package com.zumba.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import com.zumba.bean.Event;

import com.zumba.bean.Member;

import com.zumba.resource.DbConnection;

public class EventDao {

public List<Event> viewAllevent() {

List<Event> listOfevent = new ArrayList<Event>();

try {

Connection con = DbConnection.getDbConnection();

PreparedStatement pstmt = con.prepareStatement("select \* from Event");

ResultSet rs = pstmt.executeQuery();

while(rs.next()) {

Event currentEvent = new Event();

currentEvent.setEID(rs.getInt("EID"));

currentEvent.setDay(rs.getString("day"));

currentEvent.setTime(rs.getString("time"));

currentEvent.setLocation(rs.getString("location"));

listOfevent.add(currentEvent);

}

} catch (Exception e) {

System.err.println(e.toString());

}

return listOfevent;

}

public List<Event> viewEventsForMember(int MID) {

List<Event> eventsForMember = new ArrayList<Event>();

try {

Connection con = DbConnection.getDbConnection();

PreparedStatement pstmt = con.prepareStatement("select \* from Event inner join MemberEvent on Event.EID = MemberEvent.EID where MID = (?)");

pstmt.setInt(1, MID);

ResultSet rs2 = pstmt.executeQuery();

while(rs2.next()) {

Event currentEvent = new Event();

currentEvent.setEID(rs2.getInt("EID"));

currentEvent.setDay(rs2.getString("day"));

currentEvent.setTime(rs2.getString("time"));

currentEvent.setLocation(rs2.getString("location"));

eventsForMember.add(currentEvent);

}

} catch (Exception e) {

System.err.println(e.toString());

}

return eventsForMember;

}

public Event viewSpecificEvent(int EID) {

Event currentSpecificEvent = null;

try {

Connection con = DbConnection.getDbConnection();

PreparedStatement pstmt = con.prepareStatement("select \* from Event where EID = (?)");

pstmt.setInt(1, EID);

ResultSet rs3 = pstmt.executeQuery();

while(rs3.next()) {

currentSpecificEvent = new Event();

currentSpecificEvent.setEID(rs3.getInt("EID"));

currentSpecificEvent.setDay(rs3.getString("day"));

currentSpecificEvent.setTime(rs3.getString("time"));

currentSpecificEvent.setLocation(rs3.getString("location"));

}

} catch (Exception e) {

System.err.println(e.toString());

}

return currentSpecificEvent;

}

public void addNewEvent(Event newEvent) {

try {

Connection con = DbConnection.getDbConnection();

PreparedStatement pstmt = con.prepareStatement("insert into Event(day, time, location) values (?,?,?)");

pstmt.setString(1, newEvent.getDay());

pstmt.setString(2, newEvent.getTime());

pstmt.setString(3, newEvent.getLocation());

int intResult = pstmt.executeUpdate();

System.out.println("Event Dao addNewEvent=" +intResult);

}catch (Exception e) {

System.err.println(e.toString());

}

}

public void removeEvent(Event event) {

try {

Connection con = DbConnection.getDbConnection();

PreparedStatement pstmt1 = con.prepareStatement("delete from MemberEvent where EID = (?)");

PreparedStatement pstmt2 = con.prepareStatement("delete from Event where EID = (?)");

pstmt1.setInt(1, event.getEID());

pstmt2.setInt(1, event.getEID());

int intResult1 = pstmt1.executeUpdate();

int intResult2 = pstmt2.executeUpdate();

}catch (Exception e) {

System.err.println(e.toString());

}

}

}