Project 4 - A help interface

Jules Voltaire

CS3041

Prof. David Brown

04/26/2016

**User Analysis:**

This program is designed to be used by various types of users. That includes computer programmers and computer enthusiasts who may need to refresh on one or two commands, students and people who are new to Linux and would like to get familiar with Linux commands.

**Task Analysis:**

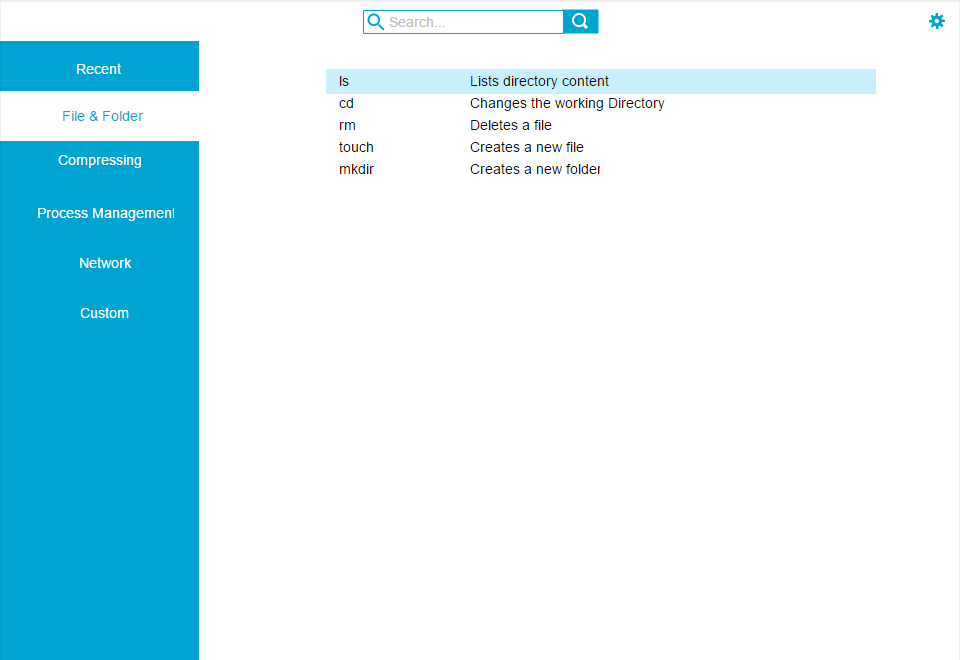
* Objects
  + Category object
    - Name
    - List containing commands
  + Command object
    - Name
    - Short Description
    - Detailed Description
    - Options e.g. -l in ls command
    - Format
    - Examples to show how to use the command
* Actions
  + Selecting a category
  + Selecting a command
  + Scrolling through categories
  + Scrolling through commands
  + Scrolling through details of a command
  + Searching for a specific command
* Examples
  + The user scrolls through the list of all the available categories
  + The user selects the File and System category to see all the commands in this category.
  + Once the user selects a category, he/she scrolls through the list of all the commands in that category. If the category is empty a label will be shown telling the user that the selected category is empty. If nothing is selected, the user is told that nothing a selected.
  + The user selects the “ls” command to view more details of.
  + Once presented with the data the user scrolls through all the details of the ls command that was previously selected.
  + The user searches for a command as opposed to navigating to it to be quicker. If an exact command is found, then that command is then shown. If there are multiple results, then the user is shown all the resulting commands to choose from. If nothing was found, then the user is notified that no commands were found and is given the option to search online for such command.

**Error Handling:**

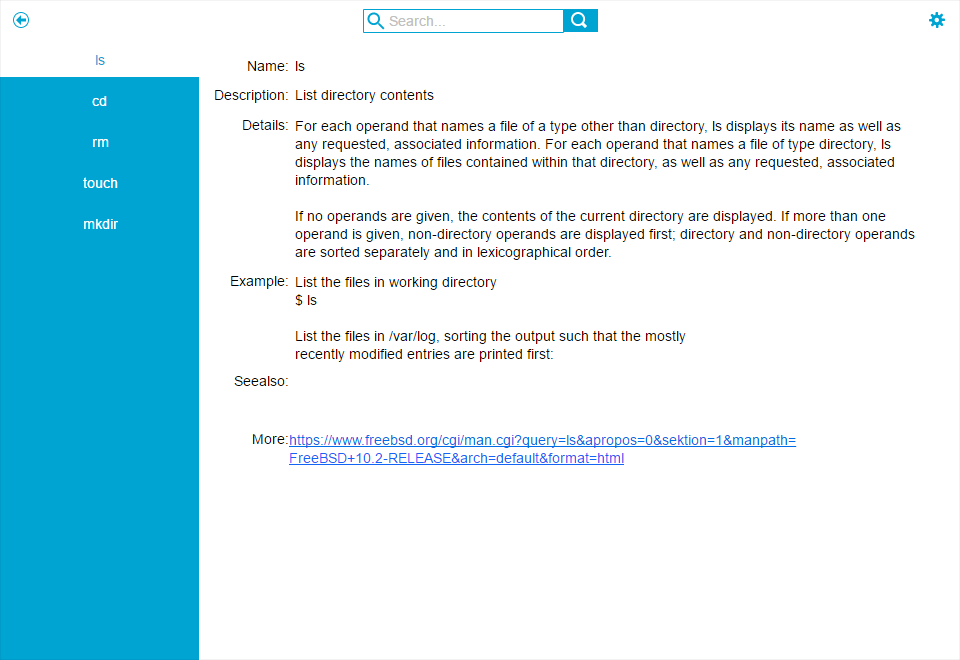
* The user searches for a command that doesn’t exist
  + Solution: The user is notified and given an option to search online
* The xml file is somehow missing.
  + Solution: A pop up dialog is displayed to notify the user of the error

**Low fidelity prototype:**

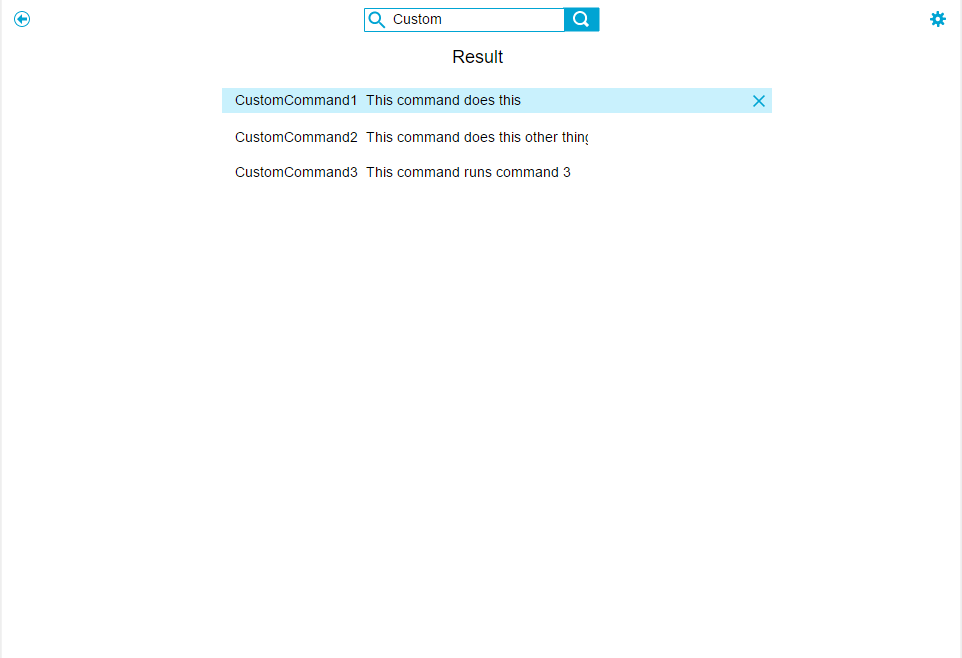
The main page showing the categories on the left and a list of commands in that category.



