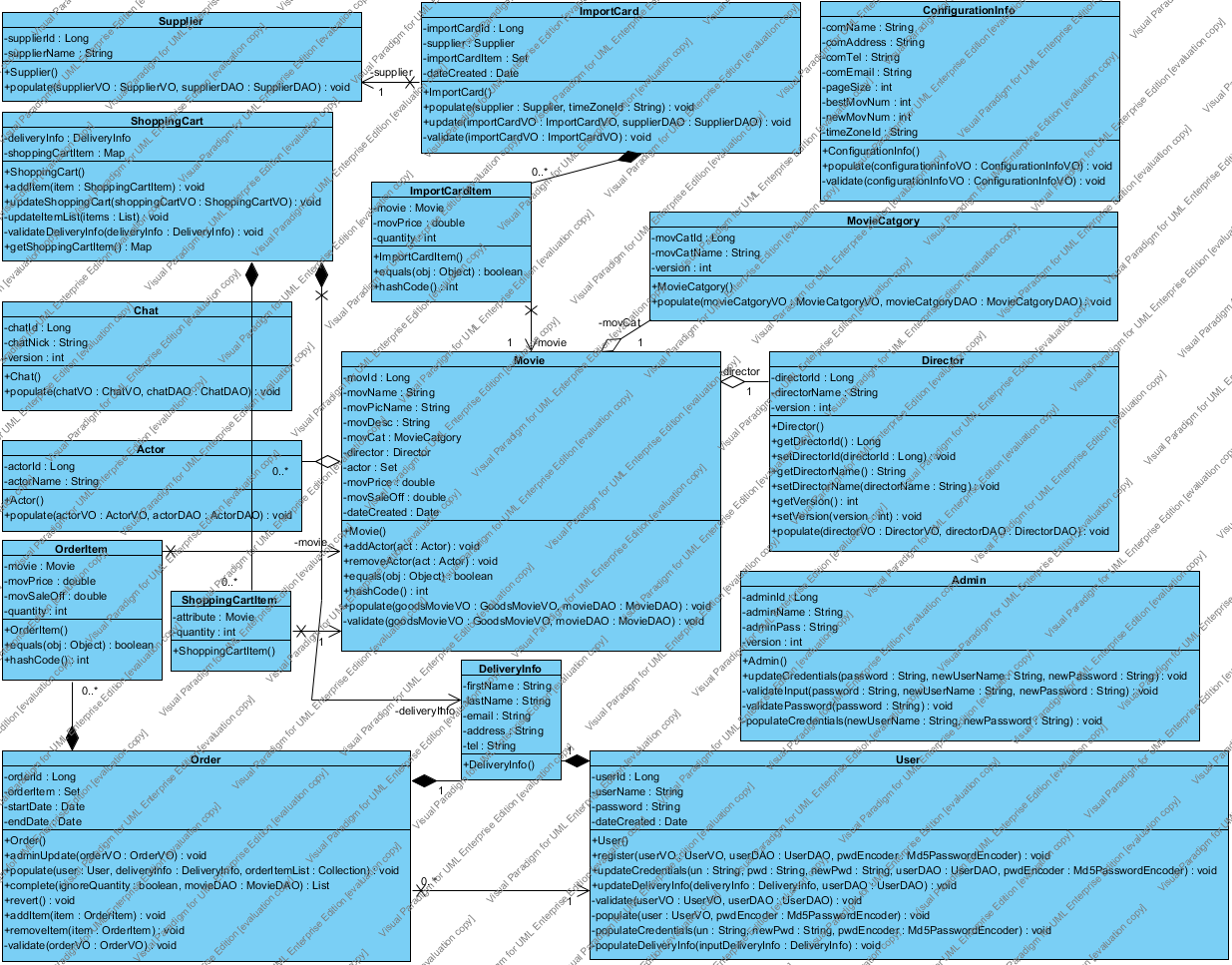
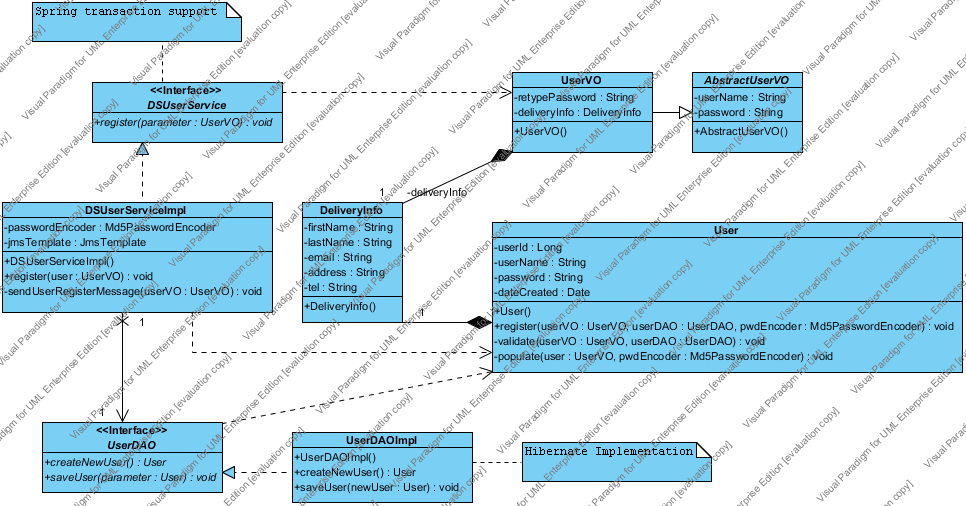
**3.3. Sơ đồ lớp:**

Domain Model Classes



DangKy\_BL

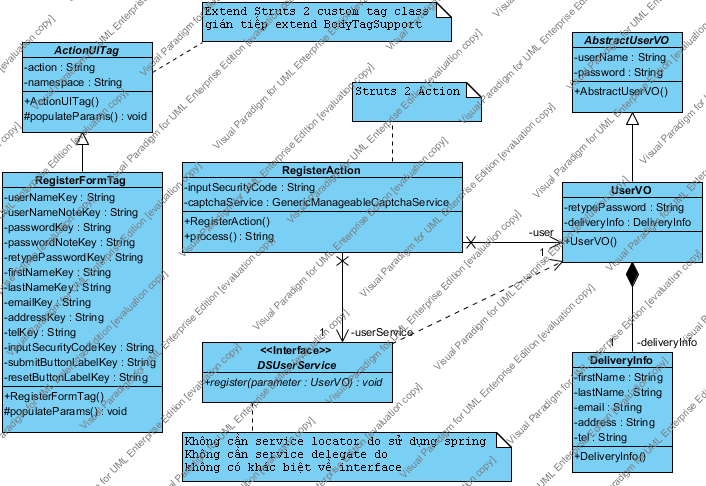


- Sử dụng Spring Transaction Support: JTA, Hibernate Transaction, JDBC Transaction. Nếu không sử dụng spring, có thể phải hard code 🡪 thay đổi khó, spring chỉ config qua xml, về nguyên tắc, spring tạo 1 proxy wrap lại object lúc init application context. Proxy này có nhiệm vụ start transaction lúc bắt đầu thực hiện method, gọi method của wrapped object, và commit transaction khi method không throw exception, trường hợp có throw exception thì rollback transaction.

