

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**BÁO CÁO THỰC TẬP  
ITSS- HEDSPI**

<b>GV Hướng dẫn</b>	Nguyễn Ngọc Dũng
<b>Công ty thực tập</b>	Việt Nam Lab-GMO Internet INC
<b>Sinh viên</b>	Chu Quang Viên
<b>Lớp</b>	AS1-HEDSPI-K54

*Hà Nội, tháng 7 năm 2013*

## Mục lục:

Giới thiệu chung .....	4
1.    Mục đích của việc thực tập.....	4
2.    Các công việc được giao .....	4
I.    Framework Seasar2& SAAstruts.....	6
1.    Seasar2 là gì ? .....	6
2.    SAAstruts là gì ? .....	7
2.1.    Struts là gì? .....	7
2.2.    Lợi thế của việc sử dụng Struts.....	8
2.3.    Vấn đề khi sử dụng Struts.....	8
2.4.    SAAstruts là gì ? .....	10
3.    Hướng dẫn cài đặt Seasar2 trên Eclipse .....	11
3.1.    Cài đặt Java .....	11
3.2.    Cài đặt Tomcat .....	11
3.3.    Cài đặt Eclipse .....	13
3.4.    Cài đặt các plugin Seasar2 .....	14
3.5.    Cài plugin cho Apache Tomcat.....	17
3.6.    Cài đặt MySQL và kết nối với Eclipse .....	20
3.7.    Xây dựng Project Seasar2.....	25
3.8.    Cài đặt Maven và tạo file war cho project Seasar2 .....	31
3.9.    Cài đặt SVN plugin và commit source code .....	34
II.    Cấu trúc của Framework SAAstruts .....	36
III.    Game TicTacToe.....	49
1.    Thành phần,cấu trúc game .....	49
2.    Các công nghệ mới và giải thuật sử dụng trong game .....	51
3.    Một số thủ thuật cần nhớ khi làm game TicTacToe .....	58
4.    Log4j.....	59
5.    Demo.....	60

## Tài liệu tham khảo

1. <http://www.w3schools.com/>
2. <http://sastruts.seasar.org/tutorial.html>
3. <http://yakinikunotare.boo.jp/orebase2/sastruts>
4. <http://www.freshdesignweb.com/beautiful-css-html-login-form-templates.html>
5. <http://vietgamedev.net/blog/171/html5-animating-sprites-c%C3%A1ch-t%E1%BA%A1o-%E1%BA%A3nh-%C4%91%E1%BB%99ng-tr%C3%AAn-game-canvas/>
6. <http://blueluanar.wordpress.com/2012/12/10/cau-hinh-log4j-su-dung-file-properties/>
7. <http://d.hatena.ne.jp/idesaku/20081226/1230263190>

# Giới thiệu chung

## 1. Mục đích của việc thực tập

- củng cố kiến thức lý thuyết đã được trang bị trong nhà trường và vận dụng những kiến thức đó vào thực tế.
- Tìm hiểu thêm các công nghệ mới đang được áp dụng trong thực tiễn.
- Rèn luyện kỹ năng thực hành thông qua việc tham gia vào các dự thực tế
- Rèn luyện kỹ năng làm việc theo nhóm.
- Rèn luyện ý thức chấp hành kỷ luật lao động, thái độ giao tiếp với mọi người.
- Phát huy tinh thần học hỏi, chủ động sáng tạo trong việc giải quyết các vấn đề của thực tiễn.
- Nâng cao khả năng sử dụng tiếng Nhật trong công việc (nhất là đối với công ty Nhật)

## 2. Các công việc được giao

- Ngày bắt đầu: 24/6/2013
- Ngày kết thúc: 27/6/2013
- Nội dung công việc: tiến hành cài đặt
  - ✓ Java SE 7
  - ✓ Tomcat 7
  - ✓ Eclipse: cài đặt thêm các plugin:
    - DbLauncher
    - Sysdeo/SQLI Tomcat Launcher
    - Dolteng
    - Kijimuna
    - Resource Synchronizer
    - SASTrutsPlugin
    - Subclipse
  - ✓ Mysql

Nghiên cứu framework Seasar2

- Ngày bắt đầu: 24/6/2013

- Ngày kết thúc: 5/7/2013
- Nội dung công việc:
  - ✓ Tìm hiểu về framework Seasar2: các tạo project, cấu trúc của project, cách thức hoạt động của framework ...
  - ✓ Tiến hành thực hiện game tic-tac-toe: sử dụng javascript để làm AI cho game

Sử dụng Mysql, Log4j, Maven

- Ngày bắt đầu: 8/7/2013
- Ngày kết thúc: 19/7/2013
- Nội dung công việc:
  - ✓ Tạo các chức năng login, logout, register
  - ✓ Sử dụng Log4j để kiểm soát lỗi khi code Java
  - ✓ Tiếp tục thực hiện game tic-tac-toe với chức năng chơi 2 người: sử dụng cơ sở dữ liệu để lưu trạng thái của bàn cờ, qua đó có thể chơi giữa hai người chơi, thêm chức năng mời người chơi và từ chối
  - ✓ Cài đặt maven plugin để tạo file war, từ đó upload lên server

Sử dụng Html5 để thiết kế giao diện và hiệu ứng

- Ngày bắt đầu: 22/7/2013
- Ngày kết thúc: 26/7/2013
- Nội dung công việc:
  - ✓ Nghiên cứu thẻ canvas trong html5: cách vẽ ảnh lên canvas, thực hiện di chuyển ảnh, xoay ảnh ..., sử dụng javascript
- Thêm animation vào game tic-tac-toe

# I. Framework Seasar2& SStruts

## 1.Seasar2 là gì ?

- ❖ Seasar2 Framework là một ứng dụng mã nguồn mở tương tự như Spring Framework của Java.Ban đầu nó được phát triển cho các nền tảng Java bởi Yasuo Higa(Nhật Bản) nhưng hiện tại thì .NET và PHP cũng được hỗ trợ rất tốt.Khi mới được phát triển và sử dụng thì Seasar2 có một số lượng lớn người dùng là người Nhật Bản nhưng kể từ khi bản tutorial bằng tiếng Anh được công bố tại hội nghị JavaOne ở Tokyo năm 2005 thì Seasar2 đã được cả thế giới biết đến và ngày càng phát triển.
- ❖ Seasar là một container có trọng lượng nhẹ ,được hỗ trợ DI (Dependency Injection) và AOP (Aspect Oriented Programming). Framework này thực hiện hóa DI và AOP thông qua một thư viện gọi là S2Container.
- ❖ Mục tiêu của framework được tóm tắt như sau:
  - ✓ Gần đây khi nhắc đến Java thì mọi người thường nói rằng “Năng suất coding Java thấp quá...”. Sự thật là khi coding với Java thì người dùng phải thực hiện nào là setting phiên phức như là tạo file định nghĩa, khi sửa source thì phải khởi động lại chương trình...
  - ✓ Seasar2 là framework được phát triển nhằm mục đích nâng cao năng suất phát triển sử dụng Java. Về việc phải viết các file setting thì hầu như không có, mặt khác khi chỉnh sửa code thì không cần phải khởi động lại chương trình mà framework sẽ tự động nhận và load sự thay đổi.
  - ✓ Để sử dụng framework Seasar2 cho Eclipse thì cần sử dụng plugin của Eclipse là Dolteng

## 2.SA Struts là gì ?

### 2.1. Struts là gì?

- ❖ Struts là một framework mã nguồn mở phục vụ việc phát triển các ứng dụng Web trên Java. Sử dụng mẫu thiết kế Model-View-Controller (MVC), Struts giải quyết rất nhiều các vấn đề liên quan đến các ứng dụng Web hướng business đòi hỏi hiệu năng cao sử dụng Java servlet và JSP. Struts cơ bản định hình lại cách các Web programmer nghĩ về và cấu trúc một ứng dụng Web. Struts được phát triển bởi Craig McClanahan và được bảo trợ bởi Apache, nhóm Jakarta vào tháng 6 năm 2001.
- ❖ Thư viện thẻ Struts để xây dựng các thành phần trình diễn trong một ứng dụng Struts là một tập hợp các thư viện Java và có thể chia ra làm các nhóm sau:

- Framework cơ sở

Framework cơ sở cung cấp các tính năng MVC cốt lõi. Nền tảng của framework cơ sở là Controller servlet: `ActionServlet`. Phần còn lại của framework cơ sở bao gồm các lớp cơ sở mà ứng dụng của bạn sẽ extend và các class tiện ích. Nổi bật nhất trong các lớp cơ sở là lớp `Action` và lớp `ActionForm`. Lớp `Action` được sử dụng bởi `ActionServlet` để xử lý các request xác định. Lớp `ActionForm` được sử dụng để capture dữ liệu từ các HTML form và được sử dụng để chuyển dữ liệu trở lại View để generate response.

- Thư viện thẻ JSP

Struts đưa ra các thư viện thẻ để hỗ trợ việc lập trình View logic trong JSP. Các thư viện thẻ JSP cho phép các JSP author sử dụng các thẻ giống HTML.

- + HTML: Sử dụng để generate các THML form tương tác với Struts API
- + Bean: Sử dụng để làm việc với các Java bean object trong JSP

+ Logic: Sử dụng để implement các logic điều kiện đơn giản trong JSP

+ Nested: Sử dụng để cho phép các mức nested tùy ý

➤ Tiles plugin

Tiles là một JSP template framework giúp thuận tiện cho việc tái sử dụng mã HTML.

➤ Validator plugin

Validator là một framework với khả năng thực hiện xác nhận tính hợp lệ của dữ liệu ở cả phía server và client.

## 2.2. Lợi thế của việc sử dụng Struts

Struts 2 được xây dựng và phát triển dựa trên nền tảng mô hình MVC nên nó thừa hưởng được đầy đủ các ưu điểm mà mô hình MVC đem lại.

- Dễ dàng tùy chỉnh (customize) chu kỳ xử lý (request lifecycles ) cho từng action
- Giải quyết hiệu quả vấn đề internationalization và localization trong các ứng dụng web
- Tự động chuyển đổi kiểu dữ liệu chuỗi truyền thống trong tham số request parameter thành các đối tượng lớp dữ liệu java => tiết kiệm được thời gian và công sức cho các lập trình viên
- Cung cấp các thẻ tag, các themes và templates giúp cho việc làm giao diện GUI trở nên dễ dàng, nhanh lẹ và tăng tính tái sử dụng.
- Tính mở rộng (Extensibility) cao thông qua việc hỗ trợ các plug-in
- Hỗ trợ portal.
- Hỗ trợ AJAX
- Dễ dàng tích hợp với Spring framework(\*) và Hibernate.

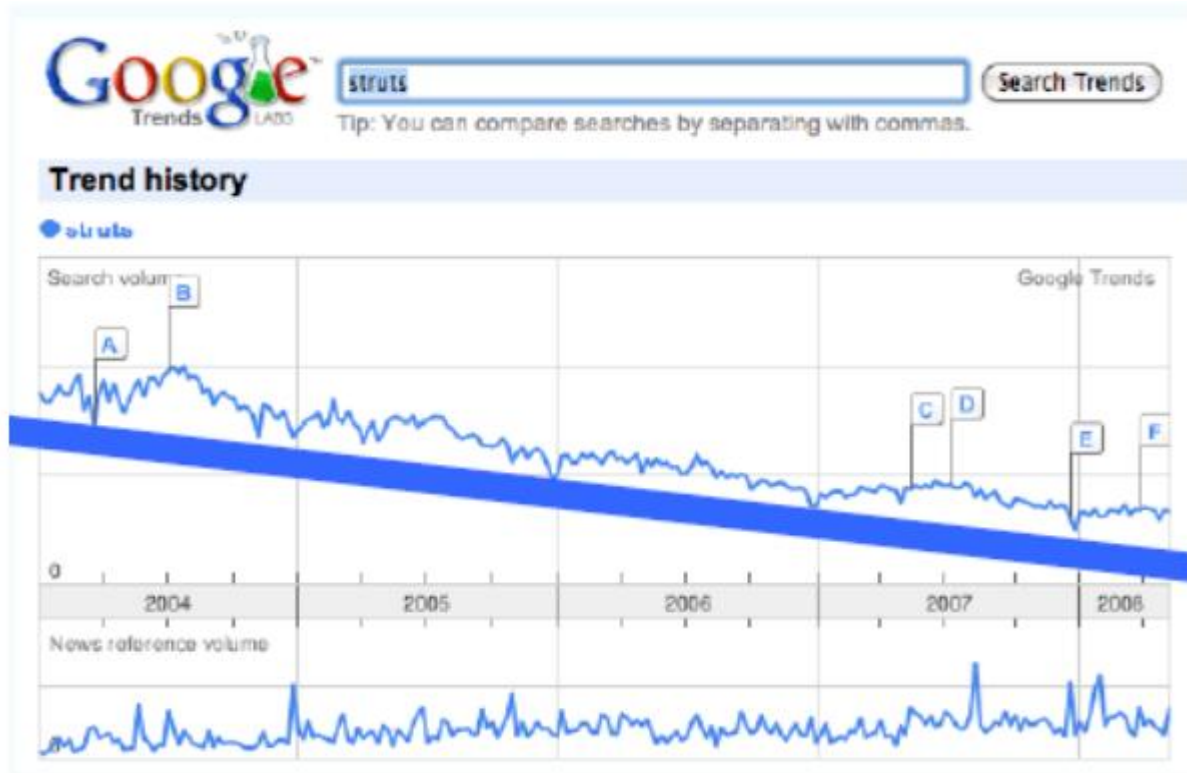
## 2.3. Vấn đề khi sử dụng Struts

- Để sử dụng MVC với chuẩn RequestDispatcher, ta cần nghiên cứu sâu với chuẩn JSP và Servlet APIs. Để sử dụng MVC với Struts, ta còn phải hiểu rõ cả framework rộng lớn và phức tạp, nó tương tự như việc tìm hiểu cả cái cốt lõi của hệ thống. Điều bất lợi này đặc biệt đáng kể với những dự án nhỏ, những dự án có

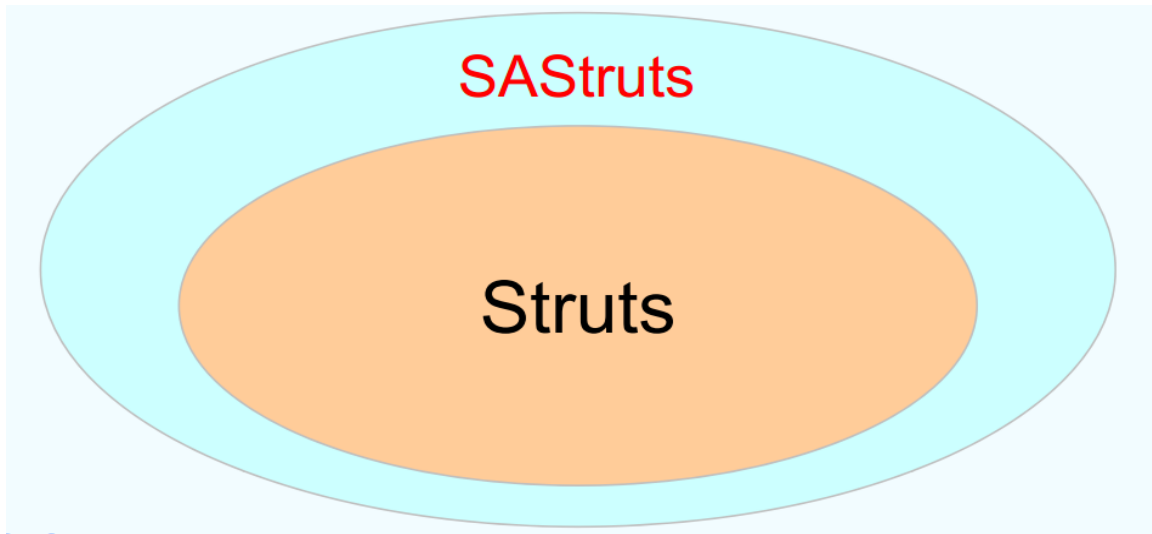


ít thời gian để thực hiện, và những lập trình viên có ít kinh nghiệm; ta mất rất nhiều thời gian trong việc nghiên cứu Struts khi thực hiện đề án.

- Sự phát triển nhanh chóng đến khủng khiếp của tập tin cấu hình
- Chinh sửa mã nguồn cần có thời gian.
- Sự gia tăng về số lượng các class Action



## 2.4. SAStruts là gì ?



SAStruts là tên viết tắt của SuperAgileStruts. Nó là một framework được phát triển dựa vào nền tảng của Struts.

❖ Lợi thế của việc sử dụng SAStruts:

struts-1.2.9.jar	536KB
sa-struts-1.0.1.jar	92KB

- Đơn giản
  - Số lượng các class là 79
- Năng suất cao
  - Thời gian chuẩn bị mã hóa ngắn.
  - Thời gian từ lúc gỡ lỗi đến phản ánh ngắn.

### 3.Hướng dẫn cài đặt Seasar2 trên Eclipse

Phần cài đặt cho Seasar2 khá là khó khăn nên em sẽ báo cáo kĩ từng bước cài đặt. Hiện tại đã có phiên bản JDK 7 nhưng phần hướng dẫn cài đặt được sử dụng trên trang tutorial của Seasar2 với phiên bản JDK 6 và Tomcat 6,nên những nếu muốn sử dụng các phiên bản mới nhất của JDK và Tomcat thì làm tương tự hướng dẫn với tên các package tương ứng với phiên bản hiện tại.Chi tiết hướng dẫn có thể xem tại đây: “[http://datadb04.u-aizu.ac.jp:8080/GT3/GT3\\_2012/s1180134/html/seasar2Installation.html](http://datadb04.u-aizu.ac.jp:8080/GT3/GT3_2012/s1180134/html/seasar2Installation.html)”

#### 3.1. Cài đặt Java

- Go to: "<http://www.oracle.com/technetwork/jp/index.html>"
- Download "JDK 6 Update xx with Java EE"

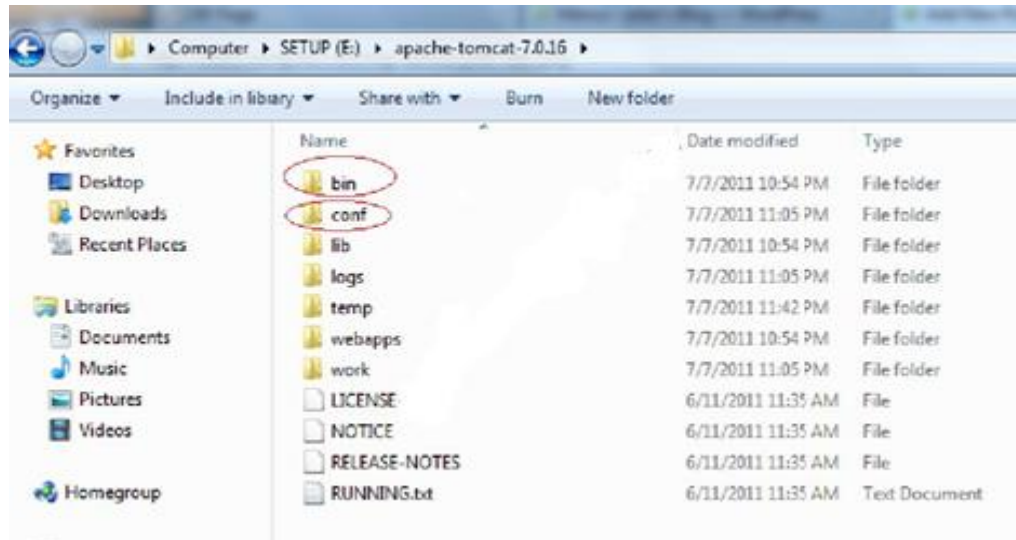
#### 3.2. Cài đặt Tomcat

- Go "<http://tomcat.apache.org>"
- Go the link of Tomcat 6.x
- In "Binary Distributions", select "Windows Service Installer"
- Execute downloaded file
  - ✓ Click "Next "
  - ✓ Agree the Tomcat's license
  - ✓ Click "Next "
  - ✓ Chose "Destination Folder" (ex. "C:\Tomcat6.0") and click "Next"
  - ✓ Decide port used by Tomcat, user name , and password, and click "Next "
  - ✓ Select JRE (ex. "C:\Program Files\Java\jre1.6.0\_07") and click "Install"
  - ✓ Check out "Run Apache Tomcat"and "Show Readme", and click "Finiah"

Hiện tại thì phiên bản Tomcat 7 không cần down về cài đặt mà có thể sử dụng trực tiếp nên thuận tiện hơn nhiều so với các phiên bản trước.Sau đây em xin hướng dẫn các config Tomcat 7.x như sau:

Sau khi download Tomcat về em đặt tại ổ E:\apache-tomcat-7.0.16

Khi config cấu hình cho Tomcat chỉ cần quan tâm tới 2 thư mục là bin và conf:



Đầu tiên hãy mở file /bin/startup.bat và thêm 1 dòng : set  
 “JAVA\_HOME=C:\Program Files\Java\jdk1.6.0\_16”  
 Các bạn chú ý set đúng đường dẫn JDK này

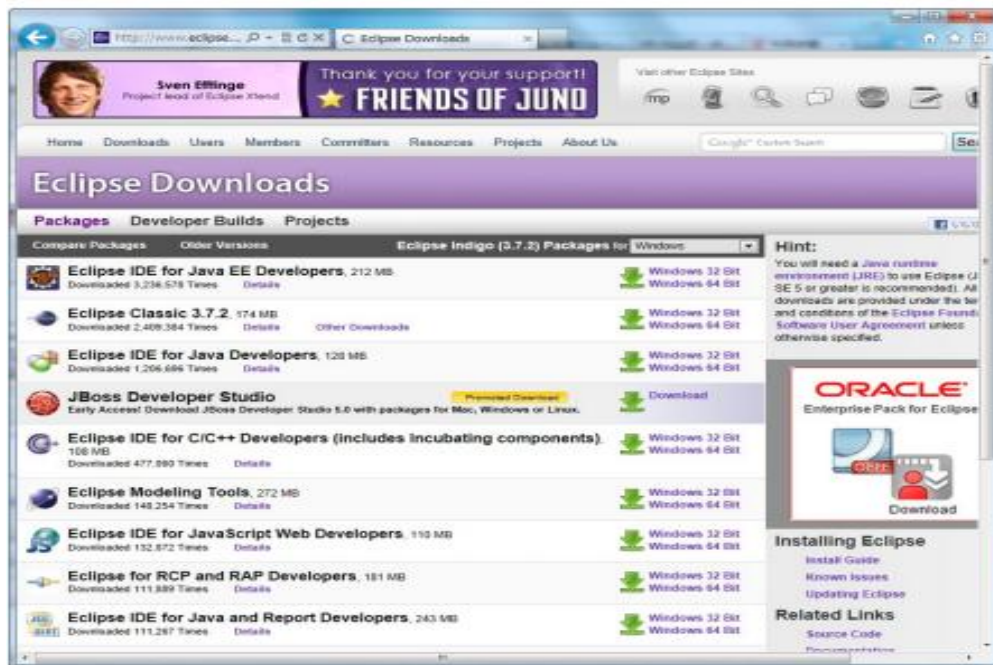
```

22 rem -----
23
24 rem Guess CATALINA_HOME if not defined
25 set "CURRENT_DIR=%cd%"
26 if not "%CATALINA_HOME%" == "" goto gotHome
27 set "CATALINA_HOME=%CURRENT_DIR%"
28 if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
29 cd ..
30 set "CATALINA_HOME=%cd%"
31 cd "%CURRENT_DIR%"
32 :gotHome
33 if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
34 echo The CATALINA_HOME environment variable is not defined correctly
35 echo This environment variable is needed to run this program
36 goto end
37 :okHome
38
39 set "EXECUTABLE=%CATALINA_HOME%\bin\catalina.bat"
40 set "JAVA_HOME=C:\Program Files\Java\jdk1.6.0_16"
41 rem Check that target executable exists
42 if exist "%EXECUTABLE%" goto okExec
  
```

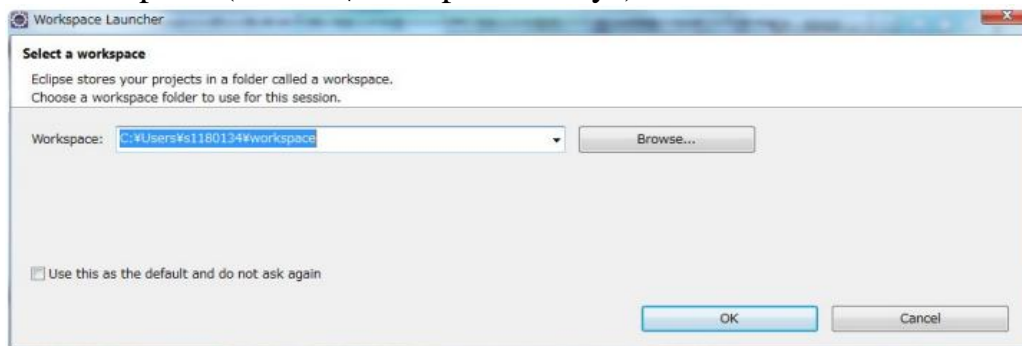
Riêng file config trong thư mục /conf/ thì ta dùng để cài đặt root cho từng project nên cũng không cần để ý tới. Sau khi sửa xong file /bin/ startup.bat thì ta save lại và chạy file /bin/startup.bat, lúc này apache Tomcat đã khởi động.

