



INTI
FORMATION

INSTITUT DES NOUVELLES TECHNOLOGIES DE L'INFORMATION

Support de projet

Développement d'une application de gestion d'une école

Contexte technologique

JAVA – J2EE – Tomcat – SPRING (MVC et IoC) – SPRING SECURITY – Hibernate – MySQL.

I. Objectif

l'établissement gère son personnel (administratif et enseignant) ainsi que ses étudiants, cours et matières. L'établissement exerce son métier avec des fiches papier, des envois de documents par e-mail ou par courrier postal. Il n'arrive plus à gérer manuellement son expansion et souhaite utiliser un Système d'Information pour lui permettre de répondre à cette croissance et pour gagner en souplesse de travail.

L'école attend plusieurs services du Système d'Information comme la gestion dynamique des personnes, la gestion des affectations des enseignants aux matières et aux cours, la gestion des étudiants, de leur promotion et les affectations des élèves à leurs cours, en gérant les absences.

II. Besoins fonctionnels

Nous voulons développer un projet de gestion d'une école (gestion des étudiants, enseignants, cours, matières...).

L'application permet :

- De gérer des utilisateurs/personnes pour les enseignants ou les étudiants (liste, création, modification, suppression, authentification) ;
- De gérer des cours/matières (liste, création, modification, suppression, tri) ;
- De gérer des sessions pour l'authentification et la navigation ;
- De gérer des langues (site multilingues) ;
- De gérer des exercices associés à des cours ;
- De gérer des services d'envoi par e-mail ou services Web.

III. Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur. pour cela, il faut répondre aux exigences suivantes :

- **Fiabilité** : L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.
- **Les erreurs** : Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre site web.
- **Ergonomie et bonne Interface** : L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.
- **Sécurité** : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les sites web.
- **Aptitude à la maintenance et la réutilisation** : Le système doit être conforme à une architecture Trois tiers et claire permettant sa maintenance et sa réutilisation.

IV. Modélisation du SI

A. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permettant de représenter les fonctionnalités du système du point de vue utilisateur se compose :

- D'acteurs (entités externes humaines ou robot/matériel utilisant le système).
- De cas d'utilisation (fonctionnalités proposées par le système).

Les acteurs utilisant le système sont :

- **Administrateur** : les employés de l'administration peuvent, après authentification, gérer :
 - les personnes (administrateurs, enseignants, étudiants).
 - les matières (Informatique, Management, Mathématiques...).
 - les promotions et l'affectation des étudiants aux promotions.
 - les cours et l'affectation des cours aux matières.
 - les absences des étudiants aux cours.

