

Cahier des charges

Portail de suivi du doctorat



Réalisé par : Pr.AHAIDOUS Khadija

Le : 30 septembre 2025

1 - Contexte et définition du projet

Le cycle de vie d'un doctorant, de son inscription initiale jusqu'à la soutenance de sa thèse, implique de nombreuses étapes administratives. Le processus actuel repose sur des procédures manuelles (échanges d'emails, documents papier...), ce qui est source d'erreurs, de retards et d'un manque de visibilité pour toutes les parties prenantes.

2 - Objectif du projet

L'objectif de ce projet est de développer une plateforme web centralisée, moderne et efficace pour dématérialiser et optimiser l'ensemble du parcours doctoral. L'application couvrira l'inscription, la réinscription annuelle et l'intégralité du processus de soutenance. Les objectifs principaux sont :

- Simplifier les démarches d'inscription, de réinscription et de demande de soutenance pour les doctorants.
- Fluidifier les étapes de validation (par les directeurs de thèse, les rapporteurs, les services administratifs).
- Centraliser l'ensemble des données et documents relatifs au parcours de chaque doctorant.
- Fournir une visibilité en temps réel sur l'état d'avancement des dossiers (inscription, soutenance, etc.).
- Garantir que les conditions préalables à la soutenance (publications, formations) sont bien remplies.
- Automatiser les tâches répétitives, les notifications et la génération de rapports.

3 – Utilisateurs cibles

Profil	Description du rôle
Candidat / Doctorant	Soumet ses dossiers (inscription, réinscription, soutenance). Suit l'avancement. Dépose son manuscrit de thèse.
Directeur de Thèse / Encadrant	Suit le statut du doctorant. Signe les documents nécessaires. Propose les membres du jury.
Personnel Administratif	Valide les dossiers d'inscription et les demandes de soutenance. Gère le processus global. Vérifie la conformité des dossiers. Planifie administrativement la soutenance.

4 - Description fonctionnelle des besoins

L'application sera organisée autour des modules fonctionnels suivants :

Module 1 : Gestion des comptes et authentification

Ce module est transverse et gère l'identité des utilisateurs :

- Création de compte pour les nouveaux candidats.
- Connexion / Déconnexion sécurisée pour tous les profils.
- Gestion des rôles et des permissions (un administrateur doit pouvoir assigner les rôles adéquats).

Module 2 : Processus d'inscription et de réinscription

Ce module gère le début du cycle et son renouvellement annuel.

- Gestion des campagnes d'inscription/réinscription par l'administration (définition des dates d'ouverture et de fermeture).
- Formulaire d'inscription dynamique pour le candidat (informations personnelles, sujet de thèse, directeur de thèse, collaboration etc.).
- Téléversement de pièces justificatives (diplômes, CV, lettre de motivation, etc.) avec vérification du format (PDF, JPG...).
- Circuit de validation en ligne :
 - 1 Le dossier est soumis par le candidat.
 - 2 Notification au directeur de thèse pour avis.
 - 3 Notification à l'administration pour vérification et validation administrative.
- Tableau de bord pour le doctorant lui permettant de suivre l'état de son dossier en temps réel.
- Processus de réinscription simplifié (reprise des données de l'année N-1, avec mise à jour des informations nécessaires).

Module 3 : Processus de soutenance

Ce module gère la fin du cycle doctoral.

- Formulaires de demande de soutenance initiés par le doctorant.
- Check-list de prérequis à la soutenance (validée par le système et l'administration) :
 - Nombre de publications scientifiques.
 - Crédits de formation doctorale obtenus.
 - Liste des documents à fournir (Demande manuscrite adressée au chef de l'établissement, rapport de thèse, rapport anti-plagiat, Rapport contenant les publications et les communications, Les copies d'attestations de formations et autorisation de soutenance)

- Proposition du jury par le Directeur de Thèse :
 - Désignation des rapporteurs, examinateurs etc.
- Autorisation de soutenance et planification administrative : Une fois les rapports favorables reçus, l'administration donne l'autorisation finale, fixe et communique la date, l'heure et le lieu de la soutenance.

Module 4 : Notification et communication

Ce module assure la communication entre les utilisateurs et le système.

- Envoi d'emails automatiques à chaque étape clé (dossier soumis, en attente de validation, validé, etc.).
- Génération de documents PDF : attestation d'inscription, autorisation de soutenance, procès-verbal de soutenance (pré-rempli).