### 3.3. Cài đặt Eclipse

- Go "<http://www.eclipse.org/>"
- Go the link of "Downloads"
- Select "Eclipse IDE for java EE Developers"



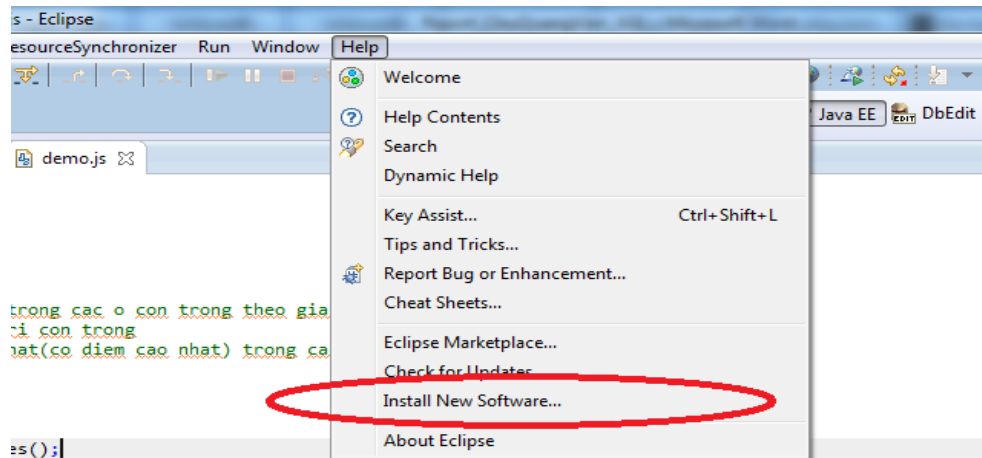
- Extract downloaded file (ex. under "C:\desktop")
- Boot Eclipse (\eclipse\eclipse.exe)
- Set "Workspace"(ex. "C:\workspace\_study")



- Check "Use this as the default and do not ask again"

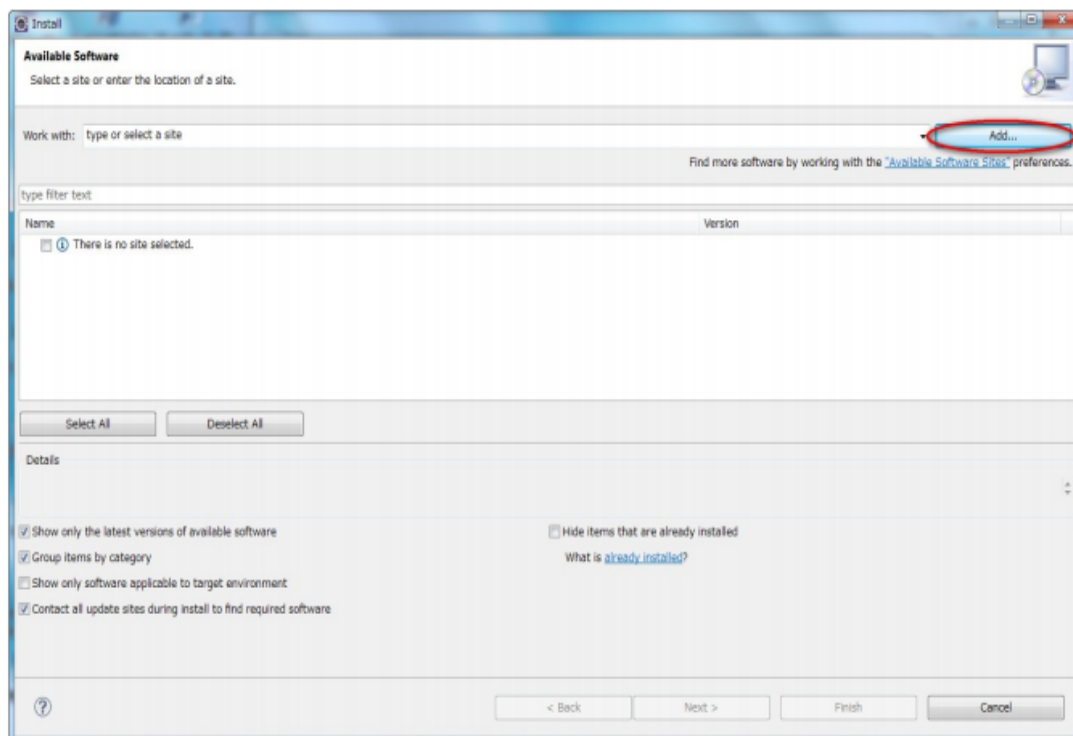
### 3.4. Cài đặt các plugin Seasar2

Để cài đặt các Plugin cho Eclipse thì trước hết ta tìm đến Help > Instal New Software



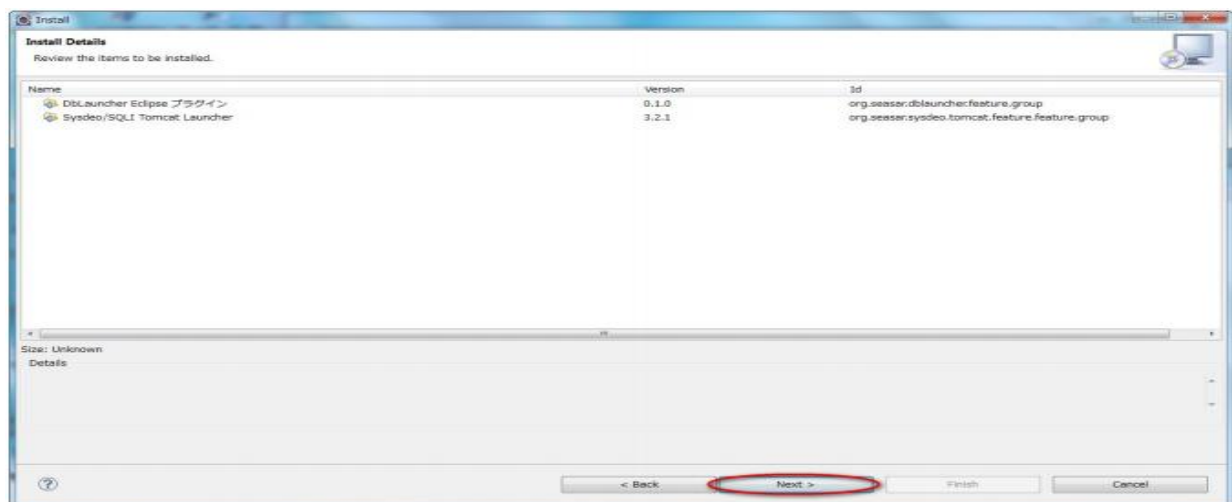
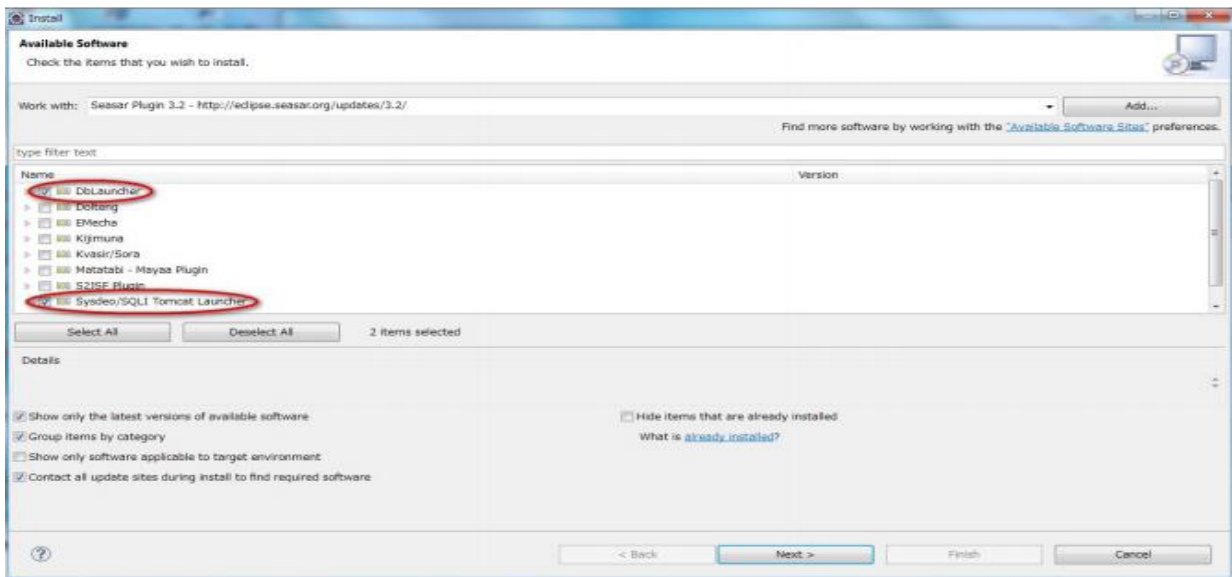
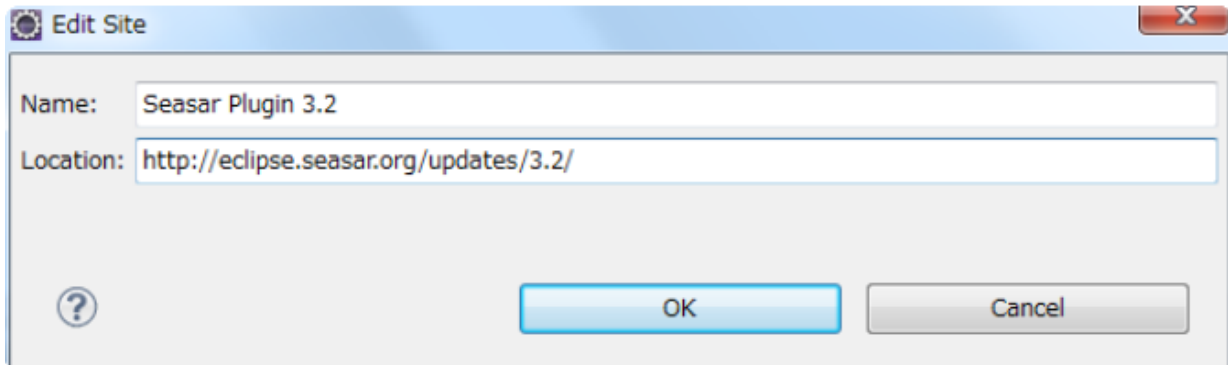
Và sau đó làm theo các bước sau đây:

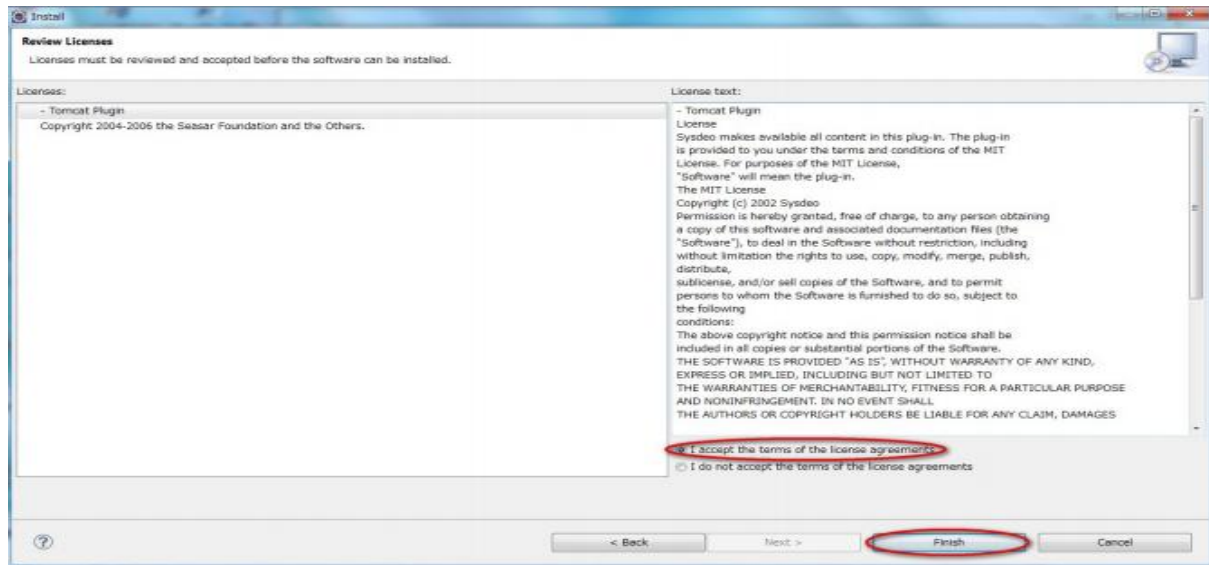
- In Install, click "Add..."





- a. Set "Name" as "Seasar Plugin 3.2" and "Location" as ["http://eclipse.seasar.org/updates/3.2/"](http://eclipse.seasar.org/updates/3.2/), and click "OK"



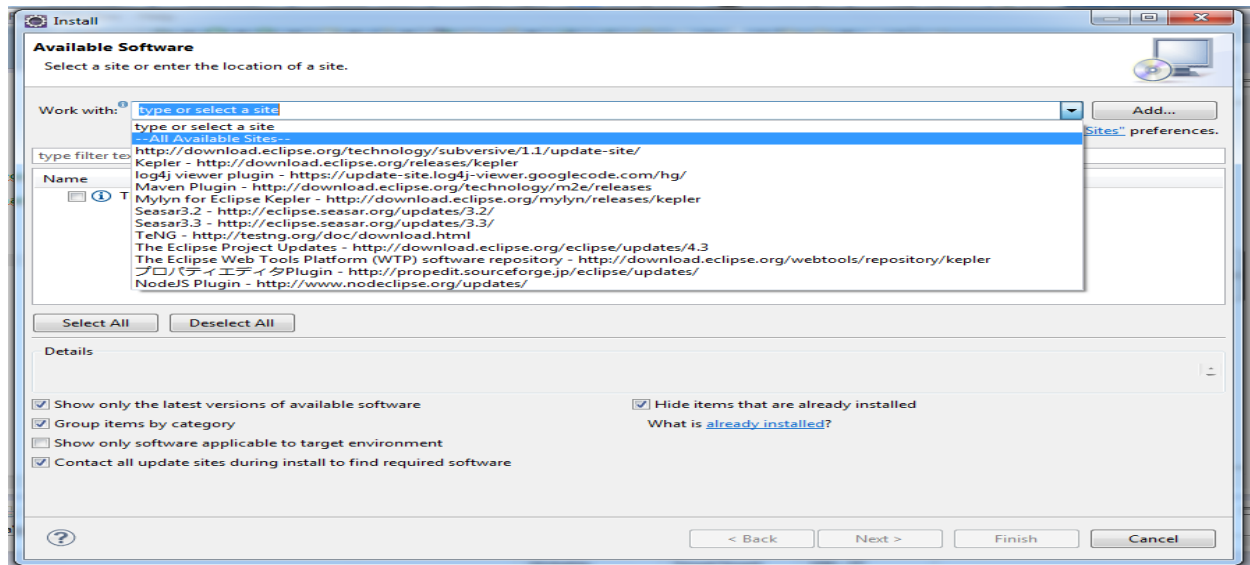


Tương tự như hình ảnh hướng dẫn cho Plugin Seasar3.2 thì ta cũng làm tương tự với các plugin sau đây:

- Choose plugins, **DbLauncher** and **Sysdeo/SQLI Tomcat Launch**
- In Install, click "Add...",
  - a. Set "Name" as "Seasar Plugin 3.3" and "Location" as ["http://eclipse.seasar.org/updates/3.2/"](http://eclipse.seasar.org/updates/3.2/), and click "OK"
- Choose plugins, **Dolteng**, **Kijimuna**, **ResourceSynchronizer**, and **SAStrutsPlugin**
- In Install, click "Add...",
  - a. Set "Name" as "Subclipse 1.6.x" and "Location" as ["http://subclipse.tigris.org/update\\_1.6.x"](http://subclipse.tigris.org/update_1.6.x), and click "OK"
- Choose plugin, Subclipse
- Click "OK" and reboot Eclipse

Sau khi đã làm đầy đủ các bước trên thì lúc này các plugin đã được add vào trong Eclipse và ta có thể kiểm tra như sau:





### 3.5. Cài plugin cho Apache Tomcat

Để cài đặt Plugin của Apache Tomcat cho Eclipse thì ta làm theo các bước sau:

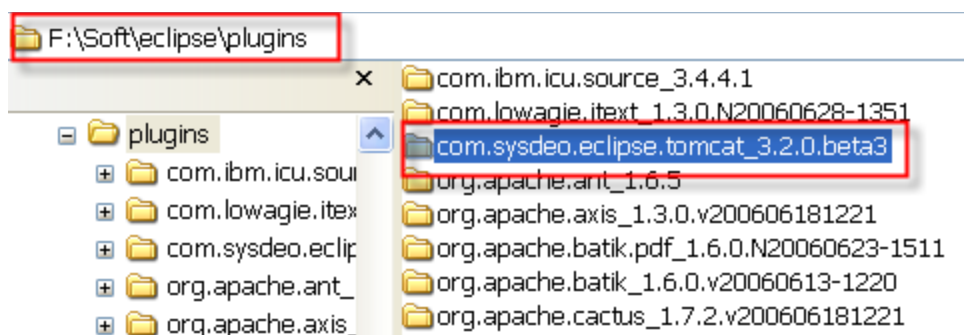
- Download Plugin Tomcat tại địa chỉ :

<http://www.eclipsestotal.com/tomcatPlugin.html#A3>

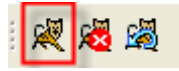
Khi download xong có dạng: tomcatPluginVxxx.zip :



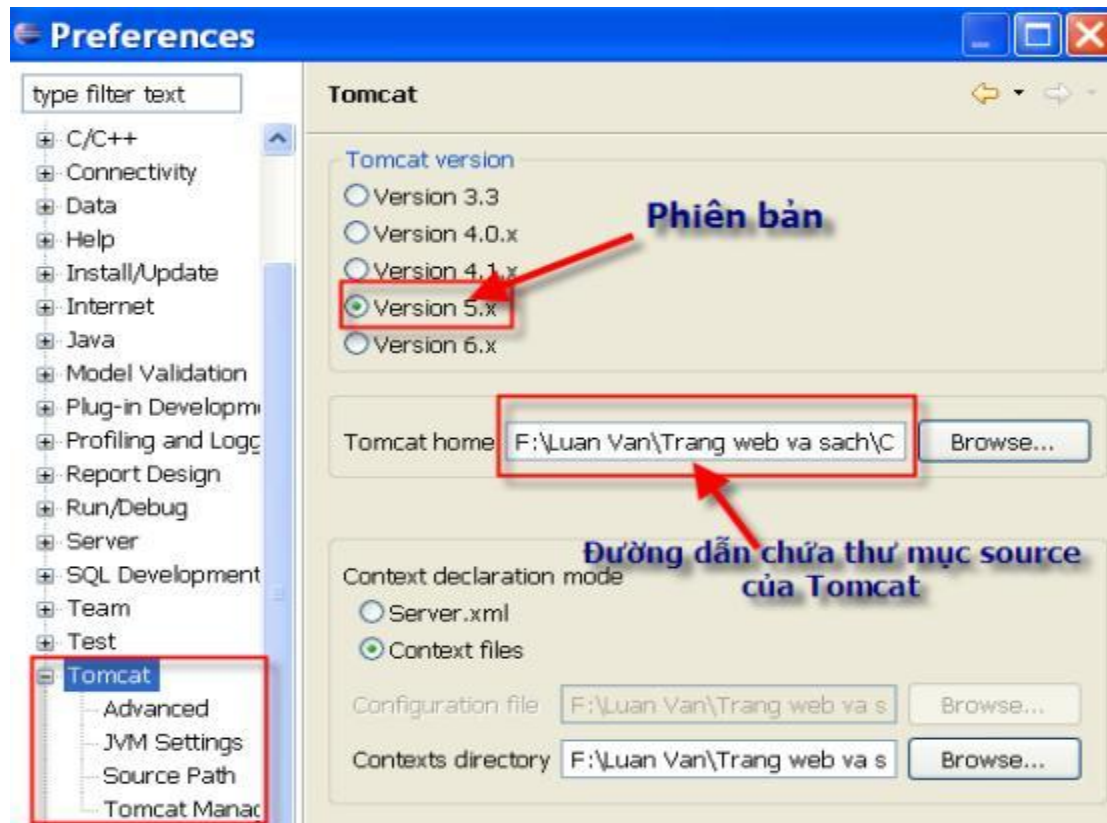
- Giải nén và đặt thư mục vừa giải nén vào nơi cài eclipse/plugin:



