

PROJET CAE

Rapport final

Tables des matières

MISE EN OEUVRE DU PROJET
RESPONSIVE DESIGN
INTERFACES
TESTS UNITAIRES
DIFFICULTES RENCONTREES
CONCLUSION

MISE EN OEUVRE DU PROJET

Nous allons présenter dans ce document **l'ensemble des ajouts effectués** par rapport aux fonctionnalités détaillées dans les documents précédents. L'application développée est une plate-forme de vente d'article de presse. **Cette application permet via une interface intuitive et plaisante de mettre en relation des rédacteurs d'articles de presse et des lecteurs** afin que chacun puisse tirer avantage de notre site.

Dans l'application réalisée, les utilisateurs ont la possibilité de se procurer des articles à l'unité selon un thème particulier ou un rédacteur préféré ou bien de s'abonner afin d'obtenir un accès illimité aux articles présents sur notre plate-forme.

Notre site accueille les journalistes, écrivains ou passionnés d'écriture souhaitant s'exprimer via la rédaction d'articles dans le domaine de leur choix.

L'ensemble des articles proposés sur notre site est soumis au **contrôle de rédacteurs en chef** dont le rôle est de s'assurer que les articles respectent nos normes de qualités rédactionnelles, ainsi que notre ligne éditoriale. Il est en effet primordial pour nous de nous assurer que les articles proposés via notre plate-forme sont correctement rédigés, relatent une véracité journalistique et permettent de voir s'exprimer la liberté d'expression des rédacteurs tout en restant dans le cadre légal.

Cet encadrement nous permet d'assurer aux utilisateurs des **articles de qualité** dans de **nombreux domaines** afin de satisfaire le plus grand nombre.

Une équipe d'administrateurs oeuvre également quotidiennement sur notre site via une interface afin de s'assurer que nos visiteurs jouissent d'un service unique. Ils sont en mesure d'agir sur l'ensemble des comptes utilisateurs et de faire respecter les conditions générales d'utilisation de notre site **MiagerOfi**.

RESPONSIVE DESIGN

Dans l'optique de fournir une interface claire et intuitive sur l'**ensemble des supports** (PC, Tablettes et smartphones), nous avons choisi d'utiliser un framework front-end nommé **Foundation**. Cet outil permet de créer de manière simple des applications web multi-support en fournissant des composants pré-configurés. Ces composants permettent:

- D'améliorer la mise en page avec des **outils GRID**
- De mettre en place une **Navigation** claire
- De créer et configurer des éléments de type **Button** facilement
- D'intégrer facilement les **Plugins** fournis par le framework



Notre application profite donc d'une interface graphique **claire et conviviale**, et ce, quelque soit le support. Cette vision multi-support s'intègre parfaitement à la notion de programmation distribuée apportée avec les EJB.

Côté serveur, MiagerOfi a la possibilité de s'exécuter sur différents serveurs. Côté client, MiageRofi peut s'exécuter et parfaitement fonctionner sur des clients hétérogènes.

INTERFACES

Voici ci-dessus quelques exemples d'interface réalisés avec le framework front-end **Foundation**.

