Documentation of flow of the application:

Document and apllication specification-

Project name- Lockerr File

Developer name- Rahul Rana

GitHub Account name- Rahulpersie66

Github repostitry link- https://github.com/Rahulpersie66/Phase-1-Project-SimpliLearn.git

IDE- Eclipse IDE used for creation this appliaction.

Programming language- JAVA is used.

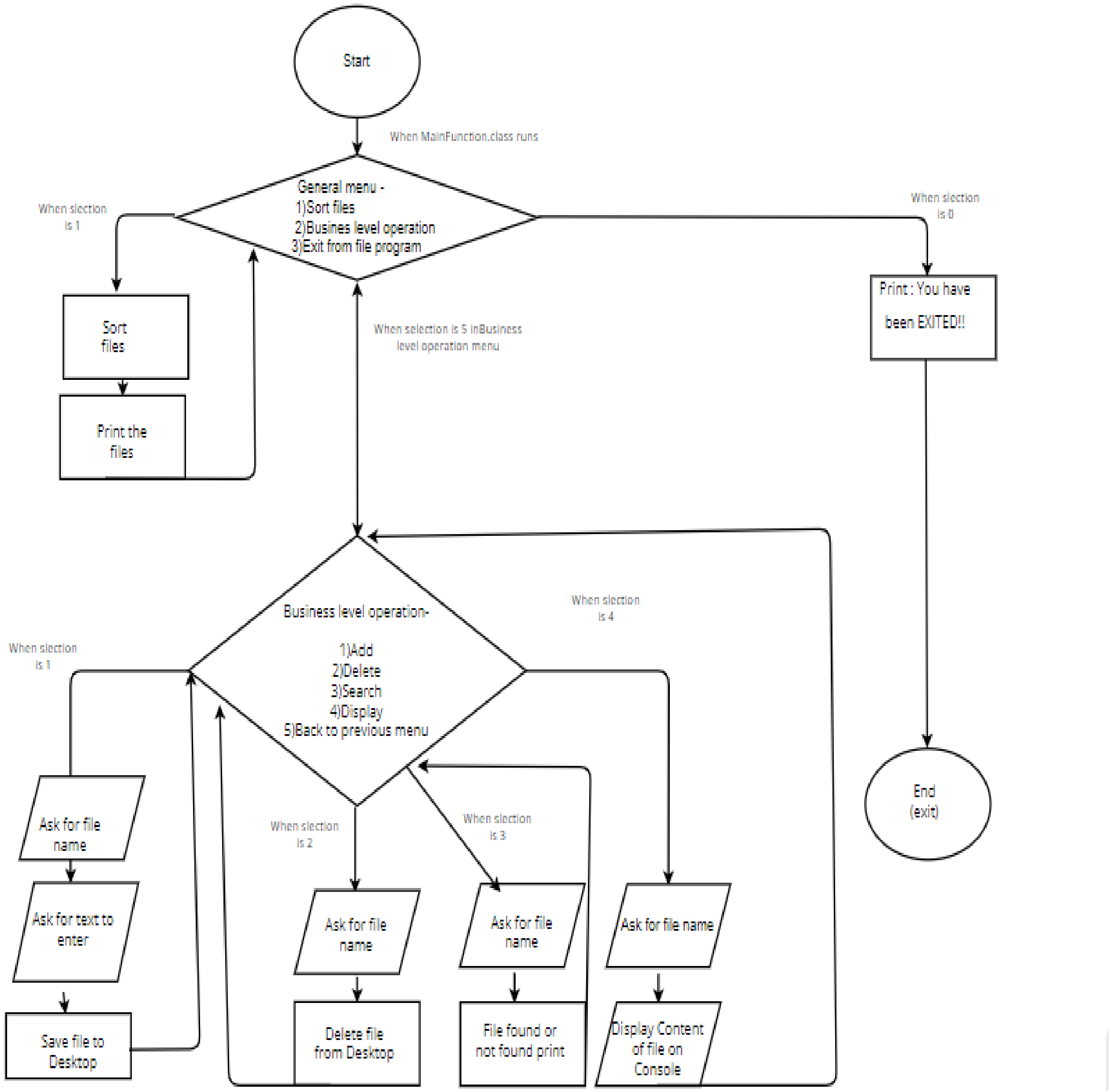
Data structure/Collection – Tree set

Searching technique- for searching files – File class function File.exists() used.

Sorting- Tree set collection has pre-defined function for sorting. By default it sort the list in ascending order.

Libraies & class used- File Output Stream, File Input Stream, File (for search file, set path to add the file, for delete file)

Data flow diagram according to algorithm-



Algorithm for File addition and deletion from desktop using File Input Stream and File Output Stream-

Step-1 Take input using console and showing output on console. And print the welcome screen alongside with Developer name and application name and note how to give input or output.

