Rapport du projet JEE

Un annuaire des anciens étudiants du Master

Nabil Bernine, Kiril Gashteovski, Maximilien Perrad.*

2011 - 2012

1 Introduction

Le projet s'inscrit dans l'unité d'enseignement *Architecture JEE*. Il s'agit de réaliser une application pour gérer un annuaire des anciens étudiants du Master en utilisant la technologie *J2EE*.

2 Cahier des charges

L'objectif de ce projet gérer à l'aide de la technologie J2EE un annuaire des étudiants du Master deuxième année. De manière plus précise, cette annuaire doit respecter un certain nombre de règles :

- Chaque étudiant est doté d'un mot de passe qui l'authentifie ;
- Les étudiants sont regroupés dans des promotions. Une promotion comporte une année, un titre, un ID et deux ou trois paragraphes pour la décrire ;
- En s'inspirant de l'annuaire actuel du Master, on récupère les informations à gérer (nom, prénom, adresse postale, adresse électronique, site WEB, nom de la société, site WEB de la société, date de naissance, etc.) ;
- Les étudiants peuvent modifier les informations qui les concernent en donnant leur mot de passe ;
- En cas d'oubli, les étudiants peuvent recevoir un mail pour leur rappeler leur mot de passe ;
- Un administrateur (qui a son propre mot de passe) peut détruire, modifier ou ajouter des entrées dans l'annuaire (c'est à dire des étudiants et des promotions).
  L'ajout de plusieurs étudiants dans une promotion doit être réalisé en une seule opération.

*M2 I2A-ARO

3 Réalisation de l'application

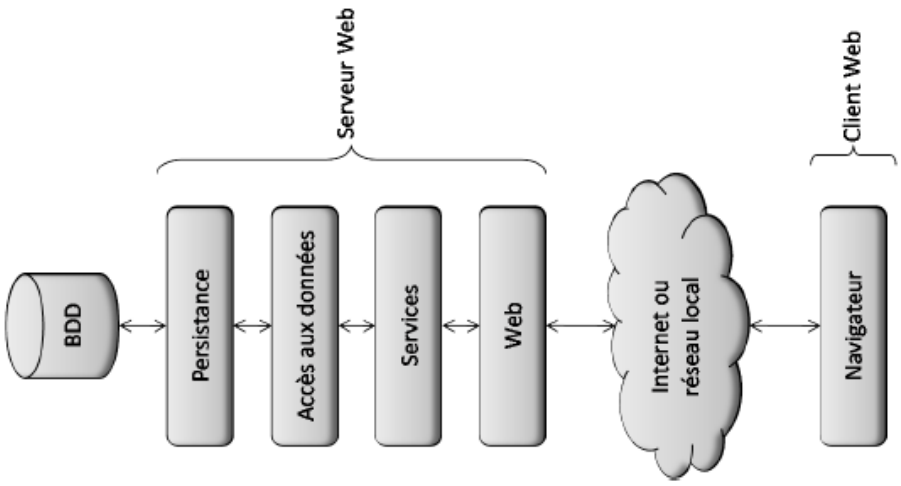

FIG. 1 – Structure de l'application

3.1 Base de données

La base de données se compose de trois (3) tables : *Promotion*, *Etudiant* et *Administrator*, définies comme suit :

1. Une table Promotion :

Field	Type	Null	Key	Default	Extra
Prom_ID	bigint(20)	NO	PRI	NULL	auto_increment
BeginingYear	int(11)	NO		NULL	
OptInitial	varchar(3)	NO	MUL	NULL	
OptNom	varchar(100)	NO		NULL	
SpecInitial	varchar(3)	NO		NULL	
SpecNom	varchar(100)	NO		NULL	
Anne	varchar(9)	NO		NULL	

TAB. 1 – Promotion

Composée d'un identifiant *Prom_ID* (clé primaire de la table), de l'année de début de la promotion *BeginingYear*, le sigle définissant la promotion *OptInitial*, dénomination de la promotion *OptNom*, le sigle définissant la spécialité *SpecInitial*, dénomination de la spécialité *SpecNom* et l'année universitaire *Anne*.
Notons que *OptInitial* et *Anne* définissent une promotion de manière unique. Aussi, tout les champs sont obligatoires.

2. Une table Etudiant :

Field	Type	Null	Key	Default	Extra
E_ID	bigint(20)	NO	PRI	NULL	auto_increment
Comentaires	varchar(200)	YES		NULL	
Couriel	varchar(50)	YES	UNI	NULL	
Nom	varchar(100)	NO		NULL	
PromID	int(11)	NO		NULL	
Web	varchar(50)	YES		NULL	

Tab. 2 – Etudiant

Définissant pour chaque étudiant, un identifiant *E_ID* (clé primaire de la table), un commentaire *Comentaires* (optionnel), l'adresse électronique *Couriel* (optionnel), un nom *Nom*, l'identifiant de sa promotion *PromID* pour accéder aux informations de sa promotion à partir de la table précédente et un site web *Web* (optionnel).

