|  |
| --- |
| **LockedMe.com**  **(Sprint Work & Project Specification)** |

**Version History:**

|  |  |
| --- | --- |
| **Author** | **Kumar Abhishek** |
| **Purpose** | **Scrum details and specifications of the application** |
| **Date** | **12th Aug 2021** |
| **Version** | **1.0** |

Contents

[1. PROJECT GITHUB LINK: 3](#_Toc79729722)

[2. Folder Structure 3](#_Toc79729723)

[3. FileManager.java 4](#_Toc79729724)

[4. LockedMeProject.java 6](#_Toc79729725)

# PROJECT GITHUB LINK:

|  |  |
| --- | --- |
| **REPOSITORY NAME** | **Phase1Project-FSD** |
| **GITHUB LINK** | <https://github.com/kumarabhishek7885/Phase1Project-FSD.git> |

|  |
| --- |
| Folder Structure |
|  |

|  |
| --- |
| FileManager.java |
| **package** com.lockedme;  **import** java.io.File;  **import** java.io.FileWriter;  **import** java.util.ArrayList;  **import** java.util.List;  **public** **class** FileManager  {  /\*\*  \* This method will return the file names from the folder  \* **@param** folderpath  \* **@return** List<String>  \*/  **public** **static** List<String> getAllFiles(String folderpath)  {  //Creating File Object  File fl = **new** File(folderpath);    //Getting All the Files into FileArray  File[] listOfiles = fl.listFiles();    //Declared a list to store file names  List<String> fileNames = **new** ArrayList<String>();    //Getting file names from array of files  **for**(File f:listOfiles)  fileNames.add(f.getName());    //Return the List of file names  **return** fileNames;  }  /\*\*  \* This method will create or append content into the file specified  \* **@param** folderpath  \* **@param** fileName  \* **@param** content  \* **@return** boolean  \*/  **public** **static** **boolean** createFiles(String folderpath, String fileName, List<String> content)  {  **try**  {  //Creating file and file writer object  File fl = **new** File(folderpath, fileName);  FileWriter fw = **new** FileWriter(fl);    //Writing to file  **for**(String s:content)  {  fw.write(s+"\n");  }  fw.close();  **return** **true**;  }  **catch**(Exception Ex)  {  **return** **false**;  }  }    /\*\*  \* This method will delete the file name if it exists.  \* **@param** folderpath  \* **@param** fileName  \* **@return**  \*/  **public** **static** **boolean** deleteFile(String folderpath, String fileName)  {  //Adding folder path with file name and creating file object  File file = **new** File(folderpath+"\\"+fileName);    **try**  {  //Deleting file  **if**(file.delete())  **return** **true**;  **else**  **return** **false**;  } **catch** (Exception e) {  // **TODO**: handle exception  **return** **false**;  }  }    /\*\*  \* This method will search the file from a folder  \* **@param** folderpath  \* **@param** fileName  \* **@return**  \*/  **public** **static** **boolean** searchFile(String folderpath, String fileName)  {  //Adding folder path with file name and creating file object  File file = **new** File(folderpath+"\\"+fileName);    //Search condition  **if**(file.exists())  **return** **true**;  **else**  **return** **false**;  }  } |
| LockedMeProject.java |
| **package** com.lockedme;  **import** java.util.ArrayList;  **import** java.util.Collections;  **import** java.util.List;  **import** java.util.Scanner;  **public** **class** LockedMeProject  {  //private static Scanner scan = new Scanner(System.in);  **static** **final** String ***folderpath***="F:\\MyPhase1Project\\LockedMeFiles";    **public** **static** **void** main(String[] args)  {  **int** goahead=1;    **do** {  //Variable declaration  Scanner obj = **new** Scanner(System.***in***);  **int** ch;    //Menu  *displayMenu*();  System.***out***.println("Enter your Choice:");  ch = Integer.*parseInt*(obj.nextLine());    //switch case  **switch**(ch)  {  **case** 1: *getAllFiles*();  **break**;    **case** 2: *createFiles*();  **break**;    **case** 3: *deleteFile*();  **break**;    **case** 4: *searchFile*();  **break**;    **case** 5: System.*exit*(0);    **default**:System.***out***.println("Invalid Option");  **break**;  }    }**while**(goahead>0);  }    **public** **static** **void** displayMenu()  {  System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  System.***out***.println("\t\tLockedMe.com");  System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  System.***out***.println("1. Display All Files");  System.***out***.println("2. Add New Files");  System.***out***.println("3. Delete a File");  System.***out***.println("4. Search a File");  System.***out***.println("5. Exit");  System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  }    /\*\*  \* This method will retrieve files  \*/  **public** **static** **void** getAllFiles()  {  //getting the file names  List<String> fileNames = FileManager.*getAllFiles*(***folderpath***);    //Edge Condition  **if**(fileNames.size()==0)  System.***out***.println("No Files in the Directory");  **else**  System.***out***.println("FILES LIST IS BELOW:\n");    //Sorting file names in ascending order  List<String> sortedList = **new** ArrayList<String>();      **for**(String str:fileNames)  sortedList.add(str.toLowerCase());  Collections.*sort*(sortedList);    //Printing O/P to console  **for**(String f:sortedList)  System.***out***.println(f);  System.***out***.println();    }  /\*\*  \* this method will Create files  \*/  **public** **static** **void** createFiles()  {  //Variables declaration  Scanner obj = **new** Scanner(System.***in***);  String fileName;  **int** linesCount;  List<String> content = **new** ArrayList<String>();    //reading file name from user  System.***out***.println("Enter file Name:");  fileName=obj.nextLine();    //Reading number of lines from user  System.***out***.println("Enter How Many lines in the file:");  linesCount = Integer.*parseInt*(obj.nextLine());    //Reading lines from user  **for**(**int** i=1;i<=linesCount;i++)  {  System.***out***.println("Enter line "+i+":");  content.add(obj.nextLine());  }    //Saving the content into the file  **boolean** isSaved= FileManager.*createFiles*(***folderpath***, fileName, content);  **if**(isSaved)  System.***out***.println("File and Data Saved Successfully");  **else**  System.***out***.println("Some error occured. please contact Kumar.abhishek7885@gmail.com ");  }    /\*\*  \* This method will Delete files  \*/  **public** **static** **void** deleteFile()  {  //code for deleting a file  String fileName;  Scanner obj = **new** Scanner(System.***in***);  System.***out***.println("Enter File Name to be Deleted:");  fileName=obj.nextLine();    //Deleting file  **boolean** isDeleted = FileManager.*deleteFile*(***folderpath***, fileName);    **if**(isDeleted)  System.***out***.println("File Deleted successfully");  **else**  System.***out***.println("Either file not there or some access issues");  }    /\*\*  \* This method will search files  \*/  **public** **static** **void** searchFile()  {  //code for Searching a file  String fileName;  Scanner obj = **new** Scanner(System.***in***);  System.***out***.println("Enter File Name to be searched:");  fileName=obj.nextLine();    //Searching file  **boolean** isFound = FileManager.*searchFile*(***folderpath***, fileName);    **if**(isFound)  System.***out***.println("File is present in the folder");  **else**  System.***out***.println("File is not present in the folder");  }    } |