

---

# Documentation technique du projet LabXpert - Angular (Part 3)

---

*Réalisé par*

**GUELSA Mouna**

**Tkarkib Lhassan**

**IdBensaleh Youssef**

2023-2024

## TABLE DES MATIÈRES

Documentation	3
1 Introduction	3
2 Conduite du projet	3
2.1 Méthodologie de travail	3
2.2 Outil de collaboration	4
3 Etude et analyse fonctionnelle	4
3.1 Besoins Fonctionnels	4
3.2 Modélisation	5
4 Etude technique et technologique	7
4.1 Technologies et outils techniques	7
5 Démonstration	10
5.1 Interface des réactifs	10
5.2 Interface des échantillons	11
5.3 Interface des mesures	11
5.4 Interface des détails d'analyses	12
5.5 Interface des échantillons	12
5.6 Interface des patients	13
6 Test	13
6.1 Test du service user	13
6.2 Test du service patient	13
7 Conclusion	14

## TABLE DES FIGURES

1	Méthode KANBAN . . . . .	3
2	Diagramme de cas d'utilisation. . . . .	5
3	Diagramme de classes. . . . .	7

## 1 Introduction

TechLab, laboratoire médical à la pointe de l'innovation, s'engage dans une transformation majeure avec le lancement du projet LabXpert. Conçu pour optimiser chaque aspect des opérations du laboratoire, LabXpert promet d'élever l'efficacité et la précision dans le traitement des analyses médicales. Cette solution complète couvre un large spectre, depuis l'enregistrement des échantillons jusqu'à la gestion des résultats, avec des fonctionnalités telles que le suivi en temps réel des analyses, la gestion des patients, le contrôle des stocks de réactifs, et bien plus encore. LabXpert, c'est l'avenir de TechLab, offrant une gestion intégrée qui promet d'optimiser les services pour un diagnostic médical plus rapide et plus précis.

## 2 Conduite du projet

Le choix de la conduite du projet est une phase d'exterminante pour accomplir le projet dans les bonnes conditions. Il faut donc bien définir le processus de développement et en déduire le planning du projet à suivre.

### 2.1 Méthodologie de travail

Afin de structurer les différentes phases de notre projet, et qui vont être explicitées par la suite, et afin de garantir une organisation optimale, il est nécessaire de fixer un cadre qui simplifie indéniablement le lancement du projet, sa progression et sa réussite par la suite. Pour cela Afin d'assurer le respect des délais et le bon déroulement du projet, nous avons opté pour la méthode agile KANBAN qui nous fournit un cadre d'agilité simple à comprendre. Cette méthode repose sur un système de visualisation de cartes dans un tableau. Les cartes ou étiquettes représentent les tâches à traiter qui correspondent aux demandes entrantes. Le tableau a l'avantage d'être facile à déchiffrer . Il indique également l'état d'avancement des tâches à traiter en affichant les cartes dans différentes colonnes : à faire ,en cours,à tester,terminé

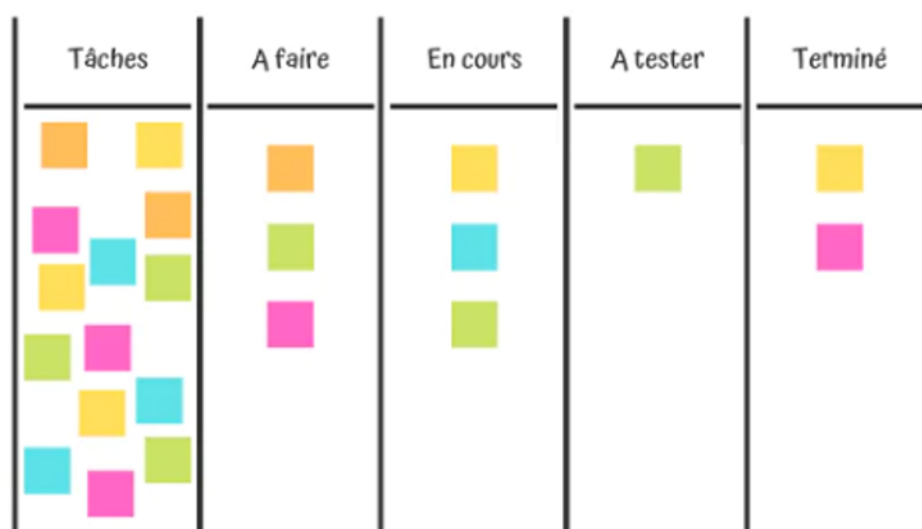
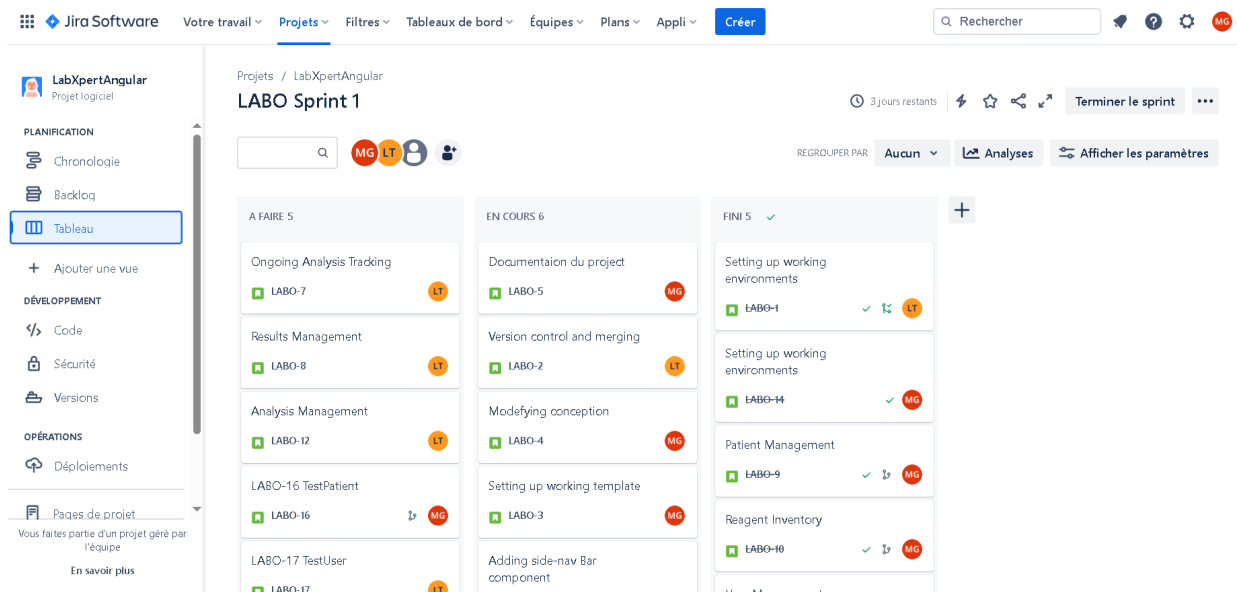


