30 mai 2022

Quentin Chassaing

Titre de développeur web et web mobile

Projet Verse

(uncensored edition)

Sommaire :

1. Présentation (abstract)
2. Présentation du projet
3. Le REAC
4. Cahier des charges
5. Spécifications fonctionnelles
6. Conception
7. Réalisation du projet
8. Présentation des fonctionnalités
9. Conclusion
10. **Présentation (abstract)**

Hi, I’m Quentin, I’m 30 years old, I live in a small town so called Figeac. Kind of small, but also pretty.

After my high school diploma, I didn’t know what to do. I was kind of lost in front of the possibilities, after high school it's a huge door that opens, and I never took the time to think about what to do next.

I turned to a degree in administration and management, followed by a master’s degree in territorial development that I haven’t finished. That was a turbulent period, stuck between failures and family issues.

But one day, I asked myself what I wanted to do, to be happy and fulfilled.

And a good friend of mine told me that I could be a developer. Maybe he knows me better than I do. In fact, I’m a video game enjoyer, I spend a lot of time on my computer. Actually, being behind my screen is kind of my comfort zone.

So, I decided to take a look at some online courses, tutorials or any resources you can find on internet. And I remember back then, when I wanted to be a video game developer without knowing what it really means. And I found some courses on OpenClassrooms to create a website with HTML and CSS. And that was a blast, I was hooked by how we can build a website. The langages, the possibilities, a revelation. I started looking for a training course that could give me enough knowledge and enough time to learn more langages. I’m craving for knowledge since I was a kid, but I also need to reassure myself.

And I found in Adrar formation the opportunity to make my journey into the world of developers.

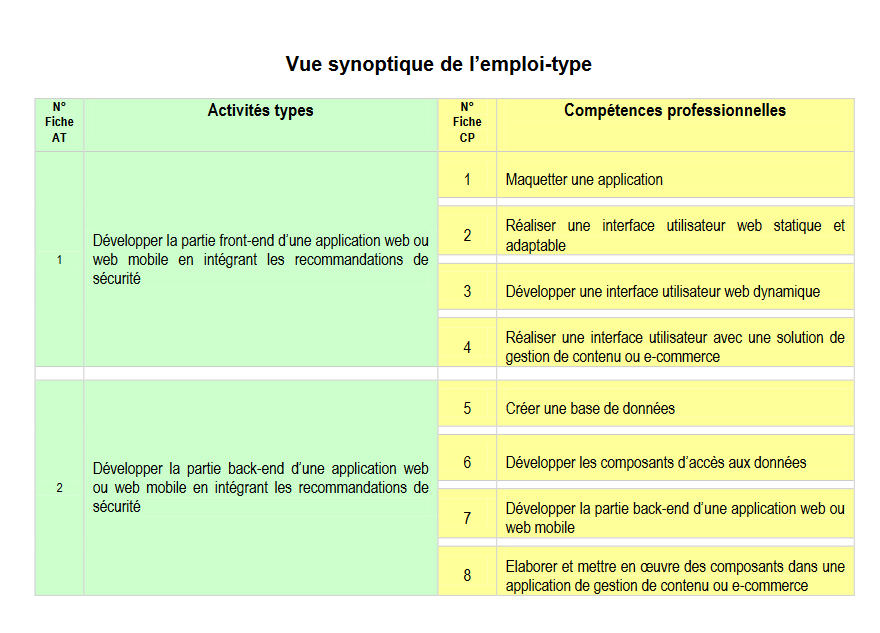
1. **Présentation du projet**

Maintenant passons à la présentation du projet, je suis entré à l’Adrar, avec une envie, un besoin. Il fallait que je puisse faire grandir mes compétences, améliorer mes connaissances qui étaient alors bien maigres en comparaison à aujourd’hui.   
Lorsque qu’on nous a parlé du projet fil rouge, je me suis alors demandé qu’est ce qui pourrait me porter et m’animer suffisamment pour que je puisse apprendre tout en m’amusant.

Le cerveau humain est plus apte à comprendre les choses si on décide de le divertir en même temps, et inconsciemment l’information s’implante bien mieux si on ressent une certaine satisfaction en réalisant quelque chose qui nous anime.

C’est ainsi, en suivant ce raisonnement que nous en arrivons à notre projet, la plateforme **Verse.**

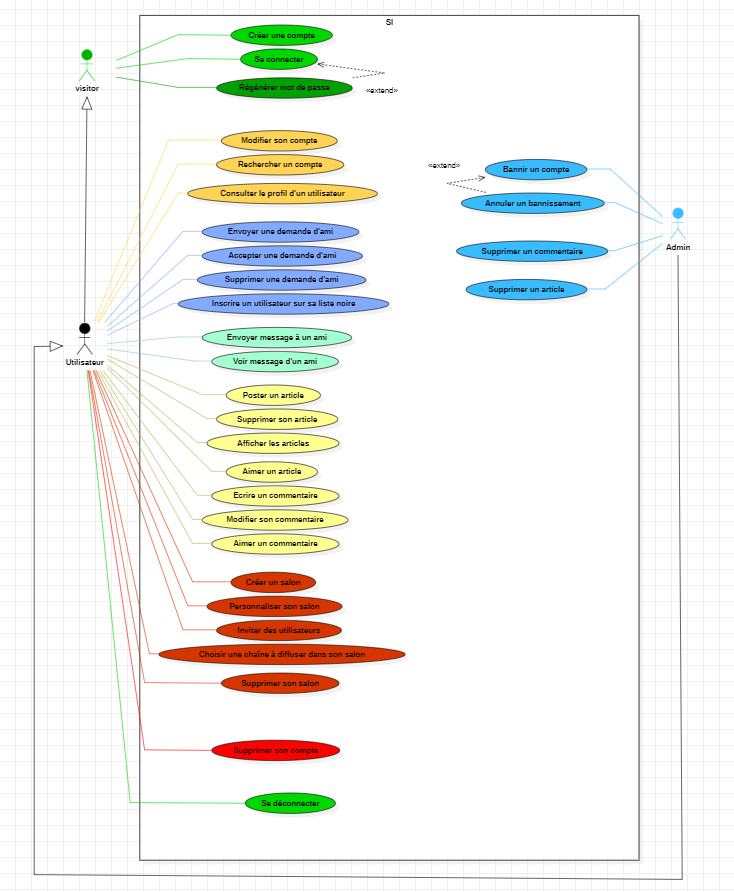
1. **Le REAC**



1. Le maquettage du site s’est fait rapidement suite à la conception du site, il faut dire que dès le départ j’avais opté pour une interface plutôt épurée et classique pour permettre à l’utilisateur de se repérer facilement dans sa navigation sur le site, sachant que l’espace de ce dernier fonctionne comme une dashboard à partir de laquelle il peut gérer toutes les fonctionnalités qui lui sont offertes sur le site.
2. Le développement du site est en majeure partie statique et responsive, l’utilisation de Bootstrap permet, en majeure partie l’adaptabilité des divers éléments affichés dans le front du site, cependant l’utilisation de média queries n’a pas été exclu afin de permettre l’adaptation sur téléphone portable. Pour le moment l’adaptation sur les nouveaux portables fold est en cours de développement.
3. Une partie du contenu du site est quant à lui géré dynamiquement par mon Php, je pense particulièrement aux informations de l’utilisateur qui sont récupéré dans le back-end afin que chaque utilisateur ait la sensation d’avoir son propre espace, il voit son contenu et les informations qu’il nous a envoyé. L’affichage des postes des utilisateurs est lui aussi géré dynamiquement, une fois qu’un utilisateur a créé son article ou post, il est affiché sur la page principale de tous les utilisateurs.
4. Afin de réaliser ce site, la base de données était cruciale, alors elle a été développée avant toutes fonctionnalités. En me basant sur le MCD/MLD et MySQL, la base de données est adaptée à l’accueil du contenu qui fera apparition sur le site, que ce soit un nouvel utilisateur, du contenu qu’il publiera ou même plus tard des salons qu’il ouvrira et pourra personnaliser comme bon lui semble.
5. Afin de relier notre base de données et notre back-end, le modèle de la connexion à la BDD et les modèles de chaque table ont été réalisé avec Php. Dans chacun des modèles, plusieurs fonctions ont été développé afin de pouvoir préparer chaque requête qui seront envoyés à la base de données et récupéré par notre back-end.
6. En plus des modèles, les contrôleurs de nos fonctionnalités ont été développé avec Php, pour terminer la boucle et entretenir cette communication du front à la base de données en passant par le back-end.
7. **Cahier des charges**

Notre entreprise se retrouve donc missionnée par le grand du streaming Twitch qui décide de mettre en place un nouveau concept, une plateforme sur laquelle nous pourrons voir l’émergence de communautés, l’idée est de donner une notion supplémentaire d’appartenance aux individus qui suivent régulièrement le contenu diffusé sur Twitch.

On aura la possibilité de partager, visionner avec nos amis, rencontrer des gens qui partagent les mêmes centres d’intérêt et de ce fait créer une sorte d’écosystème digital en dehors du site.

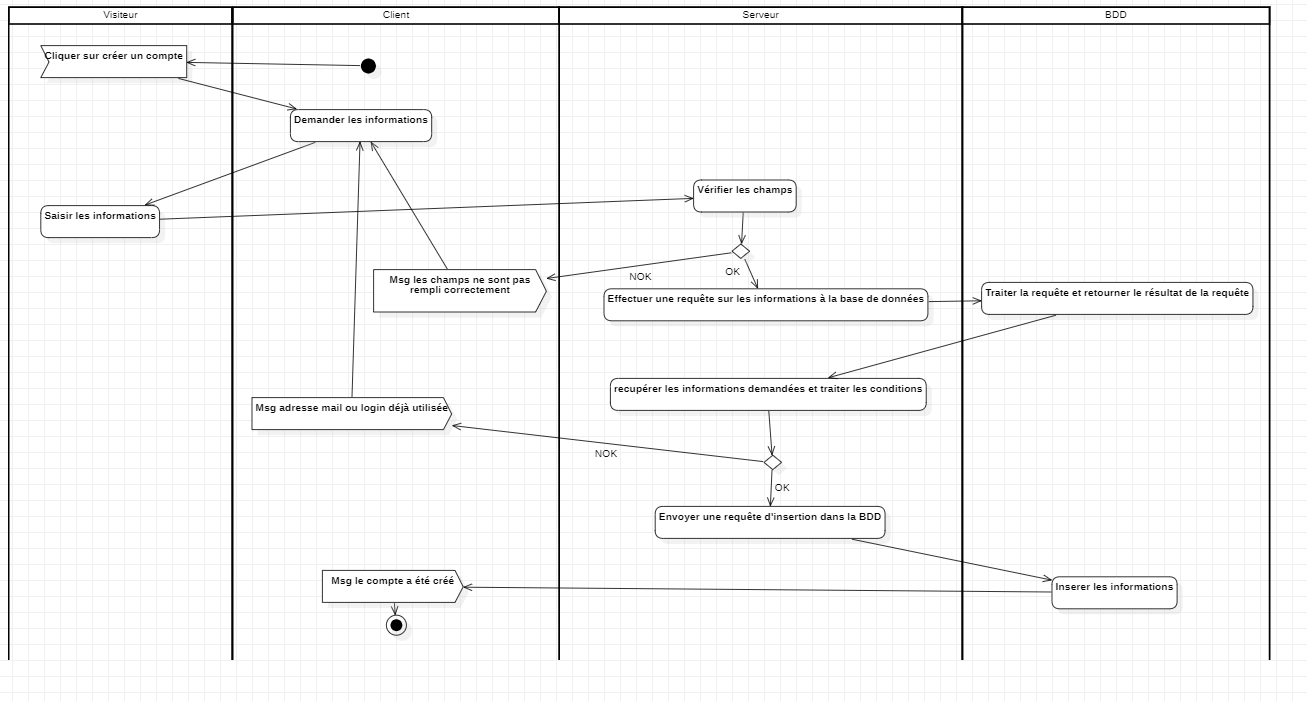
1. **Spécifications fonctionnelles**
   1. **Use Case**

Le Use Case comme on peut le voir va venir mettre en exergue les fonctionnalités qui seront à la disposition des différents acteurs, en plus des fonctionnalités propres à chacun, on peut voir qu’il y a un héritage qui se produit entre tous. Au final seul l’administrateur va posséder des actions qui lui seront propre. A contrario, le visiteur lui sera très limité en termes de droit sur la plateforme. Il devra alors créer un compte afin de pouvoir être upgradé au rang d’utilisateur et bénéficier pleinement du site.

