Specification Document

By Agis Spyropoulos

Present document is one of the deliverables in the context of the JAVA FSD course for Vodafone employees, particularly related to the Course-end Project 1 / Assessment:

> *"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."*

Course:	Java Full Stack Development
Phase:	1 - Implementing OOPS using Java with Data Structures and Beyond
Author:	Agis Spyropoulos (Agis.Spyropoulos@vodafone.com)
Company:	Vodafone
Document Name:	C:\Users\aspyr\git\repository\JavaTraining\Phase1_Proj_DMS\Documentation\01 - Specification_Document\Specification_Document.pdf
Date:	2023.01.30

Contents

Project DMS – Digitization Management System

Project Objective

Company **Lockers Pvt. Ltd.** aims to digitize their products and chose LockedMe.com as their first project to start with. The project objective is to develop a prototype of the application **DMS (Digitization Management System)**. The application prototype[1] is to be presented to the relevant stakeholders for budget approval.

Project Main Functionalities

- Displaying the Welcome screen at the start of the DMS Application. This will include the following:
  - Application name (and the developer details)
  - Options displaying the user interaction information
  - Features to accept the user input to select one of the options listed
- [F1] Retrieving the file names in an ascending order
  - **List the current file names in ascending order**. The root directory can be either empty or contain few files or folders in it
- [F2] Business-level operations (FILE OPERATIONS Menu):
  - [F2.1] **Add a user specified file** to the application (case sensitivity of the file names is ignored)
  - [F2.2] **Delete a user specified file** from the application (case sensitivity on the file name must be utilized so that the right file is deleted; a message is returned if the file is NOT found)
  - [F2.3] **Search a user specified file** from the application[2] (case sensitivity on the file name must be utilized so that the right file is retrieved; result is displayed upon successful or unsuccessful operation)
  - [F2.4] **Return to the main context** by closing the business-level operations
- [F3] Close the application

Note that **a fifth hidden functionality will be implemented** [F2.5] in the business-level operations, i.e. hidden for the FILE OPERATIONS Menu, mainly for educational/checking purposes. This functionality will NOT be implemented in a real customer environment. The functionality lists the file names as they are currently ordered in the ***database***, in order to check/verify that the filenames (by the time of running it) have remained unsorted prior to using the sort functionality that is implemented in [F1].

Project Development Methods & Tools

- Eclipse/IntelliJ: An IDE to code for the application (**using Java language**)
- Git, GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally

[1] As this is a prototyped application, the user interaction will be via a command line.

[2] The searching is done for the purpose of displaying the file attributes.

Sprints Planned & Tasks Achieved in Each One

Sprint 1 > Backbone Flow

Implementation of the application backbone (menu) flow:

- MAIN MENU displayed (not fully deployed)
- User choices entered (no exception handling)
- Flow followed **without having the relevant functionalities operational yet**
- FILE OPERATIONS Menu displayed (not fully deployed)
- Return to MAIN MENU
- Exit of the application

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
X									

Sprint 2 > Structures

Implementation of structures:

- Definition/Analysis of the Classes to be used
- Structure of each Class
- Methods & Relations
- Definition/Analysis of the ***database*** to be used
- Definition/Analysis of the ***service*** layer to be used

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
	x								

Sprint 3 > Business Logic for MAIN MENU

Implementation of the business logic that will be utilized by the MAIN MENU:

- Full Implementation of the MAIN MENU display
- Implementation of functionality "*Display File Names*"
- Implementation of ***database*** sorting
- Full Implementation of the FILE OPERATIONS Menu
- Implementation of the exit DMS functionality

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
		x	x						

Sprint 4 > Business Logic for FILE OPERATIONS Menu

Implementation of the business logic that will be utilized by the FILE OPERATIONS Menu. That implies the following:

- Implementation of the ***service*** functionality "*Add New File*"
- Implementation of the ***service*** functionality "*Delete Existing File*"
- Implementation of the ***service*** functionality "*Find Existing File*"
- Implementation of the ***service*** functionality to display File Attributes
- Implementation of the ***service*** hidden functionality to display the ***database*** filenames

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
				x	x	x			

Sprint 5 > Exception Handling

Implementation of Exception Handling. That implies the following:

- Implementation of validating user choice for the MAIN MENU
- Implementation of validating user choice for the FILE OPERATIONS Menu
- Implementation of validating user input regarding the filename string

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
							x		

Sprint 6 > Testing | Documentation

Implementation of the following:

- Test application via console, by following a smooth DMS flow (for all test cases below a screenshot[3] is stored in the GitHub "*Documentation*" folder):
    - Run DMS Application
    - Check ***database*** initialization (filenames loaded from a valid active folder)
    - Check all MAIN MENU elements displayed
    - Test successful ***service*** functionality "*Display File Names*"
    - Move to FILE OPERATIONS Menu and check all its elements
    - Test successful ***service*** functionality "*Find Existing File*"
    - Test successful ***service*** functionality "*Delete Existing File*"

[3] These are the screenshots displayed in the chapter "Algorithms & Flowcharts of the Application".

- Test successful ***service*** functionality "*Add New File*" (add various new files)
- Check that all new files are appended at the end of the ***database*** thus the ***database*** is currently unsorted (hidden functionality)
- Return to MAIN MENU
- Test again successful ***service*** functionality "*Display File Names*"; verify now that the files are sorted
- Exit DMS Application

- Test application via console, by entering invalid data (for all test cases below a screenshot[4] is stored in the GitHub "*Documentation*" folder):

  - Run DMS Application and display the MAIN MENU
  - Enter invalid choices for the MAIN MENU, i.e.
    - ✓ any integer except 1, 2, 3
    - ✓ any decimal number
    - ✓ [Enter]
    - ✓ Any string
  - Go to FILE OPERATIONS Menu and enter again invalid choices, i.e.
    - ✓ any integer except 1, 2, 3, 4 (and the hidden 11)
    - ✓ any decimal number
    - ✓ [Enter]
    - ✓ Any string
  - Input invalid filename strings, i.e. containing special characters like "?", "/", "|", "*", etc.
  - Change the active folder name, i.e. the ***ACTIVE_PATH*** string value and run again the DMS Application. Check that an error message is displayed and the application exits.

- Preparation of the Specifications Document and all the other elements that will be included inside that document, e.g. flowcharts, diagrams, figures, screenshots processed, etc.

Plan/Duration:

21jan	22jan	23jan	24jan	25jan	26jan	27jan	28jan	29jan	30jan
								x	x

Final Delivery

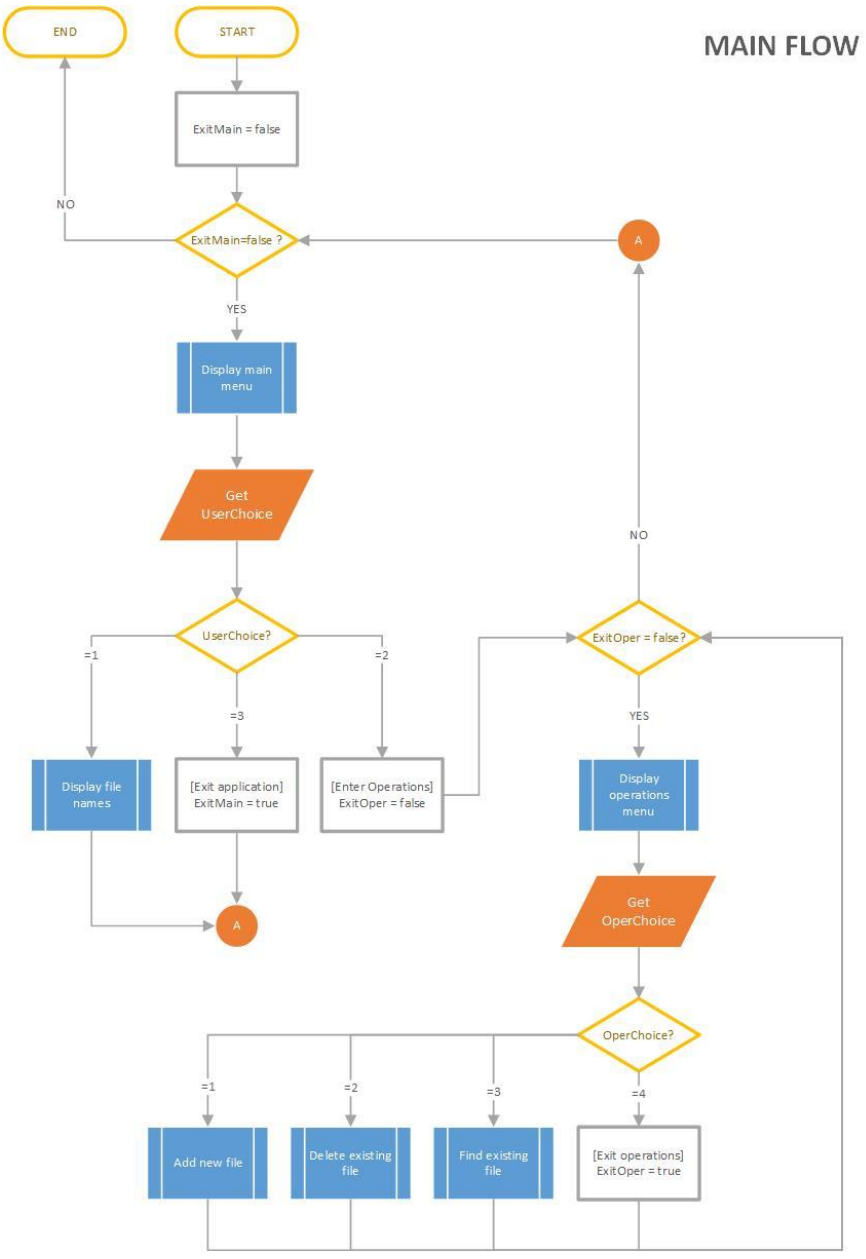
Final delivery will take place in January the 30th, 2023.

[4] These are the screenshots displayed in the chapter "Algorithms & Flowcharts of the Application".

Algorithms & Flowcharts of the Application

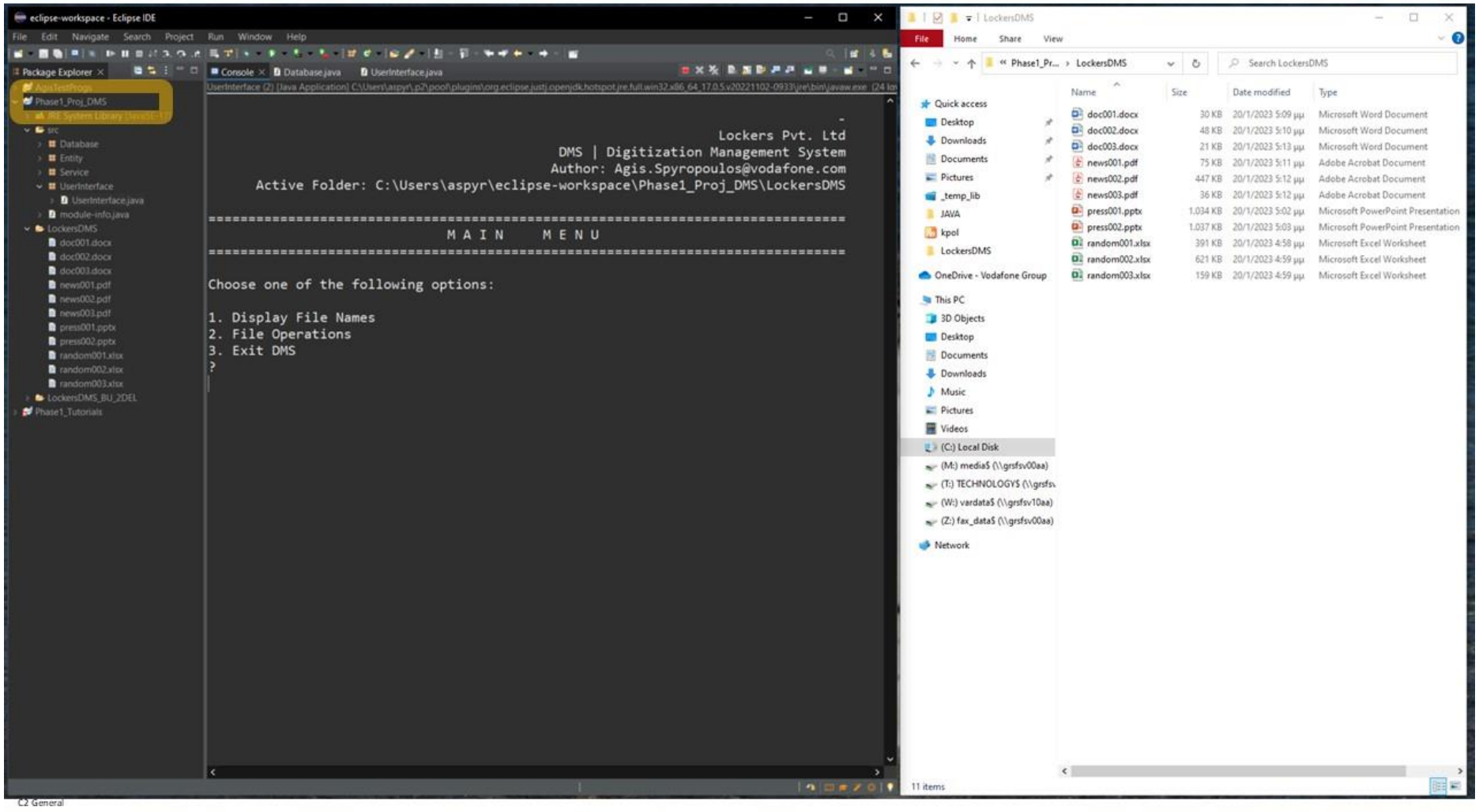
Main Application Flowchart

The main flowchart of the DMS application is the following:

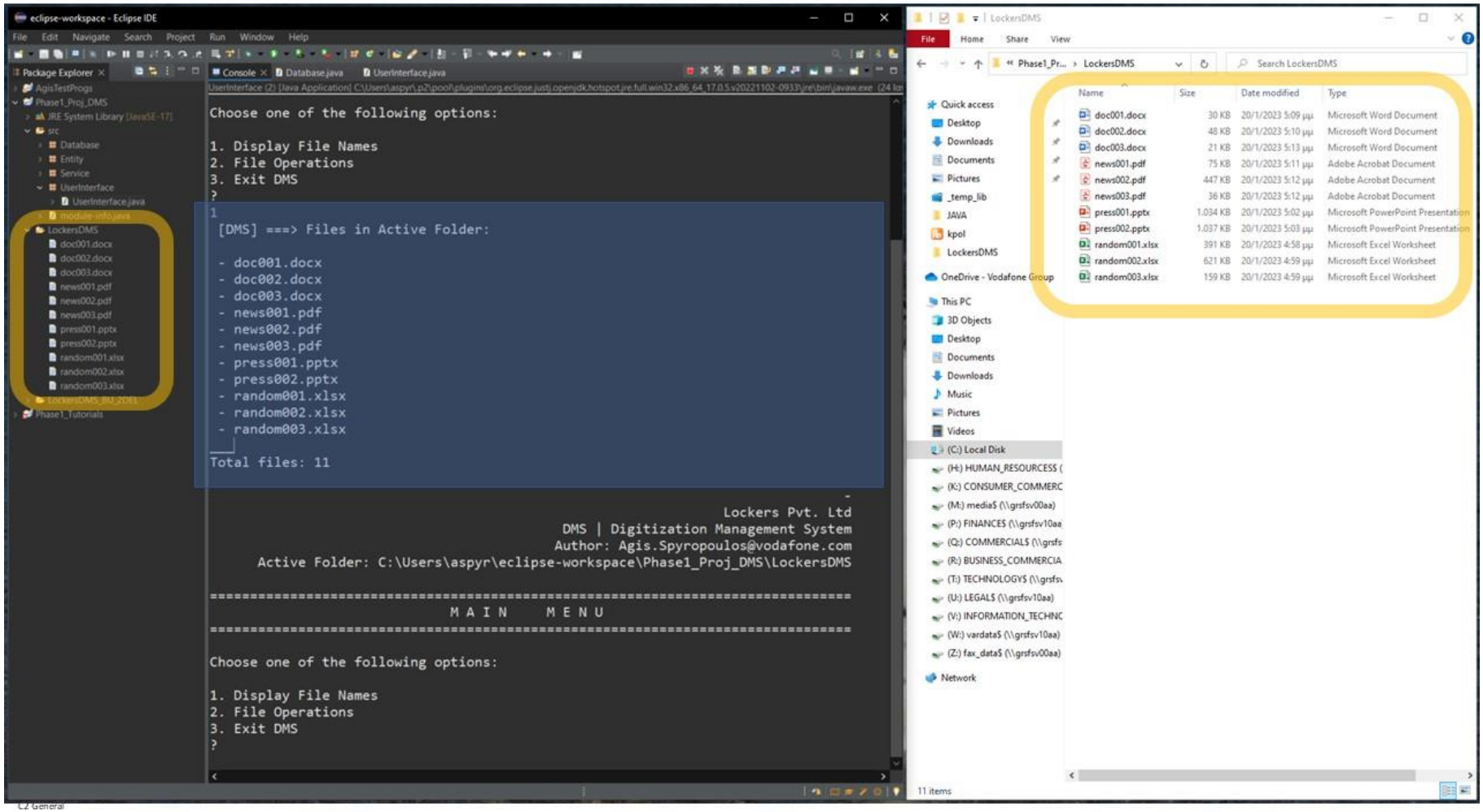


Step-By-Step Flow [Screenshots]

