**LOCKED ME APPLICATION**

**PHASE 1**

**ASSESSMENT PROJECT**

**PROJECT AND DEVELOPER DETAILS**

Locked Me – Application

Developed by Kanimozhi Mahendran

Locked me application is developed by using file management system, which creates a root directory by creating a folder in the file application. It gives user the capability to list files/folder present in this root directory as well as add and remove files from this root directory. It also gives the user a capability to search all the files with a particular name in the directory.

<https://github.com/Kanimozhi26/Phase1-Assessment-Project>

**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.

**BACKGROUND OF THE PROBLEM**

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):

**SPRINTS PLANNED AND THE TASKS ACHIEVED IN THEM**

Number of sprints planned: 3

**Sprint 1:**

* Analysing the required things needed to develop the application and creating the flow of the application.
* Designing the structure of the application and proper planning should be done.
* Creating the flow of the application and developing the program for:
* Sorting all the files in ascending order
* Business level operations Menu
* Option to exit the Application

**Sprint 2:**

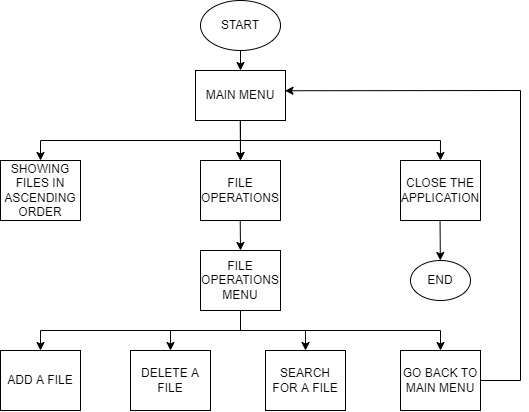
* Developing the Important Business level Operations
* Adding a File
* Deleting a File
* Searching for a file
* Option to go back to the Main Menu

**Sprint 3:**

* Documenting the steps involved in creating the application.
* Fixing the bugs that is debugging should be done.
* Testing the application with different types of user input.
* Pushing the code to GitHub repositories.

**FLOWCHART OF THE APPLICATION**

Flowchart - [Link](https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=DIA.drawio#R5Vtbk6o4EP41Pu4UEC7yyAhntNZRS9w6O4%2BU5CinkFgYR91fv4kEgcQL4yAwc3zQpEma5Ouv03SCHdBb7V9ib718RT4MO4rk7zvA7iiKIcvkmwoOiUBVtESwiAM%2FEcmZwA3%2Bg0woMek28OGm0BAjFOJgXRTOURTBOS7IvDhGu2KzXygs3nXtLaAgcOdeKEp%2FBj5eJtKuJmXyPgwWy%2FTOssSurLy0MRNslp6PdjkRcDqgFyOEk9Jq34MhxS7FJen348LV08BiGOEyHbovu1%2BGvosO7qv13rd%2B24as%2FsW0vHvhlk3YnVnTGRsxPqQwEFUEcVJ53i0DDN21N6dXdsTmRLbEq5DUZFJk%2BmCM4f7iQOXT9AltIFpBHB9IE9YB6AyxXQ5wJlrmsE5t4DETL06aMhRIgQHxEVA0ARUREJ%2FQhFVRjJdogSIvdDLpc4y2kQ%2FpfSRSy9oMEVozsH5DjA%2BM894Wo3NQ0htdB5KMC23jObwyIzZ%2B7MULiK%2B0U88bJoahh4P34jgqh10RUH%2B1BiMieXVG%2FwgGKMJbAys1xgq2kBkiSc06SQpEz%2B2Pfw5GL0T4YzB0XPJ7RM9ye87ITi6Mp7YzbRxKOXXwA1e%2F4fD6o7BUBSwpghSviTO1ZoPxyG0cNNA20MRFsjccuxS1WZ9%2BW5PJcNA7otc4eFrbwNMF8CzbpqAx720cMd5HgdYwYoaAmO0MnZnTJtBOEaAtoHUF0F7GpP5s9f6mfkrLbYqxvJs2jp9ZJjC0Azs%2BPihNY5dmcPknFMea9vrUW8fTNvntKSltC%2B9kMS8TQIKRb9EEl9TmobfZBPMiLkUQ4T7A%2F9Lyk8Zqb6wdLdv7XDP7kFYiMpdcJ1p9y1%2FLuh1rab9OdYlLmvLfSlyU8%2FbN2U87Y79UVjq%2FYXeYoIBMJHO9Lud6gONFMk3WK5%2Bic4pUwD2mdDlFCQ6CoiPHTtP%2BBO3EpOJb0u7ePLgmOpnmk6EUiaDcyahzutSaSSU%2B6zZAqrLkaJHRlXuNrkoXHgXqsrj4rH7N4hGK4DVzt910Cgc3uHf5128perTdxHThT7KbsDDea7faV1gxTfnsCvvgENyUyW9aqqzJNbJcm3JRF58NPNjqaXT4OlYv8eAFvsmCQNghmdmnqNaomSjVJ5PNE0VvE1HujvinY%2B6mqKF8Q2p02xRhKqMGqDu8VL8X0Dw1jCapAUBV1FA4RXWvGuJ56denhiw1yQ1V1yriBk%2BymrcQlTLvzpTPIS8RQ66PGGajvKgqnKj88VDd4eRju4BffW9B795ww9KpBKfIrNlsH9vK%2B9hSb5pqYbF%2FUiXthl8faxMYB2RiMK7e2ZU2kYa3dekYoHLkK0kaYkXvkGu2pg02D6FVtTuN6YFAegTwlgWMTx8IlDikbDbh4AK%2Bevfu860l68FLTTruHCeckS3SosY3kTXjQtDMmVY%2FY1p%2B36i6lz7%2FiPcCmnIluqmradLpU9zfvfs4jqrtgvNE%2BrRvkWr2F4KkefY%2FDOD8Dw%3D%3D)

****

**CORE CONCEPTS USED IN THE PROJECT**

* Class, Objects
* Methods, Interfaces
* Arrays, String Arrays
* Collections Functions
* File Handling Methods
* Exception Handling Methods
* For Loop, While Loop, If Loop, If Else Loop, Switch Case Statements
* Constructors, Scanner Classes

