/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package javaapplication8;

/\*\*

\* st10476703

\* @ofentse

\*/

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.\*;

public class SimpleMessengerApp {

private static ArrayList<User> users = new ArrayList<>();

private static HashSet<String> blockedUsers = new HashSet<>();

private static String adminUsername = "ofentse";

private static String adminPassword = "password1";

public static void main(String[] args) {

//runUnitTests(); // You can comment this out if you don't want to run tests each time

boolean running = true;

while (running) {

String[] options = {"Register", "Login", "Exit"};

int choice = JOptionPane.showOptionDialog(null, "Welcome to SimpleMessenger", "Main Menu",

JOptionPane.DEFAULT\_OPTION, JOptionPane.INFORMATION\_MESSAGE, null, options, options[0]);

switch (choice) {

case 0 -> {

String result = registerUser();

JOptionPane.showMessageDialog(null, result);

}

case 1 -> loginUser();

default -> {

running = false;

JOptionPane.showMessageDialog(null, "Goodbye!");

}

}

}

}

static String registerUser() {

String username = JOptionPane.showInputDialog("Enter username (must contain '\_' and max 5 chars):");

if (!checkUserName(username)) {

return "Invalid username.";

}

for (User u : getUsers()) {

if (u.getUsername().equals(username)) {

return "Username already exists.";

}

}

String password = JOptionPane.showInputDialog("Enter password (letter, digit & special char required):");

if (!checkPasswordComplexity(password)) {

return "Password not complex enough.";

}

String confirmPassword = JOptionPane.showInputDialog("Confirm password:");

if (!password.equals(confirmPassword)) {

return "Passwords do not match.";

}

String phone = JOptionPane.showInputDialog("Enter 10-digit cell phone number:");

if (!checkCellPhoneNumber(phone)) {

return "Invalid phone number.";

}

String ageInput = JOptionPane.showInputDialog("Enter your age:");

int age;

try {

age = Integer.parseInt(ageInput);

if (age <= 0) {

return "Invalid age.";

}

} catch (NumberFormatException e) {

return "Invalid age.";

}

String gender = JOptionPane.showInputDialog("Enter your gender:");

if (gender == null || gender.trim().isEmpty()) {

return "Gender required.";

}

String ethnicity = JOptionPane.showInputDialog("Enter your ethnicity:");

if (ethnicity == null || ethnicity.trim().isEmpty()) {

return "Ethnicity required.";

}

String date = new SimpleDateFormat("yyyy-MM-dd").format(new Date());

// Save user

getUsers().add(new User(username, password, phone, age, gender, ethnicity, date));

return "Registration successful!";

}

static void loginUser() {

String username = JOptionPane.showInputDialog("Enter username:");

String password = JOptionPane.showInputDialog("Enter password:");

if (username == null || password == null) {

return;

}

if (username.equals(getAdminUsername()) && password.equals(getAdminPassword())) {

adminMenu();

} else {

for (User u : getUsers()) {

if (u.getUsername().equals(username) && u.getPassword().equals(password)) {

if (getBlockedUsers().contains(username)) {

JOptionPane.showMessageDialog(null, "You are blocked.");

} else {

JOptionPane.showMessageDialog(null, "Welcome, " + u.getUsername() + "!");

}

return;

}

}

JOptionPane.showMessageDialog(null, "Login failed.");

}

}

static void adminMenu() {

boolean adminRunning = true;

while (adminRunning) {

String[] options = {"View Users", "Block User", "Logout"};

int choice = JOptionPane.showOptionDialog(null, "Admin Menu", "Admin",

JOptionPane.DEFAULT\_OPTION, JOptionPane.INFORMATION\_MESSAGE, null, options, options[0]);

switch (choice) {

case 0 -> {

StringBuilder userList = new StringBuilder("Registered Users:\n");

for (User u : getUsers()) {

userList.append("- ").append(u.getUsername());

if (getBlockedUsers().contains(u.getUsername())) {

userList.append(" (Blocked)");

}

userList.append("\n");

} JOptionPane.showMessageDialog(null, userList.toString());

}

case 1 -> {

String userToBlock = JOptionPane.showInputDialog("Enter username to block:");

boolean found = false;

for (User u : getUsers()) {

if (u.getUsername().equals(userToBlock)) {

getBlockedUsers().add(userToBlock);

JOptionPane.showMessageDialog(null, userToBlock + " has been blocked.");

found = true;

break;

}

} if (!found) {

JOptionPane.showMessageDialog(null, "User not found.");

}

}

default -> adminRunning = false;

