

LockedMe.com

12.08.2021

Anil Ganguly Sahoo

COMPANY NAME: LOCKERS PVT LTD

This code is developed by :- [Anil Ganguly Sahoo](#)

The code for this project is hosted at:-

<https://github.com/AnilSahoo98/LockedMe.com>



Table of Contents

<u>1.Sprint planning and Task completion.....</u>	
<u>2. core concepts used in the project.....</u>	
<u>3.Flow of Application.....</u>	
<u>4.Demonstrating the product capabilities,appearance and user interactions.....</u>	
<u>5.Unique selling points of the application.....</u>	
<u>6.conculsions.....</u>	

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

- Flow control
- File handling (java IO package)
- Exception Handling
- Collection Frame Work API
- Sorting
- OOPS concepts

Demonstrating the product capabilities, appearance and user interactions

To demonstrate the product capabilities, below are subsections configured to highlight appearance and user interactions for the project.

1. Creating the project in Eclipse
2. Writing a program in java for the entry point of the application LockedMeMain.java

Step 1: Creating a new Project in Eclipse

1. Open Eclipse
2. Go to File - New Project - Java Project - Next
3. Type the project name and click on Finish
4. Select your project name and go to File
 - 1) New Package
 - 2) Type Package Name
 - 3) com.lockedme
 - 4) Go to File Click New Class
 - 5) Type class Name
5. Enter LockedMeMain in class name, check the publicstaticvoid main (String [] args) checkbox & click on Finish

Step 2: Develop the code in Java for the Entry point of the application LockedMeMain.java(handling the Menuoption.java)

```
package com.lockedme;
```

```
public class MenuOption {
    public static void displayMenu(){
        // Creating a WelcomeScreen With Company Name And Developer Name

        Object appName = "Lockedme";
        Object developerName = "Anil Sahoo" ;
        String companyDetails =
String.format("*****\n" + "*** Welcome to
%s.com. \n" + "*** This application was developed by %s.\n" +
*****\n", appName, developerName);String
appFunction = "You can use this application to :-\n" + "• Retrieve all file names in the
\"directory\" folder\n" + "• Search, add, or delete files in \"directory\" folder.\n" +
"\n**Please be careful to ensure the correct filename is provided for searching or
deleting files.**\n";
        System.out.println(companyDetails);

        System.out.println(appFunction);
        // Creating a DisplayMenu

        String menu = "\n\n***** Select any option number from below and press
Enter *****\n\n" + "1) Retrieve all files inside \"directory\" folder\n" + "2) Display
menu for File operations\n" + "3) Exit program\n";
```

```
        System.out.println(menu);
        System.out.println("Enter The Option: \n");

    }
    public static void displayFileMenuOption() {
        // Creating a DisplayFileMenuOption to Add,Search,and Delete File
        String fileMenu = "\n\n***** Select any option number from below and
press Enter *****\n\n"
            + "1) Add a file to \"directory\" folder\n" + "2) Delete a file
from \"directory\" folder\n"
            + "3) Search for a file from \"directory\" folder\n" + "4)
Show Previous Menu\n" + "5) Exit program\n";

        System.out.println(fileMenu);

        System.out.println("Enter the option number:");

    }
}
```

Output

```
*****
** Welcome to Lockedme.com.
** This application was developed by Anil Sahoo.
*****

You can use this application to :-
• Retrieve all file names in the "directory" folder
• Search, add, or delete files in "directory" folder.

**Please be careful to ensure the correct filename is provided for searching or deleting files.**

***** Select any option number from below and press Enter *****

1) Retrieve all files inside "directory" folder
2) Display menu for File operations
3) Exit program

Enter The Option:
```

Step 3: Develop the code in Java for the Entry point of the application LockedMeMain.java (FileManager.java)

```
package com.lockedme;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

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

```
public class FileManagerMain {
```

```
    static final String folderpath ="E:\\Simplilearn\\Phase-1
    Assessment\\Lockedme.com\\Lockedme.com\\src\\directory";
```

```
public static void main(String[] args)
{

    HandleOptions.handleWelcomeScreenInput();

}

public static void getAllFiles(String directory) {
    //get file names
    //Using forEach
    List<String> fileName = LockedMeMain.getAllFiles(folderpath);
    for (String f :fileName)
        System.out.println(f);
}

public static void createFiles() {
    Scanner obj = new Scanner(System.in);
    String filename;
    int linesCount;
    List<String> content = new ArrayList<String>();
    //Read file name from user
    System.out.println("Enter file Name:");
    filename =obj.nextLine();
    //Read number of lines from user
    System.out.println("Enter how many lines in the file:");
```



```
linesCount=Integer.parseInt(obj.nextLine());
//Read line from user
for (int i = 1;i<=linesCount;i++)
{
    System.out.println("Enter line "+i+":");
    content.add(obj.nextLine());
}
//save the content into the file
boolean isSaved = LockedMeMain.createFiles(folderpath, filename,
content);
if(isSaved)
    System.out.println("File and data saved successfully");
else
    System.out.println("some error occured .please contact
admin@123.com");
//close scanner object
//    obj.close();
}
@SuppressWarnings("resource")
public static void deleteFile() {
    String fileName;
    //Read file name from user
    Scanner obj = new Scanner(System.in);
```

