**package** dxc.com.daa.dao;

**import** com.dxc.daa.model.Users;

**public** **interface** UsersDAO {

**public** **boolean** validate(String username,String password);

}

package dxc.com.daa.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.dxc.daa.dbcon.DBConnection;

import com.dxc.daa.model.Users;

public class UsersDAOImp implements UsersDAO {

Connection connection = DBConnection.getConnection();

private static final String FETCH\_USERS\_ALL = "select\*from users";

private static final String FETCH\_USERS = "select \* from users where username = ? and password = ?";

@Override

public boolean validate(String username, String password) {

boolean userExists = false;

PreparedStatement preparedStatement;

try {

preparedStatement = connection.prepareStatement(FETCH\_USERS);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

ResultSet res = preparedStatement.executeQuery();

if(res.next()) {

userExists = true;

}

} catch (SQLException e) {

e.printStackTrace();

}

return userExists;

}

}

**package** dxc.com.daa.dao;

**import** java.util.List;

**import** com.dxc.daa.model.Training;

**public** **interface** TrainingDAO {

**public** List<Training>displayrecords();

**public** **void** updateTraining( **int** sapid,**int** percentage);

}

**package** dxc.com.daa.dao;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.dxc.daa.dbcon.DBConnection;

**import** com.dxc.daa.model.Training;

**public** **class** TrainingDAOImp **implements** TrainingDAO {

Connection connection = DBConnection.*getConnection*();

List<Training>allTrainings = **new** ArrayList<Training>();

**private** **static** **final** String ***FETCH\_TRAINING\_ALL*** = "select \* from training";

**private** **static** **final** String ***RECORD\_UPDATE\_QUERY*** = "update training set percentage=? where SapId=?";

**public** TrainingDAOImp() {

**super**();

}

@Override

**public** List<Training> displayrecords() {

List<Training>allTrainings = **new** ArrayList<Training>();

ResultSet res;

**try** {

Statement stat = connection.createStatement();

res = stat.executeQuery(***FETCH\_TRAINING\_ALL***);

**while**(res.next()) {

Training training = **new** Training();

training.setSapId(res.getInt(1));

training.setEmployeeName(res.getString(2));

training.setStream(res.getString(3));

training.setPercentage(res.getInt(4));

allTrainings.add(training);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** allTrainings;

}

@Override

**public** **void** updateTraining(**int** sapid, **int** percentage) {

**try** {

PreparedStatement preparedStatment = connection.prepareStatement(***RECORD\_UPDATE\_QUERY***);

preparedStatment.setInt(1,percentage);

preparedStatment.setInt(2,sapid);

preparedStatment.executeUpdate();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**package** com.dxc.daa.model;

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

**private** String username;

**private** String password;

**public** Users() {

**super**();

}

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

**super**();

**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 + "]";

}

}

**package** com.dxc.daa.model;

**public** **class** Training {

**int** sapId;

String employeeName;

String stream;

**int** percentage;

**public** Training() {

**super**();

}

**public** Training(**int** sapId, String employeeName, String stream, **int** percentage) {

**super**();

**this**.sapId = sapId;

**this**.employeeName = employeeName;

**this**.stream = stream;

**this**.percentage = percentage;

}

**public** **int** getSapId() {

**return** sapId;

}

**public** **void** setSapId(**int** sapId) {

**this**.sapId = sapId;

}

**public** String getEmployeeName() {

**return** employeeName;

}

**public** **void** setEmployeeName(String employeeName) {

**this**.employeeName = employeeName;

}

**public** String getStream() {

**return** stream;

}

**public** **void** setStream(String stream) {

**this**.stream = stream;

}

**public** **int** getPercentage() {

**return** percentage;

}

**public** **void** setPercentage(**int** percentage) {

**this**.percentage = percentage;

}

@Override

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + ((employeeName == **null**) ? 0 : employeeName.hashCode());

result = prime \* result + percentage;

result = prime \* result + sapId;

result = prime \* result + ((stream == **null**) ? 0 : stream.hashCode());

**return** result;

}

@Override

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (getClass() != obj.getClass())

**return** **false**;

Training other = (Training) obj;

**if** (employeeName == **null**) {

**if** (other.employeeName != **null**)

**return** **false**;

} **else** **if** (!employeeName.equals(other.employeeName))

**return** **false**;

**if** (percentage != other.percentage)

**return** **false**;

**if** (sapId != other.sapId)

**return** **false**;

**if** (stream == **null**) {

**if** (other.stream != **null**)

**return** **false**;

} **else** **if** (!stream.equals(other.stream))

**return** **false**;

**return** **true**;

}

@Override

**public** String toString() {

**return** "Training [sapId=" + sapId + ", employeeName=" + employeeName + ", stream=" + stream + ", percentage="

+ percentage + "]";

}}

===========================

package com.dxc.daa.dbcon;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

public DBConnection() {

}

public static Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Connection connection = null;

try {

connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/dxc", "root", "root");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return connection;

}

}

**package** com.dxc.daa.client;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Scanner;

**import** com.dxc.daa.model.Training;

**import** dxc.com.daa.dao.TrainingDAO;

**import** dxc.com.daa.dao.TrainingDAOImp;

**import** dxc.com.daa.dao.UsersDAO;

**import** dxc.com.daa.dao.UsersDAOImp;

**public** **class** AssessApp {

UsersDAO usersDAO;

**int** choice = 0;

String userName;

String passWord;

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

**public** **void** launchAssessApp() {

UsersDAO usersDAO = **new** UsersDAOImp();

System.***out***.println("please enter userName:");

userName = scanner.next();

System.***out***.println("please enter password");

passWord = scanner.next();

**if** (usersDAO.validate(userName, passWord)) {

System.***out***.println("access granted");

}

**else** {

System.***out***.println("invalid credential");

System.*exit*(0);

}

**while** (**true**) {

System.***out***.println("M E N U ");

System.***out***.println("1. Display All Training Records : ");

System.***out***.println("2. Display Records one by One and update the percentage : ");

System.***out***.println("3. E X I T ");

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

System.***out***.println("Please enter your choice : (1-3)");

choice = scanner.nextInt();

TrainingDAO trainingDAO = **new** TrainingDAOImp();

**switch** (choice) {

**case** 1:

System.***out***.println(trainingDAO.displayrecords());

**break**;

**case** 2:

List<Training>record = **new** ArrayList<Training>();

record = trainingDAO.displayrecords();

Iterator<Training>iterator = record.iterator();

**while**(iterator.hasNext()) {

Training training = **new** Training();

training = iterator.next();

System.***out***.println(training.toString());

**if**(training.getPercentage() == 0) {

System.***out***.println("enter percentage:");

**int** percentage = scanner.nextInt();

trainingDAO.updateTraining(training.getSapId(),percentage);

}

**else** {

System.***out***.println("percentage already enterd");

}

}

System.***out***.println("Welcome to training app Update");

**break**;

}

}

}

}

**package** com.dxc.daa.client;

**public** **class** Main {

**public** Main() {

}

**public** **static** **void** main(String[] args) {

AssessApp app = **new** AssessApp();

app.launchAssessApp();

}

}