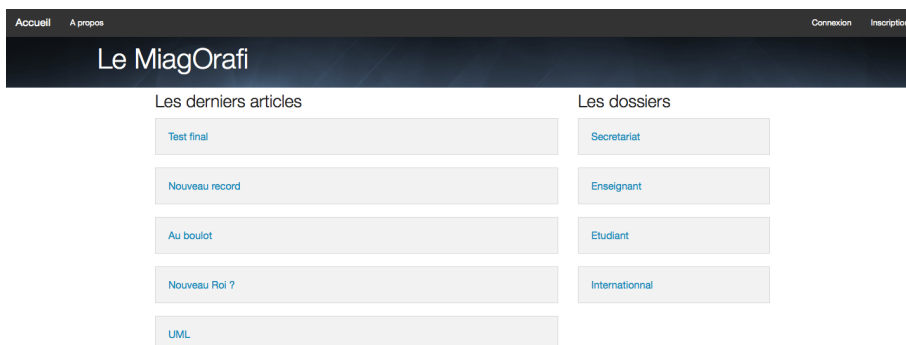


fig1. Accueil du site - Utilisateur non identifié

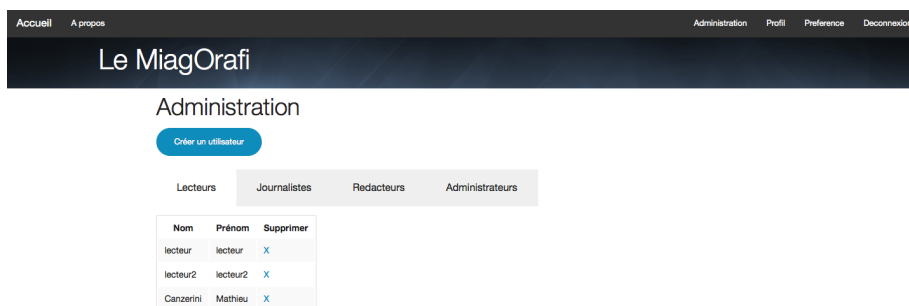


fig2. Interface d'administration - Administrateur

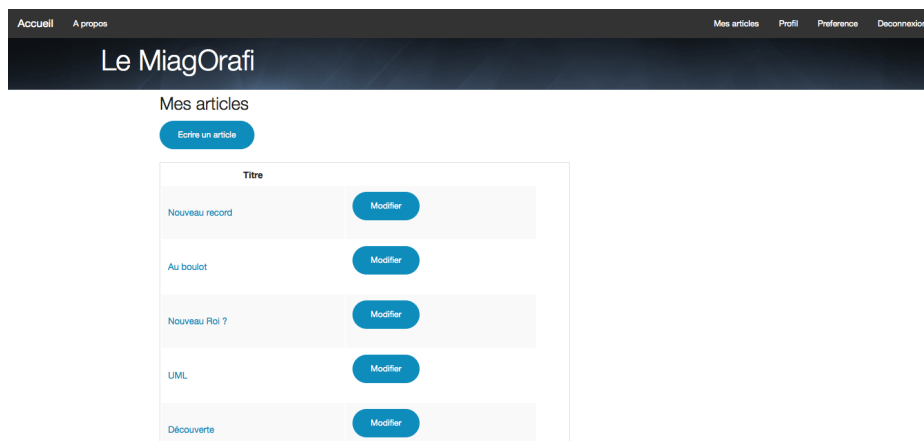


fig3. Liste des articles rédigés - Journaliste

TESTS UNITAIRES

Dans l'optique de fournir une application de qualité, nous avons décidé de réaliser des **tests unitaires** en nous appuyant sur **jUnit4**. Ces tests unitaires sont réalisés sur les fonctionnalités inhérentes à:

- La création d'objets
- La suppression d'objet
- La mise à jour des attributs d'un objet
- La lecture des informations des objets

Ces tests permettent de garantir le bon fonctionnement des EJB et le bon fonctionnement des objets métier de notre application.

DIFFICULTES RENCONTREES

Au cours de ce projet, nous avons rencontré des difficultés, principalement dû à une **méconnaissance du serveur Glassfish 3**. Nous avons mis du temps à le **configurer** afin que celui-ci communique avec la base. De ce fait, la mise en place de la persistance des données en base a été ralentie. Le serveur s'est révélé être assez capricieux et nous avons adapté en conséquence.

En cours de développement, nous avons découvert que notre projet EJB ne pouvait pas fonctionner sans certaines bibliothèques liées à Glassfish. Nous avons mis du temps à résoudre ce problème.

D'une manière générale, une fois passé les difficultés liées à la récupération des objets distants via l'API **JNDI**, nous n'avons pas rencontré d'autres problèmes particuliers.

CONCLUSION

Le développement de cette application web nous aura permis de découvrir de manière plus approfondie la notion de **programmation distribuée en JAVA**. Nous avons manipulé des **EJB** et avons ainsi **appris à les utiliser** et à nous familiariser avec.

Bien que nous ayons déjà abordé la notion de persistance avec Hibernate, nous ne l'avons encore jamais fait avec JPA. Ce projet nous aura permis de voir cette deuxième façon de faire persister les données en base.

D'une manière globale, nous avons découvert au travers d'un projet potentiellement concret comment réaliser une application web pouvant supporter une montée en charge importante. Cette façon de concevoir nous sera utile pour la suite de notre vie professionnelle.