|  |
| --- |
| **Lockedme.com**  **(Sprint work and Project specifications)** |

**Version History:**

|  |  |
| --- | --- |
| Author | Dileep kumar arji |
| Purpose | Sprint work and specification of project |
| Date | 11th Aug 2021 |
| Version | 1.0 |

Contents

[1. Module of the project 3](#_Toc79611972)

[2. Sprint wise work: 3](#_Toc79611973)

[3. Git hub link: 3](#_Toc79611974)

[4. Project code: 4](#_Toc79611975)

[5. Mapping 11](#_Toc79611976)

# Module of the project

1. Display all Files
2. Add File
3. Delete File
4. Search File

# 2. Sprint wise work:

|  |  |
| --- | --- |
| **Sprint Number** | **Modules** |
| 1 | Display all Files:  Display the file that already save in the Lockedme.com |
| 2 | Add Files:  Create new file what do you want |
| 3 | Delete File :  Clear the file data from Lockedme.com |
| 4 | Search File:  Search the file in the Lockedme.com |
| 5 | Testing  Deployment |

# 3. Git hub link:

|  |  |
| --- | --- |
| **Repository name** | ArjiDileepKumar |
| **Repository Link** | https://github.com/arjidileepkumar/LockedMe.git |

# 4. Project code:

|  |
| --- |
| **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 and the folder  \* @param folderpath  \* @return  \*/  public static List<String> getAllFiles(String folderpath)  {  //Creating File Object  File f1 = new File (folderpath);    //Getting all files into File array  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 the list  return fileNames;    }  /\*\*  \* this method will create or append content in the folder  \* @param folderpath  \* @param fileName  \* @return  \*/  public static boolean addFiles(String folderpath,String fileName,List<String> content)    {  try  {  File f = new File(folderpath,fileName);  FileWriter fw =new FileWriter(f);  for (String s:content)  {  fw.write(s+"\n");  }  fw.close();  return true;  }  catch(Exception Ex)  {  return false;  }    }      /\*\*  \* This method will delete the content in the folder  \* @param folderpath  \* @param fileName  \* @return  \*/  public static boolean deleteFile(String folderpath, String fileName)  {  //adding folder with file name and folderpath  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 content in the folder  \* @param folderpath  \* @param fileName  \* @return  \*/  public static boolean searcFile(String folderpath, String fileName)  {  //adding folder with file name and folderpath  File file = new File(folderpath+"\\"+fileName);  try  {  if(file.exists())  return true;  else  return false;  }  catch(Exception Ex)  {  return false;  }  }      } |

|  |
| --- |
| **Lockedme.java** |
| package com.lockedme;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  public class LockedMe {  // creating a folder path  static final String folderpath="G:\\My project phase 1\\LockedMeFiles";    // main menu calling for other methods  public static void main(String[] args)  {  int proceed=1;  //do while looping for menu display reapetly  do  {    //variable declaration  int d;    //Display Menu  d= displayMenu();      // switch case to calling the methods  switch(d)  {  case 1 : getAllFiles();  break;  case 2 : addFiles();  break;  case 3 : deleteFile();  break;  case 4 : searchingFiles();  break;  case 5 : System.exit(0);  break;  default : System.out.println("Invalid option");  break;    }    }while(proceed>0);    }      public static int displayMenu()  {    //variable declaration  Scanner s = new Scanner(System.in);  int d;  //Menu  System.out.println("================================================");  System.out.println("\t\tcompanyLockerpvt.Ltd");  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("=================================================");    System.out.println("Enter your choice:");  d= Integer.parseInt(s.nextLine());  return d;    }  /\*\*  \* calling the get files into main method  \*/  public static void getAllFiles()  {  //Get files names  List<String> fileNames = FileManager.getAllFiles(folderpath);    for(String f:fileNames)  System.out.println(f);  }  /\*\*  \* calling adding files into main method  \*/  public static void addFiles()  {  //Adding files  //Variable declaration  Scanner s = 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 =s.nextLine();    //Read number of lines from user  System.out.println("enter how many lines in the file:");  linesCount = Integer.parseInt(s.nextLine());    //Read Lines from user  for (var i=1;i<=linesCount;i++)  {  System.out.println("enter line"+i+":");  content.add(s.nextLine());  }  //save the content into the file  boolean isSaved = FileManager.addFiles(folderpath, fileName, content);    if (isSaved)  System.out.println("file and data saved sucessfully");  else  System.out.println("some error occured. please contact dileep");    //s.close();    }  /\*\*  \* deleting method added to main method  \*/  public static void deleteFile()  {  //variable declaration  String fileName;  Scanner s = new Scanner(System.in);    //Read File name from the user  System.out.println("enter file name:");  fileName = s.nextLine();    //deleting the file  boolean isDeleted = FileManager.deleteFile(folderpath, fileName);  if (isDeleted)  System.out.println("File sucessfully deleted");  else  System.out.println("File is not their");  //s.close();    }  /\*\*  \* searching method added to main method  \*/  public static void searchingFiles()  {  //Variable declaration  String fileName;  Scanner s = new Scanner(System.in);    //Read file name from the user  System.out.println("enter file name to be search:");  fileName = s.nextLine();    //searching the File  boolean isSearched = FileManager.searcFile(folderpath, fileName);  if (isSearched)  System.out.println("File is present in the folder");  else  System.out.println("File is not present in the folder");  //s.close();    }  } |

## 5. Mapping

