Biên soạn: TS. TRẦN CÔNG ÁN (Chủ biên)
ThS. NGUYỄN HỮU VÂN LONG

GIÁO TRÌNH

CÔNG NGHỆ J2EE

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ
2019

BIÊN MỤC TRƯỚC XUẤT BẢN THỰC HIỆN BỞI
TRUNG TÂM HỌC LIỆU TRƯỜNG ĐẠI HỌC CẦN THƠ

Trần, Công Án
Giáo trình công nghệ J2EE / Trần Công Án, Nguyễn Hữu Vân Long .– Cần Thơ : Nxb. Đại học Cần Thơ, 2019.
194 tr. : minh họa ; 24 cm
Sách có danh mục tài liệu tham khảo
ISBN: 9786049652219
1. Java (Computer program language) 2. Ngôn ngữ lập trình máy tính
I. Nhan đề. II. Nguyễn, Hữu Vân Long
005.133– DDC 23 MFN 230674
A105

LỜI GIỚI THIỆU

Nhằm góp phần làm phong phú nguồn tư liệu phục vụ nghiên cứu, học tập cho bạn đọc và sinh viên Khoa Công nghệ Thông tin và Truyền thông - Trường Đại học Cần Thơ, Nhà xuất bản Đại học Cần Thơ ấn hành và giới thiệu cùng bạn đọc giáo trình "Công nghệ J2EE" do Tiến sĩ Trần Công Án và Thạc sĩ Nguyễn Hữu Vân Long biên soạn.

Nội dung giáo trình bao gồm 05 chương, cung cấp các kiến thức nền tảng của công nghệ J2EE bao gồm tổng quan về J2EE, quan hệ giữa J2SE và J2EE, Java Server Pages, JDBC, Servlets và Enterprise Java Bean. Các kiến thức này bao phủ giúp người đọc có thể xây dựng những ứng dụng J2EE cơ bản, đồng thời là một cơ sở vững chắc để tìm hiểu thêm về các chi tiết của công nghệ J2EE cũng như các công nghệ có liên quan. Thêm vào đó, cuối mỗi chương còn có nhiều bài tập hữu ích cho bạn đọc bao gồm các bài tập củng cố lý thuyết và các bài tập thực hành. Giáo trình này là tài liệu học tập có giá trị cho sinh viên các chuyên ngành có liên quan đến ngành Công nghệ thông tin.

Nhà xuất bản Đại học Cần Thơ chân thành cám ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình "Công nghệ J2EE" được ra mắt bạn đọc.

Nhà xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ

LỜI NÓI ĐẦU

J2EE (Java 2 Platform Enterprise Edition) là một nền tảng để phát triển các ứng dụng phân tán (distributed application), theo kiến trúc đa tầng. Công nghệ J2EE phù hợp với việc phát triển các ứng dụng lớn, đòi hỏi tính dãn nở cao (scalability) của các doanh nghiệp. Mặc dù không phải là một "tiêu chuẩn" được các tổ chức chuẩn quốc tế như ISO, ECMA hay SEI,… các kiến trúc, mô hình và các công nghệ mà J2EE đề xuất được coi là một "tiêu chuẩn thực tế" (de-factor standard) cho các ứng dụng phân tán. Các ứng dụng muốn đạt được chứng nhận "tương thích với Java EE" (Java EE compliant) phải tuân thủ các yêu cầu trong đặc tả của J2EE và phải vượt qua được bộ kiểm tra của J2EE (Java EE Technology Compatibility Kit). Điều này nói lên được sự phổ biến và tầm quan trọng của nền tảng công nghệ này đối với các ứng dụng doanh nghiệp.

Trong chương trình đào tạo kỹ sư công nghệ thông tin, môn học *Công nghệ J2EE* mới được đưa vào chương trình đào tạo trong những năm gần đây nhằm nắm bắt xu hướng tất yếu của công nghệ này trong các ứng dụng doanh nghiệp. Các kiến thức này không chỉ giúp cho sinh viên biết cách thức xây dựng các ứng dụng lớn mà còn giúp cho sinh viên tiếp cận với hệ thống thông tin của doanh nghiệp thuận lợi và dễ dàng hơn.

