

Audit projet Infoger :

1. Collecte d'informations générales

a. Sources documentaires et codes :

1 / Les documents obtenues :

- Un dossier « documentation » qui comporte les spécifications fonctionnelles, un schéma de l'infrastructure, le wireframe sous format d'image, le guide d'utilisation du back office.
- Un dossier « database » comprenant différentes bases de données nommé « infoger » avec ces différentes versions.
- Un dossier « client » comprenant des fichiers de codes (php,html,css) :
- Un dossier « library » comprenant des fichiers de codes s'occupent de la gestion de la base de données (php) ;
- Un dossier « service » comprenant des fichiers de codes (3 fichier php et 1 css)
- Un dossier « service_parametre » comprenant des fichiers de codes (2 fichier php et 1 css)
- Un fichier css « connexion»
- Un fichier texte nommé README.

2 / Analyse des différents dossiers et documents :

➤ Dossier documentation :

• Spécifications de l'infrastructure :

Le fichier texte explique les différents éléments présents dans le projet ainsi que l'infrastructure en elle même de ce projet. Il cite les éléments qui sont présents dans chaque dossier du projet et donne une légère explication de celui-ci. Il donne aussi les moyens d'accès aux différents serveur et outils à dispositions (serveur DATABASE, Web...).

• Guide d'utilisations BO :

Le fichier texte comprend plusieurs screenshots du service d'utilisation du back office avec l'explication tableau de bord, liste clients, services clients. C'est un pas à pas de l'utilisation de celui-ci. Malheureusement le document est parfois peut claire du à des fautes d'orthographe et mauvaise tournure de phrase.

- infrastructure slam:

Fichier JPG comprenant le schéma d'infrastructure réseau du projet avec la spécifications des différents serveurs avec leurs adresses IP, les différents matériels utilisés (firewal, switchs,serveurs...) avec les différents vlan présents.

- maquette wireframe:

fichier JPG comprenant les liens entre les différentes pages du projet avec des flèches pour indiquer les différents liens .

- Spécifications fonctionnelles :

Un fichier texte comportant les actions que peut effectuer le programme en lui même.

- infofer,loo:

Un fichier Looping qui est un schéma conceptuel de la base de données du projet avec ces différentes tables, liens et cardinalité.

➤ **Dossier database:**

. Le dossier comprend les différentes versions de la base de donnée du projet.

➤ **Dossier client:**

. Le dossier comprend 13 fichiers dont 5 fichiers Php, 5 fichiers CSS et 3 fichiers Html . Il comprenne des lignes de odes qui ont tous des fonctionnalités différentes, modification d'un client, ajout d'un client...

➤ **Dossier library:**

Le dossier comprend 3 fichiers PHP qui on chacun des fonctionnalités différentes concernant la base de donnée comme un test de connexion à la BDD...

➤ **Dossier service:**

Le dossier comprend 3 fichiers PHP et 1 fichier CSS qui on chacun des fonctionnalités différentes concernant les différentes actions de l'option service du projet

➤ **Dossier service parametre:**

Le dossier comprend 2 fichiers PHP et 1 fichier CSS qui on chacun des fonctionnalités différentes concernant la modification des paramètres de l'option service du projet

➤ **Fichier connexion.css:**

Fichier css qui complémente la méthode de connexion.php présent dans le dossier client.

➤ **Fichier README.md:**

Fichier qui explique ce qu'est ce projet : « Back Office Web du projet de gestion des services d'INFOGER »

b. Objectif du projet :

1 / Identifiez la finalité de l'application:

C'est un Back Office Web de gestion du projet InfoGer qui comprend différents services comme l'accès au VirtualHOST au Dns, CloudComputing. Il permet d'administrer au client des services ou non. Ces clients sont des entreprises.

c. Évaluation de l'environnement technique

1 / Identifiez les versions de langages, serveurs, services, utilisées :

- Schéma conceptuel : utilisation de l'outil LOOPING
- Serveurs : DATABASE(pas d'informations sur le serveur utilisé), Serveurs applicatifs (pas d'informations sur le serveur utilisé), WEB (pas d'info), EyesOfNetwork(pas d'info), GLPI(pas d'info), DHCP (pas d'informations), AD (pas d'informations), PROXY(pas d'informations), IDS/IPS (pas d'informations)
- Langages : HTML, CSS ,PHP (pas d'informations)
- SGBD : MYSQL

2 / Examinez les serveurs utilisés (Apache, Nginx, etc.) et les accès.

