**Documentation du Projet "LabXpert"**

**Description**

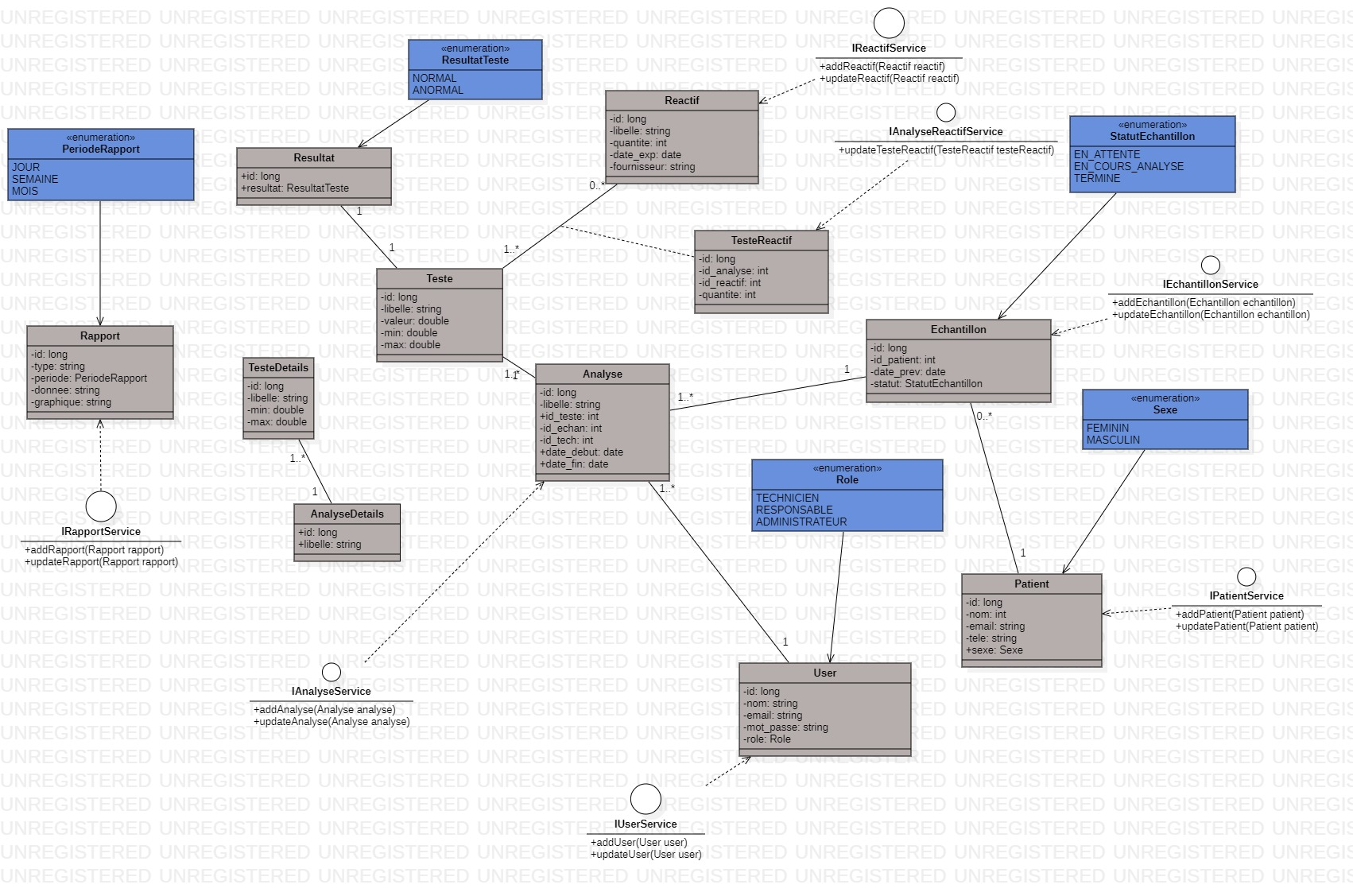
Le projet "LabXpert" a pour ambition de mettre en place un système de gestion intégrale au sein du laboratoire médical TechLab. L'objectif principal est d'optimiser les opérations du laboratoire en améliorant l'efficacité et la précision dans le traitement des analyses médicales. Le système englobe divers aspects de la gestion, allant du suivi des échantillons à la gestion des résultats, visant à fournir un service plus rapide et plus précis aux patients.

