DOCUMENTATION OF PRODUCT

Company Name => **Lockers Pvt. Ltd.**

Product Name => **LockedMe.com**

Developer's Name : **AKSHAT SANKHYAN**

GitHub Link : https://github.com/akshatsankhyan/JavaFSDPhase1Project.git

About Product

The Product "LockedMe.com" is a CLI(Command Line Interface) based application. This application can be used for various file operations such as creating new files and can write data into it, delete and search the existing files and keep track of all the files present in the directory in a sorted manner.

Language & IDE Used

Java8 & IntelliJ.

Sprint Planning

The Sprints are planned in two halves illustrated below:

Sprint 1:

- Understanding the requirements of the Product/Application.
- Designing the Flow Chart Of The Application.
- Implementing Main Menu Functionality.
- Implementing the functionality to close the application on User's Input.

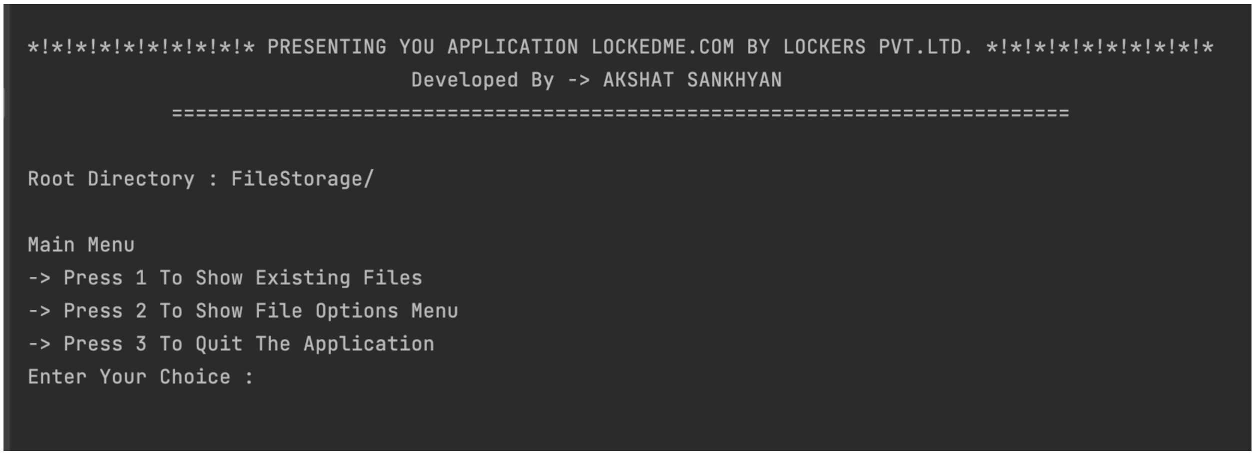
Sprint 2:

- Implementing File Menu Functionality.
- Testing File operations such as adding, deleting, searching file in the Directory.
- Optimising File operations for better results.
- Implementing the functionality to return to the Main Menu from File Options Menu.
- Creating Documentation and ReadMe file for the product.

Working Of Application

Below is the illustration of features of the application along with snapshots.

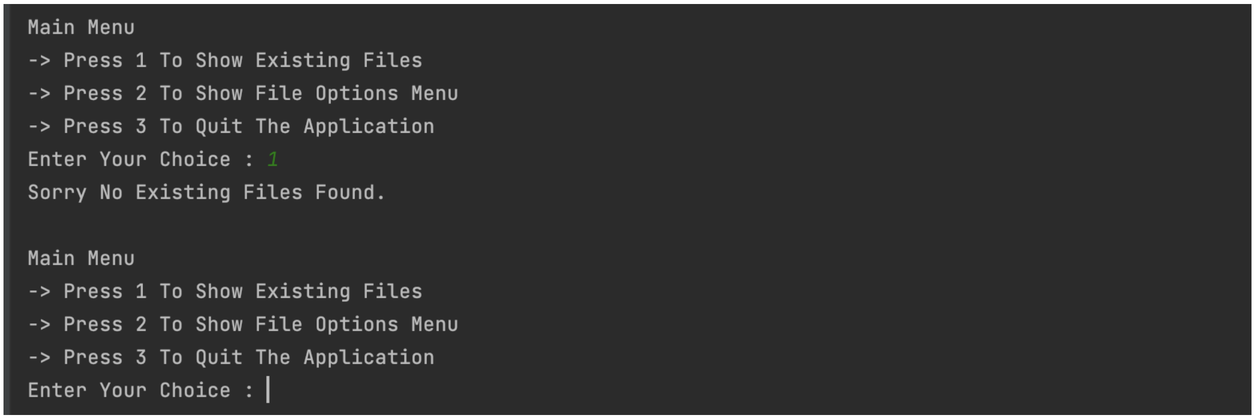
WELCOME SCREEN



```
*!*!*!*!*!*!*!*!*!* PRESENTING YOU APPLICATION LOCKEDME.COM BY LOCKERS PVT.LTD. *!*!*!*!*!*!*!*!*!*
                                Developed By -> AKSHAT SANKHYAN
            ===========================================================================

Root Directory : FileStorage/

Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice :
```

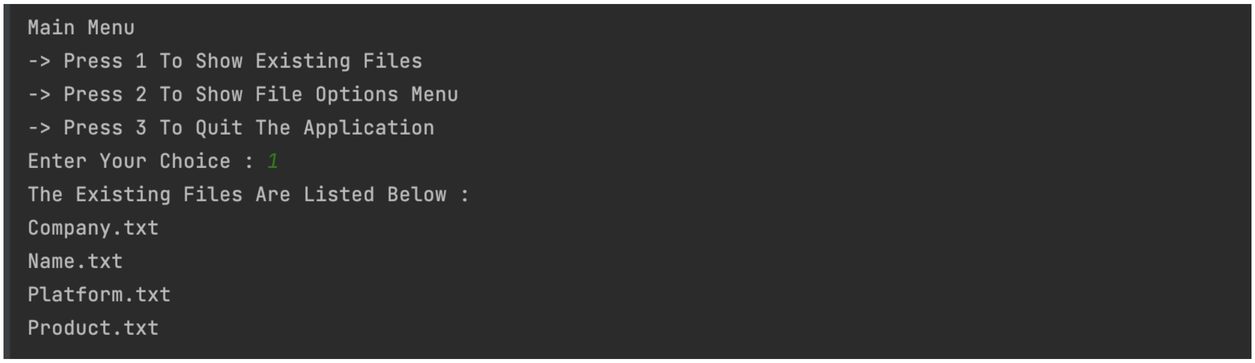
LIST OF FILES

- When there are no existing files in the directory.



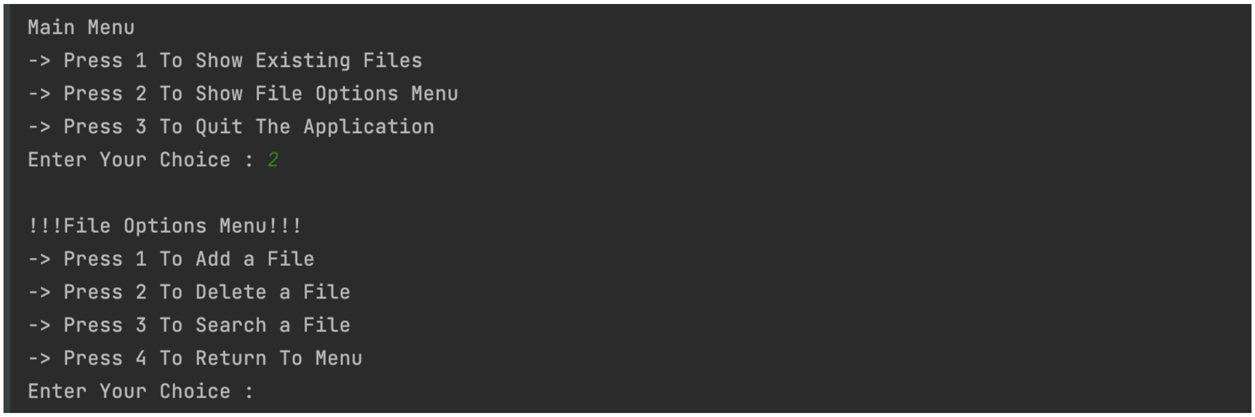
```
Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice : 1
Sorry No Existing Files Found.

Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice :
```

