

[LockedMe.com](https://www.lockedme.com)
(Sprint Work and
Project Specification)

Version History:

Author	Atharv Pratap Singh
Purpose	Screenshots of application
Date	14 ^h August 2021
Version	1.0

Java Technologies used:-

- ✓ Exception Handling
- ✓ Working with files
- ✓ Modularity
- ✓ Naming Standards
- ✓ Object Oriented Programming
- ✓ Data Structures
- ✓ Collections
- ✓ Control Structures

Contents

1. Modules in LockedMe.com Project.....	3
2. Sprint Wise Work.....	4
3. Project GITHUB Link.....	5
4. Project Code.....	6

1. Modules in LockeMe.com Project

- a. Display All files in the Directory
- b. Adding a new file and content
- c. Deleting a specific file in the Directory
- d. Search a particular file

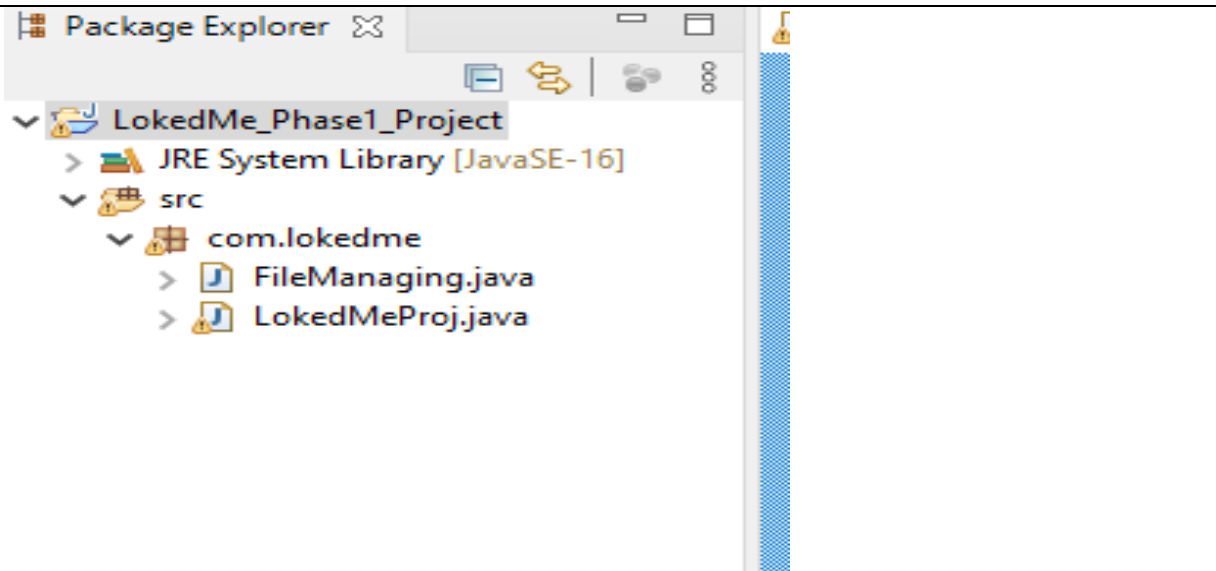
2. Sprint wise work

Sprint Number	Modules
1	Display All files Adding new File
2	Deleting a file Search a file Testing Deployment(Creating a runnable jar file)

3. Project GITHUB link:

Repository Name:-
Simplilearn_Project_P1
Github Link:-
https://github.com/apstorm/Simplilearn_Project_P1