**Objectif**

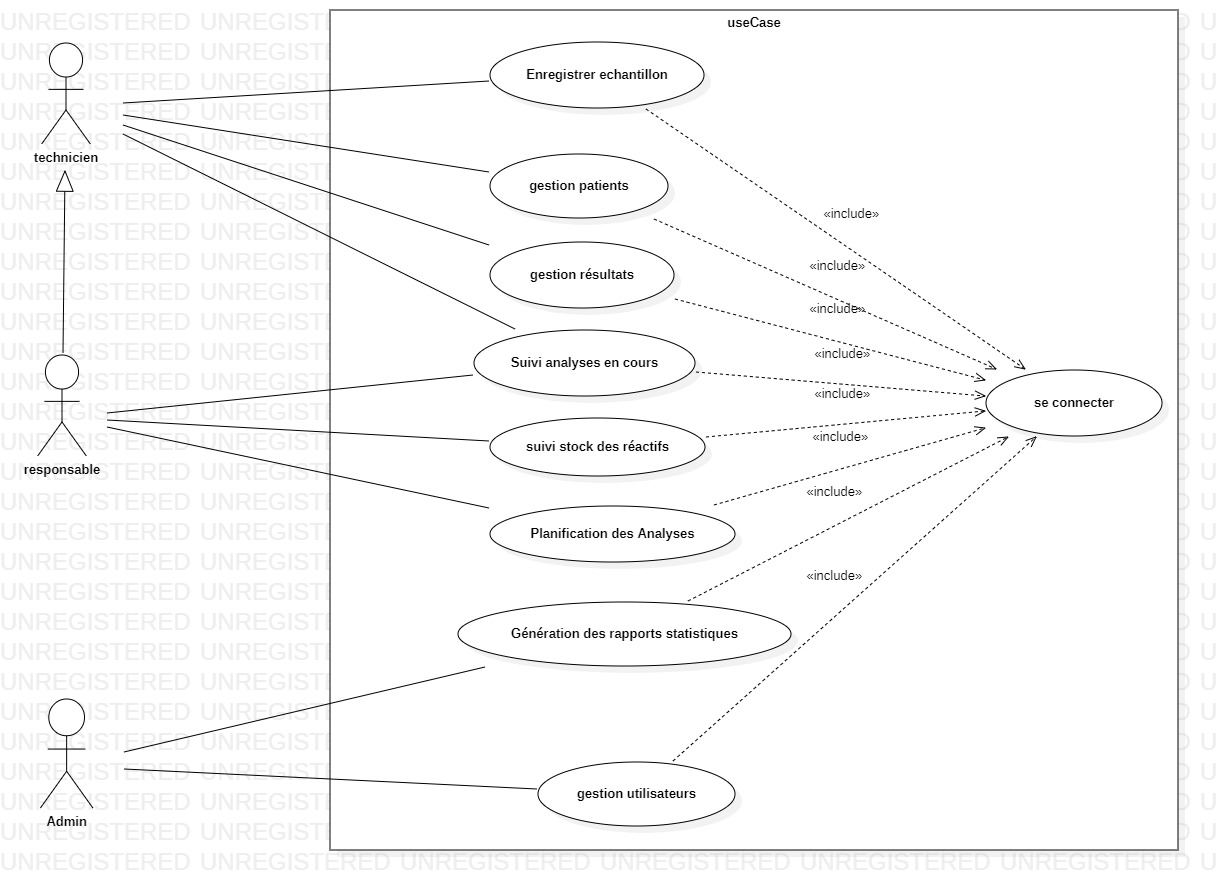
En créant ce projet, l'objectif principal est de renforcer mes compétences en développement, en mettant en œuvre une stack technique avancée et adaptée aux besoins spécifiques du laboratoire médical TechLab. L'utilisation de Java avec Spring Boot API RESTful permet de développer une interface robuste, tandis que PostgreSQL garantit une gestion fiable des données. Les outils tels que Maven, Jenkins et Git/Github facilitent le développement, le déploiement continu et la gestion des versions. SWAGGER offre une documentation claire de l'API, renforçant la communication et l'accessibilité. Cette initiative vise à acquérir une expertise approfondie dans l'analyse des besoins, la conception de bases de données, la configuration d'environnements de développement, et l'implémentation backend.

