**Virtual Key for Your Repositories:**

**LockedMe.Com**

**Developed By: Pai Najranabanu Yakub**

**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.

**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
  + 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.

**Sprints Work Details:**

I have done project in 2 sprints; are as below

**First Sprint: (done coding part)**

* Create Java App Class to display welcome with application name also include developed by.

System.*out*.println((char)**27**+ "[35m" +"\n\tWELCOME TO LockedMe.Com APPLICATION : " + (char)**27**+ "[35m" +" Developed BY: PAI NAJRANABANU Y.")**;**System.*out*.print((char)**27**+ "[30;0m")**;***appstart*()**;**

* Create appstart() method for displaying LockedMe.Com option and according to the user choice appropriate method called from same class(APP) or other(FileOperation) Class.

private static void appstart() {  
 do{  
 System.*out*.println("\n\tLockedMe.Com Options:")**;** System.*out*.print("\t-------------------------------------")**;** System.*out*.println("\n\t 1: List Files")**;** System.*out*.println("\n\t 2: File Operations")**;** System.*out*.println("\n\t 3: Close")**;** System.*out*.print("\n\tSelect Any one option:")**;** *choice*=*s*.nextInt()**;** switch(*choice*)  
 {  
 case **1**: *fo*.ListFiles()**;** break**;** //FileOperation class method called   
 case **2**: *FileOperation*()**;**break**;** // this class method called  
 case **3**: System.*exit*(**0**)**;** default :   
 System.*out*.println((char)**27**+ "[31m" + "\n\t Please select Proper Numeber for operation." + (char)**27**+ "[30;0m")**;** *appstart*()**;** }  
 System.*out*.print("\tDo you want to continue ? Y/N ")**;** *c*=*s*.next().charAt(**0**)**;** }while(*c*=='Y' || *c*=='y')**;**}

* Add Do you want to continue? for repeated process until user want.
* Create Java FileOperation Class for to perform operation on file according to the request of user.
* Create directoryfiles() methods to initialize ArrayList object with all files from the directories and also sort them in FileOperation Class.

Example:

private void directoryfiles() {  
 dp=new File("D:\\JAVA\\Phase-1\\Practise\\LockedMeCom\\File")**;** listfile=Arrays.*asList*(dp.list())**;** Collections.*sort*(listfile)**;**}

* Create ListFiles() to display all current files from specified directories in FileOperation Class.

Example:

public void ListFiles() {  
 directoryfiles()**;** Iterator itr=listfile.iterator()**;** if(listfile.size()>**0**)  
 {  
 System.*out*.println("\t------------------------------------------------------------")**;** System.*out*.println("\tAvailable Files in direcory " + dp + " are:")**;** while(itr.hasNext())  
 System.*out*.println("\t\t\t\t\t\t "+ itr.next())**;** System.*out*.println("\t------------------------------------------------------------")**;** }  
 else   
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t Directories does not contain any file..")**;** System.*out*.println((char)**27** + "[30;0m")**;** }  
}

* Called ListFiles() ; as choice 1 in switch case from App class using object of FileOperation Class:

case **1**: *fo*.ListFiles()**;** break**;** //FileOperation class method called

* Create FileOperation() in main App to display all File Operation menu and as per user selection appropriate methods called using object of FileOperation Class.

private static void FileOperation() {  
 do{  
 System.*out*.println("\n\t File Operation:")**;** System.*out*.print("\t---------------------------------")**;** System.*out*.println("\n\t 1: Add File")**;** System.*out*.println("\n\t 2: View File")**;** System.*out*.println("\n\t 3: Delete File")**;** System.*out*.println("\n\t 4: Search File")**;** System.*out*.println("\n\t 5: Back")**;** System.*out*.print("\n\tSelect Any one option:")**;** *choice*=*s*.nextInt()**;** /\*while (!Character.isDigit(choice)) {  
 System.out.println((char)27+ "[31m" + "\n\t Please do not enter any character.");  
 System.out.print((char)27+ "[37;1m");  
 System.out.print("\n Select Proper Choice:");  
 choice=s.next().charAt(0);  
 }  
 choice=choice-48;\*/  
   
 switch(*choice*)  
 {  
 case **1**: *fo*.AddFile()**;**break**;** case **2**: *fo*.ViewFile()**;**break**;** case **3**: *fo*.DeleteFile()**;** break**;** case **4**: *fo*.SearchFile()**;**break**;** case **5**: *appstart*()**;**break**;** default :   
 System.*out*.println((char)**27**+ "[31m" + "\n\t Please select Proper Numeber for file operation." + (char)**27**+ "[30;0m")**;** *appstart*()**;** }  
 System.*out*.print("\n\tDo you want to continue for File Operation? Y/N ")**;** *c*=*s*.next().charAt(**0**)**;** }while(*c*=='Y'|| *c*=='y')**;** if(*c*!='Y'|| *c*!='y')  
 {  
 *appstart*()**;** }  
}

* AddFile() used to Add File in specific Folder in FileOperation Class and called form main APP using object of class.