**LINK TO THE GITHUB REPOSITORY TO VERIFY THE PROJECT COMPLETION**

<https://github.com/Kanimozhi26/Phase1-Assessment-Project>

**SOURCE CODE**

**package** phase1\_assessment;

**import** java.util.Scanner;

**import** java.io.File;

**import** java.util.Arrays;

**import** java.io.IOException;

**public** **class** locked\_me

{

**public** **static** **void** main(String[] args)

{

System.***out***.println("\n--------------------------------\n"+"\n------LOCKEDME APPLICATION------\n"+"\n--------------------------------\n"+"\n----BY-------KANIMOZHI MAHENDRAN\n"+"\n--------------------------------\n");

File new\_folder = **new** File("new\_folder");

new\_folder.mkdir();

*optionSelection*(new\_folder);

}

**private** **static** **void** optionSelection(File new\_folder)

{

String[] array = {"1----->SORT THE FILES IN ASCENDING ORDER\n",

"2----->BUSINESS LEVEL OPERATIONS\n",

"3----->EXIT THE APPLICATION\n" };

**int**[] arr={1,2,3};

**int** len=arr.length;

System.***out***.println();

**for** (**int** i=0;i<len;i++)

{

System.***out***.println(array[i]);

}

System.***out***.println("PLEASE ENTER ANY OPTIONS YOU WOULD LIKE TO DO:");

Scanner scan=**new** Scanner(System.***in***);

**int** options=scan.nextInt();

**switch**(options)

{

**case** 1:

String[] file\_names=**new** String[100];

file\_names=new\_folder.list();

**if**(file\_names.length==0)

{

System.***out***.println("THERE IS NO FILE AND IT IS EMPTY");

}

**else**

{

Arrays.*sort*(file\_names);

System.***out***.println("LIST OF FILES: ");

**for**(String name:file\_names)

{

System.***out***.println(name);

}

}

*optionSelection*(new\_folder);

**break**;

**case** 2:

System.***out***.println("BUSINESS LEVEL OPERATIONS");

*lockedme*(new\_folder);

**break**;

**case** 3:

System.***out***.println("----THANK YOU-------\n "+"---EXITING THE APPLICATION---");

**break**;

**default**:

System.***out***.println("PLEASE ENTER CORRECT option");

*optionSelection*(new\_folder);

**break**;

}

}

**private** **static** **void** lockedme(File new\_folder)

{

System.***out***.println("1--->ADD A FILE\n"

+ "2--->DELETE A FILE\n"

+ "3--->SEARCH FOR A FILE\n"

+ "4--->GO BACK TO MAIN MENU\n");

System.***out***.println("------------PLEASE ENTER YOUR CHOICE------------");

Scanner sc=**new** Scanner(System.***in***);

**int** option=sc.nextInt();

**switch**(option)

{

**case** 1:

System.***out***.println("Adding a file\n");

System.***out***.println("ENTER THE FILE NAME TO BE ADDED");

String newFile=**new** Scanner(System.***in***).nextLine();

File add\_file=**new** File(new\_folder, newFile);

**try**

{

**if**(add\_file.createNewFile())

System.***out***.println("FILE IS ADDED");

**else**

System.***out***.println("FILE NAME ALREADY EXISTS");

}

**catch**(IOException e)

{

System.***out***.println("ERROR :"+e.getMessage());

}

*lockedme*(new\_folder);

**break**;

**case** 2:

System.***out***.println("DELETE A FILE");

System.***out***.println("ENTER THE NAME OF THE FILE TO BE DELETED:");

String file\_name=**new** Scanner(System.***in***).nextLine();

File delete\_file=**new** File(new\_folder, file\_name);

**if**(delete\_file.exists())

{

**if**(delete\_file.delete())

{

System.***out***.println("FILE IS DELETED");

}

**else**

System.***out***.println("CAN'T DELETE THE FILE");

}

**else**

System.***out***.println("THE FILE IS NOT FOUND");

*lockedme*(new\_folder);

**break**;

**case** 3:

System.***out***.println("SEARCH FOR A FILE");

System.***out***.println("ENTER THE FILE TO BE SEARCHED");

String s\_file=**new** Scanner(System.***in***).nextLine();

File search\_file=**new** File(new\_folder, s\_file);

**if**(search\_file.exists())

{

System.***out***.println("FILE IS PRESENT");

}

**else**

System.***out***.println("FILE IS NOT PRESENT");

*lockedme*(new\_folder);

**break**;

**case** 4:

System.***out***.println("GO BACK TO MAIN MENU");

*optionSelection*(new\_folder);

**break**;

**default**:

System.***out***.println("PLEASE ENTER CORRECT OPTION");

*lockedme*(new\_folder);

**break**;

}

}

}

**CREATING PROJECT IN ECLIPSE**

* Open Eclipse
* Go to File -> New -> Project -> Java Project -> Next.
* Type project name phase1 assessment project and click on “Finish.”
* Select your project and go to File -> New -> Class.
* Enter locked\_me in any class name, check the checkbox “public static
* void main(String[] args)”, and click on “Finish.”

**PUSHING TO GIT REPOSITORIES**

* Open git bash in the project folder in local path
* Create a new repository named Phase1-Assessment-Project in Git.
* Execute the following commands
* git init
* git add .
* git commit -m ”first commit”
* git remote add origin <repository link>
* git push origin master

Project folders are added to the git repository successfully.

**STEPS**

Creating a folder New folder in the project folder where the files are to be stored using mk dir command.

Creating a string array to store the options to be entered by the user.

**1--- Sort the files in ascending order**

Sorting the files using sort() method.

**2---Opening the Menu options**

Inserting the menu operations

1. Add a file
2. Delete a file
3. Search for a file
4. Go back to the main menu

**3---Exit the Application**

**2-i Add a File**

Adding a new file using file operations and if file name already exists, it will through a message that file already exists.

On successful creation of the file, it will create a file and through a message that File is created.

