



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Nhập môn Công nghệ phần mềm
Introduction to Software Engineering
(IT3180)

Lab Guide 2

MVC Architecture in Web
Application Development

Thông tin GV

Nội dung

- Giới thiệu về kiến trúc MVC
- Áp dụng kiến trúc MVC trong phát triển ứng dụng web

Separation of Concerns in Web App

- Sun Microsystem đưa ra JSP nhằm mục đích phân tách giữa tầng presentation và tầng business logic trong Servlets
- Vì thực chất mỗi trang JSP sẽ được build thành 1 servlet nhưng nhằm mục đích hiển thị hơn là điều khiển như Servlet thông thường
- Các kiến trúc này thường được đề cập đến như là các JSP Model

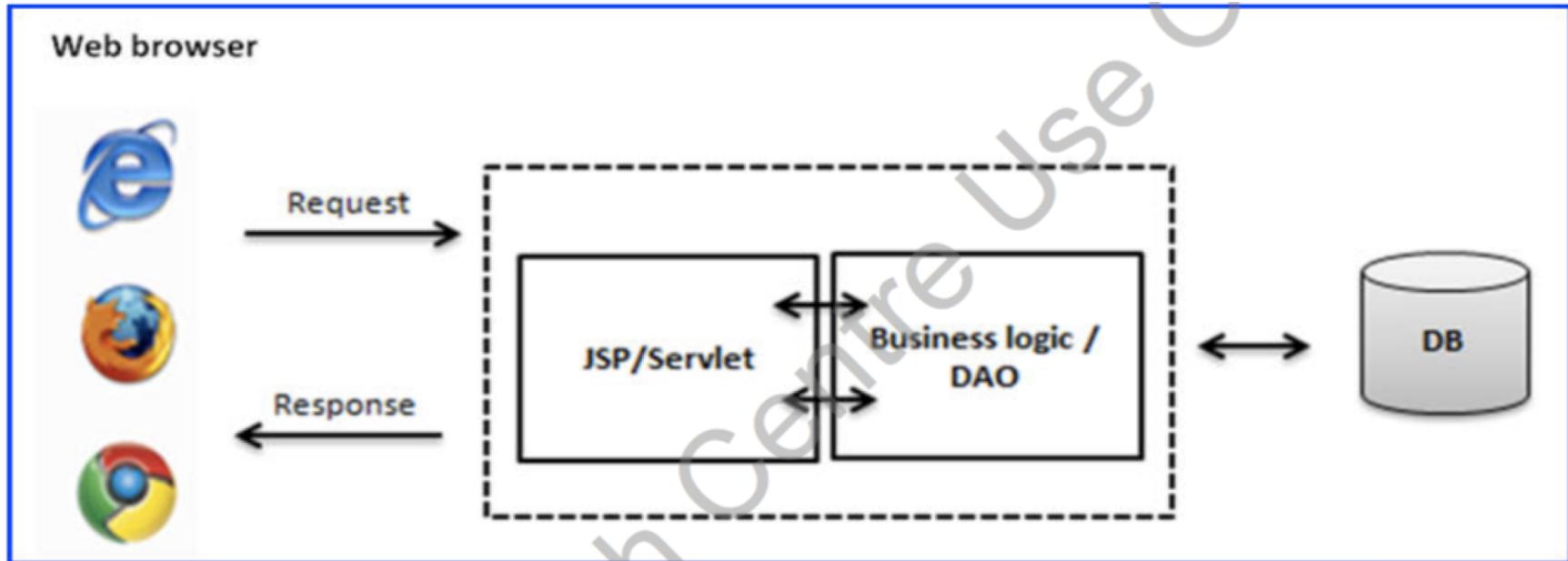
JSP Models

- Có hai mẫu thiết kế thường được sử dụng trong phát triển ứng dụng web
- JSP Model 1:
 - HTML hoặc JSP gửi các request cùng dữ liệu tới Web Container
 - Web Container sẽ gọi tới Servlet tương ứng chịu trách nhiệm xử lý request, xác minh dữ liệu, xử lý logic nghiệp vụ và tạo các response trả về cho trình duyệt

JSP Models

- Có hai mẫu thiết kế thường được sử dụng trong phát triển ứng dụng web
- JSP Model 2 (Model – View – Controller)
 - Có một Servlet trung tâm đóng vai trò làm Controller, nhận tất cả các request từ JSP
 - Controller làm việc với Model để chuẩn bị các dữ liệu cần thiết cho View và trả lại dữ liệu về cho các JSP
 - Business Logic và Presentation Logic được tách biệt rõ ràng giúp giảm tính coupling tăng tính tái sử dụng