FIGURE 1 – Méthode KANBAN

## 2.2 Outil de collaboration

### 2.2.1 Jira

Pour réussir KANBAN, nous nous servons de ‘JIRA’ qui est un outil de gestion de projet en ligne. JIRA est une plateforme multifonction qui vise à faciliter la gestion de projet en aidant à suivre les tâches, identifier les blocages et partager des informations entre les membres d’une équipe. Il est basé sur une organisation des projets en tickets, chacune représentant des tâches. De plus, JIRA permet d’assurer le suivi des tickets et la définition d’un workflow adapté aux méthodes de travail. JIRA génère aussi des graphiques et visuels qui permettent de repérer en un coup d’œil l’état des différentes missions et d’identifier les problèmes à régler en priorité pour pouvoir avancer.



## 3 Etude et analyse fonctionnelle

### 3.1 Besoins Fonctionnels

Le système de gestion du laboratoire "LabXpert" offre plusieurs fonctionnalités qui répondent aux besoins fonctionnels du laboratoire TechLab. Voici une liste détaillée de ces fonctionnalités :

- **Enregistrement des échantillons** : Le système permet aux techniciens d’enregistrer de nouveaux échantillons en spécifiant les informations pertinentes telles que le patient, le type d’analyse et la date de prélèvement.
- **Suivi des analyses en cours** : Une interface conviviale permet 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.
- **Gestion des résultats** : Les résultats des analyses sont consignés de manière systématique, permettant un accès rapide aux informations et la possibilité de partager les résultats avec les professionnels de la santé concernés.
- **Gestion des patients** : Un module dédié offre la possibilité de gérer les informations relatives aux patients, assurant une centralisation des données et une navigation facilitée.
- **Inventaire des réactifs** : Intégration d’un suivi des stocks pour garantir la disponibilité des réactifs nécessaires aux différentes analyses.

- **Gestion des utilisateurs** : Une interface d'administration permet de gérer les droits d'accès et les informations des utilisateurs, assurant une sécurité accrue des données.
- **Planification des analyses** : Possibilité de planifier les analyses en fonction de la charge de travail, optimisant ainsi l'utilisation des ressources du laboratoire.
- **Rapports statistiques** : Génération de rapports statistiques pour évaluer les performances du laboratoire, identifier les tendances et prendre des décisions basées sur les données.

Ces fonctionnalités sont conçues pour améliorer l'efficacité et la précision dans le traitement des analyses médicales, optimiser les opérations du laboratoire et fournir un service plus rapide et plus précis aux patients.