These files are stored in the New Folder that was created in the project folder

**2-ii Deleting a File**

User has the enter the file name to be deleted. If the file found in the folder, that will be deleted successfully and through a message that the file is deleted.

If the files don’t exist, it will through the message that entered file is not found.

If any error happens during deletion of file, it will through a exception that can’t delete this file.

**2-iii Searching for a file**

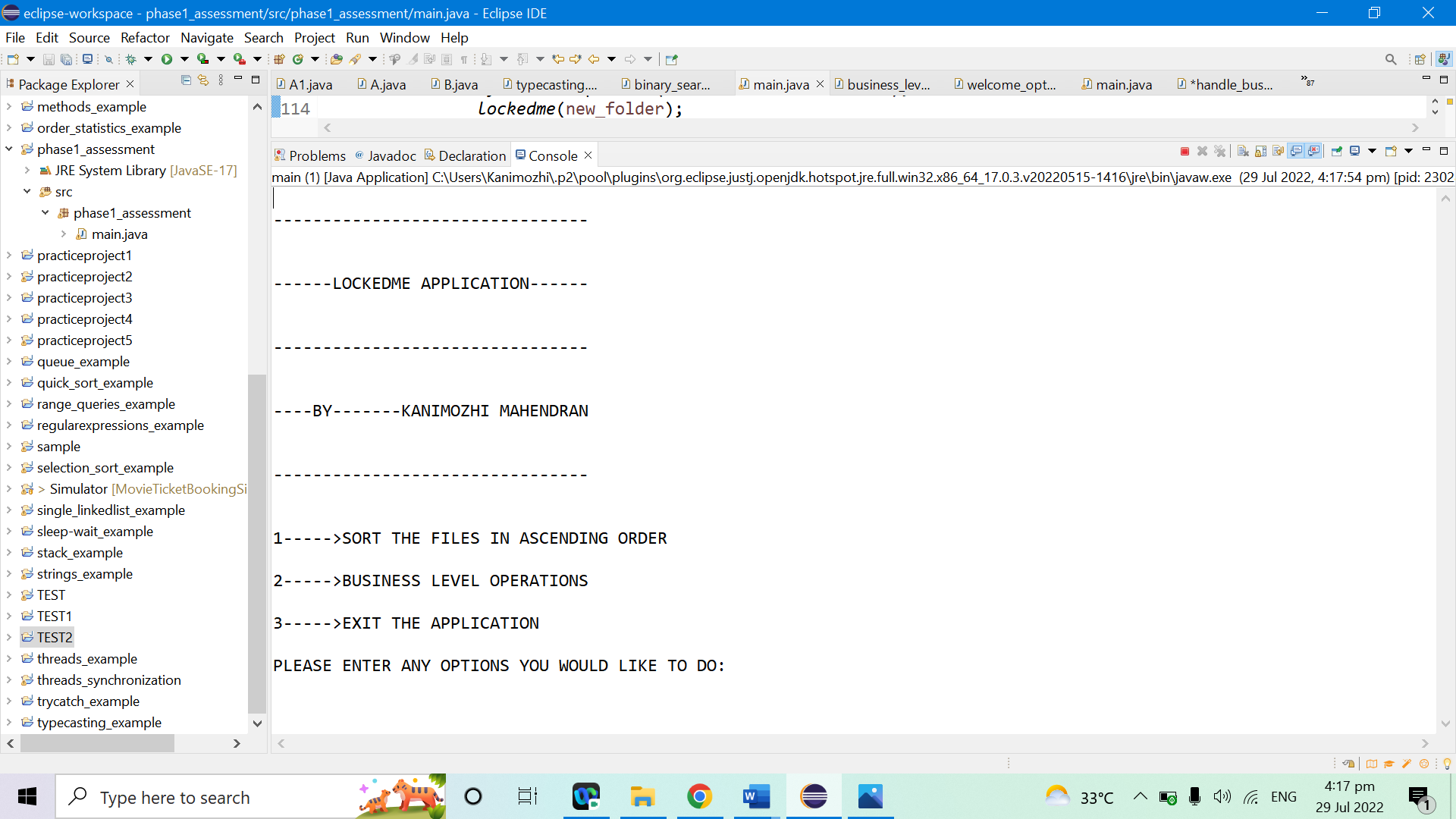
User has to enter the file to be searched.

If the file in the directory, it will return a message that the file is present, else it will return file is absent.