3. Une table Administrator pour gérer les administrateurs de l'application :

Field	Type	Null	Key	Default	Extra
AdminID	bigint(20)	NO	PRI	NULL	auto_increment
AdminPass	varchar(30)	NO		NULL	
AdminUser	varchar(30)	NO	UNI	NULL	

Tab. 3 – Administrator

Un administrateur est donné par un identifiant *AdminID* (clé primaire de la table), son mot de passe *AdminPass* et nom d'utilisateur unique à chaque administrateur *AdminUser*. Tous ces champs sont obligatoires.

3.2 Persistance des données

La persistance des données est assurée par le composant JPA.

3.3 La couche d'accès aux données (DAO)

La couche d'accès aux données (également appelée DAO pour Data Access Object) chargée de l'accès aux données et de leur manipulation, indépendamment du SGBD choisi.
Les méthodes implémentées pour la manipulation de ces données sont :

Ajout étudiant

- *addEtudiant(String name, String com, String email,String webSite, int promID) ;* Ajout d'un étudiant avec toute ses propriété comme arguments ;
- *addEtudiant(Etudiant e) ;* Ajout d'une ligne étudiant à la base de données.

Recherche étudiant

- *findEtudiantById(long id) ;* Recherche par identifiant ;
- *findEtudiantByName(String name) ;* Recherche par nom d'étudiant ;
- *findEtudiantByPromotion(int promID) ;* Recherche par promotion ;
- *findEtudiantByEMail(String eMail) ;* Recherche par adresse électronique ;
- *findAllEtudiants() ;* Trouver tous les étudiants triés par leur promotion.

