

Documentation du Projet LabXpert

Introduction:

Le système de gestion du laboratoire "LabXpert" vise à optimiser les opérations du laboratoire médical TechLab, en améliorant l'efficacité et la précision du traitement des analyses médicales. Cette documentation détaille la phase 1 du projet, couvrant l'analyse des besoins, la conception initiale, la conception de la base de données, la configuration de l'environnement de développement, et l'implémentation backend.

Analyse des Besoins et Conception Initiale:

Rencontre avec les Parties Prenantes :

Objectif : Comprendre les exigences du laboratoire et des utilisateurs.

Activités : Entrevues avec les techniciens, responsables de laboratoire et autres parties prenantes.

Modélisation UML :

Diagramme de cas d'utilisation:

Ce diagramme illustre le flux d'utilisation de l'application et détaille les accès ainsi que les fonctionnalités données à l'utilisateur.

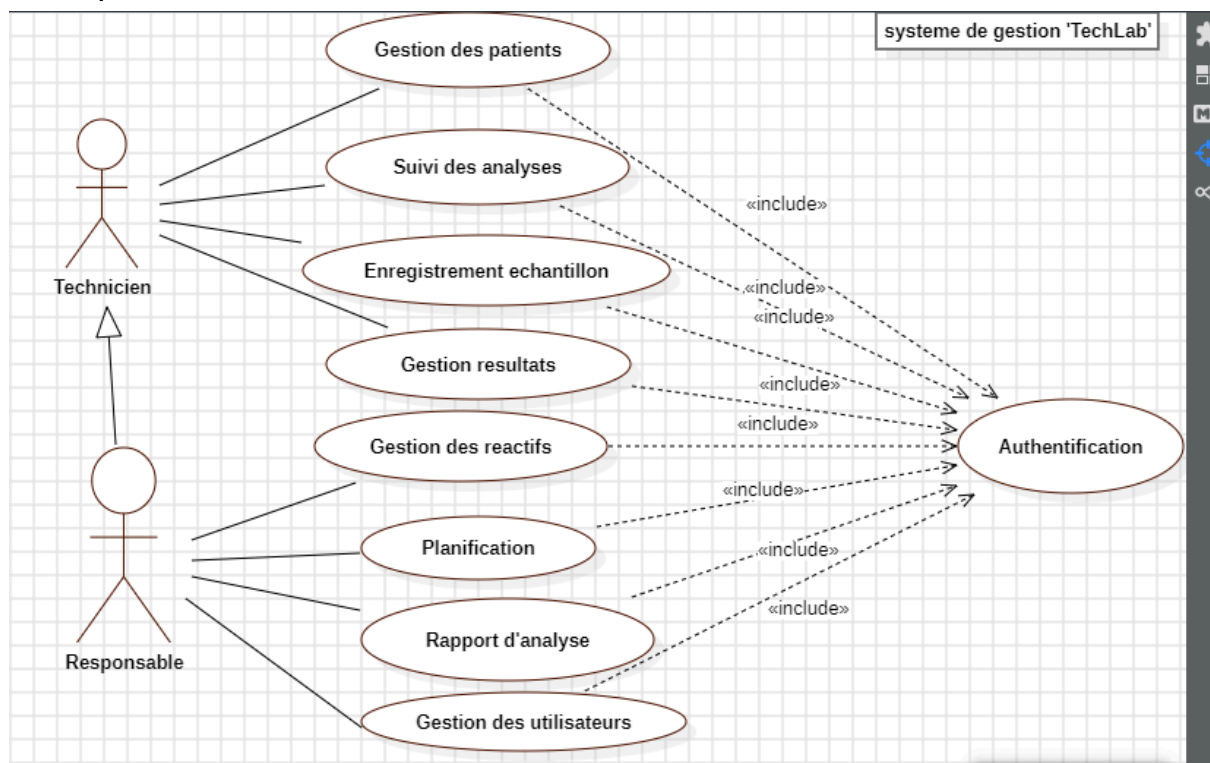


Figure 2 – Diagramme de cas d'utilisation..