Do đây là một công nghệ rất quan trọng cho các ứng dụng phân tán nên nguồn tài liệu Tiếng Anh rất phong phú. Tuy nhiên, đối với hầu hết các trường đại học tại Việt Nam thì việc đưa nội dung của công nghệ này chương trình giảng dạy mới được thực hiện trong thời gian gần đây nên nguồn tài liệu Tiếng Việt còn rất hạn chế. Vì vậy, chúng tôi đã thực hiện biên soạn quyển giáo trình *Công nghệ J2EE* này nhằm làm phong phú thêm nguồn tài liệu Tiếng Việt cho môn học này. Giáo trình này đặc biệt hướng đến độc giả là sinh viên chuyên ngành công nghệ thông tin. Nội dung của giáo trình sẽ cung cấp cho người đọc các kiến thức nền tảng, cốt lõi của J2EE chứ không bao phủ hết tất cả các chi tiết của công nghệ này. Hoàn thành các kiến thức trong giáo trình này, người đọc sẽ có được một sự khởi đầu vững chắc đối với công nghệ J2EE, qua đó việc tìm hiểu sâu hơn về công nghệ này sẽ có nhiều thuận lợi và dễ dàng.

Nội dung của giáo trình được chia thành 5 chương như sau, được trình bày theo trình tự mà người đọc nên tiếp cận:

- ✓ *Chương 1 – Giới thiệu J2EE*: Giới thiệu về J2EE và mối quan hệ giữa J2EE với J2SE, bao gồm các nội dung chính: giới thiệu về J2EE và J2SE, các khái niệm cơ bản trong J2EE, các công nghệ nổi bật trong J2EE, kiến trúc đa tầng trong J2EE và mô hình phát triển ứng dụng

MVC. Ngoài ra, cuối chương sẽ giới thiệu các công cụ dùng để phát triển các ứng dụng J2EE như Tomcat Web Server và Eclipse.

- ✓ *Chương 2 – Java Server Page*: Giới thiệu về Java Server Page, gồm các nội dung chính: vòng đời của một trang JSP, các thư viện thẻ JSP, cách thức xử lý ngoại lệ trong JSP, kỹ thuật chuyển tiếp trang và sử dụng lại trang JSP và ngôn ngữ biểu diễn biểu thức EL (Expression Language) trong JSP.
- ✓ *Chương 3 – Nối kết cơ sở dữ liệu với JDBC*: Giới thiệu về JDBC API, một giải pháp kết nối ứng dụng với các loại cơ sở dữ liệu khác nhau, các loại JDBC Driver và kịch bản thực hiện nối kết cơ sở dữ liệu, phương pháp thực thi câu lệnh SQL, các thủ tục và thao tác trên kết quả của các câu truy vấn.
- ✓ *Chương 4 – Servlets*: Giới thiệu về servlet, bao gồm các nội dung chính: kiến trúc và vòng đời của 1 servlet, quy trình hoạt động cơ bản của 1 servlet, xử lý ngoại lệ trên servlet, quản lý phiên làm việc thông qua servlet, vai trò của bộ lọc và cách sử dụng bộ lọc, và mô hình MVC với servlet và JSP.
- ✓ *Chương 5 – Enterprise Java Bean*: Làm rõ vai trò của EJB cũng như cách thức triển khai EJB trong ứng dụng, giới thiệu về EJB 3.0, các loại bean được cung cấp bởi EJB, quản lý và sử dụng EJB session bean, quản lý và sử dụng Message Driven bean, liên kết mô hình hướng đối tượng và cơ sở dữ liệu quan hệ với JPA, xây dựng quan hệ giữa các Entity class trong JPA, và sử dụng Embeddable Object.

