

## Dossier de conception

---

# Conception d'un site E-learning

### Encadré par :

- Mr. LACHGAR Mohamed

### Réalisé par :

- LECHQER Younesse
- ELEOUAD Abdelhadi

## **SOMMAIRE**

<b>I. PERIMETRE DU PROJET .....</b>	<b>2</b>
1.1. ENJEUX ET VISION DU PROJET .....	3
1.2. ARCHITECTURE FONCTIONNELLE.....	3
1.3. REFERENTIEL DES EXIGENCES .....	3
<b>II.ANALYSE ET CONCEPTION.....</b>	<b>5</b>
<b>IV.ARCHITECTURE APPLICATIVE .....</b>	<b>14</b>
<b>V. ARCHITECTURE TECHNIQUE.....</b>	<b>15</b>

# I. PRESENTATION DU PROJET

## 1.1. VISION DU PROJET

Dans le cadre du Formation concepteur réalisateur Java/JEE pour l'année universitaire 2012/2013 en pratiquant nos connaissances techniques ayant acquis durant cette formation pour réaliser projet de fin formation.

visant la conception et le développement d'un **site web E-Learning** destinée au public souhaitant bénéficier des cours et activités organisées.

Le catalogue des cours organisées, publiées par les Formateurs et valider par l'administrateur, proposant aux Etudiant un large choix et une commodité du processus d'inscription d'une part, mais également aux cours et exercices d'une autre part.

L'objectif de ce projet est d'aboutir à une interface pratique permettant aux Etudiants une expérience exceptionnelle en leur permettant de :

- Afficher ou Imprimer les cours et les exercices ;
- Posez des questions ;
- Faire des recherches.

## 1.2. ARCHITECTURE FONCTIONNELLE

Les principaux acteurs qui auront à utiliser notre système sont les suivants :

- **Administrateur AD** : il possède des droits sur le module administration.
- **Formateur FO** : il possède des droits sur le module Organisation.
- **Etudiant ET** : il possède des droits sur le module Action.

Le schéma suivant décrit le lien entre les profils et les fonctionnalités de notre système :

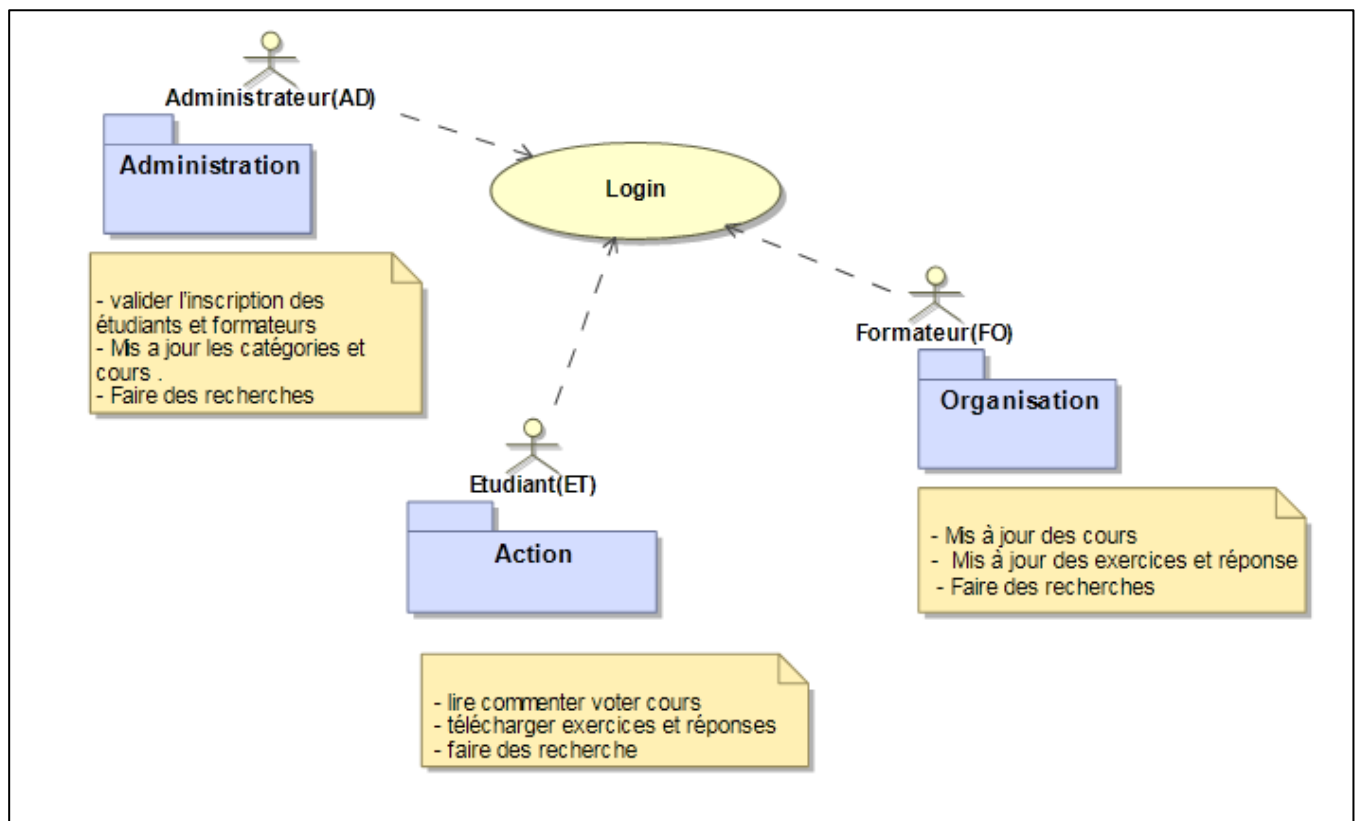


Figure 1 :Architecture fonctionnel.

## 1.3. REFERENTIEL DES EXIGENCES

### LES EXIGENCES FONCTIONNELLES

#### ▪ La recherche

La première fonction que doit offrir est la recherche. Le site doit offrir plusieurs méthodes de recherche : catégories, titre... Les résultats de la recherche doivent apparaître sur une page particulière. Le visiteur doit pouvoir aussi accéder à une classification thématique, aux nouveautés ou aux cours les plus répandues.

