

# Company Lockers Private Ltd.

Product LockedMe.com

Prototype of the Application

Name: Sachin Suresh Kale

GitHub: <https://github.com/sachinskale127/Java-FSD-Phase-1-Project.git>

The prototype of the application is operated as a CLI (Command Line Program) without GUI. Its usage is to do file operations such as create new files along with content, delete a file or search a file from a specified directory and list them afterward in sorting order.

The implementation is done with the help of Java 8 and IDE IntelliJ.

## **Sprint Planning**

The Implementation is done in two sprints which are mentioned below:

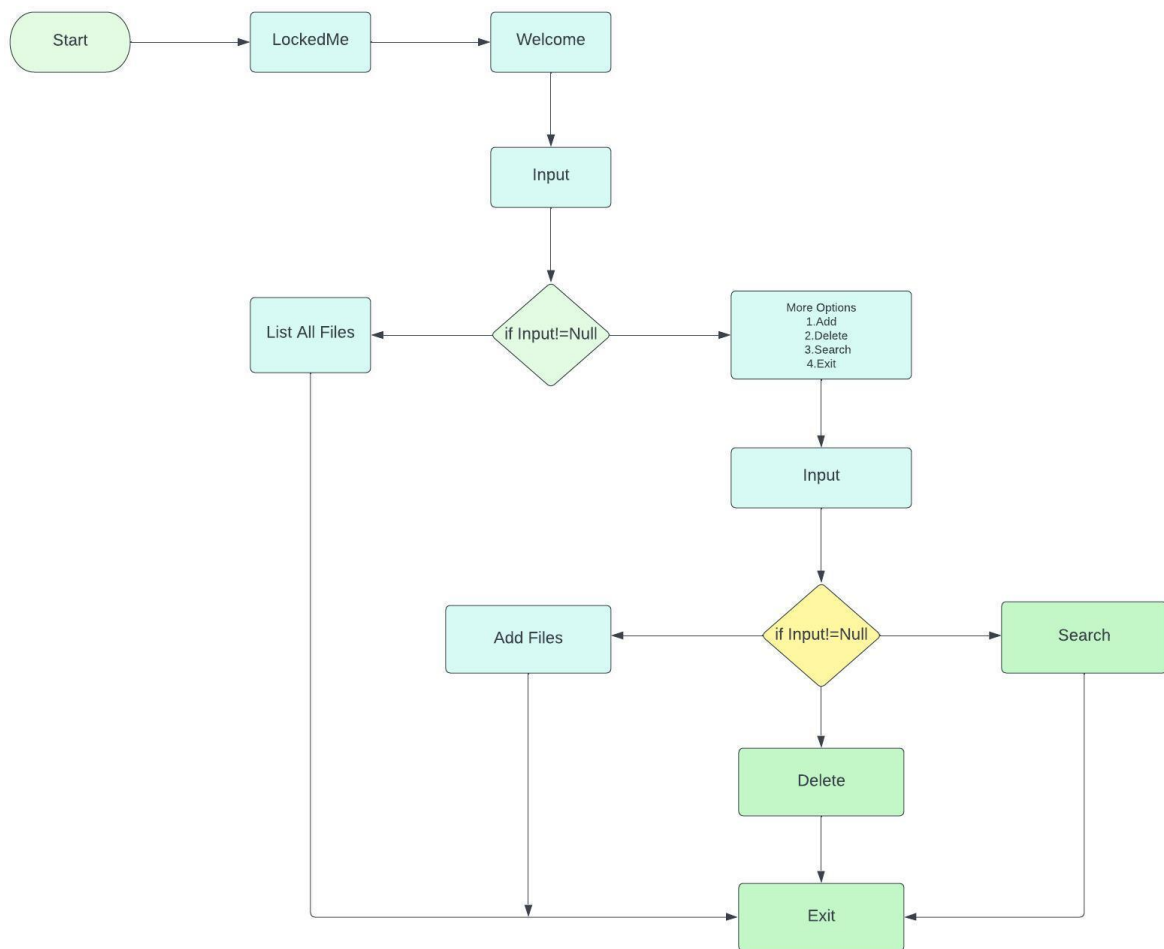
### Sprint 1:

- Clarify the specification and requirements.
- Implement view content mechanism.
- Implement list of all files in sorted order.
- Implement functionality to close the program safely.

