

Dossier de Synthèse

Lisa Foret

I. Description du projet

Ce projet consiste à créer une plate-forme de communication en ligne de type forum. Simple d'accès et intuitif. L'application doit répondre aux besoins suivants :

- Une page d'accueil avec un lien vers un formulaire de connexion et un lien vers un formulaire d'inscription
- Une connexion obligatoire pour poster un sujet ou répondre à un sujet
- Les visiteurs peuvent consulter les sujets et leurs réponses sans restriction, ces sujets sont listés en première page et triés par date de création dans l'ordre antéchronologique

II. Techno

L'application utilise les technologies suivantes :



Symfony 5.1 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.



Uikit 3.5 framework css



PHP 7.4 est un langage de programmation libre, conçu pour le développement d'application web



MySQL est un système de gestion de bases de données relationnelles



HTML (HyperText Markup Language) est le langage de balisage conçu pour la création de pages web



CSS (Cascading Style Sheets) est un langage informatique qui décrit la présentation des documents HTML.



Jmerise est un logiciel dédié à la modélisation des modèles conceptuels de données



Visual studio code est un éditeur de code multi-plateforme



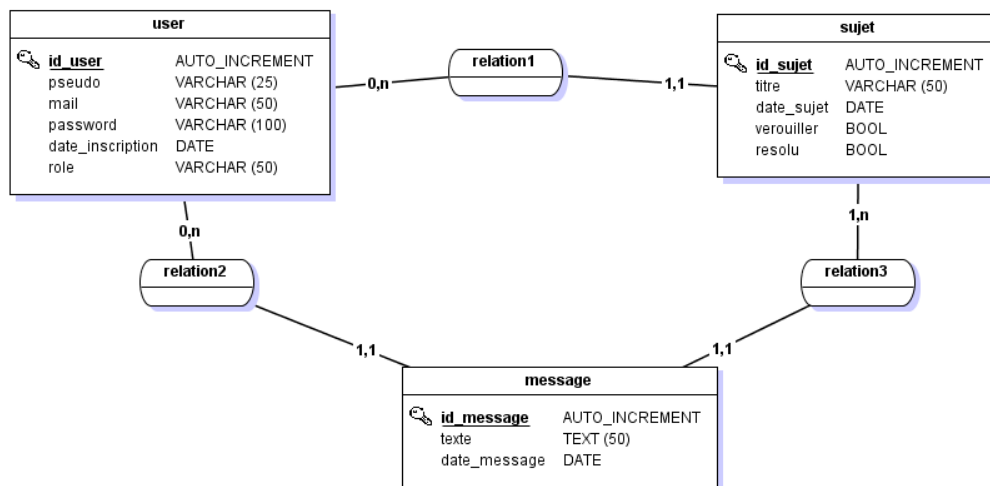
Justinmind : logiciel de conception d'interfaces web



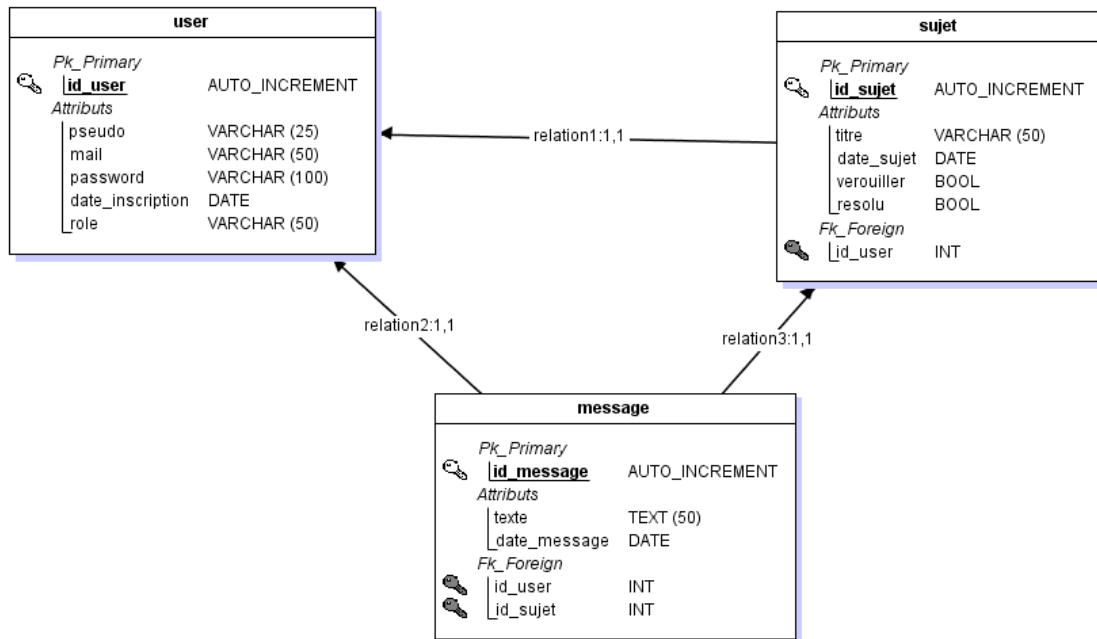
Laragon est un environnement de développement web. Il inclut PHP (langage interprété coté serveur), MySQL (gestion de la base de données) et Apache (serveur web)

III. MCD/MLD

La première étape de ce projet est de créer un schéma entités-relation de la base de données avec Jmerise. Il permet de concevoir un Modèle Conceptuel de Données (MCD) qui établit une représentation claire des données. Pour ce projet, on crée trois entités : user, sujet et message. Chacune de ces entités possède des propriétés tel que le pseudo ou le titre, mais également une propriété unique comme l'ID. Elles sont reliées entre elles par des relations, par exemple un utilisateur peut créer zéro ou plusieurs sujets (0,n), mais un sujet ne peut être écrit que par un seul utilisateur (1,1).



