**Titre** : **Audit de la Solution Logicielle de Good Food**

**1. Introduction**

L’audit de la solution logicielle de Good Food vise à évaluer la qualité, la sécurité et la performance de l’application de commande en ligne afin d'identifier les axes d'amélioration et de garantir son alignement avec les objectifs stratégiques de l'entreprise.

**Objectifs de l'audit**

Évaluer l’état actuel de l’application et de l’infrastructure.

Identifier les points faibles et risques majeurs.

Proposer des recommandations pour l’amélioration continue.

**Contexte de l'entreprise Good Food**

Good Food est une entreprise de restauration en pleine expansion avec 150 restaurants répartis en France, Belgique et Luxembourg. L’entreprise a un besoin croissant d’adapter son SI pour répondre aux nouvelles attentes de ses clients.

**Parties prenantes impliquées**

* Direction informatique.
* Équipes métiers (comptabilité, communication).
* Prestataires externes (WIM, PWI, TP System).

**Contraintes et enjeux**

* Dépendance aux prestataires externes.
* Migration des données vers l'ERP Microsoft.
* Absence de documentation technique sur l’application actuelle.

**2. Résumé Exécutif**

**Principaux constats**

* Infrastructures obsolètes (Windows Server 2008 R2).
* Absence de solution mobile.
* Difficulté de maintenance du code actuel.

**Recommandations clés**

* Refondre complètement l'application avec des technologies modernes (et open source ?).
* Mettre en place une nouvelle architecture évolutive et scalable.
* Assurer la documentation et la formation des équipes.

**Feuille de route proposée**

* Conception de l’architecture cible.
* Migration progressive des données.
* Développement des services critiques et couverture de tests à 80%.
* Déploiement et suivi post-production.

**3. Analyse de l'Architecture du Système**

## Description actuelle de l'architecture applicative

* Application développée en ASP.NET C#.
* Base de données SQL Server 2008.
* Synchronisation via WCF.

## Infrastructure existante

* Hébergement sur les serveurs de PWI.
* Dépendance à des prestataires externes pour la maintenance.

## Problématiques identifiées

* Scalabilité limitée.
* Faible résilience face aux pics de charge.

## Points d'amélioration proposés

* Adoption d'une architecture cloud scalable --> serverless.
* Séparation des services en microservices.

**4.** Analyse de la Qualité du Code

## Outils utilisés pour l'analyse

* SonarQube pour détecter les vulnérabilités et la dette technique.
* Tests automatisés pour évaluer la robustesse du code.

## Complexité du code et dette technique

* Code monolithique difficilement maintenable.
* Dépendance à des bibliothèques obsolètes.

## Adéquation aux bonnes pratiques de développement

* Manque de tests unitaires et de documentation.

## Maintenabilité et évolutivité du code

Nécessité de refondre complètement le code.

# 5. Sécurité et Conformité

## Analyse des vulnérabilités

* Absence de chiffrement des données sensibles.
* Risques d’attaques par injection SQL.

## Gestion des accès et authentification

* Vérification de la conformité avec le RGPD.
* Mise en place de script SQL pour anonymiser les données obsolètes ou d'un background service

## Plan de remédiation des risques

* Implémentation de la norme PCI DSS pour le paiement.
* Audit de sécurité régulier. Planification de pentest tous les 6 mois

# 6. Performance et Disponibilité

## Résultats des tests de charge et de stress

Temps de réponse élevé aux heures de pointe.

## Optimisation des temps de réponse

* Mise en cache des données fréquentes.
* Réduction des appels à la base de données.

## Recommandations pour améliorer la disponibilité

Utilisation d'un CDN pour améliorer les performances sur les photos des plats.

# 7. Exploitabilité et Maintenabilité

## Documentation existante

Documentation insuffisante ou inexistante.

## Processus de déploiement

* Nécessité d'automatiser les déploiements avec CI/CD.
* Intégration de SonarQube dans la pipeline pour suivre les progressions d'amélioration du code

## Surveillance et monitoring

Implémentation d'outils de monitoring proactif (grafana / prometheus par exemple)

# **8. Analyse des Processus Métier**

## Adéquation avec les besoins fonctionnels

Nécessité d’un nouveau back offices pour les franchisés.

## Identification des incohérences

Revoir la synchronisation entre la base Good Food et l’ERP. Discutez de l’utilisation de L’ERP et voir s’il peut être remplacé par une application interne. (Voir le prix de l’ERP et les économies qu’on pourrait faire si on le remplace)

**9. Livrables de l'Audit**

## Recommandations et plan d'action détaillé.

## Indicateurs de qualité (KPIs).

**10. Conclusion et Recommandations Finales**

## Synthèse des constats et recommandations.

Faire un tableau avec les constats et les recommandations pour mitiger les risques

Trier les recommandations par ordre d’importances et de temps de mise en œuvre.

## Suivi des prochaines étapes.

 Faire une road map des prochaines étapes d’évolution avec des deadlines pour pouvoir suivre les évolutions.

**Annexes**

## Glossaire des termes techniques.

## Liste des outils utilisés.

## Documentation de référence.

## Captures d'écran et logs.