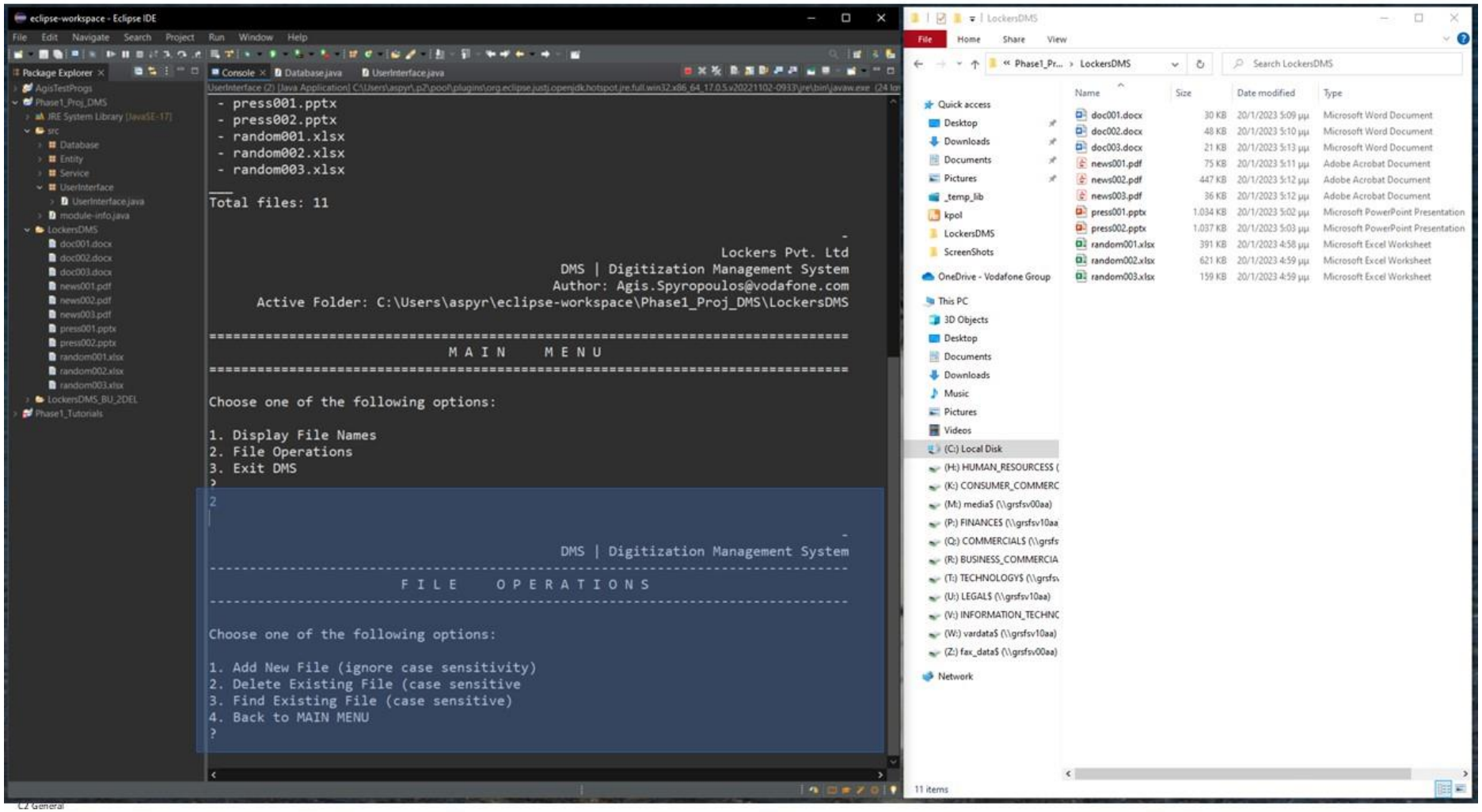
The following screenshots depict an analytical successful application flow on a step-by-step approach. The DMS application console is on the left (dark background color) whereas the actual active folder hosting the existing files is on the right (white background color).



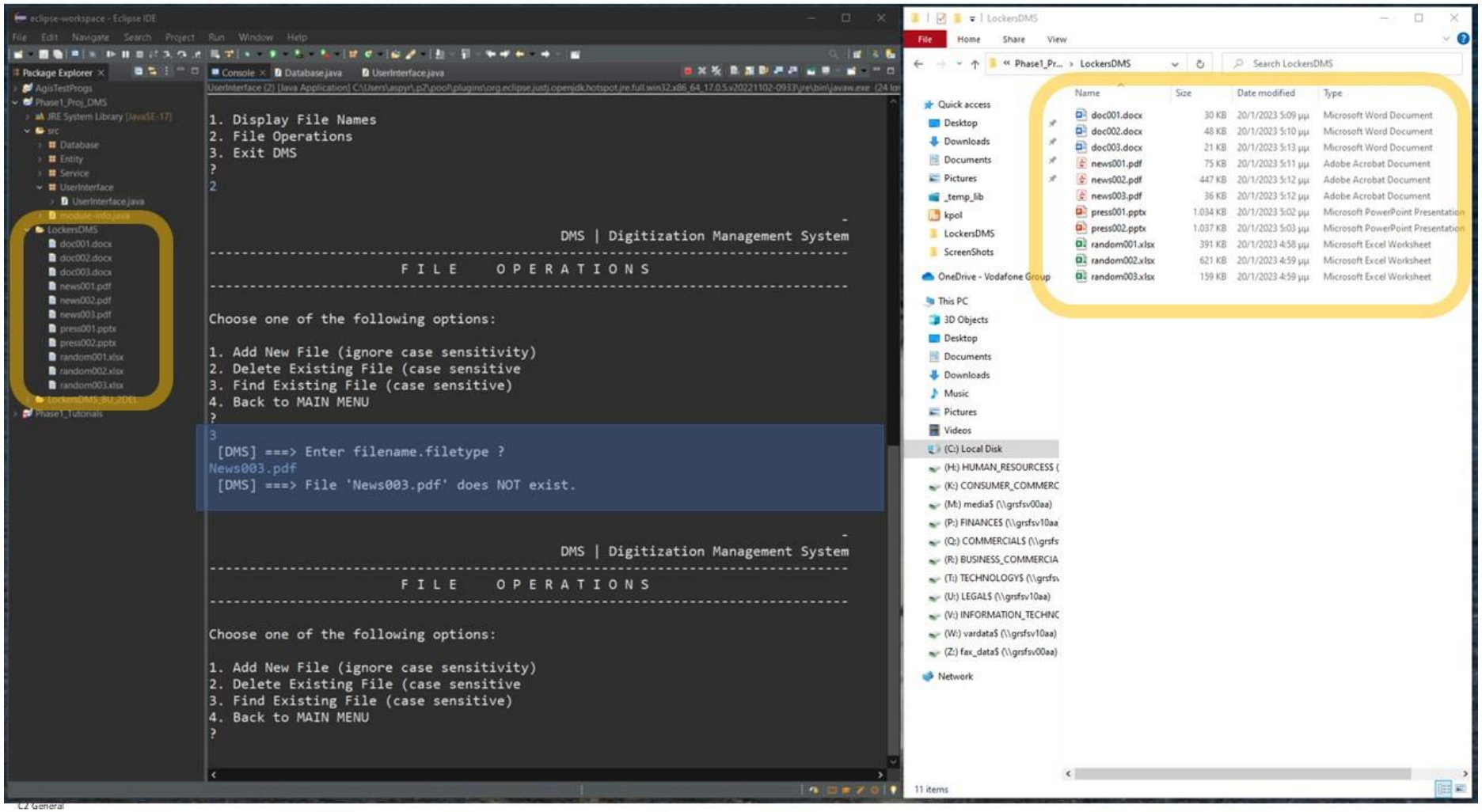
STEP-01: DMS Welcome page & MAIN MENU. In the DMS background the existing files in the active folder have already been loaded onto the ***database***.



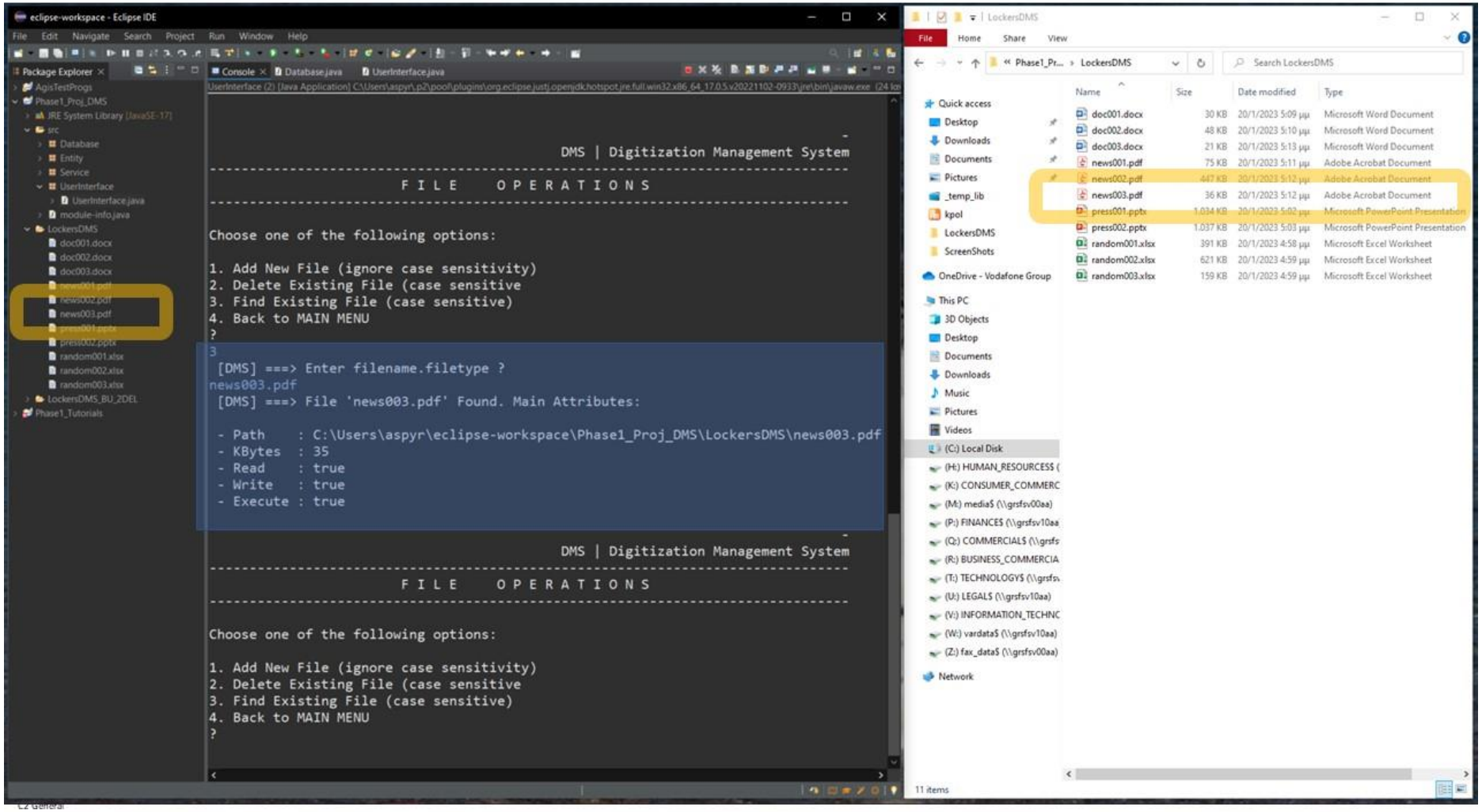
STEP-02: User Choice = 1 | ***database*** filenames are displayed on the console (refer also to the Windows folder “LockersDMS” (renamed as “SampleFiles” to the GitHub repository) on the right AND the same folder as displayed by Eclipse on the left).



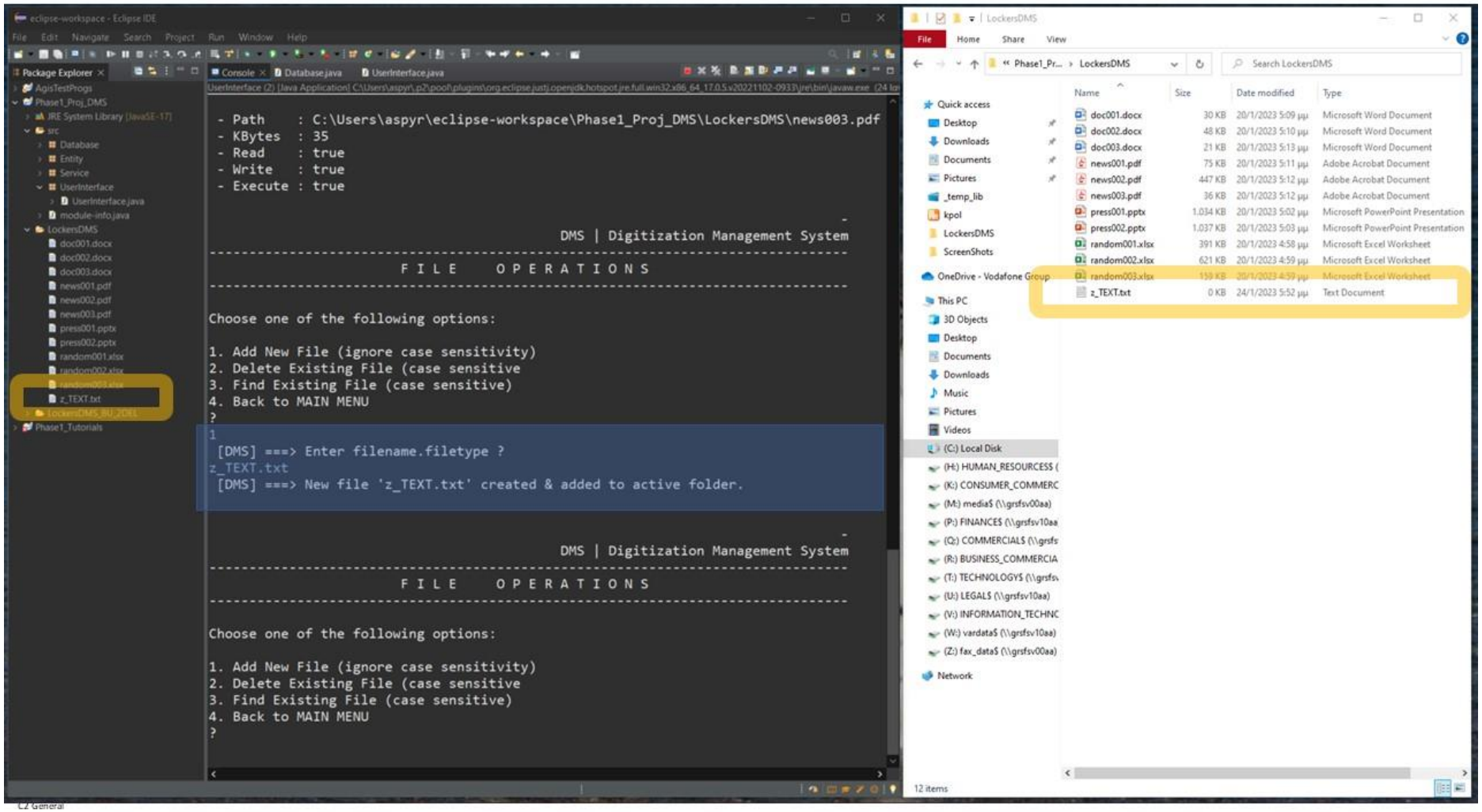
STEP-03: User Choice = 2 | Display menu with FILE OPERATIONS.



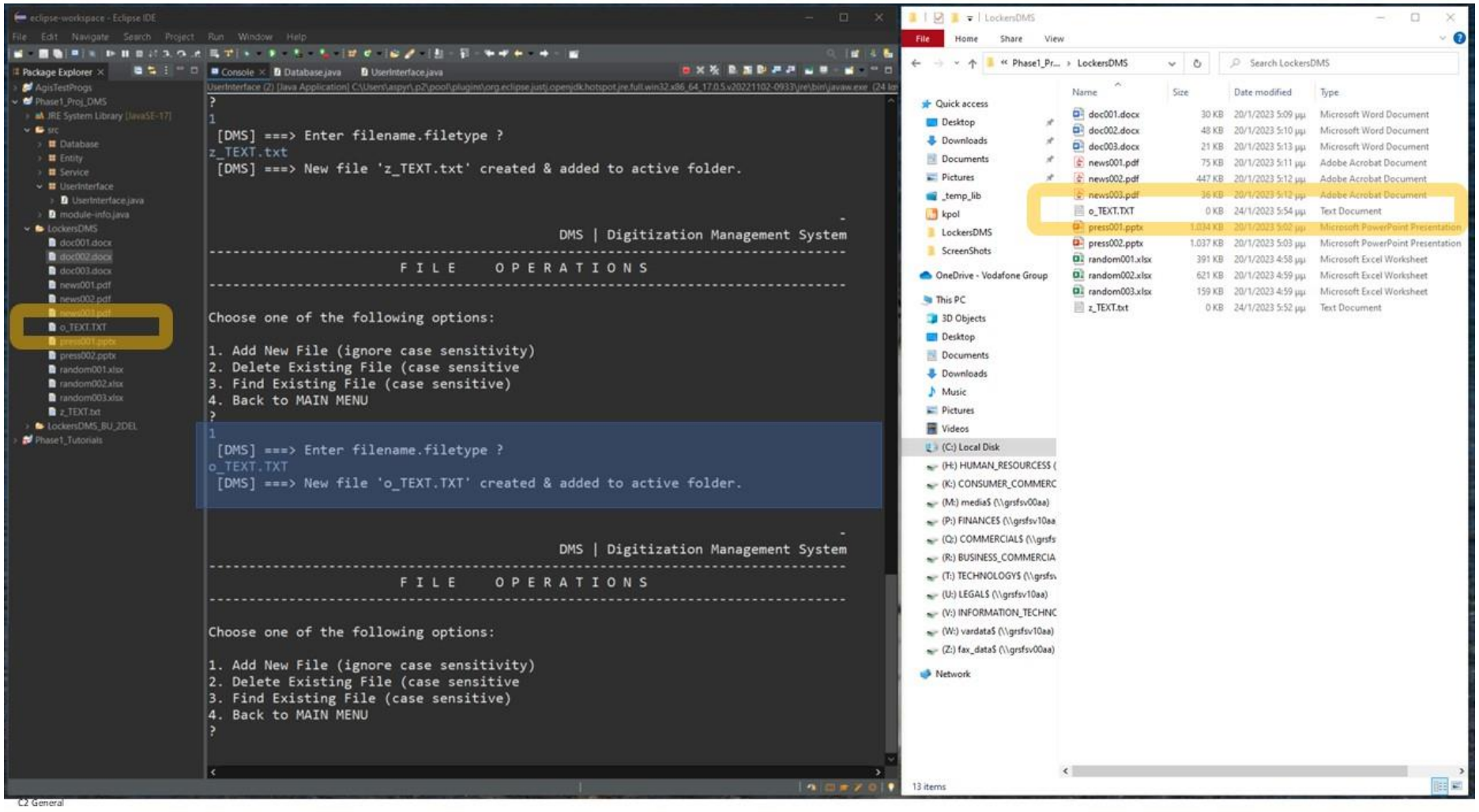
STEP-04: User Choice = 3 | User searches for file “News003.pdf” whereas the existing file is named “news003.pdf”. DMS returns a non-found message.