- When there are existing files in the directory.



```
Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice : 1
The Existing Files Are Listed Below :
Company.txt
Name.txt
Platform.txt
Product.txt
```

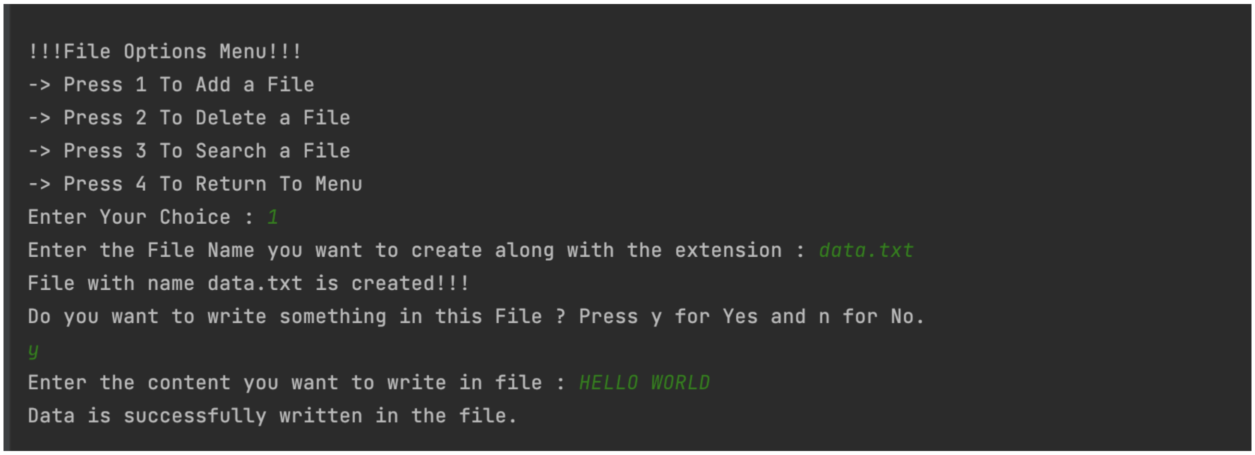
FILE OPTIONS MENU



```
Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice : 2

!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice :
```

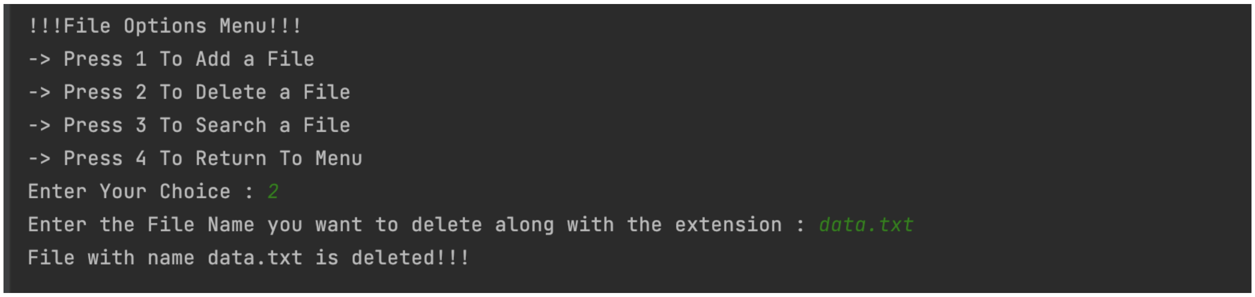
FILE OPERATIONS

- ### ADDING A FILE



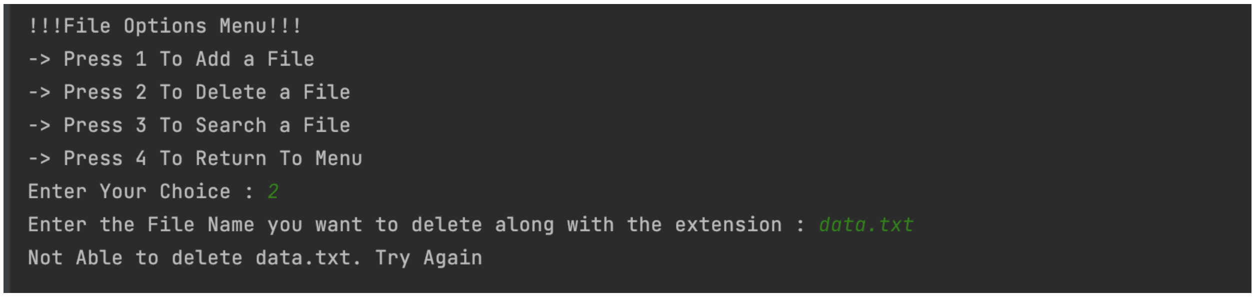
```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 1
Enter the File Name you want to create along with the extension : data.txt
File with name data.txt is created!!!
Do you want to write something in this File ? Press y for Yes and n for No.
y
Enter the content you want to write in file : HELLO WORLD
Data is successfully written in the file.
```

- ### DELETING FILE WHEN THAT FILE IS PRESENT



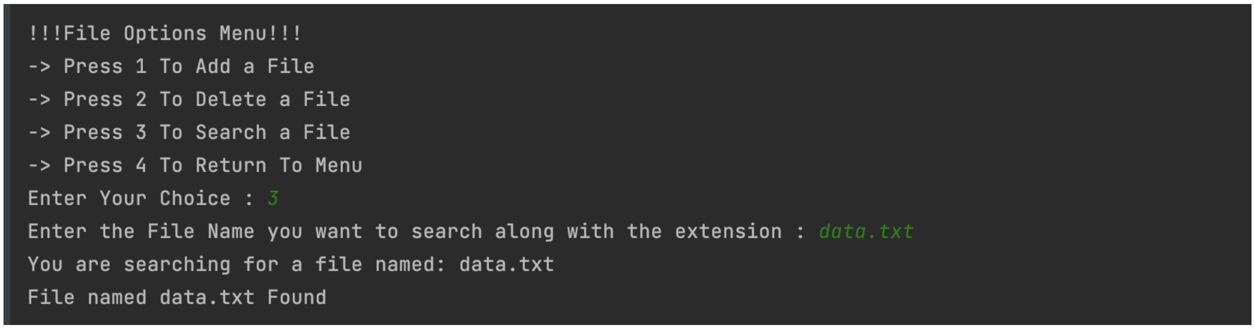
```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 2
Enter the File Name you want to delete along with the extension : data.txt
File with name data.txt is deleted!!!
```

- **DELETING FILE WHEN THAT FILE IS NOT PRESENT**



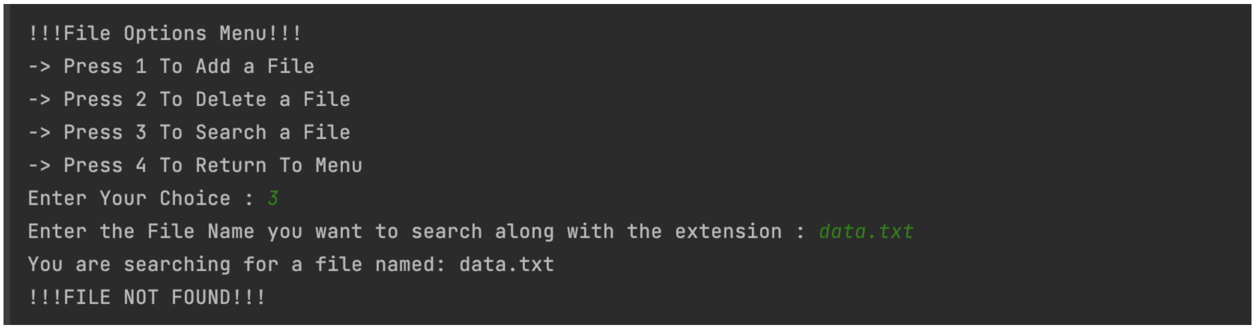
```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 2
Enter the File Name you want to delete along with the extension : data.txt
Not Able to delete data.txt. Try Again
```

