Rapport de projet TPI

Projet « Forum »

CPNV

Benjamin Muminovic

Table des matières

[1 Analyse préliminaire 2](#_Toc499021832)

[1.1 Introduction 2](#_Toc499021833)

[1.2 Objectifs 2](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 5](#_Toc499021836)

[2.1 Concept 5](#_Toc499021837)

[2.2 Stratégie de test 12](#_Toc499021838)

[2.3 Risques techniques 12](#_Toc499021839)

[2.4 Planification 12](#_Toc499021840)

[2.5 Dossier de conception 13](#_Toc499021841)

[3 Réalisation 16](#_Toc499021842)

[3.1 Dossier de réalisation 16](#_Toc499021843)

[3.2 Description des tests effectués 17](#_Toc499021844)

[3.3 Erreurs restantes 17](#_Toc499021845)

[3.4 Liste des documents fournis 17](#_Toc499021846)

[4 Conclusions 18](#_Toc499021847)

[5 Annexes 19](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 19](#_Toc499021849)

[5.2 Sources – Bibliographie 19](#_Toc499021850)

[5.3 Journal de travail 19](#_Toc499021851)

[5.4 Manuel d'Installation 19](#_Toc499021852)

[5.5 Manuel d'Utilisation 19](#_Toc499021853)

[5.6 Archives du projet **Erreur ! Signet non défini.**](#_Toc499021854)

# Analyse préliminaire

## Introduction

Ce projet a été réalisé dans le cadre du travail personnel individuel (TPI) de 4ème année de CFC Informatique. Le projet m’a été imposé par un cahier des charges fourni par mon chef de projet. Le thème a cependant été choisi selon ma demande.

Ce projet consiste en la réalisation d’un forum web qui permettra à un utilisateur de créer des postes auxquels d’autres utilisateurs pourront répondre.

Aucun travail préliminaire n’a été réalisé pour ce projet. J’ai cependant eu l’occasion de réaliser un projet pré-TPI afin de me préparer pour ce projet.

## Objectifs

Voici le descriptif du projet tel qu’il a été fourni dans le cahier des charges :

Le site web proposé permettra de publier, lire et répondre à des posts sur différents thèmes.

Le site doit couvrir les cas suivants :

* Partie « public »
  + - « Liste des posts » : une page d’accueil présentera les posts les plus récents organisés par thème et par titre. Le titre du post, son auteur, sa date de création ainsi que le nombre de réponses seront visibles. Toute autre information jugée nécessaire pourra être ajoutée. En cliquant sur le titre, le post complet avec les réponses des autres utilisateurs s’afficheront.
    - L’utilisateur pourra aussi filtrer les posts en sélectionnant un thème à disposition.
    - Un utilisateur pourra s’enregistrer sur le site. Un email lui sera envoyé pour confirmer sa demande de connexion.
* Partie « utilisateur connecté »
  + - Un utilisateur connecté pourra rentrer un post avec un titre, un lien sur une vidéo ou une image et le thème auquel il appartient (actualité, sport, films, jeux vidéos…). La date du post sera celle au moment de son insertion.
    - Un utilisateur connecté pourra publier un commentaire sur un post et répondre à un commentaire déjà existant. Il ne sera cependant pas possible de répondre à un commentaire de commentaire.
    - Il pourra signaler un post ou un commentaire inadéquat à l’administrateur.
    - Il peut voir l’ensemble des posts qu’il a publiés.
* Partie « administrateur »
* Un administrateur pourra ajouter des thèmes (musique, histoire…).
* Il pourra voir les commentaires / posts signalés comme inadéquats.
* Il pourra aussi bloquer un utilisateur si celui-ci publie des propos inappropriés ou bloquer les commentaires sur un post s’ils incitent à la haine, à la discrimination ou sont vulgaires.
* Il pourra historiser/bloquer un post lorsqu’il n’est plus d’actualité ou lorsqu’il n’est pas approprié. Ce post ne sera plus visible par les utilisateurs mais encore présent dans la base de données.

