

# Enoncé du travail pour l'examen de Java Web

## (Informatique de gestion – bloc 3)

La cotation attribuée pour le cours de Programmation Web orientée objet (théorie et labo) est basée d'une part sur un travail de groupe et d'autre part sur un examen oral.

Le travail est une application Web à écrire en Java en utilisant les technologies **Spring et Hibernate**. Ce travail doit être réalisé par groupe de deux étudiants.

*L'évaluation du travail de groupe constituera une cote de base pour l'examen oral ; cette cote pourra être sensiblement augmentée ou diminuée en fonction des réponses données par l'étudiant aux questions posées lors de l'oral.*

Lors de l'examen oral, l'étudiant devra être capable d'expliquer n'importe quelle ligne de code du programme Java réalisé en groupe. L'étudiant devra être capable d'expliquer les **Design Patterns** vu lors de son cursus.

## 1 Directives générales

- 1) Le programme consiste en une application Web à développer dans l'environnement **Eclipse** en utilisant **Spring, Maven et Hibernate**.
- 2) L'application doit être déployée sur un serveur **Tomcat**.
- 3) L'application doit accéder à une base de données **MySQL**. Les contraintes d'intégrité référentielle doivent être vérifiées (présence de **clés étrangères**). Prévoyez des contraintes assurant l'intégrité de la base de données (**check, ...**).
- 4) Vous devez rendre le dossier contenant le résultat de votre travail au plus tard le deuxième jour du blocus de janvier. Ce dossier doit être remis sous forme électronique (CD). Le jour de l'examen oral, vous devrez exécuter votre application Web sur la machine de votre choix (PC personnel ou PC de l'école) et le professeur testera devant vous les différentes fonctionnalités. *Un dossier communiqué par mail ne sera pas accepté ; une copie sur CD est nécessaire pour l'archivage des examens.* Le dossier doit comprendre :
  - une présentation du domaine d'application ;
  - le schéma conceptuel de la base de données, le schéma logique des tables, ainsi que la documentation de ces tables (définition de chaque table et chaque colonne, en précisant les identifiants et clés étrangères) ;
  - **le script SQL de la création de la base de données ;**
  - le code de l'application Web.

Le dossier doit être complet et impérativement comprendre ces 4 parties.

- 5) Votre travail sera examiné avant l'examen. C'est donc le travail que vous remettrez alors qui sera évalué et coté, et en aucun cas, une version ultérieure du programme (que vous apporteriez éventuellement le jour de l'examen).

## 2 Fonctionnalités de l'application

L'application consiste en un site de vente en ligne.

Le domaine d'application est laissé à votre appréciation. Le domaine d'application choisi doit être proposé au professeur de laboratoire pour validation. Les domaines d'application doivent être différents d'un groupe à l'autre.

### Types d'utilisateur

Il y a deux façons d'utiliser le site :

- le simple internaute non identifié (1) ; il visite le site et consulte le catalogue des produits vendus ;
- le client qui s'est identifié via login et mot de passe (2) ; il peut en outre passer commande.

Tout visiteur du site qui le désire doit donc pouvoir s'inscrire en choisissant un **login** et un **mot de passe**.

Lors d'une inscription d'un client, les informations à demander sont au minimum :

- nom et prénom
- adresse de livraison
- adresse email
- téléphone

Prévoir impérativement dans ce formulaire certains champs **facultatifs**.

Toutes les données relatives aux clients doivent être mémorisées dans la base de données.

Dans tous les cas, qu'il se soit identifié (cfr (2)) ou non (cfr (1)), le visiteur du site doit pouvoir :

- consulter le **catalogue** des catégories de produits vendus;
- lister les **produits** d'une catégorie choisie ;
- visualiser les **détails** d'un produit choisi ;
- placer un produit dans son **caddie (ou panier)** en **précisant le nombre d'exemplaires souhaités** ;
- **visualiser le contenu du caddie** à n'importe quel moment (liste des articles avec libellé, prix unitaire et quantité souhaitée, ainsi que le montant total du caddie) ;
- **modifier la quantité** d'un article choisi dans le caddie ;
- **supprimer un article** du caddie.

Les utilisateurs qui sont identifiés peuvent en outre passer commande du contenu du caddie.

## Structure du site

Chaque page doit offrir (au minimum) :

- la possibilité de s'identifier (via login et mot de passe) ;
- un message personnalisé si l'utilisateur est logé ;
- un lien vers le formulaire d'inscription pour les nouveaux clients qui désirent s'inscrire ;
- un lien vers la page accueil ;
- un lien vers la description de la société (pages statiques) ;
- un lien vers le catalogue des produits proposés à la vente (pages dynamiques) ;
- la possibilité de visualiser le contenu du caddie ;
- le choix entre au moins deux langues.