## 3.2 Modélisation

### 3.2.1 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.

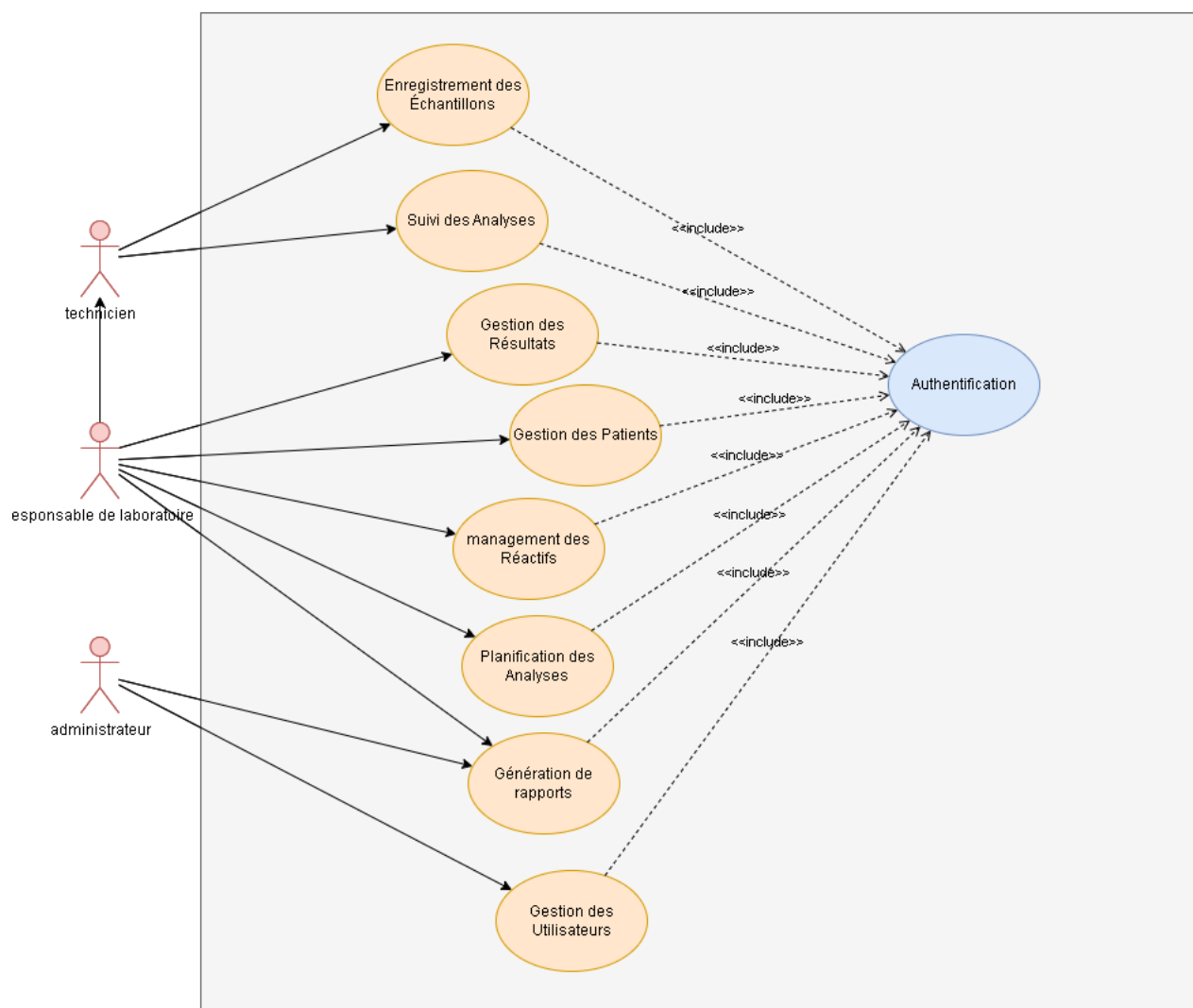


FIGURE 2 – Diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation décrit les interactions entre les utilisateurs (acteurs) et le système. Voici une description textuelle des cas d'utilisation pour le système de gestion du laboratoire "LabXpert" :

- **Enregistrer Échantillon :**
  - Acteur : Technicien
  - Description : Le technicien enregistre un nouvel échantillon en fournissant les informations nécessaires telles que le patient, le type d'analyse, et la date de prélèvement.
- **Suivre Analyses en Cours :**
  - Acteur : Technicien, Responsable de laboratoire
  - Description : Les utilisateurs peuvent suivre en temps réel l'état d'avancement des analyses en cours. Ils peuvent obtenir des détails spécifiques pour chaque échantillon.
- **Consulter Résultats :**
  - Acteur : Technicien, Professionnels de la santé
  - Description : Les utilisateurs peuvent consulter les résultats des analyses enregistrés de manière systématique. Ils peuvent également partager ces résultats avec les professionnels de la santé concernés.
- **Gérer Patients :**
  - Acteur : Personnel administratif, Technicien
  - Description : Les utilisateurs peuvent gérer les informations relatives aux patients, assurant une centralisation des données et une navigation facilitée.
- **Gérer Inventaire Réactifs :**
  - Acteur : Technicien
  - Description : Le technicien peut suivre les stocks de réactifs, assurant la disponibilité des réactifs nécessaires aux différentes analyses.
- **Gérer Utilisateurs :**
  - Acteur : Administrateur
  - Description : L'administrateur peut gérer les droits d'accès et les informations des utilisateurs, assurant une sécurité accrue des données.
- **Planifier Analyses :**
  - Acteur : Responsable de laboratoire
  - Description : Le responsable de laboratoire peut planifier les analyses en fonction de la charge de travail, optimisant ainsi l'utilisation des ressources du laboratoire.
- **Générer Rapports Statistiques :**
  - Acteur : Responsable de laboratoire
  - Description : Le responsable de laboratoire peut générer des rapports statistiques pour évaluer les performances du laboratoire, identifier les tendances et prendre des décisions basées sur les données.

