

**MOHAMMED ABUTALIB A**

**a.mohammedabutalib@mphasis.com**

## **Virtual Key for Repositories**

### **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.

**This document contains sections for:**

- Sprint planning and Task completion
- Core concepts used in project
- Demonstrating the product capabilities, appearance, and user interactions.
- Unique Selling Points of the Application
- Conclusions

The code for this project is hosted at: <https://github.com/Abutalib15/JAVA-FSD/tree/main/Phase-1/Final%20Project/Virtual%20Key%20Repository>

The project is developed by Mohammed Abutalib A.

**Sprints planning and Task completion:**

The project is planned to be completed in 1 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.
- Creating this specification document highlighting application capabilities, appearance, and user interactions.

**Core concepts used in project:**

Collections framework, File Handling, Sorting, Flow Control, Recursion, Exception Handling, Streams API .

**Demonstrating the product capabilities, appearance, and user interactions:**

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

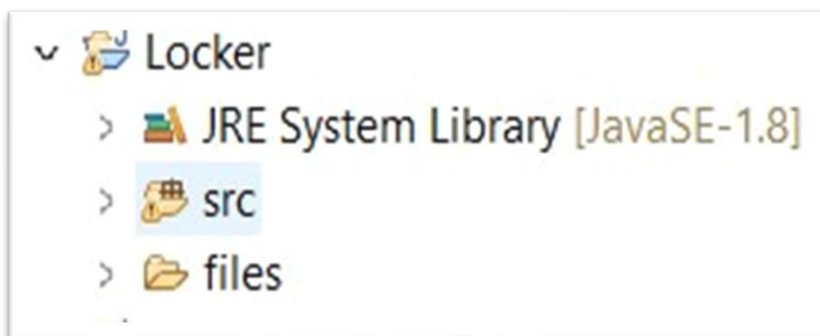
1. Creating a New project in Eclipse.
2. Write a program is java program ([Locker.java](#)).

## Step 1: Creating a new project in Eclipse

- GOTO Files Select New Java Project.
- Add source file to the project.
- Create a Class file (Locker.java).

## Step 2: Writing a program in Java application (Locker.java).

- Creating a Public main class for the Class file.
- Creating Directory file to Save, Delete, Add.



- Writing a Method For displaying Welcome screen

```
Problems @ Javadoc Declaration Console × Tasks
Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v2022051
**** lockers.pvt****
DIRECTORY : D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

MAIN MENU - Select any of the following:
1 -> List files in directory
2 -> Add, Delete or Search
3 -> Exit Program
|
```

- Writing a Method for Initiate Main Menu.

```
MAIN MENU - Select any of the following:
1 -> List files in directory
2 -> Add, Delete or Search
3 -> Exit Program
```

- Writing a method for Secondary Menu.

2

Select any of the following:

- a -> Add a file
- b -> Delete a file
- c -> Search a file
- d -> GoBack

- Write a method to create file or folder by user

```

Problems @ Javadoc Declaration Console × Tasks
Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-14
Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
a
↳ Adding a file...Please Enter a File Name : sample
File sample added to D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

```

- Write a method to Delete A file.

```

Problems @ Javadoc Declaration Console × Tasks
Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre
Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
b
↳ Deleting a file...Please Enter a File Name : sample
File sample deleted from D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

```

- Write a method to Search a File in created directory

```

Problems @ Javadoc Declaration Console × Tasks
Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\
Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
c
↳ Searching a file...Please Enter a File Name : sample
FOUND : File sample exists at D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

```

- Write a method to go back to the Main menu.

```
Select any of the following:
```

```
a -> Add a file  
b -> Delete a file  
c -> Search a file  
d -> GoBack
```

```
d
```

```
Going Back to Main Menu
```

```
MAIN MENU - Select any of the following:
```

```
1 -> List files in directory  
2 -> Add, Delete or Search  
3 -> Exit Program
```

- Write the Print Statement when exiting the Program.

```
MAIN MENU - Select any of the following:
```

```
1 -> List files in directory  
2 -> Add, Delete or Search  
3 -> Exit Program
```

```
3
```

```
Thank You
```

### **Source code:**

```
package virtualKey;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
public class Locker {
```

```
    static String DIRECTORY;
```

```
    File folder_name;
```

```

public Locker() {

    DIRECTORY = System.getProperty("user.dir");

    folder_name = new File(DIRECTORY+"/files");

    if (!folder_name.exists())

        folder_name.mkdirs();

    System.out.println("DIRECTORY : "+
folder_name.getAbsolutePath());

}

private static final String WELCOME_PROMPT =

    "\n***** Locker.pvt*****\n";

private static final String MAIN_MENU_PROMPT =

    "\nMAIN MENU - Select any of the following: \n"+

        "1 -> List files in directory\n"+

        "2 -> Add, Delete or Search\n"+

        "3 -> Exit Program";

private static final String SECONDARY_MENU_PROMPT =

    " \nSelect any of the following: \n"+

        " a -> Add a file\n"+

        " b -> Delete a file\n"+

        " c -> Search a file\n"

