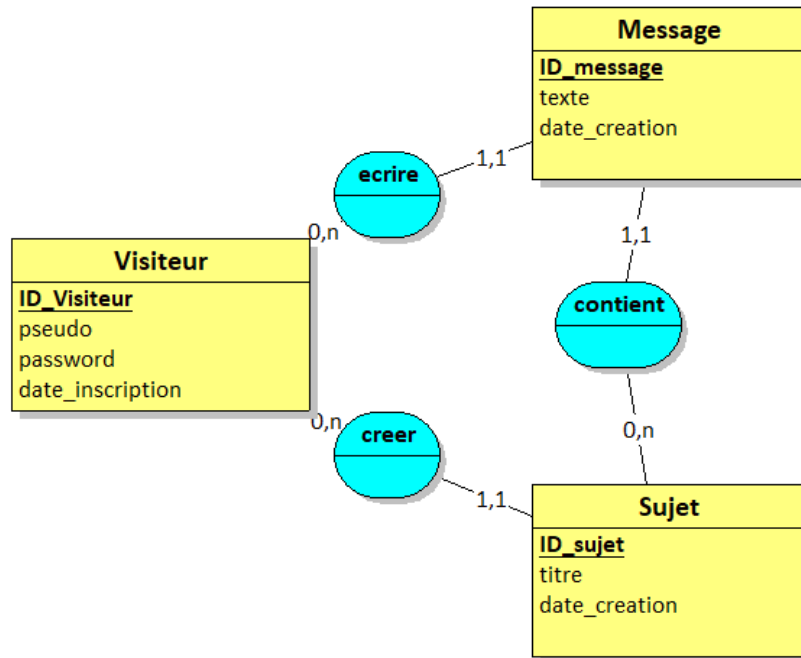


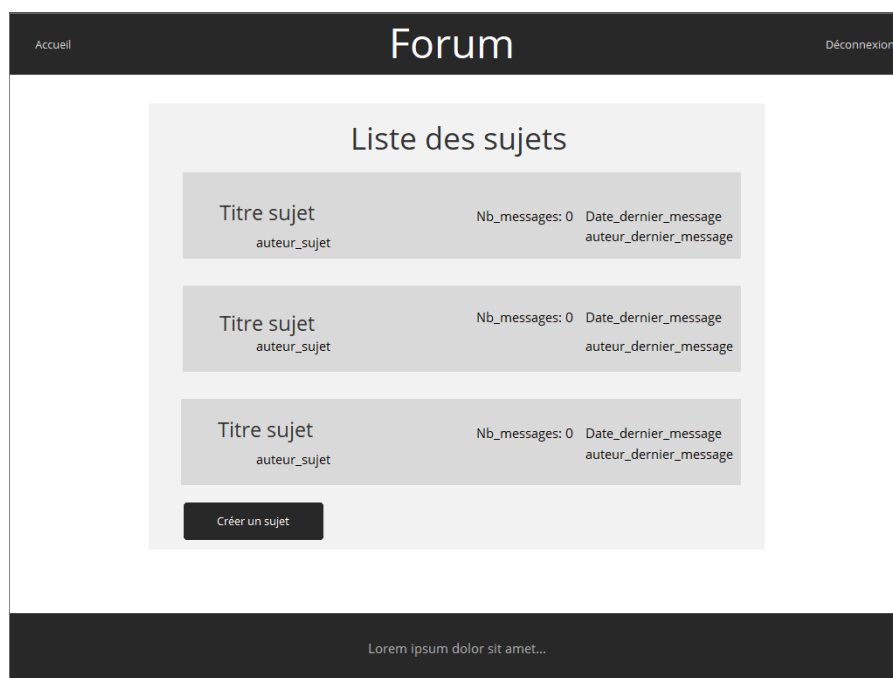
Dossier Forum MVC

1. MCD



MCD du Forum

2. Maquettage



Accueil du Forum

[Accueil](#) Forum [Déconnexion](#)

[Retour aux sujets](#)

Titre sujet

Date_creation & auteur_sujet

Auteur_message

Date_message

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariaturn aliquam pharetra eget.

