SOURCE CODE

- startPoint.java

```java
import WelcomeScreen.WelcomeScreen;

no usages
public class startPoint {
    no usages
    public static void main(String[] args)  {
        WelcomeScreen welcome = new WelcomeScreen();
        welcome.introMenuPage();
        welcome.userInput();
    }
}
```

- WelcomeScreen.java

```java
package WelcomeScreen;

import FileServices.*;
import FileSystem.*;
import java.util.*;

3 usages
public class WelcomeScreen implements Screen {

    1 usage
    final String productName = "*!*!*!*!*!*!*!*!*!* PRESENTING YOU APPLICATION LOCKEDME.COM BY LOCKERS PVT.LTD. *!*!*!*!*!*!*!*!*!*";
    1 usage
    final String developerName = "Developed By -> AKSHAT SANKHYAN";

    4 usages
    private final ArrayList<String> optionList = new ArrayList<>();


    1 usage
    public WelcomeScreen() {
        optionList.add("-> Press 1 To Show Existing Files");
        optionList.add("-> Press 2 To Show File Options Menu");
        optionList.add("-> Press 3 To Quit The Application");

    }
    1 usage
    public void introMenuPage() {
        System.out.println();
        System.out.println(productName);
        System.out.println("\t\t\t\t\t\t\t\t\t" + developerName);
        System.out.println("\t\t\t==========================================================================");
        System.out.println();
        System.out.println("Root Directory : " + DirService.getFileDir().name);
    }
    2 usages
    @Override
    public void menuOptions() {
        System.out.println();
        System.out.println("Main Menu");
        for (String str : optionList)  {
            System.out.println(str);
        }
    }
```

```
    2 usages
    @Override
    public void NavigateOption(int choice) {
        switch(choice) {
            case 1:
                DirService.showFiles();
                break;
            case 2:
                MenuService.setMenu();
                break;
            case 3:
                System.out.println("\n Thank You For Using `LOCKEDME.COM`.Hope You Liked Our Application. ");
                break;
            default:
                System.out.println("Please Enter Valid Option ");
                break;
        }
    }
    2 usages
    @Override
    public void userInput() {
        int val = 0;
        char o;
        while(val != 3) {
            this.menuOptions();
            System.out.print("Enter Your Choice : ");
            o = sc.next().charAt(0);
            val = (int)o - 48;
            this.NavigateOption(val);
        }
    }
}
```

- FileOperations.java

```
package FileUI;

import java.io.*;
import java.nio.file.Path;
import java.util.*;
import FileServices.*;

2 usages
public class FileOperations {

    6 usages
    Scanner sc = new Scanner(System.in);
    9 usages
    String filename;
    3 usages
    char ch;

    1 usage
    public void AddFile()   {
```

```java
1 usage
public void AddFile()   {
    System.out.print("Enter the File Name you want to create along with the extension : ");
    filename = sc.nextLine();

    File file = new File( pathname: DirService.Path() + "/" + filename);
    try {
        boolean flag = file.createNewFile();
        if(flag) {
            System.out.println("File with name " + filename + " is created!!!");
            System.out.println("Do you want to write something in this File ? Press y for Yes and n for No.");
            ch = sc.next().charAt(0);
            if(ch == 'Y' || ch == 'y') {
                System.out.print("Enter the content you want to write in file : ");
                sc.nextLine();
                String fileData = sc.nextLine();
                try {
                    FileWriter writeData = new FileWriter( fileName: DirService.Path() + "/" + filename);
                    writeData.write(fileData);
                    System.out.println("Data is successfully written in the file.");
                    writeData.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
        else System.out.println("Not Able to create " + filename + ". Try Again with another name");
    } catch(Exception e)    {
        e.printStackTrace();
    }
}
1 usage
public void DeleteFile()    {
    System.out.print("Enter the File Name you want to delete along with the extension : ");
    filename = sc.nextLine();

    File file = new File( pathname: DirService.Path() + "/" + filename);
    try {
        boolean flag = file.delete();
        if(flag) System.out.println("File with name " + filename + " is deleted!!!");
        else System.out.println("Not Able to delete " + filename + ". Try Again");

    } catch(Exception e)    {
        e.printStackTrace();
    }
}
1 usage
```

```
    1 usage
    public void SearchFile() {
        boolean isFound = false;

        System.out.print("Enter the File Name you want to search along with the extension : ");
        String fileName = sc.nextLine();
        System.out.println("You are searching for a file named: " + fileName);

        Path path = DirService.getFileDir().path;
        File prevFiles = path.toFile();

        File[] directoryFiles = prevFiles.listFiles();

        if (directoryFiles != null) {
            for (File directoryFile : directoryFiles)
                if (directoryFile.getName().equals(fileName)) {
                    System.out.println("File named " + fileName + " Found");
                    isFound = true;
                }
        }
        if (!isFound)
            System.out.println("!!!FILE NOT FOUND!!!");
    }
}
```

