

Peut on créer une application Php Mvc sans utiliser un framework?

Partie suivante

- [Introduction.](#)
- [Le Désign pattern MVC.](#)
- [Tp: Gérations des Stocks.](#)

Introduction

La réponse est bien évidemment oui, les frameworks sont généralement des outils qui facilitent le développement rapide des applications. Un framework est un ensemble de bibliothèques, d'outils et de règles à suivre, et il fournit un ensemble des fonctionnalités. Avec un framework, l'exécution est dirigée par le code du framework qui fait appel au code écrit par le développeur le développeur doit compléter les classes et ressources préexistantes dans le framework.

Exemple des Frameworks

Les frameworks les plus répandus dans le monde php :

- Zendframework.
- Symfony.
- CakePhp.
- ...

Il y'en a d'autres dans d'autre plates forme, comme java :

- Spring MVC.
- Strut 1 et 2.
- Jsf.
- ...

Le but de ce tutoriel est de vous montrer la manière la plus simple de créer une application web se basant sur MVC et sans l'utilisation de framework.

Le Désign pattern MVC

Presque tous les frameworks se basent sur une architecture dite mvc, qui permet la séparation de code de l'application en 3 tiers. Les avantages, une meilleure organisation de fichiers, une structure de l'application claire, ce qui facilite sa maintenance.

MVC

Le mvc est l'abréviation de **M**odèle, **V**ue et **C**ontrôleur

Modèle

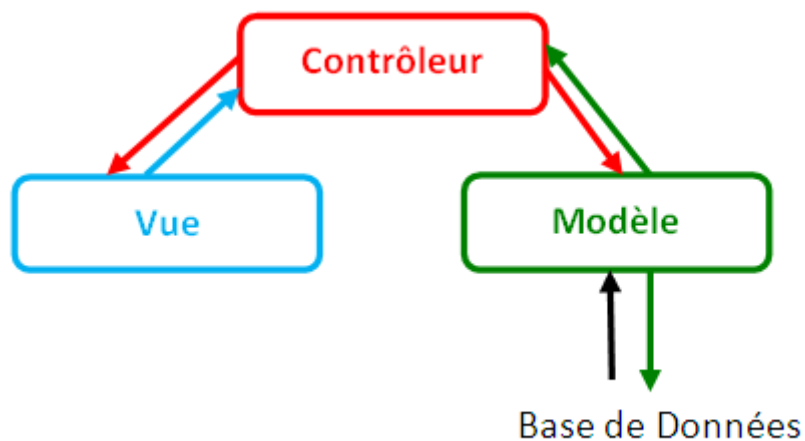
Le modèle permet à l'application d'interagir avec la base de données par l'intermédiaire des objets qui représentent la base de données au niveau de l'application. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données.

Vue

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Cette interface soit elle affiche des données provenant du modèle par l'intermédiaire du contrôleur, soit elle reçoit toutes les actions de l'utilisateur (clic de souris, saisie des informations, boutons, etc). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le contrôleur et d'interagir avec l'utilisateur. Généralement les vues sont conçues pour qu'elles ne contiennent que du HTML et quelque contrôle d'instruction du langage de programmation comme les boucles, les conditions, et les instructions d'affichage.

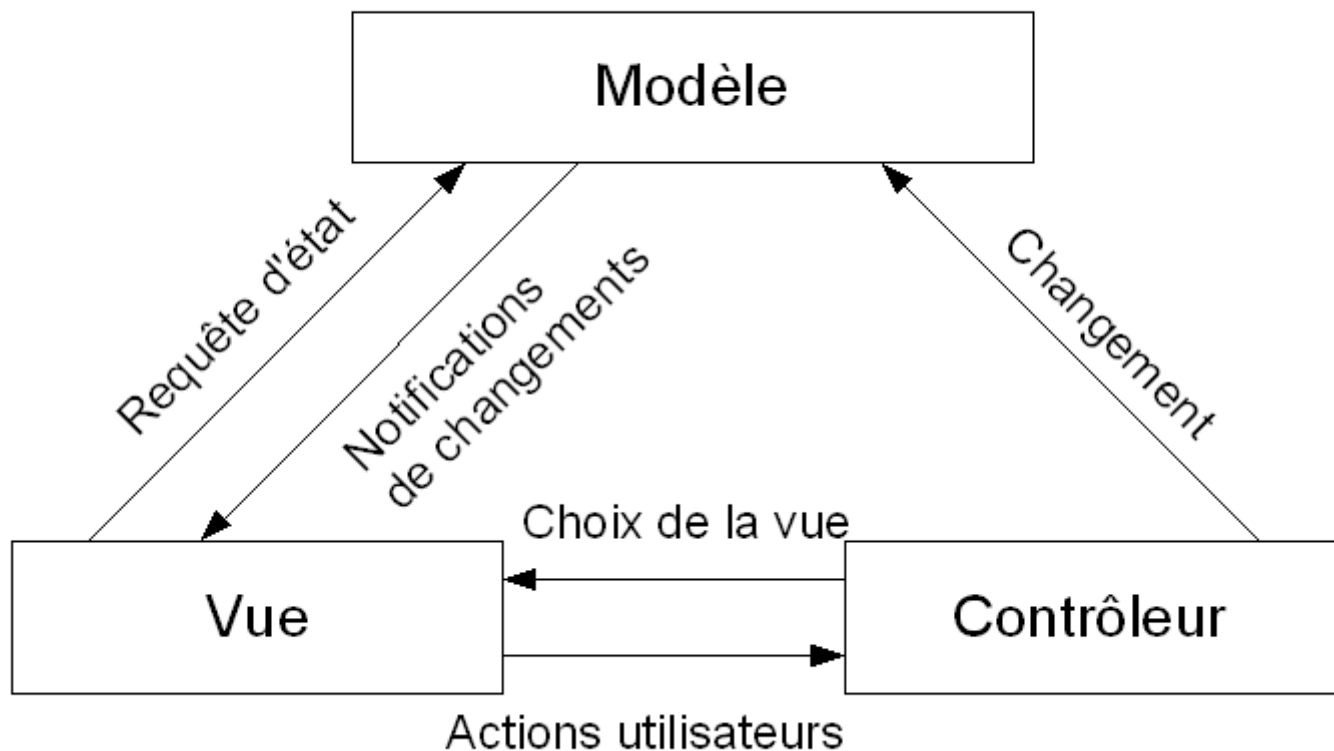
Contrôleur

Le contrôleur présente un cas d'utilisation et contient une ou plusieurs actions, il prend en charge la gestion des événements de synchronisation pour mettre à jour une base de données à travers les modèles. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondante à la demande.



Design pattern MVC.

La présentation qu'on a vu, c'est une approche au design pattern mvc, mais presque tous les cours le présente comme suit;



Ce qui manque à la première présentation, c'est la synchronisation entre la vue et le modèle passée avec le pattern Observer. Il permet de générer des événements lors d'une modification du modèle et d'indiquer à la vue qu'il faut se mettre à jour. Dans notre projet on va ignorer ce cas.

Simulation

L'architecture MVC.

Étapes orange (les numéros en orange) :

- 1 : L'utilisateur émet une requête au serveur, le dispatcher (distributeur) récupère le contrôleur et l'action à partir de l'URL, puis il cherche et exécute l'action du contrôleur, sinon il affiche la page par défaut.
- 2 : Le contrôleur à travers son action, il cherche la vue correspondante.
- 3 : La vue retournée au contrôleur.
- 4 : L'utilisateur reçoit la réponse.

Étapes bleues (les numéros en bleu) :

- 1 : L'utilisateur émet une requête au serveur, le dispatcher (distributeur) récupère le contrôleur et l'action à partir de l'URL, puis il cherche et exécute l'action du contrôleur, sinon il affiche la page par défaut.