public void AddFile()  
{  
 System.*out*.print("\n\tEnter File Name With Extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** try {  
 if(f.createNewFile())  
 {  
 System.*out*.print("\n\tEnter File Content :")**;** fco=s1.nextLine()**;** fw=new FileWriter(fnam**,**Charset.*forName*("UTF8"))**;** fw.write(fco)**;** fw.close()**;** System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Added Successfully:"+ (char)**27** + "[30;0m")**;** }  
 else  
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Already exist..:"+ (char)**27** + "[30;0m")**;** }   
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
}

called from main APP;

case **1**: *fo*.AddFile()**;**break

* ViewFile() used to Display the content of File as per user request.

public void ViewFile()  
{  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** directoryfiles()**;** n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 {  
 System.*out*.print((char)**27** + "[31m" + "\n\t " + fname +" File exist With content:"+ (char)**27** + "[30;0m")**;** n++**;** try {  
 fr=new FileReader(fnam)**;** int i=**0;** while((i=fr.read())!=-**1**)  
 {  
 System.*out*.print((char)i)**;** }  
 } catch (Exception e) {  
 //*TODO: handle exception* }  
 System.*out*.println()**;** }   
 }  
 if(n==**0**)  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File does not exist:"+ (char)**27** + "[30;0m")**;**}

called ViewFile() from main APP using object of FileOperation Class.

case **2**: *fo*.ViewFile()**;**break**;**

* DeleteFile() used to delete file with same case from the folder in FileOperation Class.

public void DeleteFile()  
{  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** f=new File("File/" + fname)**;** directoryfiles()**;** int n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 {  
 if(f.delete())  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname + " File Successfully deleted:"+ (char)**27** + "[30;0m")**;** n++**;** }   
 }  
 if(n==**0**)  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File does not exist:"+ (char)**27** + "[30;0m")**;**}

called DeleteFile() from main APP using object of FileOperation Class.

case **3**: *fo*.DeleteFile()**;** break**;**

* SearchFile() used to Search file with same case from the folder in FileOperation Class.

public void SearchFile()  
{  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** directoryfiles()**;** n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 n=**1;** }  
 if(n==**1**)  
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Found"+ (char)**27** + "[30;0m")**;** Desktop desk=Desktop.*getDesktop*()**;** try {  
 desk.open(f)**;** } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace()**;** }   
   
 }  
 else  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Not Found"+ (char)**27** + "[30;0m")**;**}

called SearchFile() from main APP using object of FileOperation Class.

case **4**: *fo*.SearchFile()**;**break**;**

* Add exit code to exit from the Application.

**Second Sprint: (done documentation and GitHub Repositories)**

* Create Algorithm.
* Flow chart.
* Taking Screenshots of Application.
* Create GitHub repositories.
* Upload all documentation and src file into the GitHub Repositories.

