**Document de Spécification Technique pour l'Application Web d'INTIA assurance**

**1. Introduction**

L'objectif de ce projet est de développer une application web pour INTIA assurance, qui permet de gérer les informations relatives aux clients et aux contrats d'assurance. Cette application doit inclure des fonctionnalités d'ajout, de modification, de suppression et de consultation des informations pour les clients et leurs assurances. Le système doit être accessible à la direction générale et aux deux succursales : INTIA-Douala et INTIA-Yaoundé.

**2. Architecture du Système**

Le système sera basé sur une architecture client-serveur :

* **Front-end** : Une interface utilisateur web, développée avec ReactJS.
* **Back-end** : Un serveur RESTful développé avec Spring Boot.
* **Base de données** : Postgresql pour stocker les informations relatives aux clients et aux assurances.
* **API** : Le back-end exposera des endpoints API sécurisés pour permettre la gestion des données.

**3. Fonctionnalités Principales**

1. **Gestion des clients** :
   * Ajout de nouveaux clients avec des informations telles que nom, prénom, adresse, numéro de téléphone, email, etc.
   * Modification des informations des clients.
   * Suppression des clients.
   * Consultation de la liste des clients et détails individuels.
2. **Gestion des assurances** :
   * Création de nouveaux contrats d'assurance pour un client.
   * Modification des contrats d'assurance.
   * Suppression de contrats d'assurance.
   * Consultation des contrats d'assurance par client ou globalement.
3. **Gestion des agences** :
   * La direction générale pourra avoir une vue d'ensemble sur toutes les branches (Douala, Yaoundé).
   * Les succursales (Douala et Yaoundé) auront accès uniquement aux données des clients et assurances de leur agence respective.

**4. Solution Proposée**

* **Front-end** : ReactJS pour créer une interface interactive et réactive. Le framework Formik sera utilisé pour la gestion des formulaires, et Axios pour les appels API vers le back-end.
* **Back-end** : Spring Boot pour créer un back-end performant avec des API RESTful, et Spring Data JPA pour les opérations de base de données.
* **Base de données** : Postgresql. Chaque succursale aura sa propre instance de données séparée par un attribut de localisation (Douala ou Yaoundé).

**5. Modèle de Données**

* **Client** : id, nom, prenom, adresse, telephone, email, branche.
* **Assurance** : id, type\_assurance, date\_debut, date\_fin, montant, client\_id, branche.
* **Branche** : id, nom, localisation (Douala ou Yaoundé).

**6. Cas d'Utilisation**

1. **Ajout d’un nouveau client** :
   * Un agent se connecte.
   * L’agent peut ajouter un nouveau client avec toutes les informations requises (nom, adresse, etc.).
2. **Modification des informations d’un client** :
   * L’agent peut modifier les informations des clients existants.
3. **Ajout d’un contrat d’assurance** :
   * L’agent peut ajouter une assurance pour un client.
4. **Suppression d’un contrat ou d’un client** :
   * Les contrats ou clients peuvent être supprimés selon certaines règles métier

* **Admin** : Accès à toutes les fonctionnalités pour gérer clients, assurances, et succursales.
* **Agent** : Accès restreint aux informations de sa propre succursale (Douala ou Yaoundé).
* **Utilisateur** : Accès pour consulter ses propres informations et les contrats qui lui sont associés.

**8. Technologies Utilisées**

* **Frontend** : ReactJS, Axios, Formik, CSS3.
* **Backend** : Java, Spring Boot, Spring Data JPA, Spring Security.
* **Base de données** : Postgresql.
* **9. Plan de Déploiement**
* **Environnement de développement** : Préparation d'une machine locale.

**10. Exigences Non-Fonctionnelles**

* **Performance** : L'application doit supporter un grand volume de transactions simultanées sans délai important.
* **Scalabilité** : Le système doit pouvoir évoluer pour inclure de nouvelles agences si nécessaire.
* **Sécurité** : Les données des clients doivent être protégées par une authentification sécurisée et des communications cryptées.