|  |
| --- |
| **LockedMe.com**  **(Project Source Code)** |

**Version History:**

|  |  |
| --- | --- |
| Author | Nikhil Jain |
| Purpose | Source Code of the application |
| Date | 10th August 2021 |
| Version | 1.0 |

Table of Contents

[1. Project GitHub Link 3](#_Toc79526653)

[2. Folder Structure 3](#_Toc79526654)

[3. FileManager.java 4](#_Toc79526655)

[4. LockedMe.java 6](#_Toc79526656)

# Project GitHub Link

|  |  |
| --- | --- |
| Repository Name | LockedMe |
| GitHub Link | <https://github.com/Niks4u2/LockedMe> |

# 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** ArrayList  \*/  **public** **static** List<String> getAllFiles(String folderpath)  {  //Creating file object  File folder = **new** File(folderpath);    //Getting all the files into file array  File[] listOfFiles = folder.listFiles();    //Declare a list to store file names  List<String> fileNames = **new** ArrayList<String>();    //Getting file names from array of files  **for**(File f : listOfFiles)  fileNames.add(f.getName());    //return the list of file names  **return** fileNames;  }    /\*\*  \* This method will create and append content to the file specified  \* **@param** folderpath  \* **@param** fileName  \* **@param** content  \* **@return** boolean  \*/  **public** **static** **boolean** createAndWriteToFile(String folderpath, String fileName, List<String> content)  {  **try**  {  //Creating file and file writer object  File file = **new** File(folderpath, fileName);  FileWriter fwrite = **new** FileWriter(file);    //Writing to file  **for**(String s : content)  fwrite.write(s+"\n");    fwrite.close();  **return** **true**;  }  **catch**(Exception ex)  {  **return** **false**;  }  }    /\*\*  \* This method will delete the file name specified if exists  \* **@param** folderpath  \* **@param** fileName  \* **@return** boolean  \*/  **public** **static** **boolean** deleteFile(String folderpath, String fileName)  {  //Creating file object  File file = **new** File(folderpath+"\\"+fileName);  **try**  {  //Deleting file  **if**(file.delete())  **return** **true**;  **else**  **return** **false**;  }  **catch**(Exception ex)  {  **return** **false**;  }  }    /\*\*  \* This method will search the file from the folder  \* **@param** folderpath  \* **@param** fileName  \* **@return** boolean  \*/  **public** **static** **boolean** searchFile(String folderpath, String fileName)  {  //Creating file object  File file = **new** File(folderpath+"\\"+fileName);    //Search condition  **if**(file.exists())  **return** **true**;  **else**  **return** **false**;  }  } |

# LockedMe.java

|  |
| --- |
| **package** com.lockedme;  **import** java.util.ArrayList;  **import** java.util.Collections;  **import** java.util.List;  **import** java.util.Scanner;  **public** **class** LockedMe  {  **private** **static** Scanner *scan* = **new** Scanner(System.***in***);  **private** **static** **final** String ***FOLDERPATH*** = "C:\\Users\\golun\\Desktop\\MyPhaseOneProject\\LockedMeFiles";    **public** **static** **void** main(String[] args)  {  //Variable declaration  **int** proceed = 1;  **int** choice;    **do**  {  *welcomeScreen*();    //Read choice from user  System.***out***.println("Please enter your choice:");  choice = Integer.*parseInt*(*scan*.nextLine());    **switch**(choice)  {  **case** 1 : *getAllFileNames*();  **break**;  **case** 2 : *addFile*();  **break**;  **case** 3 : *deleteFile*();  **break**;  **case** 4 : *searchFile*();  **break**;  **case** 5 : System.***out***.println("Thank you for using the application.");  System.*exit*(0);  **break**;  **default** : System.***out***.println("Invalid Option. Please enter correct choice between 1 to 5.");  }    }**while**(proceed != 0);  }    **public** **static** **void** welcomeScreen()  {  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.***out***.println("\t\tLockedMe.com");  System.***out***.println("\t\t Nikhil Jain");  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");    System.***out***.println("1. Display all the files");  System.***out***.println("2. Add new file");  System.***out***.println("3. Delete a file");  System.***out***.println("4. Search a file");  System.***out***.println("5. Exit\n");    System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  }    **public** **static** **void** getAllFileNames()  {  //Variable declaration  List<String> fileNames = FileManager.*getAllFiles*(***FOLDERPATH***);    //Edge condition  **if**(fileNames.size() == 0)  System.***out***.println("No files in the directory.\n");  **else**  System.***out***.println("Below is the file list:\n");    //Sorting file names in ascending order  Collections.*sort*(fileNames);    //Print output to console  **for**(String fileName : fileNames)  System.***out***.println(fileName);  System.***out***.println();  }    **public** **static** **void** addFile()  {  //Variable declaration  String fileName;  **int** linesCount=0;  **boolean** isAdded;  List<String> content = **new** ArrayList<String>();    //Read file name from user  System.***out***.println("Enter file name: ");  fileName = *scan*.nextLine();    **try**  {  //Read number of lines from user  System.***out***.println("Enter number of lines:");  linesCount = Integer.*parseInt*(*scan*.nextLine());  }  **catch**(Exception ex)  {  System.***out***.println("Please enter only integer values. To add content to the file.\n");  isAdded=**false**;  }    //Read lines from user  **for**(**int** i = 1; i <= linesCount; i++)  {  System.***out***.println("Enter line "+i);  content.add(*scan*.nextLine());  }    //Save content to file  isAdded = FileManager.*createAndWriteToFile*(***FOLDERPATH***, fileName, content);    //Print output to console  **if**(isAdded)  System.***out***.println("File added successfully.\n");  **else**  System.***out***.println("Error occured. Please try again.\n");  }    **public** **static** **void** deleteFile()  {  //Variable declaration  String fileName;  **boolean** isDeleted;    //Read file name from user  System.***out***.println("Enter file name to be deleted: ");  fileName = *scan*.nextLine();    //Check for deletion  isDeleted = FileManager.*deleteFile*(***FOLDERPATH***, fileName);    //Print output to console  **if**(isDeleted)  System.***out***.println("File deleted successfully.\n");  **else**  System.***out***.println("File not found or some access issue.\n");  }    **public** **static** **void** searchFile()  {  //Variable declaration  String fileName;  **boolean** isFound;    //Read file name from user  System.***out***.println("Enter file name to be searched: ");  fileName = *scan*.nextLine();    //Check for search result  isFound = FileManager.*searchFile*(***FOLDERPATH***, fileName);    //Print output to console  **if**(isFound)  System.***out***.println("File is present in the directory.\n");  **else**  System.***out***.println("File is not present in the directory.\n");  }  } |