- **SEARCHING FILE WHEN THAT FILE IS PRESENT**



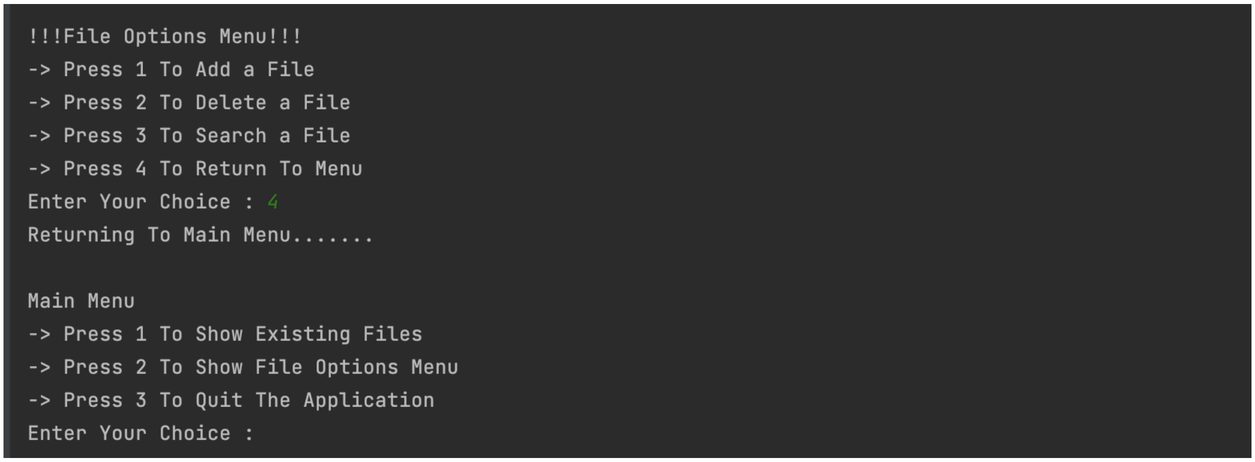
```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 3
Enter the File Name you want to search along with the extension : data.txt
You are searching for a file named: data.txt
File named data.txt Found
```

- **SEARCHING FILE WHEN THAT FILE IS NOT PRESENT**



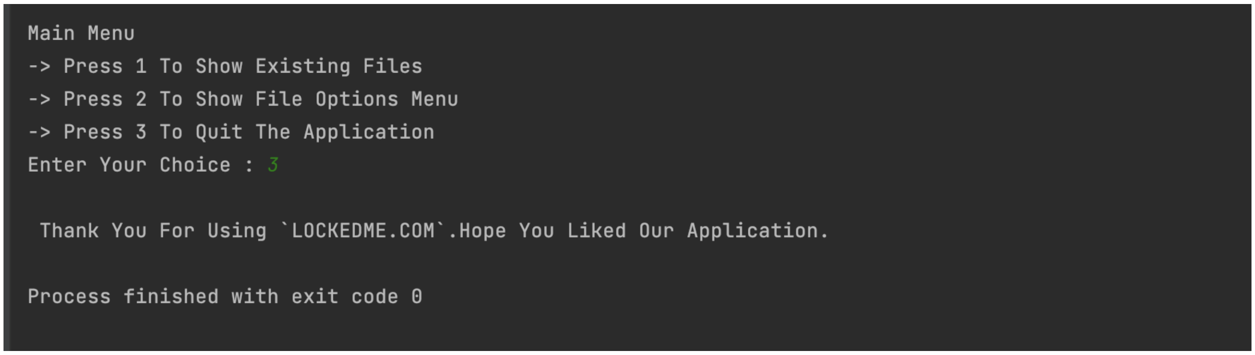
```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 3
Enter the File Name you want to search along with the extension : data.txt
You are searching for a file named: data.txt
!!!FILE NOT FOUND!!!
```

- **RETURNING TO MAIN MENU**



```
!!!File Options Menu!!!
-> Press 1 To Add a File
-> Press 2 To Delete a File
-> Press 3 To Search a File
-> Press 4 To Return To Menu
Enter Your Choice : 4
Returning To Main Menu.......

Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice :
```

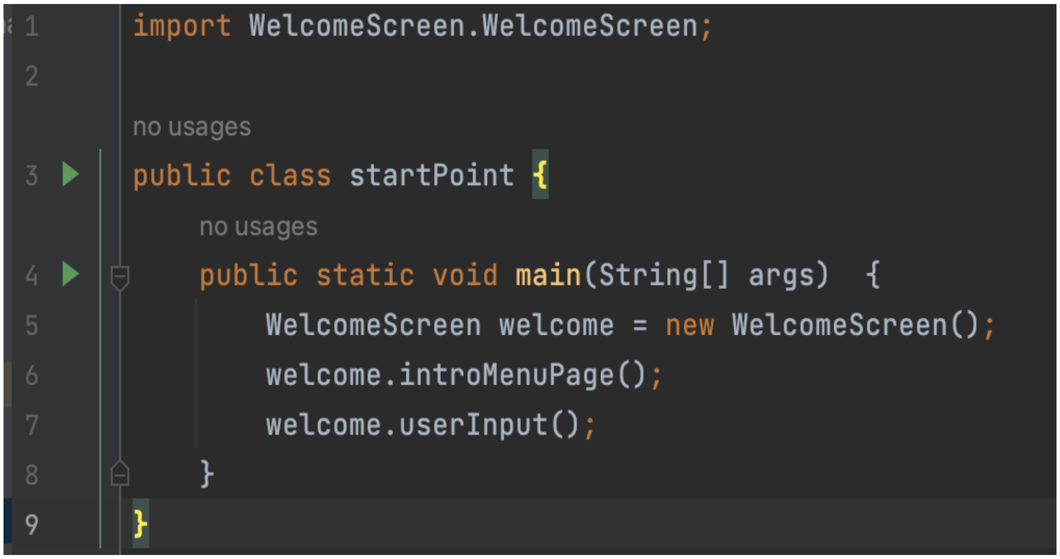
QUITTING THE APPLICATION



```
Main Menu
-> Press 1 To Show Existing Files
-> Press 2 To Show File Options Menu
-> Press 3 To Quit The Application
Enter Your Choice : 3

 Thank You For Using `LOCKEDME.COM`.Hope You Liked Our Application.

Process finished with exit code 0
```

