John Caley

Application Documentation: Virtual Key

**Purpose of the Application**:

The purpose of this application is to develop a file managing application using the Agile development process. The project requires the typical features of a file manager to be implemented; searching, insertion, deletion, etc.

**Flow of the Application**:

When the application begins, the welcome screen and main menu will be displayed for the user. From there the user will have the option to get a list of all the files in the directory, open the additional operations menu or to close the application. From the additional operations menu the user will have the option to add a new file, delete an existing file, search for an existing file or to return to the main menu. A flow diagram, “FlowDiagram.jpg” is provided in the accompanying documentation.

**Concepts Implemented**:

The application utilizes the class “Java.io.File” to read and write files from the directory. The path to the working directory is stored as a static String variable using the directory’s relative path. When a file needs to be read to written to, an instance of the class “File” is created using the String variable listed previously as well as the name input by the user.

An array list is used when listing all the files in the working directory. When the “File” object has it’s list() method called, the result is stored in a string array. The application then differentiates between files and subdirectories and stores each in its own corresponding array list.

A regular expression check is utilized in two instances in the application. The first is to differentiate between files and subdirectories when listing the contents of the working directory (files will contain a number of characters followed by a period and a few more characters at the end.) The second is used when creating a new file; if the file name taken from the user’s input doesn’t follow the criteria for a file listed above, the application inform the user the input is invalid before returning the user to the additional options menu.

Finally, exception handling is utilized whenever the application needs to read or write from the working directory. This is to ensure the application doesn’t crash in the event that the directory cannot be found for whatever reason.

**Sprints**:

*Sprint 1*

Duration: 3 Days

Purpose:

The purpose of this sprint was to first create the welcome screen displaying the application name and developer details as well as create a main menu which offered the user the option to display the files in the working directory or close the application. The method which displays the files distinguish between files and subdirectories using a regular expression checking for a file extension. The implementation of this sprint can be seen in “sprint1.png”.

*Sprint 2*

Duration: 3 Days

Purpose:

The purpose of this sprint was to implement an additional operations menu which then gave the user the option to add a new file to the directory or to return to the main menu. When the user choses to add a new file, the application prompts the user to input the file name. The implementation of this sprint can be seen in “sprint2.png”

*Sprint 3*

Duration: 3 Days

Purpose:

The purpose of this sprint was to implement an option to delete a file in the additional operations menu. When this option was selected, the user would be prompted to input the file name. The application would then check to make sure the file exists in the directory. If not a message would display telling the user the file doesn’t exist before return to the additional operations menu. If the file does exist the application would delete the file, confirm to the user the file was deleted and then return to the additional operations menu. In addition to implementing this new feature, this sprint also makes some cosmetic changes to the menu displays so it’s easier for the user to distinguish between the main menu and the additional operations menu. The implementation of this sprint can be seen in “sprint3.png”

*Sprint 4*

Duration: 3 Days

Purpose:

The purpose of this sprint was to implement the final feature of the program; the search feature. From the additional operations menu the user may select to search for a file in the directory. From there the user may input a file name and the application will check if the file exists. If so, the application will inform the user the file does indeed exist. If not, the application will likewise inform the user the file doesn’t exist. The user will then return to the additional operations menu. In addition to implementing this search feature, this sprint also fixes a minor bug where if the user made an invalid selection in the additional operations menu, the program would close. Now the user will remain on the additional operations menu should an invalid selection be made. The implementation of this sprint can be seen in “sprint4.png”