### Sprint 2:

- Implement functionality to add create files along the content.
- Implement functionality to delete a file if it is present in that user specified directory.
- Implement functionality to search a file in the same directory.
- Documentation.

## LockedMe Application Flowchart



## Output

```
*****
*      DEVELOPED BY Sachin Suresh Kale      *
*****
*      LOCKEDME.COM                        *
*****
```

```
=====
|      MAIN MENU      |
=====
| Select any one of the following: |
| 1 - List All Files  |
| 2 - More Options   |
| 3 - Exit           |
|                     |
=====
```

Enter your choice :

1

```
*****
```

\*\*\*\*\*

```
=====
|           MAIN MENU           |
=====
| Select any one of the following: |
|   1 - List All Files            |
|   2 - More Options              |
|   3 - Exit                      |
|                                |
=====
```

Enter your choice :

2

```
=====
|           SUB MENU            |
=====
| Select any one of the following: |
|   1 - Add a file                |
|   2 - Delete a file            |
|   3 - Search a file            |
|   4 - Go Back                  |
|                                |
=====
```

Enter your choice :

1

==> Adding a File...

Please enter a file name :

File 1

File Successfully Created: /Users/sackale/Downloads/Java\_FSD\_phase1/File 1

\*\*\*\*\*

\*\*\*\*\*

```
=====
|           SUB MENU            |
=====
| Select any one of the following: |
|   1 - Add a file                |
|   2 - Delete a file            |
|   3 - Search a file            |
|   4 - Go Back                  |
|                                |
=====
```

Enter your choice :

1

==> Adding a File...

Please enter a file name :

File 2

File Successfully Created: /Users/sackale/Downloads/Java\_FSD\_phase1/File 2

\*\*\*\*\*

\*\*\*\*\*

```
=====
|          SUB MENU          |
=====
| Select any one of the following: |
| 1 - Add a file                |
| 2 - Delete a file             |
| 3 - Search a file             |
| 4 - Go Back                   |
=====
```

Enter your choice :

1

==> Adding a File...

Please enter a file name :

file3

File Successfully Created: /Users/sackale/Downloads/Java\_FSD\_phase1/file3

\*\*\*\*\*

\*\*\*\*\*

```
=====
|          SUB MENU          |
=====
| Select any one of the following: |
| 1 - Add a file                |
| 2 - Delete a file             |
| 3 - Search a file             |
| 4 - Go Back                   |
=====
```

Enter your choice :

1

==> Adding a File...

Please enter a file name :

fIlE\_4

File Successfully Created: /Users/sackale/Downloads/Java\_FSD\_phase1/fIlE\_4

\*\*\*\*\*

```

=====
|               MAIN MENU               |
=====
| Select any one of the following:      |
|   1 - List All Files                  |
|   2 - More Options                    |
|   3 - Exit                            |
=====

```

Enter your choice :

1

\*\*\*\*\*

The Files in /Users/sackale/Downloads/Java\_FSD\_phase1 are:

File 1

File 2

fIlE\_4

file3

Total Number of files: 4

\*\*\*\*\*

\*\*\*\*\*

```

=====
|               MAIN MENU               |
=====
| Select any one of the following:      |
|   1 - List All Files                  |
|   2 - More Options                    |
|   3 - Exit                            |
=====

```

Enter your choice :

2

```

=====
|               SUB MENU                |
=====
| Select any one of the following:      |
|   1 - Add a file                      |
|   2 - Delete a file                  |
|   3 - Search a file                  |
|   4 - Go Back                        |
=====

```

Enter your choice :

3

==> Searching a File...

