Rapport de Documentation :

Projet "LabXpert" - Système de Gestion du Laboratoire Médical TechLab

Réalisé par :

EL AMRI HALIMA

TKARKIB LHASSAN

BOUTAHIR RACHID

Encadré par :

Youssef IdBenSalh

Sommaire

Table De Figures

Introduction

Le laboratoire médical TechLab entreprend le déploiement d'un système de gestion complet, appelé "LabXpert", visant à optimiser ses opérations. L'objectif principal est d'améliorer l'efficacité et la précision dans le traitement des analyses médicales, avec un accent particulier sur la gestion des échantillons, le suivi des analyses, la gestion des résultats, la gestion des patients, l'inventaire des réactifs, la gestion des utilisateurs, la planification des analyses, et la génération de rapports statistiques.

Fonctionnalités du Système "LabXpert"

1. Enregistrement des Échantillons

- Les techniciens peuvent enregistrer de nouveaux échantillons en spécifiant des informations cruciales telles que le patient, le type d'analyse, et la date de prélèvement.

2. Suivi des Analyses en Cours

- Interface conviviale permettant aux techniciens et aux responsables de laboratoire de suivre en temps réel l'état d'avancement des analyses en cours, avec des détails spécifiques pour chaque échantillon.

3. Gestion des Résultats

- Les résultats des analyses sont consignés de manière systématique, permettant un accès rapide et la possibilité de partager les résultats avec les professionnels de la santé concernés.

4. Gestion des Patients

- Module dédié offrant la possibilité de gérer les informations relatives aux patients, assurant une centralisation des données et une navigation facilitée.

5. Inventaire des Réactifs

- Intégration d'un suivi des stocks pour garantir la disponibilité des réactifs nécessaires aux différentes analyses.

6. Gestion des Utilisateurs

- Interface d'administration permettant de gérer les droits d'accès et les informations des utilisateurs, assurant une sécurité accrue des données.

7. Planification des Analyses

- Possibilité de planifier les analyses en fonction de la charge de travail, optimisant ainsi l'utilisation des ressources du laboratoire.

8. Rapports Statistiques

- Génération de rapports statistiques pour évaluer les performances du laboratoire, identifier les tendances et prendre des décisions basées sur les données.

Gestion des taches par JIRA

Jira est une plateforme de gestion de projet développée par Atlassian. Elle est largement utilisée pour la gestion des projets, la planification, le suivi des tâches et la collaboration au sein des équipes de développement de logiciels.

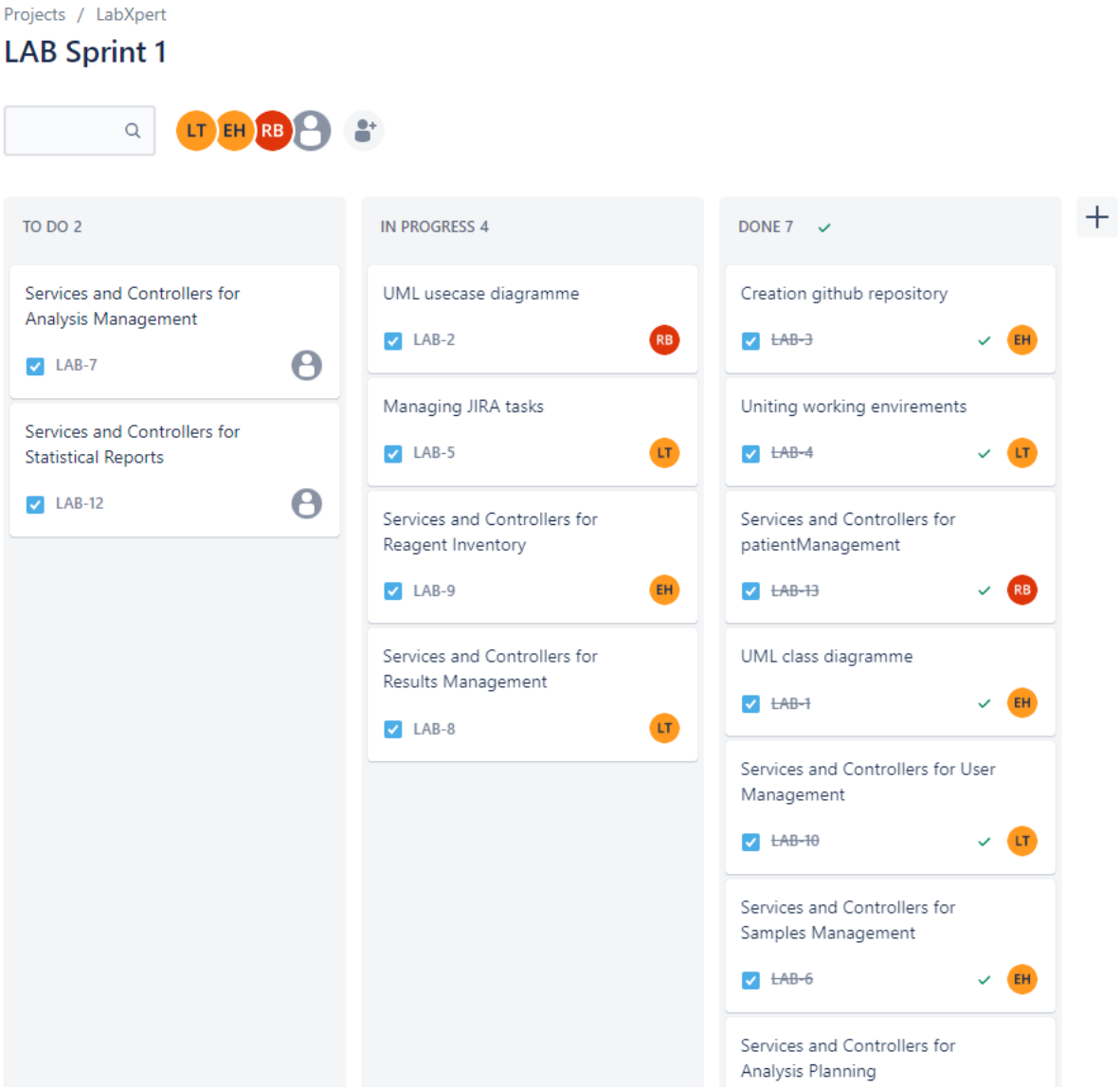

Figure 1:Jira example 1

Conception de Projet

1. Diagramme de Classe :

Le diagramme de classe représente la structure statique d'un système en identifiant les classes du système, leurs attributs, leurs relations et leurs méthodes. Il est utilisé pour visualiser la structure du système du point de vue des objets et des classes qui le composent.

