

Architecture :	Une architecture en couche en utilisant l'environnement Java EE avec une séparation entre les différentes couches (Couche persistance, Couche DAO, Couche métier, Couche Sécurité, Couche Présentation)
Hibernate :	Comme un Framework ORM permettant l'abstraction de l'accès à la base de données (avec ces méthodes prédéfinis), et la création d'une couche d'abstraction entre le projet JAVA et la base de données relationnelle (à l'aide des classes POJO).
JSF 2.x (Mojarra Implementation):	Comme un Framework MVC à base des composants qui assure l'implémentation de la couche présentation (à l'aide de ces composants) et le passage des données saisies par le client à la couche métier (à l'aide des Managed Bean). Q : Pourquoi JSF 2.x ? R : <ul style="list-style-type: none"> - View Scope ^[1] - Flash Scope ^[2] - La facilité d'intégrer AJAX par rapport au JSF 1.x.
Pretty faces :	Comme un Simple Framework d'URL Rewriting. Exemple : http://localhost:8080/nomProjet/page.xhtml devient : http://localhost:8080/nomProjet/page/hello
Primefaces :	Comme une librairie des composants graphiques qui enrichit le Framework JSF Q : Pourquoi Primefaces ? R : <ul style="list-style-type: none"> - Facile à utiliser. - Une panoplie riche de composants (+113 composants graphiques). - Une documentation détaillée.
Apache DBCP :	Comme « Database connection pooling service », qui améliore les performances de l'exécution des commandes sur une base de données
Spring ORM :	Comme un Framework qui facilite la création du « SessionFactory » du Hibernate sous forme d'une Bean, au lieu d'utiliser l'ancienne méthode du « HibernateUtil.java »
Spring Security :	Comme un Framework de sécurité pour gérer l'authentification et l'autorisation aux ressources de l'application.
Spring Context:	Comme un conteneur « IOC » et un outil du « Dependency Injection » pour éliminer le couplage fort entre les couches.
OmniFaces 2.0	Comme un « utility library » qui enrichit le framework JSF