### ▪ **La consultation**

Chaque chapitre du cours doit pouvoir être présentée en détail sur une page. Cette page doit contenir :

- Une description du chapitre.
- contenu détaillée.
- Exercice et réponse.
- Téléchargement du chapitre.

### ▪ **L'Edition**

Le formateur doit avoir la possibilité de mettre à jour des cours dans il est participer. Il doit aussi pouvoir mettre à jour les exercices et réponse des cours .

## **LES EXIGENCES NON FONCTIONNELLES**

### ▪ **Exigences de qualité**

Pour attirer de nouveaux Etudiant et pour fidéliser des Etudiant existants il est important de respecter les exigences de qualité suivantes :

- Ergonomie sobre permettant des présentations claires.
- Une présentation exhaustive de chaque cours publiée.
- Formulaire d'inscription simple et facile à remplir.

### ▪ **Exigences de performance**

Administrateur doit pouvoir gérer tous les comptes. Le catalogue de cour doit pouvoir contenir un nombre important des cours. La recherche ne doit pas affecter les ressources du système.

## II. ANALYSE ET CONCEPTION

### DIAGRAMME DE CAS D'UTILISATION

Les **diagrammes de cas d'utilisation** sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

#### **AD : ADMINISTRATEUR :**

Sommaire D'IDENTIFICATION :

---

**Objectif :** valider l'inscription des étudiants et formateurs , Mis à jour les catégories et cours . Faire des recherches des cours, formateurs ,étudiants.

**Résumé :** Cette fonctionnalité permet au Administrateur de valider l'inscription des étudiants et formateurs , Mis à jour les catégories et cours . Faire des recherches des cours, formateurs, étudiants.

Acteur : Administrateur.

---

Description DETAILLÉE :

---

☞ **PRE CONDITIONS :** LA SECRETAIRE S'EST AUTHENTIFIEE SUR LE SYSTEME

☞ **DESCRIPTION DU TRAITEMENT NOMINAL :** L'ACTEUR PEUT :

1. MIS A JOUR LES CATEGORIES ET COURS;
  2. VALIDER L'INSCRIPTION DES ETUDIANTS;
  3. CHERCHER LES COURS, FORMATEURS, ETUDIANTS ;
-

## DIAGRAMME

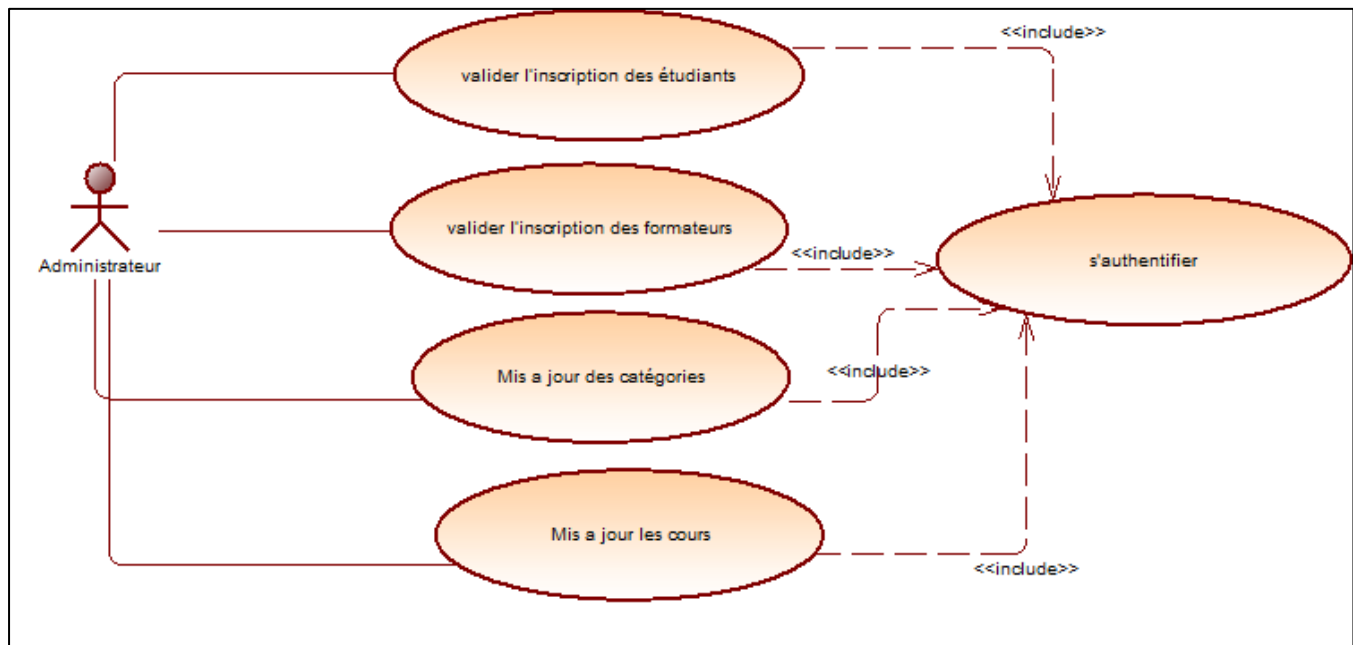


Figure 2 :diagramme de cas d'utilisation Adminstrateur

### 2.1. FO : FORMATEUR

Sommaire D'IDENTIFICATION :

**Objectif :** Mis à jour des cours, Mis à jour des exercices et réponse , Faire des recherches des cours, formateurs .

**Résumé :** Cette fonctionnalité permet au Formateur de Mettre à jour des cours, Mettre à jour des exercices et réponse , Faire des recherches des cours, formateurs .