**Les diagrammes**

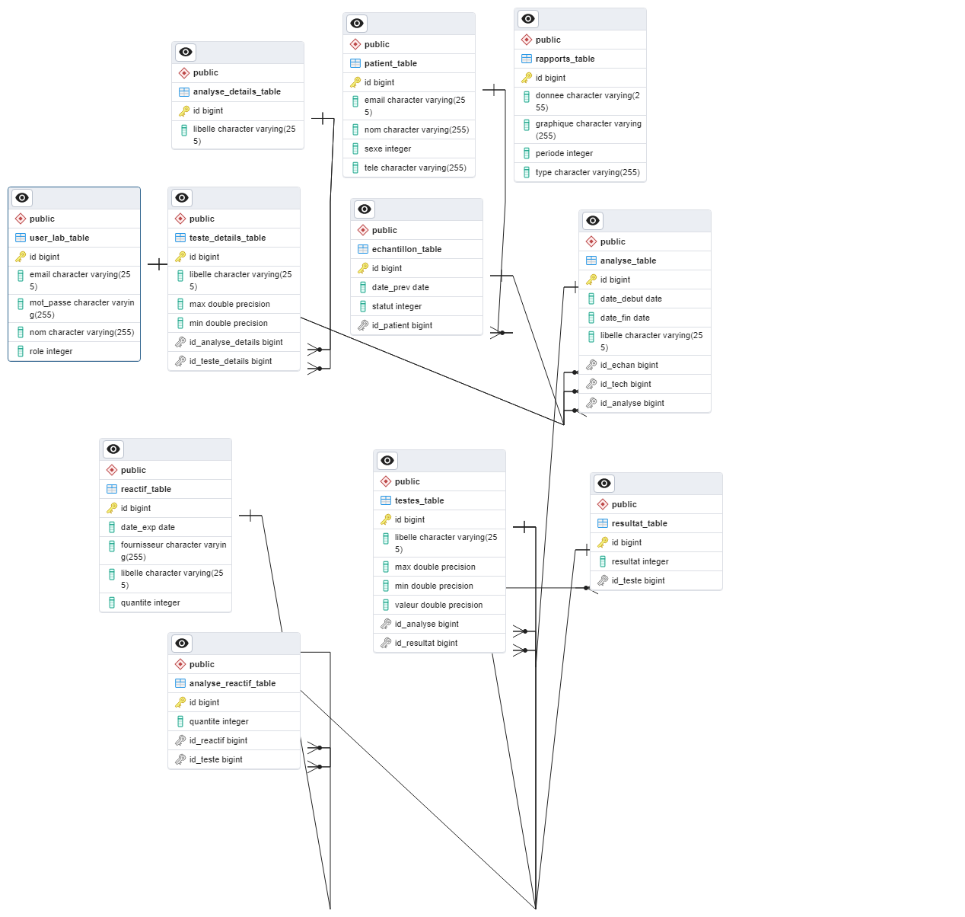
1. Diagramme de classe :



1. Diagramme de cas d’utilisation :



1. Database :



**Principales étapes :**

1. Modéliser les cas d'utilisation et les diagrammes de classes (UML) :

Comprendre les besoins des utilisateurs et définir la structure du système grâce aux cas d'utilisation et aux diagrammes de classes UML. Ces éléments fournissent une base essentielle pour la conception et le développement ultérieurs du système, en identifiant les fonctionnalités attendues et les relations entre les entités.

1. Développer les entités JPA (Mapping entre Entités) :

Créer des entités Java représentant les objets métier dans la base de données. Grâce à Java Persistance API (JPA), établir un mapping entre ces entités et les tables de la base de données. Cette approche permet de manipuler les données de manière orientée objet, assurant une cohérence entre le modèle de données et le modèle objet de l'application.

1. Création des repositories avec Jeu d'insertion :

Mettre en place des repositories pour interagir avec la base de données. Définir des méthodes dans les repositories pour effectuer des opérations telles que la récupération, la mise à jour et la suppression de données. Intégrer un jeu d'insertion dans cette phase facilite le développement, le débogage et les tests ultérieurs en peuplant la base de données avec des données de test.

**Les Outils Utilisés**

- Spring Boot API RESTful

- Apache Maven

- PostgreSQL

- Apache Tomcat

- Git et Github