```

        System.out.println("Enter file name to be deleted:");
        fileName = obj.nextLine();
        boolean isDeleted = LockedMeMain.deleteFile(folderpath, fileName);
        if(isDeleted)

            System.out.println("File deleted sucessfully");//Delete the content
in the file

        else

            System.out.println("Enter file not there or some access issuse");
//ShowingError if the content is not present in the file

    }
    public static void searchFile()
    {
        //code for searching a file

        String fileName;
        Scanner obj = new Scanner(System.in); //Read file name
from user

        System.out.println("Enter file name to be searched:");
        fileName = obj.nextLine();

        boolean isFound = LockedMeMain.searchFile(folderpath,
fileName);

```

```

        if(isFound)
            System.out.println("File is present in the folder"); //If
the content is present in the file
        else
            System.out.println(" file is not present in te folder");
//If the content is not present in the file
//
        obj.close();
    }

}
//Operation of files

```

Output

Step 4: Develop the code in Java for the Entry point of the application LockedMeMain.java (HandleOption.java)

```

package com.lockedme;

import java.util.Scanner;

public class HandleOptions
{
    public static boolean handleWelcomeScreenInput()

```



```
        System.exit(0);
        break;
    default:
        System.out.println("Please select a valid option from
above."); //In case of invailed Entry
    }
}
catch (Exception e)
{
    System.out.println(e.getClass().getName());
    handleWelcomeScreenInput();
}

}while (true);

}

public static boolean handleFileMenuOptions() { //Option to add,delete,search the file

    Scanner sc = new Scanner(System.in);
    do {
```

```
try {
    MenuOption.displayFileMenuOption();

    int input = Integer.parseInt(sc.nextLine()); //user input option
    switch (input)
    {
        case 1:
            FileManagerMain.createFiles(); //Create a file
            break;
        case 2:
            FileManagerMain.deleteFile(); // delete the file from the
            directory
            break;
        case 3:
            FileManagerMain.searchFile(); // search the file from the
            directory
            break;
        case 4:
            return false ; //return to the previous Menu
        case 5:
            // Exit
            System.exit(0);
            System.out.println("Program exited successfully."); //exit
            the program
    }
}
```

```
        return false;

        default:
            System.out.println("Please select a valid option from
above.");
    }

    } catch (Exception e) {
        System.out.println(e.getClass().getName());
        e.printStackTrace();
        //        handleFileMenuOptions();
    }

    }while (true);

    }
```

Case 1) MenuOption get all the file in the directory

***** Select any option number from below and press Enter *****

- 1) Retrieve all files inside "directory" folder
- 2) Display menu for File operations
- 3) Exit program

Enter The Option:

1
aas
aj.txt
asd
asdf.txt
hello
jbh.txt
ksk.txt
qwe
sdf.txt
somr
Somthing WRong.txt
Somthing.txt
zxsad.txt

Case 2) MenuOption Display Menu Option for Operation

***** Select any option number from below and press Enter *****

- 1) Retrieve all files inside "directory" folder
- 2) Display menu for File operations
- 3) Exit program

Enter The Option:

2
|

***** Select any option number from below and press Enter *****

- 1) Add a file to "directoryy" folder
- 2) Delete a file from "directory" folder
- 3) Search for a file from "directory" folder
- 4) Show Previous Menu
- 5) Exit program

Enter the option number:

Case 3) MenuOption Exit the program

***** Select any option number from below and press Enter *****

- 1) Retrieve all files inside "directory" folder
- 2) Display menu for File operations
- 3) Exit program

Enter The Option:

3

Program exited successfully.

Case 1) Handling Option and File Operation Add

***** Select any option number from below and press Enter *****

- 1) Add a file to "directoryy" folder
- 2) Delete a file from "directory" folder
- 3) Search for a file from "directory" folder
- 4) Show Previous Menu
- 5) Exit program

Enter the option number:

1

Enter file Name:

Simililearn.txt

Enter how many lines in the file:

2

Enter line 1:

learning

Enter line 2:

Full stack javaDeveloper

File and data saved successfully

Case 2) Handling Option and File Operation delete

```
***** Select any option number from below and press Enter *****  
1) Add a file to "directoryy" folder  
2) Delete a file from "directory" folder  
3) Search for a file from "directory" folder  
4) Show Previous Menu  
5) Exit program  
  
Enter the option number:  
2  
Enter file name to be deleted:  
Simililearn.txt  
File deleted sucessfully
```

Case 3) Handling Option and File Operation search

```
***** Select any option number from below and press Enter *****  
  
1) Add a file to "directoryy" folder  
2) Delete a file from "directory" folder  
3) Search for a file from "directory" folder  
4) Show Previous Menu  
5) Exit program  
  
