```java
package org.example;

import Functioanlity.Menu;

public class Main {
        public static final String path = "C:\\Users\\amitya2\\Downloads\\Lockedme";

        public static void main(String[] args) {
            Menu menu = new Menu();
            menu.introScreen();
            menu.mainMenu();
        }

}
```

```java
package Options;

import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class FileOptions {
    public void listAllFiles(String path) {

        if (path == null || path.isEmpty() || path.isEmpty())
            throw new NullPointerException("Path cannot be Empty or null");


        File dir = new File(path);

        if(!dir.exists())
            throw new IllegalArgumentException("Path does not exist");

        if(dir.isFile())
            throw new IllegalArgumentException("The given path is a file. A directory is expected.");

        String [] files = dir.list();

        System.out.println("\n**********************************");
        if(files != null && files.length > 0) {

            Set<String> filesList = new TreeSet<String>(Arrays.asList(files));
            System.out.println("The Files in "+ dir.getAbsolutePath() + " are: \n");
            for(String file1:filesList) {
```

```
35
36                  System.out.println(file1);
37
38              }
39
40              System.out.println("\nTotal Number of
   files: "+ filesList.size());
41          }else {
42
43              System.out.println("Directory is Empty");
44          }
45
46      }
47
48
49      public void createNewFile(String path , String
   fileName) throws IOException {
50
51          if (path == null || path.isEmpty() || path.
   isEmpty())
52              throw new NullPointerException("Path
   cannot be Empty or null");
53
54
55          if (fileName == null || fileName.isEmpty
   () || fileName.isEmpty())
56              throw new NullPointerException("File Name
    cannot be Empty or null");
57
58          File newFile = new File(path + File.separator
    + fileName);
59
60          boolean createFile = newFile.createNewFile();
61
62          if (createFile) {
63
64              System.out.println("\nFile Successfully
   Created: " + newFile.getAbsolutePath());
65
66          }else if(!createFile) {
67
```

```
68              System.out.println("\nFile Already Exist
    .. Please try again." );
69
70          }
71
72      }
73
74
75
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
91              System.out.println("\nFile deleted
    Successfully");
92
93          }else {
94
95              System.out.println("\nFile Not Found..
    Please try again." );
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 
```

```java
package Functioanlity;

import Options.FileOptions;
import org.example.Main;

import java.io.IOException;
import java.util.Scanner;

public class Menu {
    Scanner scan = new Scanner(System.in);
    FileOptions dao = new FileOptions();

    public void introScreen() {
        System.out.println();
        System.out.println(
"************************************************");
        System.out.println("*             DEVELOPED
BY Amit Yadav           *");
        System.out.println(
"************************************************");
        System.out.println("*
LOCKEDME.COM                    *");
        System.out.println(
"************************************************");
        System.out.println("\n\n");
    }
    public void exitScreen() {

        System.out.println(
"************************************************");
        System.out.println(
"************************************************");
        System.out.println(
"*                                              *");
        System.out.println("*    THANK YOU FOR
VISITING LOCKEDME.COM        *");
        System.out.println(
"*                                              *");
        System.out.println(
"************************************************");
        System.out.println(
```

```
30 "*********************************************");
31         System.out.println("\n\n");
32 
33     }
34     public void Firstprompt() {
35         System.out.println(
   "====================================");
36         System.out.println("|            MAIN MENU
                 |");
37         System.out.println(
   "====================================");
38         System.out.println("| Select any one of the
   following:  |");
39         System.out.println("|   1 - List All Files
                 |");
40         System.out.println("|   2 - More Options
                  |");
41         System.out.println("|   3 - Exit
                           |");
42         System.out.println(
   "====================================");
43         System.out.println("Enter your choice : ");
44     }
45     public void subOptions() {
46 
47         System.out.println(
   "====================================");
48         System.out.println("|            SUB MENU
                 |");
49         System.out.println(
   "====================================");
50         System.out.println("| Select any one of the
   following:  |");
51         System.out.println("|   1 - Add a file
                    |");
52         System.out.println("|   2 - Delete a file
                 |");
53         System.out.println("|   3 - Search a file
                 |");
54         System.out.println("|   4 - Go Back
                       |");
```

```
55          System.out.println(
   "=====================================");
56          System.out.println("Enter your choice : ");
57
58      }
59
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("\n
    Invalid Input \nValid Inputs :(Y/N)\n");
110                          mainMenu();
111                      }
112
113
114                  default:
115                      System.out.println("\nInvalid
    Input \nValid Input Integers:(1-3)\n");
116                      mainMenu();
117
118              }
119
120
121          }while(true);
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 \n
    Valid 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                     break;
179 
180                 case 3:
181                     System.out.println("\n==>
    Searching a File...");
182                     System.out.println("Please enter
     a file name to Search : ");
183                     file = scan.nextLine();
```

```
184                         fileName = file.trim();
185                         try {
186                             dao.searchFile(Main.path,
    fileName);
187                         }catch(NullPointerException e) {
188                             System.out.println(e.
    getMessage());
189                         }catch(IllegalArgumentException
    e) {
190                             System.out.println(e.
    getMessage());
191                         }catch(Exception e) {
192                             System.out.println(e.
    getMessage());
193                         }
194                         System.out.println("\n
    ************************************\n");
195                         break;
196                     case 4: mainMenu();
197                         break;
198 
199                     default:
200                         System.out.println("Invalid
    Input \nValid Input Integers:(1-4)");
201                         subMenu();
202 
203                 }
204 
205                 file = null;
206                 fileName = null;
207 
208             }while(true);
209 
210         }
211 
212 }
213 
```