Projet FADDEV 31-21-02

**Rotten potatoes**

**Site communautaire de notation et de critique de jeux vidéos**

Par Sébastien Leyrissoux

## **Introduction**

This project is based on the following fictive scenario : The website Rotten tomatoes want to expand his universe with a new website dedicated to the notation and review of video games by the community. To achive this objective, they wish to make a whole new interface while keeping the original website indentity with the new name Rotten potatoes. This new website will be realised from scratch with the following languages : HTML, CSS, javascript, PHP and will be linked to a mySQL database.



## **REAC**

Ce projet couvre les compétences suivantes :

1 – Maquetter une application

2 – Réaliser une interface utilisateur web dynamique

3 – Développer une interface utilisateur web dynamique

5 – Créer une base de données

6 – Développer les composants d’accès aux données

7 – Développer la partie back-end d’une application web ou web mobile

## **Analyse du besoin**

Présentation de l’entreprise

Rotten tomatoes est un site web dédié aux informations et aux critiques de films et de séries, séparées en critiques de la communauté et critiques professionnelles. On y retrouve des informations sur les dernières sorties, les films/séries les plus populaires du moment, ainsi que des listes diverses et variées comme les films par réalisateur, par année de sortie, par acteur… et divers articles écrits par les membres de l’équipe en charge du site.

Contexte – besoin

Comme cité dans l’introduction, ce projet représente une demande fictive du site Rotten tomatoes, le besoin étant d’étendre l’univers des médias traités par le site via un tout nouveau site dédié à la critique de jeux vidéo. Le but étant de retrouver les fonctionnalités présentes sur le site original, on pourra donc y visualiser les dernières sorties, les sorties prochaines, noter et poster une critique des jeux

Contraintes techniques

Temps : C’est un grand projet pour un développeur en apprentissage, réussir à mener l’intégralité du projet à son terme dans le temps impartit sera compliqué

Interface responsive : Une interface adapté aux différents formats d’écrans est devenue la norme, il faudra donc mettre en place une solution responsive.

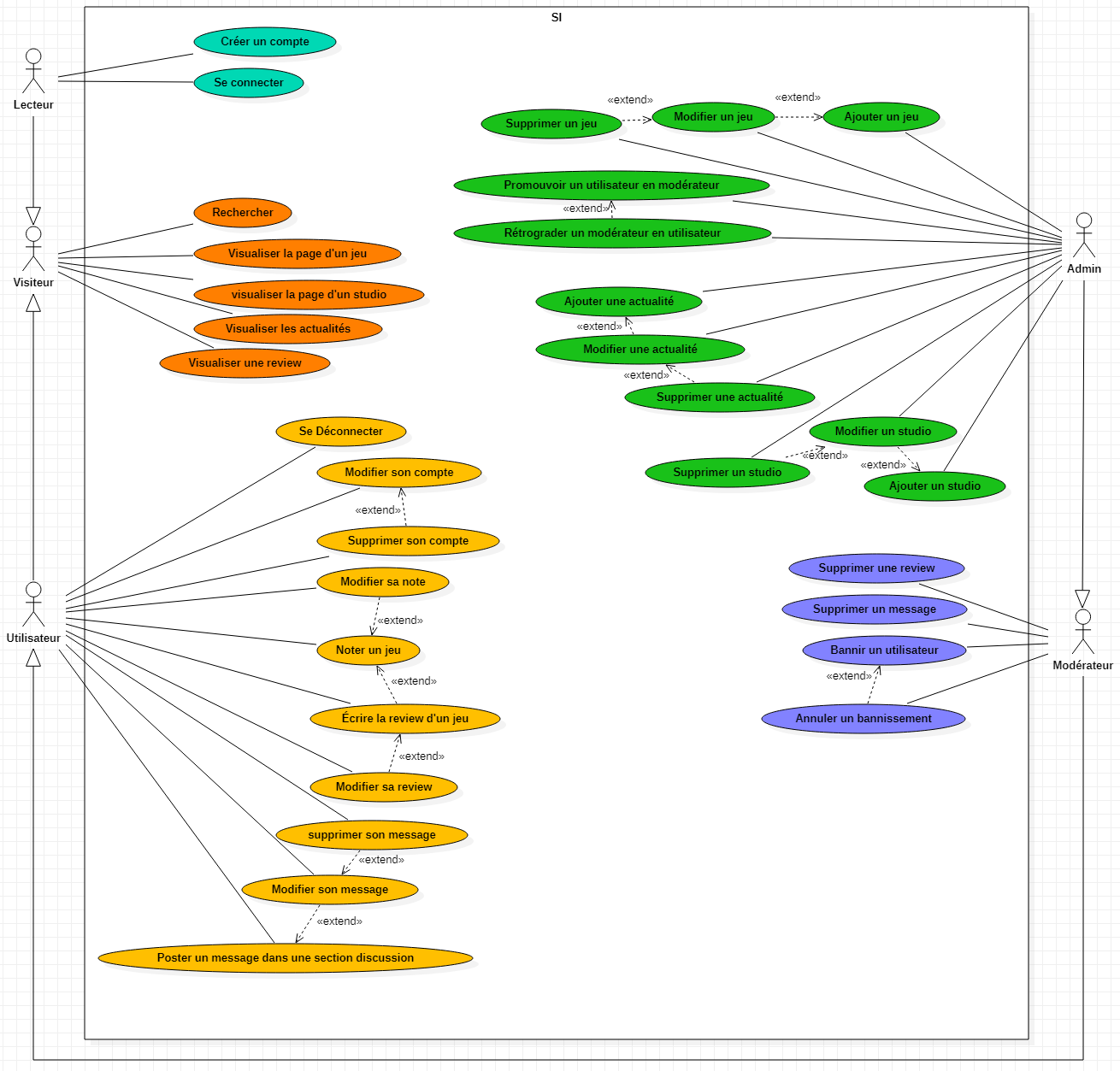
Français Anglais : Rotten tomatoes étant un site anglophone, les langues Anglais et Français serait nécessaire, mais vont nécessiter une traduction entière du site.