### 3.2.2 Diagramme de classes

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

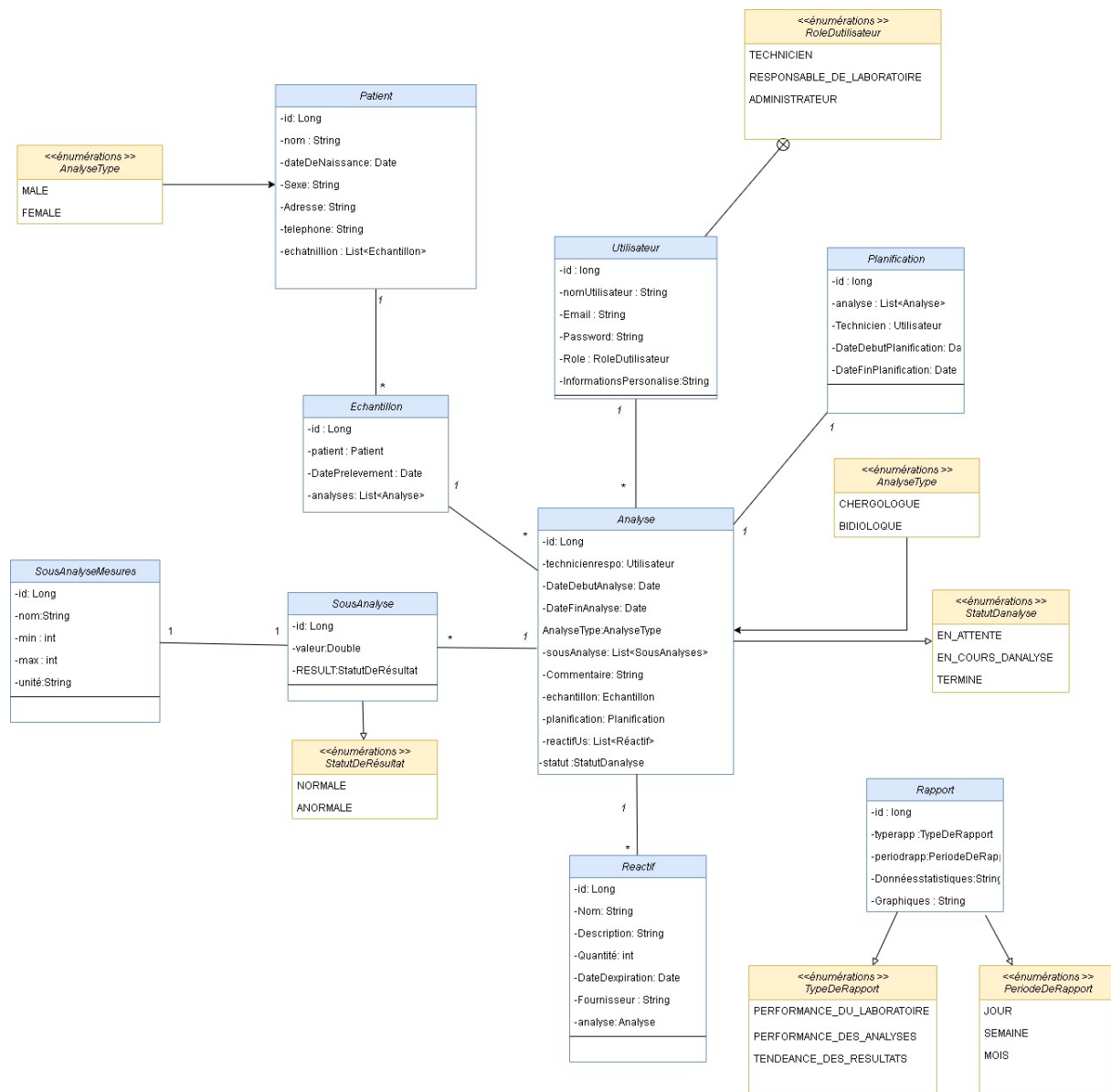


FIGURE 3 – Diagramme de classes.

## 4 Etude technique et technologique

### 4.1 Technologies et outils techniques

#### 4.1.1 Concepts DevOps

##### Intégration continue

L'intégration continue (CI) est la pratique qui permet d'automatiser l'intégration des modifications du code provenant de plusieurs contributeurs dans un même projet. Il s'agit d'une des principales bonnes pratiques DevOps, qui permet aux développeurs de fusionner fréquemment les modifications de code dans un répertoire central où les builds et les tests sont ensuite exécutés. Des outils automatisés sont utilisés pour garantir la conformité du nouveau code avant son intégration. Un système de contrôle de la version du code source est l'élément central du processus de CI. Le système de contrôle de version est également complété par d'autres contrôles tels que les tests de qualité du code, les outils de révision du style syntaxique, etc. On a utilisé



GITHUB .

#### 4.1.2 Outils

\* **Angular :**



Angular est un framework open-source développé par Google pour la création d'applications web côté client. Il est écrit en TypeScript et offre une structure robuste pour le développement d'applications web dynamiques et interactives. Voici quelques points clés concernant Angular :

