Lockedme source code

package com.simplilearn.lockme.application;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.LinkedHashMap;

import java.util.Scanner;

import com.simplilearn.lockme.model.UserCredentials;

import com.simplilearn.lockme.model.Users;

public class Authentication {

//input data

private static Scanner keyboard;

private static Scanner input;

private static Scanner lockerInput;

//output data

private static PrintWriter output;

private static PrintWriter lockerOutput;

//model to store data.

private static Users users;

private static UserCredentials userCredentials;

public static void main(String[] args) {

initApp();

welcomeScreen();

signInOptions();

}

public static void signInOptions() {

System.out.println("1 . Registration ");

System.out.println("2 . Login ");

int option = keyboard.nextInt();

switch(option) {

case 1 :

registerUser();

break;

case 2 :

loginUser();

break;

default :

System.out.println("Please select 1 Or 2");

break;

}

keyboard.close();

input.close();

}

public static void lockerOptions(String inpUsername) {

System.out.println("1 . FETCH ALL STORED CREDENTIALS ");

System.out.println("2 . STORED CREDENTIALS ");

int option = keyboard.nextInt();

switch(option) {

case 1 :

fetchCredentials(inpUsername);

break;

case 2 :

storeCredentials(inpUsername);

break;

default :

System.out.println("Please select 1 Or 2");

break;

}

lockerInput.close();

}

public static void registerUser() {

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

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

System.out.println("\* WELCOME TO REGISTRATION PAGE \*");

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

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

System.out.println("Enter Username :");

String username = keyboard.next();

if(!validateUser(username)) {

users.setUsername(username);

System.out.println("Enter password :");

String password = keyboard.next();

users.setPassword(password);

output.println(users.getUsername());

output.println(users.getPassword());

System.out.println("User Registration Suscessful !");

}else {

System.out.println("User Already Exist : 200OK, Please Try With Other UserName");

}

output.close();

}

public static boolean validateUser(String uname) {

/\*

\* boolean found = false; while(input.hasNext() && !found) {

\* if(input.next().equals(uname)) { found = true; break; } }

\*/

boolean found = false;

LinkedHashMap<String, String> existedusers = new LinkedHashMap<String, String>();

while(input.hasNext()) {

String existeduser = input.next().toLowerCase();

existedusers.put(existeduser, existeduser);

}

if(existedusers.containsKey(uname.toLowerCase())) {

found = true;

}

return found;

}

public static void loginUser() {

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

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

System.out.println("\* WELCOME TO LOGIN PAGE \*");

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

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

System.out.println("Enter Username :");

String inpUsername = keyboard.next();

boolean found = false;

while(input.hasNext() && !found) {

if(input.next().equals(inpUsername)) {

System.out.println("Enter Password :");

String inpPassword = keyboard.next();

if(input.next().equals(inpPassword)) {

System.out.println("Login Successful ! 200OK");

found = true;

lockerOptions(inpUsername);

break;

}

}

}

if(!found) {

System.out.println("User Not Found : Login Failure : 404");

}

}

public static void welcomeScreen() {

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

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

System.out.println("\* Welcome To LockMe.com \*");

System.out.println("\* Your Personal Digital Locaker \*");

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

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

}

//store credentails

public static void storeCredentials(String loggedInUser) {

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

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

System.out.println("\* WELCOME TO DIGITAL LOCKER STORE YOUR CREDS HERE \*");

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

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

userCredentials.setLoggedInUser(loggedInUser);

System.out.println("Enter Site Name :");

String siteName = keyboard.next();

userCredentials.setSiteName(siteName);

System.out.println("Enter Username :");

String username = keyboard.next();

userCredentials.setUsername(username);

userCredentials.toString();

System.out.println("Enter Password :");

String password = keyboard.next();

userCredentials.setPassword(password);

lockerOutput.println(userCredentials.getLoggedInUser());

System.out.println();

lockerOutput.println(userCredentials.getSiteName());

lockerOutput.println(userCredentials.getUsername());

lockerOutput.println(userCredentials.getPassword());

System.out.println("YOUR CREDS ARE STORED AND SECURED!");

lockerOutput.close();

}

//fetch credentials

public static void fetchCredentials(String inpUsername) {

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

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

System.out.println("\* WELCOME TO DIGITAL LOCKER \*");

System.out.println("\* YOUR CREDS ARE \*");

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

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

System.out.println(inpUsername);

while(lockerInput.hasNext()) {

//System.out.println(lockerInput.hasNext());

if(lockerInput.next().equals(inpUsername)) {

System.out.println(lockerInput.hasNext());

System.out.println("&&&&&&&&&&"+lockerInput.next().equals(inpUsername));

System.out.println("Site Name: "+lockerInput.next());

System.out.println("User Name: "+lockerInput.next());

System.out.println("User Password: "+lockerInput.next());

}

}

}

public static void initApp() {

File dbFile = new File("database.txt");

File lockerFile = new File("locker-file.txt");

try {

//read data from db file

input = new Scanner(dbFile);

//red data from locker file

lockerInput = new Scanner(lockerFile);

//read data from keyboard

keyboard = new Scanner(System.in);

//out put

output = new PrintWriter( new FileWriter(dbFile,true));

lockerOutput = new PrintWriter( new FileWriter(lockerFile,true));

users = new Users();

userCredentials = new UserCredentials();

} catch (IOException e) {

System.out.println("404 : data File Not Found "+e);

}

}

}

**package** com.simplilearn.lockme.model;

**public** **class** UserCredentials {

**private** String siteName;

**private** String loggedInUser;

**private** String username;

**private** String password;

**public** UserCredentials() {}

**public** UserCredentials(String siteName, String loggedInUser, String username, String password) {

**this**.siteName = siteName;

**this**.loggedInUser = loggedInUser;

**this**.username = username;

**this**.password = password;

}

**public** String getSiteName() {

**return** siteName;

}

**public** **void** setSiteName(String siteName) {

**this**.siteName = siteName;

}

**public** String getLoggedInUser() {

**return** loggedInUser;

}

**public** **void** setLoggedInUser(String loggedInUser) {

**this**.loggedInUser = loggedInUser;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

@Override

**public** String toString() {

**return** "UserCredentials [siteName=" + siteName +

", loggedInUser=" + loggedInUser +

", username=" + username

+ ", password=" + password + "]";

}

}

**package** com.simplilearn.lockme.model;

**public** **class** Users {

**private** String username;

**private** String password;

**public** Users() {}

**public** Users(String username, String password) {

**this**.username = username;

**this**.password = password;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

@Override

**public** String toString() {

**return** "Users [username=" + username + ", password=" + password + "]";

}

}