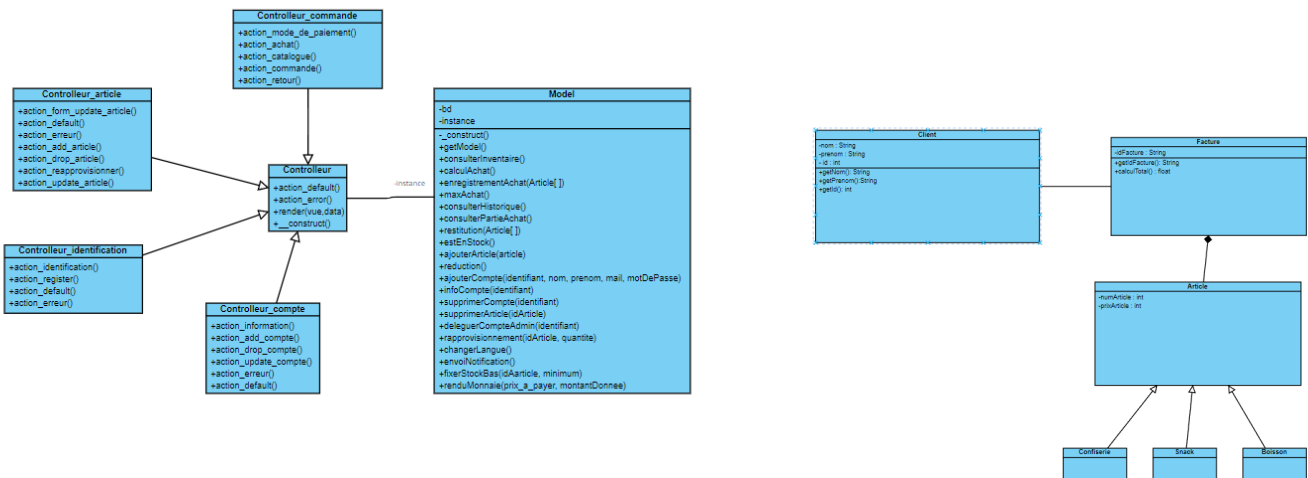


GROUPE : Darius\_Porthos

## SPÉCIFICATION DES FONCTIONS - PROJET CAISSE ENREGISTREUSE AUTOMATIQUE ASSOCIATION BDE IUT P13 Version 0.1 du 15/11/2022

### Spécification des fonctions coté serveur



### Fonctions du modèle:

**connexion(id,mdp : String):** Permet à l'utilisateur de se connecter à son compte client/admin en y inscrivant son identifiant(id : int ) et son mot de passe(mdp : String).

*Ne retourne aucune valeur (void).*

**modeDePaiement():** Définir le mode de paiement en fonction du choix de l'utilisateur.

*Ne retourne aucune valeur (void).*

**consulteInventaire():** permet de consulter le stock ainsi que les informations des produits.

*Retourne un tableau d'objet d'article (Articles[] ).*

**calculAchat(Article[] ) :** Calcule le montant des achats d'articles[] et retourne son prix total.

*Retourne un nombre flottant (float).*

**enregistrementAchats(Achats):** Permet d'enregistrer dans une base de donnée un achat effectué par le client.

*Ne retourne aucune valeur (void).*

**maxAchats(StockArticle)** : Maximise l'effectif d'achat d'un article à son stock d'articles (StockArticle : int).  
*Ne retourne aucune valeur (void).*

**consulterHistorique()** : Permet de consulter la liste des précédents achats d'un compte client depuis une base de données.  
*Retourne un l'historique des achats.*

**consulterPartieAchat()** : Permet de consulter une partie de l'historique, les 20 derniers achats.  
Retourne une partie de l'historique des achats (20 derniers achats)

**restitution(<Article>)** : Permet de réaliser une restitution d'un article (tableau d'article)  
*Retourne un true si le retour a été effectué, false sinon (boolean)*

**estEnStock()** : Vérifie la disponibilité d'un article en stock.  
*Retourne true si un article est disponible en stock, false sinon (boolean)*

**ajouterArticle(Article)** : Ajouter un article (article: Article) donné en paramètre à la base de donnée.  
*Ne retourne aucune valeur (void)*

**reduction(prix, ptsFidélité)** : Permet d'utiliser ,lors de l'achat d'un produit, des points de fidélité (ptsFidélité) pour avoir une réduction sur le prix total d'un achat (prix) .  
*Retourne le prix à payer après avoir appliqué la réduction (double)*

**ajouterCompte(id, nom, prenom, mail, motDePasse)** : ajouter un nouveau compte à la base de donnée en spécifiant son identifiant (id), son nom (nom), son prénom (prénom), son mail (mail) et son mot de passe (motDePasse).  
*Retourne true si le compte a bien été ajouté à la base de donnée, false sinon (boolean)*

**infoCompte(identifiant)** : Récupère les informations d'un compte à partir de son identifiant (idCompte).  
*Retourne un tableau d'objets Informations(<Informations>)*

**supprimerCompte(identifiant)** : Supprime le compte correspondant à l'identifiant entré en paramètre (identifiant).  
*Retourne true si le compte a été supprimé, false sinon (boolean).*

**supprimerArticle(idArticle)** : Supprime l'article correspondant à l'identifiant entré en paramètre (idArticle).  
*Retourne true si l'article a été supprimé, false sinon (boolean).*

**deleguerCompteAdmin(identifiant)** : Le compte superAdmin peut déléguer son rôle à un autre compte admin (CompteAdmin).  
*Retourne true si le superAdmin a délégué son rôle, false sinon (boolean).*

**reapprovisionner(idArticle,quantité):** Met à jour les stocks les quantités (quantité) en paramètre d'un article grâce à l'identifiant de l'article donné (idArticle).

