**SPRINT DOCUMENT**

**Total number of sprints:-2**

**Time for each sprint :-2 hours**

**Sprint 1**

**AIM:-We aim to develop the user interaction page with one functionality to display the files in a directory.**

**The following steps will be followed in oder to achieve the aim of the sprint:-**

**Step1: -**

**(i):-Designing the layout which gives the option to the user to retrieve the files from the directory.**

**(ii)-The welcome layout will contain the welcome page displaying the name of the application and the presenter.**

**(iii)-This can simply be achieved by printing the names of the website and the presenter and to make it look interactive we can use some symbols like(==,\*\*) to look good.**

**(IV):-Then an option will be displayed over the screen the total number of files present in the directory in an ascending order.**

**PATH:-“D:\\”-This is the path used in the local machine.**

**This will be achieved by using a switch case condition and define the display method and calling it.**

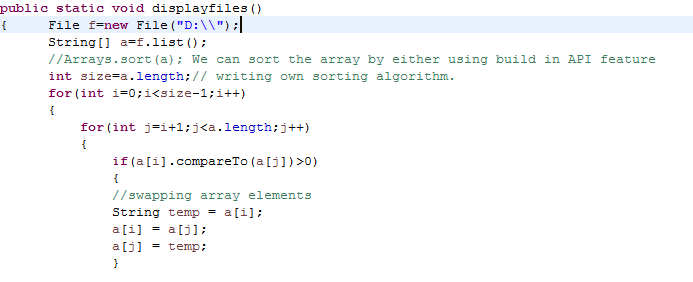
**Using the file handling concept by making a pointer to the directory**

**File f=new File (“D:\\);**

**(v)-The total files including the directories will be fetched in a strings array with the help of list() method of the file io api.**

**String[] a=f.list();**

**(vi):-Now to separate the files from the directories stored in the strings array a will be done by using a method called isFile().**

**(vii)Now we will separate the files and store them in the array with the help of isFile() method . After which the sorting algorithm will be performed on the array to display** 

**Sprint 2**

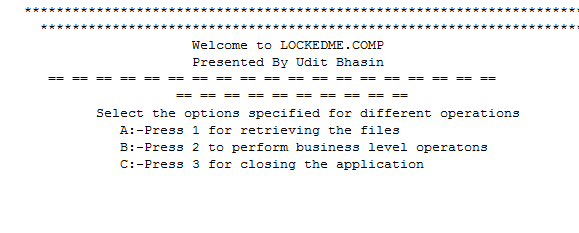
**AIM:-In this sprint we will define the other two features and combine with the output of the first sprint to fulfil the user requirement.**

**Steps**

**1-We will add two more choices to the existing choice of displaying the files .the other two choices would be:-**

**(i)performing different operation on the file.**

**(ii)closing the application**

**These two functionalities will be added to the previous functionality . The user will be provided to give an input either 1,2,3 to do the specific operation which will be implemented via switch case .   
2:-The 2nd option will be subdivided into another switch case which will allow the user to perform add/delete/search .**

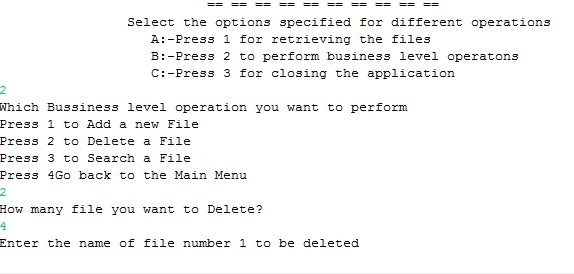
**The following functionalities will be achieved by building a separate function for each one of choice.**

**The deleting function will be implemented by taking the input in two forms:-**

**1-number of files to be deleted**

**2-the name of each file .**

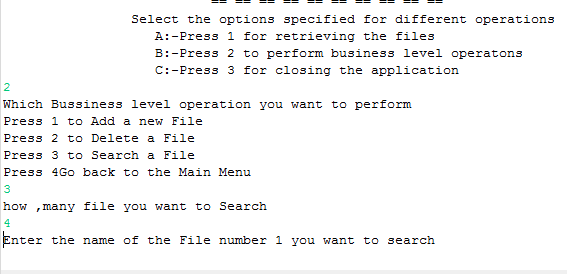
**The name of each file will be parsed to the file object pointing to the directory and delete it by using the delete method of the file API.**

****

**The searching function will be implemented by taking the input in two forms:-**

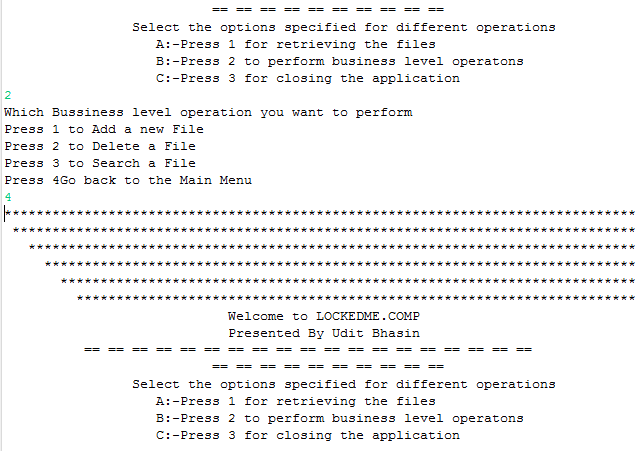
**1-number of files the user wants to search.**

**2-the name of each file .**

**The name of each file will be parsed to the file object pointing to the directory and search it by using the exist() method of the file API and giving the location of the file.**

**4:-Going back to the previous menu.**

**To achieve this we will simply define a function which will call the main function . Which will take the user to the main menu.**

****