

# I Document de Spécifications Techniques

## Projet : Application web de Gestion des clients et leurs assurances – chez INTIA ASSURANCE

### 1. Présentation Générale du Projet

La société INTIA Assurance souhaite disposer d'une application web permettant de gérer les informations relatives à ses clients et à leurs assurances. Cette application permettra à l'entreprise de :

- Ajouter un client ou une assurance.
- Modifier un client ou une assurance.
- Supprimer un client ou une assurance.
- Consulter les informations des clients et des assurances.

INTIA possède :

- Une direction générale
- Deux succursales : INTIA-Douala et INTIA-Yaoundé.

### 2. Objectifs du Projet

- Digitaliser la gestion des clients et des contrats d'assurances.
- Centraliser les informations dans une base de données unique.
- Rendre l'application accessible aux utilisateurs des 3 sites (Direction Générale, INTIA-Douala, INTIA-Yaoundé).
- Faciliter les opérations CRUD (Create, Read, Update, Delete) pour les agents de INTIA.

### 3. Architecture du Système

Couche	Technologie	Description
Frontend	Symfony (Twig) + Bootstrap 5	Génération des pages web côté serveur, responsive design, interface utilisateur adaptée
Backend	Symfony 7,2 (PHP 8,2)	Gestion des fonctionnalités métier, sécurité, logique de traitement, gestion des rôles et authentification
Base de Données	MySQL 8	Stockage relationnel des données (utilisateurs, assurances, succursales)
Hébergement	Serveur Web Apache 2	Hébergement des fichiers du projet Symfony et gestion des requêtes HTTP

## 4. Cas d'Utilisation

Acteur	Fonctionnalité	Description
Administrateur	Gérer les agents, les clients et contrats d'assurances	Créer, modifier, supprimer un agent INTIA
Agent Assurance		Ajouter, consulter, modifier, supprimer les assurances

## 5. Technologies Proposées

Symfony Bootstrap Mysql Serveur web Apache.

## 6. Contraintes Fonctionnelles

- Authentification obligatoire pour accéder à la plateforme.
- Gestion des rôles et permissions (Admin, Agent).
- Les actions de gestion doivent respecter le CRUD.

## 7. Contraintes Non Fonctionnelles

- Sécurité des données (hashage des mots de passe).
- Application responsive.
- Application rapide et optimisée.

## II Démarche de test

Type de Test	Objectif	Outils Utilisés
Tests Fonctionnels Manuels	Vérifier le bon fonctionnement des actions principales (CRUD, navigation)	Navigateur Web, Symfony Profiler
Tests Automatisés Unitaires	Vérifier le fonctionnement isolé des méthodes PHP	PHPUnit (intégré à Symfony)
Tests d'Interface	Vérifier le rendu visuel et responsive des pages	Chrome
Tests de Sécurité	Vérifier les accès utilisateurs (selon les rôles)	Symfony Security, Accès contrôlés