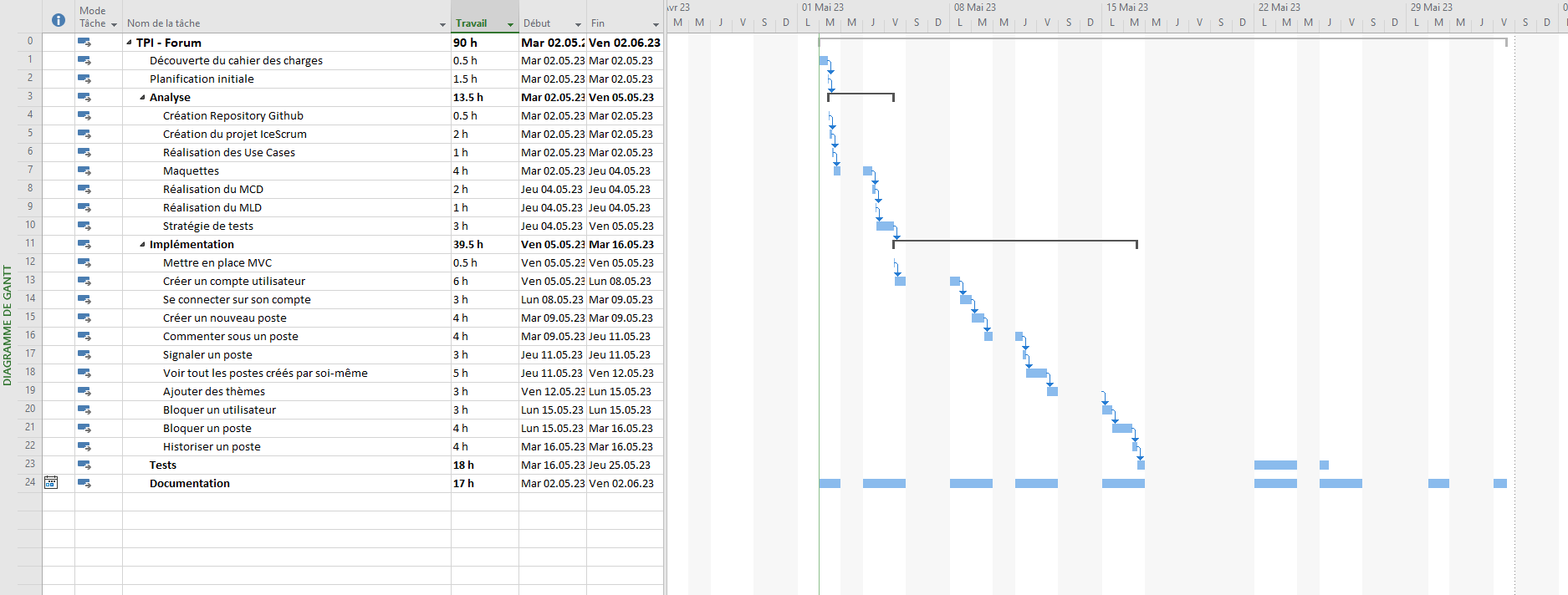
Toutes les informations persistantes seront stockées dans une base de données. Celle-ci sera laissée au choix du candidat qui devra le justifier.

Le site sera responsive, il devra permettre une lecture et une navigation conviviales et optimales, qu’il soit consulté depuis un téléphone mobile, une tablette ou un ordinateur.

Le candidat sera libre de choisir sa méthode de gestion de projet. Il devra justifier son choix.

Le candidat sera libre de choisir une architecture pour son code. Il devra justifier son choix.

