Dossier Projet

Pour le titre professionnel développeur Web / Web Mobile

S.O.M.M.A.I.R.E

[Introduction 3](#_Toc75908259)

[Résumé du dossier 3](#_Toc75908260)

[Première partie 3](#_Toc75908261)

[Seconde partie 3](#_Toc75908262)

[Premier projet 4](#_Toc75908263)

[Présentation de l’entreprise 4](#_Toc75908264)

[Présentation du projet 4](#_Toc75908265)

[Les cibles 4](#_Toc75908266)

[Enjeux 4](#_Toc75908267)

[Inventaire technique 5](#_Toc75908268)

[Principe du projet 5](#_Toc75908269)

[Cas d’utilisation 6](#_Toc75908270)

[Algorithme du script 7](#_Toc75908271)

[Exemple de document en anglais 8](#_Toc75908272)

[Second projet 10](#_Toc75908273)

[Présentation du jeu 10](#_Toc75908274)

[Présentation du projet 10](#_Toc75908275)

[Les cibles 10](#_Toc75908276)

[Spécifications fonctionnelles 10](#_Toc75908277)

[Critères 10](#_Toc75908278)

[Inventaire technique 11](#_Toc75908279)

[Arborescence du site 11](#_Toc75908280)

[Cas d’utilisation 12](#_Toc75908281)

[Maquettes et conception de l’interface 13](#_Toc75908282)

[Page d’accueil 13](#_Toc75908283)

[La page de l’admin 15](#_Toc75908284)

[Formulaire d’ajout d’un personnage 17](#_Toc75908285)

[Extrait de code Front End 19](#_Toc75908286)

[Architecture du Projet 21](#_Toc75908287)

[Modèle de données 22](#_Toc75908288)

[La base de données 22](#_Toc75908289)

[La sécurité 23](#_Toc75908290)

[Extrait de code Back End 24](#_Toc75908291)

[Tests 26](#_Toc75908292)

[TODO 27](#_Toc75908293)

# Introduction

## Résumé du dossier

### Première partie

Lors de mon stage pour la société Ibloo (installateur de fibre optique depuis 2019, basé à Tours) j’ai eu pour projet de créer une fonction qui récupère dans un rayon de 100 mètres, les informations des bâtiments autour une coordonnée se situant dans un json.

Cette fonction a pour but d’être utilisé pour vérifier l’éligibilité à la fibre optique de potentiel client.

### Seconde partie

Smite est un jeu vidéo de type MOBA (Multiplayer Online Battle Arena) développé et édité par Hi-Rez

Studios. Le développement de ce jeu a commencé en 2011 et il est sorti officiellement sur

PC, en 2014.

Le projet a pour objectif la sélection aléatoire d’un personnage de l’univers de Smite, ainsi que les objets que le joueur va devoir acheter au cours de la partie.

Cette application personnelle pourra possiblement s’étendre à un plus large public.

Premièrement, j’ai dû établir les spécifications fonctionnelles de l’application.

En second lieu, j’ai défini les spécifications techniques, ce après quoi il a fallu créer la base de données de l’application, avec toutes les liaisons nécessaires à son bon fonctionnement.

De plus il a fallu créer la maquette, choisir le design et les couleurs du site.

Ainsi, j’ai effectué le développement du site en commençant par la partie front, tout en respectant le mockup que j’avais mis en place.

Ce dossier comprend donc : L’expression des besoins, les spécifications fonctionnelles, les spécifications techniques, les documents liés à la création du site (mockup, arborescence, documentation…) et enfin, un extrait d’une documentation anglophone, que j’ai traduit moi- même, afin de l’utiliser pour la création de mon application.

# Premier projet

## Présentation de l’entreprise

Ibloo est une entreprise qui installe la fibre optique partout en France pour le particulier et le professionnel, depuis 2019.

L’entreprise étant basé à Tours, mon stage a donc été effectué entièrement en distanciel.

## Présentation du projet

Ce projet est mis en place pour que les potentiels futurs clients puissent vérifier s’ils sont éligibles à la fibre optique.

La partie que j’ai développé est une fonction qui permet de récupérer les bâtiments dans un rayon de 100 mètres de l’adresse entrée par le client. Cette fonction est réalisée en Python, j’ai donc dû apprendre le Python ainsi qu’utiliser certains modules de ce langage.

### Les cibles

Ce projet a pour cible les potentiels futurs clients qui souhaitent obtenir la fibre optique.

### Enjeux

1. Apprentissage et utilisation du langage Python.
2. Recherche dans les documentations (Notamment en anglais).
3. Utilisation des types de projections géographiques.

## Inventaire technique

**Python 3.9.5:** Utilisé pour la réalisation de la fonction

**Visual Studio Code :** L’éditeur de texte pour le développement du site

**Trello :** Utilisé pour l’organisation du projet

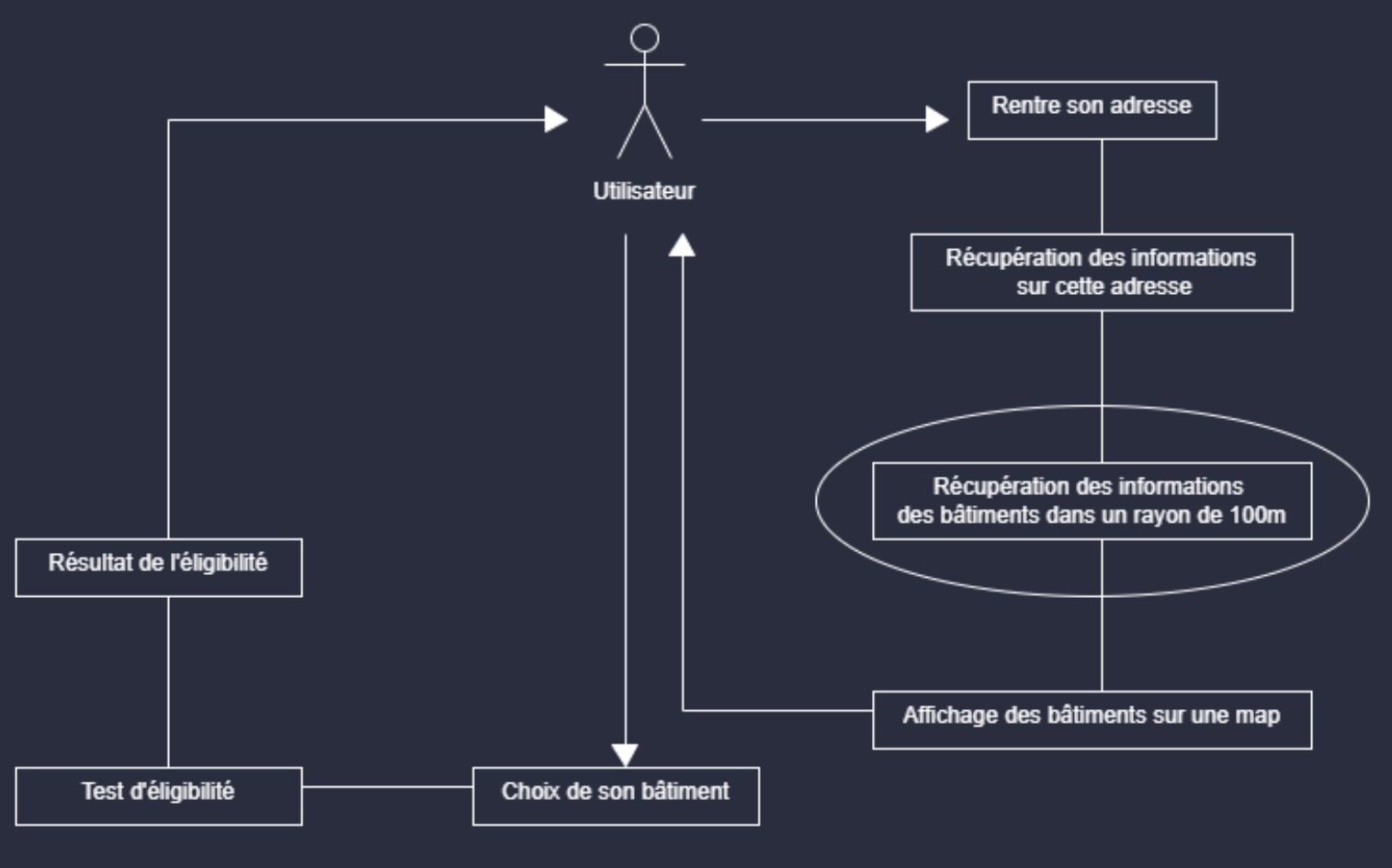
**Geo Data Viewer :** Pour visualiser sur une map.

## Principe du projet



Pour ce projet, j’ai travaillé sur une seule fonctionnalité, et en Local. J’ai donc dû me servir d’un json de test au lieu d’avoir accès directement à l’API Géo-Gouv.