```

```
    " d -> GoBack";
```

```
void showPrimaryMenu() {  
    System.out.println(MAIN_MENU_PROMPT);  
    try{  
        Scanner scanner = new Scanner(System.in);  
        int option = scanner.nextInt();  
        switch (option){  
            case 1 : {  
                showFiles();  
                showPrimaryMenu();  
            }  
            case 2 : {  
                showSecondaryMenu();  
            }  
            case 3 : {  
                System.out.println("Thank You");  
                System.exit(0);  
            }  
            default: showPrimaryMenu();  
        }  
    }  
}
```

```
        catch (Exception e){

            System.out.println("Please enter 1, 2 or 3");

            showPrimaryMenu();

        }

    }
```

```
void showSecondaryMenu() {

    System.out.println(SECONDARY_MENU_PROMPT);

    try{

        Scanner scanner = new Scanner(System.in);

        char[] input =

scanner.nextLine().toLowerCase().trim().toArray();

        char option = input[0];

        switch (option){

            case 'a' : {

                System.out.print("â†³ Adding a file...Please Enter a File Name :

");

                String filename = scanner.next().trim().toLowerCase();

                addFile(filename);

                break;

            }

        }

    }
```



```
        case 'b' : {

            System.out.print("â†³ Deleting a file...Please Enter a File Name

: ");

            String filename = scanner.next().trim();

            deleteFile(filename);

            break;

        }

        case 'c' : {

            System.out.print("â†³ Searching a file...Please Enter a File

Name : ");

            String filename = scanner.next().trim();

            searchFile(filename);

            break;

        }

        case 'd' : {

            System.out.println("Going Back to Main Menu");

            showPrimaryMenu();

            break;

        }

        default : System.out.println("Please enter a, b, c or d");

    }

    showSecondaryMenu();
```

```

    }

    catch (Exception e){

        System.out.println("Please enter a, b, c or d");

        showSecondaryMenu();

    }

}

void showFiles() {

    if (folder_name.list().length==0)

        System.out.println("The folder is empty");

    else {

        String[] list = folder_name.list();

        System.out.println("The files in "+ folder_name +" are :");

        Arrays.sort(list);

        for (String str:list) {

            System.out.println(str);

        }

    }

}

void addFile(String filename) throws IOException {

    File filepath = new File(folder_name + "/" + filename);

```

```
String[] list = folder_name.list();

for (String file: list) {

    if (filename.equalsIgnoreCase(file)) {

        System.out.println("File " + filename + " already exists at " +
folder_name);

        return;

    }

}

filepath.createNewFile();

System.out.println("File "+filename+" added to "+ folder_name);

}
```

```
void deleteFile(String filename) {

    File filepath = new File(folder_name + "/" + filename);

    String[] list = folder_name.list();

    for (String file: list) {

        if (filename.equals(file) && filepath.delete()) {

            System.out.println("File " + filename + " deleted from " +
folder_name);

            return;

        }

    }

}
```

```
        System.out.println("Delete Operation failed. FILE NOT FOUND");  
    }  
  
    void searchFile(String filename) {  
        String[] list = folder_name.list();  
        for (String file: list) {  
            if (filename.equals(file)) {  
                System.out.println("FOUND : File " + filename + " exists at " +  
folder_name);  
                return;  
            }  
        }  
        System.out.println("File NOT found (FNF)");  
    }  
  
    public static void main(String[] args) {  
  
        System.out.println(WELCOME_PROMPT);  
  
        Locker menu = new Locker();  
        menu.showPrimaryMenu();  
    }  
}
```

**Overall output:**

```
Problems Javadoc Declaration Console × Tasks
Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\

Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
a
↳ Adding a file...Please Enter a File Name : sample
File sample added to D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
b
↳ Deleting a file...Please Enter a File Name : sample
File sample deleted from D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files
```

```
Problems Javadoc Declaration Console × Tasks
<terminated> Locker [Java Application] C:\Users\DELL\Downloads\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (06-Aug-2

Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
c
↳ Searching a file...Please Enter a File Name : sample
FOUND : File sample exists at D:\Users\DELL\Documents\workspace-spring-tool-suite-4-4.15.1.RELEASE\Virtual Key\files

Select any of the following:
a -> Add a file
b -> Delete a file
c -> Search a file
d -> GoBack
d
Going Back to Main Menu

MAIN MENU - Select any of the following:
1 -> List files in directory
2 -> Add, Delete or Search
3 -> Exit Program
3
Thank You
```

### Step 3: Pushing the code to GitHub repository

- Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

- Initialize repository using the following command:

**git init**

- Add all the files to your git repository using the following command:

**git add .**

- Commit the changes using the following command:

**git commit . -m <commit message>**

- Push the files to the folder you initially created using the following command:

**git push -u origin master**

### **Unique Selling Points of the Application:**

1. The application is designed to keep on running and taking user inputs even after exceptions occur. To terminate the application, appropriate option needs to be selected.
2. The application can take any file/folder name as input. Even if the user wants to create nested folder structure, user can specify the relative path, and the application takes care of creating the required folder structure.
3. User is also provided the option to write content if they want into the newly created file.
4. The application doesn't restrict user to specify the exact filename to search/delete file/folder. They can specify the starting input, and the program searches all files/folder starting with the value and displays it. The user is then provided the option to select all files or to select a specific index to delete.
5. The application also allows user to delete folders which are not empty.

6.The user is able to seamlessly switch between options or return to previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed.

**Conclusion:**

Further enhancements to the application can be made which may include:

- Conditions to check if user is allowed to delete the file or add the file at the specific locations.
- Asking user to verify if they really want to delete the selected directory if it's not empty.
- Retrieving files/folders by different criteria like Last Modified, Type,