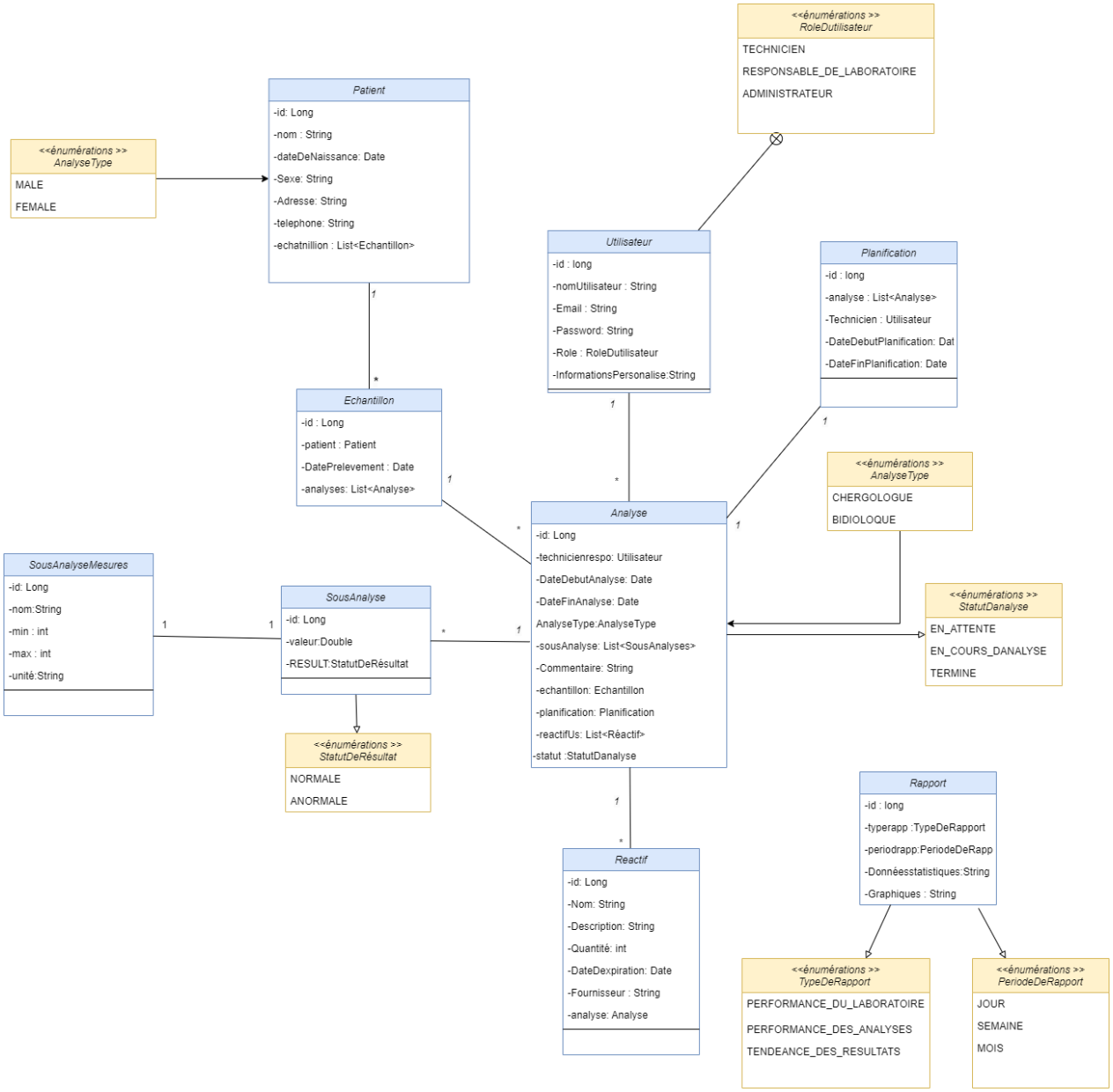

Figure 2:Diagramme de class

2. Diagramme de Cas d'Utilisation :

Le diagramme de cas d'utilisation modélise les interactions entre les acteurs externes et un système, décrivant les différents scénarios d'utilisation du système. Il met l'accent sur ce que fait le système sans entrer dans les détails internes de sa mise en œuvre.

