

IG338 - Développement avancé d'application Web

Henallux – Département technique IESN

Évaluation – Janvier 2026

1) Généralités

Le projet consistera en la conception et implémentation d'un site de vente en ligne sécurisé.

Ce projet doit être réalisé en groupe de deux ou trois étudiants.

Afin de constituer votre groupe, envoyez dès que possible un email à Mr. Bodart (bastien.bodart@henallux.be) comprenant vos noms, prénoms et adresses email.

2) Fonctionnalités

Types d'utilisateur

Deux profils d'utilisateur pourront utiliser votre site :

- le simple internaute non authentifié (1) ; il peut visiter le site et consulter le catalogue des produits vendus ;
- le client qui s'est authentifié via login et mot de passe (2) ; il peut, lui, passer une commande.

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

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

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

Attention, il faut prévoir impérativement dans ce formulaire certains champs **facultatifs**.

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

Dans tous les cas, qu'il soit authentifié (2) ou non (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** (un produit doit donc pouvoir être acheté en plusieurs exemplaires) ;
- **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 authentifiés peuvent en outre **passer commande** du contenu du caddie.

Ils doivent aussi pouvoir se déconnecter et reprendre le rôle d'utilisateur non identifié.

Structure du site

Chaque page doit offrir (au minimum) :

- la possibilité de s'identifier (via login et mot de passe) ou de se déconnecter si on est connecté (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 (page statique) ;
- 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 ;

De plus, si l'utilisateur est authentifié:

- un message personnalisé doit être affiché dans la bannière;
- l'utilisateur doit pouvoir se déconnecter ;
- l'utilisateur doit avoir la possibilité de **modifier son compte utilisateur**.

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 **dynamiques** et proviennent de 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 panier/caddie permet au client de regrouper plusieurs produits à commande. À 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.

Ne peuvent commander que les clients authentifiés.

Les commandes, ainsi que les lignes de commandes correspondantes, confirmées par les clients doivent être **enregistrées dans la base de données, et ce, avant même que la commande ne soit payée**.

Une fois la commande confirmée, l'utilisateur doit pouvoir payer en ligne.

Le site de vente en ligne doit disposer d'un mode de paiement. Au moins une technique de **paiement en ligne** doit être proposée à l'utilisateur (PayPal ou autre).

Une fois le paiement validé, la commande est marquée comme payée dans la base de données.

Votre site doit proposer des **promotions**. *À vous de faire preuve d'originalité dans la façon de proposer ces promotions !*

Les promotions doivent être calculées dans la couche business/métier à l'aide de services.

L'utilisateur doit pouvoir **choisir la langue** : il doit avoir le choix entre au minimum le français et l'anglais.

La traduction doit être gérée à 3 niveaux : libellés statiques, libellés dynamiques et messages d'erreur.

Les **libellés statiques** du site doivent être traduits dans la langue choisie par l'utilisateur ainsi que, impérativement, **certains libellés dynamiques**, par exemple les noms des catégories de produits (provenant de la base de données).

Attention également à la traduction des **messages d'erreur**, qui doit être prévue.

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

3) Technologies

Vous devez au minimum utiliser les technologies suivantes :

Langage : Java
Framework : Spring Boot
Build tool : Apache Maven
ORM : Hibernate
Serveur Web : Apache Tomcat
Test frameworks : JUnit, Mockito
Vue : JSP/HTML + CSS
DB : PostgreSQL

4) Contraintes

L'application doit être structurée en **couches**. Appliquez les principes de **l'architecture 3-tiers**.

Appliquez également l'architecture **Model – View – Controller**.

Évitez à tout prix le couplage entre les couches. Il doit être justifié. Pratiquez l'abstraction quand cela est possible.

De même, les différentes techniques de **validation** proposées dans le cours doivent être appliquées.

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.

Plusieurs **tests unitaires** de vos services doivent être présents.

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 d’un nouveau *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 application. Elle doit notamment correctement gérer l’authentification et être protégée contre les injections.

Les **mots de passe** doivent être **hashés** dans la base de données. Utilisez un algorithme récent.

Couche *View* : dans les pages Web,

- gérez la mise en forme des pages via **CSS** ;
- structurez les pages : utilisez un **template**.

La **base de données** devra être **garnie** avec **suffisamment** de données pour pouvoir tester complètement le programme (plusieurs catégories avec chacune plusieurs produits, clients déjà enregistrés...).

5) **Délivrables**

Pour remettre votre projet, un repository sera créé pour votre groupe sur la plateforme GitLab. Sur celui-ci, vous déposerez au minimum le contenu suivant :

- Le code source complet de votre application
- Le script de création (et éventuellement de population) de la base de données
- Le fichier pom.xml nécessaire pour construire l’application avec Maven
- Toute ressource nécessaire au fonctionnement de l’application

La date ultime de remise du projet est le dimanche 4 janvier à 23h59.

Toute modification du repository après cette date ne sera pas prise en compte.

6) Évaluation

Le groupe effectuera une défense orale du projet lors de la session d'examens.

Lors de celle-ci une démonstration de l'application et de ses fonctionnalités aura lieu, suivie d'une revue du code de l'application.

Chaque membre du groupe sera individuellement interrogé sur des parties de ce code, qu'il devra être capable d'expliquer correctement, ainsi que les concepts et technologies sous-jacents.

Une note sur 20 individuelle sera attribuée en fonction des critères suivants :

- Respect des spécifications du cahier des charges
- Compréhension du code et des concepts/technologies
- Gestion de la sécurité de l'application

Bon travail !