- Sau đó restart lại eclipse và khởi động lại,lúc này trên thanh công cụ của Eclipse sẽ xuất hiện biểu tượng của Tomcat Manager như sau:



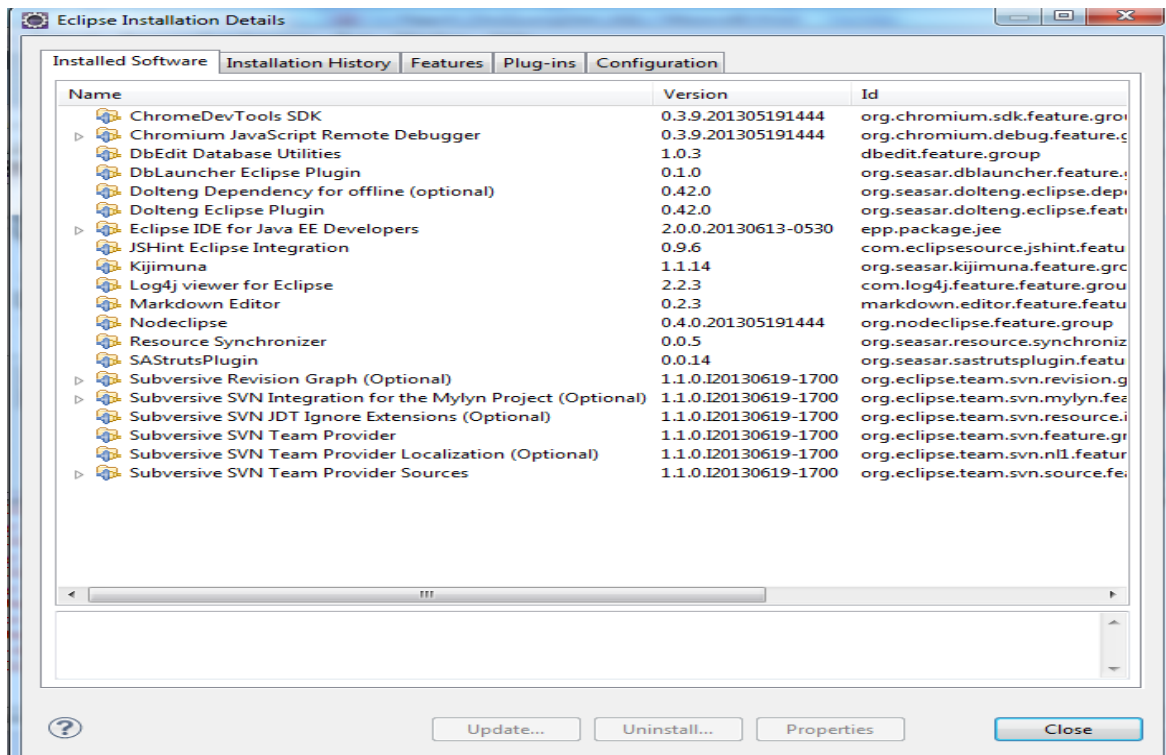
- Để cấu hình Tomcat trên Eclipse thì ta vào “Window->Preference->Tomcat” và chọn Version Tomcat muốn sử dụng,tốt nhất là nên chọn Version mới nhất của Tomcat



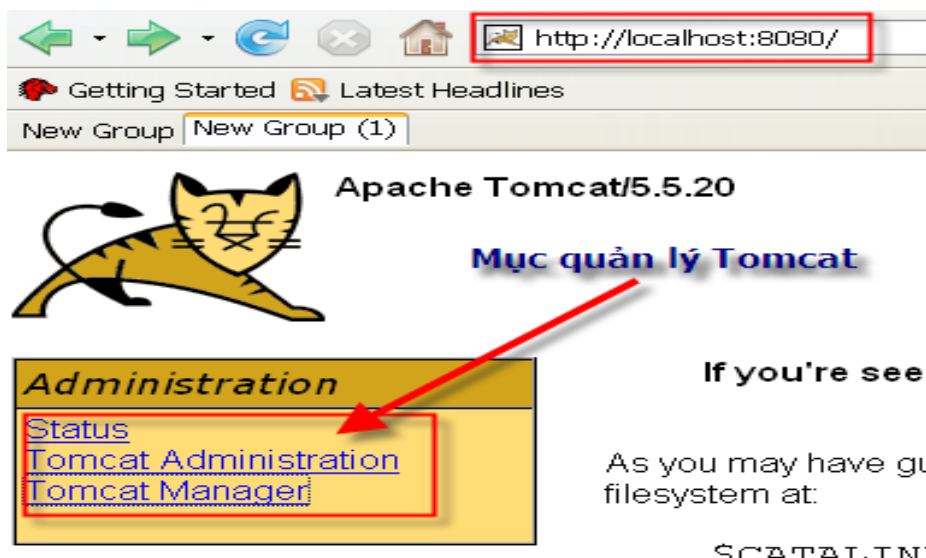
Khi cài Plugin **SQLI Tomcat Launch** của Seasar2 thì nó chỉ hỗ trợ Apache Tomcat Version 6.x,vì thế khi config Tomcat trong Eclipse sẽ chỉ xuất hiện Version 6.x,lúc ấy nếu phiên bản Tomcat mới nhất là tomcat 7 thì sẽ không sử dụng được.Để khắc phục điều ấy ta làm như sau:

- Khởi động Eclipse

- Vào Help -> About Eclipse -> Installations Details

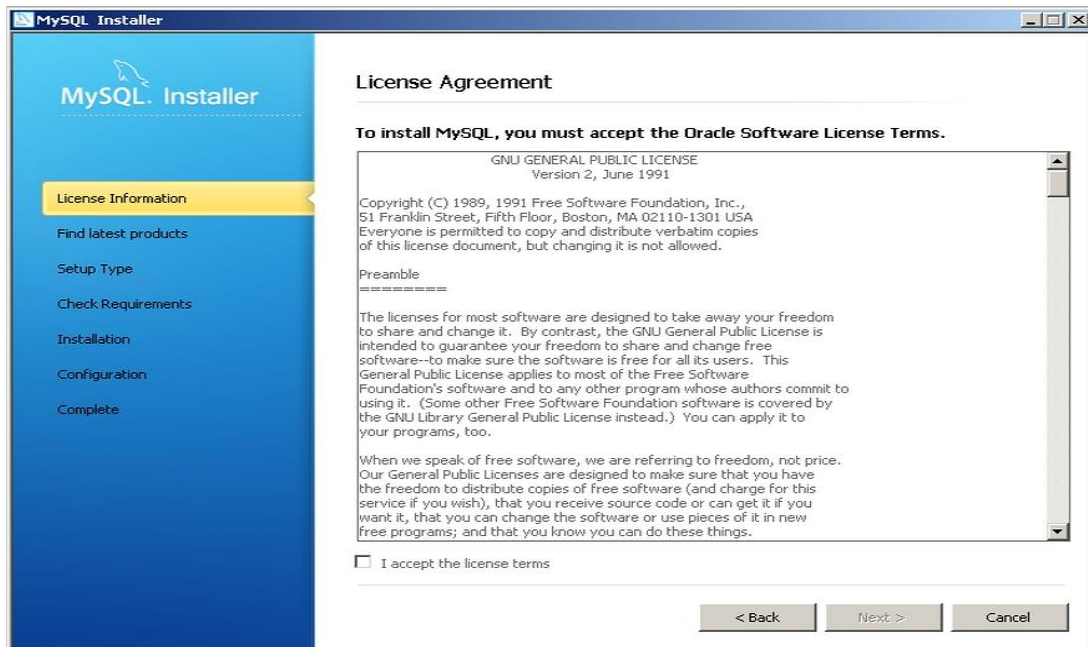
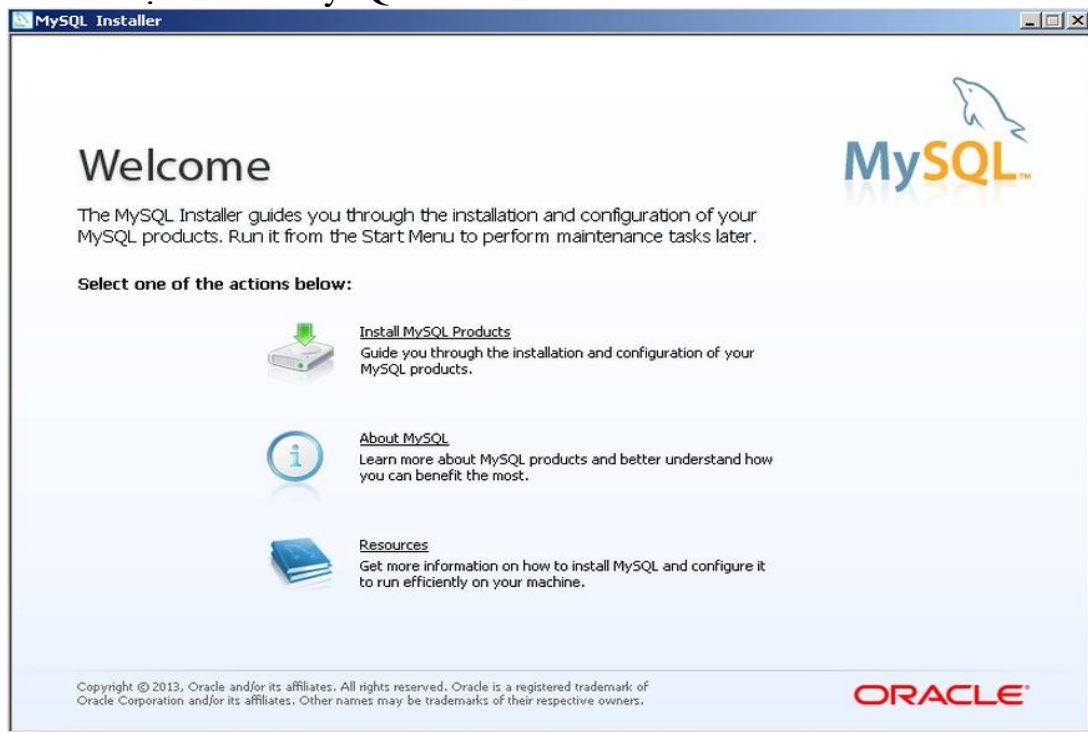


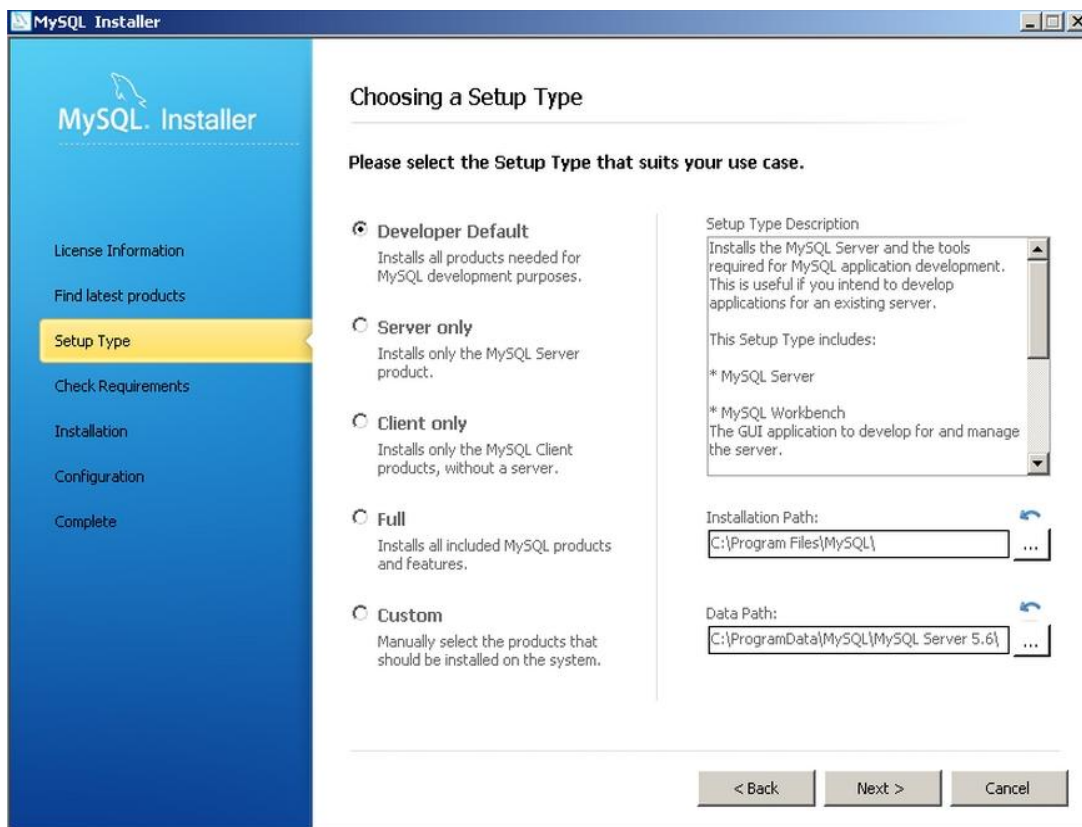
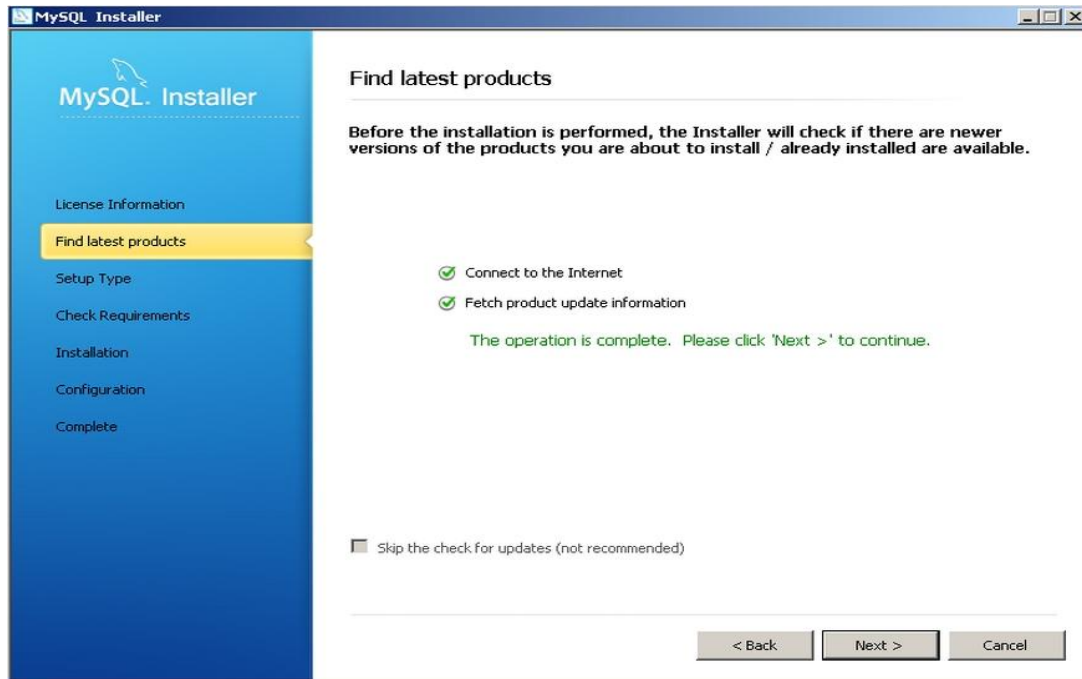
- Chọn plugin SQLI Tomcat Launch và Uninstall
- Bây giờ vào Window->Preference->Tomcat và chọn lại Version mới nhất là Tomcat 7.x và lựa chọn đường dẫn tới địa chỉ chứa Tomcat như hướng dẫn ban đầu. Cài đặt thành công bằng cách vào browser kiểm tra theo đường dẫn :<http://localhost:8080/> và kết quả như sau:



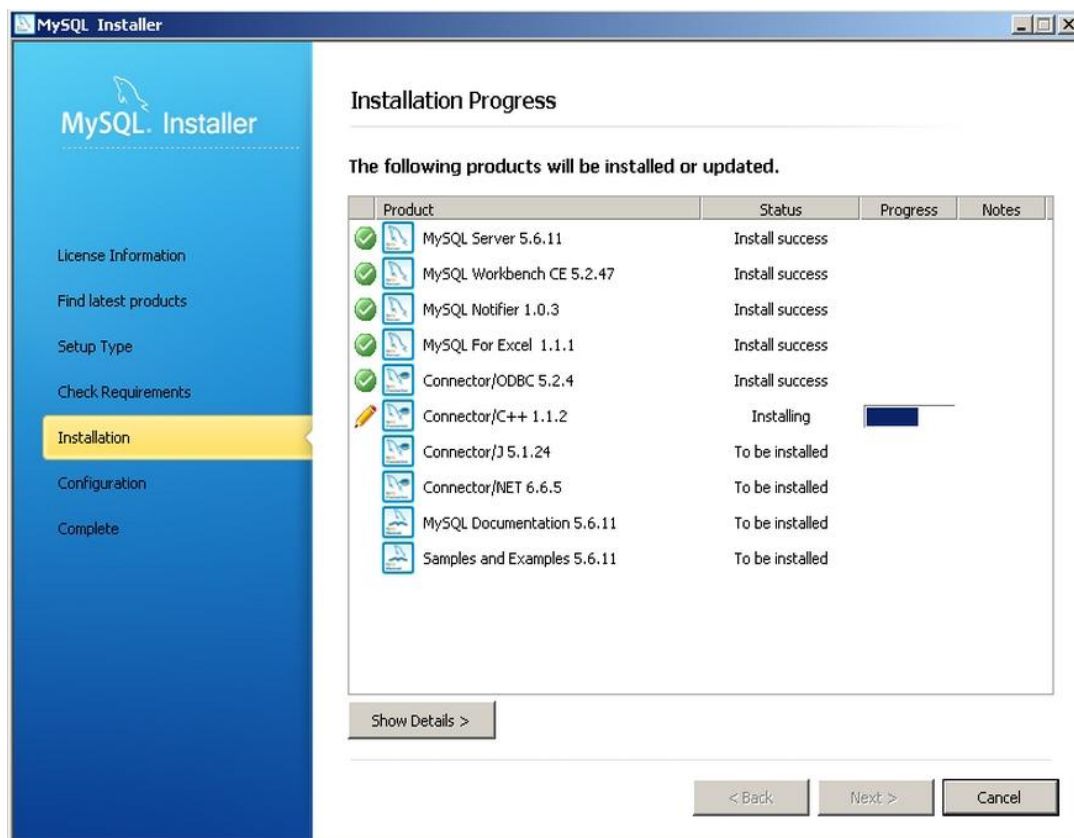
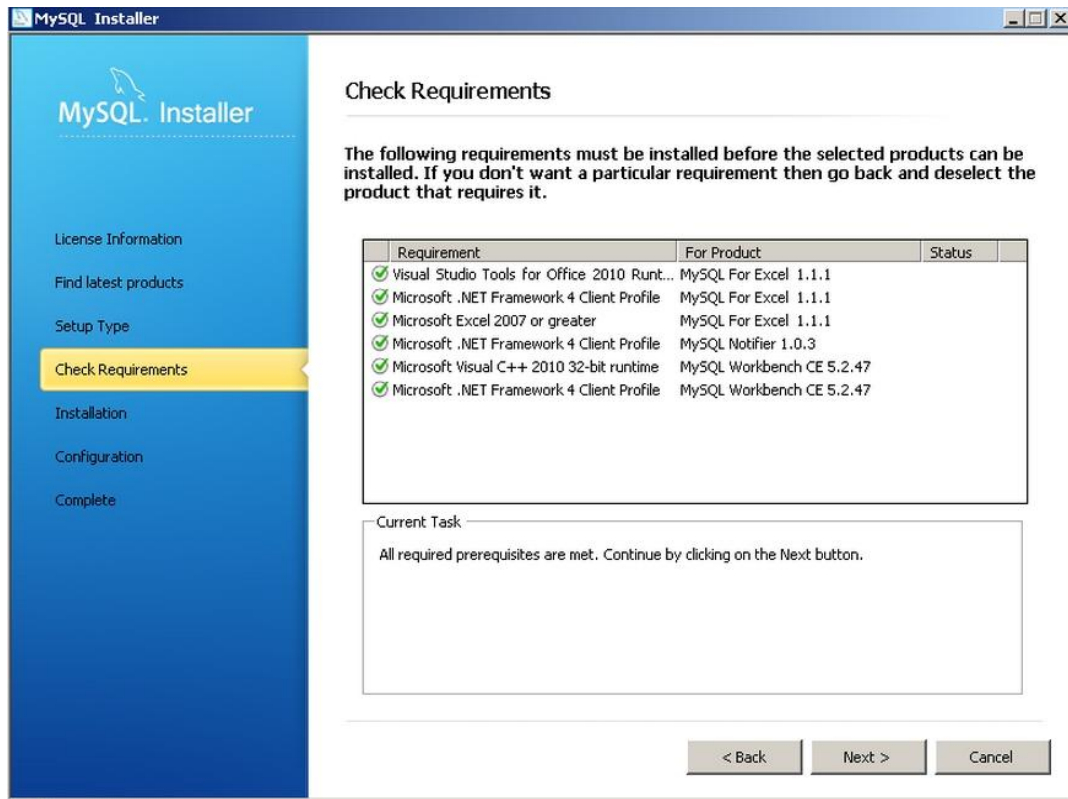
### 3.6. Cài đặt MySQL và kết nối với Eclipse

- Download Mysql theo đường dẫn:  
<http://dev.mysql.com/downloads/installer/>
- Cài đặt Mysql
  - ✓ Run file mysql-installer-community-5.6.12.1.msi
  - ✓ Chọn Install MySQL Products

