**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
    - Option to Sort the files in Ascending Order
    - 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
  + Sort all the files in an ascending order
  + 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

**Developer Details**

**Name:** Tejaswini Arun Mahore

**Email ID:**[tejaswinimahore@gmail.com](mailto:tejaswinimahore@gmail.com)

**Cohort Detail:**JG FSJD Nov 2022 COHORT 1

**GitHub Link (of the application):** https://github.com/tejaswinimahore/LockedMe.comApp.git

**Sprint Details**

**Sprint 1:**

**Sprint Planning:** Decide upon the workflow of the application and the basic algorithms involved.

**Sprint Review:** Defined the working structure of the application and decided to use an array to store data and utilise bubble sort algorithm to retrieve the data.

**Sprint Retrospective:** Successfully completed the targets of the sprint and built a good base for the next sprint.

**Sprint 2:**

**Sprint Planning:** Program the entire application based on the structural workflow decided upon in the previous sprint.

**Sprint Review:** Created the application in two separate classes. One of the classes comprises the source code of the application. The second class involves the interface of the application.

**Sprint Retrospective:** The sprint was completed within the timeframe albeit the application being riddled by a few bugs.Consequently, the entirety of the next sprint would go into debugging the code and pushing it to GitHub.

**Sprint 3:**

**Sprint Planning:** Debug the application and push it to GitHub.

**Sprint Review:** The application is working as expected now. Post the successful debugging of the application, it was reviewed again and then pushed to GitHub using Git.

**Sprint Retrospective:** Although the application is running in an ideal manner now, this entire sprint could have been avoided if not for the bugs. The only practical progress in this sprint was pushing the application to GitHub.

**Sprint 4:**

**Sprint Planning:** Create documentation for the entire process of programming the application,

**Sprint Review:** Gathered data and documented the entire process spanning the algorithms and the workflow involved in the application up to the sprints planned for programming the same.

**Sprint Retrospective:** The documentation of the entirety of the processes involved took more time than ideally expected. This sprint could not have been clubbed with the previous sprint even if no bugs were encountered in the application.

**Core Concepts Used**

There were several core concepts used in the project, specified as follows:

**Exception Handling:**One of the several core concepts involved in this application was the concept of exception handling. In the case of this application, the exception handling primarily involved utilising a try-catch block to return the output of “Wrong choice entered” if the user chose a number other than 1 to 6 for an option. Two more try-catch blocksare also used to return outputs to inform the user in the cases if they tried to delete a file that do not exist or create a file that already exists in the directory.

**Data Structures and Algorithms:**Another major core concept used in the making of the application is the usage of an array of string variables to show the files that are present in the selected directory, including the new files that are being created using the application. Also, a bubble sort algorithm is used to present the files in an ascending order in the output when the user chooses the concerned option.

**Switch Cases:**A switch case is used to give user the requisite 6 options to choose from that the premise of the application requires us to give them, namely the options to add, delete, search or list existing file, return to main context and exit the application.

**Inheritance:**One of most the vital core concept used is this case Inheritance. In this application, all the actions required in the switch cases such as retrieving, adding, or deleting a file has its actions defined in the class of “SourceCode”, and then it is called class of “LockedMe” in a switch case, with each specified case inheriting the requisite class.

**Pre-Built Classes:**In this application, we also utilize the pre-built classes that Java provides to programmers. Particularly, we used a pre-built class called Scanner class to read the user input for the corresponding choice that they have selected, forming a base for an interactive I/O of the application.

**Data Types:** A majorly overlooked core concept used in this application is that of the various data types. A Boolean data type is used in one of try-catch blocks for handling exceptions and string variables are used to store the data of file names in the arrays. Furthermore, integer data type is used to register the choice input of the user while navigating the main context of the application.

**Conclusion**

The application can be enhanced further with the following features:

1. It can be made to support all types of file formats such as pictures, word documents, basic text documents, videos, other applications, etcetera.
2. By adding a feature to display the files only of a particular extension such as .jpeg or .xlsx, it can help a user narrow down their search to very specific targets.
3. Give the application a certain degree of autonomy, such as allowing it to create different directories for separate file extensions or archiving all deleted files for temporary storage would increase the potential use-cases of the application.
4. Build the application over a private cloud instance and allow only a select group of related users to share a storage serverfor the data isolation of sensitive files.
5. Allow the users to encrypt the pre-existing existing files in the selected directory and new files that they are adding from a range of relevant cryptographic algorithms with the keys being downloaded only in an external hard drive bearing the relevant cryptographic tokens.
6. Allows users to analyse their usage of storage by profiling the types and frequency of files present, their usage patterns and other behavioural parameters and automatically archives files not used for more than a certain period of time as defined by the user.

Potential unique selling points (USPs) of the application are as follows:

1. Ease of use and wide range of applications.
2. All kinds of files can be tracked and managed via a single interface for multiple directories and the various pre-existing folders/files within them.
3. Minimalistic yet interactive design.
4. Gives users complete control and privacy over their files.
5. Grants them access to hidden files on their computers/storage which would otherwise not be possible.
6. Allows users to actively manage their storage and keep their directories clean and in an orderly manner.