- Đây chính là khái niệm AOP (Aspect Orient Programming) developer chỉ cần tập trung xây dựng bussiness logic, không cần quan tâm tới security hay transaction. Việc áp dụng AOP giúp dễ dàng hơn cho quá trình phát triển, cũng giúp dễ dàng hơn khi maintainent, và unit test (chỉ test bussiness logic, không cần test security hay transaction).

- User.register() nhận tham số UserDAO do đây là entity 🡪 do DAO sinh, 1 số ORM framework có khả năng tạo entity với constructor có tham số, tuy nhiên nếu làm vậy có thể bị gắn với 1 số framework nhất định, vậy đưa dependency vào dưới dạng tham số (spring chỉ quản lý service và dao object, entity do hibernate framework quản lý, do vậy không thể inject, cũng không nên tự tạo dependency bằng toán tử new do không thể unit test).

- Có thể thấy 2 phương thức private validate và populate (phương thức cập nhật instance variable) chính là 2 phương thức chính chưa bussiness logic được encapsulate trong User. Tạo ra khả năng sử dụng lại code. Có thể thấy mối quan hệ giữa Service Object với DAO là qua DAO interface



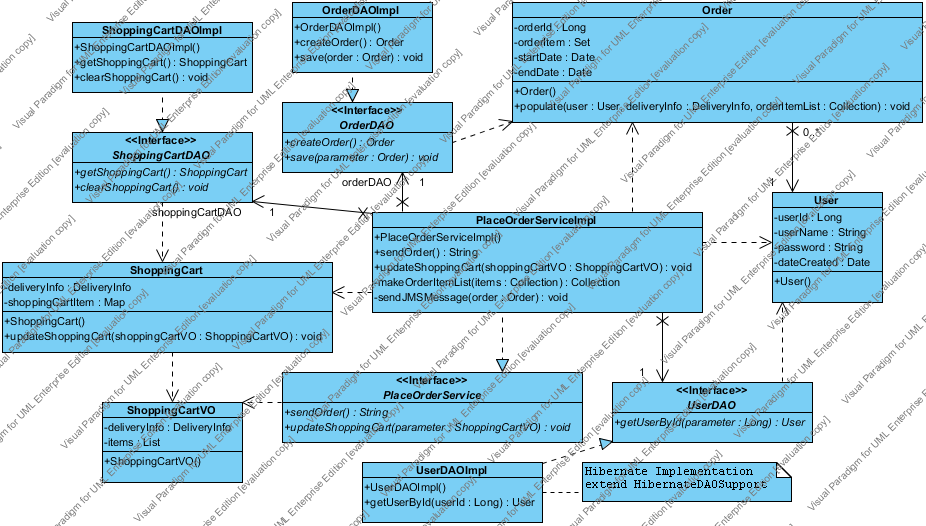
- Các trang web jsp chỉ làm nhiệm vụ layout, web được chia thành các component riêng rẽ, được thực thi dưới dạng các custom tag, thiết kế này tạo ra khả năng linh hoạt, dùng lại code, dễ dàng cho thay đổi

- Các custom tag tận dụng khả năng của struts 2 UI support. Cho khả năng quản lý theme cũng như tính linh hoạt khi không muốn sử dụng custom tag của jsp

- Struts 2 Action là sự cải tiến lớn: thực sự trở thành 1 plain old java object, không bị phụ thuộc vào framework đặc biệt là servlet api. (Sử dụng các interceptor cho các thao tác với servlet api) 🡪 tăng khả năng dùng lại code và cũng để có thể unit test. Có thể sử dụng kết hợp với spring framework để inject dependency

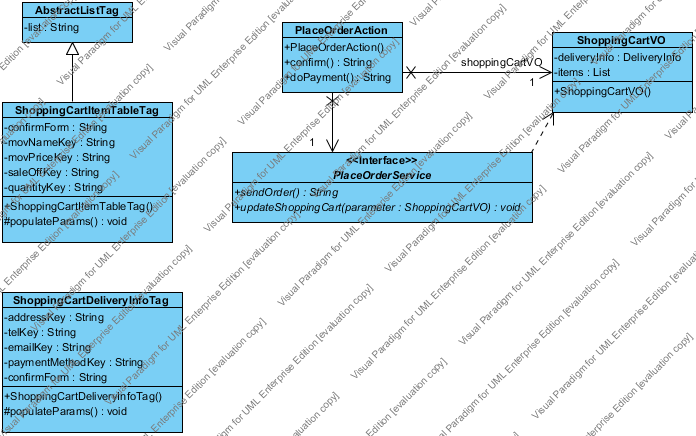
- Có thể thấy Action được spring (làm nhiệm vụ service locator) inject 1 đối tượng service, nhưng chỉ phụ thuộc ở dạng interface, không bị gắn với 1 lớp cụ thể, chính thiết kế này giúp Presentation tier có thể tách biệt 2 tầng riêng rẽ. hoạt động mà không có service thực sự 🡪 unit test

# GuiDonHang\_BL

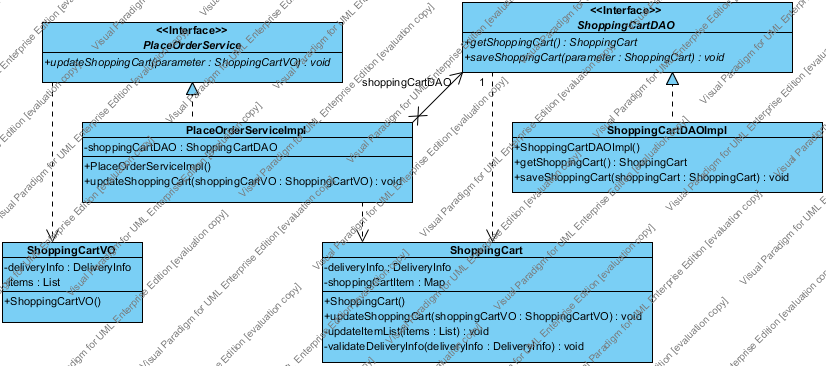


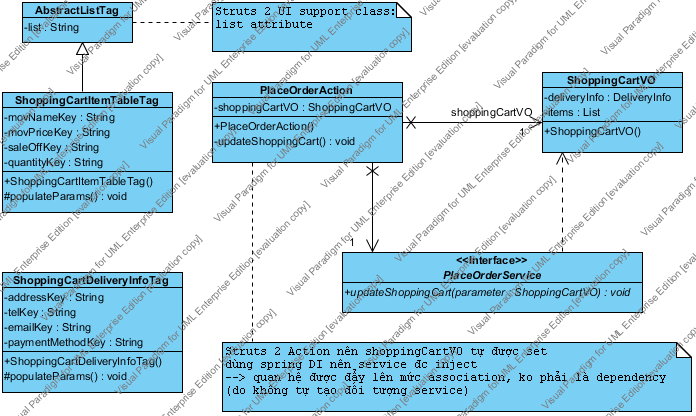
- Có thể thấy làm việc với ShoppingCart được lưu trong session không khác gì làm việc với các entity khác