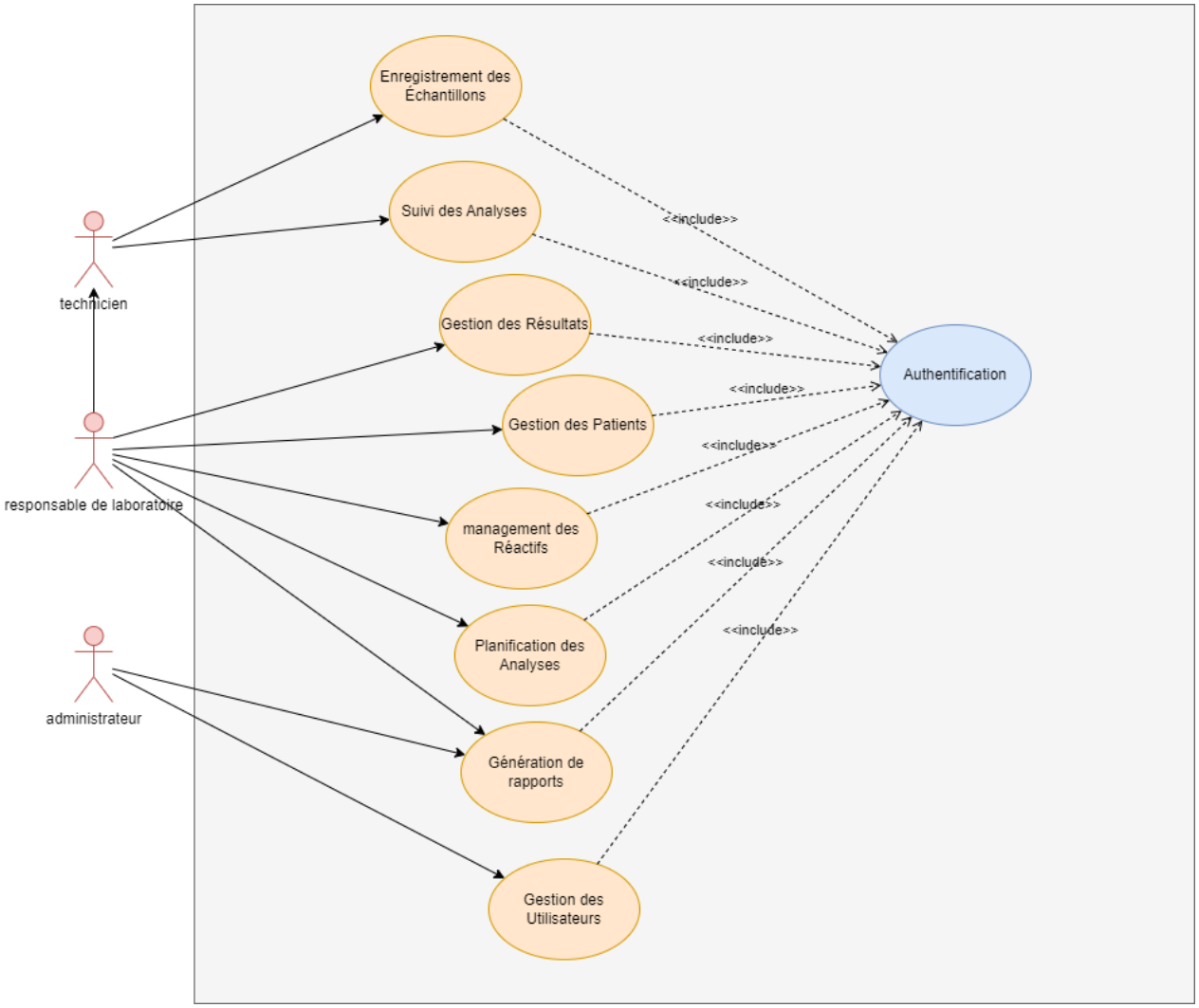

Figure 3:Diagramme de Cas des utilisations

Réalisation et outils techniques

1. Test Des contrôleurs en Postman

Postman est un outil de collaboration et de développement d'API (Interface de Programmation d'Application) qui permet aux développeurs de tester, de développer et de documenter les API plus efficacement. C'est une plateforme qui simplifie le processus de création, de test et de gestion des appels d'API. Postman offre une interface utilisateur conviviale et des fonctionnalités puissantes pour faciliter le travail avec les API.

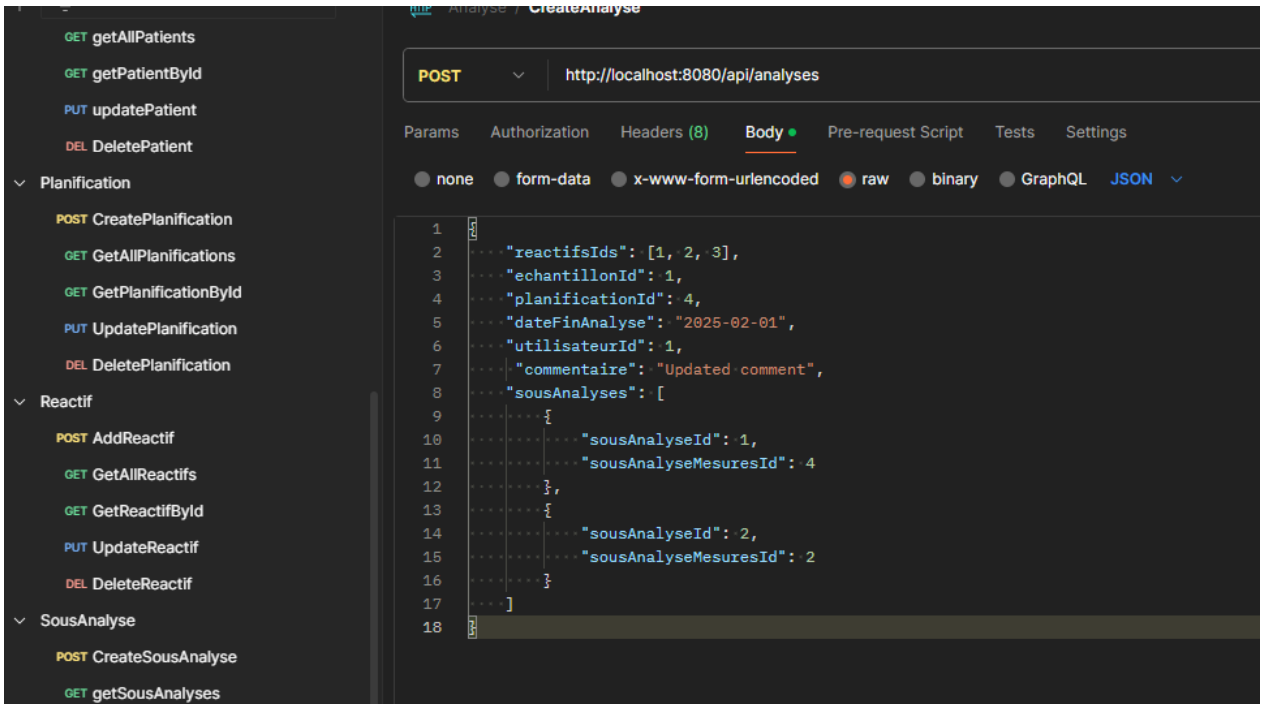

Figure 4: Analyses création api sur postman

Avant de créer une analyse, il est essentiel de d'abord établir des associations liées telles que le patient, l'utilisateur, la planification, les réactifs et les échantillons. En construisant ces relations, nous serons en mesure de réaliser une analyse de manière cohérente et complète, en assurant une intégration fluide de toutes les composantes du système. Cette approche garantit une base solide pour la gestion efficace des opérations du laboratoire médical.

