

Lockedme.com

(Sprint Work and Project Specification)

- **Version History:**

1.	Author Name	Tushar Khillare.
2.	Purpose	Specification of Project and Sprint Work.
3.	Date	8 August 2021
4.	Version	1.0

– Contents

1.Modules in the Project.	3
2.JAVA Technology Used:	3
3.Sprint Wise Work.	4
4.Project GitHub Link.....	4
5.Project Code.	5

1.Modules in the Project.

- a. Display all Files.
- b. Add File.
- c. Delete File.
- d. Search File.

2.JAVA Technology Used:

- a. Exception Handling.
- b. Working with the Files.
- c. Naming Standards.
- d. Modularity.
- e. Object Oriented Programming.
- f. Collections.
- g. Data Structures.
- h. Control Structure.

3.Sprint Wise Work.

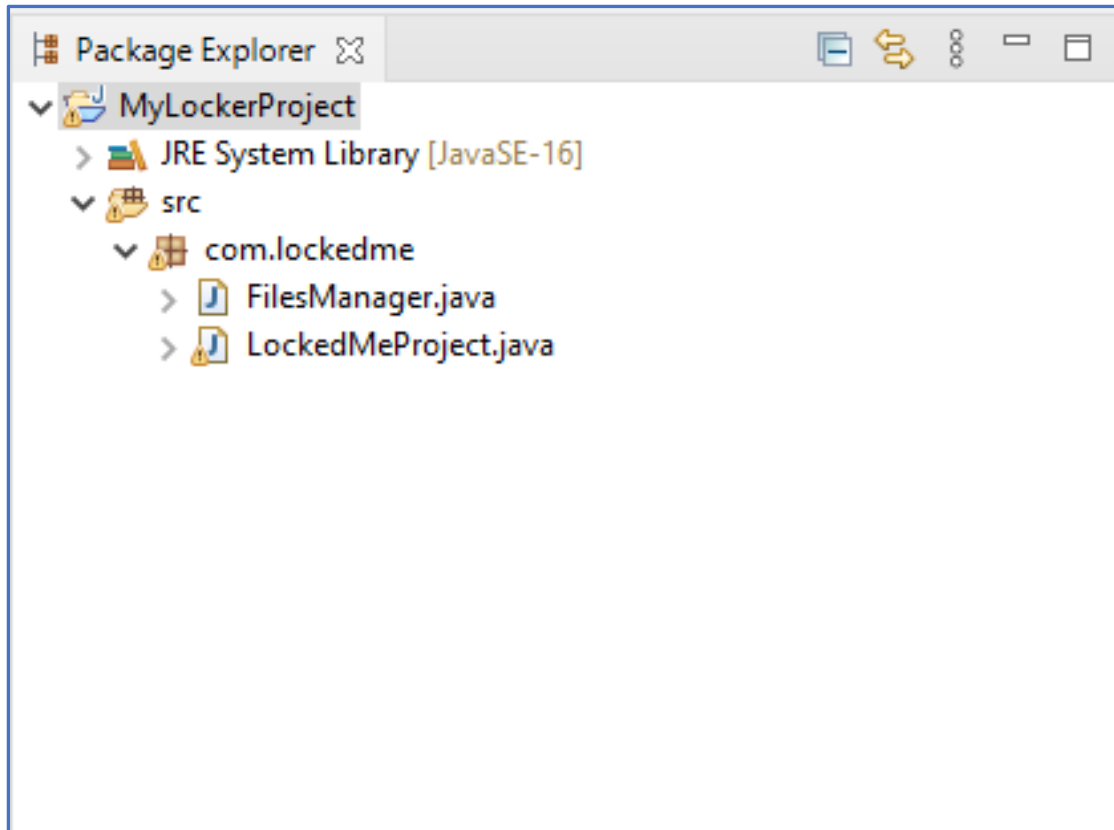
Sr no	Sprint Number	Sprint Module
1.	1	a. Display All Files – Displays all the files present in folder b. Add a new File – Adds all the files in folder as per user requirements.
2.	2	a. Delete a File – Delete the file from the folder as per user input. b. Search a File – Searches a file from the folder.
3.	3	a. Testing – Testing of a code. b. Deployment (creating a jar file.)

4.Project GitHub Link.

Repository Name.	LockedME
GitHub Link.	https://github.com/tuskillare/LockedME

5.Project Code.

A.Folder Structure.



