

# Rapport du stage

## \*Introduction :

\* Le modèle architectural **Modèle-Vue-Contrôleur** (MVC) : un concept fondamental en développement logiciel, notamment pour la création d'interfaces utilisateur .

## \*Composants du MVC :

### 1. \*Modèle (Model) :

- Le Modèle est responsable de la gestion des données et de la logique métier de l'application. - Il encapsule les données de l'application ainsi que les règles pour manipuler données ces. - Le Modèle interagit avec la base de données, les API ou toute autre source de données. - Il inclut souvent des méthodes pour créer, lire, mettre à jour et supprimer des données.

(**Définition API** : L'acronyme "API" signifie "Interface de Programmation d'Application", en anglais "Application Programming Interface". Une API est un ensemble de règles, de protocoles et de définitions qui permet à différentes applications logicielles de communiquer et d'interagir entre elles de manière standardisée.

En d'autres termes, une API fournit un moyen pour différentes applications, services ou systèmes de partager des données et de fonctionner ensemble de manière cohérente et efficace. Elle définit comment les composants logiciels doivent interagir, quels types de requêtes et de réponses sont acceptés, et quels sont les formats de données à utiliser.)

### 2. \*Vue (View) :

- La Vue est responsable de la présentation des données à l'utilisateur. - Elle génère les composants de l'interface utilisateur, tels que les boutons, les formulaires et les mises en page. - La Vue obtient ses données à partir du Modèle et les affiche de manière visuellement attrayante et compréhensible pour l'utilisateur.

### 3. \*Contrôleur (Controller) :

- Le Contrôleur agit comme un intermédiaire entre le Modèle et la Vue. - Il gère les entrées de l'utilisateur et les traduit en actions qui affectent le Modèle ou la Vue.

- Le Contrôleur reçoit les demandes de l'utilisateur, les traite et met à jour le Modèle ou la Vue en conséquence. - Il garantit que les modifications dans un composant n'affectent pas directement les autres.

## \*Fonctionnement du MVC :

### 1. \*Interaction de l'utilisateur :

- L'utilisateur interagit avec l'application via la Vue. Les actions de l'utilisateur telles que les clics, les saisies ou les gestes sont capturées par la Vue.

### 2. \*Action du Contrôleur :

- La Vue transmet l'entrée de l'utilisateur au Contrôleur.
- Le Contrôleur traite l'entrée et détermine l'action appropriée à prendre.

### 3. \*Mise à jour du Modèle :

\*- Le Contrôleur interagit avec le Modèle, qui met à jour les données de l'application en fonction de l'action. - Si nécessaire, le Modèle notifie la Vue des modifications apportées aux données.

### 4. \*Mise à jour de la Vue :

- La Vue récupère les données mises à jour depuis le Modèle et les affiche à l'utilisateur.
- L'utilisateur voit le résultat de son interaction dans la Vue mise à jour.

### \*Avantages de l'utilisation du MVC :

\***Séparation des préoccupations** : Le MVC sépare les préoccupations de l'application en composants distincts, ce qui facilite la compréhension, le développement et la maintenance.

\***Réutilisabilité** : La structure modulaire du MVC permet de réutiliser les composants dans différentes parties de l'application.

\***Développement parallèle** : Différentes équipes peuvent travailler sur différentes composantes simultanément, réduisant ainsi le temps de développement.

\***Flexibilité** : Les modifications apportées à un composant peuvent être effectuées sans affecter les autres, favorisant l'évolutivité et l'adaptabilité.

\***Tests** : Chaque composant peut être testé indépendamment, améliorant la fiabilité globale de l'application .

# MVC