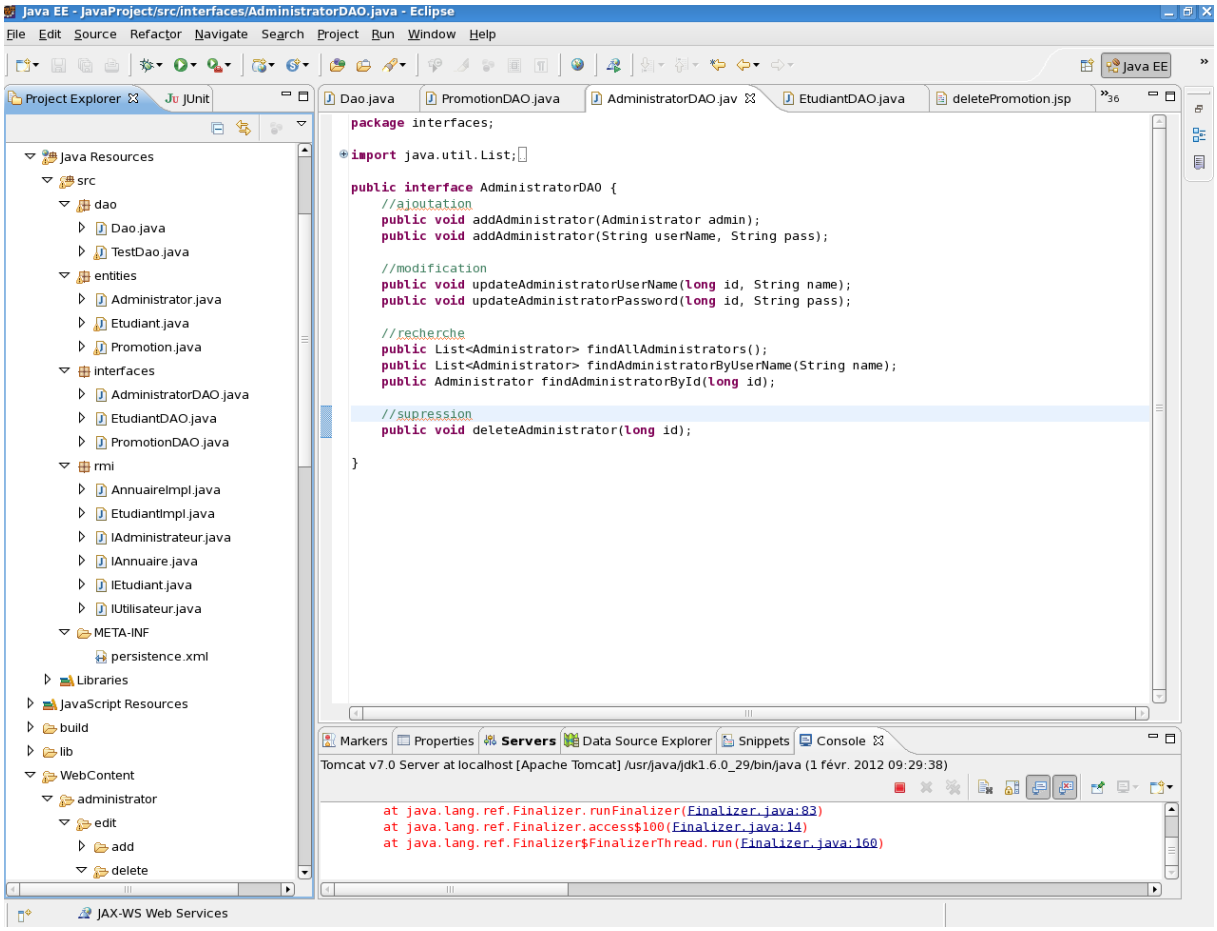

FIG. 2 – Aperçu du package explorer de l'application

Mise à jour étudiant

- *updateEtudiantName(long id, String name) ;* Mettre à jour le nom de l'étudiant ;
- *updateEtudiantEMail(long id, String eMail) ;* Mettre à jour l'adresse électronique de l'étudiant ;
- *updateEtudiantComments(long id, String comment) ;* Mettre à jour le commentaire sur l'étudiant ;
- *updateEtudiantPromID(long id, int PromID) ;* Mettre à jour l'identifiant de la promotion de l'étudiant ;
- *updateEtudiantWeb(long id, String WebSite) ;* Mettre à jour le site web de l'étudiant.

Suppression étudiant

- *deleteEtudiant(long id) ;* Supprimer les données d'un étudiant dans la base à partir de son identifiant.

Ajout promotion

- *addPromotion(String optInit, String optName, String specInit,String specName,*

String year, int debAnne) ; Ajout d'une promotion avec toute ses propriétés ;
- *addPromotion(Promotion p)* ; Ajout d'une promotion prédéfinie.

Recherche promotion

- *findPromotionById(long id)* ; Par identifiant (clé) ;
- *findPromotionByOptionInitial(String name)* ; Par nom ;
- *findPromotionBySpecialiteInitial(String name)* ; Par sigle de la spécialité ;
- *findAllPromotions()* ; Trouver toute les promotions triées par leurs sigles ;
- *findPromotionByBegYear(int year)* ; Donne une liste des promotion qui ont la même année de début (recherches par *beginingyear*) ;
- *findPromotionByOptBegYear(opt String, int year)* ; Retourne une liste des promotions qui ont un sigle de spécialité donné et année de début de promotion spécifiée (par *OptionInitial* et *BeginingYear*) ;
- *findAllPromotionsSortedByYear()* ; Retourne une liste de toutes les promotions dans la base de données, qui sont triés par année de début *BeginingYear* (le même que findAll-Promotions () mais classés par année début).

Mise à jour promotion

- *updatePromotionOptInitial(long id, String optionInitial)* ; Du sigle de l'option ;
- *updatePromotionOptNom(long id, String optionNom)* ; De la dénomination de la promotion ;
- *updatePromotionSpecInitial(long id, String specialiteInitial)* ; Du sigle de la spécialité
- *updatePromotionSpecNom(long id, String speciaNom)* ; De sa dénomination ;
- *updatePromotionYear(long id, String year)* ; De l'année de la promotion
- *updatePromotionStartYear(Identifiant de long, int year)* ; Mises à jour de l'année de début de la promotion.

Suppression promotion

- *deletePromotion(long id)* ; Suppression d'une promotion par son identifiant.

Ajout administrateur

- *addAdministrator(Administrator admin)* ; Ajout d'un administrateur prédéfinie ;
- *addAdministrator(String userName, String pass)* ; Ajout d'un administrateur en insérant les champs le définissant.

Mise à jour administrateur

- *updateAdministratorUserName(long id, String name)* ; Mise à jour du nom de l'administrateur ;
- *updateAdministratorPassword(long id, String pass)* ; Mise à jour de son mot de passe.

Recherche administrateur

- *findAllAdministrators()* ; Trouver tous les administrateurs triés par leur nom d'utilisateur ;
- *findAdministratorByUserName(String name)* ; Chercher les administrateurs par leur noms d'utilisateur ;

- *findAdministratorById(long id) ;* Par leur identifiant.

Suppression administrateur

- *deleteAdministrator(long id) ;* Supprimer un administrateur en utilisant son identifiant.