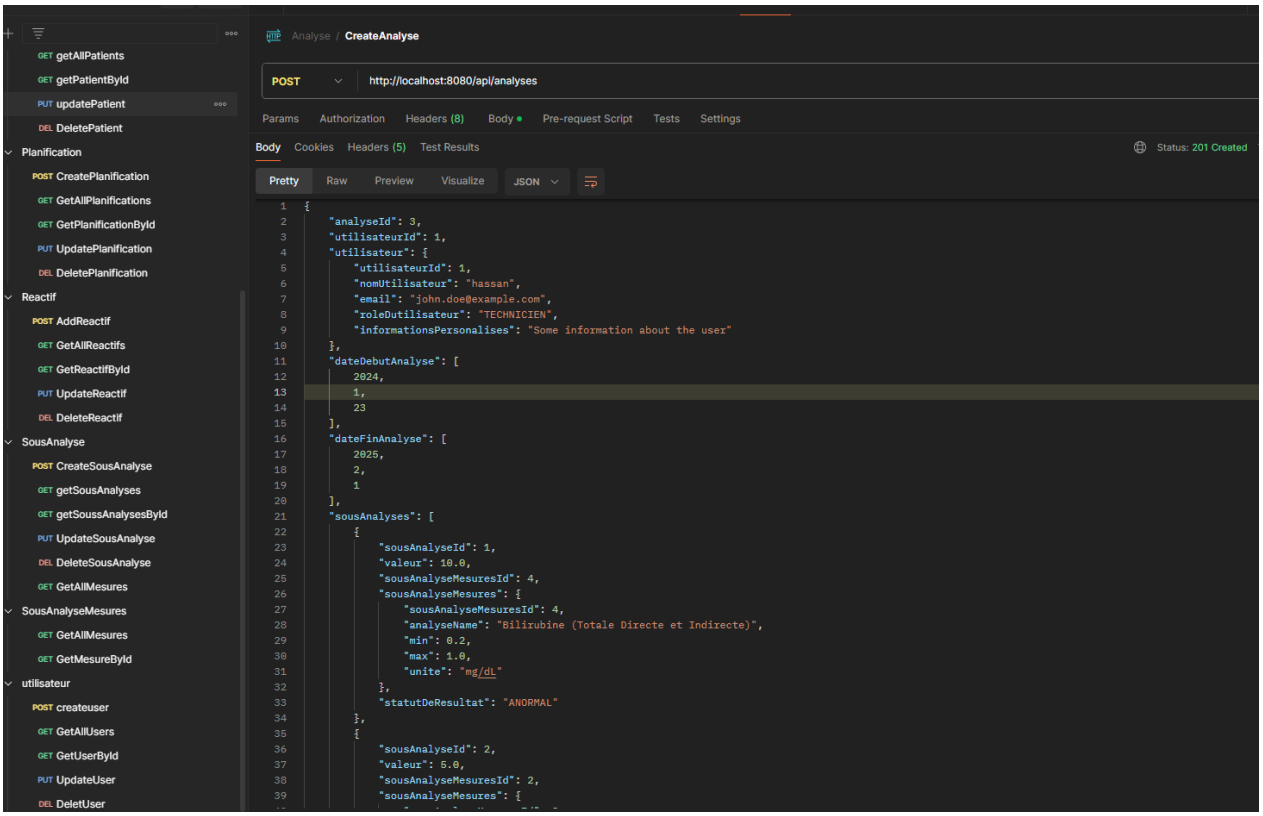

Figure 5:création api d'analyse Sur postman

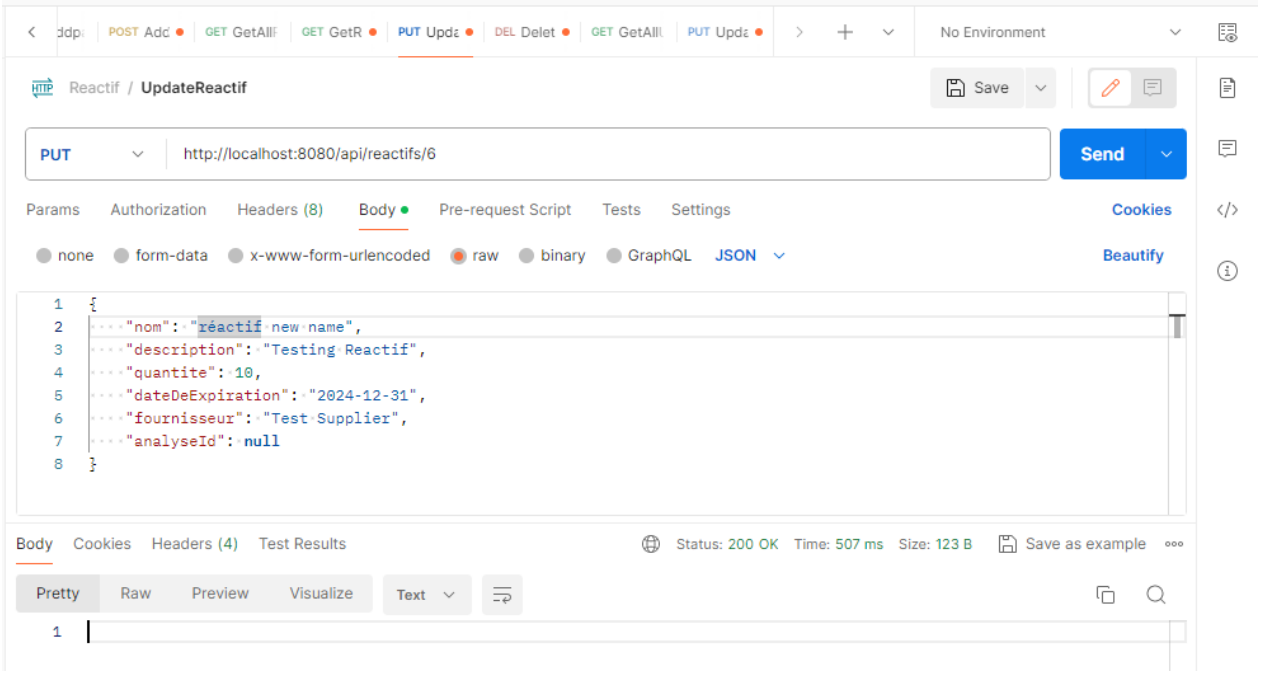

Figure 6:Modification de Réactif

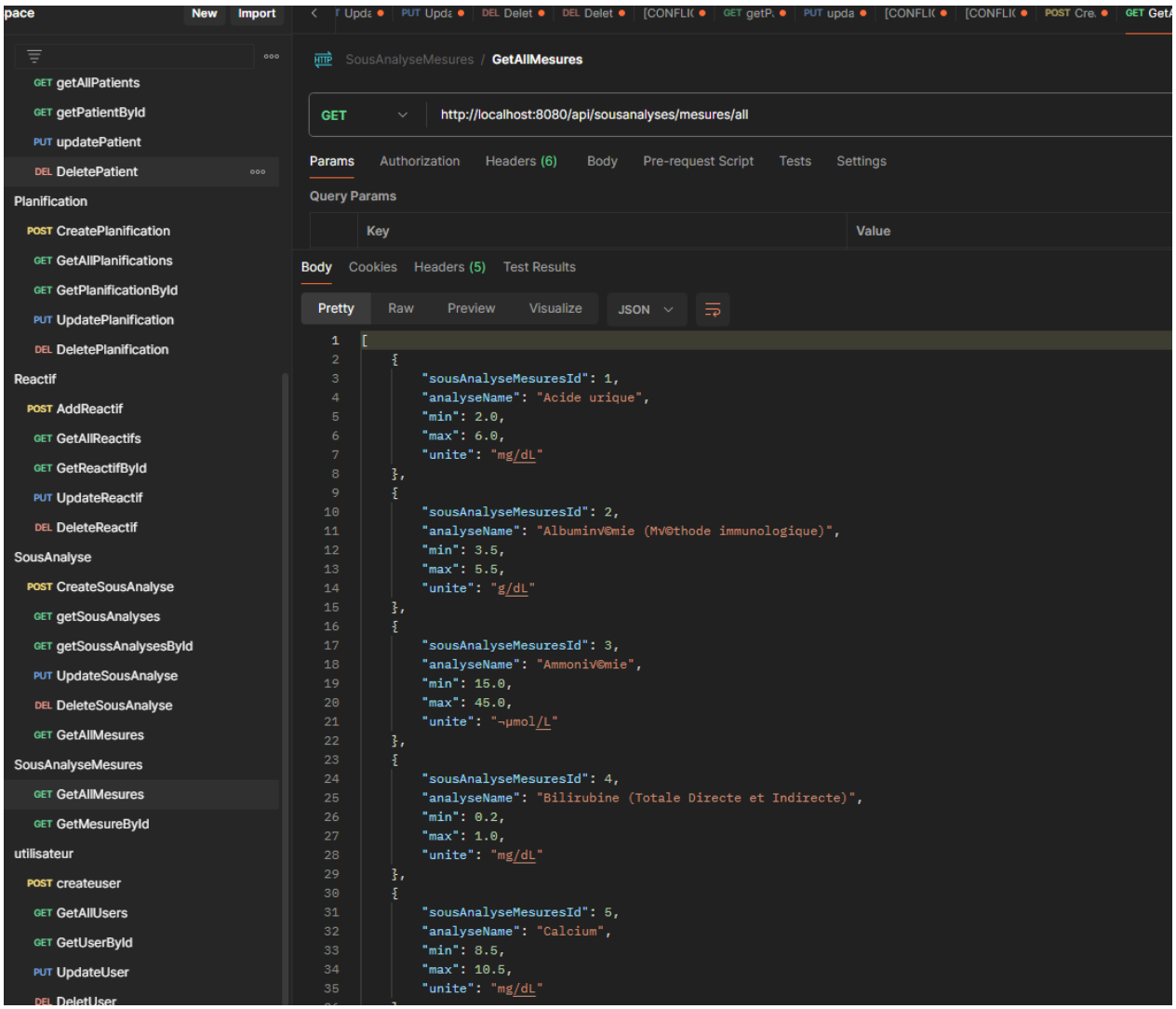

Figure 7:Récupuration de toutes sous analyses

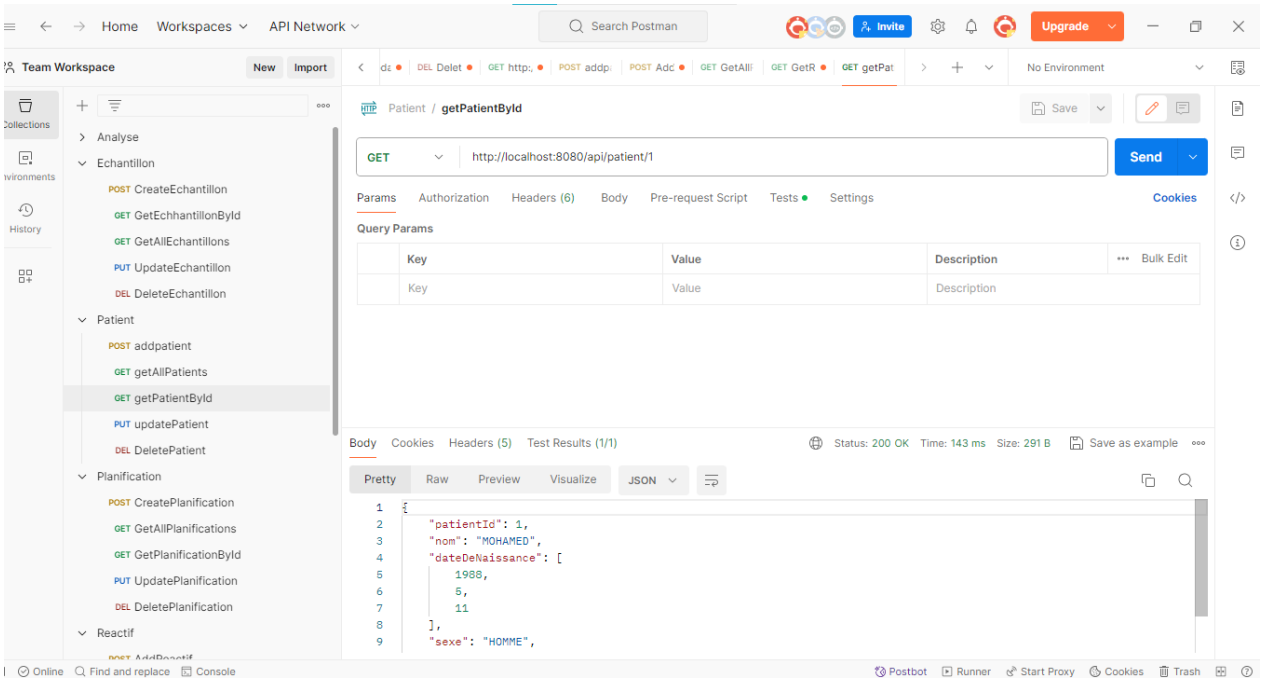

Figure 8:récupiration par ID de patient

2. Documentation par Swagger

Swagger est un ensemble d'outils open source qui permettent aux développeurs de concevoir, documenter et consommer des services web RESTful. L'objectif principal de Swagger est de simplifier le processus de développement d'API en fournissant une documentation claire et interactive, ainsi qu'en facilitant la communication entre les développeurs et les équipes travaillant sur différents services.