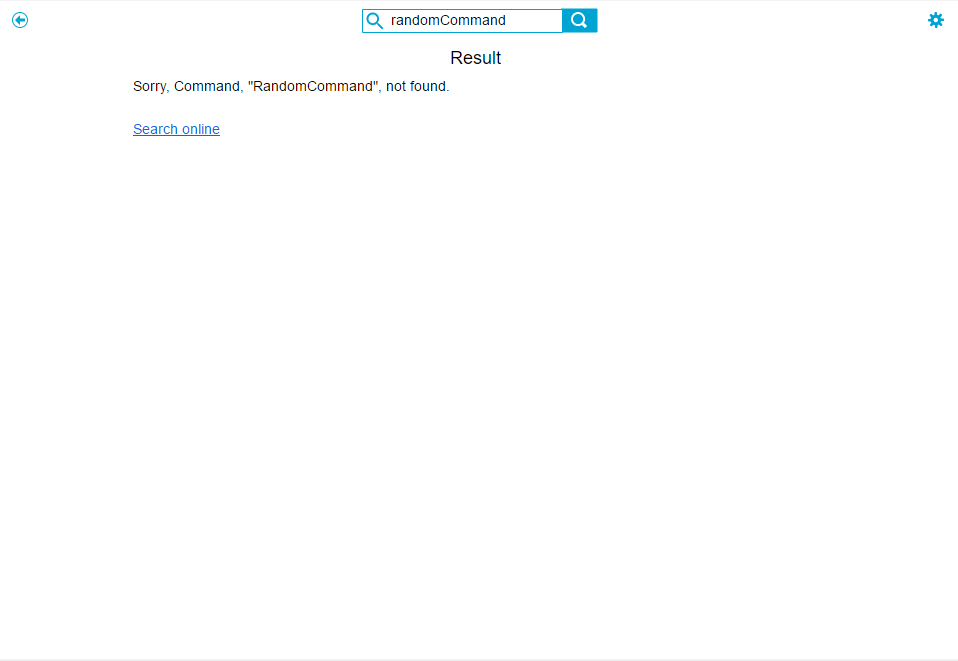
This page shows the list of commands that are in the same categories as the command that was previously selected. And on the right the details about the selected command on the left is displayed.



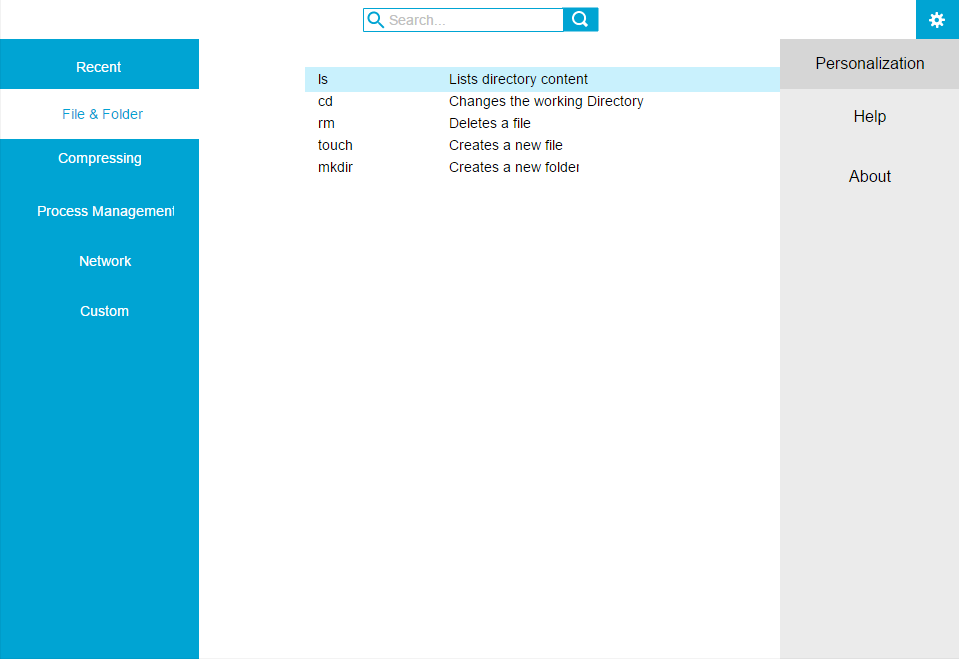
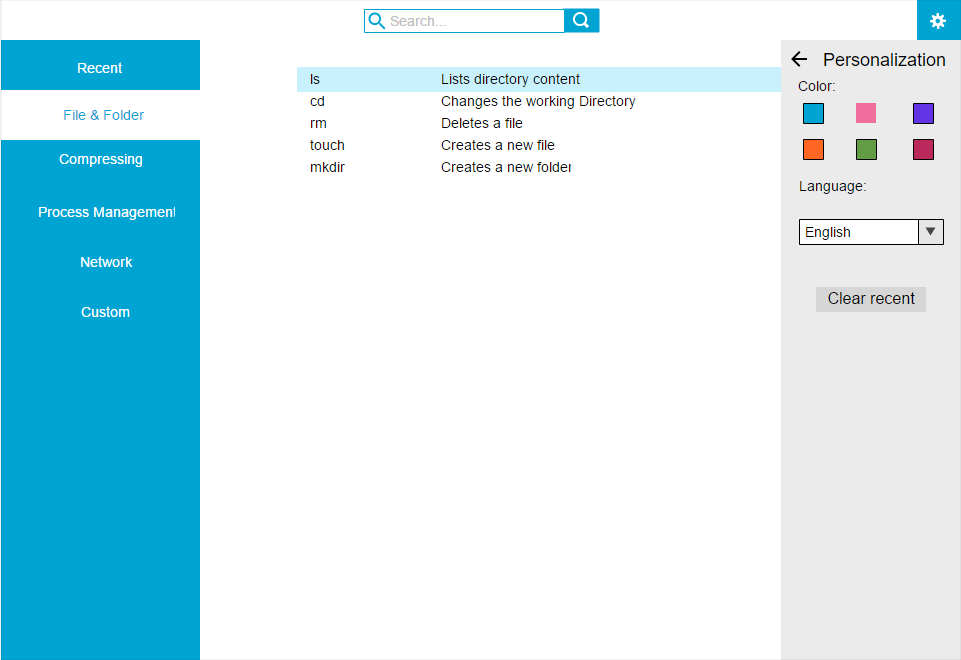
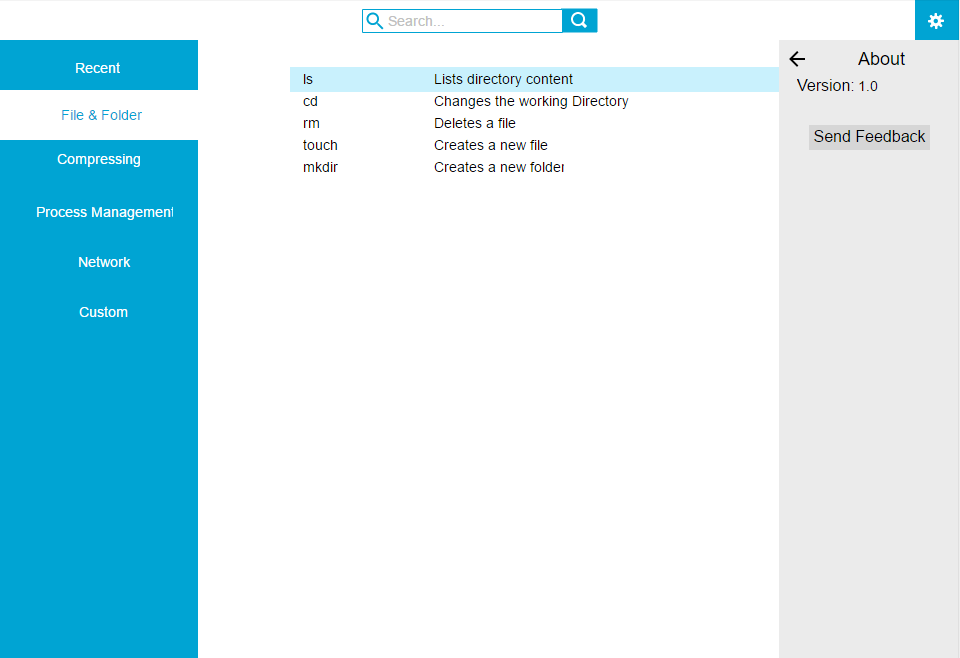
* Page showing the result from searching for a command



* Page showing that no result was found.

