

membre: El Hadji Moussa FALL , Ngoné FAYE , Khadidiatou DIARISSO

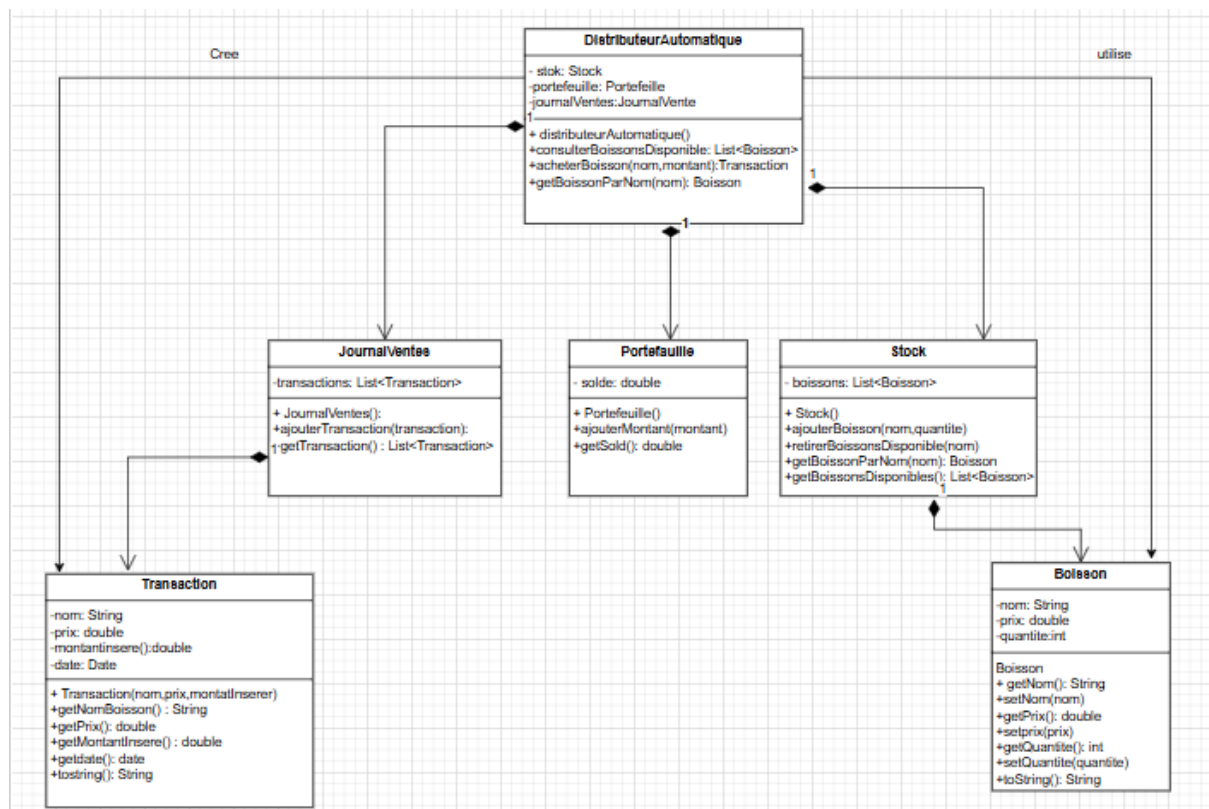
## Rapport du projet : Gestion d'un distributeur automatique de boissons

**Contexte :** Vous devez développer une application Java qui gère un distributeur automatique de boissons. Ce distributeur permet à un utilisateur de consulter la liste des boissons disponibles, d'acheter une boisson s'il dispose d'un montant suffisant et au personnel de recharger le stock des boissons...

### 1. Diagramme de classes

voici notre diagramme de classe qui matérialise les différentes classes de notre Gestionnaire D'un distributeur automatique de boissons.

- nous avons la classe distributeur qui contient une journal de vente , une portefeuille et un stock
- le journal de vente contient qui les transactions
- la classe boisson aussi contient le stock



titre:Diagramme de classes UML complet

### 2. Tests unitaires

voici les différentes tests unitaire qui ont été réalisées (les tests unitaires se trouvent sur env/page/test)

#### pour la class Distributeur

- Initialisation du distributeur et ajout de boissons dans le stock
- Vérifier que `consulterBoissonsDisponibles` retourne une liste vide si aucun stock
- Vérifier un achat réussi

- Vérifier un achat avec montant exact
- Vérifier une exception pour boisson non trouvée
- Vérifier une exception pour rupture de stock
- Vérifier une exception pour montant insuffisant
- Vérifier une recharge de stock réussie
- Vérifier une recharge avec une nouvelle boisson
- Vérifier une exception pour quantité négative lors de la recharge

#### **Pour la classe journal Vente**

- Vérifier que le constructeur initialise une liste vide
- Vérifier l'ajout d'une transaction
- Vérifier l'ajout de plusieurs transactions
- Vérifier que getTransactions retourne une copie de la liste
- Vérifier l'ajout d'une transaction null
- Vérifier que getTransactions retourne une liste vide après création
- Vérifier l'ajout de transactions identiques

#### **Pour la classe portefeuille nous avons**

- Vérifier que le constructeur initialise le solde à 0
- Vérifier l'ajout d'un montant positif
- Vérifier l'ajout de plusieurs montants positifs
- Vérifier l'ajout d'un montant zéro
- Vérifier une exception pour un montant négatif
- Vérifier la précision des calculs avec des montants décimaux
- Vérifier que getSolde retourne toujours la valeur correcte
- Vérifier que le solde reste inchangé après une tentative d'ajout négatif

#### **Pour la classe stock aussi nous avons**

- Vérifier que le constructeur initialise une liste vide
- Vérifier l'ajout d'une nouvelle boisson
- Vérifier l'augmentation de la quantité d'une boisson existante
- Vérifier une exception pour quantité négative lors de l'ajout
- Vérifier le retrait d'une boisson
- Vérifier une exception pour retirer une boisson non trouvée
- Vérifier une exception pour retirer une boisson en rupture de stock
- Vérifier getBoissonParNom avec une boisson existante
- Vérifier getBoissonParNom avec une boisson inexistante

maintenant nous allons partir sur les test d'acceptation:

### 3. Tests d'acceptance

voici les différents tests d'acceptance qui se trouvent sur (env.page.test)

- Consulter les boissons disponibles avec un stock initial
- Consulter les boissons disponibles avec un stock vide
- Acheter une boisson avec un montant suffisant
- Acheter une boisson avec le montant exact
- Acheter une boisson avec un montant insuffisant
- Acheter une boisson non existante
- Acheter une boisson en rupture de stock
- Recharger le stock d'une boisson existante
- Recharger le stock avec une nouvelle boisson
- Recharger le stock avec une quantité nulle
- Recharger le stock avec une quantité négative
- Acheter plusieurs boissons consécutivement
- Consulter les boissons après un achat
- Consulter les boissons après une recharge

**Pour la classe transactions nous avons aussi**

- Création d'une transaction pour un achat avec montant suffisant
- Création d'une transaction avec le montant exact
- Vérification de la date dans une transaction
- Enregistrement d'une transaction dans le journal des ventes
- Enregistrement de plusieurs transactions dans le journal
- Pas de transaction pour un achat avec montant insuffisant