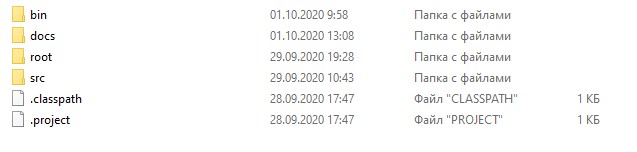
**VIRTUAL KEY FOR YOUR REPOSITORIES PROJECT**

**JAVA PROJECT AUTHOR**

Dmitry Koryanov

**JAVA PROJECT DESCRIPTION**

The root directory of the java project is “Phase1” (Screen 0 image). This directory has “bin” folder with Java code compiled, “src” folder with Java source code, ”root” folder for user files storage, “docs” folder for the screenshots & project description document.



Screen 0. Phase1 Folder Structure.

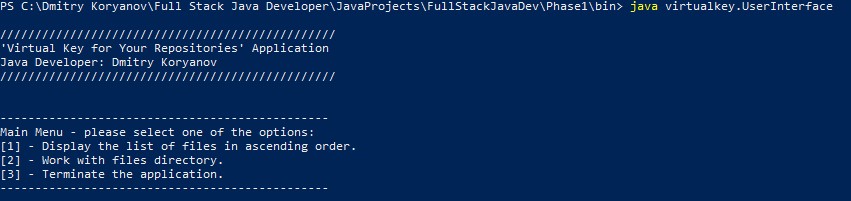
To run the application it is necessary to clone the repository from GitHUb <https://github.com/dmitrykoryanov/FullStackJavaDev> , enter “Phase1/bin”, start command line from this folder & execute “java virtualkey.UserInterface” in the command line.

The application has two menus to work with, according to the problem statement. The first menu (Screen 1 image) has 3 options:

[1] - Display the list of files in ascending order.

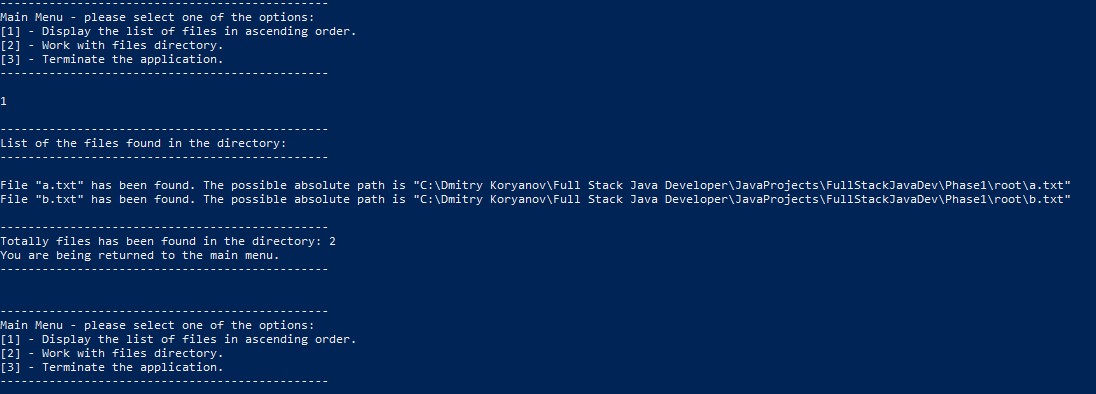
[2] - Work with files directory.

[3] - Terminate the application.



Screen 1. The main menu of the application.

Option [1] displays files in the “Phase1/root” folder in the ascending order (Screen 2 image). The total number of found files is also being displayed. Option [2] allows to work with the next menu of the application. Selection the option [3] allows user to terminate the application.



Screen 2. The list of files in the ascending order is being displayed.

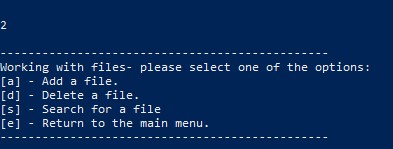
The second menu (Screen 3 image) to work with files has the following options:

[a] - Add a file.

[d] - Delete a file.

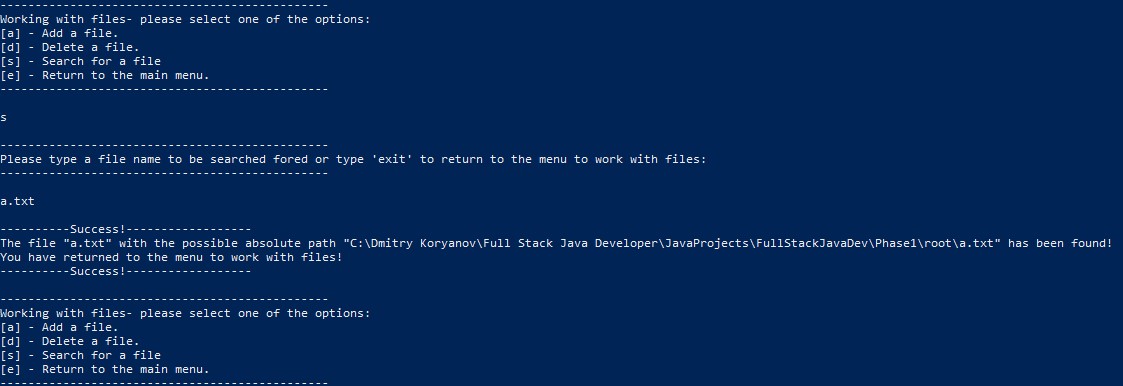
[s] - Search for a file

[e] - Return to the main menu.