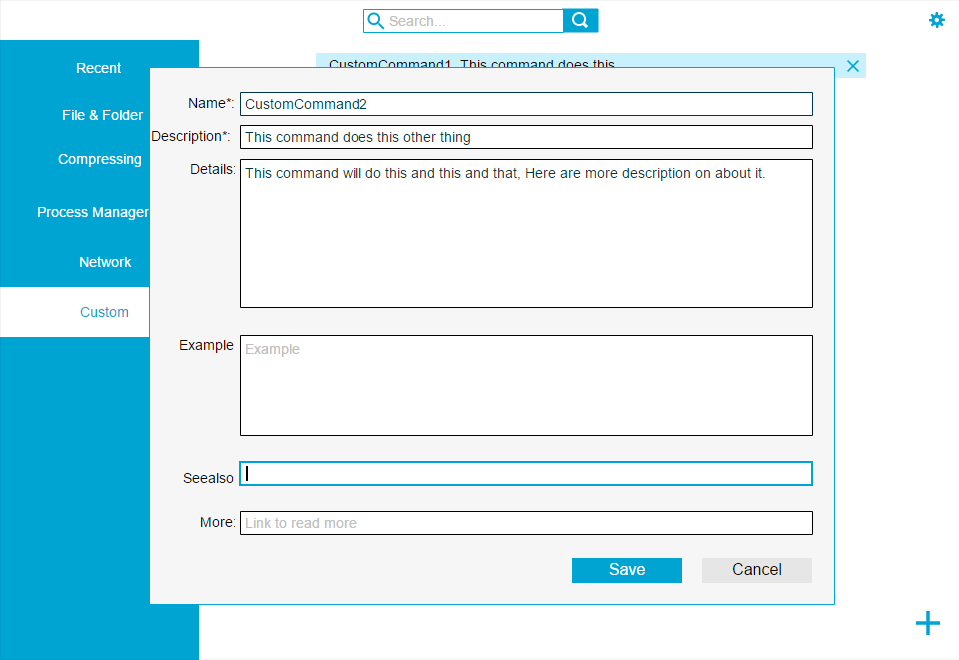


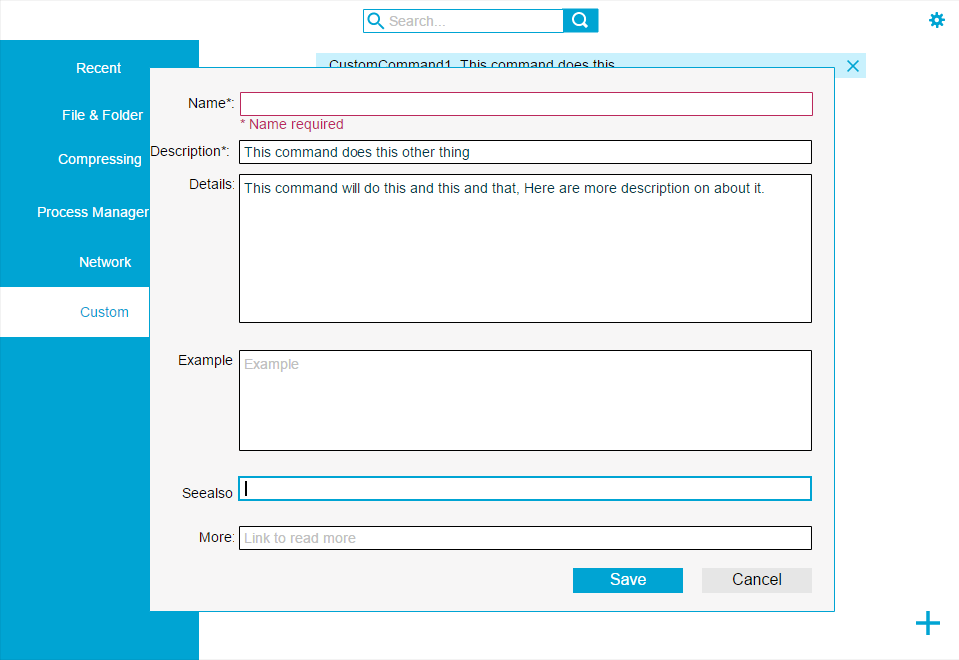
* The options available in settings (Only clearing recent category was implemented)

