**Documentation**

BOB’S FILE MANAGEMENT SYSTEM

Developed by: Gautam Das

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

OBJECTIVE:

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

PRODUCT CAPABILITIES:

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

USER INTERACTIONS:

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

Duration of Sprints required:

* I have divided this application in 2 sprints of 7 days each. In the First sprint I developed operations on adding files, deleting files.
* In the second sprint I developed operation on listing files and searching files. And also created the documentation.

GIT:

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>

In this repository you can access my works In phase 1 Project and link to that is given below, but its recommended to access my work through repository link given above.

However, I will be sharing direct link of my phase1 project below:

<https://github.com/codergautam25/Full-Stack-Projects/tree/master/out/production/Phase%201%20Project>

Steps to be followed to pull my remote repository to your system:

1. Open git and create directory by using command mkdir name (where name indicates directory name)
2. Check its location and navigate to directory and open git bash at that location
3. Create the empty git repository using command git init .

Now create pull request my using command git pull

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

1. Check the contents in directory
2. use command git add . to add files and command git commit –m “Type your message” to commit changes.

You can push to remote repositories using following commands

1. git init
2. git remote add origin “your repository URL”
3. git add .
4. git commit –m “Type your message”
5. git push origin master

JAVA CONCEPTS USED:

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

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.

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.

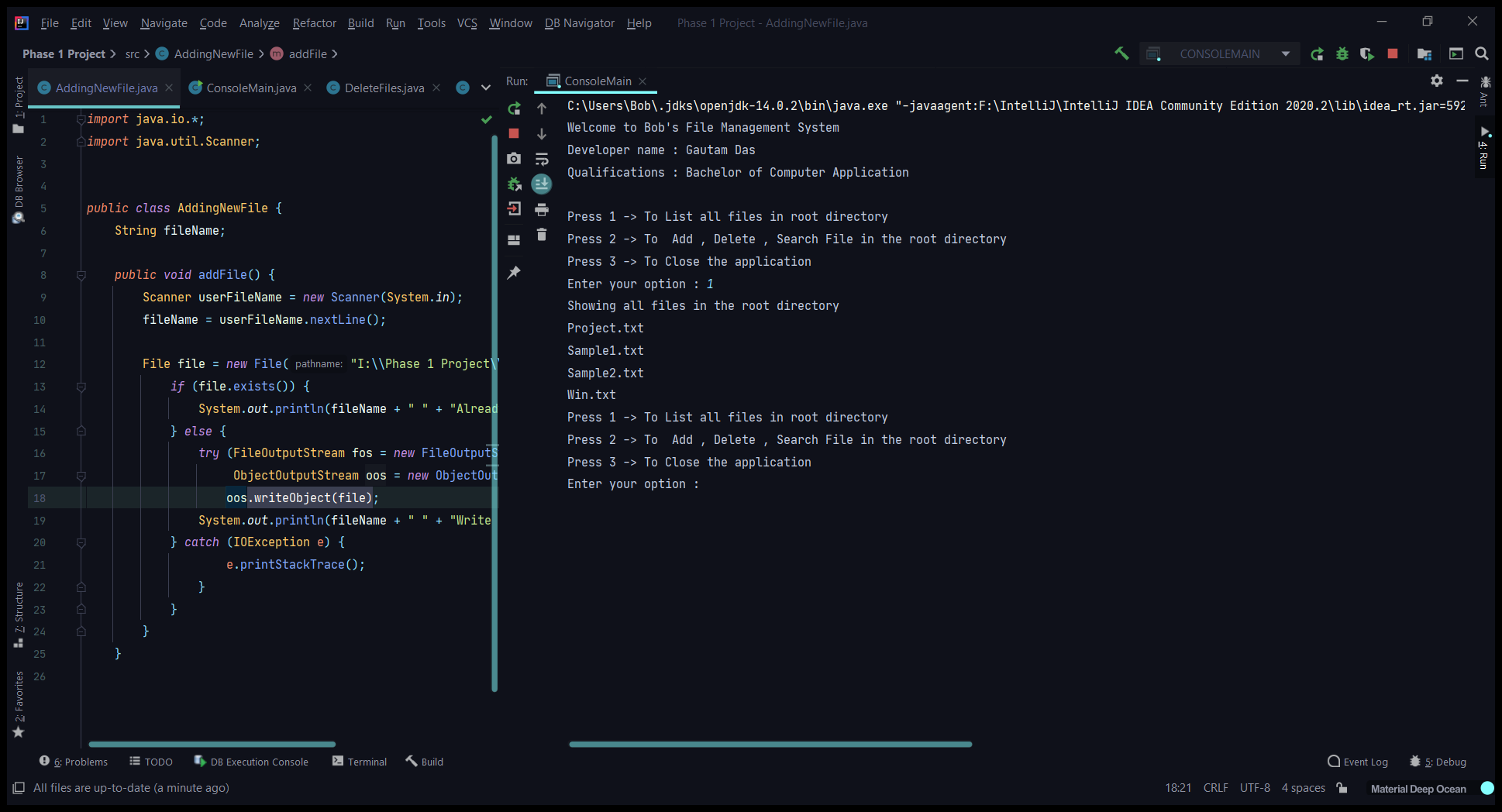
I have implemented this using switch statements. If the user user provides invalid input rather than option (1-3) it throws an invalid option and allows user to re-enter their option. As this is user input applications, all the input is taken by user which is achieved by Scanner class.

