

# Chapitre 3

---

## Etude technique

Ce chapitre sera consacré à la partie technique du projet, dans laquelle nous allons présenter tous les outils techniques utilisés pour la mise en œuvre de l'application et les justifications de chaque choix technique.

## 1. Technologie et outils de mise en œuvre

### 1.1 Choix des langages et frameworks

L'application est développée en PHP à l'aide du framework Symfony. Tout d'abord car ce sont les technologies utilisées par Hiit Consulting pour le Back-end.

#### 1.1.1 Partie Back-end

##### a. PHP

Tout d'abord, PHP est un langage gratuit qui ne nécessite aucune licence. PHP est le langage de programmation le plus utilisé dans le web. La communauté de développeurs est très active qui rend disponibles des dizaines de milliers de bibliothèques PHP de grande qualité ainsi qu'une vaste quantité de documentation et tutoriels pour le bénéfice de chacun. Ces ressources facilitent le travail et réduisent notre temps d'exécution.

##### b. Symfony Symfony

Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

#### Avantages

- Grâce à symfony on peut facilement intégrer un projet développé à partir du framework, contrairement à un projet développé en PHP natif, où il n'y a pas de normes ni règles imposées.
- Les fichiers doivent respecter une syntaxe particulière et doivent se trouver au bon endroit dans l'arborescence du projet. Cela garantit une facilité de maintenance sur le long terme, on sais rapidement dans quel fichier il faut aller pour apporter des modifications.
- L'architecture MVC (Modèle Vue Contrôleur) permet de découper le code représentant la logique métier de l'application et le code de présentation des vues.
- Favorise la réutilisation de code, et le respect des recommandations PHP-FIG (Des recommandations mondiales pour une meilleure interopérabilité entre les projets web PHP). Symfony permet donc de produire du code de qualité.

- Symfony intègre des mesures de sécurité préventives pour lutter contre les failles et attaques XSS, CSRF et injection SQL. Symfony embarque systématiquement ces mécanismes de sécurité, sans avoir à les implémenter à chaque fois.

### **inconvénients**

- Un des principal inconvénients d'un framework est le temps d'apprentissage qui est bien plus élevé que pour un langage en lui-même.
- Symfony, plus particulièrement nécessite d'apprendre plusieurs autres technologies qui lui sont propres comme YAML ou Doctrine.
- Symfony est un framework assez lourd et n'est pas idéal pour les petits projets.

### **ORM (doctrine)**



Mapping objet-relationnel (object-relational mapping en anglais), se charge de gérer l'exploitation de la base de données et la persistance des données. Il offre une couche d'abstraction qui réalise la traduction des données extraites de la base de données vers un objet propre au langage de programmation. Grace à l'ORM on travaille ainsi uniquement avec des objets sans se soucier du stockage sous-jacent des données.

Doctrine est l'ORM php le plus stable est le plus utilisé, dans l'application on va utilisé Doctrine.

### **Moteur de template (TWIG)**



Un moteur de Template est une technique de programmation qui permet de séparer distinctement l'interface graphique du reste de votre application. Il permettra d'avoir des fichiers purement html/css/javascript coté front, et un fichier purement php coté back. Dans l'application on va utilisé twig.

#### **1.1.2. Partie Front-end**

Pour la partie front end j'ai utilisé html, css, bootstrap et le javascript natif. Malgré que l'entreprise utilise l'AngularJS comme framework pour le front, je ne l'ai pas utilisé car j'ai intégré l'entreprise en tant que stagiaire développeur back-end, et ce framework nécessite ,comme symfony, un temps important pour l'apprentissage et l'adaptation.

En contre partie, j'ai utilisé le jquery qui fournit des fonctionnalités javascript compatible avec les différents navigateurs.

Ensuite, afin de faire communiquer le client et le serveur sans avoir besoin d'actualiser la page, on utilise l'AJAX et pour cela, l'application doit fournir une API REST qui va générer des réponses HTTP JSON.

## **AJAX**

(Asynchronous JavaScript and XML) , Utilisé pour envoyer et recevoir des données depuis un serveur , pour ne plus avoir besoin de recharger les pages après validation d'un formulaire ou autre interaction avec le serveur.

## **JSON**

(JavaScript Object Notation) est un format des données textuelles dérivé de la notation des objets du langage JavaScript.

## **1.2 Outils de Gestion du code, de conception et de planification**

### **1.2.1 Gestion de version**

Après le lancement de la partie de la réalisation, le code a commencé de plus en plus à se compliquer, et c'était nécessaire de sauvegarder un historique de toute mise-à-jour et modifications dans le code, et donc se servir d'un VCS.

Un VCS ou logiciel de gestion de version «Version Control System» est un logiciel qui permet conserver la chronologie de toutes les modifications qui ont été effectuées dans un code source.

#### **a. git**

Git est un système Open-Source gratuit de distribution de versions. Il est utilisé pour contrôler les version des fichiers. Vu que git est celui qui est utilisé par l'entreprise, il est très stable, et le plus utilisé, mon choix s'est directement orienté vers lui

#### **b. github**

GitHub est un dépôt distant de répertoires placés sous le contrôle de Git. Tout le code source de l'application est disponible sur le site, et accessible à partir du lien : <https://github.com/amine-hiit/HiitcoIntranet/>

### **1.2.2. GanttProject**

GanttProject est un logiciel libre de gestion de projet écrit en Java, ce qui permet de l'utiliser sur de nombreux OS tel que Windows, Linux, MacOS. Ce projet a été lancé par un étudiant de l'université

de Marne La Vallée en Janvier 2003 et est maintenant proposé sous licence libre (GNU GPL). Le chef de projet initial est Alexandre Thomas, relayé aujourd'hui par DmitryBarashev.