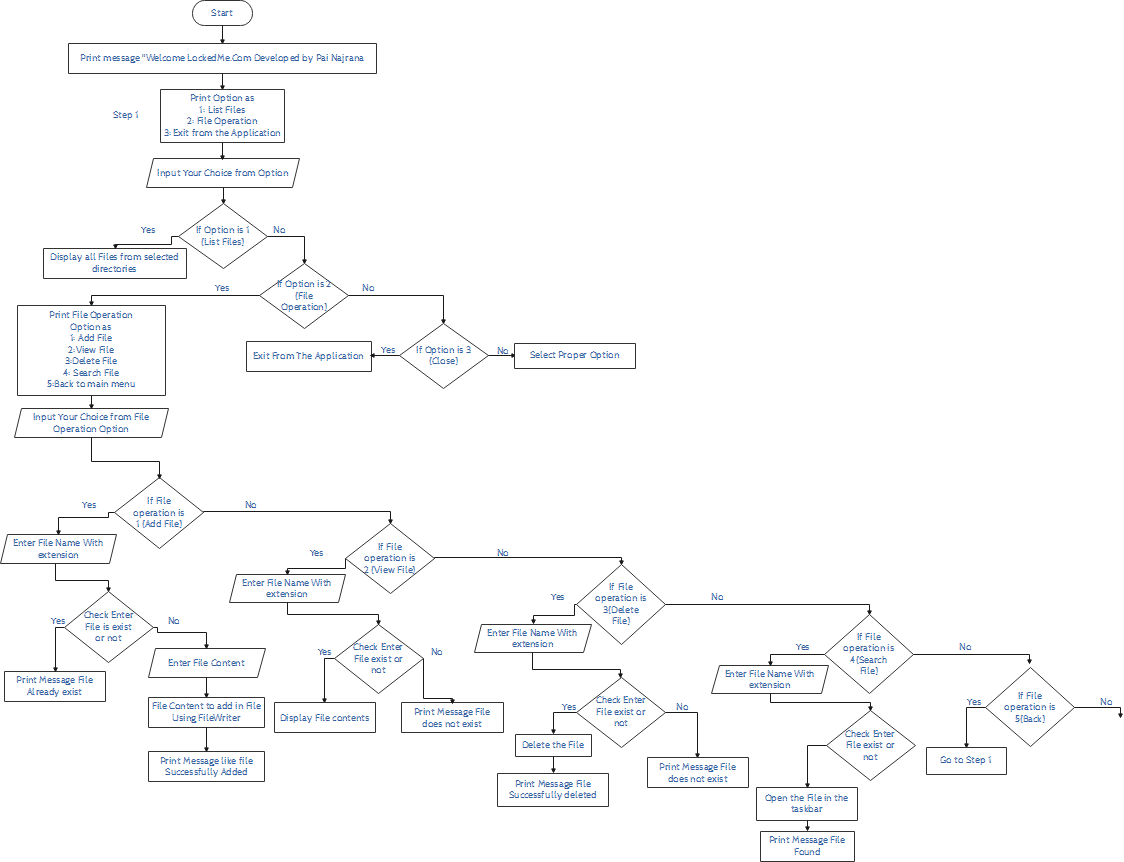
**Core Concepts used in Project:**

* Variable: Local, Global and static variable
* Object
* Access Modifier
* Flow Control (Switch case and if..else)
* Loop
* Function
* Collection (ArrayList)
* sorting
* File Handling (File Creation, FileWriter, FileReader, File Delete)
* Desktop (java.awt.Desktop for open the file in desktop)
* Exception Handling

**Algorithm:**

|  |  |
| --- | --- |
| Step 1: | Start the LockedMe.Com Application. |
| Step 2: | Display the Application name and developer name. |
| Step 3: | Display the LockedMe.Com option Menu like  1: List Files.  2. File Operation.  3. Close the Application. |
| Step 4: | Select any one option from step 3 |
| Step 5: | If Select option as 1(1: List Files) then Step 6 |
| Step 6: | Display all Files from the current directories. |
| Step 7: | User Input for “Do you want to continue?” If “Y” or ‘y” then Step 3 Otherwise Step 31 |
| Step 8: | If Select option as 2(2. File Operation) then Step 9 |
| Step 9: | Display the File Operation Option Menu like  1: Add File  2: View File  3: Delete File  4: Search File  5: Back |
| Step 10: | If Select File Operation As 1(1: Add File) then Step 11 |
| Step 11: | User Input for File Name with extension |
| Step 12: | Check user input file exist or not., if exist then step 13 else step 14. |
| Step 13: | Display message “File is already exist” |
| Step 14: | User Input for File content. |
| Step 14: | Display message like Filename File successfully added. |
| Step 15: | Display message like “Do You want to continue for file operation?” if ‘Y’ or ‘y’ then step 9 else step 3. |
| Step 16: | If Select File Operation As 2(2: View File )then Step 17 |
| Step 17: | Repeat Step 11 |
| Step 18: | Check user input file exist or not., if exist then step 19 else step 20. |
| Step 19: | Display the contents of file. |
| Step 20: | Display message “File does not exist”. |
| Step 21: | If Select File Operation As 3(3: Delete File) then Step 22 |
| Step 22: | Repeat Step 11 |
| Step 23: | Check user input file exist or not., if exist then step 24 else step 20. |
| Step 24: | Delete the file and display message File successfully deleted. |
| Step 25: | If Select File Operation As 4(4: Search File) then Step 26 |
| Step 26: | Repeat Step 11 |
| Step 27: | Check user input file exist or not., if exist then step 28 else step 20. |
| Step 28: | Open the File and Display message “File Found” |
| Step 29: | If Select File Operation As 5 then Step 3. |
| Step 30: | If select the LockedMe.Com operation as 3 (Closed) then step 31. |
| Step 31: | Close the Application |

**Flowchart:**



**Coding:**

**I have Create 2 class APP and FileOperation:**

1. **APP Class:**
2. **FileOperation Class:**

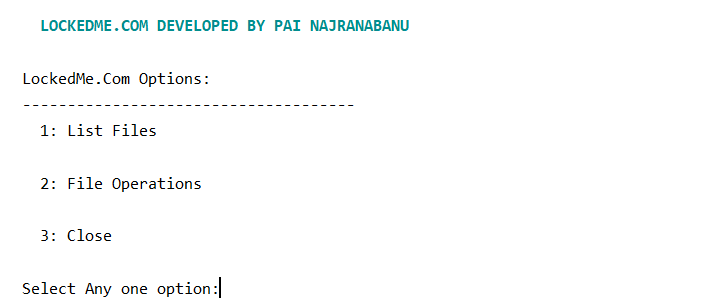
**APP Class:**

import java.util.Scanner**;**public class APP {  
 static char *c***;** static int *choice***;** static Scanner *s* = new Scanner(System.*in*)**;** static FileOperation *fo*=new FileOperation()**;** //object creation  
 public static void main(String[] args) throws Exception {  
   