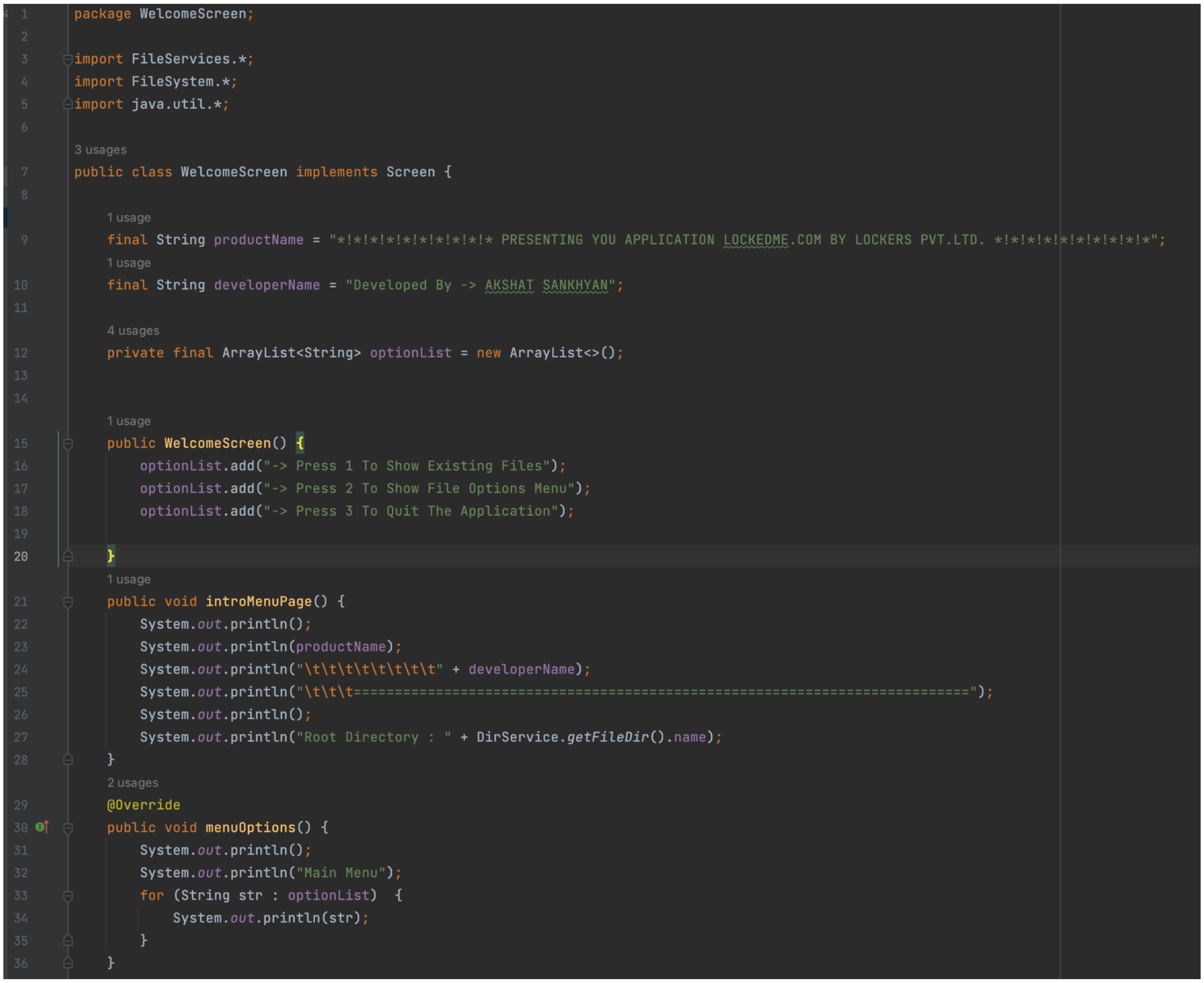
SOURCE CODE SNAPSHOTS

- **startPoint.java**

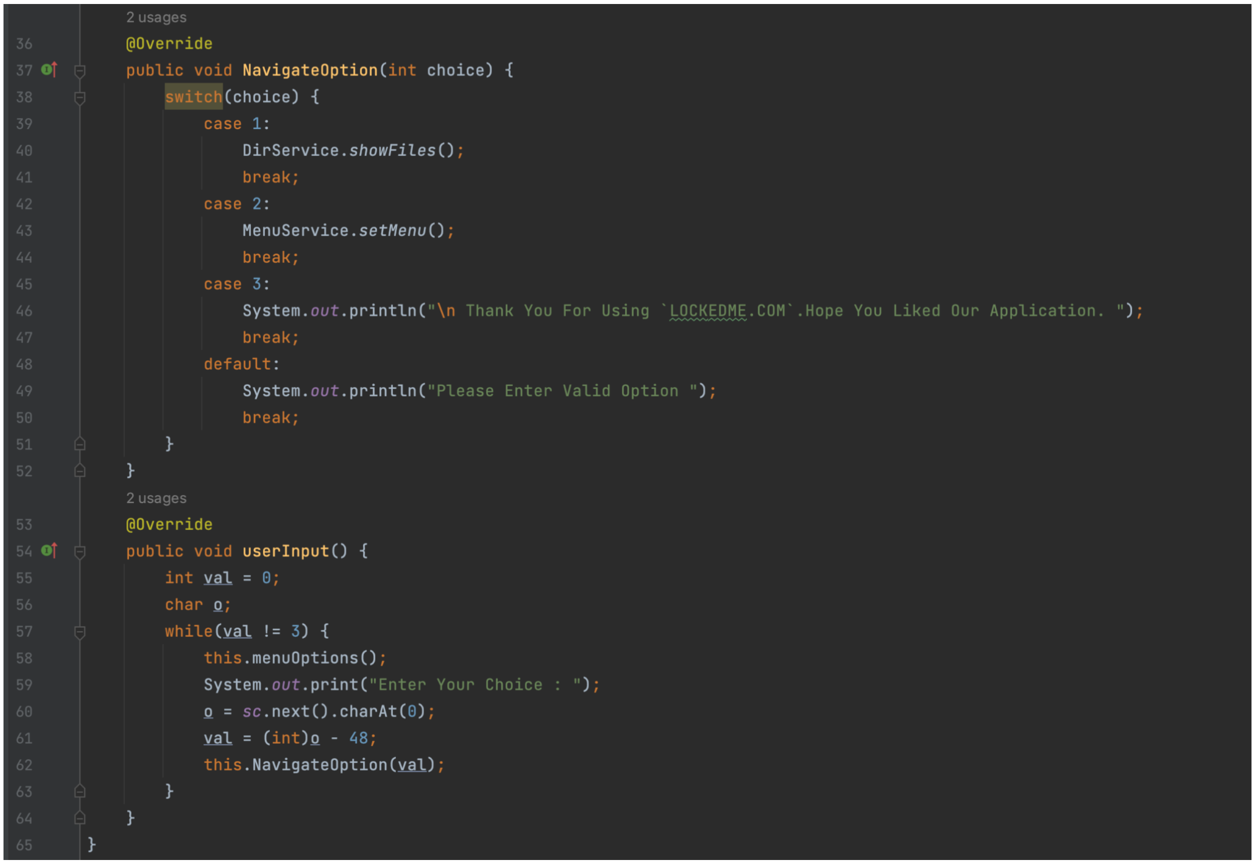


```
import WelcomeScreen.WelcomeScreen;

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

- **WelcomeScreen.java**

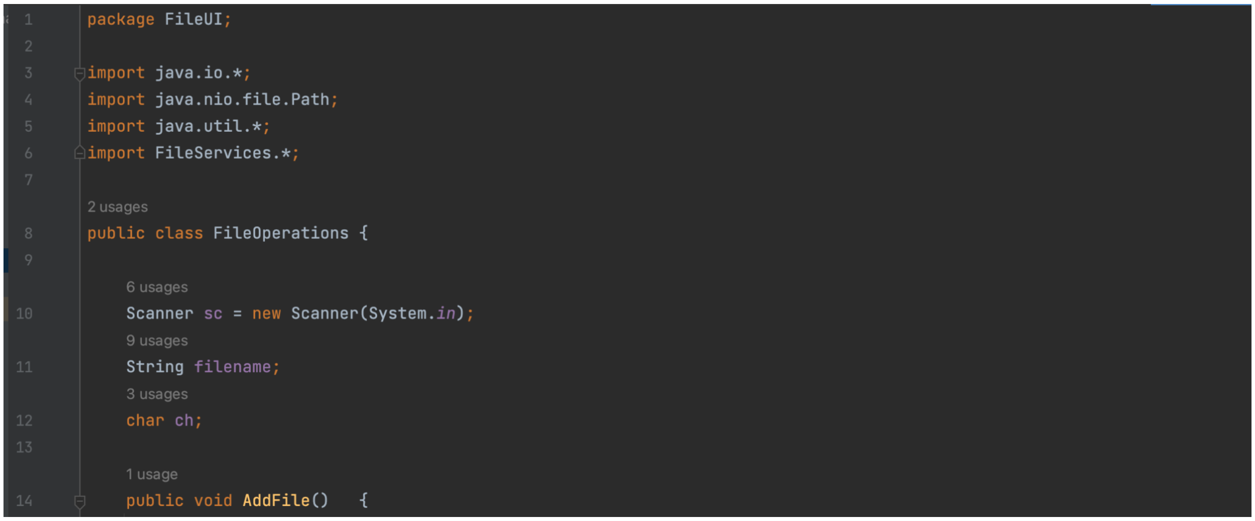


