PROJET FIL ROUGE

Au cours de cette formation, j’ai pu apprendre par la pratique et la théorie différentes compétences durant quelque mois à travers des exercices et surtout par le projet Fil Rouge.  
  
Pour cette présentation j’ai donc choisi de mettre en avant ce projet qui aborde toutes les compétences demandées en détail au détriment du stage, bien qu’enrichissant notamment pour l’approfondissement de certaines compétences mais n’étant pas complet sur certains aspects comme l’analyse fonctionnelle par exemple.

**Sommaire**

1. **Introduction**

Présentation du projet

Contexte

Technologies utilisées

Contraintes techniques

1. **Maquettage**
2. **Arborescence**
3. **Analyse Fonctionnelle**

Use Case

1. **Conception**

MCD

MLD

1. **SQL**
2. **Fonctionnalités**

Fonctionnalité Front-End

Fonctionnalités Back-End

Recherche

1. **Conclusion**
2. **Remerciements**
3. **Annexes**

**1. INTRODUCTION**

Présentation du projet : OVERHAULGAMES

L’idée : créer un site communautaire sur le jeu vidéo

Pour ce projet, j’ai voulu m’orienter vers un site web dédié au monde du jeu vidéo, un univers qui me passionne.   
L’objectif est de créer une plateforme communautaire regroupant des informations variées, des classements de jeux, un répertoire complet des titres disponibles, des actualités constamment mises à jour, et des forums interactifs.

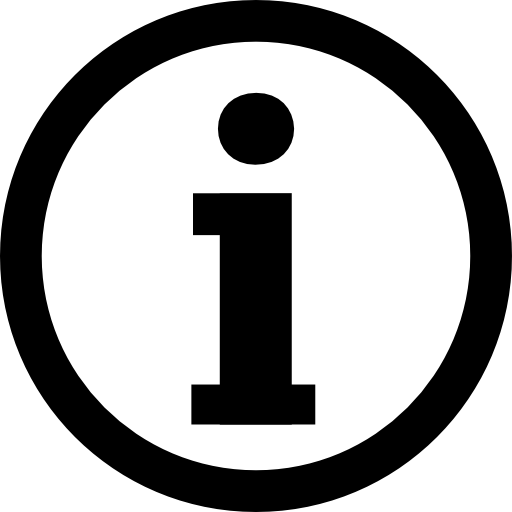
OVERHAULGAMES

Plongez dans une communauté dédiée aux jeux vidéos



Communautaire

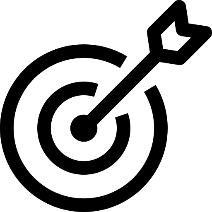
Le site se veut communautaire, offrant un espace où les passionnées comme les visiteurs occasionnels peuvent se réunir, échanger et collaborer.   
Les forums permettent des discussions approfondies sur des sujets variés, allant des nouvelles sorties de jeux aux stratégies de jeu et astuces.

Information

Comment ? Par qui ?   
L’accès à l’information sur *OverhaulGames* se fait par le biais des administrateurs, qui mettent à disposition des visiteurs des articles d’actualité et des critiques de jeux. Les internautes, quant à eux, peuvent échanger leurs connaissances et leurs expériences à travers les forums, ou directement dans les espaces de commentaires dédiés à chaque jeu. Cette interaction permet de maintenir un flux constant d'informations actualisées, tout en favorisant l'implication active de la communauté.

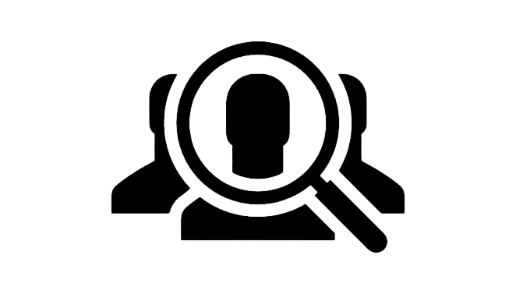
Contexte - Besoin

Les objectifs ?



Les objectifs principaux sont :

* **Le partage d’information**, via des sections actualités ou des forums, les utilisateurs ou administrateurs peuvent partager et découvrir.
* **Accès facilité à l’information**, en passant par une ergonomie optimale permettant par exemple depuis un jeu, d’accéder à ses actualités ou les forums le concernant.
* **Clarté**, un site clair sans excès d’informations inutiles ou peu pertinentes, offrant une expérience utilisateur agréable et efficace.

Les cibles ?

Le site se veut accueillant pour tous les passionnés de jeux vidéo, sans distinction d'âge, de genre ou de niveau d'expérience. Il vise à créer une communauté diversifiée où chacun peut trouver sa place et participer activement.

**Un public varié**

Hommes, femmes, jeunes et moins jeunes, *OverhaulGames* cible un large éventail de profils. Que vous soyez un adolescent découvrant les jeux vidéo, un adulte passionné depuis des années, ou un senior appréciant les jeux pour se divertir, le site s'adresse à tous.

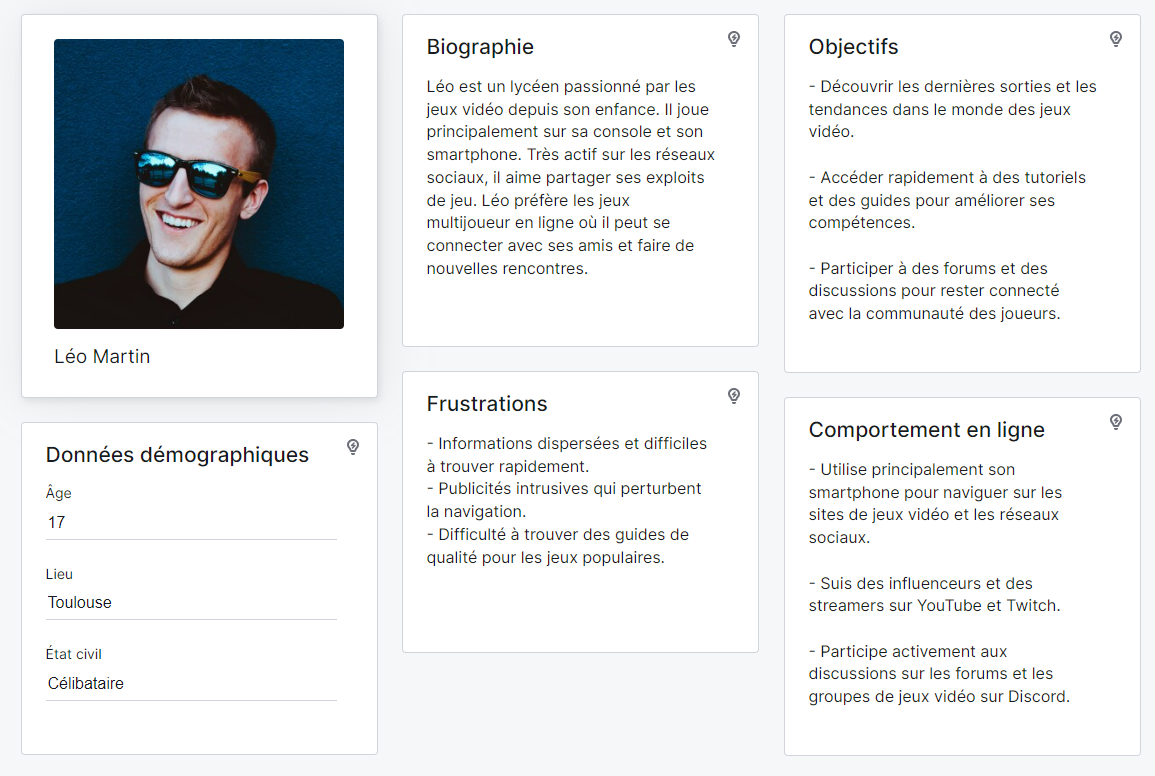
**Pour tous les niveaux d’expériences**

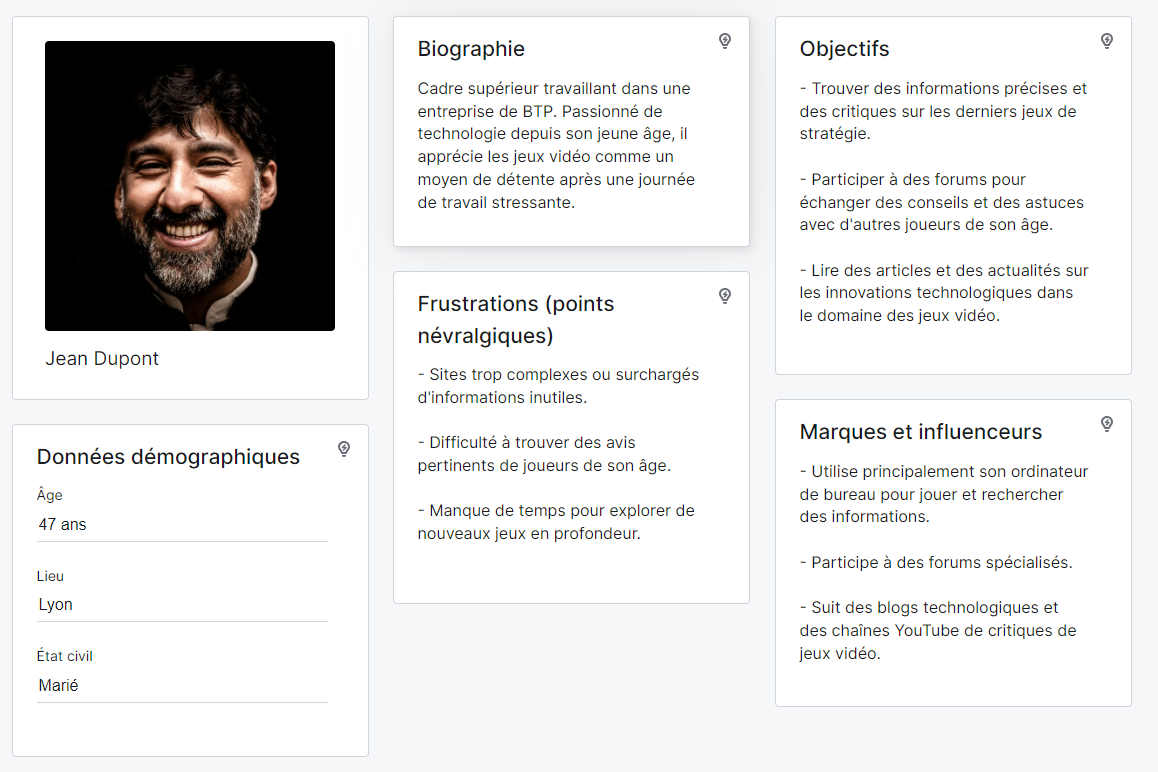
**Joueurs occasionnels :** Les joueurs qui jouent de temps en temps et cherchent des astuces simples ou des informations sur leurs jeux préférés. Ces utilisateurs apprécient une navigation facile et des contenus rapidement accessibles.

