

I. ARCHITECTURE DE NOTRE LOGICIEL

I.1. Définition

L'architecture d'un logiciel est un plan qui décrit la structure globale d'un système logiciel. Elle définit les différents composants du logiciel, leurs interrelations et leurs interactions. L'architecture logicielle est comme un plan d'architecte pour un bâtiment : elle fournit une vue d'ensemble de la structure du logiciel et de la manière dont ses différentes parties s'articulent. Les avantages qui résident dans la conception de l'architecture d'un système sont que :

- **Elle permet de concevoir des logiciels plus fiables, plus robustes et plus évolutifs.** Une architecture bien conçue permet de répartir les tâches entre différents composants, ce qui rend le logiciel plus modulaire et plus facile à maintenir. De plus, une bonne architecture permet d'anticiper les besoins futurs et de faire évoluer le logiciel plus facilement.
- **Elle facilite la communication entre les développeurs.** L'architecture logicielle sert de référence commune à tous les développeurs qui travaillent sur le projet. Cela permet de réduire les malentendus et de garantir que tous les composants du logiciel sont compatibles entre eux.
- **Elle permet de prendre de meilleures décisions en matière de technologie.** L'architecture logicielle permet d'identifier les technologies qui seront nécessaires pour développer le logiciel. Cela permet de choisir les technologies les plus appropriées et de s'assurer qu'elles sont compatibles entre elles.

I.2. Les différents types d'architecture logicielle

Il existe différents types d'architecture logicielle, chacun ayant ses propres avantages et inconvénients. Parmi les types d'architecture les plus courants, on trouve :

- **L'architecture client-serveur** : dans une architecture client-serveur, le logiciel est divisé en deux parties : un client et un serveur. Le client est l'interface utilisateur du logiciel, tandis que le serveur est responsable de l'exécution des tâches principales du logiciel.
- **L'architecture à trois couches** : dans une architecture à trois couches, le logiciel est divisé en trois couches : la couche présentation, la couche métier et la couche d'accès aux données. La couche présentation est responsable de l'affichage des données à l'utilisateur, la couche métier est responsable de l'exécution des règles métier du logiciel et la couche d'accès aux données est responsable de l'accès aux données.
- **L'architecture SOA (Service Oriented Architecture)** : dans une architecture SOA, le logiciel est composé de services autonomes qui communiquent entre eux via des messages. Les services SOA sont généralement basés sur des technologies Web, telles que SOAP et REST.
- **L'architecture micro-services** : une architecture micro-services est une variante de l'architecture SOA. Dans une architecture micro-services, le logiciel est divisé en petits services indépendants qui sont développés et déployés indépendamment les uns des autres.

I.3. Architecture de notre plateforme

Pour la conception et la réalisation de notre plateforme de free-lance pour photographe, nous allons adopter une architecture **client-serveur**. Cette architecture fonctionnera de la manière suivante :

- **Demande du client** : les clients sont des programmes exécutés sur les ordinateurs des utilisateurs finaux (Photographes, Clients, Administrateurs). Un client envoie une demande à un serveur pour obtenir un service ou des données. La demande peut être envoyée via un réseau local ou Internet.
- **Traitement de la demande par le serveur** : Le serveur reçoit la demande du client et la traite. Cela peut impliquer l'accès à une base de données, l'exécution de calculs complexes ou la récupération de fichiers.
- **Réponse du serveur** : Le serveur envoie une réponse au client. La réponse peut inclure les données demandées, le résultat d'un calcul ou un message d'erreur.
- **Réception de la réponse par le client** : Le client reçoit la réponse du serveur et l'affiche à l'utilisateur.

Cette architecture peut être représentée de la manière suivante :