* 1. **Diag activité**

Le Diagramme d’activité va nous permettre de mettre en place le raisonnement derrière chaque fonctionnalité. Il sera le fil directeur dans la réflexion derrière le développement

* + - **La création d’un compte**



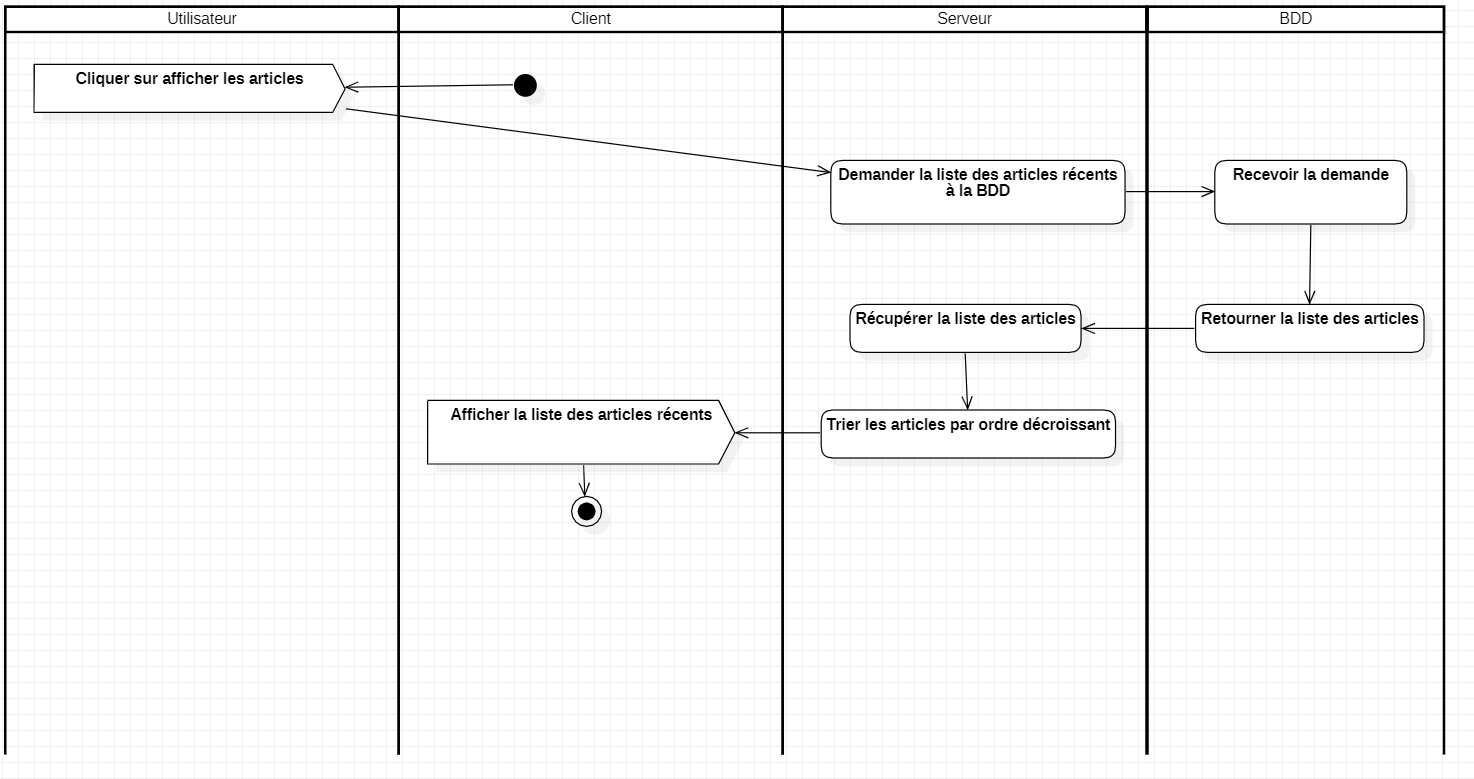
Dans la conception du diagramme d’activité de la création d’un utilisateur, feature propre au visiteur.

On peut noter qu’il est requis une action afin de lui ouvrir l’accès à la création du compte. Une fois l’action réalisée, on entame le processus de création. Il devra renseigner les différents champs nécessaire et obligatoire à l’insertion de son compte. Durant le processus, des vérifications seront effectuées afin de déterminer si certaines de ses informations sont ou non déjà présente dans la base de données.

Le but étant de ne pas avoir de doublon sur les informations sensibles pouvant nuire à l’intégrité d’un compte.

Une fois toutes ces actions effectuées. Le visiteur possède un compte et est élevé au rang d’utilisateur.

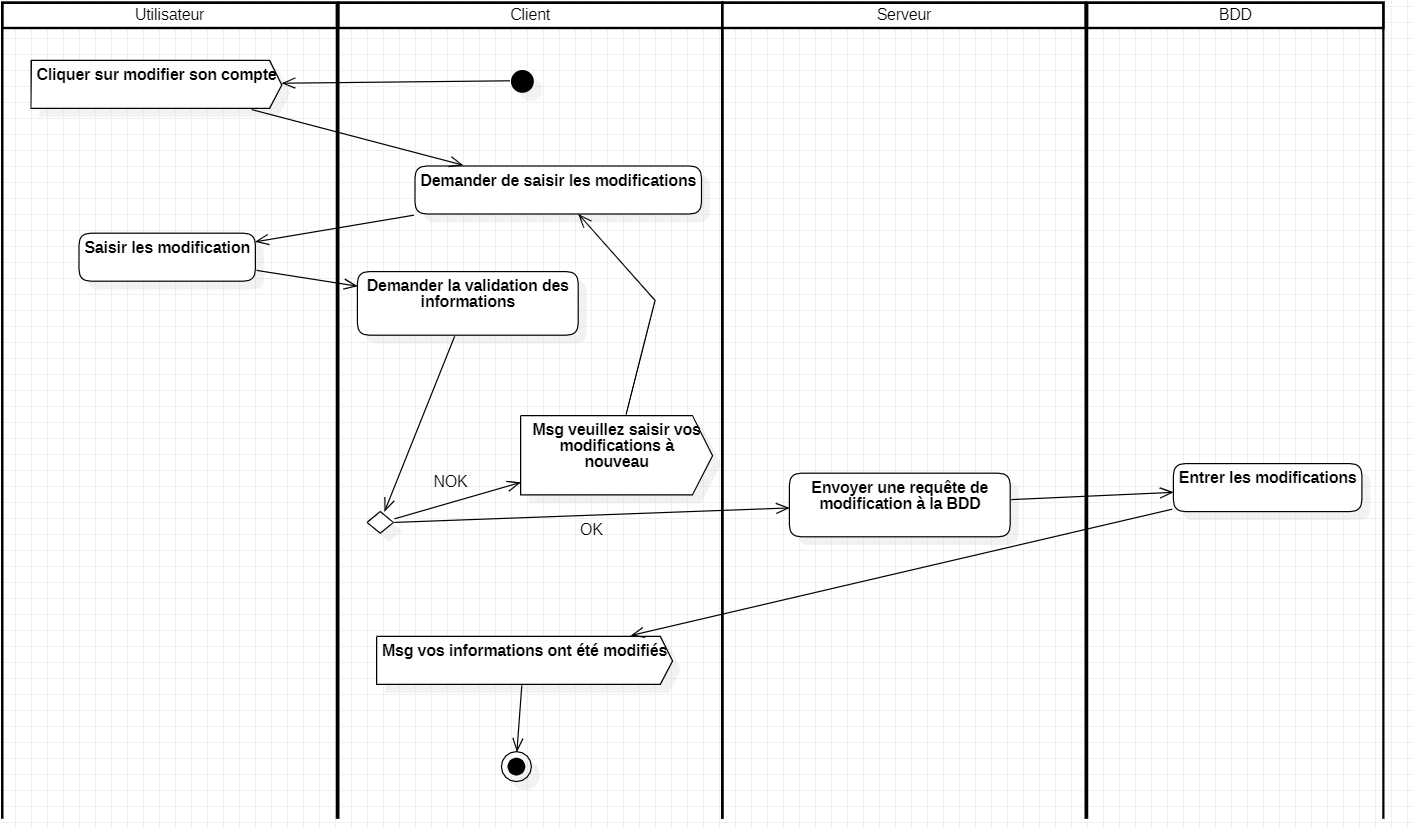
* + - **L’affichage des articles**



L’affichage des articles lui devait suivre un principe simple, il était souhaité que ces articles que l’on a appelé communément des « posts » soient accessibles de tous. En reprenant cette idée de réseau social, il était intéressant que les différents membres du site puissent partager tous ensembles et, par la suite réagir à divers contenus provenant de tous les utilisateurs.

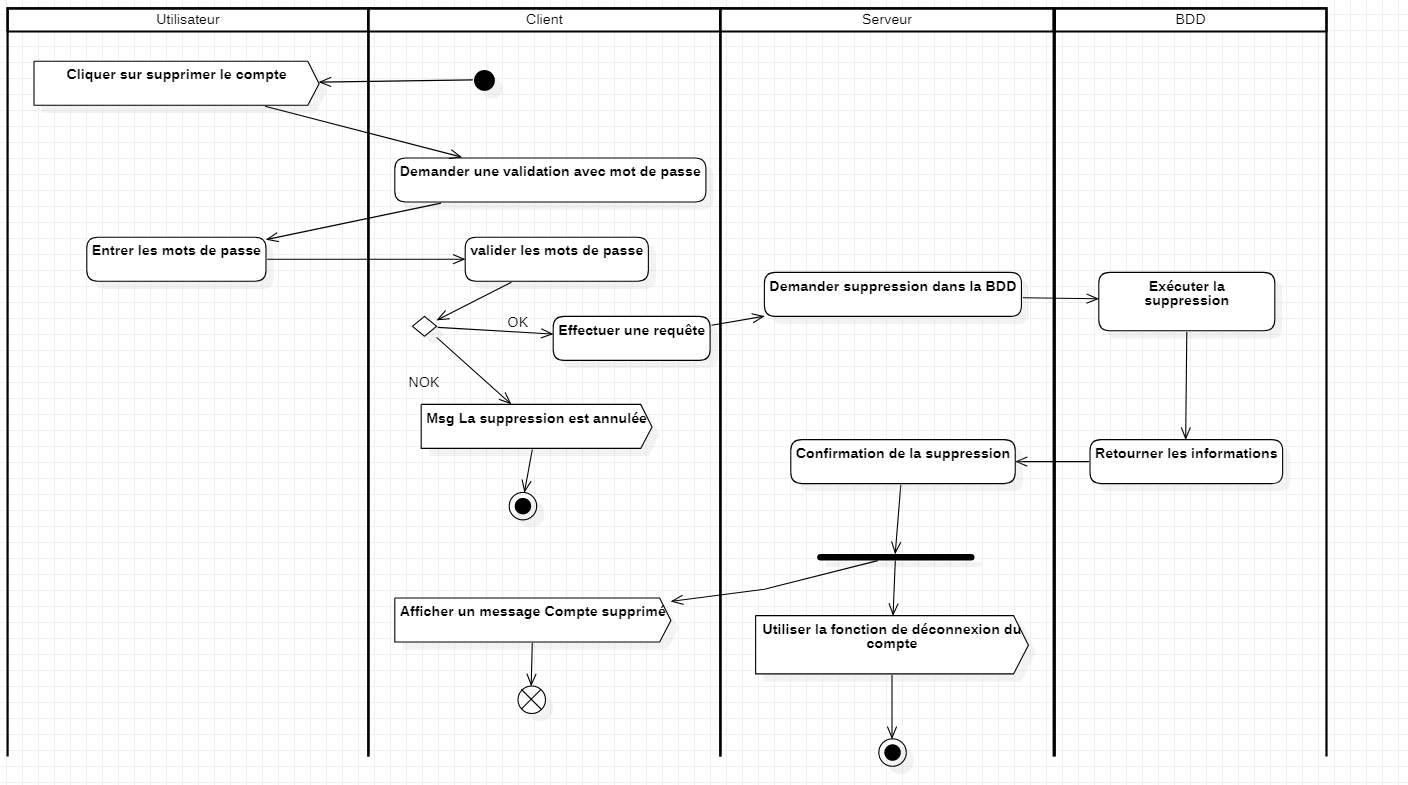
Une fois l’utilisateur sur sa page principale, il peut accéder à tous les articles qui sont classés par ordre chronologique du plus récent au plus ancien. On demande donc à notre backend d’effectuer une requête pour récupérer les articles que l’on trie et retourne à notre front.

