COURSE: JAVA FULL STACK DEVELOPMENT

JAVA FSD PROJECT PHASE 1

PROJECT NAME: FILE STORAGE

SIMPLILEARN

Developed by,

GANAVI M

Table Contents

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.

2. PROBLEM STATEMENT:

Company Lockers Pvt. Ltd. hired you as a Full Stack Developer. They aim to digitize their products and chose LockedMe.com as their first project to start with. You're asked to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for the budget approval. Your manager has set up a meeting where you're asked to present the following in the next 15 working days (3 weeks):

- ✓ Specification document - Product's capabilities, appearance, and user interactions
- ✓ Number and duration of sprints required
- ✓ Setting up Git and GitHub account to store and track your enhancements of the prototype
- ✓ Java concepts being used in the project
- ✓ Data Structures where sorting and searching techniques are used.
- ✓ Generic features and three operations:
    - ➢ Retrieving the file names in an ascending order
    - ➢ Business-level operations:
        - ▪ Option to add a user specified file to the application
        - ▪ Option to delete a user specified file from the application
        - ▪ Option to search a user specified file from the application
        - ▪ Navigation option to close the current execution context and return to the main context
    - ➢ Option to close the application

The goal of the company is to deliver a high-end quality product as early as possible.

3. SPRINTS PLANNING AND TASK COMPLETION:

The project is planned to be completed in 3 sprint. Tasks assumed to be completed in the sprint are:

Sprint 1:

1. Creating the flow of the application
2. Initializing git repository to track changes as development progresses.
3. Writing the Java program to fulfil the requirements of the project.
4. Design Classes and Methods
5. Design Exceptions
6. Code Data

Sprint 2:

1. List File Method
2. Create File Method
3. Search File Method
4. Delete File Method
5. Code the level one menu

Sprint 3:

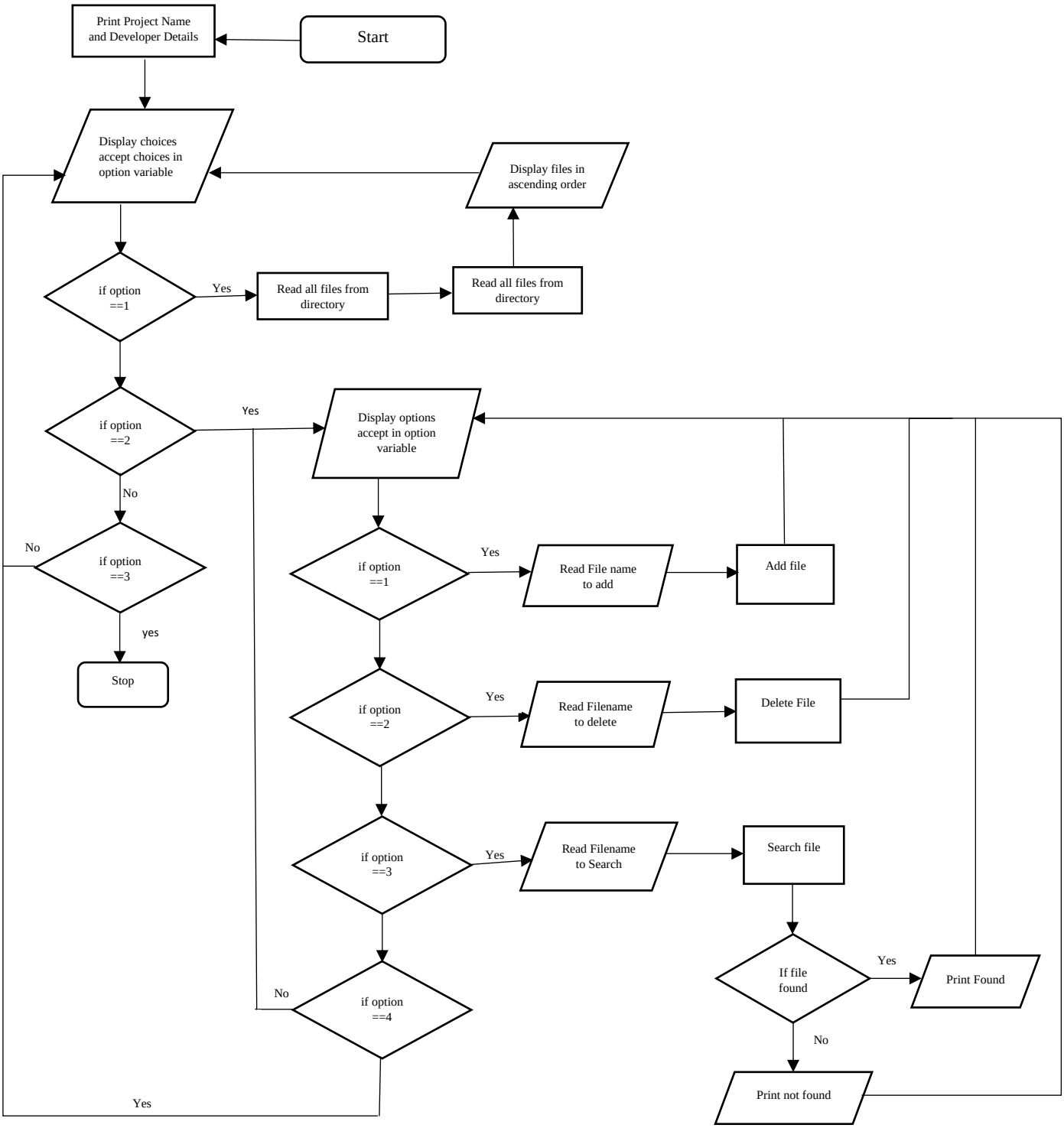
1. Code the level two menu
2. Call the Business Tier Methods in the Main File.
3. Testing the Java program with different kinds of User input
4. Pushing code to GitHub.
5. Creating this specification document highlighting application capabilities, appearance
6. Use String Formatting to display the output in a standard manner.

