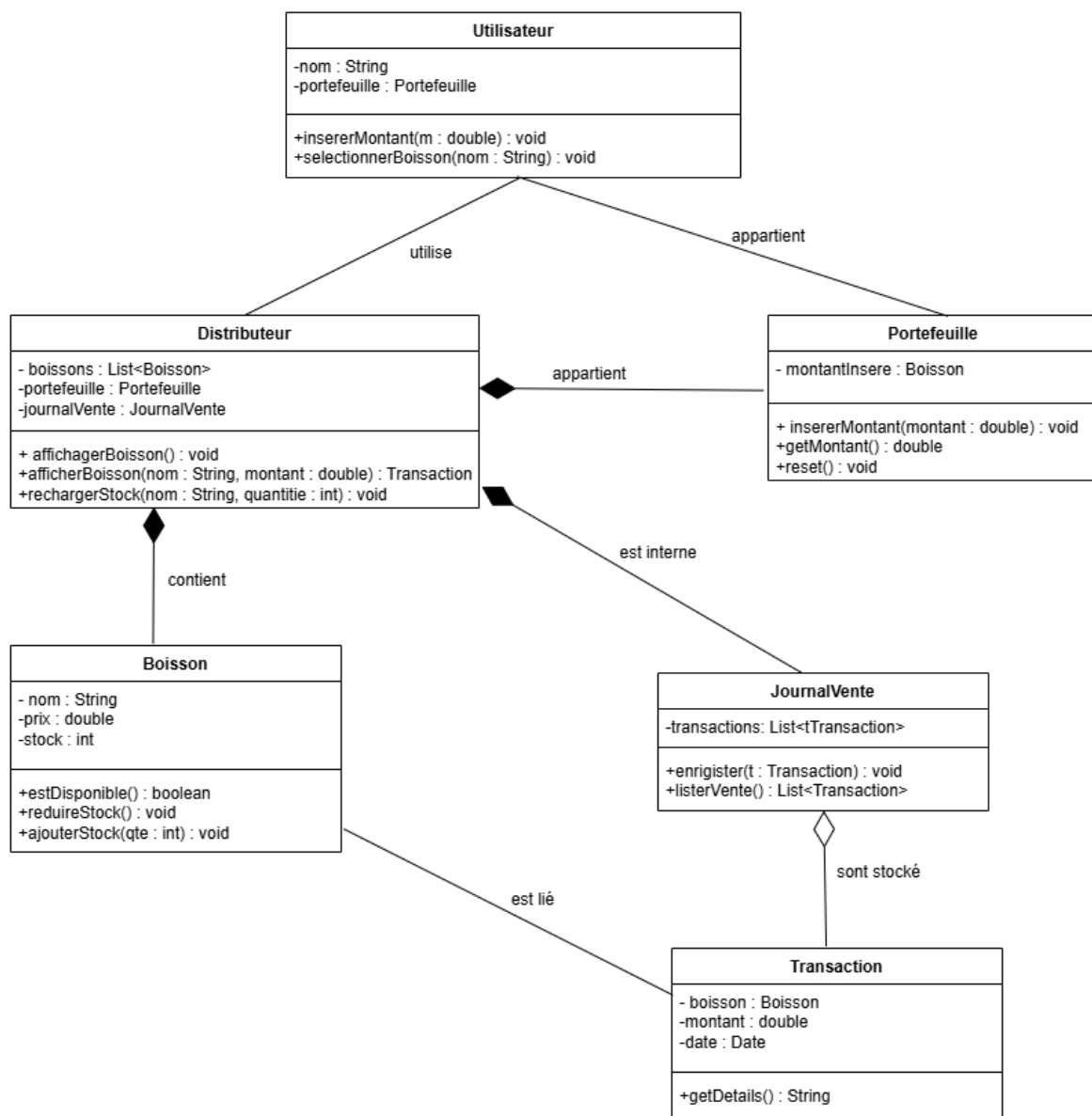


PROJET QUME : Gestion d'un distributeur automatique de boissons

👤 Membres du groupe :

- Mohamet Lamine Ba
- Mouhamadou Al Bachir Ba

A. Diagramme de classes UML du distributeur automatique :



Le diagramme représente les entités suivantes :

- **Distributeur** : point d'entrée principal du système
- **Boisson** : chaque produit avec son prix et son stock
- **Transaction** : historique des achats
- **JournalVente** : enregistre toutes les transactions
- **Portefeuille** : gestion des montants insérés
- **Utilisateur** : acteur simulant les achats

◆ Utilisateur

Représente une personne qui utilise le distributeur pour acheter des boissons.

- Il possède un Portefeuille
- Il peut insérer de l'argent et sélectionner une boisson
- Il utilise le Distributeur pour effectuer un achat.

◆ Distributeur

Cœur du système, c'est lui qui gère :

- La liste des Boisson disponibles
- Le Portefeuille temporaire pour stocker l'argent inséré
- Le JournalVente pour enregistrer les achats
- Les opérations d'achat, d'affichage, et de rechargement de stock

Il contient les Boisson, possède un Portefeuille, et enregistre des Transaction dans un JournalVente.