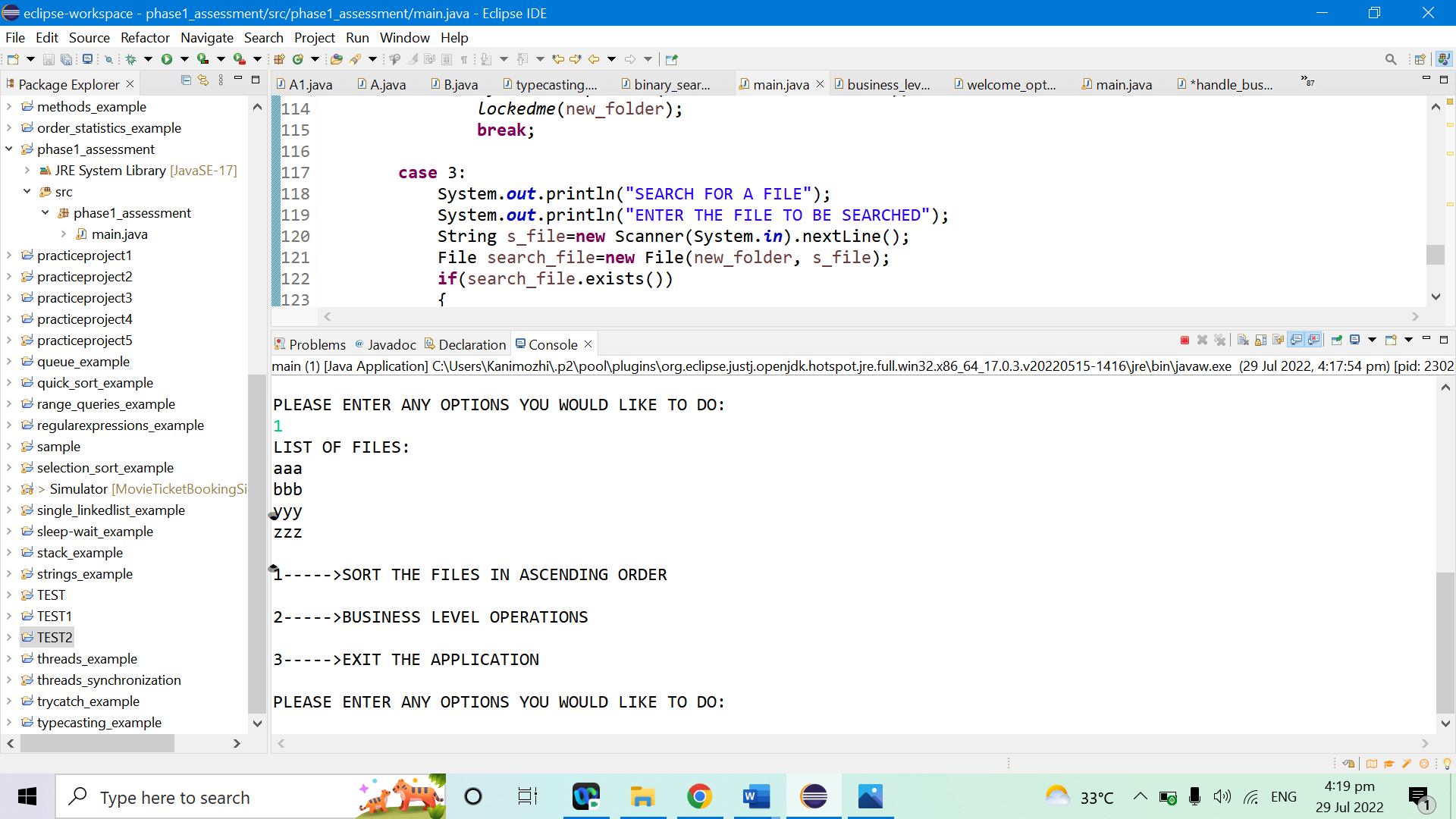
**2-iv Go back**

Going back to the main menu

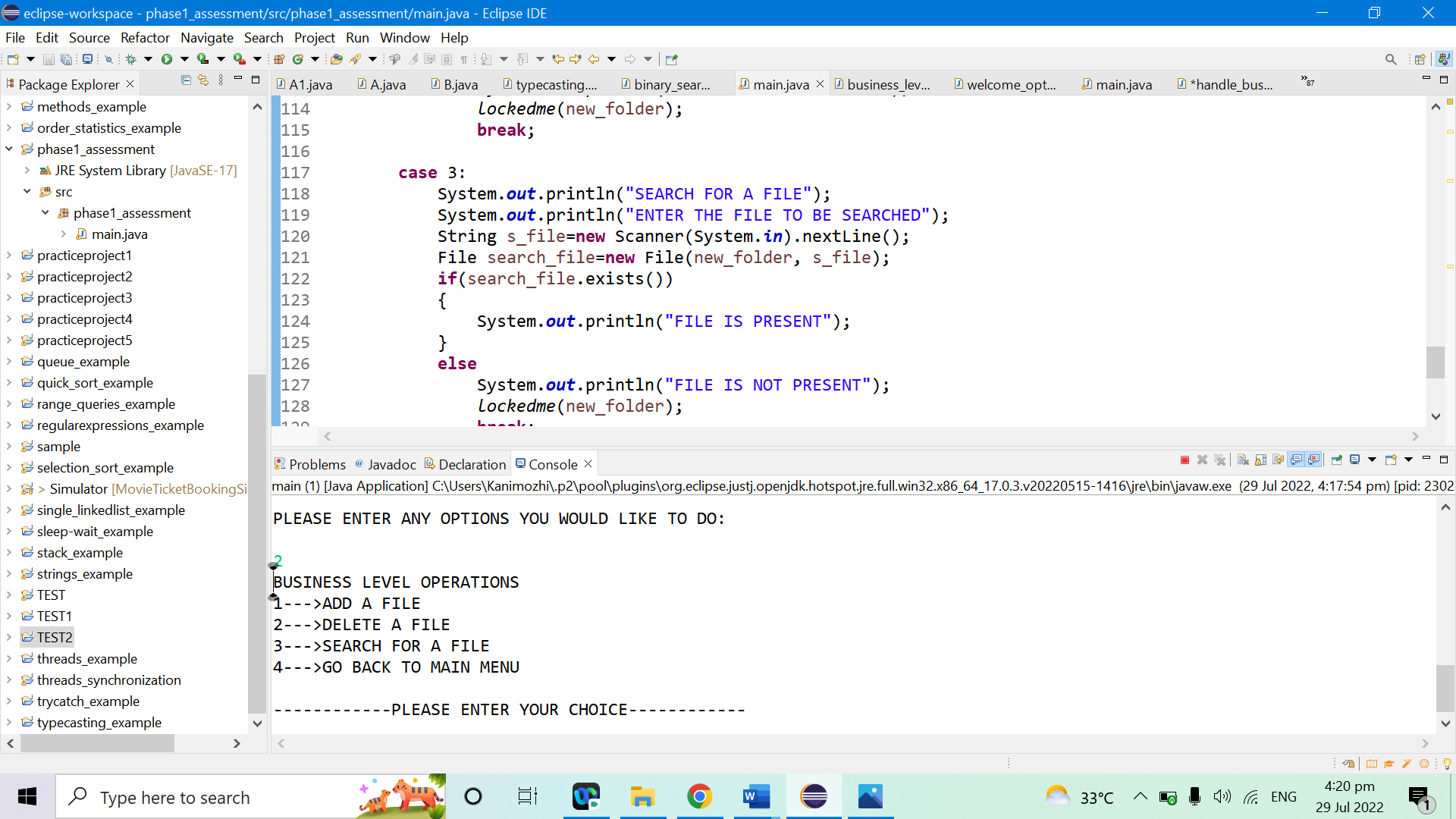