Please enter a file name to Search :

File 2

File Found at location: /Users/sackale/Downloads/Java\_FSD\_phase1

\*\*\*\*\*

\*\*\*\*\*

```
=====
|          SUB MENU          |
=====
| Select any one of the following: |
|   1 - Add a file               |
|   2 - Delete a file            |
|   3 - Search a file            |
|   4 - Go Back                  |
=====
```

Enter your choice :

3

==> Searching a File...

Please enter a file name to Search :

file 4

File Not Found.. Please try again.

\*\*\*\*\*

\*\*\*\*\*

```
=====
|          SUB MENU          |
=====
| Select any one of the following: |
|   1 - Add a file               |
|   2 - Delete a file            |
|   3 - Search a file            |
|   4 - Go Back                  |
=====
```

Enter your choice :

2

==> Deleting a File...

Please enter a file name to Delete :

FILE\_4

File deleted Successfully

\*\*\*\*\*

```

*****

=====
|          SUB MENU          |
=====
| Select any one of the following: |
|   1 - Add a file             |
|   2 - Delete a file          |
|   3 - Search a file          |
|   4 - Go Back                |
=====
Enter your choice :
4

=====
|          MAIN MENU         |
=====
| Select any one of the following: |
|   1 - List All Files          |
|   2 - More Options           |
|   3 - Exit                    |
=====
Enter your choice :
3

Are you sure you want to exit ?
(Y) ==> Yes    (N) ==> No
y

*****
*****
*                                     *
*   THANK YOU FOR VISITING LOCKEDME.COM   *
*                                     *
*****
*****

```

## Source Code

### Main. java

```

1 package org.example;
2
3 import Functionality.Menu;
4
5 public class Main {
6
7     public static final String path = "/Users/sackale/Downloads/Java_FSD_phase1";
8
9     public static void main(String[] args) {
10         Menu menu = new Menu();
11         menu.introScreen();
12         menu.mainMenu();
13     }
14
15 }

```



## Menu.java

```
1 package Functionality;
2
3 import Options.FileOptions;
4
5
6 public class Menu {
7     Scanner scan = new Scanner(System.in);
8     FileOptions dao = new FileOptions();
9
10    public void introScreen() {
11        System.out.println();
12        System.out.println("*****");
13        System.out.println("*      DEVELOPED BY Sachin Suresh Kale      *");
14        System.out.println("*****");
15        System.out.println("*      LOCKEDME.COM      *");
16        System.out.println("*****");
17        System.out.println("\n");
18    }
19
20    public void exitScreen() {
21        System.out.println("*****");
22        System.out.println("*****");
23        System.out.println("*      *");
24        System.out.println("*      THANK YOU FOR VISITING LOCKEDME.COM      *");
25        System.out.println("*      *");
26        System.out.println("*****");
27        System.out.println("*****");
28        System.out.println("\n");
29    }
30
31    public void Firstprompt() {
32        System.out.println("=====");
33        System.out.println("|          MAIN MENU          |");
34        System.out.println("=====");
35        System.out.println("| Select any one of the following: |");
36        System.out.println("| 1 - List All Files             |");
37        System.out.println("| 2 - More Options              |");
38        System.out.println("| 3 - Exit                     |");
39        System.out.println("=====");
40        System.out.println("Enter your choice : ");
41    }
42
43    public void subOptions() {
44        System.out.println("=====");
45        System.out.println("|          SUB MENU          |");
46        System.out.println("=====");
47        System.out.println("| Select any one of the following: |");
48        System.out.println("| 1 - Add a file                |");
49        System.out.println("| 2 - Delete a file            |");
50        System.out.println("| 3 - Search a file            |");
51        System.out.println("| 4 - Go Back                  |");
52        System.out.println("=====");
53        System.out.println("Enter your choice : ");
54    }
55
56 }
```



```

60 public void mainMenu() {
61
62     int choice = 0;
63     char decision = 0;
64     do {
65
66         Firstprompt();
67
68         try {
69             choice = Integer.parseInt(scan.nextLine());
70         } catch (NumberFormatException e) {
71             System.out.println("\nInvalid Input \nValid Input Integers:(1-3)\n");
72             mainMenu();
73         }
74
75
76         switch (choice) {
77
78             case 1:
79                 System.out.println();
80                 try {
81                     dao.listAllFiles(Main.path);
82                 } catch (NullPointerException e) {
83                     System.out.println(e.getMessage());
84                 } catch (IllegalArgumentException e) {
85                     System.out.println(e.getMessage());
86                 } catch (Exception e) {
87                     System.out.println(e.getMessage());
88                 }
89                 System.out.println("\n*****\n");
90                 break;
91
92             case 2:
93                 System.out.println();
94                 subMenu();
95                 break;
96
97             case 3:
98                 System.out.println("\n Are you sure you want to exit ? ");
99                 System.out.println(" (Y) ==> Yes (N) ==> No ");
100                decision = scan.nextLine().toUpperCase().charAt(0);
101                if (decision == 'Y') {
102                    System.out.println("\n");
103                    exitScreen();
104                    System.exit(1);
105                } else if (decision == 'N') {
106                    System.out.println("\n");
107                    mainMenu();
108                } else {
109                    System.out.println("\nInvalid Input \nValid Inputs :(Y/N)\n");
110                    mainMenu();
111                }
112
113             default:
114                 System.out.println("\nInvalid Input \nValid Input Integers:(1-3)\n");
115                 mainMenu();
116
117         }
118
119     } while (true);
120
121 }
122
123

