

# DOCUMENTATION OF PRODUCT

COMPANY NAME - Lockers pvt. Ltd

PRODUCT NAME - LockedMe.com

DEVELOPER'S NAME - Kartik Dixit

GITHUB LINK - <https://github.com/kardixit/JAVA-FSD-PHASE-1-PROJECT.git>

**About Product** - The Product "LockedMe.com" is a CLI(Command Line Interface) based application. This application can be used for various file operations such as creating new files. Delete and search the expand files and keep track of all the files present in the directory in a sorted manner.

## Language & IDE Used

Java8 & IntelliJ.

## Sprint Planning

The Sprints are planned in two halves illustrated below:

### Sprint 1:

- Understanding the requirements of the Product/Application
- Designing the Flow Chart Of The Application
- Implementing Main Menu Functionality.
- Implementing the functionality to close the application on User's Input.

### Sprint 2:

- Implementing File Menu Functionality.
- Testing File operations such as adding, deleting, searching file in the Directory.

- Optimising File operations for better results.
- Creating Documentation and Read me file for the product.

## WORKING OF THE APPLICATION

- WELCOME SCREEN

[illegible]

- WHEN THERE IS NO FILE

```
1
/Users/kardixit/Desktop/JAVA_FSD_Phase-1/Project_Phase_1/FileSystem/target/files/
No existing files

MAIN MENU

Please select an option from below

-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter your choice:
|
```

- WHEN THERE IS EXISTING FILES

```
Enter your choice:

1
/Users/kardixit/Desktop/JAVA_FSD_Phase-1/Project Phase 1/FileSystem/target/files/
KD.txt
MAVERICK.txt
KARAM.txt
DATACENTER.txt

MAIN MENU

Please select an option from below

-> Press 1 To Show Existing Files
```

- SHOWING FILE OPTIONS MENU

```
Enter your choice:

1
/Users/kardixit/Desktop/JAVA_FSD_Phase-1/Project Phase 1/FileSystem/target/files/
KD.txt
MAVERICK.txt
KARAM.txt
DATACENTER.txt

MAIN MENU

Please select an option from below

-> Press 1 To Show Existing Files
```

- ADD A FILE

```
COMMANDO
/Users/kardixit/Desktop/JAVA FSD Phase-1/Project Phase 1/FileSystem/target/files/COMMANDO.txt
File created Successfully : COMMANDO
FILE OPTIONS MENU

Please select an option from below

-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter your choice:
```

- DELETING A FILE

```
KD
/Users/kardixit/Desktop/JAVA FSD Phase-1/Project Phase 1/FileSystem/target/files/KD.txt
File KD.txt has been deleted successfully

FILE OPTIONS MENU

Please select an option from below

-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter your choice:
```

- SEARCH A FILE

```
MAVERICK
/Users/kardixit/Desktop/JAVA_FSD_Phase-1/Project_Phase_1/FileSystem/target/files/
File present in the Application

FILE OPTIONS MENU

Please select an option from below

-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter your choice:
```

- RETURN TO MAIN MENU

```
4

End of Business Operations

MAIN MENU

Please select an option from below

-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter your choice:
```

- QUIT THE APPLICATION

MAIN MENU

Please select an option from below

-> Press 1 To Show Existing Files  
-> Press 2 To Show File Options Menu  
-> Press 3 To Quit The Application  
Enter your choice:

3

End of application.Thank you

Process finished with exit code 0

|

# SOURCE CODE

## SOURCE CODE OF BUSINESS OPERATIONS

```
1 package FileSystem;
2
3 import ...
4
5
6
7 public class BusinessOperations {
8
9     1 usage
10
11     public static void main() throws IOException, URISyntaxException {
12
13         Scanner sc = new Scanner(System.in);
14         boolean flag = true;
15         while(flag){
16             System.out.println("FILE OPTIONS MENU");
17             System.out.println("\nPlease select an option from below\n");
18             // FileResourcesUtils.main("screen2");
19             System.out.println("-> Press 1 To Add a File");
20             System.out.println("-> Press 2 To Delete a File");
21             System.out.println("-> Press 3 To Search a File");
22             System.out.println("-> Press 4 To Return To Menu");
23             System.out.println("Enter your choice:");
24             System.out.println();
25             // System.out.println();
26             int opt = sc.nextInt();
27             switch (opt){
28                 case 1:
29                     System.out.println("\nPlease enter the file name\n");
30                     String name = sc.next();
31                     Files.CreateFile(name);
32                     break;
33                 case 2:
34                     System.out.println("\nPlease enter the file name you want to delete");
35                     String nAme = sc.next();
36                     Files.Delete(nAme);
37                     System.out.println();
38                     break;
39                 case 3:
40                     System.out.println("\nPlease enter the file name you want to search\n");
41                     String Name = sc.next();
42                     Files.searchFiles(Name);
43                     System.out.println();
44                     break;
45                 case 4:
46                     flag = false;
47                     break;
48                 default:
49                     System.out.println("\nPlease select a valid option\n");
50             }
51         }
52     }
53 }
```

```
47         System.out.println("Please select a valid option");
48     }
49 }
50 System.out.println("\nEnd of Business Operations\n");
51
52
53 }
54 }
```