**SCREENSHOTS**

INITIAL OUTPUT

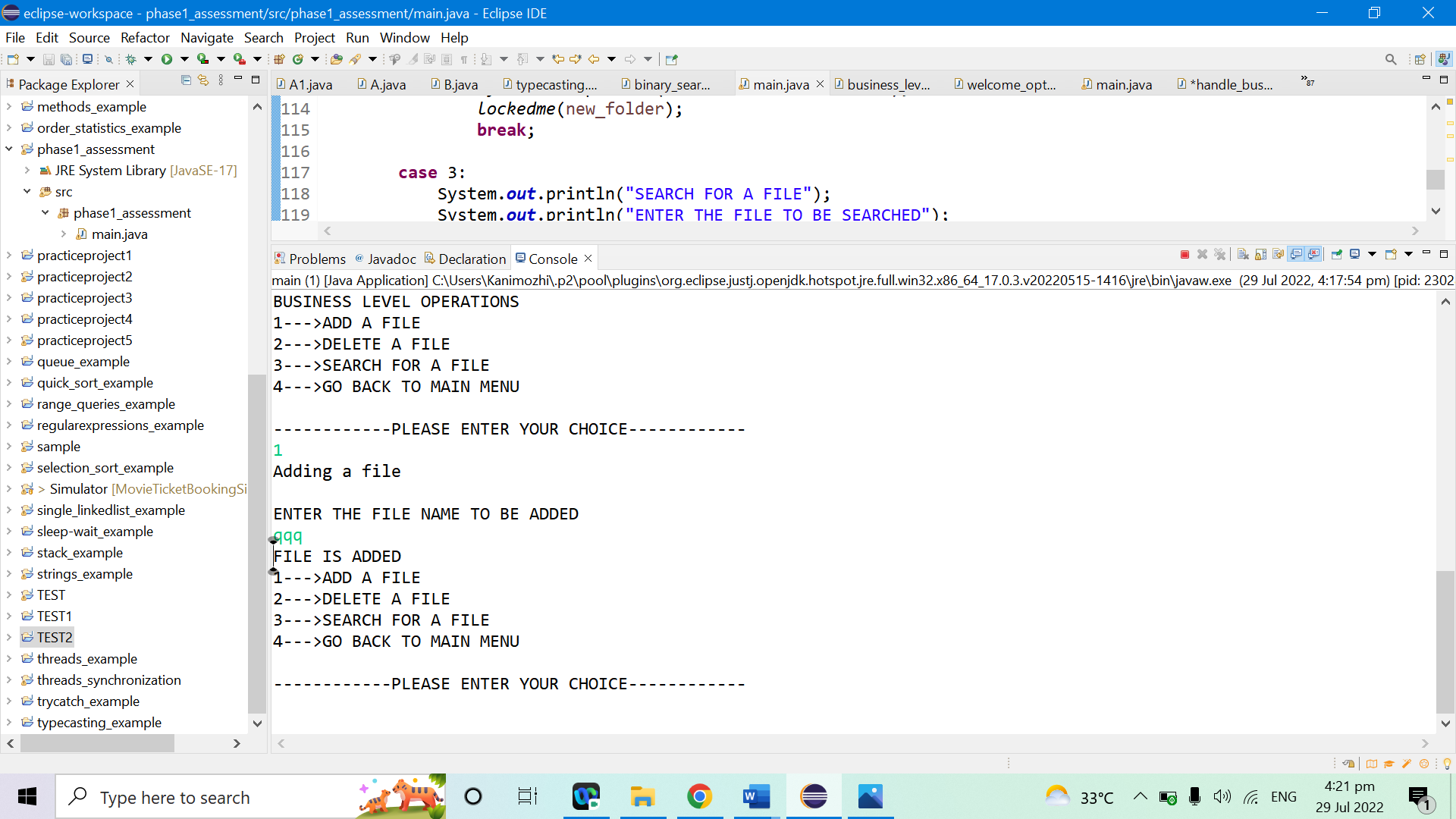
SORTING THE FILES IN ASCENDING ORDER



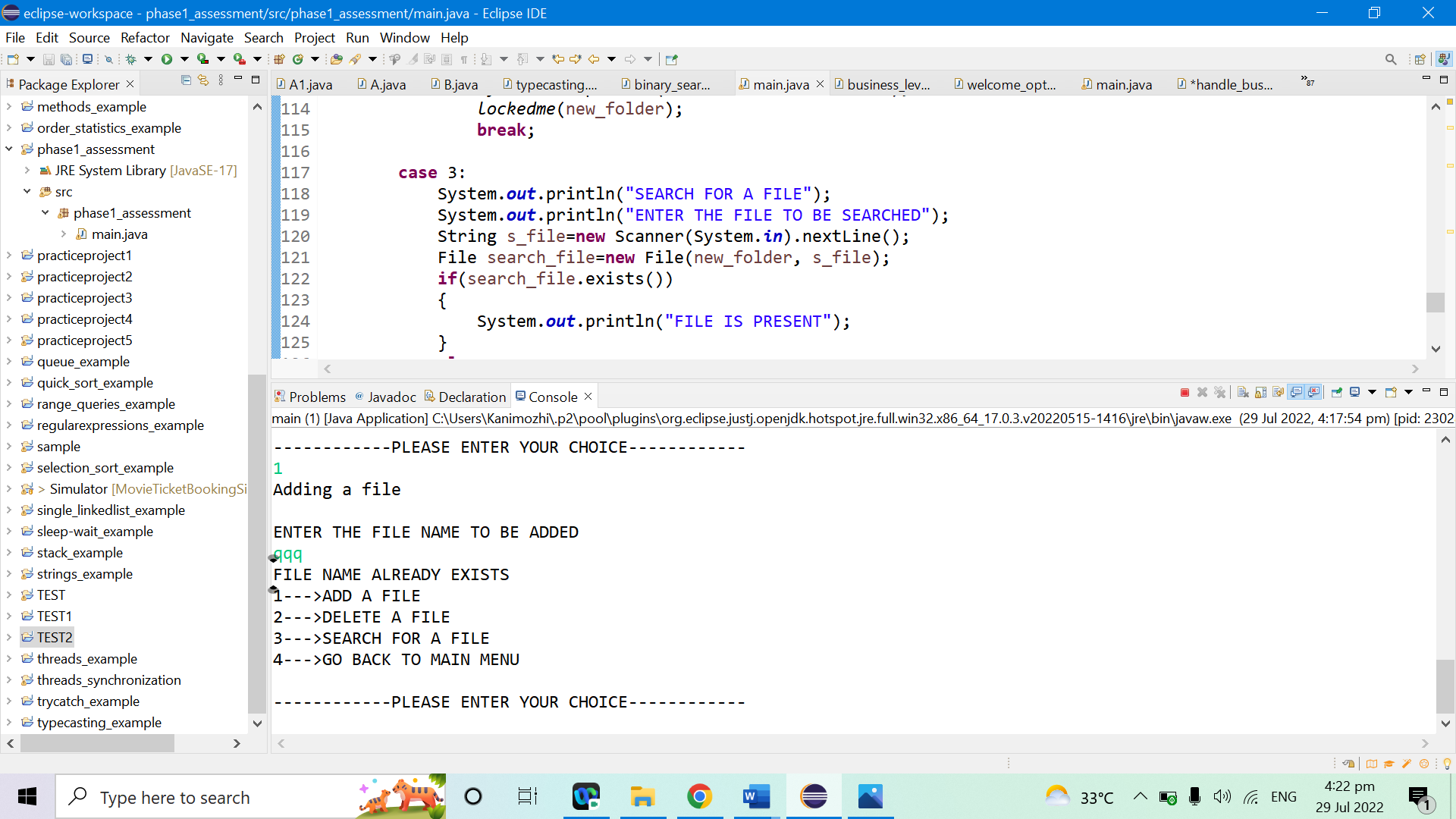
