**Cahier des Charges**

**Sujet : Conception et développement d’une plateforme de gestion des documents de développement et génération de projets Spring**

**Contexte et Problématique**

Actuellement, l'équipe de développement fait face à plusieurs limitations et problèmes dans la gestion des normes de développement interne et des documents associés :

1. **Absence de centralisation des documents métiers :**  
   Les normes de développement (frameworks, configuration, testing, sécurité, dépendances, etc.) sont actuellement dispersées et gérées sous forme de fichiers statiques (documents Word ou PDF). Cela engendre des difficultés dans :
   * La gestion des versions et l'historique des modifications.
   * La mise à jour des documents (processus manuel, non collaboratif).
   * L’accès rapide à ces normes.
2. **Manque de uniformisation et automatisation :**
   * Les développeurs souffrent d'une hétérogénéité dans les pratiques de développement (ex : nomenclature des branches Git, prise en compte irrégulière des normes).
   * L'onboarding d’un nouveau développeur est complexe, car il n'existe pas d'outils standardisés pour démarrer rapidement avec les bonnes pratiques.
3. **Besoin de génération de projets standardisés :**  
   Lorsqu’un nouveau projet Spring est démarré, il faut manuellement respecter toutes les normes définies, notamment en termes d’architecture (ex : hexagonale) et d’utilisation des dépendances internes (platforme). Un générateur de projets respectant les normes internes pourrait non seulement simplifier ce processus, mais aussi garantir la conformité des livrables.

**Objectif Général**

L’objectif principal est de concevoir et développer une **plateforme centralisée clé en main** pour :

1. Gérer les documents des normes internes avec des fonctionnalités avancées (CRUD, gestion des droits d'accès, historique des modifications , voir les documents).
2. Permettre la **génération automatique de projets Spring** conformes aux normes internes de développement en suivant une architecture prédéfinie et intégrant les dépendances internes.

**Objectifs Spécifiques**

1. **Améliorer la gestion et l’accessibilité des normes** :
   * Centraliser tous les documents relatifs aux normes de développement dans un référentiel unique, accessible via une interface dédiée.
   * Fournir une interface permettant de consulter, modifier, ajouter ou supprimer des normes.
   * Permettre le tracking complet de chaque modification (historique des changements avec auteur et date).
2. **Faciliter l’adoption des normes internes via des outils modernes :**
   * Standardiser les pratiques d'utilisation des frameworks, de la configuration, du testing et des dépendances grâce à une plateforme simple et intuitive.
3. **Simplifier la création de projets avec respect des normes internes (Générateur de projets Spring)** :
   * Proposer une interface qui permet de configurer les caractéristiques initiales d’un projet Spring (version de Java, Maven/Gradle, nom du projet, dépendances à inclure).
   * Générer automatiquement une structure de projet respectant les normes d’architecture hexagonale définie par l’entreprise (organisation des services en packages spécifiques).
   * Inclure des dépendances prédéfinies comme : platforme, foundation et autres outils internes.

#### Génération de projets Spring

* **Configurer les caractéristiques d’un projet Spring Personalizable** :  
  Lors de la génération, permettre à l'utilisateur de définir :
  + Version de Java (ex : 8, 11, 17…).
  + Système de compilation (Maven ou Gradle).
  + Nom du projet.
  + Inclusion de dépendances internes comme platforme, foundation ou autres.
  + Inclusion de starters ou modules spécifiques (Spring Boot Web, JPA, Security…).
* **Respect des normes internes pour la structure du projet généré** :  
  Produire un projet respectant la structure suivante :
  + **Service API** : DTOs, interfaces pour les APIs, use cases.
  + **Service Core** : Implémentations des use cases, gestion de la persistance (JPA, gateways).
  + **Service Content** : Contiendra la classe BeanFactory et le point d’entrée principal (Main).
* **Insertion automatique de configurations de base** :
  + Intégrer une configuration Git standardisée (nommage et gestion des branches).
  + Créer des squelettes de tests unitaires et d’intégration.
* processus de mise à jour du système.