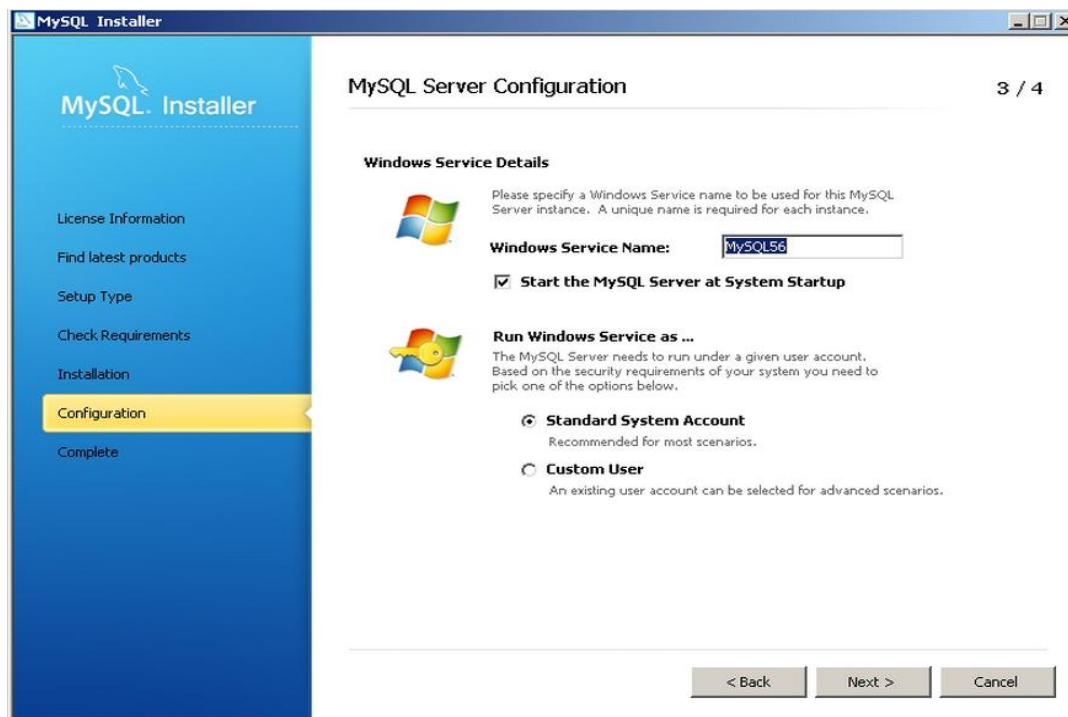
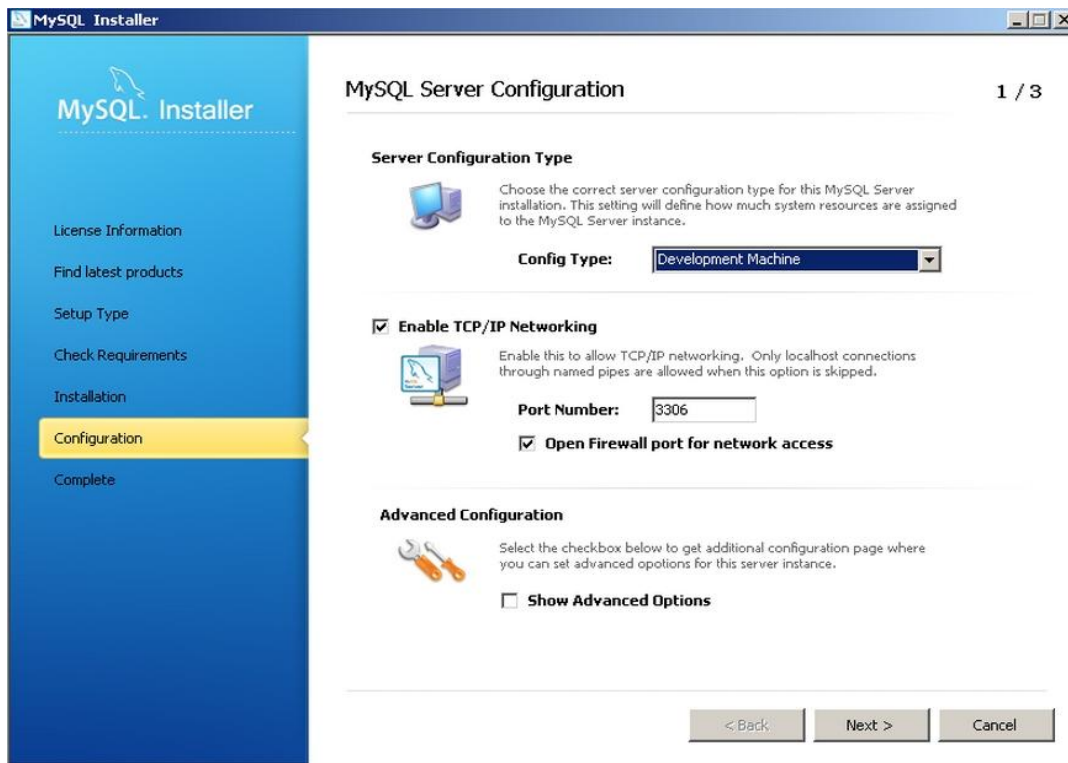


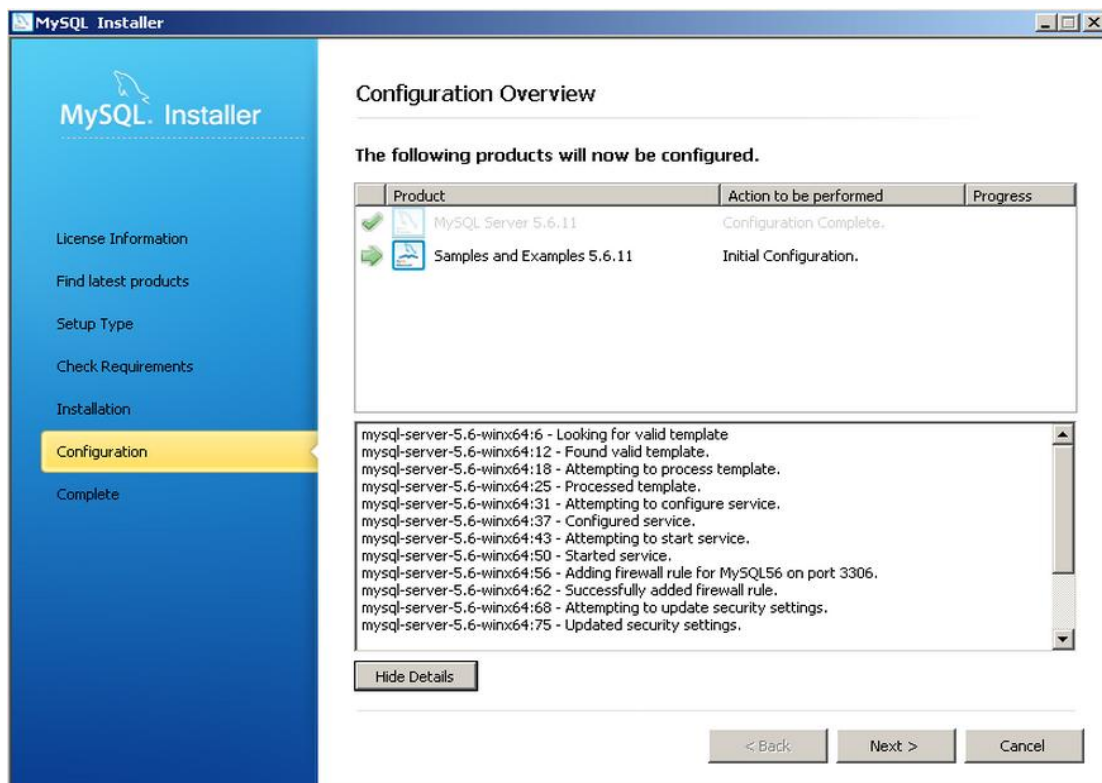
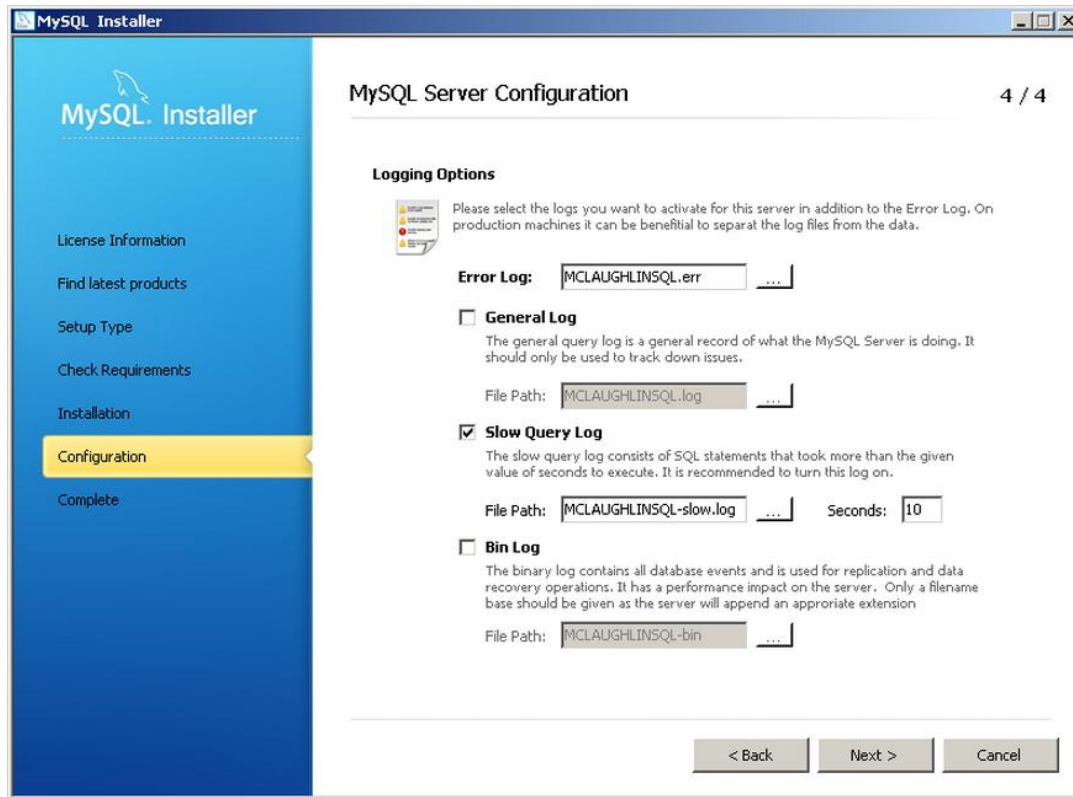






## ✓ Configuration Steps





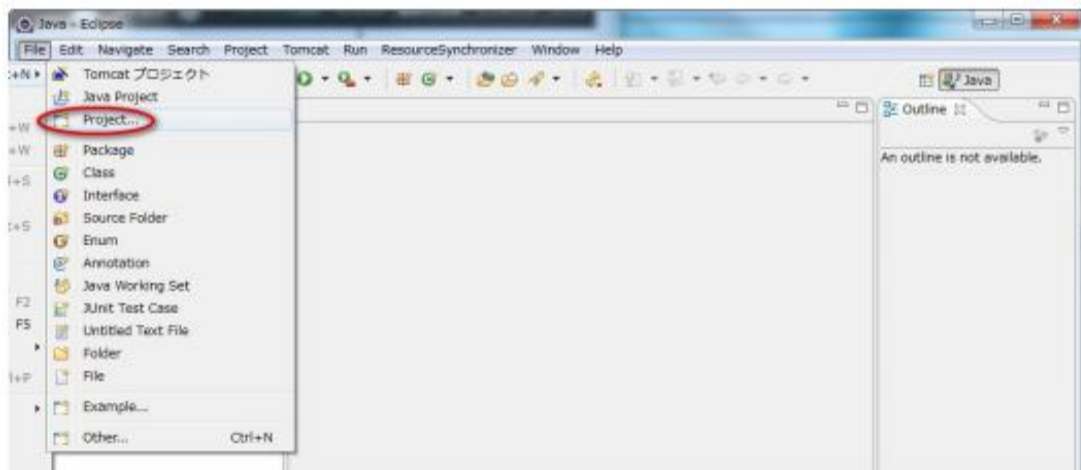


## 3.7. Xây dựng Project Seasar2

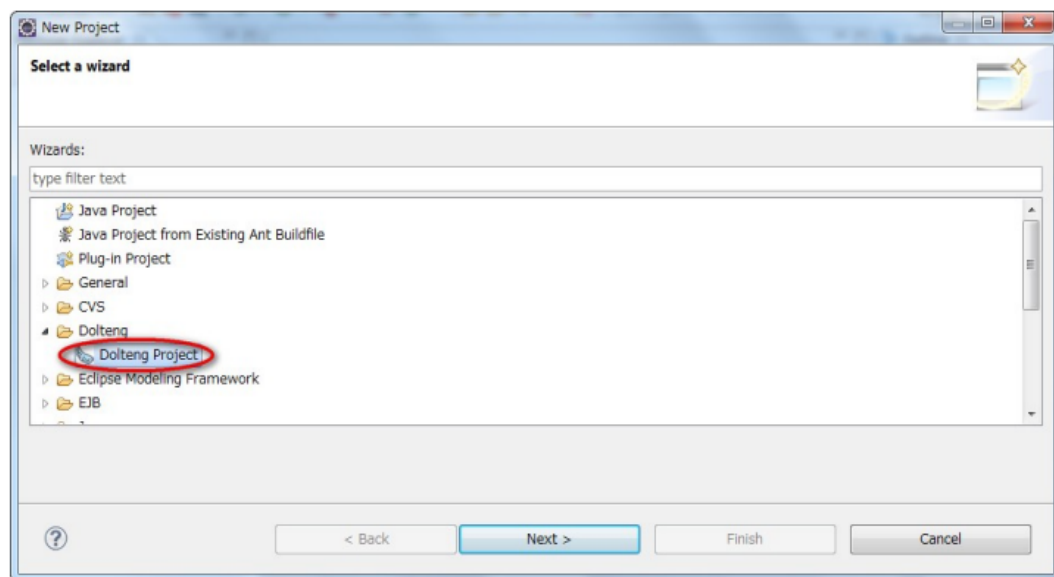
### 3.7.1. Tạo Project Seasar2

Để tạo một Project sử dụng framework Seasar2 ta thực hiện các bước như sau:

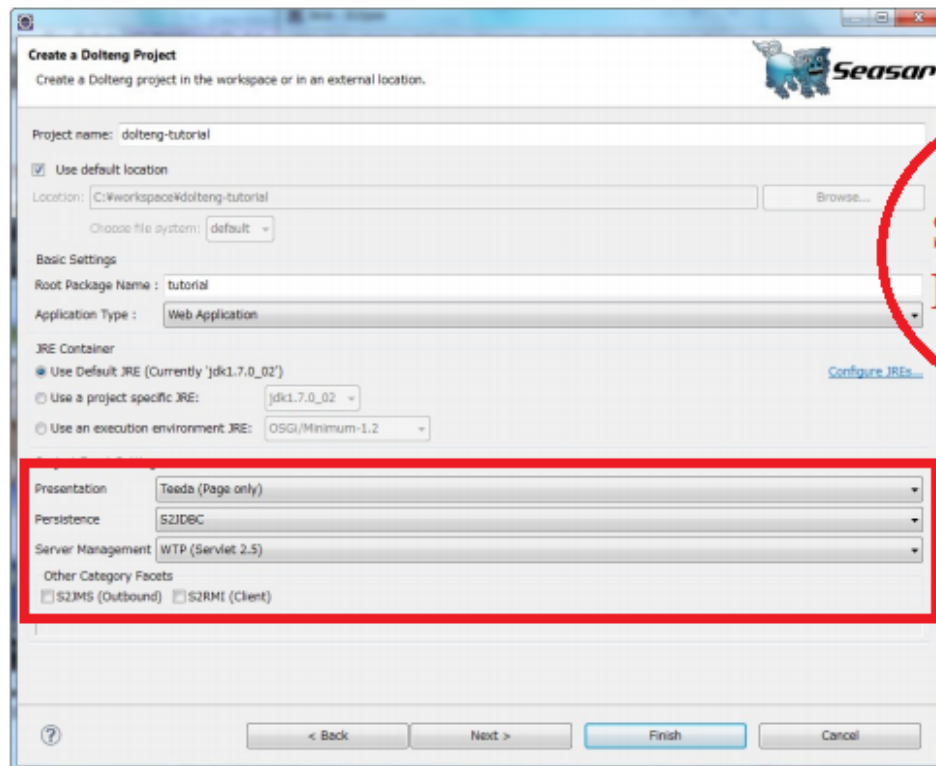
- Khởi động Eclipse, chọn New Project



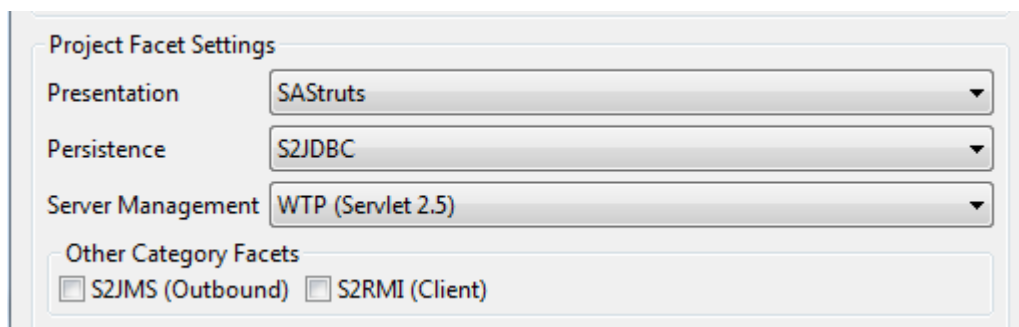
- Để tạo 1 Project Seasar2 thì Eclipse đã sử dụng một Plugin của Seasar2 là Dolteng, ta sẽ tạo một Dolteng Project



- Sau khi lựa chọn Creat Dolteng Project



- Phần lựa chọn khởi tạo Project này rất quan trọng, nếu không lựa chọn đúng sẽ không thể có được một Project có thể sử dụng được Seasar2 và SAStruts. Ta sẽ set up như sau:



- Chọn Next để tạo 1 project hoàn chỉnh.

### 3.7.2. Quản lý Database trong Eclipse

- Sử dụng DBEdit để quản lý Database từ Eclipse

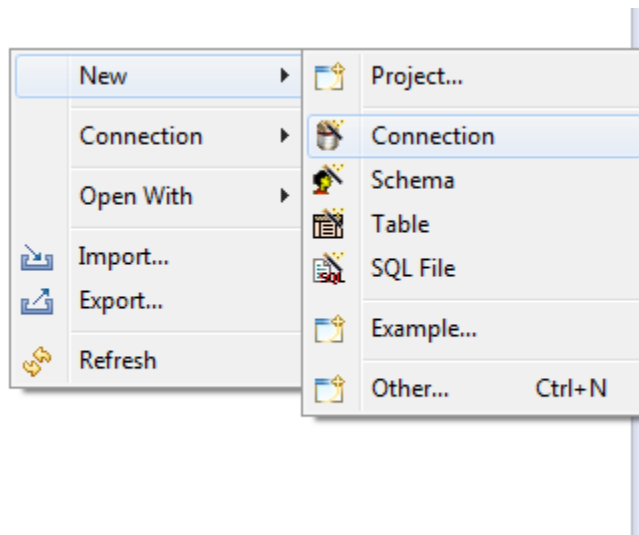
Để sử dụng DBEdit, ta vào Open Perspective và chọn DBEdit để thêm vào thanh công cụ của Eclipse



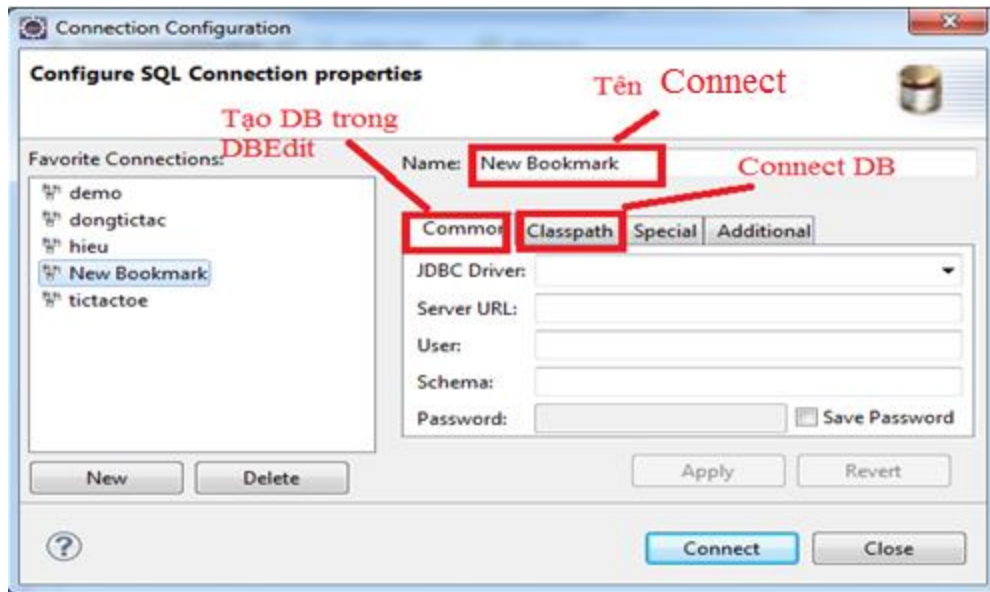
- Muốn sử dụng DBEdit ta cần sử dụng plugin connector của MySQL cho Eclipse. Ta có thể down nó tại địa chỉ sau:

“<http://dev.mysql.com/downloads/connector/j/>” . Tại đây down file .zip về và giải nén sẽ thấy file “mysql-connector-java-5.1.18-bin.jar” – đây chính là file connect database.