## Planification initiale

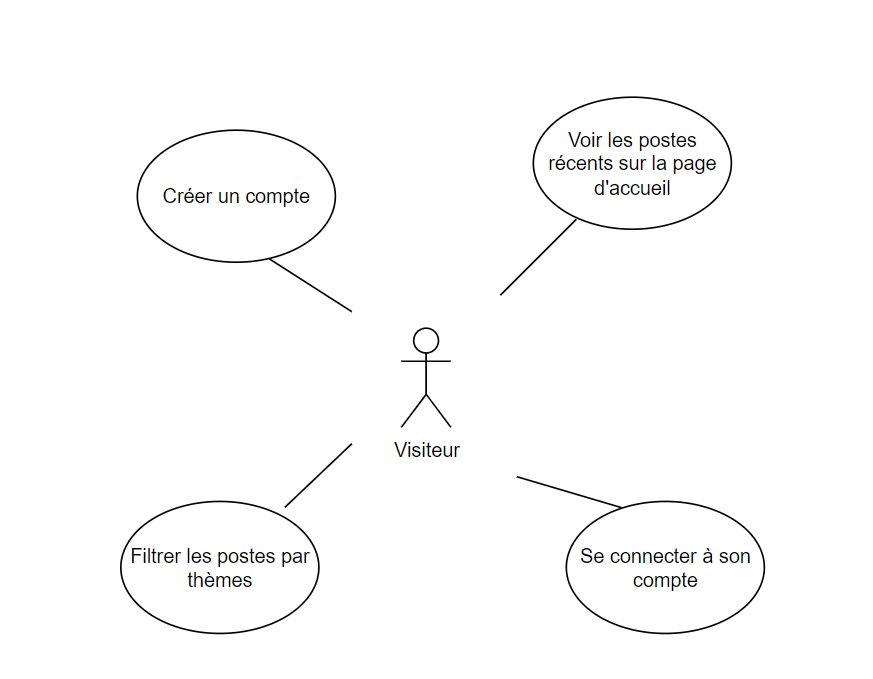


# Analyse / Conception

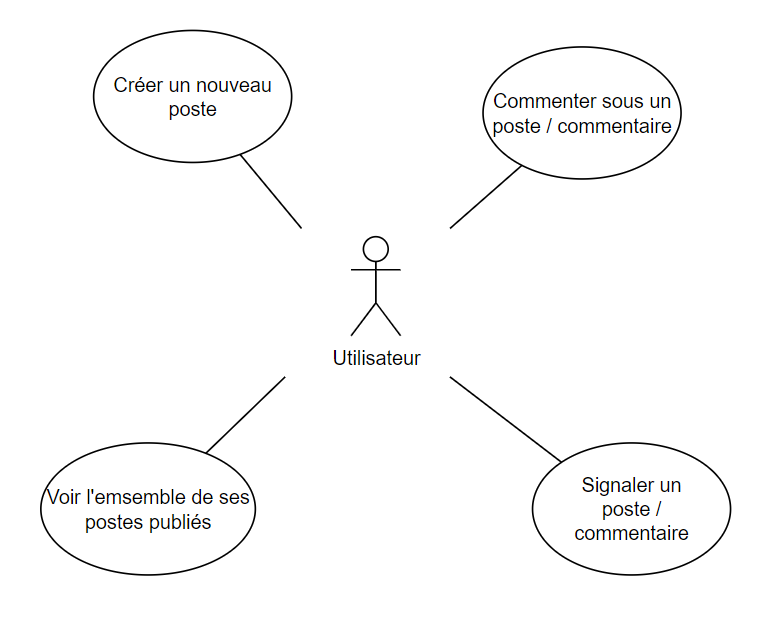
## Concept

### Use cases

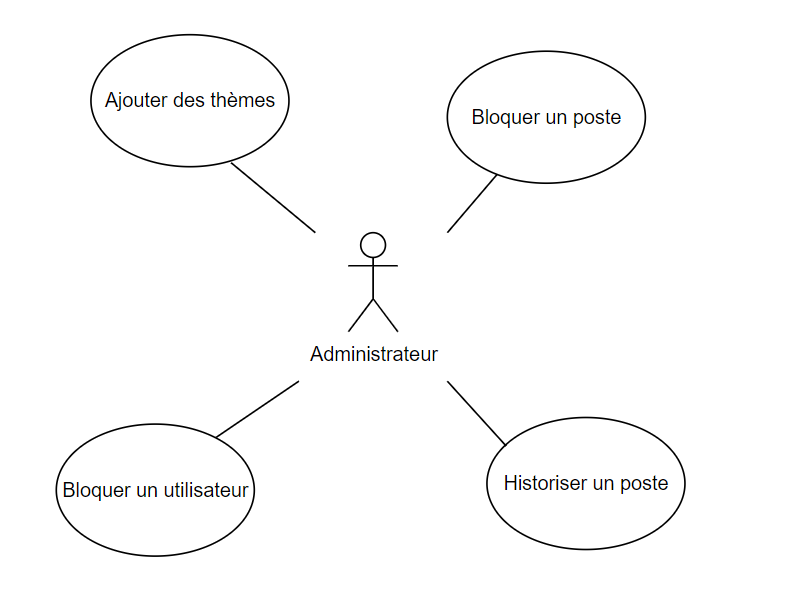
