LockedMe.com

Submitted to

Simplilearn

Submitted by

Omar Turki Alsulaiteen – KFH Forssa Trainee

GitHub Link

12 December 2022

Table of Contents

Table of Figures

1 Introduction

LockedMe is a prototype to show the power of digitizing the company. It's a command line that allows the user to parse through the directory, add a file to the directory, remove it, and search for it. It will help the company and reduce its time by avoiding unnecessary work.

2 Implementation Details

The project is implemented using the Java language. The project contains three classes, FolderManager class, the Input class, and the Main class.

2.1 FolderManager Class

The FolderManager class implements the main functionality of the program. It Manipulates the directory by adding a new file, deleting it, and searching for it.

2.1.1 Members

private String root: the path for the directory that stores the files.

private ArrayList<File> files: the collection (data structure) used to work with the directory.

2.1.2 Methods

2.1.2.1 Constructor

```
25    public FolderManager(String root) {
26        this.root = root;
27
28        File directory = new File(this.root);
29
30        files = new ArrayList<File>(Arrays.asList(directory.listFiles()));
31    }
32
```

Figure 1: Constructor

In The constructor, the root is defined. Also, all the files in the directory are added to the array list.

2.1.2.2 *Displaying the files in the directory*

```
37    public void listFiles() {
38        System.out.println("\nCurrent files in the directory:");
39        files.sort((o1, o2) -> o1.getName().compareTo(o2.getName()));
40        for (File file : files) {
41            System.out.println(file.getName());
42        }
43    }
```

Figure 2: listFiles method is used to print all the files in the directory

listFiles method is used to print all the files in the directory. In addition, all the files are sorted ascendingly using the method compare (lambda expression).

2.1.2.3 *Adding a file to the directory*

```
50    public void addFile(String fname) {
51        File newFile = new File(getRoot() + "/" + fname);
52        try {
53            if (newFile.createNewFile()) {
54                System.out.println(newFile.getName() + " Created successfully");
55                if (!files.add(newFile))
56                    System.out.println("Error adding" + newFile.getName() + " file");
57
58            } else {
59                System.out.println("A file with the same name already exists");
60            }
61        } catch (IOException e) {
62            System.out.println("IOException\n" + e.getMessage());
63        }
64
65    }
```

Figure 3: addFile method is used to add a file to the directory

The addFile method takes the file name from the user and adds it. If a file with the same name already exists, the user will be notified that a file with the same already exists.

2.1.2.4 *Deleting a file*

```
72    public void deleteFile(String fname) {
73        File file = new File(this.getRoot() + "/" + fname);
74        if (file.delete()) {
75            System.out.println("File deleted successfully");
76            if (!files.remove(file))
77                System.out.println("Error removing file from the list\n");
78
79        } else {
80            System.out.println("File not found!!!");
81        }
82
83    }
```

Figure 4: deleteFile method is used to delete a file from the directory

The deleteFile method is used to delete a file from the directory. It checks whether the file exists or not.

2.1.2.5 *Searching for a file*

```
88    public void searchFile(String fname) {
89        File file = new File(getRoot() + '/' + fname);
90        if (files.contains(file)) {
91            System.out.println(fname + " exists");
92        } else {
93            System.out.println(fname + " doesn't exist");
94        }
95    }
96 }
```

Figure 5: searchFile method

searchFile method is used to search for a file in the directory.

2.2 Input Class

Input class is used to handle the thrown exceptions from the user.

2.2.1 Handling Integer Inputs

```
105     public static int getInteger() {
106         Scanner input = new Scanner(System.in);
107         System.out.print("\nEnter a number: ");
108         int n = -1;
109         try {
110             n = input.nextInt();
111             input.nextLine();
112         } catch (InputMismatchException e) {
113             return -1;
114         }
115         return n;
116     }
```

Figure 6: getInteger method, which used to handle the user's input

The getInteger method takes input from the user and returns -1 if there is an invalid error. Later this value is used in the Main class to be handled.

2.2.2 Handling String Inputs

```
118     public static String getString(String task) {
119         Scanner input = new Scanner(System.in);
120         System.out.print(task);
121         String str;
122         try {
123             str = input.nextLine();
124         } catch (InputMismatchException e) {
125             return null;
126         }
127         return str;
128     }
129 }
```

Figure 7: getString method

getString method is used to handle the strings that the user enters. The method will return a null value if an invalid input is entered. Which will be handled later in the Main class.