**Joueurs expérimentés :** Ceux qui ont une connaissance approfondie des jeux vidéo et aiment partager leurs propres astuces, débattre de stratégies complexes, ou rédiger des critiques trouveront également leur place. Les forums avancés, les sections de commentaires détaillées et les espaces de contribution d'articles permettent à ces joueurs de s'exprimer et de se connecter avec d'autres passionnés de leur niveau.

Voici deux persona montrant l’étendue de visiteurs que le site peut viser :

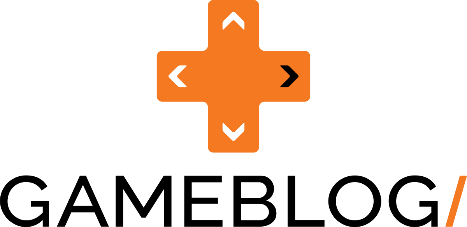
Les cibles ?





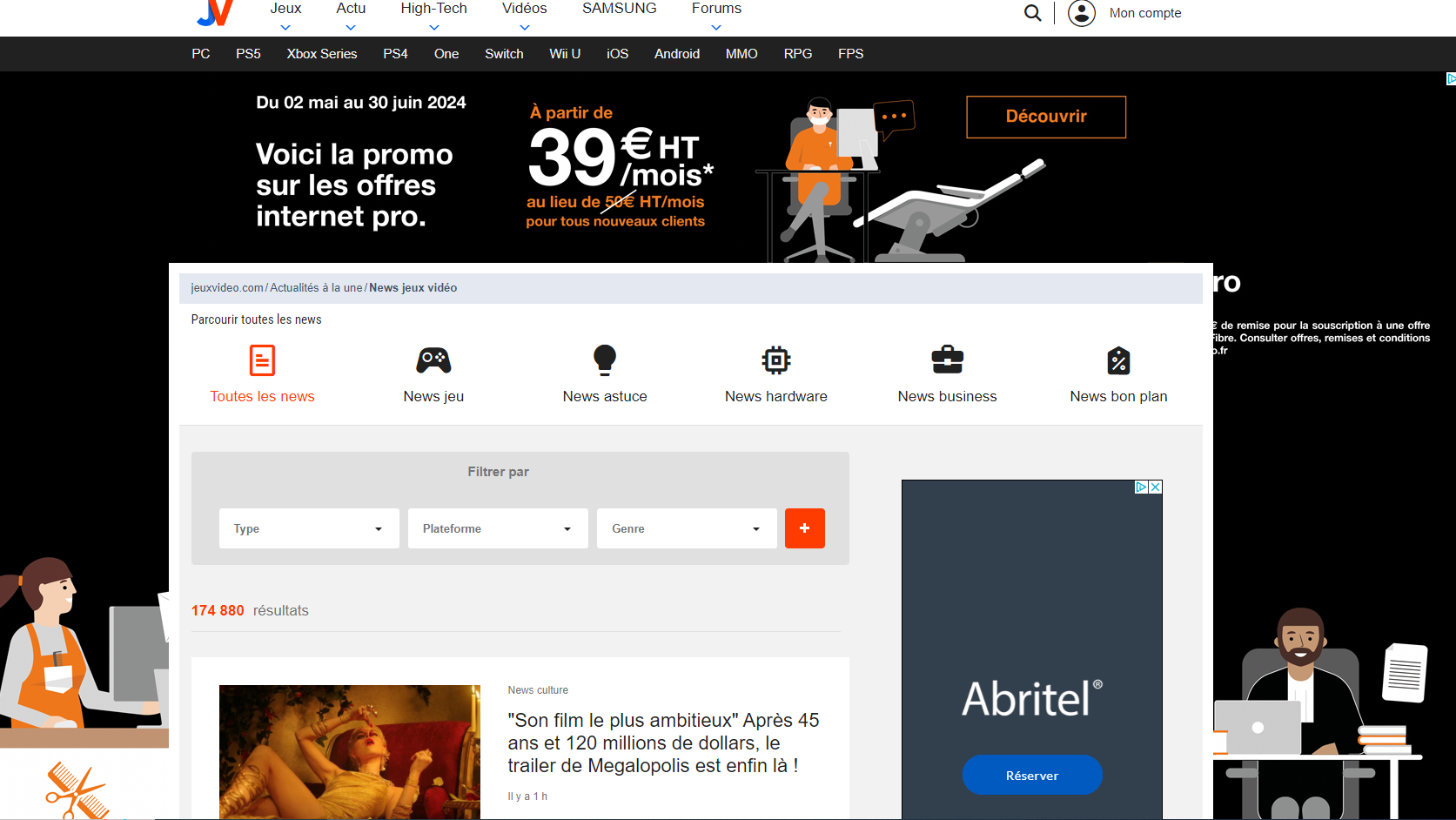
Les concurrents ?

Dans le cadre de la création de ce site communautaire, il est essentiel de comprendre l'environnement concurrentiel.   
Pour ce faire, j'ai analysé plusieurs sites bien établis dans le domaine : Gameblog, JeuxVideo.com, et Gamekult.

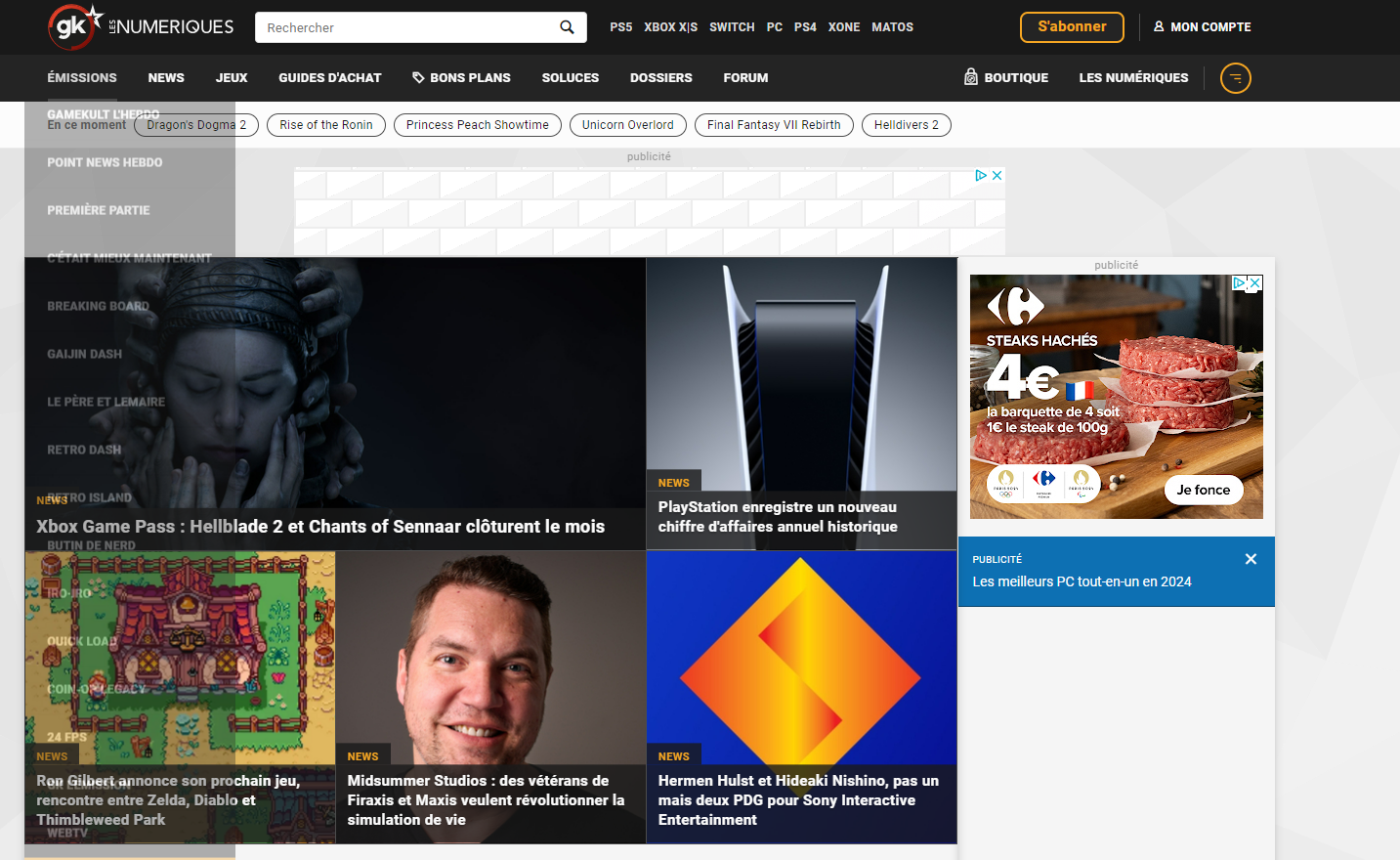
  

Généralement, chacun de ces sites possèdent un contenu riche et varié, une communauté active (voire massive pour JeuxVideos.com) et une interface dynamique.  
Grosse base de données, actualités régulièrement mises à jour ainsi que de bonnes critiques pour certains d’entre eux.

Cependant, certains de ces sites possèdent aussi de nombreuses pubs qui peuvent nuire à l’expérience utilisateur.



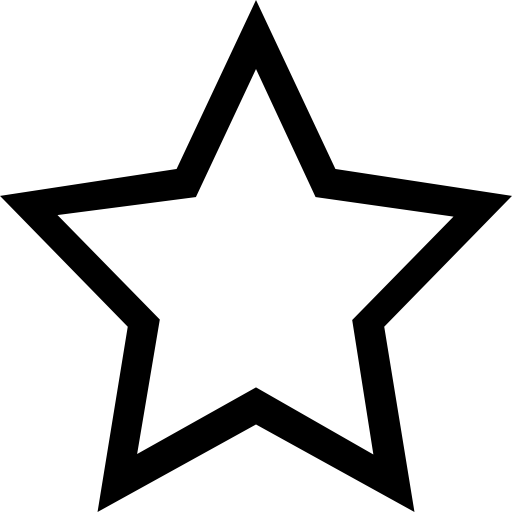
Gamekult par exemple, possède moins de pubs, mais a malheureusement une interface qui peut rendre la navigation peut être déroutante pour les nouveaux utilisateurs.



Certains ont aussi beaucoup de contenus intéressants qui sont réservés aux abonnés, ce qui peut frustrer les utilisateurs non payants.   
L’impact est donc d’avoir une communauté moins active par rapport à ses concurrents.

Que peut-on en tirer pour mon projet Fil Rouge ?

L’idée est de prendre en compte tout cela, en essayant de rendre une information facile d’accès et visible.   
  
**La création d’un compte** n’apporterait que des avantages mineurs mais très pratiques : ajouter des articles, tests en **favoris** pour avoir une information rapidement.



(Exemple : la soluce d’un jeu ou un forum parlant de plein d’astuces.)

C’est un avantage que l’on retrouve peu sur les autres sites, mettant en avant *OverhaulGames* pour sa praticité de consultation d’information.

Les utilisateurs ne seraient pas frustrés par un manque de contenu en étant pas abonné ou autres. En limitant les pubs sur la page et ne pas polluer le site de section et sous-section, on peut créer un site facile d’accès et communautaire.