**Acteur :** Formateur .

Description DETAILLÉE :

☞ **PRE CONDITIONS** : LE FORMATEUR S'INSCRIRE ET VALIDER PAR SYSTEM  
PUIS S'EST AUTHENTIFIEE

☞ **DESCRIPTION DU TRAITEMENT NOMINAL** : L'ACTEUR PEUT :

1. MIS A JOUR LES COURS ET EXERCICES ET REPONSE;
2. CHERCHER LES COURS, FORMATEURS;

**Diagramme :**

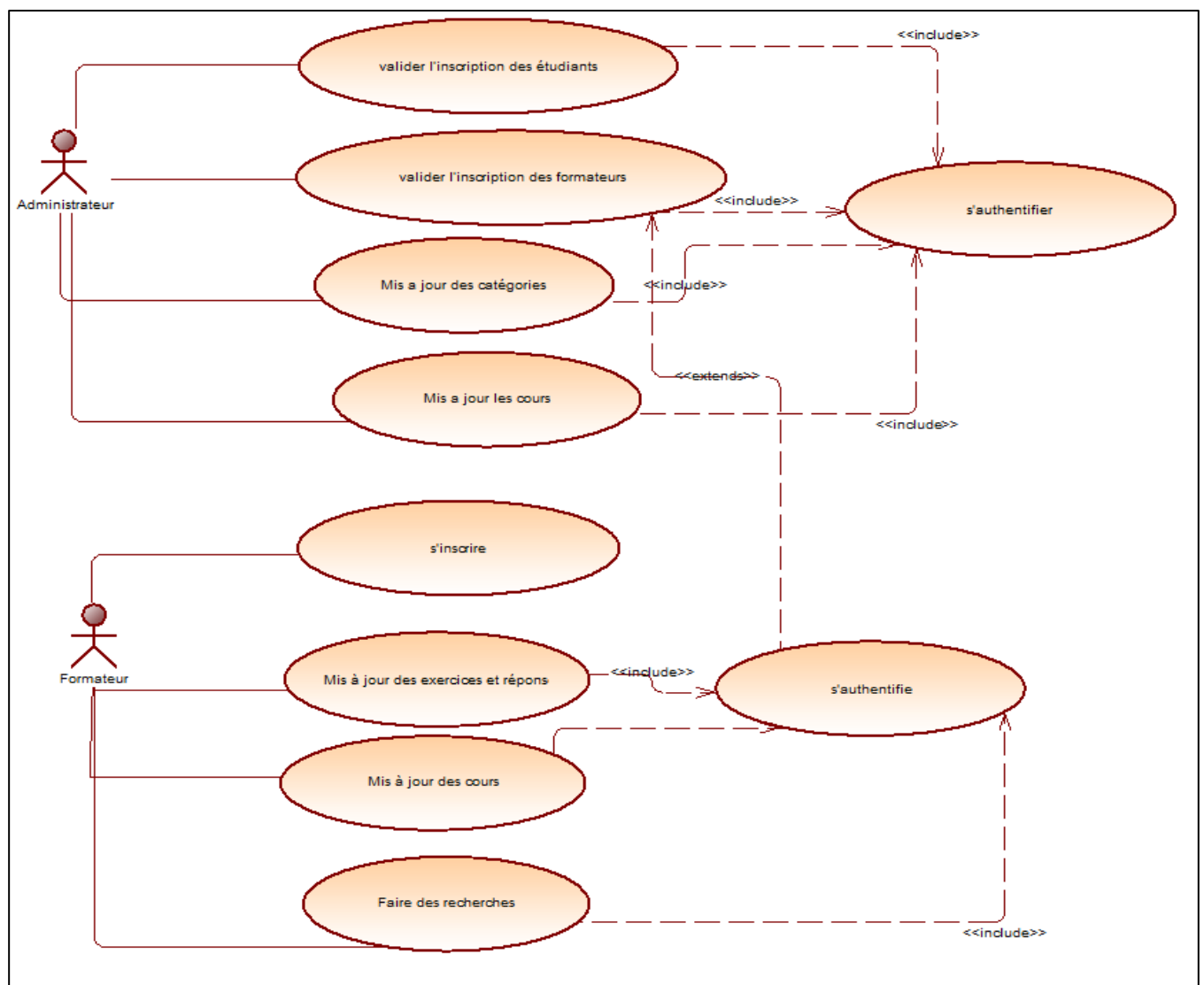


Figure 3 :diagramme de cas d'utilisation Formateur



ET : Etudiant

Sommaire D'IDENTIFICATION :

**Objectif :** lire commenter voter cours, télécharger exercices et réponses, faire des recherche.

**Résumé :** Cette fonctionnalité permet au Etudiant de lire commenter voter cours télécharger exercices et réponses, faire des recherche.

Acteur : Etudiant.

Description DETAILLÉE :

☞ **PRE CONDITIONS :** L'ETUDIANT S'INSCRIRE ET VALIDER PAR SYSTEM PUIS S'EST AUTHENTIFIEE

☞ **DESCRIPTION DU TRAITEMENT NOMINAL :** L'ACTEUR PEUT :

