

Exercice 01 : Correction

Correction de l'exercice 01

Exercice mettant en pratique la réalisation d'une architecture J2EE.

1. <u>LES COMPOSANTS DE NOTRE SI</u>	<u>2</u>
1.1. LA COUCHE METIER	2
1.2. LA COUCHE DE PERSISTANCE	3
1.3. LA COUCHE DE PRESENTATION	3
2. <u>L'ARCHITECTURE DE NOTRE SI</u>	<u>4</u>
3. <u>LES DIAGRAMMES DE SEQUENCE</u>	<u>5</u>

1. Les composants de notre SI

1.1. La couche métier

L'architecture est basée sur 3 composants principaux :

- **Panier**
- **GestionComptesClient**
- **GestionStockLivres**
- **JMS Provider**

Panier :

Ce composant est un EJB Session Stateful qui gère le contenu du panier d'articles de l'utilisateur. Il mémorise les articles commandés (inutile de le persister en BD, le panier disparaît lors d'une déconnexion involontaire).

Lors de la validation du panier, il demande au **GestionComptesClient** de traiter la commande.

Son interface :

- void addArticle(Article a) : ajouter un article dans le panier
- void annulerPanier() : annule le contenu du panier
- RetourPanier validerPanier : valide le panier qui demande au GestionComptesClient de valider la commande. Si un des livres n'est pas réservable, RetourLivre contient l'impossibilité de réservation et les livres plus libres.

GestionComptesClient :

Ce composant est un EJB Session Stateless qui gère le compte du client : son profil et ses commandes. Il ne mémorise aucune information car il mémorise en base de données directement lors de la création

Son interface :

- void creerProfil(Profil p) : création du profil de l'utilisateur
- void modifierProfil(Profil p) : modification du profil de l'utilisateur
- Profil getProfil(String user) : consultation du profil de l'utilisateur
- Retourcommande validerCommande(String user,int panier) : prend en entrée le contenu du panier et demande à GestionStockLivres (reserveLivres) de réserver les livres. RetourCommande contient l'impossibilité de réservation et les livres plus libres
- Commandes[] getCommandes(String user) : retourne les commandes de l'utilisateur
- void vendreLivreCommandé(int idCommande,int idLivre) : la Caisse utilise cette méthode pour vendre un des livres commandés. Il demande à GestionStockLivres (vendreLivre) de marquer que le livre a été vendu.

GestionStockLivres :

Ce composant est un EJB Singleton. Il est donc utilisé, indirectement, par tous les utilisateurs.

On a fait le choix du singleton, essentiellement, pour synchroniser les méthodes de mise à jour des livres afin que l'on ne puisse pas réserver un livre qui aurait été, entre temps, réservé par un autre utilisateur ou supprimé par l'administrateur.

Son interface :

- synchronized RetourDispo reserverLivres(Livre[] livres) : vérifie que les livres en entrée sont toujours disponibles et en fait la réservation. Si un des livres n'est pas disponible alors pas de réservation et retourne les livres indisponibles sinon retourne ok.
- synchronized void ajouterLivre(Livre l) : ajoute un livre dans le stock
- synchronized void supprimerLivre(int idLivre) : supprime un livre dans le stock
- synchronized void vendreLivre(int idLivre) : cette méthode est appelé par la Caisse et consiste à marquer le livre comme étant vendu
- Livre[] getLivres(Critere c) : retourne des livres en fonction du critère de filtre ou recherche.

JMS Provider :

Etant donné que les IHM Admin et Vendeur peuvent avoir à l'écran des livres en cours de consultation, on met en place un JMS Provider pour notifier ces deux IHMs de toutes les modifications sur les livres réalisées par le gestionnaire de stock.

Pour cela on crée deux MessageDriven, Admin et Vendeur, qui notifient respectivement toutes les IHM Admin et Vendeur.

Le modèle de MOM ici utilisé se justifie car le nombre de poste d'admin et de vendeur n'est pas toujours le même.

Ces IHM s'abonnent au JMS Provider lors de leurs activations.

1.2. La couche de persistance

La couche de persistance est définie par les Entity suivantes :

- Profil : les profils des utilisateurs
- Commande : les commandes des utilisateurs
- Livre : les livres du stock

1.3. La couche de présentation

Les Servlets :

Les actions de l'utilisateur faites dans le navigateur sont réalisées par des Servlets.