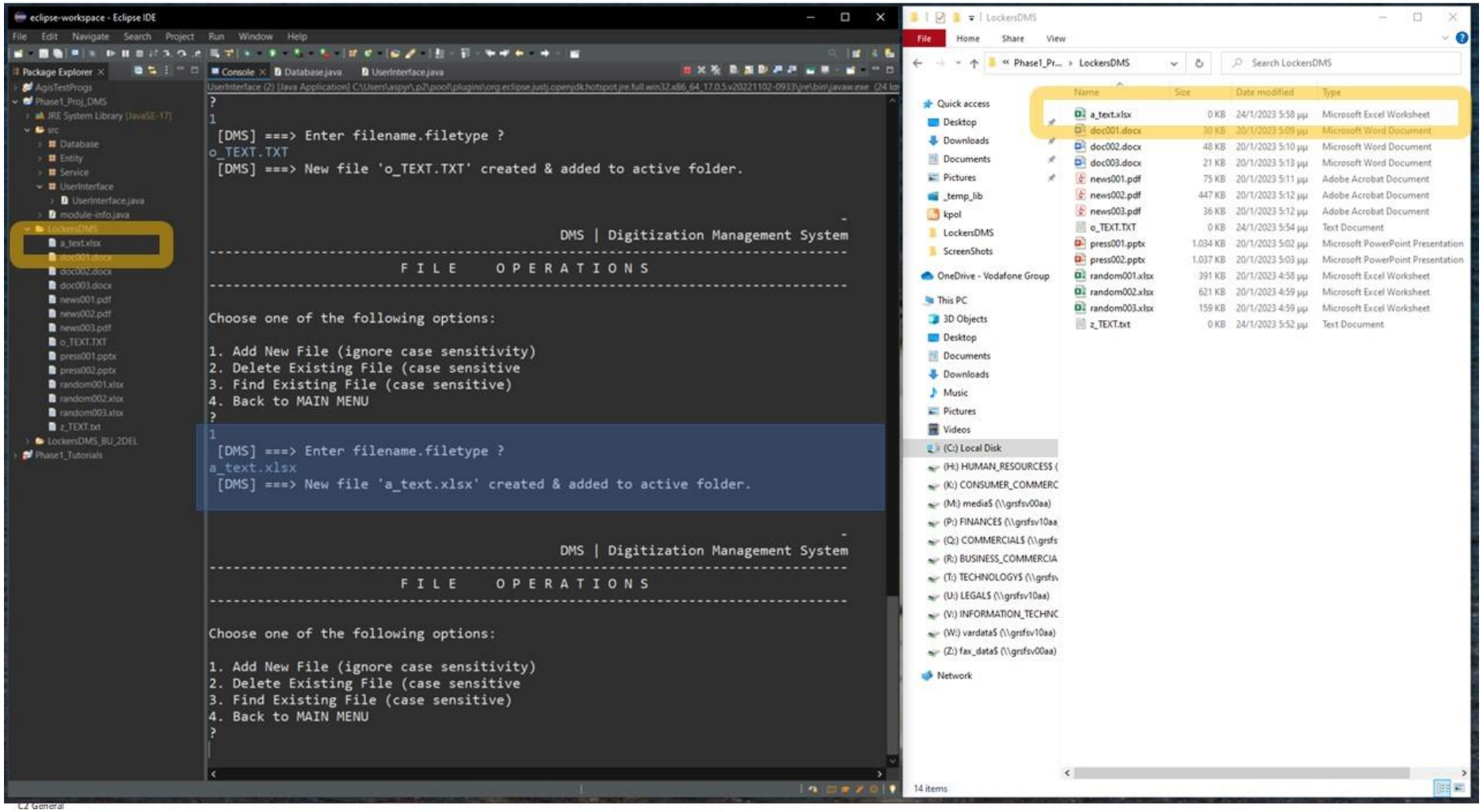
STEP-05: User Choice = 3 | User searches for file “news003.pdf”. File is found and DMS returns its attributes.



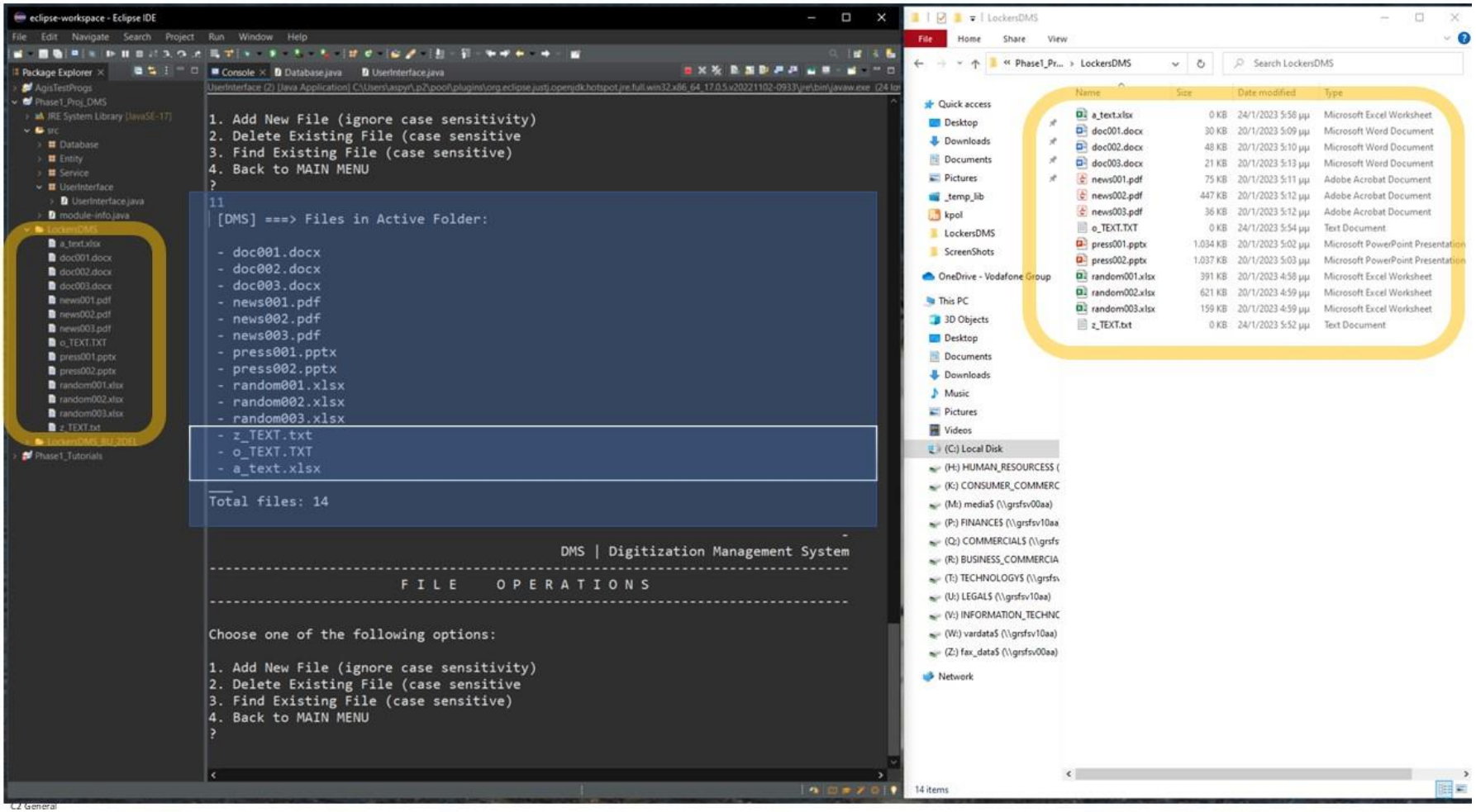
STEP-06: User Choice = 1 | User adds news file "z_TEXT.txt". The file is added in the active folder (check left and right) and it is also added to the ***database***.



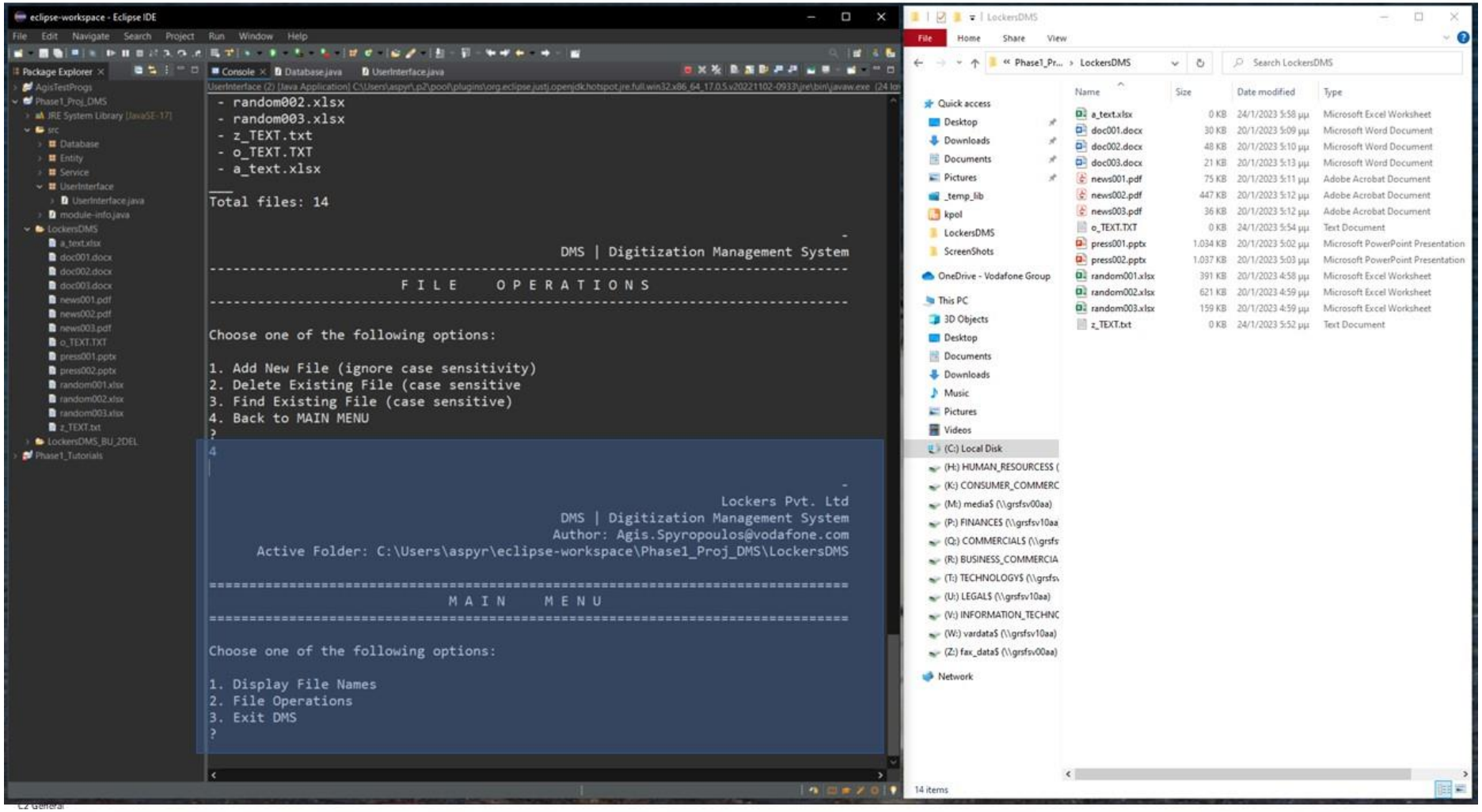
STEP-07: User Choice = 1 | User adds news file “o_TEXT.TXT”. The file is added in the active folder (check left and right) and it is also added to the ***database***.



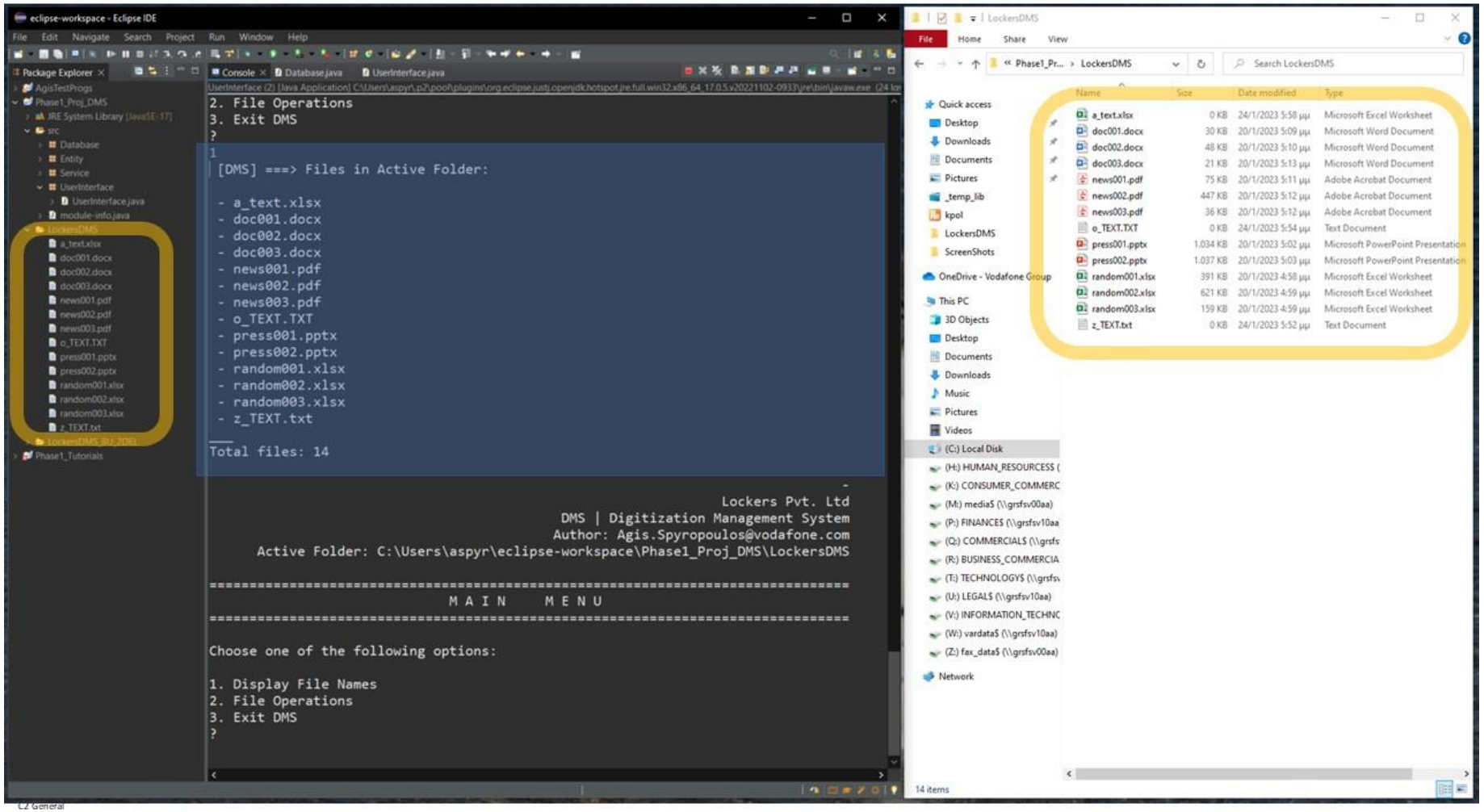
STEP-08: User Choice = 1 | User adds news file “a_text.xlsx”. The file is added in the active folder (check left and right) and it is also added to the ***database***.