1. LIRE COMMENTER VOTER COURS.
2. LIRE COMMENTER VOTER EXERCICES ET REPOSES,
3. FAIRE DES RECHERCHES.

**Diagramme :**

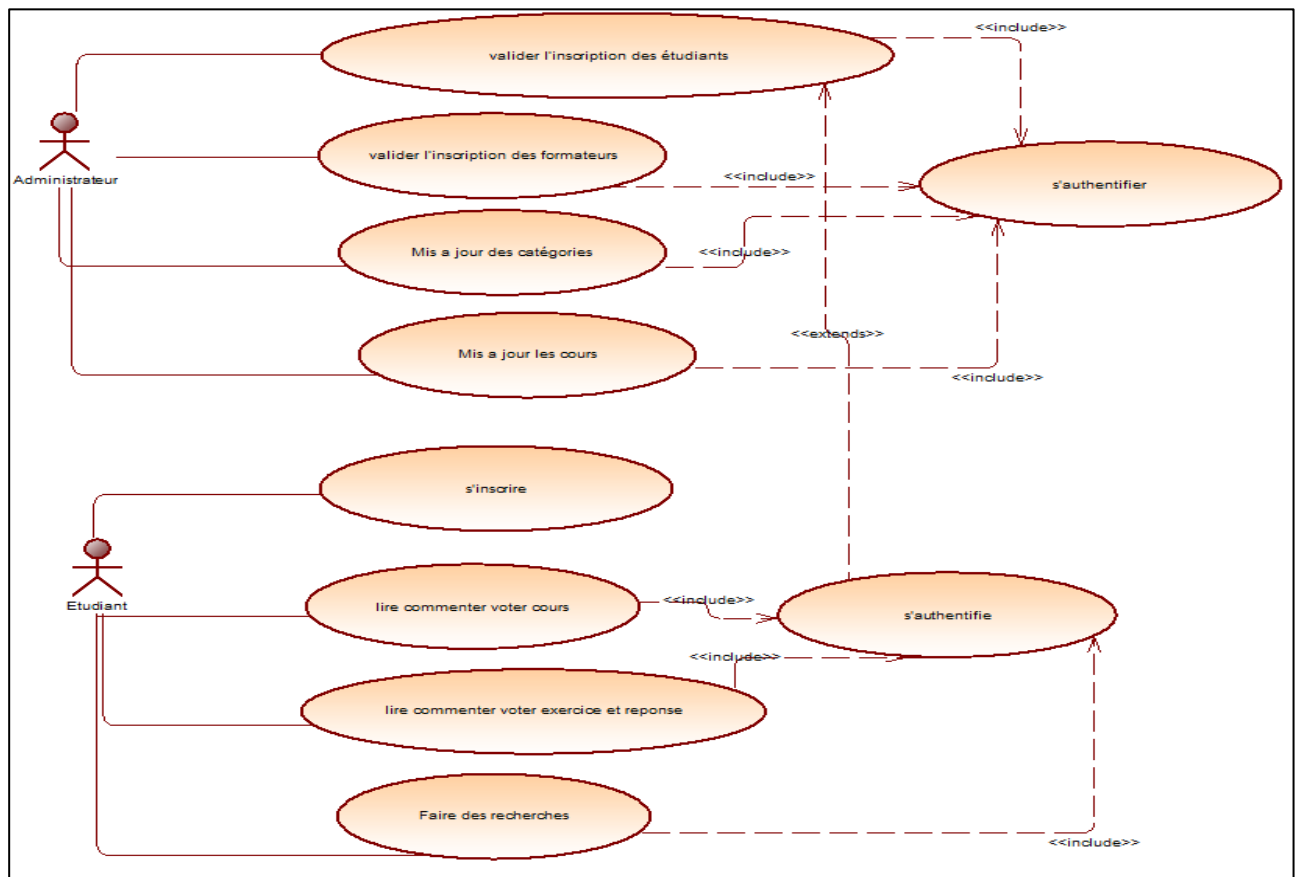


Figure 4 : diagramme de cas d'utilisation Etudiant

## DIAGRAMME DE CLASSES

Le **diagramme de classes** est un schéma utilisé en UML pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

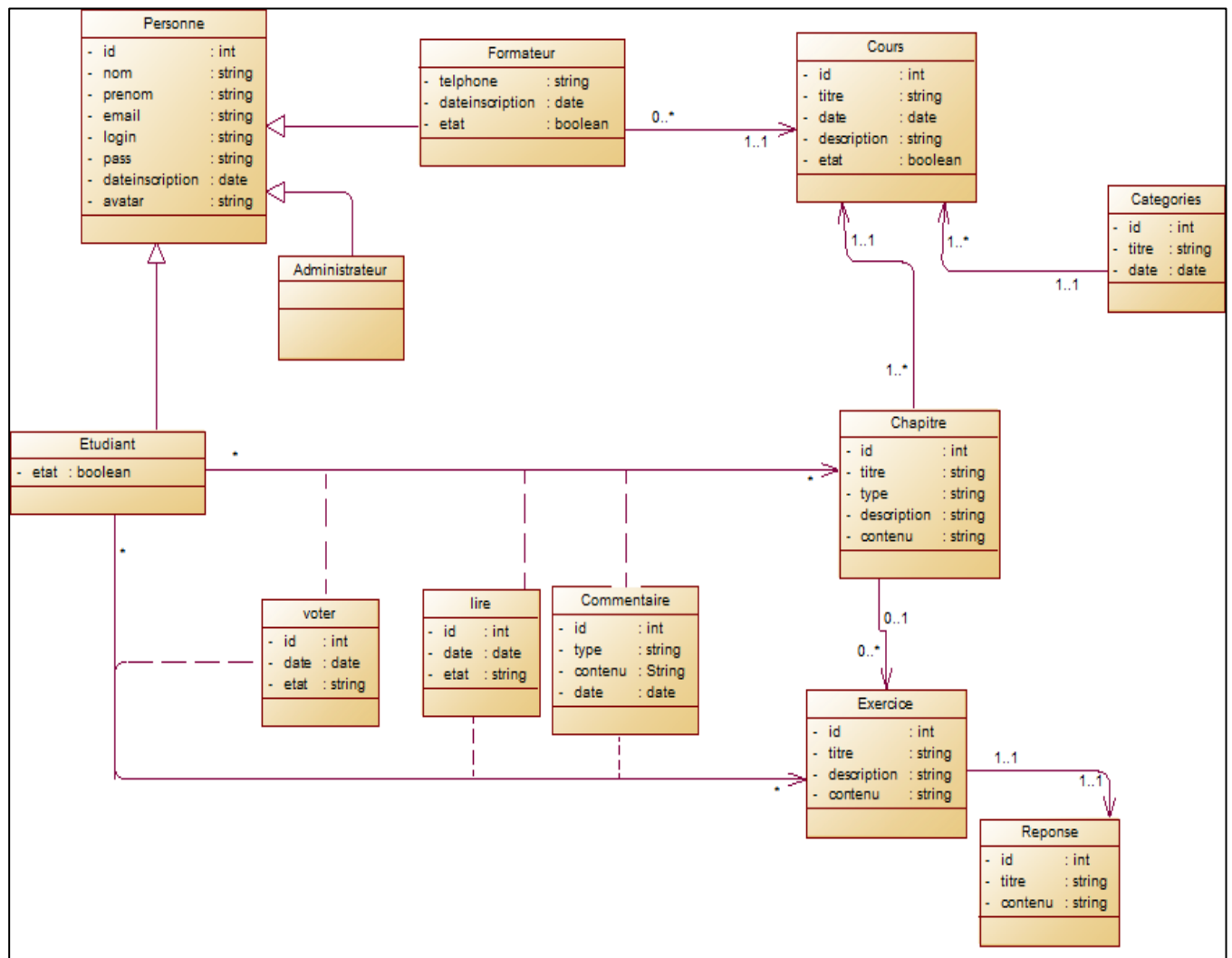


Figure 5 :diagramme de classe

## DIAGRAMME DE SEQUENCES

Le diagramme de séquence est une représentation intuitive qu'on a utilisée pour concrétiser des interactions entre nos entités.

