

## Documentation Technique et Fonctionnelle

### Objectif :

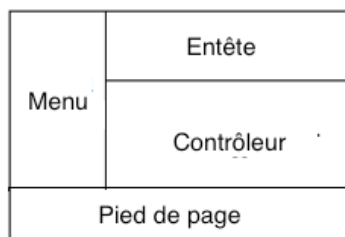
- ✓ interroger une base de données par l'intermédiaire d'une liaison client/serveur http (web)
- ✓ Utiliser un logiciel de gestion de bases de données avec interface graphique (adminer).
- ✓ Découverte de la programmation PHP objet côté serveur
- ✓ interfacier les langages HTML et PHP et réaliser des pages web interactives
- ✓ Utilisation de l'architecture MVC Model View Controller
- ✓ Réaliser des requêtes préparées et paramétrées SQL en langage PHP avec l'API PDO
- ✓ Réaliser un objet DAO (Database Access Object)

### Mise en situation Gestion Agence Immobilière :

Une agence immobilière gère les demandes d'achat d'appartements et de maisons pour le compte de propriétaires. Les clients qui souhaitent acheter un bien, effectuent des demandes ;

L'agence enregistre alors la demande avec son identifiant, son nom, son budget, le type de bien (maison, appartement) et la superficie. Les biens sont dans des localités, dont on indique le nom.

Vous travaillez pour la SSII ailTECH. Vous êtes chargé de créer une interface web permettant d'effectuer différentes requêtes SQL sur la table des demandes.

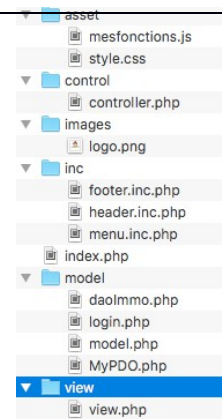
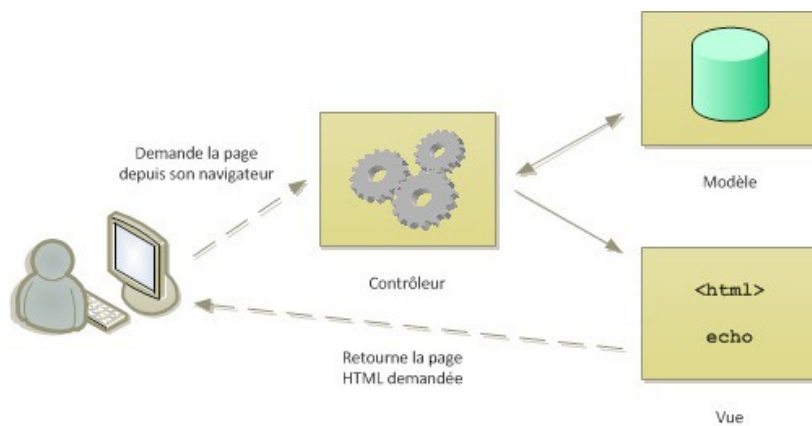


Travail réaliser sur serveur local de développement :

- ✓ J'utilise l'architecture d'application MVC
- ✓ Mon site web est sur le serveur : [oceaned.alwaysdata.net](http://oceaned.alwaysdata.net).
- ✓ La page est divisée en groupes appelés « div »
- ✓ Les différents liens des menus sont traités dans le fichier controller.php.
- ✓ L'affichage est réalisé par les méthodes de la classe Vue incluse dans le fichier view.php
- ✓ L'accès aux données est réalisé par le biais de la classe Model, elle-même composée de la classe DaoImmo (Data Acces Object).
- ✓ C'est dans cette classe DaoImmo qui hérite de la classe PDO que doivent être implémentées les méthodes contenant l'exécution des différentes requêtes SQL.