1. **Architecture MVVM (Modèle-Vue-VueModèle) :** Angular utilise une architecture basée sur le modèle MVVM qui permet une séparation claire entre la logique métier, la présentation et la gestion de l'état.
2. **Langage TypeScript :** Angular est principalement écrit en TypeScript, un sur-ensemble de JavaScript qui ajoute des fonctionnalités de typage statique à JavaScript. Cela contribue à une meilleure maintenance du code, une détection d'erreurs précoce et une meilleure intégration des outils de développement.
3. **Binding de données bidirectionnel :** Angular propose une liaison de données bidirectionnelle, ce qui signifie que les changements dans le modèle (côté logique) sont automatiquement reflétés dans la vue (côté interface utilisateur) et vice versa.
4. **Composants :** Les applications Angular sont basées sur des composants. Chaque composant encapsule une partie spécifique de la logique de l'application et de l'interface utilisateur. Ces composants peuvent être imbriqués pour construire des applications complexes.
5. **Directives :** Les directives Angular permettent d'étendre la logique HTML en ajoutant des fonctionnalités personnalisées aux éléments de la page.
6. **Services :** Les services Angular sont des objets réutilisables qui peuvent être injectés dans des composants pour fournir des fonctionnalités partagées, telles que l'accès aux données, la gestion de l'état, etc.
7. **Routing :** Angular propose un système de routage qui permet de gérer la navigation entre les différentes vues de l'application de manière fluide.
8. **Gestion des formulaires :** Angular offre des fonctionnalités avancées pour la gestion des formulaires, y compris la validation côté client.

Angular a évolué au fil du temps, et différentes versions ont été publiées. Il est souvent utilisé pour le développement d'applications d'entreprise et d'applications web complexes en raison de sa structure modulaire, de sa facilité de test et de sa prise en charge active par la communauté de développeurs.

**\* Node js :**

Il s'agit d'un environnement d'exécution JavaScript, basé sur le moteur JavaScript V8 de Chrome. Il est développé par Ryan Dahl qui est un ingénieur logiciel travaillant chez Google Brain, il a également développé le runtime Deno JavaScript et TypeScript. Node.js est multi-plateforme et open-source qui exécute du code JavaScript côté serveur, c'est-à-dire en dehors du navigateur Web. En raison de sa nature monothread, il est principalement utilisé pour les serveurs non bloquants et pilotés par les événements, un modèle d'E/S non bloquant le rend léger et efficace, il est donc préférable pour les applications en temps réel à forte intensité de données. Il est utilisé par les sites Web traditionnels et les services d'API principaux. Il est conçu avec une architecture push en temps réel qui s'exécute sur des appareils distribués. Le module HTTP (Hypertext Transfer Protocol) fournit un ensemble de classes et de fonctions pour construire un serveur HTTP.

**\* Type Script :**

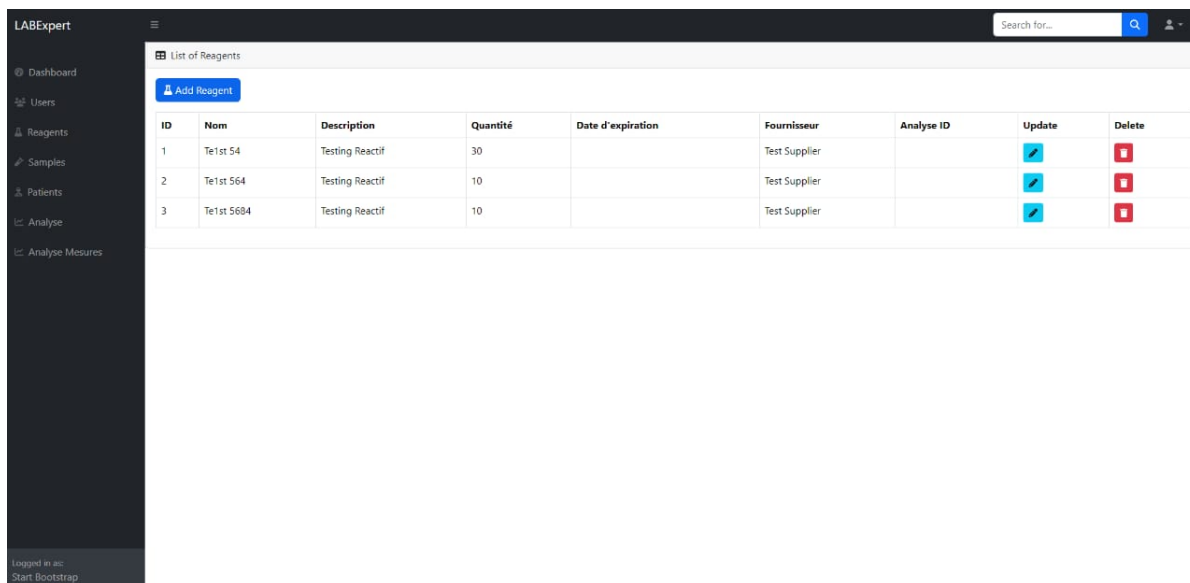
TypeScript est un langage de programmation développé par Microsoft qui s'appuie sur JavaScript en ajoutant des fonctionnalités de typage statique. L'une de ses caractéristiques distinctives est la possibilité de déclarer le type des variables, facilitant ainsi la détection précoce des erreurs lors du développement. TypeScript prend en charge les concepts de programmation orientée objet tels que les classes, l'héritage, et les interfaces, offrant ainsi une structure plus organisée pour le code JavaScript. Grâce à son typage statique, il améliore la lisibilité du code et permet une meilleure maintenance, particulièrement dans des projets complexes. TypeScript est également compatible avec JavaScript, ce qui facilite son adoption progressive dans les projets existants. Son utilisation est courante dans le développement d'applications web, en particulier avec des frameworks comme Angular.