In option 2 there are sub options, that is:

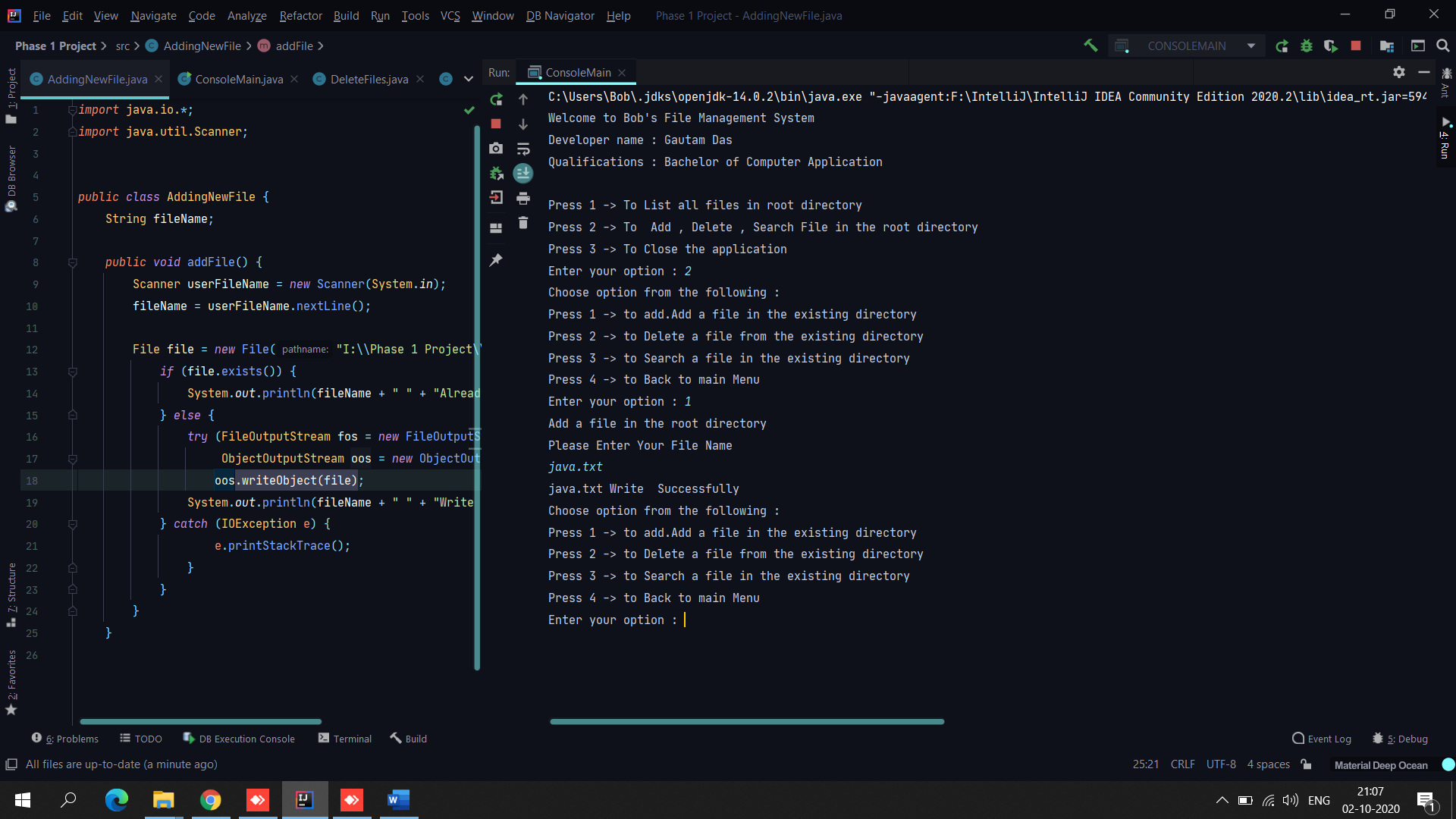
1. Option 1 to add files. User need to enter a file name to add it into directory. If name of file user trying to add exists already, then it shows message of file already exists, and user have to enter correct file name to add it to ‘rootFile’ directory.
2. Option 2 to delete files. As this is case 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.

