**Virtual Key for Your Repositories**

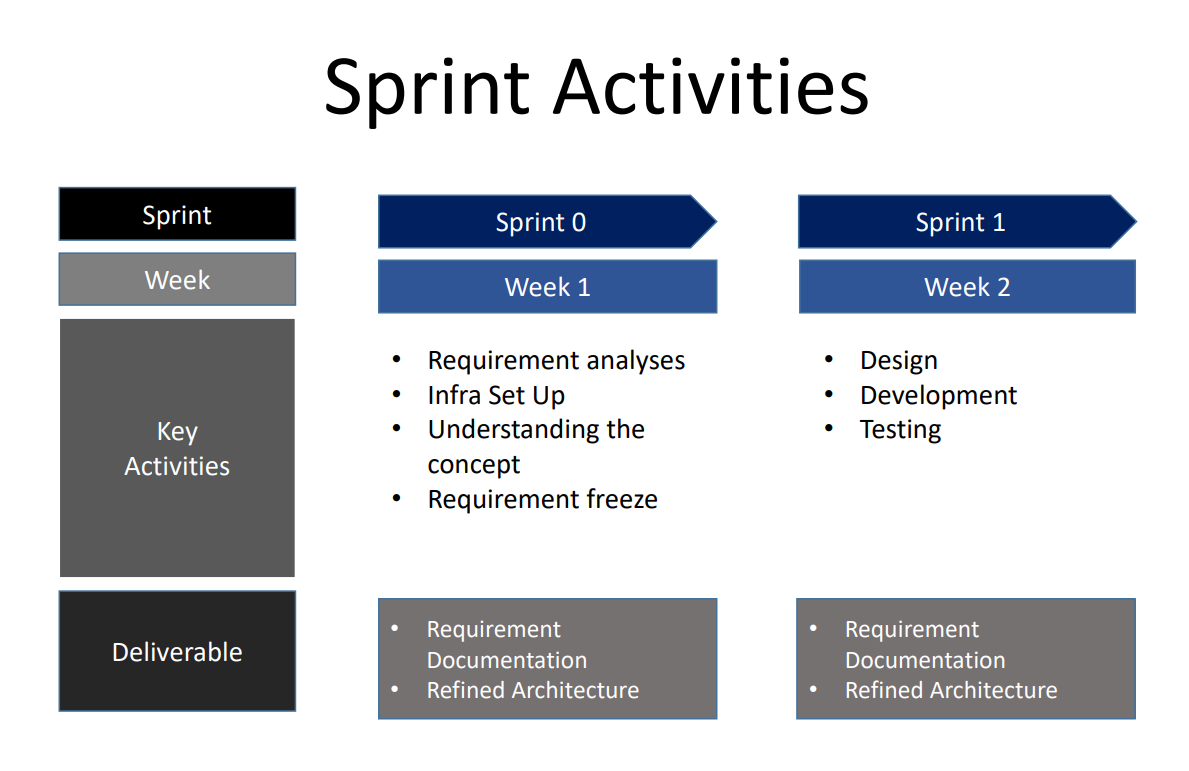
**Lockers Pvt. Ltd**

**Abstract:**

* Sprint
* Program insights
* Flow and Feature of the Application
* Required tools
* Algorithm
* Flowchart
* Program overview
* Screenshot of the Classes
* Screenshot of the different output
* Conclusion
* GitHub link for all the documents and source code

**Sprint:**

Time taken for the development for the application



**Program insights:**

* Duration of sprint required
* Setting GitHub and store the required code and documents in it.
* Installation of eclipse toolkit to develop the application
* Application developed in Java programming Language
* Java concepts used to develop the application
* Control statement like switch, while and if statements are used.
* To handle errors through exception
* Business level operation like create user, add files, search file, search user
* Check the duplicate user and files search
* Visa-versa between main menu and second menu
* Test the application
* Close the application without any unexpected termination

**Flow and Feature of the Application:**

* Planmore than two sprints to complete the application
* Document the flow of the application and prepare a flow chart
* List the core concepts and algorithms being used to complete this application
* Code to display the welcome screen. It should display:
  + Application name and the developer details
  + The details of the user interface such as options displaying the user interaction information
  + Features to accept the user input to select one of the options listed
* The first option should return the current file names in ascending order. The root directory can be either empty or contain few files or folders in it
* The second option should return the details of the user interface such as options displaying the following:
  + Add a file to the existing directory list
    - You can ignore the case sensitivity of the file names
  + Delete a user specified file from the existing directory list
    - You can add the case sensitivity on the file name in order to ensure that the right file is deleted from the directory list
    - Return a message if FNF (File not found)
  + Search a user specified file from the main directory
    - You can add the case sensitivity on the file name to retrieve the correct file
    - Display the result upon successful operation
    - Display the result upon unsuccessful operation
  + Option to navigate back to the main context
* There should be a third option to close the application
* Implement the appropriate concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

**Required Tools:**

* Eclipse IDE
* GitHub account
* Microsoft office to prepare documents

**Algorithm:**

Step 1: Start

Step 2: Display the message welcome and developer name.

Step 3: Display the Main menu options like 1 for list the all-file names, 2 for more options and 3 for exit from the application.

