DAL Package

BaseDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\*

\* @author massami

\*/

public class BaseDAL {

private static final String DATABASE\_NAME = "jdbc:mysql://localhost:3306/librarysystem\_db?useSSL=false, root, root";

static Connection con;

/\*\*

\* Get the configuration for the data base connection

\* @return the connection to the database

\* @throws Exception

\*/

public static Connection getConnection() throws Exception {

try {

if (con == null) {

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

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/librarysystem\_db", "root", "root");

}

return con;

} catch (SQLException ex) {

throw new Exception(ex.getMessage());

}

}

/\*\*

\* Prepare the call to the database

\* @param sql Query

\* @return statement

\* @throws Exception

\*/

public static CallableStatement getStatement(String sql) throws Exception{

Connection con = getConnection();

return con.prepareCall(sql);

}

}

BookDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import Model.Book;

import Model.Catalogue;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

/\*\*

\*

\* @author massami

\*/

public class BookDAL extends BaseDAL {

/\*\*

\* Insert new book to the database

\* @param book new object to be inserted

\* @throws Exception

\*/

public static void addBook(Book book) throws Exception {

String sql = "call sp\_addBook(?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setString(1, book.getAuthor());

st.setString(2, book.getGenre());

st.setString(3, book.getIsbn());

st.setString(4, book.getTitle());

st.setString(5, book.getDescription());

st.setString(6, book.getReleaseDate());

st.setString(7, book.getStatus());

st.setString(8, book.getNote());

st.setString(9, book.getLanguage());

st.executeQuery();

}

/\*\*

\* Update the information of the book

\* @param book object to be updated

\* @throws Exception

\*/

public static void updateBook(Book book) throws Exception {

String sql = "call sp\_updateBook(?,?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setInt(1, book.getId());

st.setString(2, book.getAuthor());

st.setString(3, book.getGenre());

st.setString(4, book.getIsbn());

st.setString(5, book.getTitle());

st.setString(6, book.getDescription());

st.setString(7, book.getReleaseDate());

st.setString(8, book.getStatus());

st.setString(9, book.getNote());

st.setString(10, book.getLanguage());

st.executeQuery();

}

/\*\*

\* Get all the books

\* @return all the books

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAllBook() throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchBook()";

ResultSet rs = getStatement(sql).executeQuery();

while (rs.next()) {

Book book = new Book(rs.getInt("catalogue\_id"), rs.getInt("book\_id"), rs.getString("author"), rs.getString("genre"), rs.getString("isbn"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(book);

}

return catalogueList;

}

/\*\*

\* Get the book filtered by the title

\* @param title property to be searched

\* @return list of books with the selected title

\* @throws Exception

\*/

public static ArrayList<Catalogue> getBookByTitle(String title) throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchBookByTitle(?)";

CallableStatement st = getStatement(sql);

st.setString(1, title);

ResultSet rs = st.executeQuery();

while (rs.next()) {

Book book = new Book(rs.getInt("catalogue\_id"), rs.getInt("book\_id"), rs.getString("author"), rs.getString("genre"), rs.getString("isbn"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(book);

}

return catalogueList;

}

/\*\*

\* Get only the available books for borrowing

\* @return list of available books

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAvailableBook() throws Exception{

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchAvailableBook()";

ResultSet rs = getStatement(sql).executeQuery();

while (rs.next()) {

Book book = new Book(rs.getInt("catalogue\_id"), rs.getInt("book\_id"), rs.getString("author"), rs.getString("genre"), rs.getString("isbn"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(book);

}

return catalogueList;

}

/\*\*

\* Get available books by the filtered title

\* @param title filter

\* @return

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAvailableBookByTitle(String title) throws Exception{

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchAvailableBookByTitle(?)";

CallableStatement st = getStatement(sql);

st.setString(1, title);

ResultSet rs = st.executeQuery();

while (rs.next()) {

Book book = new Book(rs.getInt("catalogue\_id"), rs.getInt("book\_id"), rs.getString("author"), rs.getString("genre"), rs.getString("isbn"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(book);