#### Visiteur



#### Utilisateur

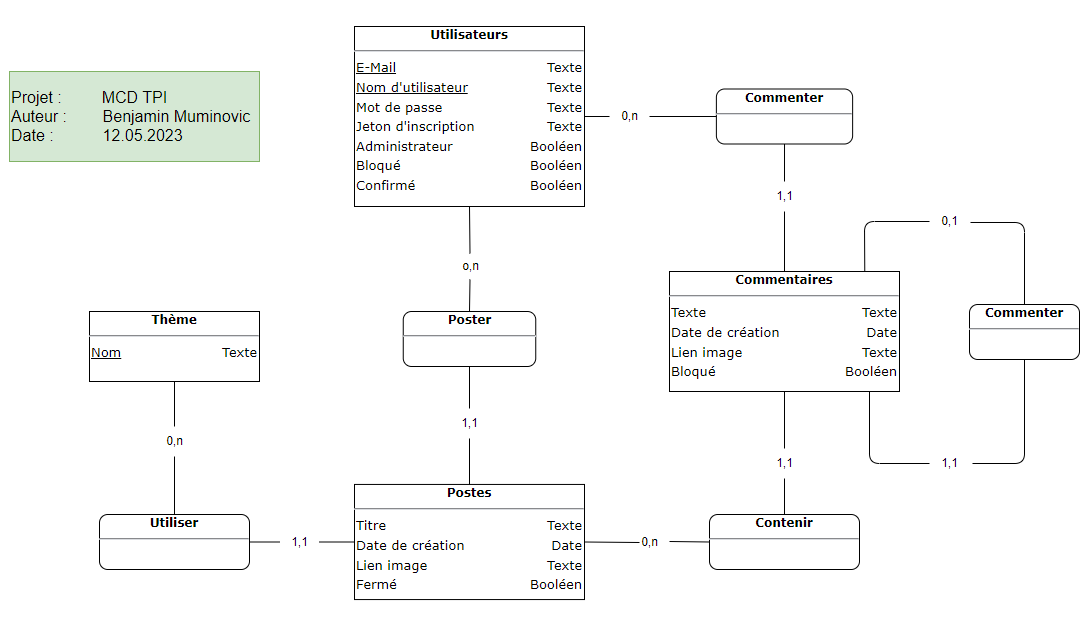


#### Administrateur



### Modèles de données

#### Modèle conceptuel de données

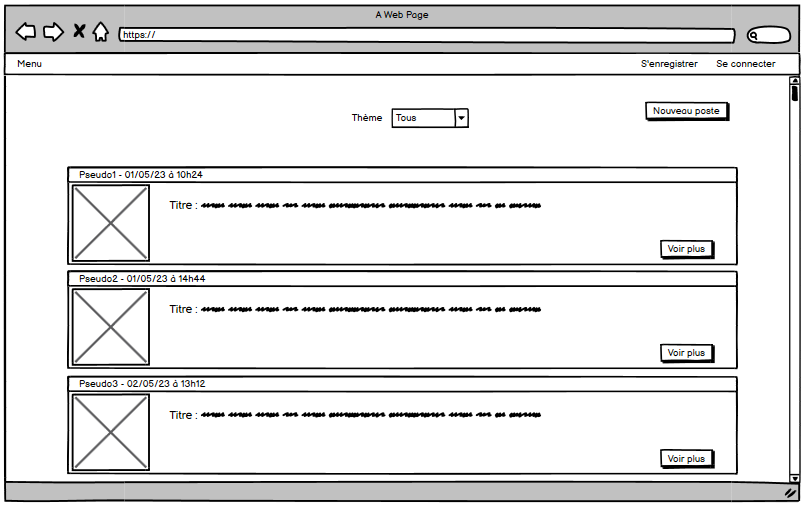


Explication de l’auto-liaison de la table « commentaires » :