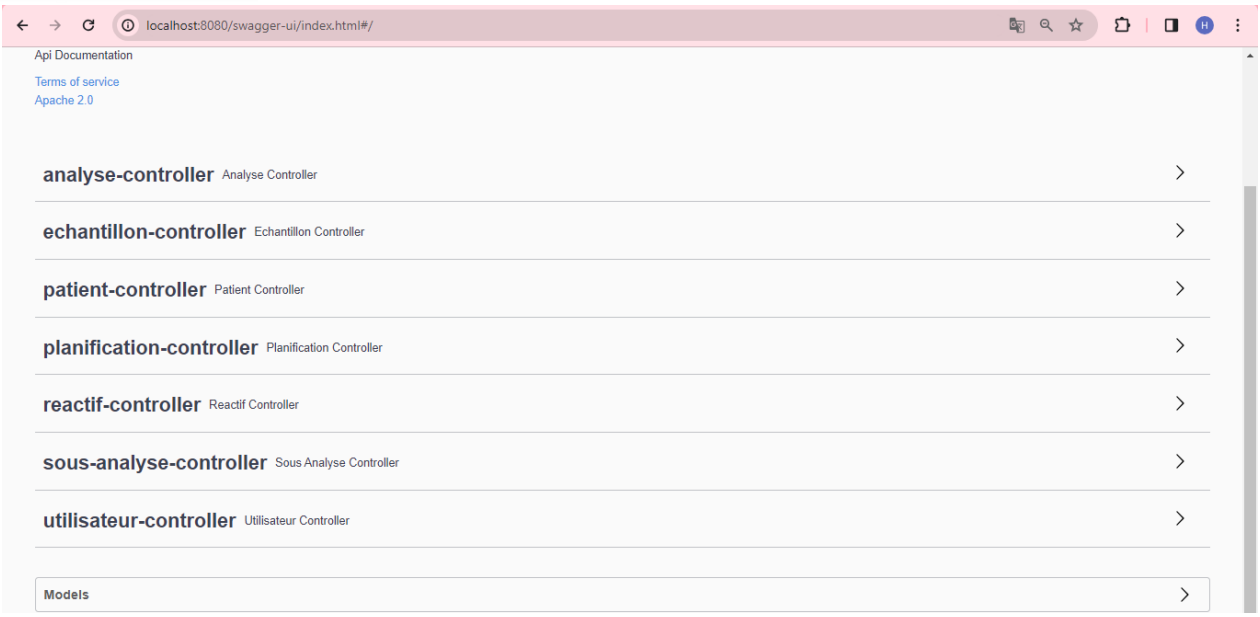

Figure 9:Documentation par Swagger

3. Validation DTO

La validation des objets de transfert de données (DTO) est une étape importante dans le processus de développement pour s'assurer que les données entrantes respectent les règles et contraintes attendues. En Java, la validation des DTO peut être effectuée à l'aide de plusieurs approches, et l'une des méthodes couramment utilisées est l'utilisation d'annotations de validation

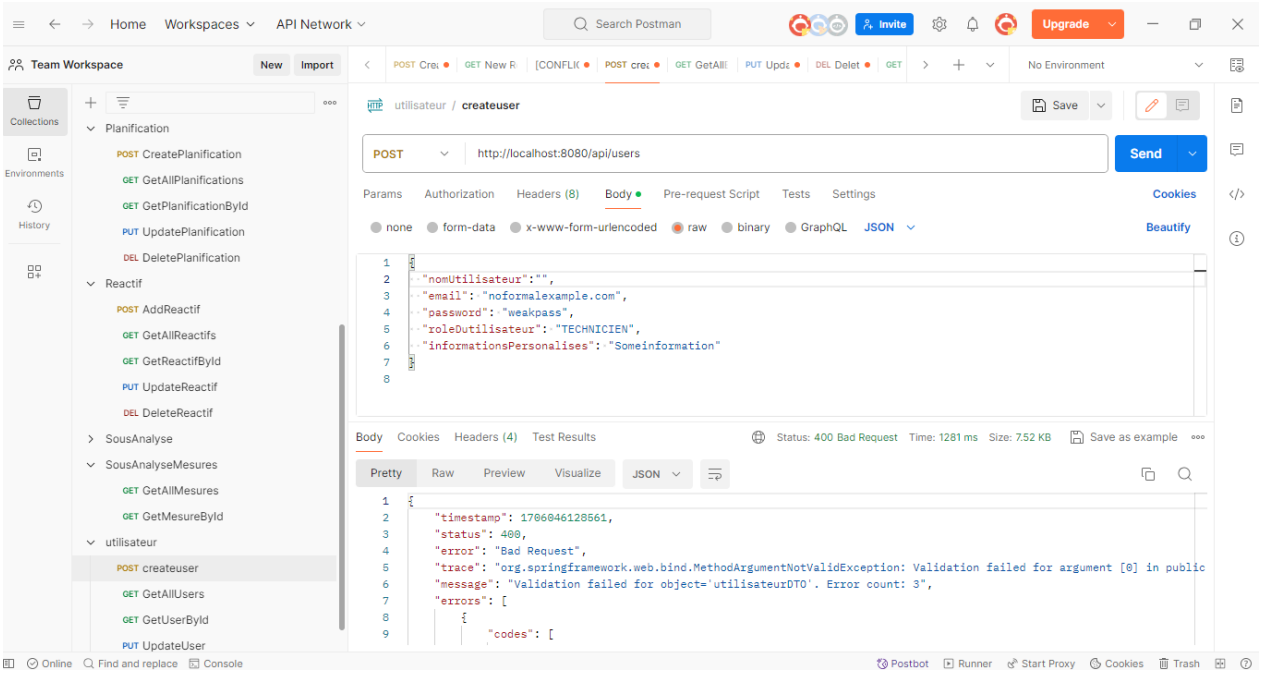

Figure 10:exemple de validation des DTO

4. Test des contrôleurs par Mockito

Tester les contrôleurs à l'aide de Mockito est une pratique courante dans le développement logiciel, en particulier dans le contexte des tests unitaires pour les applications basées sur Java. Mockito est un framework de test Java qui permet de créer des objets factices (mocks) pour simuler le comportement des dépendances

