

# 1. Mise en place de méthodes pour montrer que l'application est correcte :

## 1.1 Fonctionnalités attendues et préconditions :

- **Fonctionnalités: préconditions** (certaines fonctionnalités n'ont pas de précondition à leur bon fonctionnement)
  - **Inscription** : Pour s'inscrire l'utilisateur doit avoir une email valide et doit remplir toutes les cases obligatoires dans le formulaire.
  - **Connexion** : Pour se connecter l'utilisateur doit avoir déjà créer un compte. Il doit remplir les cases demandées sur le formulaire de connexion.
  - **Ajouter au panier** : L'utilisateur doit-être connecté sur le site.
  - **Consulter Catalogue**
  - **Consulter fiche produit**
  - **Filtrage par catégorie ou sous-catégorie**
  - **Consulter Panier** : L'utilisateur doit-être connecté sur le site.
  - **Consulter historique d'achat** : L'utilisateur doit-être connecté sur le site.
  - **Augmenter quantité d'un produit commandé** : L'utilisateur doit-être connecté et doit avoir un produit dans son panier. La quantité commandée doit-être inférieur au stock du produit.
  - **Baisser quantité d'un produit commandé** : L'utilisateur doit-être connecté et doit avoir un produit dans son panier. La quantité commandée doit-être supérieur à 0.
  - **Supprimer un produit du panier** : L'utilisateur doit-être connecté et avoir un produit dans son panier.
  - **Modifier son profil** : L'utilisateur doit être connecté sur le site.
  - **Payer** : L'Utilisateur doit-être connecté et avoir des produits dans son panier. Il doit rentrer ses informations de paiement et de livraison.
  - **Rentrer un code promo** : Le code promo doit-être valide, l'utilisateur doit-être connecté et avoir des produits dans son panier. Les codes promos ne peuvent pas se cumuler.

## 1.2 Résultats attendus du test:

- **Fonctionnalités : résultat attendus**

- **Inscription** : Les informations de l'utilisateur doivent être insérées dans la base de données. Le mot de passe doit être hâché.
- **Connexion** : L'utilisateur doit avoir accès à son profil et à toutes les fonctionnalités nécessitant une connexion.
- **Ajouter au panier** : L'utilisateur doit pouvoir ajouter un produit à son panier en cliquant sur le bouton prévu à cet effet. Le produit et ses informations doivent-être visible dans le panier. Pour cela il doit être ajouter à la base de données.
- **Consulter catalogue** : Lorsqu'il arrive sur le site l'utilisateur doit-être capable de visualiser la page d'accueil ainsi que de naviguer entre les différents produits du site sans être inscrit/connecté.
- **Consulter fiche produit** : En cliquant sur un produit l'utilisateur doit pouvoir visualiser la fiche produit contenant les informations de ce dernier.
- **Filtrage par catégorie ou sous-catégorie, prix** : L'utilisateur doit pouvoir filtrer les produits selon la catégorie, la sous-catégorie ou le prix.
- **Consulter Panier** : L'utilisateur doit pouvoir visionner son panier en cliquant sur les boutons prévus à cet effet.
- **Consulter historique d'achat** : L'utilisateur doit pouvoir visionner son historique d'achat en cliquant sur les boutons prévus à cet effet.
- **Augmenter quantité d'un produit commandé** :L'utilisateur doit pouvoir augmenter la quantité d'un produit qu'il veut commander. Cela doit modifier la quantité commandée dans la base de données pour cet utilisateur.
- **Baisser quantité d'un produit commandé** : L'utilisateur doit pouvoir baisser la quantité d'un produit qu'il veut commander. Cela doit modifier la quantité commandée dans la base de données pour cet utilisateur.
- **Supprimer un produit du panier** : L'utilisateur doit pouvoir supprimer un produit de son panier. Cela doit le supprimer de la base de donnée pour cet utilisateur.
- **Modifier son profil** : L'utilisateur doit pouvoir modifier les informations de son profil qu'il a rentré en s'inscrivant sur le site. Ces informations doivent se modifier dans la base de données.

- **Payer** : L'utilisateur doit pouvoir payer une fois qu'il a au moins un produit dans son panier. Cela doit modifier la base de données afin de supprimer son panier en plus de lui rajouter son historique d'achat.
- **Rentrer code promo** : L'utilisateur doit pouvoir bénéficier d'une réduction sur le prix d'achat lorsqu'il rentre un code présent dans la base de données. Une fois le code utilisé, la base de données doit-être modifiée afin de réduire de 1 le nombre d'utilisations restantes de ce code.

### 1.3 Jeux de test :

- Fonctionnel:
  - Test Admin: test du blocage de la page admin si l'on est pas connecté et si le role est différent de "[ROLE\_ADMIN]"; test que l'admin peut supprimer un produit; test que l'admin peut supprimer un code promo
  - Test HomePage: test qu'il y à bien 1 bouton "decouvrir" pour chaque catégorie
  - Test Register: test que /register redirige bien lorsque les informations du formulaire sont valides; test que /register ne redirige pas si les informations du formulaire ne sont pas correct
  - Test Login: test que /login fais bien la redirection si les informations du formulaire sont valides; test que /login ne redirige pas si les identifiants sont incorrect
  - Test Commande: test que quand on valide sont panier on est bien rediriger et que le panier est ajouté dans commande
  - Test Panier: test de "ajouter au panier"; test de supprimer un produit du panier; test de augmenter la quantité d'un produit dans le panier; test de baisser la quantité d'un produit dans le panier
- Unitaire: Test des entités CodePromo ; Panier; Produit pour vérifier qu'elle était bien instancier

## 2. Coût en temps et coût en mémoire :

### 2.1 Traitement optimal des données :

- Utilisation des **fonctionnalités de symfony** pour le hâchage des mots de passe
- Utilisation des **design patern** (MVC, DAO...) pour fragmenter et simplifier notre code

- Utilisation de **procédures** afin de ne pas dupliquer du code en plus de grandement faciliter l'accès à la base données en plus de réduire la taille du code.
- Utilisation de **doctrine** pour créer notre base de données de manière optimisée.
- Eviter les rechargement de pages inutiles à l'aide javascript quand c'est possible.
- Stocker le résultat des appels à la base ou des calculs dans des variables afin d'éviter de faire le calcul plusieurs fois.
- Création de fonctions afin d'éviter la duplication de code.

## 2.2 Absorption de la montée en charge par le serveur :

- Optimisation des requêtes SQL grâce aux procédures et doctrine.

## 2.3 Organisation du code:

- Nommage des variables/fonctions en rapport avec leur utilisation.
- Commentaires présents dans tout le code
- Création de la base de données avec uniquement ce qui est nécessaire et nommage cohérent des attributs.