Technologies utilisés

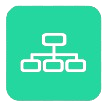
Pour la réalisation de mon projet, j'ai utilisé une variété d'outils et de technologies qui m'ont permis de concevoir, développer et organiser le site. Chaque outil a joué un rôle clé dans les différentes étapes du projet, de la création des maquettes à la gestion des bases de données, en passant par la réalisation de ce dernier. Voici un aperçu des principaux outils que j'ai utilisés :

**MAQUETTAGE**

****Figma : Editeur de graphiques vectoriels et outil de prototypage.   
Création du Zoning, Wireframe et MockUp de mon site.   
Utile aussi pour la création d’un moodboard lors de la création de la charte graphique.

Free Logo Design : Plateforme en ligne permettant de créer des logos professionnels gratuitement.   
J’ai utilisé FLD pour concevoir et personnaliser le logo de mon projet.

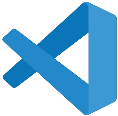
ARBORESCENCE

GlooMaps : Outil en ligne facile à utiliser pour créer des sitemaps. Il permet de dessiner rapidement des plans de site visuels, d'ajouter des pages, et de les organiser. Je l’ai utilisé pour la création de l’arborescence du site.

CONCEPTION

Looping : Logiciel de modélisation conceptuelle de données avec lequel j’ai réalisé mon MCD / MLD.

MySQL : Logiciel de gestion et d’administration de bases de données MySQL.  
Il permet de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d’effectuer toutes les opérations inhérentes à la gestion d’une base de données.

REALISATION

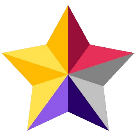
VS Code : Editeur de texte qui permet le développement d’applications pour toutes les plateformes et tous les langages.

HTML / CSS : Langages de base pour la structuration et le style des pages web. HTML est utilisé pour créer la structure et le contenu des pages, tandis que CSS est utilisé pour la mise en forme et le design visuel.   
J’ai utilisé HTML et CSS pour coder toutes les pages de mon site, en veillant à ce que le design soit un minimum cohérent.

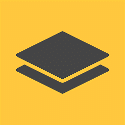
JAVASCRIPT : Langage utilisé pour ajouter des fonctionnalités interactives et dynamiques au site. Il m’a permis de créer des éléments tels que des animations.   
J'ai utilisé JavaScript pour améliorer l'expérience utilisateur et rendre le site plus réactif.

GITHUB : Plateforme de gestion de versions utilisée pour héberger et collaborer sur le code du projet.   
J'ai utilisé GitHub pour suivre les modifications et gérer les versions.

MODÉLISATION

StarUML : Outil de génie logiciel dédié à la modélisation UML.   
J'ai utilisé ce logiciel pour réaliser les diagrammes de séquences, diagrammes d'activités ainsi que le Use Case afin de planifier et documenter la structure du projet.

AUTRES OUTILS PRATIQUES

REMOVE.BG : Outil en ligne gratuit utilisé pour supprimer l'arrière-plan des images de manière automatique et précise.   
J'ai utilisé Remove.bg pour éditer les images du projet en supprimant les arrière-plans indésirables et en mettant en valeur les éléments principaux des visuels.

CANVA : Plateforme de design graphique en ligne offrant des modèles, outils et ressources pour créer des visuels de bonne qualité.   
J'ai utilisé Canva pour concevoir la présentation du projet qui me servira de support.

Word : Logiciel de traitement de texte.   
J'ai utilisé ce logiciel pour rédiger le contenu de mon projet.

Contraintes techniques

Lors de la réalisation du projet de ce projet, plusieurs contraintes techniques ont dû être prises en compte pour assurer la performance, la sécurité et l'accessibilité du site web :

* **Performance** : Le site doit se charger rapidement pour offrir une expérience utilisateur fluide, en particulier pour les utilisateurs avec des connexions internet plus lentes. Cela implique l'optimisation des images en utilisant des formats adaptés et des outils de compression, la minification des fichiers CSS et JavaScript, et l'utilisation de techniques de mise en cache pour réduire les temps de chargement des pages.
* **Compatibilité multi-navigateurs** : Assurer que le site fonctionne correctement sur les principaux navigateurs (Chrome, Firefox, Safari, Edge) et leurs différentes versions. Cela a nécessité des tests approfondis et des ajustements spécifiques pour chaque navigateur.
* **Responsivité** : Le site doit être entièrement responsive, s'adaptant à toutes les tailles d'écran, des ordinateurs de bureau aux smartphones et tablettes.
* **Sécurité** : Protéger les données des utilisateurs et assurer la sécurité du site contre les attaques courantes comme les injections SQL et le Cross-Site Scripting (XSS). L'implémentation de protocoles de sécurité et de bonnes pratiques de codage a été essentielle, incluant des pratiques de validation et de nettoyage des entrées utilisateurs, ainsi que l'authentification et l'autorisation sécurisées.
* **Accessibilité** : Rendre le site accessible aux utilisateurs ayant des handicaps, en suivant les directives WCAG (Web Content Accessibility Guidelines). Cela inclut des éléments comme des descriptions d'images (alt text) et des contrastes de couleurs adéquats.

En surmontant ces contraintes techniques, le projet *OverhaulGames* a pu être développé de manière à fournir une expérience utilisateur optimale et à répondre aux exigences modernes.

**2. MAQUETTAGE**

Le maquettage est une étape cruciale dans la conception de sites web, permettant de visualiser et de planifier l'apparence et la structure du site avant le développement. Pour mon projet, j’ai utilisé Figma.

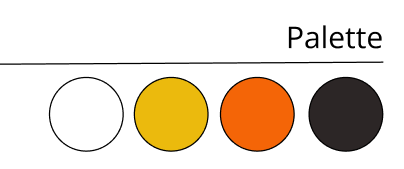
Avant de commencer le maquettage, j’ai pris un temps afin d’établir un moodboard pour faire la **charte graphique**, en réfléchissant respectivement au logo, couleurs, polices et charte éditoriale.

**La charte graphique ?**

**LE LOGO**

J’ai crée ce logo à partir d’un modèle sur Free Logo Design, qui m’inspirait la simplicité avec une manette, permettant de saisir directement sur quel genre de site on est, facilement identifiable.

LES COULEURS

J’ai choisi les couleurs jaune et orange qui sont des couleurs joyeuses avec du dynamisme. Pour faire un contraste, j’ai utilisé aussi du noir. Il s’agit d’un choix aussi esthétique que pratique : il est assez courant sur les sites de jeux vidéos d’utiliser du noir, sans oublier les raisons d’accessibilité évoqué dans les contraintes techniques.   
En choisissant des couleurs vives, on peut garantir une bonne lisibilité. J’ai tout de même fait attention à ne pas surcharger ces couleurs vives partout, qui peuvent être lassante et agressive à la longue.

J’ai donc veillé à l’utilisation de couleurs complémentaires et lisibles, ne serait-ce que pour des personnes souffrant de handicap comme le daltonisme par exemple, en utilisant des sites comme Adobe Color :





On peut voir un rapport de contraste réaliste et lisible !

LES POLICES



Pour les mêmes raisons d’accessibilité évoqué précédemment, j’ai choisis quelques variantes du **Roboto** en fonction des titres, sous titres et corps de texte car c’est une police très facile à lire, elle améliore l’expérience utilisateur en plus d’avoir un style moderne et simple. C’est l’une des polices les plus utilisés sur le web et notamment pour des raisons d’accessibilité pour les personnes souffrant de handicap comme la dyslexie.

J’ai choisi arbitrairement la police Anton pour le nom du site, qui d’après mes recherches est assez courants d’avoir une police volumineuse sur ce genre de sites. Avec le logo et cette police, on rejoint le point d’être un site facilement identifiable.

LA CHARTE EDITORIALE

La charte éditoriale de *OverhaulGames* garantit un contenu clair, informatif et engageant, adapté à tous les niveaux d'expérience des utilisateurs. Elle inclut des directives sur le ton, le style, tout en assurant la cohérence et la qualité des articles, des actualités et des discussions sur les forums.

Les mots d’ordre doivent être communauté, bienveillance et simplicité.

**Processus de Maquettage ?**

Dans un premier temps, j’ai décidé de démarrer le maquettage sur du mobile-first et en voici les raisons :

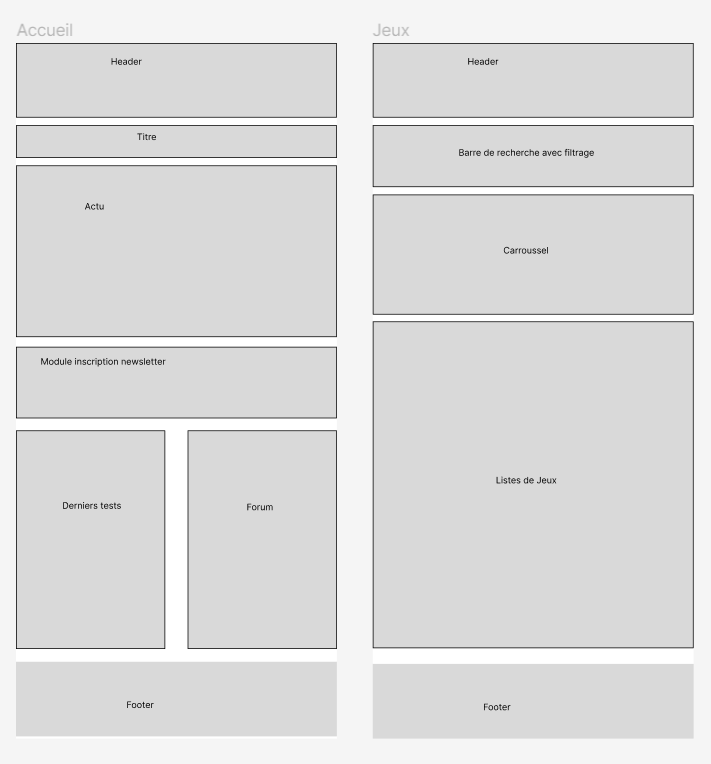
* L'utilisation des appareils mobiles pour naviguer sur Internet a considérablement augmenté ces dernières années. En adoptant une approche mobile-first, on garantit une expérience utilisateur optimale pour une majorité d'utilisateurs qui accèdent au site via leurs smartphones et tablettes.
* C’est aussi plus simple pour adapter par la suite notre site à une plus large gamme d’écran. Commencer un design pour les plus petits écrans puis évoluer vers un écran avec bien plus d’espace est relativement plus simple que l’inverse.
* Concevoir pour mobile oblige à prioriser le contenu le plus important. Les limitations d'espace sur les écrans mobiles encouragent des décisions de design qui mettent en avant les informations et les fonctionnalités clés, améliorant ainsi la clarté et la pertinence du contenu présenté.
* Et enfin il est dit que des moteurs de recherches, comme Google, favorisent les sites qui offrent une excellente expérience mobile. En utilisant une stratégie mobile-first, on peut améliorer son référencement et augmenter ses chances de se classer plus haut dans les résultats de recherche, attirant ainsi plus de visiteurs.