**\* Jasmine :**

Jasmine est un framework de test unitaire populaire pour JavaScript, souvent utilisé dans le développement d'applications web, en particulier avec des frameworks comme Angular. Sa syntaxe claire et expressive facilite la rédaction de tests, organisés en suites et spécifications. Avec des fonctions d'assertion robustes, des hooks de cycle de vie pour la configuration de l'environnement de test, et des fonctionnalités intégrées de mocking et d'espionnage, Jasmine offre une solution complète pour les tests unitaires. Son intégration avec des runners de tests comme Karma simplifie l'exécution des tests sur différents navigateurs. Jasmine encourage une approche centrée sur le comportement (BDD), incitant les développeurs à décrire le comportement attendu du code. En fournissant une documentation générée automatiquement, Jasmine facilite également la compréhension des tests et contribue à assurer la qualité du code en identifiant rapidement les erreurs.

## 5 Démonstration

### 5.1 Interface des réactifs

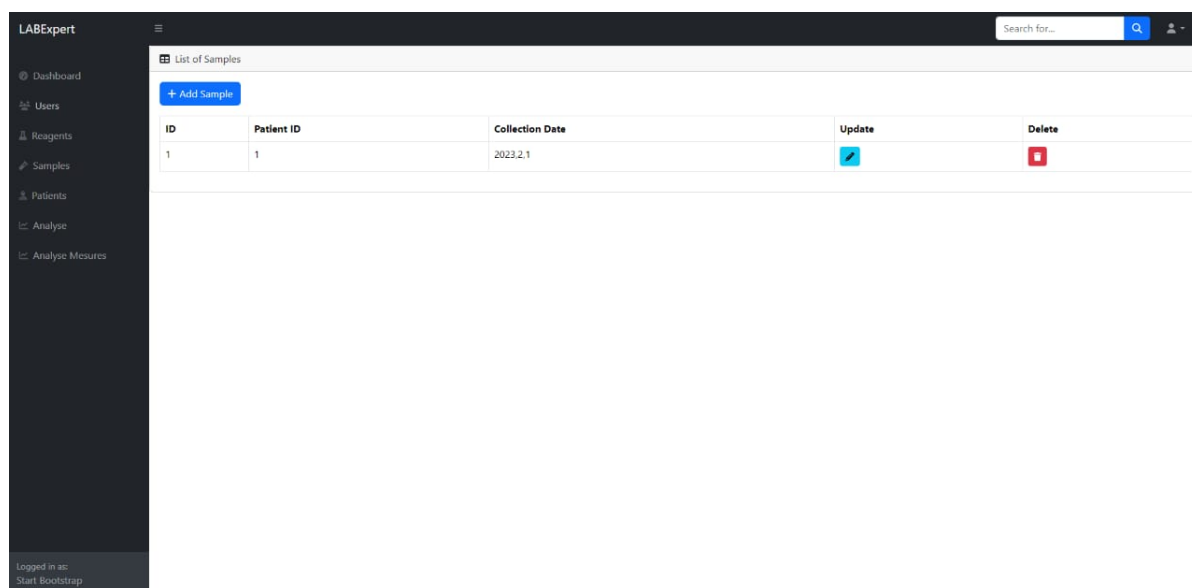


The screenshot displays the LABExpert application interface. On the left is a dark sidebar with a menu containing: Dashboard, Users, Reagents, Samples, Patients, Analyse, and Analyse Mesures. The main content area is titled "List of Reagents" and includes a blue "Add Reagent" button. Below the button is a table with the following data:

ID	Nom	Description	Quantité	Date d'expiration	Fournisseur	Analyse ID	Update	Delete
1	Te1st 54	Testing Reactif	30		Test Supplier			
2	Te1st 564	Testing Reactif	10		Test Supplier			
3	Te1st 5694	Testing Reactif	10		Test Supplier			

At the bottom left of the sidebar, it says "logged in as: Start Bootstrap". At the top right of the main area, there is a search bar with the placeholder text "Search for..." and a magnifying glass icon.

## 5.2 Interface des échantillons



LABExpert

Search for...