Enter the option number:  
3  
Enter file name to be searched:  
Simililearn.txt  
File is present in the folder
```

Step5: Develop the code in Java for the Entry point of the application

LockedMeMain.java

```
package com.lockedme;

import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;

public class LockedMeMain
{
    public static List<String> getAllFiles(String folderpath){
        //creating File Object
        File fl = new File(folderpath);
        //getting all the files into FileArray
        File[] listofFiles = fl.listFiles();
        //declare a list to store file names
        List<String> fileNames = new ArrayList<String>();

        for (File f:listofFiles) //Using ForEach to get the file names
            fileNames.add(f.getName());
    }
}
```

```
        return fileNames;

    }

    /**
     * This method will create to append content into the file specified
     * @param folderpath
     * @param fileName
     * @param content
     * @return boolean
     *
     */
    public static boolean createFiles(String folderpath,String fileName,List<String>
Content)
    {
        try
        {
            File fl = new File(folderpath,fileName);
            FileWriter fw = new FileWriter(fl); //write the userinput to the file

            for (String s : Content)
            {
                fw.write(s+ "\n");
            }
        }
    }
}
```

```
        fw.close();
        return true;
    }
    catch(Exception Ex)
    {
        return false;
    }
}

/**
 * This method will delete the file name if it exists.
 * @param folderpath
 * @param fileName
 * @return
 */

public static boolean deleteFile(String folderpath,String fileName)
{

    File file = new File(folderpath+"\\ "+fileName); //delete the file from the
directory

    try {
```

```
        if (file.delete()) // if the file exists it will delete the file
            return true;
        else
            return false;
    }
    catch(Exception ex) {
        return false;
    }
}

/**
 * This method will search the file
 * @param folderpath
 * @param fileName
 * @return
 */
public static boolean searchFile(String folderpath,String fileName) {
    File file = new File(folderpath+"\\ "+fileName);
    if(file.exists())
        return true;
    else
        return false;
}
```

```
}  
  
}
```

Pushing code to GitHub Repository

Steps to be followed:

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

cd

2. Initialize repository using the following command :

git init

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

git add .

4. Commit the changes using the following command :

git commit -m

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

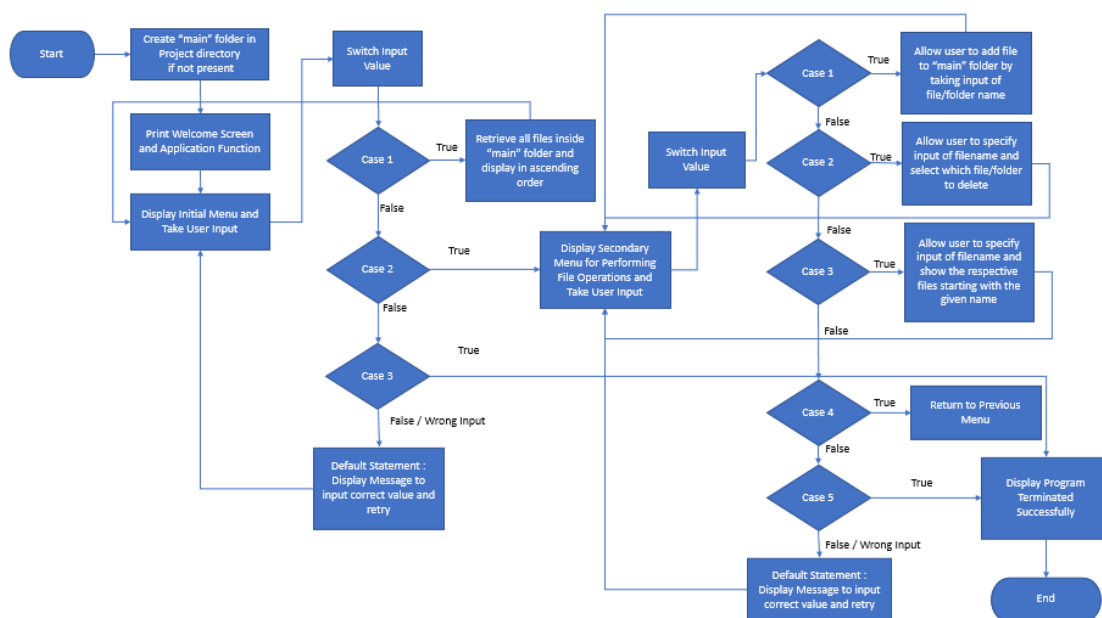
git push -u origin main

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.
7. When the option to retrieve files in ascending order is selected, user is displayed with two options of viewing the files.
 - 7.1. Ascending order of folders first which have files sorted in them,
 - 7.2. Ascending order of all files and folders inside the "main" folder.

8. The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code. Application has been developed keeping in mind that there should be very less “hard coding” of data.

Flow of the Application



Conclusions:-

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, etc.
- Allowing user to append data to the file