Après avoir effectué des recherches sur les tendances actuelles et analysé des sites similaires pour m’inspirer et établir la charte graphique en ayant identifié les meilleures pratiques, j’ai commencé le maquettage.

ZONING

Le zoning est une représentation schématique **de la structure d'une page web ou d'une application mobile**, réalisée à l'aide de formes simples et sans aucun élément graphique.

J'ai créé des zonings pour définir l'organisation de chaque page. Cela a permis de décider de la disposition générale des éléments sans se concentrer sur les détails esthétiques.



Le but était de réaliser une première idée de la disposition, qui n’est bien sûr pas définitive et sera modifié au fil du temps.

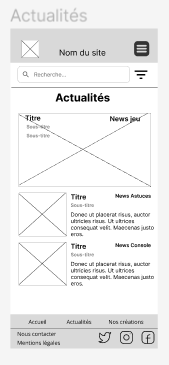
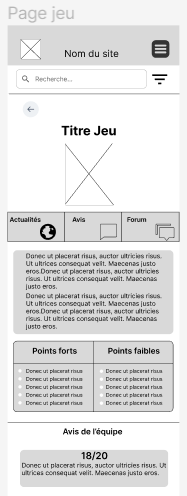
WIREFRAME

Un wireframe est un croquis qui permet de **visualiser la structure et l'organisation des éléments de l'interface**, sans se soucier des détails graphiques.

À partir des zonings, j'ai créé un wireframe pour détailler la structure de chaque page.   
Elle permet donc la visualisation :   
- Des zones de textes  
- L’emplacement des images, des vidéos, des liens ainsi que des différents éléments graphiques.

Utilisé lors de la conception d’une interface utilisateur pour définir les zones et composants qu’elle doit contenir.   
À partir d’un wireframe peut être réalisée l’interface proprement dite par un graphiste.   
La démarche de recourir à des wireframes s’inscrit dans une recherche d’ergonomie.

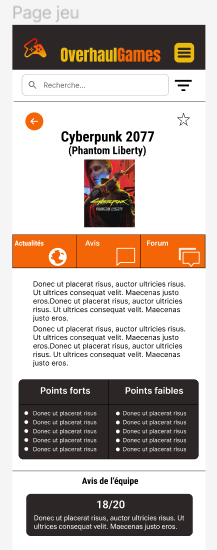
**PAGE D’ACCUEIL DES TESTS ET ACTUALITÉS**

*Ayant plus d’une quinzaine de pages pour le wireframe ainsi qu’une vingtaine pour le Mock-Up, j’ai décidé qu’il ne serait pas pertinent de toutes les montrer ici, et donc de n’intégrer que quelques pages principales pour servir d’exemple ! Le reste se trouve dans l’annexe.*

MOCK UP   
Du point de vue matériel, un mock-up se caractérise par le fait d’être un modèle visuel esthétique, fixe ou sous forme de vidéo, qui peut être modifié aux fins de personnalisation, souvent des textes.   
Il est utilisé à des fins d’études, de tests, ou de démonstration.   
Dans le domaine de la conception d’interfaces, il est une représentation du futur du site ou de l’application mobile.   
Il est un modèle réaliste de ce à quoi ressemblera la future interface utilisateur.   
C’est une simulation statique, en moyenne ou haute fidélité, de l’UI, avant le processus de développement.   
  
J'ai donc développé un mock-up détaillé, incorporant les styles visuels définitifs, les couleurs, les typographies et les images. Cela a aidé à visualiser le rendu final du site et à faire des ajustements avant le début du codage

**PAGE D’ACCUEIL DES TESTS ET ACTUALITÉS**



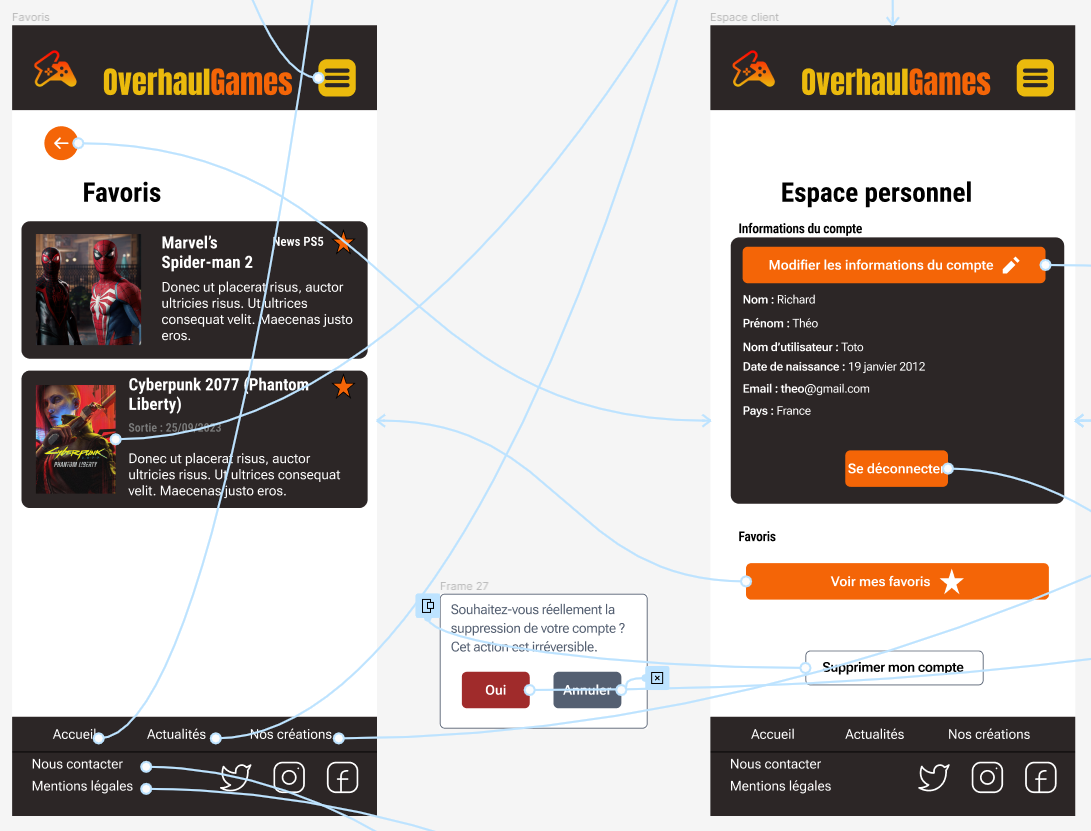
J’ai suivi la charte graphique choisie et respecté le wireframe initial.

Précédemment lors de l’explication des besoins, du contexte, je parlais de practicité d’utilisation. Ne pas surcharger de boutons et autres, tout en permettant aux utilisateurs de récupérer facilement des informations. Ici, on peut voir par exemple l’idée d’incorporer 3 boutons visibles sous la présentation du jeu, permettant d’accéder directement à des avis sur le jeu, les actualités et forums le concernant.

PROTOTYPAGE

L'une des fonctionnalités de Figma est le prototypage. Cela a permis de créer des prototypes cliquables des pages du site, simulant la navigation et l'interaction utilisateur. Ces prototypes ont été utilisés pour tester et valider les flux de navigation et l'ergonomie avant de commencer le développement.

*[Cliquez ici pour accéder au prototypage que j’ai réalisé sur Figma](https://www.figma.com/proto/GwxzYNOtVgvlvakmX3W1d7/Mockup?node-id=202-32&t=B8nyZnidPoskxtB7-0&scaling=scale-down&page-id=0%3A1&starting-point-node-id=22%3A873)*

  
  
Ici, j’ai pu tester en amont le fait qu’un pop-up sorte lorsque le bouton « supprimer mon compte » est cliqué. Le prototype sert à voir ce que va donner aussi la navigation entre les pages, et s’assurer que les fonctionnalités et l’ergonomie répondent aux attentes.

En utilisant Figma pour le maquettage, mon projet a bénéficié d'une planification visuelle détaillée et d'une collaboration efficace, assurant une conception bien pensée et prête pour le développement.

Avant la présentation du maquettage, j’ai évoqué l’utilisations des meilleurs pratiques, en voici :

UX / USER EXPERIENCE

Aspect **« subjectif »** de l'utilisation d'un produit. Elle détermine si l'utilisateur trouve l'expérience **agréable et facile**.

L'objectif principal de l'UX est de créer une expérience utilisateur positive, qui encourage l'utilisateur à revenir et à utiliser le produit ou service à nouveau.

Pour atteindre cet objectif, les designers UX s'appuient sur différentes méthodes, telles que la recherche utilisateur, l'ergonomie, l'architecture de l'information et le design d'interaction.

L’UX prend en compte :

* **La facilité d'utilisation** : l'utilisateur peut-il trouver facilement ce qu'il recherche ? Les fonctionnalités sont-elles intuitives et faciles à utiliser ?
* **L'utilité** : le produit ou service répond-il aux besoins de l'utilisateur ?
* **L'esthétique** : le produit ou service est-il agréable à regarder et à utiliser ?

**UI / USER INTERFACE**

Désigne l'ensemble des éléments visuels et interactifs qui permettent à un utilisateur d'**interagir** avec un produit, un service ou un système.

C’est l’aspect **concret** de l'utilisation d'un produit. Elle concerne tout ce que l'utilisateur peut voir, toucher, entendre et avec lequel il peut interagir.

L'objectif principal de l'UI est de créer une interface utilisateur intuitive et esthétique, qui permet à l'utilisateur d'accomplir ses tâches facilement et efficacement.

L’UI prend en compte :

* **La mise en page** : l'organisation des éléments sur l'écran est-elle claire et logique ?
* **Le choix des couleurs** : les couleurs sont-elles attrayantes et adaptées à la marque ?
* **La typographie** : les polices de caractères sont-elles lisibles et agréables à regarder ?
* **Les boutons et les icônes** : sont-ils faciles à comprendre et à utiliser ?

SEO / SEARCH ENGINE OPTIMIZATION

Ensemble des techniques visant à améliorer la position d'un site web dans les résultats des moteurs de recherche.   
L'objectif est d'**accroître la visibilité du site. Il a deux types de référencement possible :**

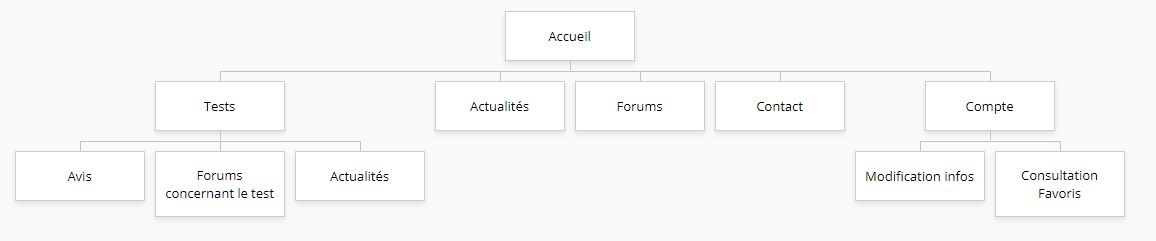
* **Le référencement naturel (SEO)** : Ensemble de techniques visant à améliorer le positionnement d'un site web dans les résultats de recherche naturels. Cela implique notamment d'optimiser le contenu du site, de créer des backlinks de qualité et d'améliorer l'expérience utilisateur.
* **Le référencement payant (SEA)** : il s'agit d'acheter de la visibilité sur les moteurs de recherche en enchérissant sur des mots-clés. Les annonces apparaissent ensuite dans les résultats de recherche payants, en haut ou sur le côté de la page.