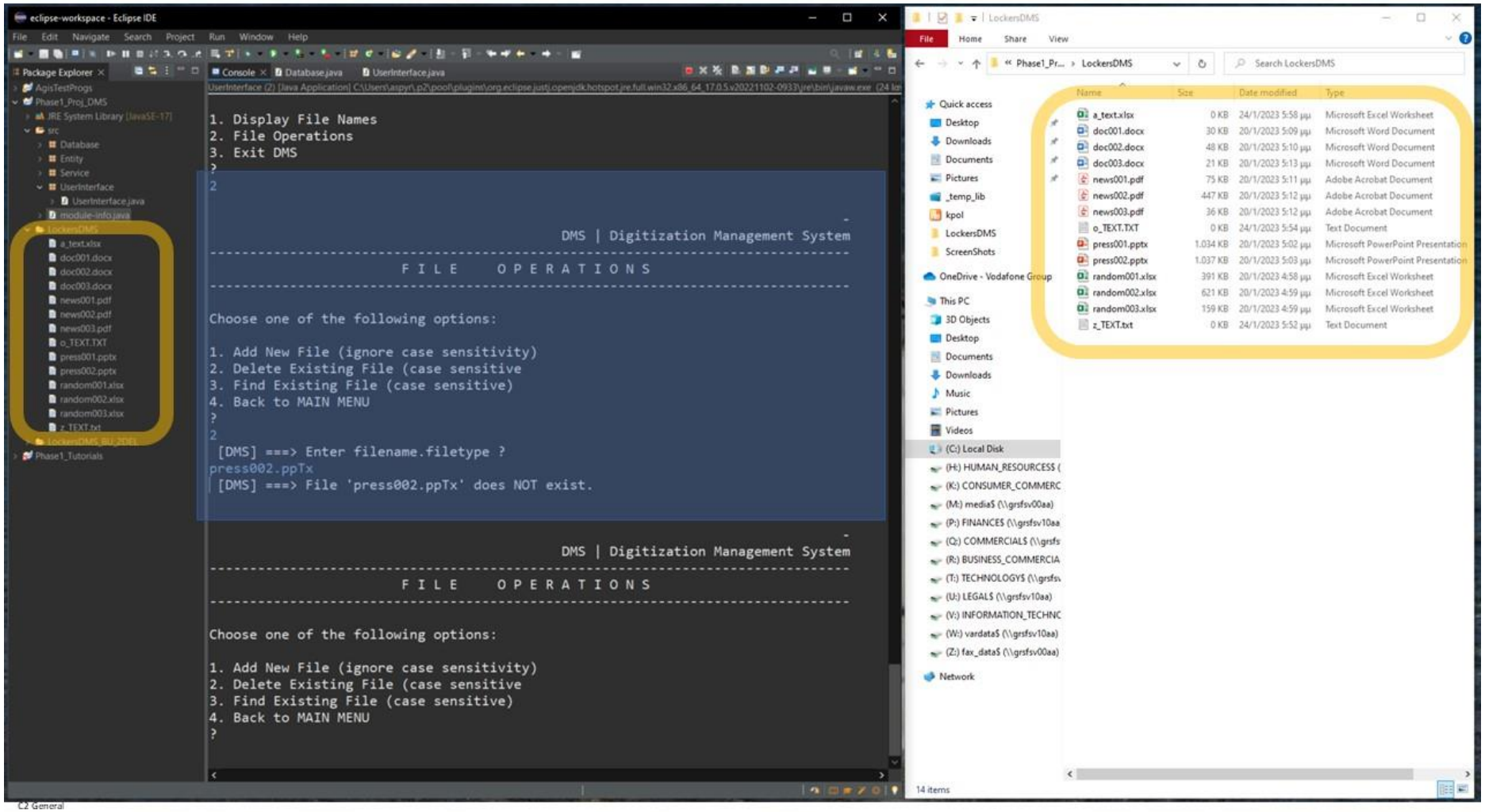
STEP-09: User Choice = 11 | This is a hidden choice implemented only for educational/checking purposes (will **NOT** be implemented in a real business environment). It displays the ***database*** files in order to check that the added files are appended at the end of the ***database***, i.e. the ***database*** is not sorted yet.



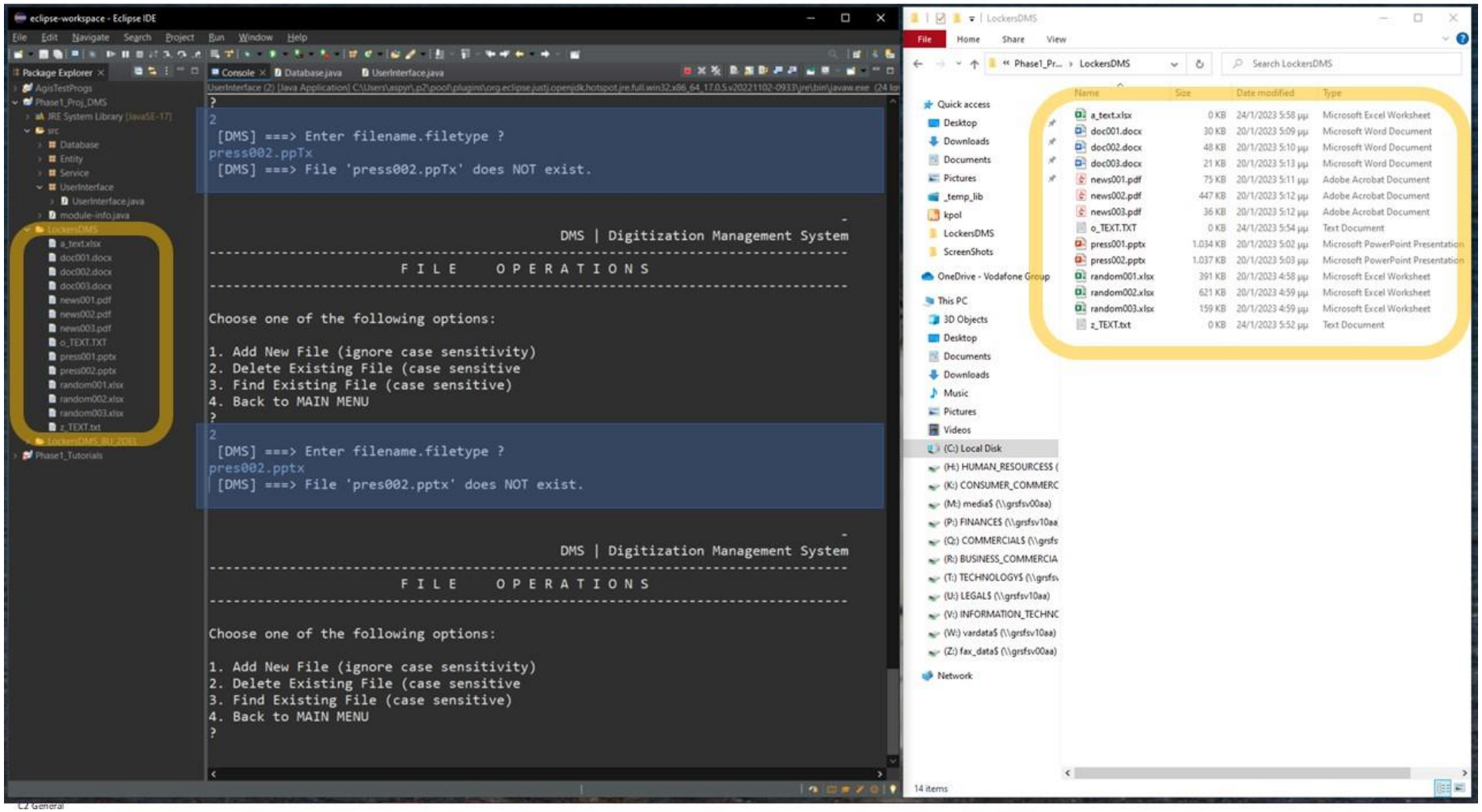
STEP-10: User Choice = 4 | Return to the MAIN MENU.



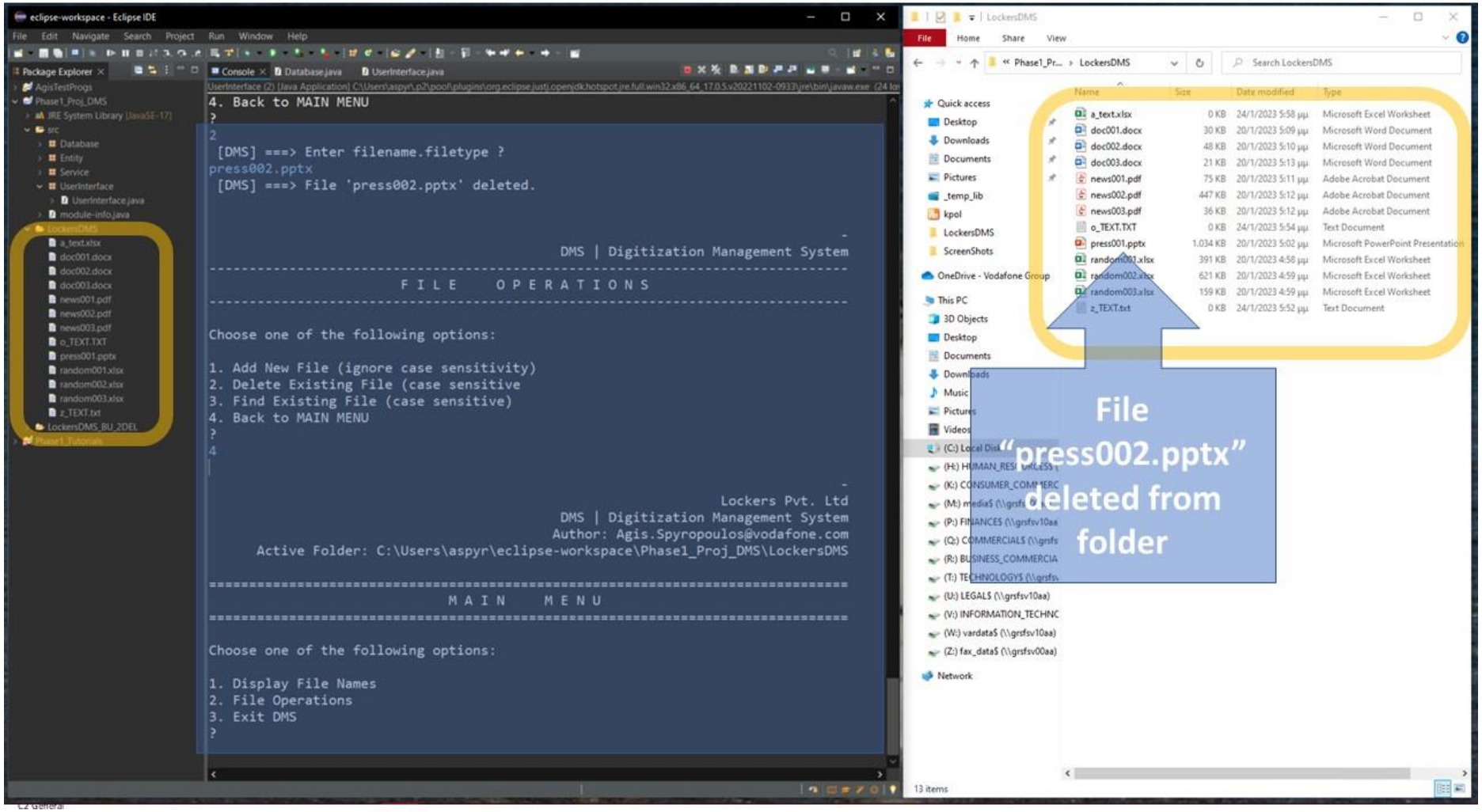
STEP-11: User Choice = 1 | File names of the java list are now sorted and displayed on the console (refer also to the Windows folder “LockersDMS” (renamed as “SampleFiles” to the GitHub repository) on the right AND the same folder as displayed by Eclipse on the left).



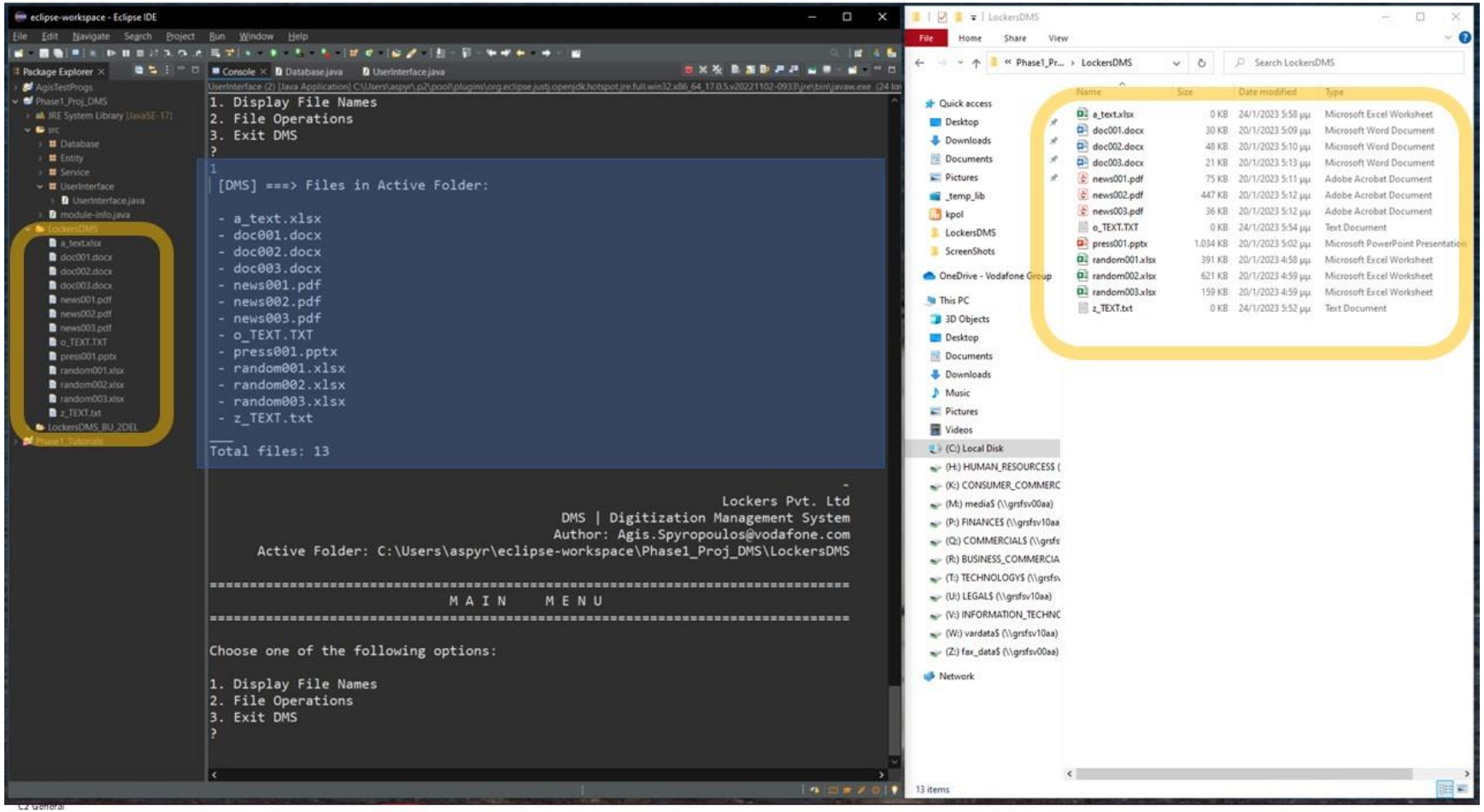
STEP-12: User Choice = 2 | Display submenu with FILE OPERATIONS. Next User Choice = 2 | Try to delete file “press002.ppTx” instead of file “press002.pptx”.



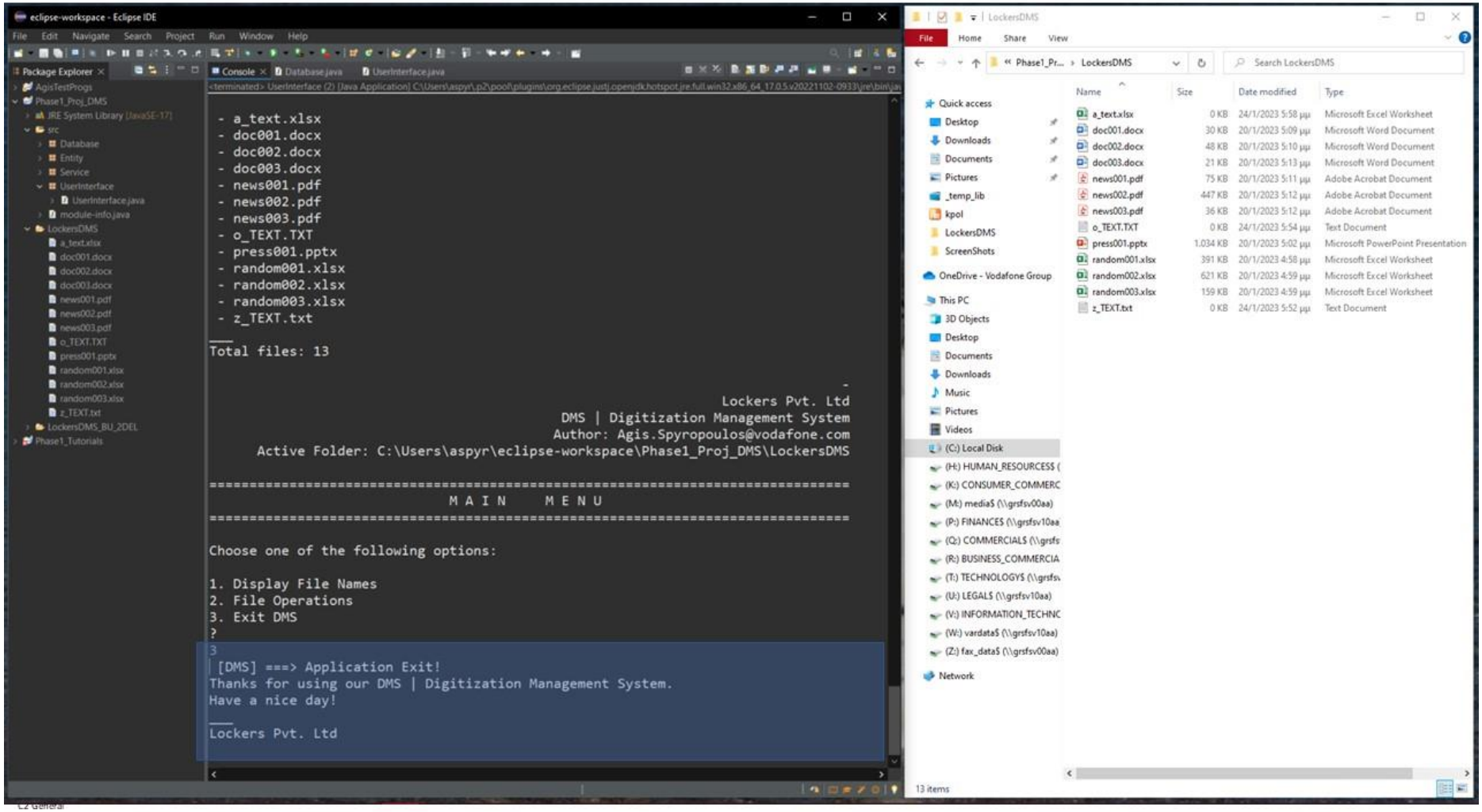
STEP-13: User Choice = 2 | Try to delete file “pres002.pptx” instead of file “press002.pptx”.



STEP-14: User Choice = 2 | Delete file "press002.txt". Next User Choice = 4 | Return to MAIN MENU.