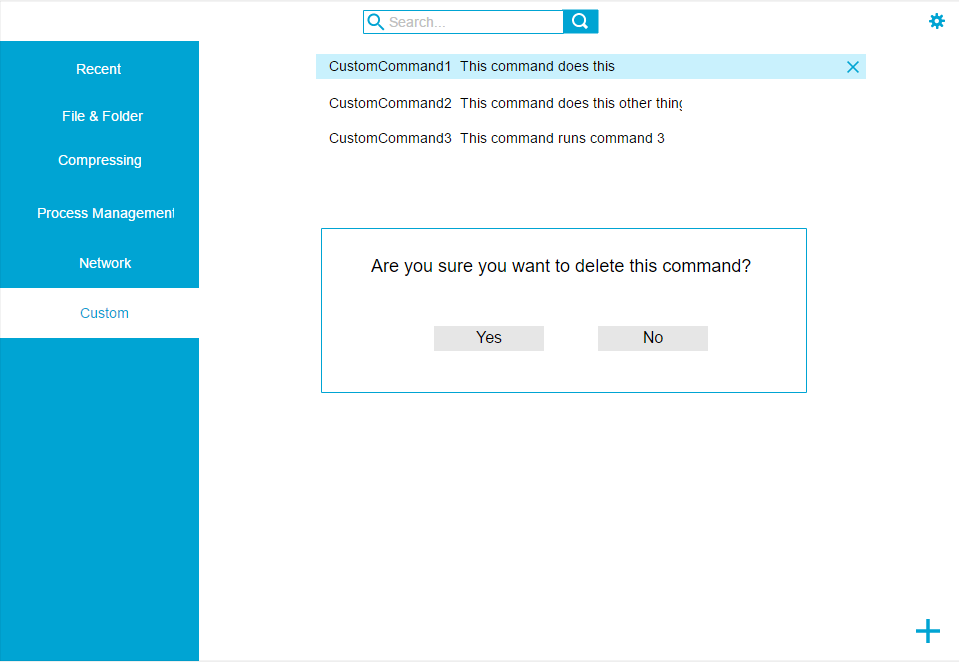
Some unimplemented features

* Adding new and editing a command to the custom category

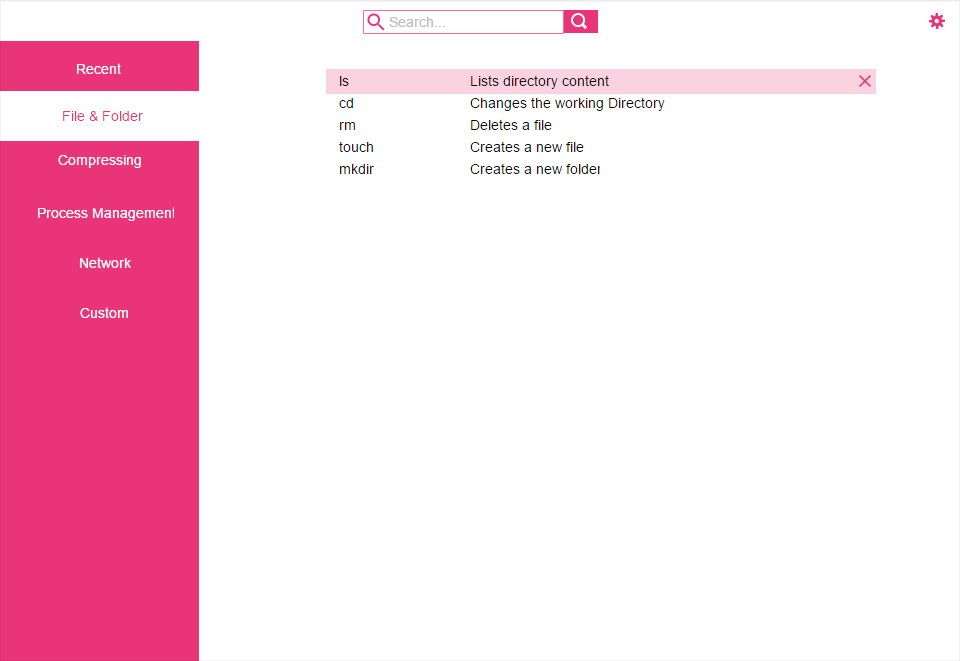




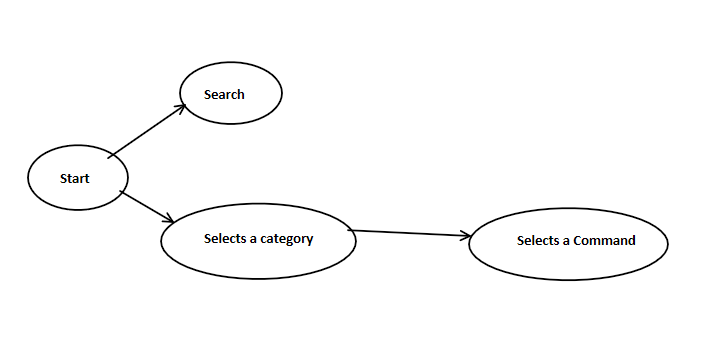
* Deleting a custom command



* Changing the system color



**Formal Specification: State Diagram**



**Initial testing:**

I started out by design and laying out all of the pages. Once done, I simulated multiple scenarios using those designs and everything workout great. Nothing needed to be changed since I didn’t start test until the designs were done.

**Implementation and testing:**

In the initial design, once the user selected a command and all the details about that command was displayed, the user had no way of telling what category he or she was currently in. To fix this problem, I added a title label in the top left corner, next to the back button, to let the user know which category, he or she is currently seeing. In the case of the user being in the main page the title just says “Home”.

The attributes of the command task object changed a little. The first change is that, the see Also section was removed due to complications about saving commands and not have duplications. The second, a formats list and options list were added. The formats list is just a list that holds different formats of the usability of the command, and the options list holds a list of options available for the use of the command e.g. -l option for the “ls” command.

**Design Rationale:**

The root page is broken down into two sections. The top which is a menu bar and the bottom part which takes up the remaining space and is a container for other pages that shows the data read from the xml file. The menu bar contains a back button that is only visible once the user navigates away from the home data page, a title next to the back button to let the user know what page/category he or she is currently seeing, a textbox to get input from the user to perform a search, a button to initiate the search and finally and toggle button to show and hide a settings panel.

The search functionality is implemented for users that knows what command he or she is looking for or remembered what it started with but don’t remember the whole command name. If an exact command is found the detailed view of the command is shown, still showing the list of other commands in the same category. If there is more than 1 result, then they are all displayed for the user to choose. If nothing was found, then the user is notified and given an option to search online for more result. For expert users, a shortcut (ctrl + F) will instantly give focus to the search textbox where the user can type and hit enter/return to perform the search. This allows the more advanced users to very quickly find the commands that he or she is looking for.

The data pages are based on a master/details design. On the main data page (Home page), the left side contains the list of categories each of which are selectable. On the right, as the user selects a category, the right side shows the list of commands in that category, each displaying the name and a short description of the command. The reason for showing the name and a short description is for expert users who don’t need to see all the details of a command and/or simply want to remind themselves of what that command does. For people who are new to Linux and students, a command can be clicked on to view more details about it.

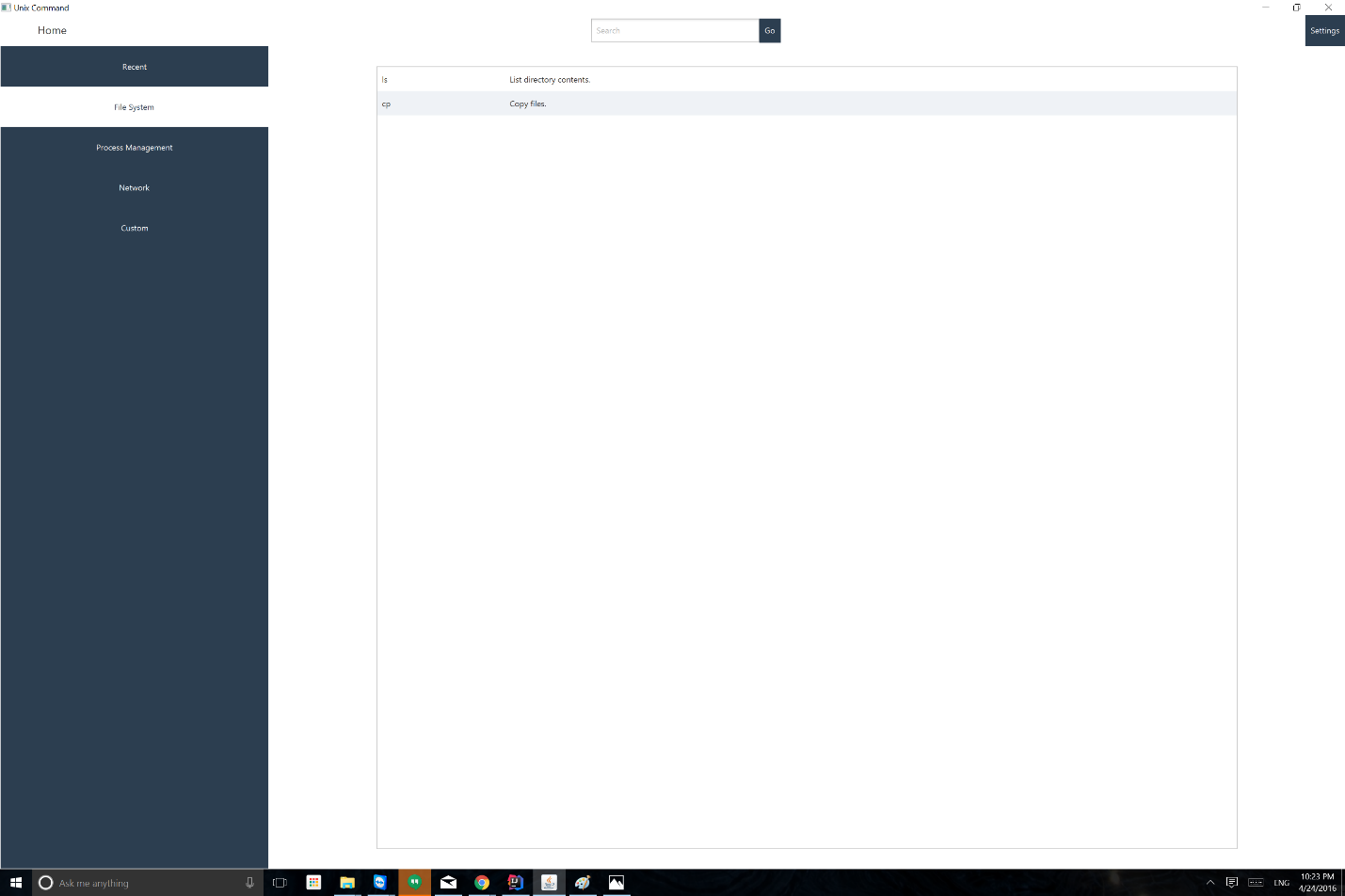
Once the user clicks on a command to learn see more details, a new page is displayed. That page is also based on a master/details design. On the left side, the list of all commands in the same category is displayed and can be selected. The reason is so that the user doesn’t have to go back in order to select a different command that is in the same category. On the right, information about the selected command is shown. That includes the name, short description, detailed description, all the different options e.g. -l option for the “ls” command, available for this command, formats on the usability of the command, examples on how to use the command and a link to a webpage where the user can get even more details about the command. In some cases, there might be a lot of white space on the right side because the list views that contains the options, examples and formats gets their sizes based on the number of items in them. So the more of those items the more the page is going to be filled. If there are too many items to fit all on the screen, the user will be able to scroll up and down.

When the program starts, a “Recent” category is added to the list of categories. Once the command objects are loading from the xml file, using the recently viewed attribute for each command, commands are added to “Recent” category. As the user views more commands, they are also added to this category without duplication. When the program is about to close, those attributes gets saved to the xml file. This category can be cleared under personalization in settings.