# GuiDonHang\_P

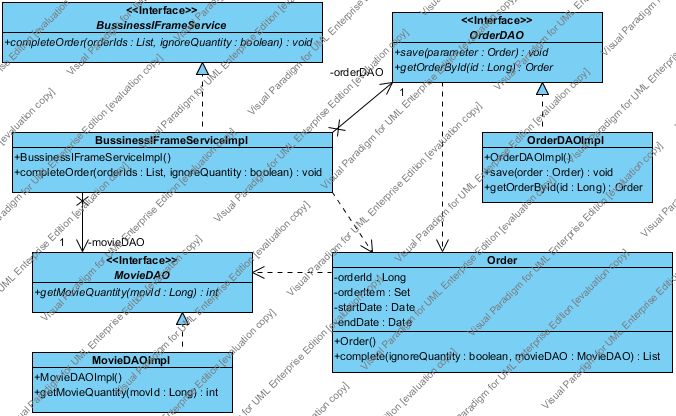


# CapNhatGioHang

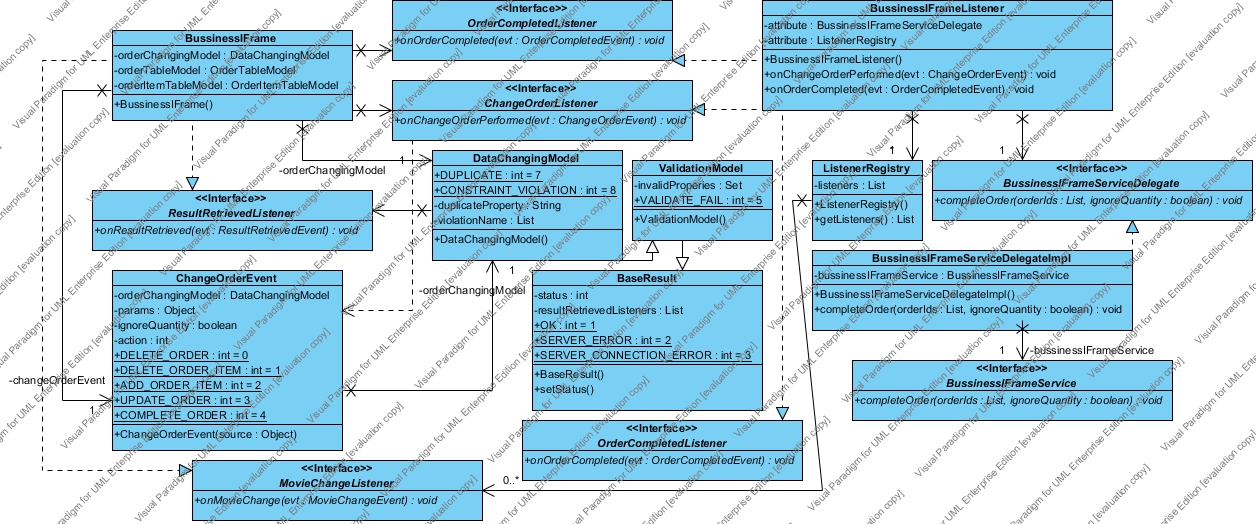




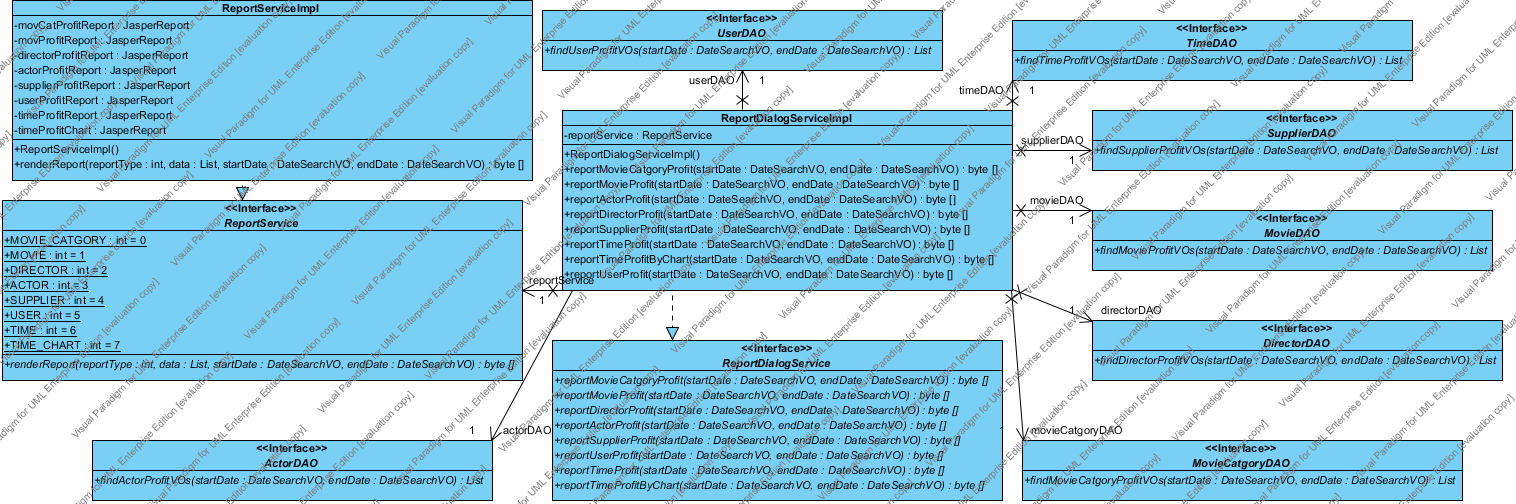
# LuuPhieuXuat\_BL



# LuuPhieuXuat\_P



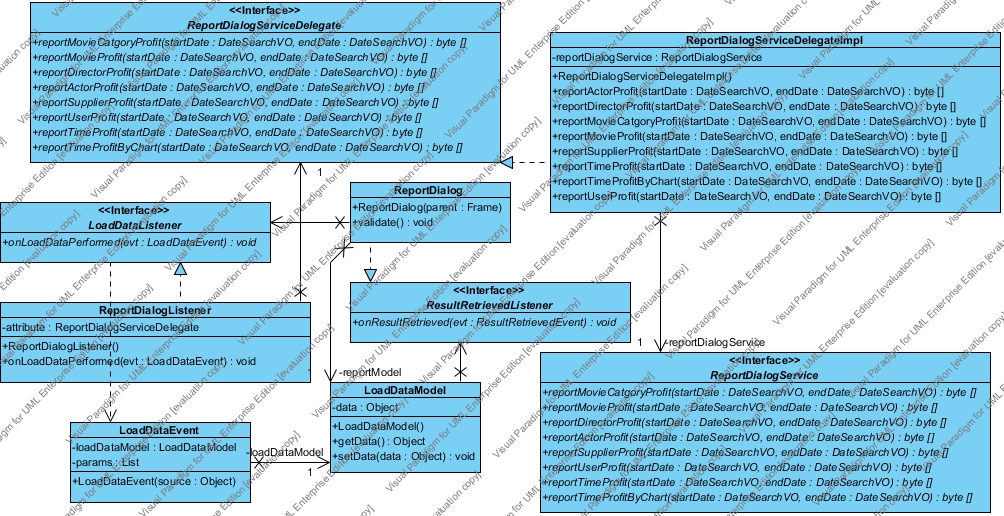
# XuatDuLieuBaoCaoDoanhThu\_BL



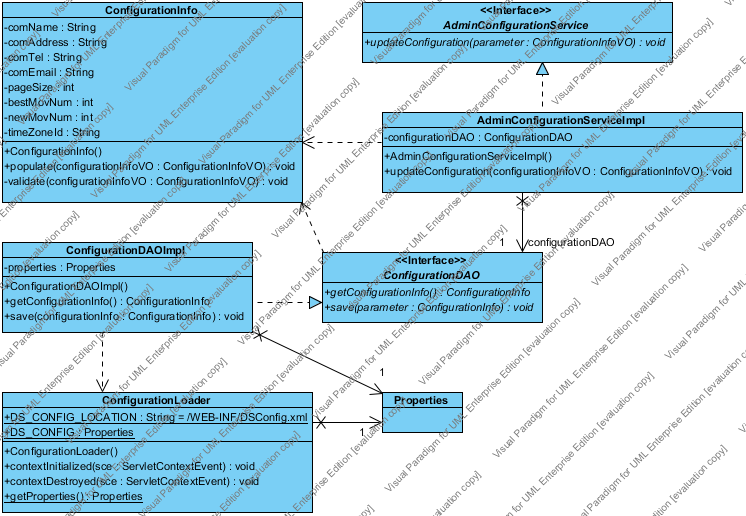
- Có thể thấy ưu điểm của thiết kế này: phần làm việc với dữ liệu , bussiness logic và phần gắn với Jasper Report được tách ra

- Trên thực tế ReportService còn phục vụ mục đích không thống kê (in phiếu nhập, phiếu xuất).

# XuatDuLieuBaoCaoDoanhThu\_P



# SuaCaiDatHeThong\_BL

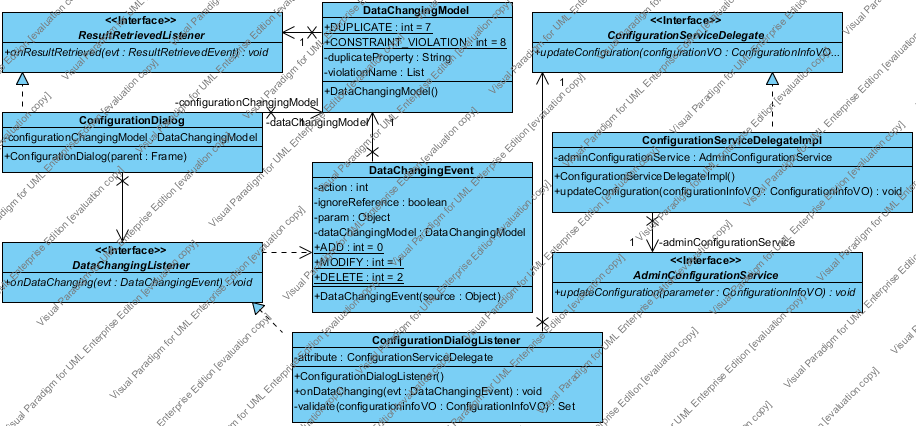


- Dễ thấy về cơ bản, cấu trúc trên không khác với các modul quản lý khác mặc dù configuration được lưu dưới dạng file xml, được load khi server start và được lưu lại dưới