

TP Développement Web – Java Enterprise Edition

Adeel Ahmad

I - L'application WebCatalog.com

L'application consiste à développer un site Internet d'e-commerce, dans lequel un internaute pourra visualiser les produits d'une boutique en ligne. La boutique comprend des catégories qui permettent d'organiser les produits. Les produits que nous souhaitons vendre sont des livres, des DVDs, des CDs de musique.

Les clients visitent la page d'accueil et choisissent une catégorie de produits en cliquant sur l'image correspondante. Une fois la catégorie choisie, les clients accèdent à la liste de ses produits. Ils peuvent choisir des produits et les ajouter à leur panier en cliquant sur le bouton prévu à cet effet. Ils peuvent naviguer dans les catégories grâce au menu de gauche. Dès que le panier n'est plus vide, des liens « Panier » et « Terminer les achats » apparaissent en haut à côté de l'icône du panier.

Lors de la visualisation du panier, les clients peuvent changer les quantités grâce au bouton « Mettre à jour ». Via des liens, ils peuvent aussi vider le panier, revenir à la saisie des produits (« Continuer les achats ») ou terminer les achats.

La finalisation de la commande commence par l'enregistrement des informations personnelles des clients dans le formulaire prévu à cet effet. Le montant total de la commande et des frais (forfaitaires) est rappelé. A l'appui du bouton « Enregistrer », des contrôles de validité sont effectués. Quand les données sont correctement saisies la commande est traitée par le site, qui affiche une page de confirmation.

La page de confirmation donne le numéro de confirmation affecté à la commande, et rappelle son contenu.

II. Conception de la base de données

Une catégorie comporte plusieurs produits et un produit est associé à une et une seule catégorie. D'où la clé étrangère `categorie_id` dans la table `produit`.

Les clients passent plusieurs commandes, et une commande est associée à un et un seul client. D'où la clé étrangère `client_id` dans la table `commande_client`.

Une commande contient plusieurs produits, et un produit peut appartenir, à plusieurs commandes. Cette association n vers n est implantée par la table `produit_commande` dont la clé est la composition des clés étrangères `produit_id` et `commande_client_id`.