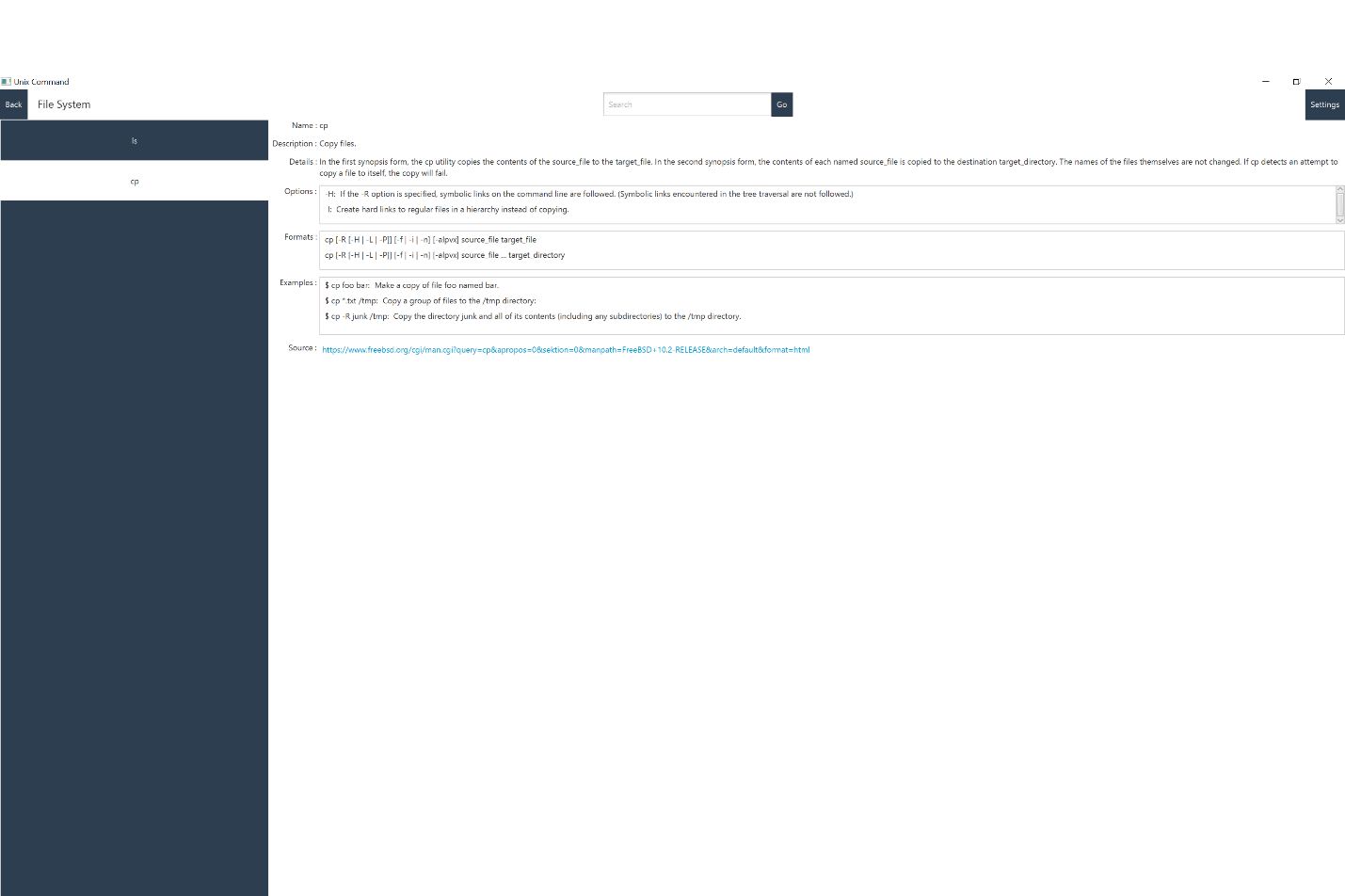
Some unimplemented features include the ability to add and edit a custom command using a form, and delete a command after the user confirms the deletion. All of those command would be added to the “Custom” category that is added to the list of categories when the program starts and saved before the program exits. Another feature is to allow customization of the system for the for the user, mainly being able to change the theme color of the program.

**Working Program:**

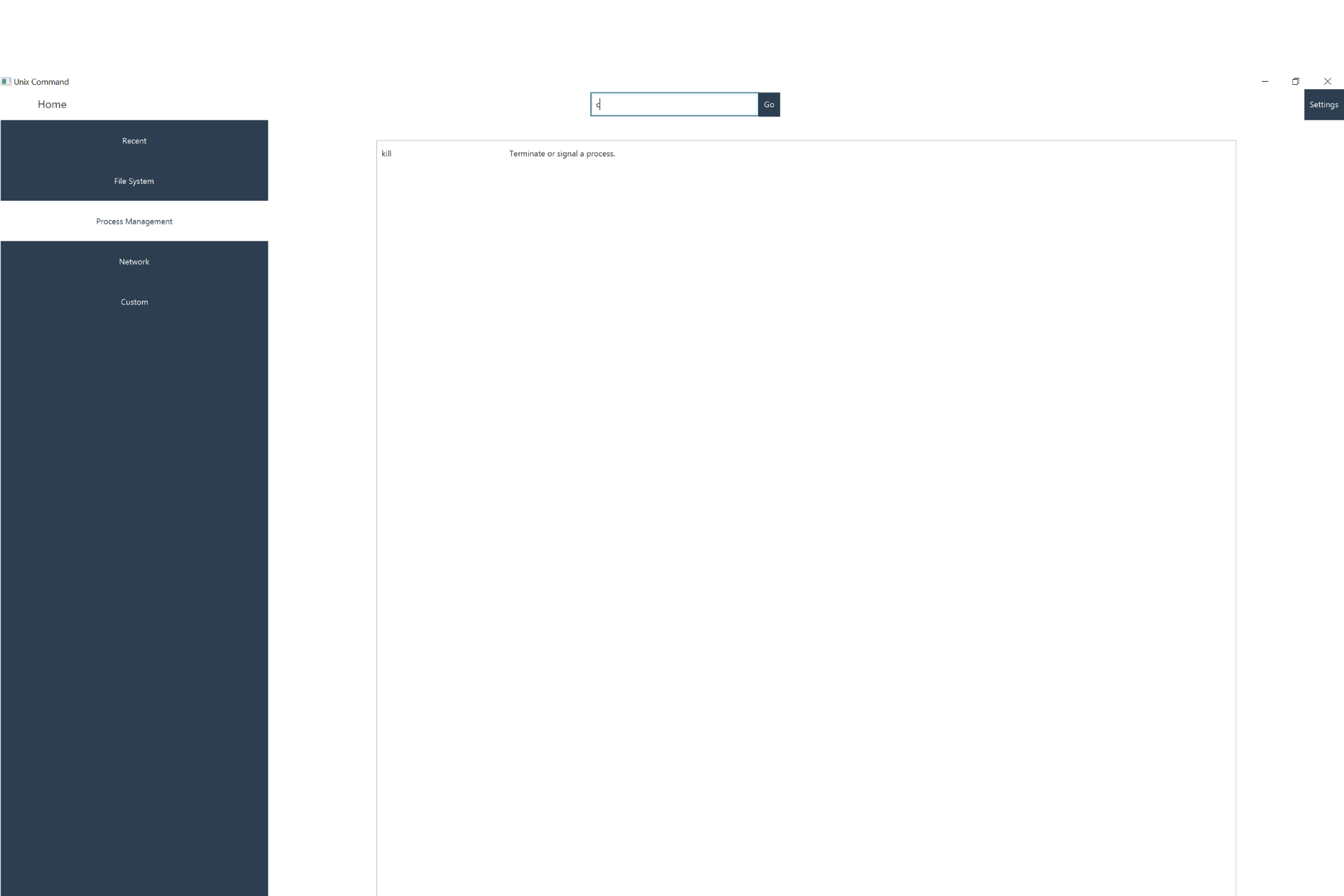
This is the main screen, showing the File System category selected. On the right, the list of commands for that category.