Nội dung của giáo trình này được biên soạn dựa trên giả thuyết là độc giả đã có kiến thức cơ bản về lập trình Java (J2SE). Do đó, các kiến thức cơ bản về ngôn ngữ lập trình Java sẽ không được giới thiệu trong giáo trình này.

Chúng tôi gửi lời cảm ơn sâu sắc đến quý Thầy Cô khoa Công nghệ thông tin và Truyền thông đã nhiệt tình góp ý cho nội dung của giáo trình này.

Do đây là phiên bản đầu tiên của giáo trình nên sẽ không tránh khỏi những thiếu sót. Chúng tôi mong nhận được sự góp ý chân thành từ quý độc giả để quyển giáo trình ngày càng hoàn thiện hơn.

Cần Thơ, tháng 06 năm 2019

NHÓM TÁC GIẢ

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

DANH MỤC THUẬT NGỮ TIẾNG ANH

Attribute	Thuộc tính
CGI	Common Gateway Interface
DBMS	Database Management System Hệ quản trị cơ sở dữ liệu
DD	Deployment Descriptor Bộ/Tập tin mô tả triển khai
Design Pattern	Mẫu thiết kế
EL	Expression Language Ngôn ngữ biểu diễn biểu thức
Enterprise Application	Ứng dụng doanh nghiệp
Exception	Ngoại lệ
Filter	Bộ lọc
HTTP Request	Thông điệp yêu cầu HTTP
HTTP Response	Thông điệp trả lời HTTP
Implicit Object	Đối tượng ẩn
J2EE	Java 2 Platform Enterprise Edition
J2EE Component	Thành phần của J2EE
J2EE Container	Bộ chứa của J2EE
JDBC	Java Database Connectivity
JSTL	Java Server page Standard Tag Library
MVC	Model – View – Controller
Parameter	Tham số
POJO	Plain Old Java Object Đối tượng Java thuần túy
Procedure	Thủ tục
Process	Tiến trình
Reuseable	Tính sử dụng lại
Savepoint	Điểm đánh dấu
Scalability	Tính giãn nở
Scope	Phạm vi
Session	Phiên giao dịch
Synchronize	Đồng bộ hoá
Thread	Luồng
Topology	Hình thái
Transaction	Giao dịch

CHƯƠNG 1

GIỚI THIỆU J2SE VÀ J2EE

Chương này giới thiệu về J2EE và mối quan hệ giữa J2EE với J2SE, bao gồm các nội dung chính: giới thiệu về J2EE và J2SE, các khái niệm cơ bản trong J2EE, các công nghệ nổi bật trong J2EE, kiến trúc đa tầng trong J2EE và mô hình phát triển ứng dụng MVC. Ngoài ra, cuối chương sẽ giới thiệu các công cụ dùng để phát triển các ứng dụng J2EE như Tomcat Web Server và Eclipse.

1.1 J2SE VÀ J2EE

Chúng ta đã được học về cách để lập trình, phát triển và triển khai 1 ứng dụng cơ bản trên nền tảng J2SE - Java 2 Standard Edition. Mặc dù ứng dụng được phát triển có thể đảm bảo về chức năng, hoàn toàn không có lỗi, bugs, và code được viết theo đúng quy phạm nhưng câu hỏi đặt ra là liệu ứng dụng đã được xem là sẵn sàng để triển khai trên các doanh nghiệp hay chưa? Thuật ngữ "Ứng dụng doanh nghiệp" (Enterprise Application) nhằm chỉ ra các ứng dụng được phát triển theo một quy trình chặt chẽ trên một nền tảng hỗ trợ đầy đủ các tính năng "Doanh nghiệp" khi triển khai.

