**Phase 1**

**Course-end Project 1**

**Project Objective :**

As a Full Stack Developer, complete the features of the application by planning the development in terms of sprints and then push the source code to the GitHub repository. As this is a prototyped application, the user interaction will be via a command line.

**Project Overview :**

The project is designed to provide a user interaction through command line which provides ability to user to manage and handle files. Files are created and stored in the form of array or list.

**Project Capability :**

The project is capable of performing operations like retrieving , creating and deleting of files . It will also have the ability to sort the existed files after taking various operations on files by user. While listing files the project will list down the present files in ascending order . Redundancy of files is also managed by the project in order to remove duplicate files and also does not allow user to create duplicate files. It will also ensure to notify the users when they are trying to delete non-existing files.

**Project Appearance :**

Project is made to interact and execute through command line. First screen after execution shows welcome page of the application and displays the menu consisting of options corresponding to the file operations.

**User Interactions :**

Users are presented with the list operations as menu in the first place and also provided with corresponding inputs to enter for every file operation. They are also asked to re-enter the option if they give invalid option as input. Every time after a completion of operation it will ask user whether to continue the particular file operation one more time or return to main menu. Simple single digits are asked as input to make it more user-friendly.

**Project Git hub link :**

https://github.com/SindhujaSeelam/javademos/blob/main/Project/src/projectpackage/FilesProject.java

**Sprint Planning :**

I have divided project into 3 sprints.

Sprint 1:

In sprint 1 I have configured all the requirements of the project at first and then designed rough outline of the project. Created main() method and static methods which invoke file operations like add(),delete(), search(). It took me 2 days to understand the project and start with basic declaration of static methods mentioned above.

Sprint 2:

In sprint 2 I have designed the first screen of the project after its execution which consists of welcome note and menu having list of operations that can be performed on the files. The menu has three options 1.Display files in ascending order 2. File handing operations menu 3. Exit Application . I have also implemented code for all the above methods using loops and conditional statements. It took me 4 days for this particular sprint.

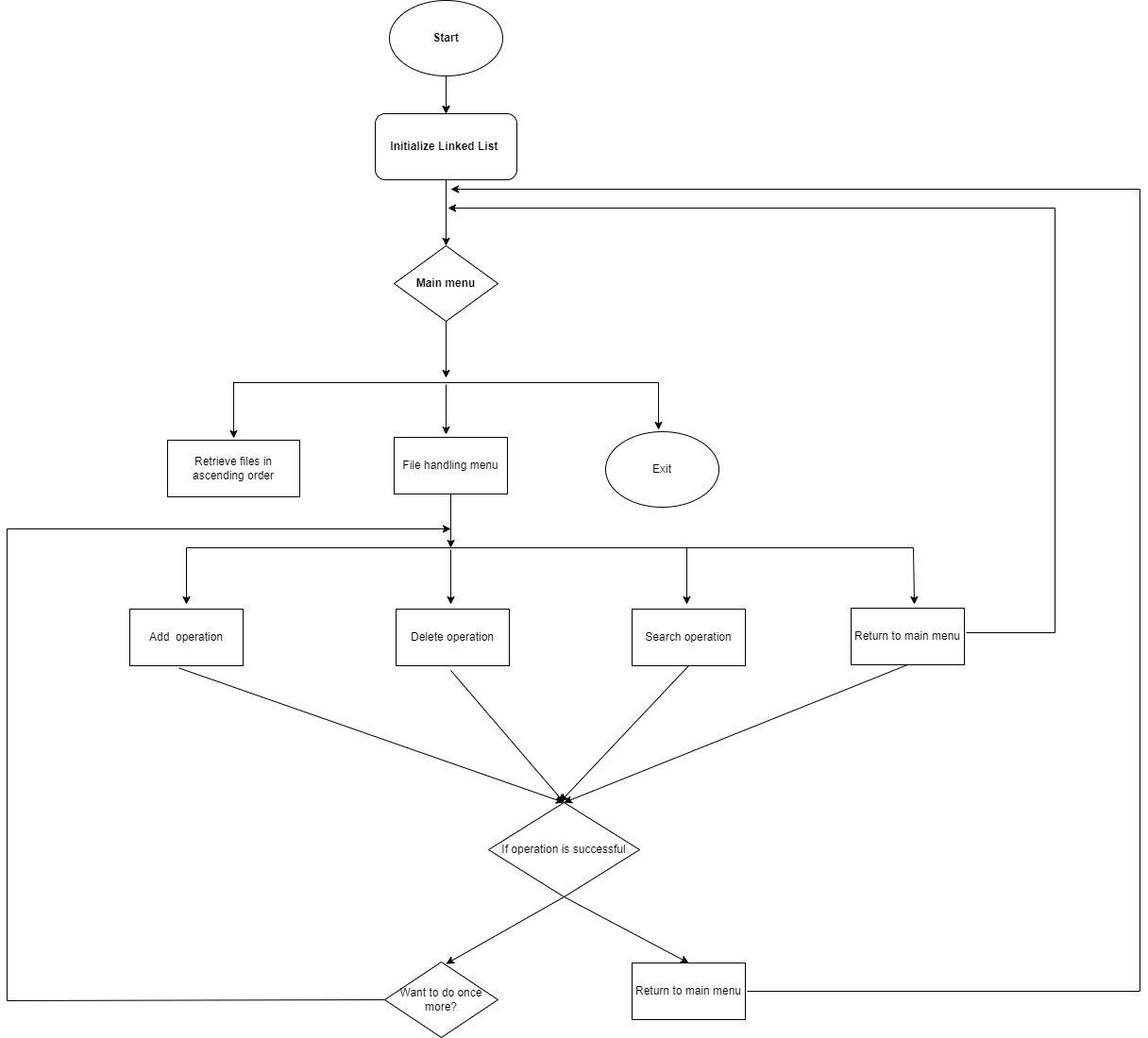
Sprint 3:

In sprint 3 I have implemented try blocks for scanner class and also defined else cases if user give a invalid input irrespective of provided options. I have implemented static methods like addmore(), deletemore(), searchmore(), to provide user a option by which they can perform any operation more than once and also gave a option if they want to return to main menu. It took me 3-4 days to assume and implement all possibilities of program from not throwing errors and exceptions.

Java Concepts Used:

* Collections(Linked List)
* Scanner Class
* Try and catch block
* Conditional Statements
* Switch statement
* Break statement
* While Loop

**Flowchart of the program**



Algorithm of Application:

Step 1: START

Step 2: run main()

Step 3: Calling Main menu

Creating Linked List

Adding some basic files to list

Step 4: Displaying main menu and taking user input from

1->printing file names in ascending order

2->To call file handing menu

3->Exit Program

Step 5 : if option 1 is given as input by user

Files are displayed in ascending order

Step 6: if option 2 is given as input by user

Then File handling menu displayed

1-> To add file

2-> To delete file

3-> To Search a

4-> To display main menu

Goto-> Step 4

Step 7: if user selects option 3 in Main menu

then program is terminated

Step 8: STOP