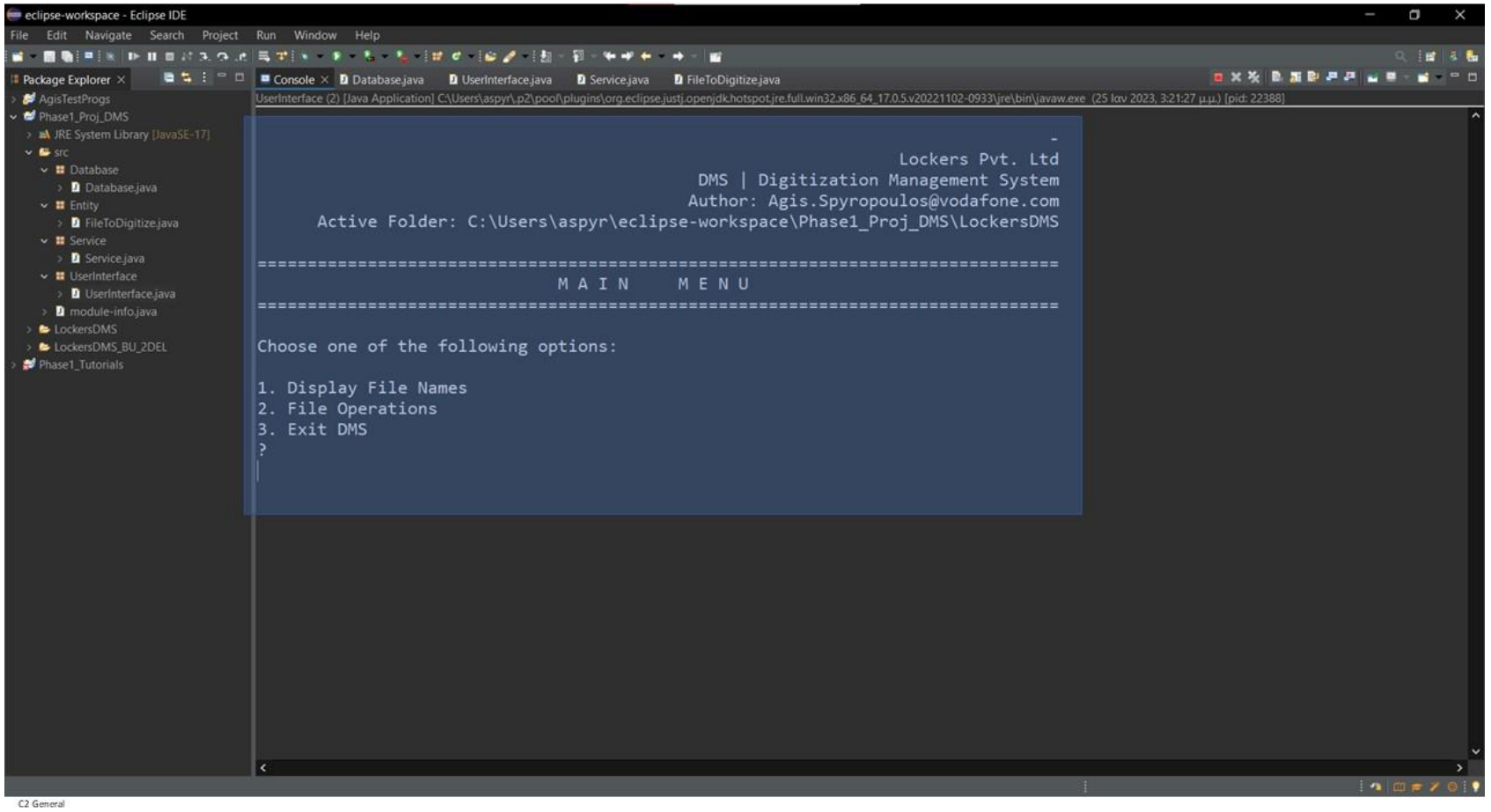


STEP-15: User Choice = 1 | File names of the java are displayed on the console (refer also to the Windows folder “LockersDMS” (renamed as “SampleFiles” to the GitHub repository) on the right AND the same folder as displayed by Eclipse on the left).

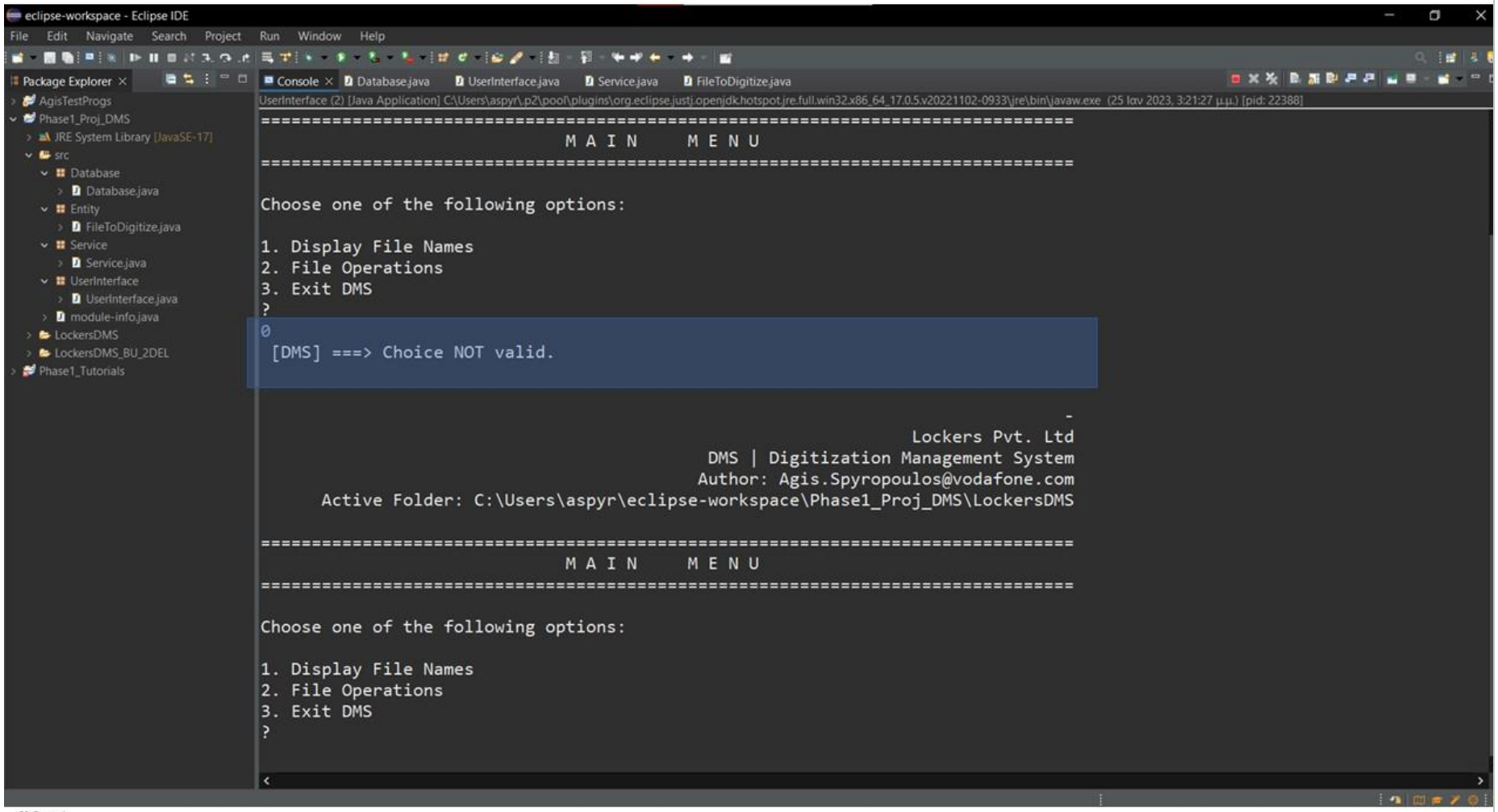


STEP-16: User Choice = 3 | Exit DMS Application.

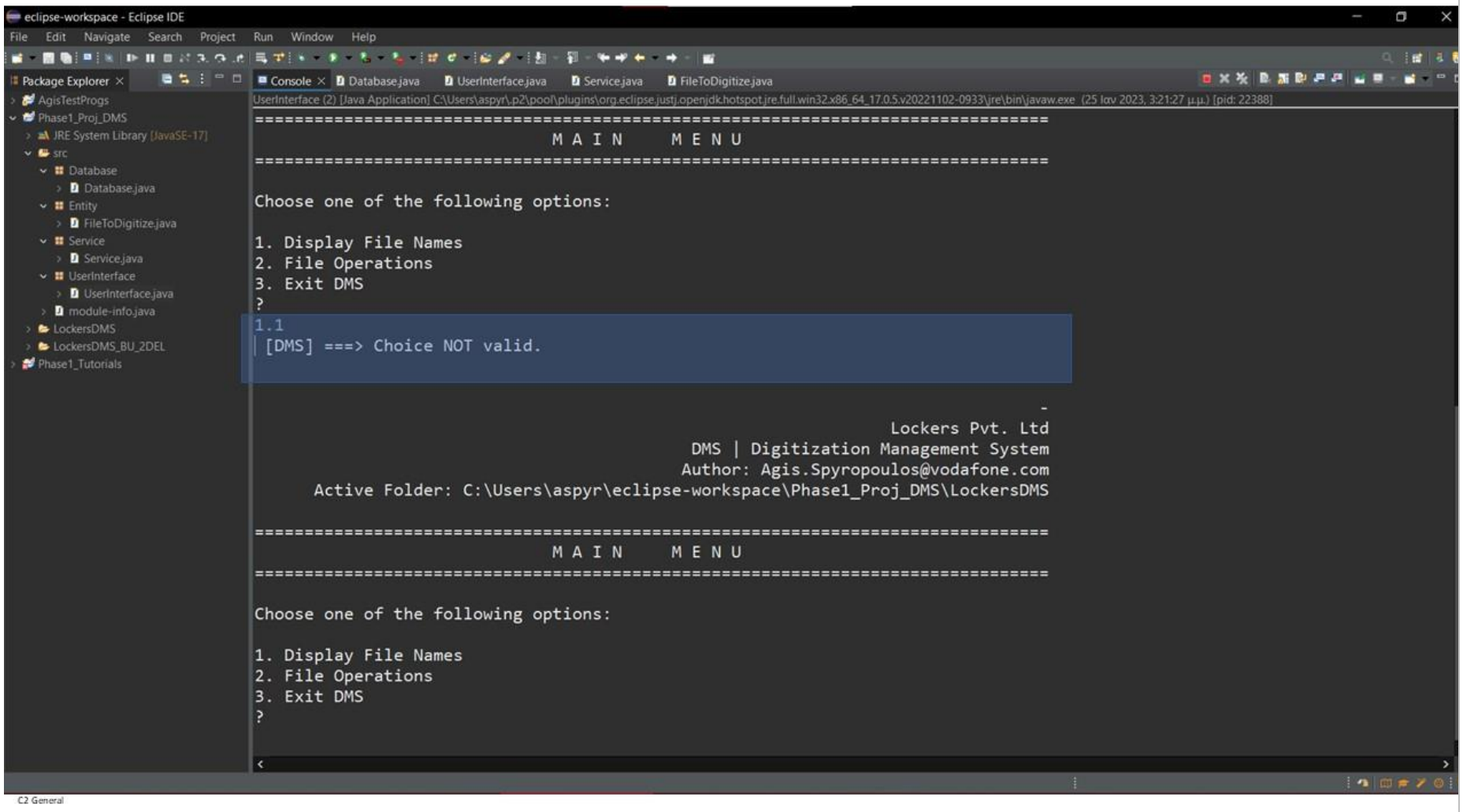
Exception Handling Cases [Screenshots]



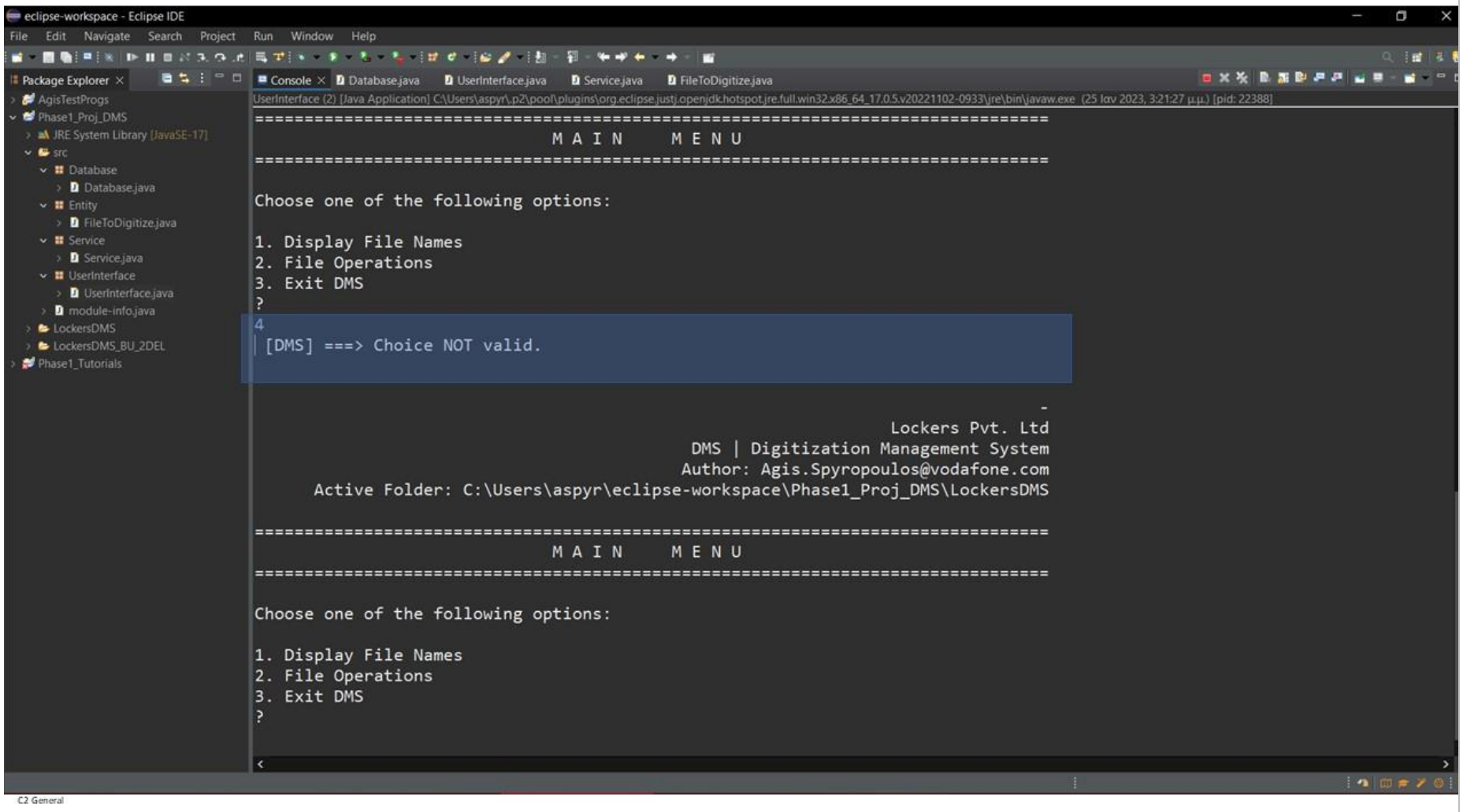
STEP-17: DMS Welcome page.



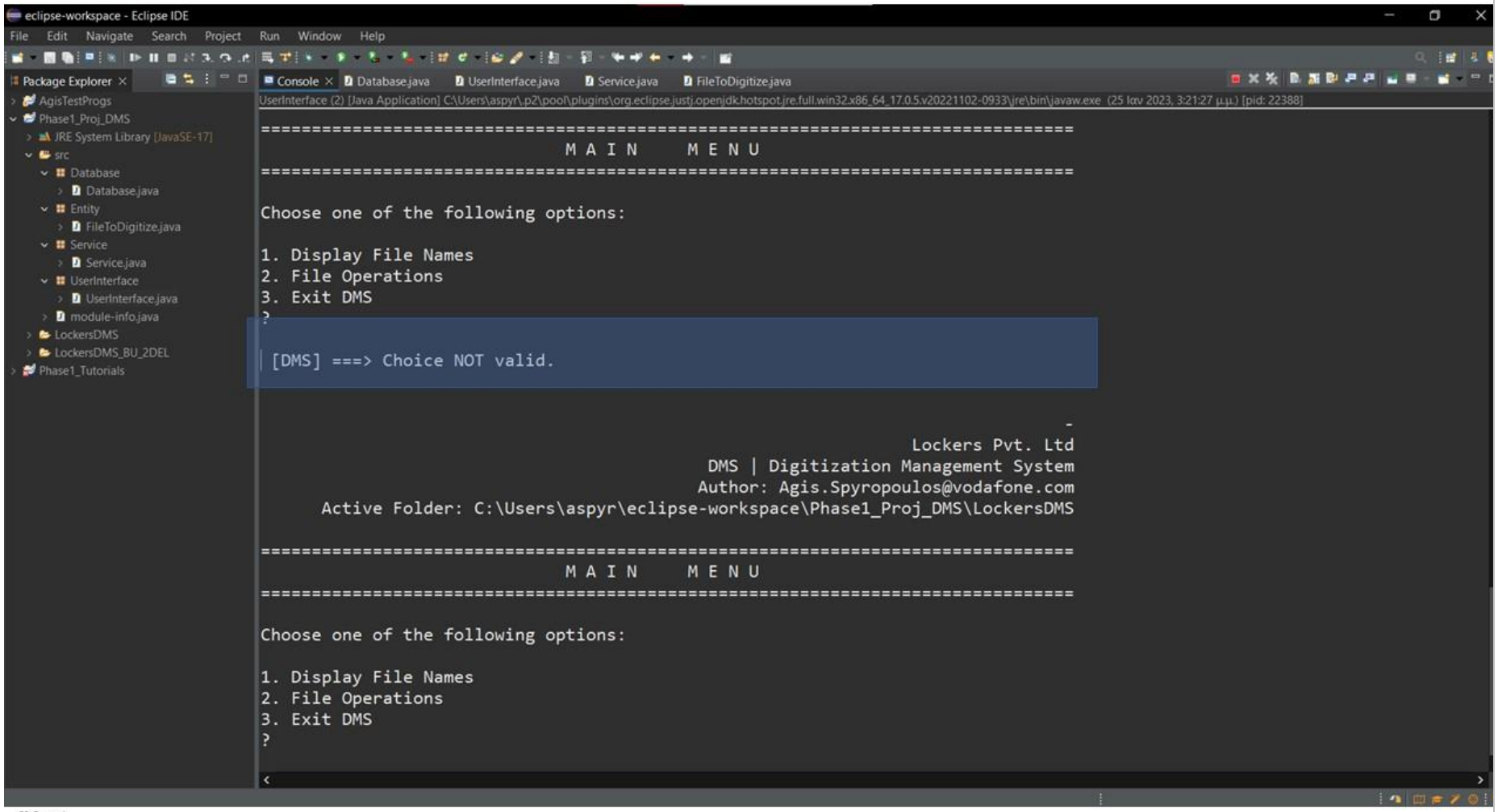
STEP-18: MAIN MENU, User Choice = 0, Choice not valid.



