# LockedMe.com (Sprint Work and Project Specification)

#### **Version History:**

Author	Atharv Pratap Singh
Purpose	Screenshots of application
Date	14 <sup>h</sup> 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
	Sprint Wise Work	
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

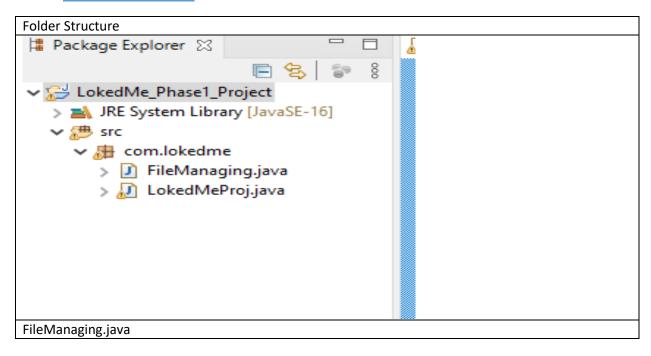
# 2. Sprint wise work

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:



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

```
public static boolean createFile(String folderPath,String fileName, List<String> content)
42⊖
43
44
            try
45
            {
46
                File fl=new File(folderPath, fileName);
                FileWriter fw=new FileWriter(fl);
47
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
         * @param fileName
65
        * @return
66
67
68⊖
        public static boolean deleteFile(String folderPath,String fileName)
69
70
            //adding folder path with filename and create file object
            File f=new File(folderPath+"\\"+fileName);
71
72
            try
73
                if(f.delete())
74
75
76
                    return true;
77
78
                else
79
80
                    return false;
```

```
73
                  if(f.delete())
 74
 75
 76
                      return true;
 77
 78
                 else
 79
                      return false;
 80
 81
 82
             catch (Exception e)
 83
 84
 85
                 return false;
 86
 87
         }
 88
         public static boolean searchFile(String folderPath,String fileName)
 89⊜
 90
 91
             //adding folder path with filename and create file object
 92
             File f=new File(folderPath+"\\"+fileName);
 93
             try
 94
                 if(f.exists())
 95
 96
                  {
                      return true;
 97
 98
 99
                 else
100
101
                      return false;
102
103
104
             catch (Exception e)
105
106
                 return false;
107
108
109
110
111
                                                         111 . 1 . 2010
                     Writable
```

LokedMeProj.java

```
    LokedMeProj.java 
    □ FileManaging.java

   package com.lokedme;
   3⊖ import java.util.ArrayList;
   4 import java.util.List;
   5 import java.util.Scanner;
   7 public class LokedMeProj {
   9⊝
          static final String folderPath=
          "F:\\EGDownloads\\simplilearn stuffs\\live class vedeos\\P-1\\Project(Phase-1)\\(LockedMe.com)-files";
  10
  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
              //Menu
  20
  21
              displayMenu();
  22
              System.out.println("enter your choice: ");
  23
  24
              choice=Integer.parseInt(obj.nextLine());
  25
  26
              switch (choice)
  27
              {
                  case 1: getAllFiles();
  28
  29
                  break;
  30
                  case 2: createFile();
  31
  32
                  break;
  33
                  case 3: deleteFile();
  34
  35
                  break;
  36
                  case 4: searchFile();
  37
  38
                  break;
  39
```

```
40
         case 5: System.exit(0);
41
         break;
42
43
       default: System.out.println("Invalid Choice..");
44
45
46
47
       }while(go>0);
48
49
50
    }
51
52⊖
     * Description: this method is used to display the menu options
53
54
55
     * Author:
             Athary Pratap Singh
56
57
     * Date:
             14/08/2021
    public static void displayMenu()
58
59⊝
60
61
           Scanner obj=new Scanner(System.in);
62 //
           int ch;
63
           64
           65
66
67
           68
69
70
71
           72
73
74 //
           ch=Integer.parseInt(obj.nextLine());
75 //
           return ch;
76
77
```

```
77
         }
 78
 79⊝
          * Description: this method is used for retrieving all
 80
 81
                         files in ascending order
 82
          * Author:
 83
                         Athacy Pratap Singh
 84
 85
          * Date:
                         13/08/2021
 86
         public static void getAllFiles()
 87⊝
 88
 89
             List<String> fileNames=FileManaging.getAllFiles(folderPath);
 90
             if(fileNames.size()==0)
 91
 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 acending 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
111
112
          * Author:
                         Athary Pratap Singh
113
          * Date:
114
                         13/08/2021
          */
115
```

```
* Date:
                         13/08/2021
114
115
116⊖
         public static void createFile()
117
118
             // Add file and content code
119
120
             //variable declaration
             Scanner obj=new Scanner(System.in);
121
122
             String fileName;
123
             int linesCount;
124
             List<String> content=new ArrayList<String>();
125
126
             //read file name from user
             System.out.println("enter file name: ");
127
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
             for(int i=1;i<=linesCount;i++)</pre>
135
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
             {
                 System.out.println("file data saved successfully");
147
148
             }
149
             else
150
                 System.out.println("some error is there. please contact system adminisstrator");
151
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⊜
         * Description: this method is used for deleting
159
160
                         the specified file in the folder
161
          * Author:
162
                         Athary Pratap Singh
163
          * Date:
164
                         14/08/2021
165
166⊖
        public static void deleteFile()
167
168
             // Deleting the specified file
            String fileName;
169
170
            Scanner obj=new Scanner(System.in);
171
            System.out.println("enter file name to be deleted: ");
172
            fileName=obj.nextLine();
173
            boolean isDeleted=FileManaging.deleteFile(folderPath, fileName);
174
175
             //deleting the file
176
            if(isDeleted)
177
178
                 System.out.println("file deleted successfully");
179
            }
180
            else
181
             {
                 System.out.println("file not there or may be some other issue...");
182
183
184
185
        }
186
187⊝
          * Description: this method is used to search
188
                         the specified file to know
189
```

```
180
              else
 181
              {
                  System.out.println("file not there or may be some other issue...");
 182
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:
                         Athacy Pratap Singh
 194
          * Date:
                          14/08/2021
 195
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
```