SHOWING BUSINESS LEVEL OPERATIONS MENU



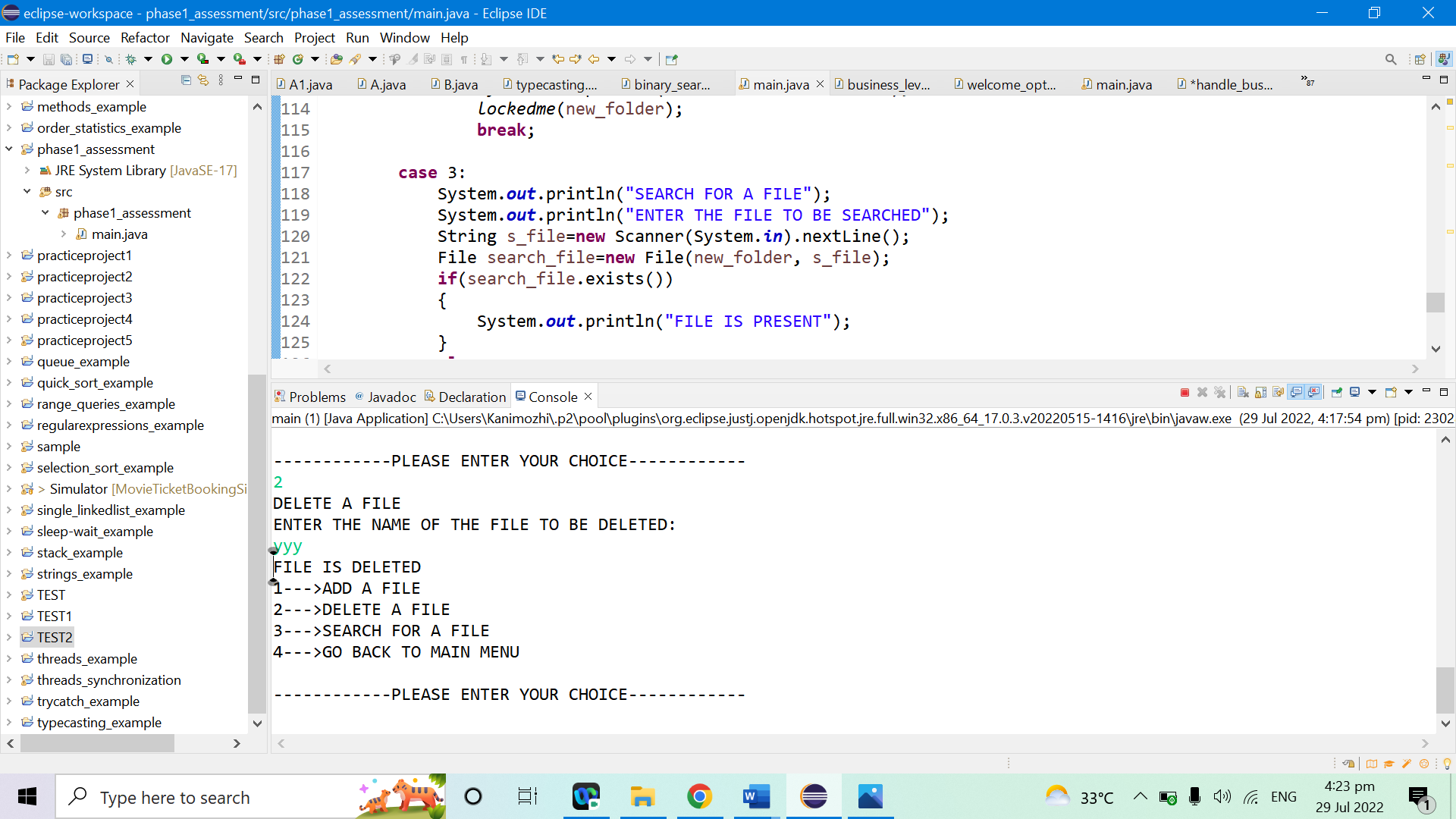
ADDING A FILE



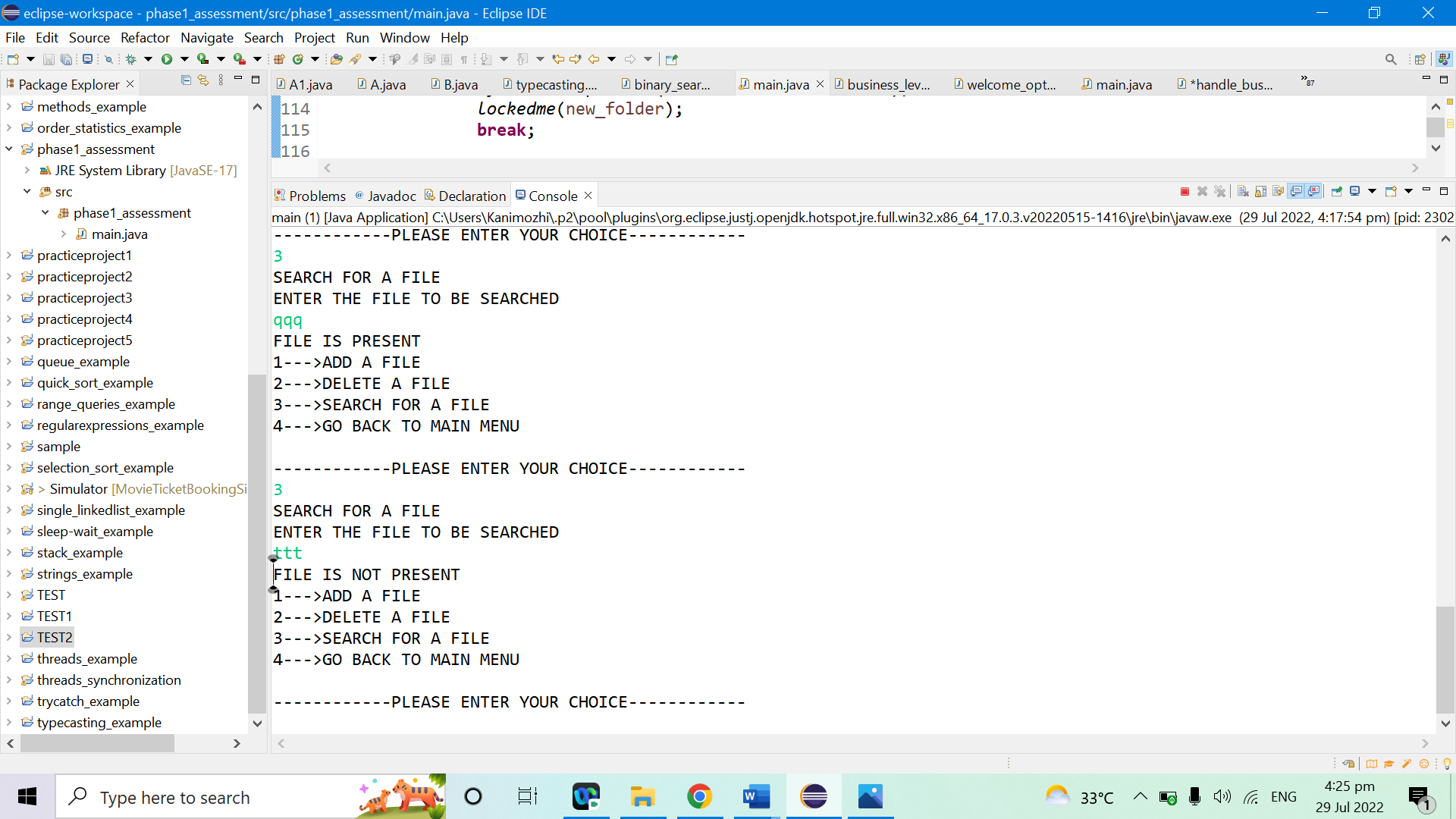