```java
package WelcomeScreen;

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" + 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);
        }
    }
```

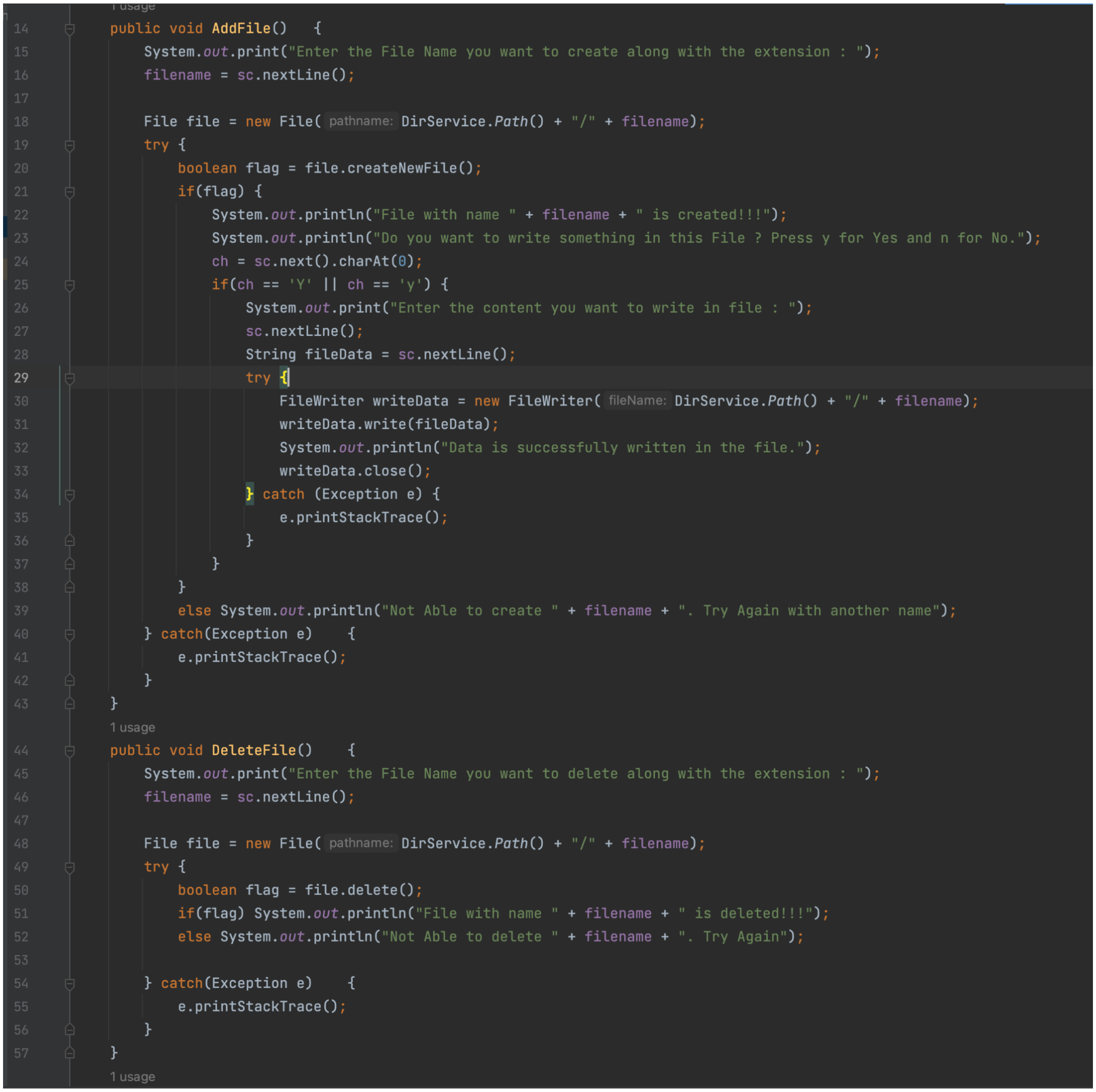


```
    2 usages
    @Override
    public void NavigateOption(int choice) {
        switch(choice) {
            case 1:
                DirService.showFiles();
                break;
            case 2:
                MenuService.setMenu();
                break;
            case 3:
                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);
        }
    }
}
```

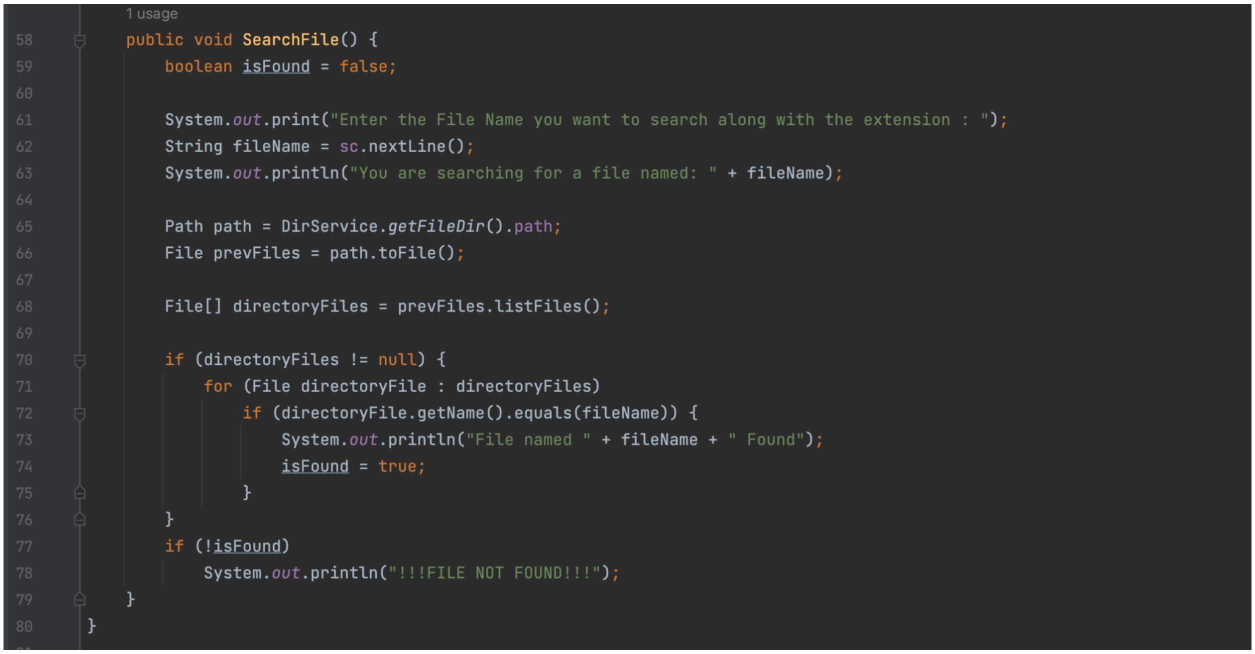
- **FileOperations.java**