dạng file xml khi server stop. Điều này dễ dẫn đến việc gắn với Servlet api, tuy nhiên với thiết kế trên, các thao tác với servlet api được đóng gói trong ConfigurationLoader là 1 application listener. Nếu yêu cầu thay đổi, cần lưu configurationinfo trên csdl, hầu hết code được dùng lại

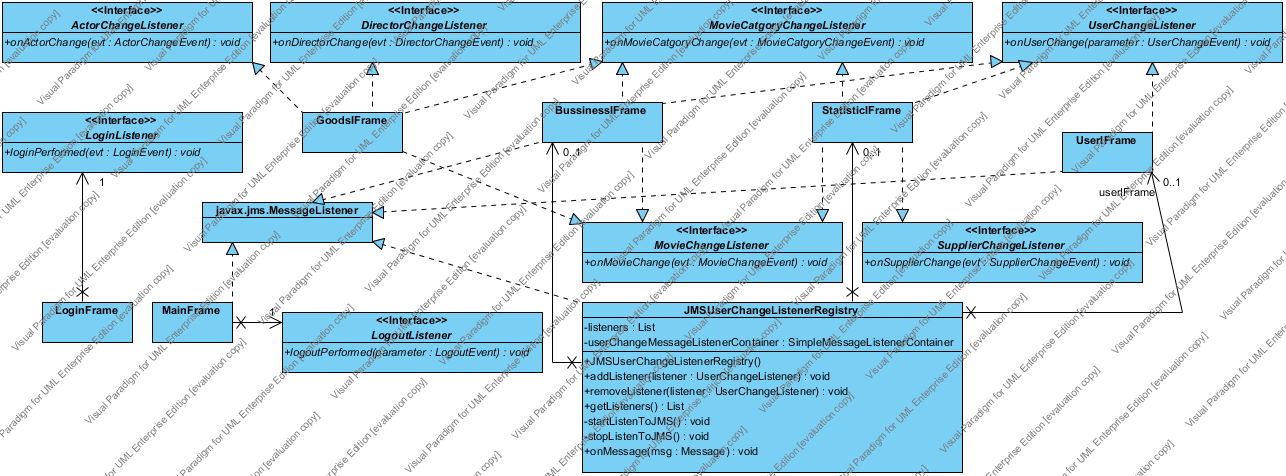
- ConfigurationInfo là 1 entity, bussiness logic được đóng gói trong này, khả năng dùng lại cao, xây dựng dễ

- Chú ý đây là enterprise application, phục vụ nhiều user, nhiều thread, configuration là dữ liệu chung cho nhiều thread vì thế cần threadsafe, vậy nên chọn lớp Properties, đây là lớp threadsafe, hơn nữa thông tin configuration là thông tin ít thay đổi mà thường chỉ đọc, vì thế việc lưu 1 đối tượng Properties dùng cho nhiều thread là có thể chấp nhận được về mặt tốc độ