- Để quản lý Database đã được tạo trong MySQL bằng DBEdit ta click chuột phải và khởi tạo New Connector

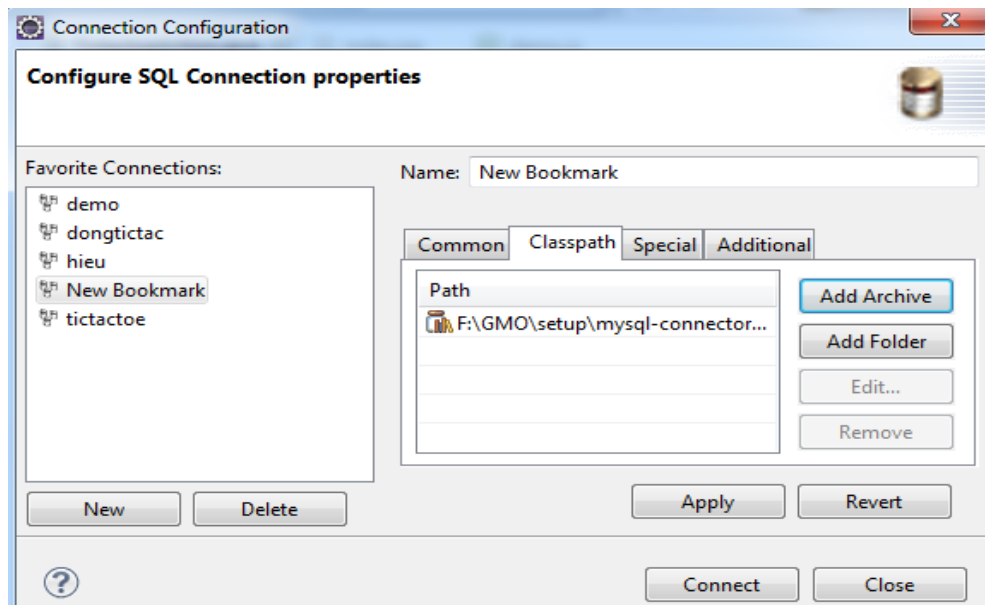


- Để đồng bộ database trong MySQL với DBEdit ta cần làm 3 bước sau:

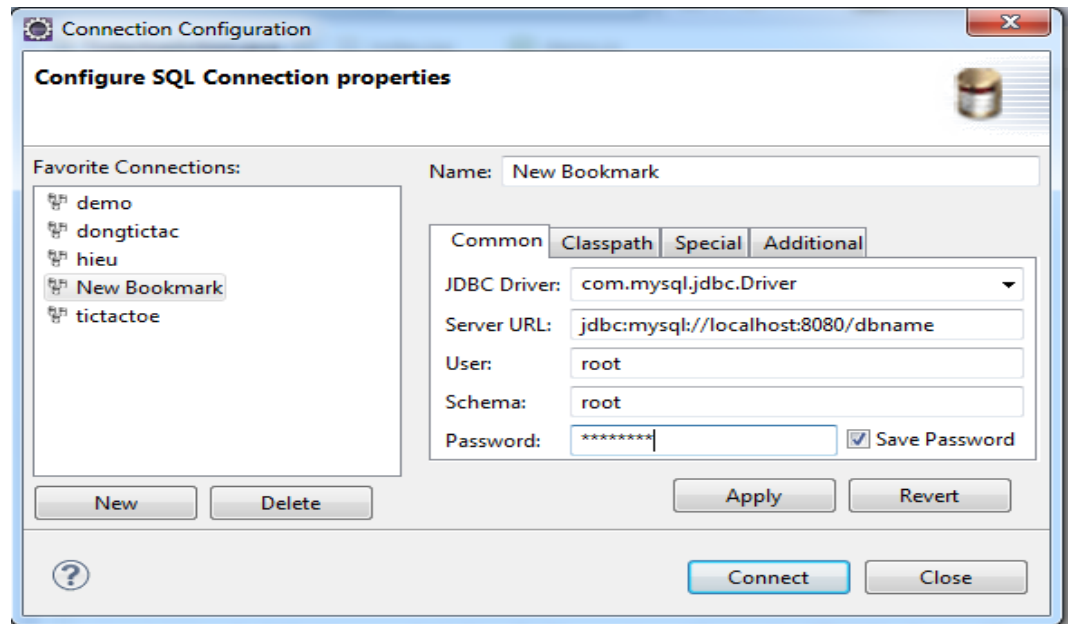


✓ Bước 1: Connect BD

Trong mục Classpath ta chọn Add Archive và chọn đường dẫn tới file connect MySQL vừa down ở trên và chọn Apply để connect.



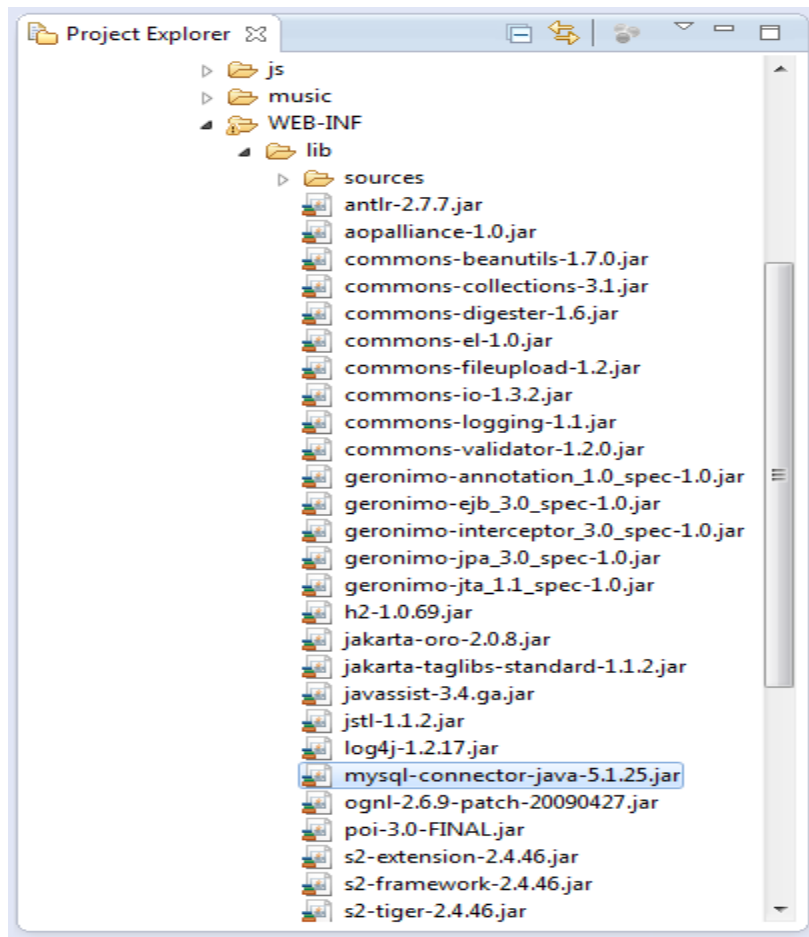
✓ Bước 2: Sau khi thiết lập connect ta vào mục common để khởi tạo database trong DBEdit, lựa chọn và thiết lập các thông số theo hình sau:



- Chọn Connect để tạo ra một Connector mới trong DBEdit, lúc này ta đã có thể quản lý Database trong Eclipse.

### 3.7.3. Config để sử dụng databaseMySQL trong project Seasar2

- Để sử dụng Database trong project Seasar2 thì ta cần config 2 file là s2jdbc.dicon và jdbc.dicon
- Download file mysql-connector-java-5.1.25-bin.jar từ URL sau đây : <http://dev.mysql.com/downloads/connector/j/>  
Sau đó Add vào thư mục WEB-INF/lib



- Khởi động Eclipse và tạo project sử dụng framework Seasar2 với plugin Dolteng. Trong thư mục src/main/resources ta config 2 file
  - ✓ **S2JDBC:** Chọn property “mysqlDialect” để sử dụng

```

s2jdbc.dicon
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
    "http://www.seasar.org/dtd/components24.dtd">
<components>
  <include path="jdbc.dicon"/>
  <include path="s2jdbc-internal.dicon"/>
  <component name="jdbcManager" class="org.seasar.extension.jdbc.manager">
    <property name="maxRows">0</property>
    <property name="fetchSize">0</property>
    <property name="queryTimeout">0</property>
    <property name="dialect">mssqlDialect</property>
    <property name="dialect">h2Dialect</property>
    <property name="dialect">db2390Dialect</property>
    <property name="dialect">db2400Dialect</property>
    <property name="dialect">db2Dialect</property>
    <property name="dialect">derbyDialect</property>
    <property name="dialect">firebirdDialect</property>
  </component>
</components>

```

Với những hệ cơ sở dữ liệu khác cũng tương tự, nếu ta cần dùng hệ cơ sở dữ liệu nào thì trong file S2jdbc.dicon ta cũng lựa chọn hệ cơ sở dữ liệu đó.

### ✓ JDBC

Lựa chọn phần tương ứng với hệ cơ sở quản trị dữ liệu sử dụng, ở đây ta dùng MySQL nên ta sẽ config với mysql như sau:



Trong file này các hệ quản trị cơ sở dữ liệu khác sẽ được để ở dạng chú thích. Trong phần property với name="URL" là nơi định nghĩa tên CSDL mà ta sử dụng, name="user" là nơi định nghĩa người dùng, ta để mặc định là root như trong MySQL đã khai báo (MySQL ta sử dụng quyền root thì khai báo root, nếu ta dùng quyền user thì phải khai báo với tên user đã khai báo trong MySQL), name="password" là nơi định nghĩa password của DB ta đã khai báo trong MySQL.

## 3.8. Cài đặt Maven và tạo file war cho project Seasar2

### 3.8.1. Maven là gì?

Hiểu một cách đơn giản maven là một công cụ build phổ biến trong thế giới lập trình Java. Theo định nghĩa chính thức Apache Maven (<http://maven.apache.org/>), maven là công cụ toàn diện về quản lý dự án phần mềm. Dựa trên mô hình đối

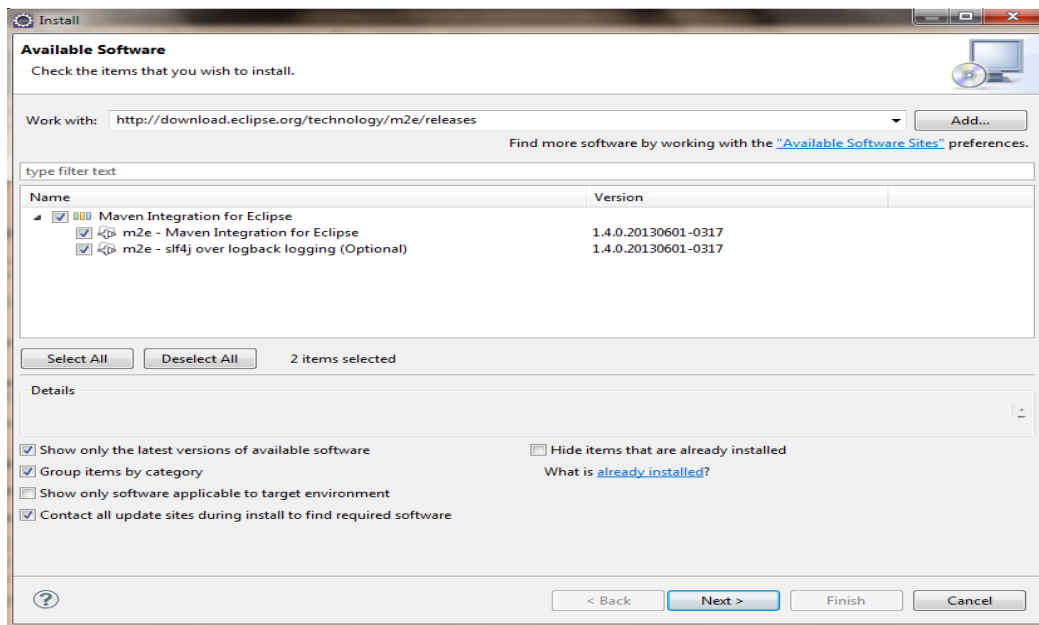
tượng dự án (project object model – POM) maven có thể quản lý build, tạo báo cáo và tài liệu cho dự án từ thông tin lưu trữ tập trung (trong file pom)

### 3.8.2. Cài đặt Maven trong project Seasar2

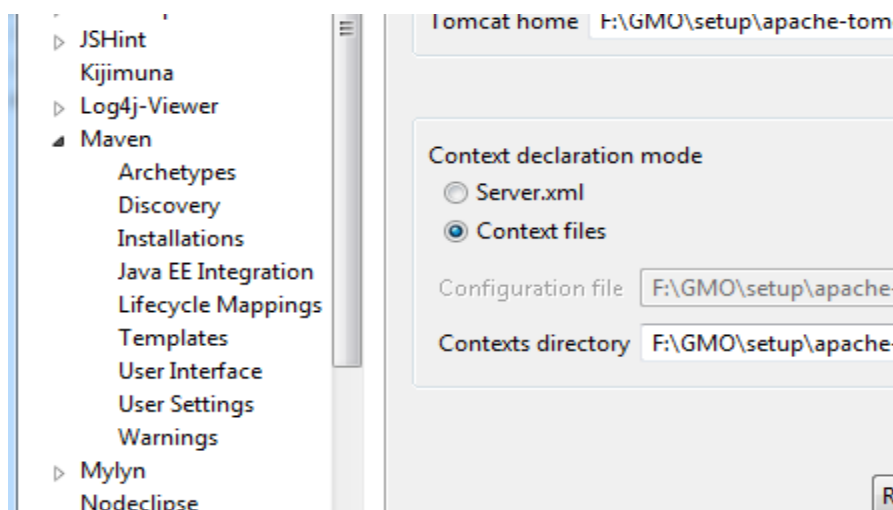
- Khởi động Eclipse, vào Help -> Install New Software

Để cài đặt Maven ta sử dụng plugin của Maven dành cho Eclipse. Ta làm như sau: Vào Help để Install Plugin cho Maven bằng URL:

“<http://download.eclipse.org/technology/m2e/releases/>”

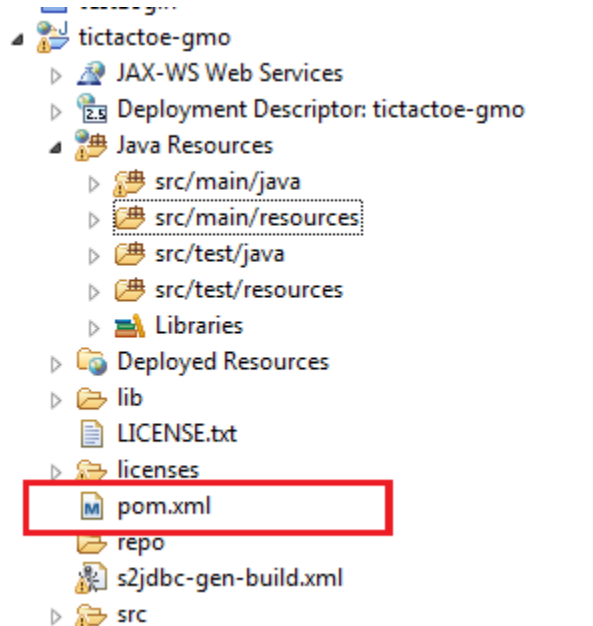


- Sau khi đã add plugin của maven xong ta khởi động lại Eclipse.

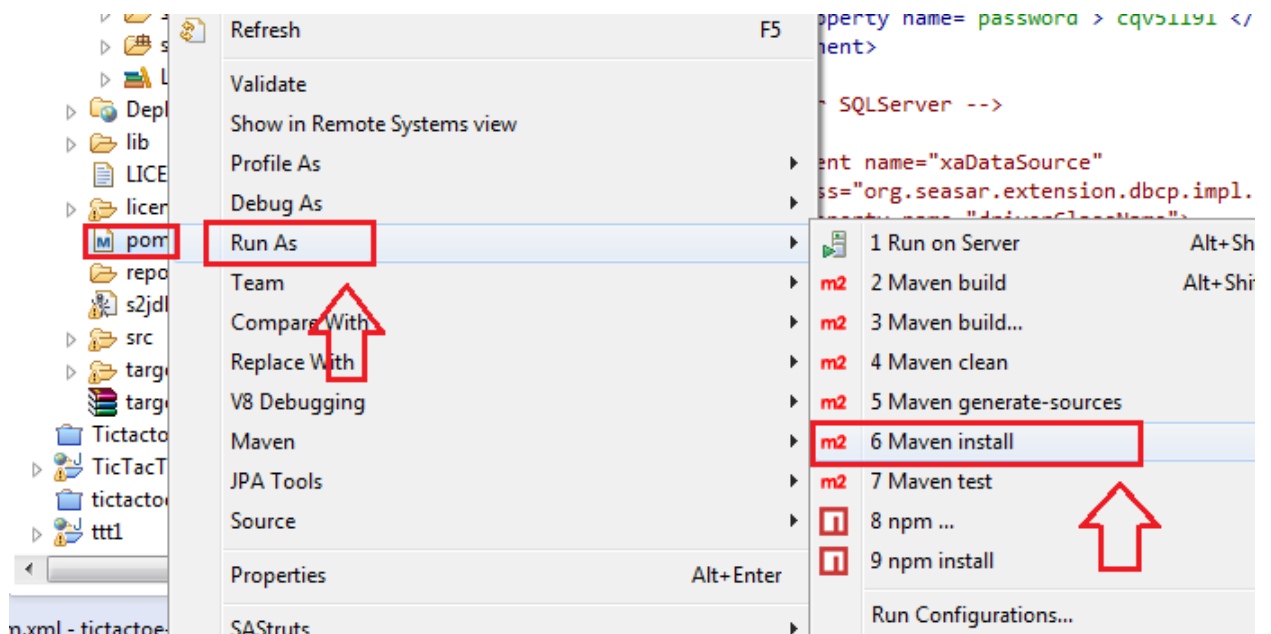




- Sau khi cài đặt Maven thì trong project Seasar2 sẽ tạo ra file pom, đây chính là file lưu trữ và quản lý thông tin tập trung của maven.



- Build Maven để tạo file war



## ➤ Kết quả

```

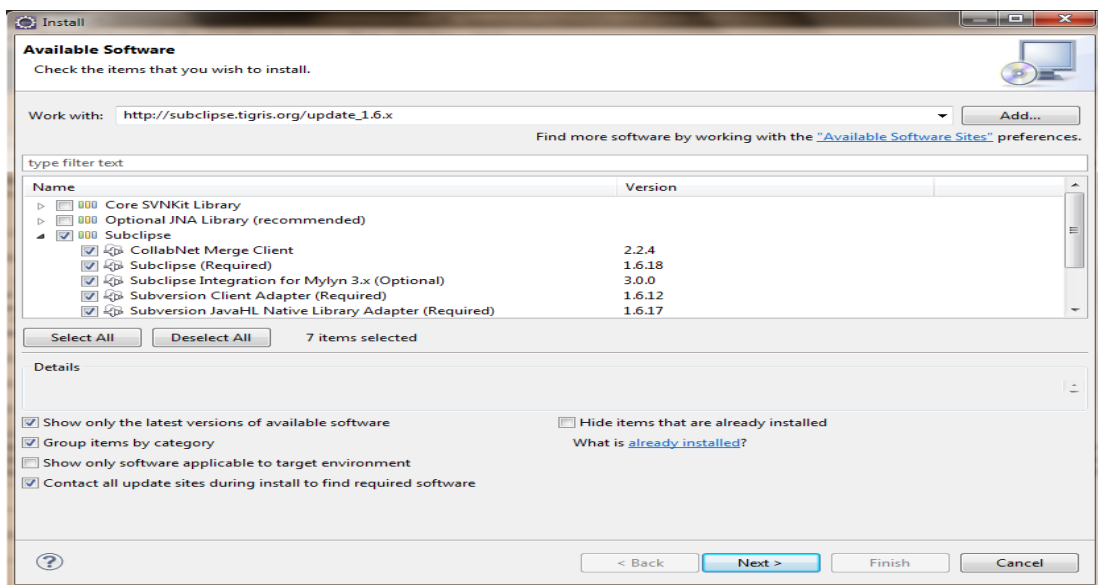
<terminated> C:\Program Files\Java\jre7\bin\javaw.exe (Jul 22, 2013 11:52:40 AM)
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ dat_tictactoe ---
[INFO] Packaging webapp
[INFO] Assembling webapp [dat_tictactoe] in [D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\target\dat_tictactoe-0.0.1]
[INFO] Processing war project
[INFO] Copying webapp resources [D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\src\main\webapp]
[INFO] Webapp assembled in [5024 msecs]
[INFO] Building war: D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\target\dat_tictactoe-0.0.1.war
[WARNING] Warning: selected war files include a WEB-INF/web.xml which will be ignored
(webxml attribute is missing from war task, or ignoreWebxml attribute is specified as 'true')
[INFO]
[INFO] >>> maven-source-plugin:2.2.1:jar (source-jar) @ dat_tictactoe >>>
[INFO]
[INFO] <<< maven-source-plugin:2.2.1:jar (source-jar) @ dat_tictactoe <<<
[INFO]
[INFO] --- maven-source-plugin:2.2.1:jar (source-jar) @ dat_tictactoe ---
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ dat_tictactoe ---
[INFO] Installing D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\target\dat_tictactoe-0.0.1.war to C:\Users\Administrator\.m2\repository\dat\dat_tictactoe\0.0.1\dat_tictactoe-0.0.1.war
[INFO] Installing D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\pom.xml to C:\Users\Administrator\.m2\repository\dat\dat_tictactoe\0.0.1\dat_tictactoe-0.0.1.pom
[INFO] Installing D:\Seasar\ eclipse_seasar\workspace\dat_tictactoe\target\dat_tictactoe-0.0.1-sources.jar to C:\Users\Administrator\.m2\repository\dat\dat_tictactoe\0.0.1\dat_tictactoe-0.0.1-sources.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 22.818s
[INFO] Finished at: Mon Jul 22 11:53:08 ICT 2013
[INFO] Final Memory: 7M/18M
[INFO]