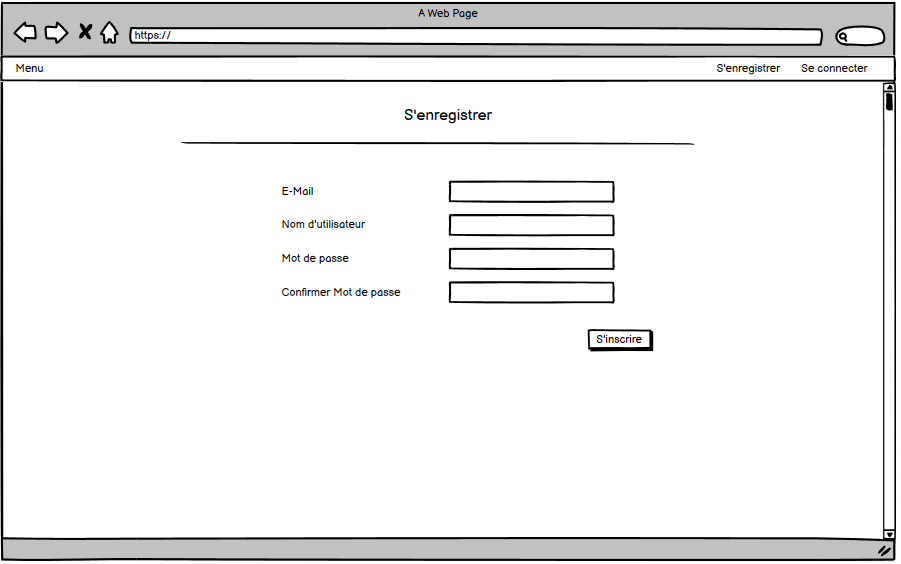
Puisqu’un commentaire ne peut peut avoir qu’un seul niveau de commentaire (on ne peut pas répondre à un commentaire de commentaire), le commentaire possédera une clé étrangère pointant vers l’id du commentaire auquel il répond (ou vaudra 0 s’il répond au poste).