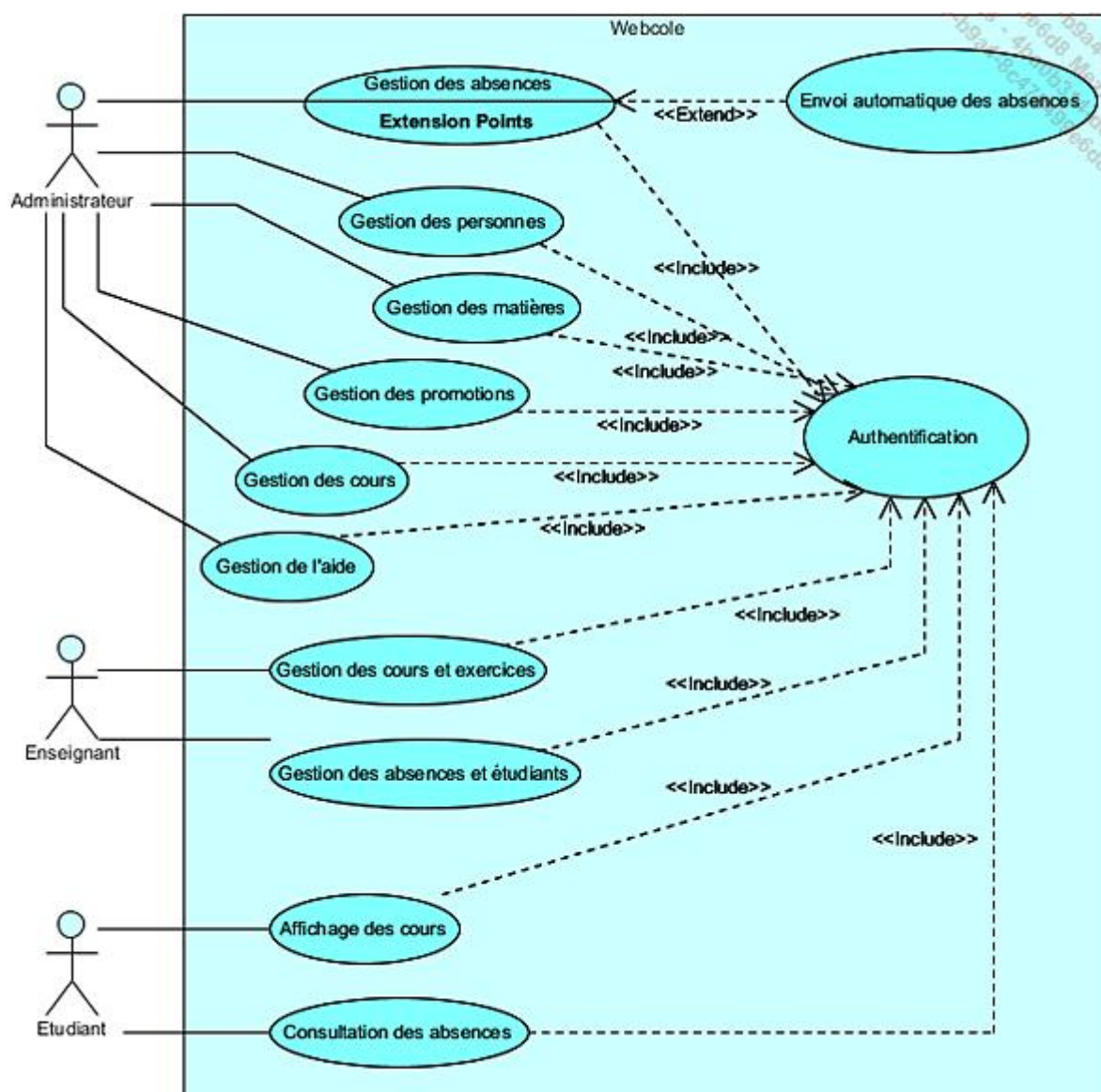
- **Enseignant** : les enseignants peuvent, après authentification :

- ajouter des cours et exercices associés.
- afficher les cours par promotion, matière et date.
- gérer les absences des élèves à leurs cours.
- afficher un récapitulatif des absences des étudiants.

- **Étudiant** : les étudiants peuvent, après authentification :

- consulter la liste des cours et les informations associées.
- afficher un tableau récapitulatif de leurs absences.

Le projet est limité à ces services, mais pourrait également gérer les retards, le cahier de textes, les motifs des absences ou autres.



Le diagramme des cas d'utilisation peut être lu comme ceci :

- Un administrateur peut gérer les absences des étudiants et enseignants, gérer les personnes (administrateurs, étudiants et enseignants), gérer les matières, gérer les cours pour les matières (Java SE : les classes, Java SE : les IHM, Java EE : les Servlets...) et gérer les promotions (BTS 1, BTS 2, Licence, Master 1, Master 2...).
- Un enseignant peut créer un nouveau cours et des exercices associés. Il peut également consulter la liste des étudiants absents.
- Un étudiant peut afficher la liste de ses cours et exercices par matière et consulter la liste de ses absences.

