virtual-key-repository

Project name: LockedMe.com

Project github link:
https://github.com/beastking1234/Phaseoneend

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 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:

1. Retrieving the file names in an ascending order

2. 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

4. Option to close the application.

Project objective:

Java Console based program allows users to input menu options for:

1) Displaying File/Folder structure.

2) Search for File/Folder recursively.

3) Add/Delete File/Folder recursively.

Component Hierarchy:

LockedMeMainProgram.java :

This program contains the Main method. It's the entry point of the program.

FileMenuOption.java :

This program contains methods for displaying menu options.

FileHandleOption.java :

This program contains methods for handling different options from user input and calls respective methods from File Operations.

FileOperation.java :

This program defines all the operations required for displaying, searching, deleting, and adding files/folder.

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 the 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

Implemented the concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

Sprints planning and Task completion

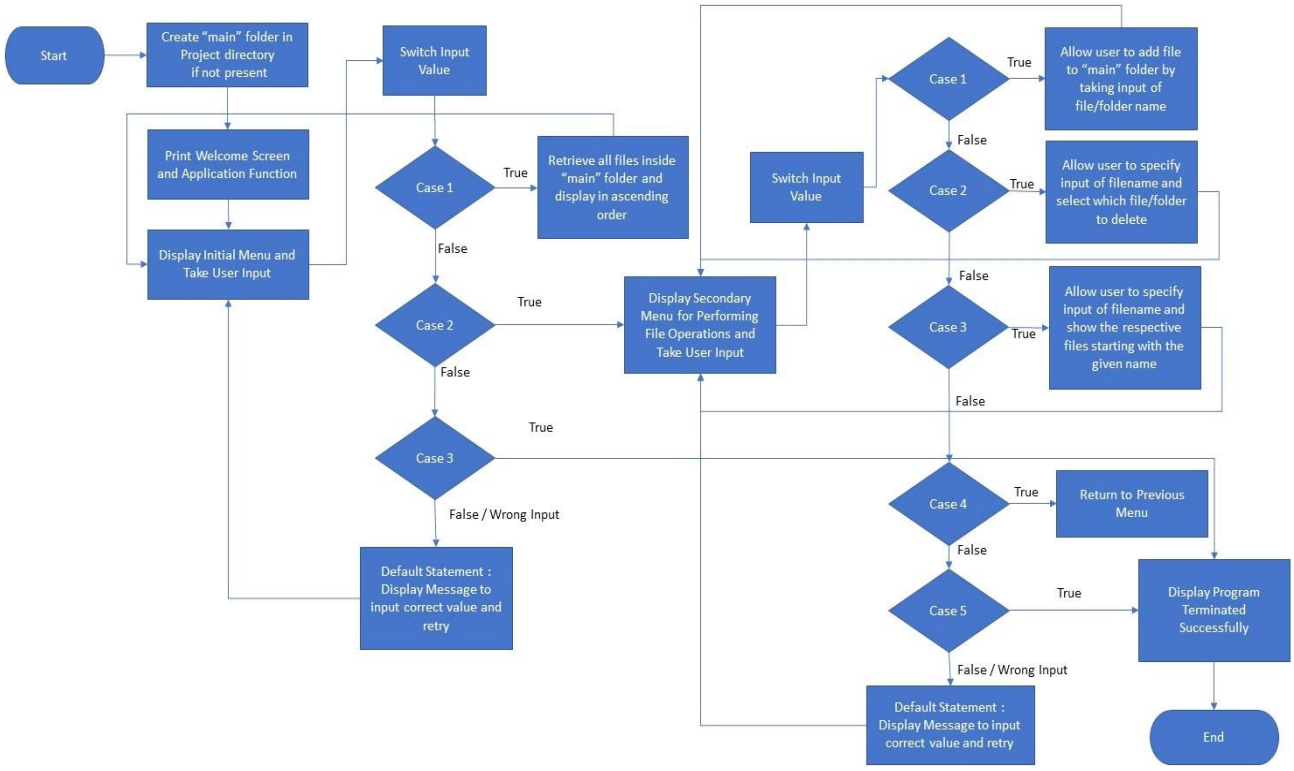
The project is planned to be completed in 2 sprint. Tasks assumed to be completed in the sprint are:

- Creating the flow of the application
- Initializing git repository to track changes as development progresses.
- Writing the Java program to fulfill the requirements of the project.
- Testing the Java program with different kinds of User input
- Pushing code to GitHub.

Core concepts used in project

Collections framework, File Handling, Sorting, Exception Handling.

Flow of Application:


Start
Create "main" folder in Project directory if not present
Print Welcome Screen and Application Function
Display Initial Menu and Take User Input
Switch Input Value
Case 1
True
Retrieve all files inside "main" folder and display in ascending order
False
Case 2
True
Display Secondary Menu for Performing File Operations and Take User Input
False
Case 3
True
False / Wrong Input
Default Statement : Display Message to input correct value and retry
Switch Input Value
Case 1
True
Allow user to add file to "main" folder by taking input of file/folder name
False
Case 2
True
Allow user to specify input of filename and select which file/folder to delete
False
Case 3
True
Allow user to specify input of filename and show the respective files starting with the given name
False
Case 4
True
Return to Previous Menu
False
Case 5
True
Display Program Terminated Successfully
False / Wrong Input
Default Statement : Display Message to input correct value and retry
End

Project Steps:

Step 1: Creating a new project in Eclipse

Step 2: Writing a program in Java for the entry point of the application (**LockedMeMain.java**).

Step 3: Writing a program in Java to display Menu options available for the user (**FileMenuOptions.java**).

Step 4: Writing a program in Java to handle Menu options selected by user (**HandleOptions.java**)

Step 5: Writing a program in Java to perform the File operations as specified by user (**FileOperations.java**)

Step 6: Pushing the code to GitHub repository