**LockedMe – Virtual Key for Repositories**

Table of content-

* Sprint planning
* DFD of application
* Operation performs by application
* Java concept used in the application
* User Interaction (Screenshots)
* Coding

**Sprint planning**- This project is completed in 2 sprints,

Below are the tasks accomplished inside the sprints.

Sprint 1 –

* Task1 – Created the initial project documentation
* Task2 – Created the basic java project with only 1 feature of app.
* Task3 - Created the git hub repository
* Task4 – Cloned the remote repository in eclipse (IDE)
* Task5 – Pushed the initial code in the repository (Main branch)
* Task6 - Created multiple feature branch from main branch and added remaining features of application
* Task7 – Merged the features branches into the main branch without any code conflict.
* Task 8 – Pull the main branch in the eclipse to reflect the changes into the local repository from the remote repository (Main Branch)

Sprint 2 –

* Task1 - Created the multiple hotfix branches to fix the open issues in the app
* Task2 – Merged the branches to main branch without any code conflict

DFD (Data Flow Diagram) –

Create the directory (D:// File handling) if not already exist

Start

+

Case 5

Case 4

Case 3

Case 2

Case 1

Switch

Take user input of his/ her choice

Print welcome screen and operations that can be performed by the application

Yes

Retrieve all files

No

Add files

Yes

No

No Yes

Search file

No

Yes

Delete file

No

Exit

Yes

No

Default

Enter correct option

**Operation performs by application**

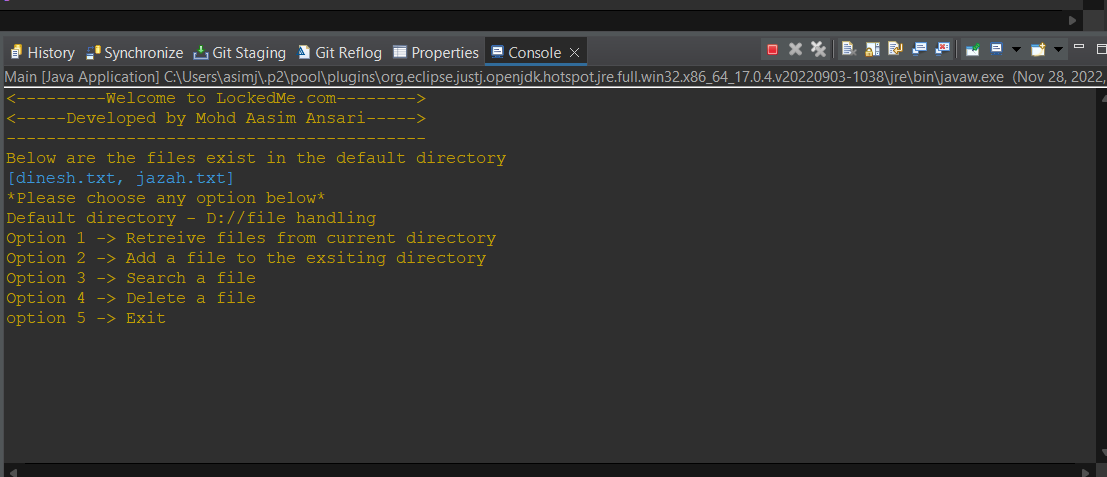
1. Retrieve all Files: - In this operation user can see all the files which is exist in current directory, if there is no file exist in current directory then a message “there is no file exist” will be shown to a user.
2. Add File: - In this operation user can add a file in a particular directory, if the directory exists then it will create the file or if the directory does not exist then first it will create the directory after that it will create the file.
3. Search File: - In this operation user can search for a file in current directory, if file exists then user will see the file and the content of the file, if file does not exist then a message “file not found” will be shown.
4. Delete File: - In this operation user can delete a file in current directory, if file exists then that file will be deleted.