Step 4: Check the user are created and file creation

Step 5: Deleted the user defined file

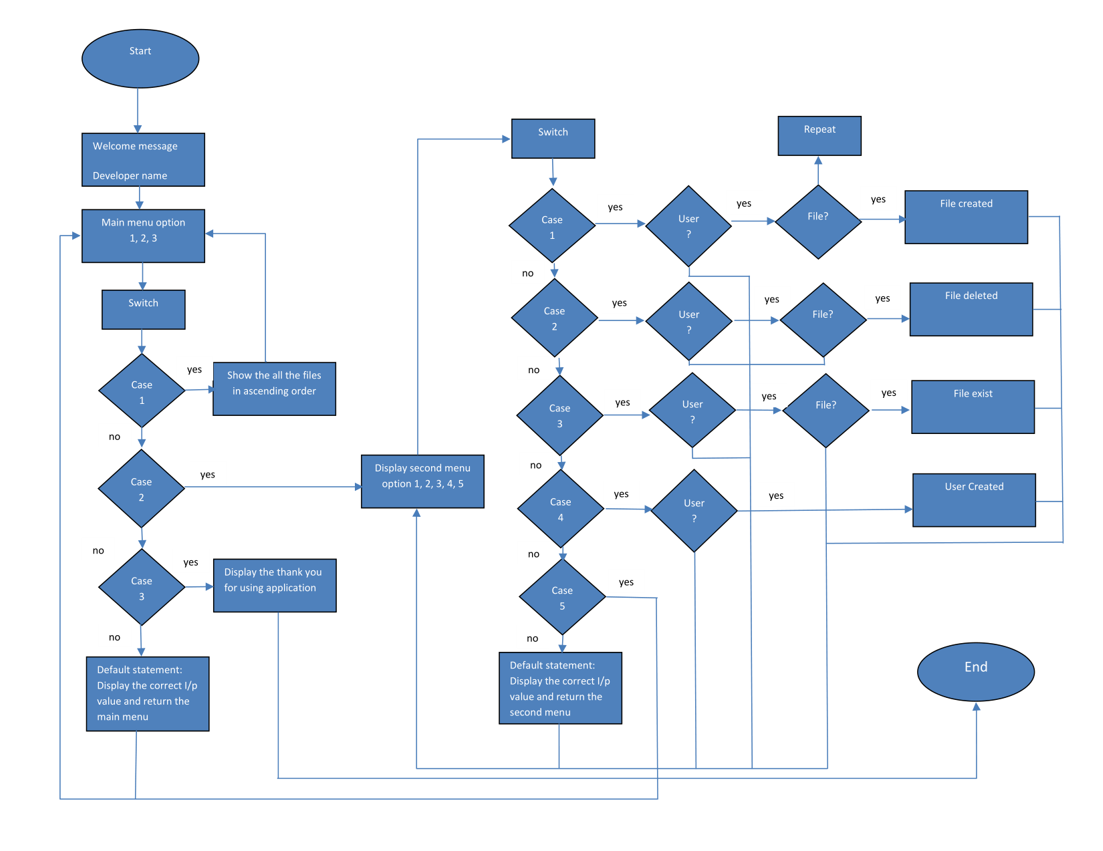
Step 6: Check the user required file

Step 7: Creating User

Step 8: Return to main menu

Step 9: Exit

**Flowchart:**

****

**Program overview:**

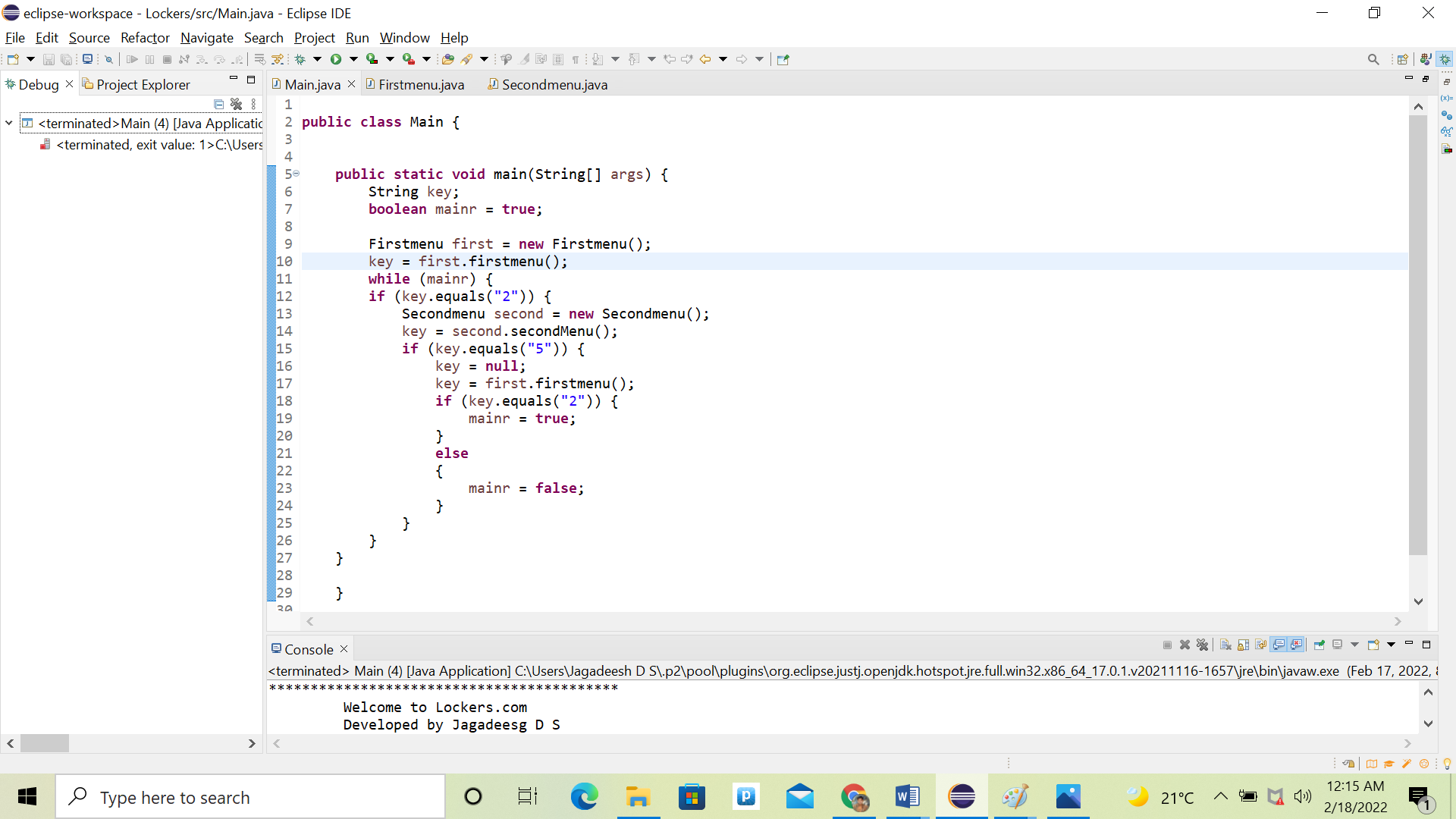
Display the Main menu options like 1 for list the all-file names, 2 for more options and 3 for exit from the application. Check the user are created or not if the user id is not created retune to the second main menu. And if user is created save the user defined file to the folder when saving check the file is already exist or not if file already exist show the message already exist and wait for the new file name to enter.

Next user can delete their file using the file name. Before deleting check, the user’s name is there or not. If the user is there asking for the file name to delete and if the file doesn’t exist return to the second main menu. If the user not there then return to the second main menu. Check the user defined file is available or not in the user created folder. If the file exists or not show the message depending on the output i.e., file exist or file is doesn’t exist. Here also user is not available return to the second main menu.

