**GitHub link: https://github.com/Saivinay08/Phase-1-LockedMeProject**

**LockedMe Application**

**Project:**

This application is able to take file input from user, delete those files when not required, able to search existing files and sort existing files in ascending order.

Product Backlog:

1. Display home page & developer details.

2. Details of User interface such as options display for user information

* To show all existing files in ascending order
* To add/delete/search files
* To close Application.

1. Features to accept the user input to select one of the options listed
2. 1st option should return the current file names in ascending order.

2nd option method called

3rd option -- To Exit application link with main menu & after completing operations.

1. 2nd option should return the details of the user interface such as options

displaying the following:

1. 2-1 option -- To add – it should show successful message once added
2. 2-2 option-- To delete- it should delete file & show successful message

-it should display message, if not deleted

8. 2-3 option -- To Search- it should search for enter file in directory - - it should display message, even if not available

1. 2-4 option-- To go back to main menu- option should be there to take to main menu

# Flowchart:

Display the welcome screen

1. Application name and developer details
2. User interface – Options for user interaction

Display options –

1. Add a file
2. Delete a file
3. Search a file
4. Back to main menu

Exit the application

Display file names in ascending order

Switch Case

User Input

A

1 2 3

Options for file operations

1. Create new File object with root directory path
2. Create an array of type File
3. Read all files from the directory to the array using listFiles()
4. Use sort function of arrays
5. Display the files using Println

Use System.exit(0) in Switch case

Switch Case

A

Search for a file

Delete a file

Back to main menu

Add a file

1 2 3 4

1. Prompt the user to enter file name to be searched
2. Convert the user input to File object
3. Use try block to search the file using exists()
4. Print success message if it exists else print file does not exist
5. Catch IOException
6. Prompt the user to enter file name to be deleted
7. Convert the user input to File object
8. Use try block to delete the file using delete()
9. Print success message if deleted else catch NoSuchFileException
10. Catch IOException
11. Prompt the user to enter file name to be created
12. Convert the user input to File object
13. Use try block to create the new file using createNewFile()
14. Print success message
15. Catch IOException

Java Concepts used:

1. Inheritance -- for using super class method in subclass directly.
2. Switch case – for selecting options.
3. Scanner—for getting input from console using default input system.in
4. Encapsulation- used getter & setter method for better code reusability.
5. IO stream File class- for creating, deleting, searching & sorting file type objects.
6. If-else condition- is used for getting conformation regarding file operations.
7. For each, arrays & sort are used to get file output in ascending order.

Sprint 1: Time – 3 Days

User Story 1: Display home page & developer details.

Task1 : create a class called Homepage & In that main method display application name & developer details by using System.out.println() command.

User Story 2: Details of User interface such as options display for user information.

* To show all existing files in ascending order
* To add/delete/search files
* To close Application.

Task 1 : create a new WelcomeOptions class & in that class create void MainContext() method.

Task 2: Using this system.out.println(), declare all 3 options required & call this method in main method. So that we can reuse this method again.

User story 3: Features to accept the user input to select one of the options listed.

Task 1: as we have 2 switch cases, code reusability. Create class InputCollector in that create also create integerinputmethod(), which contain scanner class can take input from user by default input receiver system.in as console.

Task 2: create also getter & setter method in Input collector class . call those getter method in

WelcomeOptions class contain WelcomePageselection() method.

Task 3: call this both method’s MainContext()- display user options & WelcomePageselection()- which takes input from console using scanner class.

Sprint 2: Time: 5 days

User Story 4: 1st option should return the current file names in ascending order.

Task 1: create a AscendingOrder class & create void display() method. Call the method in

WelcomeOptions class .

Task 2: in WelcomeOptions class create a switch case which able to perform options 1. Ascending order , option 2 operations for file & option 3 to close application.

Task3: In WelcomeOptions class under switch case option 1 paste this display() method.

Task 4: In WelcomeOptions class under switch case option 2 paste OperationSelect() method in Operationselector class for file change operations.

Task 5: In WelcomeOptions class under switch case option 3 contains, system.exit(0) method which helps to end application by system.out.println consisting message as thanks for using application.

User Story 5: 2nd option should return the details of the user interface such as options

displaying the following:

Task 1: create a operatationSeelector class with consists, from InputCollector class inputmethod() is used by calling object of that class.

Task 2 : Create a OptionSelect() method to display the available 3 options for add/delete/search.

Task 3: switch case also created with options 1, 2, 3 & 4 which consist of their respective method to perform operations.

User Story 6: 2-1 option -- To add – it should show successful message once added

Task 1: create AddFile class & create addfile() & call string input method with is already created input controller class.

Task 2: use File class object to create file & give input through getter method of string form InputController class.

Task 3: use If & else condition to create & conform file in root directory created or not with system.out.println() statement.

User Story 7: 2-2 option -- To delete – it should show successful message once added

Task 1: create DeleteFile class & create deletefile() & call string input method with is already created input controller class.

Task 2: use File class object to delete file & give input through getter method of string form InputController class.

Task 3: use If & else condition to delete & conform file in root directory created or not with system.out.println() statement.

User Story 8: 2-3 option -- To Search – it should show successful message once added

Task 1: create SearchFile class & create searchfile() & call string input method with is already created input controller class.

Task 2: use File class object to search file & give input through getter method of string form InputController class.

Task 3: use If & else condition to search & conform file in root directory created or not with system.out.println() statement.

User Story 9: 2-4 option -- To go back to main menu

Task 1: create exit class & create exitoption() method. Also can give switch case to display 2 options.

Task 2: option 1 is to create object & call welcomeOptions class MainContent() method--display options & welcome Pageselection() method to select the page options again.

Task 3: option 2 contains, system.exit(0) method which helps to end application by system.out.println() consisting message as thanks for using application.