**3. ARBORESCENCE**

L'arborescence est un élément fondamental de la structure d'un site web, permettant d'organiser le contenu de manière logique et intuitive pour les utilisateurs.

Sans une arborescence de site solide, un site internet n’est qu’un ensemble de pages et d’articles de blog complètement désorganisé. Les internautes ont besoin de cette architecture pour naviguer d’une page à l’autre et comprendre où trouver l’information.   
C’est un élément déterminant de l’expérience utilisateur.

Les moteurs de recherche utilisent également la structure d’un site pour déterminer quel contenu est important et ce qui est moins pertinent, faisant de l’arborescence de site un enjeu majeur du référencement naturel (SEO).  
Meilleure est la structure du site, meilleures sont les chances de se positionner sur la première page de Google.



Les visiteurs auront accès à toutes les pages, sauf la partie compte qui contient une page réservé aux utilisateurs connecté à leur compte permettant d’accéder à leur favoris. Sans oublier la modification de ses infos personnelles !

Les administrateurs auront des fonctionnalités supplémentaires, notamment par exemple pour les actualités où il leur sera proposé d’ajouter un article.

**4. ANALYSE FONCTIONNELLE**

**UML ET DIAGRAMME DE CAS D’UTILISATION**

**Introduction à l’UML**

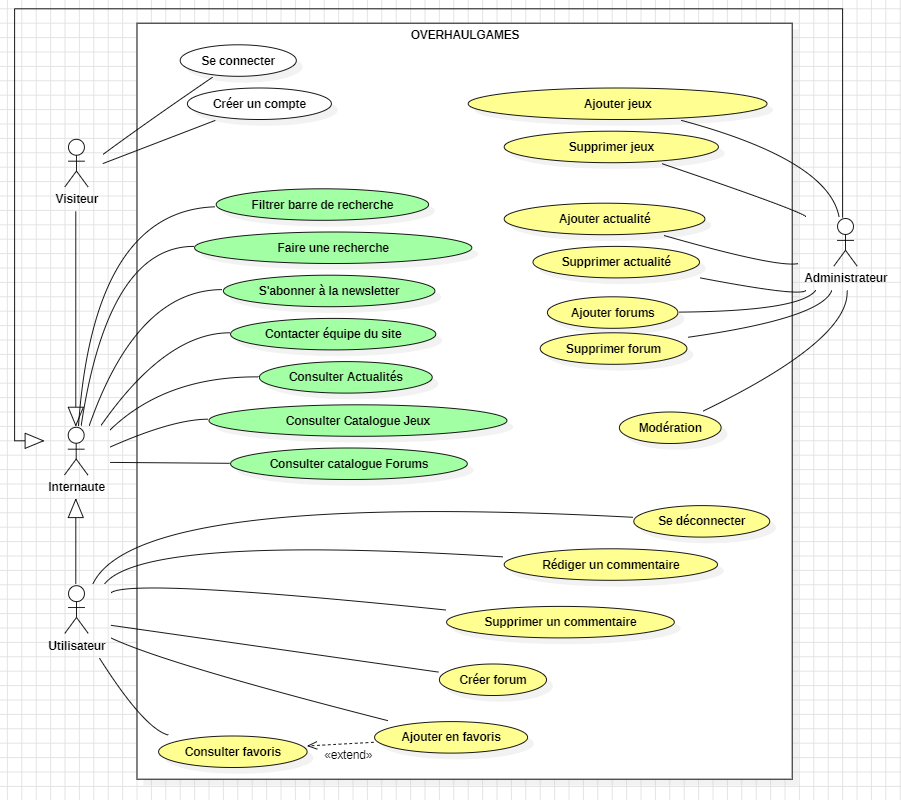
L'UML, ou *Unified Modeling Language* est un langage de modélisation standardisé utilisé dans le domaine du développement de logiciels pour **visualiser, spécifier, construire et documenter les processus, éléments et objets qui interagissent entre eux dans un système**. Il offre une manière structurée de représenter les systèmes à travers différents types de diagrammes. L'UML est largement utilisé pour ses capacités à simplifier la complexité des projets logiciels, à améliorer la communication entre les parties prenantes et à garantir une compréhension commune du système à développer.

Ici, nous allons évoquer le diagramme de Cas d’Utilisation, autrement dit le Use Case.

**Introduction au diagramme de Cas d’Utilisation**

Un **diagramme de cas d'utilisation** est un outil clé dans la modélisation UML. Il permet de visualiser les différentes façons dont un utilisateur pourrait interagir avec un système. Le diagramme de cas d'utilisation se compose de trois éléments principaux : **Acteurs**, **Cas** **d’Utilisation**, **Relations**.

DIAGRAMME DE CAS D’UTILISATION D’OVERHAULGAMES



**Acteurs :**

Un acteur représente le rôle d’un utilisateur qui intéragit avec le système sur l’application en cours de création.   
L’utilisateur peut être un utilisateur humain, une organisation, une machine ou un autre système externe.

Ici, on retrouve **4 acteurs** dans le Use Case de mon projet.   
- Visiteur   
- Utilisateur   
- Internaute  
- Administrateur

Mes motivations ici de créer ces 4 acteurs sont les suivantes :

La création d’un **administrateur** est essentielle pour ce site et il est logique qu’il puisse avoir ses propres cas d’utilisations.

**L’utilisateur** lui a ses propres caractéristiques puisqu’il représente un utilisateur lorsqu’il est **connecté**.

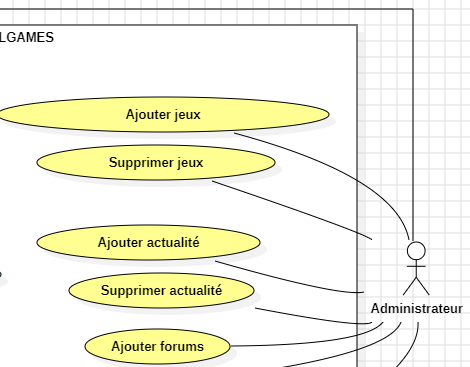
Le **visiteur** est un utilisateur lambda qui n’est pas connecté à un compte.

L’utilisateur, le visiteur ainsi que l’administrateur ont tout les 3 accès à des cas d’utilisations communs comme le fait de naviguer entre les pages des catalogues ou encore faire une recherche, s’abonner à la newsletter etc… mais ont chacun des caractéristiques propres qui ne peuvent pas se mélanger.   
Il n’est pas possible pour un utilisateur d’avoir par exemple comme cas d’utilisation « créer un compte » puisqu’il s’agit d’un acteur représentant un utilisateur déjà connecté.

C’est donc pour ces raisons que j’ai décidé de créer un 4e acteur, l’**internaute**, étant un acteur **virtuel** qui permet à tout les autres d’hérité de ses cas d’utilisations, en évitant les soucis de compatibilité entre les différents acteurs.

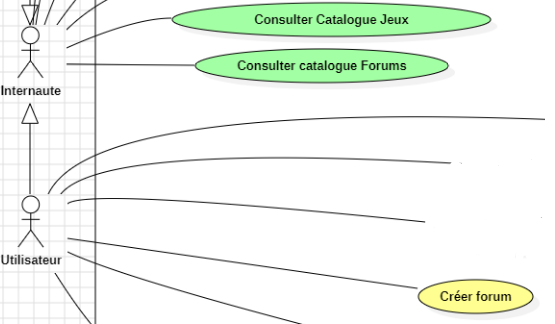
**Cas d’Utilisation :**

Un cas d’utilisation sont les fonctionnalités ou services offerts par le système en réponse aux interactions des acteurs (par exemple, la fonctionnalité pour l’administrateur d’ajouter/supprimer un jeu, une actualité ou un forum sont des cas d’utilisations, cf : ci-dessous).



**Relations :**   
Ce sont les liens entre les acteurs et les cas d'utilisation, ainsi que les dépendances entre différents cas d'utilisation. Il existe différents types de relations dans le cadre d’un diagramme de cas d’utilisation, les voici :

* **Association :** Représente une connexion ou une interaction directe entre un acteur et un cas d'utilisation.   
  ***Exemple****: Un administrateur (acteur) est associé à l'action d’ajouter un jeu au catalogue(cas d'utilisation).*
* **Généralisation (Héritage) :** Il est possible de définir une relation d’héritage entre eux afin d’éviter de surcharger les diagrammes.  
  **On utilise la généralisation lorsqu’un acteur possède un cas d’utilisation commun à un autre, tout en bénéficiant de cas spécifiques.**

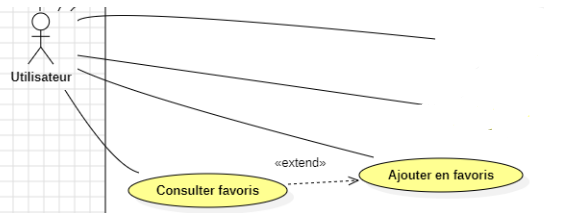


Dans cet exemple, on peut voir une notion d’héritage où l’utilisateur hérite des caractéristiques de l’internaute. On a voulu créer cet héritage afin de permettre à l’utilisateur d’accéder à toutes les fonctionnalités de bases du site dont celles en exemple : la consultation des Jeux et Forums tout en ayant ses propres caractéristiques comme la création d’un compte.

C’est un cas d’utilisation qui n’aurait pas pu être pour l’internaute car c’est une fonctionnalité réservée à un utilisateur connecté.

* **Include** :Ici, un cas d'utilisation inclut systématiquement un autre cas d'utilisation. Cela signifie que le comportement du cas d'utilisation inclus est toujours exécuté dans le cadre du cas d'utilisation de base.

***Exemple*** *: Le cas d'utilisation "Publier un commentaire" peut inclure le cas d'utilisation "Se connecter" pour s'assurer que l'utilisateur est authentifié avant de publier.*

* **Extend :** Un cas d'utilisation peut étendre le comportement d'un autre cas d'utilisation de manière conditionnelle. Cela permet d'ajouter des fonctionnalités supplémentaires sous certaines conditions.
* 

Ici, le cas d’utilisation « Consulter favoris » peut être étendu par « Ajouter en favoris » car on peut très bien consulter ses favoris sans avoir à en ajouter nécessairement sur le coup.