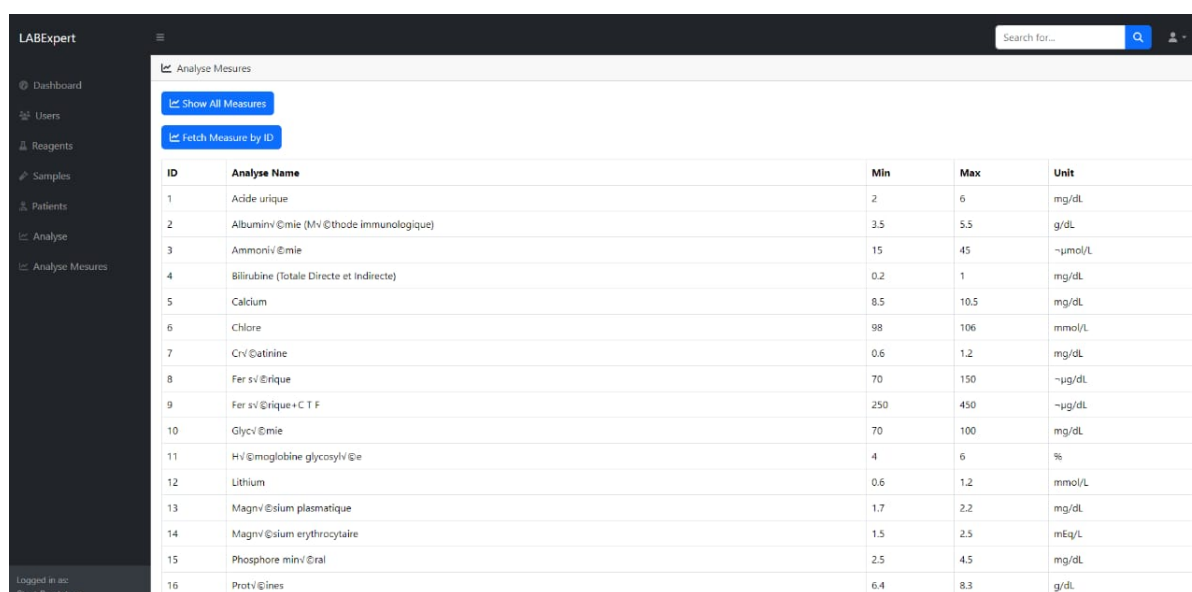
List of Samples

+ Add Sample

ID	Patient ID	Collection Date	Update	Delete
1	1	2023.2.1		

Logged in as: Start Bootstrap

## 5.3 Interface des mesures



LABExpert

Search for...

Analyse Mesures

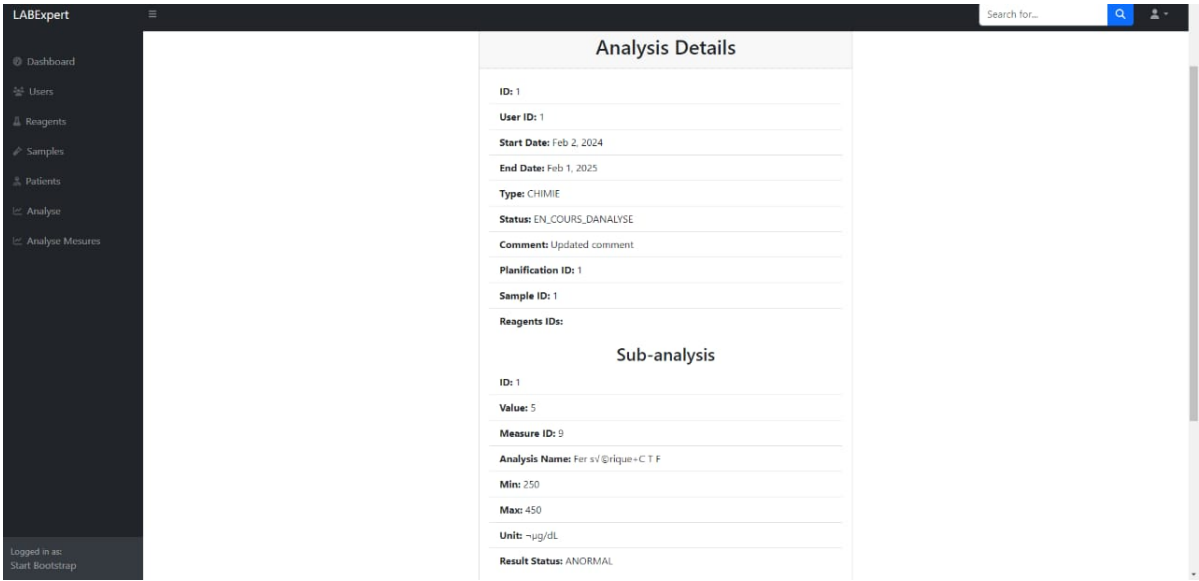
Show All Measures

Fetch Measures by ID

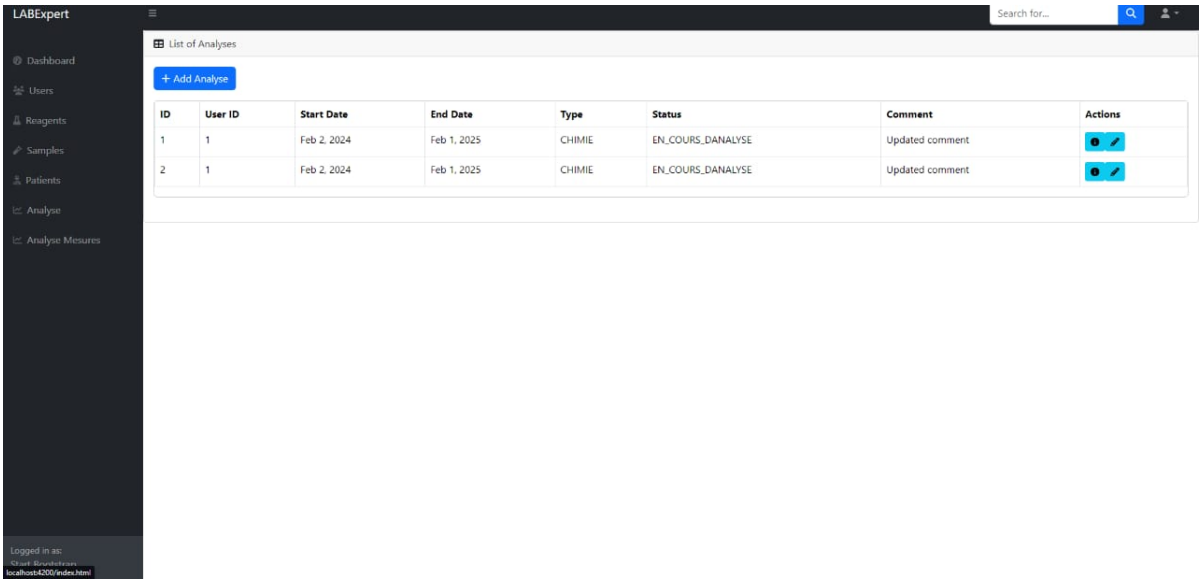
ID	Analyse Name	Min	Max	Unit
1	Acide urique	2	6	mg/dL
2	Albuminiv @mie (Mv @thode immunologique)	3.5	5.5	g/dL
3	Ammoniv @mie	15	45	~µmol/L
4	Bilirubine (Totale Directe et Indirecte)	0.2	1	mg/dL
5	Calcium	8.5	10.5	mg/dL
6	Chlore	98	106	mmol/L
7	Criv @atrine	0.6	1.2	mg/dL
8	Fer sv @rique	70	150	~µg/dL
9	Fer sv @rique + C T F	250	450	~µg/dL
10	Glycv @mie	70	100	mg/dL
11	Hv @moglobine glycosylv @e	4	6	%
12	Lithium	0.6	1.2	mmol/L
13	Magnv @sium plasmatique	1.7	2.2	mg/dL
14	Magnv @sium erythrocytaire	1.5	2.5	mEq/L
15	Phosphore minv @ral	2.5	4.5	mg/dL
16	Protv @lines	6.4	8.3	g/dL