[Editer](#) [Supprimer](#)

Auteur_message

Date_message

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariaturn aliquam pharetra eget.

REPONDRE

Votre message

envoyer

Lorem ipsum dolor sit amet...

Model d'un sujet

3. Rendu visuel effectif

[Accueil](#) Hello S2LF [Se déconnecter](#)

Liste des Topics

Titre du topic1
Crée par S2LF le 28/01/2020 à 09:20
X

Titre du topic 2
Crée par S2LF le 28/01/2020 à 09:22
X

topic
Crée par Test le 31/01/2020 à 10:25
X

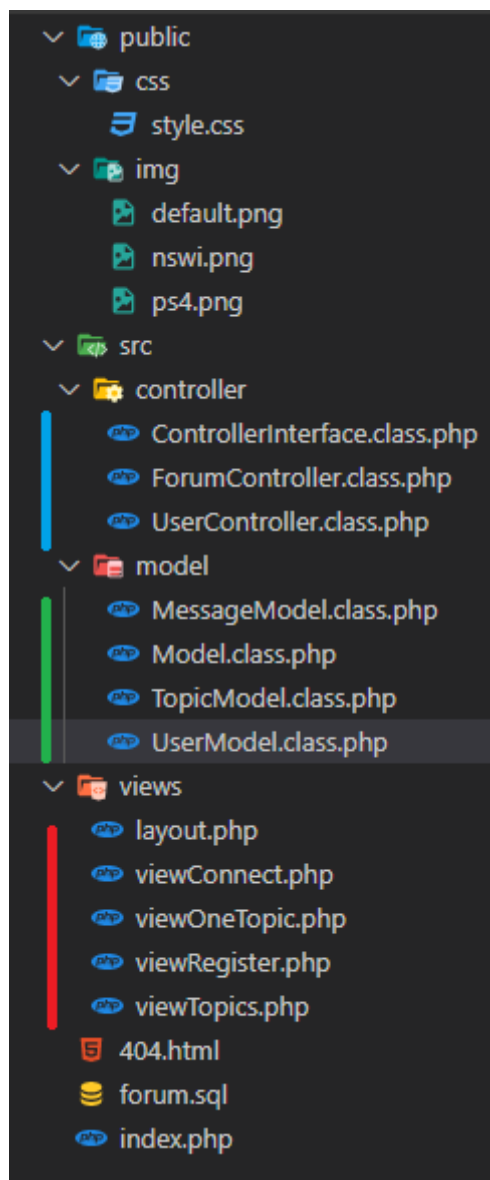
[CREATE TOPIC !](#)

Accueil du Forum



Model d'un sujet

4. Explication du Design Pattern



Pour cette exercice le Design Pattern a été le MVC :

4.1 Model

- **abstract Model** -> Model type avec la connexion à la BDD. Les autres model hérite de celui-ci.

- **Topic Model** -> Rassemble les fonctions CRUD et les requêtes SQL lié à la table Topic.

- **MessageModel** -> Rassemble les fonctions CRUD et les requêtes SQL lié à la table Message.

- **User Model** -> Rassemble les fonctions CRUD et les requêtes SQL lié à la table User.

4.2 Controller

- **Controller Interface** -> Implément chaque Controller et l'oblige a utilisé une fonction IndexAction, fonction par défaut quand la classe est appelé.

- **Forum Controller** -> Rassemble les fonctions CRUD lié aux Sujets et aux Messages. Selon les besoin, récupère ou retourne des données au Model ou à la View.

- **User Controller** -> Rassemble les fonctions CRUD lié à l'Utilisateur. Selon les besoin, récupère ou retourne des données au Model ou à la View.

4.3 Views

- **Layout** -> La fonction du Layout est d'afficher les éléments commun à chaque pages (Header, Nav, Footer). Y est présent aussi le Doctype, Head (feuilles de styles , CDN, JQuery ...). Chaque View va afficher son contenu au sein de la variable \$Page.

- **view Topics** -> Affiche la liste des Sujets. Un utilisateur connecté pourra créer / éditer / supprimer un sujet qu'il a créé.