**5. CONCEPTION**

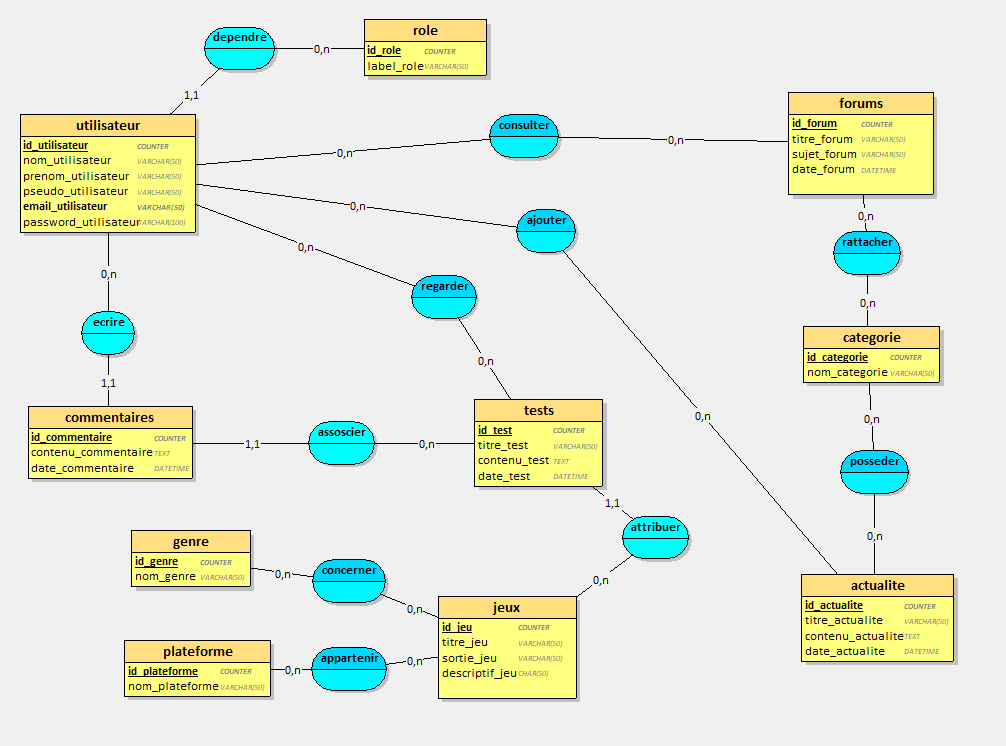
MCD

Le MCD est un schéma représentant la **structure du système d’information** du point de vue des données, c’est-à-dire les dépendances ou **relations entre les différentes données du système d’information**.

Le modèle conceptuel de données (MCD) a pour but de décrire de façon formelle les données qui seront utilisés par le système d’information.  
Les concepts de base sont : entités (ou objet, classe), relation ou association, propriétés ou attributs.

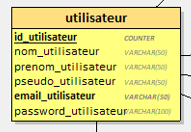
Il s’agit donc d’**une** **représentation des données**, facilement compréhensible, permettant de décrire le système d’information à l’aide d’entités.

MCD OverhaulGames



Le MCD s’appuie sur une méthode de conception et **permet ainsi d’obtenir facilement le modèle relationnel qui servira à la construction de la base de données dans le SGBDR**.  
Ces « relations » vont permettre de créer les tables de la base de données.

Les caractéristiques du MCD :

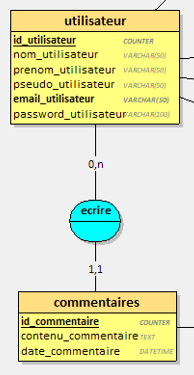
LES ENTITÉS

Les entités représentent les objets du domaine étudié qui possèdent des propriétés distinctes. Dans le contexte de mon projet, on peut retrouver des entités comme "Utilisateur", "Jeux", "Commentaires", "Favoris", etc.

Ici, je vais détailler l’entité « utilisateur ».   
Elle se compose :

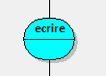
* D’une clé primaire **id\_utilisateur,** avec comme typage « ***counter*** » comme il s’agit d’une donnée qui sera « ***auto-incrémenter*** » à chaque création d’un utilisateur, c’est-à-dire qu’il est unique et se mettra à jour dès qu’un profil sera créer.
* De quelque attributs composant un utilisateur comme son nom, prenom, etc… avec comme typage « ***varchar*** », qui indique que cet caractéristique contiendra une « ***string*** » avec une valeur plus ou moins grande selon l’utilité.
* D’un **email\_utilisateur** qu’on peut voir en gras. Cela veut dire que j’ai précisé que cet identifiant sera unique, et que chaque utilisateur aura un email différent.

LES CARDINALITÉS

Les cardinalités permettent d’exprimer le nombre minimum et maximum d’occurrences d’une entité par rapport à une autre (liées par une association).  
Dans le cas de l’entité « *utilisateur* » reliée à l’entité « *commentaires* » qui vient définir les commentaires des utilisateurs, on constate une occurrence 0,n et 1,1.

Littéralement cela signifie qu’un utilisateur peut écrire aucun ou plusieurs commentaires, et qu’un commentaire doit être écrit par au moins un utilisateur et pas plus. On parle ici de **Type 1**.

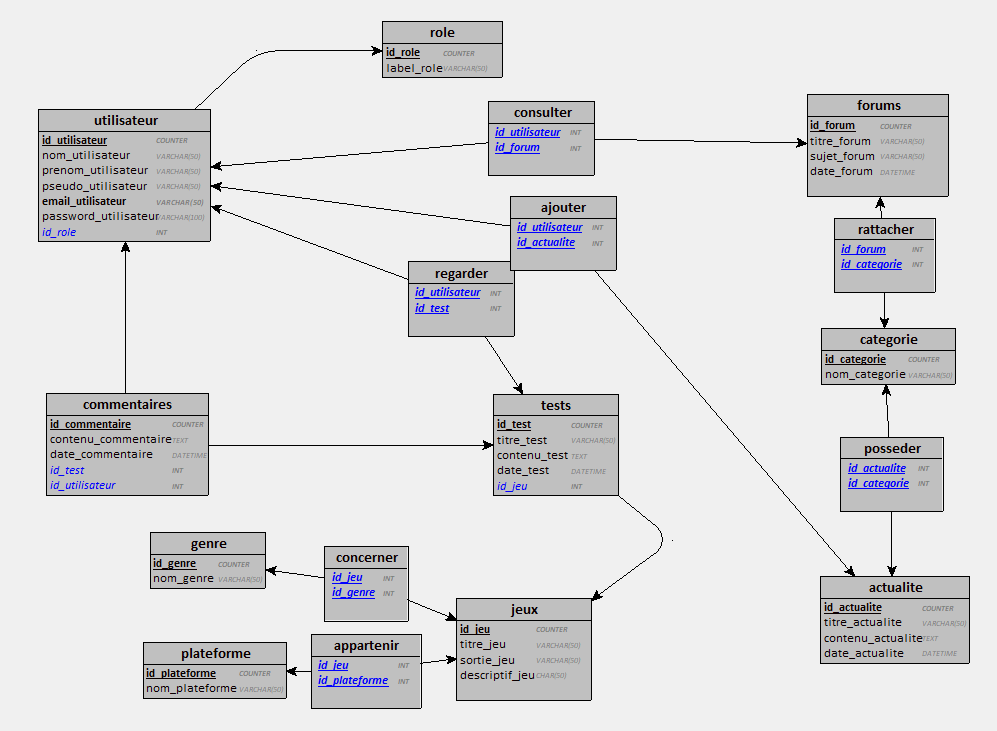
LES ASSOCIATIONS

Une association est une connexion entre des entités.   
Elle permet de lier des entités représentant chacune des objets clairement définis mais qui sont liés par un événement qui peut ne pas être très clairement représenté par une autre entité.   
L’association n’existe qu’à travers les entités qu’elle relie.

MLD

Le modèle relationnel permet d’organiser les données (attributs) en « relations » qui se transformeront en « tables » dans la base de données.  
Il est basé sur le concept de dépendances fonctionnelles et est aussi appelé Modèle Logique de Données (MLD).

MLD OverhaulGames



*Un exemple d’une transformation d’une des tables en fonction des cardinalités se trouve plus bas, page 27*

Passer du MCD au MLD

1. **Une entité du MCD devient une relation, c’est-à-dire une table.**

Une table contiendra donc un ensemble d’enregistrements.   
Une ligne correspond à un enregistrement.   
Une colonne correspond à un champ.

**Son identifiant devient la clé primaire de la relation.**   
La clé primaire permet d’identifier de façon unique un enregistrement dans la table.   
Les valeurs de la clé primaire sont donc uniques.

**Les autres propriétés deviennent les attributs de la relation.**

1. **Une association de type 1,N**

(C’est-à-dire qui a les cardinalités maximales positionnées à « 1 » d’un côté de l’association et à « n » de l’autre côté) qui est une association qui se traduit par la création d’une clé étrangère dans la relation correspondante à l’entité côté « 1 ».

Cette clé étrangère référence la clé primaire de la relation correspondant à l’autre entité.

1. **Une association de type N:N**

(C’est-à-dire qui a les cardinalités maximales positionnées à « n » des 2 côtés de l’association comme **0,n - 0,n** / **1,n - 1,n / 1,1 – 1,1)** qui est une association qui se traduit par la création d’une table dont la clé primaire est composée des clés étrangères référençant les relations correspondant aux entités liées par l’association.  
Les éventuelles propriétés de l’association deviennent des attributs de la relation.

RELATION

Une relation est un **sous-ensemble du produit cartésien d’une liste d’attributs**.   
Elle possède un nom et contient une liste d’attributs.  
  
CLÉ PRIMAIRE

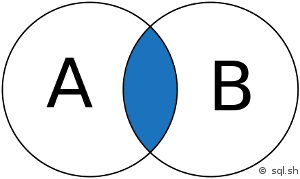
La clé primaire d’une relation est l’attribut ou l’ensemble d’attributs dont tous les autres attributs de la relation sont en Dépendance Fonctionelle.  
**Une relation ne possède qu’une seule clé primaire**.

CLÉ(S) ÉTRANGÈRE(S)

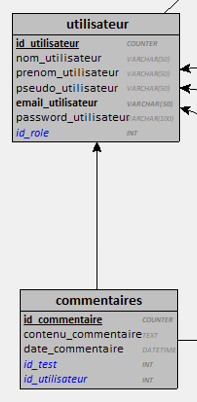
Une clé étrangère d’une relation est un attribut ou un ensemble d’attributs **qui a un rôle de clé primaire dans une autre relation**.  
Une relation possède 0 à plusieurs clés étrangères.   
Ce sont les clés étrangères qui permettent de faire les liens entre les relations.

INNER JOIN

Les jointures en SQL permettent d’**associer plusieurs tables dans une même requête**.  
Cela permet d’exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables en même temps de manière efficace.  
Les jointures SQL consistent à associer les lignes de deux tables en associant l’égalité des valeurs d’une colonne d’une seconde table.

  
On nomme ce concept : la condition.

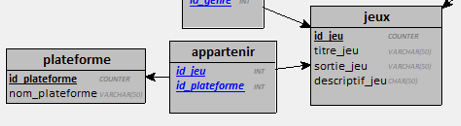
L’intérêt des jointures : Elles sont tout d’abord un moyen d’optimiser les processus, en améliorant leur vitesse d’exécution.  
Elles permettent également d’éviter les répétitions d’informations.