4. Project Code:

<div data-bbox="203 338 1412 934"><p>Folder Structure</p><p>The screenshot shows the Package Explorer of an IDE. The project 'LokedMe_Phase1_Project' is expanded, revealing the 'JRE System Library [JavaSE-16]' and the 'src' folder. The 'src' folder is further expanded to show the package 'com.lokedme', which contains two Java files: 'FileManaging.java' and 'LokedMeProj.java'.</p><pre>graph TD; LokedMe_Phase1_Project --> JRE_System_Library[JRE System Library [JavaSE-16]]; LokedMe_Phase1_Project --> src; src --> com_lokedme[com.lokedme]; com_lokedme --> FileManaging_java[FileManaging.java]; com_lokedme --> LokedMeProj_java[LokedMeProj.java];</pre></div> <p>FileManaging.java</p>

```
LokedMeProj.java  FileManaging.java  ⌕
1  package com.lokedme;
2
3  import java.io.File;
4
5
6
7
8  public class FileManaging
9  {
10     /**
11     * this method will return the file names from folder
12     * @param folderPath
13     * @return list<String>
14     */
15     public static List<String> getAllFiles(String folderPath)
16     {
17         //file object creation
18         File fl=new File(folderPath);
19
20         //getting all files inside array of file
21         File[] listOfFiles=fl.listFiles();
22
23         //declare list to store file names
24         List<String> fileNames=new ArrayList<String>();
25
26         for(File f:listOfFiles)
27         {
28             fileNames.add(f.getName());
29         }
30
31         //return list
32         return fileNames;
33     }
34 }
35 /**
36 * this method create or append content into the specified file
37 * @param folderPath
38 * @param fileName
39 * @param content
40 * @return
41 */
42 public static boolean createFile(String folderPath,String fileName, List<String> content)
```



```
42 public static boolean createFile(String folderPath,String fileName, List<String> content)
43 {
44     try
45     {
46         File fl=new File(folderPath, fileName);
47         FileWriter fw=new FileWriter(fl);
48         for(String s:content)
49         {
50             fw.write(s+"\n");
51         }
52         fw.close();
53         return true;
54     }
55     catch (Exception e)
56     {
57         return false;
58     }
59 }
60
61
62 /**
63  * this method will delete the file name specified
64  * @param folderPath
65  * @param fileName
66  * @return
67  */
68 public static boolean deleteFile(String folderPath,String fileName)
69 {
70     //adding folder path with filename and create file object
71     File f=new File(folderPath+"\""+fileName);
72     try
73     {
74         if(f.delete())
75         {
76             return true;
77         }
78         else
79         {
80             return false;
```

```

73     {
74         if(f.delete())
75         {
76             return true;
77         }
78         else
79         {
80             return false;
81         }
82     }
83     catch (Exception e)
84     {
85         return false;
86     }
87 }
88
89 public static boolean searchFile(String folderPath,String fileName)
90 {
91     //adding folder path with filename and create file object
92     File f=new File(folderPath+"\\ "+fileName);
93     try
94     {
95         if(f.exists())
96         {
97             return true;
98         }
99         else
100         {
101             return false;
102         }
103     }
104     catch (Exception e)
105     {
106         return false;
107     }
108 }
109 }
110 }
111

```

```
LokedMeProj.java  FileManaging.java
1 package com.lokedme;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.Scanner;
6
7 public class LokedMeProj {
8
9     static final String folderPath=
10     "F:\\EGDownloads\\Simplilearn stuffs\\live class vedeos\\P-1\\Project(Phase-1)\\(LockedMe.com)-files";
11     public static void main(String[] args)
12     {
13         int go=1;
14         do
15         {
16             //Variable Declaration
17             Scanner obj=new Scanner(System.in);
18             int choice;
19
20             //Menu
21             displayMenu();
22
23             System.out.println("enter your choice: ");
24             choice=Integer.parseInt(obj.nextLine());
25
26             switch (choice)
27             {
28                 case 1: getAllFiles();
29                 break;
30
31                 case 2: createFile();
32                 break;
33
34                 case 3: deleteFile();
35                 break;
36
37                 case 4: searchFile();
38                 break;
39

```

```

39
40         case 5: System.exit(0);
41         break;
42
43
44         default: System.out.println("Invalid Choice..");
45         break;
46
47     }
48     }while(go>0);
49
50 }
51
52 /**
53  * Description: this method is used to display the menu options
54  *
55  * Author:      Atharv Pratap Singh
56  *
57  * Date:        14/08/2021
58  */
59 public static void displayMenu()
60 {
61     Scanner obj=new Scanner(System.in);
62     // int ch;
63
64     System.out.println("*****");
65     System.out.println("\t\tCompany Lockers Pvt. Ltd.");
66     System.out.println("*****");
67     System.out.println("**** 1. Display all files *****");
68     System.out.println("**** 2. Add new file *****");
69     System.out.println("**** 3. Delete a file *****");
70     System.out.println("**** 4. Search a file *****");
71     System.out.println("**** 5. Exit *****");
72     System.out.println("**** Enter your choice: *****");
73     System.out.println("*****");
74     // ch=Integer.parseInt(obj.nextLine());
75     // return ch;
76
77