 System.*out*.println((char)**27**+ "[35m" +"\n\tWELCOME TO LockedMe.Com APPLICATION : " + (char)**27**+ "[35m" +" Developed BY: PAI NAJRANABANU Y.")**;** System.*out*.print((char)**27**+ "[30;0m")**;** *appstart*()**;** }  
  
 private static void appstart() {  
 do{  
 System.*out*.println("\n\tLockedMe.Com Options:")**;** System.*out*.print("\t-------------------------------------")**;** System.*out*.println("\n\t 1: List Files")**;** System.*out*.println("\n\t 2: File Operations")**;** System.*out*.println("\n\t 3: Close")**;** System.*out*.print("\n\tSelect Any one option:")**;** *choice*=*s*.nextInt()**;** switch(*choice*)  
 {  
 case **1**: *fo*.ListFiles()**;** break**;** //FileOperation class method called   
 case **2**: *FileOperation*()**;**break**;** // this class method called  
 case **3**: System.*exit*(**0**)**;** default :   
 System.*out*.println((char)**27**+ "[31m" + "\n\t Please select Proper Numeber for operation." + (char)**27**+ "[30;0m")**;** *appstart*()**;** }  
 System.*out*.print("\tDo you want to continue ? Y/N ")**;** *c*=*s*.next().charAt(**0**)**;** }while(*c*=='Y' || *c*=='y')**;** }  
  
 private static void FileOperation() {  
 do{  
 System.*out*.println("\n\t File Operation:")**;** System.*out*.print("\t---------------------------------")**;** System.*out*.println("\n\t 1: Add File")**;** System.*out*.println("\n\t 2: View File")**;** System.*out*.println("\n\t 3: Delete File")**;** System.*out*.println("\n\t 4: Search File")**;** System.*out*.println("\n\t 5: Back")**;** System.*out*.print("\n\tSelect Any one option:")**;** *choice*=*s*.nextInt()**;** /\*while (!Character.isDigit(choice)) {  
 System.out.println((char)27+ "[31m" + "\n\t Please do not enter any character.");  
 System.out.print((char)27+ "[37;1m");  
 System.out.print("\n Select Proper Choice:");  
 choice=s.next().charAt(0);  
 }  
 choice=choice-48;\*/  
   
 switch(*choice*)  
 {  
 case **1**: *fo*.AddFile()**;**break**;** case **2**: *fo*.ViewFile()**;**break**;** case **3**: *fo*.DeleteFile()**;** break**;** case **4**: *fo*.SearchFile()**;**break**;** case **5**: *appstart*()**;**break**;** default :   
 System.*out*.println((char)**27**+ "[31m" + "\n\t Please select Proper Numeber for file operation." + (char)**27**+ "[30;0m")**;** *appstart*()**;** }  
 System.*out*.print("\n\tDo you want to continue for File Operation? Y/N ")**;** *c*=*s*.next().charAt(**0**)**;** }while(*c*=='Y'|| *c*=='y')**;** if(*c*!='Y'|| *c*!='y')  
 {  
 *appstart*()**;** }  
 }   
}

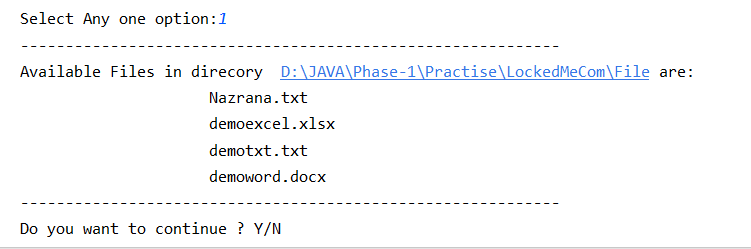
**FileOperation Class:**

import java.awt.Desktop**;**import java.io.File**;**import java.io.FileReader**;**import java.io.FileWriter**;**import java.io.IOException**;**import java.nio.charset.Charset**;**import java.util.Arrays**;**import java.util.Collections**;**import java.util.Iterator**;**import java.util.List**;**import java.util.Scanner**;**public class FileOperation {  
 String fname**,**fnam**,**fco**;**File f**,**dp**;** FileWriter fw**;**int n**;**FileReader fr**;** Scanner s1=new Scanner(System.*in*)**;** List<String> listfile**;** public void ListFiles() {  
 directoryfiles()**;** Iterator itr=listfile.iterator()**;** if(listfile.size()>**0**)  
 {  
 System.*out*.println("\t------------------------------------------------------------")**;** System.*out*.println("\tAvailable Files in direcory " + dp + " are:")**;** while(itr.hasNext())  
 System.*out*.println("\t\t\t\t\t\t "+ itr.next())**;** System.*out*.println("\t------------------------------------------------------------")**;** }  
 else   
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t Directories does not contain any file..")**;** System.*out*.println((char)**27** + "[30;0m")**;** }  
 }  
 private void directoryfiles() {  
 dp=new File("D:\\JAVA\\Phase-1\\Practise\\LockedMeCom\\File")**;** listfile=Arrays.*asList*(dp.list())**;** Collections.*sort*(listfile)**;** }  
 public void AddFile()  
 {  
 System.*out*.print("\n\tEnter File Name With Extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** try {  
 if(f.createNewFile())  
 {  
 System.*out*.print("\n\tEnter File Content :")**;** fco=s1.nextLine()**;** fw=new FileWriter(fnam**,**Charset.*forName*("UTF8"))**;** fw.write(fco)**;** fw.close()**;** System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Added Successfully:"+ (char)**27** + "[30;0m")**;** }  
 else  
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Already exist..:"+ (char)**27** + "[30;0m")**;** }   
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
 }  
 public void ViewFile()  
 {  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** directoryfiles()**;** n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 {  
 System.*out*.print((char)**27** + "[31m" + "\n\t " + fname +" File exist With content:"+ (char)**27** + "[30;0m")**;** n++**;** try {  
 fr=new FileReader(fnam)**;** int i=**0;** while((i=fr.read())!=-**1**)  
 {  
 System.*out*.print((char)i)**;** }  
 } catch (Exception e) {  
 //*TODO: handle exception* }  
 System.*out*.println()**;** }   
 }  
 if(n==**0**)  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File does not exist:"+ (char)**27** + "[30;0m")**;** }  
 public void DeleteFile()  
 {  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** f=new File("File/" + fname)**;** directoryfiles()**;** int n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 {  
 if(f.delete())  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname + " File Successfully deleted:"+ (char)**27** + "[30;0m")**;** n++**;** }   
 }  
 if(n==**0**)  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File does not exist:"+ (char)**27** + "[30;0m")**;** }  
 public void SearchFile()  
 {  
 System.*out*.print("\n\tEnter File Name with extension:")**;** fname=s1.nextLine()**;** fnam="File/" +fname**;** f=new File(fnam)**;** directoryfiles()**;** n=**0;** for (String fn : listfile)  
 {  
 if(fn.equals(fname))  
 n=**1;** }  
 if(n==**1**)  
 {  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Found"+ (char)**27** + "[30;0m")**;** Desktop desk=Desktop.*getDesktop*()**;** try {  
 desk.open(f)**;** } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace()**;** }   
   