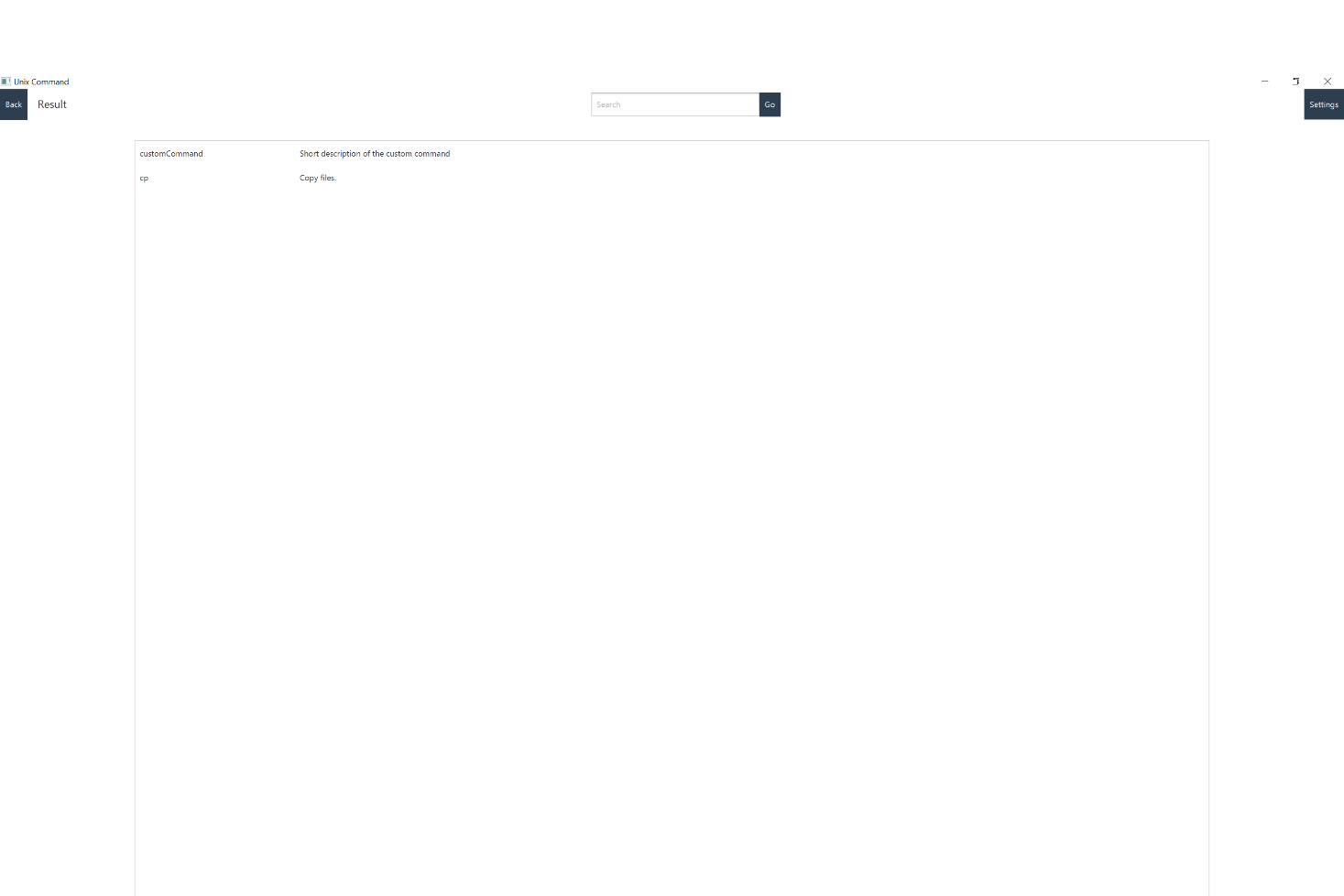
On this screen, on the left, the list of all the commands in the same category is displayed, and on the right, the details of the selected command are shown. In this picture there is a lot of white space on the right. That is because each individual ListView containing the list of options, the list of examples and the list of formats resizes as items gets added. Since there aren’t a lot of each item, then there is a lot of white space.



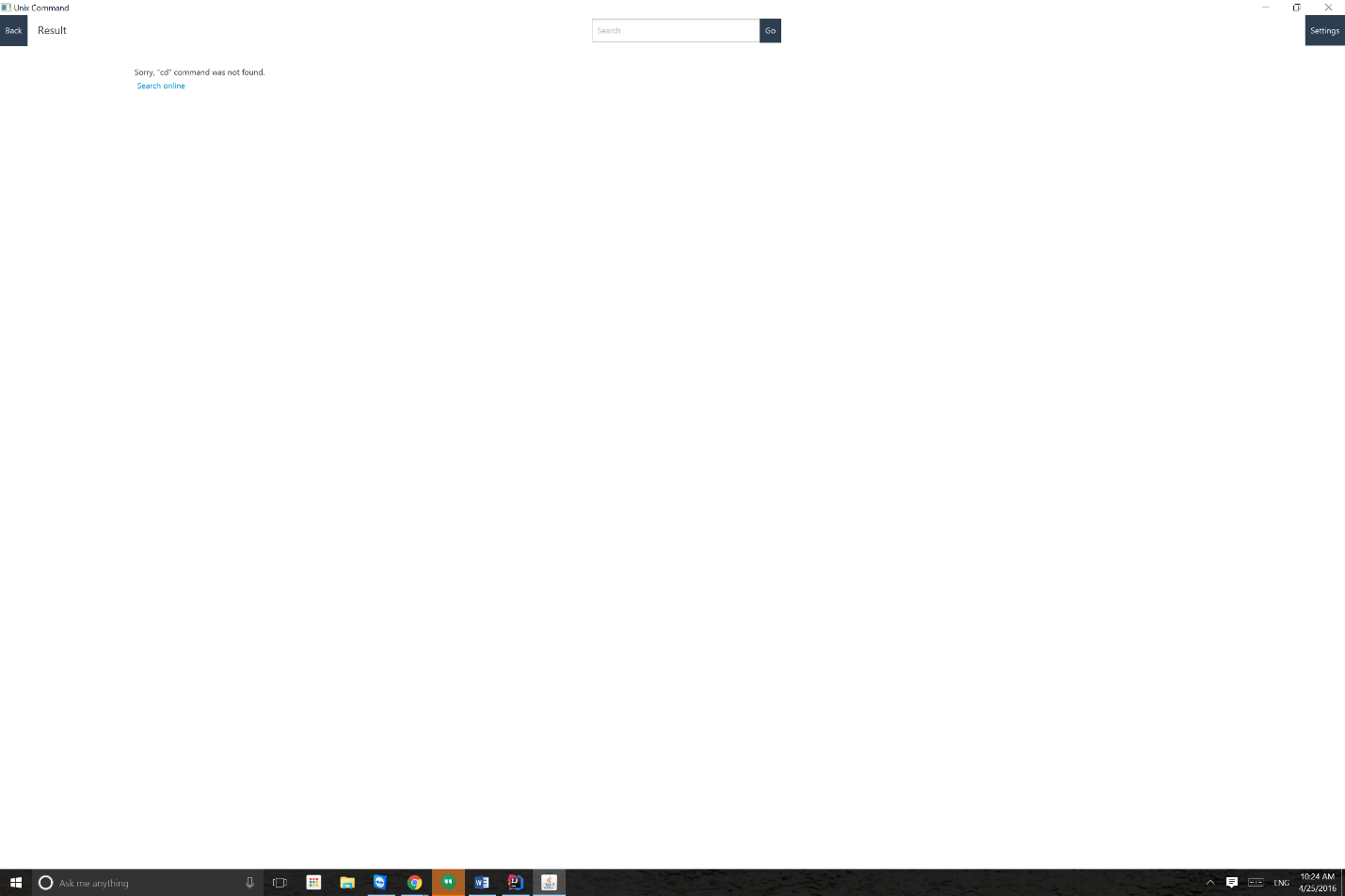
This shows the screen as the user types a command to be searched.