3.4 Application Web

L'arborescence de l'application web est donné par la figure suivante (FIG.3) :

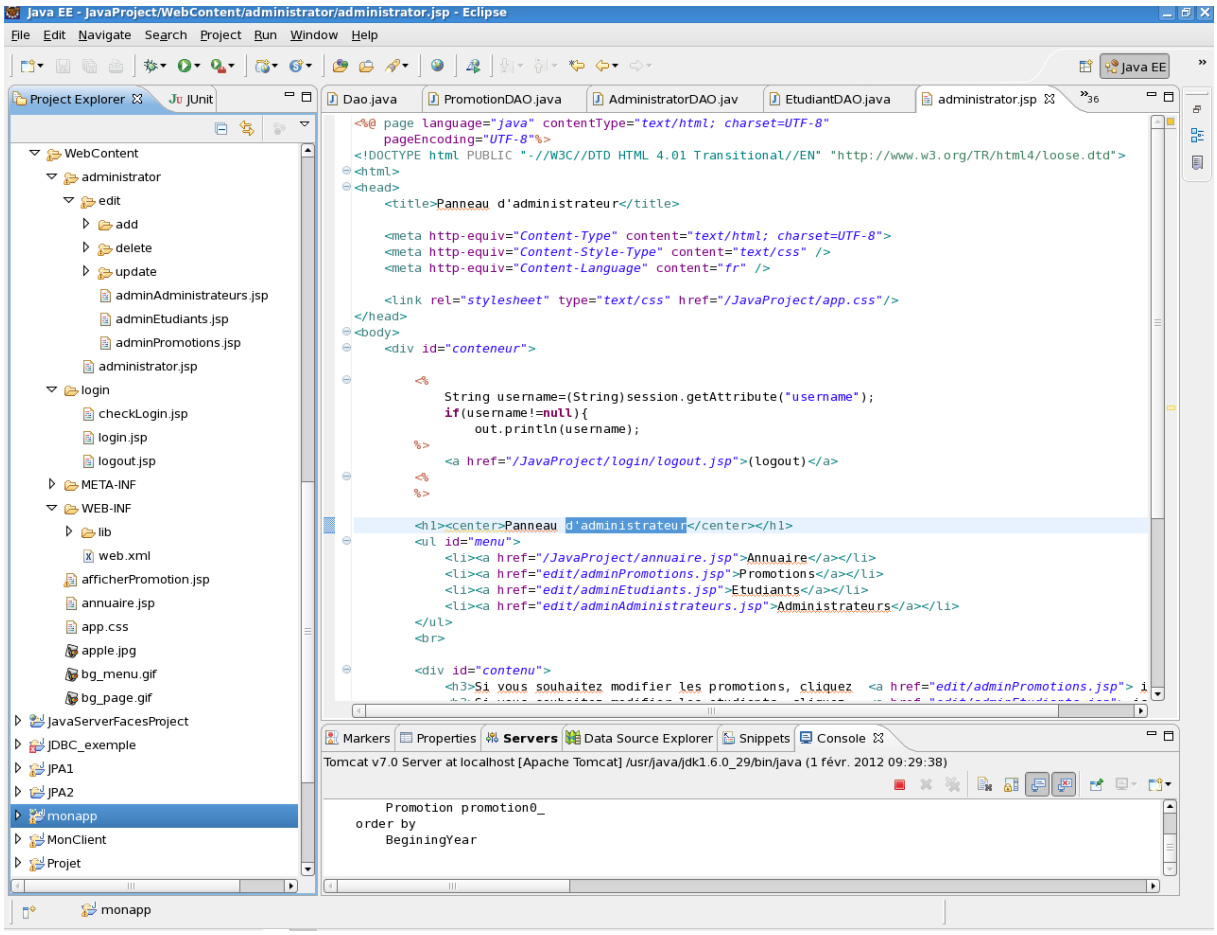

FIG. 3 – Arborescence de l'application web

Dans le dossier *Administrator*, sont placés toute les pages visibles uniquement par les utilisateur connectés en tant qu'administrateur. La page *administrator.jsp* est le panneau de l'administrateur.

Dans le dossier *Edit*, on a les pages suivantes :
adminAdministrateurs.jsp, c'est une page avec des choix pour changer les données des administrateurs (ajout, suppression, mise à jour) ;
adminEtudiants.jsp, c'est une page avec des choix pour changer les données des étudiants (ajout, suppression, mise à jour) ;

adminPromotions.jsp, c'est une page avec des choix pour changer les données des promotions (ajout, suppression, mise à jour).

Ainsi que les sous dossiers suivant :

add, contient les pages qui ajoutent les étudiants, les promotions et les administrateurs ;
delete, contient les pages qui suppriment les étudiants, les promotions et les administrateurs ;
update, contient les pages qui mettent-à-jour les étudiants, les promotions et les administrateurs.

Dans le dossier *login* sont placées toutes les pages qui sont liées à la connexion / déconnexion des processus (checkLogin.jsp, login.jsp, logout.jsp).
Aussi, dans le dossier parent du contenu web, on a placé un fichier *app.css*, qui est le CSS utilisé pour une bonne présentation de notre application. Et toute les librairies externe utilisées par l'application sont placées dans le répertoire *WEB-INF/lib*.

Quand on lance l'application, on on obtient la vue suivante (FIG.4). Chacun peut voir la première page *annuaire.jsp*.