## Cas d’utilisation

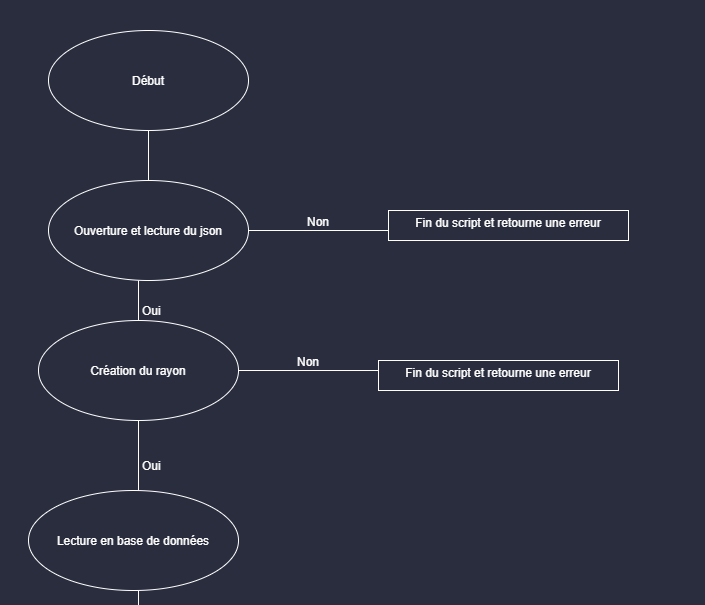


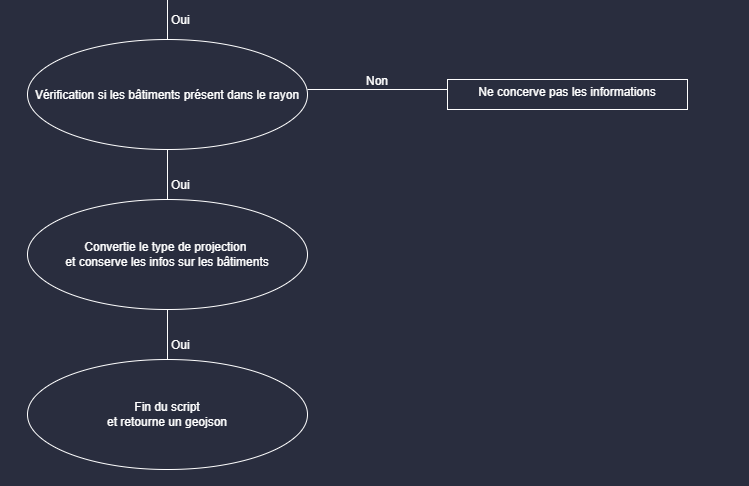
L’utilisateur rentre son adresse, l’API gouvernementale Géo-Gouv va alors envoyer un json avec des coordonnées qui correspondent à celle de l’adresse entrée par l’utilisateur.

Ensuite, une fonction va récupérer les informations des bâtiments dans un rayon de 100 mètres. Suite à cela, l’utilisateur verra s’afficher sur une map les bâtiments et devra choisir le sien.

Enfin une autre fonction va vérifier ensuite s’il est éligible à la fibre puis renverra à l’utilisateur le résultat.

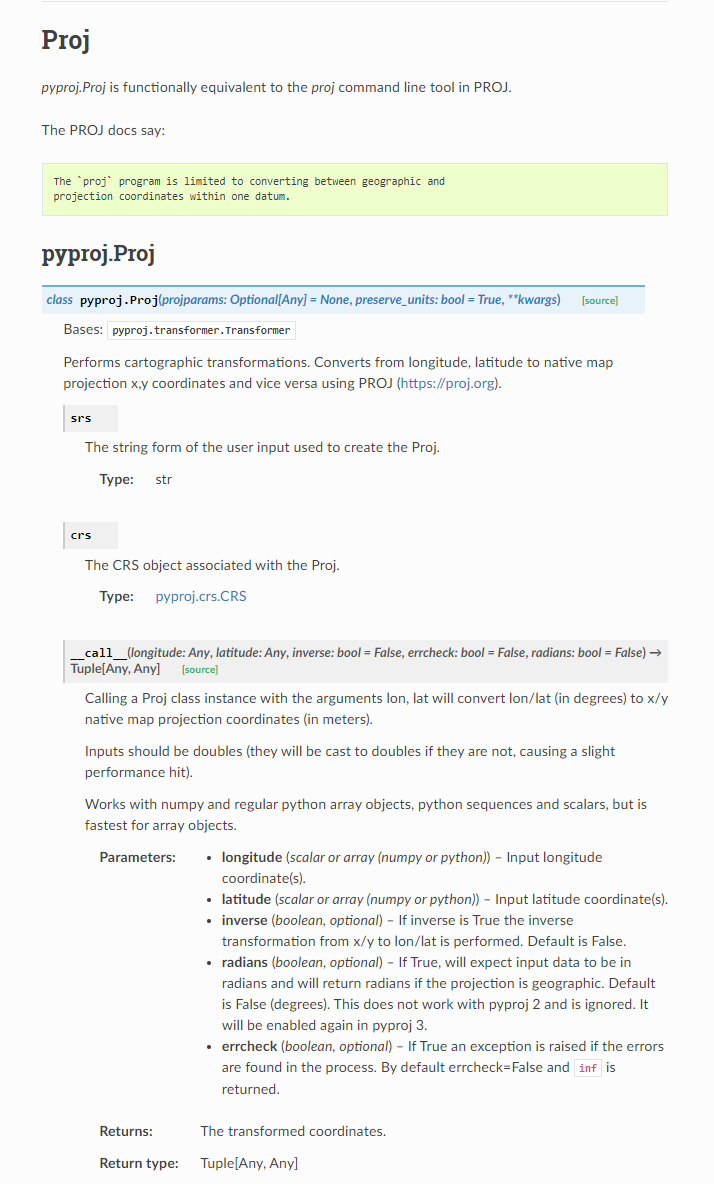
## Algorithme du script





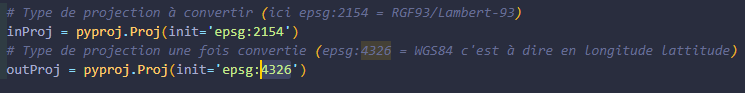
## Exemple de document en anglais

Je me suis servi de plusieurs documents afin d’apprendre le Python, mais aussi pour exploiter des modules. Voici un exemple de documentation expliquant la fonction Proj du module Pyproj.



Cette fonction sert à convertir les projections géographiques. Je m’en suis servi pour convertir du RGF93 (Projection géographique Française) en WGS 84 (Projection géographique Mondial).

*Variable de préparation à la conversion*



*Conversion de projection*



# Second projet

## Présentation du jeu

Smite est un jeu vidéo d'arène de bataille en ligne multijoueur en vue à la troisième personne, où les joueurs incarnent des divinités. Développé et édité par Hi-Rez Studios, le jeu est sorti le 25 mars 2014. Ce jeu servira donc de support pour ce projet.

## Présentation du projet

Ce projet est mis en place afin de pimenter une partie en y rajoutant du hasard et donc, de la difficulté. En effet, cette application choisit aléatoirement le personnage à jouer ainsi que les objets que le joueur devra acheter au cours d’une partie.

Il y a aussi une partie Back office avec la gestion complète des divinités.

### Les cibles

Ce site s’adresse aux joueurs ayant un minimum de connaissances sur le jeu, sur les dieux mais aussi sur les objets d’équipement des personnages.

### Spécifications fonctionnelles

Tirage aléatoire d’un personnage avec ces objets.

Gestion complète des personnages stockés en base de données :

* + Insertion de données
  + Mise à jour de données
  + Suppression de données
  + Affichage des données
  + Recherche des données

### Critères

L’écriture du code a été réalisée en POO (Programmation Orienté Objet).