◆ Boisson

Classe représentant un produit en vente.

- Comporte un nom, un prix et un stock
 - Fournit des méthodes pour vérifier la disponibilité, réduire ou recharger le stock
- Elle est contenue dans le Distributeur.

◆ Portefeuille

Gère l'argent inséré par l'utilisateur ou stocké temporairement par le distributeur.

- Permet d'insérer, consulter ou réinitialiser un montant

Il appartient soit à un Utilisateur, soit à un Distributeur.

◆ Transaction

Représente un achat effectué par un utilisateur.

- Contient la Boisson achetée, le montant payé, et la date
- Permet d'obtenir un résumé de l'achat

Chaque Transaction est créée lors d'un achat et enregistrée dans le JournalVente.

◆ JournalVente

Historique interne au distributeur.

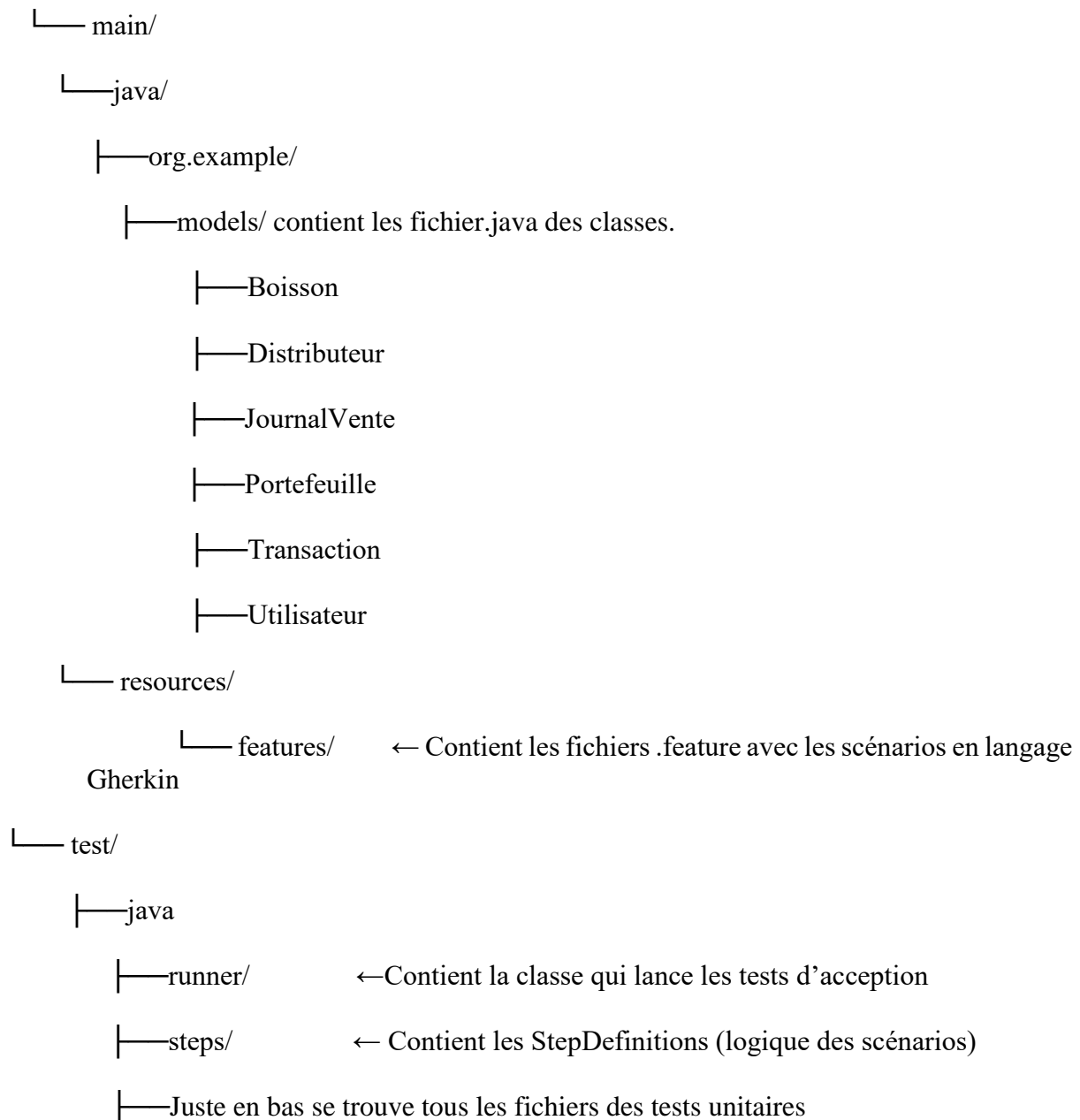
- Contient toutes les Transaction effectuées
- Permet d'enregistrer ou de lister les ventes

Il est lié au Distributeur et stocke les Transaction.