2.3 Main Class

The Main class contains the command line. It includes three commands. It starts with a welcome message. After that, the user must enter one, two, or three to continue. The commands are as follows:

1. List the files in the current directory
2. Add/Delete/Search for a file
3. Exit the program

```
34              case (1):
35                  directory.listFiles();
36                  break;
37              case (2):
38                  loop2: while (true) {
39
40                      int k = 0;
41                      String fileName;
42                      System.out.println("\nPlease choose one of the following options\n1. Add a file to the directory");
43                      System.out.println("2. Delete a file \n3. Search for a file\n4. Go back to previous settings");
44                      k = Input.getInteger();
45
46                      // The second switch statement is used to handle the add, delete and the search
47                      // options are: add, delete and search for a file.
48                      switch (k) {
49                      case (1):
50                          fileName = Input.getString("Enter the file name for the new file: ");
51                          directory.addFile(fileName);
52                          break;
53                      case (2):
54                          fileName = Input.getString("Enter the file name you want to delete: ");
55                          directory.deleteFile(fileName);
56                          break;
57                      case (3):
58                          fileName = Input.getString("Enter the file name you want to search for: ");
59                          directory.searchFile(fileName);
60                          break;
61                      case (4):
62                          System.out.println();
63                          break loop2;
64                      default:
65                          System.out.println("Invalid value, please enter a valid value");
66                      }
67                      Input.getString("\nPress Enter to Continue");
68                      System.out.println();
69                  }
70                  break;
71
72              case (3):
73                  System.out.println("Thank you for using LockedMe. See you next time!!!");
74                  break loop;
75              default:
76                  System.out.println("Invalid value, please enter a valid value");
```

Figure 8: The main part of the Main class (command line)

3 Output Screenshots

```
Welcom to LockedMe.com
Developer details:
Developer's name: Omar Alsulaiteen
Job Title: LockedMe Developer

Please choose one of the following options
1. List all the files in the directory
2. Add/Delete/Search for a file
3. Exit the program
```

Figure 9: Welcome screenshot

```
Please choose one of the following options
1. List all the files in the directory
2. Add/Delete/Search for a file
3. Exit the program

Enter a number: 1

Current files in the directory:
newFile.txt
test.txt
trolololololo.txt
xyz.txt

Press Enter to Continue
```

Figure 10: Displaying files in the directory screenshot

```
Please choose one of the following options
1. List all the files in the directory
2. Add/Delete/Search for a file
3. Exit the program

Enter a number: 2

Please choose one of the following options
1. Add a file to the directory
2. Delete a file
3. Search for a file
4. Go back to previous settings

Enter a number: 1
Enter the file name for the new file: omar.txt
omar.txt Created successfully

Press Enter to Continue
```

Figure 11: Adding a file screenshot

```
Please choose one of the following options
1. Add a file to the directory
2. Delete a file
3. Search for a file
4. Go back to previous settings

Enter a number: 2
Enter the file name you want to delete: omar.txt
File deleted successfully

Press Enter to Continue


Please choose one of the following options
1. Add a file to the directory
2. Delete a file
3. Search for a file
4. Go back to previous settings

Enter a number: 2
Enter the file name you want to delete: omar.txt
File not found!!!

Press Enter to Continue


Please choose one of the following options
1. Add a file to the directory
2. Delete a file
3. Search for a file
4. Go back to previous settings

Enter a number: 3
Enter the file name you want to search for: omar.txt
omar.txt doesn't exist

Press Enter to Continue
```

Figure 12: Deleting, and searching for a file

4 Sprints

The sprints are included in the GitHub repository. It will also be included here. In addition, the product backlog and the progress log are included in this document.

4.1 Product Backlog

LockedMe.com File Management Project

- LockedMe.com is a website for LockedMe Co. used as a prototype to digitize the company.

- The program supports displaying files in the directory.
    * The files are sorted in ascending order.

- The program allows users to add/delete/search for a file in the current directory.

    * The user can add files to the directory.
        # The file name must be unique. If it is not, a message will be displayed to the user.

    * The user can delete files from the directory.
        # The program will check if the file exists before deleting it.

    * The user can search for a file.

    * An option to return to the previous menu (The main one).

- The program is done by creating a Java project, and it contains.
    * Class FolderManager to manage the project. It contains
        # addFile method to add a file to the directory.
        # deleteFile method to delete a file from the directory.
        # SearchFile method to search for a file
        # listFiles method to list the files in the directory.

    * Class Main for the command line.

- Exception handling for the program

    * Exception handling for all of the user's inputs. It includes

        # The numbers entered when parsing through the menu

        # The files names entered by the user

    * Files I/O

- Sorting algorithms will be used to sort the files.

4.2 Sprint 1

Features included in sprint 1:

- GitHub repository to manage the project

- Create a Java project

    * Create a class FolderManager to manage the project. It contains

        # addFile method to add a file to the directory.

        # deleteFile method to delete a file from the directory.

        # SearchFile method to search for a file

        # listFiles method to list the files in the directory.

    * Create a class Main for the command line.