Le diagramme de séquence décrit un scénario qui modélise une exécution particulière d'un cas d'utilisation du début jusqu'à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation.

### Lire chapitre :

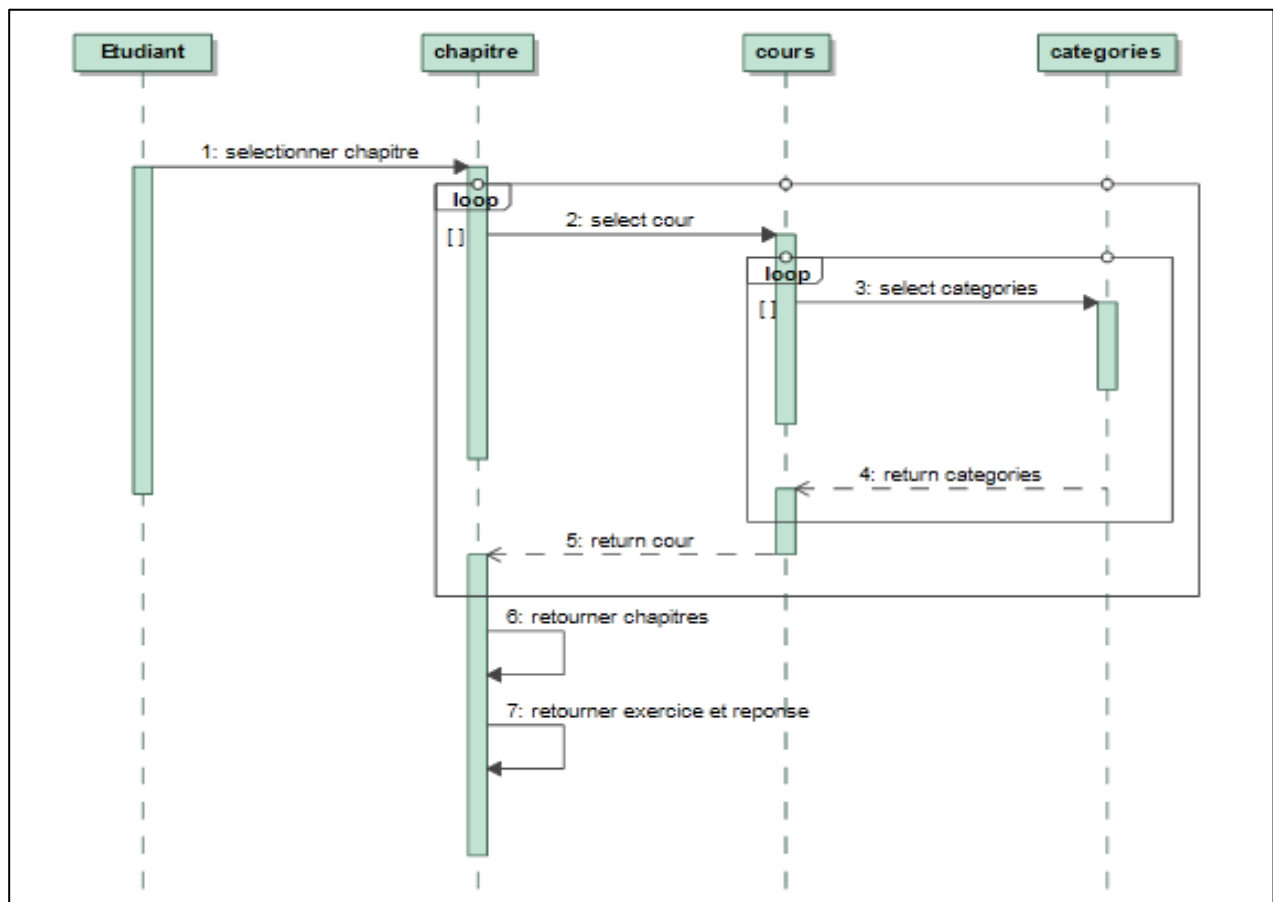


Figure 6 : diagramme de séquence chapitre

### Ajouter cour- chapitre :

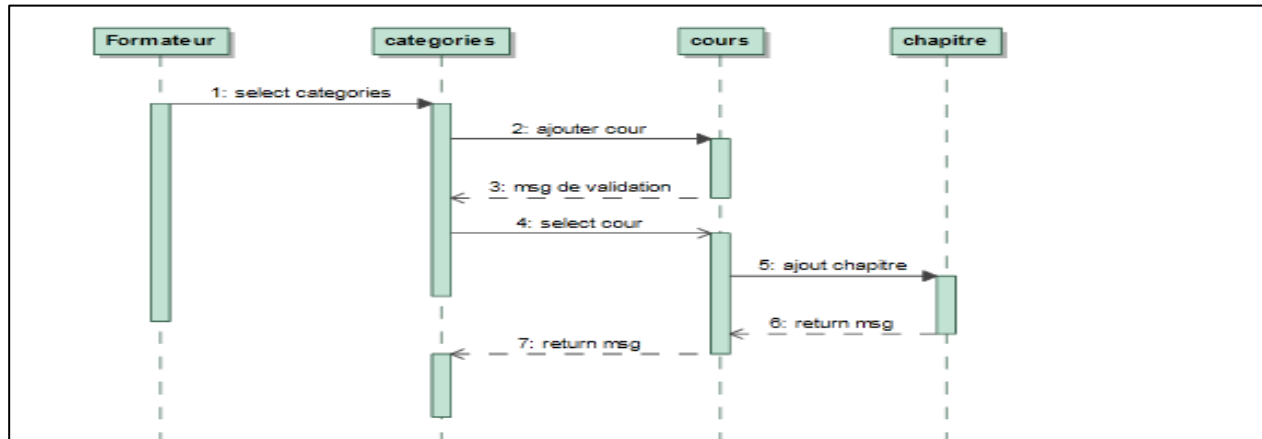


Figure 7 : diagramme de séquence cours

