

Document de stratégie de test

1 Sommaire exécutif

Ce document regroupe une stratégie de test.

2 Historique des Versions

Date	Version	Commentaires
16/05/2022	0.1	

3 Contenu

1	Sommaire Exécutif.....	1
2	Historique de Version.....	1
3	Contenu.....	1
4	Objet.....	2
5	Demande de projet d'architecture et contexte.....	3
6	Description et portée du projet d'architecture.....	3
7	Vue d'ensemble de la vision de l'architecture.....	3
8	Procédures spécifiques et changement de périmètre.....	3
9	Rôles, responsabilités et livrables.....	3
10	Critères et procédures d'acceptation.....	3
11	Plan et calendrier du projet d'architecture.....	3
12	Approbations.....	3

Document de stratégie de test

4 Objet

Le présent document est une stratégie de test auquel le PoC se conforme.

5 Préparer et définir sa stratégie de tests

La stratégie de tests permet de réussir la phase de tests, ce document indique les prérequis nécessaires à la recette et guide les testeurs durant la phase de test. C'est un peu comme un « cahier des charges » avant de lancer un projet. Ci-dessous les points à indiquer dans la stratégie de tests.

- Périmètre
- Moyens de tests
- Environnement et URL
- Méthodologies et Procédures
- Type de tests prévues
- Jeux de données
- Criticité des incidents de fonctionnement
- Partie prenantes
- Planning

6 Rédiger les scénarios et les cas de tests

Rédaction des scénarios de test , en utilisant les terminologies Given That / When / Then.

7 Exécution des tests

Tests Unitaires : tests techniques des **développeurs** après leurs développements (prérequis aux tests fonctionnels).

Tests d'Intégration : tests techniques des couches services et repository (anciennement DAO).

Tests E2E (End-To-End) : tests de bout en bout de l'ensemble des fonctionnalités projet de l'application.

Tests automatisés : tests automatisés en utilisant gitLab avec le CI/CD.

Document de stratégie de test

8 Résultats

Les tests fonctionnels, de Bout en Bout et TNR permettent de vérifier que le logiciel est conforme aux spécifications décrites dans les Users stories. Ils peuvent même permettre de combler des trous fonctionnels. Ils assurent la qualité d'un logiciel.