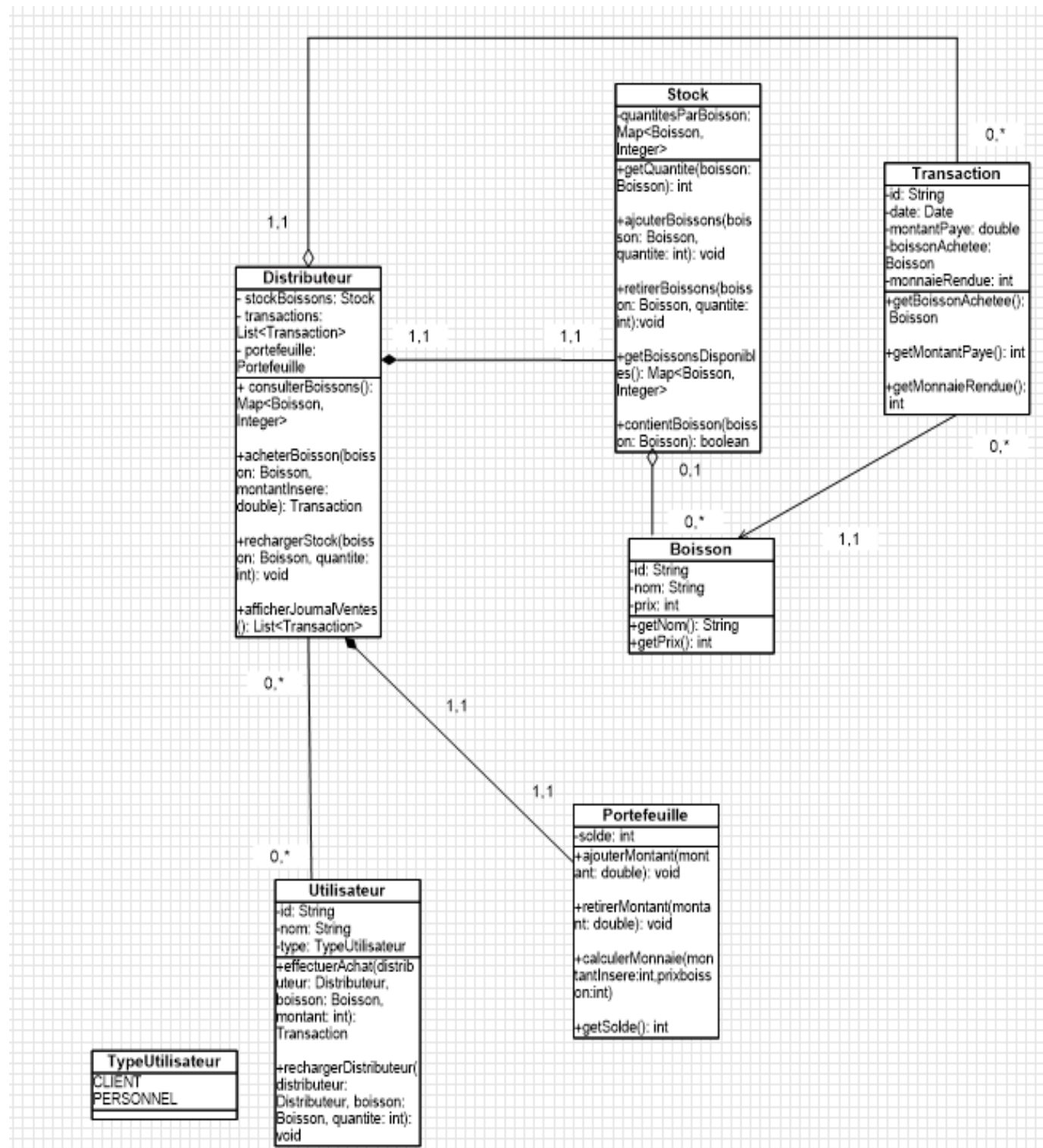


ARAME DIENG

IBRAHIMA THIAM

MARIE SOLANGE SIGA T. DIOUF

DIAGRAMME DE CLASSE



ROLE DE CHAQUE CLASSE

Distributeur :

_Composant central du distributeur, responsable de la coordination de la gestion des stocks, du traitement des achats, de la gestion de l'argent et de la tenue de l'historique des transactions. Il sert d'interface principale pour les interactions avec les utilisateurs.

Stock :

Gérer l'inventaire des boissons, faire le suivi de la quantité de chaque boisson disponible dans le distributeur et gérer les ajouts ou les retraits du stock.

Boisson :

Représente une boisson individuelle vendue par le distributeur, définie par son identifiant unique, son nom et son prix.

Transaction :

Enregistre les détails d'un seul achat, y compris la boisson achetée, la quantité insérée, la monnaie rendue et l'horodatage.

Utilisateur :

Représente une personne qui interagit avec le distributeur, soit en tant que client achetant des boissons, soit en tant que personnel réapprovisionnant le distributeur.

TypeUtilisateur :

Définit le type d'utilisateur (client ou personnel) pour contrôler l'accès à des actions spécifiques, telles que le réapprovisionnement.

Portefeuille :

Gère l'argent à l'intérieur du distributeur, gère les montants insérés, calcule la monnaie et suit le solde de trésorerie de la machine.

LES RELATIONS PRINCIPALES ENTRE LES CLASSES

Distributeur-Stock :

Le Distributeur contient exactement un objet Stock, qui fait partie intégrante de la machine et ne peut pas exister indépendamment.

Distributeur utilise Stock pour gérer la disponibilité des boissons et mettre à jour les stocks lors d'achats ou de réassorts.

Distributeur-Portefeuille :

Le distributeur contient un objet Portefeuille, qui est un composant central de la machine et ne peut exister séparément.