## Inventaire technique

**HTML 5 :** Utilisé pour la version statique du site web.

**CSS 3 :** Pour l’habillage du site.

**Javascript :** Pour la rédaction du dossier projet.

**PHP 7.4.9 :** Utilisé pour rendre le site dynamique et pour la création des fonctions.

**Google Chrome et Opera GX :** Pour les différents test et l’optimisation du site.

**Visual Studio Code :** L’éditeur de texte pour le développement du site.

**Git :** Pour le contrôle des versions.

**GitHub :** Pour le stockage dans le cloud.

## Arborescence du site

Accueil

Connexion

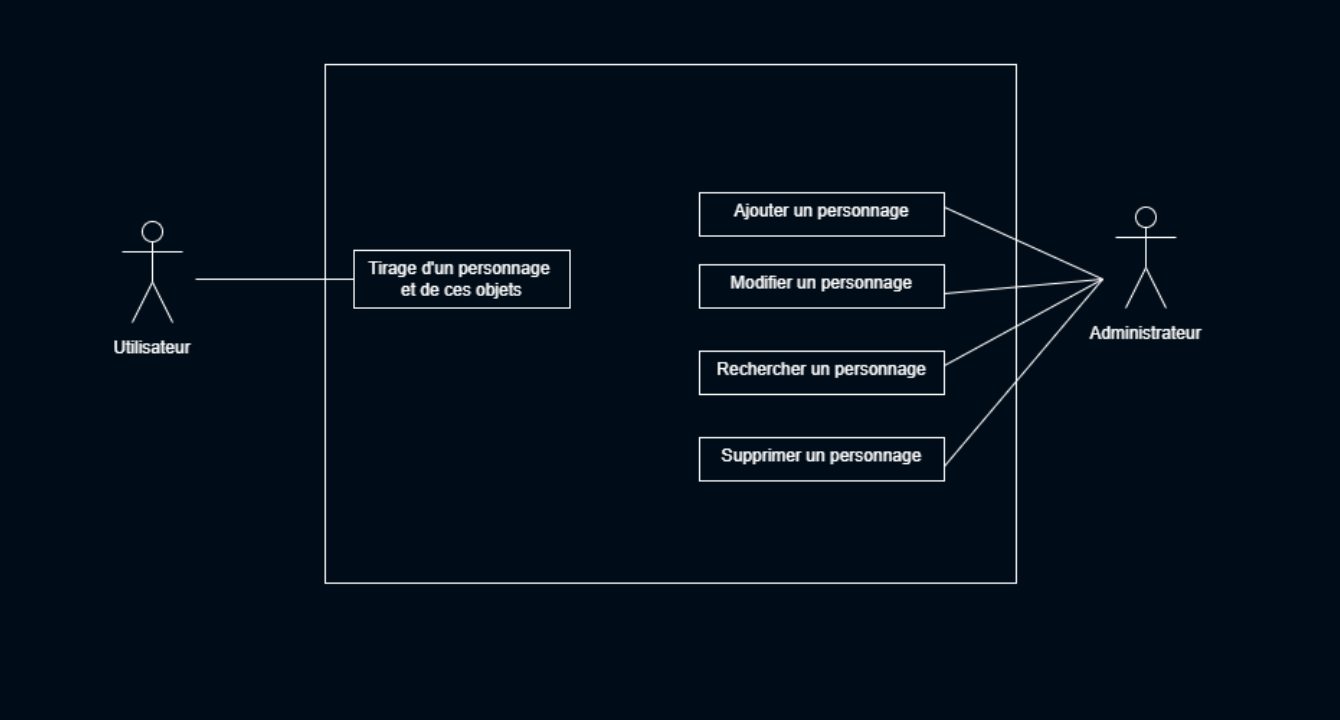
A l’Administration

Tableau de bord

Editer un personnage

Ajouter un personnage

## Cas d’utilisation



L’utilisateur n’a accès qu’à la page d’accueil et ne peut donc effectuer que des tirages au sort.

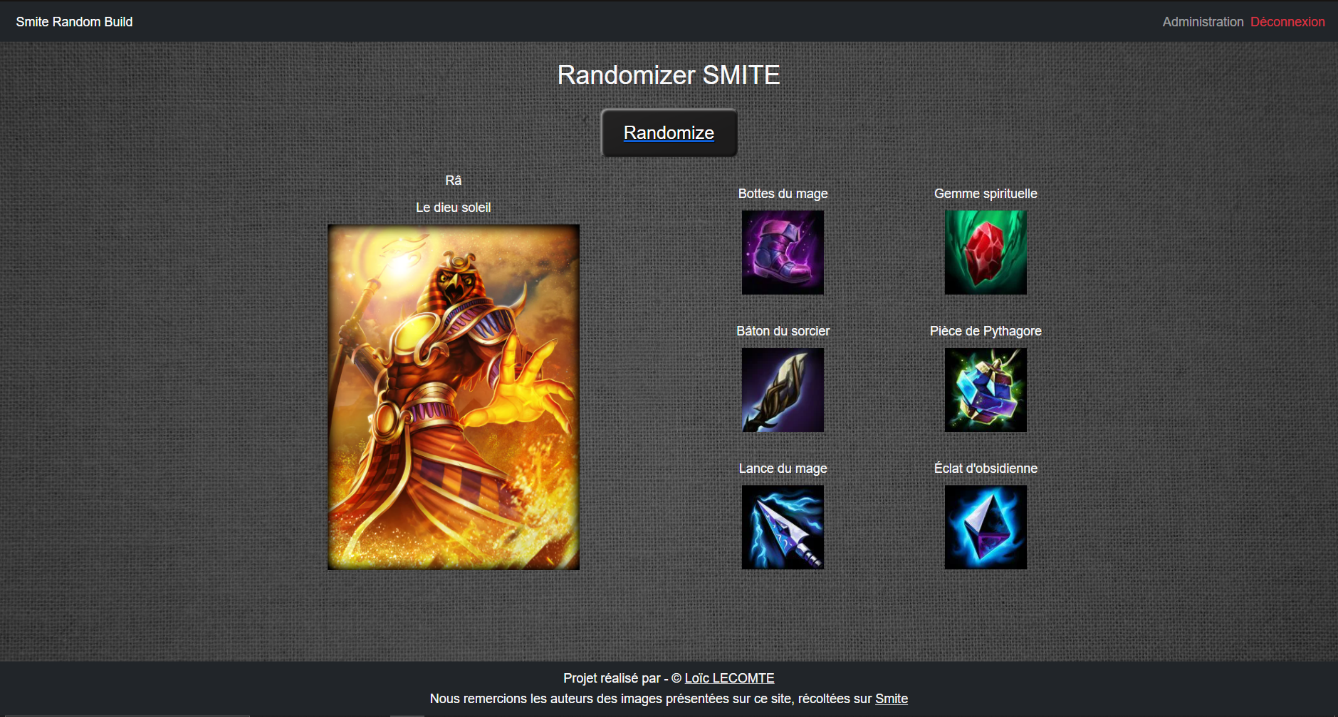
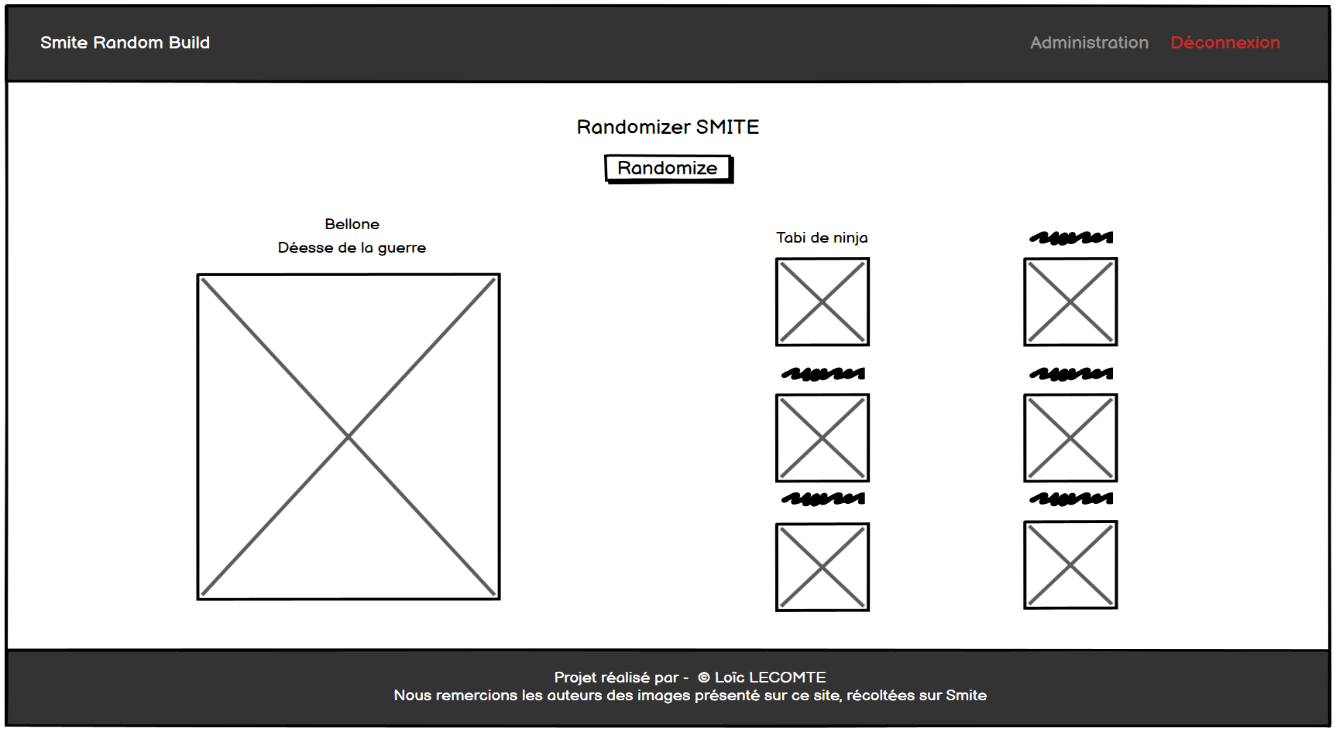
L’administrateur, lui, a accès a un back office afin de pouvoir gérer les divinités.