### Commenter chapitre :

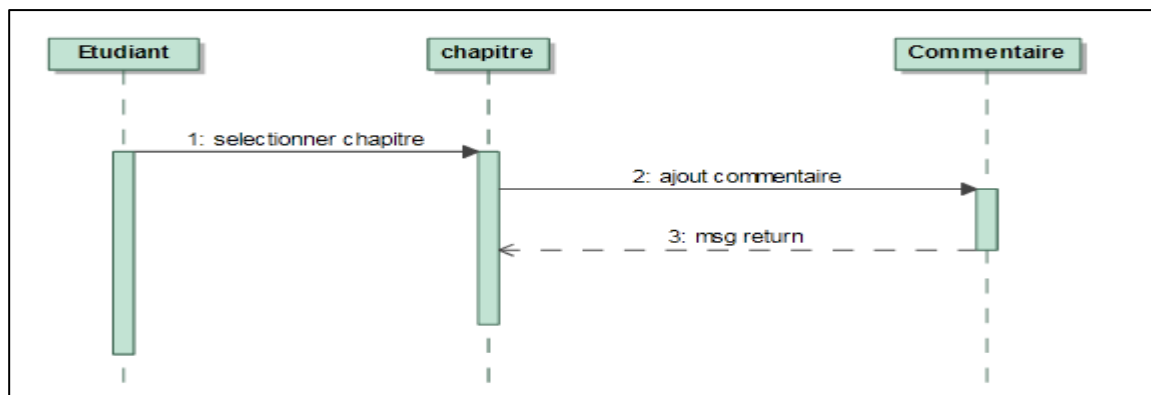


Figure 8 : diagramme de séquence commenter chapitre

**Diagramme d'activité :**

Le **diagramme d'activité** est un diagramme permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multithreads ou multi-processus). Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow).

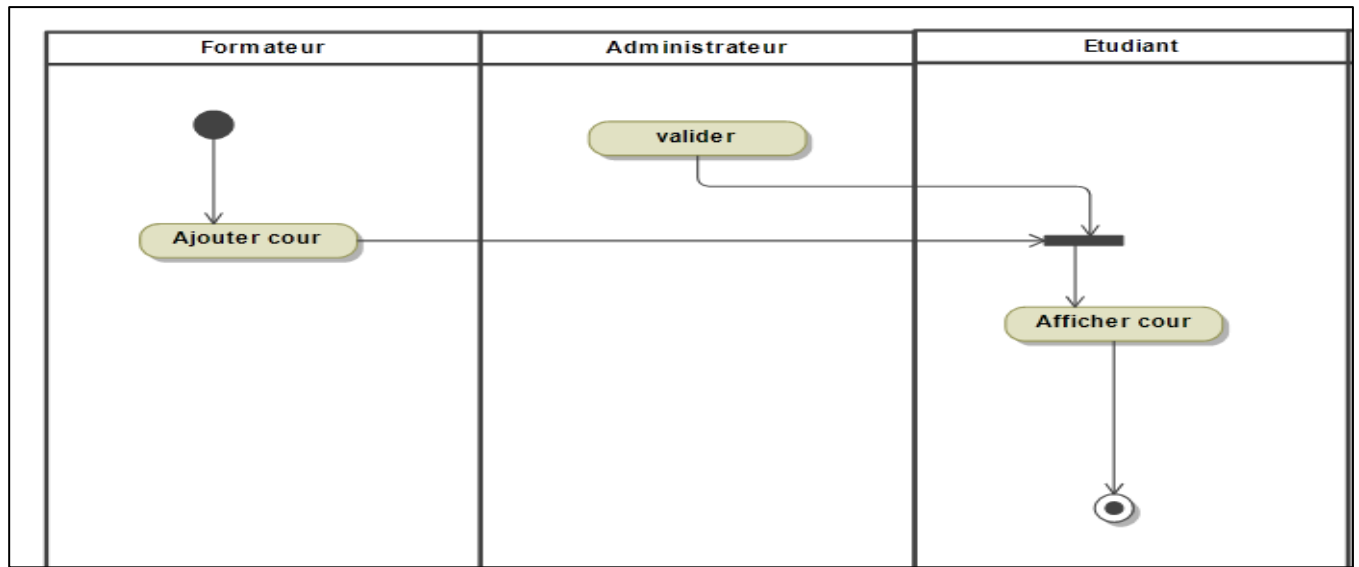
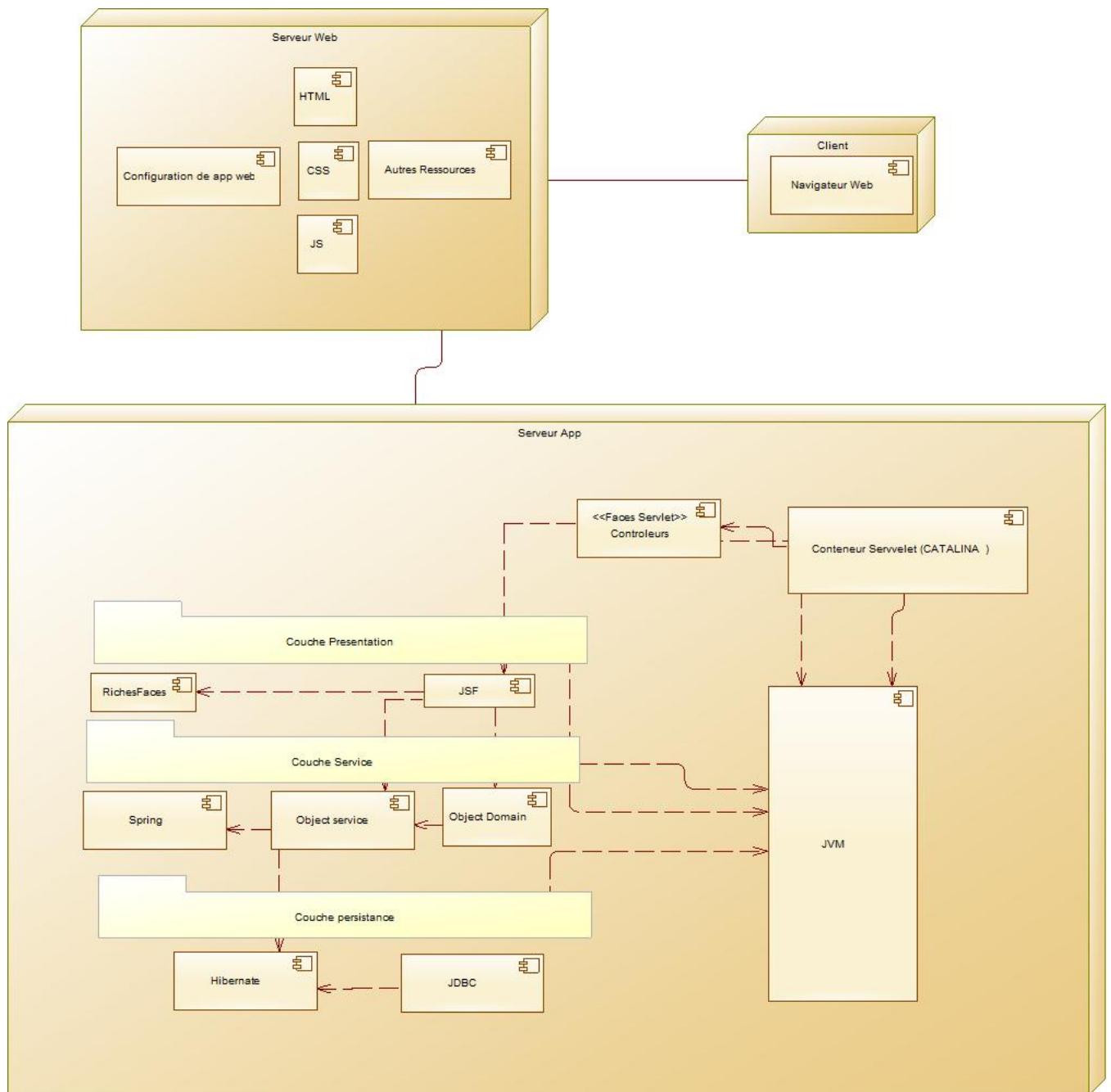