- 2 : Le contrôleur a travers son action, il demande au modèle d'effectuer une interaction avec la base de données
- 3 : le modèle interagit avec le serveur de la base de données, en demandant de récupérer une liste de données par exemple, en utilisant une Requête "SELECT".
- 3 : Le serveur de la base de données retourne la réponse au modèle.
- 4 : Le modèle de son rôle retourne la réponse au contrôleur.
- 5 : Le contrôleur reçoit les données.
- 6 : Le contrôleur cherche la vue, il la transmet les données.
- 7 : Une vue contenant les données organisées par Html.
- 8 : L'utilisateur reçoit la réponse.

Le cas de saisie des données ou modification ou suppression, une interface (formulaire par exemple) doit s'afficher en passant par **les étapes orange**. Le contrôleur reçoit les données provenant de cet interface, il les vérifie et les valide, il signale les erreurs à l'utilisateur avant de demander au modèle d'effectuer le traitement **"les étapes bleu"**.

Tp: Géstions des Stocks

Une société voulait automatisé son sytème d'information, en développant une application web (Intranet) qui gère les ventes, les fournisseurs, et les stocks.

Pré-requis

Pour réaliser ce Tp, vous devez avoir une bonne connaissance en langage Php et Poo programmation orienté objet.

Outils

Avoir le serveur d'application Apache(Ou autre conteneur Php), serveur de base de données Mysql(Ou autre SGBDR).

Partie de Modélisation

La modélisation consiste à créer une représentation simplifiée d'un problème, pour facilité de le comprendre.

Diagramme de Cas d'utilisation

Ce diagramme présente les services, un service sera présenté sous forme d'un contrôleur.

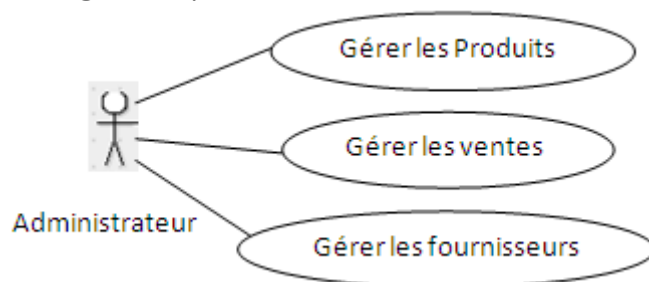


Diagramme de cas d'utilisation.

Diagramme de Classe

Ce qui concerne le développement, nous se concentrons que sur les services de Gestion de produits et stocks

Diagramme de classe.

Diagramme de Séquence

Nous voulons développer que la partie qui concerne la gésion des Produits et leurs Stocks, voilà le diagramme de séquence qui présente les défférentes actions.

Un diagramme de séquence pour le cas d'utilisation "Gérer Les Produits".

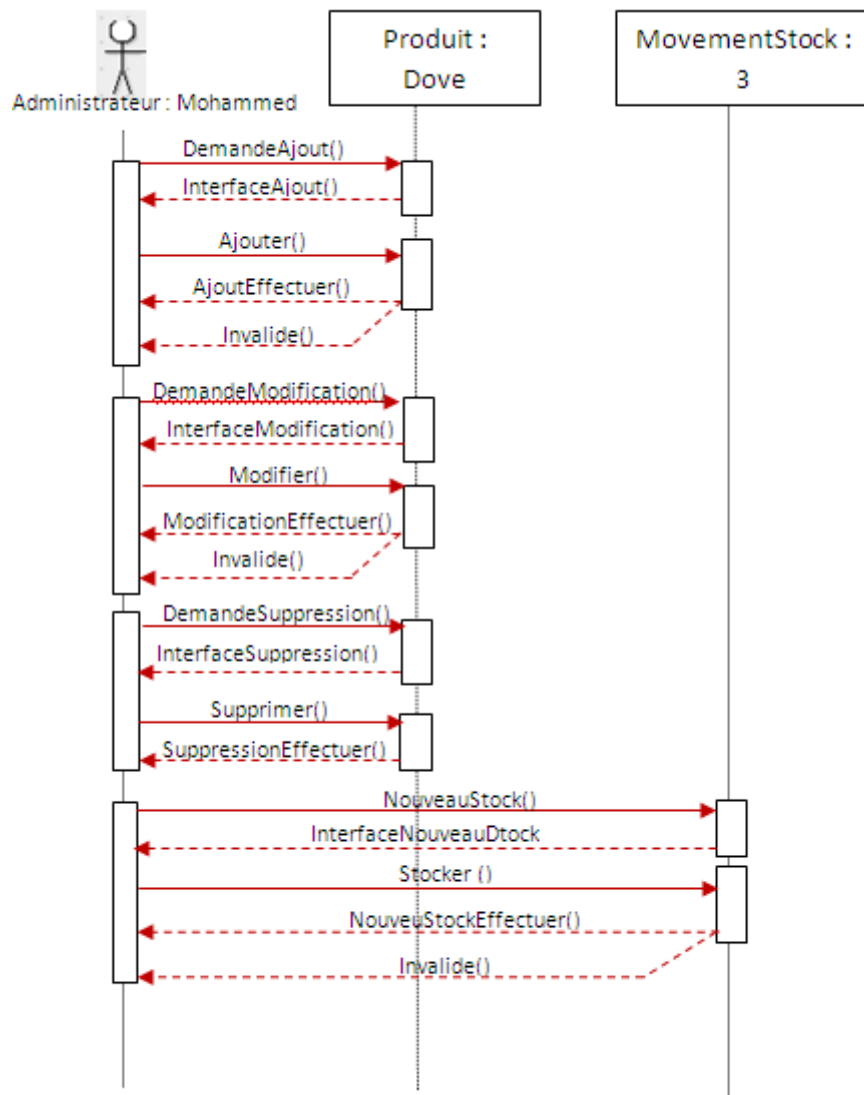


Diagramme de séquence.

La partie de conception et Modélisation étant fini, Maintenant le web designer(Généralement un infographiste) va présenté le projet sous forme des maquettes(Images), et on confirme par la suite avec le client. Après la confirmation le programmeur et l'intégrateur web peuvent commencer la partie des programmation.

Voilà les maquettes qu'on va utiliser.

Home : L'accueil.



Stock : liste des produits et leurs gérations.



Stock : L'ajout d'un produit.

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Ajouter un nouveau produit

Designation :	<input type="text" value="Panthène"/>
Prix Unitaire :	<input type="text" value=""/> (DH) * Erreur Prix Unitaire
Tva :	<input type="text" value=""/> (%) * Erreur tva
Quantité :	<input type="text" value="0"/>
Famille :	<input type="text" value="Shampooing"/> ▼
<input type="button" value="Ajouter"/>	

Programmation en Pratique 2012

Stock : modification d'un produit.

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Modifier un produit

Id :	<input type="text" value="1"/>
Designation :	<input type="text" value="Head & Shoulders"/>
Prix Unitaire :	<input type="text" value="21.3"/> (DH)
Tva :	<input type="text" value="3.2"/> (%)
Quantité :	<input type="text" value="295"/>
Famille :	<input type="text" value="Shampooing"/> ▼
<input type="button" value="Modifier"/>	

Programmation en Pratique 2012

Stock : suppression d'un produit.

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Supprimer un produit

Id :	<input type="text" value="1"/>
Designation :	<input type="text" value="Head & Shoulders"/>
Prix Unitaire :	<input type="text" value="21.3"/> (DH)
Tva :	<input type="text" value="3.2"/> (%)
Quantité :	<input type="text" value="295"/>
Famille :	<input type="text" value="Shampooing"/>
<input type="button" value="Supprimer"/>	

[Retour](#)

Programmation en Pratique 2012

Stock : gestion de stock d'un produit.