B. Structure du Projet :

Projet_QUME_L3/

|— src/



C. Technologies utilisées :

- Java 17
- Maven
- JUnit 5
- Cucumber 7.14.0
- IntelliJ IDEA

D. Liste avec plus de 30 tests unitaires proposés :

Les tests suivants vérifient le bon fonctionnement du système :

Classe BoissonTest :

- testCreationBoisson()
- testEstDisponibleTrue()
- testEstDisponibleFalse()
- testReducStock()
- testAjouterStock()
- testPrixBoisson()

Classe PortefeuilleTest :

- testInitialMontant()
- testAAssezPourTrue()
- testAAssezPourFalse()
- testInsererMontant()
- testReset()
- testInsererMontantNegatif()

Classe DistributeurTest :

- testChercherBoissonExistante()
- testChercherBoissonInexistante()
- testAchatReussi()
- testAchatMontantInsuffisant()
- testAchatBoissonInexistante()
- testAchatBoissonRupture()
- testRechargeStock()
- testRechargeStockBoissonInconnue()

Classe TransactionTest :

- tesTransactionData()
- testTransactionDetailFormat()
- testTransactionAvecMontantNegatif()
- testTransactionHorodatageNonNull()
- testTransactionBoissonNonNull()

Classe JournalVenteTest :

- testJournalVideAuDebut()
- testEnregistrerTransaction()
- testListerTransactionsContientTransaction()

Classe UtilisateurTest :

- testCreationUtilisateur()
- testInsererMontantDansPortefeuille()
- testSelectionnerBoissonReussie()
- testSelectionnerBoissonEchecMontantInsuffisant()

E. Liste des 15 scénarios en langage Gherkin :

Scenario: 1 - Affichage des boissons disponibles

Given le distributeur contient les boissons suivantes
When l'utilisateur consulte la liste des boissons
Then il voit la boisson "Coca"

Scenario: 2 - Achat réussi d'une boisson

Given le distributeur contient une boisson "Fanta" à 400 FCFA avec 5 en stock
When l'utilisateur insère 500 FCFA et achète "Fanta"
Then l'achat est réussi et le stock diminue de 1

Scenario: 3 - Achat avec montant insuffisant

Given le distributeur contient une boisson "Fanta" à 400 FCFA avec 5 en stock
When l'utilisateur insère 100 FCFA et tente d'acheter "Fanta"
Then l'achat échoue avec message "Montant insuffisant."

Scenario: 4 - Achat d'une boisson en rupture de stock

Given le distributeur contient une boisson "Sprite" à 300 FCFA avec 0 en stock
When l'utilisateur insère 500 FCFA et tente d'acheter "Sprite"
Then l'achat échoue avec message "Boisson en rupture de stock."

Scenario: 5 - Recharge de stock

Given le stock de "Eau" est à 0
When le personnel recharge "Eau" avec 5 unités
Then le stock de "Eau" est maintenant à 5

Scenario: 6 - Achat d'une boisson inexistante

When l'utilisateur tente d'acheter "Thé"
Then l'achat échoue avec message "Boisson introuvable."

Scenario: 7 - Transaction enregistrée après achat

Given une boisson "Coca" est achetée
Then une transaction est enregistrée dans le journal

Scenario: 8 - Montant inséré visible dans le portefeuille

When l'utilisateur insère 1000 FCFA
Then le portefeuille contient 1000 FCFA

Scenario: 9 - Réinitialisation du portefeuille après achat

Given le portefeuille contient 500 FCFA

When un achat est effectué
Then le portefeuille est réinitialisé à 0

Scenario: 10 - Affichage du journal de vente
Given deux boissons sont achetées
When l'utilisateur consulte le journal des ventes
Then il voit 2 transactions enregistrées

Scenario: 11 - Recharge avec quantité négative
When le personnel tente de recharger "Coca" avec -5 unités
Then le stock ne change pas

Scenario: 12 - Création d'une nouvelle boisson
When le personnel ajoute une nouvelle boisson "Thé" à 300 FCFA
Then "Thé" apparaît dans la liste des boissons

Scenario: 13 - Tentative d'achat avec portefeuille vide
Given le portefeuille est vide
When l'utilisateur tente d'acheter "Fanta"
Then l'achat échoue avec message "Montant insuffisant."

Scenario: 14 - Recharge d'une boisson inexistante
When le personnel tente de recharger "Jus d'ananas"
Then le système affiche "Boisson non trouvée."

Scenario: 15 - Affichage formaté des boissons
Given le distributeur contient "Fanta" à 400 FCFA avec 3 en stock
When l'utilisateur consulte la liste
Then il voit "Fanta - 400 FCFA (3 en stock)"

Conclusion

Ce projet nous a permis de mettre en œuvre une architecture orientée objet, des tests unitaires robustes avec JUnit, et des tests d'acceptance via Cucumber. Il illustre l'importance de la validation continue et du découpage fonctionnel dans le développement logiciel.