



## Sujet de stage : **Développement d'une Application de Gestion des Appels d'Offres**

### **1. Introduction**

L'objectif de ce document est de définir les exigences fonctionnelles et non fonctionnelles pour le développement d'une application de gestion des appels d'offres. Cette application sera développée en utilisant Spring Boot pour la partie backend et Angular pour la partie frontend.

### **2. Objectifs**

L'application doit permettre aux utilisateurs de :

- Créer, modifier et supprimer des appels d'offres.
- Validation et publication ou refus et clôture des appels d'offres.
- Consulter la liste des appels d'offres valider et publier.
- Consulter la liste des appels d'offres refuser et clôturer.
- Consulter la liste de tous les appels d'offres.
- Gérer les utilisateurs et leurs permissions.

### **3. Exigences fonctionnelles**

#### **3.1 Authentification et autorisation**

Les utilisateurs doivent pouvoir s'inscrire et se connecter.

Différents niveaux d'autorisation doivent être définis (administrateur, utilisateur normal, etc.).

Seuls les utilisateurs autorisés peuvent créer, modifier ou supprimer des appels d'offres.

#### **3.2 Gestion des appels d'offres**

Les utilisateurs autorisés peuvent créer de nouveaux appels d'offres en spécifiant les détails tels que la description, les dates de début et de fin, les critères de sélection, etc.

Les appels d'offres doivent pouvoir être modifiés ou supprimés par leurs créateurs.

Les utilisateurs doivent pouvoir consulter la liste des appels d'offres disponibles avec des filtres de recherche.

### **3.3 Soumission des offres**

Les utilisateurs autorisés peuvent soumettre des offres pour les appels d'offres disponibles en fournissant les informations requises.

Les soumissions d'offres doivent être horodatées et leur statut doit être suivi (en attente, acceptée, rejetée, etc.).

### **3.4 Gestion des utilisateurs**

Les administrateurs doivent pouvoir gérer les utilisateurs en ajoutant, modifiant ou supprimant des comptes.

Les informations utilisateur telles que le nom, l'email et le rôle doivent être enregistrées.

## **4. Exigences non fonctionnelles**

### **4.1 Performance**

L'application doit être rapide et réactive, même lorsqu'elle est utilisée par de nombreux utilisateurs simultanément.

### **4.2 Sécurité**

Toutes les données sensibles doivent être stockées de manière sécurisée.

Les utilisateurs doivent être authentifiés avant d'accéder à des fonctionnalités sensibles.

### **4.3 Interface utilisateur conviviale**

L'interface utilisateur doit être conviviale et intuitive pour permettre une utilisation facile de l'application.

Les validations côté client doivent être mises en place pour assurer la saisie correcte des données par les utilisateurs.

### **4.4 Extensibilité**

L'architecture de l'application doit être conçue de manière à permettre l'ajout de nouvelles fonctionnalités à l'avenir.

## **5. Technologies**

Backend : Spring Boot (Java)

Frontend : Angular (TypeScript)

Base de données : MySQL, PostgreSQL, ou toute autre base de données relationnelle.

Communication entre le frontend et le backend : API RESTful.

## **6. Livrables**

Code source de l'application.

Documentation technique.

Instructions d'installation et de déploiement.

**Ceci est un document interne et ne doit pas être partagé  
avec des utilisateurs non autorisés.**