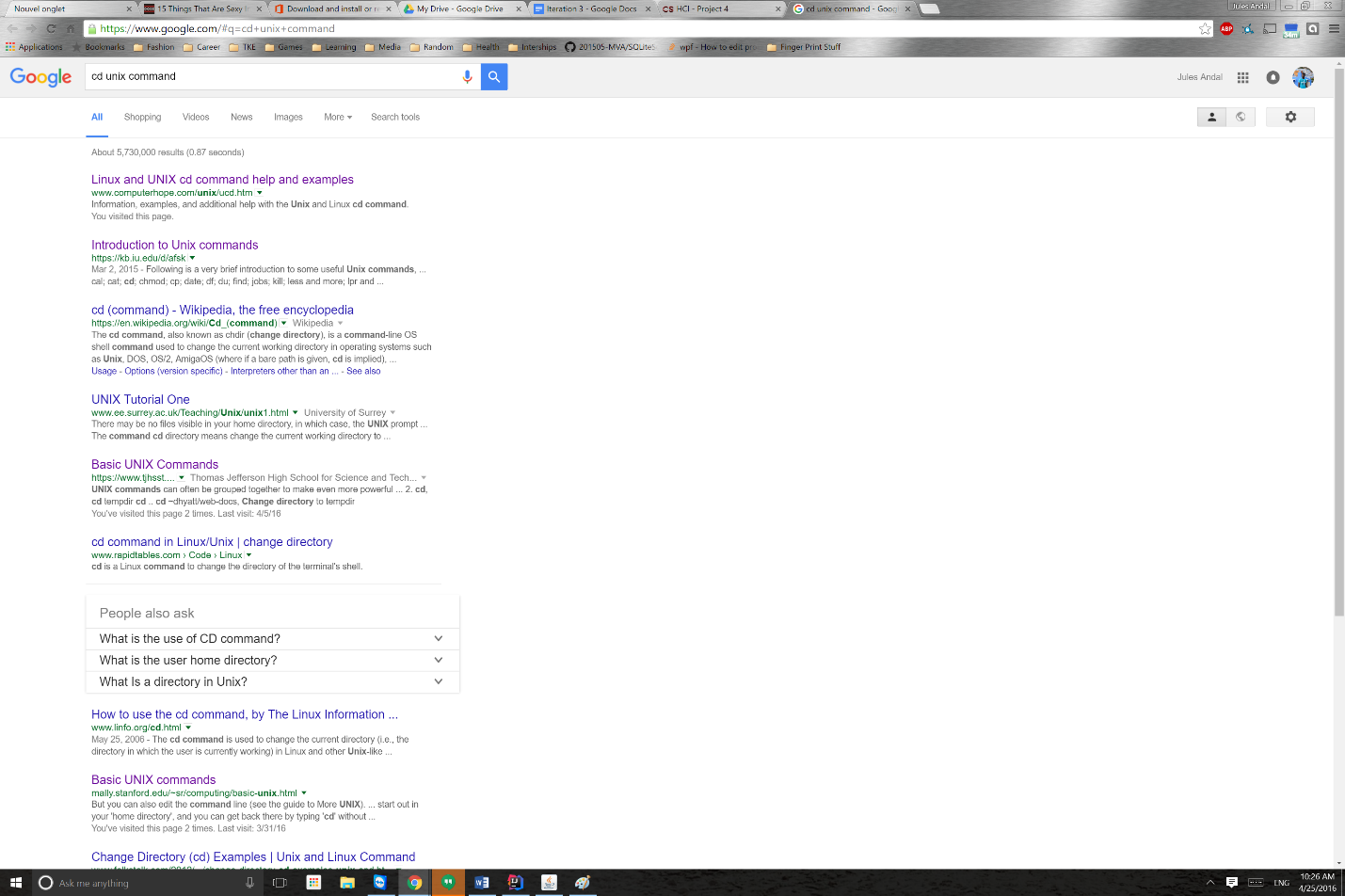
This screen shows the result when there are multiple commands found form the searched



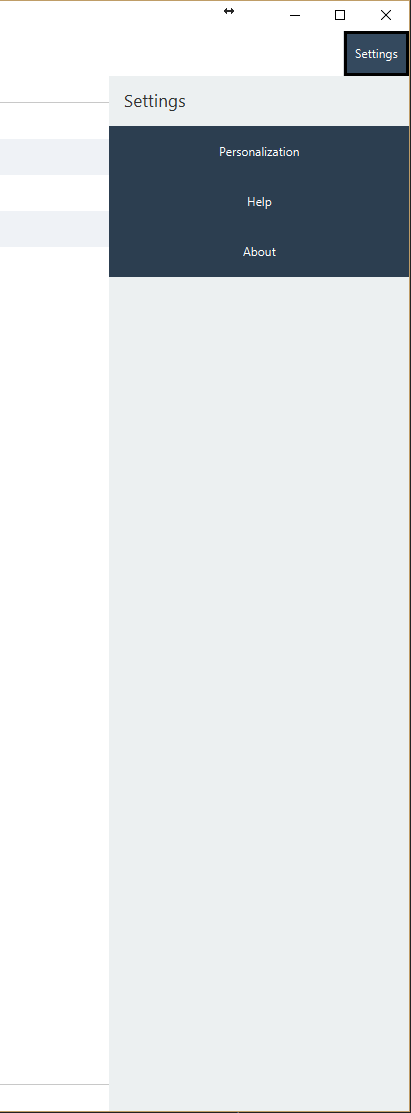
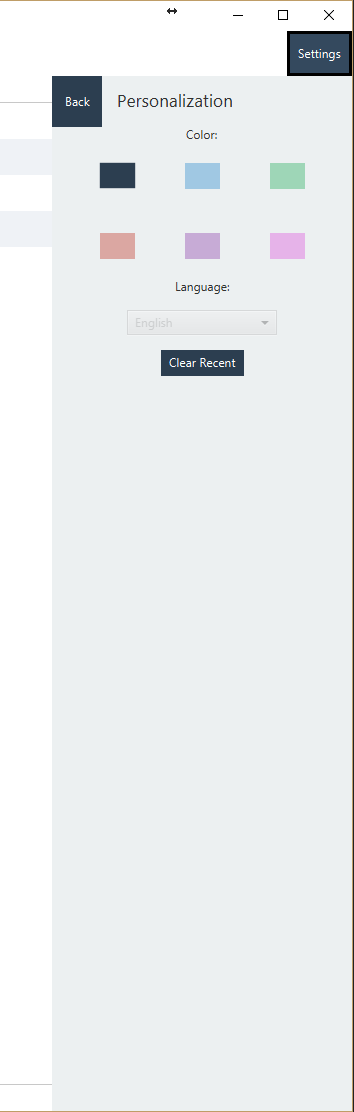
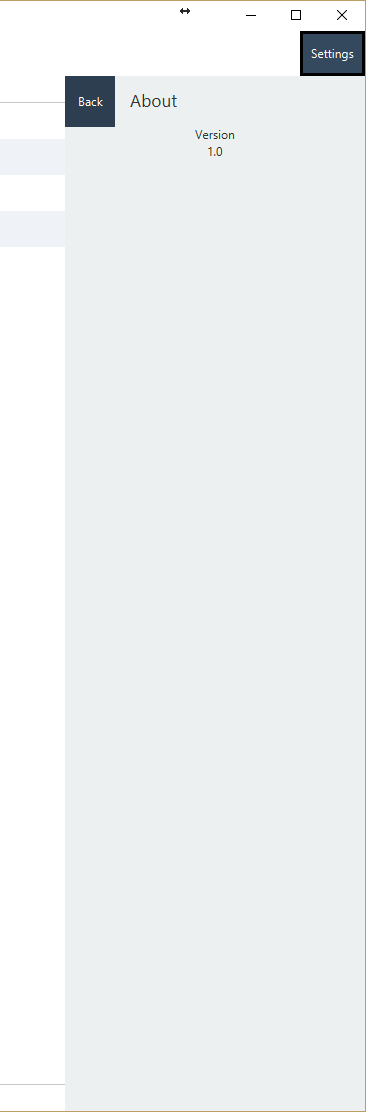
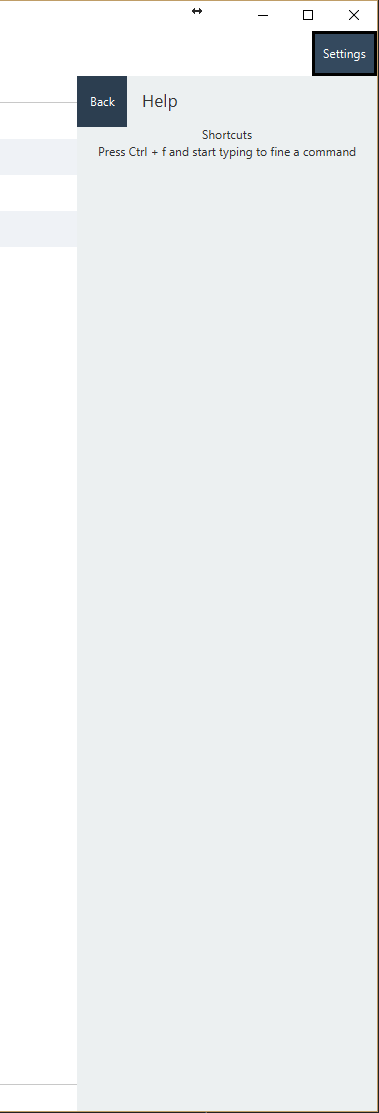
This screen is shown when no result was found from the search.