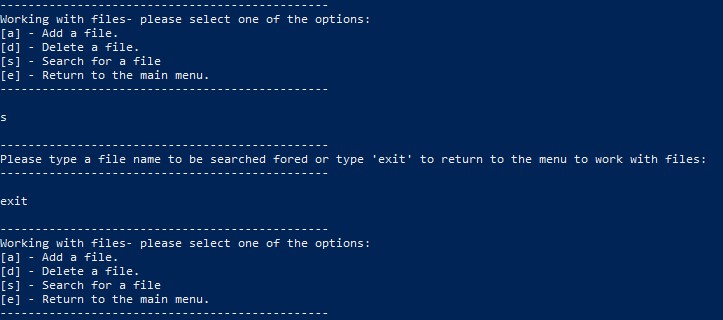
Screen 3. The second menu of the application – working with files.

Option [e] returns the user to the main menu. Options [s] (Screen 4 image) allows a user to search a for a file. Either if the file is found or not, the user gets a corresponding message in the screen, after that the menu to work with files is being displayed again.



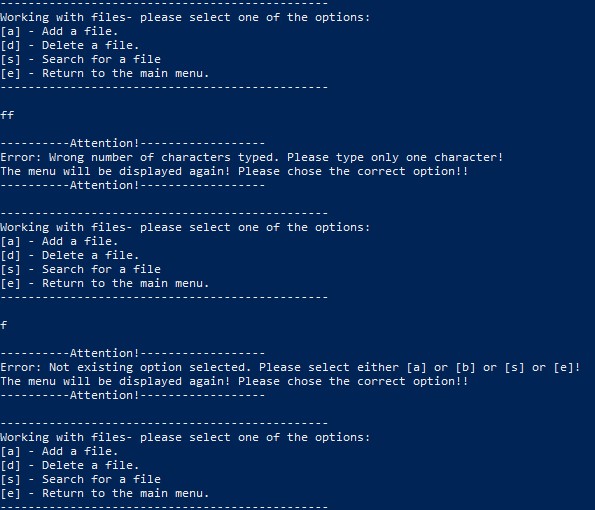
Screen 4. Searching for a file.

In case user has changed his/her mind while being inside options [a], [d] or [s], he/she can type “exit” to return to the menu to work with files.



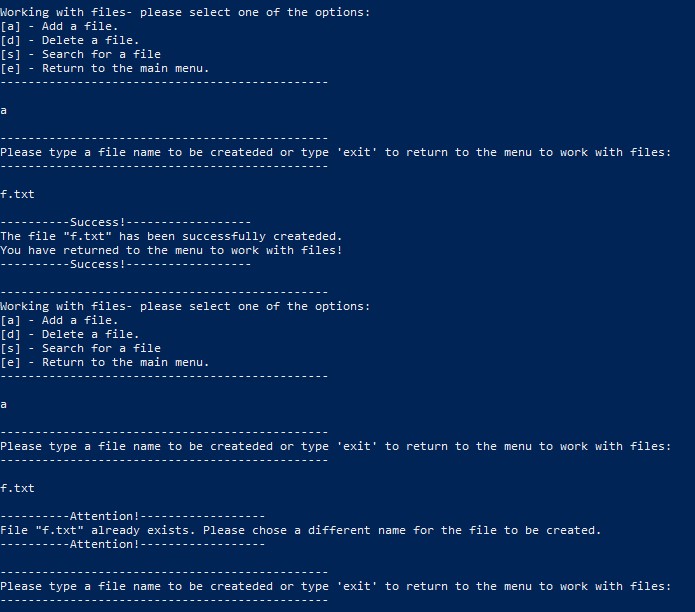
Screen 5. There is a possibility to exit from options [a]. [d] or [s] without performing corresponding actions.

In case the user chooses not existing option, or enters some rubbish, a corresponding message is being displayed to him/her (Screen 6 image). The same relates to the main menu of the application.



Screen 6. The messages being displayed in case user chooses a wrong option. The similar messages are being displayed in the main menu.

Option [a] allows user to add an empty file to the “Phase1/root” folder with the specified name. Upon the successful result a corresponding message is being displayed and user is being returned to the menu, in case of a failure the user is being suggested to choose a different file name to be added (Screen 7 image).



Screen 7. Upon successful addition of a file to the directory user gets a corresponding message and being returned to the menu. In case of a failure user is being suggested to choose a different file name.

Option [d] works in a similar way, as option [a]. It allows user to delete a file from a file storage (“Phase1/root” folder) with the specified name. In case of a success a corresponding message is being displayed and user is being returned to the menu, in case of a failure user is being suggested to choose a different file name (Screen 8 image).

**JAVA ALGORITHMS USED**

The case sensitivity for working with files is being inherited from the file system. In case the file system is case insensitive (Windows), the application would not distinguish case sensitivity of file names, and vice versa for UNIX.

The source java code of the application is well documented, please refer to it for the details of Java constructions and algorithms used in the application.

**SPRINTS**

The application has been developed in two sprints. In the first sprint graphical interface (menus) were developed, without implementing the logic of corresponding actions. In the second sprint the code has been fully completed and commented, along with the project description. The same can be tracked in GitHub repository.