JSP Model 1

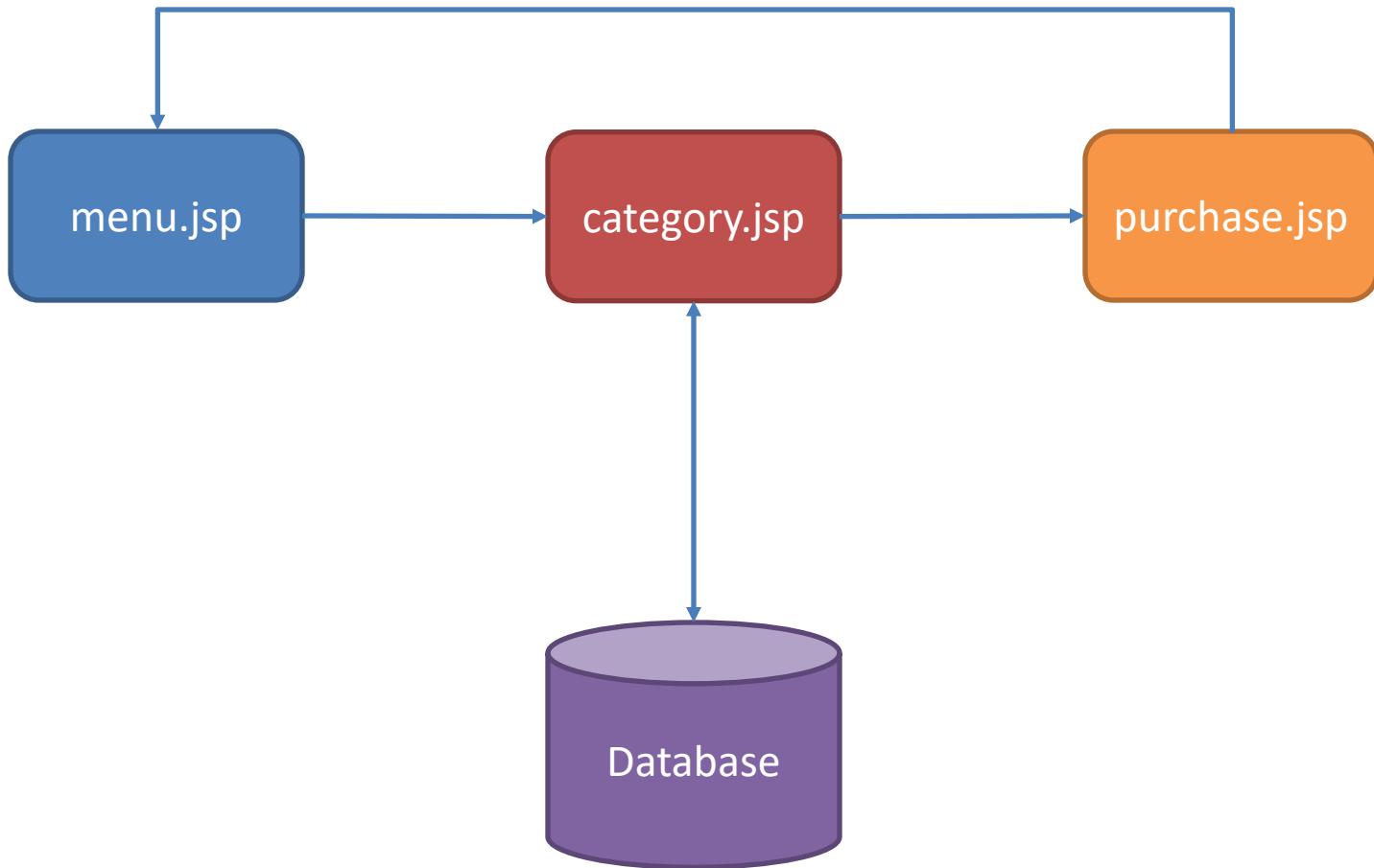


- Không có các Servlet ngoài các JSP
- Client request được gửi trực tiếp tới trang JSP, trang JSP tương tác với bean model và làm nhiệm vụ điều hướng người dùng tới một trang khác để view

Đặc điểm của JSP Model 1

- Page-centric architecture
- Ứng dụng chứa một loạt các trang JSP có liên quan đến nhau và các trang JSP đó làm nhiệm vụ điều khiển tất cả các vấn đề trong ứng dụng
- Việc xử lý logic nghiệp vụ và điều khiển được "mixed" trong code JSP dưới dạng các JavaBean Forms, scriptlet, expressions