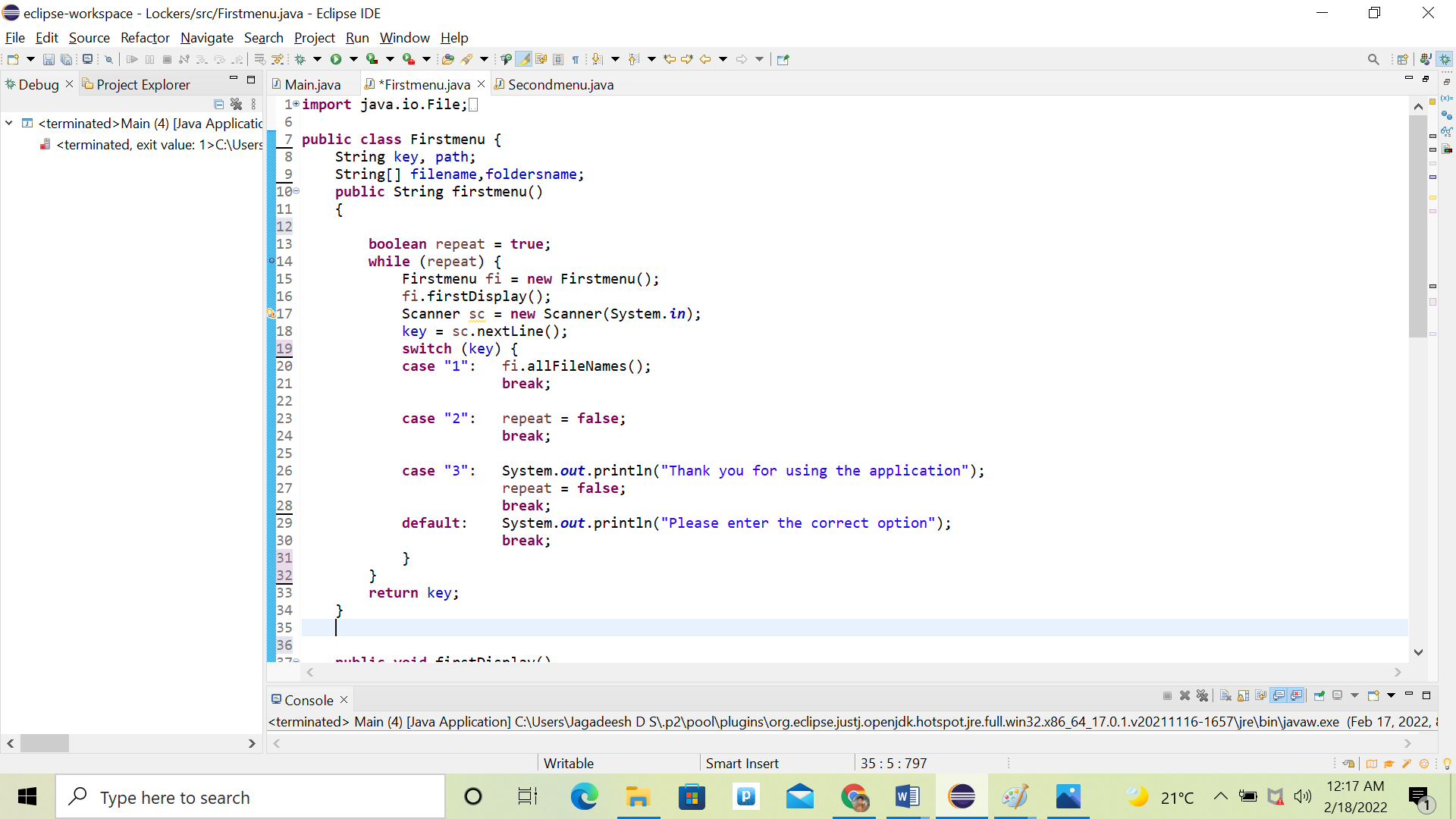
Creating the user for the save their file in the application. When creating the user check user exist or not if user exist show the already exist and if not create the user and return to main menu option and check the file are available in the all-file list and close the application without any uninterrupted termination.

**Screenshot of created class in the source code:**

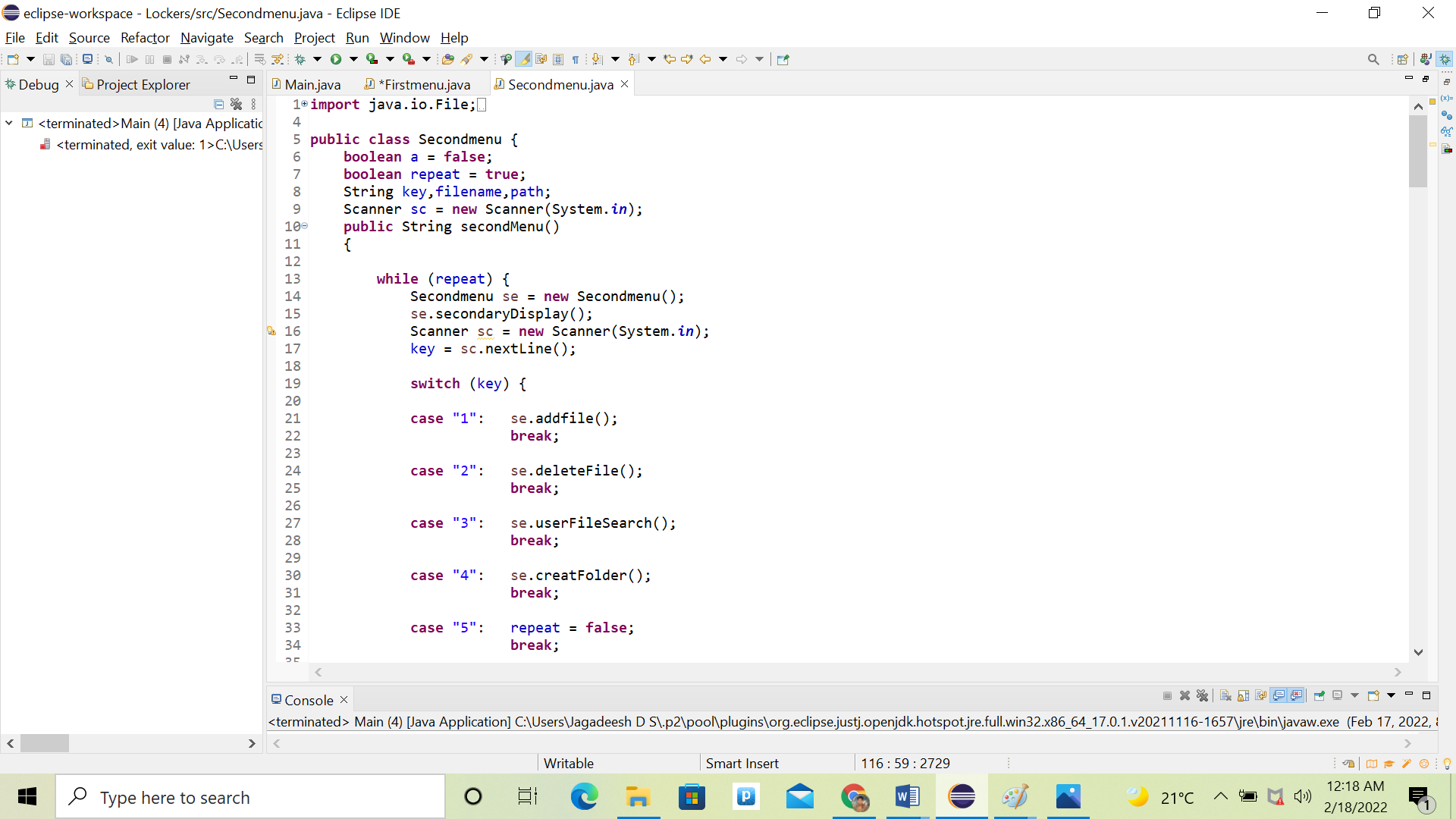
Menu Class:



Firstmenu class:

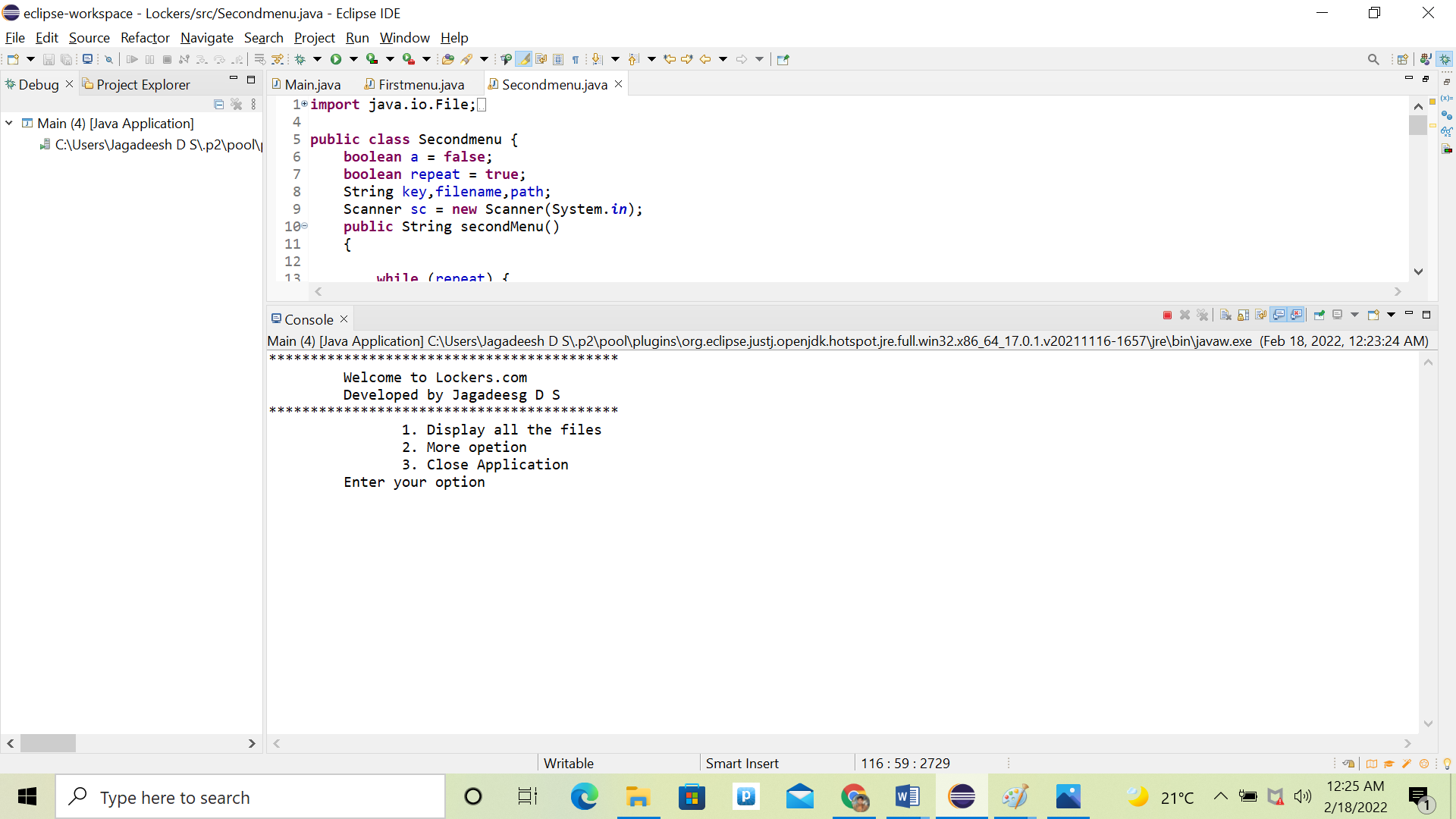


Secondmenu Class:

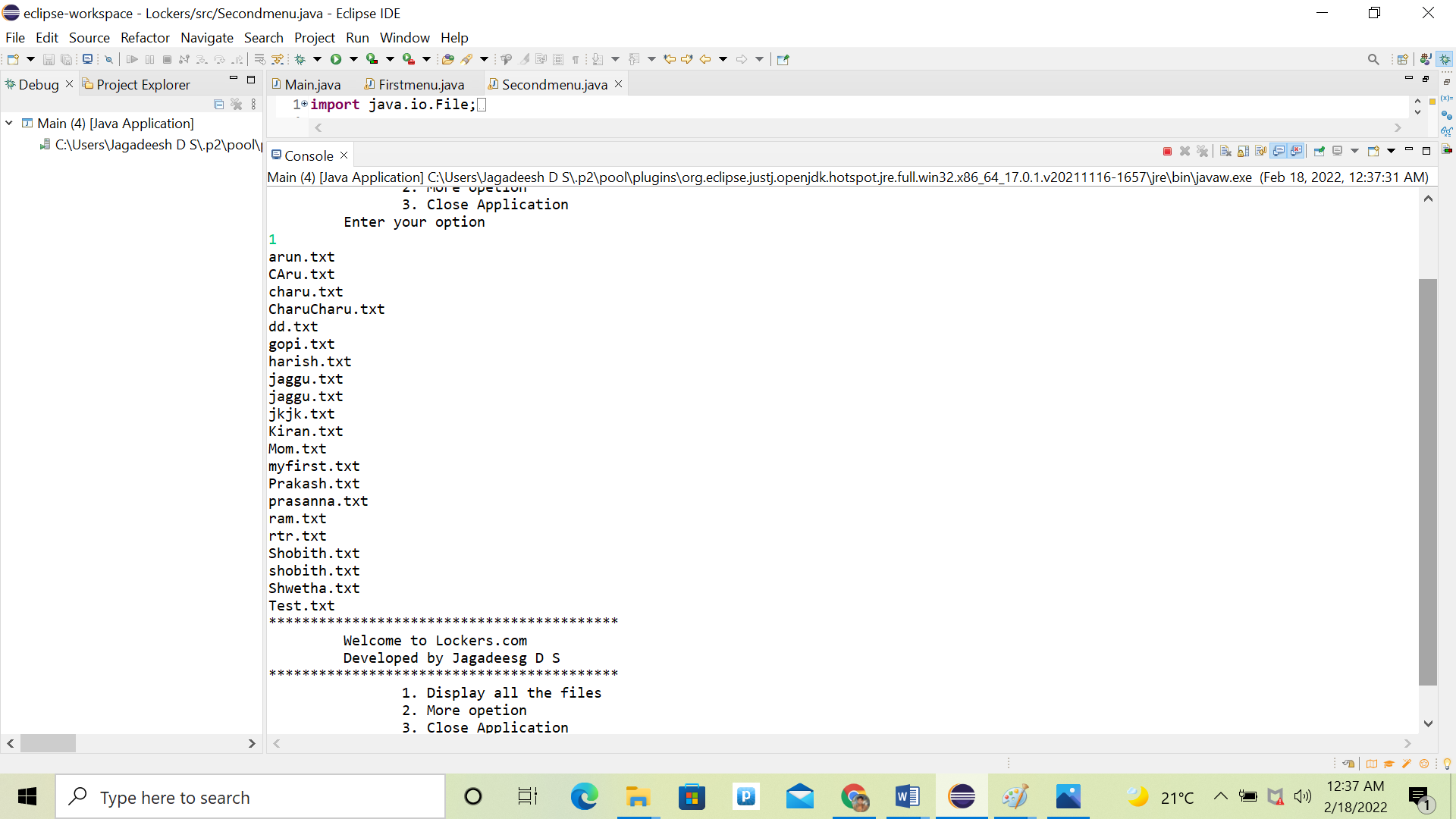


**Screenshot different output:**

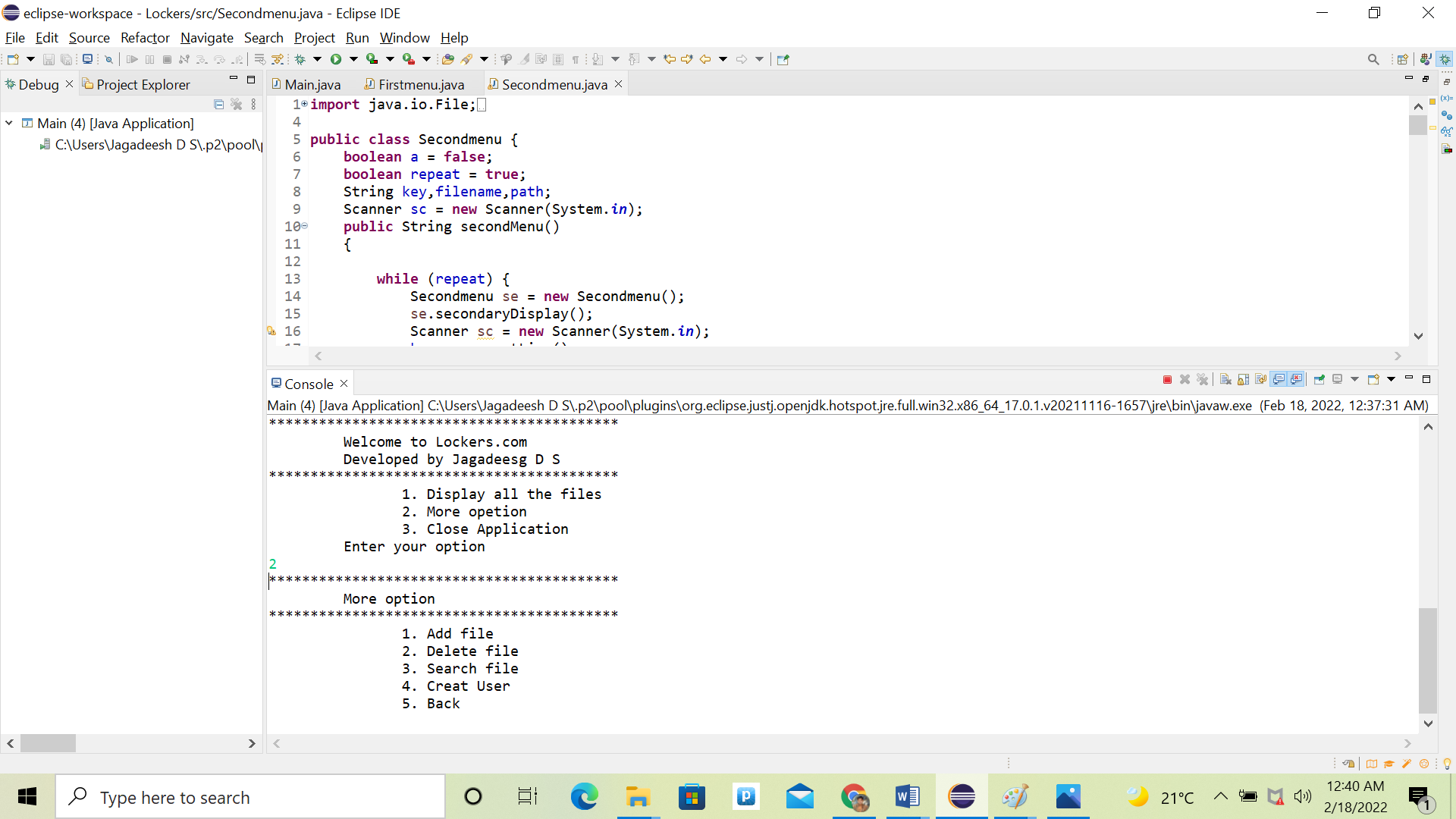
Welcome display and main menu options



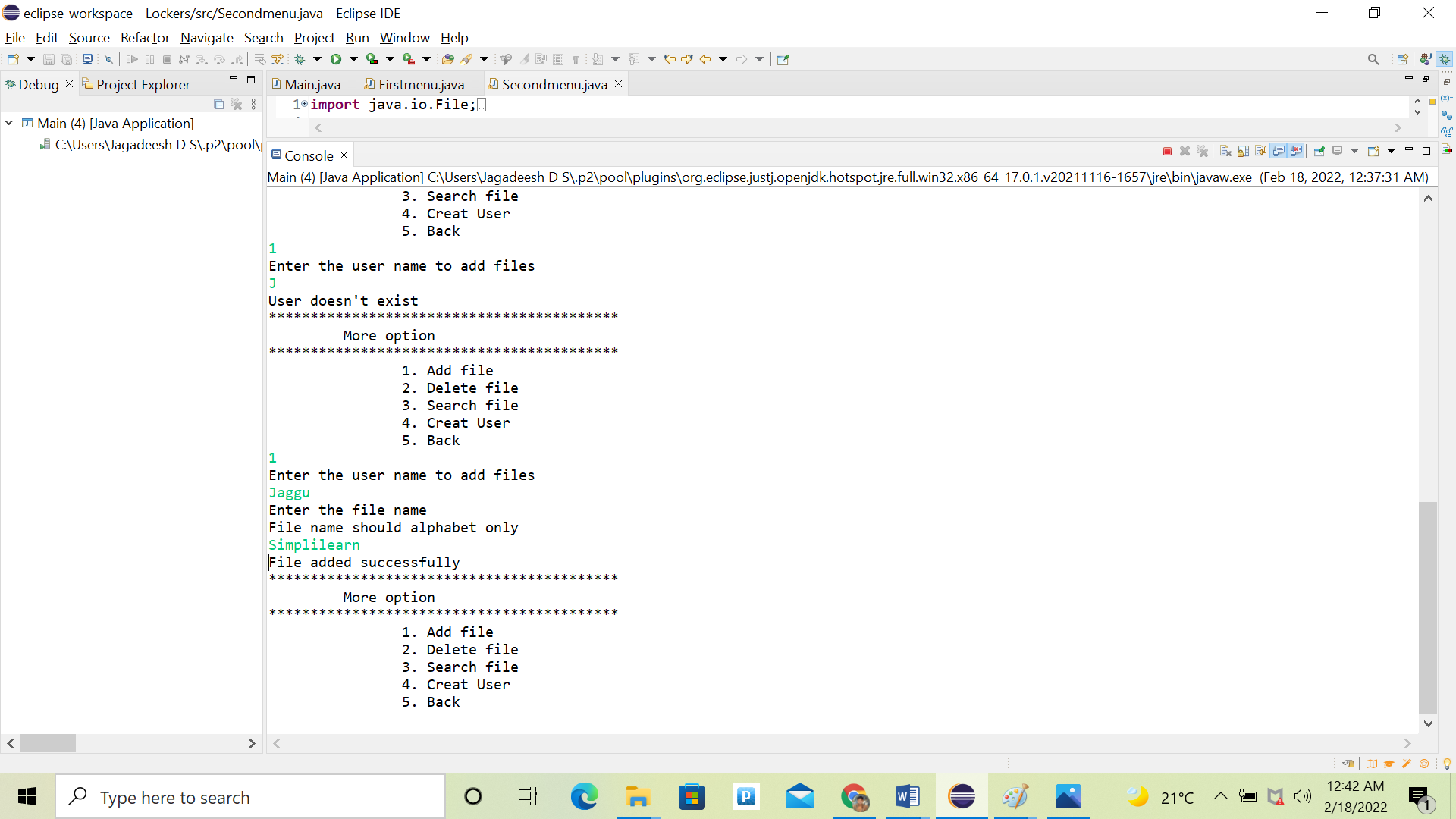
Sorted files list:



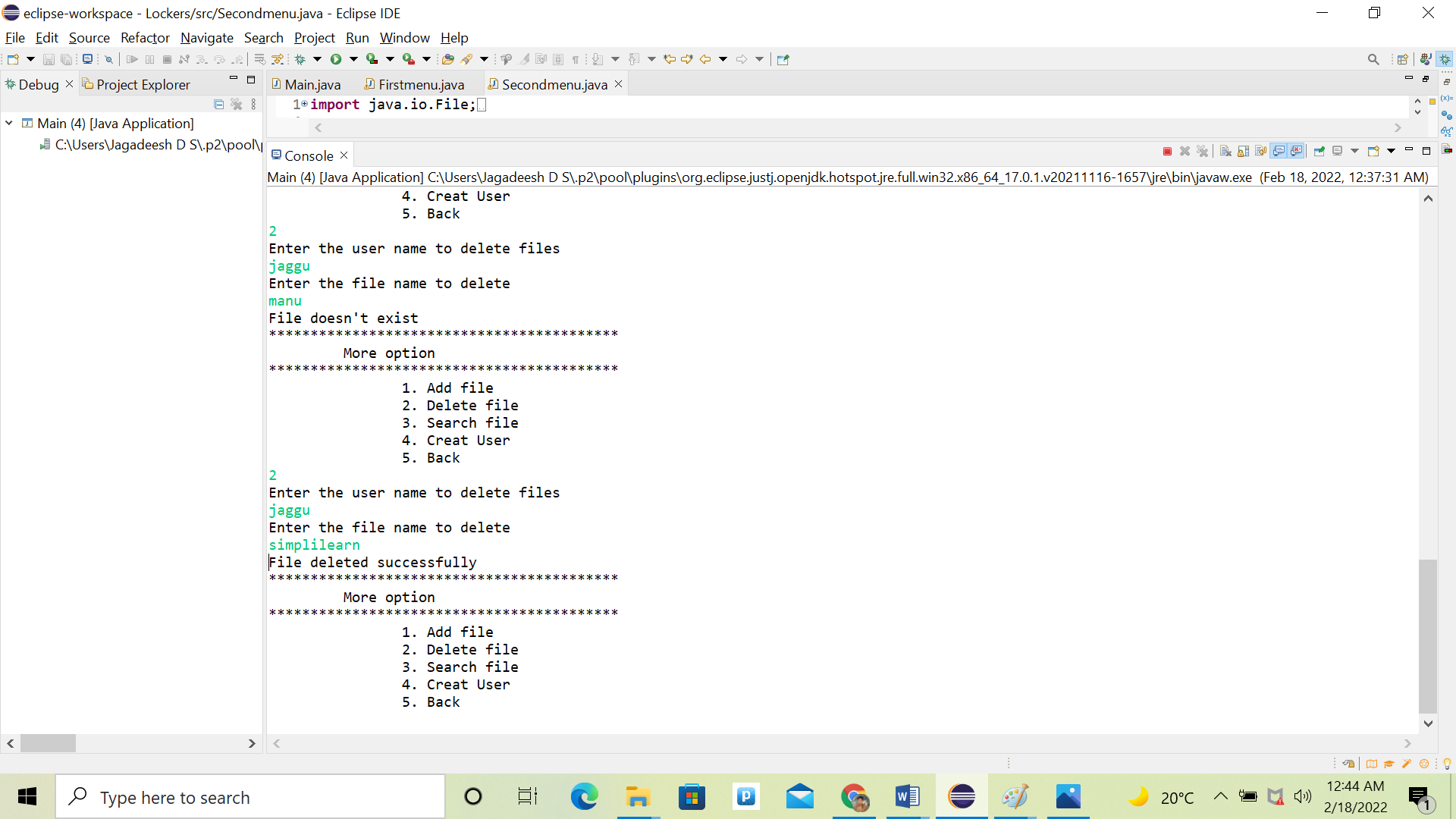
Second menu option:



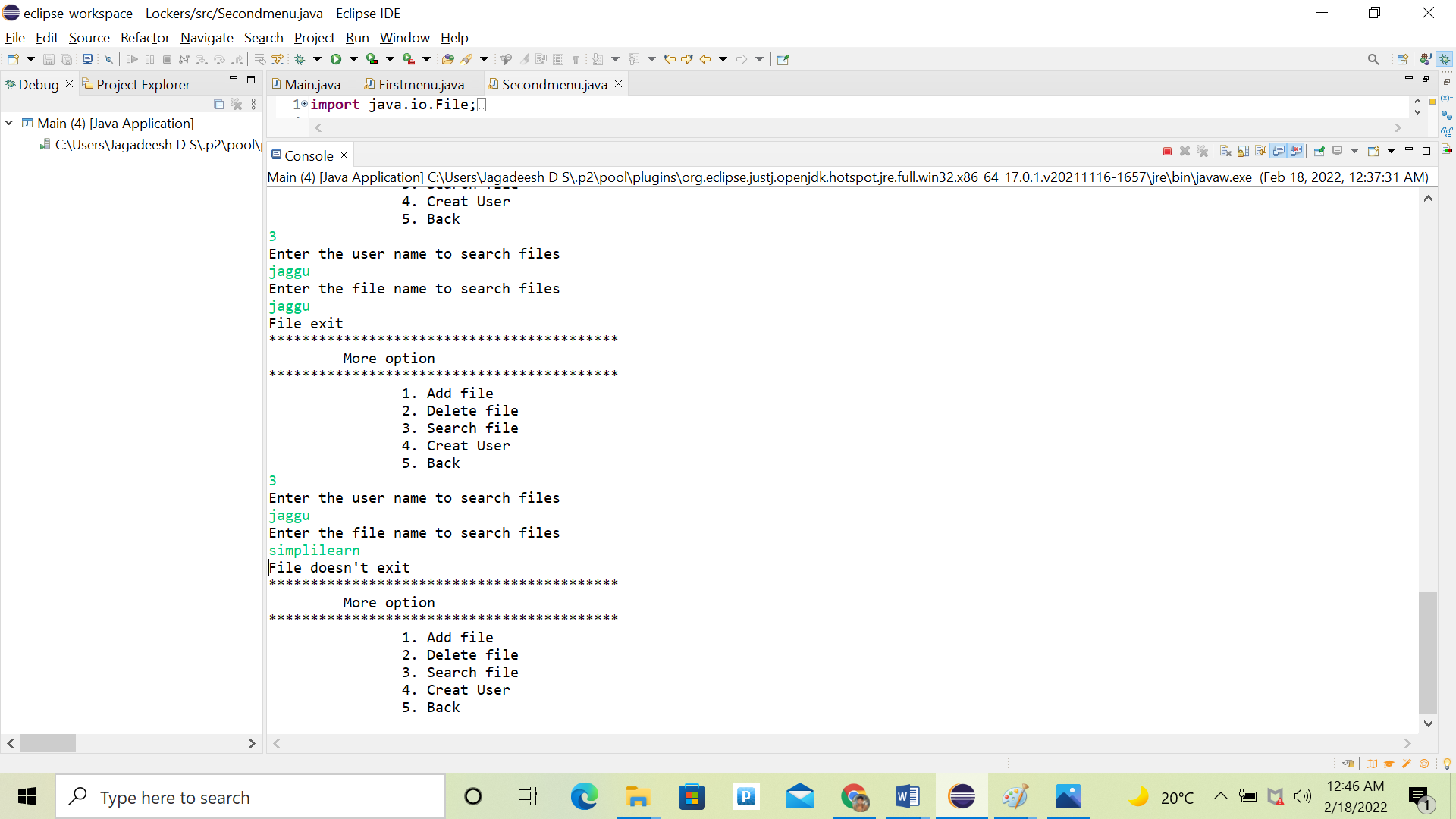
Add file option:



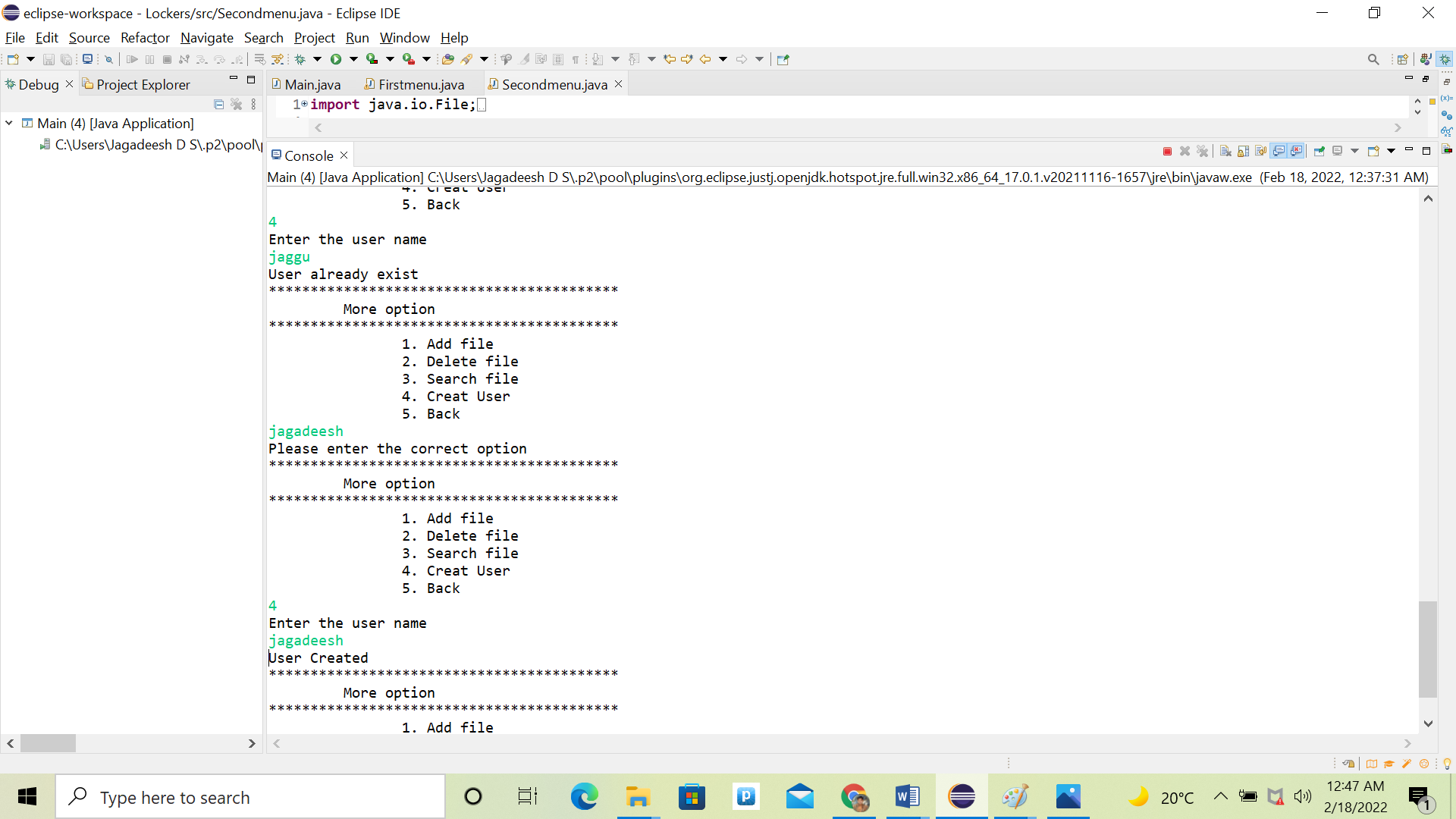
Delete file:



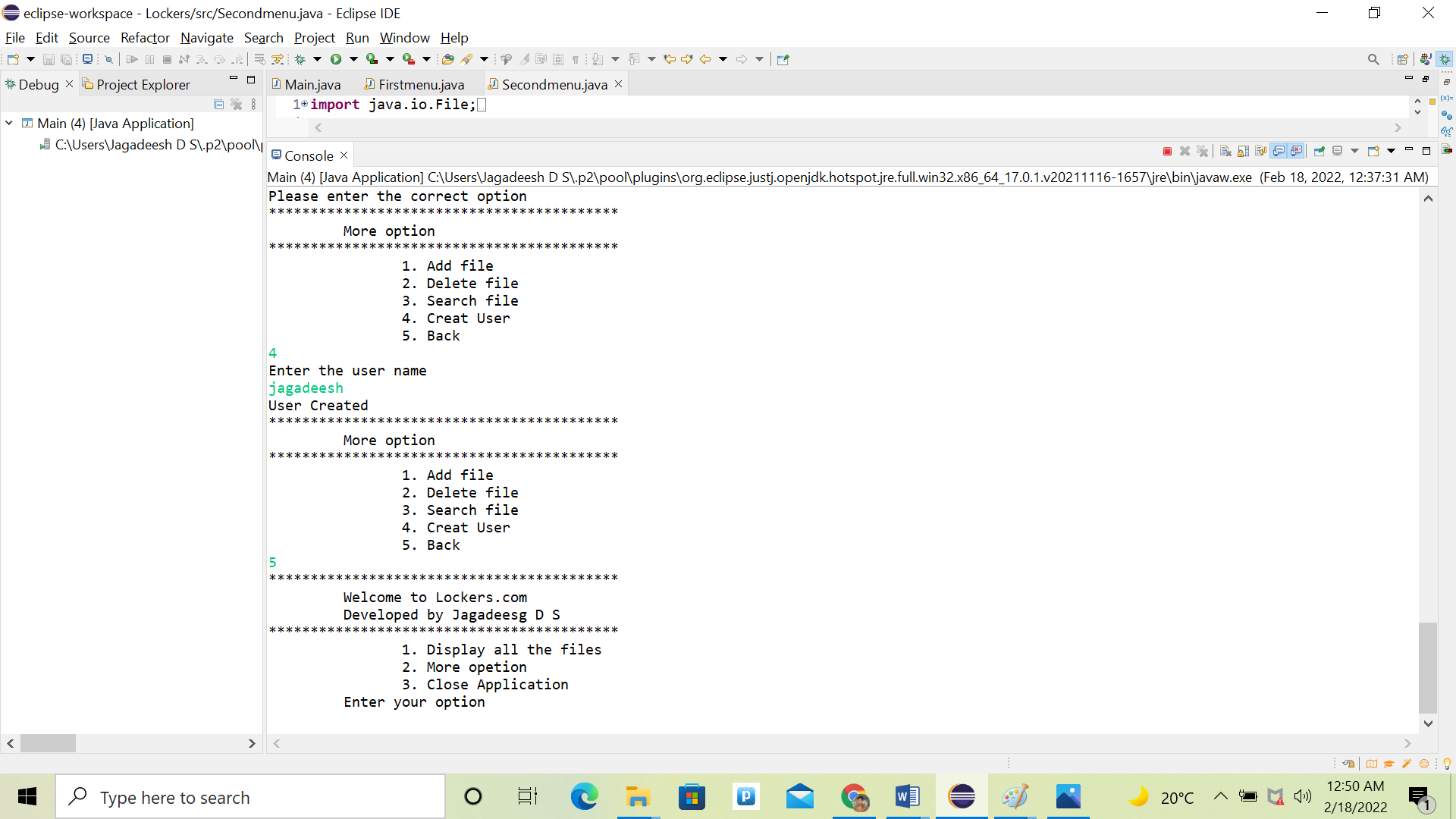
Search file:



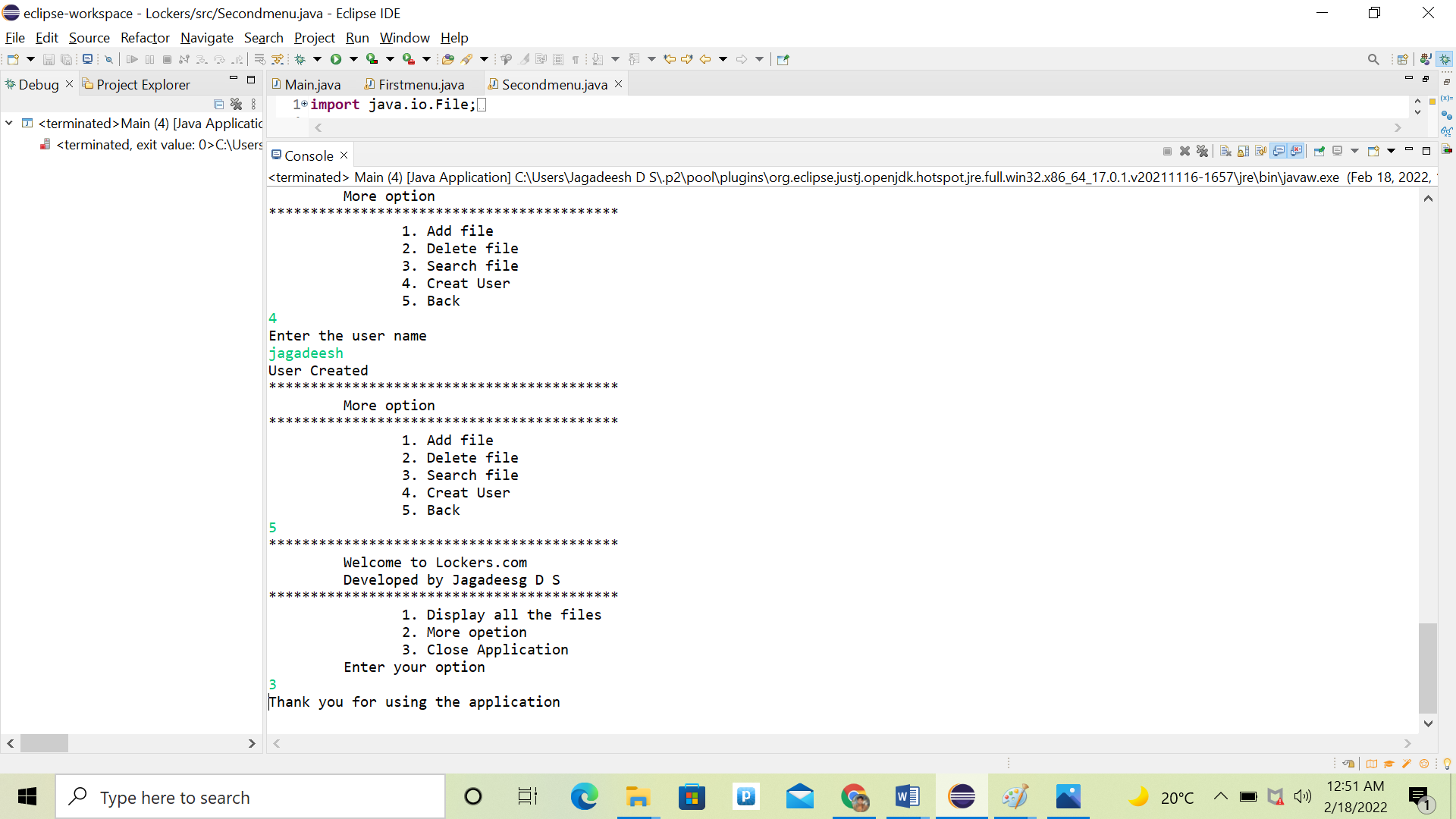
Create user:



Back to main menu



Exit application:



**Conclusion:**

Further enhancement can be done as per the user requirement

**GitHub link for all the documents and source code:**