Le catalogue des produits disponibles à la vente doit être proposé d'abord **par catégories de produits**. Les catégories sont dynamiques, c'est-à-dire qu'elles doivent être stockées dans la base de données. *Toute modification apportée aux catégories dans la base de données doit donc apparaître sur le site de vente en ligne.*

L'utilisateur doit pouvoir choisir une catégorie de produits et demander l'affichage des **produits** de la catégorie choisie. Comme pour les catégories, les produits sont stockés dans la base de données. *Toute modification apportée aux produits dans la base de données doit donc apparaître sur le site de vente en ligne.* La liste des produits correspondant à la catégorie choisie doit proposer pour chaque produit, au minimum le libellé du produit, le prix unitaire ainsi qu'un lien vers le **détail du produit**.

Un clic sur le lien **détail du produit** présente le produit et toutes ses caractéristiques. Cette page doit contenir un bouton permettant **d'ajouter** le produit au **caddie** (ou panier) du client.

Un même produit doit pouvoir être placé dans le caddie en **plusieurs exemplaires**. Le client doit donc pouvoir choisir le nombre d'exemplaires.

La notion de caddie permet au client de simuler l'achat de plusieurs produits avant de valider sa commande. A tout moment (c'est-à-dire quelle que soit la page), l'utilisateur doit pouvoir **visualiser le contenu de son caddie**.

La page décrivant le contenu du caddie doit comprendre un bouton permettant de **passer commande**. Avant d'enregistrer la commande, une confirmation doit être demandée à l'utilisateur.

Les **commandes** confirmées par les clients doivent être **enregistrées dans la base de données**.

Votre site doit proposer des **promotions** qui varient de semaine en semaine. *A vous de faire preuve d'originalité dans la façon de proposer ces promotions !*

L'utilisateur doit pouvoir **choisir la langue** : il doit avoir le choix entre au minimum le français et l'anglais. Les libellés statiques du site doivent être traduits dans la langue choisie par l'utilisateur ainsi qu'au minimum les noms des catégories de produits (provenant de la base de données). Attention également à la traduction des messages d'erreur.

Tant qu'un utilisateur reste **connecté** (après s'être identifié via login et mot de passe corrects), un **message personnalisé** doit être affiché dans la bannière reprenant son login.

### 3 Contraintes

- L'application doit être structurée en **couches**. Appliquez les principes de **l'architecture 3-tiers**. Appliquez également l'architecture **Model – View – Controller**. Evitez à tout prix le couplage entre les couches.
- Les pages de présentation des catégories des produits mis en vente ainsi que les pages listant les produits d'une catégorie doivent être des **pages dynamiques**. Les caractéristiques des produits et des catégories de produits doivent être enregistrées dans la base de données. L'affichage des catégories et des produits nécessitent donc des accès à la base de données.
- Les données entrées par l'utilisateur doivent être vérifiées. **Vérifications** minimales :
  - les champs *obligatoires* doivent être remplis ;
  - les champs de type *numérique* ne contiennent que des chiffres ;
  - saisie de *mot de passe* : prévoir deux champs de saisie de mot de passe ; les deux valeurs introduites doivent être égales ;
  - l'adresse *email* doit avoir un format correct (ex : contenir @ et . ) ;
  - certains champs doivent avoir une *longueur minimale* et *maximale* (téléphone, code postal, ...) ;
  - la quantité commandée doit être *positive* ;
  - le *login* proposé par l'utilisateur ne doit *pas déjà exister* dans la base de données.
- Sécurisez au maximum votre programme. Les données introduites par l'utilisateur doivent être **validées à plusieurs niveaux**, au minimum :
  - Dans le **formulaire** : la validation des données introduites par l'utilisateur via des formulaires doit être gérée autant que possible ;
  - Dans la **base de données** : placer le maximum de contraintes (primary key, unique, foreign key, **check** sur les valeurs permises dans les colonnes, ...)

**Sachez justifier pourquoi** ces vérifications sont nécessaires à tous les niveaux !

- Les **mots de passe** doivent être **cryptés** dans la base de données.
- Couche View : dans les pages Web,
  - gérez la mise en forme des pages via **CSS** ;
  - structurez les pages : utilisez un **template**.
- Vous devez impérativement utiliser une **HashMap** pour stocker les produits commandés par le client. Autrement dit, vous devez stocker le caddie dans une HashMap.
- Toutes les **collections** utilisées doivent être **typées** (HashMap, ArrayList, ...).
- La **base de données** devra être **garnie** avec **suffisamment** de données pour pouvoir tester complètement le programme (plusieurs catégories avec plusieurs produits, clients déjà enregistrés, ...)
- Aucun retard ne sera accepté pour la remise du dossier ; tout **retard** entrainera une **cote nulle**.
- **L'orthographe** interviendra dans la cotation.

## 4 Bonus

- Une fonctionnalité de **recherche** ;
- L'utilisation d'**Ajax** ;
- Gestion des **paiements** sécurisée.

*L'originalité de votre travail sera prise en compte.  
Le design quant à lui n'intervient pas dans la cotation.*

***Bon travail !***

***Françoise Dubisy***