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Stockage de Produit

N°	Date de Stock	Quantité
13	03/03/2012	20
14	29/02/2012	50
Nouveau Stock	<input type="text" value="30/2/2012"/> * Date invalide	<input type="text"/> * Quantite invalide
<input type="button" value="Ajouter au Stock"/>		

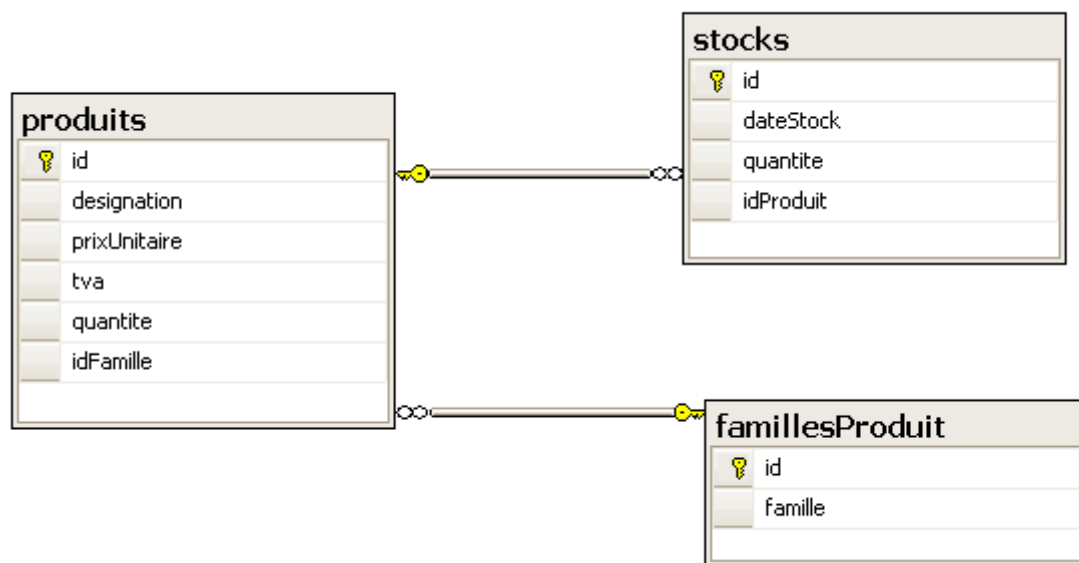
[Retour](#)

Programmation en Pratique 2012

Partie de programmation

1 - Création de la base de données.

Commençant par la création de la base de données. on se basant sur la partie encadré en orange du diagramme de classe. Le modèle physique de données (MPD) correspondant à partir de diagramme de classe.



Modèle Physique de Données(MPD).

j'ai utilisé la ligne de commande pour la création, vous pouvez utiliser l'assistant phpMyAdmin si vous utiliserez le SGBDR mysql

Ligne de Commande : création de base de données.

Sélectionner C:\WINDOWS\system32\cmd.exe - mysql -u root

Console - Windows Trust 3.0 (Service Pack 3: v5512)

(C) 1985-2008 Microsoft Corp.

C:\Documents and Settings\Administrateur>e:

E:\>cd E:\Program Files\wamp5\mysql\bin

E:\Program Files\wamp5\mysql\bin>mysql -u root

ERROR 2003 (HY000): Can't connect to MySQL server on 'localhost' (1006)

E:\Program Files\wamp5\mysql\bin>mysql -u root

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 1

Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database stockProduits;

Query OK, 1 row affected (0.00 sec)

mysql> show databases;

Database
information_schema
babel_db
gestion_stock
gestionstock
labo
mabase
mabasezend
mvc_db
mysql
phpmyadmin
qcm_db
stockproduits
test
testjpggraph
world

15 rows in set (0.16 sec)

```

mysql> use stockProduits;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE `famillesproduit` (
  ->   `id` int(11) NOT NULL auto_increment,
  ->   `famille` varchar(50) NOT NULL,
  ->   PRIMARY KEY (`id`)
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE `produits` (
  ->   `id` int(11) NOT NULL auto_increment,
  ->   `designation` varchar(50) NOT NULL,
  ->   `prixUnitaire` double NOT NULL,
  ->   `tva` double NOT NULL,
  ->   `quantite` int(11) NOT NULL,
  ->   `idFamille` int(11) NOT NULL,
  ->   PRIMARY KEY (`id`),
  ->   KEY `idFamille` (`idFamille`)
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE `stocks` (
  ->   `id` int(11) NOT NULL auto_increment,
  ->   `dateStock` date NOT NULL,
  ->   `quantite` int(11) NOT NULL,
  ->   `idProduit` int(11) NOT NULL,
  ->   PRIMARY KEY (`id`),
  ->   KEY `idProduit` (`idProduit`)
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_stockproduits |
+-----+
| famillesproduit         |
| produits                |
| stocks                  |
+-----+
3 rows in set (0.00 sec)

```

```
mysql> CREATE TRIGGER ajoutStock AFTER INSERT ON stocks
-> FOR EACH ROW
-> UPDATE produits SET quantite = quantite + NEW.qu
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show triggers;
```

Trigger	Event	Table	Statement
ajoutStock	INSERT	stocks	UPDATE produits SET qu

```
1 row in set (0.00 sec)
```

```
mysql> insert into famillesProduit values ('', "Shampooi
Query OK, 1 row affected, 1 warning (0.06 sec)
```

```
mysql> insert into famillesproduit values ('', "Savon")
Query OK, 1 row affected, 1 warning (0.03 sec)
```

```
mysql> select * from famillesproduit;
```

id	famille
1	Shampooing
2	Savon

```
2 rows in set (0.00 sec)
```

```
mysql> insert into produits values('', "Head & Shoulder
Query OK, 1 row affected, 1 warning (0.03 sec)
```

```
mysql> insert into produits values('', "Taws", 3.5, 1.2
Query OK, 1 row affected, 1 warning (0.03 sec)
```

```
mysql> select * from produits;
```

id	designation	prixUnitaire	tva	quantite
1	Head & Shoulders	21	3.2	0
2	Taws	3.5	1.2	0

```
2 rows in set (0.00 sec)
```

```
mysql> insert into stocks values('', now(), 50, 1);
Query OK, 1 row affected, 1 warning (0.05 sec)
```

```
mysql> select * from produits;
```

id	designation	prixUnitaire	tva	quantite
1	Head & Shoulders	21	3.2	50
2	Taws	3.5	1.2	0

2 rows in set (0.00 sec)

```
mysql> insert into stocks values('', now(), 30, 2);
Query OK, 1 row affected, 1 warning (0.01 sec)
```

```
mysql> insert into stocks values('', now(), 20, 1);
Query OK, 1 row affected, 1 warning (0.03 sec)
```

```
mysql> select * from produits;
```

id	designation	prixUnitaire	tva	quantite
1	Head & Shoulders	21	3.2	70
2	Taws	3.5	1.2	30

2 rows in set (0.00 sec)

```
mysql>
```

2 - L'architecture de l'application.

Créant Maintenant un dossier "GestionStockProduit" dans le root www ou dans le dossier ou vous avez spécifié le DocumentRoot lors de l'installation. voilà l'architecture de l'application .

GestionStockProduit

```
|----- index.php
|----- Controllers
|         |----- Controller.Controller.php
|         |----- Produit.Controller.php
|         |----- Vente.Controller.php
|         |----- Fournisseur.Controller.php
|         |----- Home.Controller.php
|----- Models
|         |----- Model.Model.php
|         |----- FamilleProduit.Model.php
|         |----- Produit.Model.php
|         |----- Stock.Model.php
|
|----- Views
|         |----- Produit
|             |----- index.phtml
|             |----- ajouter.phtml
|             |----- modifier.phtml
|             |----- supprimer.phtml
|             |----- stocker.phtml
|         |----- Vente
|             |----- index.phtml
|         |----- Fournisseur
|             |----- index.phtml
|         |----- Home
|             |----- index.phtml
|         |----- Style
|             |----- style.css + images
|         |----- Script
|             |----- script.js + (jQuery)
```

L'architecture de l'application(MVC).