 }  
 else  
 System.*out*.println((char)**27** + "[31m" + "\n\t " + fname +" File Not Found"+ (char)**27** + "[30;0m")**;** }  
}

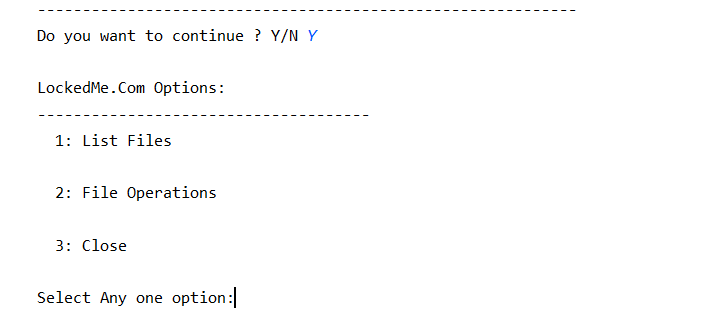
**Execute the program: ( I have changed the color scheme as classic Light)**



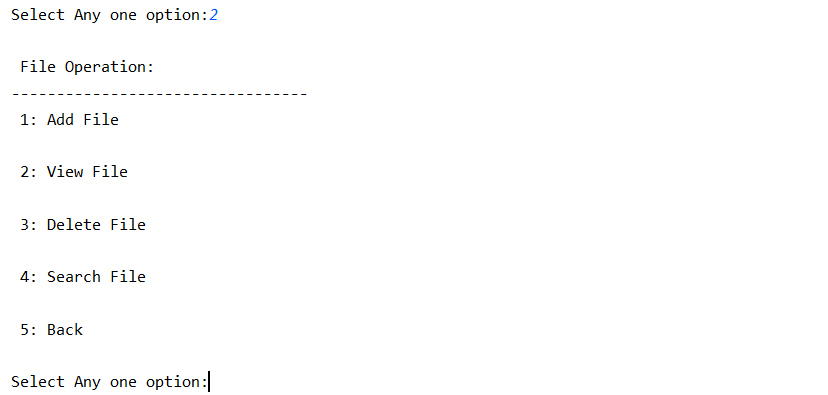
* If select any one option as 1, Its showings all files that exist in a folder:



* If you want to continue operation please press ‘Y’ or ‘y’:

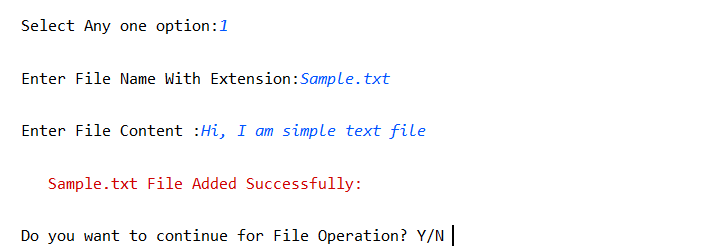


* If select any one option as 2, File Operation Option:

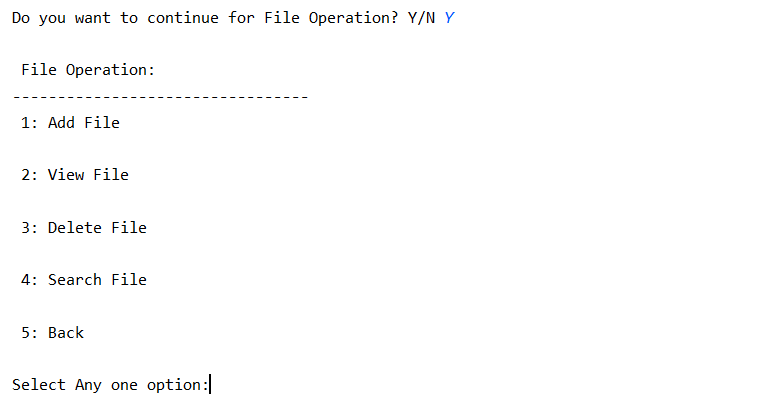


* Want to add file then please select any one option as 1:

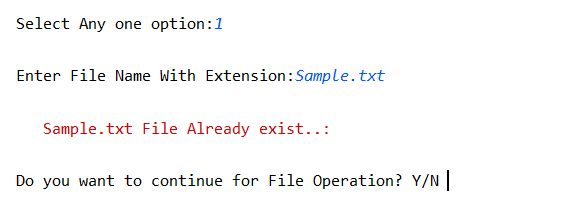
1. Enter File name with extension because it allows to create any file that you want.
2. And Enter Content of file:



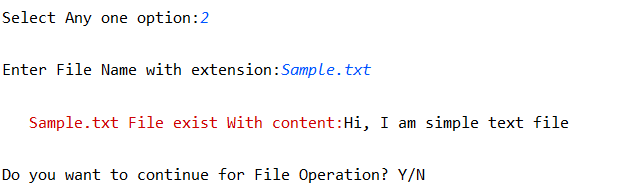
* If you want to continue operation please press ‘Y’ or ‘y’:



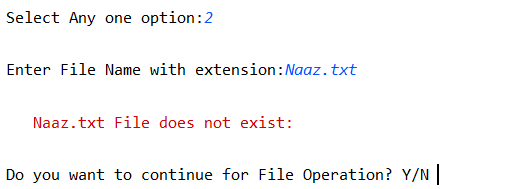
* If want to add file, but file is already exist then:



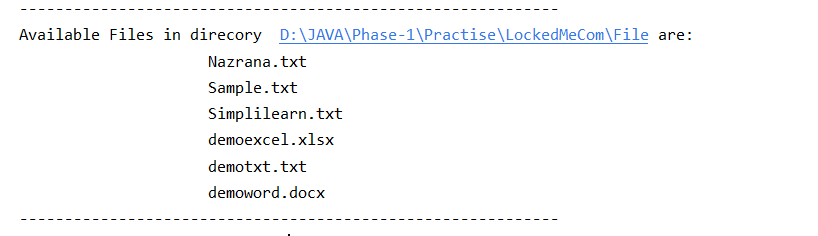
* Want to View file content then please select any one option as 2:

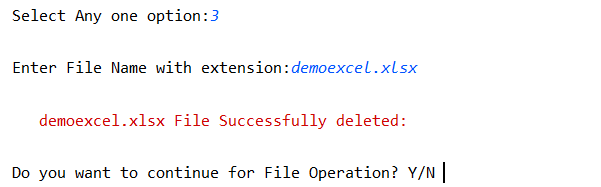


* If File does not exist then:

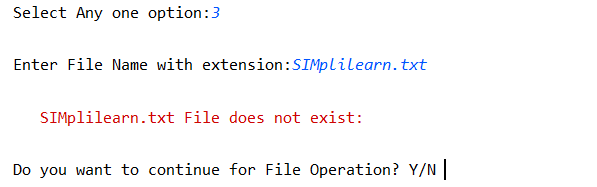


* Want to delete a File, select option as 3(delete file with case sencitive):