Figure 9 : diagramme d'activité

## Diagramme de déploiement :

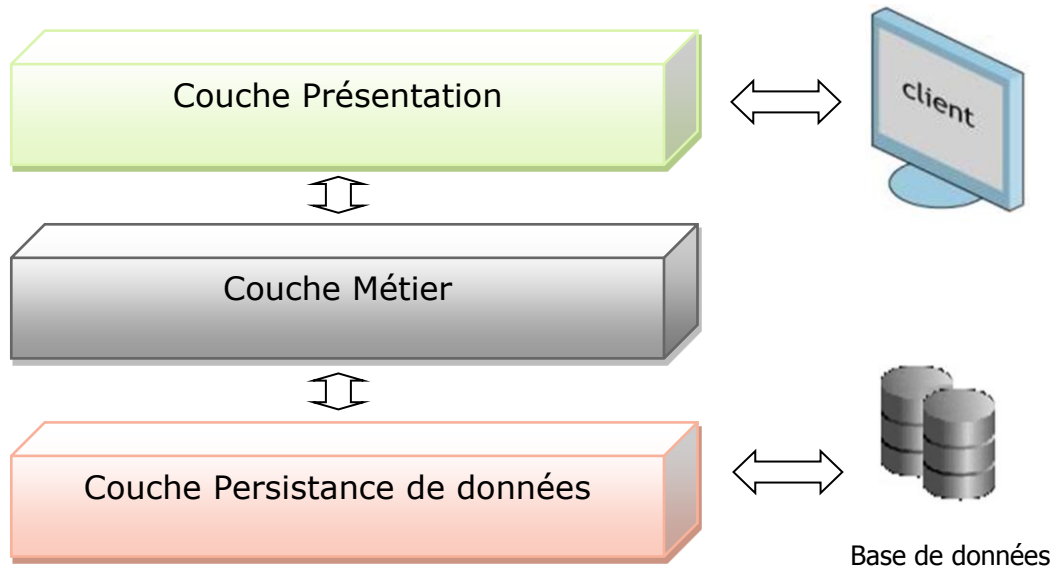
En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.



### III. ARCHITECTURE APPLICATIVE

Notre application est constituée de trois couches.

La figure suivante présente l'architecture Java EE d'une application web en trois couches :

