|  |
| --- |
| **Company Locked**  **(Sprint work and Project Specification)** |

**Version History**

|  |  |
| --- | --- |
| **Author** | Ibrahim Sheik |
| **Purpose** | Screenshots of the application |
| **Date** | 11th Aug 2021 |
| **Version** | 1.0 |

Contents

[1.Modules in the Project: 3](#_Toc79662379)

[2.Java Technologies Used: 3](#_Toc79662380)

[3.Sprint wise work: 3](#_Toc79662381)

[Sprint Number 3](#_Toc79662382)

[Modules 3](#_Toc79662383)

[3.1.Display All Files: 3](#_Toc79662384)

[3.2.Create File: 3](#_Toc79662385)

[3.3.Delete File: 4](#_Toc79662386)

[3.4.Search File: 4](#_Toc79662387)

[3.5.Testing File: 4](#_Toc79662388)

[3.6.Deployment(Creating Jar File): 4](#_Toc79662389)

[4.Project Git Hub Link: 4](#_Toc79662390)

[Repository Name 4](#_Toc79662391)

[GitHub Link 4](#_Toc79662392)

[5.Project Code: 5](#_Toc79662393)

[1.Folder Structure: 5](#_Toc79662394)

[2.FileManger: 5](#_Toc79662395)

# 1.Modules in the Project:

1. Main Menu Screenshot
2. 2.Display All Files
3. Create A File
4. 4. Delete A File
5. 5. Search File
6. 6.Exit

# 2.Java Technologies Used:

1.Naming Standards

2.Exceptional Handling

3.Working with Files

4.Modularity

5.Object Oriented Programming

6.Collections

7.Control Structures

8.Data Structure

# 3.Sprint wise work:

|  |  |
| --- | --- |
| Sprint Number | Modules |
| 1 | Display All Files  Create File |
| 2 | Delete File  Search File  Exit |
| 3 | Testing File  Deployment(Creating jar File) |

# 3.1.Display All Files:

In this method all the files present in the directory is displayed.

# 3.2.Create File:

In Create File option user can create a new file .

# 3.3.Delete File:

In these user can delete any file in directory by selecting option Delete File.

3.4.Search File:  
User can search any file in the directory by using Search File Option.

3.5.Testing File:  
Testing File is done by complier to verify the code.by this we can rectify errors and can get output/result.

# 3.6.Deployment(Creating Jar File):

By creating a jar file we can run program in Command Prompt. Jar File can be run without any Development Kit Application. In Jar File user can access program easily.