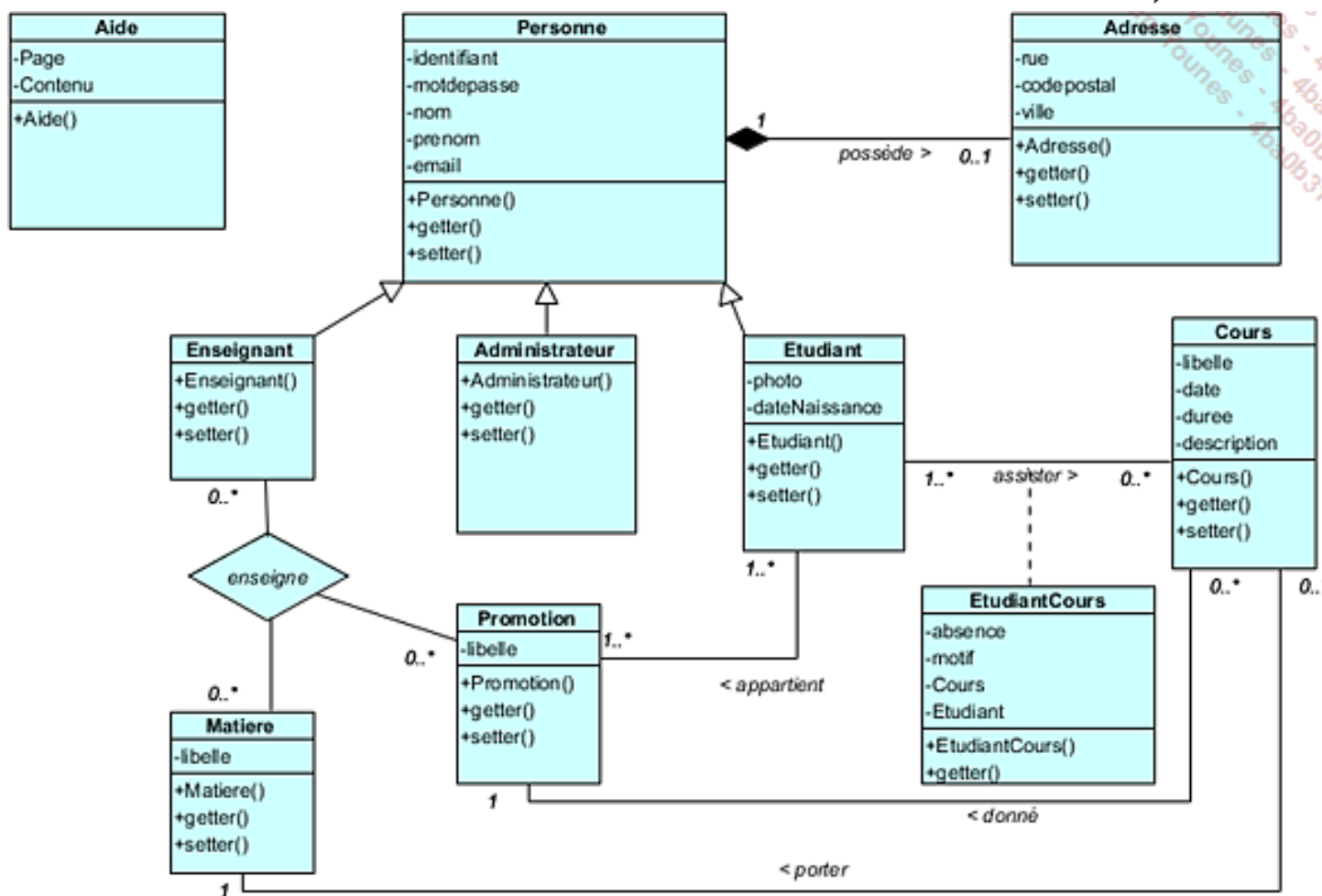
B. Diagramme de classes du projet

Le diagramme de classes présente la classe mère **Personne** permettant de gérer les administrateurs, les étudiants et les enseignants possédant éventuellement une **Adresse**.

La classe **Cours** concerne une **Matiere** et une **Promotion**.

Les enseignants peuvent être responsables d'une matière et les étudiants appartiennent à une **Promotion**.

Les présences et absences des étudiants aux cours sont gérées par l'association **EtudiantCours** permettant de conserver l'absence à l'aide d'un booléen et le motif de l'absence si elle existe.