* + - **La modification du compte**



Afin de pouvoir modifier son compte, l’utilisateur va devoir accéder à une page où il peut trouver toute ses informations. Et, à partir de là lui est offert la possibilité de modifier ses informations. Dans un rendu général, la modification de compte se déroule de la sorte. Des informations concernant la modification lui seront demandés, par la suite l’utilisateur devra entrer les nouvelles informations et les anciennes dans un processus de sécurisation de l’opération. Le client procèdera alors la mise à jour des nouvelles informations de l’utilisateur. Et l’utilisateur sera notifié du succès de l’opération.

* + - **La suppression d’un compte**



L’utilisateur a toujours la possibilité de supprimer son compte s’il le souhaite. Nous nous sommes tournés vers le principe de suppression complète du compte.

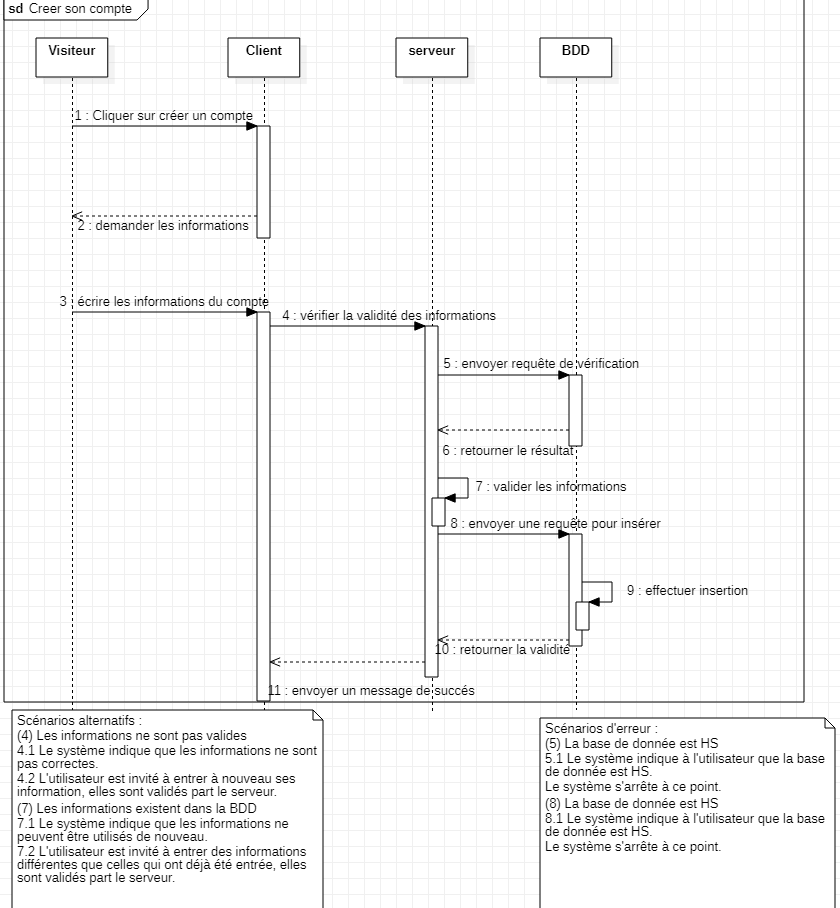
Si l’utilisateur fait la démarche de cliquer sur la suppression de son compte. Il sera alors envoyé vers une fenêtre où il devra entrer son mot de passe deux fois afin de vérifier qu’il est sûr de sa décision. Une fois ceci fait notre back va vérifier si le mot de passe correspond bien à cet utilisateur et va alors valider ou non l’opération.

Si l’opération est bien valide, la requête de suppression est envoyée à la base de données qui va supprimer la ligne correspondante à cet utilisateur. L’utilisateur est alors supprimé et devient à nouveau un visiteur avec les droits qui en découle.

* 1. **Diag de séquence**

Alors qu’il est très proche du diagramme d’activité, le diagramme de séquence quant à lui possède une particularité et pas des moindre, il va nous servir de feuille de route pour évaluer au mieux les scénarios alternatifs et les scénarios d’erreurs. Encore une fois, un diagramme nous permettant d’améliorer le développement de chaque fonctionnalité en prenant compte des difficultés pouvant être rencontrés lors de leur utilisation.

* + - **La création d’un compte**



Je vais accorder une importance particulière aux nouveautés qu’apporte ce diagramme de séquence. On y voit donc l’apparition des scénarios alternatifs et d’erreurs. Du côté des scénarios alternatifs, on va relever l’invalidation des informations. Ce scénario-là peut survenir de diverses façons.

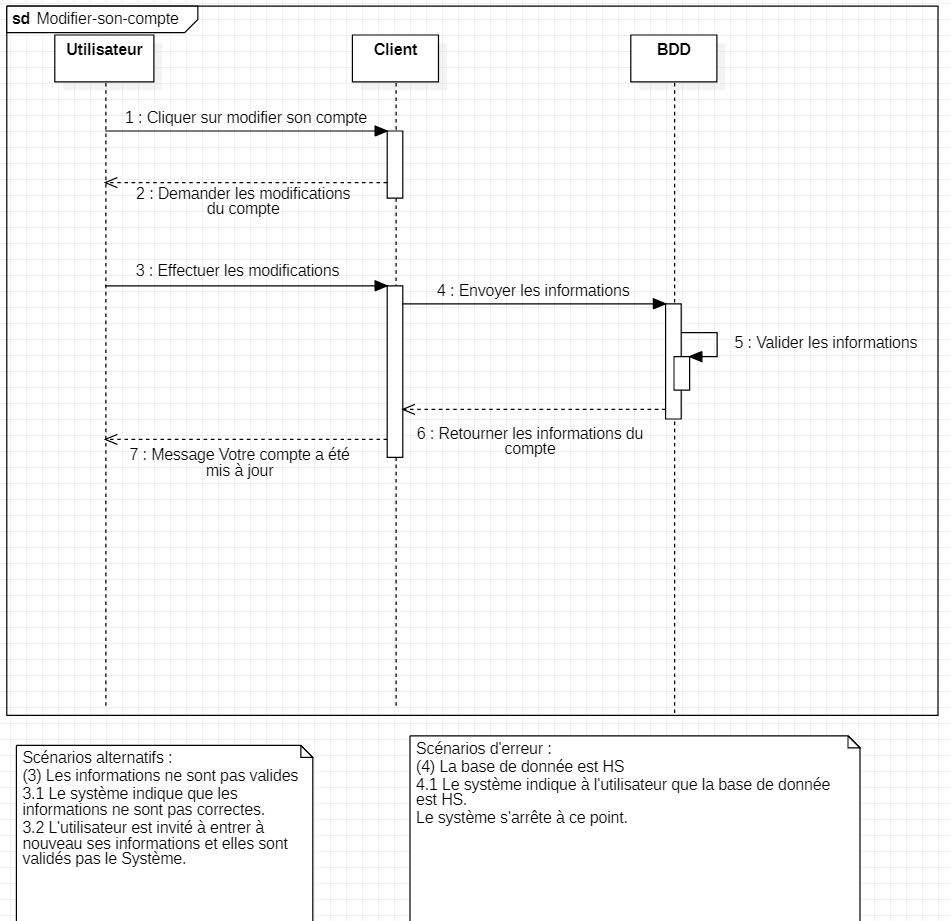
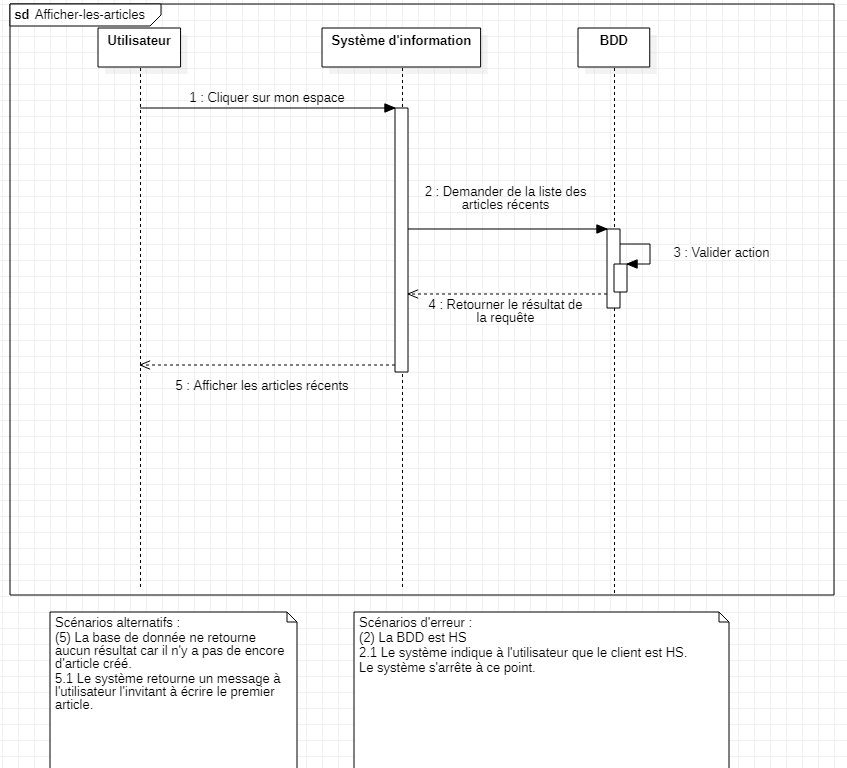
Cela peut correspondre à des caractères employés qui sont invalides, un nombre de caractères ne correspondant à ce que nous allons imposer, des caractères qui ne sont pas employables, des champs manquants.

Dans un deuxième temps, il s’agit d’un scénario qui survient lors de la vérification côté serveur, la base de données nous indique qu’une information du champ existe d’ores et déjà dans la base de données ce qui entraine un retour à l’étape d’inscription des champs côté client.

En termes de scénario d’erreur on en a relevé un seul. Il s’agit de la base de données qui est injoignable, cet évènement peut survenir deux fois lors de la création d’un compte utilisateur.