```
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()   {
```

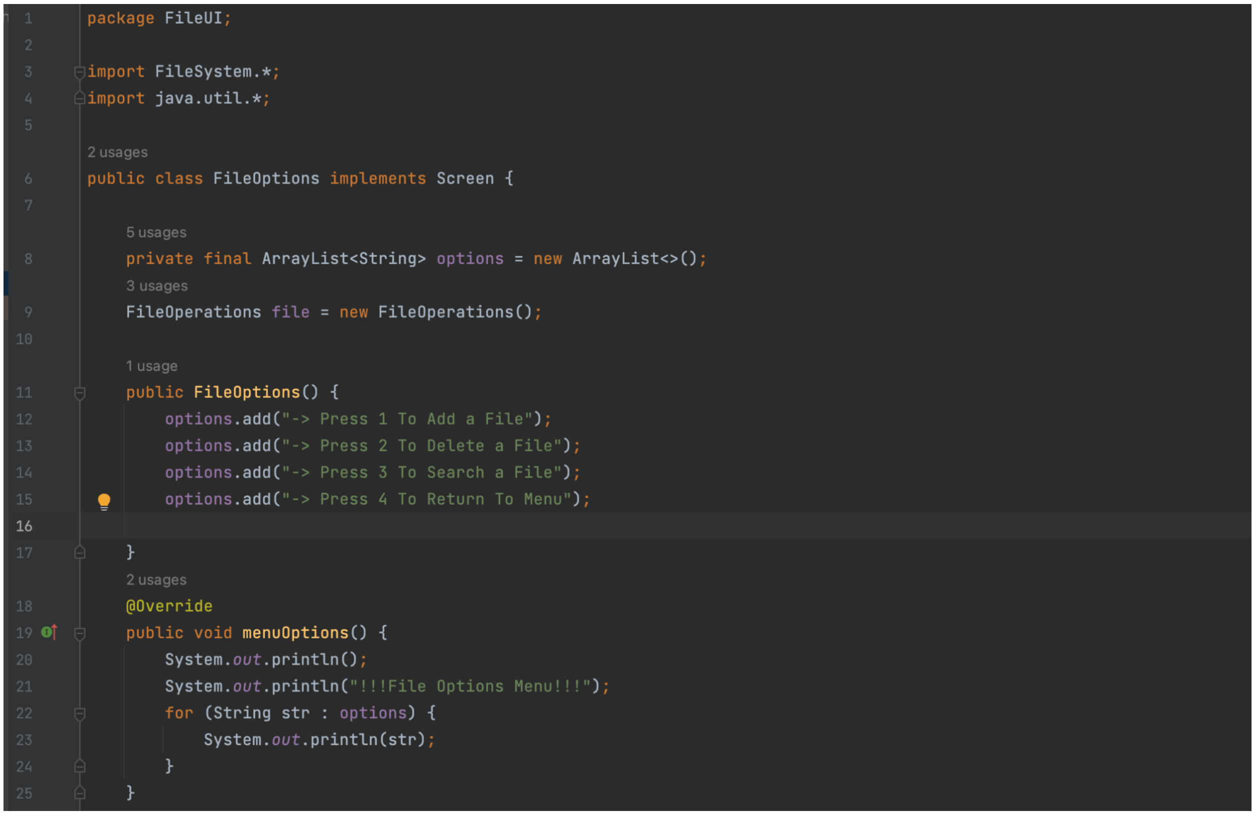


```
    1 usage
    public void AddFile()   {
        System.out.print("Enter the File Name you want to create along with the extension : ");
        filename = sc.nextLine();

        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
```

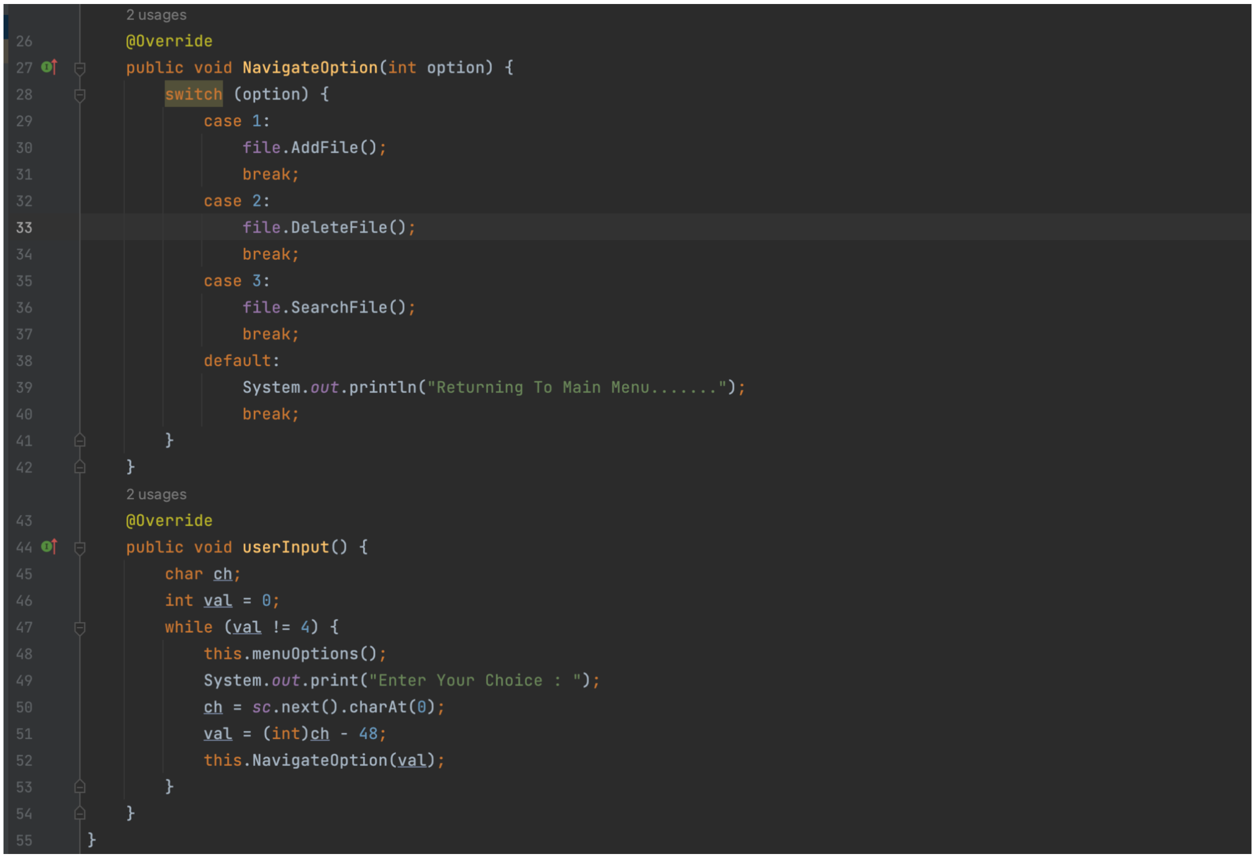


```java
    1 usage
    public void SearchFile() {
        boolean isFound = false;