```

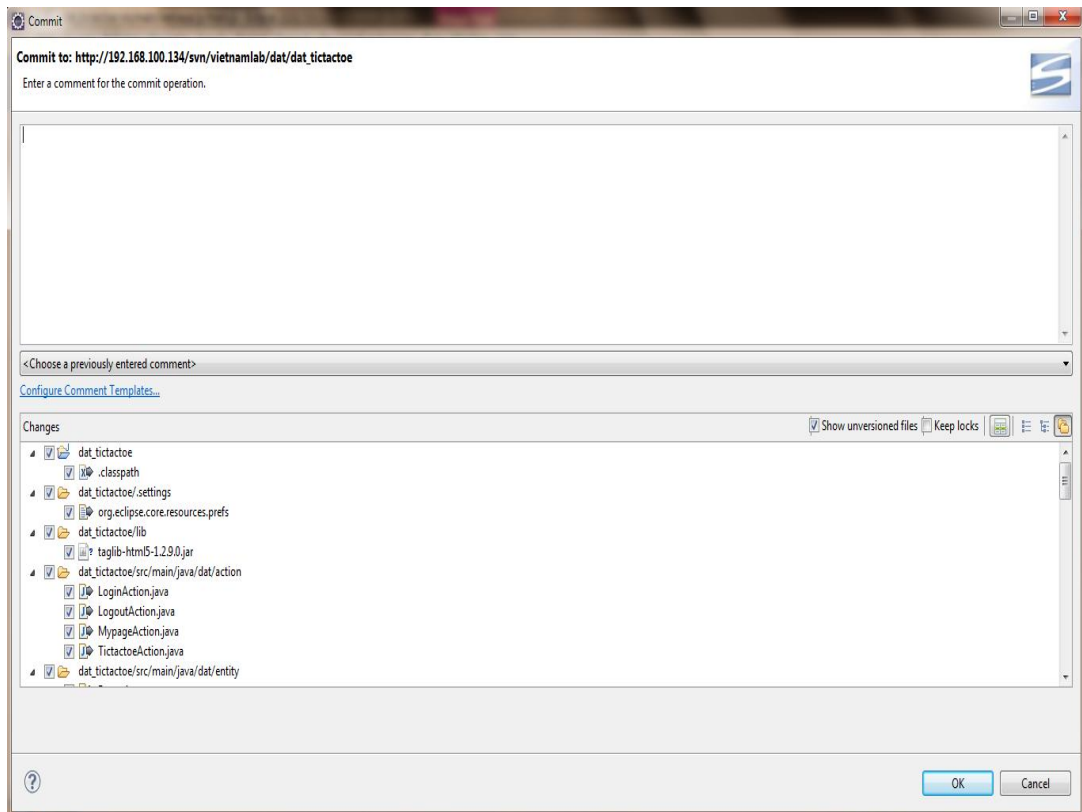
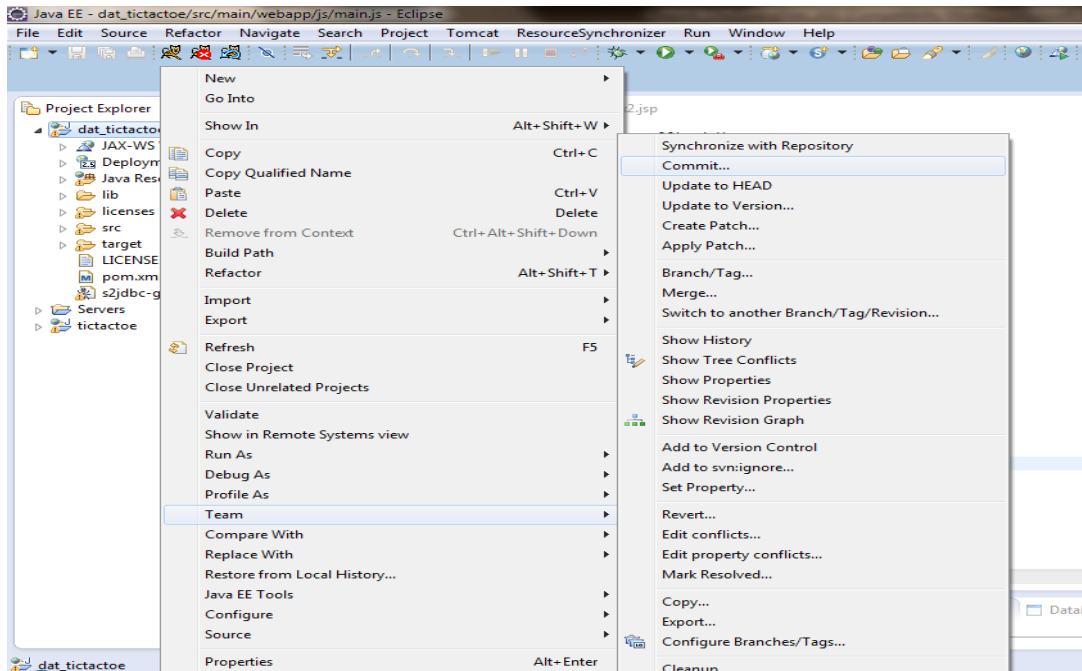
## 3.9. Cài đặt SVN plugin và commit source code

- Để cài đặt SVN cho Eclipse ta sử dụng một plugin của SVN cho Eclipse, đó là Subclipse, để sử dụng được subclipse ta cần add nó vào Eclipse thông qua đường dẫn sau : [http://subclipse.tigris.org/update\\_1.6.x](http://subclipse.tigris.org/update_1.6.x)



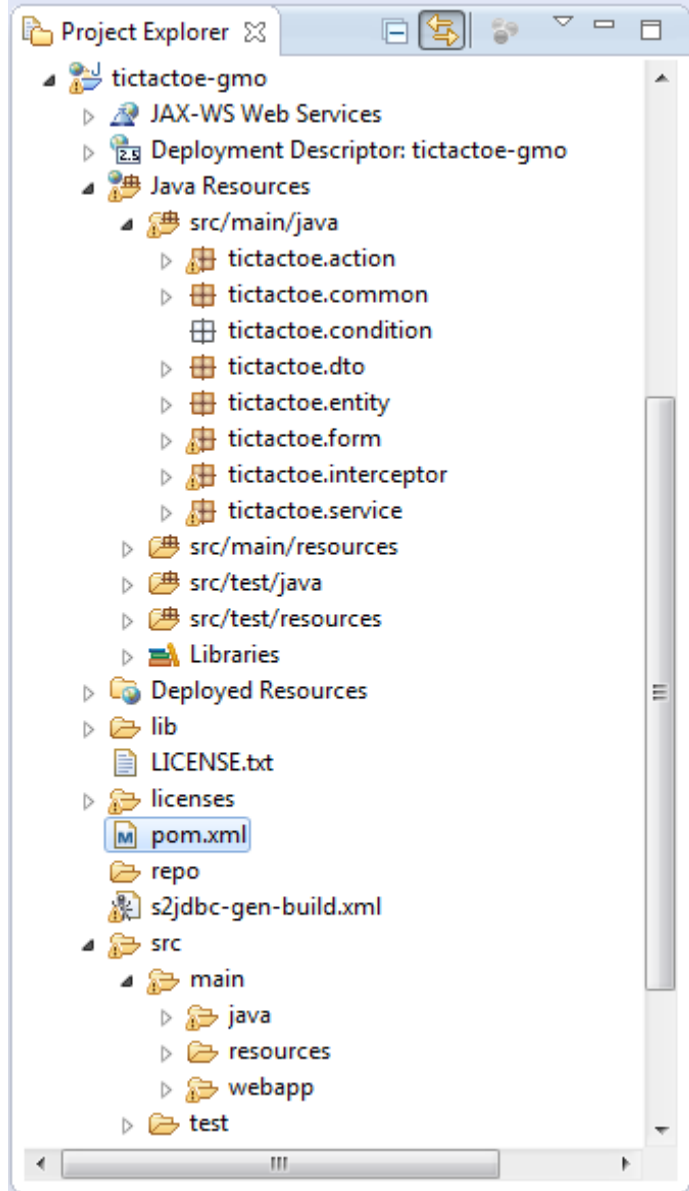
- Sau khi add xong ta cần khởi động lại Eclipse

- Sau khi cài đặt, để sử dụng ta làm theo hướng dẫn sau :



## II. Cấu trúc của Framework SAStruts

SAStruts là 1 framework có cấu trúc rất rõ ràng

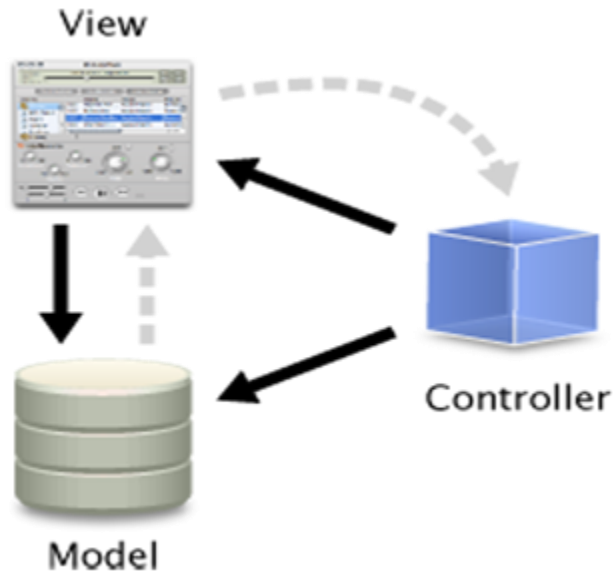


Trong đó :

- Src/main/java : thư mục chứa file java để xử lý
- Src/main/resources : thư mục chứa các file config như s2jdbc.dicon hay jdbc.dicon
- Src/main/webapp : thư mục chứa các file JSP(ngoài ra còn có javascrip,css,..) dùng để hiển thị trên brower.

Trong thư mục main/java lại chia thành các rootpackage riêng biệt như action , entity , dto, service,form,... Mỗi rootpackage lại có những tác dụng riêng biệt làm nên điểm mạnh của framework Seasar2.

Trước khi đi vào phân tích từng rootpackage trong framework này,ta mô tả nó một cách đơn giản như sau : **‘SAstruts là một framework có cấu trúc dựa theo mô hình MVC,trong đó Action là Controller,Entity là Model và JSP chính là View’**. Ta sẽ nhắc lại mô hình MVC là gì ?



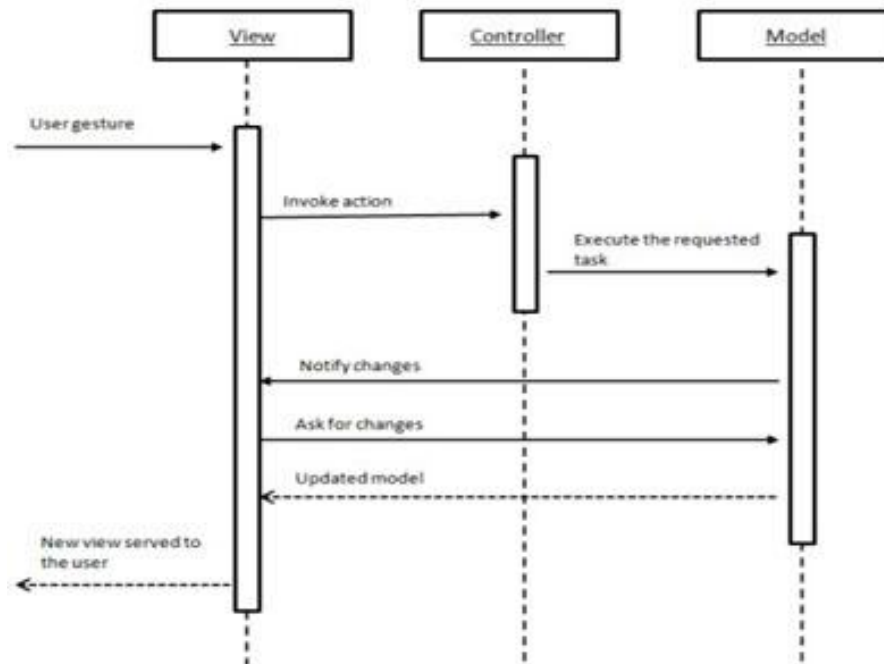
❖ Thành phần MVC ra sao ?

- Model : Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...
- View : Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images...Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.
- Controller : Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.

❖ MVC làm việc như thế nào?

- Nhìn lại sơ đồ phía trên, ta thấy có mũi tên nét liền và những mũi tên nét đứt. Những mũi tên nét đứt được hình thành trên quan điểm của người dùng mà không phải là của những nhà thiết kế phần mềm thực sự. Do đó chúng ta chỉ quan tâm đến những mũi tên còn lại.
- Đây là một cách đơn giản để mô tả lại luồng sự kiện được xử lý trong MVC:
  - ✓ User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi.(submit)

- ✓ Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model.
- ✓ Model nhận thông tin và thực thi các yêu cầu.
- ✓ Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.



Với cách thức làm việc của mô hình MVC thì ta cũng đã khái quát được cách thức làm việc của SAstruts :

#### ❖ Action

➤ Quan hệ giữa Action class và URL :

RootPackage này chứa các file Action. File Action sẽ được khởi động khi có request. Trong Struts thì mọi quan hệ của Action và URL đều được mô tả trong file « struts-config.xml ». Tuy nhiên SAstruts đã khắc phục điều ấy và tự động gọi tới lớp Action khi có request từ URL.

Phân tích 1 ví dụ sau :

Truy cập tới <http://localhost:8080/sa-struts-tutorial/login/>

Trong đó sa-struts-tutorial là tên WebProject. SAstruts sẽ thực hiện việc chuyển đổi từ URL sang Action theo các bước sau :

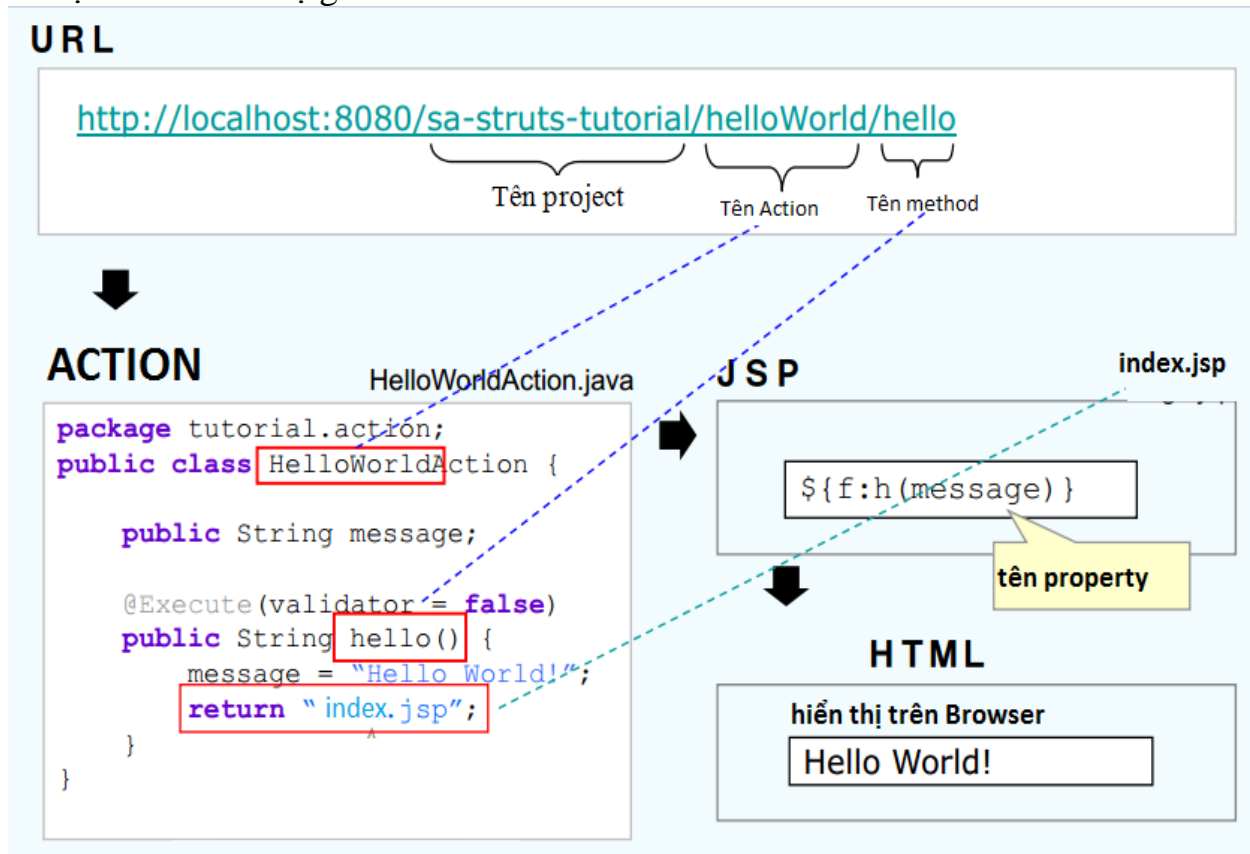
- ✓ Chuyển số gạch(/) cuối cùng của đường dẫn sau tên project thành Action(/login/ thành /loginAction). Trong trường hợp không có số gạch thì thêm Action vào sau.

- ✓ Chuyển ký tự ngay sau dấu gạch(/)thành chữ HOA(/loginAction thành /LoginAction)
- ✓ Chuyển dấu gạch(/LoginAction) thành dấu chấm (.LoginAction)
- ✓ Gắn nó vào ngay đầu của  
rootpackage.action(rootpackage.action.LoginAction).Trong đó tên của rootpackage được ta đặt khi khởi tạo project.Ở ví dụ này ta đặt là tutorial.
- ✓ Cuối cùng tương ứng với /login/ chính là tutorial.action.LoginAction

Trong trường hợp không tồn tại đường dẫn đi sau tên project mà có rootpackage.action.IndexAction thì sẽ tự động gọi đến nó.( Ví dụ khi truy cập vào <http://localhost:8080/sa-struts-tutorial/> thì sẽ được gọi đến tutorial.action.IndexAction).

Như vậy ta có thể dễ dàng thấy giữa class Action và url có sự liên quan chặt chẽ với nhau.

Ví dụ về mối liên hệ giữa Action và URL



#### ➤ Chức năng của Action class

Action class là nơi đáp ứng các yêu cầu từ web page, được thực hiện để xử lý yêu cầu gửi lên từ phía các file jsp bằng cách sử dụng các thành phần tài nguyên lấy từ Form, Entity, Dto, Service, .... Nó sẽ xử lý tính toán để trả về kết quả chứa nội dung

cần hiển thị và đưa kết quả lên trang jsp để hiển thị. Khi sử dụng Action cần chú ý cách đặt tên sao cho đồng bộ với file JSP (như phân tích phần trên).

➤ Phương thức thực hiện trong Action :

Trong Action class thì phương thức thực hiện dùng để quản lý các request. Tên của phương thức được đặt tự do, gắn kèm với chú thích @Execute, giá trị trả về là String và không có tham số truyền vào

```
@Execute
public String xxx() {
    ...
    return ...;
}
```

Trong phương thức thực hiện của class Action thì giá trị trả về là một đường dẫn. Đường dẫn sẽ dẫn tới folder trong view/xxx cùng tên với XxxAction, trong folder này chứa file JSP tương ứng với tên file JSP trong phương thức khai báo. Nếu file JSP chưa tồn tại thì sẽ tự động sinh ra. (Trong Eclipse chỉ cần để chuột vào đúng vị trí của đường dẫn và Ctrl 8 thì file JSP sẽ tự động sinh ra). Ngoài ra nếu muốn đường dẫn quay trở lại JSP đã tồn tại thì ta sử dụng "?redirect=true" đặt ngay sau vị trí nét sổ của đường dẫn (ví dụ : return "/login/?redirect=true"). Khi thực hiện trả về đường dẫn thì khi thực hiện phương thức đó, mọi kết quả sẽ được gửi tới trang JSP tương ứng trong đường dẫn nếu ở trang JSP có lời gọi giá trị đó thông qua \${f:h()}.

Class Action sẽ nhận được request thông qua submit hoặc được các form truyền tới. Và sau khi tính toán trả về kết quả thì các kết quả sẽ được gửi lên JSP thông qua \${f:h()}.

Xét ví dụ đơn giản sau :

✓ class AddAction



```

package tutorial.action;

import org.apache.log4j.BasicConfigurator;

public class AddAction {

    static Logger logger = Logger.getLogger(AddAction.class);

    @Required
    @IntegerType
    public String arg1;

    @Required
    @IntegerType
    public String arg2;

    public Integer result;

    @Execute(validator = false)
    public String index() {
        BasicConfigurator.configure();
        logger.error("Sample error with log4j");
        return "index.jsp";
    }

    @Execute(input = "index.jsp")
    public String submit() {
        result = Integer.valueOf(arg1) + Integer.valueOf(arg2);
        return "index.jsp";
    }
}

```

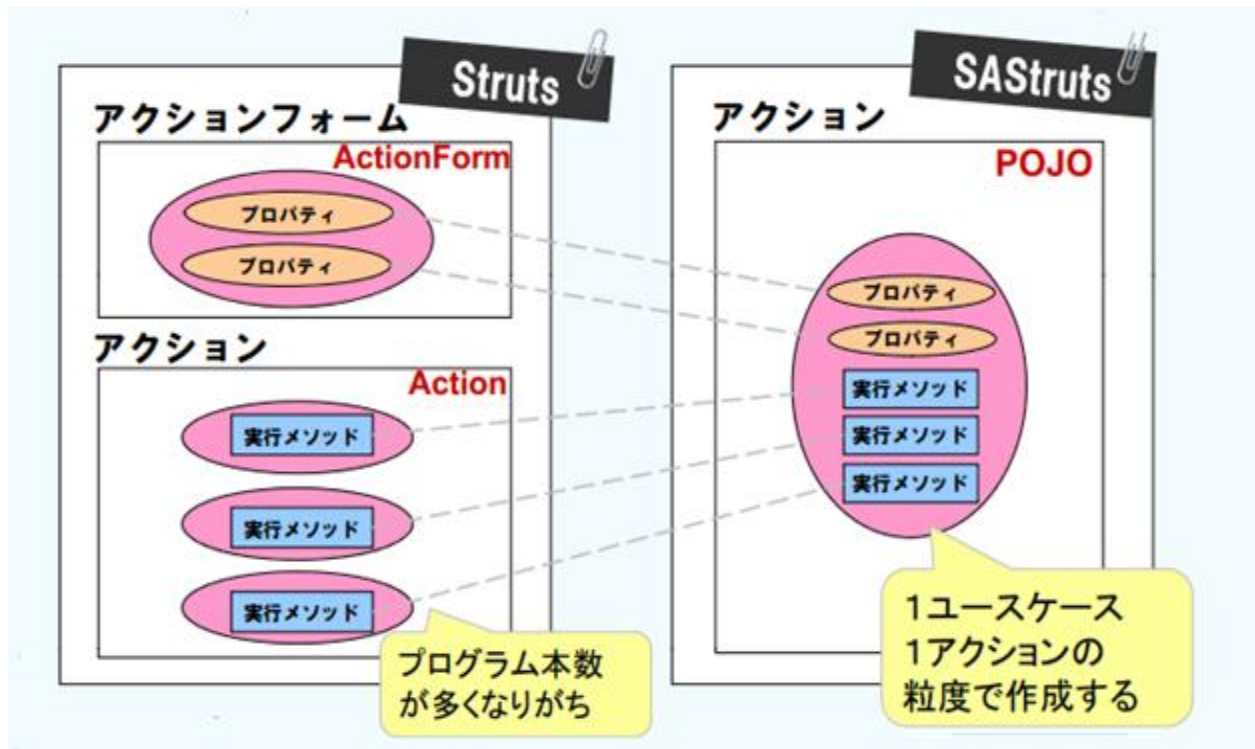
✓ index.jsp

```

<html>
<head>
<title>Add</title>
</head>
<body>
<html:errors/>
<s:form>
<html:text property="arg1"/> + <html:text property="arg2"/>
= ${f:h(result)}<br />
<input type="submit" name="submit" value="Submit This!"/>
</s:form>
</body>
</html>

```

Ta thấy 2 input đầu vào định nghĩa trong thẻ `<html :text>` với thuộc tính `property`. Khi ấy form sẽ gửi giá trị lên để Action xử lý. Theo Struts thì đầu vào input sẽ được định nghĩa trong `ActionForm` nhưng `SAStruts` đã thay đổi bằng cách dùng class Action như một POJO (Plain Old Java Object) và định nghĩa đồng thời cả `ActionForm` và `Action` trong cùng 1 class `Action`. Tuy nhiên để rõ ràng thì ta vẫn nên phân biệt rõ ràng `ActionForm` và `Action` để tiện cho việc debug sau này.



