Project 1: LockedMe.com

Student: Abilio Junior

SimpliLearn Project LockedMe.com – Virtual Key Repositories

Contents

Sprint Planning

The project was divided in two sprints.

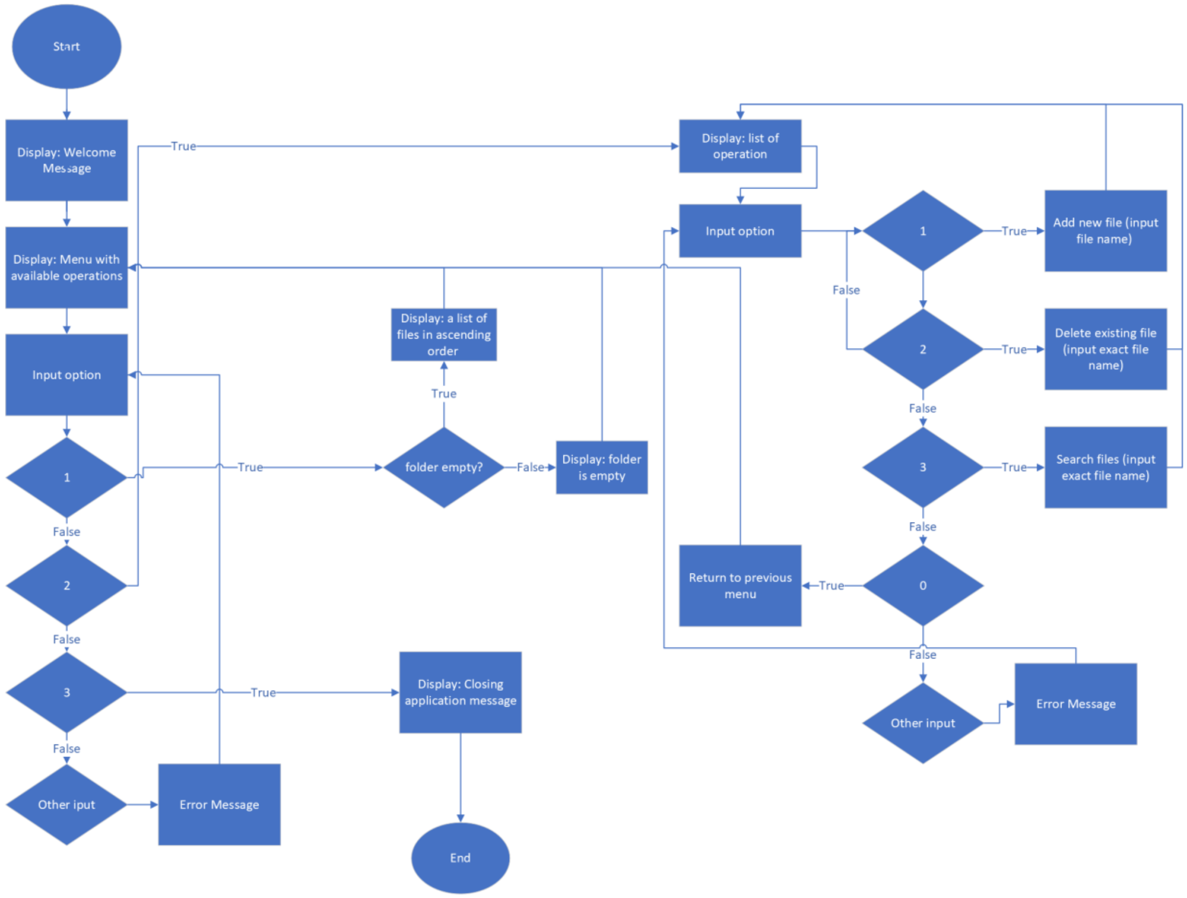
1- Understand the project design the flow of the application according to the requirements
2- Initialize Git and Connect GitHub, Develop the application and document the application.