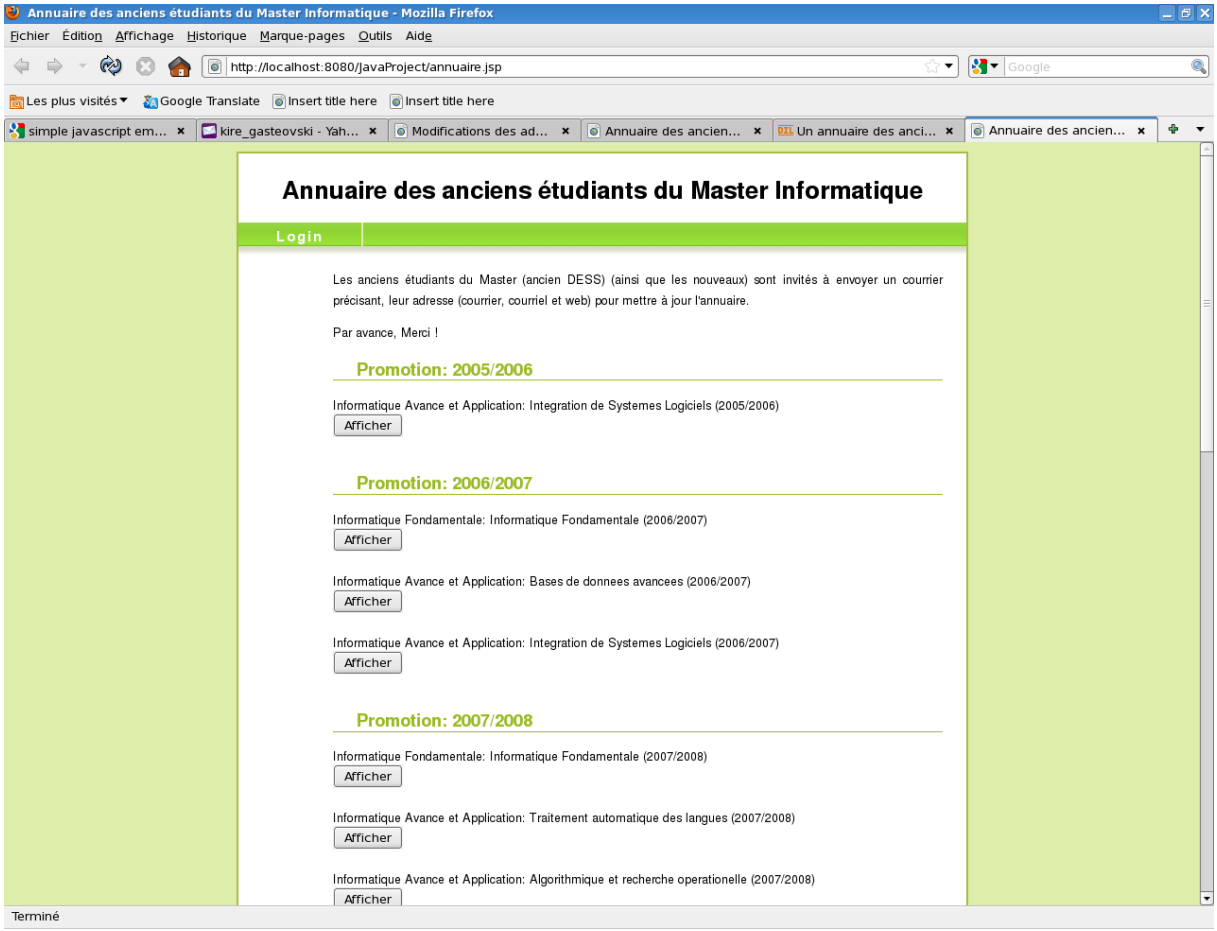

FIG. 4 – Lancement de l'application web

Dans la première page sont présentés les promotions triés par leur année. Si quelqu'un clique sur le boutons d'une promotion donnée, l'application va à la page *afficherPromotion.jsp* et tous les élèves et leurs informations sont affichés à partir de cette promotion. Aussi, un menu *Annuaire* est inclus, si le client clique dessus, il remonte à *annuaire.jsp*.

Seul un administrateur peut se connecter pour voir le panneau administrateur et peut modifier les données pour les étudiants, les promotions et même d'autres administrateurs (ou de changer ses propres données).

Si un administrateur veut se connecter, il doit aller sur le menu *Connexion* qui le renvois à la page de *connexion/login.jsp*, dans lequel on peut voir un journal sous forme de la figure suivante (Fig.5). L'administrateur renseigne son nom d'utilisateur et mot de passe.

