LockedMe Source Code

App.java

```
package frontEnd;

import java.util.*;
import ascending.Ascending;
import operations.Operations;

public class App {
  static Scanner sn = new Scanner(System.in);

        public static void info() {

                String appHeader =
String.format("**************************************************\n"
                                + "************* Welcome to LockedMe.com  *************\n"
                                + "**************************************************\n");

    String developerName = "Dev - Rajat Kumar Soni\n";

                String developerEmail = "DevEmail - rajatsoni8585@gmail.com\n";

                String appDetails = "You can use this application to :-\n" + "--> Retrieve all file names in a
given folder\n"
                                + "--> Add, delete or search files\n";

                System.out.println(appHeader);

          System.out.println(developerName);

                System.out.println(developerEmail);

                System.out.println(appDetails);

        }
        public static void main() {
                System.out.println("");
                System.out.println("Main Menu");
                System.out.println("Press 1 to show file in Ascending Order");
                System.out.println("Press 2 to view file operations");
                System.out.println("Press 3 to Exit from the application");

                int choice = sn.nextInt();
                handle(choice);
        }
        public static void handle(int num) {
                switch(num) {
                        case 1:
                                Ascending.ascendingOrder();
                                break;
                        case 2:
                                Operations.FileOperations();
                                break;
                        case 3:
                                System.out.println("Terminated :( ");
                                System.exit(0);
                                break;
                        default:
```

```
                        System.out.println("Invalid input");
                }
                main();
        }
        public static void main(String[] args) {
                info();
                main();

        }
}
```

Ascending.java

```
package ascending;

import java.io.*;
import java.util.*;

public class Ascending {

        static String directory= "/Users/rajasoni/Desktop/LockedMe/storage";
        public static void ascendingOrder() {

                File[] files = new File(directory).listFiles();
                Set<String> a = new TreeSet<>();
                for(File file : files) {
                        if (!file.isFile()) {
                                continue;
                        }
                        a.add(file.getName());
                }
                a.forEach(i->System.out.println(i));

        }
}
```

Operations.java

```
package operations;

import java.io.*;
import java.nio.file.*;
import java.util.*;
import frontEnd.App;

public class Operations {

        static Scanner sn = new Scanner(System.in);
        static String directory = "/Users/rajasoni/Desktop/LockedMe/storage";

        public static void FileOperations() {

                System.out.println("");
                System.out.println("Press 1 to Add a file");
                System.out.println("Press 2 to Delete a file");
                System.out.println("Press 3 to Search a file");
                System.out.println("Press 4 to go Back to the Main Menu");
                String choice = sn.nextLine();
                handle(choice);
        }

        public static void handle(String num) {
                switch(num) {
                        case "1":
                                System.out.println("You selected Add Operation");
                                add();
                                break;
                        case "2":
                                System.out.println("You selected Delete Operation");
                                delete();
                                break;
                        case "3":
                                System.out.println("You selected Search Operation");
                                search();
                                break;
                        case "4":
                                System.out.println("Going Back to Main Menu");
                                App.main();
                                break;
                        default:
                                System.out.println("Invalid input");
                }
                FileOperations();
        }

        // to add a file

        public static void add() throws InvalidPathException {
                System.out.println("Enter the file path (ex: /Users/Desktop/t.txt)");
                String input = sn.nextLine();
                Path path;
                try {
                        path = Paths.get(input);
                } catch (Exception e) {
                        System.out.println("Invalid input");
                        return;
                }

                if (!Files.exists(path)) {
                        System.out.println("No such file exist");
                        return;
                }else {
                        System.out.println("File is present");

                }

                String newPath = directory + "/" + path.getFileName();
                int i = 0;
                while (Files.exists(Paths.get(newPath))) {
                        i++;
                        newPath = directory + "/" + i + "_" + path.getFileName();
                }

                try {
                        Files.copy(path,  Paths.get(newPath));
                        System.out.println("file has been stored");
                } catch (IOException e) {
                        System.out.println("Not able to store the file");
                System.out.println(e);
            }

        }

        // to delete a file

        public static void delete() throws InvalidPathException {
            System.out.println("Enter the file path (ex: c.txt)");
            String input = sn.nextLine();
            String Path = directory + "/" + input;
            Path path;

            try {
                path = Paths.get(Path);
            } catch (Exception e) {
                System.out.println("Invalid input");
                return;
            }

            if (!Files.exists(path)) {
                System.out.println("No such file existed,thus cannot be deleted");
                return;
            } else {
                System.out.println("File is present");
            }

            File Delete = new File(Path);
            try {
                Delete.delete();
                System.out.println("File is deleted");
            }
            catch (Exception e) {

                System.out.println("Not able to delete file");
                System.out.println(e);
            }
        }

        //to search a file

        public static void search() throws InvalidPathException{
            System.out.println("Enter the file to search (ex: a.txt)");
            String input = sn.nextLine();
            String Path = directory + "/" + input;
            Path path;

            try {
                path = Paths.get(Path);
            } catch (Exception e) {
                System.out.println("Invalid input");
                return;
            }

            if(!Files.exists(path)) {
                System.out.println("No such file exist");
                return;
            } else {
                System.out.println("File is present");
            }

        }
```

}