**Java concept used in the application**

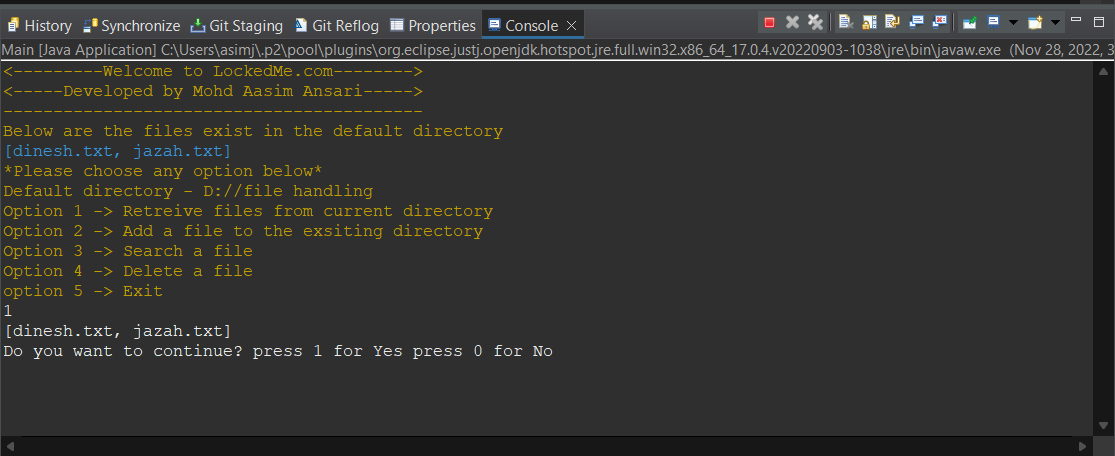
1. File Handling
2. Exception Handling
3. Collections
4. Sorting
5. Searching (Linear Search)

User Interaction (Screenshots)

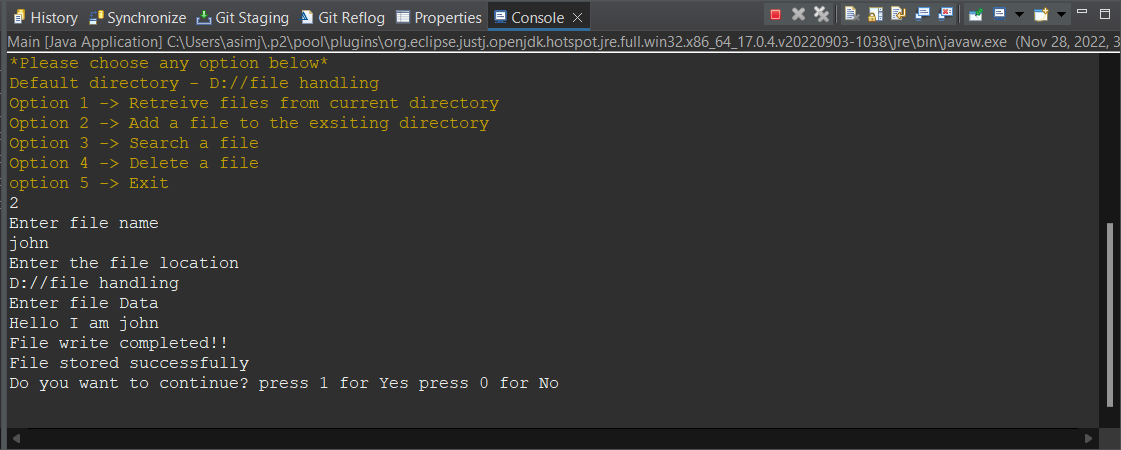
Welcome Screen



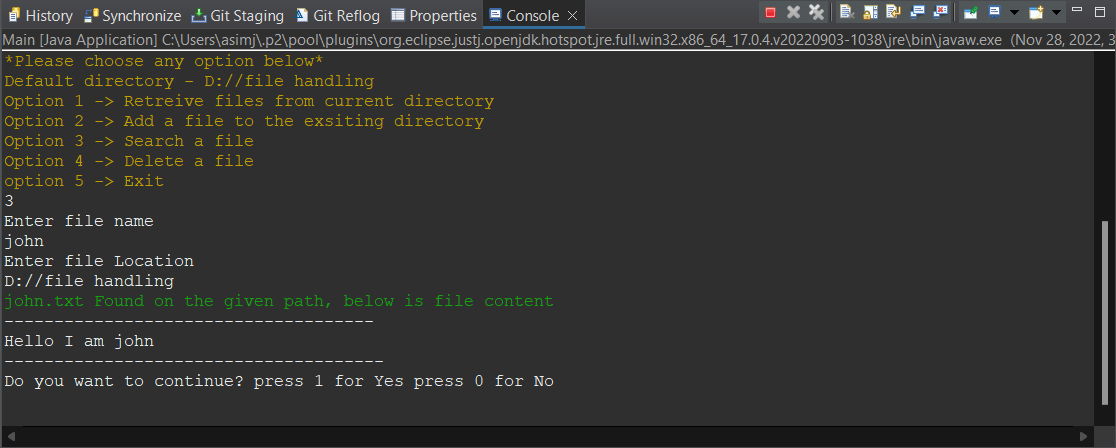
Option 1 (Retrieve all files from current directory)