4. ALGORITHM AND FLOWCHART:

Algorithm-

1) Start
2) Print Options to all files, file menu and cancel
3) If choice is 1
    a) Read all files in the directory
    b) Sort files in ascending order using Collections.sort()
    c) Print all the files
    d) Goto 3
4) If the choice is 2
    a) Print Options to add, search and delete files
    b) If the choice is 1
        i) Accept file name from the user
        ii) Create a new file with that name
        iii) Goto a
    c) If the choice is 2
        i) Accept file name from the user
        ii) Delete the file with that name
        iii) Goto a
    d) If the choice is 3
        i) Accept file name from the user
        ii) Search for file with that name
        iii) If file found print found
        iv) Else print Not found
        v) Goto a
    e) If the choice is 4 goto a
    f) Goto 2
5) If the choice is 3 goto 7
6) Stop

Flowchart of the System:



5. CORE CONCEPTS IMPLEMENTED:

1. Encapsulation: Class in the data layer encapsulates all the file data by making them private and providing public methods to access them.

2. Abstraction: Abstracts all the implementation details from the presentation layer class and the interface.

3. Inheritance: The FileOptionsScreen.java extend/ inherit from the Runtime Exception class to provide user-defined exception messages if the file name provided by the user is not a valid name and if the file name provided to delete by the user is not available.

4. Polymorphism: Overrides the toString method to provide custom string output. It also overrides the compare. To method of the comparator method to provide the sorting capability. It also has overloaded constructors.