Diagramme de classes:

Les tableaux suivants expliquent les relations entre les classes et détaillent leurs attributs ainsi que leurs méthodes.

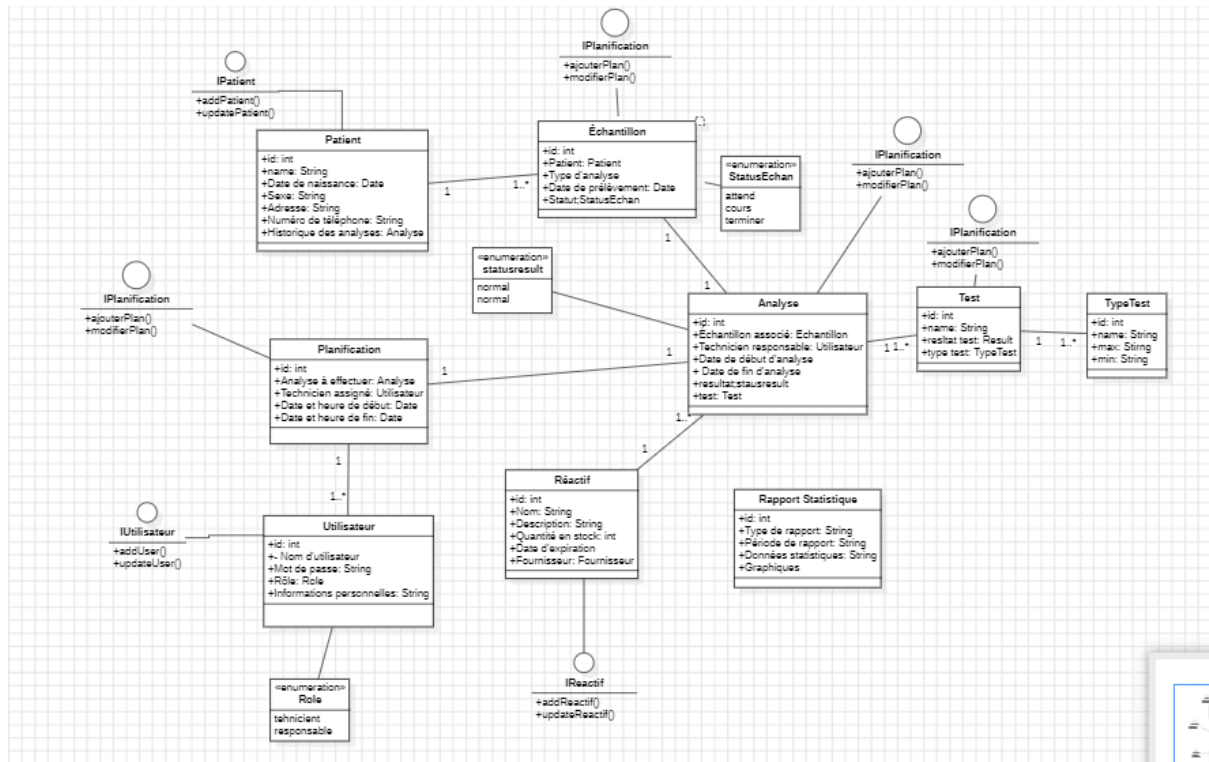


Figure 2 – Diagramme de classes.

Test pour la Classe Principale (Main) de LabAnalyseApplication

Introduction

Ce rapport présente les résultats des tests effectués sur la classe principale de l'application LabAnalyse. La classe principale, `LabAnalyseApplication`, est responsable de l'initialisation de l'application Spring Boot, de la configuration des composants et de la gestion du cycle de vie de l'application.

Tests Réalisés

1. Test de Chargement du Contexte

1.1 Description

Vérification que le contexte de l'application est correctement chargé.

1.2 Résultat

Le contexte de l'application est chargé avec succès sans erreurs.