#### Modèle logique de données

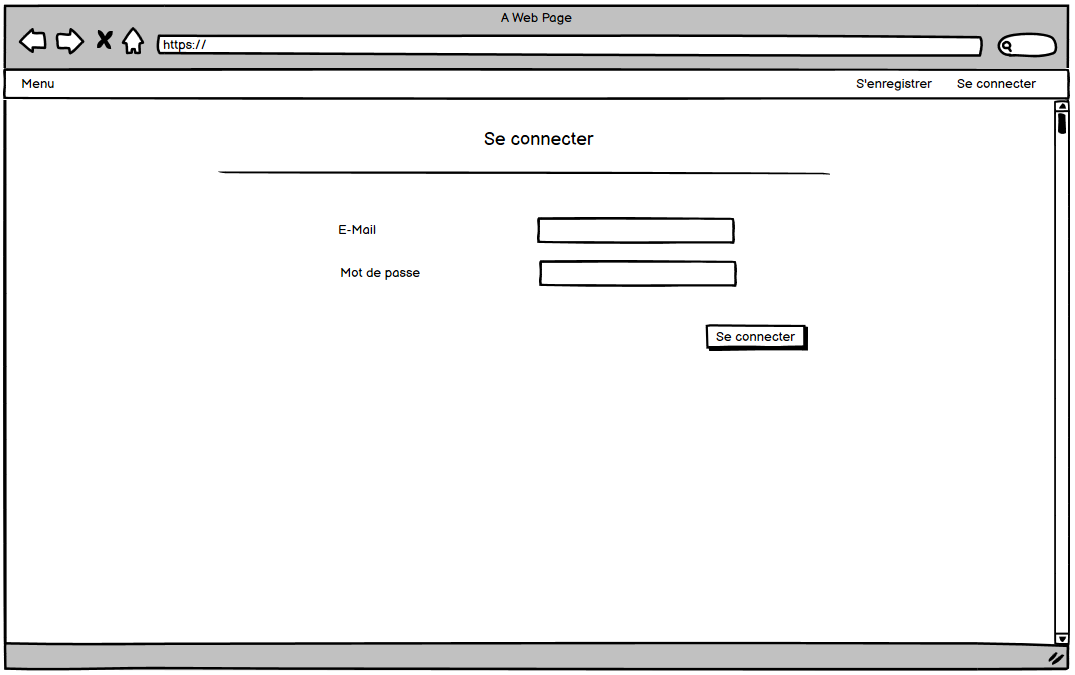
### Maquettes



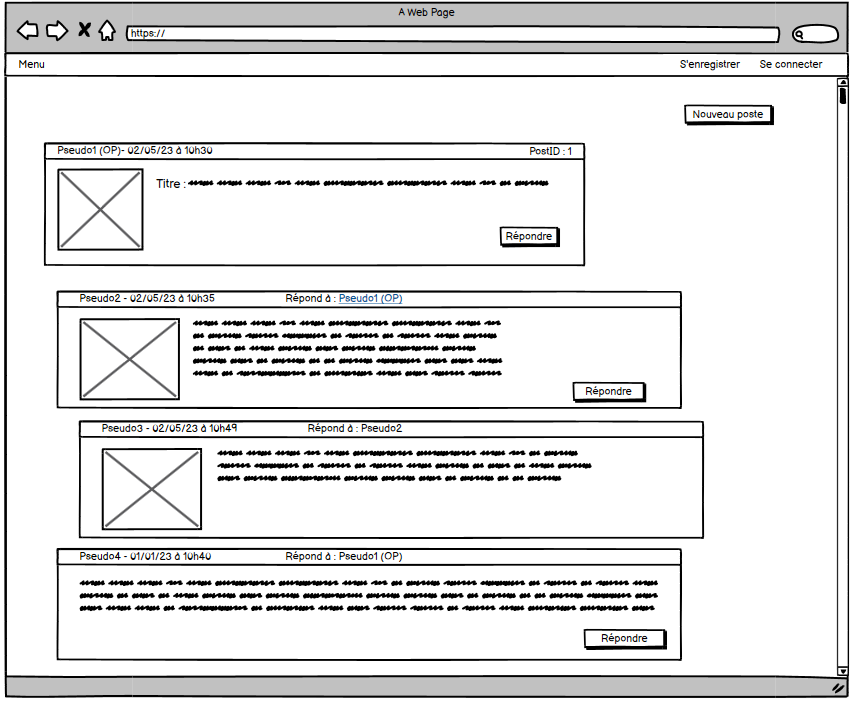
**Maquette 1 - Page d'accueil**



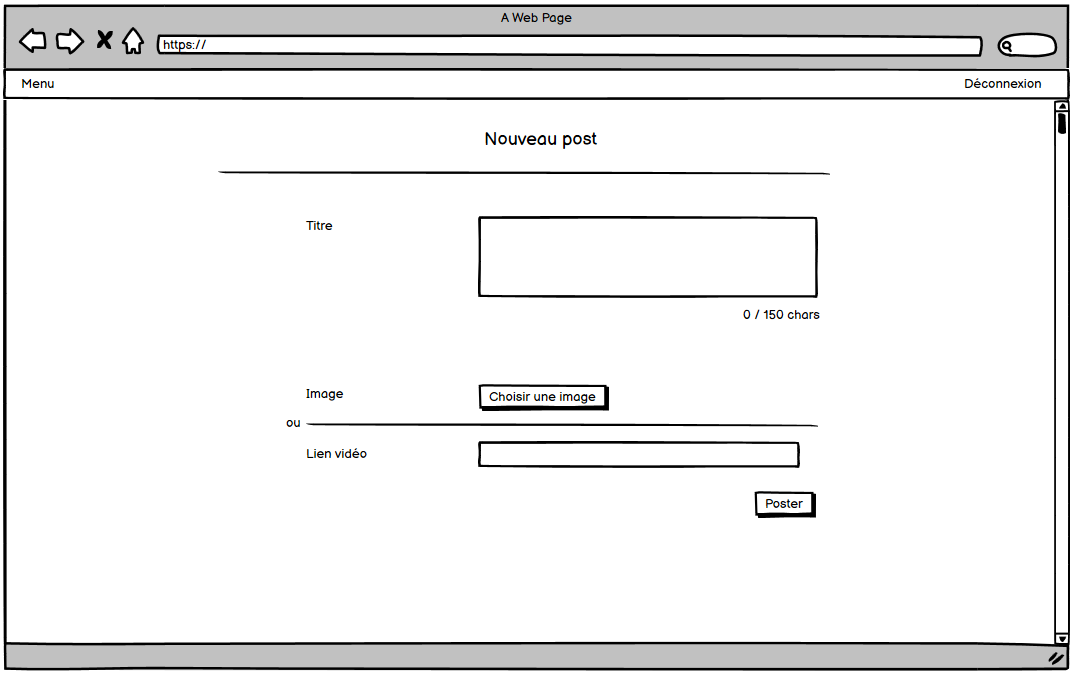
**Maquette 2 - Page d'accueil**



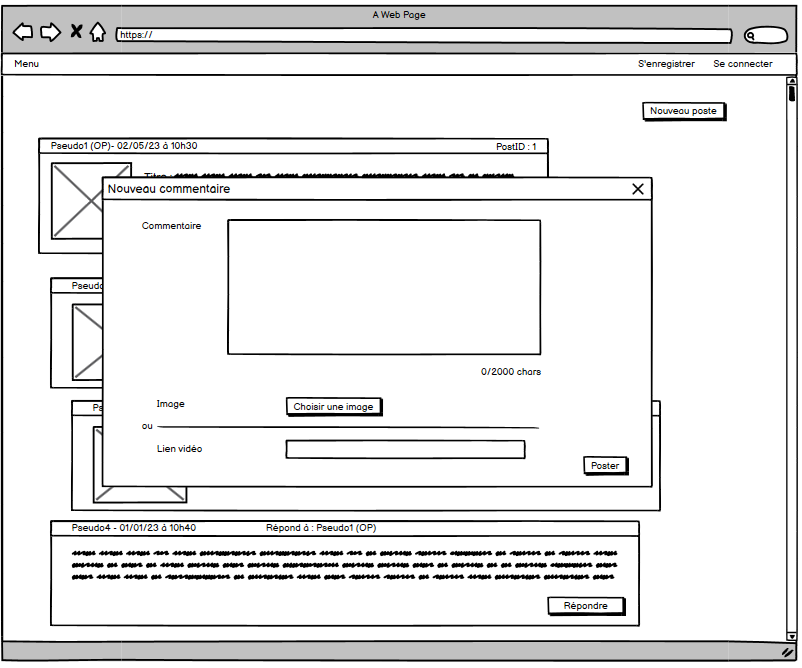
**Maquette 3 - Page de connexion**



**Maquette 4 - Post avec ses commentaires**



**Maquette 5 - Page de création d'un nouveau post**



**Maquette 6 - Menu pour poster un commentaire**

## Stratégie de test

*Décrire la stratégie globale de test:*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?).*
* *les testeurs extérieurs éventuels.*

### Tests fonctionnels

Les tests fonctionnels sont réalisés à la fin de chaque sprint afin de valider la réalisation des stories dans IceScrum

Chaque story possède une liste de tests qui garantit que la fonctionnalité est correctement implémentée. Par exemple : si un utilisateur fourni un nom d’utilisateur trop long pour la base de données pendant la création de son compte, un message d’erreur s’affichera.

### Tests d’acceptation

## Risques techniques

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

### Outils pour la réalisation

Hardware et OS : Un ordinateur du CPNV (Dell Optliplex 7040) avec Windows 10

Logiciels :

* IDE : Visual Studio Code
* Client DB : HeidiSQL
* MCD : Draw.io
* MLD : MySQL Workbench
* Maquettes : Balsamiq WireFrames
* Gestion Git : Github

Languages et Framworks :

* PHP 8.2
* TailwindCSS 3.3.2
* DaisyUI : 2.51.6