- Serveurs DATABASE (IP ADDRESS : 192.168.100.210/24), (Connexion(id : dev_web/ password :123+aze))
- Serveurs applicatifs (pas d'informations)
- WEB (IP ADDRESS : 192.168.10.201/24), (Connexion(id : srv/ password :123+aze))
- EyesOfNetwork (IP ADDRESS : 192.168.100.210/24),
- GLPI (IP ADDRESS : 192.168.10.202/24),
- DHCP (IP ADDRESS : 192.168.100.254/24),
- AD (IP ADDRESS : 192.168.100.200/24),
- PROXY(IP ADDRESS : 192.168.10.254/24),
- IDS/IPS (IP ADDRESS : 192.168.5.254/24)

3 / Vérifiez les versions des bibliothèques externes ou frameworks utilisés (si applicables).

Aucun

4 / Identifiez les applicatifs/logiciels services nécessaires

- Looping
- IDE (VS code, phpstorm, ...)
- SGBDR (MYSQL, phpmyadmin...)

5 / Regardez la structure de la base de données et ses accès (si il y a lieu).

L'accès à la base de donnée s'effectue grâce à des codes d'accès :

- Un accès admin : |id : infoger_admin, |password : 123+aze\$
- Un accès utilisateur : |id : infoger_user1, |password : 123+aze\$

La base de donnée utilisée est une base de donnée relationnel avec pour chaque table une clé primaire et sur certaines tables, des clés secondaires implémentés. Des principes de cardinalité sont présent pour assurer l'association des clés ce qui créer des nouvelles tables.

d/ Exécution du projet :

1/ Testez l'application pour comprendre ses fonctionnalités principales.

Sur le tableau de bord du back office l'option accès client et fonctionnelle et l'option services n'est pas fonctionnelle.

Sur la page liste des clients on a bien l'indication de tous les clients disponibles avec leurs adresses... . L'option de détails des informations et fonctionnelle et celle des services aussi. Sur cette page une option de retour au tableau de bord et disponible et fonctionnelle.

Sur la page informations, les informations du client sont bien indiquées et l'option de modification est bien fonctionnelle. Ainsi lors de la modification les éléments sont bien modifiés dans la base de données lorsque que l'on clique sur le bouton enregistrer. Sur cette page une option de retour au tableau de bord et disponible et fonctionnelle ainsi qu'une option de retour à la liste client elle-même fonctionnelle.

Sur la page services du client sélectionné les services VirtualHost permet à un client d'obtenir un serveur virtuel http pour lui sur le serveur physique d'Infogier , l'option d'activation et de désactivation est opérationnelle et fonctionne, aussi sur cette page une option de retour au tableau de bord et disponible et fonctionnelle ainsi qu'une option de retour à la liste client elle-même fonctionnelle.

L'option de paramètres du Virtual HOST peut indiquer le chemin d'accès et le nom VirtualHost. Sur cette page une option de retour au tableau de bord et disponible et fonctionnelle ainsi qu'une option de retour aux services client elle-même fonctionnelle.

Le service DNS permet d'attribuer une adresse IP au nom de domaine que le client possède, celui-ci n'est pas disponible ainsi que ces options.

Le service Cloud computing permet de stocker les données des sites sur un serveur distant celui-ci n'est pas disponible ainsi que ces options.

2. Analyse de l'architecture et du code :

a. Structure des fichiers sources

1/ Étudiez l'organisation du projet :

➤ Fonctionnalités du projet(Back office) :

1/ Library :

- `bdd_manager` : c'est une classe nommée « BDDManager » comporte une méthode de connexion à la base de données, une méthode de vérification de connexion (boolean true/false), une méthode pour exécuter une requête SQL sur la base de donnée, une méthode pour pouvoir insérer des données dans la base de donnée, une méthode pour pouvoir modifier des données dans la base de donnée.
- `bdd_infoger.php` : c'est une classe nommée « BDD_infoger » qui hérite des méthodes et membres de la classe « BDDManager », on y retrouve toute les requêtes SQL qui interagiront avec la base de donnée.
- `test_connexion.php` : c'est un fichier comportant un en tête html avec une balise php dans laquelle on insère les méthodes de la classe « BDDManager ». Elle permet donc d'effectuer un test de connexion à la base de donnée.