```

```

77     }
78
79     /**
80      * Description: this method is used for retrieving all
81      *               files in ascending order
82      *
83      * Author:      Atharv Pratap Singh
84      *
85      * Date:        13/08/2021
86      */
87     public static void getAllFiles()
88     {
89         List<String> fileNames=FileManaging.getAllFiles(folderPath);
90
91         if(fileNames.size()==0)
92         {
93             System.out.println("no files in the directory");
94         }
95         else
96         {
97             System.out.println("file list is below: \n");
98
99             // Shows all files in folder but ascending remains
100            for(String s:fileNames)
101            {
102                System.out.println(s);
103            }
104        }
105    }
106
107     /**
108      * Description: this method is used to add a new file
109      *               along with some lines of content inside
110      *               it
111      *
112      * Author:      Atharv Pratap Singh
113      *
114      * Date:        13/08/2021
115      */

```

```

114      * Date:      13/08/2021
115      */
116      public static void createFile()
117      {
118          // Add file and content code
119
120          //variable declaration
121          Scanner obj=new Scanner(System.in);
122          String fileName;
123          int linesCount;
124          List<String> content=new ArrayList<String>();
125
126          //read file name from user
127          System.out.println("enter file name: ");
128          fileName=obj.nextLine();
129
130          //read number of lines
131          System.out.println("enter how many lines in file ?");
132          linesCount=Integer.parseInt(obj.nextLine());
133
134          //read lines from user input
135          for(int i=1;i<=linesCount;i++)
136          {
137              System.out.println("enter line "+i+":");
138              content.add(obj.nextLine());
139          }
140
141
142          //saving the content into the file
143          boolean isSaved=FileManaging.createFile(folderPath, fileName, content);
144
145          if(isSaved)
146          {
147              System.out.println("file data saved successfully");
148          }
149          else
150          {
151              System.out.println("some error is there. please contact system adminisstrator");
152          }

```

```

151         System.out.println("some error is there. please contact system adminisstrator");
152     }
153
154     //closing scanner object
155     obj.close();
156 }
157
158 /**
159  * Description: this method is used for deleting
160  *             the specified file in the folder
161  *
162  * Author:     Atharv Pratap Singh
163  *
164  * Date:       14/08/2021
165  */
166 public static void deleteFile()
167 {
168     // Deleting the specified file
169     String fileName;
170     Scanner obj=new Scanner(System.in);
171     System.out.println("enter file name to be deleted: ");
172     fileName=obj.nextLine();
173
174     boolean isDeleted=FileManaging.deleteFile(folderPath, fileName);
175     //deleting the file
176     if(isDeleted)
177     {
178         System.out.println("file deleted successfully");
179     }
180     else
181     {
182         System.out.println("file not there or may be some other issue...");
183     }
184 }
185
186
187 /**
188  * Description: this method is used to search
189  *             the specified file to know

```

```

180         else
181         {
182             System.out.println("file not there or may be some other issue...");
183         }
184     }
185 }
186
187 /**
188  * Description: this method is used to search
189  *              the specified file to know
190  *              whether is it present or not
191  *              in the folder
192  *
193  * Author:      Atharv Pratap Singh
194  *
195  * Date:        14/08/2021
196  */
197 public static void searchFile()
198 {
199     // code for Searching a specified file
200     String fileName;
201     Scanner obj=new Scanner(System.in);
202     System.out.println("enter the file to be searched: ");
203     fileName=obj.nextLine();
204
205     boolean isFound=FileManaging.searchFile(folderPath, fileName);
206
207     if(isFound)
208     {
209         System.out.println("file is found, present");
210     }
211     else
212     {
213         System.out.println("file not found, absent");
214     }
215 }
216
217 }
218

```


THANK YOU