So với một ứng dụng thông thường, một ứng dụng doanh nghiệp được tạo ra để giải quyết các vấn đề phức tạp và luôn có thay đổi phát sinh phù hợp với nhu cầu của doanh nghiệp. Dữ liệu trong các ứng dụng doanh nghiệp cần được lưu trữ an toàn, bền vững, chẳng hạn: hoá đơn của khách hàng, thông tin đặt vé chuyến bay… Ứng dụng doanh nghiệp cung cấp nhiều giao diện làm việc tương ứng các thao tác khác nhau như giao diện dành cho khách hàng đặt hàng, giao diện thực hiện nghiệp vụ của nhân viên… Ngoài ra, ứng dụng doanh nghiệp hỗ trợ việc truyền tải dữ liệu trên các hệ thống kết nối từ xa, kết hợp xử lý trên đa dữ liệu từ đa nguồn...Cuối cùng nhưng không kém quan trọng là ứng dụng cần đảm bảo tất cả các thao tác diễn ra đúng đắn, chính xác, hiệu suất làm việc tốt nhất. Những đặc điểm vừa nêu trên có thể được tóm gọn lại bằng một câu, đó là: *Tính mạnh mẽ khi đối mặt với sự phức tạp* cần phải có của ứng dụng doanh nghiệp.

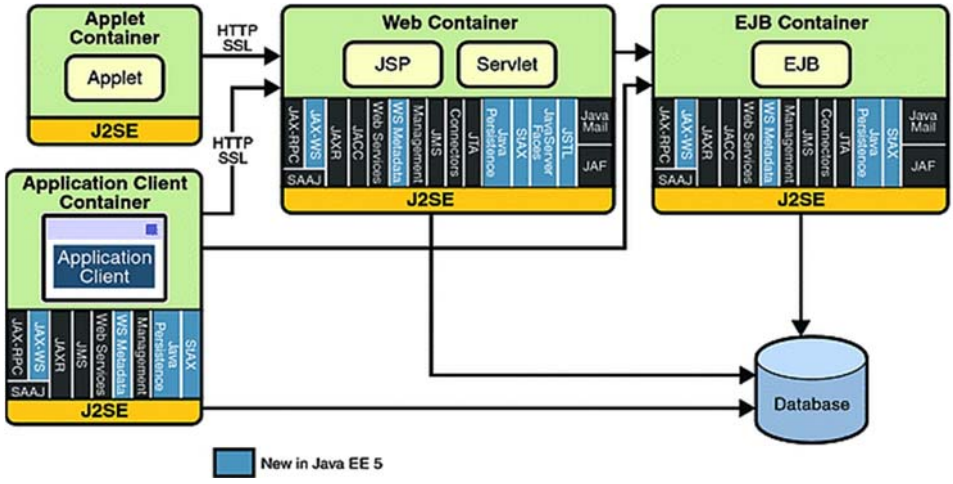
Trong quy trình phát triển ứng dụng, từng thành phần của ứng dụng không được phát triển từ cơ bản mà sẽ sử dụng nền tảng hỗ trợ để thực hiện điều này, gọi là các *Application server*. Giống như bộ thư viện Swing, Application server cung cấp các dịch vụ, tính năng hữu ích như phương thức truyền tải thông tin giữa các máy tính khác nhau, phương thức quản lý cơ sở dữ liệu, phương thức quản lý các giao dịch với người dùng trong ứng dụng…

Trong trường hợp ứng dụng được xây dựng dựa trên nhiều Application server khác nhau về kiến trúc thì chúng ta cần một giải pháp để có thể xây dựng ứng dụng mà không cần quan tâm đến nền tảng Application Server nào được sử dụng cả. Giải pháp tuyệt vời này đã được Java và Sun Microsystems giới thiệu qua một tập các thư viện, gọi là Java 2 Platform Enterprise Edition – J2EE.