Project 1: LockedMe.com

Java concepts used in the project

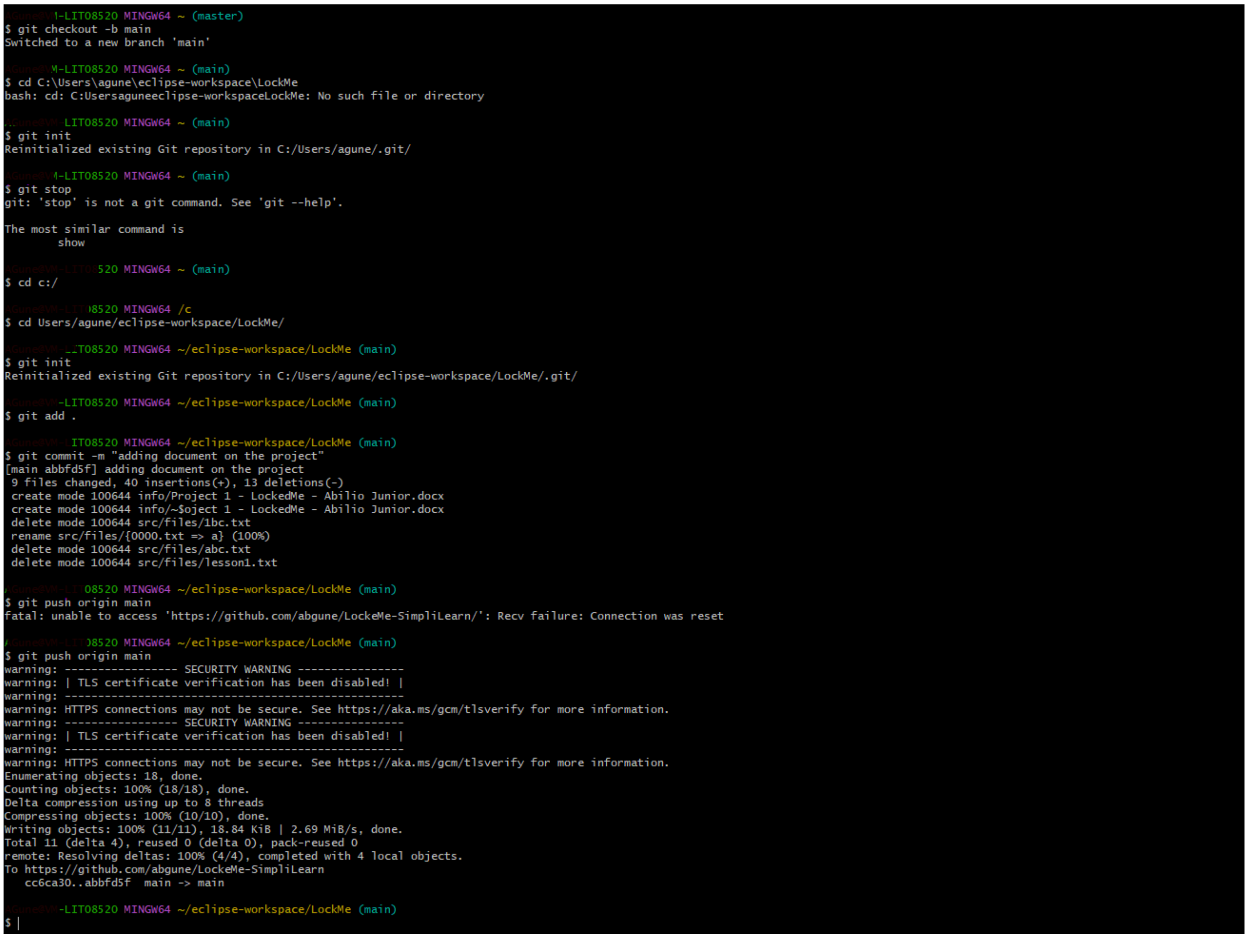
- Collections Framework (List)
- Exception Handling
- File Handling

Flow Diagram


Start
Display: Welcome Message
Display: Menu with available operations
Input option
1
False
2
False
3
False
Other iput
Error Message
True
folder empty?
True
Display: a list of files in ascending order
False
Display: folder is empty
True
Display: Closing application message
End
True
Display: list of operation
Input option
1
True
Add new file (input file name)
False
2
True
Delete existing file (input exact file name)
False
3
True
Search files (input exact file name)
False
0
True
Return to previous menu
False
Other input
Error Message