### 1.3.3. StarUML



StarUML : est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL. StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0.

## 2. Architecture technique

L'application permet la centralisation de flux d'information dans l'entreprise, pour cela une solution basée sur l'architecture client-serveur est la plus adaptée pour notre application.

### 2.1. Le client:

Dans l'application le client devra être léger et facile à déployer, il sera donc un navigateur Web à travers lequel l'utilisateur peut accéder à notre application.

### 2.2. Serveur web (apache2)



C'est un serveur HTTP qui permet à un serveur de communiquer avec un navigateur en utilisant le protocole HTTP(S). Apache est le serveur HTTP le plus populaire. C'est donc lui qui met à disposition la plupart des sites internet du WWW.

### 2.3. Serveur de donnée (MySQL)

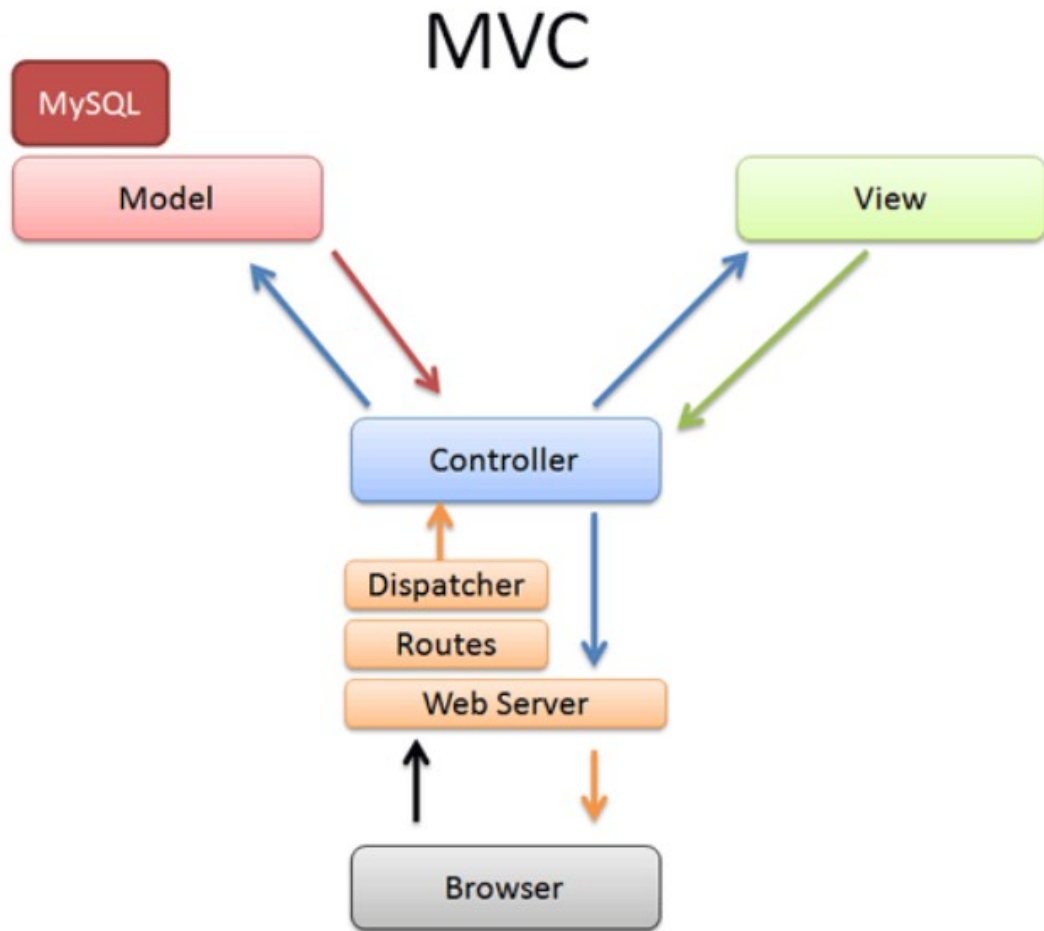


Fournit au serveur web les données requises pour répondre au client. Dans le Projet on a utilisé MySQL comme serveur de données, car il est open source, le plus adapté pour les bases de données relativement petites et il est assez complet au niveau de fonctionnalités.

## 3. Architecture logicielle:

L'architecture logique est la manière dont les composantes logiques d'une solution sont organisées et intégrées. Il est primordiale à la conception de tout système informatique de choisir le modèle d'architecture qui lui sera adéquat pouvant assurer un bon fonctionnement, des meilleurs performances ainsi que la réutilisation et l'interconnexion fiable de ce système avec d'autres.

C'est à cet effet qu'on opte pour le modèle MVC qui sera également très pratique pour gérer l'interaction entre les différents composants de l'application.



*Figure 1: Schéma de l'architecture MVC*

L'architecture MVC (modèle, vue et contrôleur) est une architecture à trois couches utilisée pour la programmation client/serveur et d'interface graphique.

C'est un modèle architectural très puissant qui intervient dans la réalisation d'une application. Il tire sa puissance de son concept de base qui est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

**Contrôleur:** son rôle est de générer la réponse à la requête HTTP demandée par notre visiteur. Il est la couche qui se charge d'analyser et de traiter la requête de l'utilisateur. Le contrôleur contient la logique de l'application web, il utilise les autres composants : les modèles et les vues.

**Modèle:** élément qui contient les données ainsi que de la logique en rapport avec les données: validation, lecture et enregistrement. Dans notre application par exemple, le modèle représente les congés, les employés, les emails...

**Vue:** Partie visible d'une interface graphique son rôle est d'afficher les pages. Elle est appelée et rendue par le Contrôleur vers l'utilisateur, et dans notre application, les vues sont générées par **TWIG**.