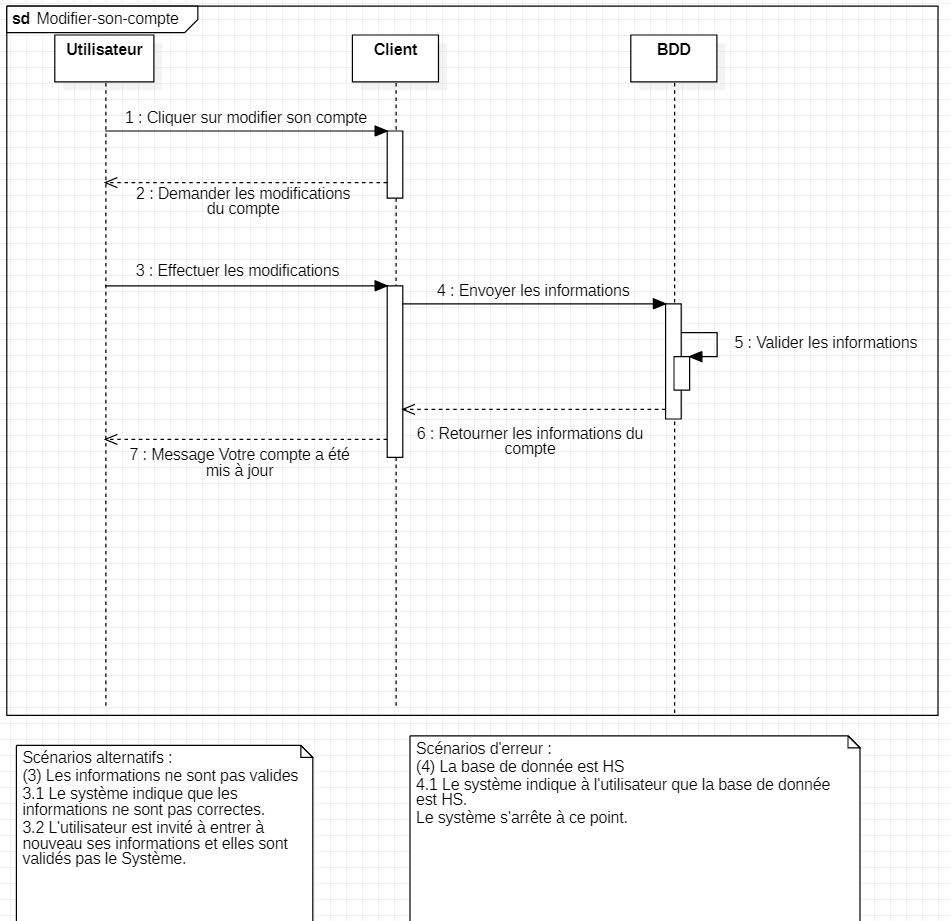
* + - **L’affichage des articles**

Lors de la réalisation de ce diagramme, un seul scénario alternatif nous est venu à l’esprit. L’absence d’article et l’impossibilité d’afficher un article sur l’écran de l’utilisateur. Afin de pallier à cet éventualité, l’idée d’émettre un message qui prendra la place des articles en attendant qu’un premier post soit publié.



Ici on retrouve encore une fois le scénario d’erreur qui survient lors de la requête qui doit normalement se charger de récupérer et trier tous les articles en base de données. Sauf que cette dernière n’est pas accessible.

* + - **La modification d’un compte**

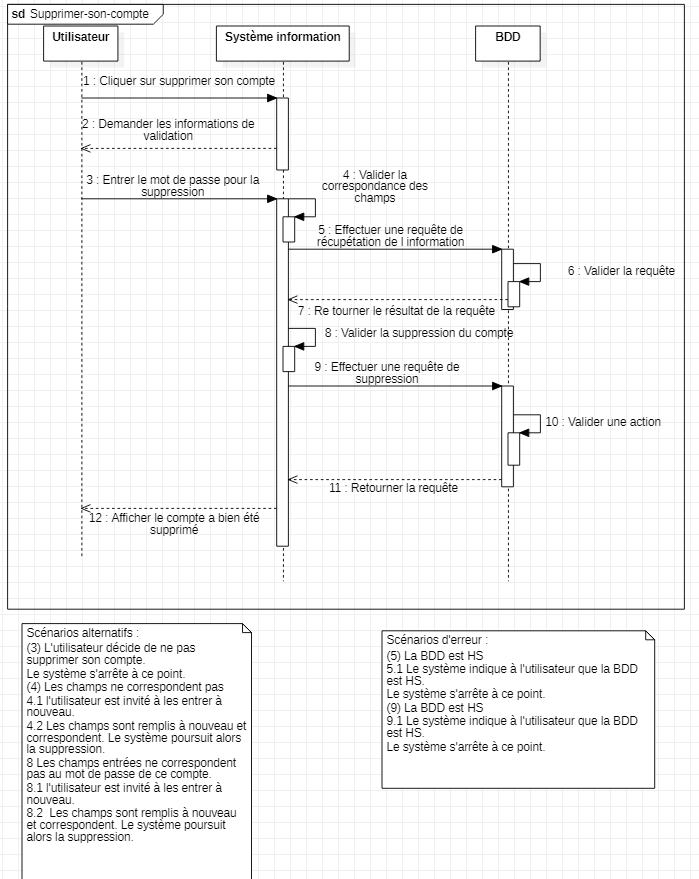


Lors de la modification du compte on rencontre un scénario alternatif qui est que les informations entrées par l’utilisateur ne correspondent pas.

Il s’agit ici de vérifier que les informations du compte ne sont pas erronées. C’est une sécurité supplémentaire pour s’assurer au maximum qu’une tierce personne ne change pas les informations du compte à l’insu de son utilisateur.

Lors de la mise à jour des informations de l’utilisateur, on a relevé un possible scénario d’erreur nous renvoyant à une base de données qui est inaccessible et met fin au système.

* + - **La suppression d’un compte**



La suppression du compte sous la forme du diagramme de séquence doit permettre de relever si l’utilisateur est en train de supprimer son compte où s’il a cliqué ici par erreur.

On cherche avant tout à sécuriser cette suppression qui est avant tout définitive.

On va prendre en compte plusieurs scénarios alternatif avec des fins diverses.

Dans un premier temps, on a l’utilisateur qui décide de faire machine arrière est arrête de lui-même le processus.

S’il entre ses informations et qu’elles ne sont pas correctes, le système devra le notifier qu’il y a un problème de champs qui sont différents. L’utilisateur sera alors invité à entrer à nouveau les informations.

Par la suite après vérification dans la base de données, il peut lui être retourné que le mot de passe utilisé lors de la suppression du compte ne correspond pas au sien. Il s’agit là d’une dernière sécurité afin d’empêcher au maximum qu’une suppression puisse être effectuée par une autre personne.

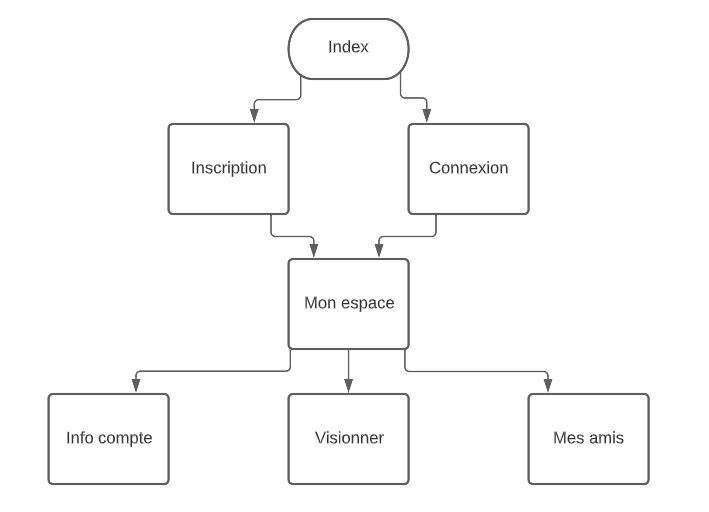
* 1. **Maquettage**

Tout d’abord la présentation de l’organigramme du site. Dans sa conception nous partons d’un visiteur qui arrive sur le site.

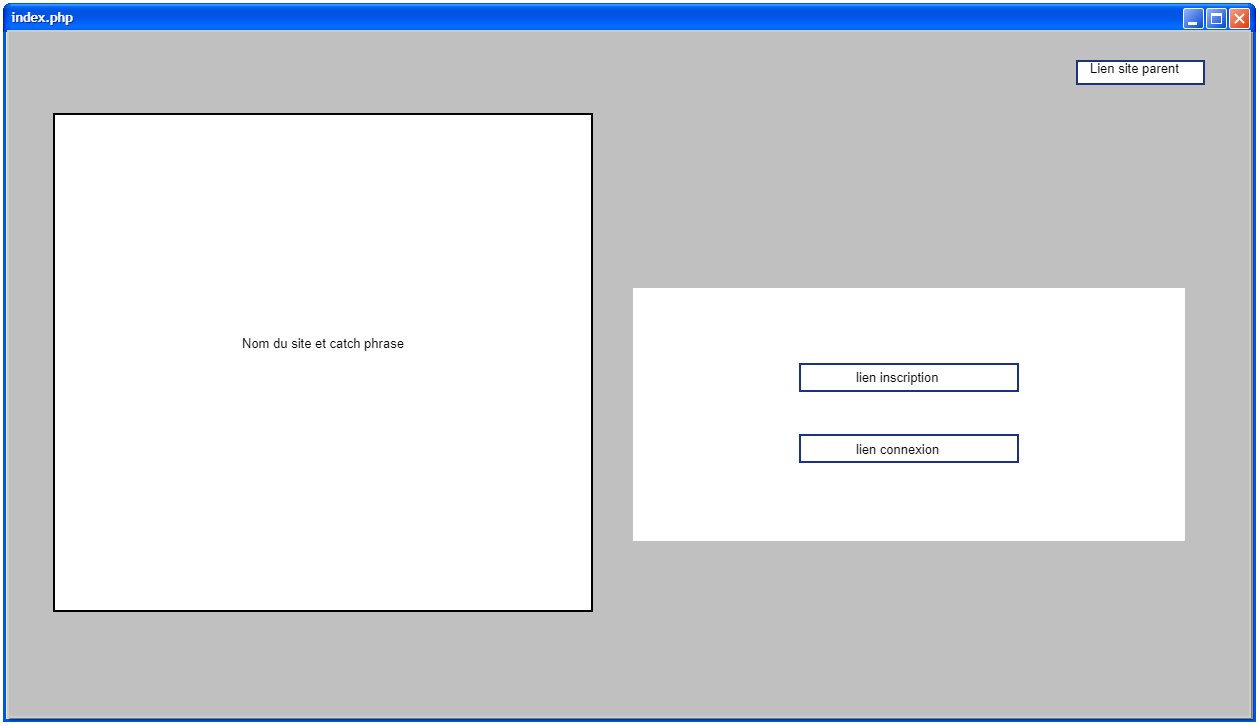
Il se retrouve sur la première page qui nous sert à filtrer qui est un utilisateur et qui est un visiteur.

