Cahier de charge sur les fonctionnalités des l’application de gestion d’une église

# Contexte

## Organisation du document

Ce document concerne les spécifications fonctionnelles du futur système ainsi que les recommandations sur l’infrastructure et réseaux.

La partie qui suit, contiendrait une description succincte de l’ensemble de nos recommandations.

Ensuite, chaque module du futur système fera l’objet d’une section dédiée et y seront présentées les différentes fonctionnalités à mettre en place.

La partie à propos de nos recommandations sur l'infrastructure et les réseaux précède notre conclusion pour clôturer ce document.

# Description générale du futur système

## Architecture logicielle du système

De nos jours, 2 options d'architecture sont mises à notre disposition. Nous avons d’une part celle dite desktop et d’autre part celle multi-tiers basée sur le Web.

### Architecture desktop

Une application construite autour de ce modèle n’est destinée qu’à une plateforme unique bien définie (OS X, Linux, Windows, etc.). Cette façon de procéder implique donc nécessairement une installation de l’application pour chaque ordinateur client. Ces types d’applications sont développées avec la technologie comme Visual Basic, Windev, etc …

1. **Avantages**

* Généralement, cette architecture offre une plus forte interactivité au niveau de l’interface graphique.
* Du fait que l’application a été conçue pour une plateforme spécifique, elle a donc accès aux fonctionnalités natives de cette dernière.

1. **Inconvénients**

* La maintenance peut occasionner des soucis d’incompatibilités lors :
  + d'une mise à jour de l’OS en question (ex : passer de Windows 10 à Windows 11)
  + d’un changement de plateforme (ex : passer d'OS X à Linux).

Il faudra alors revoir en intégralité l’application avant de pouvoir l’utiliser de nouveau.

* Il faudra également développer plusieurs versions de la même application pour la rendre accessible sur plusieurs plateformes (ex : une version pour mobiles (Android, iOS), une autre pour Windows, etc.).

### Architecture multi-tiers basée sur le web

La principale différence ergonomique avec l’architecture présentée plus haut réside dans le fait qu’il n’y a aucun logiciel à installer pour les clients. Ces derniers n’auront besoin que d’un navigateur Web, normalement fourni avec la plateforme, qui serait alors la porte d’entrée de l’application. Ainsi, elle deviendrait multiplateforme.

En optant pour cette architecture, le développement de l’application se fera sur plusieurs couches, notamment :

* une couche d’accès aux données, responsables des opérations vers le système de gestion de la base des données,
* une couche métier, qui sera en charge de l’implémentation des logiques métiers et les règles de gestion du système, et une couche de présentation.

1. **Avantages**

* L’application est accessible n’importe où à tout moment tant que le client est connecté aux réseaux locaux de l’entreprise ou à l’Internet (sous réserve de la mise en place des dispositifs de sécurité adéquate),
* La maintenance de l’application est plus aisée.

1. **Inconvénient**

* Son développement est plus chronophage étant donné qu' il sera forcément sur plusieurs couches.

### Notre recommandation

Malgré l’interactivité appréciable proposée par l’architecture desktop, nous recommandons d’opter pour la mise en place d’une application multi-tiers orienté web.

En effet, nous jugeons qu’il serait préférable que les informations soient plus accessibles et également que l’application soit plus indépendante de la plateforme où elle est utilisée.

La future application sera disponible sur tous les sites et même en dehors via Internet par l’intermédiaire de la mise en place des VPN (Virtual Private Network). Cf recommandation sur l'infrastructure et réseaux.

## Choix technologiques

Les choix des technologies nécessaires à la réalisation de l’application s’avèrent très importantes dans la réussite d’une telle mission.

### Plateforme de développement

Ces plateformes sont principalement constituées de frameworks avec le ou les langage(s) de programmation pour leur exploitation. Nous distinguons parmi les plus en vogues dans le monde professionnel :

1. **Fullstack Javascript**

De nos jours, Javascript est devenu un langage de programmation réputée, notamment grâce à l’émergence de différents frameworks, tels que Node.JS pour la partie back-end et Angular et React assurant le côté front-end pour ne citer qu’eux, qui forment ensemble l’univers Javascript.

1. **Avantages**

* La rapidité est remarquable.
* L’unicité du langage de programmation, à savoir Javascript, est aussi un atout considérable pour les développeurs de l’application.