J2EE có thể được xem là một tập hợp các đặc tả về giao diện lập trình ứng dụng (API -Application Programming Interface), một kiểu kiến trúc tính toán phân tán, và các định nghĩa về quy trình đóng gói các đối tượng trong ứng dụng. J2EE còn là 1 tập hợp chứa các thành phần (component), vật chứa (container) và các dịch vụ (service) đã được chuẩn hoá nhằm phục vụ cho việc phát triển và triển khai các ứng dụng phân tán trên các kiến trúc hệ thống khác nhau.

Mục tiêu chính của J2EE là hỗ trợ các hệ thống phức tạp thực thi các nghiệp vụ trên quy mô rộng lớn. Ứng dụng doanh nghiệp không phải là một phần mềm nhỏ gọn có thể được cài đặt và vận hành trên một máy tính đơn lẻ mà nó cần nhiều rất nhiều tài nguyên để đạt được hiệu suất tối ưu nhất. Vì vậy triển khai ứng dụng đồng nghĩa với triển khai từng mảng của ứng dụng này trên các hệ thống khác nhau nhưng cần phải có một sự kết nối cho các thành phần, các giao diện, các dịch vụ của từng mảng. Tất cả những điều này đã được chuẩn hoá và đặc tả chi tiết trong J2EE. Như vậy, một ứng dụng doanh nghiệp có thể được xây dựng bằng cách sử dụng các API của J2SE kết hợp với các công nghệ hỗ trợ trong J2EE.

Nhiều người hiểu nhầm rằng sự ra đời của J2EE đã thay thế cho J2SE. Điều này hoàn toàn không đúng vì bản chất của J2SE chính là thành phần cốt lõi mà J2EE xây dựng trên. Mô hình dưới đây tổng quát hoá kiến trúc và các công nghệ mà J2EE hỗ trợ với nền tảng là J2SE.



Hình 1.1 Kiến trúc nên tảng của công nghệ J2EE (Source: Java EE Tutorial, Oracle)

1.2 LỢI ÍCH CỦA J2EE TRONG PHÁT TRIỂN ỨNG DỤNG DOANH NGHIỆP

J2EE cung cấp hàng loạt dịch vụ trọng yếu để người dùng có thể hoàn toàn tập trung vào việc phát triển xây dựng nghiệp vụ cho ứng dụng mà không cần quan tâm đến hạ tầng triển khai. Hiểu một cách đơn giản, các dịch vụ J2EE cung cấp giống như điện và nước sử dụng trong sinh hoạt. Thực tế, chúng ta không quan tâm đường dây điện được thiết kế như thế nào, các ống nước bố trí ra sao mà chỉ cần sử dụng được chúng để phục vụ nhu cầu của chúng ta. Việc sử dụng điện và nước hoàn toàn thông qua các thiết bị điều khiển tiêu chuẩn như: vòi nước, công tắc điện, ổ cắm điện, ống xả nước...Nếu như hệ thống điện và nước không sẵn dùng, mọi thứ đều phải được bắt đầu từ số 0 khiến cho chi phí xây dựng, lắp đặt, kết nối cho các hệ thống này trở nên đắt đỏ và tốn nhiều thời gian.

Tương tự như vậy, trong việc phát triển một ứng dụng doanh nghiệp, tại sao chúng ta không tự xây dựng một kiến trúc hạ tầng phù hợp của riêng chúng ta, nghĩa là bắt đầu mọi thứ từ số không? Câu trả lời rất rõ ràng: Nếu làm như vậy, bất lợi đầu tiên là chi phí rất lớn về thời gian và tiền bạc. Bất lợi thứ hai là khả năng chia sẻ và hợp tác với các kiến trúc của doanh nghiệp khác. Thay vào đó, sử dụng J2EE, ta có được các vật chứa, các bộ kết nối và các thành phần để làm được việc này. Cụ thể hơn, nếu ứng dụng doanh nghiệp được phát triển trên nền tảng J2EE thì sẽ có khả năng cộng tác hoặc thực thi trên bất kỳ nền tảng nào, miễn là sử dụng J2EE làm hạ tầng.