Distributeur s'appuie sur Portefeuille pour traiter les paiements et calculer la monnaie des transactions.

Distributeur-Transaction :

Le distributeur tient à jour une collection d'objets Transaction pour consigner l'historique des achats. Les transactions peuvent exister indépendamment.

Distributeur crée et stock des objets Transaction pour chaque achat et les récupère pour l'affichage de l'historique.

Distributeur-Utilisateur :

Plusieurs objets Utilisateur (clients ou personnel) peuvent interagir avec plusieurs objets Distributeur, ce qui représente une relation de non propriété lâche.

L'utilisateur effectue des actions telles que l'achat ou le réapprovisionnement sur Distributeur.

Stock-Boisson :

Le stock gère plusieurs objets Boisson, en suivant leurs quantités. Les objets Boisson peuvent exister indépendamment du stock.

Stock utilise Boisson pour représenter les boissons qu'elle gère.

Boisson-Transaction :

Chaque transaction fait référence exactement à une boisson (la boisson achetée), tandis qu'une boisson peut être impliquée dans plusieurs transactions.

Enregistrement de la transaction qui a permis d'acheter Boisson.

LISTES DES TESTS

- **TESTS UNITAIRES**

Fichier BoissonTest

1.testGetId(): Vérifie que l'ID de la boisson est correctement retourné.

2.testGetNom(): Vérifie que le nom de la boisson est correctement retourné.

3.testGetPrix(): Vérifie que le prix de la boisson est correctement retourné.

4.testEqualsSameBoisson(): Teste l'égalité entre deux objets Boisson ayant le même ID.

5.testEqualsDifferentBoisson(): Teste l'inégalité entre deux objets Boisson ayant des IDs différents.

Fichier PortefeuilleTest

- 6. testPortefeuilleInitialSolde()** : Vérifie que le solde initial du portefeuille est correct.
- 7. testAjouterMontant()** : Teste l'ajout d'un montant positif au portefeuille.
- 8. testRetirerMontantSuffisant()** : Teste le retrait d'un montant lorsque le solde est suffisant.
- 9. testRetirerMontantInsuffisant()** : Teste le retrait d'un montant lorsque le solde est insuffisant.
- 10. testCalculerMonnaie()** : Teste le calcul correct de la monnaie à rendre

Fichier StockTest

- 11. testAjouterBoissons()**: Teste l'ajout de nouvelles boissons au stock.
- 12. testAjouterBoissonsExistantes()**: Teste l'ajout de boissons existantes, vérifiant l'incrémentation de la quantité.
- 13. testRetirerBoissonsSuffisantes()**: Teste le retrait d'une quantité suffisante de boissons.
- 14. testRetirerBoissonsInsuffisantes()**: Teste le retrait d'une quantité insuffisante de boissons.
- 15. testContientBoissonTrue()**: Teste si le stock contient une boisson existante.
- 16. testGetQuantite()**: Teste l'obtention de la quantité d'une boisson spécifique.
- 17. testGetBoissonsDisponibles()**: Teste si la map des boissons disponibles est correctement retournée (et est une copie).

Fichier TransactionTest

- 18. testVerificationInfoTransaction()**: Vérifie si une transaction est créée avec les bonnes informations et si les getters fonctionnent.

Fichier DistributeurTest

- 19. testConsulterBoissonsAvecStock()**: Teste l'affichage des boissons après rechargement du stock.
- 20. testAcheterBoissonSuccess()**: Teste l'achat réussi d'une boisson avec montant suffisant.
- 21. testAcheterBoissonMontantInsuffisant()**: Teste l'achat d'une boisson avec un montant insuffisant.

22. testAcheterBoissonRuptureDeStock(): Teste l'achat d'une boisson en rupture de stock.

23. testAcheterBoissonNonExistante(): Teste l'achat d'une boisson qui n'existe pas dans le stock.

24. testRechargerStock(): Teste la recharge correcte du stock du distributeur.

25. testAfficherJournalVentesApresAchat(): Teste l'affichage du journal des ventes après un achat.

Fichier UtilisateurTest

26. testEffectuerAchatClientSuccess(): Teste un achat réussi par un client (y compris la mise à jour des soldes).

27. testEffectuerAchatClientSoldeInsuffisant(): Teste un achat par un client avec un solde insuffisant.

28. testEffectuerAchatPersonnelNonPermis(): Teste qu'un personnel ne peut pas effectuer d'achat.

29. testRechargerDistributeurPersonnelSuccess(): Teste la recharge du distributeur par un membre du personnel.

30. testRechargerDistributeurClientNonPermis(): Teste qu'un client ne peut pas recharger le distributeur.

• TESTS D'ACCEPTANCE

Fichier DistributeurSetps et distrubuteur.feature

Scénario 1: Achat Réussi d'une Boisson par un Client

Scénario 2: Achat avec Montant Insuffisant

Scénario 3: Achat d'une Boisson en Rupture de Stock

Scénario 4: Recharge du Stock par le Personnel

Scénario 5: Un Client Tente de Recharger le Stock (Échec)

Scénario 6: Consultation des Boissons Disponibles

Scénario 7: Affichage du Journal des Ventes (Après Achat)

Scénario 8: Affichage du Journal des Ventes (Aucun Achat)

Scénario 9: Achat d'une Boisson à Prix Exact

Scénario 10: Portefeuille du Distributeur après Multiples Ventes

Scénario 11: Tentative de Retrait de Montant Négatif du Portefeuille

Scénario 12: Tentative d'Ajout de Montant Négatif au Portefeuille

Scénario 13: Achat d'une Boisson après Déplétion Complète du Stock

Scénario 14: Recharge de Stock d'une Nouvelle Boisson

**Scénario 15: Vérification de la Consistance du Solde du Client
après un Achat Échoué**