File - F:\MyWork\FullJavaStack\Projects\LockedMe - Project\src\com\simplilearn\project\LockedMe.java

```java
package com.simplilearn.project;

import com.simplilearn.project.FileManager;

import java.io.IOException;
import java.util.Scanner;

/*
    ========================================
        Title:  LockedMe.com site
        Author: Anas Al-Mughamsi
        Date:   1 Oct 2021
    ========================================
 */

public class LockedMe {

    private static Scanner input;

    public static void main(String[] args) throws IOException {
        welcomeMenu();
        showMenu();
    }

    public static void welcomeMenu() {

        System.out.println("-------------------------------------");
        System.out.println("Welcome to LockedMe.com site");
        System.out.println("-------------------------------------");

        System.out.println("\n\n\t\t*************************************");
        System.out.println("\t\tWelcome the main menu of LockedMe.com");
        System.out.println("\t\t*************************************");
        System.out.println("\t\t--Develop by Anas Al-Mughamsi--");

    }

    public static void showMenu() throws IOException {


        System.out.println("\nSelect one of these options below: - ");
        System.out.println("1. Show all files");
        System.out.println("2. File Managements");
        System.out.println("3. Exit");

        input = new Scanner(System.in);
        int selection = input.nextInt();

        switch (selection) {
            case 1: FileManager.showAllFiles();
                break;
            case 2: managementMenu();
                break;
            case 3: FileManager.closeApp();
                break;
            default:
                System.out.println("INVALID OPTION!!\nPlease try again");
                showMenu();
        }
```

```
    }

    public static void managementMenu() throws IOException {
        System.out.println("1. Add a file.");
        System.out.println("2. Delete a file.");
        System.out.println("3. Search a file.");
        System.out.println("4. Return to main menu.");
        System.out.println("5. Exit");

        input = new Scanner(System.in);
        int selection = input.nextInt();

        switch (selection) {
            case 1: FileManager.addFile();
                break;
            case 2: FileManager.deleteFile();
                break;
            case 3: FileManager.searchFile();
                break;
            case 4: showMenu();
                break;
            case 5: FileManager.closeApp();
                break;
            default:
                System.out.println("INVALID OPTION!!\nPlease try again\n");
                managementMenu();
        }
    }

}
```

```java
package com.simplilearn.project;

import java.io.File;
import java.io.IOException;
import java.util.*;

public class FileManager {

    /*  this class has
     1. show all files
     2. add file
     3. delete file
     4. search file
     5. return to main menu
     6. Close app - done
     */

    static final File folderPath = new File("F:\\MyWork\\FullJavaStack\\Projects
\\Phase - 01\\FilesForLockedMe");
    LockedMe lockedMe = new LockedMe();
    private static Scanner input = new Scanner(System.in);


    public static void showAllFiles() throws IOException {

        if(folderPath.list().length == 0) {
            System.out.println("\t--Folder is empty--");
            LockedMe.showMenu();
        }
        File[] showFiles = folderPath.listFiles();

        for (int i = 0; i < showFiles.length; i++) {
                System.out.println(showFiles[i].getName());
        }
        LockedMe.showMenu();

    }

    public static void addFile() throws IOException {

        System.out.println("Enter the name of file: ");
        try {

            String fileName = input.nextLine();
            fileName = fileName + ".txt";
            File newFile = new File(folderPath + "\\" + fileName);

            if (newFile.createNewFile()) {
                System.out.println(fileName + " File created");

            } else {
                System.out.println("File already exist in this folder, at
location => " + newFile);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        LockedMe.showMenu();
    }
```

```
    public static void deleteFile() throws IOException {

        if(folderPath.list().length == 0) {
            System.out.println("\t--Folder is empty--");
            LockedMe.showMenu();
        }

        System.out.println("Enter the name of file you want to delete (Enter '0'
return previous menu) : ");
        String deleteFile = input.nextLine();

        if (deleteFile.equals("0")) {
            LockedMe.managementMenu();
        }

        deleteFile = deleteFile + ".txt";

        File fileToDelete = new File(folderPath,deleteFile);
        if(fileToDelete.exists() && fileToDelete.isFile()) {
            fileToDelete.delete();
            System.out.println("File '" + deleteFile + "' deleted from " +
folderPath);
            LockedMe.managementMenu();
        } else {
            System.out.println("File not exist... please try again");
            deleteFile();
        }
    }

    public static void searchFile() throws IOException {

        if(Objects.requireNonNull(folderPath.list()).length == 0) {
            System.out.println("\t--Folder is empty--");
            LockedMe.managementMenu();
        }

        System.out.println("Write the name of file you want to search: ");
        String searchValue = input.nextLine();
        searchValue = searchValue + ".txt";

//        File[] fileName = folderPath.listFiles();

        String[] fileName = folderPath.list();

        for(String name: fileName) {
            if(searchValue.equals(name)) {
                System.out.println("File [" + searchValue + "] found at location
 " + folderPath);
                LockedMe.managementMenu();

            } else {
                System.out.println("File not Found");
                LockedMe.managementMenu();
            }
        }

    }
```

```
    public static void closeApp() {
        System.out.println("Thank you for using our app");
        System.exit(1);
    }

}
```