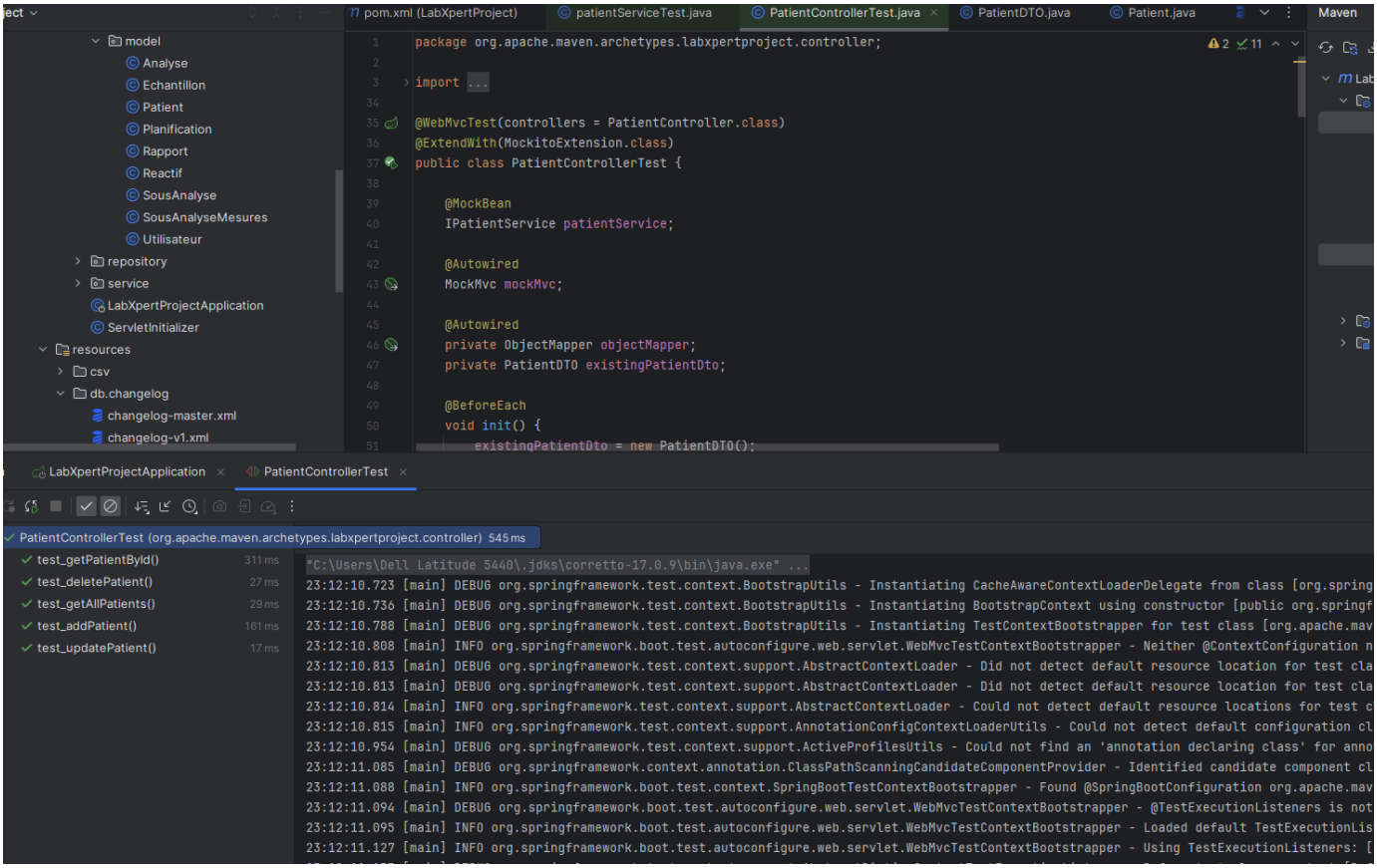

Figure 11:Test des contrôleurs par Mockito

5. Test des Services par H2 et Junit

Tester des services avec H2 et JUnit est une approche courante dans le développement Java, particulièrement lorsqu'on souhaite réaliser des tests unitaires ou d'intégration. H2 est une base de données en mémoire légère, tandis que JUnit est un framework de test unitaire pour Java. Voici un exemple simple de test de service avec H2 et JUnit :

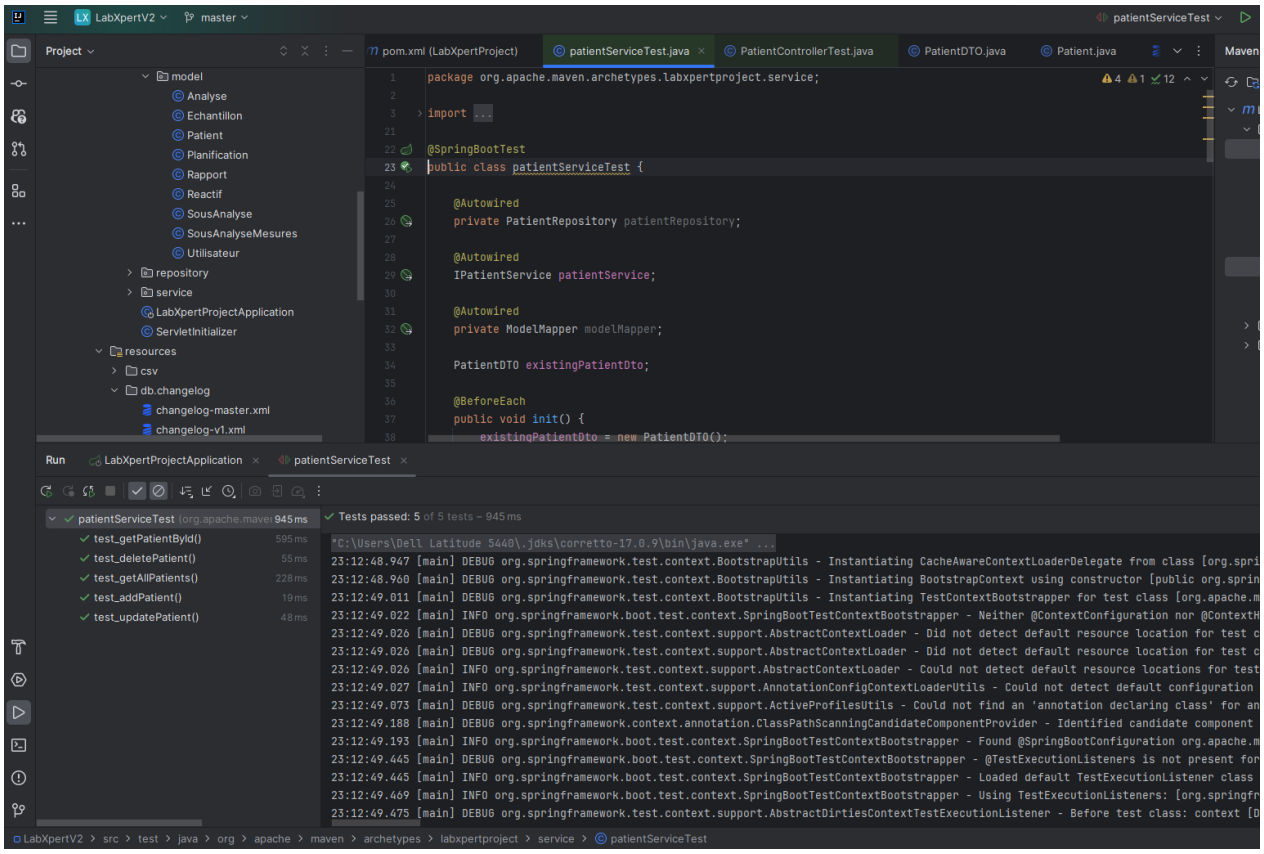

Figure 12:Test des Services par H2 et Junit

6. Concepts DevOps

Gestion des Versions

La gestion des versions sur GitHub avec Git implique l'utilisation de certaines fonctionnalités et bonnes pratiques pour suivre les modifications apportées à un projet, collaborer efficacement et garantir la stabilité du code.