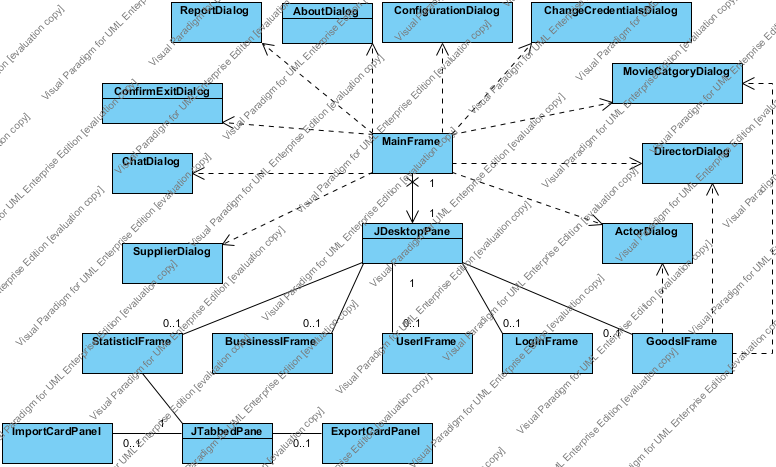
SuaCaiDatHeThong\_P



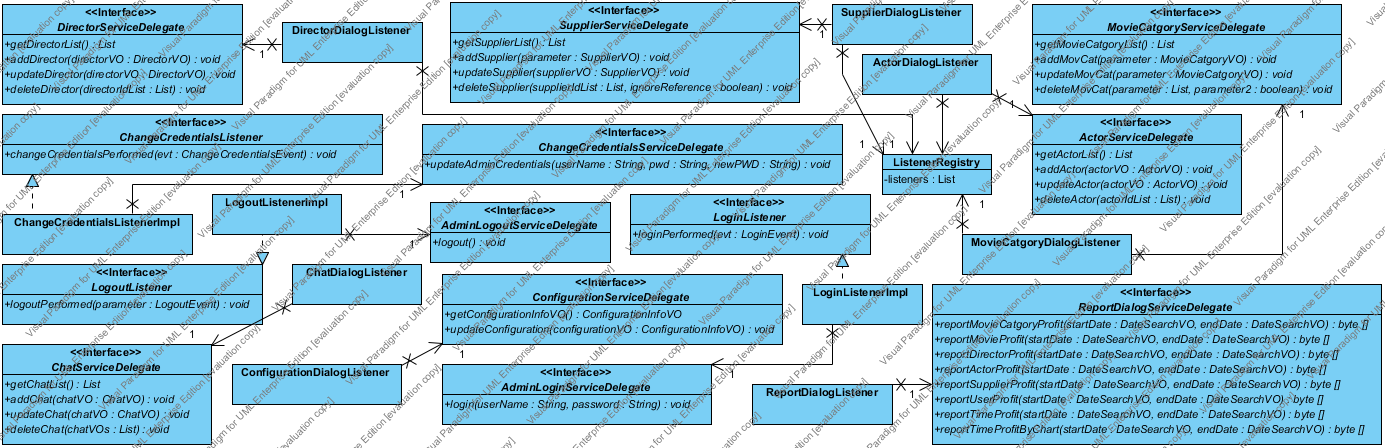
# FrameAndListener



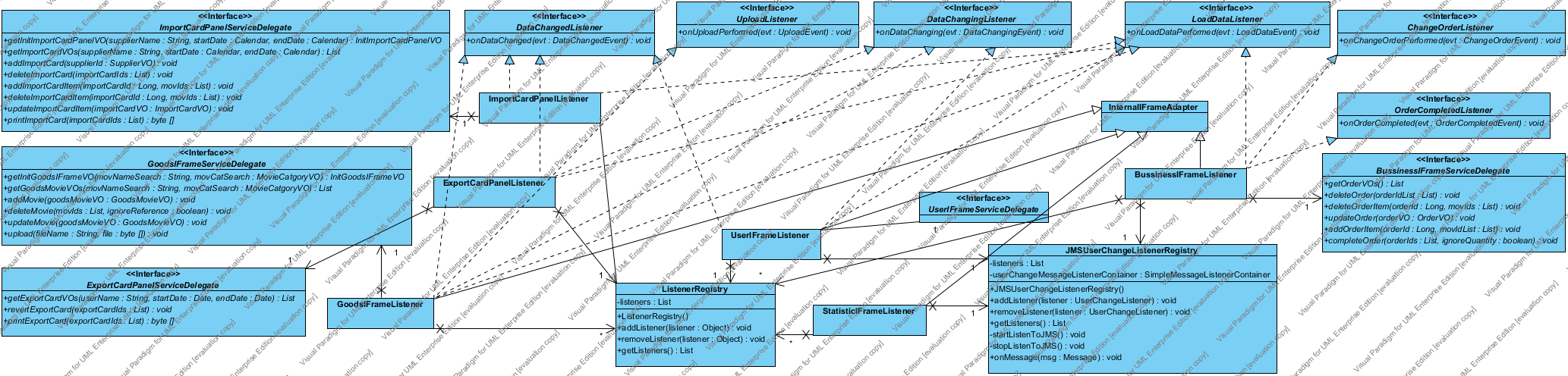
# Frame



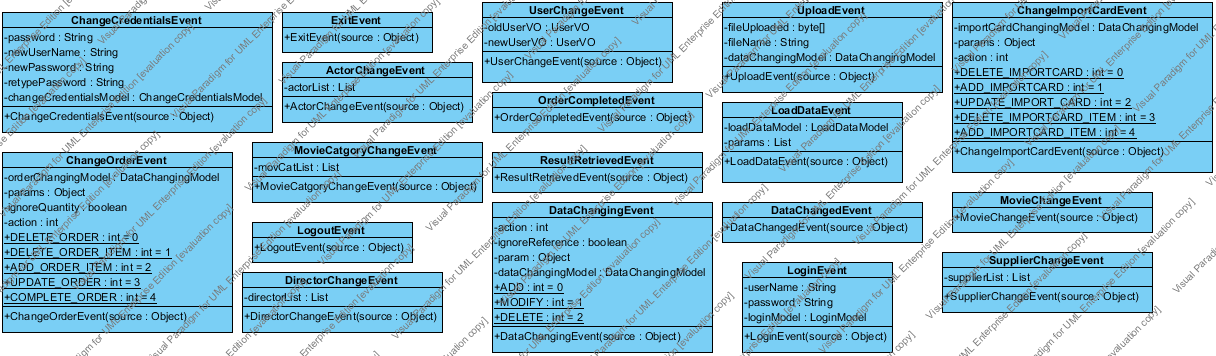
# DialogListener



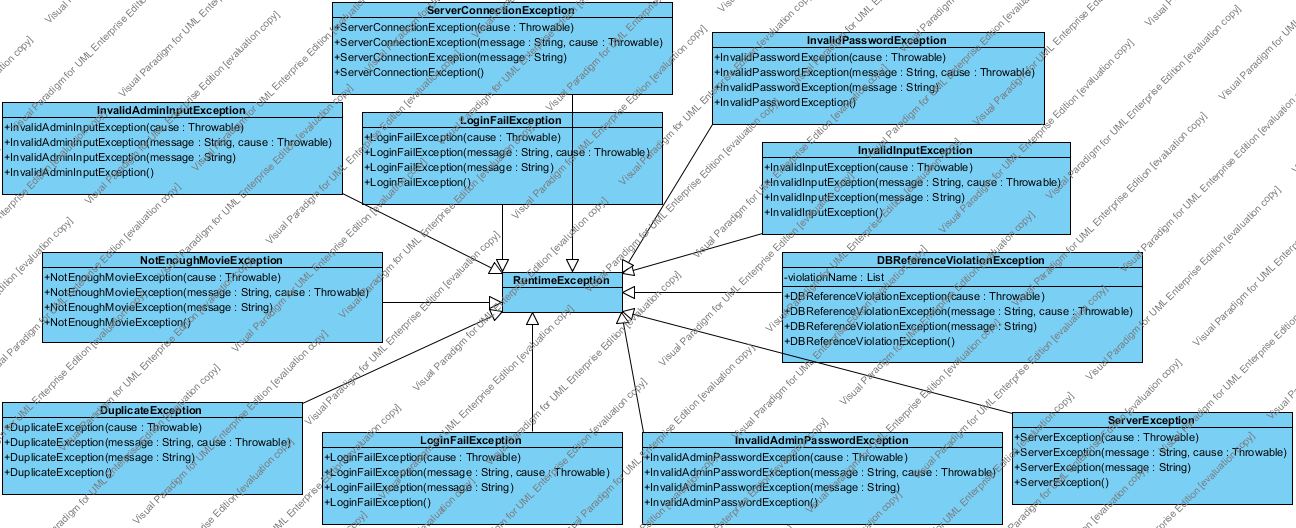
# FrameListener



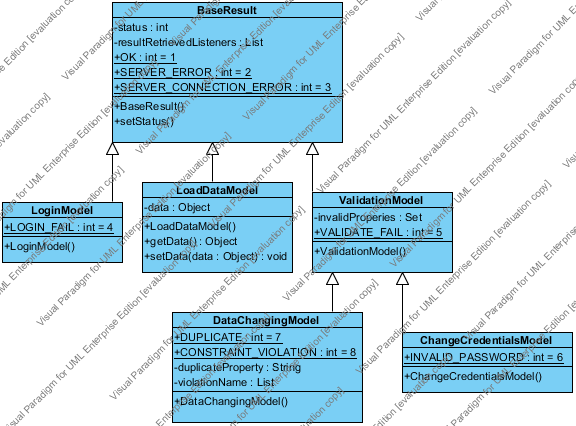
# Event



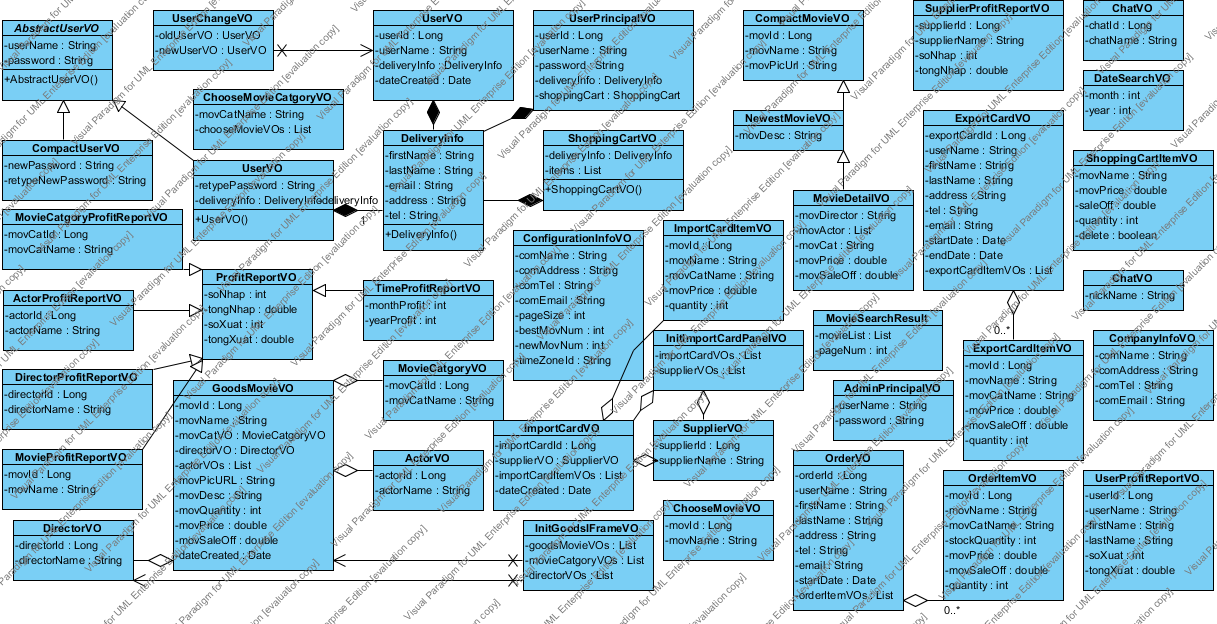
# Exception



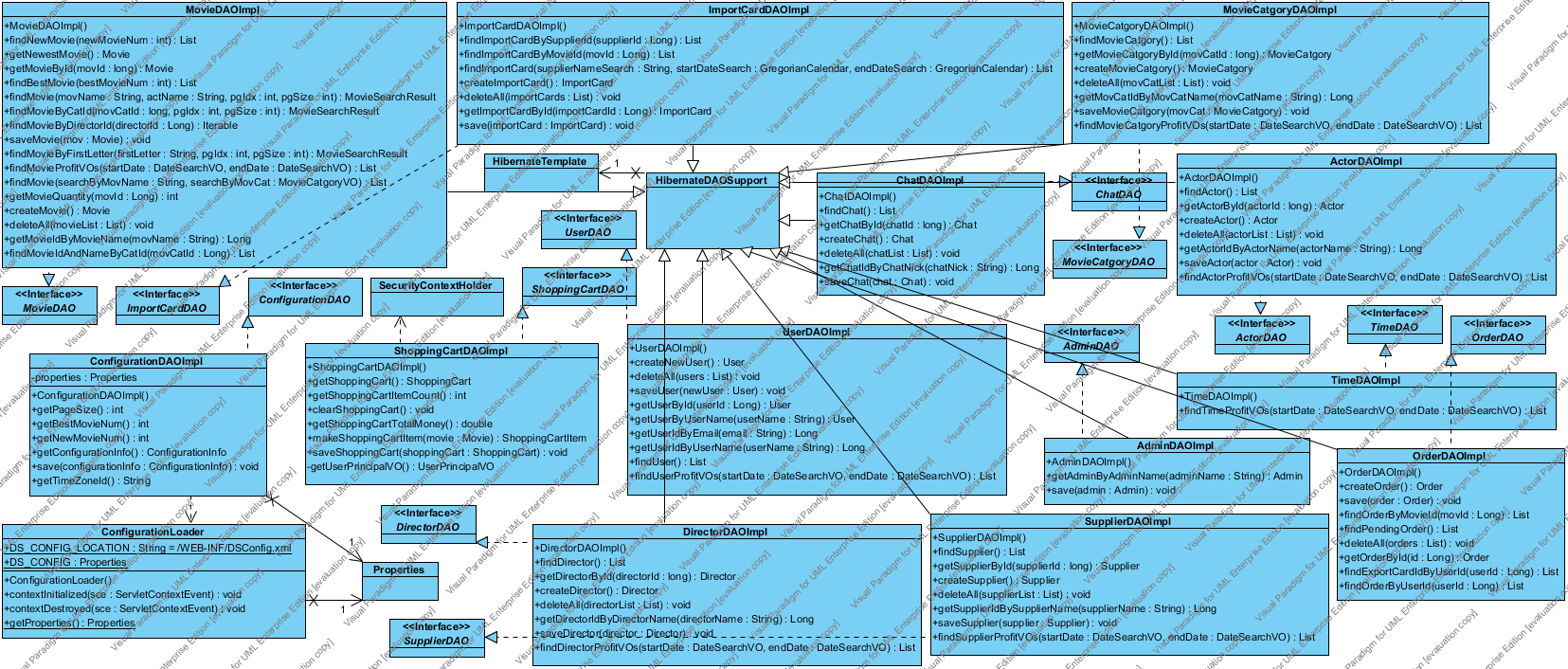
