LockedMe – Virtual Key for Repositories

The code for this project is hosted at https://github.com/nrai06/-LockedMe-Application-
The project is developed by Neha Rai

Sprints planning and Task completion

Project work divided into 2 Sprint 10 days, Sprint work details are –

1st – Sprint Work Details -

- Initializing git repository to track changes as development progresses.
- Create Java class to show the files from the folder in ascending order.
- Create Java class for Option to add a user specified file to the application.
- Create Java class for Option to search a user specified file from the application with case sensitive.
- Create Java class for Option to read a user specified file from the application with case sensitive.
- Push the Java class in the git repository.

2nd – Sprint Work Details –

- Create Java class for Option to delete user specified file from the application with case sensitive.
- Create Main Java class as a welcome class to Navigation option to close the current execution context and return to the main context.
- Welcome Class is the main class to execute all the classes.
- Test Again All the Java class.
- Migrate all the Java classes to the welcome class and execute the application.
- Test applications make the required changes.
- Push the Java class in the git repository.

Core concepts used in project

Collections framework, File Handling, Sorting, Flow Control, Exception Handling, Loop, Algorithm, sorting etc.

Java Classes to use in project –

1. **WelcomeScreen Class -** Used to welcome the user, Main menu and File operation menu.
2. **ShowFile Class -** Used to Show Files names in an ascending order present in the directory.
3. **AddFile Class -** Used to create a new file, input taken from user for filename and some write up in file.
4. **SearchFile Class** - Search a user specified file from the main directory.
5. **DeleteFile Class** -Delete a user specified file from the existing directory list.
6. **WriteFile Class** - Write the content of the file from the existing directory list.

Details Concept for Each class with code and Result –

1.WelcomeScreen Class -

- First this class displays the welcome screen and then Application name and the developer details.
- The details of the user interface such as options displaying the user interaction information
- Features to accept the user input to select one of the options listed.
- The first option should return the current file names in ascending order. The root directory can be either empty or contain few files or folders in it.
- The second option should return the details of the user interface such as options displaying the following.
    i.Add a file to the existing directory list
    ii.Delete a user specified file from the existing directory list
    iii.Search a user specified file from the main directory
    iv. ShowAllFile to Files names in an ascending order present in the
    v.Option to navigate back to the main context.
- After Performing operation it will return to the main menu if you want to continue or exit .
- Don't want to continue type e and it will exit the application .

1)Code for WelcomeScreen Class

```
1  package userInterface;
2
3  import java.util.Scanner;
9
10 public class WelcomeScreen {
11
12
13     public static void main(String[] args) {
14
15         System.out.println("*********************************************************************************");
16         System.out.println();
17         System.out.println("                    Welcome to LockedMe  Application                    ");
18         System.out.println();
19         System.out.println("                             Developed by : Neha Rai                    ");
20         System.out.println("                          Email Id:nr506066@gmail.com                   ");
21         System.out.println();
22         System.out.println("*********************************************************************************");
23
24
25
26         while(true) {
27
28
29             System.out.println("Enter your Choice ");
30
31             System.out.println("1. Show Files names in an ascending order present in the directory   ");
32             System.out.println("2. File Menu options");
33             System.out.println("3. Exit");
34             System.out.println();
35
36
37             Scanner input = new Scanner(System.in);
38             int choice = input.nextInt();
```

```
40
41              if(choice==1) {
42                  ShowAllFiles sf = new ShowAllFiles();
43                  sf.show();
44
45              }
46              else if(choice==2) {
47                  while(true) {
48                      System.out.println(" ********************** FILE MENU OPTION  ********************** ");
49                      System.out.println();
50                      System.out.println("a : Add a new File");
51                      System.out.println("b : write in  a new File");
52                      System.out.println("c : Delete a file");
53                      System.out.println("d : Search a file");
54                      System.out.println("e : Exit (from file menu)");
55                      System.out.println();
56                      System.out.println(" ************************************************************** ");
57
58                      String choice1=input.next();
59
60                      if(choice1.equals("e")) {
61                          System.out.println(" Moved to the main menu ");
62                          System.out.println();
63                          break;
64                      }
65                      switch(choice1) {
66
67                      case "a":
68                          System.out.println(" Enter file name to be added in the folder ");
69                          AddFile af=new AddFile();
70                          af.createFile();
71                          break;
72
73                      case "b":
74                          System.out.println(" Enter file name to written in the folder ");
75                          WriteFile wf=new WriteFile();
76                          wf.writefile();
77                          break;
78
79                      case "c":
80                          System.out.println(" Enter file name to be deleted from the folder ");
81                          DeleteFile df= new DeleteFile();
82                          df.delete();
83                          break;
84
85
86                      case "d":
87                          System.out.println(" Enter file name to be seached ");
88
89                          break;
90
91
92                      }
93
94                  }
95
96              }
97              else if(choice==3) {
98                  System.out.println(" Thanks for using Lockedme.com services. ");
99                  System.out.println();
100                 break;
101
102             }
103             else {
104                 System.out.println(" Please Enter a Correct Choice ");
105                 System.out.println();
106             }
```

OutPut Result for WelcomeScreen Class -Welcome to user ,Main menu and FileOperation menu -

```
**********************************************************************************

                    Welcome to LockedMe  Application

                            Developed by : Neha Rai
                          Email Id:nr506066@gmail.com

**********************************************************************************
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

2
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ******************************************************************
```

2)Code for ShowAll Files

```java
package operations;

import java.io.File;
import java.util.Arrays;


public class ShowAllFiles {

    public void show() {
        // Creates an array in which we will store the names of files and directories
        String[] pathnames;

        // Creates a new File instance by converting the given pathname string
        // into an abstract pathname
        File f = new File("C:\\Users\\nrai6\\eclipse-workspace\\LockedMe\\Files");

        // Populates the array with names of files and directories
        pathnames = f.list();
        Arrays.sort(pathnames);
        System.out.println("Files names in an ascending order ");
        // For each pathname in the pathnames array
        for (String pathname : pathnames) {
            // Print the names of files and directories

            System.out.println(" |- " + pathname);
```