B. FilesManager.java

```
package com.lockedme;

import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;

public class FilesManager
{
    /**
     * This method will return the file name 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 filesArrays.
    File[] listOfFiles = fl.listFiles();

    //Declare a list to store the file name.
    List<String> fileNames = new ArrayList<String>();

    for(File f:listOfFiles)
        fileNames.add(f.getName());

    //Return the fileNames.
    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
    {
        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 will delete the file name if exist.

```

```

* @param folderpath
* @param fileName
* @return
*/
public static boolean deleteFiles(String folderpath,String fileName)
{
    //Initializing the path with file name and creating the object.
    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 if exist.
* @param folderpath
* @param folderName
* @return
*/

public static boolean searchFiles(String folderpath,String fileName)
{
    //Initializing the path with file name and creating the object.
    File file = new File(folderpath+"\\\\"+fileName);

    try
    {
        if(file.exists())
            return true;
        else
            return false;
    }
    catch(Exception Ex)
    {
        return false;
    }
}

```

```
}
```

C. LockedMeProject.java

```
package com.lockedme;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class LockedMeProject
{
    static final String folderpath="D:\\Phase1 Project\\LockedMeFiles"; //FolderPath of files.
    public static void main(String[] args)
    {
        int proceed = 1;

do
    {
        //Variable Declaration.
        int ch;

        //Display Menu.
        ch=displayMenu();

        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("Invaild Option");
                    break;
        }
    }while(proceed>0);
}

public static int displayMenu()
```



```
{
    Scanner obj = new Scanner(System.in);
    int ch;

System.out.println("+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*");
    System.out.println("\t\tLockedMe.com");

System.out.println("+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*");
    System.out.println("1). Display All Files.");
    System.out.println("2). Add a New File");
    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 : ");
    ch = Integer.parseInt(obj.nextLine());

    return ch;
}

/**
 * This method will return the file name from the folder.
 * @param folderpath
 * @return List<String>
 */
public static void getAllFiles()
{
    //getting files name
    List<String> fileNamees = FileManager.getAllFiles(folderpath);

    if(fileNamees.size()==0)
        System.out.println("Files not found in Directory");
    else
    {
        System.out.println("File list is below:\n");
        for(String f:fileNamees)
            System.out.println(f);
    }
}

/**
 * This method will create or append content into the file specified.
 * @param folderpath.
 * @param fileName.
 * @param content.
```

```

* @return boolean.
*/
public static void createFiles()
{
    //Variable Declaration.
    Scanner obj = new Scanner(System.in);
    String fileName;
    int linesCount;
    List<String> content = new ArrayList<String>();

    //Read files names from the user.
    System.out.println("Enter the File Name:");
    fileName=obj.nextLine();

    //Read number of Lines form the user.
    System.out.println("Enter how many lines in the files:");
    linesCount=Integer.parseInt(obj.nextLine());

    //Read lines from the user.
    for(int i=1;i<=linesCount;i++)
    {
        System.out.println("Enter line "+i+":");
        content.add(obj.nextLine());
    }

    //Save content into the file.
    boolean isSaved= FileManager.createFiles(folderpath,fileName,content);

    if(isSaved)
        System.out.println("Files and Data saved Sucessfully");
    else
        System.out.println("Error occured.Please contact admin@.com");
}

/**
 * This method will delete the file name if exist.
 * @param folderpath
 * @param fileName
 * @return
 */
public static void deleteFiles()
{
    //Code for Deleting File.
    String fileName;
    Scanner obj = new Scanner(System.in);//User input.
    System.out.println("Enter the file name to be Deleted:");
    fileName = obj.nextLine();

```

```

        //Deleting the File.
        boolean isDeleted = FileManager.deleteFiles(folderpath,fileName);

        if(isDeleted)
            System.out.println("File deleted Successfully");
        else
            System.out.println("File not there");

    }

    /**
     * This method will Search the File if exist.
     * @param folderpath
     * @param fileName
     * @return
     */

    public static void searchFiles()
    {
        //Code for Searching the File.
        String fileName;
        Scanner obj = new Scanner(System.in);//User input.
        System.out.println("Enter the file name to be Searched:");
        fileName = obj.nextLine();

        //Searching the File.
        boolean isFound = FileManager.searchFiles(folderpath,fileName);

        if(isFound)
            System.out.println("File searched Successfully");
        else
            System.out.println("File not Found");

    }
}

```