### L'architecture applicative MVC : Définition

L'architecture [MVC](#), pour Modèle-Vue-Contrôleur, est un modèle de conception (design pattern en anglais). L'architecture MVC est un design pattern utilisé dans le cas de développement d'applications avec interface graphique. Il est très utilisé dans la programmation Web, notamment par certains frameworks PHP (Symfony, Cake PHP, Zend, etc) mais aussi dans de très nombreux langages : C#, Java, Swing, etc..



Le modèle MVC possède de nombreux avantages :

- ✓ il permet de structurer son code de façon claire, logique et efficace,
- ✓ son adoption par de nombreux développeurs tend à uniformiser les codes des sites Web,
- ✓ il facilite la maintenance, le débogage et la réutilisabilité du code,
- ✓ la compréhension de cette architecture MVC facilite l'apprentissage des frameworks.

En contrepartie, le modèle MVC demande un certain temps pour l'apprentissage et nécessite une grande rigueur lors du développement.

## L'architecture applicative MVC : les composants

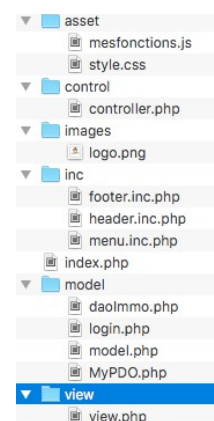
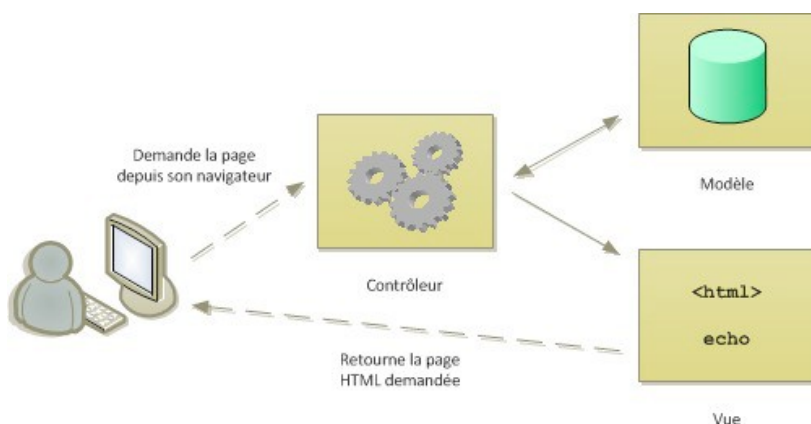
La partie **Modèle** d'une architecture MVC détermine quelles données l'application doit contenir. Si l'état de ces données change, alors le modèle va généralement avertir la vue et parfois le contrôleur.

La partie **Vue** détermine comment les données de l'application doivent être affichées.

La partie **Contrôleur** fait le lien entre le Modèle et la Vue. Concrètement, le contrôleur possède plusieurs actions. Chaque action correspond à l'affichage d'une page Web. Le contrôleur doit :

- ✓ récupère les paramètres nécessaires pour effectuer l'action (dans \$\_GET par exemple),
- ✓ utilise le modèle pour mettre à jour certaines données dans le cas d'ajout, de modification ou de suppression dans la base de données,
- ✓ récupère les données nécessaires pour l'affichage de la page Web grâce au modèle,
- ✓ appelle la vue en lui transmettant les données pour obtenir une page Web\*.

*\* Seule la dernière étape est nécessaire, les 3 premières sont facultatives et dépendent de l'action à réalisée.*



Il existe plusieurs manières d'implémenter un site Web en utilisant l'architecture MVC. Nous allons en voir une.

## UI\* de l'application *(\*User Interface, interface utilisateur)*

Vous devez télécharger le dossier compressé du projet GAI de la SSI ailTECH « ailTECH\_GAI.zip » sur github : <https://github.com/OceaneDamour/Ailtech>.

La table de données « demande » et « personne » doit être présente dans votre base de données. Pour ceux qui l'ont perdu reprendre le fichier demande.sql.

### Travail réalisé

#### 1) Finalisation du fichier « controller.php » et des classes Model et DaoImmo

On teste la valeur de la variable « cas » dans un commutateur ([switch](#)). Cette structure est équivalente à une succession de if() / elseif(). La variable « cas » indique la source de l'événement ici un clic en provenance de la partie HTML.