1. **Inconvénient**

* Il s’agit d’une technologie jeune ce qui signifie qu’elle n'a pas encore fait ses preuves dans des applications d’entreprise. Par exemple, elle ne prend pas en charge la création d’applications transactionnelles.

1. **PHP et ses frameworks**

Le PHP est un des langages de développement d’applications les plus connus dans le monde du Web dynamique.

1. **Avantages**

* Cette plateforme a été conçue principalement pour les sites d’informations et les sites vitrines, ce qui lui a valu sa réputation.

1. **Inconvénient**

* ​​​​Elle est inadaptée pour les applications de grande envergure telles que celles d’entreprises malgré l’apparition de frameworks plus matures comme Symfony ou Lavarel.

1. **.NET**

Quant à .NET, il a été développé par Microsoft Corporation dans le but de répondre aux exigences des grandes entreprises en termes de développement d’applications.

1. **Avantages**

* C’est une plateforme mature, rapide, performante, maintenable et stable, tout ce dont la réalisation d’une application destinée aux grandes firmes requiert.

1. **Inconvénient**

* Il y a d’abord son coût financier vu qu’il faut payer une licence logicielle pour pouvoir en faire usage.
* Du côté serveur, nous sommes dans l’obligation de nous servir de Windows Server ; c’est-à-dire qu’il est impossible de l’utiliser sans s’appuyer sur les technologies de Microsoft Corporation.

1. **Java et ses frameworks**

Les technologies Java ne sont plus à présenter tellement elles sont omniprésentes. En plus d’être un langage de programmation très connu avec une grande communauté de développeurs, Java, c'est également un ensemble de solutions proposées pour l’élaboration d’applications à but professionnel.

1. **Avantages**

* Il n’a rien à envier à .NET en termes de fiabilité en plus d’être open-source contrairement à son homologue de chez Microsoft Corporation.
* Puisque cette technologie est implantée dans le monde du Web depuis ses débuts, les ressources la concernant abondent. Cela est également dû aux nombreux experts de cette dernière l’utilisant au quotidien. Elle évolue donc sans cesse.
* Poursuivons notre comparaison avec .NET, Java dispose de ce que nous appelons une « machine virtuelle » ce qui lui procure la capacité d’être multiplateforme.

1. **Inconvénient**

* Comme nous l’avons vu, Java est très complet certes, cependant son point faible demeure dans sa complexité. Il est très difficile à appréhender pour un non-initié.

Notre recommandation

Pour le bien du projet, il serait plus judicieux d’opter pour la plateforme Java qui cumule le plus de points forts tout en ayant des points faibles facilement remédiables.

### Système de gestion de base de données (SGBD)

Il est à préciser qu’il s’agit également d’un élément important pour réussir à concevoir une application vu qu’il persiste toutes les informations et se charge d’assurer l’intégrité des données. C’est pour cela que la performance de l’application globale se base surtout sur sa gestion des données, en d’autres termes, de son SGBD. Nous pouvons en considérer 2 grands types :

1. **Les SGBD open-source**

Parmi ces derniers, les plus référencés et appréciés sont MySQL et PostgreSQL qui sont des SGBD dits « relationnels ».

1. **Avantages**

* Ils suffisent largement pour des applications assez sophistiquées étant donné que ces SGBD sont complets, du moins pour leur prix.

1. **Inconvénients**

* Nous nous rendons vite compte de leurs limites lors de la conception d'applications plus complexes et avancées. Effectivement, certaines fonctionnalités sont soit complexes à mettre en œuvre soit inexistantes.

1. **Les SGBD payants**

Dans le monde industriel, les plus célèbres pour leur robustesse et leur performance sont : Oracle Database et Microsoft SQL Server. Ils sont également des SGBD « relationnels ».

1. **Avantages**

* Ils sont à la fois stables et très performants.

1. **Inconvénients**

* Aujourd’hui, il n’y a pas grand-chose à leur reprocher à part leur coût d’exploitation.

Notre recommandation

Pour le bien du projet, il serait plus judicieux d’opter pour PostgreSQL Database qui cumule le plus de points forts et pour une bonne performance et aussi une stabilité.

# Infrastructure et sécurité

Cette partie du document rassemble nos recommandations concernant le réseau, le serveur et l'architecture en place.