2/ Tableau de bord :

Pour que cette page il faut ces fichiers : `tableau_bord.html`, `tableau_bord.css`,

- Bouton Accès Client (fonctionnelle) : cette fonctionnalité redirige vers la page de « Liste des Clients ».
- Bouton Accès Service (non fonctionnelle) :

3/ Liste des Clients :

Pour cette page il faut les fichiers `lister_client.php`, `liste_client.css`, `afficher_client.php`,

- Bouton Ajout d'un client (fonctionnelle) : pour cette fonctionnalité il faut ces fichiers : `nouveau_client.html`, `nouveau_client.css`, `creer_client.php`
- Bouton Informations (fonctionnelle) : cette fonctionnalité redirige vers la page « Informations Clients »
- Boutons Services (fonctionnelle) : cette fonctionnalité redirige vers la page « Liste des services d'un utilisateur »

4/ Informations Clients :

Pour cette page il faut les fichiers client_x.html, client_x.css

- Bouton Modifier client (fonctionnelle) : cette fonctionnalité redirige vers la page «Modification »

5/ Modification :

Pour cette page il faut les fichiers modifier_client.php, modifier_client.css

- Bouton Enregistrer (fonctionnelle) :

6/ Liste des services d'un utilisateur:

Pour cette page il faut les fichiers service_client.php, service_client.css

- Bouton Activer/Désactiver Virtual Host (fonctionnelle): pour cette fonctionnalité il faut ces fichiers : changer_statut.php,
- Bouton Activer/Désactiver DNS (non-fonctionnelle):
- Bouton Activer/Désactiver Cloud Computing (non-fonctionnelle):
- Bouton Paramètres Virtual HOST(fonctionnelle) : cette fonctionnalité redirige vers la page « Paramètres du service »

7/ Paramètre du service DNS:

Pour cette page il faut les fichiers parametre_service.php, parametre_service.css

- Bouton Enregistrer (fonctionnelle): pour cette fonctionnalités il faut le fichier : modifier_parametre.php

8/ Fonctionnalité commune à toutes les pages :

- Bouton retour tableau de bord (fonctionnelle): cette fonctionnalités redirige vers la page « Tableau de Bord »
- Bouton retour a la liste des clients (fonctionnelle): cette fonctionnalités redirige vers la page « Liste des clients »

b. Analyse du code source

1. Audit général :

- Point d'entrée : tableau_bord.html
- Convention de nommage :
 - les variables sont notées de cette manière «motvariable_motvariable »
 - Chaque méthodes possède une indication d'actions
 - Des commentaire sont positionnée aux endroits utiles
- Aucun framework n'est utiliser pour ce projet, ni de bibliothèques ou de fonctions natives

2. Recherche des modèles de conception :

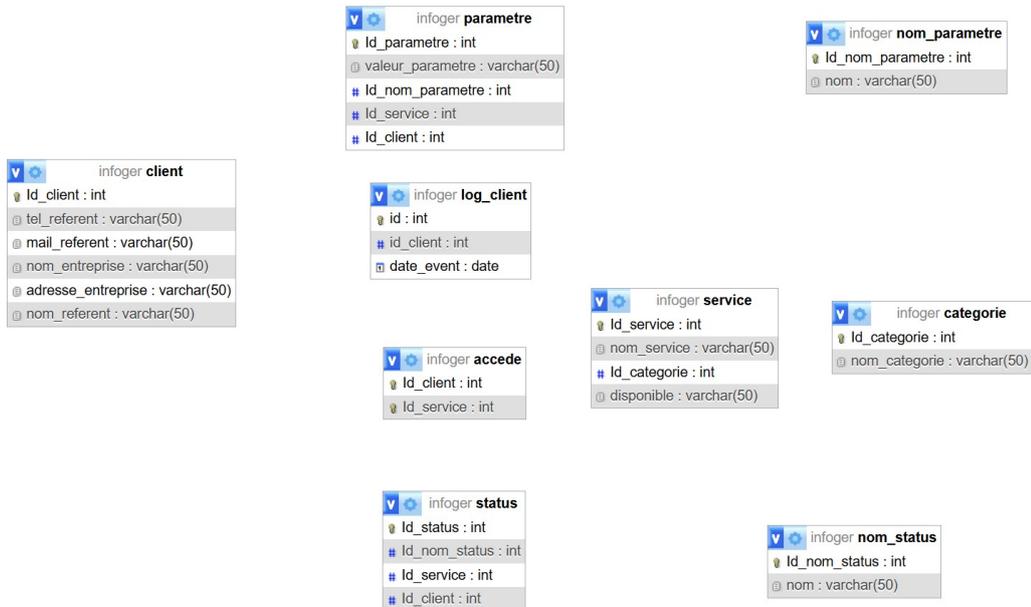
La plupart des méthodes de l'application sont dirigés vers des classes qui gère la base de données. En effet ce n'est pas un contrôleur qui gère la direction mais les méthodes en elles mêmes.