Le design pattern se base sur l'archetecture 3 tiers en séparant les codes dans les dossiers models views et controllers

Chaque contrôleur correspond à un dossier porte le même nom dans le dossier views, et contenant des vues, chacune de ces vues correspond elle aussi à une action de contrôleur.

Actions de controleur Produit	Les Vues corresponds
index() {}	index.phtml
ajouter() {}	ajouter.phtml
modifier() {}	modifier.phtml
supprimer() {}	supprimer.phtml

3 - Création de Models.

La présentation de la base de données au niveau de l'application se fait à travers des models, Ce sont eux qui vont interroger la base de données.

Dans le dossiers de models, on crée les fichiers (~.Model.php).

- Model.Model.php : fichier contenant la classe abstraite Model, regroupe des attributs et méthodes communs
- FamilleProduit.Model.php : fichier contenant la classe FamilleProduit présente la table famillesproduit.
- Produit.Model.php : fichier contenant la classe Produit présente la table produits.
- Stock.Model.php : fichier contenant la classe Stock présente la table stocks.

Model : Model.Model.php

```
<?php

/* Models : Model une classe abstraite, classe mère de toutes les modèles */

abstract class Model{

    //Tableau sera rempli par les nom des attributs no valide.
    protected $ErrorAttribut = array();

    //Tableau contiendra les attributs no valide et les messages d'erreur.
    protected $listError = array();

    //Méthodes de Validation des Attributs

    /**
     * @desc Spécifie le message d'erreur pour
     * les éventuels attributs no valide.
     */
}
```



```

        * @param    str $attribut    Nom de l'attribut
        * @param    str $messageError    Message d'erreur
    */

    public function AddModelError($attribut, $messageError){
        global $ErrorAttribut;
        global $listError;
        for($i=0; $i<count($ErrorAttribut); $i++){
            if($ErrorAttribut[$i] === $attribut)
                $listError[$attribut] = $messageError;
        }

        /**
        lide.
        * @desc    Retourne le message d'erreur de l'attribut no va
        * @param    str $attribut    Nom de l'attribut
        * @return    $listError or ""    Retourne un tableau
        ou vide
        */

        public function ValidationMessage($attribut){
            global $listError;
            if(sizeof($listError)>0);
                return $listError[$attribut];

            return "";
        }

        /**
        êcher une
        * @desc    Vérifie s'il y des attributs no valides pour emp
        action sur la base de données.
        * @return    1 or 0    validé ou no validé
        */

        public function IsValid(){
            global $ErrorAttribut;
            if(count($ErrorAttribut)==0)

```

```

        return 1;
    else
        return 0;
    }
}

?>

```

Model : Produit.Model.php

```

<?php
    /* Models : classe Produit */
    //Inclusion de Classe Mère
    require_once('Model.Model.php');
    class ProduitModel extends Model{
        //Les Attributs : chaque attribut correspond a une colonne de la
        a table produit.
        private $id;
        private $designation;
        private $prixUnitaire;
        private $tva;
        private $quantite;
        private $idFamille;

        //Accesseurs et modificateurs des attributs privé (getters et setters)

        /*
            * @desc retourne id de l'objet courant
            * @return id
        */
        public function getId(){
            return $this->id;
        }
    }

```

```

/*
    * @desc    retourne designation de l'objet courant
    * @return   designation
*/
public function getDesignation(){
    return $this->designation;
}

/*
    * @desc change la valeur de l'attribut designation après
la validation
    * @param   str $value    valeur de l'attribut designatio
n
*/
public function setDesignation($value){
    if(!empty($value))
        $this->designation = $value;
    else{
        global $ErrorAttribut;
        $ErrorAttribut[] = "designation";
    }
}

/*
    * @desc    retourne prixUnitaire de l'objet courant
    * @return   prixUnitaire
*/
public function getPrixUnitaire(){
    return $this->prixUnitaire;
}

/*
    * @desc    change la valeur de l'attribut prixUnitaire apr
és la validation
    * @param   str $value    valeur de l'attribut prixUn
itaire

```

```

*/
public function setPrixUnitaire($value){
    if(is_numeric($value))
        $this->prixUnitaire = $value;
    else{
        global $ErrorAttribut;
        $ErrorAttribut[] = "prixUnitaire";
    }
}

/*
    * @desc    retourne tva de l'objet courant
    * @return    tva
*/
public function getTva(){
    return $this->tva;
}

/*
    * @desc change la valeur de l'attribut tva après la valid
    * @param    str $value    valeur de l'attribut tva
*/
ation
public function setTva($value){
    if(is_numeric($value))
        $this->tva = $value;
    else{
        global $ErrorAttribut;
        $ErrorAttribut[] = "tva";
    }
}

/*
    * @desc    retourne quantite de l'objet courant

```

```

        * @return    quantite
    */
    public function getQuantite(){
        return $this->quantite;
    }
    /*
        * @desc      change la valeur de l'attribut quantite apr
és la validation
        * @param      str $value    valeur de l'attribut quanti
te
    */
    public function setQuantite($value){
        if(is_numeric($value))
            $this->quantite = $value;
        else{
            global $ErrorAttribut;
            $ErrorAttribut[] = "quantite";
        }
    }

    /*
        * @desc      retourne IdFamille de l'objet courant
        * @return     tva
    */
    public function getIdFamille(){
        return $this->idFamille;
    }
    /*
        * @desc      change la valeur de l'attribut IdFamille "Nous
n' avons pas besoin de validation puisque on
va selecter la famille de produit à partir d'une l
iste"
        * @param      str $value    valeur de l'attribut IdFamill
e

```

```

    */
    public function setIdFamille($value){
        $this->idFamille = $value;
    }

    //CRUD : Create    Red    Update    Delete

    //Ajouter
    public function addProduit(){
        $rqt = 'INSERT INTO produits (designation, prixUnitaire,
tva,
                                quantite, idFamille) VALUES ("'.$this->designation
.',"',
                                '.$this->prixUnitaire.', '.$this->tva.',
                                '.$this->quantite.', '.$this->idFamille.')';
        mysql_query($rqt);
    }
    //Modifier
    public function updateProduit($idProduit){
        $rqt = 'UPDATE produits SET designation = "'.$this->desig
nation."',
                                prixUnitaire = '.$this->prixUnitaire.', tva = '.$t
his->tva.',
                                quantite = '.$this->quantite.',
                                idFamille = '.$this->idFamille.' ' WHERE id = '.$id
Produit;
        mysql_query($rqt);
    }
    //Supprimer
    public function deleteProduit($idProduit){
        $rqt = 'DELETE FROM produits WHERE id = '.$idProduit;
        mysql_query($rqt);
    }
    //Lecture
    //    Liste des Produits

```

```

        public function listProduit(){
            $tab = array();
            $rqt = mysql_query("SELECT * FROM produits");
            while($data = mysql_fetch_assoc($rqt))
                $tab[] = $data;
            return $tab;
        }
        //Recherche
        public function findProduit($idProduit){
            $rqt = mysql_query("SELECT * FROM produits WHERE id = ".$
idProduit);
            $data = mysql_fetch_assoc($rqt);
            if(count($data)>0){
                $this->id = $data['id'];
                $this->designation = $data['designation'];
                $this->prixUnitaire = $data['prixUnitaire'];
                $this->tva = $data['tva'];
                $this->quantite = $data['quantite'];
                $this->idFamille = $data['idFamille'];
            }
            return $this;
        }
    }

?>

```

Model : FamilleProduit.Model.php

```

<?php

/* Models : classe FamilleProduit */
require_once('Model.Model.php');
class FamilleProduit extends Model{
    private $id;