GIT Repository

GIT - https://github.com/abgune/LockeMe-SimpliLearn

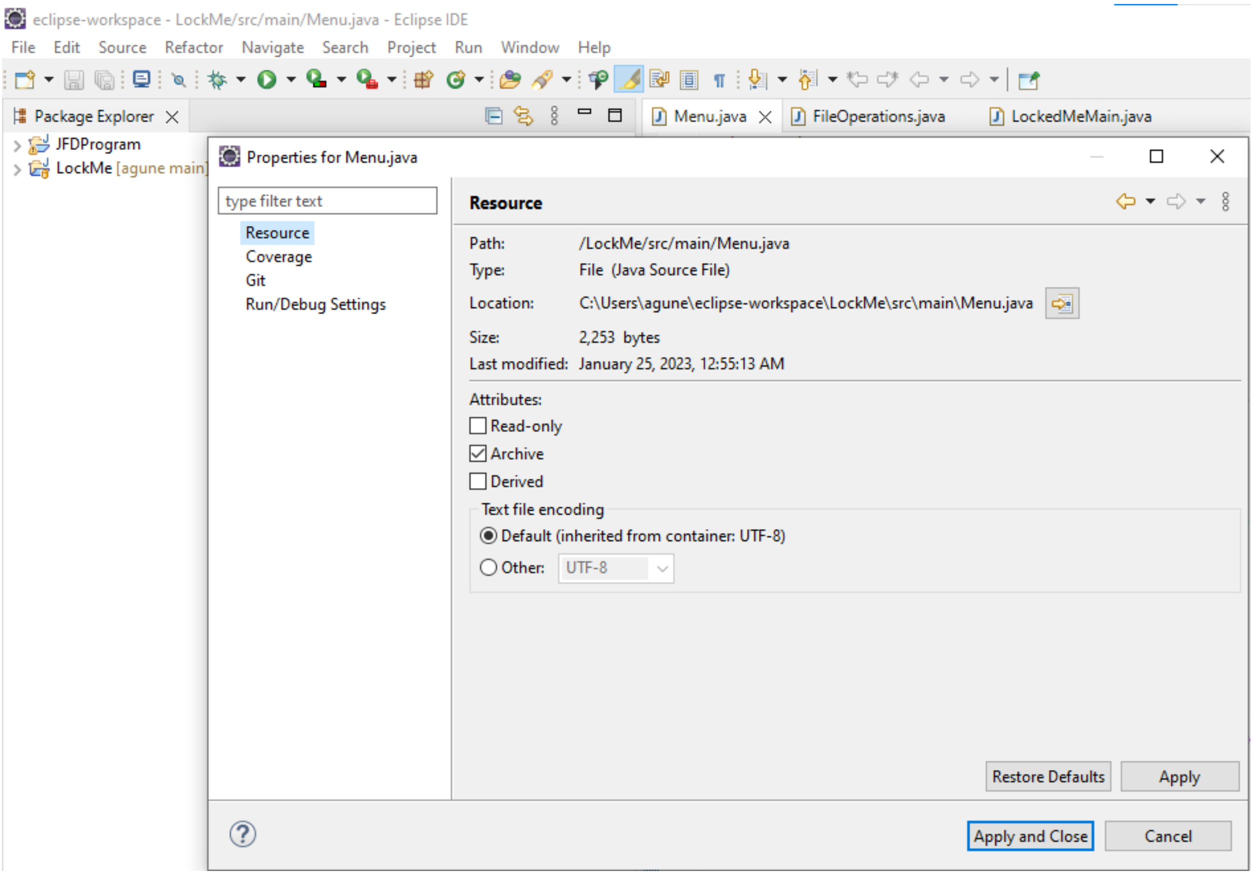


```
[redacted]1-LIT08520 MINGW64 ~ (master)
$ git checkout -b main
Switched to a new branch 'main'

[redacted]M-LIT08520 MINGW64 ~ (main)
$ cd C:\Users\agune\eclipse-workspace\LockMe
bash: cd: C:Usersaguneeclipse-workspaceLockMe: No such file or directory

[redacted]LIT08520 MINGW64 ~ (main)
$ git init
Reinitialized existing Git repository in C:/Users/agune/.git/

[redacted]1-LIT08520 MINGW64 ~ (main)
$ git stop
git: 'stop' is not a git command. See 'git --help'.

The most similar command is
        show

[redacted]520 MINGW64 ~ (main)
$ cd c:/

[redacted]08520 MINGW64 /c
$ cd Users/agune/eclipse-workspace/LockMe/

[redacted]IT08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ git init
Reinitialized existing Git repository in C:/Users/agune/eclipse-workspace/LockMe/.git/

[redacted]-LIT08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ git add .

[redacted]IT08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ git commit -m "adding document on the project"
[main abbfd5f] adding document on the project
 9 files changed, 40 insertions(+), 13 deletions(-)
 create mode 100644 info/Project 1 - LockedMe - Abilio Junior.docx
 create mode 100644 info/~$oject 1 - LockedMe - Abilio Junior.docx
 delete mode 100644 src/files/1bc.txt
 rename src/files/{0000.txt => a} (100%)
 delete mode 100644 src/files/abc.txt
 delete mode 100644 src/files/lesson1.txt

[redacted]08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ git push origin main
fatal: unable to access 'https://github.com/abgune/LockeMe-SimpliLearn/': Recv failure: Connection was reset

[redacted]08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ git push origin main
warning: ----------------- SECURITY WARNING ----------------
warning: | TLS certificate verification has been disabled! |
warning: ---------------------------------------------------
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
warning: ----------------- SECURITY WARNING ----------------
warning: | TLS certificate verification has been disabled! |
warning: ---------------------------------------------------
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 18.84 KiB | 2.69 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/abgune/LockeMe-SimpliLearn
   cc6ca30..abbfd5f  main -> main

[redacted]-LIT08520 MINGW64 ~/eclipse-workspace/LockMe (main)
$ |
```