## Serveur

Pour servir l’application, un laptop avec les configurations minimales suivantes suffit:

* Système d'exploitation : Windows
* RAM : 8Go
* Processeur : Core i5, 8è génération
* Stockage 1To (SSD si possible)

Concernant les améliorations qui peuvent être apportées, nous proposons les suivantes :

* Utilisation du débit de connexion pour éviter l’échec d'envoi de données, l’idéal, c'est 1 Gb/s.
* Utilisation de VPN pour une communication directe et sécurisée entre les sites.
* Utilisation de routeurs pour diffuser le WIFI pour les ordinateurs portables ne pouvant pas être câblés.
* Augmentation du nombre de câbles réseaux.
* **Topologie réseau :** il est recommandé d’isoler le serveur dans un DMZ (demilitarized zone) pour une performance et une sécurité optimale. Cette isolation sera plus déterminante lorsque le nombre de services ou d’utilisateurs augmentent.
* **Segmentation :** il faut également isoler les groupes d’utilisateurs suivant une structure logique de l’organisation (par exemple par Service). Pour ce faire, le réseau local doit être segmenté en plusieurs VLAN. Cela sera bénéfique à la fois en termes de performance et de sécurité.

## Sécurité

La prévention est un des aspects les plus importants de la sécurité informatique. Voici nos recommandations en termes de sécurité:

* **Antivirus:**  Les ordinateurs doivent être équipés de logiciels antivirus ce qui est très important surtout s’ils sont connectés à Internet.
* **Politique de sécurité à jour:** Il faut disposer d’un document qui explique les différents enjeux de la sécurité, les procédures à suivre pour prévenir et résoudre les incidents.
* **Pare-feu :** Il est primordial pour un réseau qui est connecté à Internet de disposer d’un pare-feu. Celui-ci assurera à la fois la sécurité du DMZ (demilitarized zone) et du réseau local. Il existe plusieurs solutions. Pour une solution non payante, nous recommandons pfSense qui est basé sur FreeBSD.

# Modules fonctionnels

L’application web recommandée aura les caractéristiques ci-après :

* l’application intégrera et agrègera toutes les informations provenant de tous les modules disponibles.
* l’accès sera protégé par un nom d'utilisateur (login) et un mot de passe,
* elle sera composée de plusieurs modules qui comportera plusieurs fonctionnalités (les détails seront dans la partie qui suit),
* chaque utilisateur aura accès à des fonctionnalités configurées par rapport à leur profil d’appartenance,
* l’application devra être accessible à partir d’un navigateur web et adoptera la technique de “responsive design” garantissant l’affichage sur tous les écrans mobile ou desktop,
* D’ailleurs, elle sera disponible dans tous les sites dès que le réseau local sera présent,
* Après la mise en place des réseaux privés virtuels, l’application pourra être accédée via Internet, offrant notamment aux dirigeants de pouvoir d’avoir accès à toutes les informations qu’ils souhaitent obtenir,
* La page d’accueil de l’application devra mettre en exergue les différents modules disponibles en considérant les autorisations de l’utilisateur connecté,
* il y aura un système de notification qui sera visible dans la page d’accueil. L’alerte pouvant venir de différents modules (ex demande validée, etc…).

## Gestion de croyants

Ce module permet la création et la gestion des dossiers des croyants de l'église, incluant leurs informations personnelles.

* CRUD paramétrable selon les églises.
* Gestion des dossiers des croyants par secteur.
* Gestion transfert entre église.

## Gestion de “Mpandray”

Ce module est dédié aux “Mpandray” et permet la gestion complète de ces personnes.

* Devenir “Mpandray”.
* Nouveau “Mpandray” provenant d'une autre église .
* Payer les cotisations par mois de chaque “Mpandray”.
* Note les réunions entre “Mpandray”.

## Gestion des Immobiliers de l’église

Ce module est dédié aux matériels liés au fonctionnement de l'église.

* Il offre des fonctionnalités pour gérer les immobiliers : bancs, , sonorisation, ... .
* Permet de lister les immobiliers afin de voir leur état dans le but de rénovation et l'innovation des matériels.

## Gestion des ressources et documents

Ce module est dédié aux numérisations de tous les documents.

* Bibliothèque de médias (photos, vidéos, anciens documents) des événements passés ou des actes.
* Classement automatique ou manuel des documents.
* Recherche par mots-clés, tags, noms de fichiers ou métadonnées, dates.
* Accès rapide aux documents fréquemment consultés par l'utilisateur actuel.
* Notification des mises à jour des documents (Suppression , Ajout ).
* Destruction automatique des documents selon les délais réglementaires (À voir selon la règle de gestion)

## Gestion des événements

Ce module est dédié aux événements et aux programmes de l'église tous les mois et chaque année .