1.3 MÔ HÌNH ỨNG DỤNG ĐA TẦNG TRONG J2EE

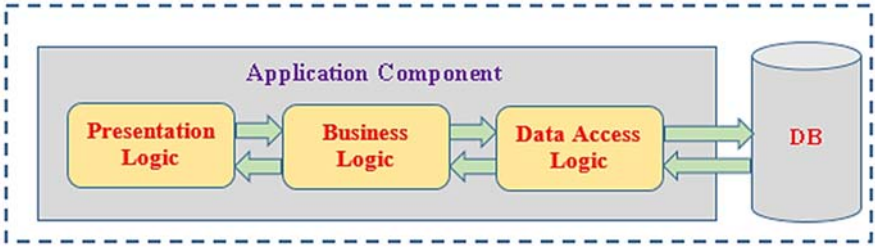
J2EE sử dụng mô hình ứng dụng theo kiểu đa tầng. Về mặt logic, ứng dụng được chia thành các thành phần theo chức năng và các thành phần này sẽ được phân tầng phụ thuộc vào môi trường mà thành phần đó được cài đặt. Một ứng dụng thường được chia thành 3 thành phần đó là:

- Phần trình bày (Presentation logic): cung cấp các chức năng điều khiển cho người dùng và cho phép người dùng tương tác với ứng dụng.
- Phần xử lý nghiệp vụ (Business logic): xử lý các tiến trình, giải quyết các yêu cầu từ người dùng. Ví dụ: dựa trên số giờ làm việc và tiền lương định mức chi trả mà tính lương nhân viên.
- Phần lưu trữ, thao tác trên cơ sở dữ liệu (Data Access Logic): đảm nhận việc đọc, ghi dữ liệu diễn ra tại bất kỳ nơi đâu, bất kỳ thời điểm nào.

Như vậy, việc tổ chức 3 thành phần logic này như thế nào chính là giải pháp tổ chức cho kiến trúc phân tầng của một ứng dụng. Một số kiểu tổ chức kiến trúc phân tầng thường thấy:

- Kiến trúc phân tầng đơn giản nhất là đơn tầng (Single-tier architecture): tất cả các thành phần logic của ứng dụng đều được cài đặt, vận hành trên cùng một nguồn tài nguyên (một máy tính đơn nhất). Ưu điểm là dễ thiết kế. Nhược điểm là cả 3 thành phần logic đều nằm trên cùng nguồn tài

nguyên nên cập nhật, bảo trì cho toàn bộ hệ thống sẽ phát sinh chi phí rất lớn. Ngoài ra, tính hợp tác trong thực thi nghiệp vụ của ứng dụng trong một môi trường lớn là rất kém, khó trao đổi và tổng hợp dữ liệu từ nhiều nguồn khác nhau. Hình 1.2 mô tả kiến trúc ứng dụng theo mô hình đơn tầng.



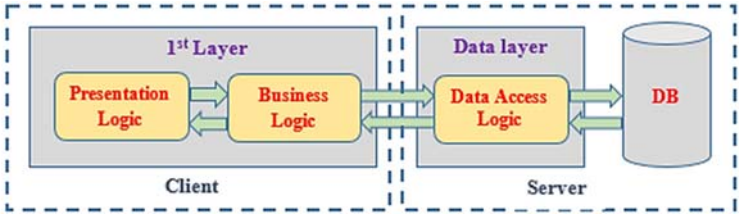
Hình 1.2 Mô hình kiến trúc đơn tầng

– Kiến trúc hai tầng (Two-tier architecture): còn gọi là mô hình Client – Server. Trong kiến trúc này, chức năng của một ứng dụng được chia thành 2 phần (chương trình) thực thi riêng rẽ nhau: 1 chương trình Client và 1 chương trình Server như được mô tả trong Hình 1.3.