Application Development Process:

Application was developed using eclipse and JDK version 11

Project 1: LockedMe.com



Resource

Path: /LockMe/src/main/Menu.java
Type: File (Java Source File)
Location: C:\Users\agune\eclipse-workspace\LockMe\src\main\Menu.java
Size: 2,253 bytes
Last modified: January 25, 2023, 12:55:13 AM

Attributes:
□ Read-only
☑ Archive
□ Derived

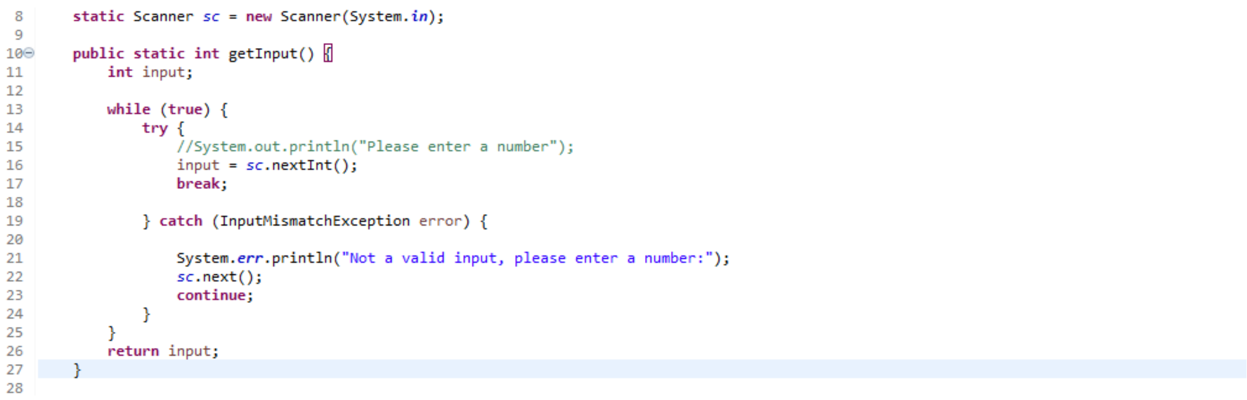
Text file encoding
● Default (inherited from container: UTF-8)
○ Other: UTF-8

The project was implemented in 3 classes:

Class: Menu

Method 1: getInput()

Responsible to receive a correct integer input to be used on the menus switch.

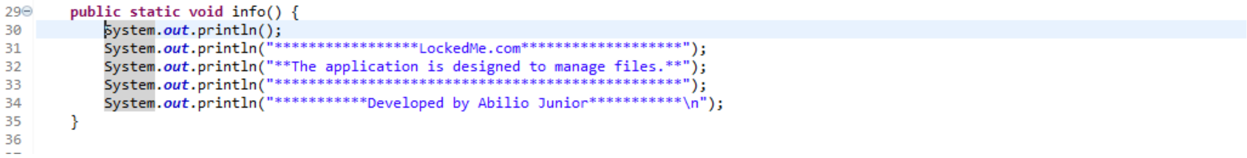


```java
    static Scanner sc = new Scanner(System.in);

    public static int getInput() {
        int input;

        while (true) {
            try {
                //System.out.println("Please enter a number");
                input = sc.nextInt();
                break;

            } catch (InputMismatchException error) {

                System.err.println("Not a valid input, please enter a number:");
                sc.next();
                continue;
            }
        }
        return input;
    }

