**Documentation**

BOB’S FILE MANAGEMENT SYSTEM

Developed by: Gautam Das

Email: [gautamdas251998@gmail.com](mailto:gautamdas251998@gmail.com)

Project Objective:

To develop an application which is used to perform operations on files using java.

Specifications:

1. User Interaction:

All the interactions of the user will be in the console only. This is a console-based application.

1. Product capabilities:

This product is capable of operations such as, adding files, deleting files, searching files and listing the files.

The flow and features of the application:

When the user opens the screen, he would get the welcome screen and see the details of the developer and below that main menu displays. Main menu consists of 3 options.

1. Option 1 to list all files in ascending order.
2. Option 2 to add, delete, and search files. All the interactions are from user inputs
3. Option 3 to exit the application.

I have implemented this using switch statements. If the user user provides invalid input rather than option (1-3) it throws an invalid option and allows user to re-enter their option. As this is user input applications, all the input is taken by user which is achieved by Scanner class.

In option 2 there are sub options, that is:

1. Option 1 to add files. User need to enter a file name to add it into directory. If name of file user trying to add exists already, then it shows message of file already exists, and user have to enter correct file name to add it to ‘rootFile’ directory.
2. Option 2 to delete files. As this is case sensitive application, user have to enter correct file name to delete it from directory. If the name of file which user try to delete doesn’t exists, then it shows file not found message. If the file exists then it deletes the file from that ‘rootFile’ directory.
3. Option 3 is provided to search a file user want to search. This is a case sensitive application. User have to enter correct file name to search file, from the ‘rootFile’ directory if file exists it returns found message. If file is not found, it returns ‘File not found’ message.
4. Option 4 is provided to the user to navigate to main menu. If the user selects option 4 then the application navigates user to the main menu.

Duration of Sprints required:

* I have divided this application in 2 sprints of 7 days each. In first 2 days I made a plan to how to create the application, the next 4 days I developed the application code for Main menu (Console), adding and deleting file and the last day of the first sprint I tested the Application.
* In the second sprint I developed code for listing files, searching files and adding case-sensitivity to the deleting and searching file operations. Then I tested the whole application. Afterwards I prepared the documentation and flow chart.

Flow Chart of The Existing System

Enter your option

Display welcome screen and developer details and switch case details

Start

Start using switch case

False

False

Case1

True

Case2

True

List all the files

Add, Delete, Search file

Case3

Enter your option

Exit the application

Finish

True

True

True

False

False

False

Back to main menu

Case4

Case3

Search file

Delete file

Add file

Case1

Case2

Tools and Technology:

* IDE used – Intellij Idea
* Technology used – Java
* Version - 14.0.1
* For Documentation – MS Word

GIT Source Code Link:

Source code of this applications has been pushed to GitHub. One can access that repository by clicking on below link:

<https://github.com/codergautam25/Full-Stack-Projects.git>

CORE JAVA CONCEPTS USED:

File handling, Exception handling, loops, Scanner inputs, condition statements, OOPS concepts, Arrays, Sorting, Searching, Collections, Java.io.

Few Functions and methods implemented in this application:

1. delete() : This returns Boolean value. It returns true if and only if directory is successfully deleted.
2. FileOutputStream(): This FileOutputStream constructor takes an argument of File class variable.
3. ObjectOutputStream() : This ObjectOutputStream() constructor takes an argument of FileOutputStream object;
4. writeObject() : This takes an argument of File class object.
5. file.exists(): This functions returns True if File already exists in the given path or returns false if File does not exists.
6. Arrays.sort() : It by default sorts Array in ascending order
7. System.in : System.in is an InputStream which is typically connected to keyboard input of console programs.
8. commonly passed to a command line Java application via command line arguments, files etc.
9. Switch Statements: Used to select one of many code blocks to be executed.
10. do While loop: The while statement continually executes a block of statements while a particular condition is true.
11. The new keyword is a Java operator is used to creates the object
12. Scanner: The Scanner class is used to get user input, and it is found in the java.util package.

Conclusion and USP:

This application interface is easy and simple to use by an end-user. Through this application a user can add, delete, search file. As well as it can list all the files that are already presented in the root directory. In future I will improvise the application by adding some security options for the application and storing files in online Database.

The USP of this application is that user can’t add file with same name and the best part is that everything is offline in this application and this gives a level of satisfaction to a user from security point of view.