**Source Code for the project – Virtual Key for your Repositories**

package VirtualKey;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Arrays;

import java.util.Scanner;

public class VirtualKeyTest{

public static void main(String[] args) {

File folder = new File("D:\\Project\\GitHub\\Virtual Key For Repository");

folder.mkdirs();

Scanner scan = new Scanner(System.in);

while(true){

System.out.println("Prototype of the Application Page>\n");

System.out.println("These are the functions: \n");

System.out.println("1. To display the files in ascending order\n");

System.out.println("2. Here are some Business level operation menu \n");

System.out.println("3. Exit from the application");

System.out.println("Which function you want to proceed : \n");

int choice = scan.nextInt();

switch(choice)

{

case 1:

File arr[] = folder.listFiles();

Arrays.sort(arr);

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

System.out.println(arr[i]);

break;

case 2:

Boolean temp = true;

while(temp) {

System.out.println("Option 1 :- To Add a file in the existing directory");

System.out.println("Option 2 :- To Delete a file from the existing directory. ");

System.out.println("Option 3 :- To Search a user specified file from the directory");

System.out.println("Option 4 :- Back to the previous menu");

System.out.println("Option 5 :- Application Exit Successfully");

int choice2 = scan.nextInt();

switch (choice2) {

case 1:

System.out.println("Enter the File Name to create");

String s = scan.next();

File file = new File(folder, s);

try {

boolean value = file.createNewFile();

if (value) {

System.out.println("The new file is created.");

} else {

System.out.println("The file already exists.");

}

} catch (Exception e) {

e.getStackTrace();

}

break;

case 2:

System.out.println("Enter the File Name");

String name = scan.next();

File file1 = new File(folder, name);

try {

boolean value = file1.delete();

if (value) {

System.out.println("File deleted Successfully.");

} else {

System.out.println("File Not Found");

}

} catch (Exception e) {

e.getStackTrace();

}

break;

case 3:

System.out.println("Enter a keyword to search");

String search = scan.next();

boolean flag1 = false;

File arr1[] = folder.listFiles();

System.out.println("File Found :\n");

for (int i = 0; i < arr1.length; i++) {

if (arr1[i].getName().startsWith(search)) {

flag1 = true;

System.out.println(arr1[i]);

}

}

if (flag1 == false) {

System.out.println("No file found");

}

case 4:

temp = false;

break;

case 5:

System.out.println("Exit from the application Succesfully");

System.exit(0);

default:

System.out.println("Input correct value and retry");

}

}

break;

case 3:

System.out.println("Application Exit Successfully");

System.exit(0);

default:

System.out.println("Input correct value and retry");

break;

}

}

}

}