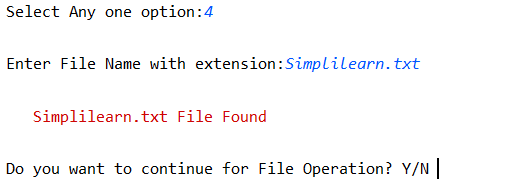




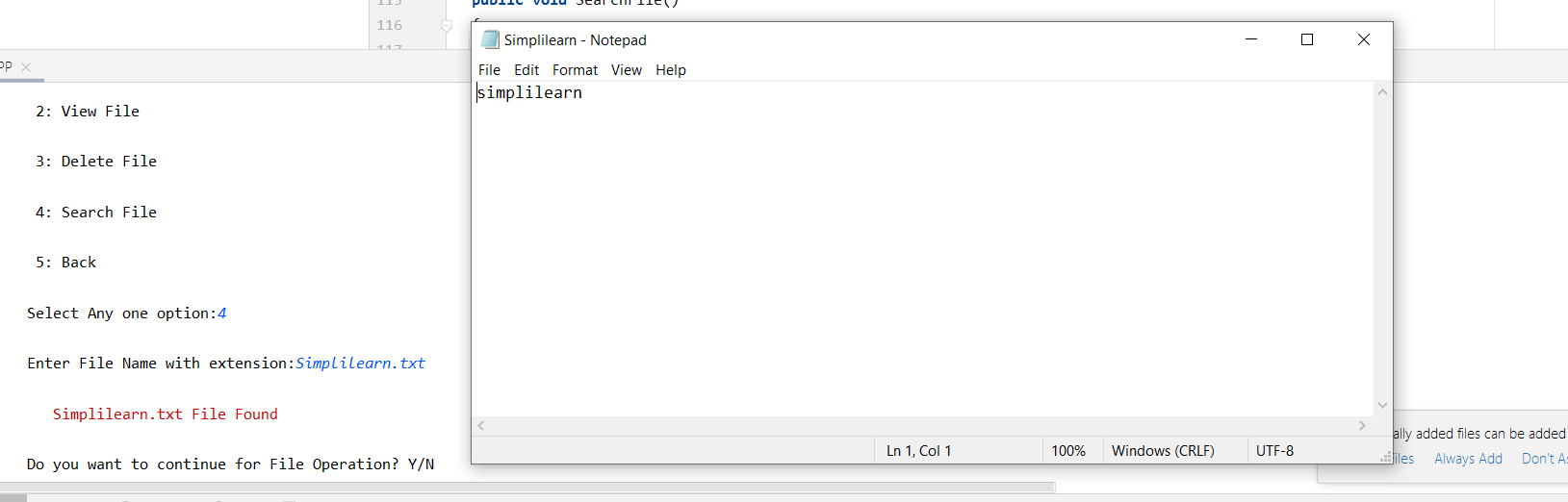
* Case sensitive example for delete:



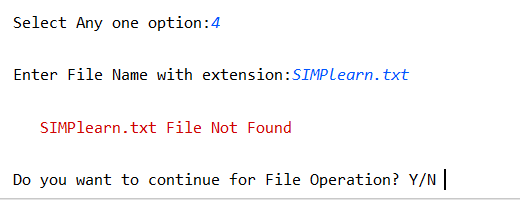
* Want to search a File, select option as 4: (if file exist then its open a file)



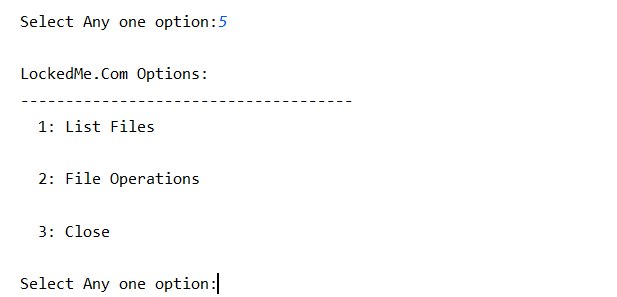
It open the file:



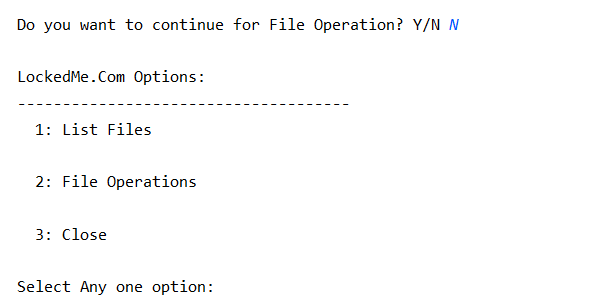
* Case sensitive example for Search:



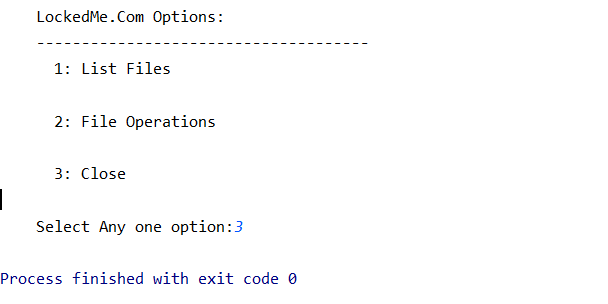
* Want to move main menu then select option as 5 or do you want to continue for file operation as ‘N’ or ‘n’:



OR

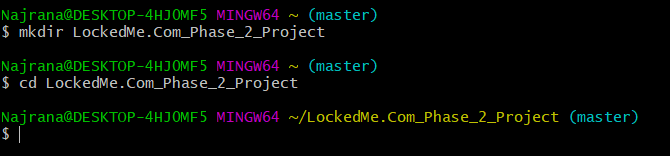


* Select 3 for close the Application:

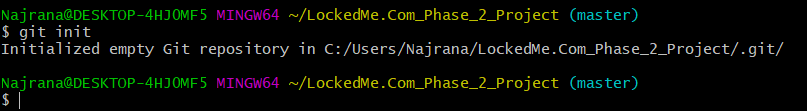


**Pushing the Code in to GitHub Repository**:

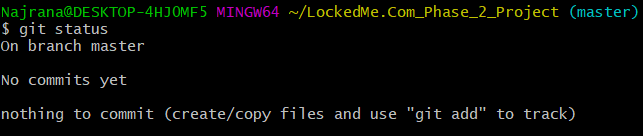
* Open Git Bash
* Create Directory name as LockedMe.Com\_Phase\_2\_Project.