Screenshots of this application:

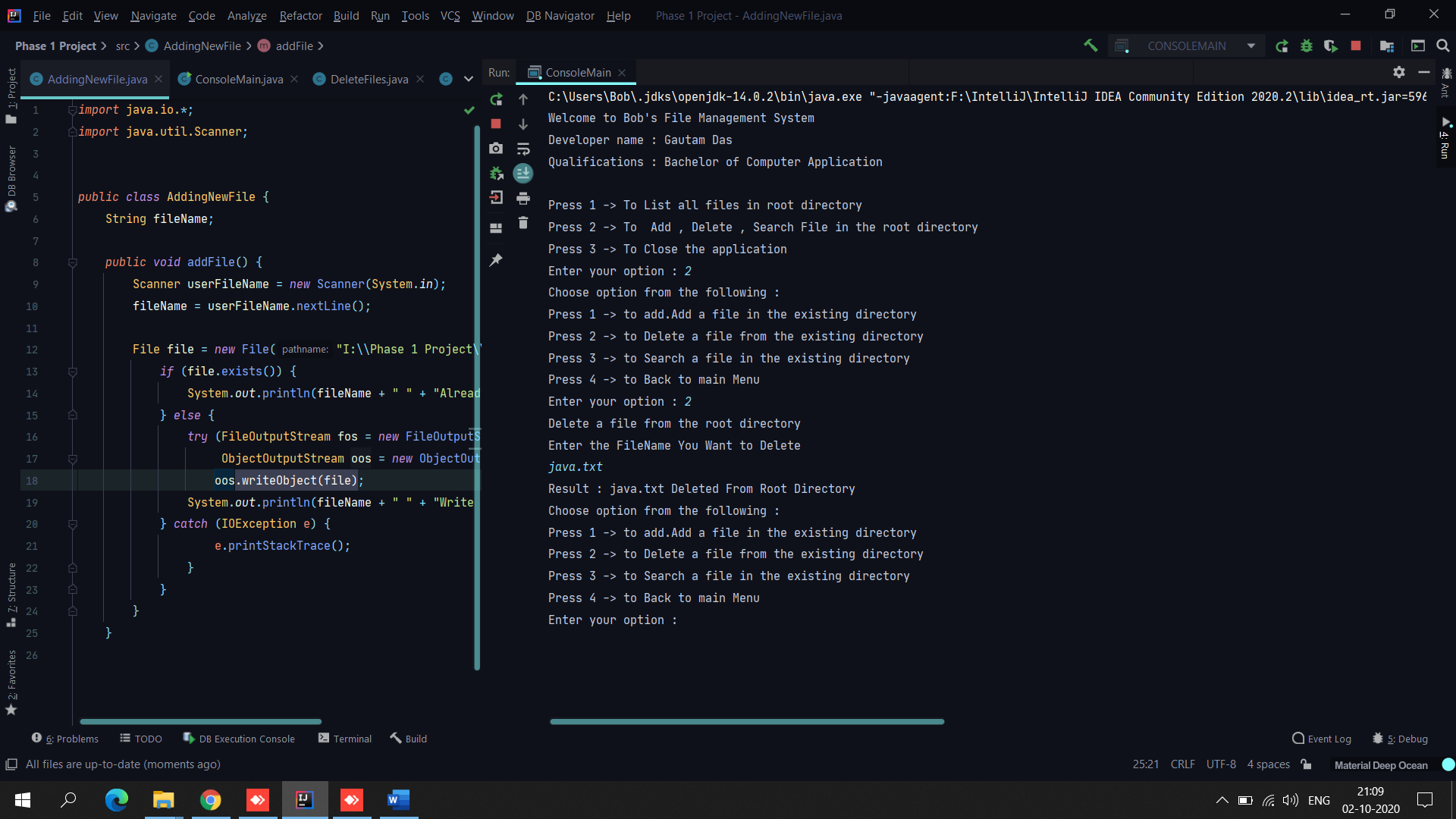
1. Listing all the files that is already exists in the root directory.