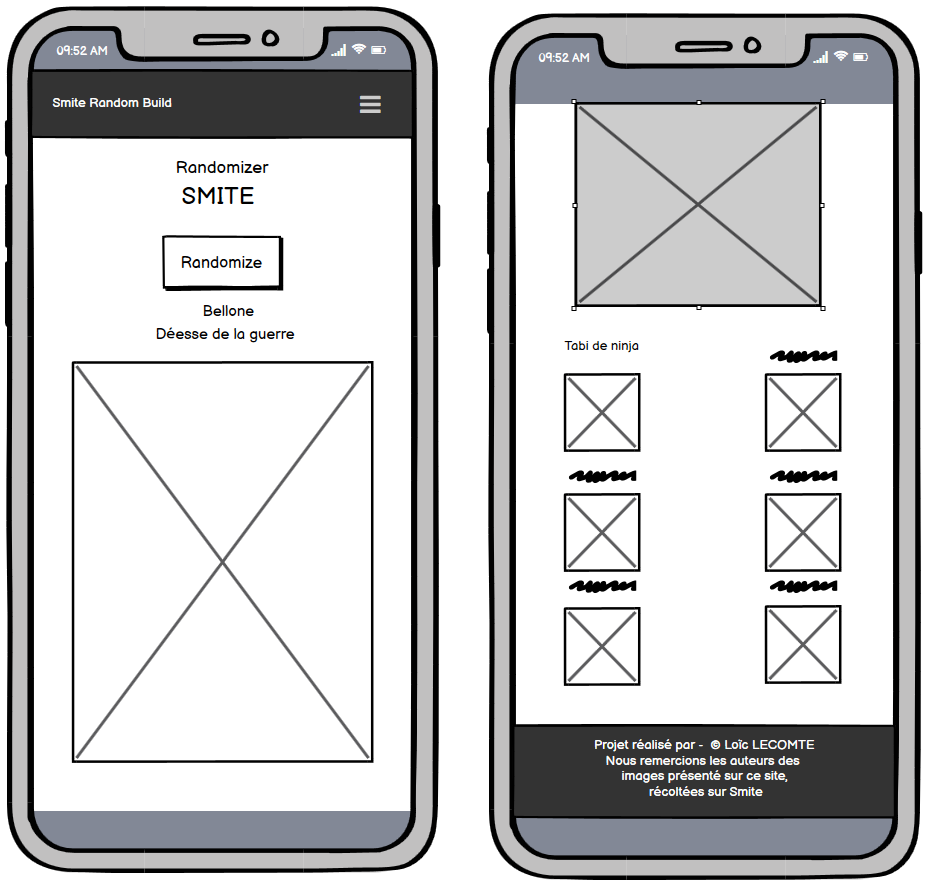
Il peut ainsi en créer, en supprimer, les modifier ou les rechercher.

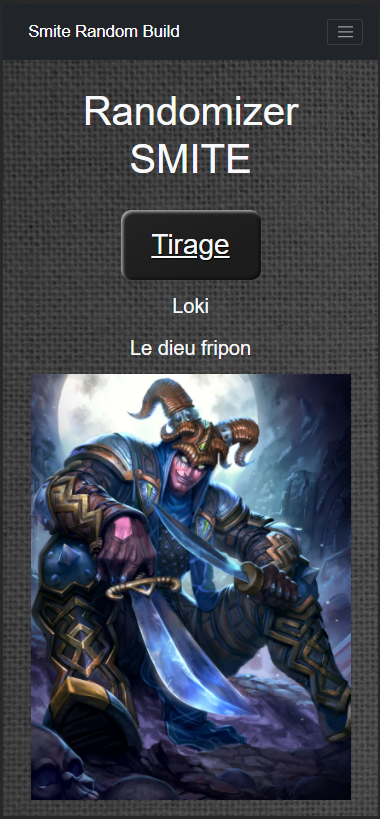
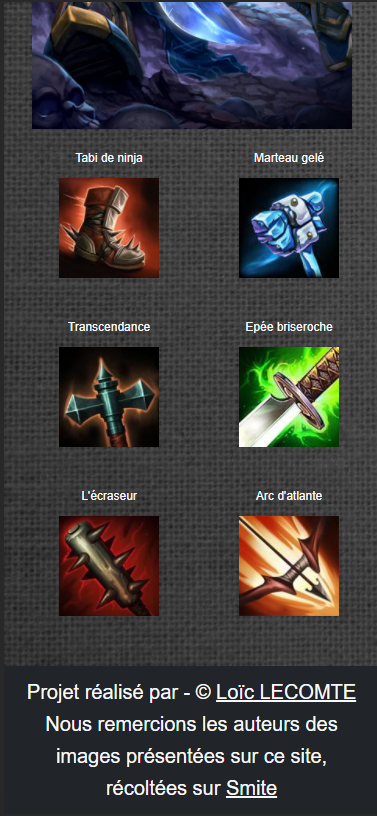
## Maquettes et conception de l’interface

### Page d’accueil

Obtention aléatoire du dieu et de ses objets et possibilité de refaire un tirage.





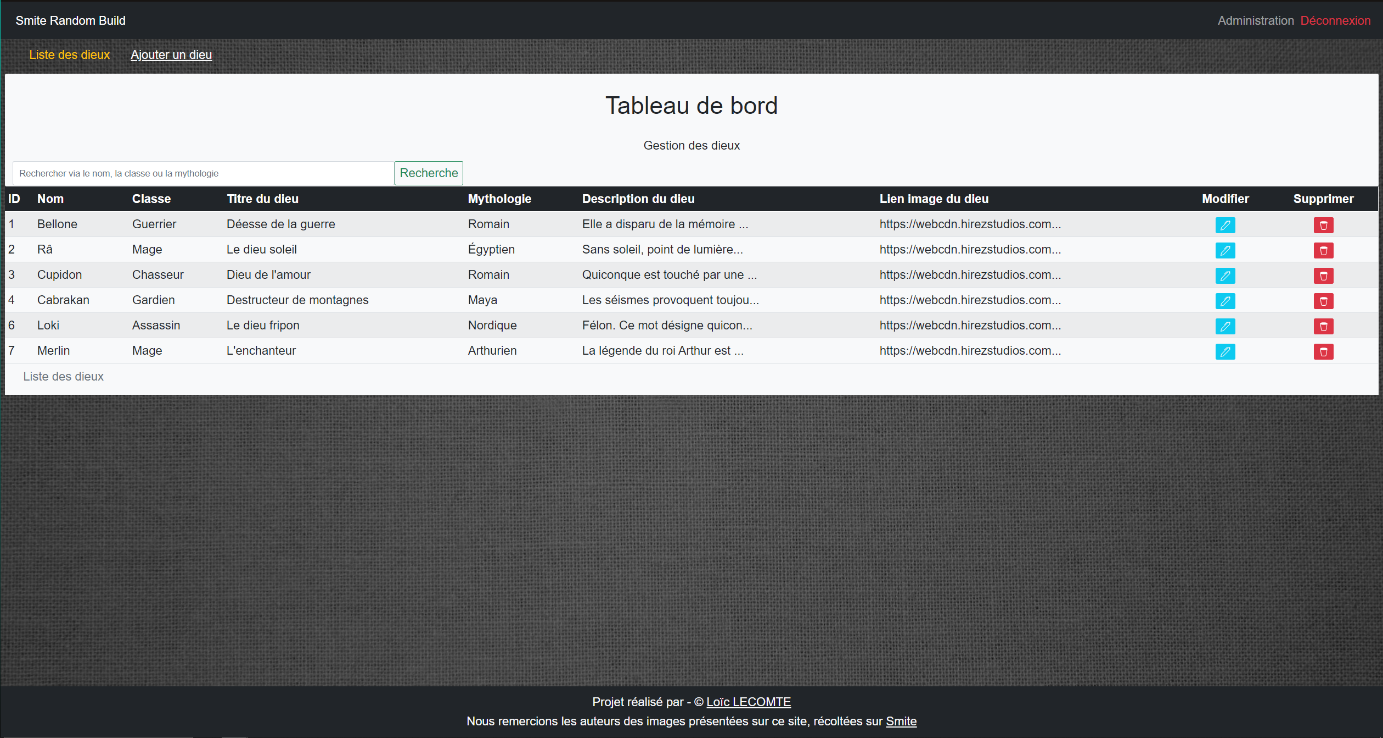
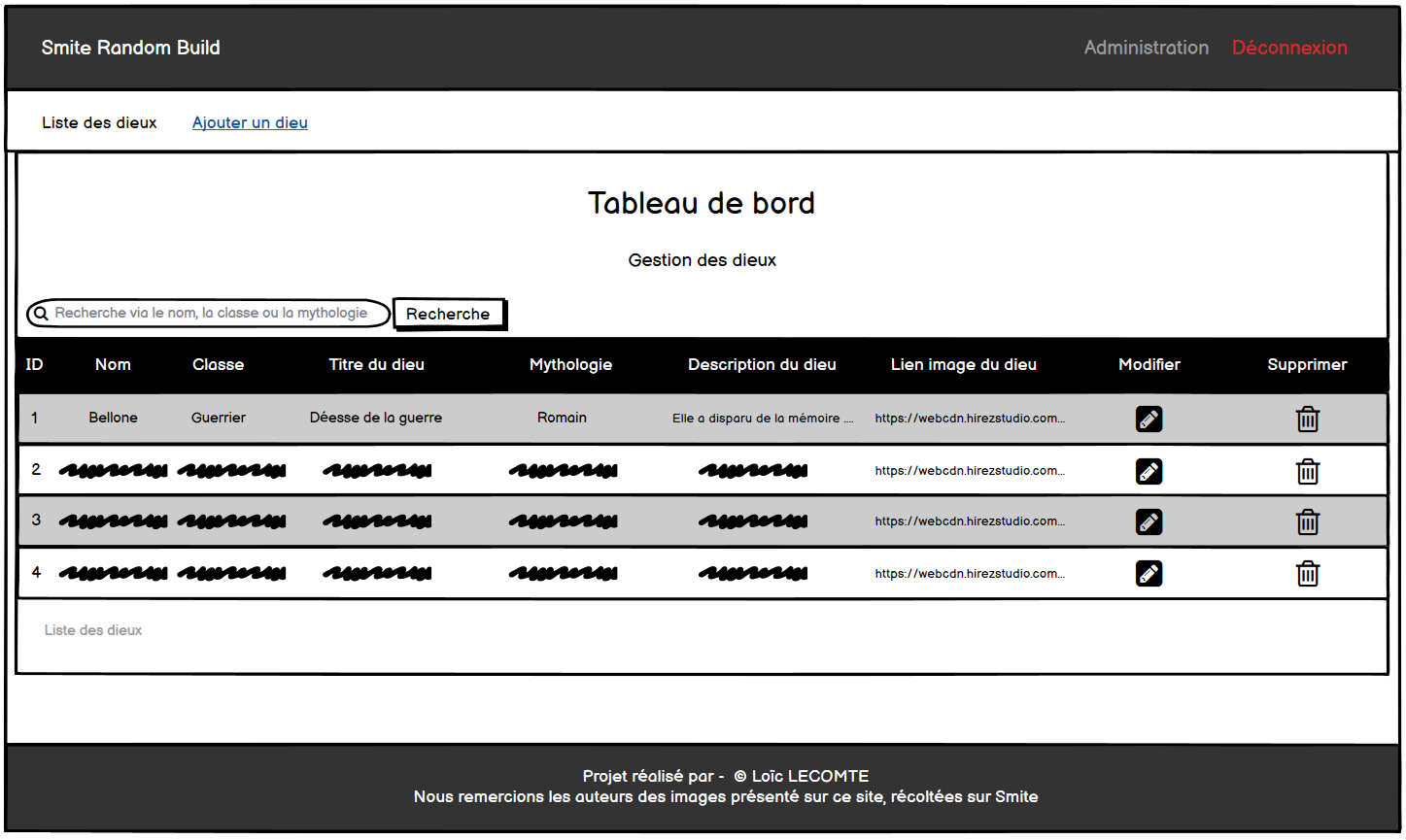
### La page de l’admin