ADDING A FILE WITH FILE NAME ALREADY EXISTS IS NOT POSSIBLE



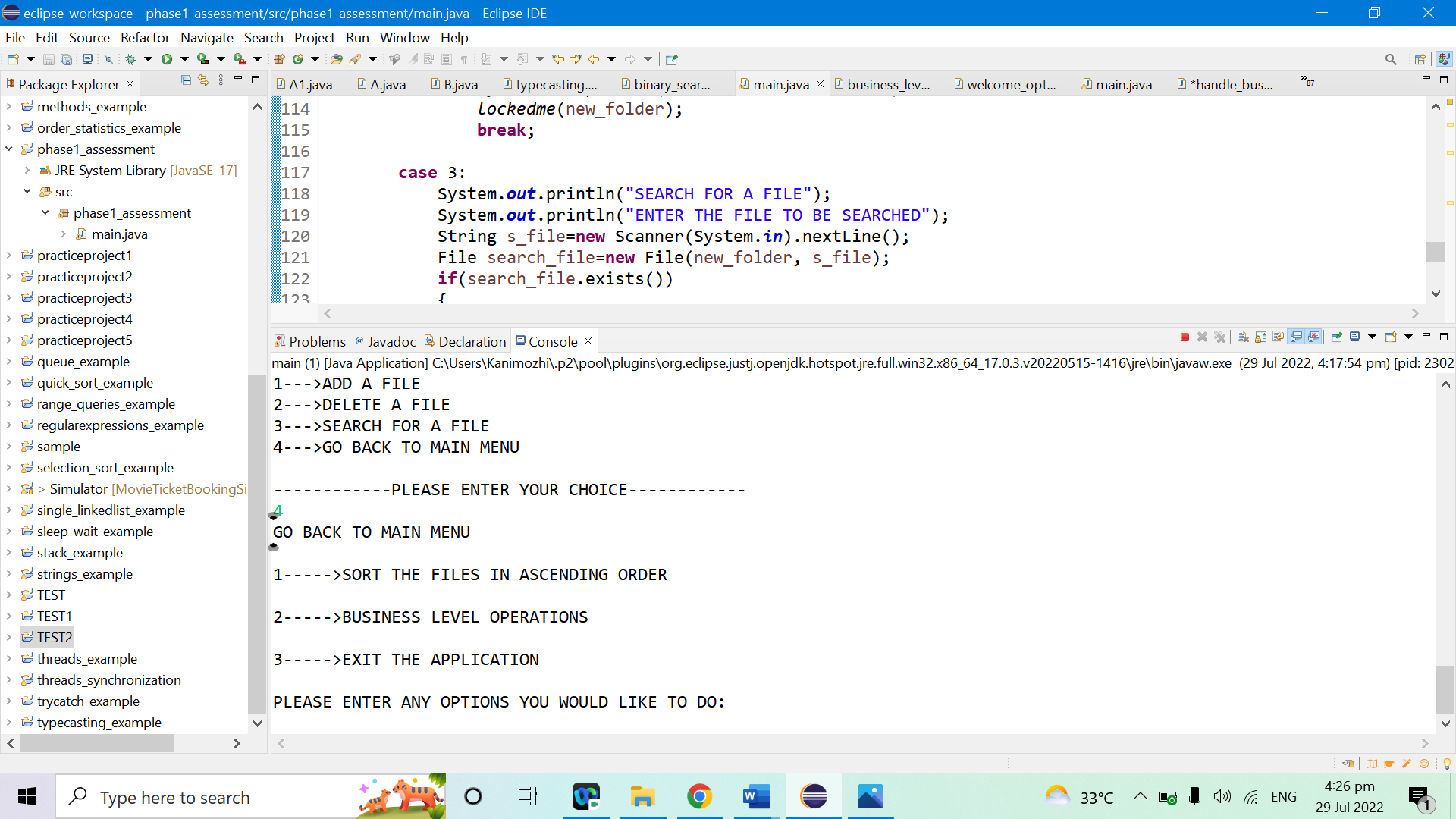
DELETING THE FILE (yyy)