```

```

124 public void subMenu() {
125     String file = null;
126     String fileName = null;
127     int choice = 0;
128
129     do {
130
131         subOptions();
132
133         try {
134             choice = Integer.parseInt(scan.nextLine());
135         } catch (NumberFormatException e) {
136             System.out.println("Invalid Input \nValid Input Integers:(1-4)");
137             subMenu();
138         }
139
140
141         switch (choice) {
142             case 1:
143                 System.out.println("\n==> Adding a File...");
144                 System.out.println("Please enter a file name : ");
145                 file = scan.nextLine();
146                 fileName = file.trim();
147                 try {
148                     dao.createNewFile(Main.path, fileName);
149                 } catch (NullPointerException e) {
150                     System.out.println(e.getMessage());
151                 } catch (IOException e) {
152                     System.out.println("Error occurred while adding file..");
153                     System.out.println("Please try again..");
154                 } catch (Exception e) {
155                     System.out.println("Error occurred while adding file..");
156                     System.out.println("Please try again..");
157                 }
158                 System.out.println("\n*****\n");
159                 break;
160
161             case 2:
162                 System.out.println("\n==> Deleting a File...");
163                 System.out.println("Please enter a file name to Delete : ");
164                 file = scan.nextLine();
165                 fileName = file.trim();
166                 try {
167                     dao.deleteFile(Main.path, fileName);
168                 } catch (NullPointerException e) {
169                     System.out.println(e.getMessage());
170                 } catch (IOException e) {
171                     System.out.println("Error occurred while Deleting File..");
172                     System.out.println("Please try again..");
173                 } catch (Exception e) {
174                     System.out.println("Error occurred while Deleting File..");
175                     System.out.println("Please try again..");
176                 }
177                 System.out.println("\n*****\n");
178
179             case 3:
180                 System.out.println("\n==> Searching a File...");
181                 System.out.println("Please enter a file name to Search : ");
182                 file = scan.nextLine();
183                 fileName = file.trim();
184                 try {
185                     dao.searchFile(Main.path, fileName);
186                 } catch (NullPointerException e) {
187                     System.out.println(e.getMessage());
188                 } catch (IllegalArgumentException e) {
189                     System.out.println(e.getMessage());
190                 } catch (Exception e) {
191                     System.out.println(e.getMessage());
192                 }
193                 System.out.println("\n*****\n");
194                 break;
195             case 4: mainMenu();
196                 break;
197
198             default:
199                 System.out.println("Invalid Input \nValid Input Integers:(1-4)");
200                 subMenu();
201
202         }
203
204         file = null;
205         fileName = null;
206
207     } while (true);
208 }
209
210 }
211
212 }
213