        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**

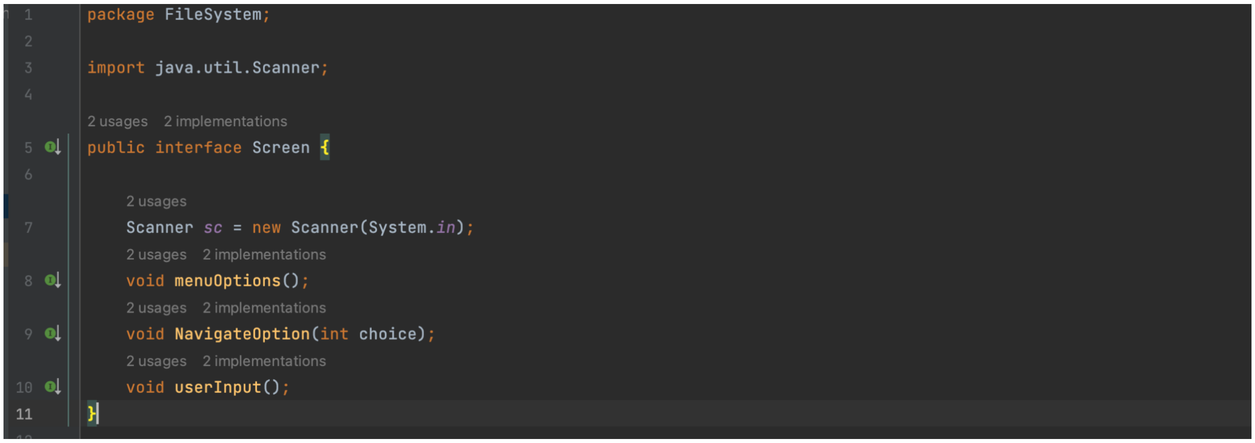


```java
package FileUI;

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);
        }
    }
```



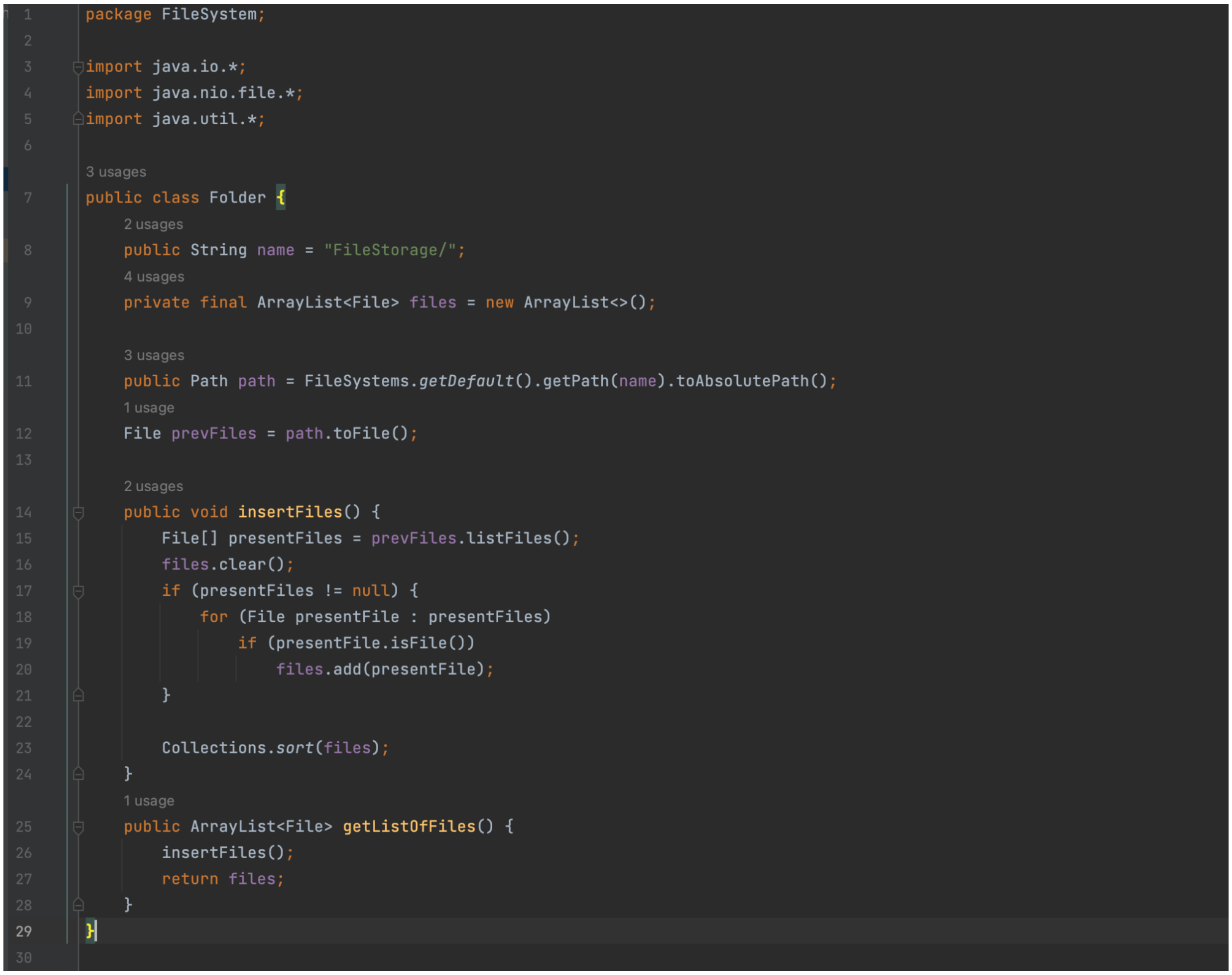
```
    2 usages
    @Override
    public void NavigateOption(int option) {
        switch (option) {
            case 1:
                file.AddFile();
                break;
            case 2:
                file.DeleteFile();
                break;
            case 3:
                file.SearchFile();
                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**



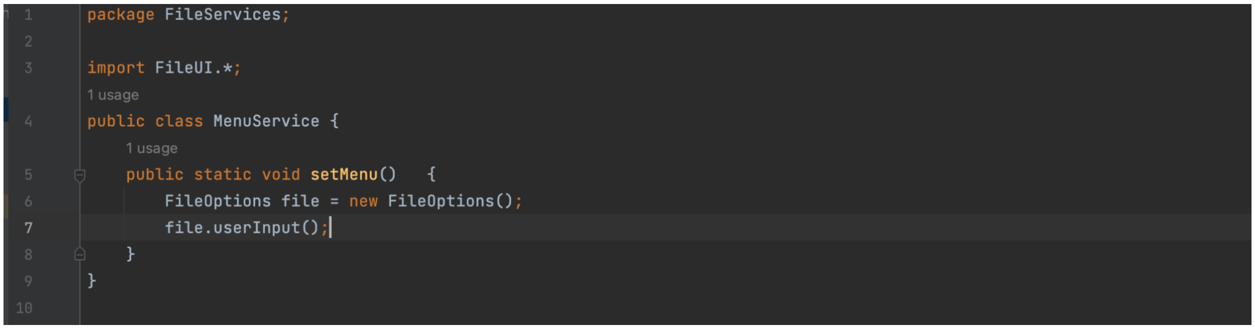
```
package FileSystem;