OutPut Result for ShowAllFile Class -

```
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

1
Files names in an ascending order
 |- a.txt
 |- d.txt
 |- emp.txt
 |- lockedme.txt
 |- s.txt
 |- vx.txt

Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit
```

3) Code for Add file

```java
package operations;

import java.io.File;

import java.io.IOException;
import java.util.Scanner;

public class AddFile {
    public void createFile() {

        try {

            Scanner filename = new Scanner(System.in);
            String filename1 = filename.next();

            File file = new File("C:\\Users\\nrai6\\eclipse-workspace\\LockedMe\\Files\\" + filename1 );
            if (file.createNewFile()) {
                System.out.println("File created: " + file.getName());
            } else {
                System.out.println("File already exists.");
            }
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }

}
```

OutPut Result for AddFile

```
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

2
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ********************************************************************
a
 Enter file name to be added in the folder
test.txt
File created: test.txt
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ********************************************************************
```

4)Code for WriteFile

```java
package operations;

import java.io.FileWriter;

public class WriteFile {

    public void writefile() {

        try {
            System.out.println("Please enter file name to write with extension, ex- filename.txt");
            System.out.println();
            Scanner filename = new Scanner(System.in);
            String filename1 = filename.next();

            FileWriter myWriter = new FileWriter(" C:\\Users\\nrai6\\eclipse-workspace\\LockedMe\\Files " + filename1);
            System.out.println("Enter your message");
            Scanner msg=new Scanner(System.in);
            String msg1=msg.nextLine();

            myWriter.write(msg1);
            myWriter.close();
            System.out.println("Successfully wrote to the file.");
        }
        catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
```

OutPut Result for WriteFile

```
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

2
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ****************************************************************
b
 Enter file name to written in the folder
test.txt
Enter your message
Hello lockedme Application
Successfully wrote to the file.
```

5)Code for DeleteFile

```java
package operations;

import java.io.File;

public class DeleteFile {

    public void delete() {

        Scanner filename = new Scanner(System.in);
        String file = filename.next();

        File file1=  new File("C:\\Users\\nrai6\\eclipse-workspace\\LockedMe\\Files\\" +file);
        if (file1.delete()) {
            System.out.println(" Your Have Successfully deleted the file " + file1.getName());
        } else {
            System.out.println("File Not Found -FNF .");

        }
    }
}
```

OutPut Result for Delete File

```
                    Welcome to LockedMe  Application

                            Developed by : Neha Rai
                          Email Id:nr506066@gmail.com

********************************************************************************
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

2
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ***************************************************************
c
 Enter file name to be deleted from the folder
d.txt
 Your Have Successfully deleted the file d.txt
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ***************************************************************
c
 Enter file name to be deleted from the folder
neha.txt
File Not Found -FNF .
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)
```

6)Code For Search File

```java
package operations;

import java.io.File;

public class SearchFile {

    void search() {


        Scanner filename = new Scanner(System.in);
        String file = filename.nextLine();

        File file1=  new File("C:\\Users\\nrai6\\eclipse-workspace\\LockedMe\\Files\\" +file);
        if(file1.exists()) {
            System.out.println(file1.getName() + " is found");
        }else {
            System.out.println("The file does not exist");
        }
```

OutPut Result for Search File

```
********************************************************************************
Enter your Choice
1. Show Files names in an ascending order present in the directory
2. File Menu options
3. Exit

2
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 ****************************************************************
d
 Enter file name to be seached
a.txt
a.txt is found
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)
```

```
 **********************************************************************
d
 Enter file name to be seached
n.txt
The file does not exist
 ********************** FILE MENU OPTION  **********************

a : Add a new File
b : write in  a new File
c : Delete a file
d : Search a file
e : Exit (from file menu)

 **********************************************************************
```

Pushing the code to GitHub repository

● Open your command prompt and navigate to the folder where you have created your files.
cd <folder path>
● Initialize repository using the following command:
git init
● Add all the files to your git repository using the following command:
git add .
● Commit the changes using the following command:
git commit . -m <commit message>
● Push the files to the folder you initially created using the following command:
git push -u origin master

Unique Selling Points of the Application

- The application is designed to keep on running and taking user inputs even after exceptions occur. To terminate the application, appropriate option needs to be selected.
- The application can take any file/folder name as input. Even if the user wants to create nested folder structure, user can specify the relative path, and the application takes care of creating the required folder structure.
- User is also provided the option to write content if they want into the newly created file.
- The application doesn't restrict user to specify the exact filename to search/delete file/folder. They can specify the starting input, and the program searches all files/folder starting with the value and displays it. The user is then provided the option to select all files or to select a specific index to delete.
- The application also allows users to delete folders which are not empty.
- The user is able to seamlessly switch between options or return to previous menu

even after any required operation like adding, searching, deleting or retrieving of files are performed.

- When the option to retrieve files in ascending order is selected, user is displayed with two options of viewing the files.
  i. Ascending order of folders first which have files sorted in them,
  ii. Ascending order of all files and folders inside the “main” folder.
- The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code. Application has been developed keeping in mind that there should be very less “hardcoding” of data.

Conclusions

Further enhancements to the application can be made which may include:

- Conditions to check if user is allowed to delete the file or add the file at the specific locations.
- Asking user to verify if they really want to delete the selected directory if it’s not
- empty.
- Retrieving files/folders by different criteria like Last Modified, Type, etc.
- Allowing user to append data to the file