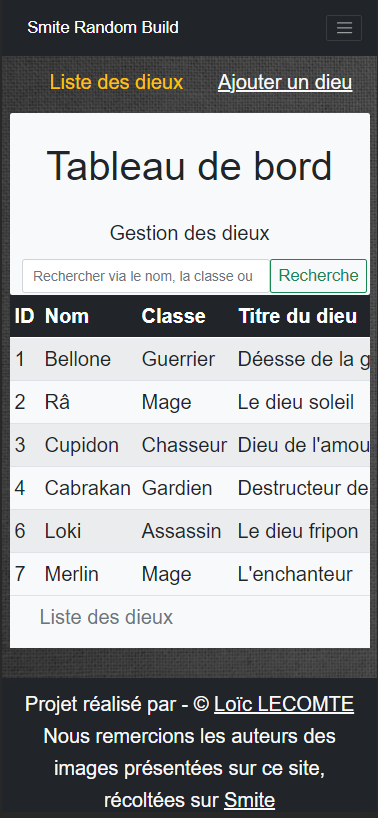
Page uniquement accessible par l’administrateur.

Visibilité sur toutes les divinités et leurs informations.

Possibilité d’accès à la page d’édition ou d’ajout d’un dieu.

Possibilité de suppression d’un personnage.

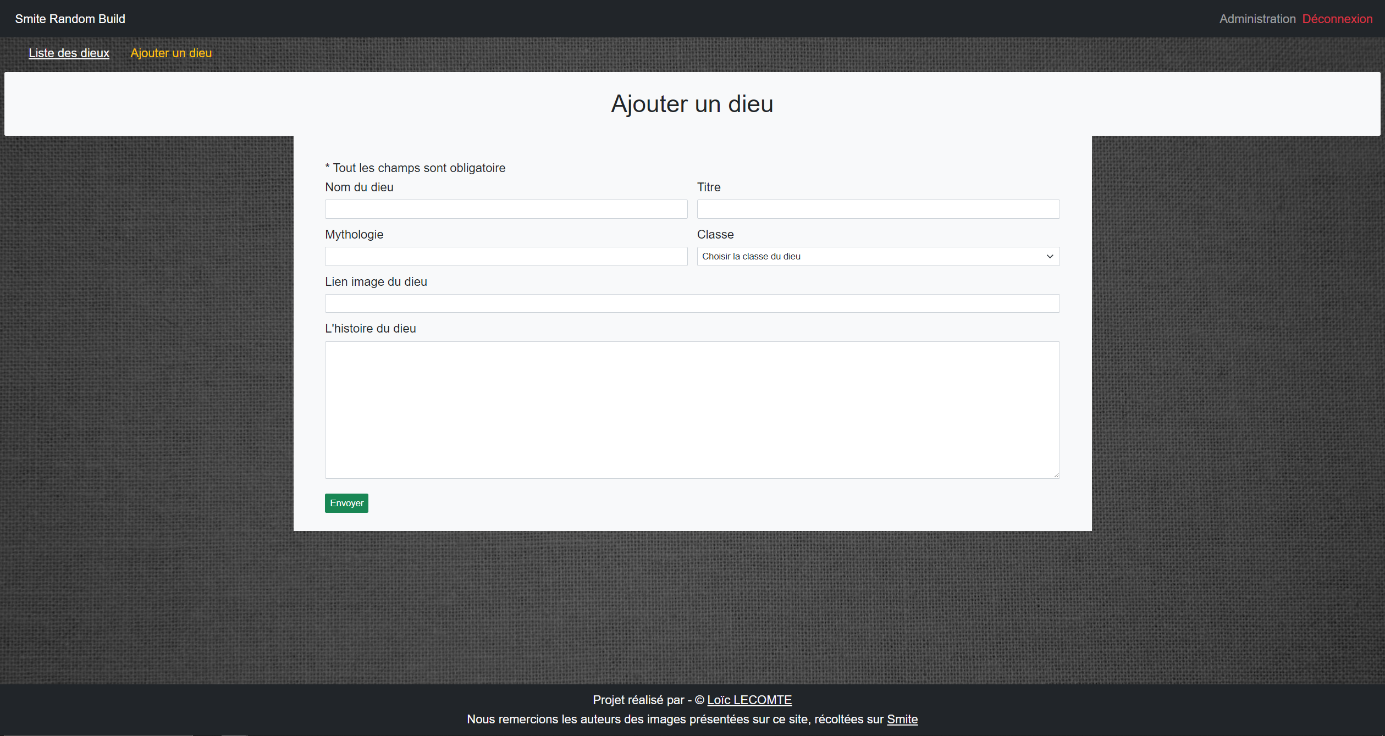


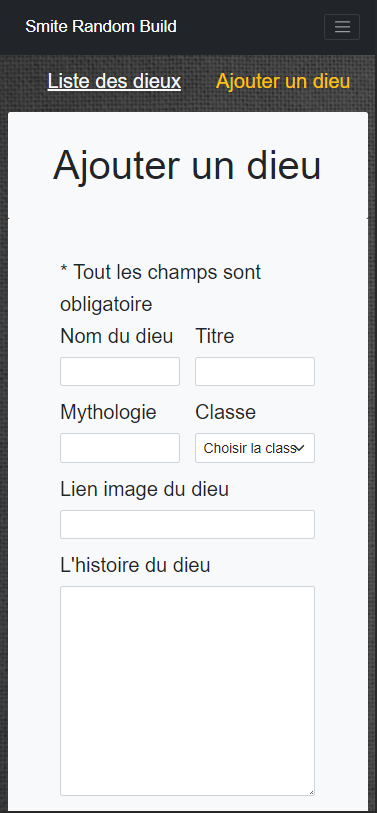
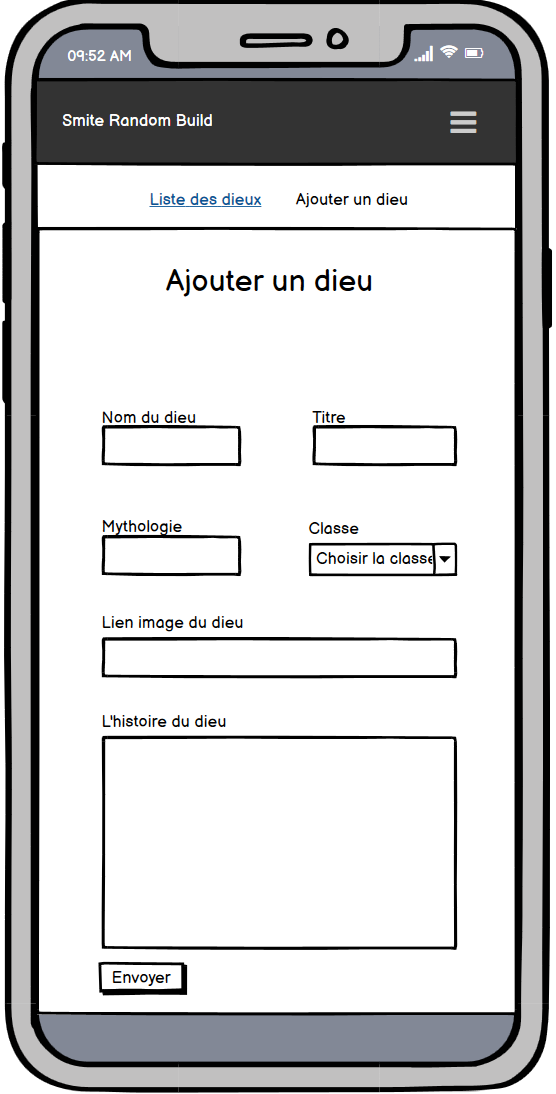
 

### Formulaire d’ajout d’un personnage

Voici le formulaire pour ajouter un personnage, la page d’édition est similaire à ce formulaire.

Dans les champs du nom, du titre et du lien de l’image, il est impossible de mettre une donnée déjà existante dans la base de données.