}

}

}

// ✅ Boolean methods

static boolean checkUserName(String username) {

return username != null && username.contains("\_") && username.length() <= 5;

}

static boolean checkPasswordComplexity(String password) {

if (password == null || password.length() < 6) {

return false;

}

boolean hasLetter = false, hasDigit = false, hasSpecial = false;

for (char c : password.toCharArray()) {

if (Character.isLetter(c)) {

hasLetter = true;

} else if (Character.isDigit(c)) {

hasDigit = true;

} else {

hasSpecial = true;

}

}

return hasLetter && hasDigit && hasSpecial;

}

static boolean checkCellPhoneNumber(String number) {

return number != null && number.matches("\\d{10}");

}

// ✅ Test block

static void runUnitTests() {

System.out.println("Running unit tests...");

assert checkUserName("usr\_") : "Username check failed (valid)";

assert !checkUserName("userlong") : "Username check failed (too long)";

assert !checkUserName("user") : "Username check failed (missing '\_')";

assert checkPasswordComplexity("A1@bcde") : "Password check failed (valid)";

assert !checkPasswordComplexity("abcdef") : "Password check failed (no digit, special)";

assert !checkPasswordComplexity("abc123") : "Password check failed (no special char)";

assert checkCellPhoneNumber("0123456789") : "Phone check failed (valid)";

assert !checkCellPhoneNumber("123456") : "Phone check failed (too short)";

assert !checkCellPhoneNumber("phone123") : "Phone check failed (invalid chars)";

System.out.println("All unit tests passed!");

}

/\*\*

\* @return the users

\*/

public static ArrayList<User> getUsers() {

return users;

}

/\*\*

\* @param aUsers the users to set

\*/

public static void setUsers(ArrayList<User> aUsers) {

users = aUsers;

}

/\*\*

\* @return the blockedUsers

\*/

public static HashSet<String> getBlockedUsers() {

return blockedUsers;

}

/\*\*

\* @param aBlockedUsers the blockedUsers to set

\*/

public static void setBlockedUsers(HashSet<String> aBlockedUsers) {

blockedUsers = aBlockedUsers;

}

/\*\*

\* @return the adminUsername

\*/

public static String getAdminUsername() {

return adminUsername;

}

/\*\*

\* @param aAdminUsername the adminUsername to set

\*/

public static void setAdminUsername(String aAdminUsername) {

adminUsername = aAdminUsername;

}

/\*\*

\* @return the adminPassword

\*/

public static String getAdminPassword() {

return adminPassword;

}

/\*\*

\* @param aAdminPassword the adminPassword to set

\*/

public static void setAdminPassword(String aAdminPassword) {

adminPassword = aAdminPassword;

}

}

// ✅ User class

class User {

private String username;

private String password;

private String phone;

private String gender;

private String ethnicity;

private String regDate;

private int age;

public User(String username, String password, String phone, int age, String gender, String ethnicity, String regDate) {

this.username = username;

this.password = password;

this.phone = phone;

this.age = age;

this.gender = gender;

this.ethnicity = ethnicity;

this.regDate = regDate;

}

/\*\*

\* @return the username

\*/

public String getUsername() {

return username;

}

/\*\*

\* @param username the username to set

\*/

public void setUsername(String username) {

this.username = username;

}

/\*\*

\* @return the password

\*/

public String getPassword() {

return password;

}

/\*\*

\* @param password the password to set

\*/

public void setPassword(String password) {

this.password = password;

}

/\*\*

\* @return the phone

\*/

public String getPhone() {

return phone;

}

/\*\*

\* @param phone the phone to set

\*/

public void setPhone(String phone) {

this.phone = phone;

}

/\*\*

\* @return the gender

\*/

public String getGender() {

return gender;

}

/\*\*

\* @param gender the gender to set

\*/

public void setGender(String gender) {

this.gender = gender;

}

/\*\*

\* @return the ethnicity

\*/

public String getEthnicity() {

return ethnicity;

}

/\*\*

\* @param ethnicity the ethnicity to set

\*/

public void setEthnicity(String ethnicity) {

this.ethnicity = ethnicity;

}

/\*\*

\* @return the regDate

\*/

public String getRegDate() {

return regDate;

}

/\*\*

\* @param regDate the regDate to set

\*/

public void setRegDate(String regDate) {

this.regDate = regDate;

}

/\*\*

\* @return the age

\*/

public int getAge() {

return age;

}

/\*\*

\* @param age the age to set

\*/

public void setAge(int age) {

this.age = age;

}

}