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**



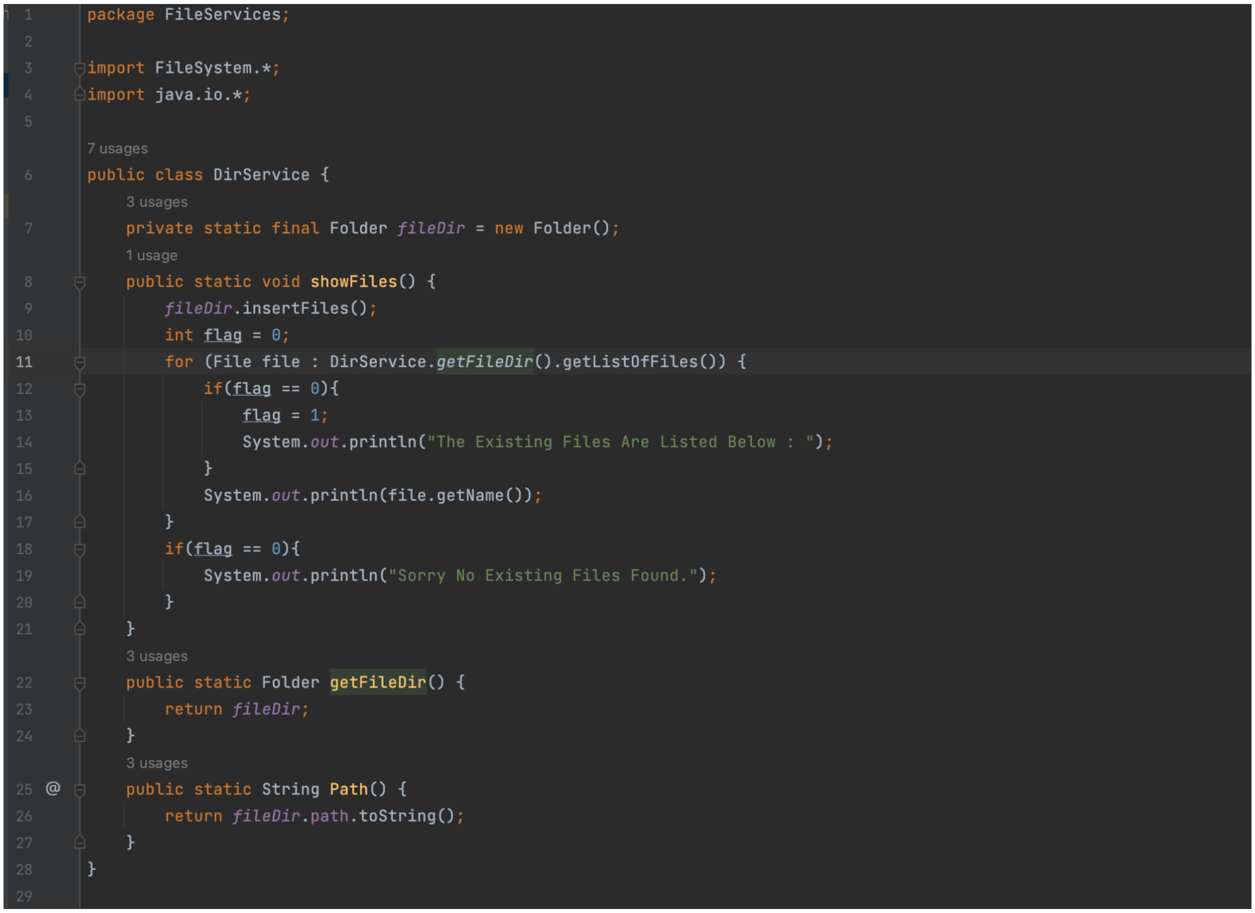
```
package FileSystem;

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**



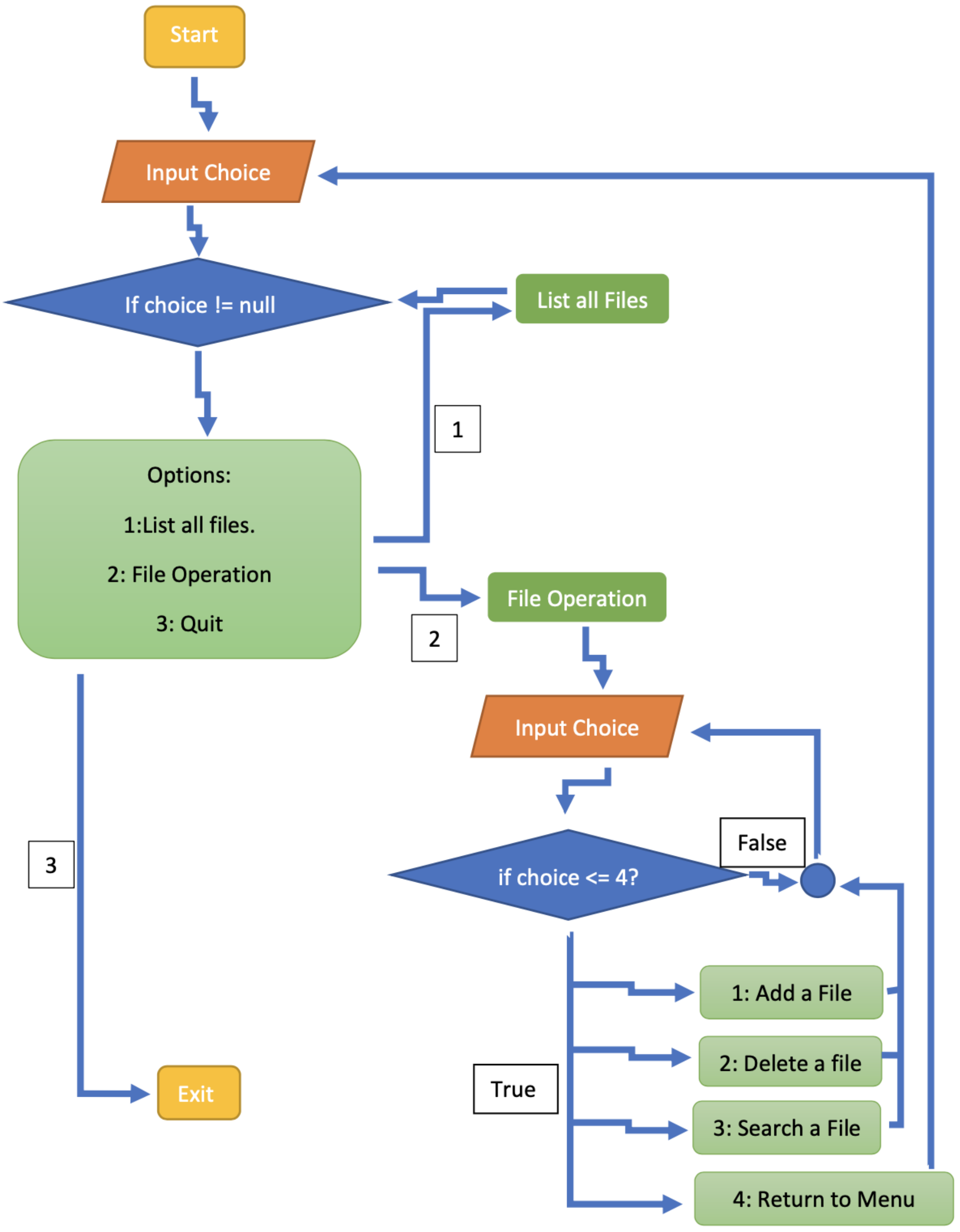
```
package FileServices;

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

```

- **DirService.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();
    }
}
```

FLOW CHART


Start
Input Choice
If choice != null
List all Files
1
Options:
1:List all files.
2: File Operation
3: Quit
2
File Operation
Input Choice
False
if choice <= 4?
3
1: Add a File
2: Delete a file
True
3: Search a File
Exit
4: Return to Menu

ALGORITHM

```
    ● Start
    ● Input From The User
    ● While val != 3 perform next Step
    ● Switch(val)

        Case1: List all files in the directory and go back to Step 2 afterwards.

        Case2: Go To Step 5

        Case3: Go To Step 6

        default: Return back to Step 2

        [End Of Switch Case Block]

[End Of While Loop]

    ● Input another choice val from user for File Operations
    ● While loop val != 4 perform the next Step.
    ● Switch(val)

        Case1: Add a file.

        Case2: Delete a file.

        Case3: Search a file.

        Case4: Return to Step 2.

            ● default: Return back to Step 5.

        [End Of Switch Case Block]

[End Of While Loop]

    ● End The Program.
    ● Stop.
```

CONCLUSION

- The Prototype has a good interface with CLI(Command Line Interface).
- The Prototype is platform independent and robust.
- We can enhance the prototype by a good GUI for more friendly and better user interaction.
- This Prototype can also be upgraded with adding authentication and security features to make it more secure and reliable.
- Users can easily use this prototype and perform the operations.