*Retourne true si les stocks ont été mis à jour, false sinon (boolean).*

**changerLangue():** Passe la langue du site web de français à anglais ou inversement.

*Retourne true si le site à changer de langue, false sinon (boolean).*

**envoiNotification() :** Envoie un message en cas de stock bas.

*Retourne une chaîne de caractère alertant le manque de stock (String).*

**fixerStockBas(IdArticle,minimum):** Fixe la quantité entrée en paramètre comme étant le "stock bas" pour l'article dont l'id (le numéro de l'article) est renseigné en paramètre.

*Ne retourne aucune valeur (void).*

**renduMonnaie(prixAPayer, montantDonné) :** Calcule, dans le cas d'un paiement par espèce, la monnaie à rendre en fonction du prix de l'achat (prixAPayer) et le montant qui a été donné (montantDonné).

*Retourne la monnaie à rendre (float).*

## Fonctions des controller :

### Controller:

**\_\_construct():** constructeur par défaut de la classe Controller

**action\_default():** action par défaut du controller

**render(vue, data):** affiche la vue entrée en paramètre et les données (ici un tableau de data) à fournir à la vue.

*Ne retourne aucune valeur (void).*

**action\_error(messageErr):** méthode transmettant un message d'erreur entré en paramètre (messageErr : String)

*Ne retourne aucune valeur (void).*

### Controller\_identification:

**action\_register():** méthode contrôlant la saisie d'enregistrement du compte de l'utilisateur et affiche un message d'erreur si les données saisies sont incorrectes.

*Ne retourne aucune valeur (void)*

**action\_login():** méthode contrôlant la saisie d'identification du compte de l'utilisateur et affiche un message d'erreur si les données saisies sont incorrectes.

*Ne retourne aucune valeur (void)*

**Controller\_article:**

**action\_form\_update\_article()**: méthode contrôlant la mise à jour de la base de données émise par le formulaire des articles.

*Ne retourne aucune valeur (void).*

**action\_add\_article()** : méthode pour contrôler l'ajout d'un article à la base de données et affichant un message d'erreur en cas d'erreur.

*Ne retourne aucune valeur (void).*

**action\_drop\_article()** : méthode pour contrôler la suppression d'un article de la base de données et affichant un message d'erreur en cas d'erreur.

*Ne retourne aucune valeur (void).*

**action\_reapprovisionner()** : méthode pour contrôler le réapprovisionnement d'un article et et affichant un message d'erreur en cas d'erreur.

*Ne retourne aucune valeur (void).*

**action\_update\_article()** : méthode contrôlant la modification d'un article existant de la base et affichant la vue message.

*Ne retourne aucune valeur (void).*

**Controlleur\_compte:**

**action\_information()**: méthode contrôlant et transmettant les informations lié à un identifiant.

*Ne retourne aucune valeur (void).*

**action\_add\_compte()** : méthode pour ajouter un article à la base et affichant la vue message

*Ne retourne aucune valeur (void).*

**action\_drop\_compte()** : méthode pour supprimer un article de la base et affichant la vue message

*Ne retourne aucune valeur (void).*

**action\_update\_compte()** : méthode contrôlant la modification d'un compte existant de la base et affichant la vue message

*Ne retourne aucune valeur (void).*

**Controlleur\_commande:**

**action\_mode\_de\_paiement()** : méthode contrôlant et transmettant les données du formulaire à remplir pour choisir le mode de paiement lors d'un achat.

*Ne retourne aucune valeur (void).*

**action\_achat()**: méthode transmettant les données du formulaire à remplir pour choisir les articles à acheter.

*Ne retourne aucune valeur (void).*

**action\_catalogue():** méthode transmettant la liste des articles qui peuvent être passés en commande.  
*Ne retourne aucune valeur (void).*

**action\_commande():** méthode transmettant les articles sélectionnés dans le panier pour pouvoir passer la commande.  
*Ne retourne aucune valeur (void).*

**action\_retour():** méthode transmettant les données du formulaire à remplir pour la procédure du retour d'un article.  
*Ne retourne aucune valeur (void).*

### Liste des vues :

**view\_begin.php** : Header de chaque vue.

**view\_error404.php** : Visualise une page erreur404 en cas de page web non trouvée.

**view\_end.php** : Footer de chaque vue.

**view\_message.php** : affiche un message défini par la variable \$message (souvent utilisé par la méthode **action\_error** de **Controller**).

**view\_commande.php** : Visualise le formulaire de commande avec le contrôle d'**action\_form\_update\_article()** de **Contrôleur\_article**.

**view\_inscription.php** : Visualise le formulaire d'inscription avec le contrôle d'**action\_register()** et d'**action\_identification()** du contrôleur **Contrôleur\_identification**.

**view\_retour.php** : Visualise le formulaire de retour d'article avec le contrôle d'**action\_retour()** du **contrôleur\_commande** et d'**action\_update\_article** de **Contrôleur\_article**.

**view\_compte.php** : Visualise la page de gestion des comptes.

**view\_inventaire.php** : Visualise la page de l'inventaire qui inclut la liste des comptes avec le contrôle d'**action\_update\_article()**, **action\_drop\_article()** et **action\_add\_article()** de **controller\_article**.

**view\_list\_compte.php** : Visualise la liste des comptes récupérés sur la base de données.

**view\_list\_article.php** : Visualise la liste des articles récupérés sur la base de données.

**view\_informations\_compte.php** : Visualise les informations d'un compte en fonction de son id contrôlé par **action\_informations()** de **controller\_compte** et contrôle la suppression et la modification d'un compte avec **action\_drop\_compte()** et **action\_update\_compte()** de **controller\_compte**.