# Model



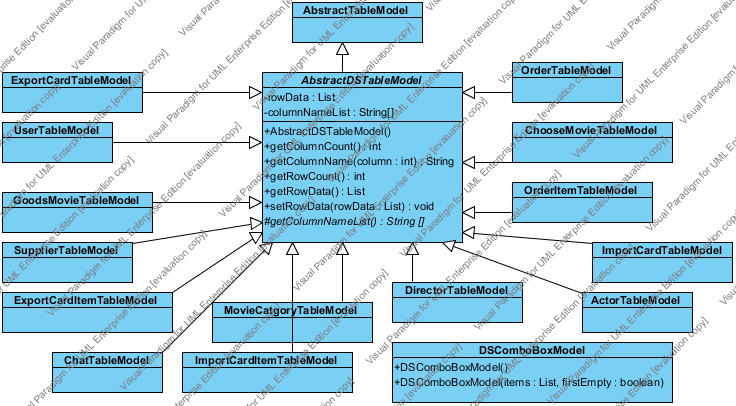
# ValueObject



# Dao

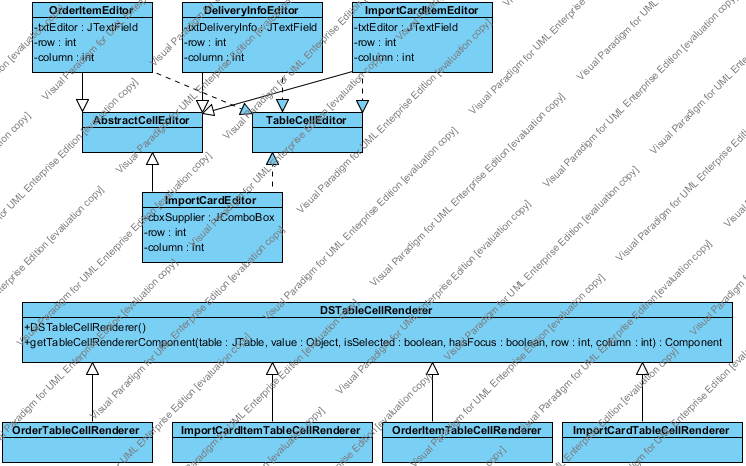


# TableModel



- AbstractDSTableModel là một lớp sử dụng generic programming, extend AbstractTableModel và implement hầu hết các lớp cần thiết, khi cần xây dựng table model cho một bảng cụ thể nào đó vd bảng Actor, chỉ việc extend AbstractDSTableModel<ActorVO> và gọi phương thức setData(List data)