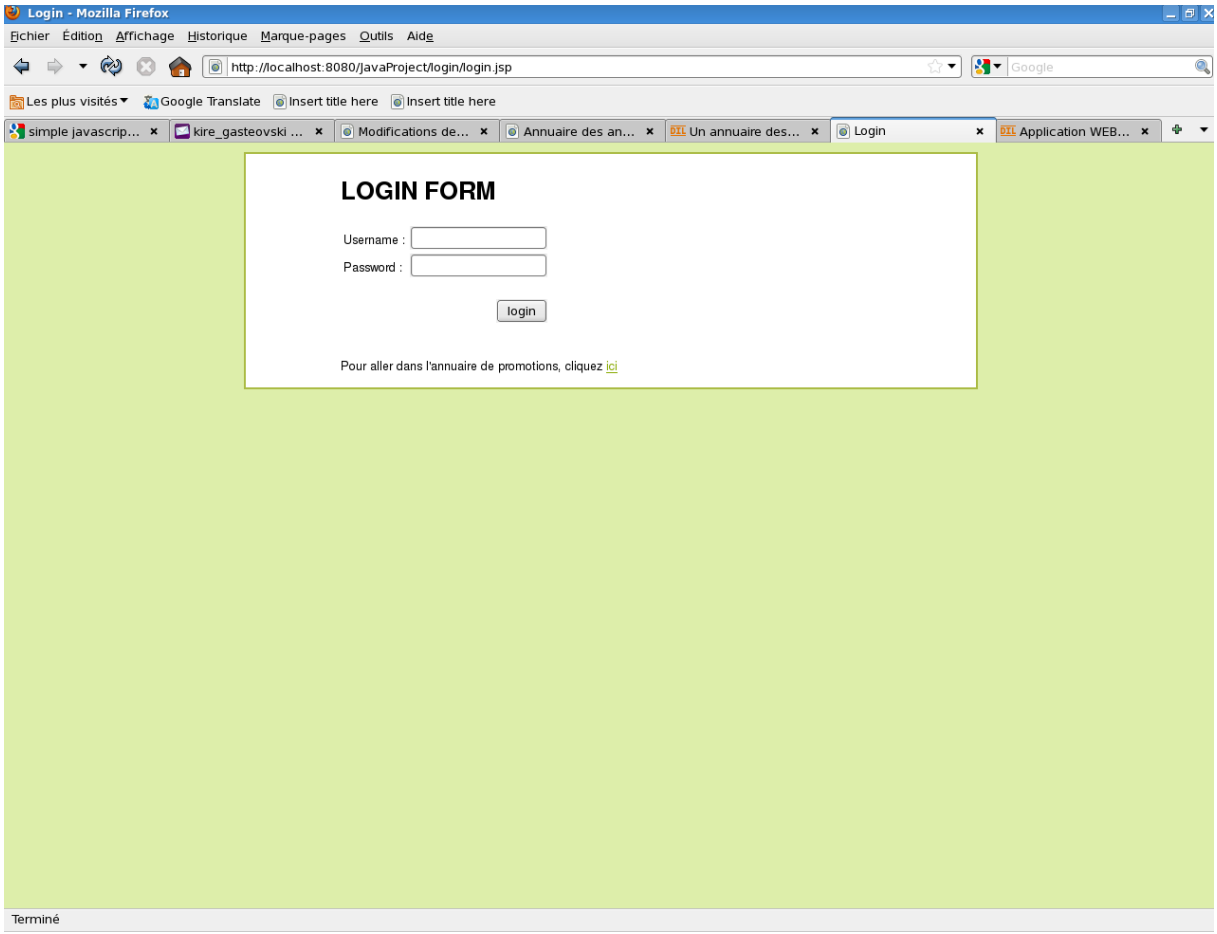

Fig. 5 – Vue d'administrateur

À partir de ce menu, on peut revenir en arrière dans le menu *Annuaire* en cliquant sur le bouton lien *ici* du message : « Pour aller dans l'annuaire de promotions, cliquez ici ».

Lorsque l'administrateur soumet les données identifiant et mot de passe, il est renvoyé à la page *login/checkLogin.jsp* et les données présentées sont vérifiées avec la base de données (*dans la table Administrateur*).
Si le nom d'utilisateur et mot de passe ne correspondent pas avec toutes les données dans la table *Administrateur*, alors la page *checkLogin.jsp* donne le message suivant : « Vérification de connexion Nom et mot de passe invalide ! Retour ».
Le lien *Retour* nous ramène à la première page *annuaire.jsp*. Si l'administrateur existe dans la base de données, alors la page *checkLogin.jsp* montre le message suivant : « Vérification de connexion Bienvenu [username] ! Retour ». Le lien *Retour* renvoie à la première page *annuaire.jsp*.

Lorsque l'administrateur est connecté, on peut voir un nouveau menu *Admin* dans la première page *annuaire.jsp* dans le panneau d'administrateur et également dans le coin supérieur gauche on peut voir le nom de l'administrateur et un lien [*déconnexion*] (voir FIG.6).