  ✓ Client đảm nhận chức năng về giao diện người dùng như: tạo form để nhập dữ liệu, tạo các thông báo cho người dùng…

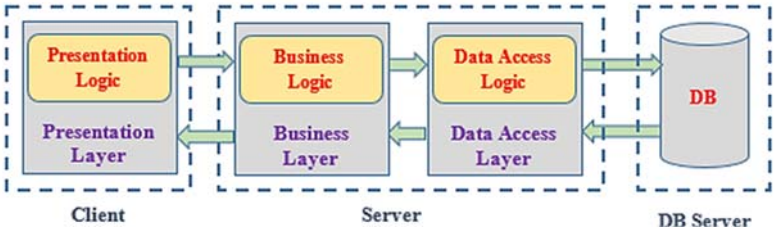
  ✓ Server đảm nhận chức năng về lưu trữ và truy xuất dữ liệu.

  ✓ Business Logic được cài đặt ở Client hoặc Server, dẫn đến phát sinh 2 kiến trúc con là Fat-Client và Fat-Server. Thông thường mô hình này yêu cầu cần phải thiết lập 1 giao thức (protocol) để thực hiện truyền thông Client và Server.



Hình 1.3 Mô hình kiến trúc hai tầng

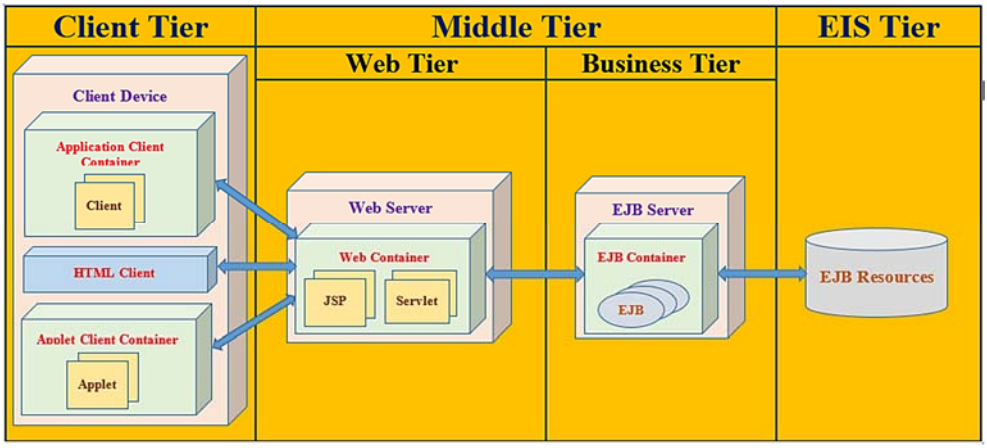
– Kiến trúc ba tầng (Three-tier architecture): mỗi tầng chứa một thành phần logic của ứng dụng. Việc tổ chức các tầng trong cùng một máy hoặc trên nhiều máy khác nhau phụ thuộc vào khả năng mở rộng của doanh nghiệp. Giải pháp kiến trúc 3 tầng này đáp ứng rất tốt với những thay đổi về cách thức xử lý nghiệp vụ của doanh nghiệp bởi vì mọi sự thay đổi chỉ diễn ra trên Business Layer nơi chứa các xử lý nghiệp vụ mà không ảnh hưởng đến 2 tầng còn lại. Kiến trúc này được mô tả trong Hình 1.4.



Hình 1.4 Mô hình kiến trúc 3 tầng Client-Server-Database

– Kiến trúc n tầng (n-tier architecture): thường được áp dụng cho các ứng dụng mà các chức năng được tổ chức theo hình thức phân tán. Các ứng dụng triển khai theo kiến trúc này cần đảm bảo được hạ tầng về mạng và khả năng xử lý trên từng tầng phải đạt được tiêu chuẩn nhất định.

