

R5-8 - Qualité de développement

# PLAN DE TEST ET VALIDATION – Projet de Gestion de Bibliothèque

**Projet :** Système de Gestion de Bibliothèque Universitaire

**Groupe :** Naharro Guerby, Bonnard Nathan, Le Bastard Théo

**Date de rendu :** 13/02/2026

---

---

# SOMMAIRE

SOMMAIRE.....	2
<b>I. Introduction.....</b>	<b>3</b>
<b>II. STRATÉGIE DE TEST.....</b>	<b>3</b>
<b>III. ENVIRONNEMENT DE TEST.....</b>	<b>4</b>
<b>IV. SCÉNARIOS DE TEST (CAHIER DE RECETTE).....</b>	<b>4</b>
A. Installation et Démarrage.....	4
B. Authentification et Sécurité.....	4
C. Gestion du Catalogue (Livres).....	5
D. Workflow des Emprunts.....	5
E. Administration et Statistiques.....	6
<b>V. RAPPORT D'ANOMALIES (CORRIGÉES).....</b>	<b>7</b>
<b>VI. CRITÈRES DE RÉUSSITE.....</b>	<b>7</b>

---

# I. Introduction

Ce projet s'inscrit dans le cadre de la modernisation des infrastructures numériques de l'université. L'objectif est de réaliser une **application web complète (Full Stack)** permettant la gestion automatisée d'une bibliothèque universitaire.

Le projet servira de support pour évaluer :

- La maîtrise de l'architecture client-serveur (React/Django).
- La qualité du code et la gestion de base de données.
- La capacité à déployer une solution conteneurisée (Docker).

Équipe de développement :

- Naharro Guerby
  - Bonnard Nathan
  - Le Bastard Théo
- 

# II. STRATÉGIE DE TEST

L'objectif est de valider le bon fonctionnement de l'application sur les environnements Windows et Linux (via Docker). Les tests couvrent trois niveaux :

1. **Tests d'Installation** : Vérification du déploiement Docker.
  2. **Tests Fonctionnels (Black Box)** : Validation des fonctionnalités utilisateur via l'interface React.
  3. **Tests d'Intégration** : Vérification de la cohérence des données entre le Front, l'API et la BDD.
-

### III. ENVIRONNEMENT DE TEST

**OS Hôte** : Mac OS (M1).

**Conteneurisation** : Docker Desktop.

**Navigateurs** : Google Chrome, Safari.

**Base de données** : PostgreSQL 16 (Image Docker officielle).

---

### IV. SCÉNARIOS DE TEST (CAHIER DE RECETTE)

#### A. Installation et Démarrage

Scénario	Étapes	Résultat Attendu	Statut
Lancement Docker (Mac OS)	1. docker-compose down -v  2. docker-compose up --build	Les conteneurs se lancent sans erreur "Entrypoint not found". La BDD s'initialise.	OK
Entrée des donnée en base	Vérifier les logs au démarrage ou l'interface.	Base de donnée remplie	OK

#### B. Authentification et Sécurité

Scénario	Étapes	Résultat Attendu	Statut
Connexion Admin	Login avec admin / admin.	Redirection vers le Dashboard. Le menu "Admin" apparaît.	OK
Connexion Élève	Login avec Ambre / pwd123.	Redirection vers l'accueil. Pas de menu "Admin".	OK
Erreur Login	Tenter un mot de passe faux.	Message d'erreur "Identifiants invalides" (Erreur 401).	OK
Déconnexion	Cliquer sur "Déconnexion".	Retour à la page de Login. Token supprimé.	OK

## C. Gestion du Catalogue (Livres)

Scénario	Étapes	Résultat Attendu	Statut
Affichage Grille	Aller sur la page d'accueil.	Les livres s'affichent en grille (4 par ligne sur PC).	OK
Recherche	Taper "Python" dans la barre.	La liste ne montre que les livres contenant "Python".	OK
Ajout Livre (CRUD)	1. (Admin) Cliquer sur "Nouveau Livre".  2. Remplir le formulaire.	Le livre apparaît immédiatement dans la liste et en BDD.	OK
Modification	(Admin) Changer le stock d'un livre.	La modification est prise en compte.	OK

## D. Workflow des Emprunts

Scénario	Étapes	Résultat Attendu	Statut
Emprunt Standard	(Élève) Cliquer sur "Emprunter".	Message succès. Stock du livre décrémenté de 1.	OK
Stock Épuisé	1. Trouver un livre avec Stock = 0.  2. Tenter d'emprunter.	Bouton désactivé ou message d'erreur "Stock épuisé".	OK
Vérification Date	Aller dans "Mes Emprunts".	La date de retour est fixée à J+30.	OK
Retour Livre	1. (Admin) Dashboard -> Retours.  2. Cliquer sur "Rendre".	L'emprunt passe en "Retourné". Stock réincrémenté (+1).	OK

## E. Administration et Statistiques

Scénario	Étapes	Résultat Attendu	Statut
Dashboard	Charger la page Dashboard.	Les compteurs (Utilisateurs, Livres, Prêts) ne sont pas à 0.	OK
Suppression User	Gestion Users -> Supprimer "Tom".	L'utilisateur disparaît de la liste.	OK
Détection Retard	1. Simuler un prêt passé.  2. Vérifier Dashboard.	Le prêt apparaît dans la section "En Retard" (Rouge).	OK

---

## V. RAPPORT D'ANOMALIES (CORRIGÉES)

Voici les bugs majeurs rencontrés durant le développement et leur résolution :

Description	Cause	Solution Apportée
Crash au démarrage sur Windows ("entrypoint.sh not found").	Gestion des fins de ligne (CRLF vs LF).	Suppression du fichier script et injection des commandes via docker-compose command.
Dates de retour invisibles dans le tableau.	Nom de variable incorrect (deadline vs return_date).	Harmonisation du Frontend avec le sérialiseur Django.
Impossible de supprimer un utilisateur.	Contraintes de clé étrangère (Emprunts actifs).	Ajout d'une gestion d'erreur (try/catch) et alerte utilisateur.

---

## VI. CRITÈRES DE RÉUSSITE

Le projet est considéré comme valide car :

1. **Stabilité** : Aucune erreur 500 (Server Error) lors des scénarios nominaux.
2. **Conformité** : Toutes les fonctionnalités du Cahier des Charges sont présentes.
3. **Performance** : Le chargement des pages catalogue est inférieur à 1 seconde.

**Conclusion** : L'application est apte au déploiement.

---