```

Project 1: LockedMe.com

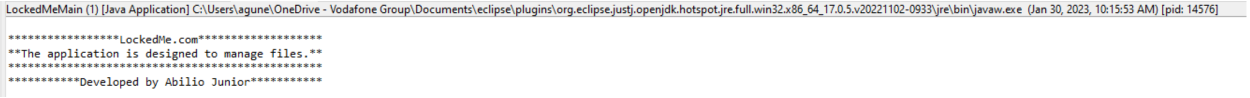
Method 2: info()

Displays the information of the application and the developer name



```java
public static void info() {
    System.out.println();
    System.out.println("*****************LockedMe.com*******************");
    System.out.println("**The application is designed to manage files.**");
    System.out.println("************************************************");
    System.out.println("***********Developed by Abilio Junior***********\n");
}
```

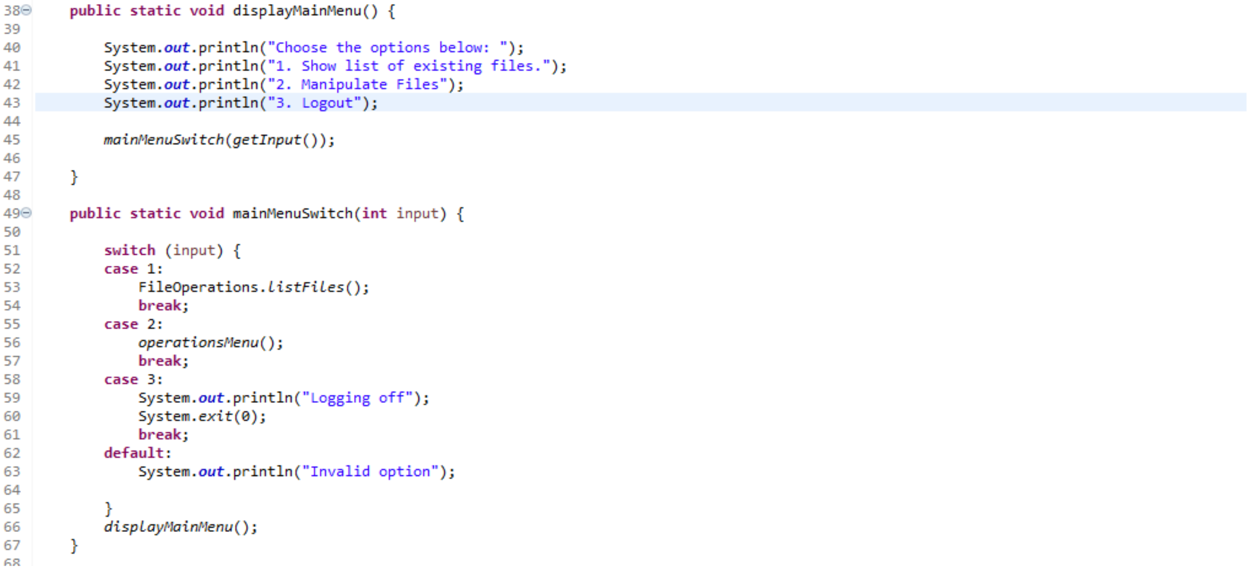
Output:



```
LockedMeMain (1) [Java Application] C:\Users\agune\OneDrive - Vodafone Group\Documents\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe  (Jan 30, 2023, 10:15:53 AM) [pid: 14576]

*****************LockedMe.com*******************
**The application is designed to manage files.**
************************************************
***********Developed by Abilio Junior***********
```

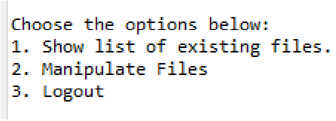
Method 3, 4: displayMainMenu(), mainMenuSwitch(int input)

Displays the information on the initial menu, and the method 3 is the switch which helps navigate on the menu options



```java
public static void displayMainMenu() {

    System.out.println("Choose the options below: ");
    System.out.println("1. Show list of existing files.");
    System.out.println("2. Manipulate Files");
    System.out.println("3. Logout");

    mainMenuSwitch(getInput());

}

public static void mainMenuSwitch(int input) {

    switch (input) {
    case 1:
        FileOperations.listFiles();
        break;
    case 2:
        operationsMenu();
        break;
    case 3:
        System.out.println("Logging off");
        System.exit(0);
        break;
    default:
        System.out.println("Invalid option");

    }
    displayMainMenu();
}
```