Step-2 Start from entering input correspond to option such as sort files, business level operation or exit.

Step-3 Load all files on the desktop in the Tree set object to show output on the console. Since tree set are already sorted so no need to construct function for sorting the files. Just print the Tree set object.

Step-4 Then build Business level operation show menu and build corresponds functions

Step-5 For addition of file use File Output Stream and add the path where you want to create file.

Step-6 For deletion of file use File class delete function to delete files from desktop and then reflect the changes on tree set object.

Step -7 After completing this operation reflect file name in the Tree Set object created in main function class.

Step-8 Create search operation using File class and check, if file exist on the desktop or not using inbuild function of class **file.exists().**

Step -9 Build show content function using File Input Stream and show the data enter in the file.

Step -10 And give a option to user to go back to previous menu by entering required number using switch case.

Step -11 After getting back to previous menu user can exit the program by entering 0 as input in console, all other input so do call function and for invalid input it should ask user to enter number until user enter number. This can be done by while loop alongside scanner class function **hasNextInt** function.

Step by Step process from sprint planning to product release-

Sprint 1-

Sprint Planning – In this phase, I construct a plan to search for best framework to store string file names that are already presented on desktop and how to fetch their name and develop a Welcome menu using switch case and how to come back from another menu to this welcome menu again. And not exiting until specified number is entered by the user.

Objective of sprint 1-

-to construct a switch case for welcome menu

-to construct a switch case for Business level operation menu

-to find out best collection to store the string name of files

- to fetch all file name of desktop and then store in the collection type oject

Process & development work-

Constructed a welcome menu which consists of developer name, product name and note how to enter the input in the console and what input it is going to take. Constructed this menu using switch case that

-if 1 is entered it will print sort list of files

-if 2 is entered it will go for business level operation

-if 0 is entered it will go for exit from program.

While user entered 2 then it should go for business level operation switch case with option-

-if 1 add the file

-if 2 delete the file

-if 3 search for the file

-if 4 display the file

-if 5 go back to previous menu.

So to go back to previous menu just call the welcome menu function from the class.

After a research from internet and guided by my course teacher, I selected tree set as my collection set for storing file names as it has in build feature of sorting files in **ascending order**.

Now main task was to fetch files name that were presented in desktop and then to store into the tree set object. So after some research and study I used **File System view** and its function to abstract desktop path and file class file list function with for each loop to abstract each file name from desktop.

Sprint retrospective-

That the program was still accepting the invalid string input and then get crashed, during the selection from menu and exited the program due to input mismatch exception. So to overcome it added the while loop with scanner class function **hasNextInt** to make it work for and will terminate program because of input mismatch. As a consequence, it work only when integer is entered and for other input it will ask for enter a number.

Sprint 2-

Sprint planning- In this phase of sprint 2, I decided to add function such as add files, delete files, search file, show content of file and sort file name functionality to program.

Sprint Objective-

-to add file function and how to make it work to add files on desktop

-to delete file function and how to make it work to delete files on desktop

-to build search function and how to search file from desktop

-to build display file and how to display content of the files

-to sort file names and print them

Process and development work-

Make another class **FileOutputStreamDemo** and constructed their function to add/create file and add it on desktop. By using File output Stream constructed a file that a user want and enter data in it, but in bytes form data is entered. And then call the function from main class switch case option corresponds to choice 1 and also reflect the changes in the Tree Set object.

Constructed another function in same class to delete file and delete it from a desktop. By using File class and its function **file delete** and then reflect the changes in the tree set object where data is stored. And added this function of file delete corresponds to switch case selection of business level operation choice 2.

After this constructed another function to search a user enter or user specified file present or not. For that I used file class and its function **file.exists()** to check whether it is presented on desktop or not. And then print File found if there else not. And added this operation corresponds to choice 3.

Similarly, added Display function corresponds to choice 4 of business level switch case which works on basis of **File Input steam** class to print the data entered in files and to print it using while loop alongside File input stream class function is read to print data inside the file.

To sort file, I just used tree set object to print file name as it is. Because tree set has ability to sort file in ascending order, so we just need to print it as it is. And added this option corresponds to Welcome menu choice 1.

Conclusion-

This application can create, delete, search, display or sort files list from desktop in a efficient way using console as input and output.

Can enhance application by-

→Making it work to read and display every type of file such as docs, pdf,excel etc.

→ Already presented files data should displayed in string format and not in byte format.

→Make it user friendly by making it application of android or pc rather than taking input in console.

→Should update file name.

→can enhance it by creating multiple files or deleting multiple files.

Unique selling points-

→No need to create file by clicking to create file and can add data at same time while creating, no need to open other application for it.

→This one application can handle 5 task –add, delete, search, display and sort task just by entering simple input number.

→ Efficent in way as using tree set collection and provide a simple interface to use.