## SOURCE CODE FOR MAIN

```
1  package FileSystem;
2
3  import ...
4
5  no usages
6
7
8  ▶ public class Main {
9      no usages
10
11     ▶ public static void main(String[] args) throws IOException, URISyntaxException {
12
13         Screens screens = new Screens();
14         Scanner sc = new Scanner(System.in);
15
16         Screens.main();
17
18         boolean flag = true;
19
20         while (flag){
21             System.out.println("MAIN MENU");
22             System.out.println("\nPlease select an option from below\n");
23             // FileResourcesUtils.main("screen1");
24             System.out.println("-> Press 1 To Show Existing Files");
25             System.out.println("-> Press 2 To Show File Options Menu");
26             System.out.println("-> Press 3 To Quit The Application");
27             System.out.println("Enter your choice:");
28             System.out.println();
29             // System.out.println();
30             int opt = sc.nextInt();
31             switch (opt){
32                 case 1:
33                     Files.ListAllFiles();
34                     System.out.println();
35                     // System.out.println();
36                     break;
37                 case 2:
38                     BusinessOperations.main();
39                     break;
40                 case 3:
41                     flag = false;
42                     break;
43                 default:
44                     System.out.println("\nPlease select a valid option\n");
45                     System.out.println();
46                     break;
47             }
48         }
49
50         System.out.println("End of application.Thank you");
51     }
52 }
```

## SOURCE CODE FOR SCREENS

```
1 package FileSystem;
2
3 import java.util.ArrayList;
4
5 5 usages
6 public class Screens {
7
8     1 usage
9     public static void main(){
10         Screens view = new Screens();
11         System.out.println();
12         view.WelcomeString();
13     }
14
15     1 usage
16     public void WelcomeString(){
17         int count = 20;
18         String productName = "*!*!*!*!*!*!*!* PRESENTING YOU APPLICATION LOCKEDME.COM BY I
19         String developerName = "\t\t\t\tDeveloped By -> KARTIK DIXIT";
20         System.out.println();
21         System.out.println(productName);
22         System.out.println(developerName);
23         System.out.println("=====");
24         System.out.println();
25     }
26 }
```

