



Plan d'Assurance Qualité Logiciel (PAQ)



TRAVAIL PRÉSENTÉ À
Ms Driss Sebbar

DANS LE CADRE DU PROJET
Catalogue de Plantes Médicinales

PAR L'ÉQUIPE :

Latifa DIDI ALAOUI

Chaima ROUITA

Nouhayla ELASSOULI

Nihal TIZGHA

Année universitaire 2024/2025

Sommaire

I.	Introduction	4
1.	Contexte du projet	4
2.	Objectif du PAQ	4
II.	Références	4
	Méthodologies utilisées	4
	Documents de référence :	4
III.	Organisation de la Qualité	5
1.	Responsabilités des membres	5
2.	Outils collaboratifs.....	5
IV.	Critères de Qualité Logicielle	5
	Fonctionnalité.....	5
	Fiabilité :	5
	Performance :	5
	Utilisabilité :	6
	Sécurité :	6
V.	Gestion des Exigences	6
	Traçabilité :	6
	Gestion des modifications :	6
VI.	Processus d'Assurance Qualité	6
1.	Revue des spécifications :	6
2.	Tests :	6
3.	Revue de code	6
4.	Automatisation des tests.....	7
5.	Déploiement	7
VII.	Gestion des Outils	7
	Design et Conception :	7
	Environnements de développement :	7
	Gestion de version :	7
	Suivi de projet :	7
	Outils de test :	7
VIII.	Planning Qualité.....	8
IX.	Documentation et Livrables.....	8

X.	Conclusion	8
-----------	-------------------------	----------

I. Introduction

1. Contexte du projet

Le projet Catalogue de Plantes Médicinales a pour objectif de développer une application intuitive et fiable qui offre aux utilisateurs des informations complètes sur les plantes médicinales, y compris leurs propriétés, utilisations et précautions.

Cette plateforme intégrera des fonctionnalités avancées telles qu'une recherche multicritères, des recommandations personnalisées et une gestion des interactions avec d'autres médicaments.

L'application est développée avec Flutter et Firebase pour la version mobile et pour la version web Java Spring Boot + Angular, des maquettes fonctionnelles de l'application mobile ont été réalisées avec Figma pour assurer une conception orientée utilisateur.

2. Objectif du PAQ

Mettre en place un cadre structuré pour garantir la qualité du produit final en définissant des processus, des outils et des normes adaptés à la gestion d'un projet collaboratif.

II. Références

Méthodologies utilisées :

- Agile/Scrum pour la gestion du projet.

Documents de référence :

- Spécifications fonctionnelles du système.
- Cahier des charges initial décrivant les besoins métiers.
- Maquettes réalisées sur Figma pour l'interface utilisateur mobile.
- Rapports d'audit qualité (SonarQube, Selenium et Jmeter).

III. Organisation de la Qualité

1. Responsabilités des membres

- **Chef de projet** : Gestion des tâches via Azure DevOps et supervision globale.
- **Tous les membres** : Collaboration et suivi via Azure DevOps, Jira, et Git.
- **Développeurs** :
 - ✓ Implémentation des fonctionnalités backend (Spring Boot) pour le web.
 - ✓ Implémentation des fonctionnalités mobiles (Firebase pour le backend, Angular pour le frontend, et Flutter pour le mobile).
- **Testeurs QA** : Validation de la conformité via Selenium, SonarQube, et tests de performance avec JMeter.
- **Designer UX/UI** : Création et validation des maquettes sur Figma
- **Tous les membres** : Collaboration et suivi des tâches sur Azure DevOps et Jira

2. Outils collaboratifs

- **Azure DevOps** : Suivi des tâches, CI/CD, gestion des pipelines.
- **Jira** : Gestion des sprints et suivi des anomalies.
- **Git** : Gestion de version pour suivre et fusionner les modifications.
- **Figma** : Création des maquettes pour valider les aspects visuels et ergonomiques.
- **JMeter** : Outil pour les tests de performance et de charge.
- **Jenkins** : Outil d'intégration continue pour automatiser les builds et les déploiements.

IV. Critères de Qualité Logicielle

Fonctionnalité

- Résultats pertinents pour la recherche multicritères.
- Recommandations adaptées basées sur les besoins de santé.

Fiabilité :

- Gestion des défaillances en cas de surcharge ou d'erreur utilisateur.
- Tests automatisés pour garantir une exécution robuste

Performance :

- Temps de réponse des recherches < 3 secondes.