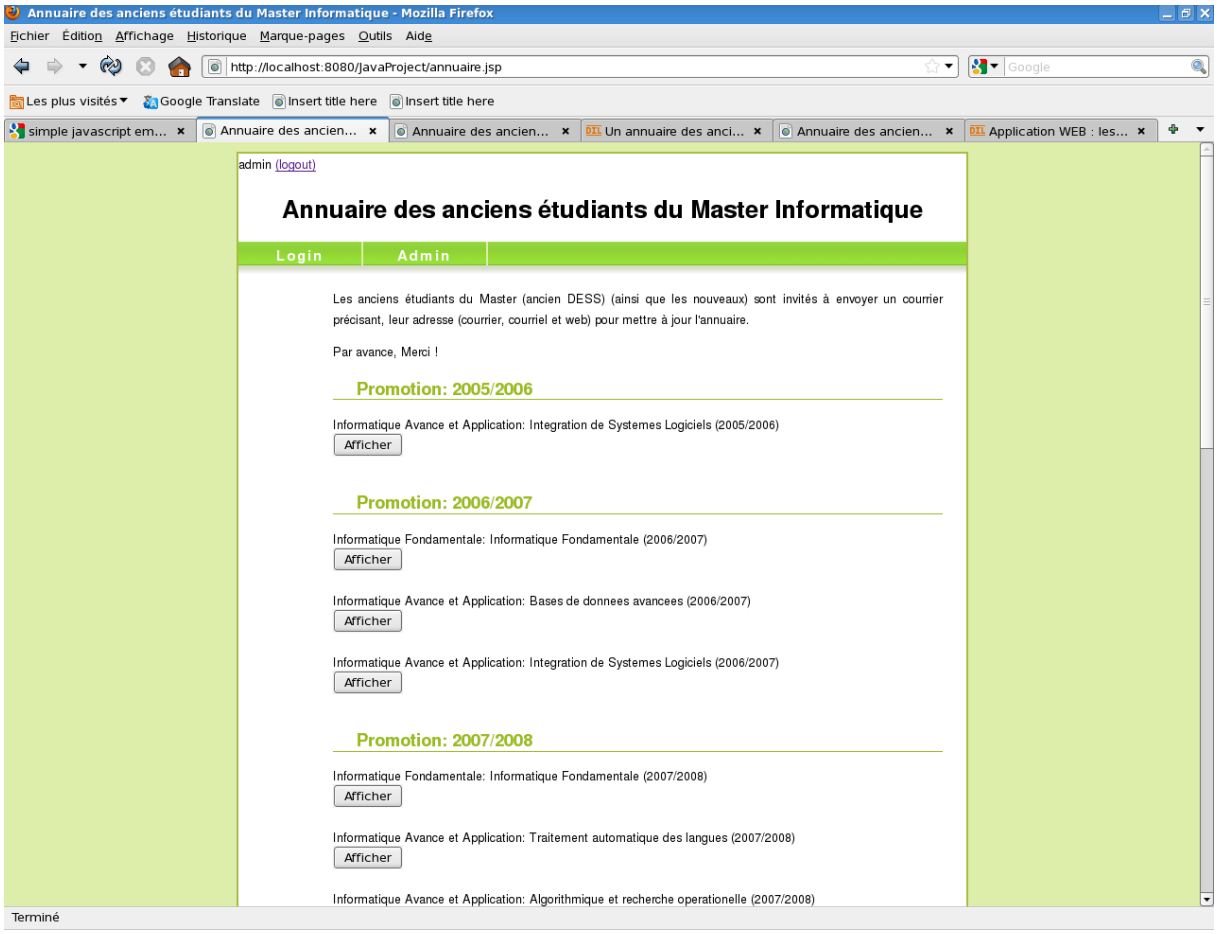

FIG. 6 – Vue d'administrateur

Cela signifie que lorsque l'administrateur veut aller dans le panneau administrateur pour modifier les données, il devra cliquer sur le menu *Admin*. Si un administrateur veut se déconnecter, il va tout simplement cliquer sur le lien *Déconnexion* dans le coin supérieur gauche, à côté de son nom d'utilisateur administrateur. Après avoir cliqué sur *déconnexion*, il va à la page *login/logout.jsp* et l'application donne le message suivant : « [admin username] déconnecté », ce qui signifie que l'administrateur a été déconnecté et désormais les pages dans le dossier *administrateur* sont indisponibles.

Le panneau d'administrateur ressemble à la figure suivante(FIG.7).