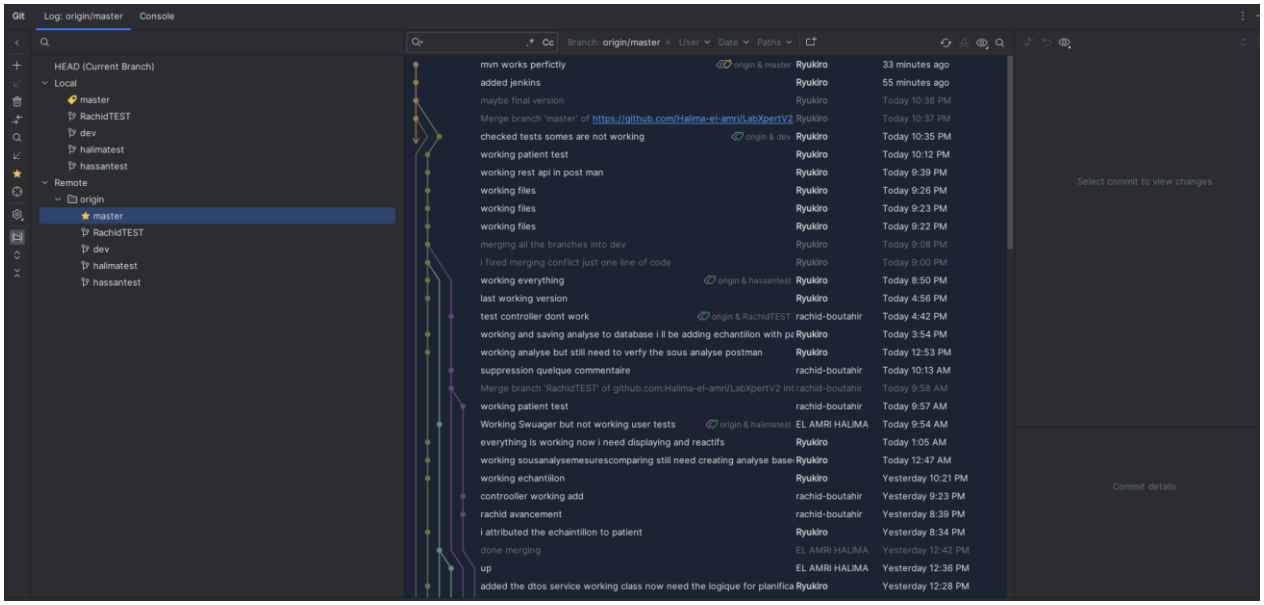

Figure 13:Gestion des versions sur GitHub

- **Intégration Continue (CI):**
  - L'intégration continue (CI) automatise l'incorporation des modifications du code provenant de plusieurs contributeurs dans un projet. Cette pratique DevOps encourage des fusions fréquentes dans un référentiel central, avec des builds et tests automatiques assurant la conformité du code avant son intégration. Un système de contrôle de version, tel que GitHub, est essentiel dans ce processus.

- **Jenkins:**
  - Jenkins, outil d'automatisation open-source écrit en Java, est dédié à la livraison continue. Il facilite la construction et le test en temps réel des projets, permettant une intégration aisée des modifications et une distribution rapide des nouvelles versions. Jenkins offre une flexibilité accrue avec ses plugins, adaptés aux besoins spécifiques de chaque projet.

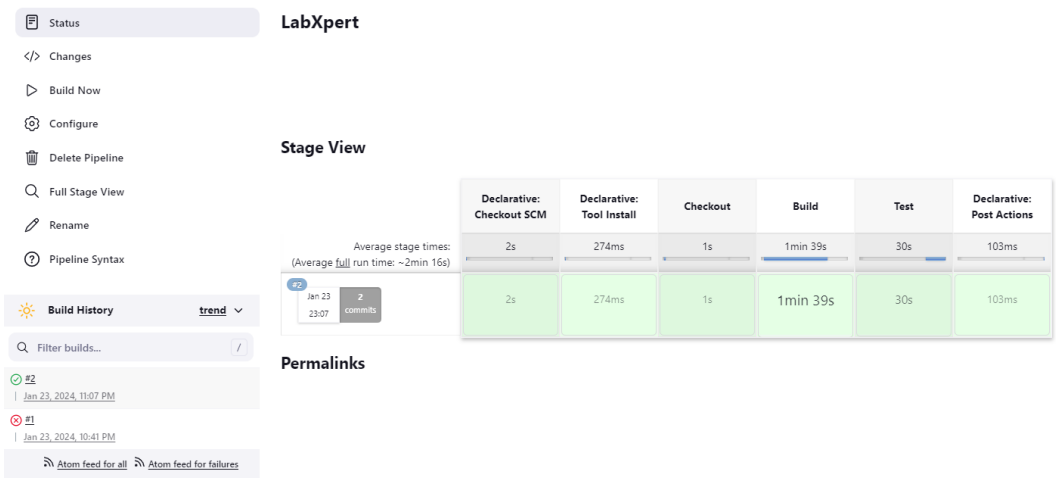

Figure 14:Jenkins

Conclusion

Notre backend s'appuie sur une architecture solide avec une API RESTful développée en utilisant Spring Boot. La gestion des dépendances est optimisée grâce à Apache Maven, tandis que PostgreSQL assure la persistance des données. Apache Tomcat est notre serveur d'application pour le déploiement. Les tests unitaires sont réalisés avec JUnit et Mockito, garantissant la qualité du code. Jenkins orchestre notre CI/CD, facilitant des déploiements continus. Git et Github sont utilisés pour la gestion de version, et Swagger assure une documentation claire de notre API. La gestion des journaux est assurée par Log4j, Liquibase et Jasper, complétant ainsi une stack complète et robuste.