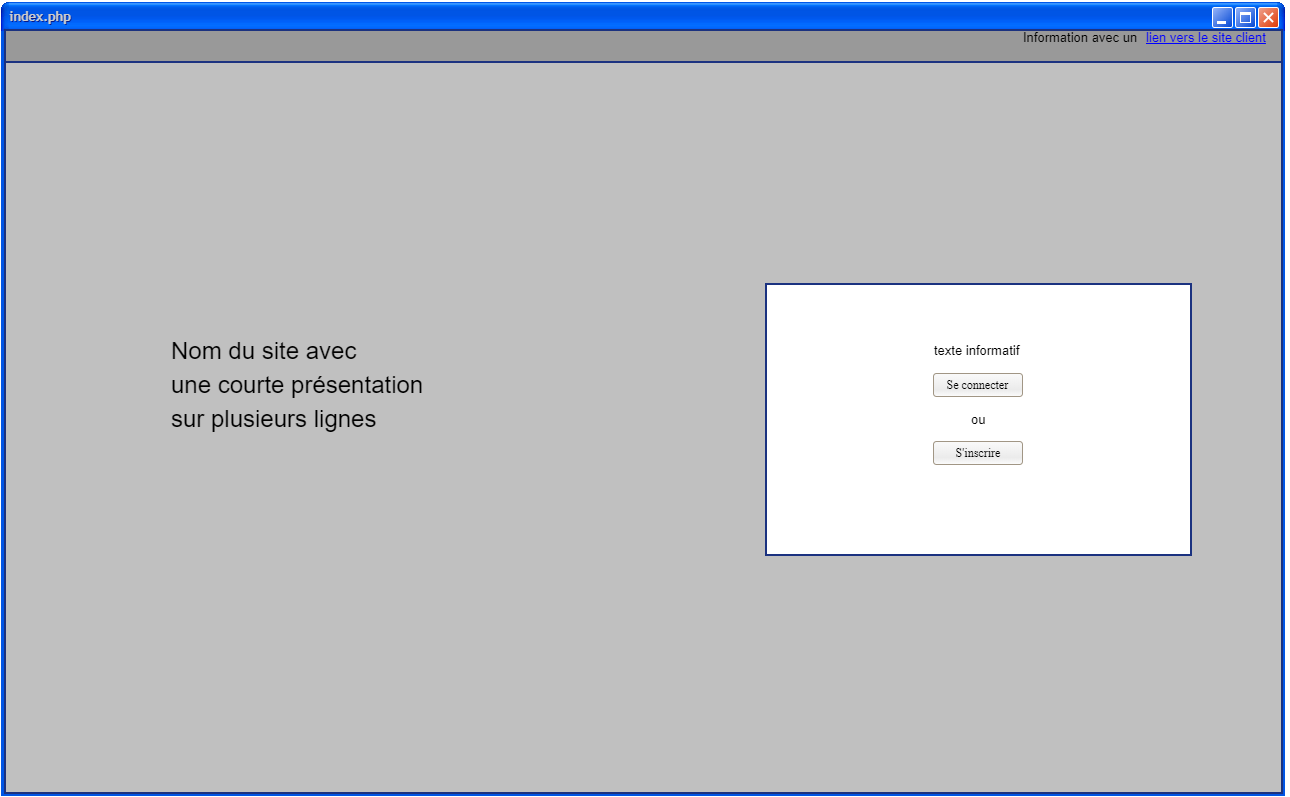
Nous offrons la possibilité de se connecter à l’utilisateur, et, à contrario la possibilité de créer un compte si l’on est visiteur.

Dans les deux cas, les acteurs seront une fois la connexion/inscription terminée redirigé vers leur espace à partir duquel ils auront la possibilité d’accéder à toutes les fonctionnalités du site.



* + - **Les zonings et wireframes**

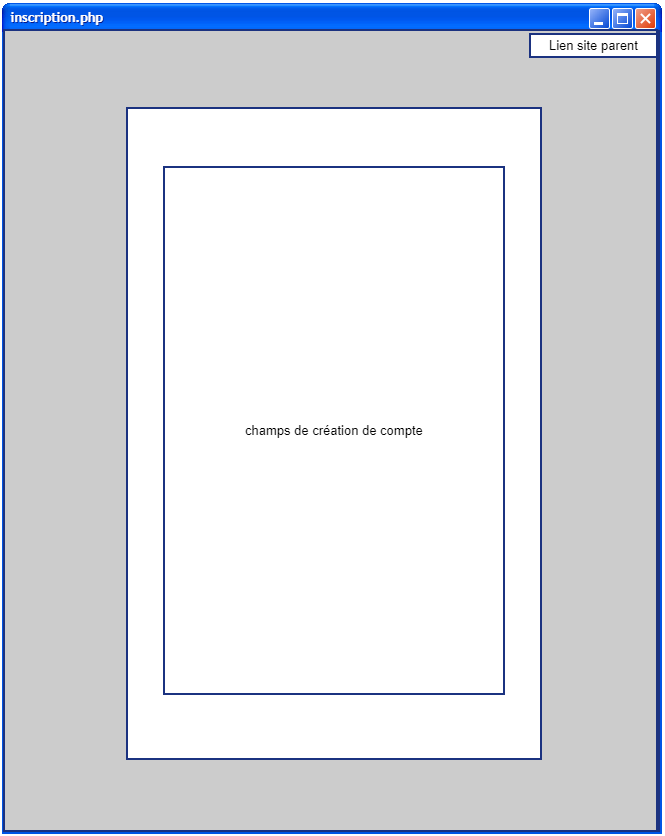


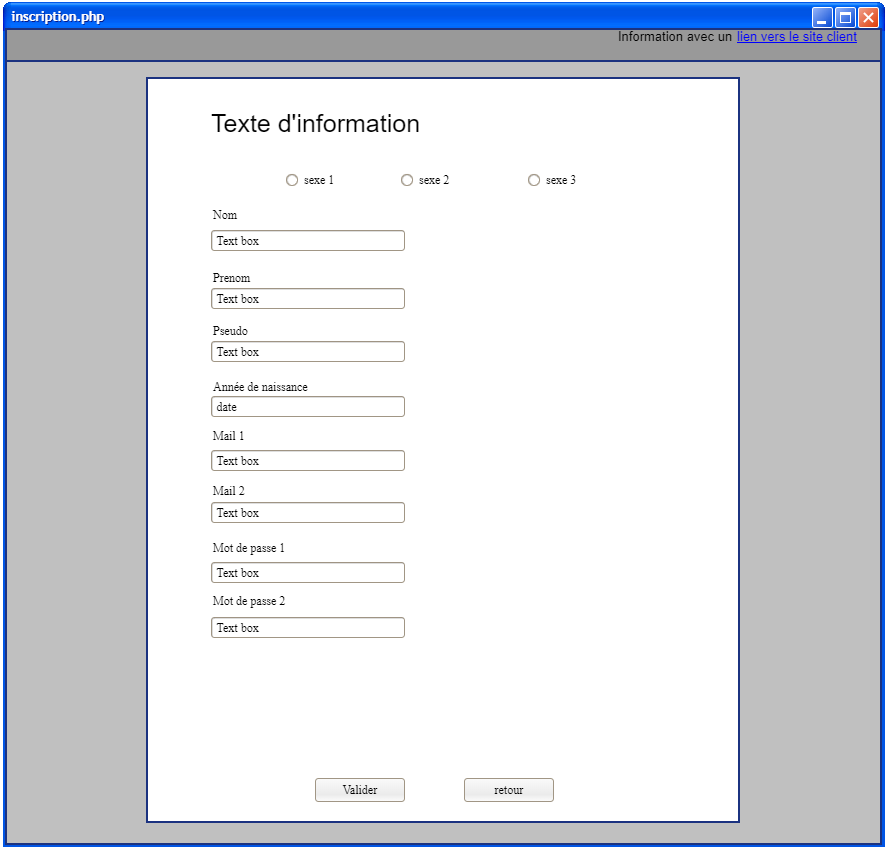


La page d’accueil arbore une maquette très simple. Peu d’informations doivent y figurer pour seule directive, qui devra par la suite se retrouver sur les autres pages du site, un lien vers le Twitch. Ensuite pour ce qui est de la mise en forme de la page il va falloir conserver de l’espace pour obtenir un design minimaliste.

Seuls deux boutons figurent sur cette page afin de permettre à l’utilisateur de savoir où il doit se tourner pour progresser sur le site.

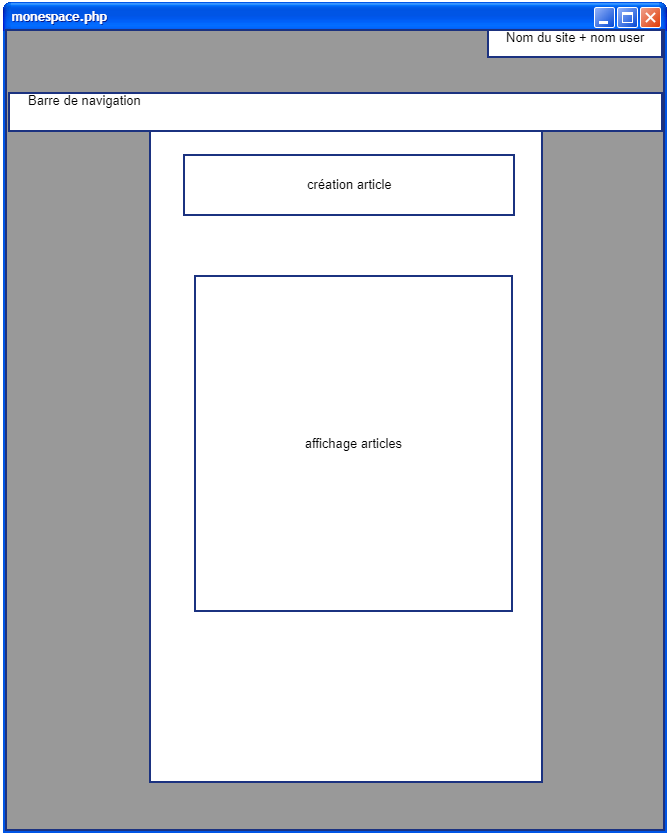
On va suivre le cheminement d’un visiteur qui décide de créer un compte :



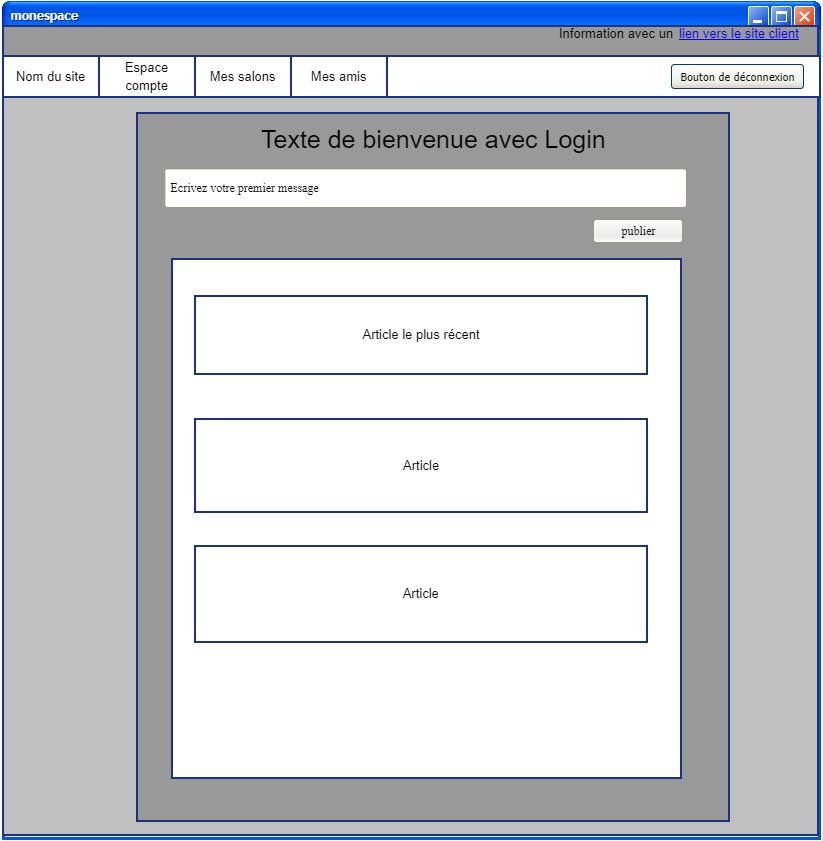


Notre visiteur cherchant à s’inscrire arrive donc sur cette page où il est invité à entrer ses informations personnelles. Pour réaliser la création de son compte.

Une fois le compte finalisé, notre anciennement visiteur est connecté à son compte utilisateur qu’il vient tout juste de créer.