Input sẽ được truyền đến ActionForm, như vậy giá trị của arg1 và arg2 chính là 2 giá trị được gửi đến. Khi JSP thực hiện submit thì tại class Action, method submit sẽ được gọi và thực hiện việc tính tổng của 2 giá trị nhập vào. Kết quả result được gửi lên và được JSP nhận thông qua việc thực hiện  $\$(f:h(result))$  ;

Sau đây là kết quả :

The image shows two screenshots of a web browser at the URL `localhost:8080/dolteng-tutorial/add/`. The browser's file explorer shows the directory structure: `Seasar2-SAsruts`, `HTML5`, `Japanese`, and `WebDesign`. The form consists of two input fields separated by a plus sign and an equals sign, followed by a 'Submit This!' button. In the left screenshot, the input fields are empty. In the right screenshot, the first input field contains the value '4' and the second input field contains the value '5', and the result '9' is displayed next to the equals sign.

### ❖ ActionForm :

Chứa các lớp để lấy các thông tin từ các input trong web page. Các biến trong ActionForm khi khai báo sẽ được đi kèm với các chú thích như `@Required`, `@Minlength`, `@Maxlength` tùy theo các điều kiện đặt ra cho đầu vào. Trong lớp Action, nếu muốn sử dụng các lớp ActionForm thì cần phải có khai báo là

@Resource và để nó lấy được giá trị thì cần thêm chú thích @ActionForm(Nếu không có khai báo này thì Action sẽ không sử dụng được các giá trị trong input truyền tới Form). Các giá trị của ActionForm đó sẽ được sử dụng trong hàm có chú thích @Execute của class Action.Chú ý khi đặt tên cho các đối tượng ActionForm cần tuân theo quy tắc XxxForm là tên ActionForm thì tên của đối tượng ActionForm là xxxForm.

```
@ActionForm
@Resource
protected AddForm addForm;
```

Ví dụ :

✓ UserForm : chứa 2 đầu vào là username và password

```
package tictactoe.form;

import org.seasar.struts.annotation.EmailType;

public class UserForm {
    /*
     * Form lưu trữ các biến được dùng trong Action, được dùng trong file .jsp
     */

    @Required
    @Maxlength(maxlength= 40)
    @Minlength(minlength= 5)
    public String userName="";

    @Required
    @Maxlength(maxlength = 20)
    @Minlength(minlength = 6)
    public String password="";
}
```

✓ LoginAction : sử dụng UserForm

```
public class LoginAction {

    @Resource
    @ActionForm
    public UserForm userForm;

    public HttpServletRequest request;

    @Resource
    public UserService userService;
```

❖ Entity :

Chứa các lớp tương đồng với các bảng trong DB, mỗi lớp Service sẽ tương ứng với mỗi bảng trong DB, mỗi biến trong một lớp Service là một thực thể trong cơ sở dữ liệu, mỗi lớp cần chú thích @Entity và @Table để đánh dấu lớp Entity. Các thuộc

tính của lớp này cần phải có tên giống như tên cột trong DB hoặc có chú thích @comment(neme="tên cột) trên tên biến. Tất cả các biến phải được để public. Khi xây dựng lớp như vậy sẽ rất tiện ích khi thư viện sẽ tự hiểu và thực hiện truy vấn để trả về đúng lớp Entity

Ví dụ : lớp UserEntity lưu trữ các trường trong table « user »



```

UserEntity.java
package tictactoe.entity;

import javax.persistence.Column;

@Entity
@Table(name = "user")
public class UserEntity {
    /*
     * Entity lưu trữ các trường trong DB
     */

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Integer id user;

    @Column(name = "email_user")
    public String email user;

    @Column(name = "name_user")
    public String name user;

    @Column(name = "password")
    public String password;

    @Column(name="state")
    public String state;

    public String toString() {
        return name user;
    }
}

```

#### ❖ Service :

Đây là nơi thực hiện các truy vấn cơ sở dữ liệu, trả về các dữ liệu Entity khi thực hiện Select, hay trả về int khi Update.

Ví dụ : lớp UserService là nơi thực thi các truy vấn CSDL với UserEnttiy

```

UserService.java
package tictactoe.service;

import java.util.List;

public class UserService extends AbstractService<UserEntity> {

    Logger logger = Logger.getLogger(UserService.class);

    public UserEntity findByUserNamePw(String userName, String password){
        return select().where("name_user = '"+ userName + "' and password = '"+ password + "'").getSingleResult();
    }

    public List<UserEntity> findAllByUserName() {
        return select().getResultList();
    }

    public int insertUser(UserEntity userEntity){
        return insert(userEntity);
    }

    public List<UserEntity> findByOnline(){
        return select().where("state = 'online']").getResultList();
    }

    public List<UserEntity> findByOnlineNotMeUsingName(String userName){
        return select().where("name_user != '"+ userName + "' and state='online']").getResultList();
    }

    public List<UserEntity> findByOnlineNotMeUsingEmail(String userEmail){
        return select().where("email_user != '"+ userEmail + "' and state='online']").getResultList();
    }

    public UserEntity findByUserName(String userName){
        return select().where("name_user = '"+ userName + "'").getSingleResult();
    }
}

```

#### ❖ Dto :

Đây là nơi chứa lớp để lưu trữ các dữ liệu toàn cục(SESSION). Để nhận biết cần thêm chú thích @Component(instance = InstanceType.SESSION) ở trên lớp. Sau khi được gán giá trị mỗi đối tượng thuộc lớp này có thể sử dụng ở các lớp Action mà không cần khởi tạo. Để xóa giá trị cần thêm comment @RemoveSession vào một hàm và khi hàm này được gọi sẽ thực hiện xóa giá trị SESSION.  
Ví dụ : lớp UserDto : lưu SESSION mỗi khi người dùng đăng nhập

```

UserDto.java
package tictactoe.dto;

import java.io.Serializable;

@Component(instance = InstanceType.SESSION)
public class UserDto implements Serializable {
    private static final long serialVersionUID = 1L;

    public String userName;
}

```

- ❖ **Common**: thư mục này chứa các hằng số phương thức để có thể dùng chung và thường các hằng số phương thức thường để static  
Ví dụ : lớp Constants chứa các phương thức static ONLINE,OFFLINE

```

Constants.java
package tictactoe.common;

public class Constants {
    public static final String ONLINE = "online";
    public static final String OFFLINE = "offline";
}

```

- ❖ **Interceptor** : thư mục chứa các lớp Interceptor giúp thực hiện kiểm tra đồng loạt một điều kiện được định nghĩa sẵn trong các class Interceptor.  
Ví dụ : lớp LoginConfirmInterceptor kiểm tra SESSION cho các page trong project, nếu người dùng chưa đăng nhập thì đều thực hiện Interceptor yêu cầu người dùng đăng nhập

```

LoginConfirmInterceptor.java
package tictactoe.interceptor;

import javax.annotation.Resource;

public class LoginConfirmInterceptor extends AbstractInterceptor {

    private static final long serialVersionUID = 1L;

    @Resource
    protected UserDto userDto;

    @Override
    public Object invoke(MethodInvocation invocation) throws Throwable {
        // この条件がtrueであればログイン済みと判断
        // この条件がfalseであればログインページへ移動
        return (!isExecuteMethod(invocation) || isLoggedIn()) ? invocation
            .proceed() : "/login/?redirect=true";
    }

    private boolean isExecuteMethod(MethodInvocation invocation) {
        return invocation.getMethod().isAnnotationPresent(Execute.class);
    }

    private boolean isLoggedIn() {
        // Sessionにユーザー情報が登録されているかチェック
        return (userDto != null && userDto.userName != null);
    }
}

```

Để có thể sử dụng Interceptor thì cần phải config file "customize.dicon ", tại đây ta sẽ định nghĩa các lớp Action có thể sử dụng Interceptor

```
<initMethodname="addAspectCustomizer">
    <arg>"aop.traceInterceptor"</arg>
    <!-- ここ重要(追加必須) -->
    <arg>true</arg>
</initMethod>
<!-- SESSIONのチェック設定はここから -->
<initMethodname="addCustomizer">
    <arg>

    <componentclass="org.seasar.framework.container.customizer.AspectCustomizer">

    <propertyname="interceptorName">"loginConfirmInterceptor"</property>
    <!-- ここ重要(追加必須) -->
    <propertyname="useLookupAdapter">true</property>

    <!-- このActionだけセッションチェックを行なう -->
    <initMethodname="addClassPattern">
        <arg>"tictactoe.action"</arg>
        <arg>"MypageAction"</arg>
    </initMethod>
    <initMethodname="addClassPattern">
        <arg>"tictactoe.action"</arg>
        <arg>"TictactoeAction"</arg>
    </initMethod>
    <initMethodname="addClassPattern">
        <arg>"tictactoe.action"</arg>
        <arg>"UserAction"</arg>
    </initMethod>
    </component>
    </arg>
</initMethod>
<!-- SESSION のチェック設定はここまで -->
```

### ❖ Sử dụng chú thích và thông báo lỗi

Trong SAstruts thì các chú thích có nhiệm vụ quan trọng và cũng là đặc trưng riêng của SAstruts so với các framework khác. Ta sẽ đi phân tích nhiệm vụ, chức năng của một số chú thích hay sử dụng và quan trọng. Chú thích trong SAstruts khi sử dụng đều bắt đầu bằng @

- ✓ Execute : Đây là chú thích đi kèm các phương thức hoạt động .Khi chú thích Execute thường có các option đi kèm
  - Validator : Được sử dụng để kiểm tra các dữ liệu có hợp lệ hay không. Nếu validator = true nghĩa là kiểm tra có hiệu lực. Khi ấy lỗi sẽ được đẩy lên trang JSP và hiển thị bằng <html: error/>. Nếu validator = false thì không. Nó thường dùng để kiểm tra các lỗi đầu vào trong

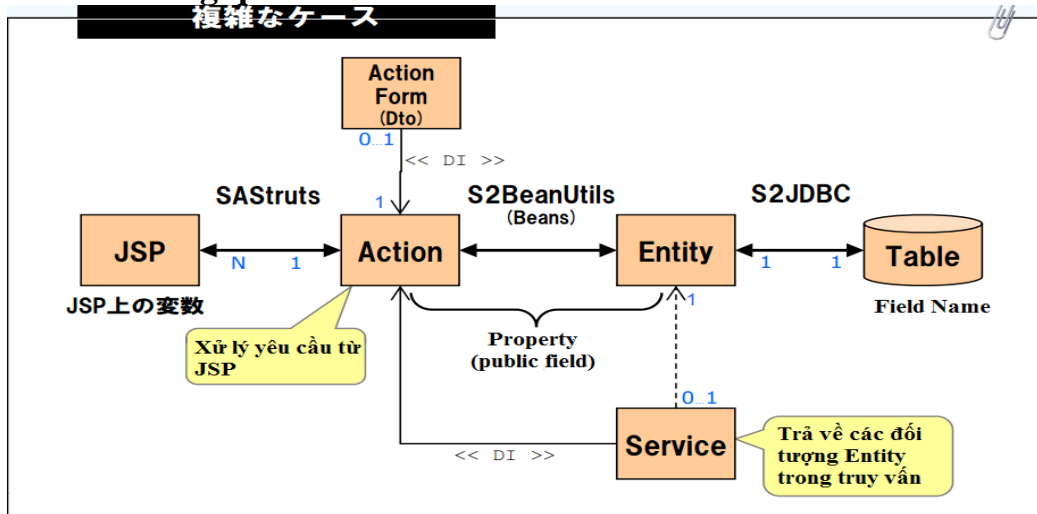


ActionForm. Các thông báo lỗi có thể được sửa đổi thêm bớt trong file application.properties.

- Nếu muốn thêm một số thông báo lỗi thì ta có thể sử dụng option saveError với ActionMessenger.
  - Nếu muốn thêm nhiều thông báo lỗi thì có thể tự viết các hàm báo lỗi bằng cách sử dụng ActionMessenger và gửi giá trị lên validate trong @Execute
  - Ngoài ra còn nhiều option khác như input,urlParttem,...
- ✓ ActionForm: Chú thích này được sử dụng khi trong Action muốn sử dụng ActionForm. Nếu không có chú thích này thì ActionForm được khai báo trong lớp Action sẽ không hoạt động.
  - ✓ Required: Chú thích kiểm tra các điều kiện cần thiết.
  - ✓ Minlength: Chú thích kiểm tra độ dài nhỏ nhất của chuỗi kí tự (thường dùng để kiểm tra chuỗi kí tự đầu vào trong ActionForm)
    - minlength: có giá trị integer, xác định giá trị nhỏ nhất của chuỗi kí tự nhập vào
  - ✓ Maxlength: Chú thích kiểm tra độ dài lớn nhất của chuỗi kí tự
    - maxlength: có giá trị integer, xác định giá trị lớn nhất của chuỗi kí tự nhập vào
  - ✓ IntegerType: Chú thích kiểm tra việc có thể chuyển đổi sang kiểu Integer
  - ✓ EmailType: Chú thích kiểm tra xem đó có phải là địa chỉ mail
  - ✓ Ngoài ra còn nhiều chú thích khác được sử dụng nhưng trên đây là những chú thích em đã sử dụng trong game TicTacToe. Nếu muốn tham khảo các chú thích khác có thể xem tại

["http://sastruts.seasar.org/annotationReference.html"](http://sastruts.seasar.org/annotationReference.html)

## ❖ Mô hình tổng quát về SAStruts Framework



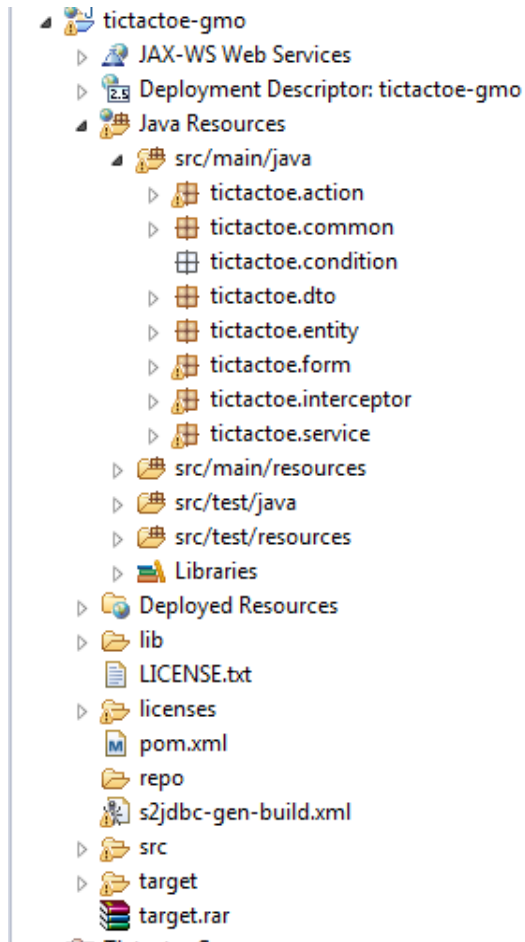


## III. Game TicTacToe

### 1. Thành phần, cấu trúc game

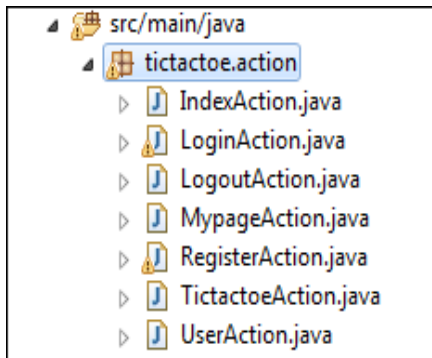
Game TicTacToe được sử dụng Framework Seasar2 nên cấu trúc rõ ràng và tuân theo cấu trúc chung của 1 framework seasar2.

❖ Đây là cấu trúc tổng thể của game

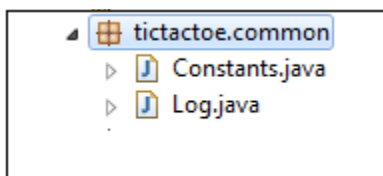


❖ Sau đây là cấu trúc theo từng phần của game

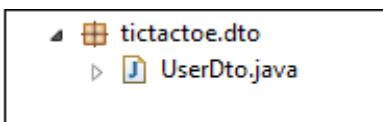
- Action: Chứa các class Action, mỗi Action là 1 chức năng của game.



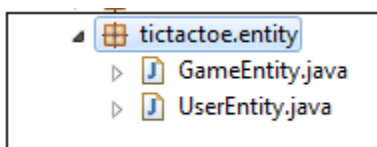
- ✓ LogicAction: Xử lý đăng nhập cho người dùng, sau khi người dùng đăng nhập thì sẽ lưu Session lại và chuyển người dùng tới trang chủ. Nếu người dùng chưa có tài khoản thì sẽ phải đăng ký.
  - ✓ RegisterAction: Xử lý việc đăng kí của người dùng
  - ✓ MypageAction: Trang này em không xử lý nhiều, chỉ thử nghiệm việc truy xuất cơ sở dữ liệu và sử dụng Dto
  - ✓ TictactoeAction: Xử lý việc người chơi với máy sử dụng giải thuật Heuristic. Có 2 chức năng cho máy chơi trước hoặc người chơi trước.
  - ✓ UserAction: Xử lý chức năng cho 2 người chơi với nhau.
- Common: Chứa các biến static được định nghĩa



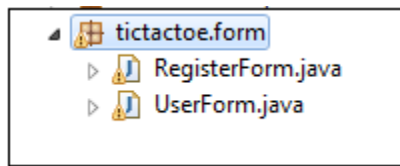
- Dto: Lưu session



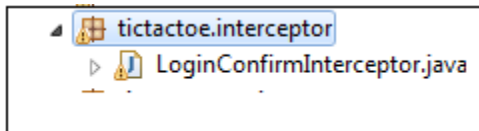
- Entity: Chứa các lớp Entity, mỗi lớp là 1 bảng trong CSDL



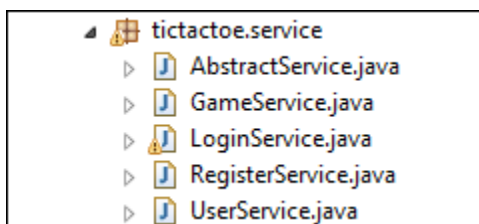
- Form: Chức các ActionForm, lưu trữ input của các trang JSP gửi tới



- Interceptor: Lưu trữ file Interceptor để kiểm tra session trong các Action được khai báo trong config



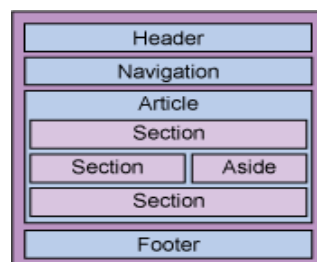
- Service: Lưu trữ các file sử dụng tài nguyên Entity để thực hiện truy vấn cơ sở dữ liệu



## 2. Các công nghệ mới và giải thuật sử dụng trong game

- **HTML5:** Sử dụng HTML5 để làm giao diện và hiệu ứng animation cho game. Hiện tại thì em đã sử dụng HTML5 để làm giao diện cho game và hoàn thành (50%). Em đã thu được một số kiến thức như sau:

- ✓ Cấu trúc chung: Thiết kế trang HTML5 thường có một vùng Header (Tiêu đề), một vùng Navigation (Chuyên hướng), một vùng Article (Bài viết) có chứa ba Section (phần), một Aside (Nhận xét), và cuối cùng, là vùng Footer (Chân trang).



- ✓ **Vùng Header** để chứa tiêu đề và phụ đề trang. Bạn sử dụng thẻ

`<header>` để tạo ra nội dung cho vùng Header của trang.

```
<header>
  <h1>Heading Text</h1>
  <p> Text or images can be included here</p>
  <p> Logos are frequently placed here too</p>
</header>
```

- ✓ **Vùng Navigation** (Chuyển hướng) của trang tạo ra bằng cách sử dụng thẻ `<nav>`. Phần tử `<nav>` định nghĩa một vùng đặc biệt dành cho việc chuyển hướng. Thẻ `<nav>` nên được sử dụng để chuyển hướng trang web chính, không dùng để thiết lập các liên kết có chứa trong các vùng khác của trang.

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Our Products</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
</nav>
```

- ✓ **Vùng Article:** Trang web mà bạn đang thiết kế có chứa một phần Article, giữ nội dung thực sự của trang. Bạn sử dụng thẻ `<article>` để tạo vùng này, và thẻ đó định nghĩa nội dung có thể được sử dụng độc lập với các nội dung khác được tìm thấy trên trang này. Vùng Article có chứa ba vùng Section. Bạn tạo ra các vùng này bằng cách sử dụng thẻ `<section>`. Một `<section>` chứa các vùng thành phần của nội dung trang web có liên quan. Thẻ `<section>` và thẻ `<article>` nữa có thể chứa các header (tiêu đề), các footer (chân trang), hoặc bất kỳ các thành phần nào khác cần thiết để hoàn thành phần này. Thẻ `<section>` dùng cho nội dung tạo nhóm. Nội dung cho cả hai thẻ `<section>` và thẻ `<article>` thường bắt đầu bằng một `<header>` và kết thúc bằng một `<footer>`, với nội dung cho

thẻ này ở giữa. Thẻ <section> cũng có thể chứa các thẻ <article>, cũng giống như thẻ <article> có thể chứa các thẻ <section>. Thẻ <section> nên được dùng để tạo nhóm các thông tin giống nhau, và thẻ <article> nên được sử dụng cho các thông tin như một bài viết hoặc một blog mà chúng có thể bị gỡ bỏ và được đặt trong một ngữ cảnh mới mà không ảnh hưởng đến ý nghĩa của nội dung. Thẻ <article> như tên gọi của nó, cung cấp một gói thông tin đầy đủ. Ngược lại, thẻ <section> chứa thông tin liên quan, trừ thông tin không thể được đặt trong một ngữ cảnh khác với chính nó, do nghĩa của nó sẽ bị mất.