- Chargement des images et vidéos en moins de 5 secondes.
- Tests de performances réalisés avec JMeter pour évaluer la charge et la scalabilité

Utilisabilité :

- Interfaces intuitives validées via les maquettes Figma.
- Facilité d'accès aux informations et recommandations.

Sécurité :

- Vérifications régulières via SonarQube pour détecter les vulnérabilités.

V. *Gestion des Exigences*

Traçabilité :

Lier chaque exigence utilisateur à des tâches spécifiques sur Azure DevOps/Jira.

Gestion des modifications :

Procédure Agile avec revue par l'équipe avant toute modification.

VI. *Processus d'Assurance Qualité*

1. Revue des spécifications :

Validation initiale par l'équipe QA des spécifications fonctionnelles.

Validation des exigences initiales et alignement avec les maquettes sur Figma.

2. Tests :

- **Unitaires** : Validation des modules backend et frontend.
- **Intégration** : Tests des interactions API/backend avec Angular, orchestres via Jenkins.
- **Système** : Simulation de scénarios réels avec Selenium WebDriver.
- **Performance** : Tests de charge et de performance réalisés avec JMeter.
- **Code qualité** : Vérifications avec SonarQube

3. Revue de code

- Analyse du code pour s'assurer qu'il respecte les standards définis

4. Automatisation des tests

- Utilisation de Selenium WebDriver pour automatiser les tests
- Utilisation de JMeter pour automatiser les tests de performance et de charge.
- **Jenkins** : Utilisé pour orchestrer l'exécution des tests automatisés et faciliter le déploiement des builds.

5. Déploiement

- **Jenkins** : Automatisation du processus de déploiement pour garantir des mises à jour fréquentes et fiables.

VII. *Gestion des Outils*

Design et Conception :

Figma : Prototypage des interfaces et validation UX/UI.

Enterprise Architect : Outil utilisé pour la conception des diagrammes UML (cas d'utilisation, classes, séquences, etc.).

Environnements de développement :

VS Code : Développement du frontend (Angular et Flutter).

IntelliJ IDEA : Développement du backend (Spring Boot).

Gestion de version :

Git : Suivi des modifications et gestion des branches.

GitHub/GitLab : Hébergement du code source et collaboration en équipe.

Suivi de projet :

Azure DevOps : Suivi des tâches et intégration continue (CI/CD).

JIRA : Gestion des sprints et priorisation des exigences.

Outils de test :

Selenium WebDriver : Tests automatisés pour les scénarios End-to-End (E2E).

Postman : Test des API backend.

SonarQube : Audits de qualité du code pour identifier les bugs et vulnérabilités.

JMeter : Tests de performance et de charge pour évaluer la scalabilité des applications.

VIII. *Planning Qualité*

- **Semaine 1-2** : Finalisation de conception et des maquettes sur Figma et validation des interfaces avec les parties prenantes.
- **Semaine 3-4** : Développement du backend (Spring Boot) et du frontend (Angular et Flutter) avec intégration des maquettes validées.
- **Semaine 5** : Analyse statique du code avec SonarQube, correction des anomalies, et validation des règles de qualité.
- **Semaine 6** : Phase de test avec Selenium et Jmeter , suivi de la résolution des anomalies et validation finale.
- **Semaine 7** : Phase de déploiement la Configuration des pipelines de déploiement dans Jenkins.

IX. *Documentation et Livrables*

Maquettes sur Figma :

Guide visuel des interfaces utilisateur.

Rapports de tests (selenium,Jmeter) :

Résultats des tests manuels et automatisés.

Rapports d'audit :

Résultats des analyses de qualité via SonarQube.

Rapport finale du projet : ce document contient les détails du travail.

X. *Conclusion*

Le Plan d'Assurance Qualité (PAQ) établit les standards nécessaires pour garantir que l'application **Catalogue de Plantes Médicinales** répondra aux attentes des utilisateurs en matière de fonctionnalité, de sécurité et de performance, tout en respectant les meilleures pratiques de développement logiciel.

Il assure également que l'application mobile et web respectera les exigences de qualité grâce à l'intégration d'outils collaboratifs modernes tels que **Figma , Git , Azure DevOps et Jenkins** . Ces outils permettent une gestion fluide des tâches et assurent une livraison conforme aux attentes des parties.