Le MCD une fois terminé est converti en MLD (Modèle logique de données) qui permet de modéliser les données en prenant en compte l'organisation des données. Le MLD affiche les clés étrangères (ex ; Id_user dans la table sujet) et permet ainsi de mieux comprendre les relations entre les entités. Jmerise génère également le script de la BDD.



IV. Maquettage

La maquette permet de visualiser le site pour évaluer l'ergonomie du site, et d'analyser l'enchaînement de la navigation et l'agencement des blocs d'information. Ici on utilise JustinMind.

[Accueil](#)
[Profil](#)
[Connexion](#)
[Déconnexion](#)

LISTE SUJETS

[Nouveau sujet](#)

Titre du sujet	Pseudo	Date de création	Statut	Résolu	Vérouiller	DELETE
Titre du sujet	Pseudo	Date de création	Statut	Résolu	Vérouiller	DELETE
Titre du sujet	Pseudo	Date de création	Statut	Résolu	Vérouiller	DELETE
Titre du sujet	Pseudo	Date de création	Statut	Résolu	Vérouiller	DELETE

Accueil

Profil

Connexion

Déconnexion

TITRE DU SUJET

MESSAGE

PSEUDO Date de création

REPONSE MESSAGE

PSEUDO Date de création

Delete

Réponse message

Envoyer

V. Fonctionnalités

Inscription/connexion/déconnexion

Espace administrateur

Consulter un sujet

Ajouter/delete/edit un sujet/message

VI. Migration

Le framework symfony fournit beaucoup de commandes différentes, on peut grâce à la console, créer ou modifier des entités avec la commande suivante : 'php bin/console make:entity', la console nous demande ensuite le nom de l'entité et ses propriétés. Il existe une commande pour directement créer une entité user : 'php bin/console make:user', la console nous propose également de choisir une propriété unique ici ça sera l'email. Ensuite on utilise la commande : 'php bin/console make:controller' pour créer nos contrôleurs.

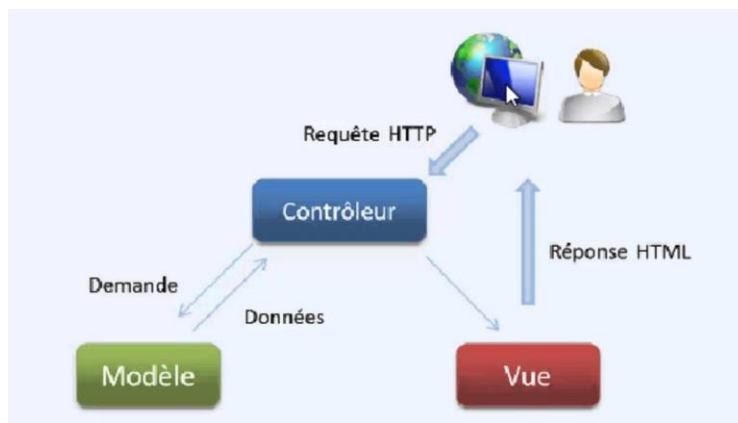
Tous les fichiers sont générés automatiquement, les entités avec leur getters & setters ainsi que les repository de chaque entité, et les contrôleurs. Après avoir créé nos entités on peut les migrer vers la base de données avec les commandes suivantes : 'php bin/console make:migration' & 'php bin/console doctrine* :migrations :migrate', cela permet de générer automatiquement toutes nos tables et nos champs . Doctrine est une librairie PHP utilisé pour travailler avec les bases de données, Symfony nous fournit cet outil. C'est un ORM (Object Relational Mapping), il s'agit d'un moteur qui fait la relation entre la BDD et l'application.

VII. Architecture

Un Design Pattern est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différentes applications. Symfony est basé sur un design pattern en particulier le MVC (Modèle Vue Contrôleur).

Le MVC sépare le code en trois couches :

- Le modèle, c'est la couche qui permet de traiter les informations dans la base de données, il comprend les dossiers 'entity' et 'repository', c'est ce dernier qui accède à la BDD grâce aux requêtes DQL.
- La vue, c'est ce que l'utilisateur obtiendra dans son navigateur.
- Le contrôleur lui gère les "interactions" entre les différents fichiers, il reçoit la requête venant de l'utilisateur puis il appelle le modèle pour obtenir certaines données, il renvoie à la vue pour l'afficher sur le navigateur de l'utilisateur



VIII. Sécurité

Sécurité

Faible XSS ou cross-site scripting consiste à injecter un script dans une page pour provoquer une action bien définie. L'utilisateur exécute le script sans s'en rendre compte dès l'ouverture de la page. Permet également de transmettre des données depuis un site vers un autre.

Symfony s'en prémunit avec l'utilisation de Twig.

Injection SQL : modification d'une requête envoyée à la base de données pour en détourner l'utilisation.

Faible CSRF