```

```

    private $famille;

    //Accesseurs et modificateurs des attributs privé (getters et setters)

    public function getId(){
        return $this->id;
    }

    public function getFamille(){
        return $this->famille;
    }

    public function setFamille($value){
        if(!empty($value))
            $this->famille = $value;
        else{
            global $ErrorAttribut;
            $ErrorAttribut[] = "famille";
        }
    }

    //Ajouter
    public function addFamilleProduit(){
        $rqt = 'INSERT INTO famillesProduit (famille) VALUES ('. $this->famille. ')';
        mysql_query($rqt);
    }

    //Modifier
    public function updateFamilleProduit($idCategorie){
        $rqt = 'UPDATE famillesProduit SET famille = '. $this->famille. '
        WHERE id = '. $idCategorie;
        mysql_query($rqt);
    }

    //Supprimer
    public function deleteFamilleProduit($idCategorie){
        $rqt = 'DELETE FROM famillesProduit

```



```

        WHERE id = ".$idCategorie;
    mysql_query($rqt);
}

//Liste des Produits
public function listFamilleProduit(){
    $tab = array();
    $rqt = mysql_query("SELECT * FROM famillesProduit");
    while($data = mysql_fetch_assoc($rqt))
        $tab[] = $data;
    return $tab;
}

//Recherche
public function findFamilleProduit($idCategorie){
    $tab = array();
    $rqt = mysql_query("SELECT * FROM categories
        WHERE id = ".$idCategorie);
    while($data = mysql_fetch_assoc($rqt))
        $tab[] = $data;
    return $tab;
}

//Liste Sous-ensemble
public function listFamilleProduit_Produit(){
    $tab = array();
    $rqt = mysql_query("SELECT * FROM produits
        WHERE idCategorie = ".$this->id);
    while($data = mysql_fetch_assoc($rqt))
        $tab[] = $data;
    return $tab;
}

}

```

?>

Model : Stock.Model.php

```
<?php

/* Models : classe Stocks */
//Inclusion de Classe Mère
require_once('Model.Model.php');
class StockModel extends Model{
    private $id;
    private $date;
    private $quantite;
    private $idProduit;

    //Accesseur des attributs privé (getter et setter)
    public function getId(){
        return $this->id;
    }

    public function getDate(){
        return $this->date;
    }
    public function setDate($value){
        if(strpos($value, '/') )
            $dt = explode('/', $value);
        elseif(strpos($value, '-') )
            $dt = explode('-', $value);
        if(checkdate((int) $dt[1], (int) $dt[0], (int) $dt
[2]))
            $this->date = $dt[2].'/'.$dt[1].'/'.$dt[0]
;
        else{
            global $ErrorAttribut;
            $ErrorAttribut[] = "date";
        }
    }
}
```

```

    }

    public function getQuantite(){
        return $this->quantite;
    }

    public function setQuantite($value){
        if(is_numeric($value))
            $this->quantite = $value;
        else{
            global $ErrorAttribut;
            $ErrorAttribut[] = "quantite";
        }
    }

    public function getIdProduit(){
        return $this->idProduit;
    }

    public function setIdProduit($value){
        $this->idProduit = $value;
    }

    //Ajouter
    public function addStock(){
        $rqt = 'INSERT INTO stocks (dateStock, quantite, idProduit)
        VALUES ("'.$this->date.', '.$this->quantite.',
        '.$this->idProduit.')';
        mysql_query($rqt);
    }

    //Modifier
    public function updateStock($idStock){
        $rqt = 'UPDATE stocks SET dateStock =
        DATE_FORMAT('.$this->date.', "%Y/%m/%d"), quantite
        =
        '.$this->quantite.', idProduit = '.$this->idProduit.'
    t.'

```

```

        WHERE id = '.$idStock;
        mysql_query($rqt);
    }
    //Supprimer
    public function deleteStock($idStock){
        $rqt = 'DELETE FROM stocks WHERE id = '.$idStock;
        mysql_query($rqt);
    }
    //Liste des stocks
    public function listStock(){
        $rqt = "SELECT * FROM stocks";
        while($data = mysql_fetch_assoc($rqt))
            $tab[] = $data;
        return $tab;
    }
    //Recherche
    public function findStock($idStock){
        $rqt = "SELECT * FROM stocks WHERE id = ".$idStock
;
        while($data = mysql_fetch_assoc($rqt))
            $tab[] = $data;
        return $tab;
    }
    //Liste de Stock (Sous-ensemble)
    public function listStockProduit(){
        $tab = array();
        $rqt = mysql_query("SELECT id, DATE_FORMAT(dateStock, '%d/%m/%Y')
. $this->idProduit);
        as date, quantite FROM Stocks WHERE idProduit = "
        while($data = mysql_fetch_assoc($rqt))
            $tab[] = $data;
        return $tab;
    }

```

```
}  
?>
```

4 - Dispatcher, Layout, Feuille de style .

Le dispatcher reçoit l'url et récupère le contrôleur, l'action et l'id.

Notre travail consiste sur l'envoi du nom de contrôleur, l'action, et l'id à travers l'url.

index.php

```
<div id="Menu">  
  <h3>Menu</h3>  
  <ul>  
    <li><a href="?controller=Home">Home</a></li>  
    <li><a href="?controller=Stock">Stocks</a></li>  
    <li><a href="?controller=Fournisseur">Fournisseurs</a></li>  
  </ul>  
</div>  
<div id="main">  
  <?php  
    /*Traitement de la Requetes*/  
    $controller = 'Home';  
    $action = 'index';  
    $id = '';  
    if(!empty($_GET['controller'])) {  
      $controller = $_GET['controller'];  
      if(!empty($_GET['action'])) {  
        $action = $_GET['action'];  
        if(!empty($_GET['id']))  
          $id = $_GET['id'];  
      }  
    }  
    echo $controller;  
  ?>  
</div>
```

L'envoi du nom de contrôleur

Récupération de nom de contrôleur, l'action et l'id, qui ont été déjà initialiser par des valeurs par défaut.

Affichage :

Gestion des Stocks

Stock

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Programmation en Pratique 2012

On ajoutant cette partie de code qui va chercher le fichiers contenant le contrôleur demandé puis l'inclure, s'il y a une action il va l'appeler, et s'il y a un id, il va le transmettre dans l'action.

index.php

<?php

```
/*Traitement de la Requettes*/
```

```
$controller = 'Home';
```

```
$action = 'index';
```

```
$id = '';
```

```
if(!empty($_GET['controller'])) {
```

```
    $controller = $_GET['controller'];
```

```
    if(!empty($_GET['action'])) {
```

```
        $action = $_GET['action'];
```

```
        if(!empty($_GET['id']))
```

```
            $id = $_GET['id'];
```

```
    }
```

```
}
```

```
//On inclut le fichier s'il existe et s'il est spécifié
```

```
if(is_file('Controllers/'.$controller.'.Controller.php')) {
```

```
    include 'Controllers/'.$controller.'.Controller.php';
```

```
    $class = $controller."Controller";
```

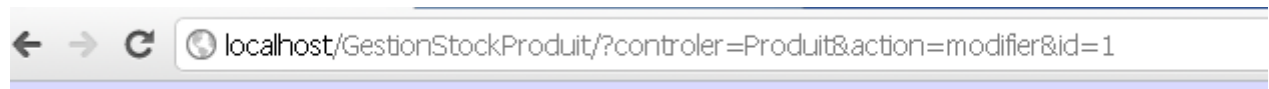
```
    $objet = new $class();
```

```
    $objet->$action($id);
```