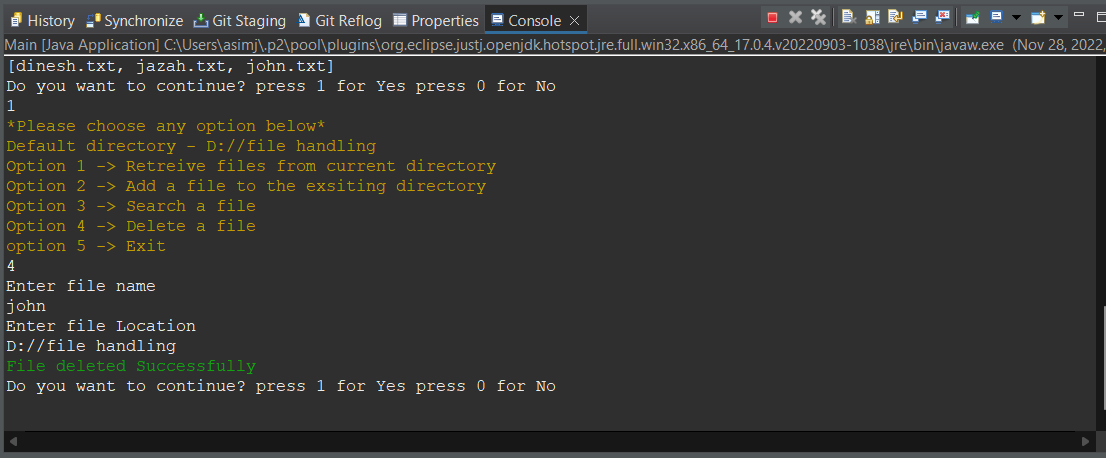
Option2 – Add files to the directory



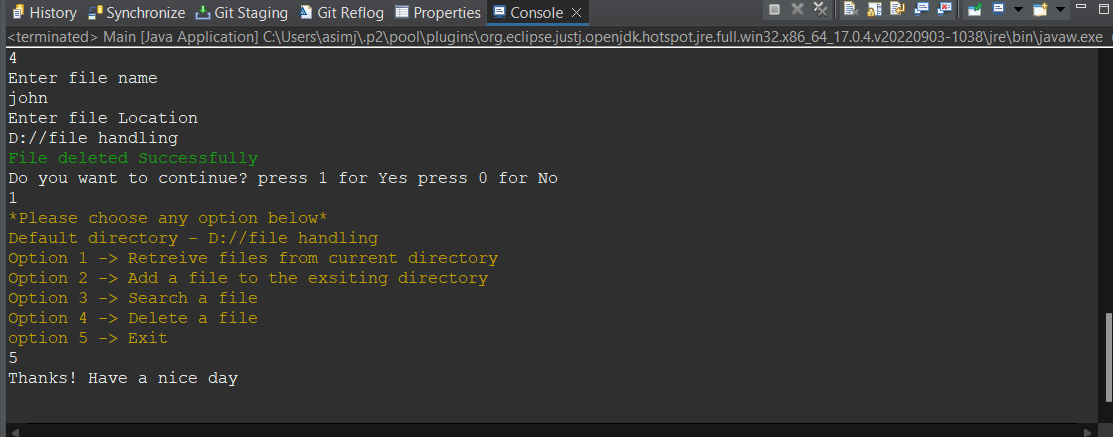
Option3 – Search file in the directory



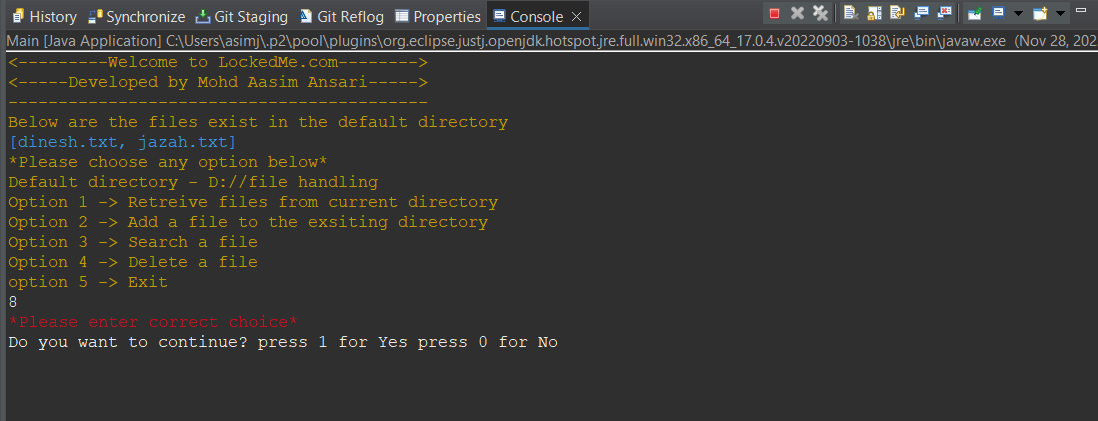
Option4 – Delete a file



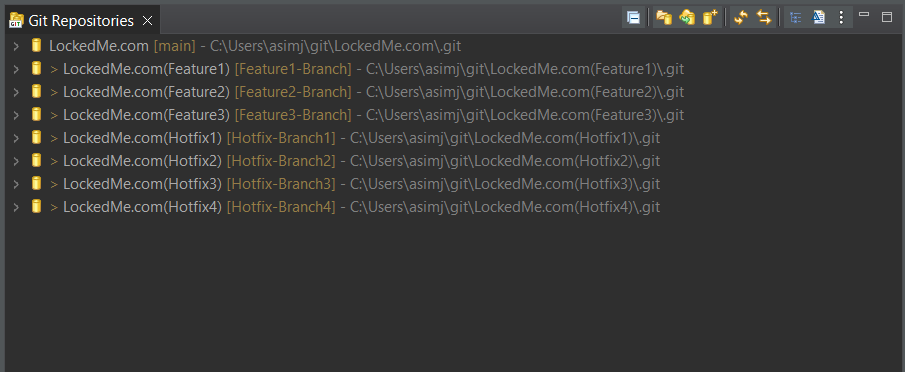
Option5 – Exit



Default case



Git Hub Multiple branches Local repositories



**Coding**

**Main.java**

package com.lockers.main;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.List;

import java.util.Scanner;

import java.io.File;

import com.lockers.service.Service;

public class Main {

public static final String ANSI\_RED = "\u001B[31m";

public static final String ANSI\_GREEN = "\u001B[32m";

public static final String ANSI\_RESET = "\u001B[0m";

public static final String ANSI\_YELLOW = "\u001B[33m";

public static final String CYAN = "\033[0;36m";

static int exit =0;

public static void main(String[] args) throws IOException

{

File file = new File("D://file handling");

//Creating the directory

file.mkdir();

Service serv = new Service();

int choice=0;

int option;

System.out.println(ANSI\_YELLOW+"<---------Welcome to LockedMe.com-------->");

System.out.println("<-----Developed by Mohd Aasim Ansari----->");

System.out.println("------------------------------------------");

System.out.println("Below are the files exist in the default directory");

List<String> files1 =serv.getAllFiles();

System.out.println(CYAN+files1+ANSI\_RESET);

do {

System.out.println(ANSI\_YELLOW+"\*Please choose any option below\*");

System.out.println("Default directory - D://file handling");

System.out.println("Option 1 -> Retreive files from current directory");

System.out.println("Option 2 -> Add a file to the exsiting directory");

System.out.println("Option 3 -> Search a file");

System.out.println("Option 4 -> Delete a file");