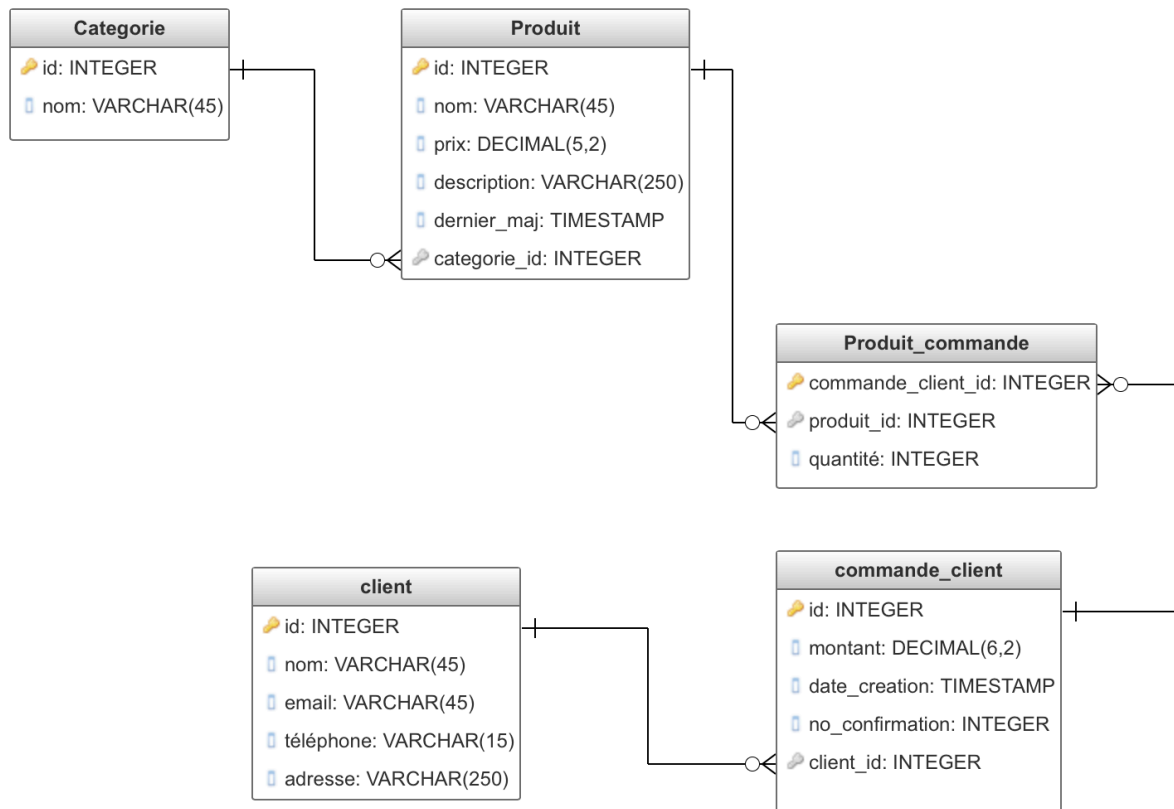


Figure 1. Le schéma de la base de données

III. Conception

Cette application JSP/EJB/JPA peut être organisée très classiquement selon une architecture n-tiers avec le patron MVC2. Le tiers client correspond au navigateur Web. Le tiers Web contient un servlet (ContrôleurServlet) qui joue le rôle de contrôleur unique et cinq pages JSP pour les vues (index.jsp, categories.jsp, panier.jsp, enregistrement.jsp et confirmation.jsp). La partie commune à toutes les pages (tête et pied) est dans Template.jsp. Les cinq pages JSP appellent ce template en lui passant le titre de la page et le fichier de contenu spécifique. Ces contenus sont des pages JSP (contenuIndex.jsp, contenuCategories.jsp, contenuPanier.jsp, contenuEnregistrement.jsp, contenuConfirmation.jsp). Le tiers applicatif (la partie modèle du MVC) comporte des EJB session (stateless), des entités JPA et des classes Java (POJO).

L'EJB GestionnaireCommandes gère l'enregistrement des commandes en s'aidant d'une couche d'EJB « façade ». Ces EJB héritent de FacadeAbstraite et font le lien au niveau applicatif avec les Entités JPA (FacadeCategorie, FacadeClient, FacadeProduit, FacadeCommandeClient, FacadeProduitCommande). Cinq entités JPA font le lien avec les tables de la base de données (Categorie, Client, Produit, CommandeClient, ProduitCommande). Trois POJO gèrent les vérifications des champs saisis (Valideur), le panier (Panier) et les items contenus dans le panier (ItemPanier). Enfin, le tiers des données contient une vase de données (MySQL) conforme au schéma présenté précédemment.