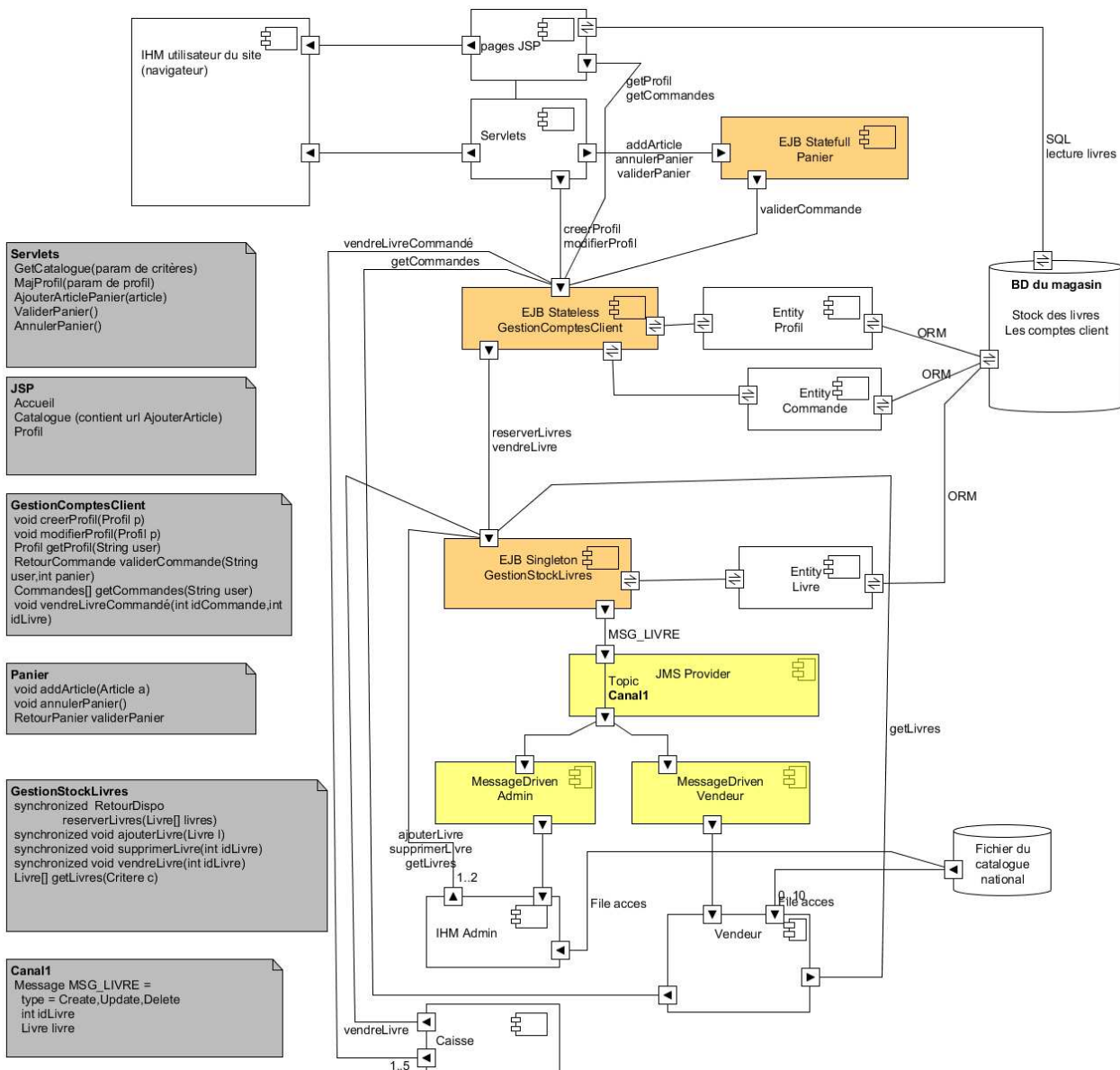
On a les servlets suivantes :

- GetCatalogue(param de critères) : détermine les livres du "catalogue" (les livres de la BD) en fonction de critères de filtre et de recherche. Il fait des requêtes SQL directement dans la BD (select). Il construit la page Catalogue.jsp qu'il retourne.
- MajProfil(param de profil) : réalise l'enregistrement en BD du profil de l'utilisateur. Il utilise l'EJB GestionComptesClient pour enregistrer le profil dans la BD. Il retourne la page Accueil.jsp.
- AjouterArticlePanier(article) : ajoute un livre dans le panier. Il utilise l'EJB Panier. Cette servlet est appelé en Ajax (sans rechargement de la page).
- ValiderPanier() : demande à l'EJB GestionComptesClient de valider la commande des livres. Il retourne la page Accueil.jsp
- AnnulerPanier() : efface le panier. Il retourne la page Accueil.jsp

Les pages JSP :

- Accueil.jsp : cette page affiche la page d'accueil du site qui permet d'accéder au catalogue pour commander les livres.
- Catalogue.jsp : cette page permet à l'utilisateur d'afficher les livres du catalogue en fonction de ses critères de filtre et de recherche et le contenu de son panier. Il peut sélectionner les livres qu'il veut commander. Cette sélection exécute une requête Ajax sur le servlet AjouterArticlePanier. Une fois sa commande terminer un bouton permet de valider sa commande qui appelle le servlet ValiderPanier
- Profil.jsp : cette page affiche un formulaire de création ou modification du profil de l'utilisateur. Un bouton permet de valider le formulaire qui appelle le servlet MajProfil.