****

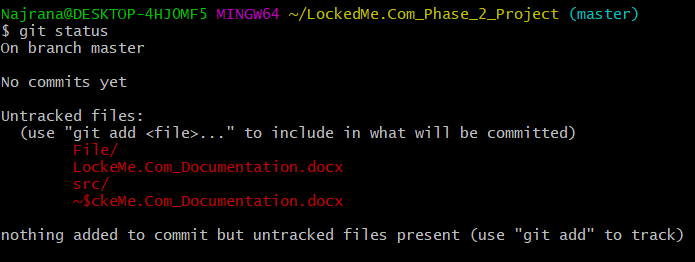
* Initialize the git directory using ***git init*** Command.



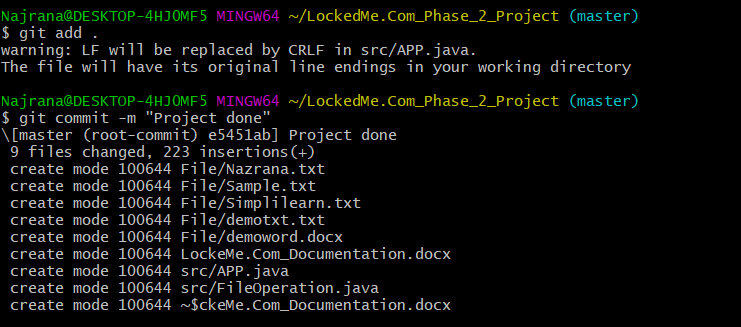
* Check status using ***git status***:



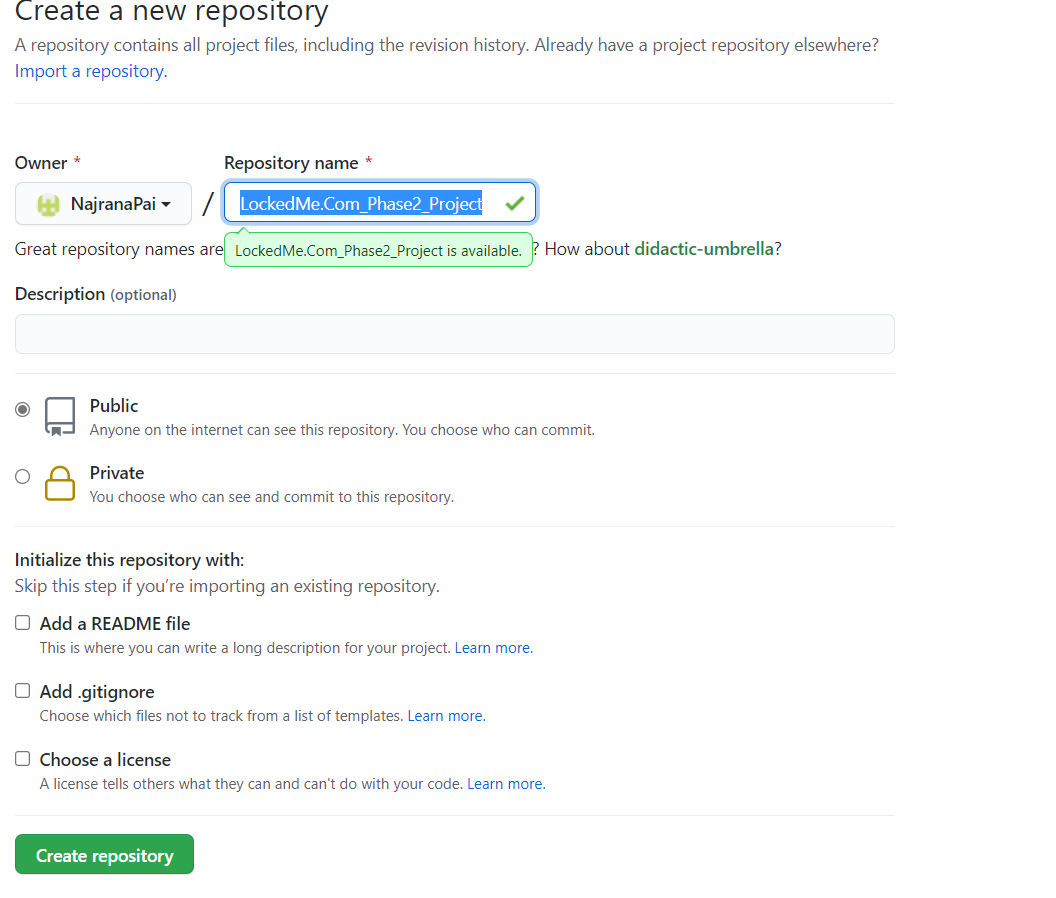
* Copy the files that you want to in GitHub Repositories:
* Now check status using ***git status***:



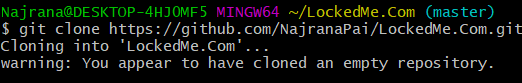
* Perform ***git add .*** and ***git commit -m “Project done”:***



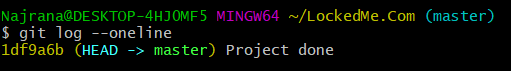
* Create New Repository in [Create a New Repository (github.com)](https://github.com/new)



* ***git clone*** repository link:



* ***git log --oneline***:



* ***git push -u origin master:***