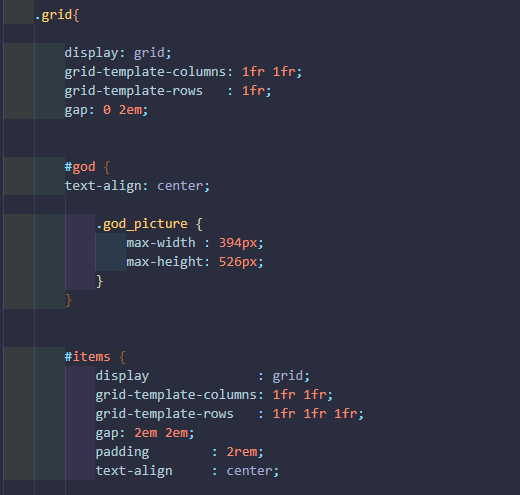


## Extrait de code Front End

**Extrait HTML de la page d’accueil**

****

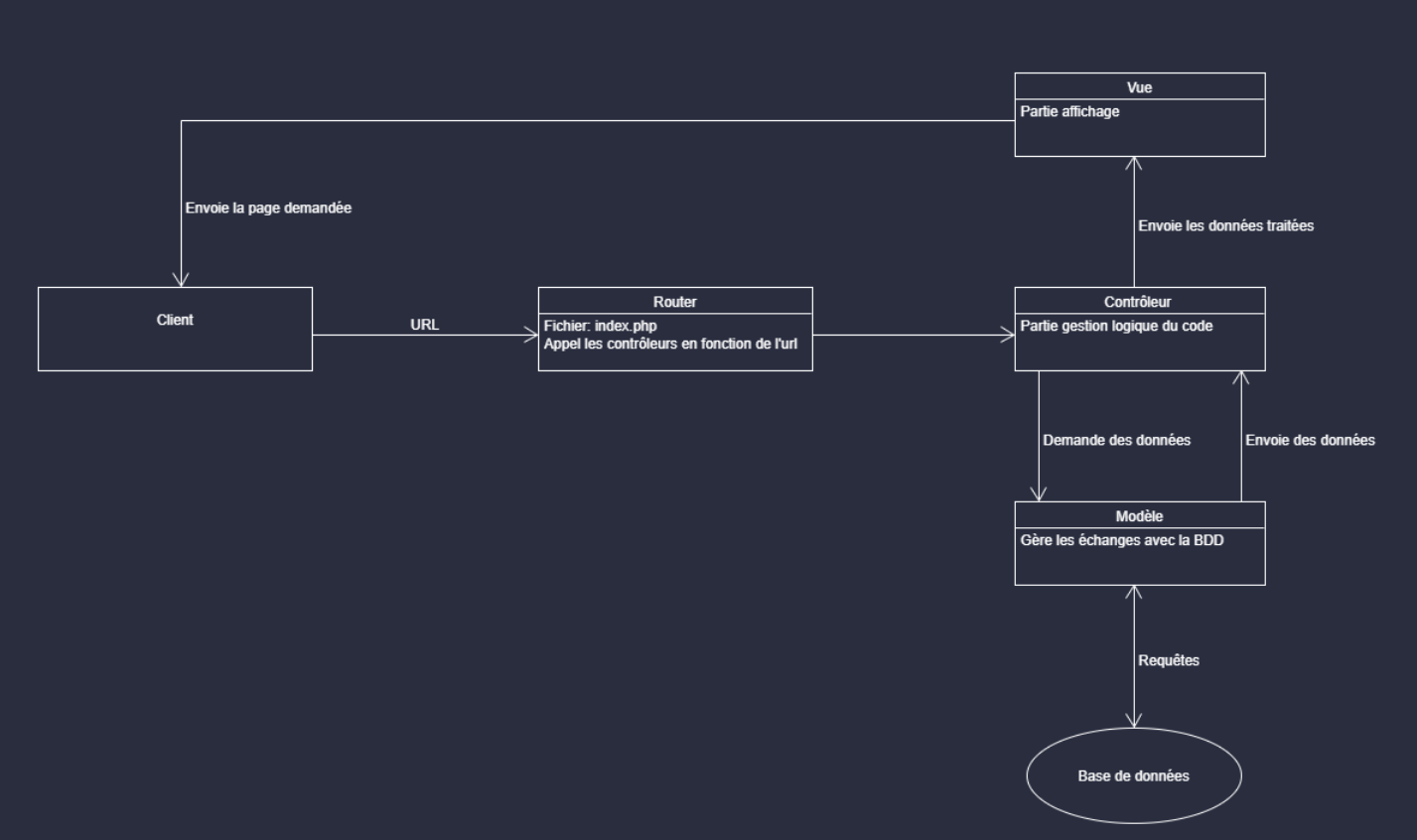
**Extrait du CSS de cette même page**

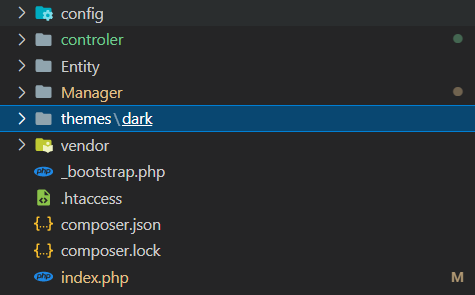
****

**Extrait de Javascript représentant l’affichage d’une info bulle lors du focus sur une image**



## Architecture du Projet





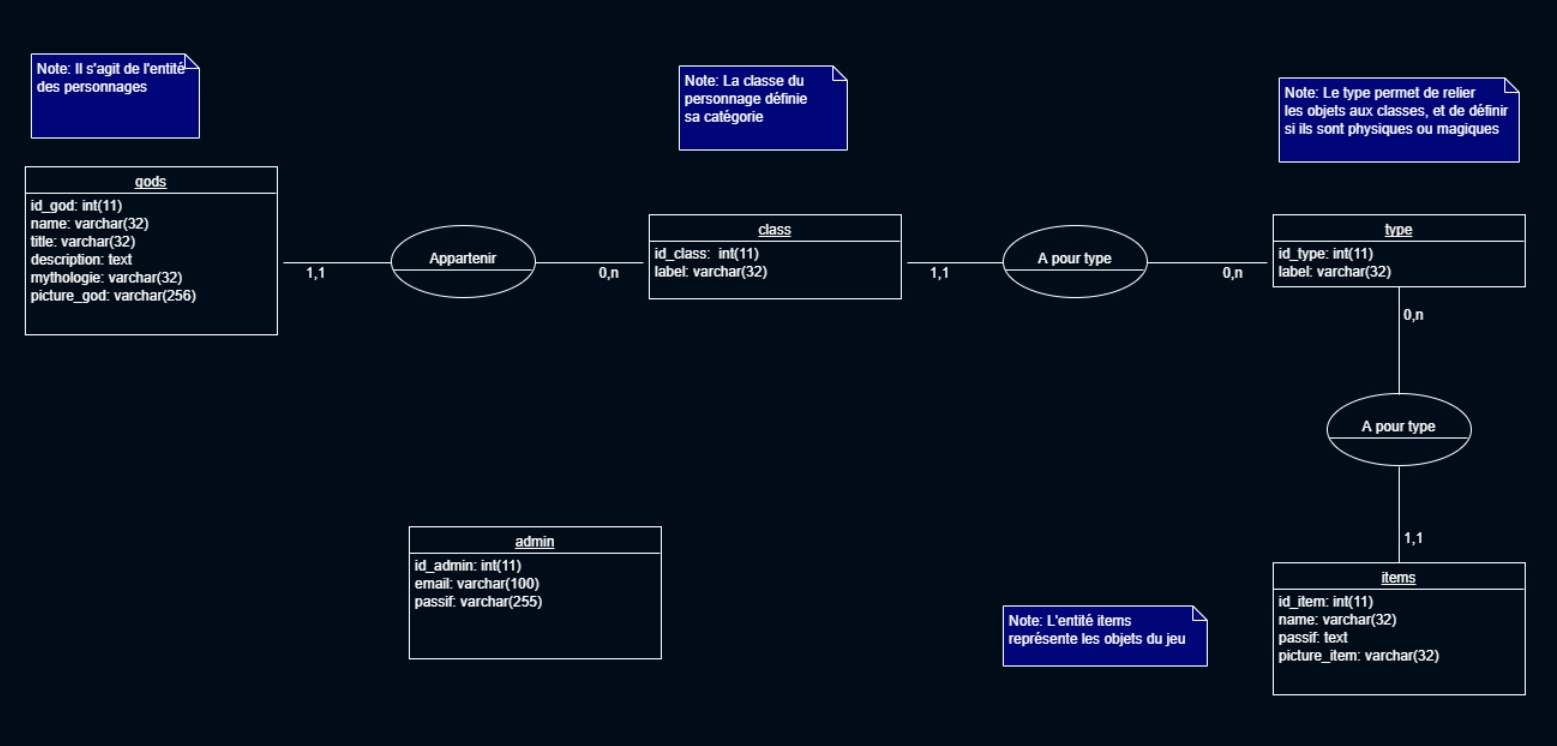
Mon router est le fichier index.php

Mes contrôleurs se situent dans le dossier controler

Mes modèles sont dans les dossiers Entity et Manager

Et mes vues sont dans le sous dossier « dark » du dossier themes.

## Modèle de données

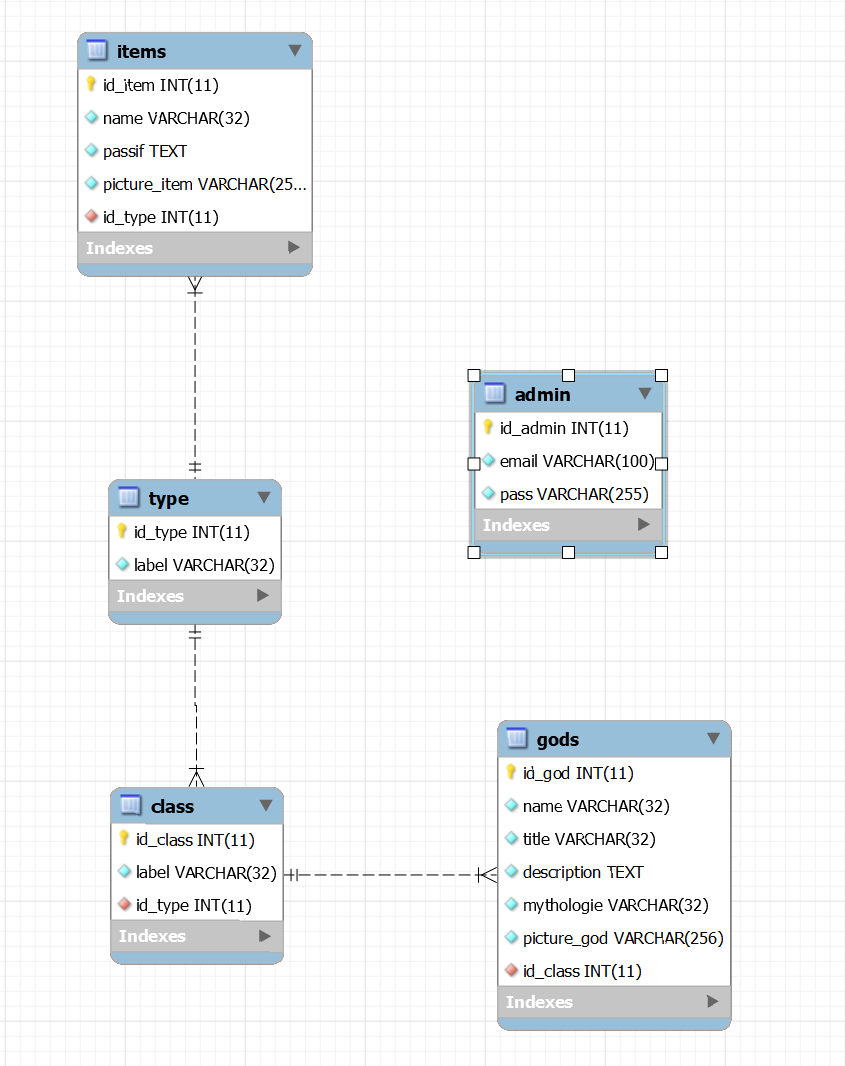


Un dieu ne peut avoir qu’une seule classe. Par contre, une classe peut ne pas avoir de personnage ou en avoir plusieurs (relation One to Many).