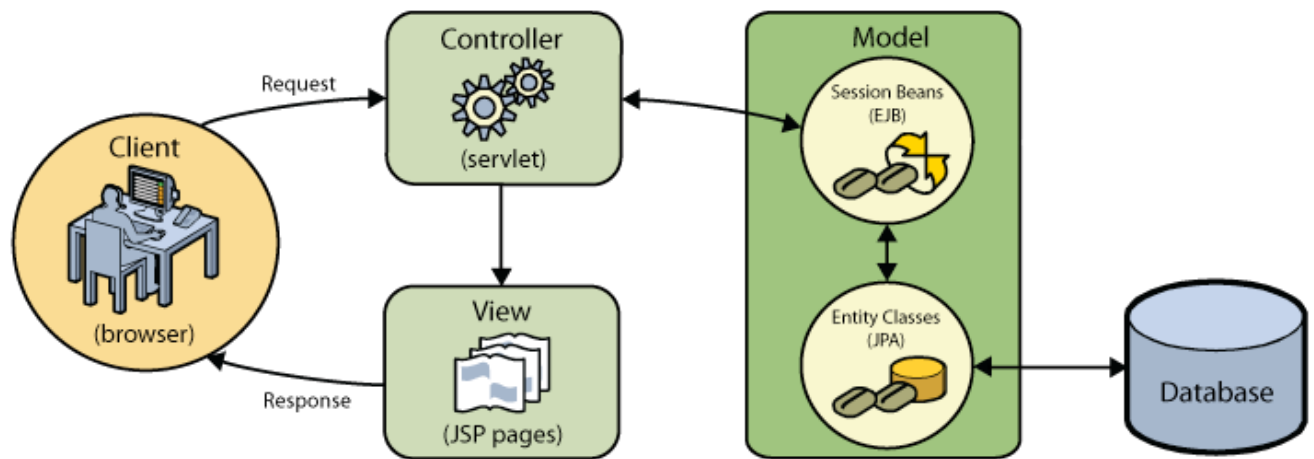


Figure 2. Architecture n-tiers et MVC

IV - Processus métier de vente en ligne

Le schéma ci-dessous montre la cinématique du front office à mettre en place.

Le processus métier est décomposé en 3 parties :

- La navigation dans les produits du site
- L'ajout au panier et la création de la commande
- La gestion du compte client.

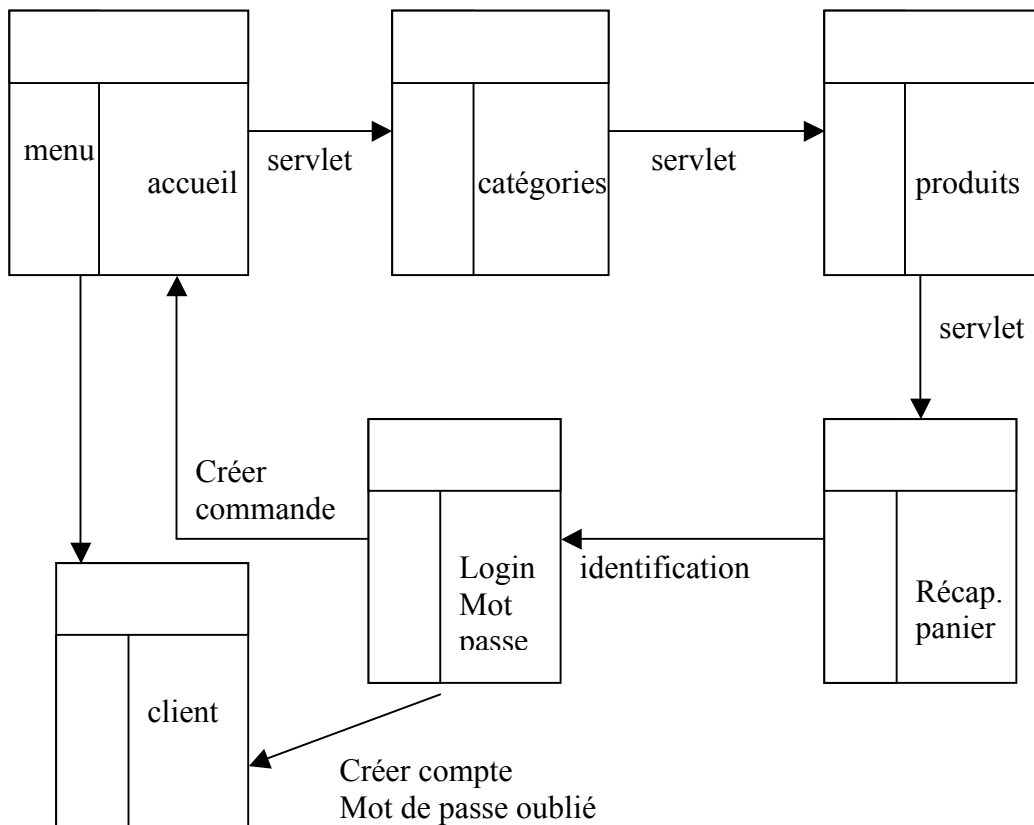
V - Navigation dans la boutique

La navigation dans la boutique de vente en ligne démarre en page d'accueil du site. Le page d'accueil offre un bandeau (bannière) dans la partie haute de chaque page web. Un menu se situe dans la partie droite de la page et propose 3 accès rapides :