}

return catalogueList;

}

/\*\*

\* Book logical delete

\* @param id id of the book

\* @throws Exception

\*/

public static void deleteBook(int id) throws Exception {

String sql = "call sp\_deleteBook(?)";

CallableStatement st = getStatement(sql);

st.setInt(1, id);

st.executeQuery();

}

}

BorrowDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import static DAL.BaseDAL.getStatement;

import Model.Book;

import Model.Borrow;

import Model.Catalogue;

import java.sql.CallableStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

/\*\*

\*

\* @author massami

\*/

public class BorrowDAL extends BaseDAL{

/\*\*

\* Save data of the borrowed transaction

\* @param borrow information about the borrow

\* @throws Exception

\*/

public static void addBorrow(Borrow borrow) throws Exception{

String sql = "call sp\_addBorrow(?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setString(1, borrow.getBorrowDate());

st.setString(2, borrow.getDueDate());

st.setString(3, borrow.getReturnDate());

st.setInt(4, borrow.getMemberId());

st.setInt(5, borrow.getCatalogueId());

st.executeQuery();

}

/\*\*

\* Save data of the returned item

\* @param borrow data of the returned item

\* @throws Exception

\*/

public static void returnItem(Borrow borrow ) throws Exception{

String sql = "call sp\_returnItem(?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setString(1, borrow.getBorrowDate());

st.setString(2, borrow.getDueDate());

st.setString(3, borrow.getReturnDate());

st.setInt(4, borrow.getMemberId());

st.setInt(5, borrow.getCatalogueId());

st.executeQuery();

}

/\*\*

\* Get the ID of the selected member

\* @param name of the member

\* @return id

\* @throws Exception

\*/

public static int getMemberId(String name) throws Exception {

String sql = "call sp\_searchMemberId(?)";

CallableStatement st = getStatement(sql);

st.setString(1, name);

ResultSet rs = st.executeQuery();

if(rs.next())

return rs.getInt("member\_id");

else

return 0;

}

/\*\*

\* Get the items that were borrow by the members

\* @param name

\* @return list of borrowed items

\* @throws Exception

\*/

public static ArrayList<Catalogue> getBorrowedItem(String name) throws Exception{

ArrayList<Catalogue> catalogueList = new ArrayList<Catalogue>();

String sql = "call sp\_searchBorrowedItem(?)";

CallableStatement st = getStatement(sql);

st.setString(1, name);

ResultSet rs = st.executeQuery();

while(rs.next()){

Catalogue catalogue = new Book();

catalogue.setId(rs.getInt("catalogue\_id"));

catalogue.setTitle(rs.getString("title"));

catalogue.setDescription( rs.getString("description"));

catalogue.setReleaseDate(rs.getString("release\_date"));

catalogue.setStatus(rs.getString("status"));

catalogue.setNote(rs.getString("note"));

catalogue.setLanguage(rs.getString("language"));

catalogueList.add(catalogue);

}

return catalogueList;

}

}

DvdDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import Model.Catalogue;

import Model.Dvd;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.ResultSet;

import java.util.ArrayList;

/\*\*

\*

\* @author massami

\*/

public class DvdDAL extends BaseDAL {

/\*\*

\* Insert DVD information to the database

\* @param dvd

\* @throws Exception

\*/

public static void addDvd(Dvd dvd) throws Exception {

String sql = "call sp\_addDvd(?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setString(1, dvd.getDistributor());

st.setString(2, dvd.getCast());

st.setString(3, dvd.getCredits());

st.setString(4, dvd.getTitle());

st.setString(5, dvd.getDescription());

st.setString(6, dvd.getReleaseDate());

st.setString(7, dvd.getStatus());

st.setString(8, dvd.getNote());

st.setString(9, dvd.getLanguage());

st.executeQuery();

