

Rapport de développement – Simulation hospitalière (PCO)

Ce document présente la démarche suivie pour implémenter la simulation hospitalière en appliquant le **Test-Driven Development (TDD)** et décrit les décisions techniques adoptées pour respecter l'intégralité des tests fonctionnels et de concurrence.

1. Méthodologie

Phase	Description
Exécution initiale des tests	Les tests fournis ont été lancés dès le départ afin d'identifier l'ensemble des échecs et de cartographier le travail à réaliser.
Boucle TDD	Pour chaque test en échec : 1. <i>Comprendre</i> le comportement attendu. 2. Implémenter la modification minimale permettant de faire passer ce test. 3. Relancer la suite complète afin de vérifier l'absence de régression. 4. Refactoriser le code et ajouter les garde-fous nécessaires (vérification d'arguments, conditions de sortie, etc.).
Documentation incrémentale	Des commentaires ont été ajoutés au fur et à mesure directement dans le code, afin d'explicitier l'intention de chaque changement (verrouillage, facturation, contrôle de stock). Cette discipline évite le phénomène de « code muet » observé lors du laboratoire précédent.
Intégration finale	Une fois la totalité des tests dans l'état <i>vert</i> , le point d'entrée <i>main</i> a été exécuté. Les erreurs restantes (incohérences de fonds, problèmes de concurrence) ont alors été corrigées lors d'un dernier passage.

2. Synchronisation : choix des **mutex**

Les seules zones réellement critiques concernent l'accès concurrent aux variables *money* et *unpaidBills*. Un verrou exclusif (*PcoMutex*) suffit à garantir l'exclusion mutuelle ; l'utilisation d'un sémaphore, plus lourde, n'apporterait pas de bénéfice supplémentaire dans ce contexte.

3. Validation

Invariant vérifié	Valeur attendue	Valeur obtenue
Somme globale des fonds (hors cotisations assurance)	4 700 €	4 700 €
Nombre total de patients	900	900
Suite Google Test	31 / 31 tests verts	✓

Ces résultats confirment la conservation des ressources ainsi que la bonne gestion de la concurrence.

4. Utilisation de l'IA

Aucune portion de code n'a été générée automatiquement. L'intelligence artificielle (ChatGPT 5) a uniquement été sollicitée pour la mise en forme ainsi que la reformulation la présente documentation. Le but étant de rendre la documentation concise et professionnel.

5. Mot de la fin

Nous avons fais l'erreur sur le labo précédant de ne pas commenter nôtre code. Nous avons fait tout l'inverse sur ce labo. Lorsque l'on travail sur le code écrit par quelqu'un d'autre, comprendre ce qu'a voulu faire le développeur avant nous n'est pas chose aisée. L'utilisation massifs des commentaires permet de se plonger dans le code plus facilement.

Aussi, les occasions de faire du vrai TDD ne sont pas nombreuses. Sur ce laboratoire, les tests on rendu bien plus simple l'implémentation des fonctionnalités. Si le temps le permettait, pleins d'autres testes pourraient être ajoutés par exemple sur les facturations et l'argent de manière générale.

Nous avons perdu passablement de temps dessus lorsque nous avons réalisé que les fonds n'étaient pas conforme en fin de programme.