1. Voir les produits
2. Créer mon compte
3. Mon panier

La partie centrale est une zone de contenu permettant d'afficher :

- différents types de messages commerciaux en home page
- les catégories et les produits du site lorsque l'internaute clique sur le lien Voir les produits
- un formulaire de création de compte lorsque l'internaute clique sur Créer mon compte. Le compte client sera une étape essentielle pour la création de la commande et la validation de celle-ci à l'aide d'un moyen de paiement.
- Un tableau récapitulatif des articles sélectionnés par le client et ajoutés à son panier. Le panier doit permettre de connaître le montant total de la commande en cours, de modifier les quantités, de supprimer un produit du panier.



Depuis la page d'accueil et en sélectionnant "Voir les produits"

L'internaute pourra sélectionner un produit, visualiser son descriptif et ajouter ce produit à son panier. L'internaute aura la possibilité de visualiser son panier à tout moment, puis de valider celui-ci afin de créer une commande. Lorsque la commande est créée puis validée, l'internaute est redirigé vers la page d'accueil du site.

Pour valider la commande, l'internaute devra s'identifier en fournissant son login et son mot de passe. Dans le cas où, l'internaute ne dispose pas de compte, il pourra le créer lors de la validation de commande ou directement à partir du menu.

L'internaute aura la possibilité de demander son mot de passe par email pour effectuer son identification.

L'accès au panier doit être disponible depuis toutes les pages du site web.

Développement de la Webapp de l'application

Page d'accueil de la e-boutique



L'écran ci-dessus permet d'accueillir le client sur la e-boutique. Il est constitué d'un bandeau sur la partie haute de l'écran. La partie centrale est simplement d'un message d'accueil. La partie droite est un menu qui offre 3 fonctionnalités principales :

- Accueil : Retour à la page d'accueil
- Categories : montre la liste des catégories disponibles dans la e-boutique
- Votre panier : montre le contenu du panier de l'utilisateur internet.

Afin d'offrir de bonnes performances ainsi qu'une navigation plus pratique sur le site Internet, la sélection de liens pour s'effectuer sous la forme de requête AJAX.

Page Catégories de la e-boutique



La page catégories montre la liste des catégories stockées en base de données. Chaque catégorie est sélectionnable et permet d'afficher une liste de produits. Le menu reste accessible aux utilisateurs. La partie centrale du site Internet est rechargé pour montrer la liste des catégories :

- 1) les libellés des catégories.
- 2) Un message d'accueil dans la zone centrale pour indiquer à l'utilisateur de sélectionner une catégorie.

Page Produits de la e-boutique



La sélection d'une catégorie permet d'afficher dans la zone centrale la liste des produits qui sont à disposition dans notre e-boutique. Le menu ainsi que la liste des catégories restent accessibles. La partie centrale affiche l'intitulé de la catégorie sélectionnée puis en dessous la liste des produits liés à cette catégorie. Pour chaque produit, un lien « Ajouter au panier » permet de sélectionner le produit afin de le stocker dans le panier, puis éventuellement par la suite de valider le contenu du panier afin de passer une commande sur le site e-boutique. Le panier doit être géré en session dans la partie web de l'application.

Le lien ajouter au panier conduit directement au récapitulatif du panier.

Page Récapitulatif Panier



La page récapitulatif panier monte l'ensemble des articles ainsi que leur prix. L'utilisateur a la possibilité de valider son panier et de créer une commande.

Page Création de commande

Suite à la validation du panier, l'utilisateur non loggé arrive sur une page lui indiquant de se connecter ou de créer un compte. Dans le cas où l'utilisateur est loggé, un formulaire lui indique de saisir son adresse de livraison, puis un bouton payer permet de créer la commande avec son adresse de livraison et de la sauvegarder en base de données. Une page de confirmation et de remerciement est affichée suite à la création de et à l'enregistrement de la

commande. Les traitements de connexion, de création de compte, de « paiement » sont des actions Struts.