}

/\*\*

\* Update the information of the DVD to the database

\* @param dvd

\* @throws Exception

\*/

public static void updateDvd(Dvd dvd) throws Exception {

String sql = "call sp\_updateDvd(?,?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setInt(1, dvd.getId());

st.setString(2, dvd.getDistributor());

st.setString(3, dvd.getCast());

st.setString(4, dvd.getCredits());

st.setString(5, dvd.getTitle());

st.setString(6, dvd.getDescription());

st.setString(7, dvd.getReleaseDate());

st.setString(8, dvd.getStatus());

st.setString(9, dvd.getNote());

st.setString(10, dvd.getLanguage());

st.executeQuery();

}

/\*\*

\* Get the list of all DVDs

\* @return list of DVDs

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAllDvd() throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchDvd()";

ResultSet rs = getStatement(sql).executeQuery();

while (rs.next()) {

Dvd dvd = new Dvd(rs.getInt("catalogue\_id") ,rs.getInt("dvd\_id"), rs.getString("distributor"), rs.getString("cast"), rs.getString("credits"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(dvd);

}

return catalogueList;

}

/\*\*

\* Get the list of DVDs filtered by the title

\* @param title filter

\* @return list of DVDs

\* @throws Exception

\*/

public static ArrayList<Catalogue> getDvdByTitle(String title) throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchDvdByTitle(?)";

CallableStatement st = getStatement(sql);

st.setString(1, title);

ResultSet rs = st.executeQuery();

while (rs.next()) {

Dvd dvd = new Dvd(rs.getInt("catalogue\_id"), rs.getInt("dvd\_id"), rs.getString("distributor"), rs.getString("cast"), rs.getString("credits"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(dvd);

}

return catalogueList;

}

/\*\*

\* Get the list of available DVDs for borrowing

\* @return list of DVDs

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAvailableDvd() throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchAvailableDvd()";

ResultSet rs = getStatement(sql).executeQuery();

while (rs.next()) {

Dvd dvd = new Dvd(rs.getInt("catalogue\_id") ,rs.getInt("dvd\_id"), rs.getString("distributor"), rs.getString("cast"), rs.getString("credits"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(dvd);

}

return catalogueList;

}

/\*\*

\* Get the list of available DVDs filtered by title

\* @param title filter

\* @return list of DVDs

\* @throws Exception

\*/

public static ArrayList<Catalogue> getAvailableDvdByTitle(String title) throws Exception {

ArrayList<Catalogue> catalogueList = new ArrayList();

String sql = "call sp\_searchAvailableDvdByTitle(?)";

CallableStatement st = getStatement(sql);

st.setString(1, title);

ResultSet rs = st.executeQuery();

while (rs.next()) {

Dvd dvd = new Dvd(rs.getInt("catalogue\_id"), rs.getInt("dvd\_id"), rs.getString("distributor"), rs.getString("cast"), rs.getString("credits"), rs.getString("title"), rs.getString("description"), rs.getString("release\_date"), rs.getString("status"), rs.getString("note"), rs.getString("language"));

catalogueList.add(dvd);

}

return catalogueList;

}

/\*\*

\* Local delete of the DVD object

\* @param id

\* @throws Exception

\*/

public static void deleteDvd(int id) throws Exception {

String sql = "call sp\_deleteDvd(?)";

CallableStatement st = getStatement(sql);

st.setInt(1, id);

st.executeQuery();

}

}

HistoryDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import static DAL.BaseDAL.getStatement;

import Model.History;

import java.sql.CallableStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

/\*\*

\*

\* @author massami

\*/

public class HistoryDAL extends BaseDAL{

/\*\*

\* Get the transaction history of the selected member

\* @param name

\* @return List of borrowed items

\* @throws Exception

\*/

public static ArrayList<History> getTransactionHistory(String name) throws Exception{

ArrayList<History> historyList = new ArrayList();

String sql = "call sp\_searchTransactionHistory(?)";

CallableStatement st = getStatement(sql);

st.setString(1, name);