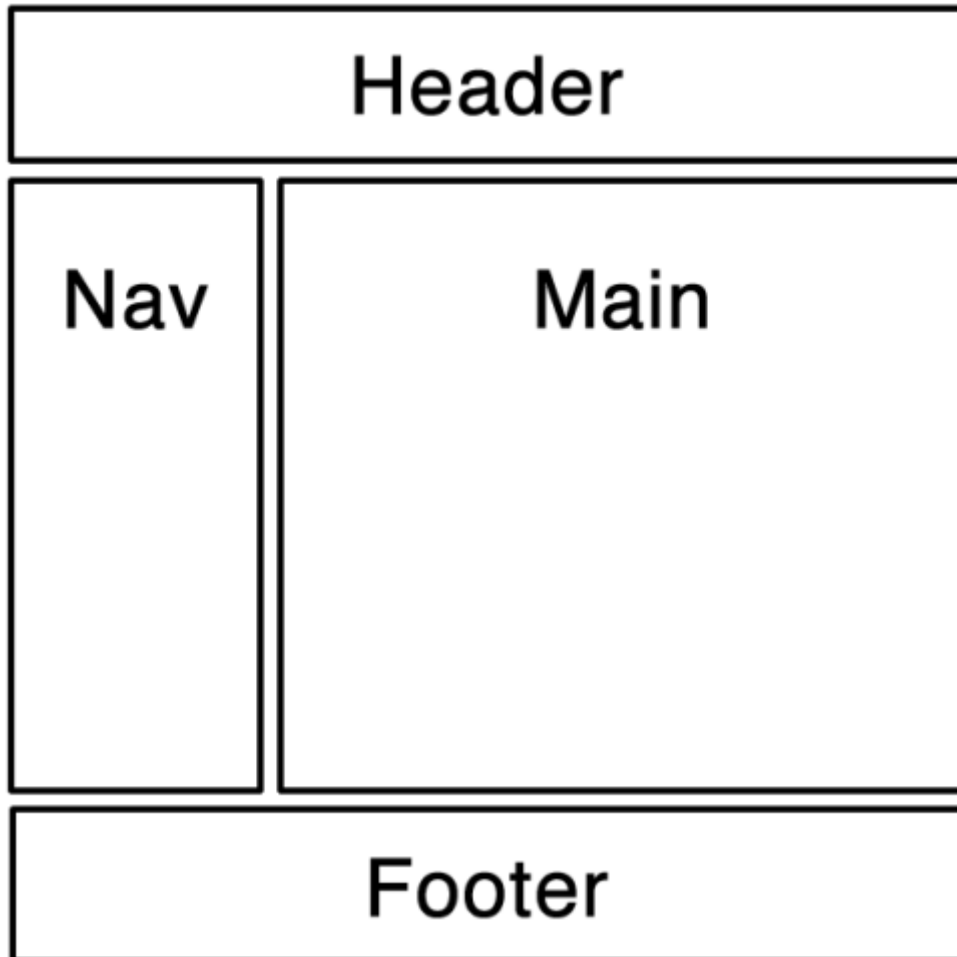
```
}
```

?>

L'url



Layout ou Master plan c'est la mise en page de site ou le plan graphique de site. voilà le gabarit de notre site.



layout.

Il reste une autre tâche à effectuer par ce dispatcher(index.php), la connexion avec le serveur de la base de données et la sélection de la base. Le code final de index.php :

index.php

```
<?php
    session_start();
?>
<html>
    <head>
        <title></title>
```

```

        <link href="Views/Style/style.css" rel="stylesheet" type=
"text/css" />
    </head>
    <body>
        <div id="wrap">
            <div id="header">
                <h1>Gestion des Stocks</h1>
            </div>
            <div id="content">
                <div id="Menu">
                    <h3>Menu</h3>
                    <ul>
                        <li><a href="?controler=Home
">Home</a>
                        </li>
                        <li><a href="?controler=Prod
uit">Stocks
                        </a></li>
                        <li><a href="?controler=Four
nisseur">
                            Fournisseurs</a></li>
                    </ul>
                </div>
                <div id="main">
                    <?php
                        //Connexion au serveur de ba
se de données
                        mysql_connect('localhost', '
root', '');
                        mysql_select_db('stockProdui
ts');
                        /*Traitement de la Requettes
*/
                        $controler = 'Home';
                        $action = 'index';
                        $id = '';

```



```

    )){
        if(!empty($_GET['controller'])
        $controller = $_GET['c
        on' ]))){
            if(!empty($_GET['acti
            $action = $_G
            if(!empty($_G
            $id =
            $_GET['id'];
        }
    }
    //On inclut le fichier s'il existe et s'il
    est spécifié
    on' ].
    if(is_file('Controllers/'.$c
    '.Controller.php')){
        include 'Controllers/'.$cont
        '.Cont
        roller.php';
        $class = $controller."Control
        ler";
        $objet = new $class();
        $objet->$action($id);
    }
    //Fermeture de connexion ave
    c Serveur de DB
    mysql_close();
    ?>
</div>
</div>
<div id="footer">
    Programmation en Pratique 2012
</div>

```

```
</div>
</body>
</html>
```

Dans le contrôleur Home on crée une action index();

Home : page par défaut

```
<?php
    /* Controlers : Home*/
    class HomeController{

        //Action index
        public function index(){
            echo 'Je suis dans l\'action du contrôleur Home';
        }
    }
?>
```

Affichage :



5 - Contrôleur et vue .

Maintenant que nous avons créé le cerveau* de l'application, on va développer les actions de contrôleur produit

Chaque contrôleur a au moins une action (index) qu'elle correspond à une vue particulière (index.phtml).

Prenant le cas de demande de site pour la 1^o fois, le contrôleur et son action par défaut sont "HomeController" et "index", il doit appeler la vue "index.phtml" situant dans le dossier "Home" dans Views.

Contrôleur : Home.Controller.php

```
<?php
    /* Controlers : Home*/
    //Inclusion de Classe Mère
    require_once('Controller.Controller.php');
    class HomeController extends Controller{

        //Action index
        public function index() {
            include(dirname(__FILE__) . '/../Views/Home/index.phtml');
        }
    }
?>
```

Vue : Views/Home/index.phtml

```
<!--
    Remarque : Dans les vues on trouve que de code html et quelque
    instructions comme (boucle, if, echo) puisqu'elles ont dédiés
    seulement pour interagir avec l'utilisateur.
-->
<h1>Programmation en Pratique</h1>
<h2>Gestion des Produits</h2>
```

Affichage :



On doit inclure chaque fois la vue dans chaque action correspondante, hors on peut développer une fonction qui nous retourne dans chaque action la vue correspondante. Cette fonction doit être utilisable dans tous les contrôleurs, c'est pour cela qu'elle doit être déclarer dans la classe mère, nous avons aussi besoin d'une fonction qui permet la redirection.

Contrôleur : Controllers/Controller.Controller.php

```
<?php

    //inclusion les models , utilisés par les actions de controleur
    include(dirname(__FILE__).'../Models/Produit.Model.php');
    include(dirname(__FILE__).'../Models/FamilleProduit.Model.php')
;

    include(dirname(__FILE__).'../Models/Stock.Model.php');
```

```

/* Controlers : Classe Abstraite */
abstract class Controller{
    /*Un tableau pour Récupère le données de contrôleur (provenant
    éventuellement de models) pour usage futur(Dans les vues)
    */

    protected $viewData = array();

    //Retourne la vue correspondante
    protected function View($action){
        //Récupérer le controleur a partir de classe fille
        .
        $controller = str_replace('Controller', '', get_class($this));

        //La vue correspondant
        $view = $controller . '/' . $action.'.phtml';
        include(dirname(__FILE__).'../Views/'.$view);
    }

    //Rederiction vers une action d'un contrôleur
    protected function RederictAction(){
        /*Récupération des paramètres : contrôleur(action index par défaut ou action d'un
        contrôleur précis avec ou sans id identifiant d'un objet qui aurait subi l'action*/

        $arg_list = func_get_args();
        if(count($arg_list)== 1)
            $url = 'controller='.$arg_list[0];
        elseif(count($arg_list)== 2)
            $url = 'controller='.$arg_list[0]. '&action='.$arg_list[1];
        else
            $url = 'controller='.$arg_list[0]. '&action='.$arg_list[1]. '
            &id='.$arg_list[2];
        echo '<script type="text/javascript">

```

```

                                window.location = "?" . $url . "'" </scri
pt>';
                                }
                                }
?>

```

Contrôleur : Controllers/Home.Controller.php

```

<?php
    /*  Controlers : Home*/
    //Inclusion de Classe Mère
    require_once('Controller.Controller.php');
    class HomeController extends Controller{

        //Action index
        public function index(){
            $this->View( __FUNCTION__);
        }
    }
?>