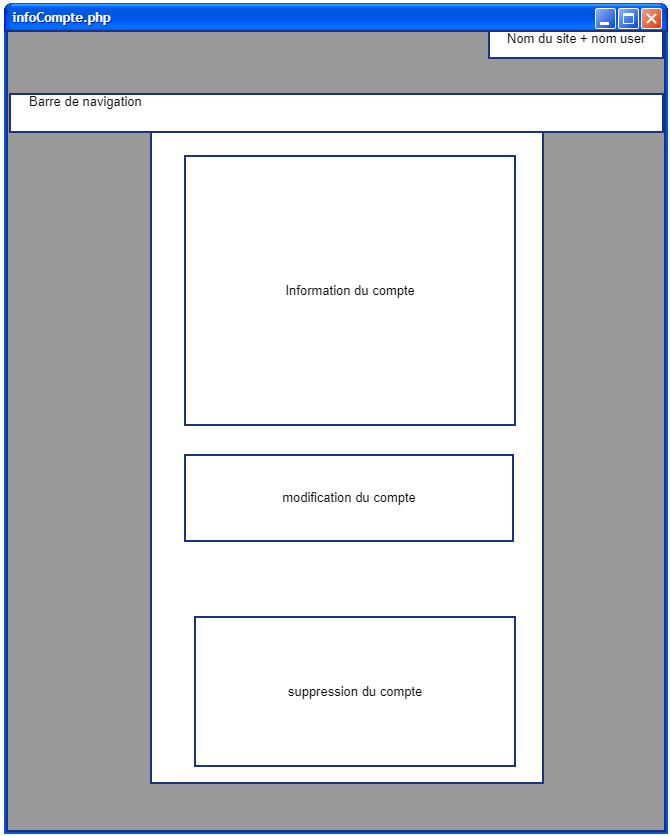


Une fois connecté notre nouvel utilisateur se retrouve sur son espace à partir duquel il a accès à toutes les fonctionnalités qui lui sont offertes par le site Verse au travers du menu de navigation. Cependant cette page est aussi celle où il a la possibilité d’écrire un article afin de partager avec les autres utilisateurs.



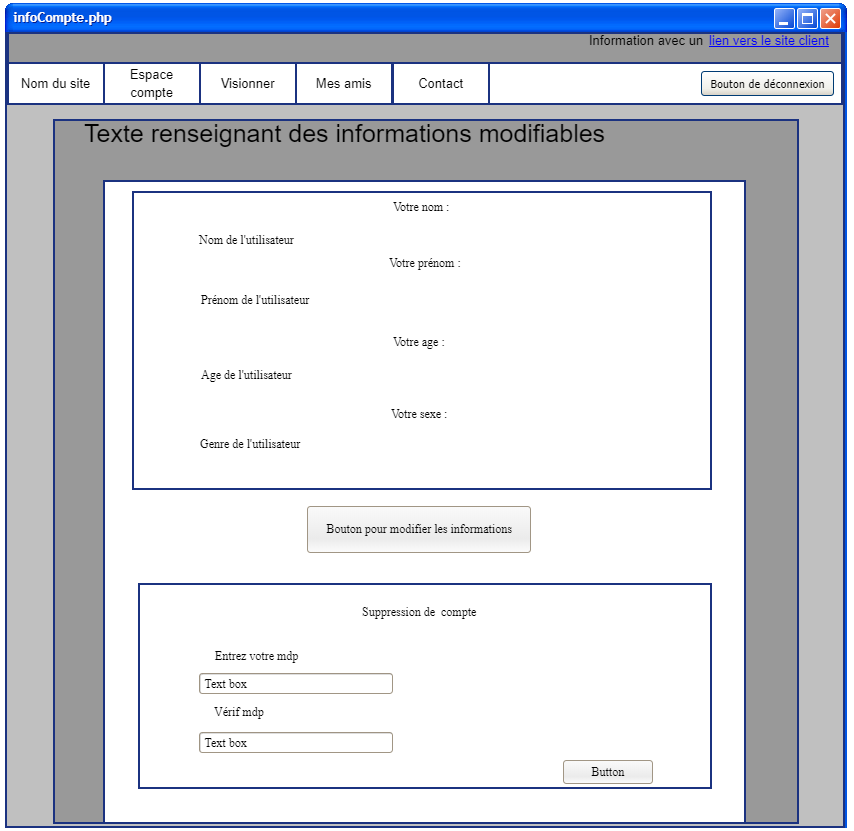
Mais c’est aussi l’endroit où il peut consulter les derniers articles poster par les autres utilisateurs.

Vient ensuite le moment où l’utilisateur désire voir ou modifier ses informations, ou même, décide de supprimer son compte :



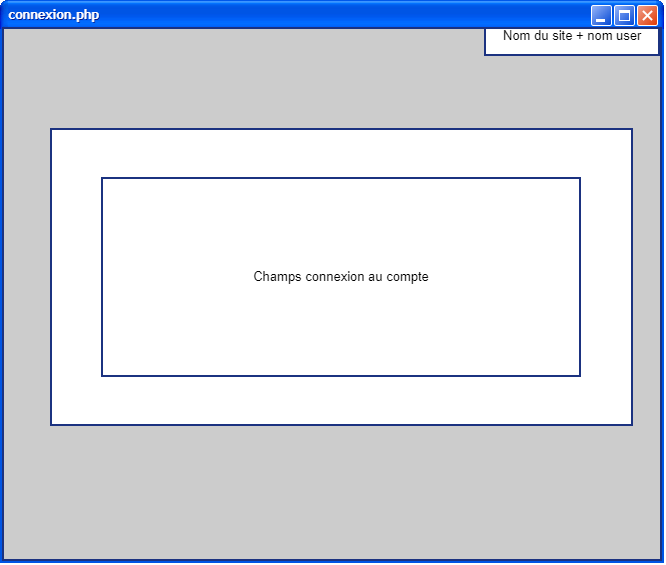
L’utilisateur se retrouve dans l’espace où il peut accéder aux informations qu’il a entré.

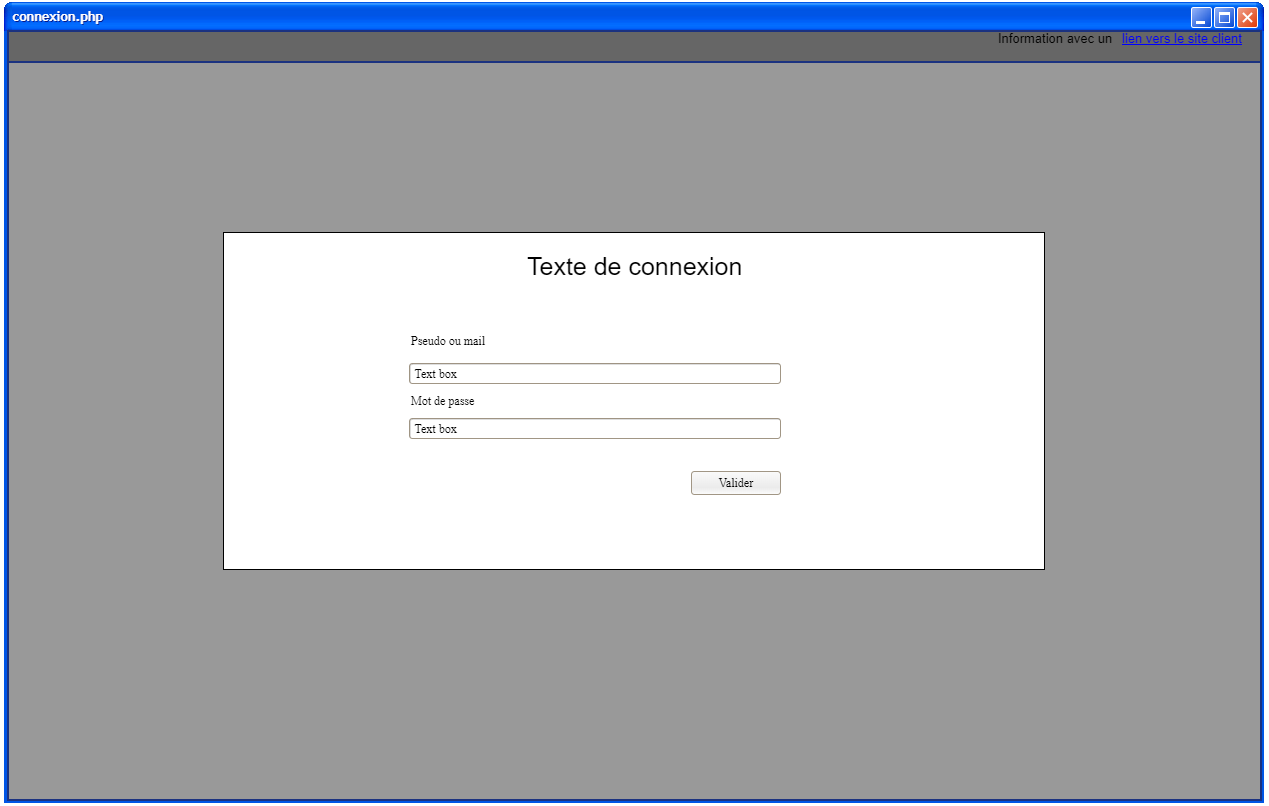
En plus de ça, il a la possibilité d’y modifier son login ainsi que son mot de passe en cliquant sur le bouton de modification qui lui ouvrira une fenêtre modale sur cette page en lui demandant d’entrer ses informations et de les valider selon qu’il souhaite modifier son login ou son mot de passe.



Et dans des cas extrêmes, il peut décider de supprimer son compte. Il lui est demandé d’y entrer son mot de passe et de le vérifier. Une fois ceci fait, son compte sera alors supprimé et il sera renvoyé vers la page d’index.

Imaginons un utilisateur, donc une personne qui possède déjà un compte, décide de se connecter au site. Sur la page d’accueil il va donc cliquer sur se connecter. Ce qui l’enverra sur la page de connexion.





L’utilisateur accède donc à la page connexion où on lui demande d’entrer ses informations. Une fois l’opération effectuée, il peut cliquer sur valider.

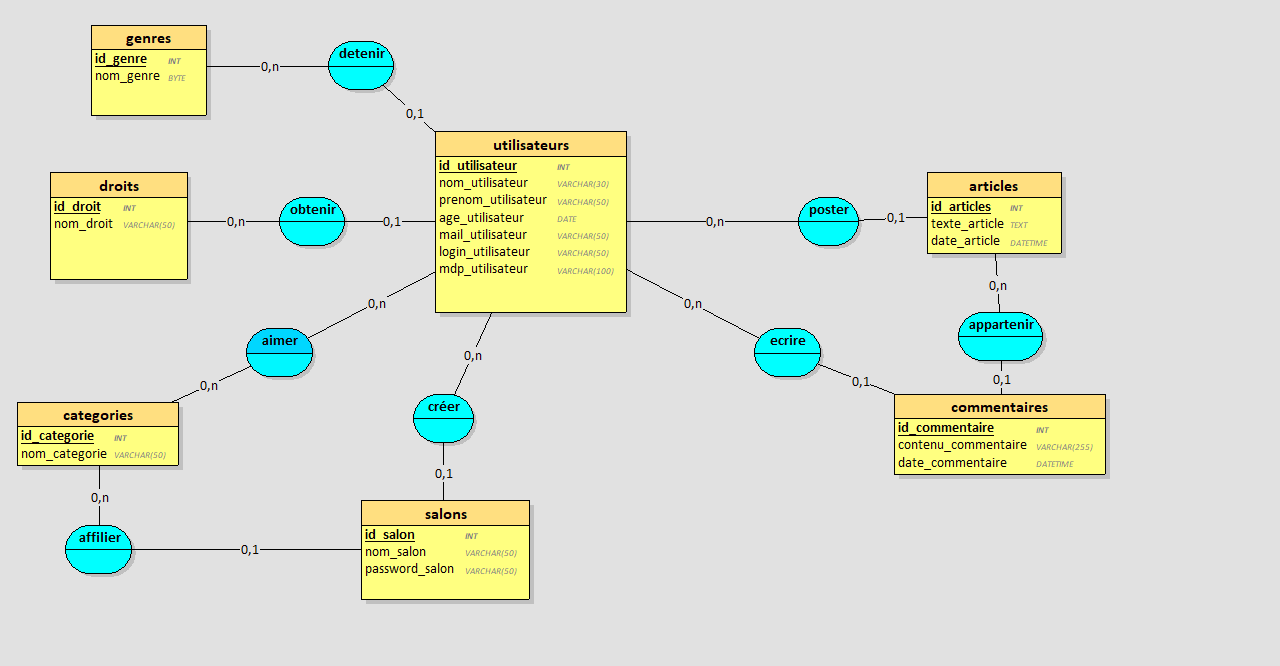
Ceci fait, il est connecté à son compte et se retrouve sur la page principale de son espace.