Une classe a toujours un type, et seulement un type. Un type peut ne pas avoir de classe ou en avoir plusieurs (relation One to Many).

Un objet a toujours un type, et seulement un type et un type peut ne pas avoir d’objet ou en avoir plusieurs (relation One to Many).

## La base de données



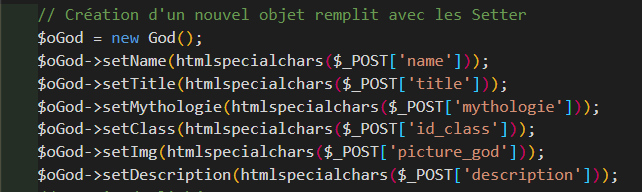
## La sécurité

Pendant mon projet, j’ai été confronté à deux failles de sécurité :

La faille XSS

C’est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page.

Pour s’en protéger, je me suis servis de la fonction htmlspecialchars() qui convertit tous les caractères spéciaux en entités HTML



L’injection SQL

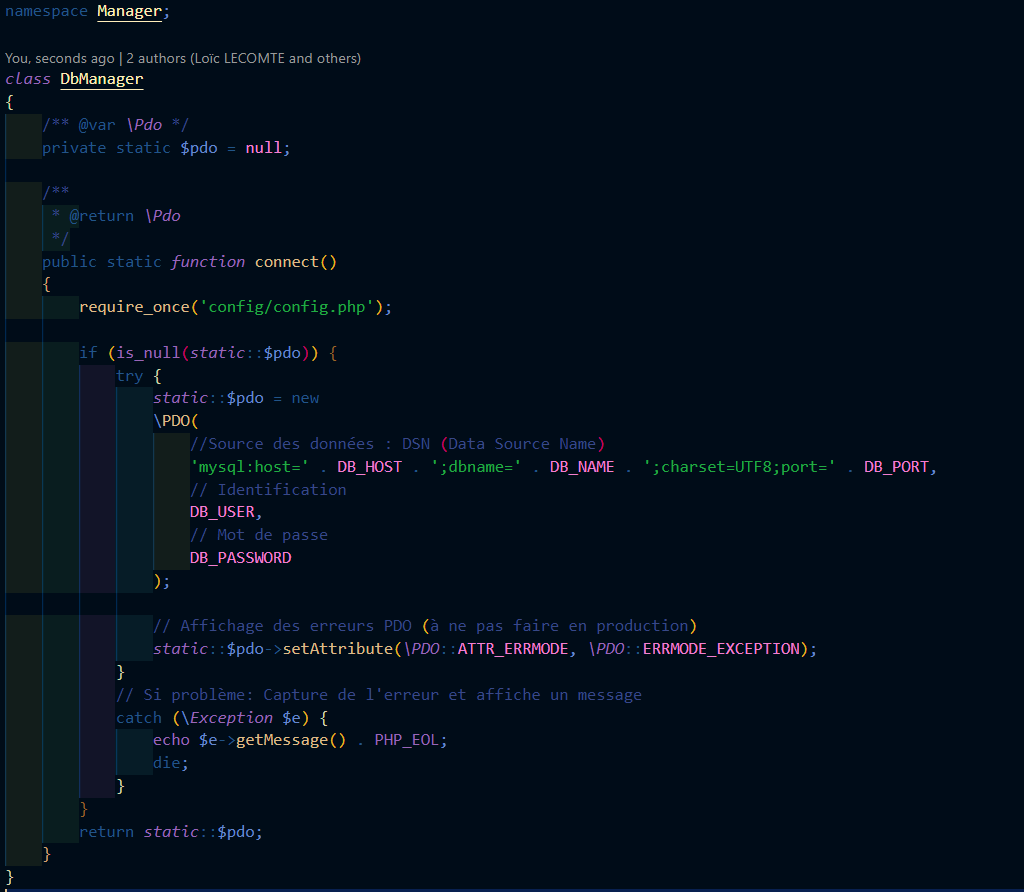
C’est une faille de sécurité qui modifie une requête SQL existante pour afficher des données cachées, ou pour écraser des valeurs importantes, ou encore exécuter des commandes dangereuses pour la base.