```

Maintenant on va se concentrer sur les actions définie dans le diagramme de séquence,

- Contrôleur : Produit
- Actions : index(toujours par défaut), ajouter, modifier, supprimer, et stocker
- Vues : index.phtml(liste des produits), ajouter.phtml, ...stocker.phtml

Contrôleur : Produit.Controller.php.

```

<?php
    /*  Controlers : Produit*/
    //Inclusion de Classe Mère
    require_once('Controller.Controller.php');
    class ProduitController extends Controller{

        //Action index
        public function index(){

```

```

        $produit = new ProduitModel();
        //met dans viewData une liste des produits
        $this->viewData['listProduit'] = $produit->listPro
duit();

        $this->View(__FUNCTION__);
    }

```

Vue : Produit/index.phtml

```

<h1>Listes de Produits</h1>
<table>
    <tr>
        <th>N°</th><th>Designation</th>
        <th>Prix Unitaire (Dh)</th><th>Tva(%)</th>
        <th>Quantité</th><th>Suppression</th>
        <th>Modification</th><th>Stockage</th>
    </tr>
<?php
    //On récupère la liste a partir de viewData
    foreach($this->viewData['listProduit'] as $produit) :
        echo'<tr><td>'.$produit['id'].'</td>
            <td>'.$produit['designation'].'</td>
            <td>'.$produit['prixUnitaire'].'</td>
            <td>'.$produit['tva'].'</td>
            <td>'.$produit['quantite'].'</td>
            <td><a href="?controler=Produit&action=mo
difier&id=
                '.$produit['id'].'">Modifier</a></t
d>
            <td><a href="?controler=Produit&action=su
pprimer&id=
                '.$produit['id'].'">Supprimer</a></
td>
            <td><a href="?controler=Produit&action=st
ocker&id=

```

```

        '.$produit['id'].'>Stocker</a></td>
    </tr>';
endforeach;
?>
</table>
<a href="?controller=Produit&action=ajouter">Ajouter un nouveau produit<
/a>

```

Affichage :

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Listes de Produits

N°	Désignation	Prix Unitaire (Dh)	Tva[%]	Quantité	Suppression	Modification	Stockage
1	Head & Shoulders	21.3	3.2	295	Modifier	Supprimer	Stocker
4	Panthène	15	3.5	30	Modifier	Supprimer	Stocker
5	Camy	25.3	3.5	10	Modifier	Supprimer	Stocker
6	Detole	15	3.5	500	Modifier	Supprimer	Stocker
7	Cadum	21	3.5	50	Modifier	Supprimer	Stocker

[Ajouter un nouveau produit](#)

Programmation en Pratique 2012

Action : ajouter()

Contrôleur : Produit.Controller.php

```

//Action :ajouter

    public function ajouter(){
        $produit = new ProduitModel();
        $this->viewData['produit'] = $produit;
        $famille = new FamilleProduit();
        $this->viewData['listFamille'] = $famille->listFamilleProduit();
    }

```



```

/*Pour éviter que les message d'erreurs ne soient
afficher
au premier appel de la vue, on vérifier */
if(!isset($_POST['designation']))
    $this->View(__FUNCTION__);
else{
    $produit->setDesignation($_POST['designation']);
    $produit->AddModelError("designation", " * Erreur
Désignation");

    $produit->setPrixUnitaire($_POST['prix']);
    $produit->AddModelError("prixUnitaire", " * Erreur
Prix Unitaire");

    $produit->setTva($_POST['tva']);
    $produit->AddModelError("tva", " * Erreur tva");
    $produit->setQuantite($_POST['quantite']);
    $produit->AddModelError("quantite", " * Erreur Qua
ntité");

    $produit->setIdFamille($_POST['famille']);
    //Validation des données.
    if($produit->IsValid()){
        $produit->addProduit();
        //rederiction vers index.phtml
        $this->RedirectAction("Produit");
    }else
        $this->View(__FUNCTION__);
    }
}

```

Vue : Produit/ajouter.phtml

```

<h1>Ajouter un nouveau produit</h1>
<form id="form1" method="post">
    <table>
        <tr>
            <td>Designation : </td>

```

```

        <td>
            <input type="text" id="designation" name="
designation"
            value="<?php echo $this->viewData['produit']->getD
esignation();?>" />
            <span id="erreur">
                <?php
echo $this->viewData['produit']->ValidationMessage
("designation");
                ?>
            </span>
        </td>
    </tr>
    <tr>
        <td>Prix Unitaire : </td>
        <td>
            <input type="text" id="prix" name="prix"
            value="<?php echo $this->viewData['produit']->getPrixUnit
aire();?>" />(DH)
            <span id="erreur">
                <?php
echo $this->viewData['produit']->ValidationMessage
("prixUnitaire");
                ?>
            </span>
        </td>
    </tr>
    <tr>
        <td>Tva : </td>
        <td>
            <input type="text" id="tva" name="tva"
            value="<?php echo $this->viewData['produit']->getT
va();?>" />(%)
            <span id="erreur">
                <?php
echo $this->viewData['produit']->Validatio
nMessage("tva");
                ?></span>
            </td>
        </tr>
    <tr>
        <td>Quantité : </td>
        <td><input type="text" id="quantite" name="quantit
e"
            value="<?php echo $this->viewData['produit']->getQuantite
();?>" />
    </tr>

```

```

        <span id="erreur">
        <?php
        echo $this->viewData['produit']->ValidationMessage
("quantite");
        ?></span>
    </td>
</tr>
<tr>
    <td>Famille : </td>
    <td>
        <select id="famille" name="famille">
        <?php
            foreach($this->viewData['listFamille'] as $cat)
                echo '<option value="'. $cat['id']. '">
                    '.$cat['famille'].'</option>
                ';
            ?>
        </select>
    </td>
</tr>
<tr><td colspan='2'><input type="submit" value="Ajouter"
/></td></tr>
</table>
</form>

```

Affichage :

Action : modifier()

Contrôleur : Produit.Controller.php

```

//Action : Modifier

    public function modifier($id){
        $produit = new ProduitModel();
    }

```

```

        $this->viewData['produit'] = $produit->findProduit($id);
        $famille = new FamilleProduit();
        $this->viewData['listFamille'] = $famille->listFamillePro
duit();

        if(!isset($_POST['designation']))
            $this->View( __FUNCTION__);
        else{
            $produit->setDesignation($_POST['designation']);
            $produit->AddModelError("designation", " * Erreur
 Désignation");

            $produit->setPrixUnitaire($_POST['prix']);
            $produit->AddModelError("prixUnitaire", " * Erreur
 Prix Unitaire");

            $produit->setTva($_POST['tva']);
            $produit->AddModelError("tva", " * Erreur tva");
            $produit->setQuantite($_POST['quantite']);
            $produit->AddModelError("quantite", " * Erreur Qua
 ntité");

            $produit->setIdFamille($_POST['famille']);
            if($produit->IsValid()){
                $produit->updateProduit($id);
                //rederiction vers index.phtml
                $this->RedirectAction("Produit");
            }else
                $this->View( __FUNCTION__);
        }
    }
}

```

Vue : Produit/modifier.phtml

```

<h1>Modifier un produit</h1>

<form id="form1" method="post">
    <table>
        <tr>

```

```

        <td>Id : </td>
        <td><input type="text" id="id" name="id" readonly=
"true"
        value="<?php echo $this->viewData['produit
']->getId();?>" />
        </td>
    </tr>
    <tr>
        <td>Designation : </td>
        <td><input type="text" id="designation" name="desi
gnation"
        value="<?php echo $this->viewData['produit']->getD
esignation();?>" />
        <span id="erreur"><?php
echo $this->viewData['produit']->ValidationMessage
("designation");
?></span>
        </td>
    </tr>
    <tr>
        <td>Prix Unitaire : </td>
        <td><input type="text" id="prix" name="prix"
        value="<?php echo $this->viewData['produit']->getPrixUnit
aire();?>" />(DH)
        <span id="erreur"><?php
echo $this->viewData['produit']->ValidationMessage("prixU
nitaire");
?></span>
        </td>
    </tr>
    <tr>
        <td>Tva : </td>
        <td><input type="text" id="tva" name="tva"
        value="<?php echo $this->viewData['produit']->getT
va();?>" />(%)
        <span id="erreur"><?php