ResultSet rs = st.executeQuery();

while (rs.next()) {

History history = new History(rs.getString("title"), rs.getString("due\_date"), rs.getString("borrow\_date"), rs.getString("name"));

historyList.add(history);

}

return historyList;

}

}

MemeberDAL

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DAL;

import Model.Member;

import Model.Person;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Hashtable;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Rom

\*/

public class MemberDAL extends BaseDAL {

/\*\*

\* Insert the member to the database

\* @param member selected member

\* @throws Exception

\*/

public static void addMember(Member member) throws Exception {

String sql = "call sp\_addMember(?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setString(1, member.getFirstName());

st.setString(2, member.getLastName());

st.setString(3, member.getEmailAddress());

st.setString(4, member.getContactNumber());

st.setString(5, member.getBirthdate());

st.setString(6, member.getAddress());

st.setString(7, member.getSuburb());

st.setString(8, member.getCity());

st.setString(9, member.getPostalCode());

st.executeQuery();

}

/\*\*

\* Update the information of the member

\* @param member object to be updated

\* @throws Exception

\*/

public static void updateMember(Member member) throws Exception {

String sql = "call sp\_updateMember(?,?,?,?,?,?,?,?,?,?)";

CallableStatement st = getStatement(sql);

st.setInt(1, member.getId());

st.setString(2, member.getFirstName());

st.setString(3, member.getLastName());

st.setString(4, member.getEmailAddress());

st.setString(5, member.getContactNumber());

st.setString(6, member.getBirthdate());

st.setString(7, member.getAddress());

st.setString(8, member.getSuburb());

st.setString(9, member.getCity());

st.setString(10, member.getPostalCode());

st.executeQuery();

}

/\*\*

\* Get the list of members

\* @return list of members

\* @throws Exception

\*/

public static ArrayList<Person> getAllMember() throws Exception {

ArrayList<Person> personList = new ArrayList();

String sql = "call sp\_searchMember()";

ResultSet rs = getStatement(sql).executeQuery();

while (rs.next()) {

Member member = new Member(rs.getInt("person\_id"), rs.getString("birthdate"), rs.getString("address"), rs.getString("suburb"), rs.getString("city"), rs.getString("postal\_code"), rs.getString("first\_name"), rs.getString("last\_name"), rs.getString("email\_address"), rs.getString("contact\_number"));

personList.add(member);

}

return personList;

}

/\*\*

\* Get the list of members filtered by the name

\* @param name filter

\* @return list of members

\* @throws Exception

\*/

public static ArrayList<Person> getMemberByName(String name) throws Exception {

ArrayList<Person> personList = new ArrayList();

String sql = "call sp\_searchMemberByName(?)";

CallableStatement st = getStatement(sql);

st.setString(1, name);

ResultSet rs = st.executeQuery();

while (rs.next()) {

Member member = new Member(rs.getInt("person\_id"), rs.getString("birthdate"), rs.getString("address"), rs.getString("suburb"), rs.getString("city"), rs.getString("postal\_code"), rs.getString("first\_name"), rs.getString("last\_name"), rs.getString("email\_address"), rs.getString("contact\_number"));

personList.add(member);

}

return personList;

}

/\*\*

\* Deactivation of the selected member

\* @param id

\* @throws Exception

\*/

public static void deactivateMember(int id) throws Exception {

String sql = "call sp\_deactivateMember(?)";

CallableStatement st = getStatement(sql);

st.setInt(1, id);

st.executeQuery();

}

/\*\*

\* Get the list of members filtered by name

\* @return list of members filtered by the name

\* @throws Exception

\*/

public static ArrayList<String> getAllMemberName() throws Exception {

ArrayList<String> personList = new ArrayList();

String sql = "call sp\_searchPerson()";

CallableStatement st = getStatement(sql);

ResultSet rs = st.executeQuery();

while (rs.next()) {

personList.add(rs.getString("name"));

}

return personList;

}

}

UserDAL