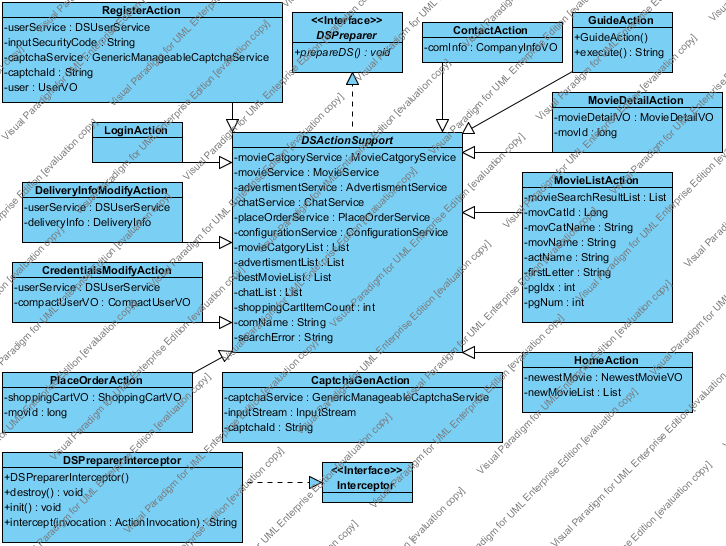
# TableCellRenderer\_Editor



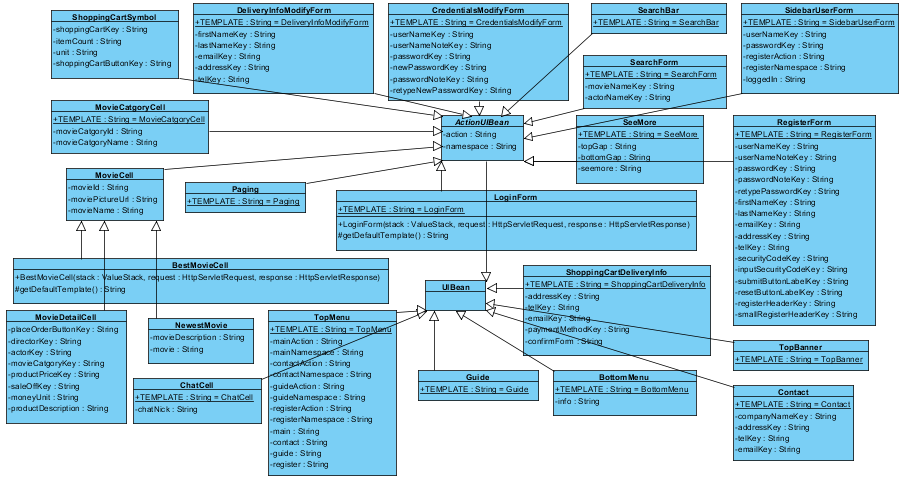
- Các lớp AbstractCellEditor interface TableCellEditor là thuộc Swing

- DSTableCellRenderer thực thi render table cell cơ bản (Jlabel, và đánh dấu dòng), các bảng tùy vào yêu cầu khác nhau extend lớp này và override phương thức getTableCellRendererComponent để thay đổi việc render (in các cột đậm tùy theo yêu cầu từng bảng).