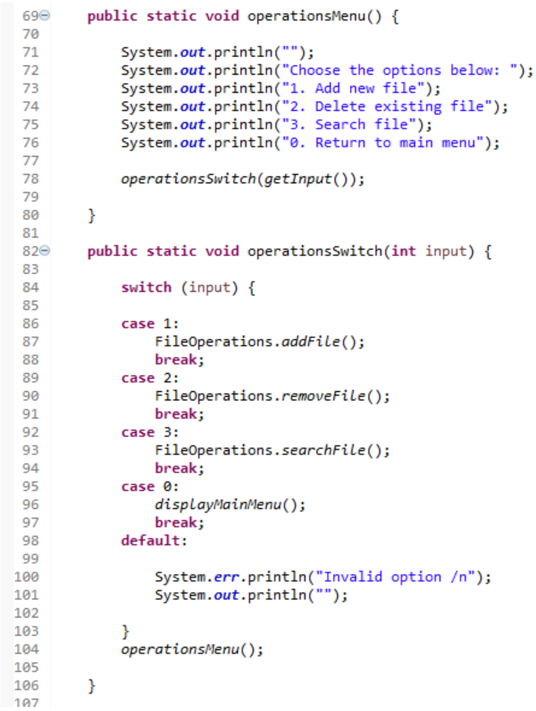
Output:



```
Choose the options below:
1. Show list of existing files.
2. Manipulate Files
3. Logout
```

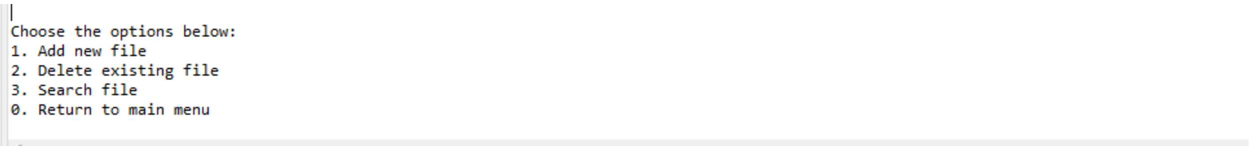
Method 5, 6: operationsMenu(), operationsSwitch(int input)

Displays the second menu for operations with the fille, and the method 6 is the switch which helps navigate on the menu options



```java
    public static void operationsMenu() {

        System.out.println("");
        System.out.println("Choose the options below: ");
        System.out.println("1. Add new file");
        System.out.println("2. Delete existing file");
        System.out.println("3. Search file");
        System.out.println("0. Return to main menu");

        operationsSwitch(getInput());

    }

    public static void operationsSwitch(int input) {

        switch (input) {

        case 1:
            FileOperations.addFile();
            break;
        case 2:
            FileOperations.removeFile();
            break;
        case 3:
            FileOperations.searchFile();
            break;
        case 0:
            displayMainMenu();
            break;
        default:

            System.err.println("Invalid option /n");
            System.out.println("");

        }
        operationsMenu();

    }
```