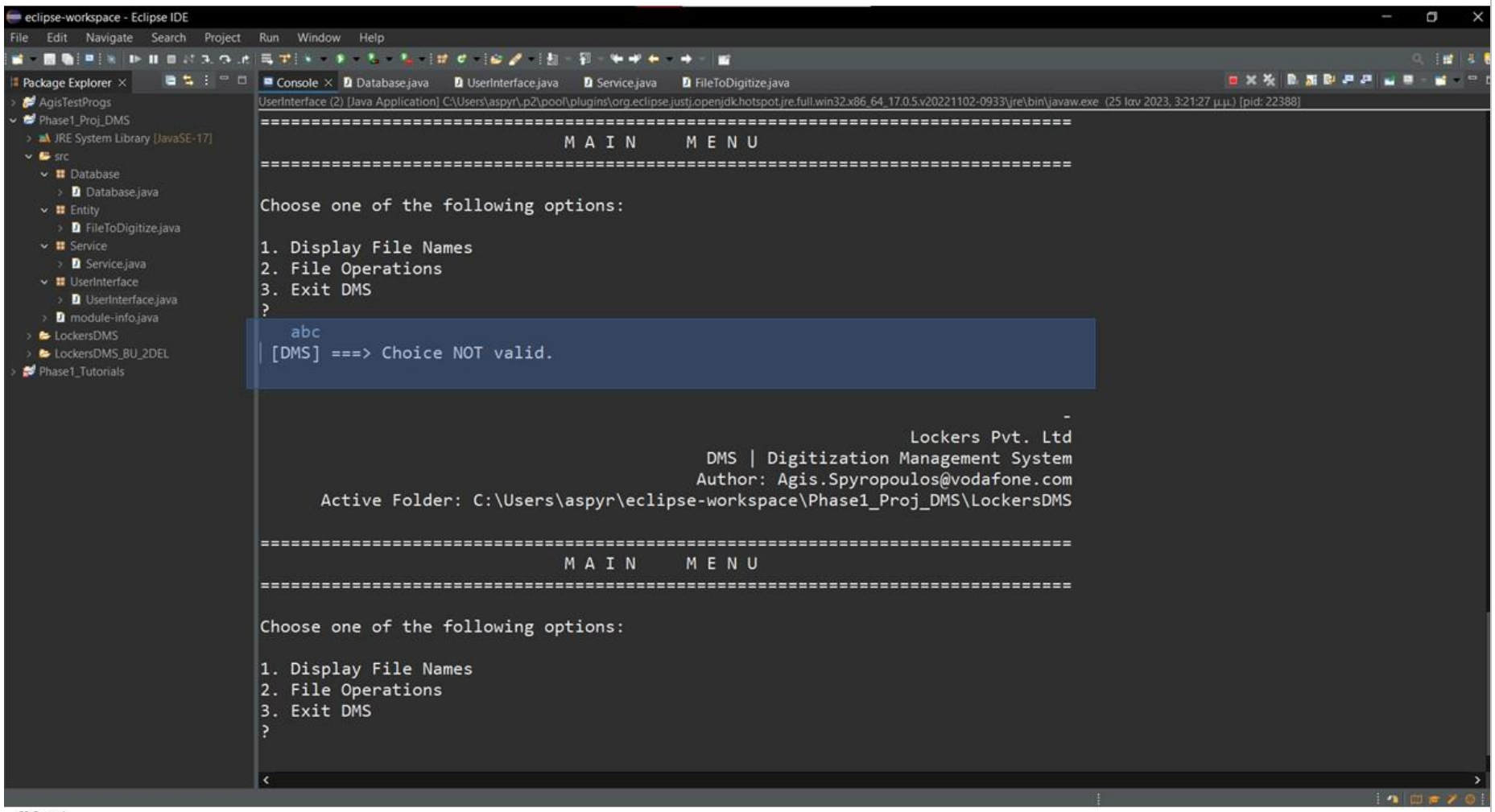
STEP-19: MAIN MENU, User Choice = 1.1, Choice not valid.



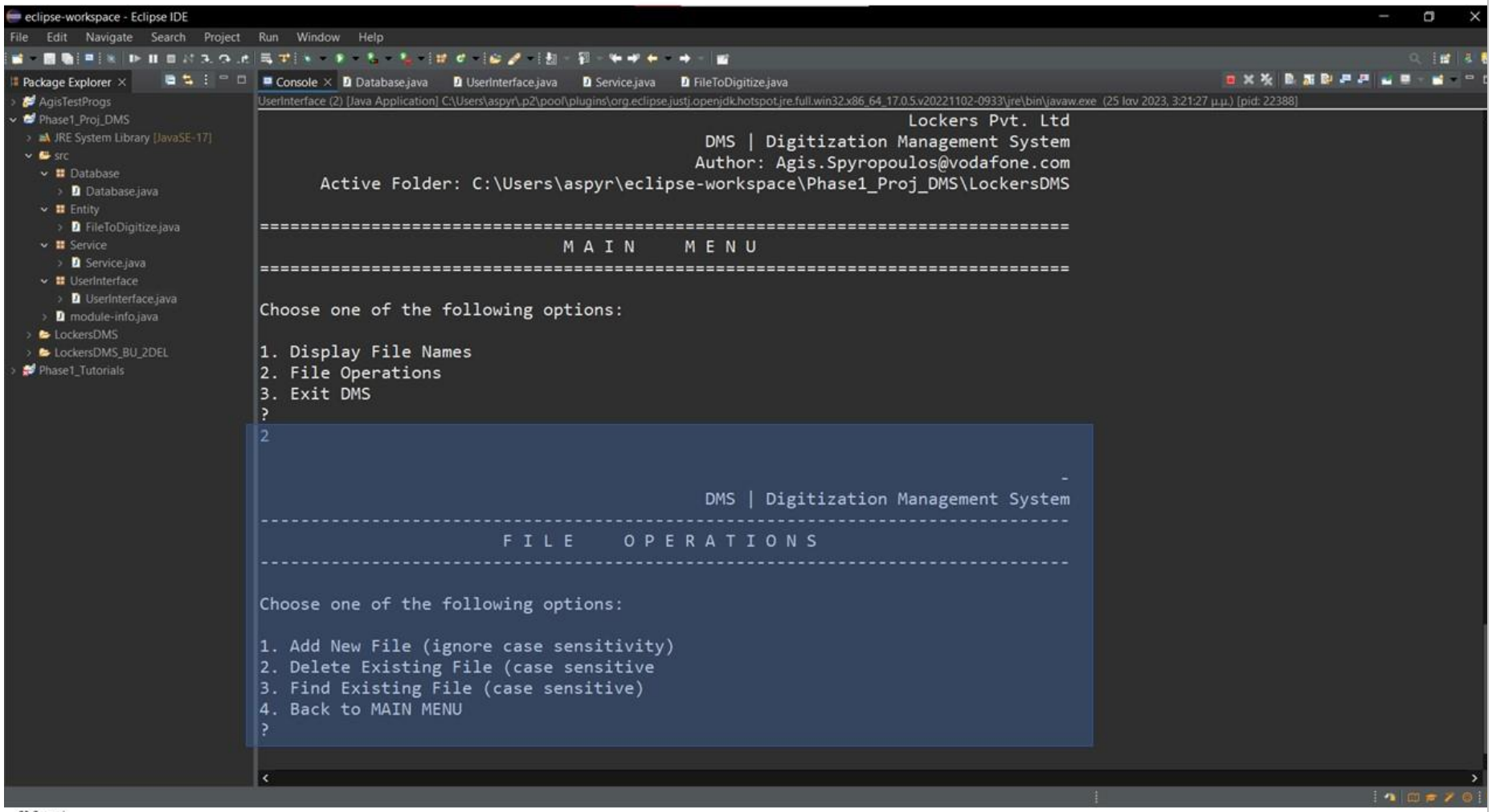
STEP-20: MAIN MENU, User Choice = 4, Choice not valid.



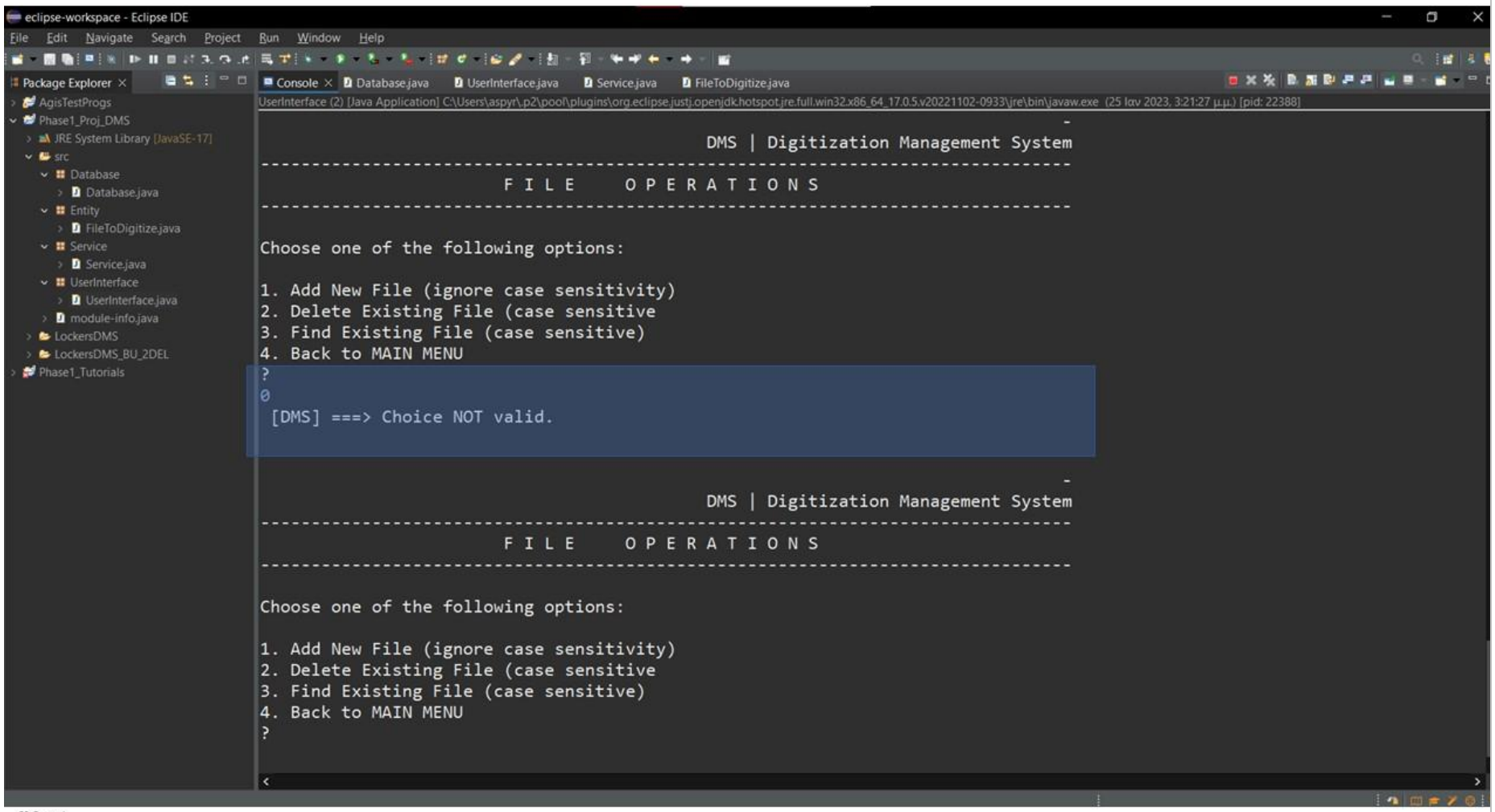
STEP-21: MAIN MENU, User Choice = [ENTER], Choice not valid.



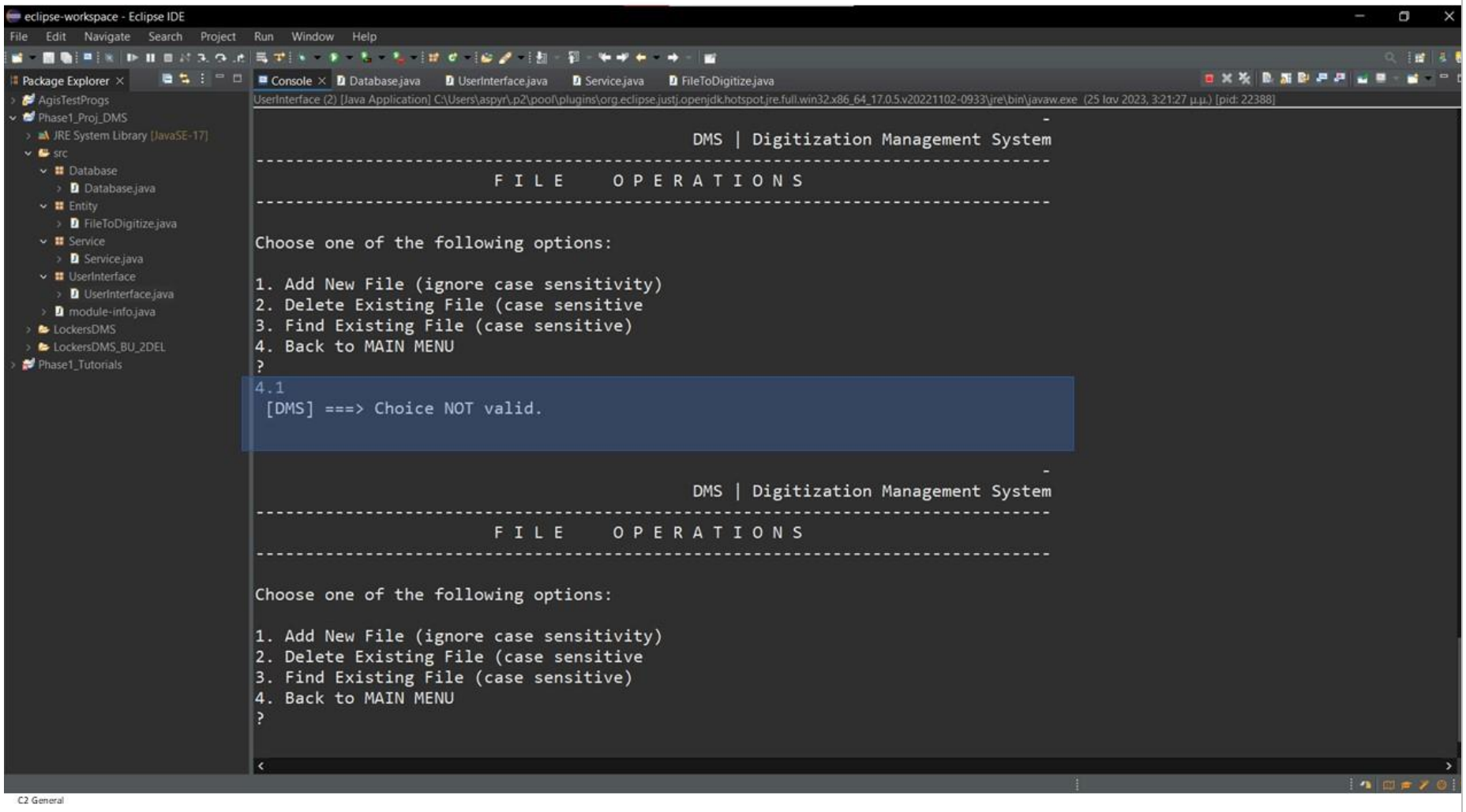
STEP-22: MAIN MENU, User Choice = [ANY_STRING], Choice not valid. In this case there are also preceding and trailing spaces.