3. Analyse de la sécurité :

Sur ce projet les injections sql sont possibles et il est même possible d'accéder directement à la base de donnée via les identifiants qui sont inscrit en dur sur le code.

3. Analyse de la base de donnée :

➤ Structure et relations :



Le schéma correspond avec la base existante, cependant on ne voit pas les relations entre les tables. Toutes les tables sont liées à la table « client ».

Les autres relations :

- La table « nom_parametre » possède un lien avec la table « parametre »
- La table « service » possède un lien avec la table « parametre », la table « accede », la table « status », la table « categorie »
- La table « nom_status » possède une relation avec la table « status »

➤ Qualité des requêtes SQL

Les requêtes présentes dans le dossier *bdd_infoger* sont peu complexes et facilement déchiffrables. Cela permet donc de ce repérer facilement lors de la lecture de ces requêtes.

4. Analyse des dépendances externes :

- Je ne vois aucune bibliothèques utilisés seulement la dépendance aux autres classe du projet

5. Tests et validation :

➤ Tests manuels :

- La plupart des fonctionnalités du projet fonctionne, seulement la liste des services dans la page « Tableau de Bord » et l'activation de certains services dans la page de contrôle des services pour un utilisateur (activation DNS et CloudComputing)

➤ Logs et erreurs :

La table de logs de la base de données infoger présente 3 colonnes(*id, id_client, date_event*). On remarque que la plupart des connexions ont été effectués entre l'année 2023 et 2024, et qu'aucune erreurs n'en est remonté.

7. Documentation et restitution de l'audit :

➤ Lister et identifier :

Documents :

- Un dossier « documentation » qui comporte les spécifications fonctionnelles, un schéma de l'infrastructure, le wireframe sous format d'image, le guide d'utilisation du back office.
- Un dossier « database » comprenant différentes bases de données nommé « infoger » avec ces différentes versions.
- Un dossier « client » comprenant des fichiers de codes (php,html,css) :
- Un dossier « library » comprenant des fichiers de codes s'occupent de la gestion de la base

- de données (php) ;
- Un dossier « service » comprenant des fichiers de codes (3 fichier php et 1 css)
 - Un dossier « service_parametre » comprenant des fichiers de codes (2 fichier php et 1 css)
 - Un fichier css « connexion»
 - Un fichier texte nommé README.

Services :

- Looping
- IDE (VS code, phpstorm, ...)
- SGBDR (MYSQL, phpmyadmin...)

➤ **Rapport de l'analyse du code :**

➤ **Forces et faiblesses du projet :**

Le projet est construite d'une manière simple et facilement compréhensible (code). Des erreurs sont présentes comme l'inscription des données de connexion directement dans le code, l'absence de sécurité dans les requêtes SQL. Une fonction n'est pas assez visible « l'ajout d'un client ».

➤ **Proposez un plan d'amélioration :**

- Option de modification de suppression et d'ajout d'un client dans la même page.
- Option de modification de suppression et d'ajout d'un service dans la même page.
- Option de modification de suppression et d'ajout d'un paramètre dans la même page.
- Interface de connexion qui nous permettra d'accéder au service.
- Barre de navigation en haut de page pour naviguer dans les pages.
- Nouvelle interface d'activation et de désactivation de service pour un client.
- Option d'association de paramètres à un service.