```

## FileOptions.java

```
1 package Options;
2
3 import java.io.File;
4
5 public class FileOptions {
6     public void listAllFiles(String path) {
7         if (path == null || path.isEmpty() || path.isEmpty())
8             throw new NullPointerException("Path cannot be Empty or null");
9
10        File dir = new File(path);
11
12        if(!dir.exists())
13            throw new IllegalArgumentException("Path does not exist");
14
15        if(dir.isFile())
16            throw new IllegalArgumentException("The given path is a file. A directory is expected.");
17
18        String [] files = dir.list();
19
20        System.out.println("\n*****");
21        if(files != null && files.length > 0) {
22            Set<String> fileList = new TreeSet<String>(Arrays.asList(files));
23            System.out.println("The Files in " + dir.getAbsolutePath() + " are: \n");
24            for(String file1:fileList) {
25                System.out.println(file1);
26            }
27
28            System.out.println("\nTotal Number of files: " + fileList.size());
29        }else {
30            System.out.println("Directory is Empty");
31        }
32    }
33
34    public void createNewFile(String path , String fileName) throws IOException {
35        if (path == null || path.isEmpty() || path.isEmpty())
36            throw new NullPointerException("Path cannot be Empty or null");
37
38        if (fileName == null || fileName.isEmpty() || fileName.isEmpty())
39            throw new NullPointerException("File Name cannot be Empty or null");
40
41        File newFile = new File(path + File.separator + fileName);
42
43        boolean createFile = newFile.createNewFile();
44
45        if (createFile) {
46            System.out.println("\nFile Successfully Created: " + newFile.getAbsolutePath());
47        }else if(!createFile) {
48            System.out.println("\nFile Already Exist.. Please try again. ");
49        }
50    }
51
52    public void deleteFile(String path , String fileName) throws IOException {
53
54    }
```

```

76 public void deleteFile(String path , String fileName) throws IOException {
77
78     if (path == null || path.isEmpty() || path.isEmpty())
79         throw new NullPointerException("Path cannot be Empty or null");
80
81
82     if (fileName == null || fileName.isEmpty() || fileName.isEmpty())
83         throw new NullPointerException("File Name cannot be Empty or null");
84
85     File newFile = new File(path + File.separator + fileName);
86
87     boolean deleteFile = newFile.delete();
88
89     if (deleteFile) {
90         System.out.println("\nFile deleted Successfully");
91     }
92     else {
93         System.out.println("\nFile Not Found.. Please try again." );
94     }
95 }
96
97 }
98
99 }
100
101
102
103 public void searchFile(String path , String fileName){
104
105     if (path == null || path.isEmpty() || path.isEmpty())
106         throw new NullPointerException("Path cannot be Empty or null");
107
108
109     if (fileName == null || fileName.isEmpty() || fileName.isEmpty())
110         throw new NullPointerException("File Name cannot be Empty or null");
111
112     File dir = new File(path);
113
114     if(!dir.exists())
115         throw new IllegalArgumentException("Path does not exist");
116
117     if(dir.isFile())
118         throw new IllegalArgumentException("The given path is a file. A directory is expected.");
119
120
121     String [] fileList = dir.list();
122     boolean flag = false;
123
124     Pattern pat = Pattern.compile(fileName);
125
126     if(fileList != null && fileList.length > 0) {
127         for(String file:fileList) {
128             Matcher mat = pat.matcher(file);
129             if(mat.matches()) {
130                 System.out.println("File Found at location: " + dir.getAbsolutePath());
131                 flag = true;
132                 break;
133             }
134         }
135     }
136
137     if(flag == false)
138         System.out.println("File Not Found.. Please try again.");
139
140 }
141
142 }
143 }
144

```

## **Conclusion**

- 1: The prototype is robust and platform independent.
- 2: User can easily use the prototype and safely exit out of it.
- 3: The prototype has a good interface with CLI (Command Line Interface).
- 4: As a developer, we can enhance it by introducing several new features such as appending in a file or overwriting a file and the file details for which user selected.
- 5: This prototype though is robust but user can only interact it with terminal or CLI so we can develop a good GUI interface for more better user-friendly.