```
<article>
  <section>
    <header>
      <h1>This is a video section</h1>
    </header>
    <p><video
src="http://people.xiph.org/~maikmerten/demos/BigBuckBunny.ogv"
      controls autoplay loop>
      <div class="no-html5-video"><p>This video will work in
        Mozilla Firefox or Google Chrome only. </p>
      </div>
    </video></p>
  </section>
</article>
```

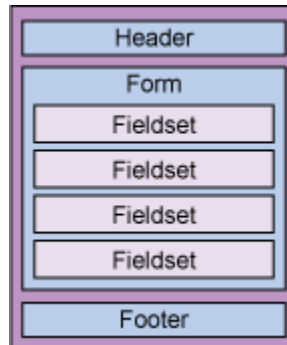
- ✓ **Vùng Footer:** Phần tử <footer> chứa thông tin về một trang, bài viết, hoặc một phần, chẳng hạn như tác giả hoặc ngày viết bài. Là phần cuối trang, nó có thể chứa bản quyền hoặc các thông tin pháp lý quan trọng khác

```
<footer>
  <p>Copyright 2011 Acme United. All rights reserved.</p>
</footer>
```

## ➤ Sử dụng HTML5 trong project

Hiện tại thì em mới chỉ sử dụng HTML5 vào việc làm 2 chức năng Login và Logout. HTML5 có nhiều ưu điểm hơn các thế hệ HTML trước đó vì  
HTML5 = HTML + CSS + JavaScript

Khi sử dụng HTML5 thì em mới chỉ sử dụng được Form và một số thuộc tính mới của thẻ <input>. Các Form trong HTML5 đã có nhiều thay đổi lớn. Một số nhiệm vụ mà trước đây đòi hỏi phải dùng JavaScript thì bây giờ có thể dễ dàng được thực hiện mà không cần nó. Dưới đây là bố cục của Form trong HTML5:



Trong HTML5, có hai thuộc tính mới đã được thêm vào thẻ <form> là: "autocomplete" và "novalidate". Thuộc tính "autocomplete" cho phép xuất hiện danh sách các gợi ý giống như khi tìm kiếm trong Google. Thuộc tính "novalidate" tắt tính năng xác nhận hợp lệ cho form, điều này rất có ích trong quá trình thử nghiệm. Thẻ <fieldset> (thiết lập trường) có ba thuộc tính mới là: disable (vô hiệu hóa), name (tên), và form. Thuộc tính disable làm cho thẻ <fieldset> không hoạt động. Thuộc tính name đặt tên cho thẻ <fieldset>. Giá trị thuộc tính form là mã định danh (ID) của form này hay các form thuộc về thẻ <fieldset>. Trong HTML5, một thẻ <fieldset> có thể nằm ngoài form này hay các form mà nó thuộc về. Khi một thẻ <fieldset> được đặt bên ngoài form, bạn phải thiết lập thuộc tính form của thẻ <fieldset> để cho thẻ <fieldset> có thể được liên kết với form hoặc các form đúng. Thẻ <label> (nhãn), định nghĩa một sự phân loại cho một phần tử đầu vào, có một thuộc tính mới là: form. Giá trị thuộc tính form là ID của form này hay các form mà thẻ <label> thuộc về. Thẻ <label> cũng có thể được đặt ở bên ngoài của form, vì vậy ở đây thuộc tính form cũng được sử dụng để

liên kết <label> với form thích hợp. Thẻ <input> (đầu vào) có một số kiểu mới cũng như các thuộc tính nâng cao khả năng sử dụng của form.

Sau đây là file Login sử dụng HTML5 của em:

```
<div id="login" class="animate form">
  <form action="" method="">
    <h1>LOG IN</h1>
    <p>
      <label for="username" class="uname" data-icon="u">User Name</label>
      <input id="username" name="username" required=""
        type="text" autofocus=""
        placeholder="Hãy nhập UserName vào đây"
        pattern="[a-zA-Z0-9_]{5,40}"
        title="Username có độ dài từ 5-40 kí tự và không chứa kí tự đặc biệt như $, @, !, #" />
      <!--
      <html:errors property="username"/>
      -->
    </p>
    <p>
      <label for="password" class="youpasswd" data-icon="p">
        Password</label>
      <input id="password" name="password" required=""
        type="password" placeholder="Hãy nhập Password vào đây"
        pattern="[a-zA-Z0-9]{6,20}"
        title="Password có độ dài từ 6-20 kí tự và không chứa kí tự đặc biệt" />
      <!--
      <html:errors property="password"/>
      -->
    </p>
    <p class="keeplogin">
      <input type="checkbox" name="loginkeeping" id="loginkeeping"
        value="loginkeeping" /> <label for="loginkeeping">Lưu
        mật khẩu lại</label>
    </p>
    <p class="login button">
      <input name="index" type="submit" value="Login" />
    </p>
    <p class="change_link">
      Not a member yet ? <a href='<:url value="/register/">'>

```

Ở đây em sử dụng các thuộc tính mới như là:

- ✓ Thuộc tính **required** tạo điều kiện thuận lợi cho việc đặt một trường bắt buộc như là một điều kiện tiên quyết để hoàn thành biểu mẫu. Điều này là hợp lệ với các trường văn bản, tìm kiếm, URL, điện thoại, email, mật khẩu, các nút chọn ngày, số, hộp kiểm tra, nút tròn, và các trường tập tin
- ✓ Thuộc tính **autofocus** bảo đảm rằng trọng tâm đầu vào sẽ ở trên trường này khi trang được mở ra. Đây là một chức năng cho phép người dùng truy cập ngay lập tức vào biểu mẫu ngay khi trang được mở.
- ✓ Thuộc tính **placeholder** đặt văn bản vào giữa các dấu ngoặc kép bên trong trường này dưới dạng văn bản màu xám nhạt. Thuộc tính

placeholder sẽ gợi ý cho người dùng biết trường này phải chứa những gì và được hiển thị thế nào khi trường này rỗng. Nếu bạn di chuyển đến một trường khác mà không nhập dữ liệu, thì văn bản placeholder sẽ được hiển thị. Khi bạn sử dụng thuộc tính autofocus với thuộc tính placeholder, văn bản placeholder sẽ biến mất do trọng tâm đang được đặt vào trường autofocus này.

- ✓ Trường **pattern** mô phỏng các chức năng của một biểu thức JavaScript chính quy (regex) truyền thống. Đầu vào phải khớp với cấu trúc mẫu của biểu thức chính quy được định nghĩa trước khi nó được xác nhận hợp lệ. Trường này làm việc với các kiểu text, search, url, telephone, email, và password

Ngoài việc sử dụng HTML5 trong việc thiết kế giao diện em còn sử dụng HTML5 vào việc tạo hiệu ứng ảnh động khác nhưng đang trong quá trình tìm hiểu nên em sẽ trình bày ngắn gọn những gì mình tìm hiểu được như sau:

- ✓ Bản chất của hình ảnh động vẫn là những hình ảnh tĩnh được ghép lại với nhau theo một quy tắc nhất định mà chúng ta đã được biết đến: 24 hình giây. Mắt con người không có khả năng cảm nhận hình ảnh một cách liên tục mà khi ánh sáng đập vào võng mạc sẽ để lại những ghi nhớ trong đó phải một thời gian ngắn nó mới mất đi. Nếu một hình được lưu trữ trong một khoảng thời gian 1/24s và được thay bằng hình tiếp theo thì chúng ta sẽ không thể nhận biết được đó là hình ảnh mới. Điều này dẫn đến hiện tượng chúng ta có cảm nhận rằng hình ảnh là động chứ không còn là hình tĩnh nữa.
- ✓ Cơ chế
  - +Bước 1: Vẽ hình đầu tiên

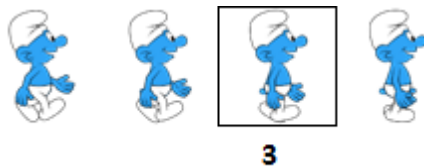




+Bước 2: Xóa màn hình vẽ hình thứ 2



+ Bước 3: Làm lại bước 2 với thứ tự hình được tăng lên.



Như vậy ta sẽ có ảnh chuyển động liên tục. Hiện tại em mới tìm hiểu đến cơ chế sử dụng thẻ <canvas> và dùng javascript để update ảnh làm cho nó di chuyển.

### ➤ **Giải thuật Heuristic khi chơi với máy**

Khi cho người chơi với máy em dùng giải thuật Heuristic để giúp cho máy có những nước đi thông minh hơn khi chơi. Khi thiết kế thì em cho game có 2 chức năng là người chơi trước và máy chơi trước. Giải thuật sẽ được trình bày ngắn gọn như sau:" Máy tính sẽ thử lần lượt đánh vào các ô xem ô nào có khả năng cáo nhất sẽ đánh vào đó. Cách đánh giá điểm số là cứ mỗi hàng, cột có 2 ô trống và 1 ô là nước đi của máy sẽ tính là 1 điểm, có 1 ô trống và 2 ô của máy sẽ tính là 10 điểm, có 3 ô của máy(chiến thắng) sẽ tính 100

điểm, người lại với đối thủ là -1, -10, -100, từ đó cộng tất cả số điểm lại chọn ra ô có số điểm cao nhất để đi ".

### 3. Một số thủ thuật cần nhớ khi làm game TicTacToe

Mặc dù đã nói trong phần cấu trúc Seasar2 nhưng ở đây em xin nhắc lại cách thức truyền dữ liệu giữa Action(Java) và file JSP

#### ❖ Truyền dữ liệu từ javascript về Java

Truyền dữ liệu từ JSP về Java thông qua Form, trong Form sử dụng button với type="submit" để trực tiếp gửi dữ liệu về Java, các dữ liệu gửi về sẽ được Java xác nhận bằng thuộc tính name="property\_name" của nó trên JSP. Nếu muốn truyền dữ liệu từ Javascript thì thông qua JSP ta làm như sau:

- ✓ Khai báo thẻ input có type= "hidden" trong JSP và xác định nó bằng thuộc tính name, Java sẽ xác nhận dữ liệu gửi từ JSP tới thông qua name

```
<c:forEach var="user" items="${list}" varStatus="status">
  <form action='<c:url value="/user/play/" />' method="get">
    <input type="hidden" name="invitePlayer" value="{f:h(userName)}">
    <input type="hidden" name="exceptPlayer" value="{f:h(user)}">
  <tr>
    <td><c:out value="{f:h(user)}" /></td>
    <td><input type="submit" value="Invite"><br /></td>
  </tr>
</form>
</c:forEach>
```

Còn Javascript sẽ tham chiếu tới giá trị mà JSP lấy từ Java về thông qua id

```
function reload() {
  document.forms["state_form"].submit();
}

function newGame() {
  document.getElementById("idplayer").value = (player === "X" ? "X" : "O");
  document.getElementById("idstate").value = "000000000";
  reload();
}
```

```
<form id="state_form" method="post">
  <input type="hidden" id="idplayer" name="nextPlayer" value="">
  <input type="hidden" id="idstate" name="cells" value="">
</form>
```

- ✓ Tại trang Java, lấy giá trị gửi về trong thẻ hidden bằng cách sử dụng `getParameter` của lớp `HttpServletRequest`

```
nextPlayer = request.getParameter("nextPlayer");
cells = request.getParameter("cells");
```

- ❖ Truyền dữ liệu từ Java đến JSP thông qua `${f:h()},{f:u()}`

- ✓ Phía JSP

```
Xin chào bạn ${f:u(userName) }
```

- ✓ Phía Java

```
request = RequestUtil.getRequest();
request.setAttribute("list", userOnlines);
request.setAttribute("list_invite", games);
```

## 4. Log4j

Em sử dụng Log4j để đẩy lỗi từ java về Log4j. Nó hoàn toàn riêng biệt với Eclipse nên sử dụng đơn giản và rất nhẹ, tiện ích cho việc debug. Để sử dụng được log4j thì trước hết phải config trong file `log4j.properties` như sau:

```
log = D://log4j
log4j.category.tictactoe = DEBUG, FILE
#log4j.rootLogger = DEBUG, FILE

# Define the file appender
log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=${log}/log5.html

# Define the layout for file appender
log4j.appender.FILE.layout=org.apache.log4j.HTMLLayout
log4j.appender.FILE.layout.Title=HTML Layout Example
log4j.appender.FILE.layout.LocationInfo=true
```

Khi ấy muốn kiểm tra lỗi ta chỉ cần khai báo Log4j trong file java và log nó ra. Theo config của em thì toàn bộ log sinh ra sẽ được đặt trong ổ D/log4j/

```

Logger logger = Logger.getLogger(UserAction.class);

@Execute(validator = false, input = "index.jsp")
public String index() {
    /**
     * Kiểm tra danh sách số người online trong csdl và lấy ra những người
     * được mời và những người mời chơi sau đó truyền tham số vào list và
     * list_invite để tham số truyền lên client
     */
    userName = userDto.userName;
    List<UserEntity> userOnlines = userService
        .findByOnlineNotMeUsingName(userDto.userName);
    logger.debug("UserOnlines : " + userOnlines);
    List<GameEntity> games = gameService.findInvite(userName);
    logger.debug("ListInvite: " + games);

    request = RequestUtil.getRequest();
    request.setAttribute("list", userOnlines);
    request.setAttribute("list_invite", games);

    return "index.jsp";
}

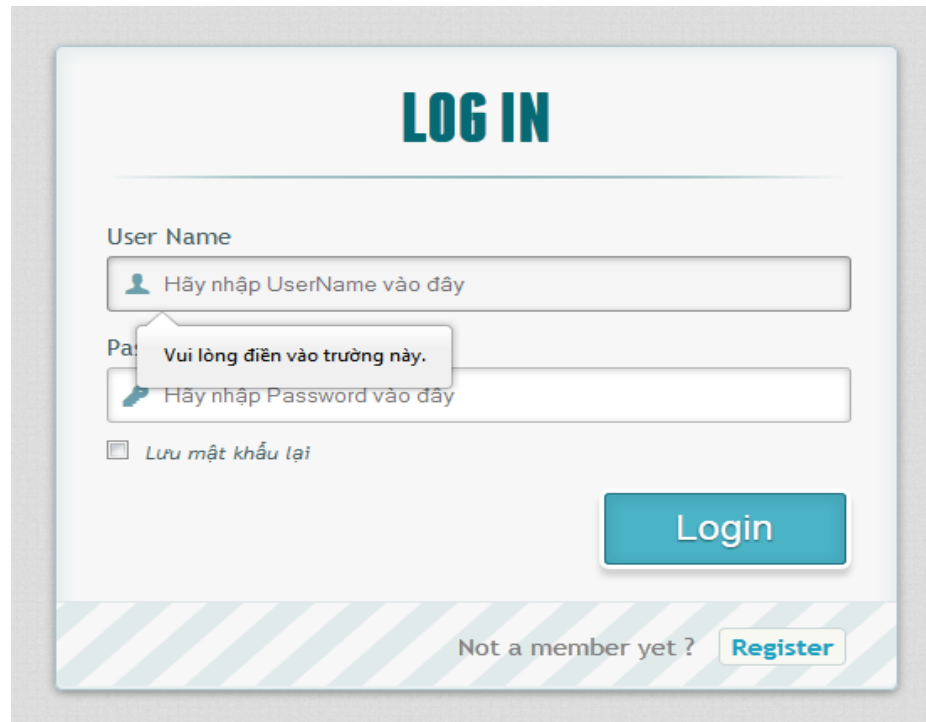
```

## 5. Demo

### ❖ Login

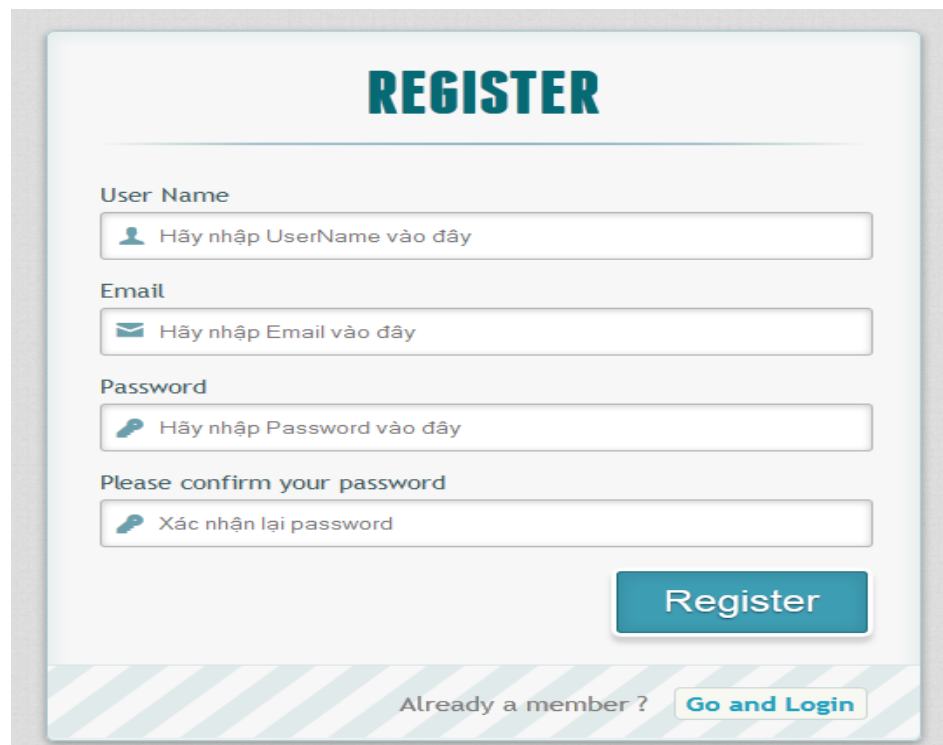
#### ✓ Giao diện chính

- ✓ Sử dụng HTML5 và ActionMessenger đẩy lỗi từ Action lên



The image shows a login form titled "LOG IN". It has two input fields: "User Name" and "Password". The "User Name" field has a placeholder text "Hãy nhập UserName vào đây". The "Password" field has a placeholder text "Hãy nhập Password vào đây". A tooltip with the text "Vui lòng điền vào trường này." is pointing to the "Password" field. Below the fields is a checkbox labeled "Lưu mật khẩu lại". A "Login" button is at the bottom right. At the very bottom, there is a link "Not a member yet ? Register".

## ❖ Register



The image shows a register form titled "REGISTER". It has four input fields: "User Name", "Email", "Password", and "Please confirm your password". The "User Name" field has a placeholder text "Hãy nhập UserName vào đây". The "Email" field has a placeholder text "Hãy nhập Email vào đây". The "Password" field has a placeholder text "Hãy nhập Password vào đây". The "Please confirm your password" field has a placeholder text "Xác nhận lại password". A "Register" button is at the bottom right. At the very bottom, there is a link "Already a member ? Go and Login".

❖ Mypage

**Chuc mung vienbk91 da dang nhap thanh cong**

Danh sach nguoi dung trong DB:

UserName	Password	State
vienbk91	123456	online
datbk91	123456	online
dongbk91	123456	online
hieubk91	123456	online
sachbk91	123456	online

[TicTacToe Online](#)    [Logout](#)    [TicTacToe with Computer](#)

❖ TicTacToe with Computer

---

[←Click de quay lai: vienbk91](#)

O	X	
X	O	

Computer play first

You play first

❖ TicTacToe two people

- ✓ Hiển thị những người online và những người mời mình chơi

Xin chào bạn vienk91

[Quay lại](#) [Logout](#)

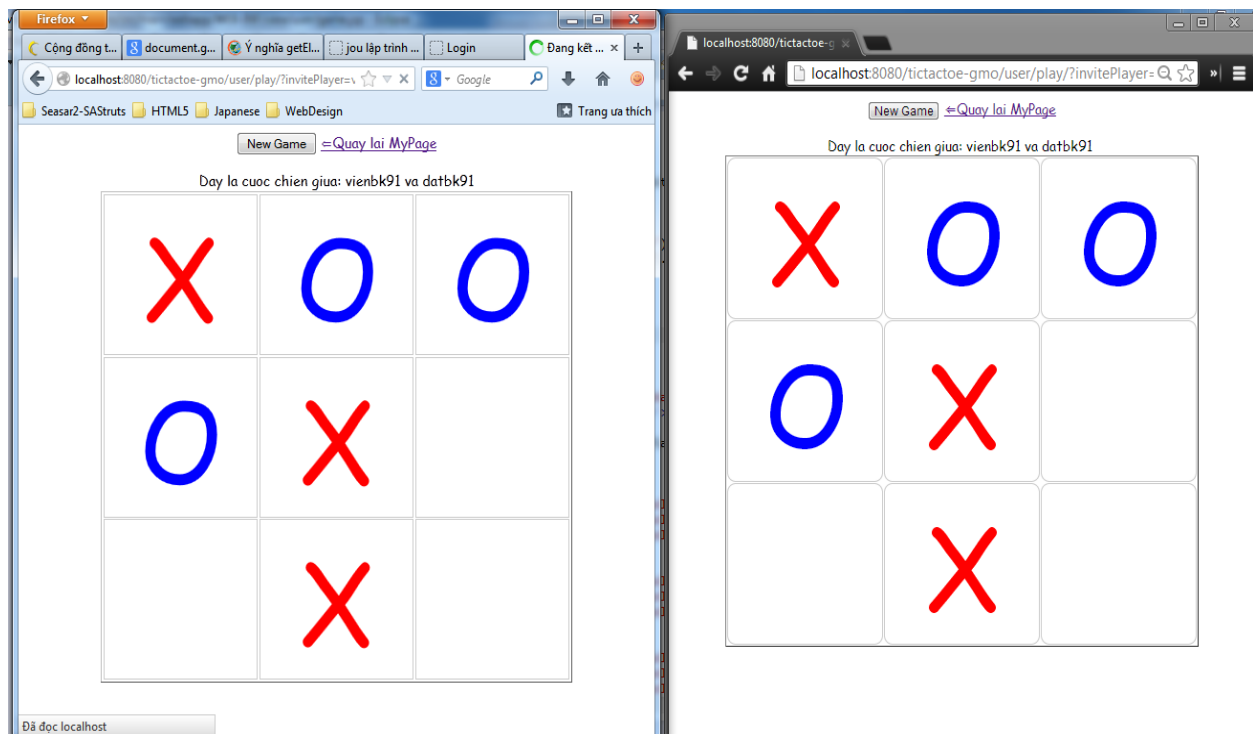
Danh sách những bạn đang Online:

UserName	Invite
datbk91	<input type="button" value="Invite"/>
dongbk91	<input type="button" value="Invite"/>
hieubk91	<input type="button" value="Invite"/>
sachbk91	<input type="button" value="Invite"/>

Danh sách những người mời bạn chơi:

UserName	Except
dongbk91	<input type="button" value="Except"/>

- ✓ Chơi 2 người(thử nghiệm trên 2 browser)



- ✓ Thông báo kết quả khi có thắng và thua