Logged in as: Start Bootstrap

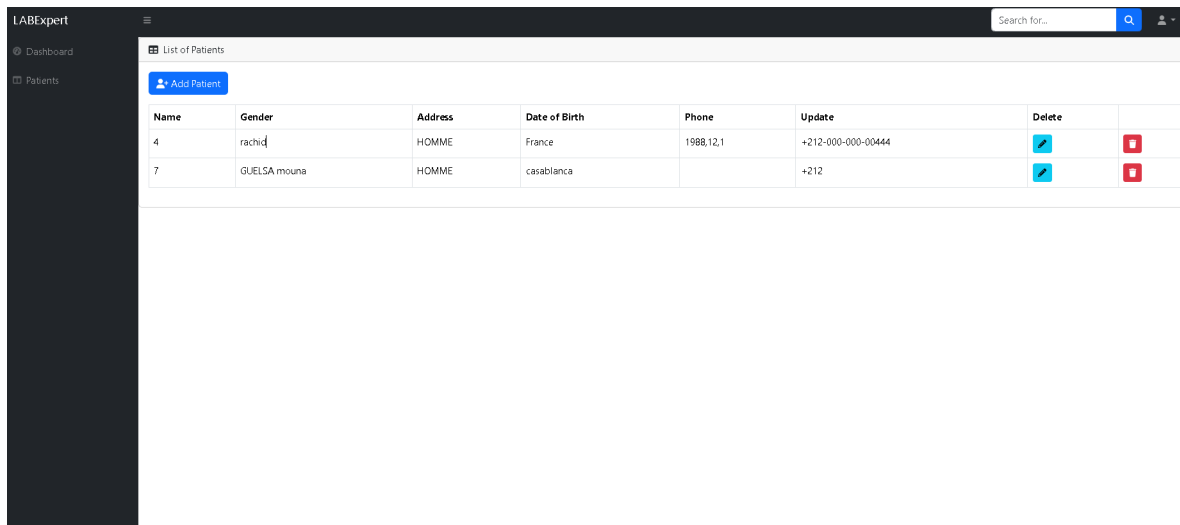
5.4 Interface des détails d’analyses





5.5 Interface des échantillons



## 5.6 Interface des patients



Name	Gender	Address	Date of Birth	Phone	Update	Delete
4	rachid	HOMME	France	1988,12,1	+212-000-000-00444	
7	GUELSA mouna	HOMME	casablanca		+212	


## 6 Test

### 6.1 Test du service user

Chrome est contrôlé par un logiciel de test automatisé.

**Karma v 6.4.2 - connected; test: complete;** DEBUG

Chrome 121.0.0.0 (Windows 10) is idle

 4.3.0 Options

5 specs, 0 failures, randomized with seed 58427 finished in 0.161s

PatientService


- should retrieve a patient by ID
- should update a patient
- should be created
- should retrieve the list of patients
- should add a new patient

### 6.2 Test du service patient

Chrome est contrôlé par un logiciel de test automatisé.

**Karma v 6.4.2 - connected; test: complete;** DEBUG

Chrome 121.0.0.0 (Windows 10) is idle

 4.3.0 Options

5 specs, 0 failures, randomized with seed 53776 finished in 0.041s

PatientService

- should retrieve the list of patients
- should be created
- should update a patient
- should retrieve a patient by ID
- should add a new patient

## 7 Conclusion

En résumé, opter pour Angular dans la mise en œuvre de LabExpert offre une base solide pour le développement d’une application web robuste et interactive. Les fonctionnalités avancées d’Angular, combinées à l’utilisation de Jasmine pour les tests unitaires, permettent de créer une solution fiable et de haute qualité. Avec son écosystème dynamique, Angular simplifie le processus de développement, offrant ainsi une expérience fluide pour la création réussie de LabExpert.