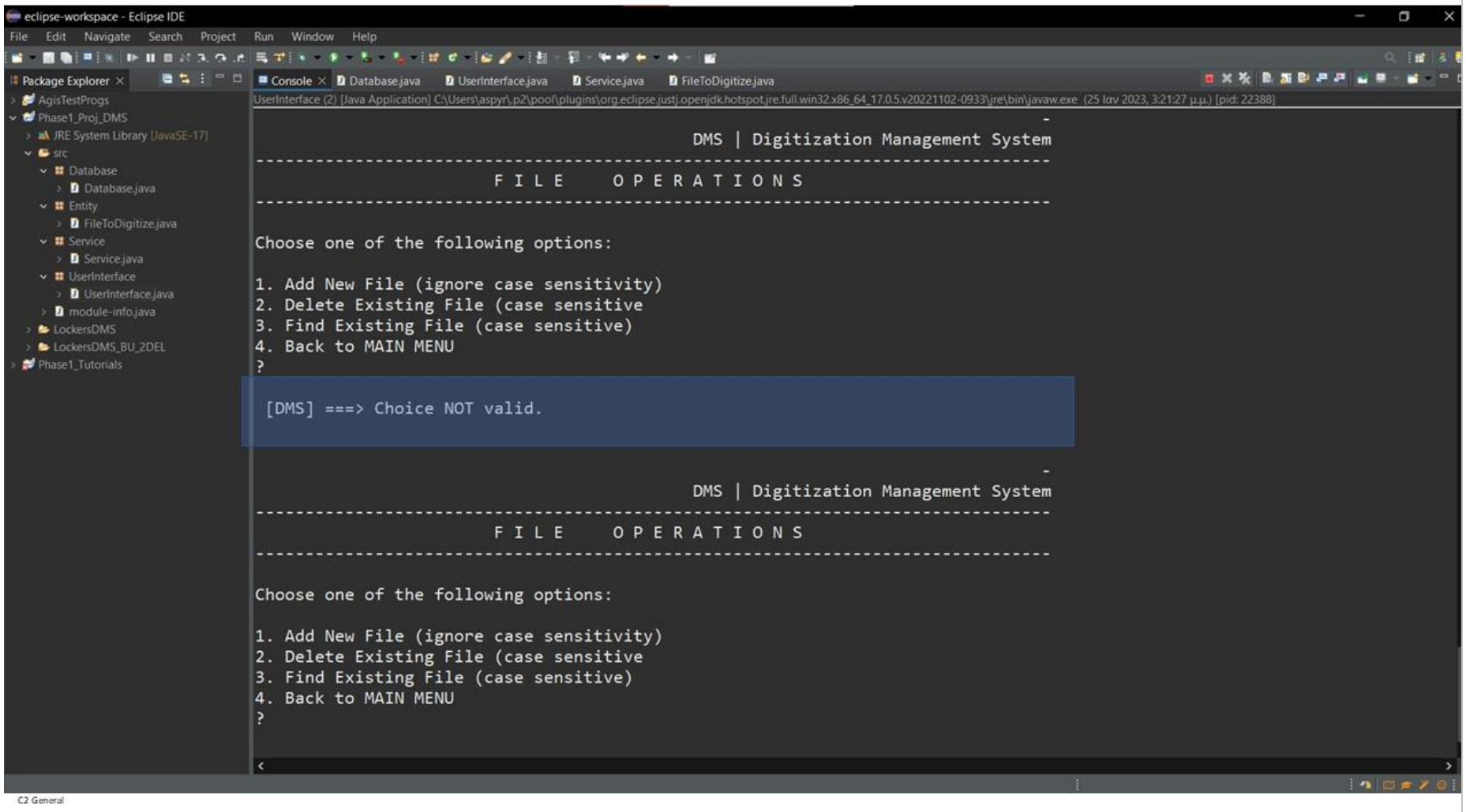
STEP-23: MAIN MENU, User Choice = 2, Display FILE OPERATIONS Menu.



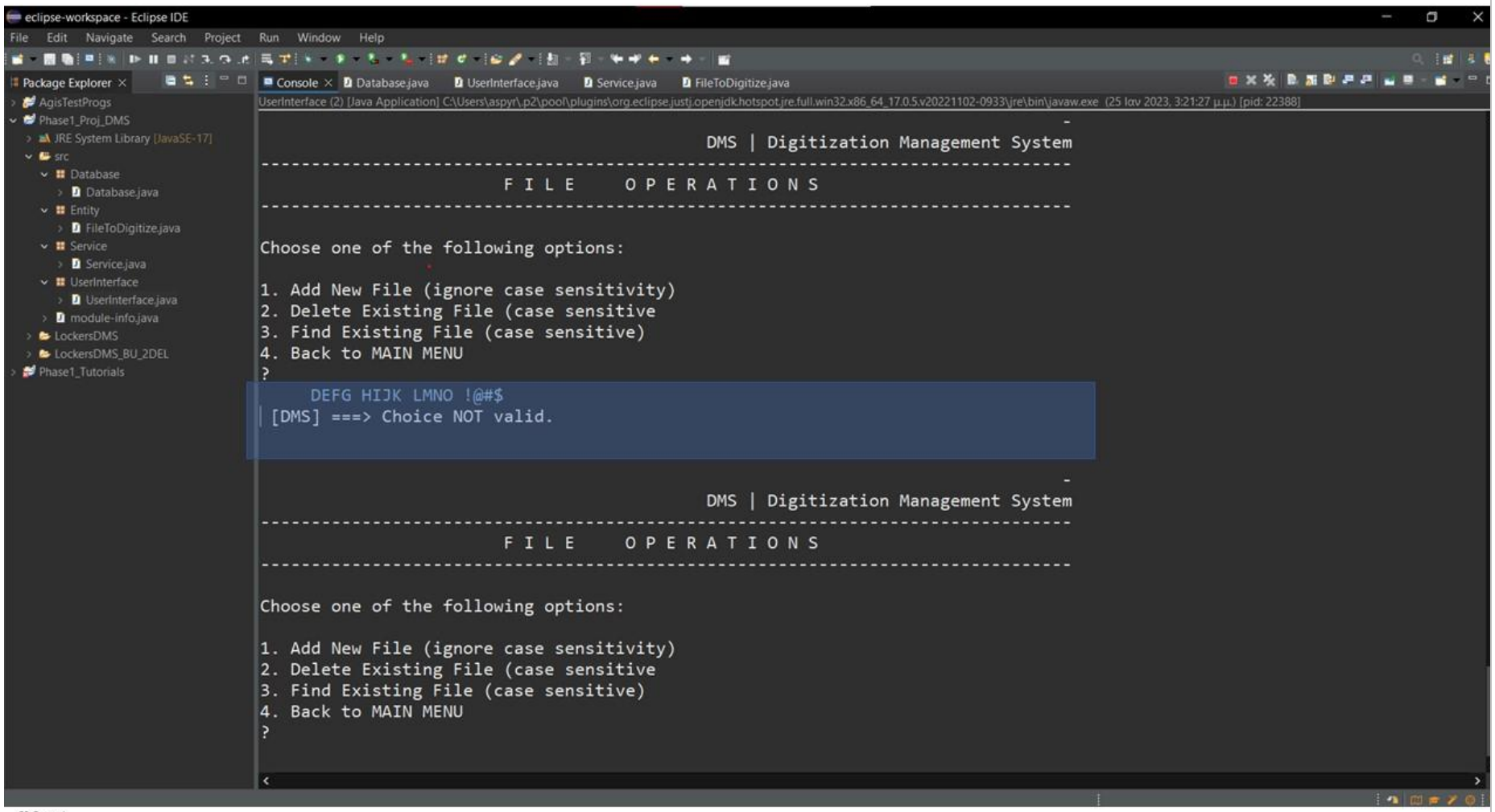
STEP-24: FILE OPERATIONS Menu, User Choice = 0, Choice not valid.



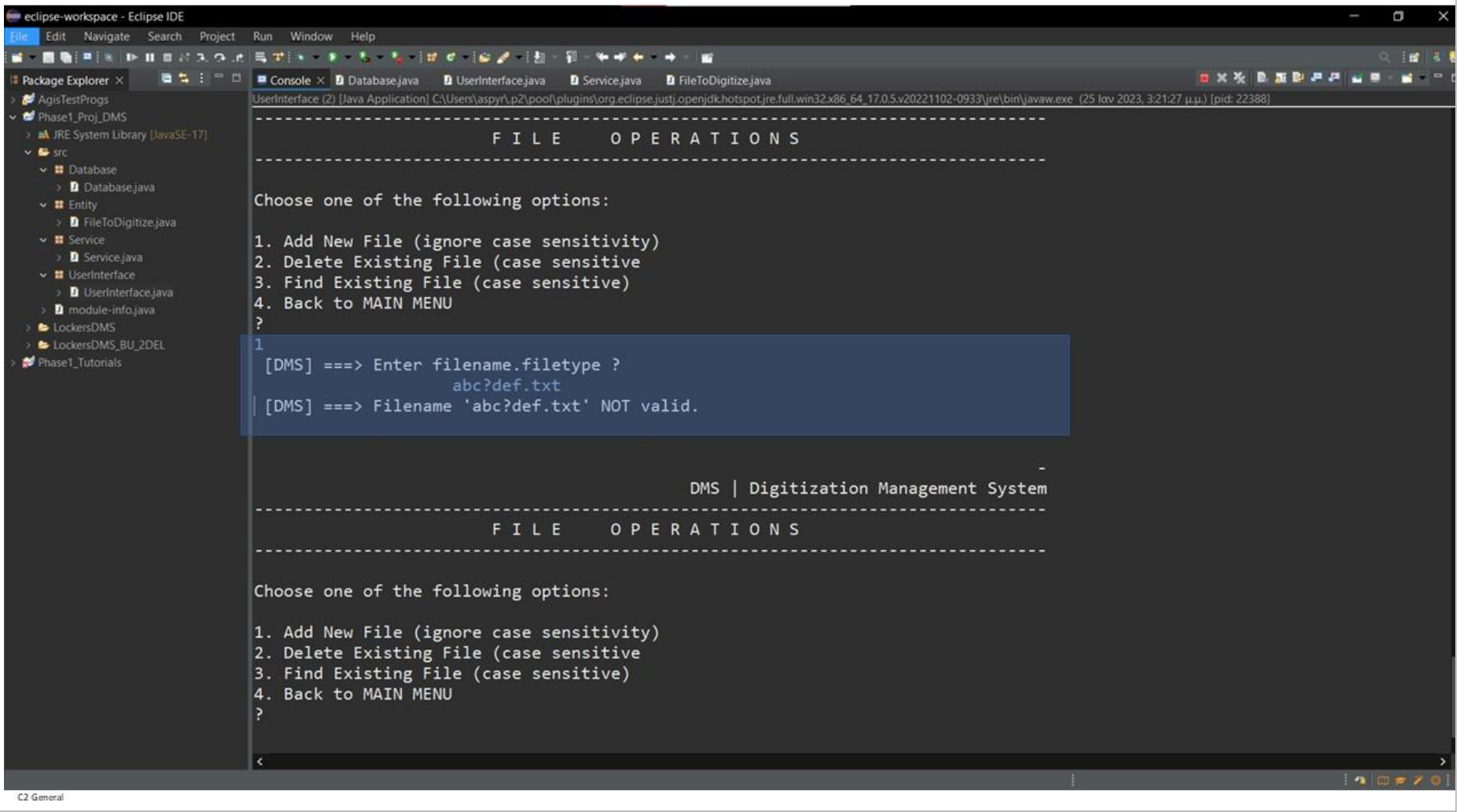
STEP-25: FILE OPERATIONS Menu, User Choice = 4.1, Choice not valid.



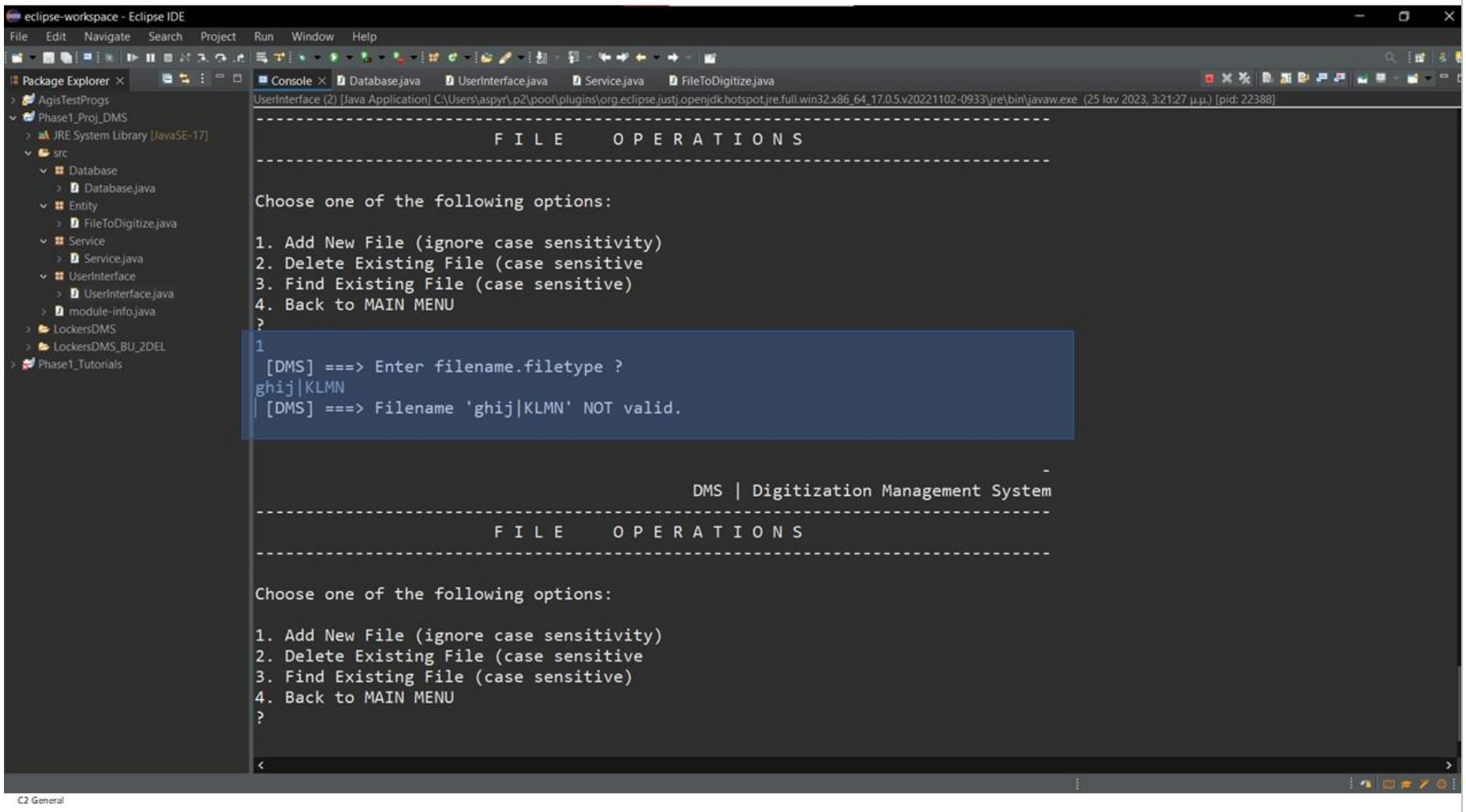
STEP-26: FILE OPERATIONS Menu, User Choice = [ENTER], Choice not valid.



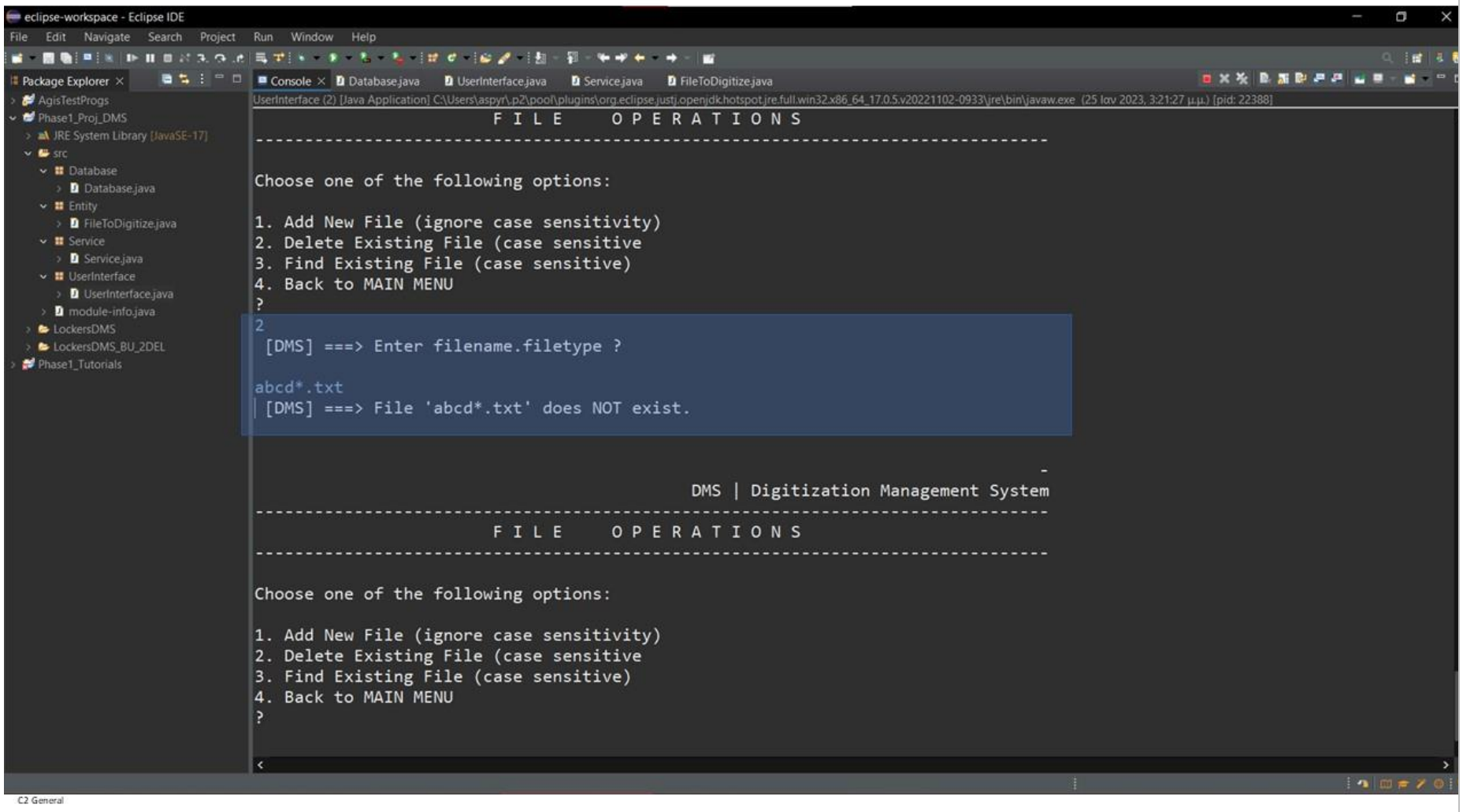
STEP-27: FILE OPERATIONS Menu, User Choice = [ANY_STRING], Choice not valid. Here the string is entered with preceding and trailing spaces.