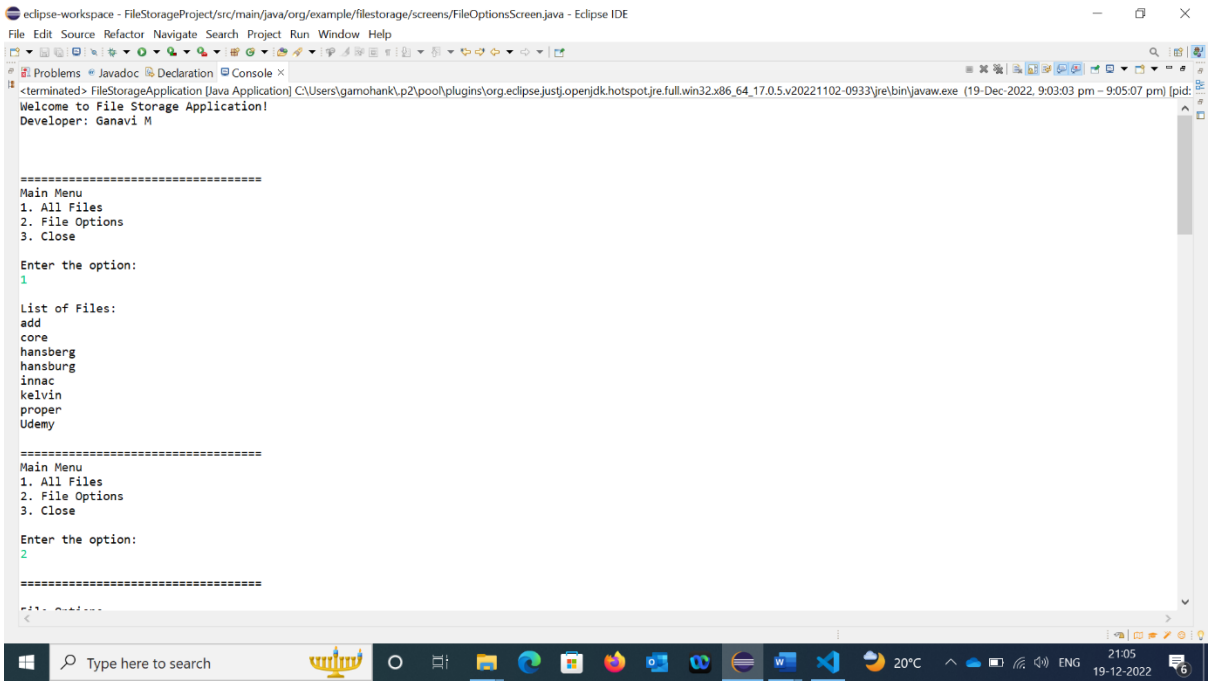
5. Linked List was used to store the list of files in the directory and Collections.sort() was used to sort them.

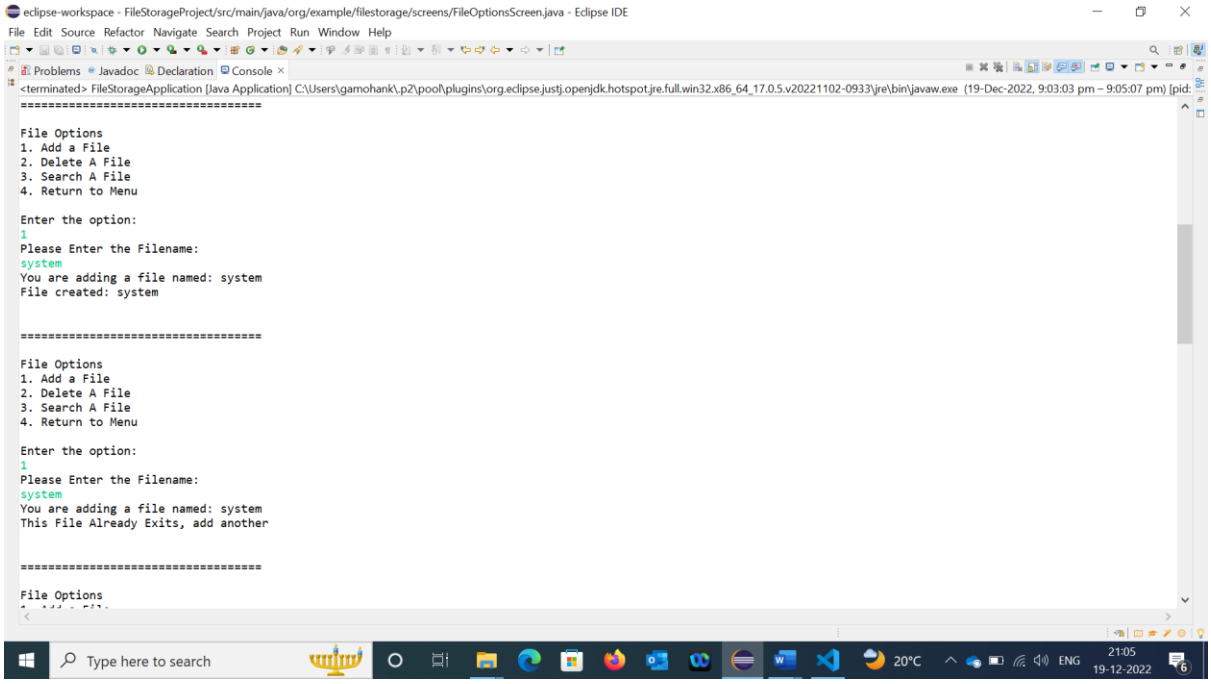
6. PROJECT GIT REPOSITORY:

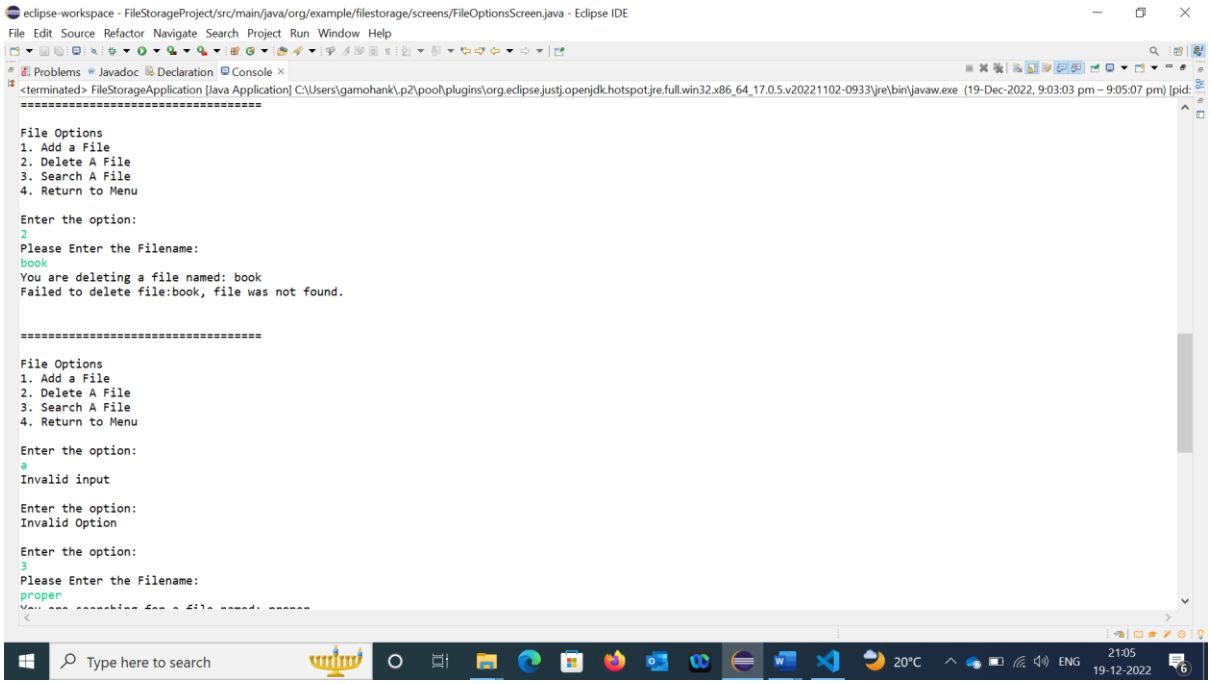
Link to the GitHub Repository -

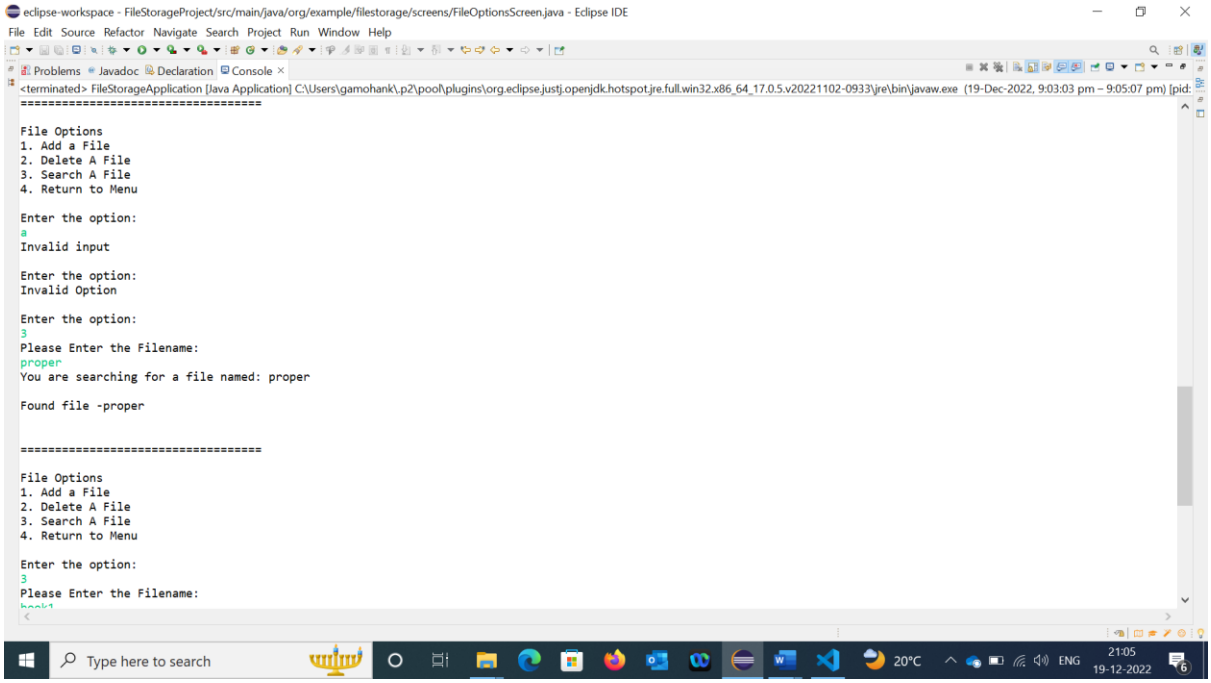
https://github.com/GanaviM/FileStorage_Application_JAVA_FSD_Phase1