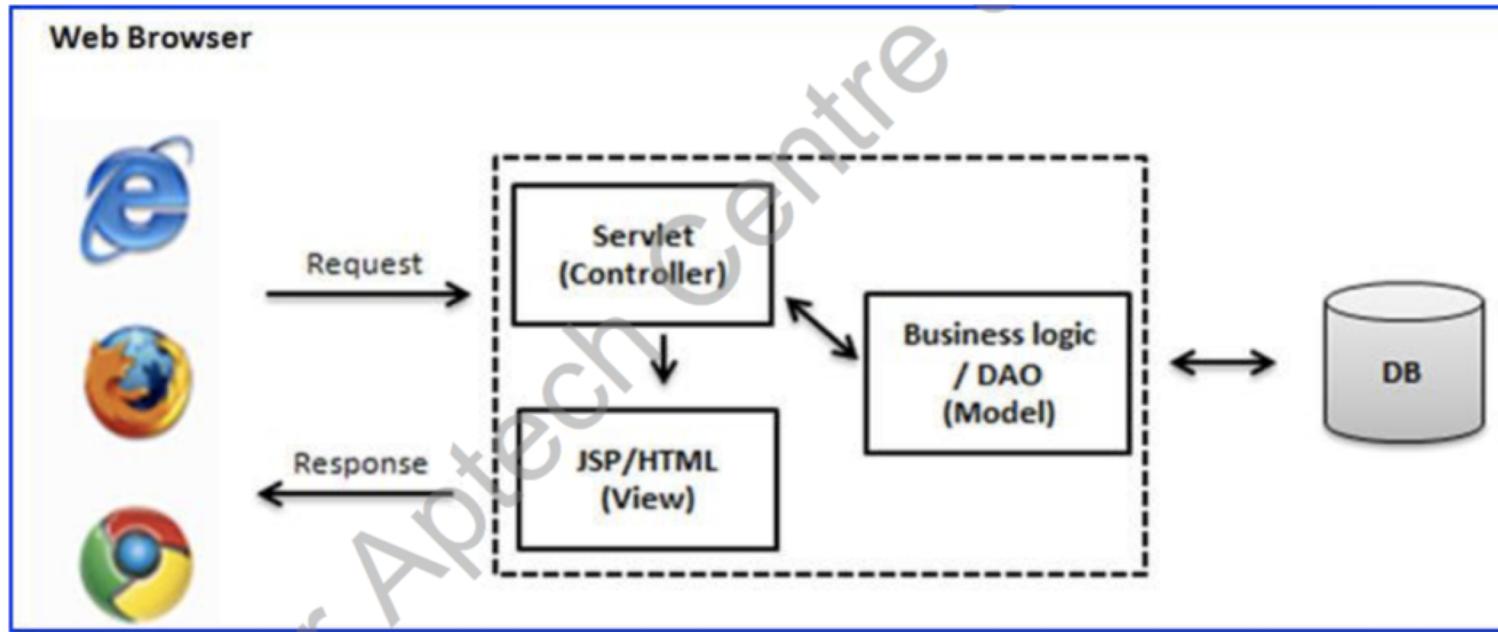
Đặc điểm của JSP Model 1



Đặc điểm của JSP Model 1

- **Ưu điểm:** giúp việc phát triển ứng dụng trở nên đơn giản hơn vì không có Servlet nào trong ứng dụng
- **Nhược điểm:**
 - Business logic và presentation logic trộn lẫn vào nhau trong trang JSP sẽ khiến cho ứng dụng quy mô lớn và phức tạp khó bảo trì
 - Mỗi một trang JSP vừa chịu trách nhiệm xử lý điều khiển, logic nghiệp vụ và hiển thị kết quả cho người dùng ==> JSP model 1 khó mở rộng

JSP Model 2 - Model/View/Controller



- Thêm một thành phần gọi là Controller
- Tách biệt giữa Business Logic và Presentation Logic
- Request được gửi tới Controller, Controller làm nhiệm vụ tương tác với các Model và xử lý business logic, sau đó tạo các response kèm dữ liệu trả về cho các JSP để hiển thị