1. **Conception**
   1. **MCD**

Pour la conception du site internet, nous avons réalisé un MCD qui nous donne la marche à suivre pour la réalisation de la base de données.

Le MCD que je vais vous présenter ne représente qu’une partie du MCD final.

Cela permettra plus de lisibilité sur la conception des fonctionnalités qui sont déjà en place sur le site.

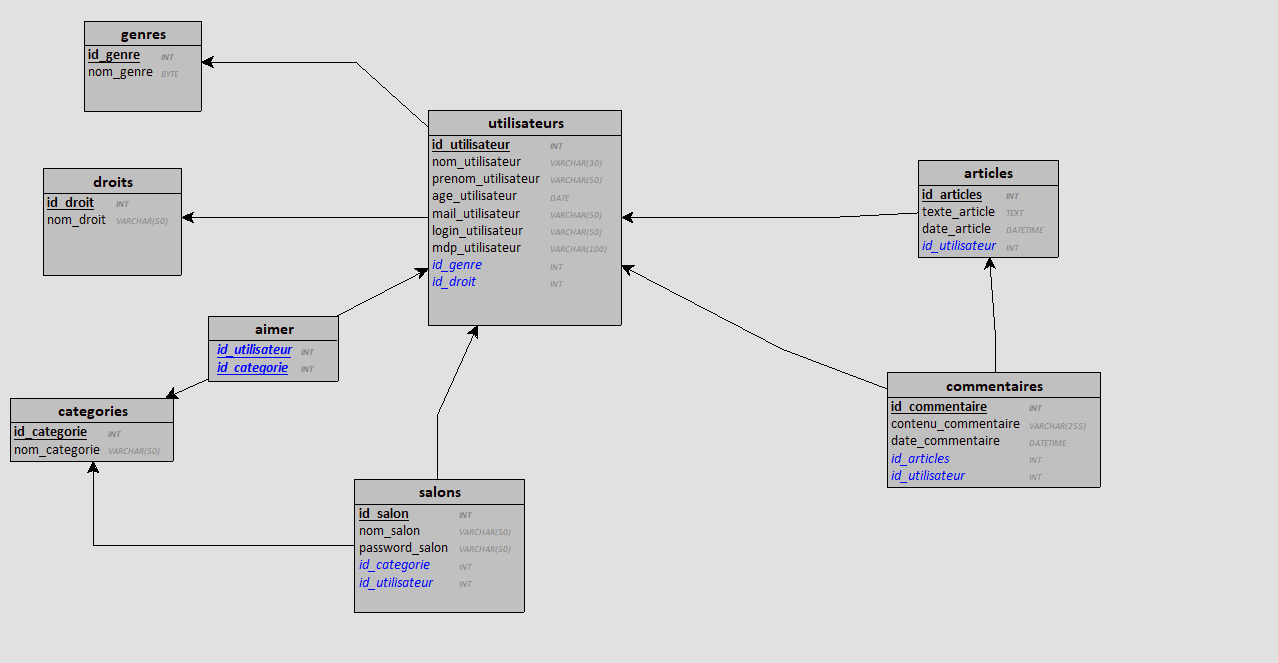


Notre MCD met donc en évidence le rôle central de l’utilisateur autour duquel toutes les autres tables vont graviter.

Nous n’avons qu’une seule table d’association sur ce modèle, entre Utilisateurs et Catégories, c’est ici le seul endroit où l’on peut retrouver des relations multiples entre les deux tables. L’utilisateur peut aimer plusieurs catégories, et les catégories peuvent se retrouver chez plusieurs utilisateurs.

Pour aller plus loin, nous allons observer le MLD pour pouvoir observer les différentes liaisons entre les tables et voir comment les clés étrangères ont été distribués entre les différents acteurs de ces modèles.

Grace au MLD on peut observer en direct les relations entre les tables, les clés étrangères étant distribué entre les tables. On voit immédiatement que la table utilisateur récupère les clés étrangères de la table Genres et Droits. On peut observer qu’entre la table Catégories et utilisateurs émerge la table d’association aimer qui récupère en clés primaires l’id\_utilisateur et l’id\_categorie.

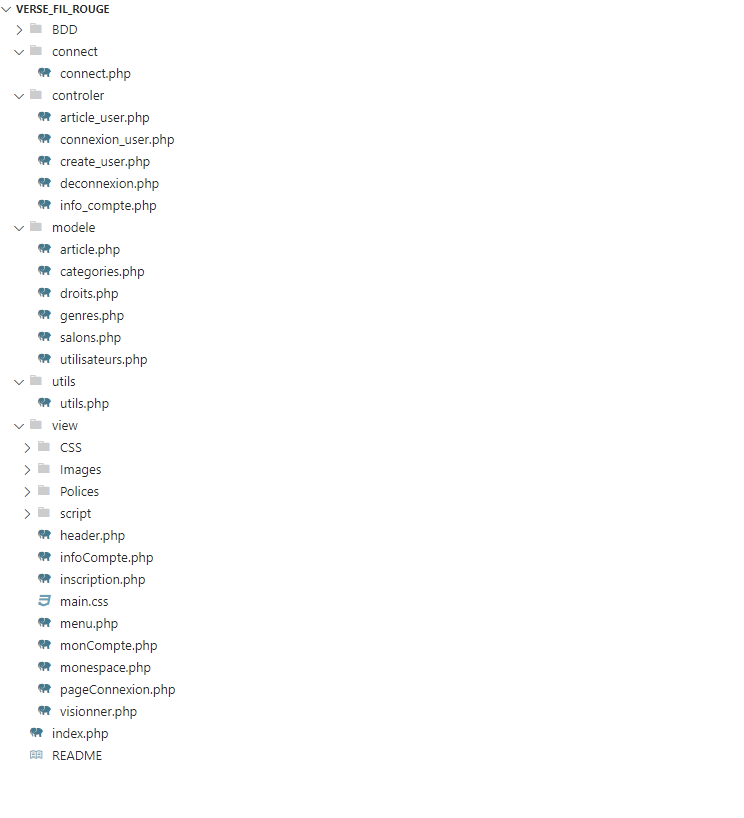


Les salons ne pouvant être créé que par un seul utilisateur vont alors récupérer son id en clé étrangère et selon la conception actuelle, les salons ne pourront avoir qu’une seule catégorie ce pourquoi elle récupère en clé étrangère id\_categorie.

1. **Arborescence des fichiers**

Pour développer le site internet de Verse, nous avons opté pour un modèle MVC afin d’obtenir une forme d’ordre à l’intérieur de nos fichiers. Et permettre une maintenance bien plus facile et adaptée au développement continu de cette plateforme.

Le travail collaboratif n’étant pas écarté par la suite, il était important, dans la conception, de pouvoir aussi miser sur le futur du site.



Afin de rendre les fichiers plus lisibles et bien plus reconnaissables j’ai donc refactorisé les différentes fonctions du site.

1. **Outils techniques Utilisés**
   1. **Le Front-end**

Pour la partie architecture de la page le squelette et la mise en forme des éléments trois langages entre dans la danse accompagnée d’un Framework.

Tout d’abord on a HTML5, qui vient s’occuper du squelette de la page. Chaque élément concentré sous son architecture accompagné de Bootstrap qui vient permettre une première mise en forme de notre page grâce à ses librairies.

Le coup de pinceau de CSS3 qui vient améliorer le rendu visuel, la mise en page de notre site, lui aussi accompagné de Bootstrap.

Et enfin Javascript qui fait une petite apparition avec l’affichage d’une fenêtre modale.

* 1. **Le Back-end**

Pour la base de données nous avons PhpMyAdmin, notre application qui nous a accompagné dans la gestion de cette dernière. Le langage qui nous a servi dans la conception de la BDD est MySQL. (Voir annexe\_1)

Pour ce qui est des fonctionnalités présentes sur le site, la gestion des utilisateurs, l’affichage dynamique et la communication entre le client et la base de données, PHP.

* 1. **Le versionnage**

Sans oublier, bien sûr, GitHub qui m’a accompagné tout au long du projet pour les sauvegardes de toutes les étapes de la conception au développement du projet en lui-même.

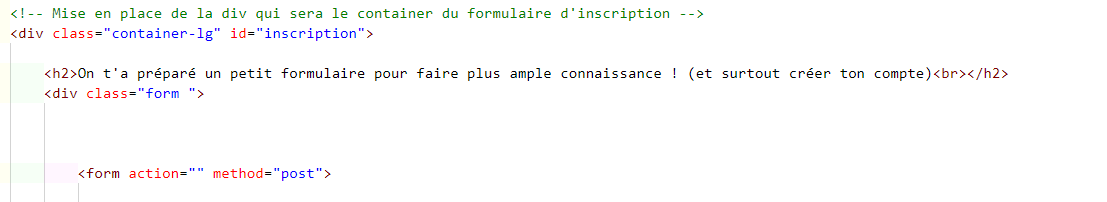
1. **Fonctionnalités**

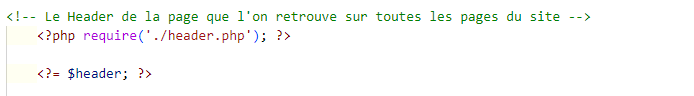
Il est maintenant temps de présenter ce qui représente le cœur de notre projet, le code. Encore un peu de patience, dans un premier temps, je vais présenter ce que le visiteur va voir lorsqu’il va cliquer sur le bouton rejoindre le site :

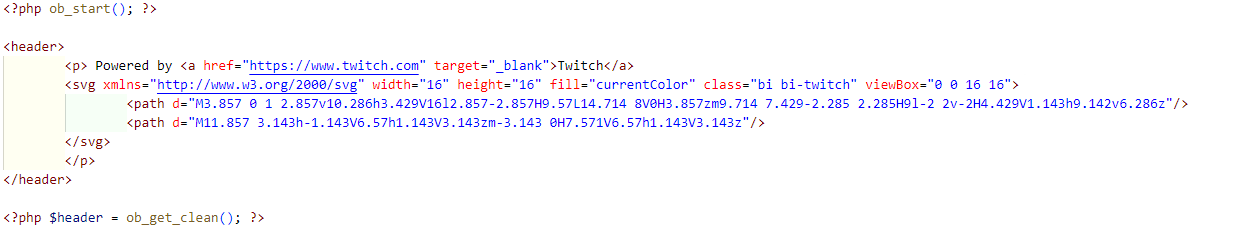
Pour le HTML, on a tout d’abord un header que l’on vient importer via du php.



On retrouve la fonction native de Php ob\_start() et ob\_get\_clean() qui permettent l’enregistrement du html que l’on entre à l’intérieur de ses deux dernières.