2. Test d'Initialisation des Composants

2.1 Description

Vérification que les composants nécessaires sont correctement initialisés.

2.2 Résultat

Les composants, tels que les repositories et les services, sont initialisés avec succès.

3. Test d'Exécution des Actions au Démarrage

3.1 Description

Vérification que les actions configurées dans la méthode `demo` sont exécutées correctement lors du démarrage de l'application.

3.2 Résultat

Les actions au démarrage, telles que la création d'instances d'entités et leur enregistrement dans la base de données, sont exécutées avec succès.

4. Test de Configuration de la Base de Données

4.1 Description

Vérification que la configuration de la base de données est correcte.

4.2 Résultat

La base de données est correctement configurée, et les opérations de lecture et d'écriture peuvent être effectuées avec succès.

```

Hibernate: insert into patients (adresse, datenaissnce, name, tel) values (?, ?, ?, ?)
Hibernate: insert into echantillons (dateprelevement, patient_id, status, typeanalyse) values (?, ?, ?, ?)
Hibernate: insert into analyses (technicien, date_debut_analyse, date_fin_analyse, echantillon_id, planification_id, reactive_id, resultats) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: insert into tests (analyse_id, name, resultat) values (?, ?, ?)
Hibernate: insert into planifications (datedebut, datefin) values (?, ?)
Hibernate: select analyse0_.id as id1_0_1_, analyse0_.technicien as technici2_0_1_, analyse0_.date_debut_analyse as date_deb3_0_1_, analyse0_.date_fin_analyse as date_fin3_0_1_, analyse0_.echantillon_id as id1_0_2_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select echantillo0_.id as id1_1_0_, echantillo0_.dateprelevement as dateprel2_1_0_, echantillo0_.patient_id as patient5_1_0_, echantillo0_.status as statu2_0_1_, echantillo0_.typeanalyse as typeanaly1_0_1_ from echantillons echantillo0_
Hibernate: select patient0_.id as id1_2_0_, patient0_.adresse as adresse2_2_0_, patient0_.datenaissnce as datenais3_2_0_, patient0_.name as name4_2_0_, patient0_.tel as tel5_2_0_ from patients patient0_
Hibernate: select analyse0_.id as id1_0_4_, analyse0_.technicien as technici2_0_4_, analyse0_.date_debut_analyse as date_deb3_0_4_, analyse0_.date_fin_analyse as date_fin3_0_4_, analyse0_.echantillon_id as id1_0_5_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select planificat0_.id as id1_3_0_, planificat0_.datedebut as datedebu2_3_0_, planificat0_.datefin as datefin3_3_0_, analyse1_.id as id1_0_1_, analyse1_.technicien as technici2_0_1_, analyse1_.date_debut_analyse as date_deb3_0_1_, analyse1_.date_fin_analyse as date_fin3_0_1_, analyse1_.echantillon_id as id1_0_2_, analyse1_.dateprelevement as dateprel2_1_0_, analyse1_.patient_id as patient5_1_0_, analyse1_.status as statu2_0_1_, analyse1_.typeanalyse as typeanaly1_0_1_ from planifications planificat0_
Hibernate: update analyses set technicien=?, date_debut_analyse=?, date_fin_analyse=?, echantillon_id=?, planification_id=?, reactive_id=?, resultats=? where id=?
Hibernate: insert into users (informationpersonnelle, motpasse, name, planification_id) values (?, ?, ?, ?)
Hibernate: insert into reactives (dateexpiration, description, fournisseur, name, quantitestock) values (?, ?, ?, ?, ?)
Hibernate: select analyse0_.id as id1_0_1_, analyse0_.technicien as technici2_0_1_, analyse0_.date_debut_analyse as date_deb3_0_1_, analyse0_.date_fin_analyse as date_fin3_0_1_, analyse0_.echantillon_id as id1_0_2_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select echantillo0_.id as id1_1_0_, echantillo0_.dateprelevement as dateprel2_1_0_, echantillo0_.patient_id as patient5_1_0_, echantillo0_.status as statu2_0_1_, echantillo0_.typeanalyse as typeanaly1_0_1_ from echantillons echantillo0_