- **view One Topic** -> Affiche un Sujet. Un utilisateur connecté pourra créer / éditer / supprimer un message qu'il a écrit.

- **view Register** -> Affiche la page pour s'enregistrer.

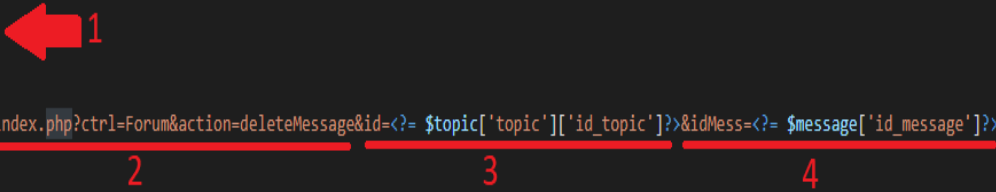
- **view Connect** -> Affiche la page de connexion.

5. Illustration d'une fonctionnalité (MVC) : Supprimer un message.

En illustration, j'ai choisi de montrer la fonctionnalité qui permet de supprimer un message. Nous allons partir de l'utilisateur, donc de la **Vue**, puis suivre le chemin de la donnée vers le **Contrôleur** puis le **Model**

5.1 ViewOneTopic

```
<?php
if(isset($_SESSION['user'])){
}
<div>
  <a id="delete" href="index.php?ctrl=Forum&action=deleteMessage&id=<?= $topic['topic']['id_topic']?>&idMess=<?= $message['id_message']?>"><i> Supprimer</i></a>
</div>
<?php
}
```



1. Cette fonctionnalité est réservée aux personnes enregistrées (qui ont donc une session User). A noter que pour l'instant toute personne enregistrée peut supprimer un message.

2. Le lien renvoie donc vers index.php (front controller) en GET avec comme 2 premières variables :

- *ctrl=Forum* -> le contrôleur appelé est ForumController

- *action*=deleteMessage -> la fonction appelé est deleteMessageAction()

3. La variable *id* contient l'id du sujet courant. Cette variable me servira ensuite à la redirection vers le sujet concerné par la suppression du message

4. La variable *idMess* contient l'id du message concerné par la suppression.

5.2 ForumController

```
public function deleteMessageAction() { 1
    $id_topic = $_GET['id']; 2
    $id_mess = $_GET['idMess']; 2
    $model = new MessageModel(); 3
    $model->deleteMessage($id_mess); 3
    header("Location:index.php?ctrl=Forum&action=viewOneTopic&id=$id_topic"); 4
}
```

1. La fonction *deleteMessageAction()* a pour but de récupérer les informations permettant de définir quel message doit être supprimé. Après avoir envoyé cette information au bon *Model* la fonction redirige vers le sujet concerné par la suppression.

2. Les variables passés en *GET* sont récupéré. Par sécurité il faudrait qu'elle soient filtré.

3. On appelle le *Model* concerné par la fonction, ici *MessageModel* puis on exécute la fonction *deleteMessage()* avec l'argument nécessaire à l'exécution de la requête SQL, ici l'id du message à supprimer.

4. On redirige vers le sujet concerné par la suppression du message. A terme on pourrait renvoyer une erreur si la suppression n'a pas marché.

5.3 MessageModel

```
public function deleteMessage($id){
    2 | $sql = "DELETE FROM message 1
    WHERE id_message = $id";
    3 | $stmt = self::$connexion->prepare($sql);
    $stmt->execute();
}
```

1. La fonction *deleteMessage()* nécessite un argument : l'id du message à supprimer.

2. On prépare la requête SQL : on veut supprimer dans la table *message* la ligne comprenant l'id correspondant à l'argument passé dans la fonction (voir 1.).

3. On prépare et on exécute la requête SQL

6. Utilisation de Framework

Utilisation de JQuery :

- Dans la liste des messages, un bouton [*Editer*] permet de faire apparaître/disparaître le formulaire d'édition.
- Utilisation de *Confirm()* pour la déconnexion et la suppression de Sujet/Message.

Aucun Framework CSS n'a été utilisé et aucune ligne de code n'a été maltraité durant la réalisation de ce projet.