### Base de données

Cette partie contient une description ainsi qu’une justification de la longueur de certains champs du MLD.

**Users** :

* **Email** : 254 charactères, c’est la longueur maximale d’une adresse mail selon le RFC 5321
* **Username** : La plupart des sites utilisent une limite de 15 charactères.
* **Password**:255 charactères, c’est la longueur recommandée pour un mot de passe hashé selon documentation php de la fonction de hashage *PASSWORD\_HASH* (la fonction qui sera utilisé dans ce projet).
* **Register\_token** : Ce champ contient une chaîne de charactères unique utilisé pour la confirmation de l’inscription d’un utilisateur. Le token est fait de 20 charactères hexadécimaux générés aléatoirement. Cela donne un nombre de 16^20 tokens possibles.
* **Admin** : Booléen, sert à savoir si un utilisateur a des droits administrateurs.
* **Banned** : Booléen, défini si un utilisateur est banni. Un utilisateur banni ne pourra plus poster ou commenter.
* **Confirmed** : Booléen, défini si le compte utilisateur est confirmé.

Ces champs booléens sont de type « tinyint » dans le MLD car il n’y pas de type booléen sur MySQL Workbench, la valeur sera booléenne 0 ou 1.

**Posts** :

* **Title** : 150 charactères, cela semble être une taille correcte pour un titre
* **Image\_link** : Ce champ contient soit un lien vers une vidéo Youtube soit un lien vers une image stockée

Par le serveur. Les deux paragraphes ci-dessous expliquent pourquoi 45 charactères suffisent.

