import java.util.ArrayList;

import java.util.Scanner;

class User {

String name;

int userID;

double currentMonthUnits;

double billAmount;

boolean isBillPaid;

public User(String name, int userID) {

this.name = name;

this.userID = userID;

this.currentMonthUnits = 0;

this.billAmount = 0;

this.isBillPaid = false;

}

public void addUnits(double units) {

this.currentMonthUnits = units;

}

public void calculateBill() {

double ratePerUnit = 5.0; // Assuming the rate per unit is 5

this.billAmount = this.currentMonthUnits \* ratePerUnit;

this.isBillPaid = false;

}

public void payBill() {

if (!isBillPaid) {

System.out.println("Bill of amount " + billAmount + " paid successfully!");

this.isBillPaid = true;

} else {

System.out.println("Bill is already paid.");

}

}

public void viewBill() {

System.out.println("User: " + name + ", UserID: " + userID);

System.out.println("Units consumed: " + currentMonthUnits);

System.out.println("Bill amount: " + billAmount);

System.out.println("Bill paid: " + (isBillPaid ? "Yes" : "No"));

}

}

public class EBManagement {

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

static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

while (true) {

System.out.println("1. Add User");

System.out.println("2. Add Consumption Units");

System.out.println("3. Calculate Bill");

System.out.println("4. View Bill");

System.out.println("5. Pay Bill");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

addUser();

break;

case 2:

addConsumptionUnits();

break;

case 3:

calculateBill();

break;

case 4:

viewBill();

break;

case 5:

payBill();

break;

case 6:

System.exit(0);

break;

default:

System.out.println("Invalid choice. Try again.");

}

}

}

private static void addUser() {

System.out.print("Enter user name: ");

String name = scanner.next();

System.out.print("Enter user ID: ");

int userID = scanner.nextInt();

users.add(new User(name, userID));

System.out.println("User added successfully.");

}

private static void addConsumptionUnits() {

System.out.print("Enter user ID: ");

int userID = scanner.nextInt();

User user = findUserByID(userID);

if (user != null) {

System.out.print("Enter units consumed: ");

double units = scanner.nextDouble();

user.addUnits(units);

System.out.println("Units added successfully.");

} else {

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

}

}

private static void calculateBill() {

System.out.print("Enter user ID: ");

int userID = scanner.nextInt();

User user = findUserByID(userID);

if (user != null) {

user.calculateBill();

System.out.println("Bill calculated successfully.");

} else {

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

}

}

private static void viewBill() {

System.out.print("Enter user ID: ");

int userID = scanner.nextInt();

User user = findUserByID(userID);

if (user != null) {

user.viewBill();

} else {

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

}

}

private static void payBill() {

System.out.print("Enter user ID: ");

int userID = scanner.nextInt();

User user = findUserByID(userID);

if (user != null) {

user.payBill();

} else {

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

}

}

private static User findUserByID(int userID) {

for (User user : users) {

if (user.userID == userID) {

return user;

}

}

return null;

}

}