* Il permet de gérer tous les événements et de les réorganiser des calendriers.
* Il offre des historiques et des rapports pour chaque événement.
* Gestion des comités d’organisation de chaque événement.
* Gestion des rôles sur les créations et la validation d’un événement.
* Notifications et rappels pour un événement à venir.
* Gestion du déroulement d’une séance d’un événement

## Gestion de stock des biens Consommables

Ce module est dédié à la gestion des stocks. Chaque ligne de stock sera statuée / flaguée selon les termes utilisés. Ceci permet aux utilisateurs du module de suivre à tout moment :

* La gestion des mouvements de stock : entrée / sortie. (Le vin, pain de communion, matériels de l'église, ...).
* L’état de stock à l’instant T.
* L’inventaire et ajustement de stock.
* Permet de régénérer des commandes pour les fournisseurs en cas de rupture

## Gestion de facturation

Ce module permet de créer des factures fournisseurs, de suivre leur état en temps réel et de les payer.

* Gérer les factures les fournisseurs

## Gestion de caisse

Ce module permet de suivre et de gérer les transactions de caisse au sein de l'église. Il inclut l'enregistrement des entrées et sorties d'argent, la gestion des fonds de caisse, et la possibilité de générer des rapports détaillés sur l'utilisation des liquidités.

* Gestion des finances liés aux activités de l’église.
* Gestion des rôles pour les créations et les validations lors des transactions.
* Régénérations des balances sur la gestion de caisse.
* statistique et analyse sur la finance.
* Création de rapports sur les finances. ( mail automatique ).
* Ligne Crédits.

## Gestion du bureau de l’eglise

Ce module permet de gérer les activités rattachées au bureau de l'église et chaque membre du bureau.

* Gestions des accessibilités à l'application par les utilisateurs

## Gestion des “Diakona”

Ce module permet de gérer les activités des “Diakona” dans l’église.

* Gestion des “Diakona” par secteur.
* Gestion des “Loholona”.
* Export en documents des activités.

## Gestion des organisations au sein de l’église

Ce module permet de gérer les activités de chaque organisation présents dans l’église.

* gestion des membres de leur bureau respectif.
* Proposition des événements par les organisations.

## Tableau de bord

Ce module permet de gérer et de consulter le tableau de bord selon le profil de l’utilisateur. Et aide à la décision.

* Analyse et statistiques des “Mpandray” les plus actifs par secteur par date.
* Analyse et statistiques des apports par rapport aux événements .
* Analyse et statistiques des dépenses par date
* Analyse sur toutes les sources de revenus et la productivité.

# Spécifications détaillées des modules

Avant d’illustrer les différents spécifications de chaque module, les règles générales suivantes sont à appliquer selon besoin :

* L’affichage des informations résultats de la recherche multicritère est contrôlée par le niveau d’accès du compte.
* Des champs libres, dynamiques et paramétrables sont mis à disposition des gestionnaires de cette application pour des informations spécifiques. La mise en place de ces champs devrait être en accord avec les deux parties .
* Les différentes dates de création sont automatiques. leur modification dépend de la règle imposée au moment de l'exécution de l’ouvrage.
* Un système de notification accompagne chaque module. Cette notification est déclenchée à chaque nouvelle enregistrement, modification et suppression.

## Paramétrage

Dans l’application intégrée du système, il y a des informations qui ne devraient pas être figées. On aura les fonctionnalités classées comme standard (CRUD) :

* Create : créer
* Read / retrieve : lister et/ou rechercher
* Update : modifier ou mettre à jour
* Delete : supprimer

On retrouvera alors les mêmes champs qui sont concernés que l’on pourra paramétrer.

# Conclusion

La réussite d'un tel projet engendrera des bénéfices incontestables. En créant un système intégré, nous pouvons garantir une rapidité de traitement uniforme dans tous les services, éliminant ainsi les silos d'information et améliorant considérablement les relations avec les patients.

Parallèlement, ce système intégré permet également de réduire considérablement les risques de perte de données. Grâce à une gestion centralisée et sécurisée des informations, nous pouvons garantir l'intégrité des données et minimiser les vulnérabilités potentielles.

Il est destiné à servir de fondement solide lors de l'implémentation des différents modules. Les recommandations formulées au niveau de l'architecture et des réseaux visent à assurer la disponibilité continue du système et sa capacité à évoluer avec les besoins futurs de l'organisation.