2. L'architecture de notre SI

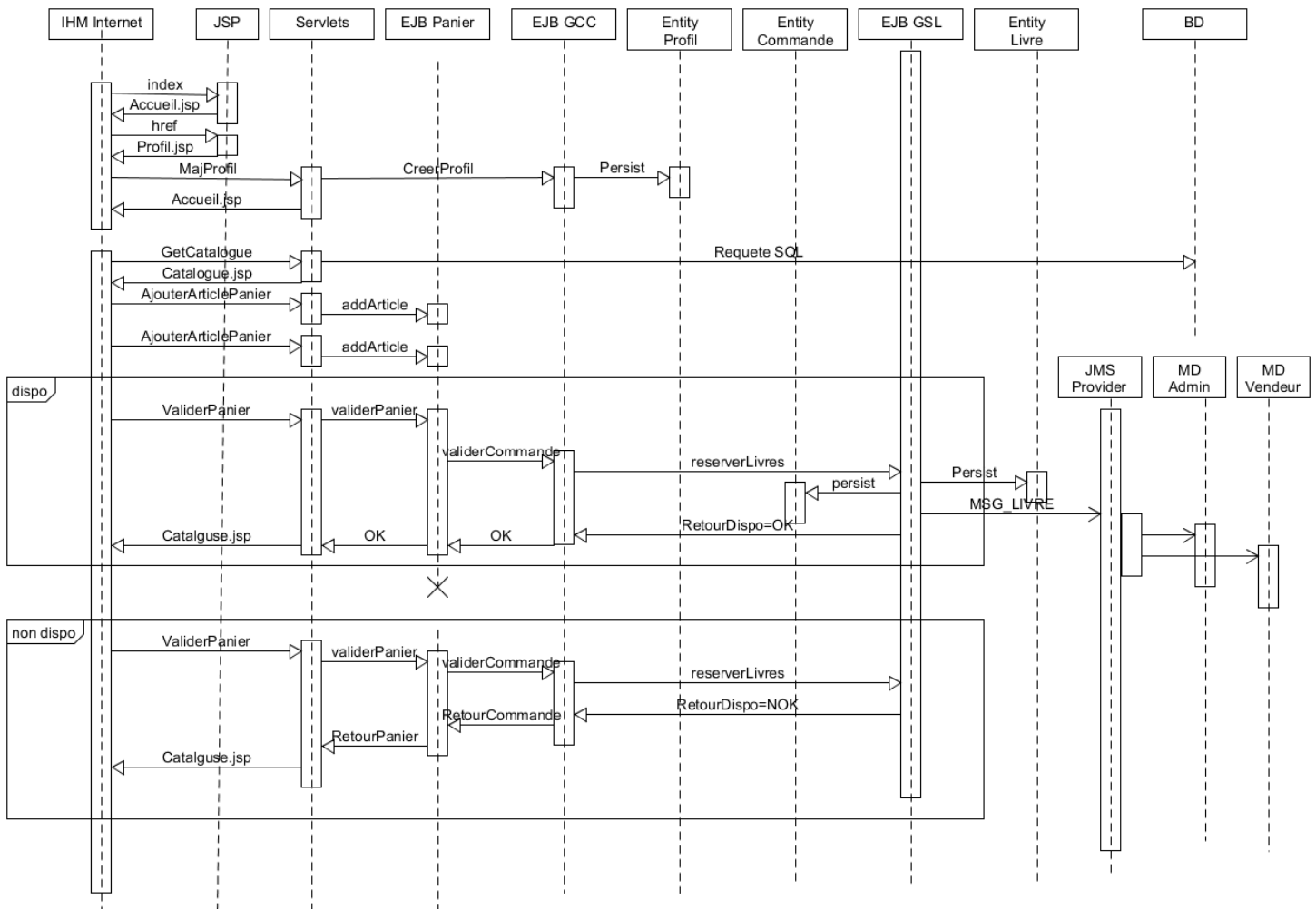


3. Les diagrammes de séquence

Les diagrammes de séquence permettent de vérifier que les **use-cases** (fonctionnalités demandées) se projettent correctement dans l'architecture.

Le diagramme suivant permet de vérifier comment sont réalisées les use-case suivantes :

- La connexion au site
- La création du profil
- La consultation du catalogue
- La commande de livres dans le cas de livres disponibles
- La commande de livres dans le cas de livres indisponibles



Le diagramme suivant permet de vérifier comment sont réalisées les use-case suivantes :

- La consultation des livres par le vendeur
- La consultation des commandes d'un utilisateur du site par le vendeur
- La vente des livres d'une commande par la Caisse
- La vente de livre par la Caisse
- L'ajout d'un nouveau livre par l'administrateur