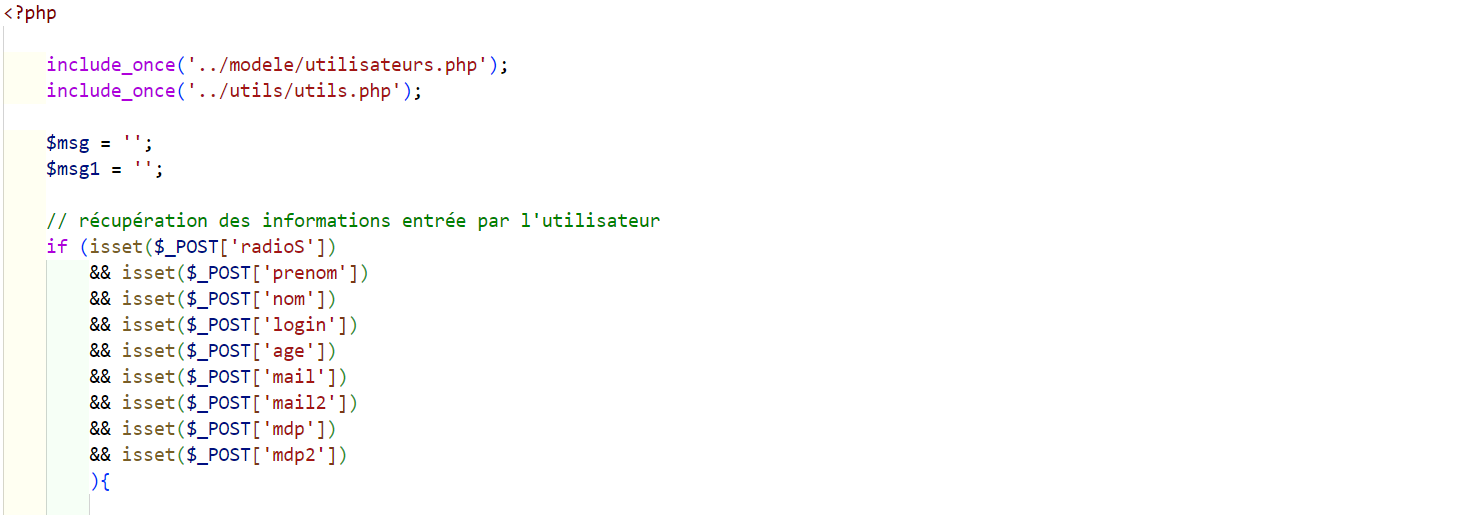


Ici on voit la balise <div> qui va englober l’intégralité du formulaire, c’est en utilisant une classe Bootstrap que je commence la mise en forme de l’ensemble du document, la classe container-lg étant une classe propre au framework.

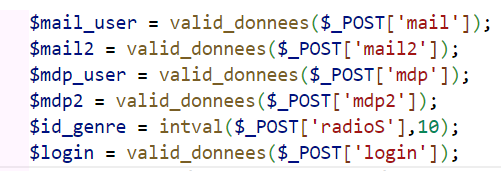
Le formulaire quant à lui, à des fins de protection des informations qui seront entrées, va utiliser la super globale $\_POST pour récupérer les informations qui seront entrées

Une fois que les champs ont été remplis et que le visiteur appuie sur le bouton valider. Que se passe-t-il ? Eh bien c’est ici que le Php va entrer en jeu.

On va directement dans notre fichier create\_user.php, le controler en charge de l’inscription alias de la création de notre nouvel utilisateur.

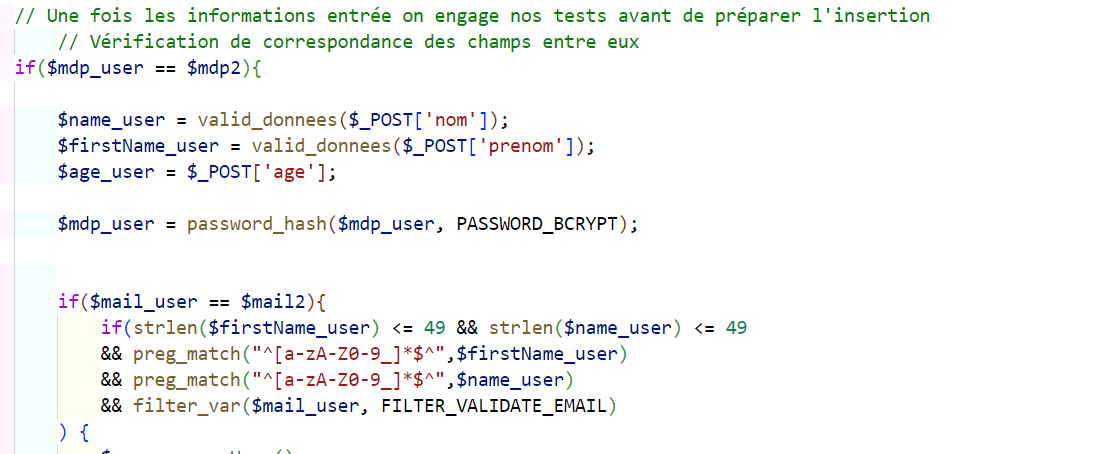


Notre super globale $\_POST va alors vérifier que les champs sont bel et bien tous remplis lors du submit de notre bouton de validation.



On va alors appliquer sur les champs récupérés une fonction valid\_donnees() afin de se protéger des injections XML.

Ceci fait on va enchainer avec nos premières vérifications afin de s’assurer que les champs de mail/mot de passe et vérification du mail/mot de passe soient identiques. Une fois les mots de passe validés, il est directement haché.



Pour mieux comprendre la suite on va d’abord jeter un œil à notre modele utilisateurs.php.

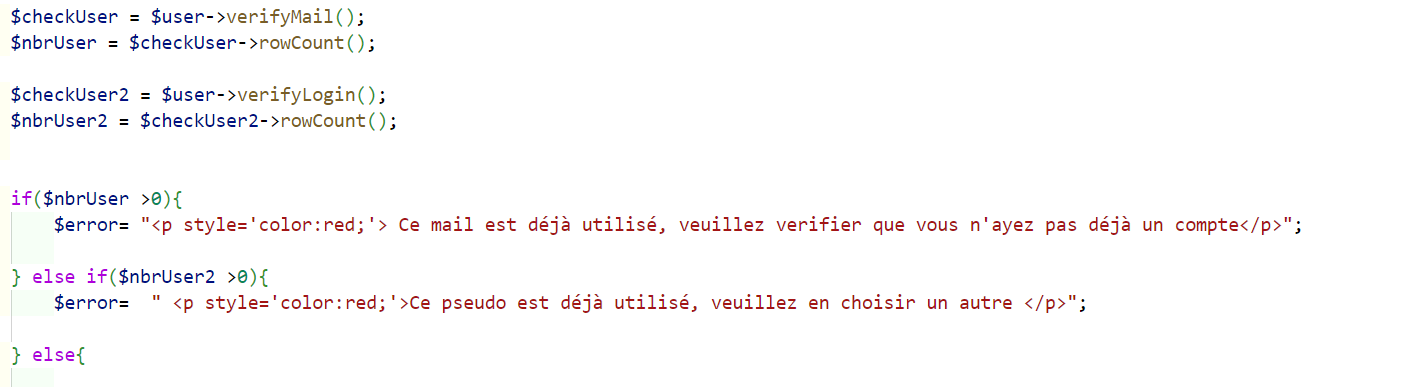
Voici la classe utilisateur qui va nous permettre la création de l’utilisateur, elle est directement connectée à la base de données par notre constructeur qui va instancier une nouvelle classe BDD à l’intérieur du constructeur de notre classe User.

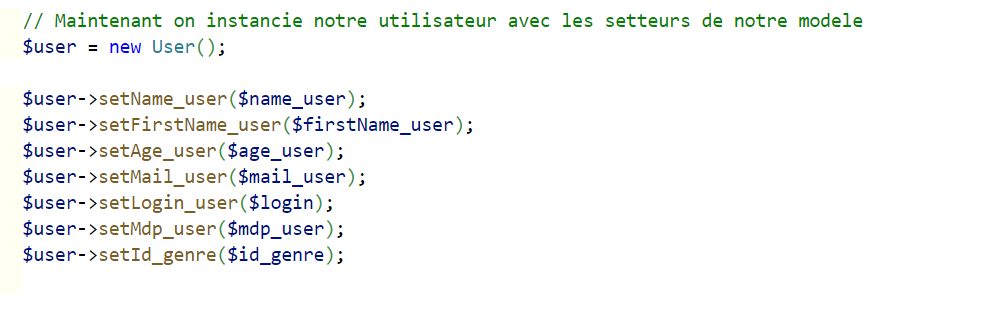


On va alors retrouver dans cette classe l’intégralité des éléments de la table utilisateur.

Pour en revenir à notre contrôleur, on a récupéré l’intégralité des informations inscrites dans nos champs, on a validé les champs doubles, on a haché le mot de passe. Il nous reste plus qu’à entamer la création de notre utilisateur.

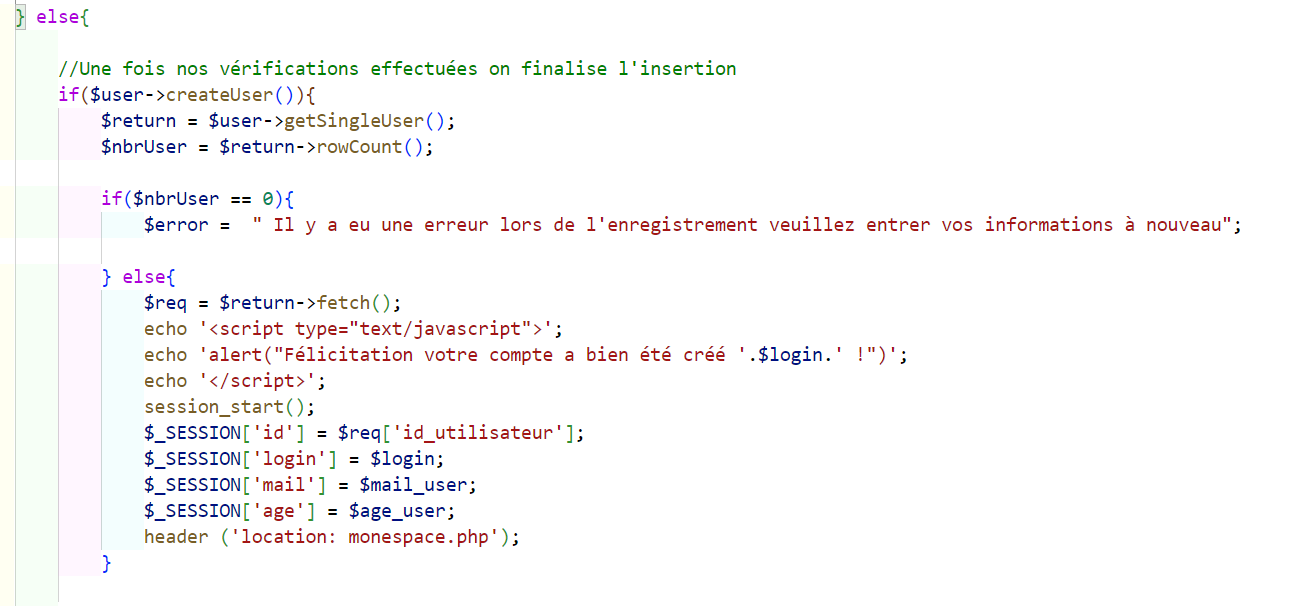
Une fois ceci fait nous allons pouvoir vérifier si le mail ou le login de l’utilisateur sont déjà utilisés sur un autre compte utilisateur. Ce sont les seules informations uniques et propre à chaque utilisateur.





Ici nous allons vérifier si à partir du mail et du login utilisateur nous trouvons des résultats identiques dans notre base de données avec un message informant le visiteur quel élément, le cas échéant est déjà inscrit dans nos données.

Si les informations entrées par notre visiteur sont uniques, alors nous pouvons l’inscrire dans notre base de données et l’accueillir chez **Verse.**



Si tout est bon et que l’insertion se fait correctement, l’utilisateur est créé. En cas d’erreur un message est affiché pour informer l’utilisateur que son inscription est interrompue suite à une erreur.

L’insertion de l’utilisateur se fait avec l’aide d’une requête préparée afin de se protéger des injection Sql. Chaque information qui ont été set auparavant sont passée en paramètre de la fonction et une fois appelée dans le contrôleur, la requête est envoyée directement à la base de donnée qui l’exécute.



Et, si tout se passe correctement, un message informe notre utilisateur que son compte a bien été créé, nous ouvrons ainsi une session avec ses informations sur le site et l’envoyons directement sur son espace.

1. **Conclusion**

**Pour conclure /merci, /monPremierProjet, /avenirDuProjet, /monAvenir, /j’aimeraisBienAvoirMonTitre**