4.3 Sprint 2

Features included in sprint 2:

- Exception handling for the program
    * Exception handling for all of the user's inputs. It includes
        # The numbers entered when parsing through the menu
        # The files names entered by the user

    * Files I/O

- Documentation
    - Documentation will be created explaining the project

4.4 Progress log

20 NOV 2022:

* Omar Alsulaiteen: Created the java project, the FolderManager class, and the Main class.

21 NOV 2022:

* Omar Alsulaiteen: Created the listFiles method.

22 NOV 2022:

* Omar Alsualiteen: Created the addFile method.

23 NOV 2022:

* Omar Alsulaiteen: Created the delteFile method.

24 NOV 2022:

* Omar Alsulaiteen: Created the searchFile method.

27 NOV 2022:

* Omar Alsulaiteen: Created the Input class to handle the user input.

28 NOV 2022:

* Omar Alsulaiteen: Created the Main class, which includes the command line.

29 NOV 2022:

* Omar Alsulaiteen: Created the Documentation file.

30 NOV 2022:

* No Progress

1 DEC 2022:

* Omar Alsulaiteen: Created the flowchart to explain the project.

5 Flowchart

An activity diagram is used to show the flow of the program.

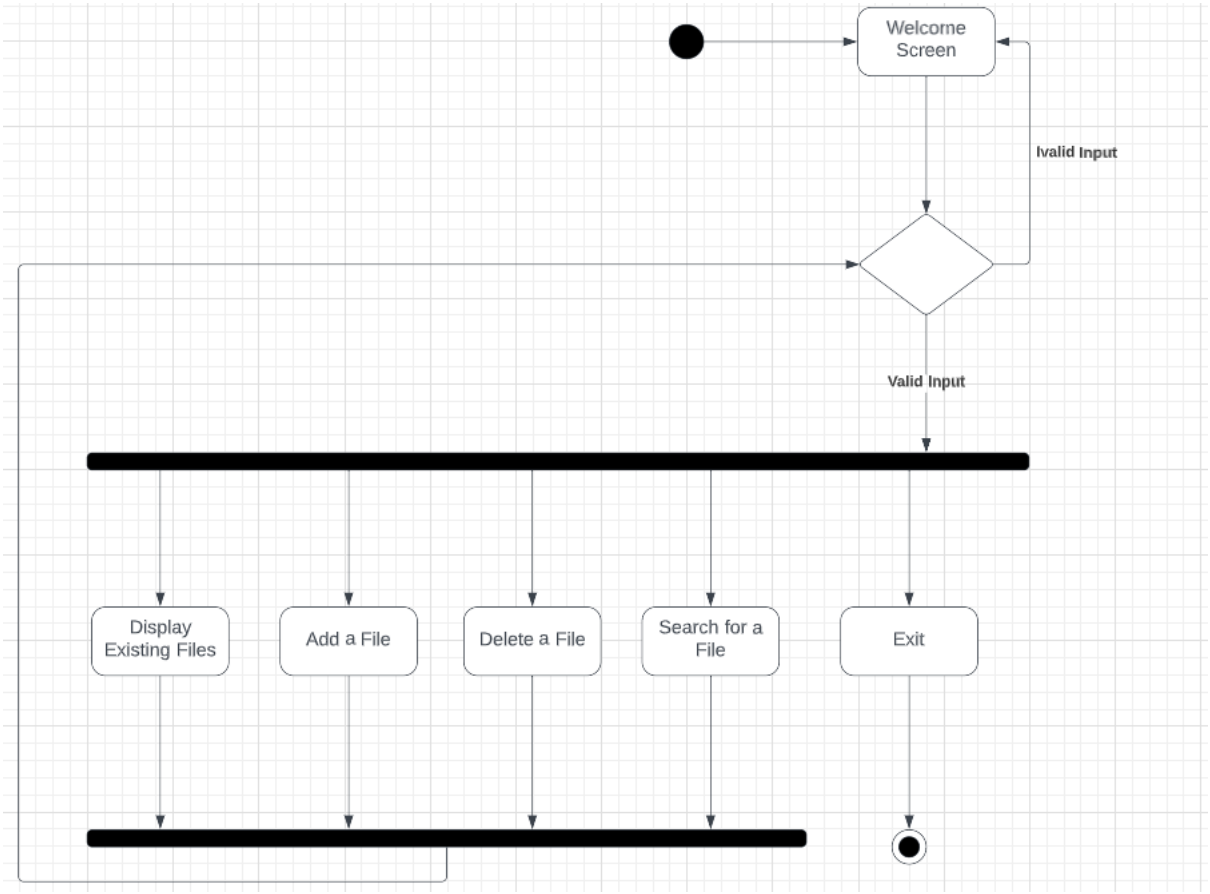

Figure 13: Activity Diagram

6 Conclusion

LockedMe.com is a great opportunity to make LockedMe Co. do a better, faster, and more efficient job. Expanding the project into production will result in a big profit for the company.