking for modifications to a file called “oldfile.txt” on June 19, 2013 (filter –dm “6/19/2013” oldfile.txt). If the user only enters “filter”, they will instead be taken to several linked sub-screens that will ask questions one at a time for what the user will be inputting. For example, the first screen will display (“Which type of filter would you like to choose?” with options 1, 2, and 3 for Modification, Creation, and Removal).

*Error Handling:*

Any entry of incorrect data that does not match with File Stalker’s predefined set of commands will result in a simple error message being displayed. This message will show which of the user’s search tags were incorrect and remind the user of the ‘help’ command. The user will immediately have the option to enter their tags once again.

*After Data Entry:*

After the user has entered in valid filtering parameters, a loading message will be displayed until the system has finished parsing the log. If the user then enters the print command, the contents of the log will then be displayed through the program’s window and, after printing has finished, the option to save the log as a text file will be given to the user. The option to save the filtered log as a text file will be displayed to the user with a simple ‘y’ or ‘n’ request. If they choose ‘yes’, the program will give a short message saying that their log has been successfully saved to their current directory. After this, and if they had previously chosen ‘n’ from the last question, they will be returned to the selection screen of File Stalker to enter in new commands.

**Eight Golden Rules of User Interface Design:**

*Strive for Consistency*

The system is meant to mimic UNIX commands in style and attempts to integrate pre-existing command functionality where appropriate. The program is meant to be run through the terminal, and keeping the styles consistent should allow users to feel most comfortable using the software.

In particular, commands will be ideally one word or short phrases followed by options. To filter results, for example one might type “filter –dm” to filter output results by the date modified.

*Enable Frequent Users to Use Shortcuts*

The program will support guiding a user through their options as well as letting more experienced users type a single line of command. If a user types “filter” for example, the program can walk the user through new submenus relating to filtering. As shown before, however, the user typing in a single command like “filer –dm” should also be acceptable. This will allow new users to avoid confusion and experienced users to avoid the frustration of always needing to be guided through each step.

*Offer Informative Feedback*

If users input an invalid command, the program will not simply crash or sit idly without feedback. Instead, the program will tell the user: “Invalid command. Please reenter.” It will also remind them of the help/menu display of valid command listings.

 Additionally, whenever any aspect of the program is successfully executed, the user will receive a relevant text update. Most of the time this will be in the form of on-screen data that the user requested. Sometimes there may not be an obvious in-program change however, such as when a user requests that the data be output to a text file. In this case and other similar cases, the program will display a short informative message such as “output.txt created successfully.”

*Design Dialogs to Yield Closure*

As mentioned in the previous section, no actions the user asks the program to execute will go unacknowledged. Most closure will come in the form of the actual data appearing on-screen, however the previously mentioned short messages will enable the user to always feel informed about what the program is doing. Appending a simple “Start” message as the first line and “End” message as the final line of all displayed and output data may also let the user know the full scope of a successfully executed command.

*Offer Simple Error Handling*

User input errors will be solved by requesting the user re-input valid commands. If the program itself has trouble, it will display relevant text such as “Directory not found.” If the program runs into unexpected errors while running it will attempt to display as much information to the user that it can regardless.

*Permit Easy Reversal of Actions*

The filtering system does not actually change the data collected, it merely sorts it for the user based on requested fields. As such, if the user is unhappy with the look of their output they can clear the screen and request a new set of fields to filter by. In the case of output file creation, the program should check to make sure no previously existing file will be overwritten and request if the user wishes to overwrite if such a file exists. The program cannot revert file changes once they occur, but it can ensure that the user is positive of his or her actions before they are executed.

*Support Internal Locus of Control*

Output should always be based on what the user requests. In particular, the filtering system will allow users to feel that the program is displaying data based on their wishes and not mechanically showing data that they do not either understand or care about.

*Reduce Short-Term Memory Load*

Allowing the menu to be accessed multiple times throughout a single use of the program will enable users to have a wide variety of command options without needing to memorize them all. This is also enforced by reminding the user how to access the command menu whenever they enter an incorrect command.

F**ile Stalker Command List:**

“ “ = required in the filtering tag

< > = optional in the filtering tag

If no specified file or directory is given, the log will be filtered in the context of the entire directory tree.

*Commands*:

* help (displays a list of all of the commands of the system)
* cd <directory> (changes the currently monitored directory)
* filter (brings user to several sub-screens to ask the user step-by-step how they would like to filter the log)
* filter –m <file or directory> - filters the log to only display modifications
* filter –c <directory> - filters the log to only display creations
* filter –r <directory> - filters the log to only display removals
* filter –dm “date” <file or directory> - filters the log to only display modifications during the date specified
* filter –dc “date” <directory> - filters the log to only display creations on the date specified
* filter – dr “date” <directory> - filters the log to only display removals on the date specified
* output – outputs the log based on current filters to the screen