Phase 1 Project Abstract

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 in Brief:

This project is built to perform the operations as required related to files as adding new files, deleting an existing file, printing the list of files, searching for the file and etc. With the basic UI so that user can input the required operation to start by entering the relevant option from the displayed menu which have choices like retrieving file in ascending, second choice to move control/flow to file handling operations adding file, deleting file, searching for file and if not present returning the result statement, also returning to main menu, finally third choice to end the program.

Sprint Planning:

I have divided the sprint into 4 parts.

* Initially, in the first sprint I took the time to understand the requirements and design the rough outline of the project and just created the basic flow creation and layout of structure. Took a day of time to briefly understand and plan according to the requirements and availabilities.
* In Second sprint I have proceeded with creating the main menu and its control, in such a way that I get main menu with a continuous iteration after every operation completed in it, so I have placed the printing of files in ascending order as first option, the second option was left blank for later assigning of file handling operations and at last I have used the exit command Systems.exit(0) as 3rd option, so that User can exit from program whenever required. This sprint took me 3-4 days of time.
* After second sprint created a TreeSet to store the filenames, as TreeSet doesn’t allow duplicates and follow natural ordering, also that is what we need, due to that reason I have taken TreeSet data Structure.
* In third sprint I have created the static method for the file handling operations as mentioned adding file, deleting file and searching for file, also said in requirements that the control has to revert back to main menu so I have made as fourth option, for every input statement took care of exceptions by using try and catch blocks around them. Even when the operations are performed the user will see the result of the operation in the UI screen. So that the user can act according to it. This sprint took me 4-5 days of time.
* I have make sure that whatever other lines code apart from input statements might be raising the exceptions to be surrounded with try and catch blocks, and tested the code with different input and different datatypes which might not be feasible to the Scanner class to accept at that particular line of code. This sprint included merging the units of and some modifications to code like by adding the call of file handling operators in the main menu and doing the final run along with some of the noted general test cases. This fourth sprint took me 2-4 days to complete.

Java Concepts Used:

* Class, Objects, Methods, and Variables
* Exceptional Handling
* Collections(TreeSet)
* Control Flow Statements
* Scanner Class
* Looping Statements
* Jump Statements

Project GitHub Link:

[javademos/Origin.java at main · rChappidi10836/javademos · GitHub](https://github.com/rChappidi10836/javademos/blob/main/phaseprojects/phaseprojects/src/com/lockedMe/Origin.java)

Conclusion:

The Project can be further developed more as there is lot of scope left, like for searching the user can input more accurate code which makes it more efficient, the lines code can be more manipulated to increase readability. Even the project can go more further if it converted into applications handling. The unique point about the project is that once the user start using the project through User Interface there is not need to run the project again with precise navigation and option selection, the code is designed in such a way that it will automatically reverts back from file handling operations menu to main context.

Flowchart of Algorithm:

Diagram

Description automatically generated

Algorithm of Application:

Step 1: START

Step 2: run main

Creating Treeset

Step 3: Calling Main menu

Step 4: Displaying main menu and taking user input from

1->printing file names in ascending order

2->Calling file handing menu

3->End of Program

Step 5: if option 2 chosen to call File handling

Then File handling menu displayed

1-> To add file

2-> To delete file

3-> To Search and access File

4-> goto step 4

Step 6: if user selects option 3 in Main menu

then program is terminated

Step 7: STOP