# Classe Validation

**Opération** : pregMatchTele(content : Chaîne de caractères)

**Références croisées :** Cas d’utilisation : Création d’un compte, Modification d’un compte(acheteur), Modification d’un compte (administrateur).

**Préconditions**: Le numéro de téléphone a été saisi par l’usager lors de la création ou la modification d’un compte.

**Post-conditions :** Retourne une valeur booléenne pour confirmer si la chaîne de caractères saisie est un numéro de téléphone.

**Opération** : pregMatchCodePostal(content : Chaîne de caractères)

**Références croisées** : Cas d’utilisation : Création d’un compte, Modification d’un compte(acheteur), Modification d’un compte (administrateur).

**Préconditions** : Le code postal a été saisi par l’usager lors de la création ou la modification d’un compte.

**Post-conditions** : Retourne une valeur booléenne pour confirmer si la chaîne de caractères saisie est un code postal.

**Opération** : pregMatchCredit(content : Chaîne de caractères)

**Références croisées** : Cas d’utilisation : Création d’un compte, Modification d’un compte (acheteur), Modification d’un compte (administrateur).

**Préconditions** : Le numéro de carte de crédit a été saisi par l’usager lors de la création ou la modification d’un compte.

**Post-conditions** : Retourne une valeur booléenne pour confirmer si la chaîne de caractères saisie est un numéro de carte de crédit.

# Classe AdministrateurDAO

**Opération** : read(Chaîne de caractères : codeAdministrateur)

**Références croisées** : Cas d’utilisation : Identification

**Préconditions** : Le code administrateur a été saisi par l’administrateur.

**Post-conditions** : Retourne un objet de type Administrateur contenant les informations sur un administrateur.

# Classe AcheteurDAO

**Opération** : create(x : Acheteur)

**Références croisées** : Cas d’utilisation : Création d’un compte

**Préconditions** : L’usager a saisi toutes les informations sur un compte Acheteur et les informations ont été validées.

**Post-conditions** : Retourne une valeur booléenne pour confirmer l’insertion dans la base de données, les informations sur un acheteur ayant été saisies auparavant par l’usager.

**Opération** : read(Chaîne de caractères : courriel)

**Références croisées** : Cas d’utilisation : Création d’un compte, Modification d’un compte, Identification

**Préconditions** : Le courriel a été saisi par l’usager.

**Post-conditions** : Retourne un objet de type Acheteur contenant les informations sur un acheteur.

**Opération** : update(x : Acheteur)

**Références croisées** : Cas d’utilisation : Modification d’un compte (acheteur), Modification d’un compte (administrateur).

**Préconditions** : Les informations sur un acheteur ont été saisies et validées.

**Post-conditions** : Retourne une valeur booléenne confirmant la mise à jour des informations sur un acheteur dans la base de données.

# Classe AdministrateurDAO

**Opération** : read(Chaîne de caractères : id)

**Références croisées** : Cas d’utilisation : Identification

**Préconditions** : L’usager a saisi son nom d’usager afin d’effectuer une connexion sur le site.

**Post-conditions** : Retourne un objet de type Administrateur contenant les informations sur un administrateur.

# Servlet Login.java

**Opération** : processRequest

**Références croisées** : Identification

**Préconditions** : L’usager saisit son code d’usager et son mot de passe. Les informations sont reçues comme des paramètres de la fonction.

**Post-conditions** :

* Si les champs du formulaire ont été remplis correctement, on vérifie si l’usager existe dans la base de données.
* Si les données saisies par l’usager correspondent à des données dans la base de données, l’usager est connecté sur le site et la procédure retourne la page à afficher dépendant si c’est l’acheteur ou l’administrateur qui a demandé la connexion.

# Servlet Logout.java

**Opération** : processRequest

**Références croisées** : Identification, Gestion des produits, Gestion du panier

**Préconditions** : L’usager doit déjà être connecté.

**Post-conditions** : L’usager est déconnecté et le site revient sur la page index.jsp.

# Servlet CreationCompte

**Opération** : processRequest

**Références croisées** : Gestion d’un compte

**Préconditions** : L’usager a saisi les informations sur son profil personnel.

**Post-conditions** :

* Les données saisies sont validées.
* Si les données sont correctes, le compte de l’usager est inséré dans la base de données.
* L’acheteur est connecté et le site affiche la page du panier.

# Servlet ModifierCompte

**Opération** : processRequest

**Références croisées** : Gestion d’un compte

**Préconditions** : L’usager doit être connecté. La requête reçoit toutes les informations sur un compte de l’acheteur.

**Post-conditions** : Après la validation des données, les informations sont modifiées dans la base de données.

# Servlet MAJcompte

**Opération** : processRequest

**Références croisées** : Gestion d’un compte

**Préconditions** : L’usager doit être connecté. La requête reçoit le courriel de l’acheteur comme paramètre de la méthode.

**Post-conditions** : La méthode récupère toutes les informations sur le compte de l’acheteur avec le courriel reçu comme paramètre et envoie ses informations à la page de modification de compte.