EXEMPLE

L’entité utilisateur devient une table, ses propriétés des attributs qui constitueront des objets.  
Son identifiant « **id\_utilisateur** » devient la clé primaire de cette table.

Pour la table **commentaires** avec une association de type 1,1 / 0,n : on peut observer la création d’une clé étrangère (Foreign Key) « **id\_utilisateur** » issue de l’entité voisine **utilisateur**.

On parle ici de **Type N**.

Pour la table **jeux** avec une association de type 0,n / 0,n : on peut observer la création d’une relation (table) qui possède **les deux clés primaires** des deux entités voisines.

1. **SQL**

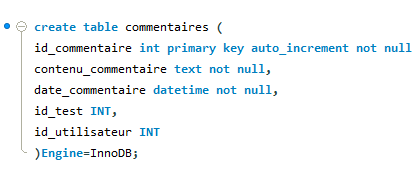
**ROLE DE MY SQL**

MySQL permet le stockage de données dans des tables séparées.  
Ces tables permettent de classer les données selon leurs caractéristiques.   
Elles sont organisées grâce à des lignes et des colonnes.

CRÉATION DE LA BASE DE DONNÉE SQL

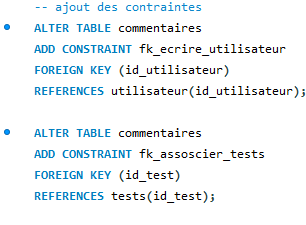
Pour la création de ma base de données en MySQL, j’ai utilisé Workbench.





Sur ce screen, on peut voir la création de la base de données avec la requête « CREATE DATABASE ».

J’ai fait un « use » juste derrière afin de préciser que chaque donnée que j’apporterais concerneront la BDD « overhaulgames ».  
Puis, on peut observer juste en dessous la création de la table « commentaires », où je viens donc apporter les attributs déjà décidé depuis le MCD, en prenant soin de préciser au préalable les « id\_test » et « id\_utilisateur » en prévision pour ajouter les Foreign Key :



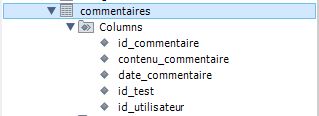
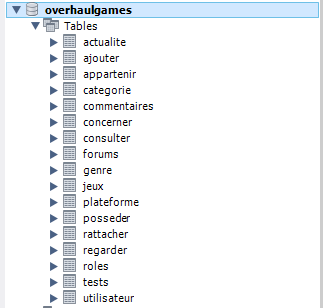
J’utilise la requête « ALTER TABLE » en premier lieu pour dire que je veux modifier la table renseignée, qui est donc « commentaires » ici.

Derrière, un « ADD CONSTRAINT » me permet d’ajouter une key, donc je précise son nom (« fk\_ecrire\_utilisateur » soit *« Type de key\_Nom de table d’association\_ Table qui est visée »)*

On précise que c’est une « FOREIGN KEY » en précisant quel attribut de la table « commentaires » est visé, puis à quel tables et attributs il fait référence.

VUE SUR MYSQL WORKBENCH

Voici ce à quoi ressemble l’arborescence depuis MySQL Workbench, avec toutes les tables de la base sur la gauche et l’exemple d’une des tables, ici « commentaires » sur la droite pour voir les colonnes et le typage.





1. **FONCTIONNALITÉS**

FONCTIONNALITÉ FRONT-END

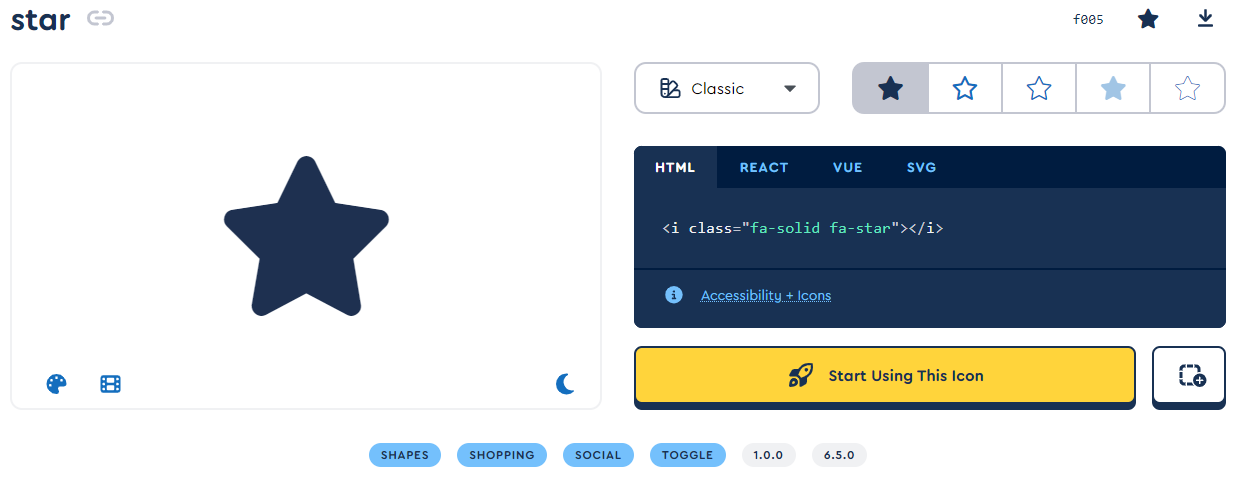
Dans cette partie, je vais vous présenter la fonctionnalité « Ajouter en favoris ».  
Comme évoqué plus tôt, c’est une fonctionnalités essentielle du site afin d’aider à l’accès pratique et rapide d’une information.

CODE HTML

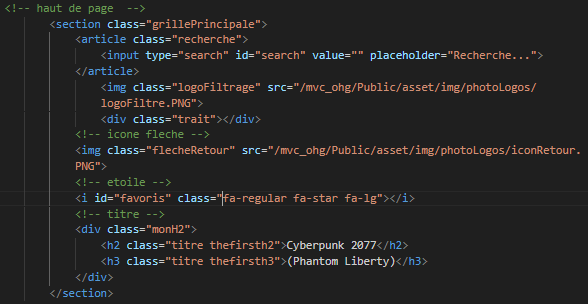


Ici, je viens implémenter le lien permettant d’accéder à la bibliothèque d’icones du site « Font Awesome ». On le met dans la partie « head » de la structure HTML.

Ensuite, dans le body, j’ai créé une section (comme on peut le voir avec la balise <section>) où j’ai renseigné quelque éléments.

Voici ce à quoi ressemble l’icône sur le site 

À l’aide de la documentation, on peut facilement personnaliser l’icône :



J’ai créé un ID « favoris » pour dans un premier temps me permettre d’appeler cet élément avec du Javascript, puis dans un second temps en profiter et lui donner quelque propriété en CSS comme par exemple sa couleur orangée pour correspondre à la charte graphique.

On peut aussi décider de la taille de l’icône avec une classe native, qui est « fa-lg ».

CODE JAVASCRIPT

Ici, on vient créer une variable « fav » me permettant de la relier à mon icône à l’aide de l’ID « favoris » précédemment créé.



Je viens ensuite créer deux fonctions :

* **addFav()** : Fonction me permettant « d’ajouter en favoris »
* **delFav()** : Fonction me permettant de « supprimer un favoris »

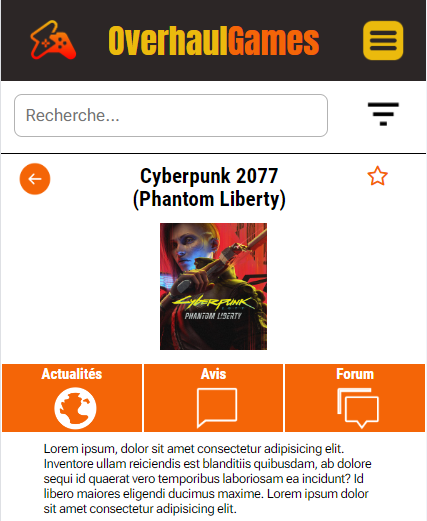
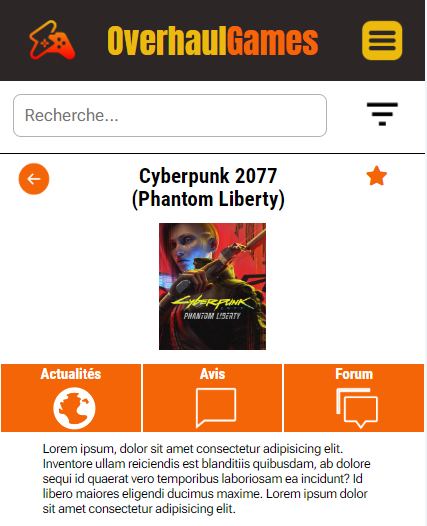
Je vais donc définir ce que sont les classes de l’icône : **« fa-regular »** permet à l’icône de n’afficher que les contours et **« fa-solid »** permet à l’icône de s’afficher complètement remplit.

La fonction addFav() va donc supprimer la classe « fa-regular » pour ajouter « fa-solid », et inversement pour la fonction delFav().   
  
Enfin, je viens créer un événement lors du click sur l’icône à l’aide de « addEventListener ».

Lorsqu’on clique sur l’icône, on vérifie si dans la « classList » est contenu la class « fa-solid ». Dans ce cas-là on lance la fonction pour supprimer un favori.  
À l’inverse, si il n’est dedans, c’est qu’il y a à la place « fa-regular » et on lance donc la fonction pour ajouter un favori.

Lors d’un click, l’icône sera donc soit remplie, soit vide et permettra à l’utilisateur de **visualiser rapidement** si un test, article ou autre est bien ajouté ou non en favoris.

RENDU FONCTIONNALITÉ

FONCTIONNALITÉS BACK-END

FONCTIONNALITÉ - INSCRIPTION

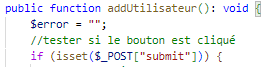
Le code HTML est assez simple, il est composé d’une balise « form », avec plusieurs « div » contenant chacune un « input » et un « label ». Il se trouve dans l’annexe page A REMPLIR

CODE PHP

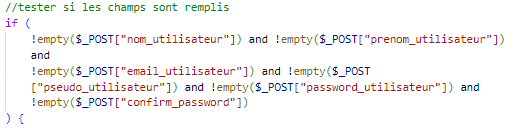
Ayant travaillé sur un modèle **MVC** (Modèle – Vue – Controller), le code ci-dessous correspond à la fonctionnalité d’**inscription** dans le **controller**.

