|  |
| --- |
| **Project specification and Sprints**  **VirtualLocked.com** |

Version history:

|  |  |
| --- | --- |
| Author | Shivang Dhasmana |
| Purpose | Screenshot |
| Date | 12/08/1996 |
| Version | 1.0 |

Contents

[1. Module in the project: 3](#_Toc80061012)

[2. Java Technology 3](#_Toc80061013)

[3. Sprint wise Work: 3](#_Toc80061014)

[4. Project github link 3](#_Toc80061015)

[5.Project Code 4](#_Toc80061016)

# 1. Module in the project:

1.Display Files-It will display will all the files in the particular Folder

2.Add Files- if we want to add the new files with some info, we used this feature

3.Delete Files-if we want to delete files from the folder

4.Search Files- if we want to search the folder from the folder

# 2. Java Technology

* Exceptional handling
* Working with files
* Naming Standard
* Modularity
* Oops concept
* Collections
* Control Structure
* Data Structure

# 3. Sprint wise Work:

|  |  |
| --- | --- |
| Sprint number | Modules |
| 1 | Display All files  Add new files |
| 2 | Search files  Delete files  Deployment in jar |

# 4. Project github link

|  |  |
| --- | --- |
| Repository Name |  |
|  |  |
| GitHub Link |  |
|  |  |

# 5.Project Code

|  |
| --- |
| Folder structure: |
|  |
| File manager:   |  | | --- | | package com.locked;  import java.io.File;  import java.io.FileWriter;  import java.util.ArrayList;  import java.util.List;  public class FileManager  {  /\*\*  \* this method all the files name from the folder  \* @param folderpath  \* @return list of string  \*/  public static List<String> getAllFiles(String folderpath)  {    //creating file object  File fl=new File(folderpath);    //Getting all the files into FileArray  File[] listofFiles = fl.listFiles();    //Declare a List to Store files names  List<String> fileNames=new ArrayList<String>();    for(File f:listofFiles)  fileNames.add(f.getName());    //return the list  return fileNames;    }    /\*\*  \* tThis method will create or append content into the file specified  \* @param folderpath  \* @param fileName  \* @param content  \* @return boolean  \*/  public static boolean writeContentToFile(String folderpath,String fileName,List<String> content)  {  try  {  File fl = new File(folderpath, fileName);  FileWriter fw = new FileWriter(fl);    for (String s:content)  {  fw.write(s+"\n");  }  fw.close();  return true;    }catch(Exception Ex)  {  return false;  }  }    /\*\*  \* This method delete the file if it exists  \* @param folderpath  \* @param fileName  \* @return  \*/  public static boolean deleteFile(String folderpath, String fileName)  {  File file=new File(folderpath+"\\"+fileName);  try  {  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  \*/  public static boolean searchFile(String folderpath, String fileName)  {  File file=new File(folderpath+"\\"+fileName);    if(file.exists())  return true;  else  return false;  }      } | |
| Virtual locked me code:   |  | | --- | | package com.locked;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  public class VirtualLocked  {  static final String folderpath="D:\\My phase1 Project\\Lockedfiles";  public static void main(String[] args)  {  int proceed=1;      do    {  //variable declaration      Scanner obj= new Scanner(System.in);  int ch;  //Menu  displayMenu();  System.out.println("Enter the choice:");  ch=Integer.parseInt(obj.nextLine());    switch(ch)  {  case 1 : getAllFiles();  break;  case 2 : createFiles();  break;  case 3 : deleteFiles();  break;  case 4: searchFiles();  break;  case 5: System.exit(0);  break;  default : System.out.println("Invalid option");  break;    }  }while(proceed>0);        }  public static void displayMenu()  {  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\tCompany Lockers pvt ltd");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("1.Display all files");  System.out.println("2.Add files");  System.out.println("3.Delete files");  System.out.println("4.Search files");  System.out.println("5.Exit");    System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");    }  /\*\*  \* Method for all files  \*/  public static void getAllFiles()  {  List<String> fileNames=FileManager.getAllFiles(folderpath);    for(String f:fileNames)  System.out.println(f);  }  /\*\*  \* Method for creating files  \*/  public static void createFiles()  {  Scanner obj = new Scanner(System.in);  String fileNames;  int linesCount;  List<String> content = new ArrayList<String>();    //Read file name user  System.out.println("Enter the File Name");  fileNames=obj.nextLine();    //read no. of line from the user  System.out.println("Enter how many lines in the files");  linesCount=Integer.parseInt(obj.nextLine());    //read no. of line from the user  for(int i=1;i<=linesCount;i++)  {  System.out.println("Enter the line "+i+":");  content.add(obj.nextLine());  }  //save the content into the file  boolean isSaved = FileManager.writeContentToFile(folderpath, fileNames, content);  if(isSaved)  System.out.println("File and data save successfully");  else  System.out.println("Some error occurred Please contact @admin");  //close obj  //obj.close();    /\*\*  \*Method For deleting files  \*/  }  public static void deleteFiles()  {  String fileName;  Scanner obj=new Scanner(System.in);  System.out.println("Enter file name to be deleted");    fileName=obj.nextLine();    boolean isDeleted = FileManager.deleteFile(folderpath, fileName);    if(isDeleted)  System.out.println("File deleted Succussfully");  else  System.out.println("Either file not there or some access issue");  // obj.close();  }  /\*\*  \* Method for searching files  \*/    public static void searchFiles()  {  String fileName;  Scanner obj=new Scanner(System.in);  System.out.println("Enter file name to be Search:");    fileName=obj.nextLine();    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");  //obj.close();  }  } | |