Ứng dụng J2EE có kiến trúc đa tầng bao gồm Client Tier, Middle Tier và Back-End Tier. Mô hình phân tầng của ứng dụng J2EE được minh hoạ trong Hình 1.5



Hình 1.5 Mô hình kiến trúc phân tầng của ứng dụng J2EE

– Client Tier: là thành phần thực thi phía client. Đây có thể là 1 ứng dụng Java đơn giản, các trang HTML tĩnh, các trang HTML động hoặc các Applets trên trình duyệt web.

– Middle Tier: gồm hai thành phần con là Web Tier và Business Tier.

  ✓ Web Tier: là nơi các web container quản lý các servlet và các trang JSP xử lý yêu cầu đến từ Client Tier, là trung gian xử lý nghiệp vụ của ứng dụng với Back-End Tier. Người dùng thực hiện yêu cầu dịch vụ từ Client Tier đến Web Tier. Web Tier sẽ xử lý các yêu cầu và trả về kết quả cho người dùng. Một truy vấn như vậy sẽ dẫn đến truy vấn tương ứng từ Web Tier đến một trong nhiều các thành phần xử lý nghiệp vụ (các EJBs) trên Business Tier và hơn nữa có thể dẫn đến sự tương tác giữa Business Tier và Back End Tier nếu cần thiết.

- ✓ Business Tier: gồm các EJB (Enterprise Java Beans) Container của J2EE, dùng để quản lý các EJB. Các EJB chứa xử lý nghiệp vụ (business rule) của ứng dụng và làm nhiệm vụ tương tác với với Back-End Tier tùy thuộc vào nghiệp vụ cần xử lý, đồng thời trả về các kết quả cho Web Tier.

– Back-End Tier: là nơi để chứa dữ liệu cho các ứng dụng, thể hiện qua cơ sở dữ liệu quan hệ.

Ví dụ, một ứng dụng mua bán trực tuyến trên web vận hành theo kiến trúc trên có thể được minh họa như sau:

– Người dùng tương tác với giao diện của trang web (Client Tier), chọn 1 thao tác bất kỳ (mua, bán, đổi, trả…) được hiển thị trên giao diện.

– Tại Middle Tier, các Servlet và JSP sẽ tương tác với các dịch vụ được yêu cầu tương ứng (mua, bán, đổi, trả…) được đóng gói trong các EJBs và trả về trang web với nội dung chứa kết quả.

– Tại Back-End Tier, dữ liệu trong kho lưu trữ được cập nhật lại (giảm số lượng hàng, thêm số lượng hàng…) dựa trên các xử lý nghiệp vụ được thực hiện trên Middle Tier.

1.4 MÔ HÌNH MVC

Những tiêu chuẩn cần đảm bảo khi xây dựng một ứng dụng doanh nghiệp gồm:

– Construction and testing (xây dựng và kiểm thử): cách thức xây dựng và phát triển? Công nghệ lựa chọn để phát triển?

– Reuseable (có thể sử dụng lại): ứng dụng có sử dụng các thành phần đã được chuẩn hoá hay không?

– Scalability (tính dãn nở): ứng dụng sẽ thực thi thế nào khi có nhiều giao dịch với nhiều người dùng?

– Security (an ninh): ứng dụng có các cơ chế gì để bảo vệ hệ thống trước các cuộc tấn công, phát tán mã độc…

Các mô hình thiết kế phần mềm (desgin pattern) được đề xuất sử dụng nhằm đảm bảo việc xây dựng ứng dụng luôn thỏa mãn các tiêu chuẩn trên. Một trong những design pattern được sử dụng nhiều trong thiết kế ứng dụng hiện nay là mô hình MVC (Model, View và Controller). MVC là giải pháp hoàn hảo cho ứng dụng lớn và chạy trên các nền tảng khác nhau như: thiết bị di động, tablet hay các trình duyệt web khác nhau trên các hệ điều hành khác nhau. Trong MVC, các thành phần được thiết kế tách biệt bằng cách đưa giao diện người dùng vào View, đưa các xử lý nghiệp vụ và lưu trữ dữ liệu vào Model và các thành phần điều khiển vào Controller.