Le code se compose de **4** conditions, qui sont des vérifications :

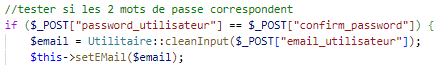
* On teste dans un premier temps si le bouton est **cliqué**, afin de lancer le script



* On vérifie dans un second temps si les **champs sont remplis** avec la fonction « empty » en précisant l’inverse avec « ! » pour le « contraire de vide »

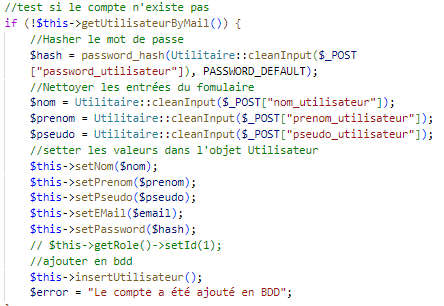


* Derrière, dans un troisième temps, on teste si les **deux mots de passes renseignés correspondent bien**, on **nettoie** l’entrée des emails pour « set » l’email renseigné avant de passer à la dernière vérification



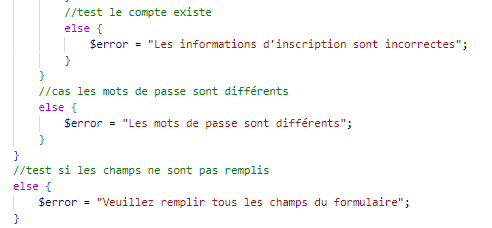
* On teste donc en dernier si le compte n’existe pas déjà (à l’aide d’une fonction créer au préalable « **getUtilisateurByMail** » qui comme son nom l’indique vient chercher un utilisateur à l’aide de l’email dans la BDD)

On peut donc hasher le mot de passe, « setter » les autres champs renseignés (en n’oubliant pas de les nettoyer juste au-dessus pour plus de sécurité) avant d’enfin ajouter dans la base les données à l’aide la fonction « **insertUtilisateur()**».



**Gestion des erreurs :**

On retrouve aussi la gestion des erreurs dans les « else », on peut observer par exemple un qui gère l’erreur d’un compte déjà existant.

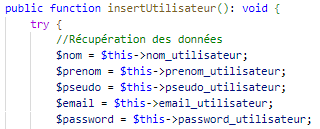


**Modèle**

Le code ci-dessous se trouve dans la partie **Modèle** et plus précisément dans l’objet Utilisateur qu’on a créé en prenant soin d’extends l’objet BddConnect qui est un objet permettant de se connecter à la base de données. Cela nous d’effectuer les requêtes que l’on veut en MySQL.

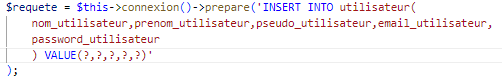


Dans cette fonction, on récupère les données avec « **$this** » qui sont d’ailleurs un peu au dessus de cette partie du code, qu’on met dans des variables.



Ensuite, on crée une variable $requête où on vient lui appliquer des fonctions native, qui sont connexion() (connexion à la base de données), puis prépare (on lui demander de préparer la requête).

Cette requête est donc un INSERT INTO dans la table utilisateur pour remplir les champs voulus. On rajoute un VALUE pour préparer nos 5 paramètres.



On rajoute alors sur la variable $requête un **bindParam(),** fonction native.



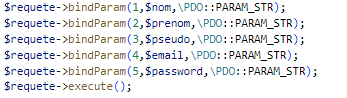
On renseigne dans la fonction le chiffre 1 correspondant au premier « ? » dans VALUE pour le connecter à la colonne « nom\_utilisateur » de la table « utilisateur » préalablement évoqué dans la requête « INSERT INTO ».

En deuxième paramètre, on renseigne ce qu’on va « set » dans la colonne visée par le 1, soit ma variable $nom qui sera remplie par le nom renseignée dans le champ correspondant qui vise nom\_utilisateur pour cet exemple (« $nom = $this->nom\_utilisateur » sur le premier screen).

En troisième et dernier paramètre, on trouve « \PDO ::PARAM\_STR » qui représente les types de données CHAR, VARCHAR ou les autres types de données sous forme de chaîne de caractères SQL.

On répète ce processus pour le nombre de colonne qu’on a choisi de remplir, soit ici 5 (nom, prenom, email, password).

On rajoute à la variable la fonction **execute()** qui va exécuter la requête.



On a pour finir la fonction le catch, venant après le try, gérant donc les messages d’erreurs.

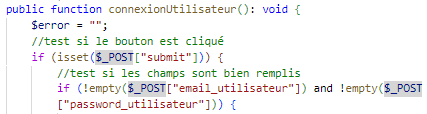


FONCTIONNALITÉ – CONNEXION  
Pour ce qui est du code HTML, il est similaire au code HTML de l’inscription avec encore moins de champs. Il se trouve donc lui aussi dans l’annexe, page A REMPLIR

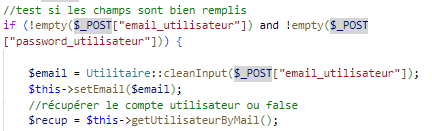
CODE PHP

**Controller**

Ici, le code est similaire à celui de l’inscription. 4 conditions, avec une gestion des erreurs. Les deux premières conditions sont les mêmes (Bouton cliqué ? Champs bien remplis ?)

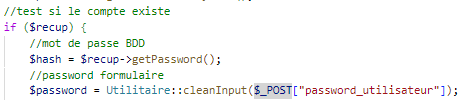


Dans la deuxième condition, on nettoie l’email et on vient « set » l’email. Ensuite, on crée une variable $recup à laquelle on attribue le compte correspondant à l’email.



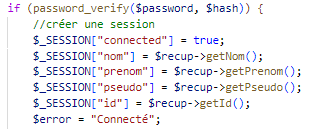
Dans la troisième condition, on fait un « if($recup) » pour vérifier si un compte rattaché à l’adresse email renseigné est bien existant.

On crée une variable $hash à laquelle on attribue le compte récupérer à l’aide de $recup, en demandant le mot de passe avec le getter « getPassword() ».  
On nettoie le champ du mot de passe qu’on met dans une variable $password.



Enfin, pour la dernière condition, on utilise la fonction native « password\_verify » qui sert à vérifier que le hachage donné correspond au mot de passe renseigné.

Si c’est le cas, on crée une session avec la superglobale $\_SESSION.

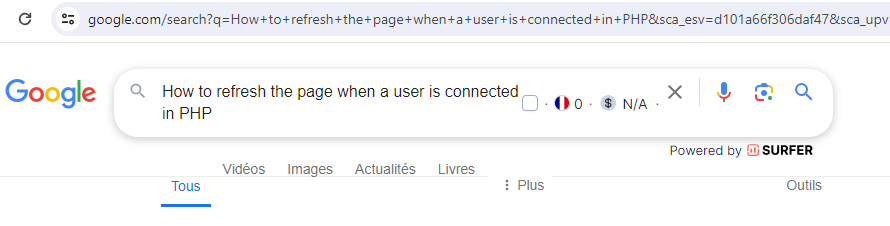


Une **superglobale** signifie simplement que cette variable est disponible dans tous les contextes du script. Il n’est pas nécessaire de faire « **global $variable ; »** avant d’y accéder dans les fonctions ou les méthodes.

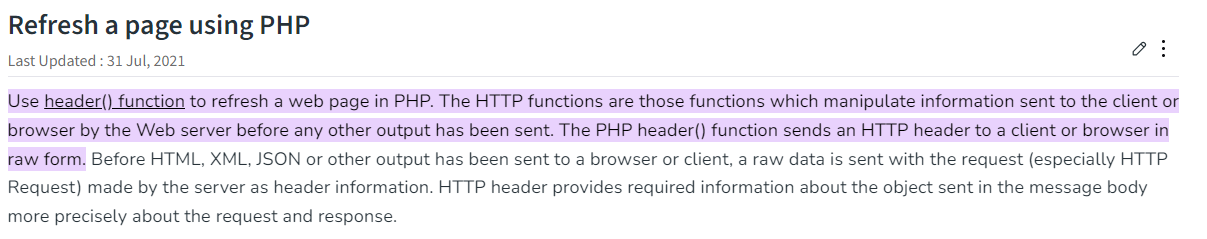
Après que l’utilisateur ait une session connecté, je fais un header(), fonction native, pour rafraichir la page sur laquelle l’utilisateur était avant la connexion. J’ai fais une recherche sur cette fonction, où j’en parle dans la partie recherche ci-dessous.

TRADUCTION : RECHERCHE EN ANGLAIS

When a user is connected to his account, I wanted the page to be automatically refreshed where he was previously. I already knew that a function like this exists because we studied one of them in the training (like « window.location.reload » in JavaScript) but I wanted a function in PHP for include it in my script at the moment when the condition for a user to be logged in is completed. So I searched that in the browser :

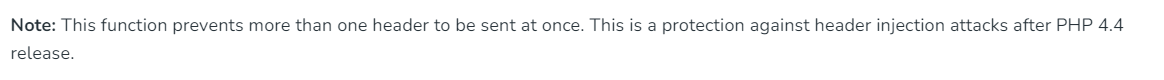


Je suis donc tombé sur le site internet « GeeksForGeeks » et vu cette réponse :

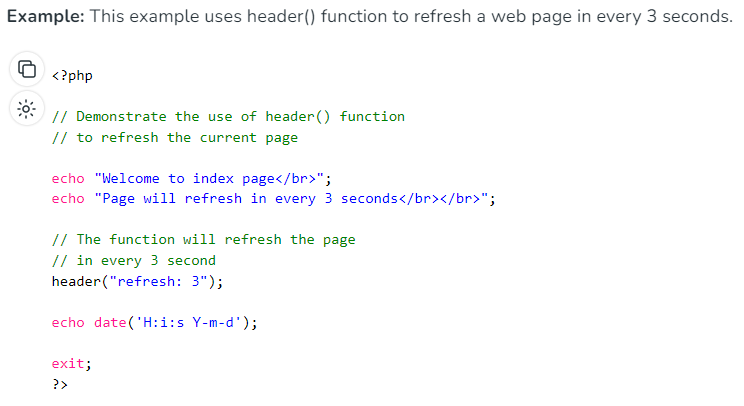


Ici, il est dit que pour recharger une page en utilisant PHP, il faut utiliser la fonction header().

Les fonctions HTTP sont des fonctions qui manipule des informations en les envoyant au client ou navigateur par le serveur web, avant l’envoi de tout autre sortie.   
Il est dit aussi que cette fonction envoie un en-tête HTTP (toujours au client/navigateur) mais sous forme brute.

Avant que l’HTML, XML, JSON ou tout autre sortie soit envoyée, des données brutes sont envoyées avec la requête (en particulier la requête HTTP) effectué par le serveur sous forme d’en-tête. Ce dernier fournit des informations requises par l’objet envoyé sur le corps du message, plus précisément sur le requête et la réponse.

Il est aussi précisé que cette fonction empêche l’envoi de plus d’une en-tête à la fois. C’est une protection contre les attaque à injections dans les en-têtes après la version 4.4 de PHP.



Exemple : Cet exemple utilise la fonction header() pour rafraichir une page internet toutes les 3 secondes.

J’ai donc adapté cette fonction pour la déclenché après un script lorsque l’utilisateur est connecté.



Il se trouve aussi dans l’annexe page A REMPLIR une fonctionnalité **Déconnexion**.

1. **CONCLUSION**