# 

# 4.Project Git Hub Link:

|  |
| --- |
| Repository Name |
| **FirstPhaseProject** |
| GitHub Link |
|  |

# 5.Project Code:

|  |
| --- |
| 1.Folder Structure: |
|  |
| 2.FileManger: |
| **package** com.lockedme;  **import** java.io.File;  **import** java.io.FileWriter;  **import** java.util.ArrayList;  **import** java.util.List;  **public** **class** FileManger  {  /\*\*  \* This method return file names  \* **@param** folderpath  \* **@return** list<String>  \*/  **public** **static** List<String> getAllFiles(String folderpath)  {    //Creating File Object  File f1 =**new** File(folderpath);    //Getting all the files into FileArray  File[] listofFiles =f1.listFiles();    //Declare a list to store file names  List<String> fileNames =**new** ArrayList<String>();  **for**(File f:listofFiles)  {  fileNames.add(f.getName());  }  //return  **return** fileNames;  }  /\*\*  \* This method is for create content in files  \* **@param** folderpath  \* **@param** fileName  \* **@param** content  \* **@return** boolean  \*/    **public** **static** **boolean** createFiles(String folderpath,String fileName,List<String> content)  {  //create files in folder  **try**  {  File fl=**new** File(folderpath,fileName);  FileWriter fw=**new** FileWriter(fl);  **for**(String s:content)  {  fw.write(s+"\n");  }  fw.close();  **return** **true**;  }  //return false if file is not created  **catch**(Exception ex)  {  **return** **false**;  }  }  /\*\*  \* This Method is for Deleting files  \* **@param** folderpath  \* **@param** fileName  \* **@return** boolean  \*/  **public** **static** **boolean** deleteFile(String folderpath,String fileName)  {  //create path for deleting file  File file=**new** File(folderpath+"\\"+fileName);  **try**  {  **if**(file.delete())  **return** **true**;  **else**  **return** **false**;  }  **catch**(Exception ex)  {  **return** **false**;  }  }  /\*\*  \* This Method is for Searching files  \* **@param** folderpath  \* **@param** fileName  \* **@return** boolean  \*/  **public** **static** **boolean** searchFile(String folderpath,String fileName)  {  //create file search option and file object  File file=**new** File(folderpath+"\\"+fileName);  **if**(file.exists())  **return** **true**;  **else**  **return** **false**;  }  } |
| Company Locked Pvt.Ltd |
| **package** com.lockedme;  **import** java.util.ArrayList;  **import** java.util.List;  **import** java.util.Scanner;  **public** **class** CompanyProject  {  **static** **final** String ***folderpath***="D:\\MyFirstProject\\ProjectFiles";  **public** **static** **void** main(String[] args)  {  **int** proceed=1;  **do**  {  //variable declaration  Scanner obj =**new** Scanner(System.***in***);  **int** c;    //Menu  *displayMenu*();  System.***out***.println("Enter your choice:");  c=Integer.*parseInt*(obj.next());    **switch**(c)  {  **case** 1: *getAllFiles*();  **break**;  **case** 2: *createFile*();  **break**;  **case** 3: *deleteFile*();  **break**;  **case** 4: *searchFile*();  **break**;  **case** 5: System.*exit*(0);  **break**;  **default**: System.***out***.println("Invalid Option");  }    }  **while**(proceed>0);  }    **public** **static** **void** displayMenu()  {  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.***out***.println("\t\tCompany Locked Pvt.Ltd.");  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.***out***.println("1.Display all files\n2.Add new File\n3.Delete a File\n4.Search a File\n5.Exit");  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");    }    **public** **static** **void** getAllFiles()  {  //Getting File Name  List<String> fileNames=FileManger.*getAllFiles*(***folderpath***);  **if**(fileNames.size()==0)  {  System.***out***.println("NO Files exists in Directory");  }  **else**  {  System.***out***.println("Files List below :\n");  **for**(String f:fileNames)  System.***out***.println(f);  }  }    **public** **static** **void** createFile()  {  //Variable declaration  Scanner obj=**new** Scanner(System.***in***);  String fileName;  **int** linesCount;  List<String> content =**new** ArrayList<String>();    //Read file name from user  System.***out***.println("ENTER FILE NAME:");  fileName=obj.nextLine();    //Read number of line from user  System.***out***.println("Enter how many line in the files:");  linesCount=Integer.*parseInt*(obj.nextLine());    //Read lines from user  **for**(**int** i=1;i<=linesCount;i++)  {  System.***out***.println("ENTER LINE"+i+":");  content.add(obj.nextLine());  }    //save the content into the file  **boolean** isSaved=FileManger.*createFiles*(***folderpath***,fileName,content);    **if**(isSaved)  System.***out***.println("File and data saved succesfully");  **else**  System.***out***.println("some error occured.Please contact admin@sk.com");        }    **public** **static** **void** deleteFile()  {  //Deleting file and creating file obj  String fileName;  Scanner obj=**new** Scanner(System.***in***);    //Read file name from user  System.***out***.println("Enter file name to delete:");  fileName=obj.nextLine();    **boolean** isDeleted =FileManger.*deleteFile*(***folderpath***, fileName);    **if**(isDeleted)  System.***out***.println("file is Deleted Sucessfully");  **else**  System.***out***.println("file not found ");    }    **public** **static** **void** searchFile()  {  //Code for Searching a file  String fileName;  Scanner obj=**new** Scanner(System.***in***);    //Read file from user  System.***out***.println("Enter file name to search:");  fileName=obj.nextLine();    **boolean** isFound =FileManger.*searchFile*(***folderpath***, fileName);    **if**(isFound)  System.***out***.println("file is present ");  **else**  System.***out***.println("file not present ");    }    } |