System.out.println("option 5 -> Exit"+ANSI\_RESET);

Scanner sc = new Scanner(System.in);

option = sc.nextInt();

switch(option)

{

case 1: {

//To get all files in specified directory

List<String> files =serv.getAllFiles();

System.out.println(files);

break;

}

case 2: {

sc.nextLine();

System.out.println("Enter file name");

String fName = sc.nextLine();

// sc.nextLine();

System.out.println("Enter the file location");

String fLocation = sc.nextLine();

System.out.println("Enter file Data");

// sc.nextLine();

String fData = sc.nextLine();

boolean status = serv.addFile(fName,fLocation, fData);

if(status)

System.out.println("File stored successfully");

else

System.out.println("File not stored! Enter proper File location");

break;

}

case 3: {

sc.nextLine();

System.out.println("Enter file name");

String fName = sc.nextLine();

System.out.println("Enter file Location");

String fLocatoin = sc.nextLine();

try {

BufferedReader br = serv.searchFile(fName, fLocatoin);

if(br!=null) {

System.out.println(ANSI\_GREEN+fName+".txt Found on the given path, below is file content"+ANSI\_RESET);

System.out.println("-------------------------------------");

String st;

// Condition holds true till

// there is character in a string

while ((st = br.readLine()) != null)

// Print the string

System.out.println(st);

System.out.println("--------------------------------------");

}

else

{

System.out.println(ANSI\_RED+"File not found!"+ANSI\_RESET);

}

br.close();

}

catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

break;

}

case 4: {

//To delete a file

sc.nextLine();

System.out.println("Enter file name");

String fName = sc.nextLine();

System.out.println("Enter file Location");

String fLocatoin = sc.nextLine();

if(serv.deleteFile(fName, fLocatoin))

{

System.out.println(ANSI\_GREEN+"File deleted Successfully"+ANSI\_RESET);

}

else

{

System.out.println("File not found");

}

break;

}

case 5: {

exit=1;

choice=0;

break;

}

default:{

System.out.println(ANSI\_RED+"\*Please enter correct choice\*"+ANSI\_RESET);

break;

}

if(exit!=1)

{

System.out.println("Do you want to continue? press 1 for Yes press 0 for No");

choice = sc.nextInt();

}

}

while(choice==1);

System.out.println("Thanks! Have a nice day");

}

}

**Service.java**

package com.lockers.service;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.io.File;

import javax.swing.plaf.synth.SynthOptionPaneUI;

public class Service {

Map<String,File> map = new HashMap<String,File>();

//Add file method

public boolean addFile(String fileName,String fileLocation ,String fileData )

{

boolean status = false;

File file = new File(fileLocation);

//Creating the directory

file.mkdir();

try {

FileWriter fw = new FileWriter(fileLocation+"/"+fileName+".txt");

fw.write(fileData);

System.out.println("File write completed!!");

fw.close();

status = true;

}

catch(IOException e)

{

e.printStackTrace();

}

return status;

}

//File deletion method

public boolean deleteFile(String fName, String fLocation)

{

boolean status = false;

try

{

File f= new File(fLocation+"/"+fName+".txt"); //file to be delete

if(f.delete()) //returns Boolean value

{

status = true;

}

}

catch(Exception e)

{

e.printStackTrace();

}

return status;

}

//File search method

public BufferedReader searchFile(String fName, String fLocation) throws FileNotFoundException

{

BufferedReader br =null;

File directoryPath = new File(fLocation);

//List of all files and directories

String contents[] = directoryPath.list();

// System.out.println("List of files and directories in the specified directory:");

for(int i=0; i<contents.length; i++) {

if(contents[i].equalsIgnoreCase(fName+".txt"))

{

File file = new File(fLocation+"/"+fName+".txt");

br = new BufferedReader(new FileReader(file));

}

}

return br;

}

//Get all files method

public List<String> getAllFiles()

{

List<String> files =new ArrayList<>();

String fLocation = "D://file handling";

File directoryPath = new File(fLocation);

//List of all files and directories

String contents[] = directoryPath.list();

// List of files and directories in the specified directory:

for(int i=0; i<contents.length; i++)

{

files.add(contents[i]);

}

//Sorting all the files names

Collections.sort(files);

return files;

}

}