5 – Règles de gestion et paramétrage

Règles de gestion clés

L'application devra implémenter les règles de gestion suivantes :

- **Durée du doctorat** : Un doctorant ne peut pas se réinscrire si son inscription initiale date de plus de 3 ans, sauf dérogation exceptionnelle accordée par le Pôle d'Études Doctorale (PED).
- **Durée maximale du doctorat** : Un doctorant ne peut pas dépasser 6 ans. Le système devra lever une alerte lorsque cette limite est approchée ou atteinte.
- **Prérequis à la soutenance** : La soumission d'une demande de soutenance ne sera possible que si le doctorant a rempli un ensemble de conditions :
 - Au moins 2 articles journaux (Q1/Q2)
 - Au moins 2 conférences (ou équivalent)
 - Compléter 200h de formation

Paramétrage de l'Application

Afin de garantir la flexibilité et l'évolutivité de la plateforme, certaines données devront être administrables via une interface de gestion réservée au personnel administratif.

- **Gestion des Campagnes** : L'administrateur doit pouvoir créer des campagnes d'inscription et de réinscription en définissant leurs dates d'ouverture et de fermeture.
- **Gestion des Prérequis** : Les critères de validation pour la soutenance (nombre de publications, heures de formation, etc.) doivent être configurables.
- **Gestion des Modèles de Documents (Templates)** : L'application utilisera des modèles pour générer des documents officiels. Le contenu de ces modèles doit être éditable par l'administrateur. Cela inclut :
 - Les modèles pour les emails de notification.
 - Le modèle de l'autorisation de soutenance.

6 – Les exigences techniques

Architecture logicielle

L'application devra être développée en respectant une architecture micro-services. Cette approche vise à découpler les différentes parties de l'application pour améliorer la maintenabilité, la résilience et l'évolutivité.

- **Frontend** : Une application monopage (Single Page Application - SPA) sera développée avec le framework Angular. Elle communiquera avec le backend via des API REST.
- **Backend** : Le backend sera composé de plusieurs micro-services développés avec Spring Boot et orchestrés à l'aide de l'écosystème Spring Cloud. Chaque micro-service correspondra à un domaine fonctionnel (ex: service des utilisateurs, service des inscriptions, service des soutenances).
- **Communication Asynchrone** : Pour les communications interservices qui ne nécessitent pas de réponse immédiate (ex: notifications, déclenchement de traitements longs), le bus de messages Apache Kafka sera utilisé. Cela permettra de construire une architecture événementielle (Event-Driven).
- **Traitements par Lots (Batch)** : Les tâches planifiées et le traitement de données en masse (ex: génération de rapports de fin de mois, archivage de dossiers) seront implémentés avec Spring Batch.

- **Passerelle API (API Gateway) :** Un service Spring Cloud Gateway sera mis en place. Il agira comme unique point d'entrée pour l'application frontend, sécurisant et routant les requêtes vers les micro-services appropriés.
- **Découverte de Services (Discovery Service) :** Un mécanisme de découverte de services (Spring Cloud Netflix Eureka) sera utilisé pour permettre aux micro-services de s'enregistrer et de se trouver dynamiquement sur le réseau.
- **Gestion de la Configuration Centralisée :** La configuration des différents micro-services (Ex: adresses de bases de données, paramètres applicatifs) devra être externalisée et gérée de manière centralisée à l'aide de Spring Cloud Config. Cela permet de modifier la configuration sans avoir à redéployer les services.
- **Résilience et Tolérance aux Pannes :** Pour gérer les défaillances partielles inhérentes à une architecture distribuée, la bibliothèque Resilience4j devra être implémentée.
 - Circuit Breaker: Pour éviter d'appeler un service qui est en panne de manière répétée.
 - Retry : Pour retenter automatiquement une opération qui a échoué de manière transitoire.
 - Fallback : Pour fournir une réponse alternative ou par défaut lorsqu'un service est indisponible.

La base de données

Chaque micro-service devra posséder sa propre base de données pour garantir le découplage (pattern "Database per service"). Le choix se portera sur une base de données relationnelle comme MariaDB, PostgreSQL ou MySQL.

Environnement de développement et déploiement

- **IDE :** IntelliJ Ultimate edition
- **Gestion de version :** Le code source sera géré avec Git et publié sur Github.
- **Conteneurisation :** L'ensemble des services (frontend, backend, bases de données) devra être conteneurisé avec Docker.

Livrables attendus

- **Code source** : L'intégralité du code source de l'application (frontend, micro-services) devra être accessible sur Github.
- **Présentation finale** : Support de présentation + démonstration couvrant les principaux scénarios d'utilisation
- **Journal de bord** : Fichier .md à inclure dans le répertoire, contenant :
 - Les tâches réalisées par chaque membre de l'équipe.
 - Les problèmes et difficultés rencontrés.