- FileOptions.java

```
package FileUI;

import FileSystem.*;
import java.util.*;

2 usages
public class FileOptions implements Screen {

    5 usages
    private final ArrayList<String> options = new ArrayList<>();
    3 usages
    FileOperations file = new FileOperations();

    1 usage
    public FileOptions() {
        options.add("-> Press 1 To Add a File");
        options.add("-> Press 2 To Delete a File");
        options.add("-> Press 3 To Search a File");
        options.add("-> Press 4 To Return To Menu");

    }
    2 usages
    @Override
    public void menuOptions() {
        System.out.println();
        System.out.println("!!!File Options Menu!!!");
        for (String str : options) {
            System.out.println(str);
        }
    }
```

```
    2 usages
    @Override
    public void NavigateOption(int option) {
        switch (option) {
            case 1:
                file.AddFile();
                break;
            case 2:
                file.DeleteFile();
                break;
            case 3:
                file.SearchFile();
                break;
            default:
                System.out.println("Returning To Main Menu.......");
                break;
        }
    }
    2 usages
    @Override
    public void userInput() {
        char ch;
        int val = 0;
        while (val != 4) {
            this.menuOptions();
            System.out.print("Enter Your Choice : ");
            ch = sc.next().charAt(0);
            val = (int)ch - 48;
            this.NavigateOption(val);
        }
    }
}
```

- Screen.java

```
package FileSystem;

import java.util.Scanner;

2 usages  2 implementations
public interface Screen {

    2 usages
    Scanner sc = new Scanner(System.in);
    2 usages  2 implementations
    void menuOptions();
    2 usages  2 implementations
    void NavigateOption(int choice);
    2 usages  2 implementations
    void userInput();
}
```

- Folder.java

```java
package FileSystem;

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

3 usages
public class Folder {
    2 usages
    public String name = "FileStorage/";
    4 usages
    private final ArrayList<File> files = new ArrayList<>();

    3 usages
    public Path path = FileSystems.getDefault().getPath(name).toAbsolutePath();
    1 usage
    File prevFiles = path.toFile();

    2 usages
    public void insertFiles() {
        File[] presentFiles = prevFiles.listFiles();
        files.clear();
        if (presentFiles != null) {
            for (File presentFile : presentFiles)
                if (presentFile.isFile())
                    files.add(presentFile);
        }

        Collections.sort(files);
    }
    1 usage
    public ArrayList<File> getListOfFiles() {
        insertFiles();
        return files;
    }
}
```

- MenuService.java

```java
package FileServices;

import FileUI.*;
1 usage
public class MenuService {
    1 usage
    public static void setMenu()   {
        FileOptions file = new FileOptions();
        file.userInput();
    }
}
```

- DirService.java

```java
package FileServices;

import FileSystem.*;
import java.io.*;

7 usages
public class DirService {
    3 usages
    private static final Folder fileDir = new Folder();
    1 usage
    public static void showFiles() {
        fileDir.insertFiles();
        int flag = 0;
        for (File file : DirService.getFileDir().getListOfFiles()) {
            if(flag == 0){
                flag = 1;
                System.out.println("The Existing Files Are Listed Below : ");
            }
            System.out.println(file.getName());
        }
        if(flag == 0){
            System.out.println("Sorry No Existing Files Found.");
        }
    }
    3 usages
    public static Folder getFileDir() {
        return fileDir;
    }
    3 usages
    public static String Path() {
        return fileDir.path.toString();
    }
}

```