Comme menus nous avons : *Annuaire*, *Promotions*, *Etudiants*, *Administrateurs*. Ils renvoient vers les pages suivantes, respectivement :

Annuaire *annuaire.jsp* est la première page de l'application ;

Promotions *administrator/edit/adminPromotions.jsp* La page d'administration des promotions peut être consulté également par le lien *ici* dans le message suivant « Si Vous souhaitez mo-

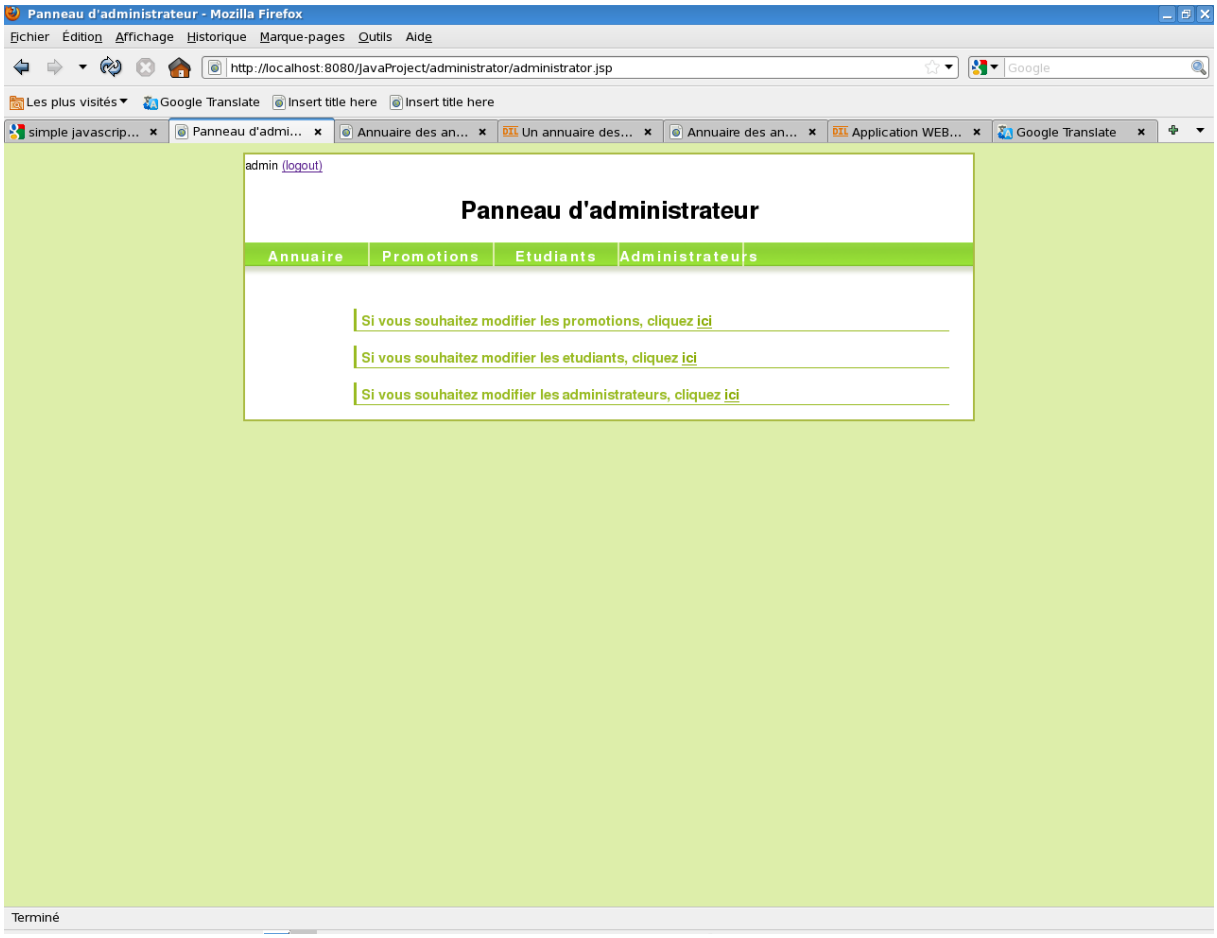

FIG. 7 – Vue d'administrateur

dificateur les promotions, cliquez ici » ;

Etudiants *administrator/edit/adminEtudiants.jsp* La page d'administration des étudiants peut être accessible par le lien *ici* dans le message suivant « Si Vous souhaitez modificateur les etudiants, cliquez ici » ;

Administrateurs *administrator/edit/adminAdministrateurs.jsp* La page d'administration des administrateurs peut aussi être accessible par le lien *ici* dans le message suivant « Si Vous souhaitez modificateur les administrateurs, cliquez ici » ;

En cliquant sur chacun de ces menus, l'administrateur peut ajouter, modifier ou supprimer des données pour les étudiants, les promotions et les administrateurs.

4 Conclusion

L'application a été créée avec *jdk1.6_029* et elle fonctionne avec. Aussi, au cours du développement de ce projet, nous avons remplacé le *JDK* avec la nouvelle version *jdk1.7.0_02*. Nous l'avons testé, et l'application marche aussi avec ce *JDK*, donc nous avons décidé de le changer dans le *classpath*.