Output:



```

Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
```

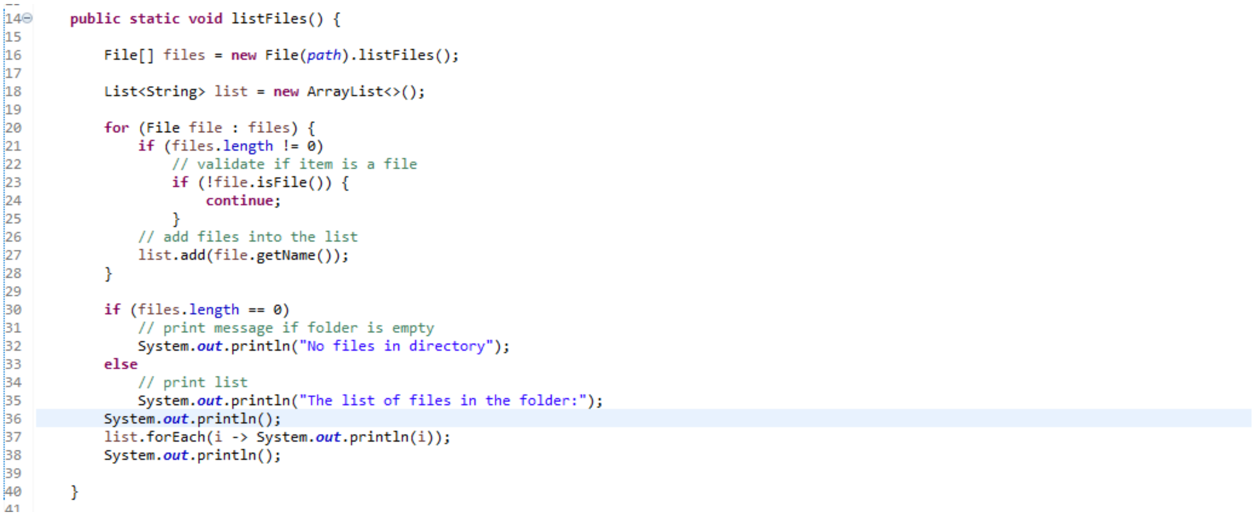
Class: FileOperations

Defined a static path that the files will be searched, created and deleted from.

```java
static String path = "src/files/";
```

Project 1: LockedMe.com

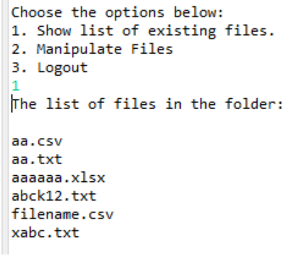
Method 1: listFiles()



```java
    public static void listFiles() {

        File[] files = new File(path).listFiles();

        List<String> list = new ArrayList<>();

        for (File file : files) {
            if (files.length != 0)
                // validate if item is a file
                if (!file.isFile()) {
                    continue;
                }
            // add files into the list
            list.add(file.getName());
        }

        if (files.length == 0)
            // print message if folder is empty
            System.out.println("No files in directory");
        else
            // print list
            System.out.println("The list of files in the folder:");
        System.out.println();
        list.forEach(i -> System.out.println(i));
        System.out.println();

    }
```

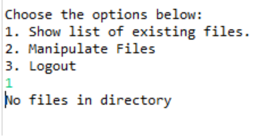
Output:

When not empty:



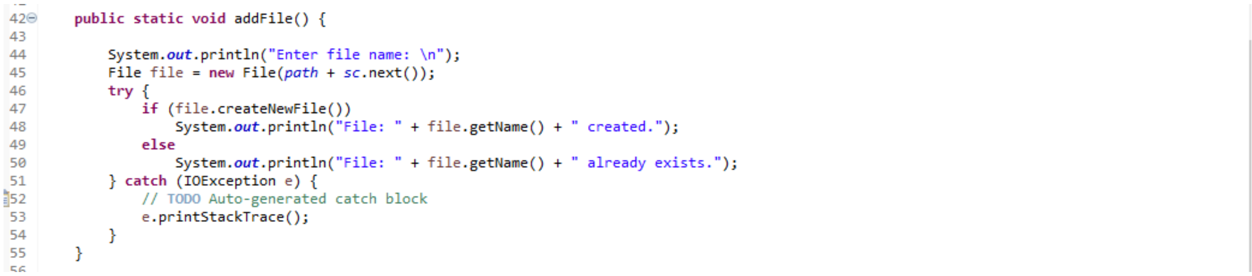
```
Choose the options below:
1. Show list of existing files.
2. Manipulate Files
3. Logout
1
The list of files in the folder:

aa.csv
aa.txt
aaaaaa.xlsx
abck12.txt
filename.csv
xabc.txt
```

When empty:



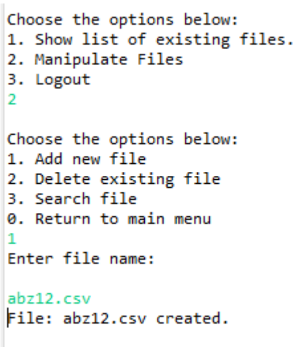
```
Choose the options below:
1. Show list of existing files.
2. Manipulate Files
3. Logout
1
No files in directory
```

Method 2: addFile()



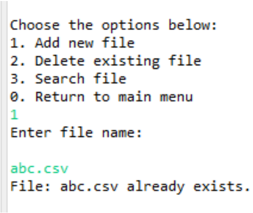
```java
    public static void addFile() {

        System.out.println("Enter file name: \n");
        File file = new File(path + sc.next());
        try {
            if (file.createNewFile())
                System.out.println("File: " + file.getName() + " created.");
            else
                System.out.println("File: " + file.getName() + " already exists.");
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

Project 1: LockedMe.com



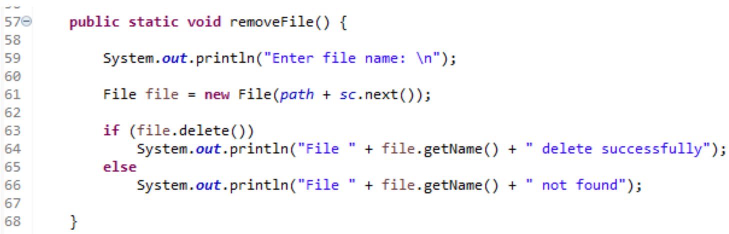
```
Choose the options below:
1. Show list of existing files.
2. Manipulate Files
3. Logout
2

Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
1
Enter file name:

abz12.csv
File: abz12.csv created.
```

When filename already exists on the folder:



```
Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
1
Enter file name:

abc.csv
File: abc.csv already exists.
```

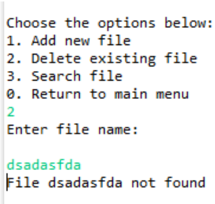
Method 3: removeFile()



```java
    public static void removeFile() {

        System.out.println("Enter file name: \n");

        File file = new File(path + sc.next());

        if (file.delete())
            System.out.println("File " + file.getName() + " delete successfully");
        else
            System.out.println("File " + file.getName() + " not found");

    }
```

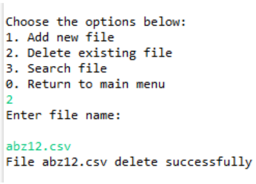
Output:

When name not matched



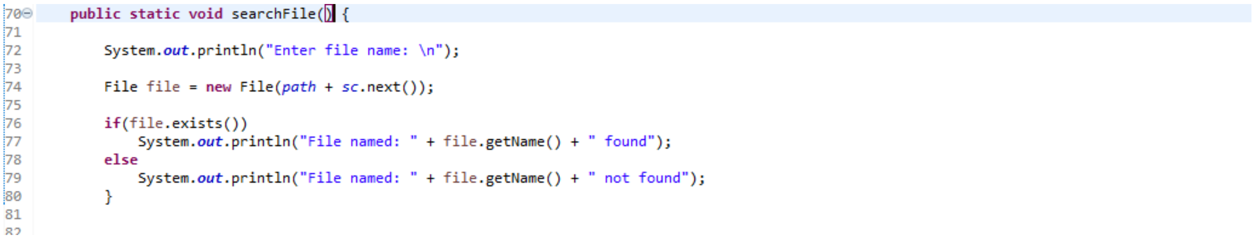
```
Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
2
Enter file name:

dsadasfda
File dsadasfda not found
```

When name is matched



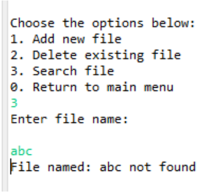
```
Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
2
Enter file name:

abz12.csv
File abz12.csv delete successfully
```

Method 4: searchFile()



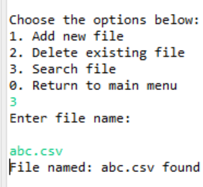
```
    public static void searchFile() {

        System.out.println("Enter file name: \n");

        File file = new File(path + sc.next());

        if(file.exists())
            System.out.println("File named: " + file.getName() + " found");
        else
            System.out.println("File named: " + file.getName() + " not found");
        }
```

When file name is not matched



```
Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
3
Enter file name:

abc
File named: abc not found
```

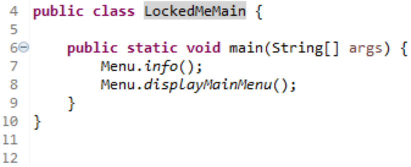
When file name is matched



```
Choose the options below:
1. Add new file
2. Delete existing file
3. Search file
0. Return to main menu
3
Enter file name:

abc.csv
File named: abc.csv found
```

Class: LockedMeMain

Main class to execute the aplication



```
public class LockedMeMain {

    public static void main(String[] args) {
        Menu.info();
        Menu.displayMainMenu();
    }
}
```

Conclusion

The application is designed to allow customer to add, search and list all files in a predefined folder to not confuse him with how you name the path.

Files are listed in ascending order when displayed.

It can add any type of file without restrictions, it doesn't allow to create an existing file as it might delete existing file content.

Doesn't crash if the user inputs incorrect charters while navigating in the menu and it only closes the application if the user select the option to do it.

Enhancements:

Consider adding am option to the customer to input the folder in which he would like to do the operations of listing files, add, remove, or search.

Create user management class to give users access to which type of permissions they have on the folder or files.

Ability to list all files that the name match or are contained in the input of the user input.