V. Maquettes de la plate-forme

NB : On vous laisse le choix concernant les maquettes des écrans Web, de choisir les feuilles de styles ainsi que les thèmes.

Les maquettes d'écran facilitent la compréhension des cas d'utilisation. Les utilisateurs se repèrent facilement grâce à ces schémas visuels et peuvent ainsi valider les choix d'analyse.

Le projet sera découpé selon le modèle suivant avec un en-tête, un menu de navigation et un pied de page.



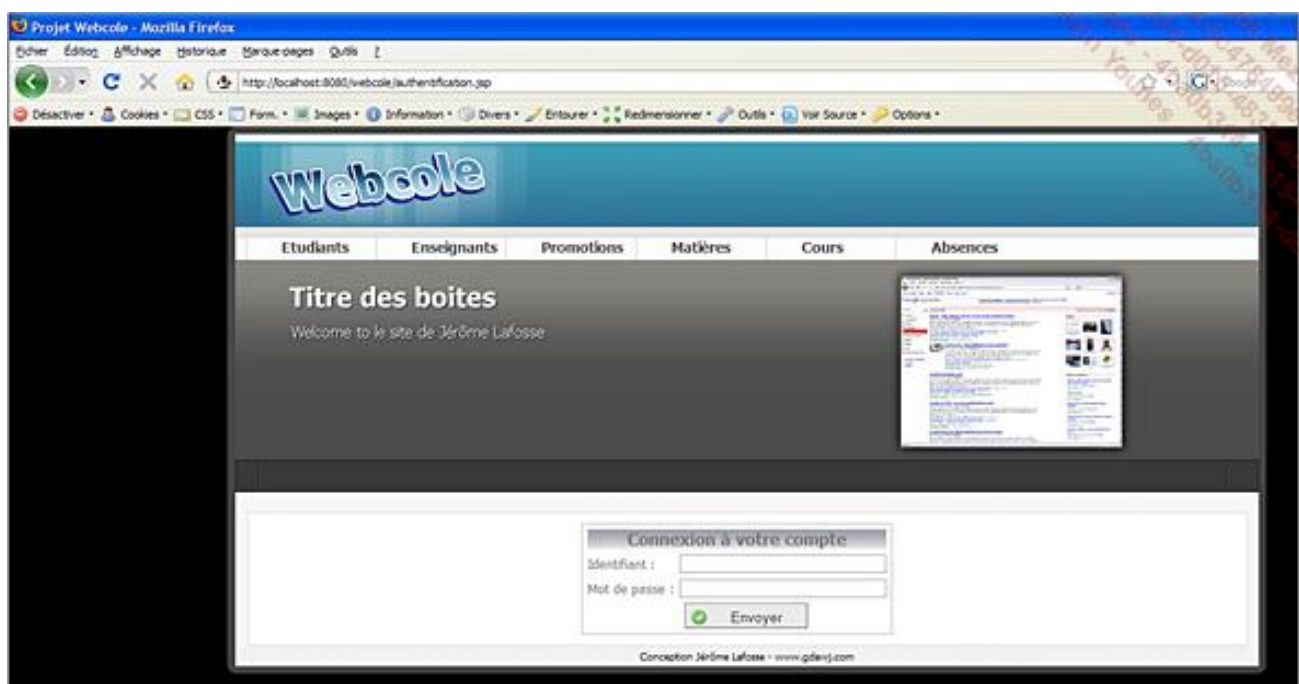
A. Authentification

Cette page permet au personnel (administrateur, étudiant ou enseignant) de se connecter et se déconnecter du système.

Le personnel saisit alors son identifiant et son mot de passe.

Si l'authentification est correcte, l'utilisateur est redirigé vers la page d'accueil et un message d'invitation est alors affiché sur la totalité des pages du site jusqu'à la prochaine déconnexion.

Ce message permet d'afficher l'identifiant de l'utilisateur actuellement connecté.



B. Gestion de l'aide

Ce service permet de gérer un système dynamique d'aide en ligne pour le projet.