* + Lien vidéo : Un lien de vidéo YouTube est toujours long de 43 charactères car l’URL ne change pas (32 charactères) et l’ID des vidéos et toujours de 11 charactères de longueur
    - Exemple lien Youtube : *https://www.youtube.com/watch?v=xxxxxxxxxxx*
  + Lien d’image : Le répertoire ou sont stockés les images uploadées par les utilisateurs est (relatif à la racine du site) : «. */view/content/posts\_img/* ». Cela fait 25 charactères pour le chemin. Le nom du fichier fait de 10 charactères hexadécimaux aléatoires plus l’extensions (4-5 charactères). Cela fait en tout 40 charactères au maximum
* **Closed** : Défini si le post est ouvert, signalé, archivé ou banni
  + 0 =ouvert
  + 1 = Signalé
  + 2 = Banni
  + 3 = Archivé

**Comments** :

* **Text** : 2000 charactères, cela semble être une taille correcte pour un titre
* **Image\_link** : Ce champ contient soit un lien vers une vidéo Youtube soit un lien vers une image stockée
* **Banned**: Comme le « Closed » :
  + 0 = ouvert
  + 1 = signalé
  + 2 = banni
  + 3 = archivé

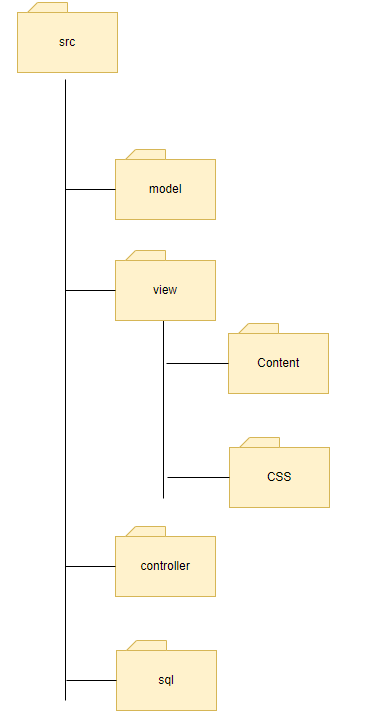
# Réalisation

## Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

### Description des répertoires

**

Le code est organisé selon le modèle MVC

Le code du projet est contenu dans **src/**

**model/** Contient les fonctions communiquant avec la base de données. Ces fonctions sont organisées dans des fichiers en fonction

**view/** contient les fichiers *.php* utilisés par le controller pour afficher les pages du site. Il contient aussi 2 sous-dossiers :

* **Content/** Contient les images publiées par les utilisateurs
* **CSS/** Contient les fichiers CSS utilisés par les pages dans **view/**, En particulier le fichier généré par TailwindCSS contenant toutes les classes Tailwind utilisées.

**controller/** contient le fichier controller dans lequel existent toutes les fonctions pouvant être appelées par le fichier index.php

**sql/** Contient le script de création de la base de donnée du site ainsi que le script d’insertion de données d’exemples et du compte administrateur.

### Description des fichiers

Model :

* **categoryManager.php** : Contient les fonctions utilisées pour la gestion des catégories
* **commentManager.php** : Contient les fonctions utilisées pour la gestion des commentaires
* **dbConnector.php** : Contient les fonctions utilisées pour la connexion avec la DB et l’exécution de commandes SQL.
* **fileManager.php** : Contient la fonction utilisée pour la gestion des fichiers uploadés depuis le site (images des posts).
* **postManager.php** : Contient les fonctions utilisées pour la gestion des posts
* **userManager.php** : Contient les fonctions utilisées pour la gestion des utilisateurs

View :

* **addCategory.php** :
* **closed.php** :
* **commentComment.php** :
* **gabarit.php** :
* **home.php** :
* **login.php** :
* **lost.php** :
* **newPost.php** :
* **post.php** :
* **register.php** :
* **reported.php** :
* **user.php** :

Controller :

* **controller.php** :
* **mailManager.php** :

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

Toutes les fonctionnalités demandées dans le CDC ont été implémentées.

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

*https://stackoverflow.com/questions/386294/what-is-the-maximum-length-of-a-valid-email-address*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

### Installation DB

Le script de création de la base de données est « *src/sql/creation\_db.sql* »

## Manuel d'Utilisation