package com.project.lockedme;

import java.io.File;

import java.io.IOException;

import java.util.Arrays;

import java.util.Set;

import java.util.TreeSet;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class OperationsDAO {

public void listAllFiles(String path) {

if (path == null || path.isEmpty() )

throw new NullPointerException("Path cannot be Empty or null");

File dir = new File(path);

if(!dir.exists())

throw new IllegalArgumentException("Path does not exist");

if(dir.isFile())

throw new IllegalArgumentException("The given path is a file. A directory is expected.");

String [] files = dir.list();

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

if(files != null && files.length > 0) {

Set<String>filesList = new TreeSet<String>(Arrays.asList(files));

System.out.println("The Files in "+ dir.getAbsolutePath() + " are: \n");

for(String file1:filesList) {

System.out.println(file1);

}

System.out.println("\nTotal Number of files: "+ filesList.size());

}else {

System.out.println("Directory is Empty");

}

}

public void createNewFile(String path , String fileName) throws IOException {

if (path == null || path.isEmpty() )

throw new NullPointerException("Path cannot be Empty or null");

if (fileName == null || fileName.isEmpty())

throw new NullPointerException("File Name cannot be Empty or null");

File newFile = new File(path + File.separator + fileName);

boolean createFile = newFile.createNewFile();

if (createFile) {

System.out.println("\nFile Successfully Created: " + newFile.getAbsolutePath());

}else if(!createFile) {

System.out.println("\nFile Already Exist.. Please try again." );

}

}

public void deleteFile(String path , String fileName) throws IOException {

if (path == null || path.isEmpty() )

throw new NullPointerException("Path cannot be Empty or null");

if (fileName == null || fileName.isEmpty() )

throw new NullPointerException("File Name cannot be Empty or null");

File newFile = new File(path + File.separator + fileName);

boolean deleteFile = newFile.delete();

if (deleteFile) {

System.out.println("\nFile deleted Successfully");

}else {

System.out.println("\nFile Not Found.. Please try again." );

}

}

public void searchFile(String path , String fileName){

if (path == null || path.isEmpty() )

throw new NullPointerException("Path cannot be Empty or null");

if (fileName == null || fileName.isEmpty() )

throw new NullPointerException("File Name cannot be Empty or null");

File dir = new File(path);

if(!dir.exists())

throw new IllegalArgumentException("Path does not exist");

if(dir.isFile())

throw new IllegalArgumentException("The given path is a file. A directory is expected.");

String [] fileList = dir.list();

boolean flag = false;

Pattern pat = Pattern.compile(fileName);

if(fileList != null && fileList.length > 0) {

for(String file:fileList) {

Matcher mat = pat.matcher(file);

if(mat.matches()) {

System.out.println("File Found at location: " + dir.getAbsolutePath());

flag = true;

break;

}

}

}

if(flag == false)

System.out.println("File Not Found.. Please try again.");

}

}