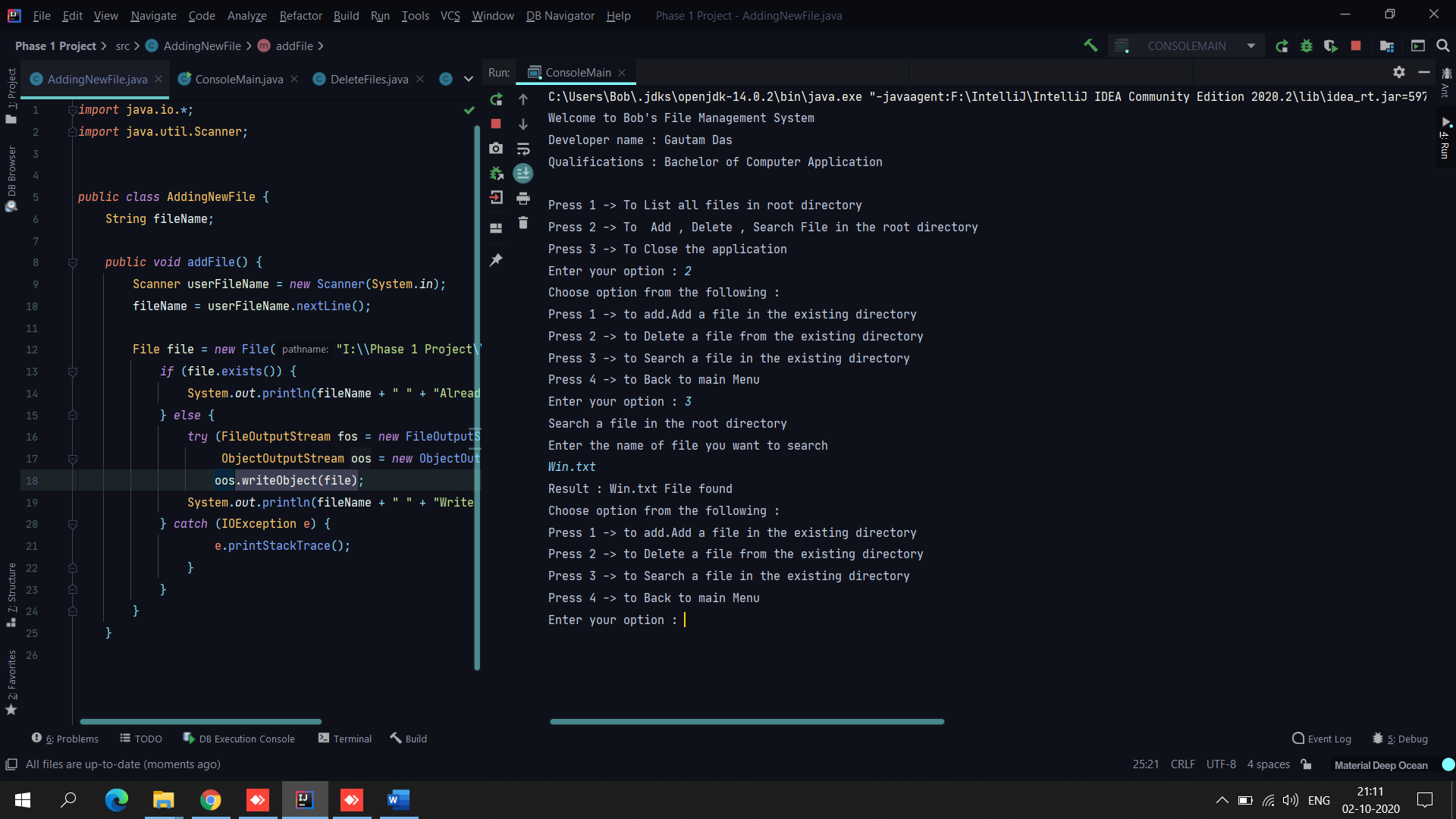
1. Adding a new file in the root directory.



1. Deleting a file from root directory.



1. Searching a file in the root directory.



Conclusion:

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 is present in the root directory.