## SOURCE CODE FOR FILES

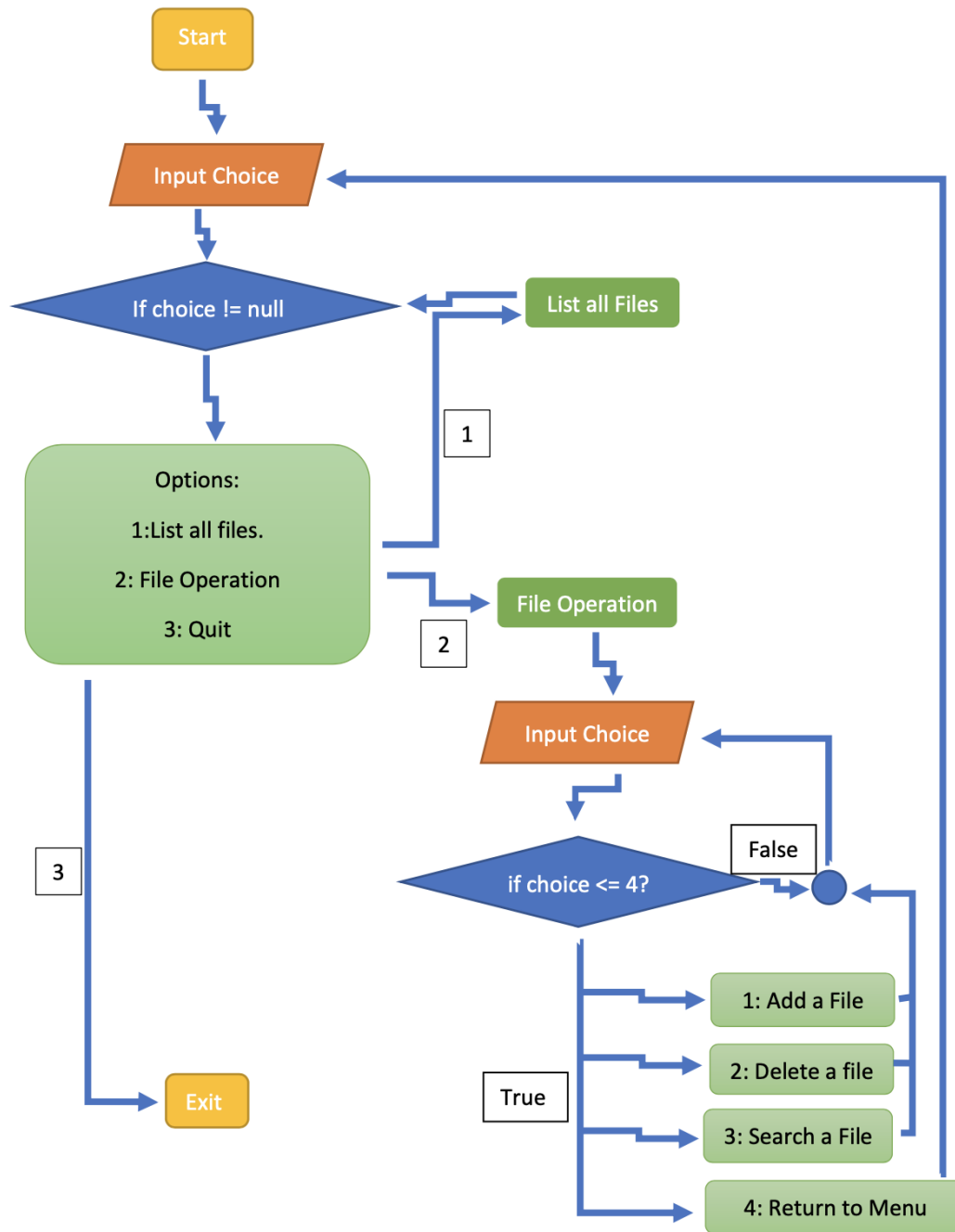
```
1 package FileSystem;
2
3 import java.io.*;
4 import java.io.IOException;
5
6 public class Files {
7     final static String dir = System.getProperty("user.dir")+"/FileSystem/target/files/";
8     public static void CreateFile(String name){
9         // String dir = System.getProperty("user.dir");
10        System.out.println(dir+name+".txt");
11        try{
12            File obj = new File( pathname: dir+name+".txt");
13            if (obj.createNewFile()){
14                System.out.println("File created Successfully : "+name);
15            }
16            else {
17                System.out.println("File already present! ");
18            }
19        } catch (IOException e){
20            System.out.println("An error occurred");
21            e.printStackTrace();
22        }
23    }
24    public static void searchFiles(String name){
25        String[] ls = getFiles();
26        if (name == null || ls == null) {
27            System.out.println("\nFiles does not exist\n");
28        } else if (ls != null) {
29            for (String str:ls){
30                // System.out.println(str+" "+name+".txt"+str.substring(0,str.length() - 4));
31
32                if (str.substring(0,str.length() - 4).equals(name)){
33                    System.out.println("File present in the Application");
34                }
35            }
36        }
37    }
38 }
```

```

35         }
36     }
37 }
38     System.out.println("File not present in the application");
39 }
    2 usages
40 public static String[] getFiles(){
41     System.out.println(dir);
42     File obj = new File(dir);
43     String contents[] = obj.list();
44     return contents;
45 }
    1 usage
47 public static void ListAllFiles(){
48     String[] ls = getFiles();
49     if(ls.length==0){
50         System.out.println("No existing files");
51     }else{
52         for(String str:ls){
53             System.out.println(str);
54         }
55     }
56
57     System.out.println();
58
59 }
    1 usage
61 public static void Delete(String name){
62     File file = new File( pathname: dir+"/"+name+".txt");
63     System.out.println(file);
64     // searchFiles(name);
65     if (file.delete()){
66         System.out.println("File "+name+".txt has been deleted successfully");
67     }else {
68         System.out.println("Unable to delete the file");

```

# FLOW CHART



# CONCLUSION

- The Prototype has a good interface with CLI(Command Line Interface).
- The Prototype is platform independent and robust.
- We can enhance the prototype by a good GUI for more friendly.
- This Prototype can also be upgraded with adding authentication and security features to make it more secure and reliable.
- Users can easily use this prototype and perform the operations.