# StrutsAction



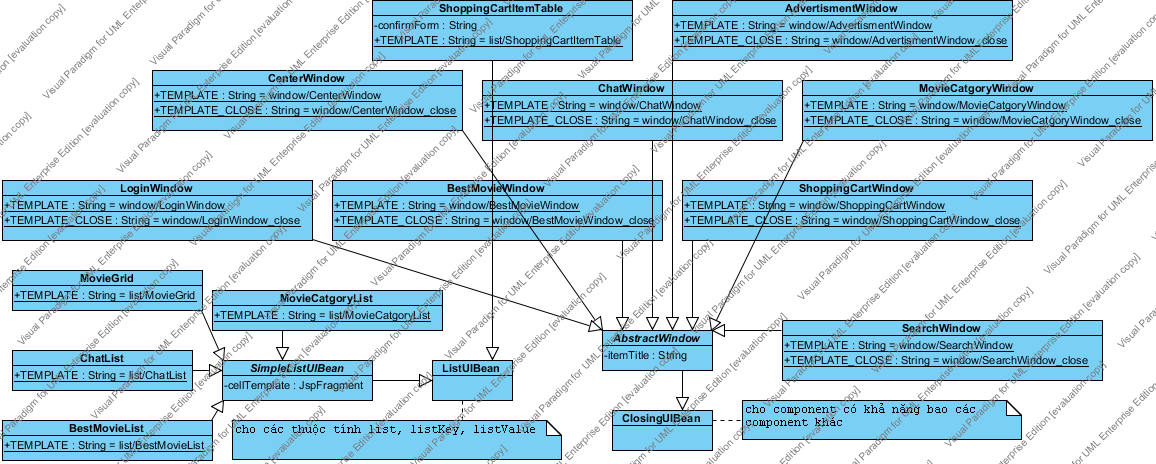
# Struts2Component



- Các lớp đều có thuộc tính TEMPLATE, chính là đường dẫn đến file \*.ftl 🡪 đây là các lớp làm việc với Freemarker api, để render ra html.

- Các lớp này extend ActionUIBean, lớp này cung cấp 2 thuộc tính dùng lại khá nhiều là action và namespace. ActionUIBean extend UIBean, lớp này cung cấp các thuộc tính dùng chung như css, id, theme, onclick, onFocus, disable, tabIndex... là các thuộc tính mà 1 thẻ html thường có. Đây là lớp của Struts 2

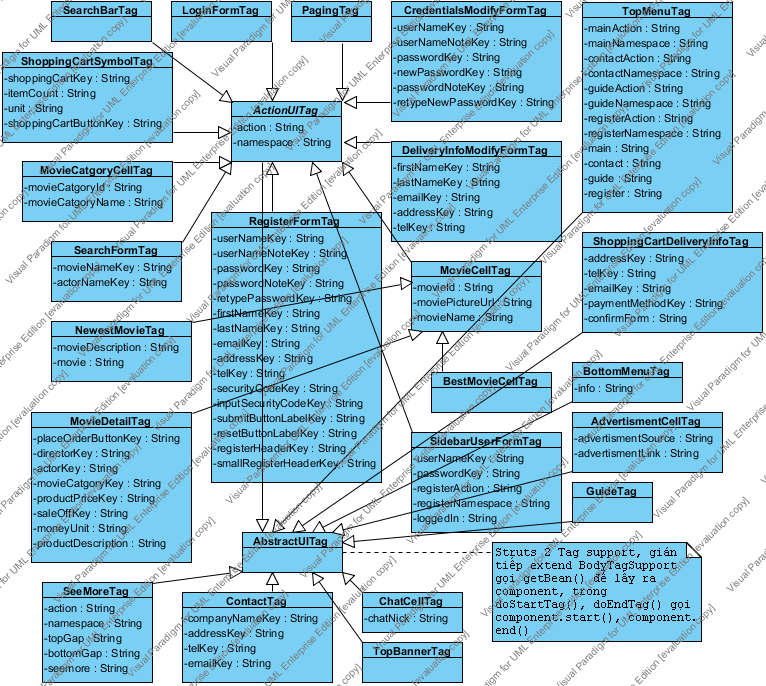
# Struts2Component\_List



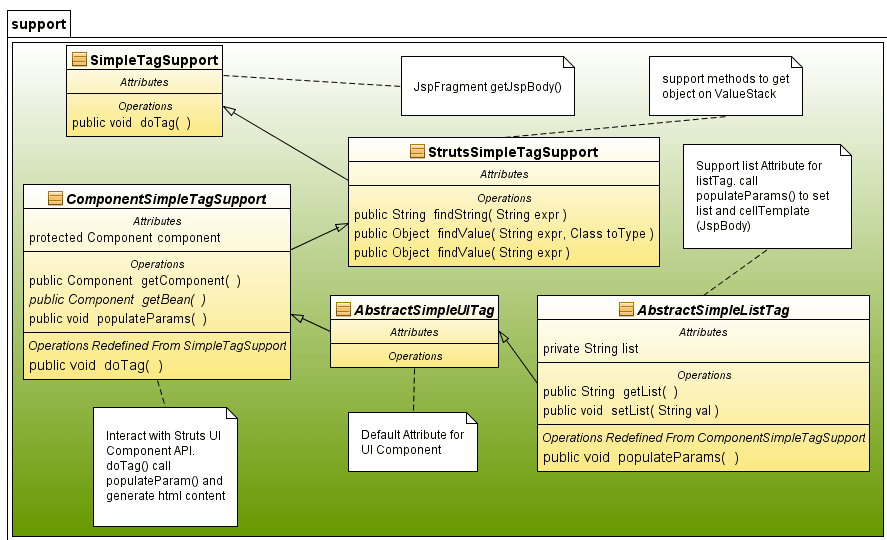
- Đây là các lớp sinh các component đặc biệt: window: chứa các tag khác, list: chứa và lặp các tag khác

- Chú ý lớp SimpleListUIBean: cho thuộc tính JspFragment là thuộc tính cho phép list component sử dụng để render html lặp lại kết quả render tag con (template)

# Struts2Tag



- Các custom tag không trực tiếp làm việc với Freemarker api mà qua struts 2 component api. Các custom tag có nhiệm vụ chuyển lại giá trị các attribute của customtag cho các component tương ứng với custom tag đó



# Struts2Tag\_List