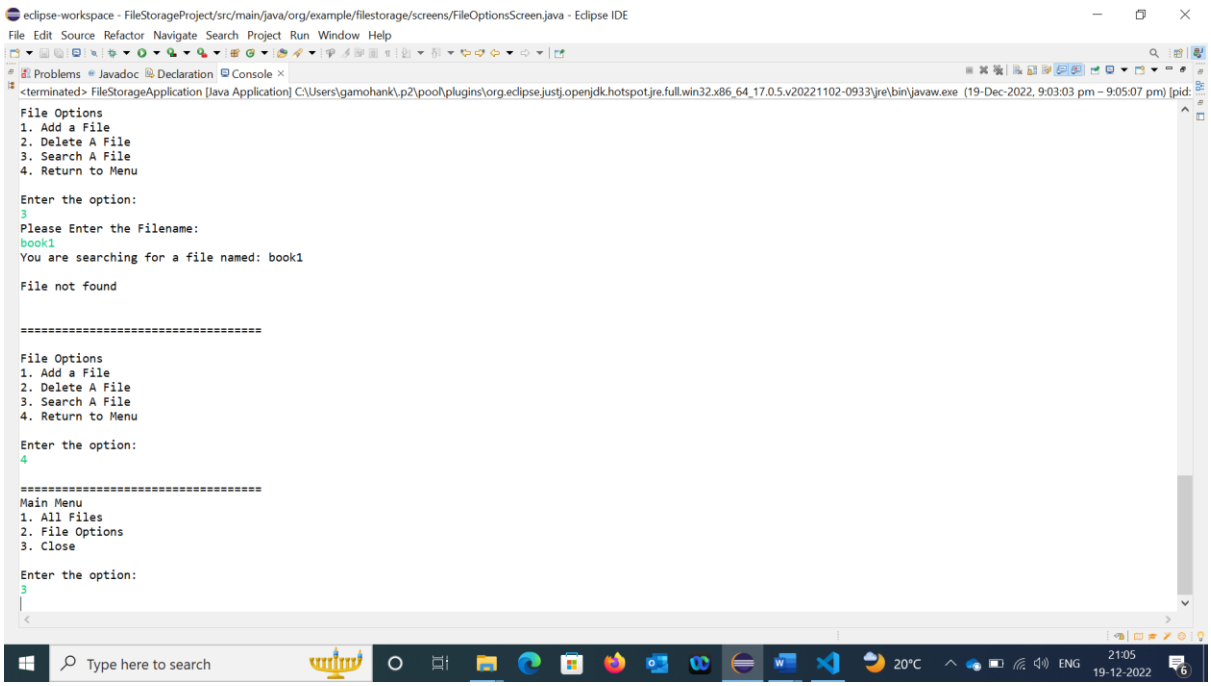
7. SCREENSHOTS OF PROGRAM OUTPUT:






eclipse-workspace - FileStorageProject/src/main/java/org/example/filestorage/screens/FileOptionsScreen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
<terminated> FileStorageApplication [Java Application] C:\Users\gamohank\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (19-Dec-2022, 9:03:03 pm – 9:05:07 pm) [pid:
==========
File Options
1. Add a File
2. Delete A File
3. Search A File
4. Return to Menu
Enter the option:
2
Please Enter the Filename:
book
You are deleting a file named: book
Failed to delete file:book, file was not found.
==========
File Options
1. Add a File
2. Delete A File
3. Search A File
4. Return to Menu
Enter the option:
a
Invalid input
Enter the option:
Invalid Option
Enter the option:
3
Please Enter the Filename:
proper
Type here to search
20°C
ENG
21:05
19-12-2022


eclipse-workspace - FileStorageProject/src/main/java/org/example/filestorage/screens/FileOptionsScreen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
<terminated> FileStorageApplication [Java Application] C:\Users\gamohank\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (19-Dec-2022, 9:03:03 pm – 9:05:07 pm) [pid:
==========
File Options
1. Add a File
2. Delete A File
3. Search A File
4. Return to Menu
Enter the option:
a
Invalid input
Enter the option:
Invalid Option
Enter the option:
3
Please Enter the Filename:
proper
You are searching for a file named: proper
Found file -proper
==========
File Options
1. Add a File
2. Delete A File
3. Search A File
4. Return to Menu
Enter the option:
3
Please Enter the Filename:
Type here to search
20°C
ENG
21:05
19-12-2022



8. CONCLUSION:

The file manager was created in 3 sprints. Special attention was paid to implement the pillars of the OOPM. Best practices were followed as and when possible. Rigorous testing was done to ensure that there are no spontaneous exits, and all exceptions are handled. Some Exceptions are handled using Custom exception classes. The throws and throw keywords were used to handle exceptions. The Comparable interface was used to provide the compare To method to help in sorting.