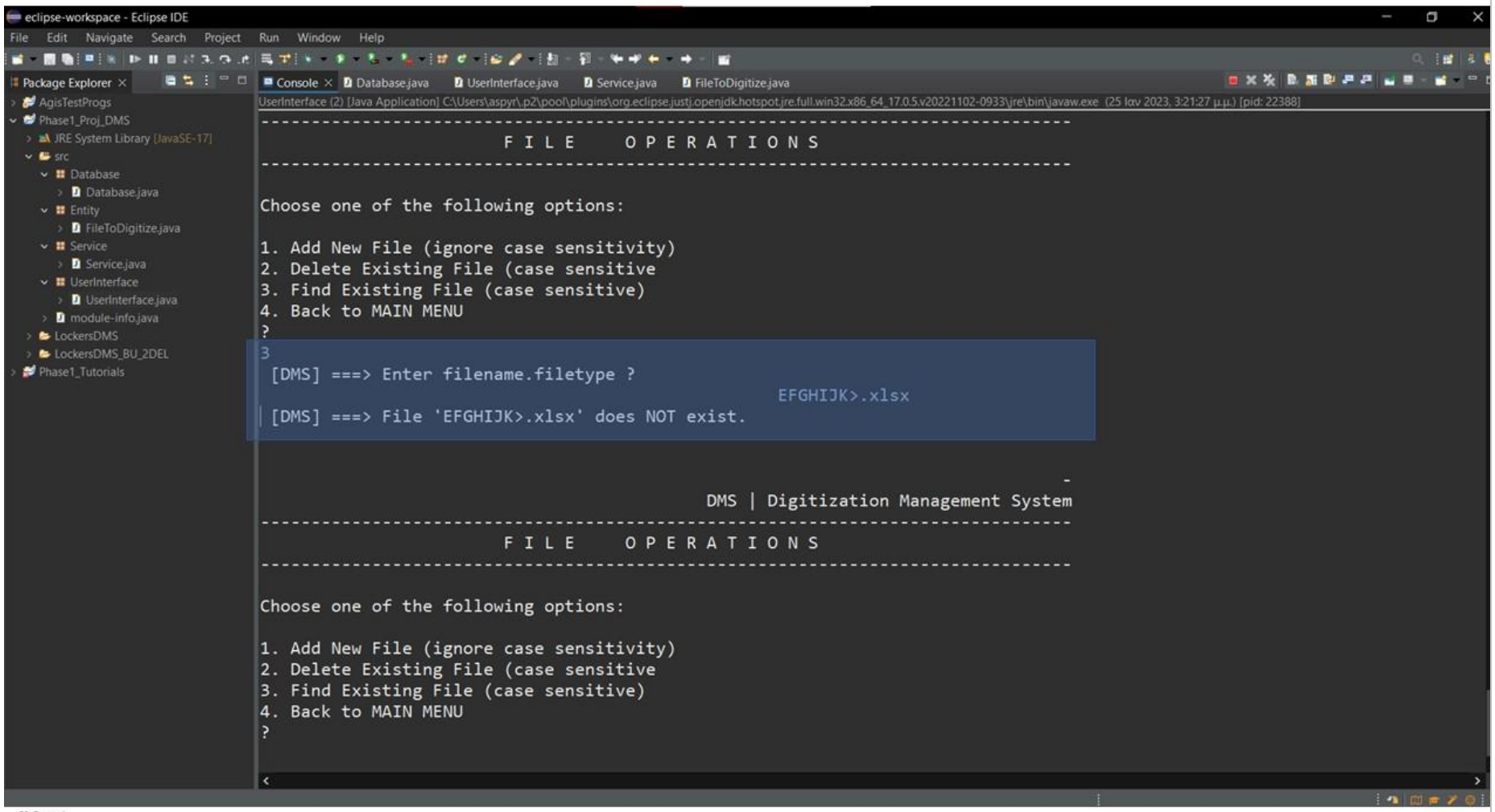
STEP-28 FILE OPERATIONS Menu, User Choice = 1, Add New File, filename is not valid. Here the string is entered with preceding and trailing spaces.



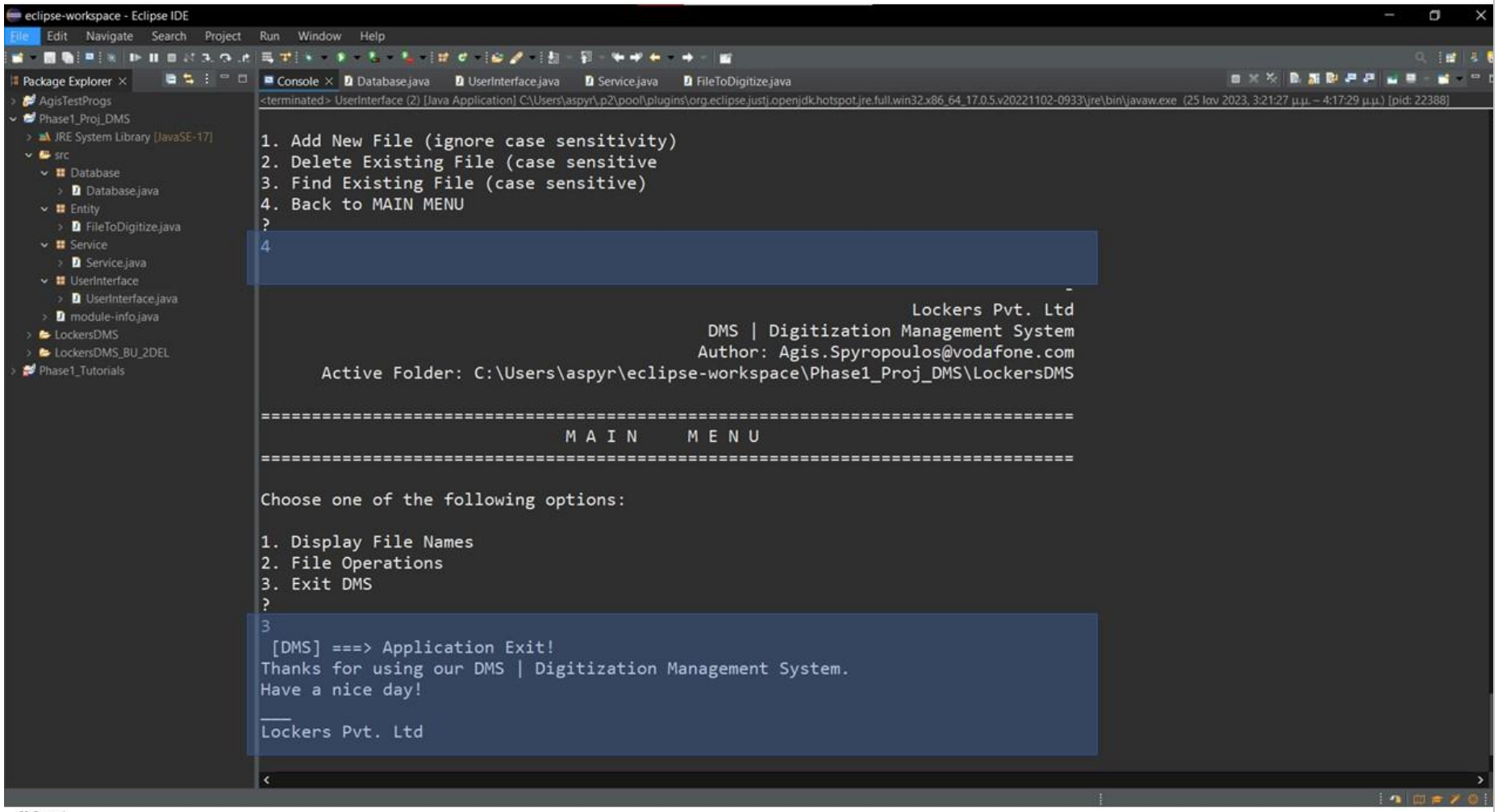
STEP-29: FILE OPERATIONS Menu, User Choice = 1, Add Another New File, filename is not valid.



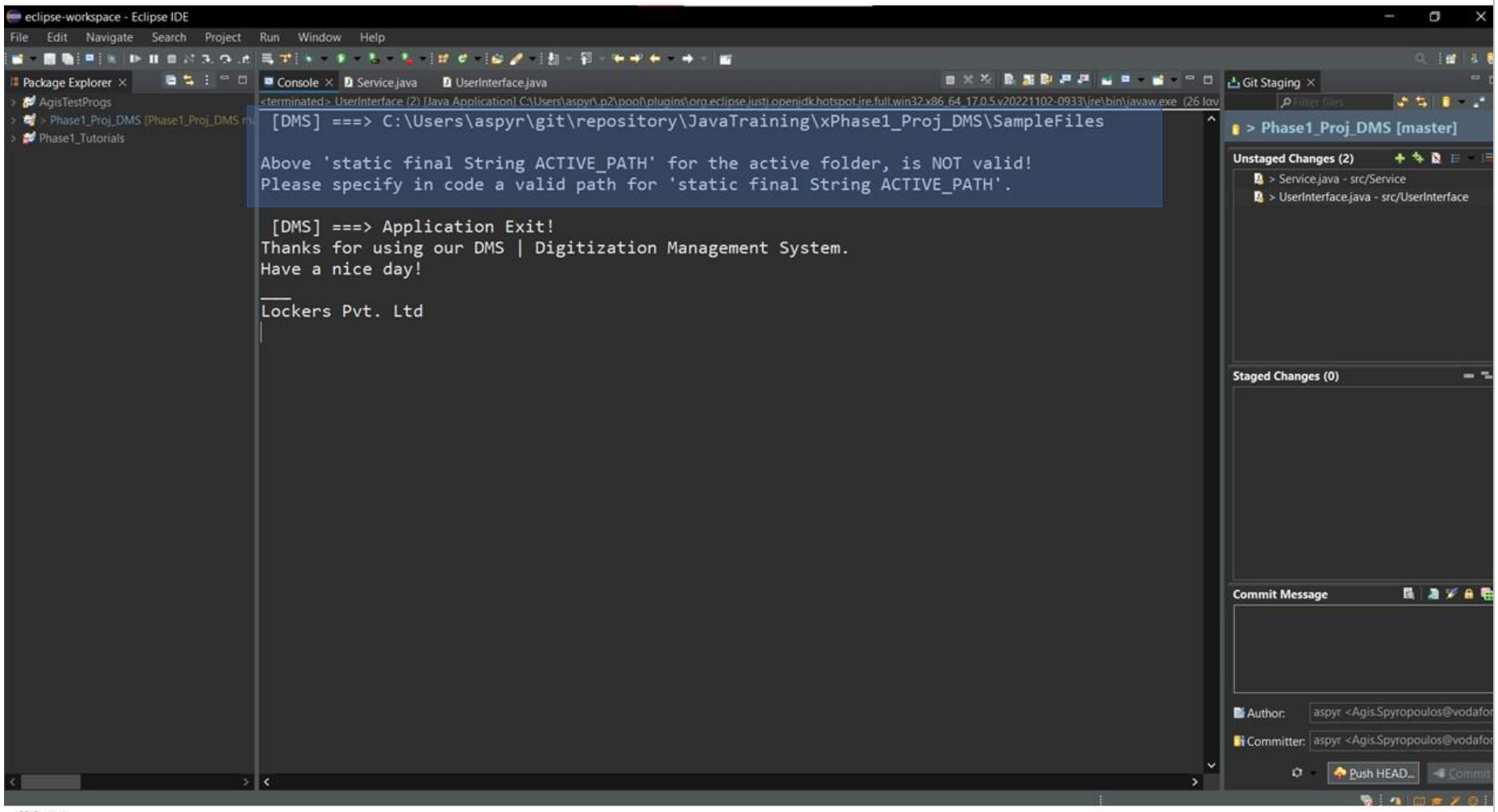
STEP-30: FILE OPERATIONS Menu, User Choice = 2, Delete Existing File, filename does not exist.



STEP-31: FILE OPERATIONS Menu, User Choice = 3, Find Existing File, filename does not exist. Here the string is entered with preceding and trailing spaces.



STEP-32: FILE OPERATIONS Menu, User Choice = 4, Back to MAIN MENU and then User Choice = 3, Exit DMS.



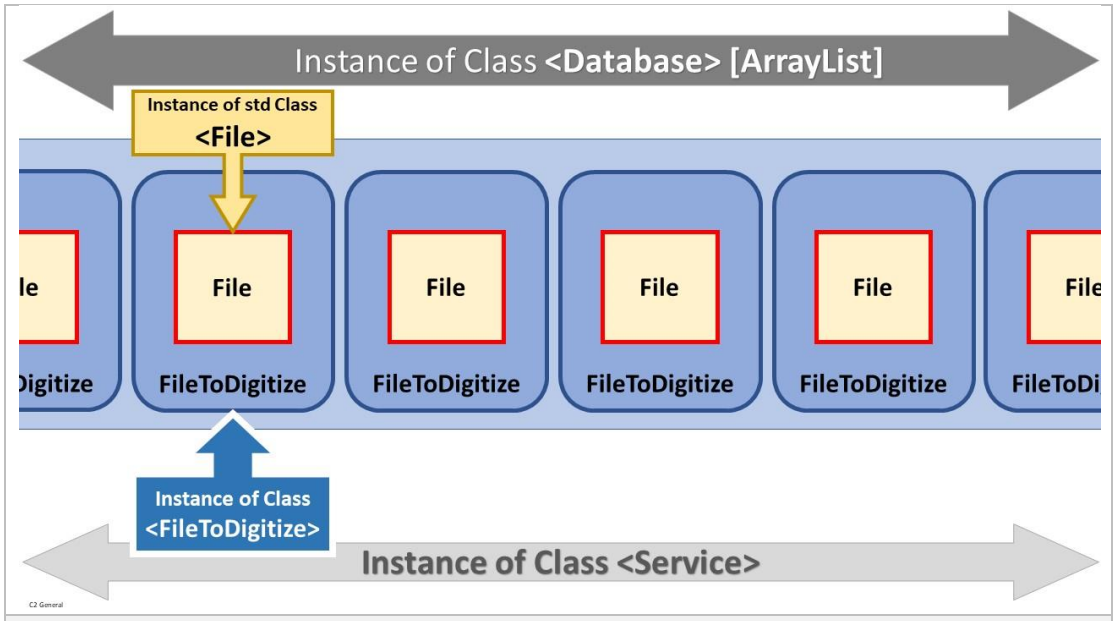
STEP-33: Case that a non-existing OR an invalid path name is initially pre-defined (set) as the DMS Active Folder (it has "…xPhase" instead of "…Phase").

Core Concepts Used in the Project

Entities

Class Name	Class Description
FileToDigitize	**ENTITY LAYER:** It is used to store the ***FileToDigitize*** objects. Contains the "file to be digitized" in the form of the standard ***File*** type.
Database	**DATABASE LAYER:** It is used to store the ***FileToDigitize*** objects in the form of an ***ArrayList***.
Service	**SERVICE LAYER:** It is adapted on the ***Database*** and used for the Business Logic of the project.
UserInterface	Stores the ***main*** program.

The relations between the entities above is depicted by the following figure:



Core concepts used for the implementation of the DMS Application.

Major Loop Structures

There will be two (2) major loop structures in the form of ***while***, one nested into the other (refer also to the application flowchart, presented in the chapter "*Algorithms & Flowcharts of the Application*"). The first one is related to the user choice concerning the MAIN MENU of the application and the second one is related to the user choice concerning the submenu FILE OPERATIONS, as described in the chapter "*Project DMS – Digitization Management System*".

Sorting Techniques

Sorting in the ***Database*** *ArrayList* is implemented via the ***Comparator*** interface using the **compareTo** function implemented for the entity ***FileToDigitize***.

Exception Handling | Validating Menu Choices

The user menu choices are checked in order to be valid integer numbers within the set of choices defined and allowed:

MAIN MENU has the following three (3) options:	FILE OPERATIONS menu has the following four (4) options:
▪ [1] Display File Names ▪ [2] File Operations ▪ [3] Exit DMS	▪ [1] Add New File ▪ [2] Delete Existing File ▪ [3] Find Existing File ▪ [4] Back to MAIN MENU

The DMS response to invalid user choices is presented by the screenshots in chapter "Algorithms & Flowcharts of the Application".

Note also that for the FILE OPERATIONS menu there has been also implemented a "hidden" choice (number 11) only for educational/testing purposes, as follows: User Choice 11 displays the files as they are currently ordered in the ***database***. This has been implemented in order to test that the objects ***FileToDigitize*** are unsorted[5] inside the ***database*** prior to using functionality "*Display File Names*" of the MAIN MENU.

[5] This of course assumes that some additions of new files (FILE OPERATIONS menu, user choice 1) have taken place before.

Exception Handling | Validating Filename Strings

While running the function which creates and inserts the file to the ***Database***, the filename string input by user is also checked for filename validity via the standard ***createNewFile*** method.

Links to the GitHub Repository

The whole project has been developed in the local repository "*C:\Users\aspyr\git\repository\JavaTraining\Phase1_Proj_DMS*" and it has been uploaded to the following public GitHub link:

https://github.com/aspyrop/Phase1_Proj_DMS.git

Full project documentation exists in the above remote repository, in folder "*Documentation*". Folder "*SampleFiles*" plays the role of the DMS active folder containing some initial (sample) files.

Conclusion on Enhancing the Application

The DMS Application could be enhanced if instead of console it is implemented as a web-based application, i.e. having a graphical user interface GUI so that user has full control on the requests. The active folder could then be selected by user, so there could be each time different packets of files for digitization.

APPENDIX I: Assessment Description as Provided by the Training Faculty

Virtual Key for Your Repositories

Course-end Project 1

DESCRIPTION

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 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.

The flow and features of the application:

- Plan more 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

You must use the following:

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop the prototype
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs

Following requirements should be met:

- The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.
- Document the step-by-step process starting from sprint planning to the product release.
- Application should not close, exit, or throw an exception if the user specifies an invalid input.
- You need to submit the final specification document which includes:

- Project and developer details
- Sprints planned and the tasks achieved in them
- Algorithms and flowcharts of the application
- Core concepts used in the project
- Links to the GitHub repository to verify the project completion
- Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)