Page Création du client

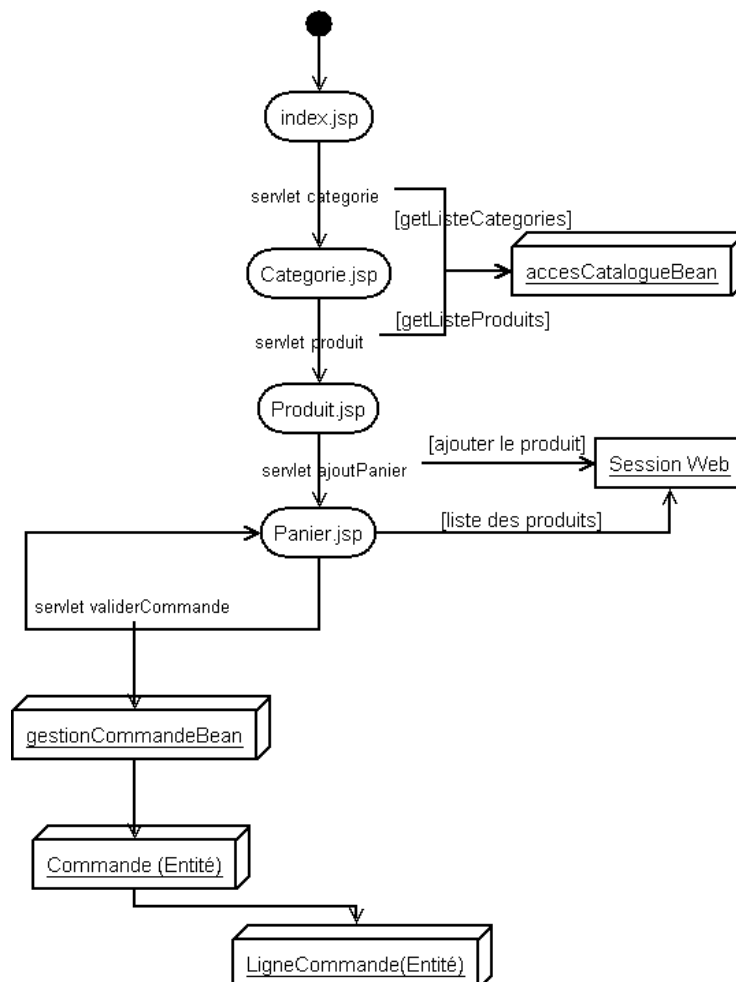
Cette page consiste en un formulaire dans lequel le client saisit son nom, son prénom, son adresse email, son adresse (rue, code postal, ville, pays), son numéro de téléphone ainsi que son mot de passe. Ces champs sont tous obligatoires pour valider la création de compte. Le login du client correspond à l'adresse email saisie. Il ne doit exister qu'un seul compte par adresse email. La validation du formulaire de création de compte déclenche une action Struts pour créer le client et le sauvegarder en base de données. Le client créé est automatiquement loggé sur le site. Il peut alors valider sa commande (retour au récapitulatif panier) ou continuer de naviguer sur le site (retour à la page d'accueil).

Architecture de l'application

L'application sera découpée selon une architecture 3-tiers :

- 1) le tiers web hébergera la partie web de l'application, à savoir les pages JSP, les pages HTML, les images et animations, les codes Javascripts ainsi que les servlets.
- 2) Le tiers métier comportera l'ensemble des composants métiers de l'application J2EE sous la forme d'Enterprise JavaBeans. Les composants EJB développés dans le cadre de notre e-boutique seront :
 1. accesCatalogueBean : contient les services d'accès au catalogue de la e-boutique, en particulier pour l'accès aux catégories et aux produits
 2. gestionCommandeBean : contient les services de gestion des commandes
 3. les EJB entités Commande et LigneCommande permettent de gérer l'accès aux données des commandes de la e-boutique.
- 3) les accès à la bases de données se feront au travers d'une source de données (DataSource) entièrement gérée par le serveur d'applications JBoss

Le schéma ci-dessous montre la structure de l'application J2EE e-boutique :



Paramétrage des EJBs entités Commande

Les EJBs entités Commande et LigneCommande permettent d'automatiser les tâches de création, de recherche et de mise à jour des commandes en base de données.