Pour s’en protéger, j’ai utilisé des requêtes préparées.



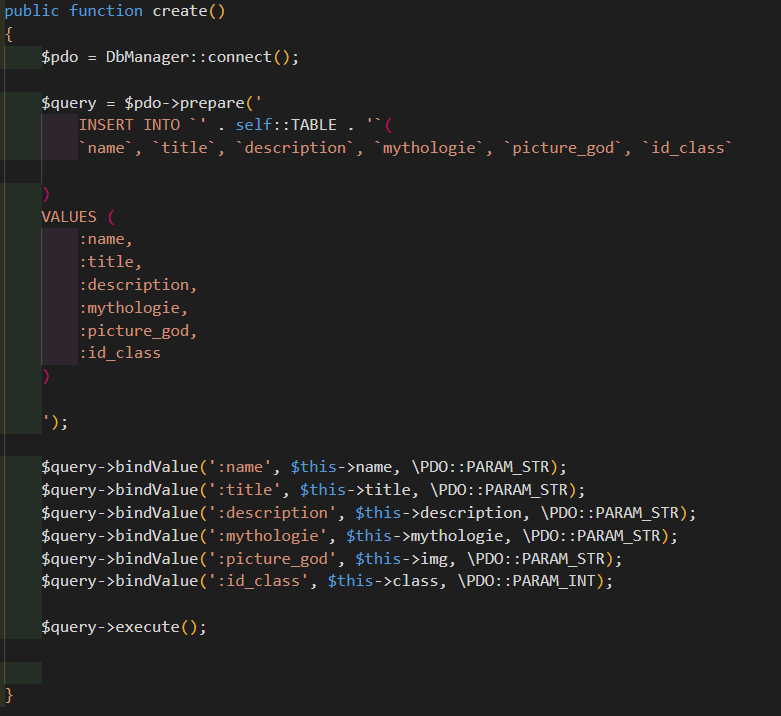
## Extrait de code Back End

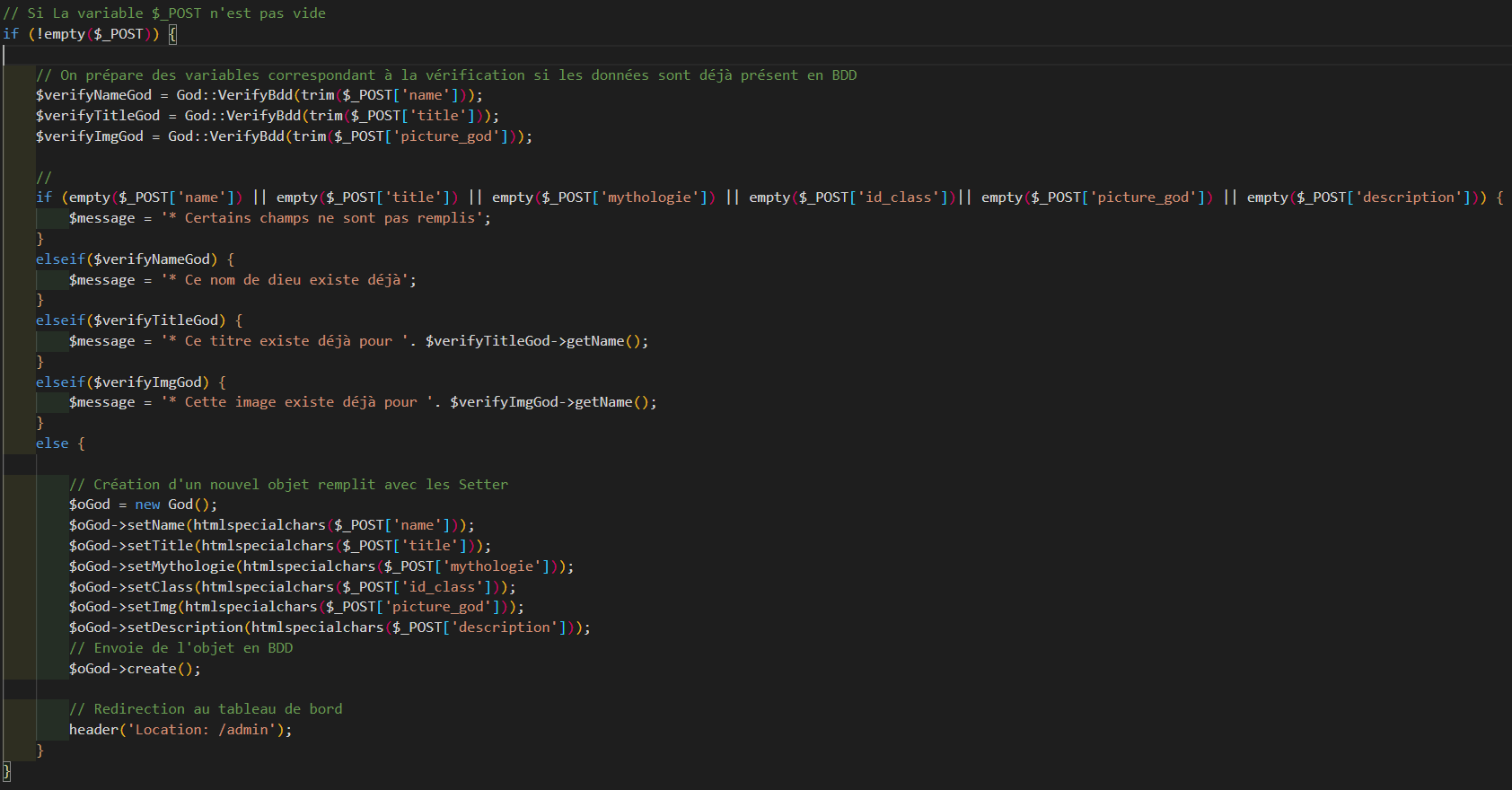
**Classe DbManager pour la connexion en base de données**

****

Les constantes stockant la DSN (Data Source Name), sont archivés dans le fichier config.php

**Classe God pour la gestion des dieux : fonction d’ajout d’une divinité**

****

**Mon contrôleur de la page pour ajouter une divinité**

## Tests

Seul un administrateur connecté peut gérer les personnages enregistrés.

On va donc tester l’affichage, l’insertion, l’édition et la suppression des données.

|  |  |  |
| --- | --- | --- |
| Module | Tests | Résultat du test |
| Afficher les personnages | On vérifie qu’il y ait bien des comptes dans la base de données, si oui, on les affiche dans un tableau | On affiche toutes les informations de tous les personnages présents en base de données |
| Ajouter un personnage | L’administrateur a un formulaire pour créer un compte | * Si le formulaire est vide, on affiche un message d’erreur demandant à l’administrateur de remplir tous les champs. * Si le nom existe déjà en base de données, alors un message d’erreur apparaît. * Si le titre ou le lien de l’image existe déjà en base de données, alors un message d’erreur est retourné avec le nom du personnage possédant déjà cette information * Sinon, le personnage est créé puis stocké en base de données, et l’utilisateur est redirigé vers le tableau de bord. |
| Modifier un personnage | L’admin clique sur le bouton pour éditer un compte, puis est redirigé sur un formulaire remplit par défaut par les infos du personnage sélectionné | * Si le formulaire est vide, on affiche un message d’erreur demandant à l’utilisateur de remplir le champ. * Si le nom, le titre ou le lien de l’image existe déjà en base de données (sauf si c’est le nom, le titre et/ou le lien de l’image du personnage en cours de modification), alors un message d’erreur apparait. * Sinon les informations sont modifiées en base de données et l’administrateur est redirigé vers la page de gestion des comptes. |
| Supprimer un personnage | On clique sur le bouton pour supprimer un compte.  Une fenêtre de validation apparait alors. | * Si on valide le personnage est supprimer en base de données * Sinon l’action est annulée. |

## TODO

* Affinage du CSS
* Toute la partie CRUD sur les objets
* Limiter le nombre de dieux affichés et créer une pagination en AJAX
* Ajouter animation sur les images
* Ajouter des métadonnées pour le référencement naturel
* CRUD Web Storage pour garder les 3 dernières tirages