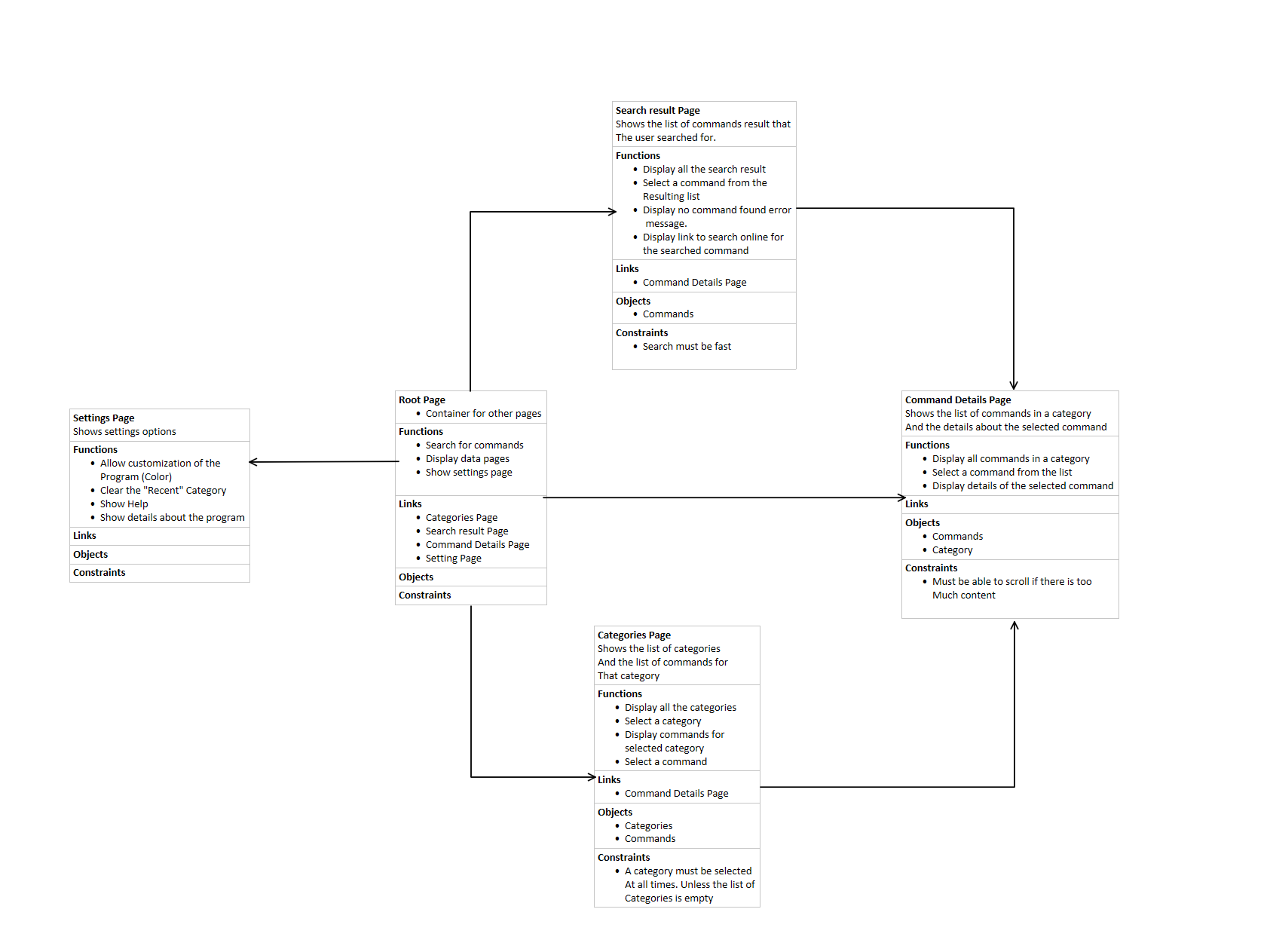
If the user clicks on search online link



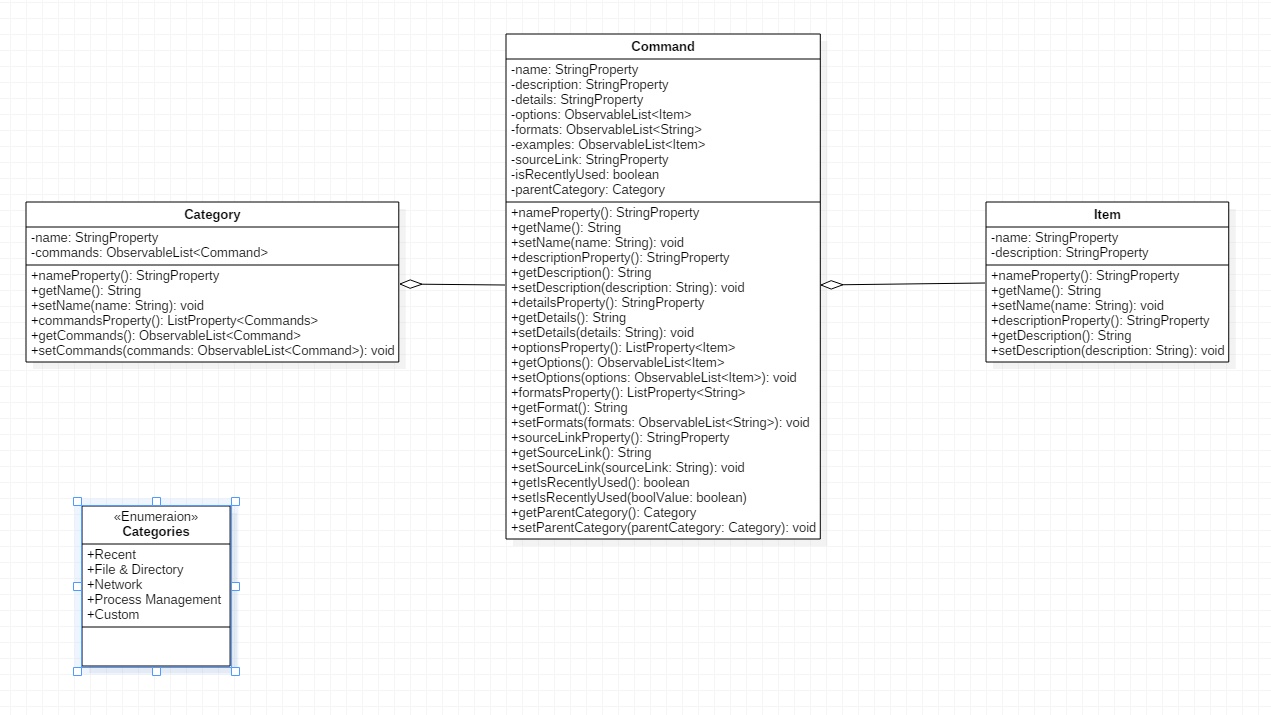
Settings Options

**Content Diagram:**

****

**Class/Model Diagram:**

****