Les commandes comportent un id, un numéro, une liste de lignes commandes (relation oneToMany).

Les lignes commandes comportent un id, un code, une désignation, une quantité, un prix unitaire et sont associées à une commande (relation ManyToOne)

Le mapping des EJBs entités commandes et lignes commandes est réalisé automatiquement à l'aide du système d'annotations et du fichier persistence.xml (à placer dans le répertoire META-INF de votre module EJB).

Le fichier persistence.xml peut être défini comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>

<persistence>
  <persistence-unit name="managerBoutique">
    <jta-data-source>java:/MySqlDS</jta-data-source>
    <class>fr.ulco.ejb.LigneCommande</class>
    <class>fr.ulco.ejb.Commande</class>
    <properties>
      <property value="org.hibernate.dialect.MySQLDialect" name="hibernate.dialect"/>
      <property value="update" name="hibernate.hbm2ddl.auto"/>
      <property value="true" name="hibernate.show_sql"/>
    </properties>
  </persistence-unit>
</persistence>
```

Paramétrage des EJBs Session accesCatalogueBean et gestionCommandeBean

L'Enterprise Java Bean Session accessCatalogueBean offre des méthodes pour l'accès aux catégories et aux produits. Pour des raisons de performance, cet EJB sera de type Session Stateless. Les services proposés par cet EJB sont indépendants des clients.

L'interface de ce composant est la suivante :

```
@Remote
public interface accesCatalogueBeanRemote {
    public ArrayList<Categorie> getListCategories();
    public ArrayList<Produit> getListProduits(int id);
}
```

L'Enterprise Java Bean Session gestionCommandeBean offre des méthodes pour l'accès aux commandes passées sur le site e-boutique.

L'interface de ce composant est la suivante :

```
@Remote
public interface GestionCommandeBeanRemote {
    public Commande createCommande();
    public List<Categorie> gererCommande();
    public void validerCommande(Commande c);
}
```

Pour manipuler les EJBs entités Commande et LigneCommande, le composant entityManager devra être défini au niveau de la classe de l'EJB Session :

```
public class GestionCommandeBean implements GestionCommandeBeanRemote,
GestionCommandeBeanLocal {
    @PersistenceContext(unitName="managerBoutique")
    EntityManager em;

}
```

Le servlet appelle l'EJB à l'aide d'interface remote par le service JNDI et accède la méthode validerCommande(Commande c)

```
// On enregistre les lignes dans la commande
c.setLignesCommandes(liste);

// On fait appel coté métier à la validation de la commande
remote.validerCommande(c);
```

Pour l'insertion d'une entité « Commande » dans la base de données, l'EJB compose la méthode validerCommande(Commande c)

```
public void validerCommande(Commande c){
    c.setNumero("69");
    em.persist(c);
    System.out.println("commande enregistré : "+c);
}
```

Bean Commande :

```
@Entity
@Table(schema="boutique", name="commandes")
public class Commande implements Serializable{
    private int id;
    private String numero;
    private Collection<LigneCommande> lignesCommandes;

    @Id
    @Column(name="pk_commande")
    @GeneratedValue
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }

    @Column(name="numero")
    public String getNumero() {
        return numero;
    }
    public void setNumero(String numero) {
        this.numero = numero;
    }

    @OneToMany(targetEntity=LigneCommande.class, mappedBy="commande",
cascade=CascadeType.ALL)
    public Collection<LigneCommande> getLignesCommandes() {
        return lignesCommandes;
    }
    public void setLignesCommandes(Collection<LigneCommande> lignesCommandes) {
        this.lignesCommandes = lignesCommandes;
    }

    public String toString(){
        return "id "+ id + " numero " + numero;
    }
}
```