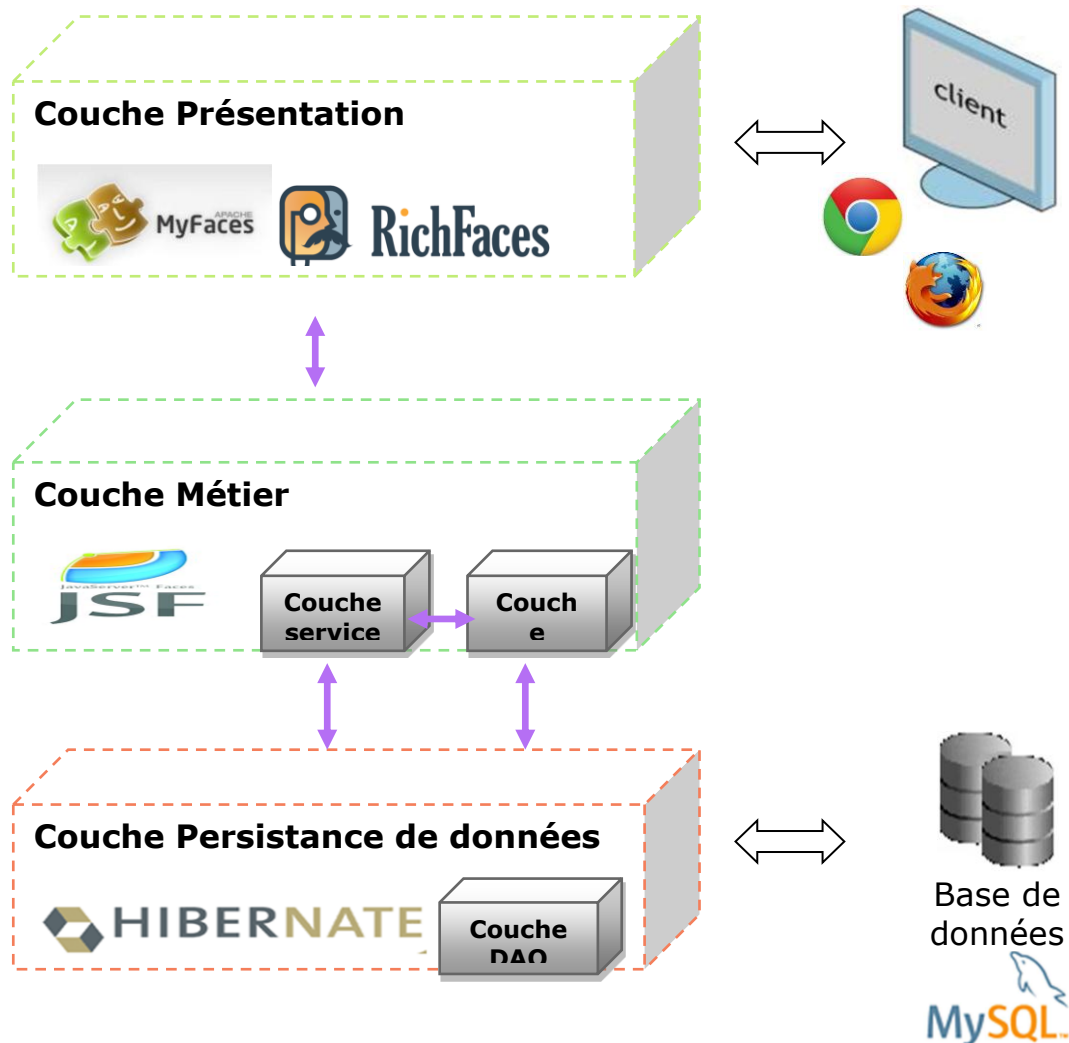


Le concept de partitionner une application en couches et de garder toute la logique de l'application dans ces couches distinctes et séparées, a été introduite bien avant l'approche orientée objet. Ainsi une application est divisée en trois couches logiques, chacune traitant des fonctions spécifiques :

- ➔ Présentation : interface usager et présentation.
- ➔ Logique du logiciel à produire (besoins, services de l'entreprise) : les règlements de l'entreprise et la logique de l'application.
- ➔ Logique des données : Base de données et intégration des services de l'entreprise.

Ce concept nous permet de créer des composants indépendants et de les déployer sur des plates-formes différentes. En fait, ce concept est très utilisé dans le développement des applications multi-tier. Plus tard, il fut adapté au modèle de conception Model-View-Controller (MVC) qui est un modèle très commun pour développer des applications distribuées et multi-tier.

## IV. ARCHITECTURE TECHNIQUE



### 1. CHOIX TECHNOLOGIQUES

#### → JSF

Java Server Faces est un framework de développement d'application Web en Java permettant de respecter le modèle d'architecture MVC et basé sur des composants côté présentation.



## ➔ FACELET

- Technologie de présentation pour le développement d'applications web en Java ;
- Système de templating : permet de définir des vues de JSF en utilisant des calibres du modèle HTML ;
- Réduit la quantité de code nécessaire pour intégrer des composants dans la vue.

## ➔ RICHFACES

Librairie de composants JSF pour le développement d'applications web riches (RIA) Rich Internet Application avec Ajax.

## ➔ SPRING

- Spring est un framework open source J2EE pour applications 3-tiers ;
- Faciliter l'intégration des différentes technologies qui a priori sont pour le moins hétérogènes.

## ➔ HIBERNATE

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle.

## 2. OUTILS :



### **TOMCAT**

L'utilisation d'un serveur Java EE est obligatoire pour le développement de pages Web dynamiques en Java EE. Un serveur HTTP classique reçoit des requêtes HTTP et renvoie des réponses mais il ne connaît pas les Servlets, les JSP... Il est donc essentiel d'utiliser un programme appelé moteur de Servlets qui est contenu dans le serveur Java EE et qui permet de pallier ce manque.



### **ECLIPSE**

Eclipse est l'environnement de développement (spécialisé pour le langage Java) qui sera utilisé dans ce projet. Le choix d'Eclipse repose essentiellement sur sa gratuité, sa facilité d'utilisation, sa puissance de développement et surtout ses nombreux plugins (bibliothèques additives).



### **APACHE**

Apache est le serveur Web le plus utilisé sur Internet. Dans une architecture en production, il est recommandé d'utiliser un serveur Web en frontal d'un

serveur d'applications. Ces recommandations sont également appliquées dans le cas de l'utilisation d'un conteneur Web comme Tomcat.

L'utilisation d'un serveur Web en frontal est nécessaire dans ce projet pour des raisons de performance, de sécurité et de flexibilité.



### **Coupler Tomcat et le serveur web Apache**

L'intégration d'un serveur Tomcat avec un serveur Web se fait au travers d'un connecteur configuré au sein de Tomcat et d'une extension ajoutée au serveur Web. Un connecteur Tomcat est une classe Java qui supporte un protocole réseau spécifique et propriétaire. La librairie d'extension du serveur Web permet un dialogue entre les deux serveurs.



### **MySQL**

Le SGBD MySQL est supporté par un large éventail d'outils. MySQL est surtout installé pour les applications Web, ce SGBD est solide et utilisé par de grands groupes spécialisés dans l'Internet. Plusieurs pilotes natifs de type 4 sont disponibles pour MySQL et sont conseillés pour une utilisation en Java.