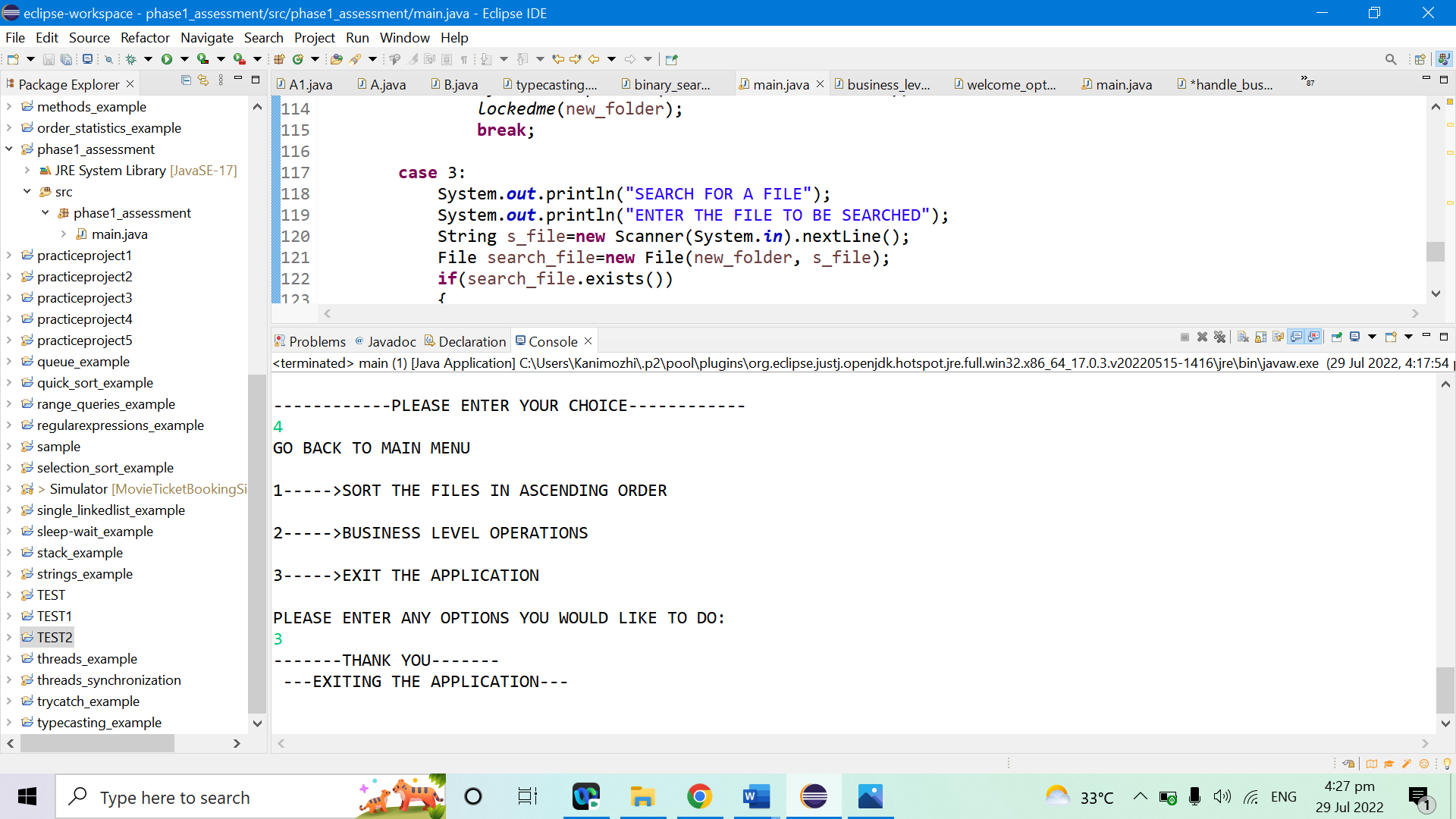
SEARCHING FOR A FILE



GOING BACK TO MAIN MENU



EXITTING THE APPLICATION



**CONCLUSION**

There is a huge scope of improvement in this Locked me project, one can easily add the functionality for navigating within folders, and adding, deleting files within a folder of the root directory. Also, a functionality to write within the text files can be added. The USP of the project is that once fully developed it can act as a complete file management system and can be used by individuals to safe keep data in serialised form. That is an additional feature that needs to be implemented thereby one can keep his data in a secure format in the Locked me app.