Hibernate: select patient0_.id as id1_2_0_, patient0_.adresse as adresse2_2_0_, patient0_.datenaissnce as datenais3_2_0_, patient0_.name as name4_2_0_, patient0_.tel as tel5_2_0_ from patients patient0_
Hibernate: select analyse0_.id as id1_0_4_, analyse0_.technicien as technici2_0_4_, analyse0_.date_debut_analyse as date_deb3_0_4_, analyse0_.date_fin_analyse as date_fin3_0_4_, analyse0_.echantillon_id as id1_0_5_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select analyse0_.id as id1_0_4_, analyse0_.technicien as technici2_0_4_, analyse0_.date_debut_analyse as date_deb3_0_4_, analyse0_.date_fin_analyse as date_fin3_0_4_, analyse0_.echantillon_id as id1_0_5_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select reactive0_.id as id1_5_0_, reactive0_.dateexpiration as dateexpe2_5_0_, reactive0_.description as descript3_5_0_, reactive0_.fournisseur as fournisseur4_5_0_, reactive0_.name as nom6_5_0_, reactive0_.quantitestock as quantitestock7_5_0_ from reactives reactive0_
Hibernate: update analyses set technicien=?, date_debut_analyse=?, date_fin_analyse=?, echantillon_id=?, planification_id=?, reactive_id=?, resultats=? where id=?
Hibernate: insert into tapprtests (max, min, test_id) values (?, ?, ?)
Hibernate: select analyse0_.id as id1_0_1_, analyse0_.technicien as technici2_0_1_, analyse0_.date_debut_analyse as date_deb3_0_1_, analyse0_.date_fin_analyse as date_fin3_0_1_, analyse0_.echantillon_id as id1_0_2_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select echantillo0_.id as id1_1_0_, echantillo0_.dateprelevement as dateprel2_1_0_, echantillo0_.patient_id as patient5_1_0_, echantillo0_.status as statu2_0_1_, echantillo0_.typeanalyse as typeanaly1_0_1_ from echantillons echantillo0_
Hibernate: select patient0_.id as id1_2_0_, patient0_.adresse as adresse2_2_0_, patient0_.datenaissnce as datenais3_2_0_, patient0_.name as name4_2_0_, patient0_.tel as tel5_2_0_ from patients patient0_
Hibernate: select analyse0_.id as id1_0_4_, analyse0_.technicien as technici2_0_4_, analyse0_.date_debut_analyse as date_deb3_0_4_, analyse0_.date_fin_analyse as date_fin3_0_4_, analyse0_.echantillon_id as id1_0_5_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_
Hibernate: select analyse0_.id as id1_0_4_, analyse0_.technicien as technici2_0_4_, analyse0_.date_debut_analyse as date_deb3_0_4_, analyse0_.date_fin_analyse as date_fin3_0_4_, analyse0_.echantillon_id as id1_0_5_, analyse0_.dateprelevement as dateprel2_1_0_, analyse0_.patient_id as patient5_1_0_, analyse0_.status as statu2_0_1_, analyse0_.typeanalyse as typeanaly1_0_1_ from analyses analyse0_

```

Conclusion

La classe principale `LabAnalyseApplication` a été testée avec succès, et toutes les fonctionnalités essentielles ont été vérifiées. Le contexte de l'application est correctement chargé, les composants nécessaires sont initialisés, et les actions au démarrage sont exécutées conformément aux attentes. La configuration de la base de données est correcte, assurant ainsi le bon fonctionnement de l'application dans son ensemble.

Ce rapport peut être adapté en fonction des spécificités de votre application et des actions configurées dans la méthode `demo`. Vous pouvez également envisager l'ajout de tests supplémentaires pour les aspects spécifiques de la classe principale de votre application.