Đặc điểm của JSP Model 2

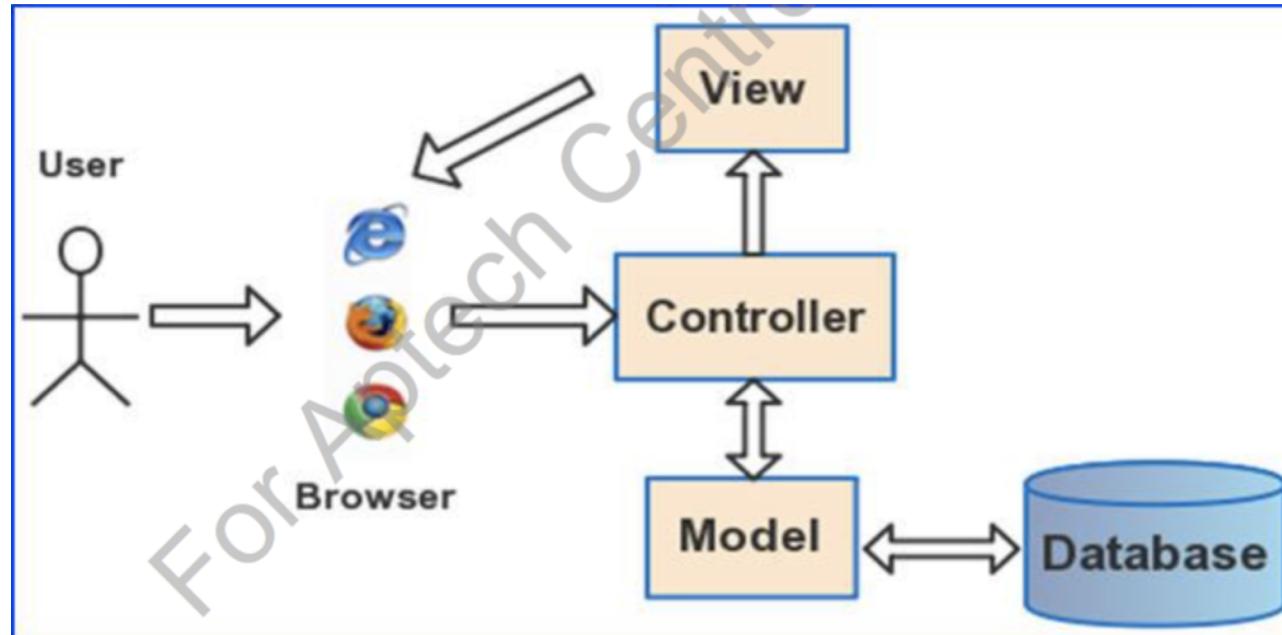
- Servlet-centric architecture
- Servlet – Controller:
 - Chịu trách nhiệm xử lý các client request và khởi tạo các Model (Java objects) để thực hiện các thao tác nghiệp vụ
 - Quyết định trang JSP nào request sẽ được chuyển tiếp
- JSP – View:
 - Chịu trách nhiệm điều khiển thành phần View
 - Nhận các đối tượng được tạo ra bởi Servlet
 - Trích xuất các nội dung động cần thay đổi trong các form view

Đặc điểm của JSP Model 2

- Ưu điểm:
 - Dễ dàng bảo trì và mở rộng
 - Dễ kiểm thử các chức năng vì tách biệt các servlet cho các chức năng khác nhau

Model – View – Controller (MVC)

- MVC là một khuôn mẫu kiến trúc phần mềm
- Mẫu thiết kế này cho phép phân tách giữa logic ứng dụng và giao diện người dùng



Relationship between Components

- View và Controller
 - Controller chịu trách nhiệm tạo và lựa chọn các view
- Model và View
 - View phụ thuộc vào Model
 - Nếu một thay đổi thực hiện trên Model, cần thiết phải thực hiện các thay đổi trên các trang View tương ứng hiển thị thông tin liên quan đến Model cho người dùng
- Model và Controller
 - Controller phụ thuộc vào Model
 - Nếu một thay đổi thực hiện trên các API liên quan đến Model, cần thiết phải thực hiện các thay đổi tương ứng trên các Controller

MVC trong phát triển ứng dụng Web

- Model đóng gói các dữ liệu và cả các logic nghiệp vụ có liên quan đến dữ liệu, sử dụng Java Bean hoặc các POJO (Plain Old Java Objects), database API hoặc các file XML
- View thể hiện trạng thái hiện tại của Model sử dụng HTML hoặc JSP
- Controller làm nhiệm vụ cập nhật trạng thái của Model và tạo ra 1 hoặc nhiều view sử dụng Servlet

Try Lab 02

- Thể hiện kiến trúc MVC trong phát triển một trang Login đơn giản
- Yêu cầu sinh viên liệt kê và vẽ sơ lược mô hình kiến trúc của Login Modul sau khi thực hành xong lab