```

```

        echo $this->viewData['produit']->ValidationMessage
("tva");

        ?></span>
    </td>
</tr>
<tr>
    <td>Quantité : </td>
    <td><input type="text" id="quantite" name="quantit
e"
value="<?php echo $this->viewData['produit']->getQ
uantite();?>" />

    <span id="erreur"><?php
echo $this->viewData['produit']->ValidationMessage
("quantite");

    ?></span>
</td>
</tr>
<tr>
    <td>Famille : </td>
    <td>
        <select id="famille" name="famille">
            <?php
foreach($this->viewData['listFamille'] as $cat) :
                if($cat['id'] == $this->viewData['produit'
]->getIdFamille())
                    echo '<option value="'. $cat['id']. '
" selected>'
                        . $cat['famille']. '</option>'
;
                else
                    echo '<option value="'. $cat['id']. '
                        . $cat['famille']. '</option>'
;
            endforeach;
        ?>

```

```

        </select>
    </td>
</tr>
<tr><td colspan='2'><input type="submit"
value= "  Modifier  " /></td></tr>
</table>
</form>

```

Affichage :

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Modifier un produit

Id :	<input type="text" value="1"/>
Designation :	<input type="text" value="Head & Shoulders"/>
Prix Unitaire :	<input type="text" value="21.3"/> (DH)
Tva :	<input type="text" value="3.2"/> (%)
Quantité :	<input type="text" value="295"/>
Famille :	<input type="text" value="Shampooing"/> <div> Shampooing Savon </div>
<input type="button" value="Modifier"/>	

Programmation en Pratique 2012

Action : supprimer()

Contrôleur : Produit.Controller.php

```

//Action : Supprimer

public function supprimer($id){
    $produit = new ProduitModel();
    $this->viewData['produit'] = $produit->findProduit($id);
    $famille = new FamilleProduit();

```

```

        $this->viewData['listFamille'] = $famille->listFamillePro
duit();

        if(isset($_POST['id'])){
            $produit->deleteProduit($id);
            //redirection vers index.phtml
            $this->RedirectAction("Produit");
        }
        $this->View(__FUNCTION__);
    }

```

Vue : Produit/modifier.phtml

```

<h1>Supprimer un produit</h1>

<form id="form1" method="post">
    <table>
        <tr>
            <td>Id : </td>
            <td><input type="text" id="id" name="id" readonly=
"true"
                value="<?php echo $this->viewData['produit
']->getId();?>" />
            </td>
        </tr>
        <tr>
            <td>Designation : </td>
            <td>
                <input type="text" id="designation" name="designat
ion"
                    readonly="true" value="<?php
                        echo $this->viewData['produit']->getDe
signation();?>" />
            </td>
        </tr>
        <tr>
            <td>Prix Unitaire : </td>

```



```

                <td>
                    <input type="text" id="prix" readonly="true"
e" name="prix"
                    value="<?php
                    echo $this->viewData['produit']->getPrixUn
itaire();?>" />(DH)
                </td>
            </tr>
            <tr>
                <td>Tva : </td>
                <td>
                    <input type="text" id="tva" readonly="true"
"
                    name="tva" value="<?php
                    echo $this->viewData['produit']->getTva();
?>" />(%)
                </td>
            </tr>
            <tr>
                <td>Quantité : </td>
                <td>
                    <input type="text" id="quantite" readonly=
"true"
                    name="quantite" value="<?php
                    echo $this->viewData['produit']->getQuanti
te();?>" />
                </td>
            </tr>
            <tr>
                <td>Famille : </td>
                <td>
                    <input type="text" id="famille" readonly="
true"
                    name="famille" value="<?php
                    foreach($this->viewData['listFamille'] as $cat)

```

```

] -> getIdFamille())
                                if($cat['id'] == $this->viewData['produit']
                                echo $cat['famille'];
                                ?>" />
                                </td>
                                </tr>
                                <tr><td colspan='2'><input type="submit" value= " Suppri
mer " />
                                </td></tr>
                                </table>
                                </form>
                                <a href="?controller=Produit">Retour</a>

```

Affichage :

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Supprimer un produit

Id :	<input type="text" value="1"/>
Designation :	<input type="text" value="Head & Shoulders"/>
Prix Unitaire :	<input type="text" value="21.3"/> (DH)
Tva :	<input type="text" value="3.2"/> (%)
Quantité :	<input type="text" value="295"/>
Famille :	<input type="text" value="Shampooing"/>
<input type="button" value="Supprimer"/>	

[Retour](#)

Programmation en Pratique 2012

Action : stocker()

Contrôleur : Produit.Controller.php

```
//Action : Stocker
```

```

    public function stocker($id){
        $stock = new StockModel();
        $stock->setIdProduit($id);
        $this->viewData['stock'] = $stock;
        $this->viewData['listStock'] = $stock->listStockProduit()
;

        if(!isset($_POST['date']))
            $this->View( __FUNCTION__);
        else{
            $stock->setDate($_POST['date']);
            $stock->AddModelError("date", " * Date invalide");
            $stock->setQuantite($_POST['quantite']);
            $stock->AddModelError("quantite", " * Quantite inv
alide");

            if($stock->IsValid()){
                $stock->addStock();
                //rederiction vers index.phtml
                $this->RedirectAction("Produit");
            }else
                $this->View( __FUNCTION__);
        }
    }

    //Autre Actions : ....

```

Vue : Produit/modifier.phtml

```

<h1>Stockage de Produit </h1>
<form id="form1" method="post">
    <table>
        <tr><th>N°</th><th>Date de Stock</th><th>Quantité</th></tr>
    <tbody>
        <tr>
            <td><input type="text" value="N°" />
            <td><input type="text" value="Date de Stock" />
            <td><input type="text" value="Quantité" />
        </tr>
    </tbody>
</table>
<input type="submit" value="Ajouter" />
</form>

<?php
    //Listes des Stock d'un produit
    foreach($this->viewData['listStock'] as $stock)

```

```

        echo '<tr><td>'.$stock['id'].'</td><td>'.$stock['date'].'</td><td>'.$stock['quantite'].'</td></tr>';
    }

    ?>
    <tr>
        <td>Nouveau Stock</td>
        <td>
            <input type="text" id='date' name='date'
            value="<?php echo $this->viewData['stock']->getDate();?>" />
            <br>
            <span id="erreur">
                <?php echo $this->viewData['stock']->ValidationMessage("date");?>
            </span>
        </td>
        <td>
            <input type="text" id='quantite' name='quantite'
            value="<?php echo $this->viewData['stock']->getQuantite();?>" />
            <br>
            <span id="erreur">
                <?php echo $this->viewData['stock']->ValidationMessage("quantite");?>
            </span>
        </td>
    </tr>
    <tr><td colspan='3'><input type="submit" value="Ajouter au Stock" />
    </td></tr>
</table>
<a href="?controller=Produit">Retour</a>

```

</form>

Affichage :

Gestion des Stocks

Menu

- [Home](#)
- [Stocks](#)
- [Fournisseurs](#)

Stockage de Produit

N°	Date de Stock	Quantité
13	03/03/2012	20
14	29/02/2012	50
Nouveau Stock	<input type="text" value="30/2/2012"/> * Date invalide	<input type="text"/> * Quantite invalide
<input type="button" value="Ajouter au Stock"/>		

[Retour](#)

Programmation en Pratique 2012

[Partie précédente](#)