1.1) Après avoir étudié la liaison entre les fichiers menu.php, header.php et le fichier controller.php afin de comprendre les différents états de la variable « cas », complète la structure switch pour les cas « Personne », « Genre », « Ville » et « Superficie »

1.2) Un clic sur le bouton « ville » transmet l'information cas="villeprecise".

✓ Détermine la méthode appelée dans la classe DaoImmo et complète cette méthode.

1.3) Pour les cas « budgetInferieur » et « budgetGenre » vous devez appeler, pour chaque cas, une méthode de la classe Model qui elle même appelle une méthode de la classe DaoImmo. Le résultat retourné par la classe Model est transmis à la méthode afficheTab de la classe Vue.

✓ Les méthodes des classes Model et DaoImmo sont réaliser.

1.4) réaliser le même travail pour les cas « budget moyen », « budget mini », « budget maxi », « budget > moyenne » et « nombre de biens ».

1.5) Complète la méthode formulaireModif de façon à ce que le formulaire soit rempli avec l'enregistrement à modifier.

✓ Vérifie que cela corresponde au cas= "modifier" et choix= "modifier"

1.6) Gère le cas " VoiciLesModif " afin de modifier l'enregistrement sélectionné.

1.7) Réalise le travail nécessaire pour le cas "modifier".

1.8) Le cas « inserer » provoque l'appel de la méthode formulaireNouveau de la classe Vue.

✓ Réalise cette méthode qui affiche un formulaire vide permettant l'ajout d'un nouvel enregistrement.

✓ Le cas doit être égal à " nouveau ".

1.9) Le cas nouveau doit provoquer l'appel de méthodes à réaliser dans les classes Model, DaoImmo afin de permettre l'insertion de l'enregistrement avec les valeurs contenues dans le formulaire.

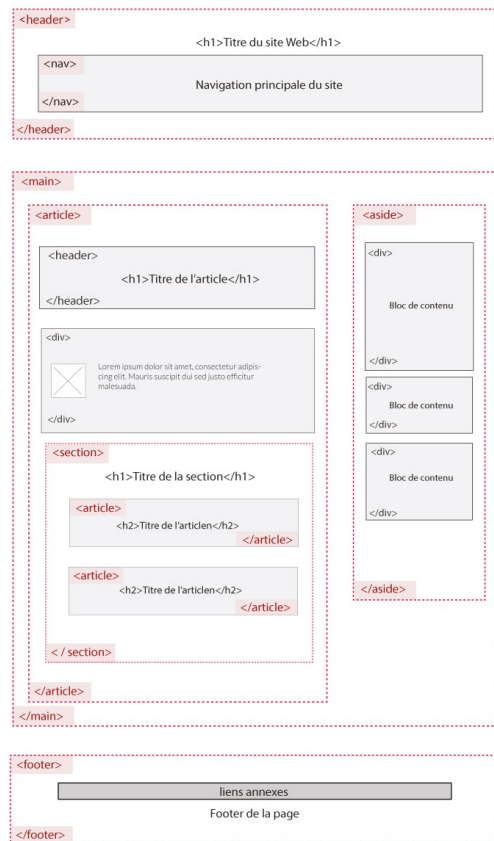
1.10) Les noms des colonnes (première ligne) contenant les noms des champs de l'enregistrement doivent être des liens permettant d'effectuer un tri.

## 2) HTML

2.1) le client souhaite convertir l'interface en web adaptatif (responsive Web Design) en respectant le format HTML 5 .

Réalise en Bootstrap le responsive

### STRUCTURE D'UNE PAGE HTML



© joetheme.com

## 3) Évolution

- ✓ L'entreprise souhaiterait mettre en place un trigger qui insert des lignes dans une table d'historique dès qu'un utilisateur fait des modifications sur la base.
  1. Voir dossier demande.sql