Charte graphique à respecter : il faudra respecter un minimum la charte graphique de rotten tomatoes pour créer un lien visuel entre les deux sites, ce qui limitera les choix créatifs possibles.

Langages choisis : le site étant développés avec les langages HTML, CSS, Javascript, PHP, il faudra s’adapter aux contraintes de ces langages pour développer les fonctionnalités du site.

## **Spécifications fonctionnelles**

Use case

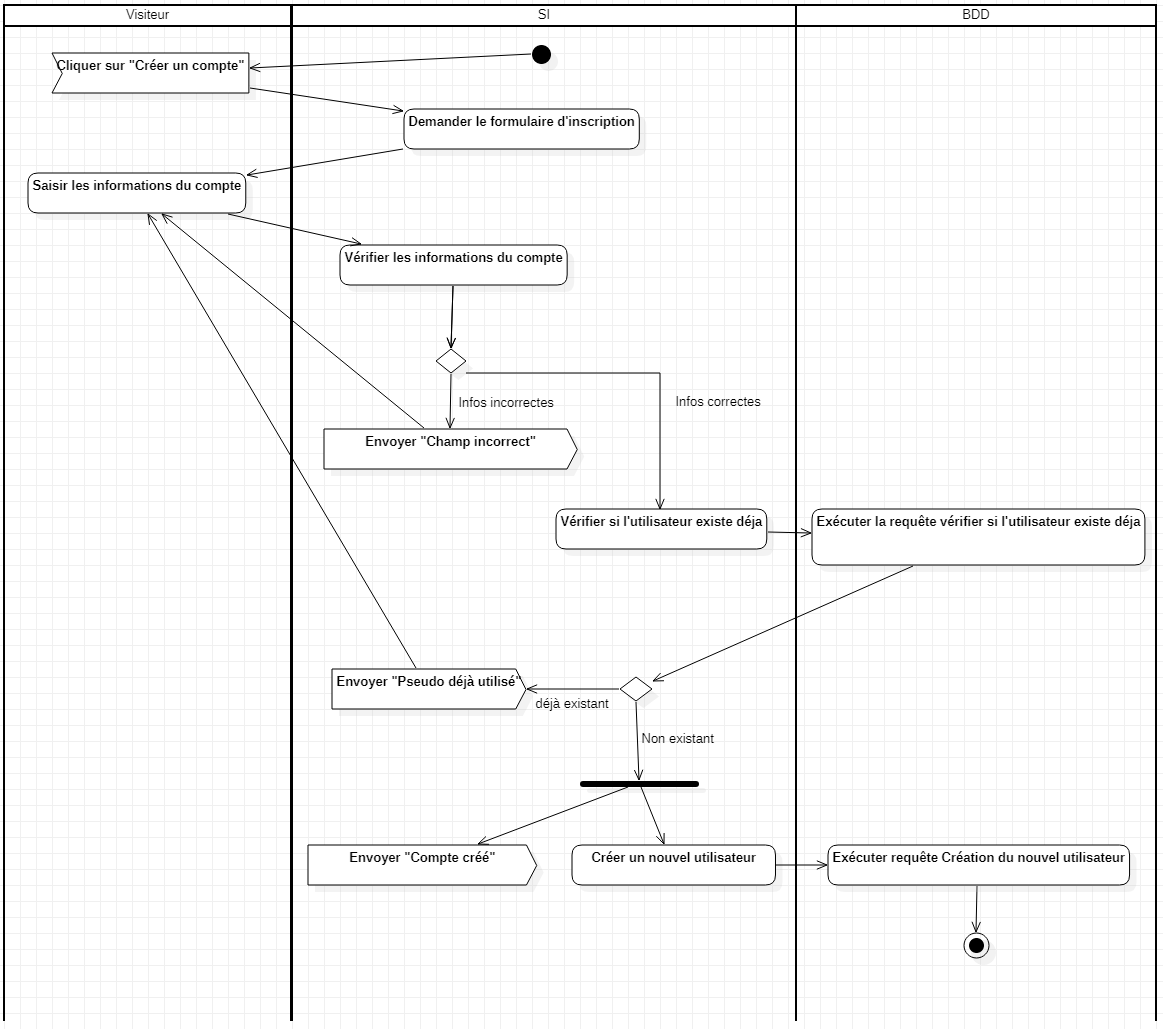


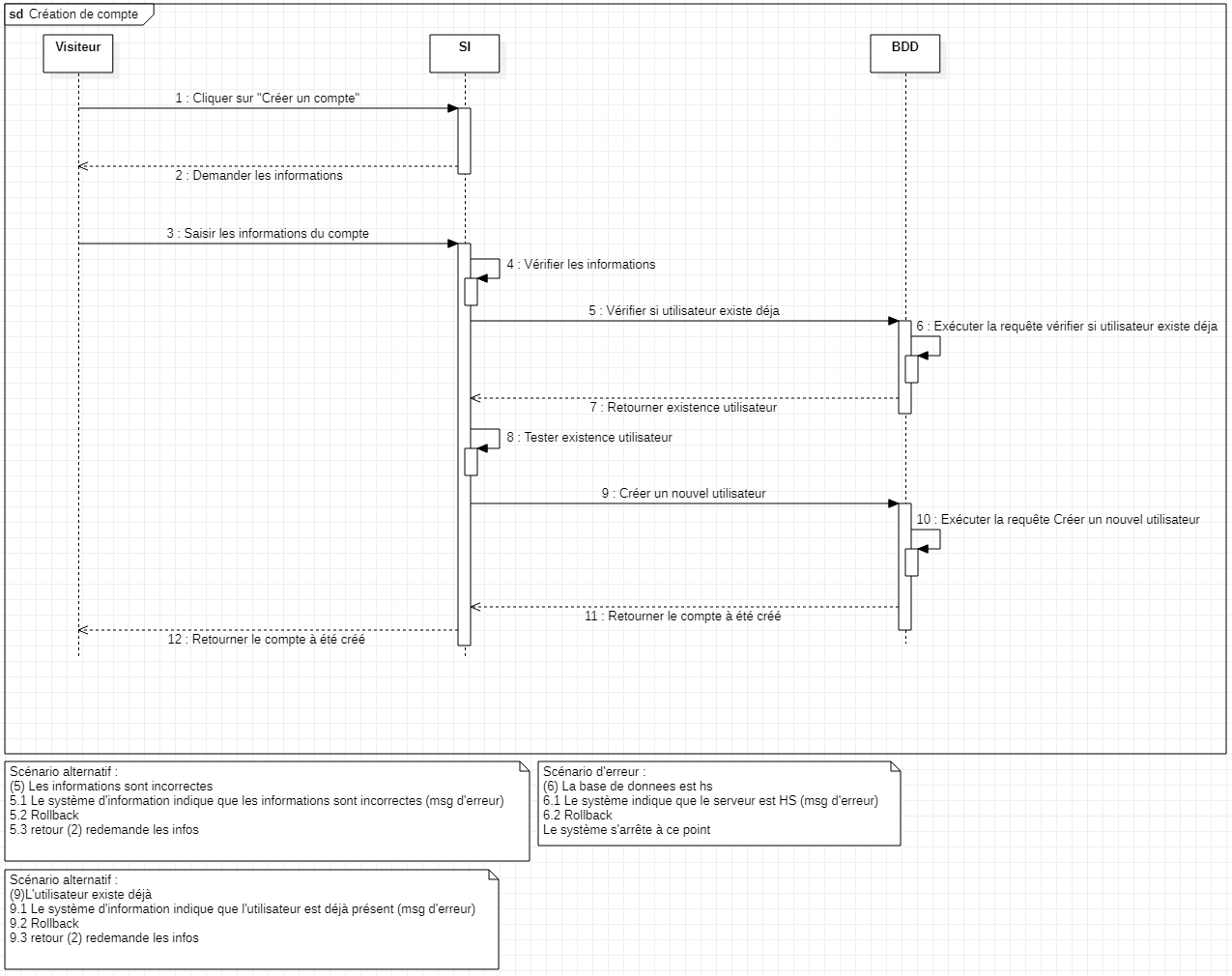
Les fonctions du diagramme des cas d’utilisations sont réparties sur 5 acteurs : Le visiteur, qui représente toute personne navigant sur le site sans compte, l’utilisateur, toute personne qui navigue sur le site en étant connectée à son compte, le lecteur, qui représente les fonctions de création de compte et de connexion auxquelles le visiteur a accès mais pas l’utilisateur, le modérateur, un utilisateur qui dispose des fonctions de modération de la communauté et l’administrateur qui est en charge de la gestion du contenu du site.

Dans la partie suivante seront présentés les diagrammes d’activité et de séquence des fonctions suivantes : création de compte, connexion, déconnexion, modification du compte et suppression du compte. Ces fonctions me permettront d’illustrer chaque partie du CRUD.

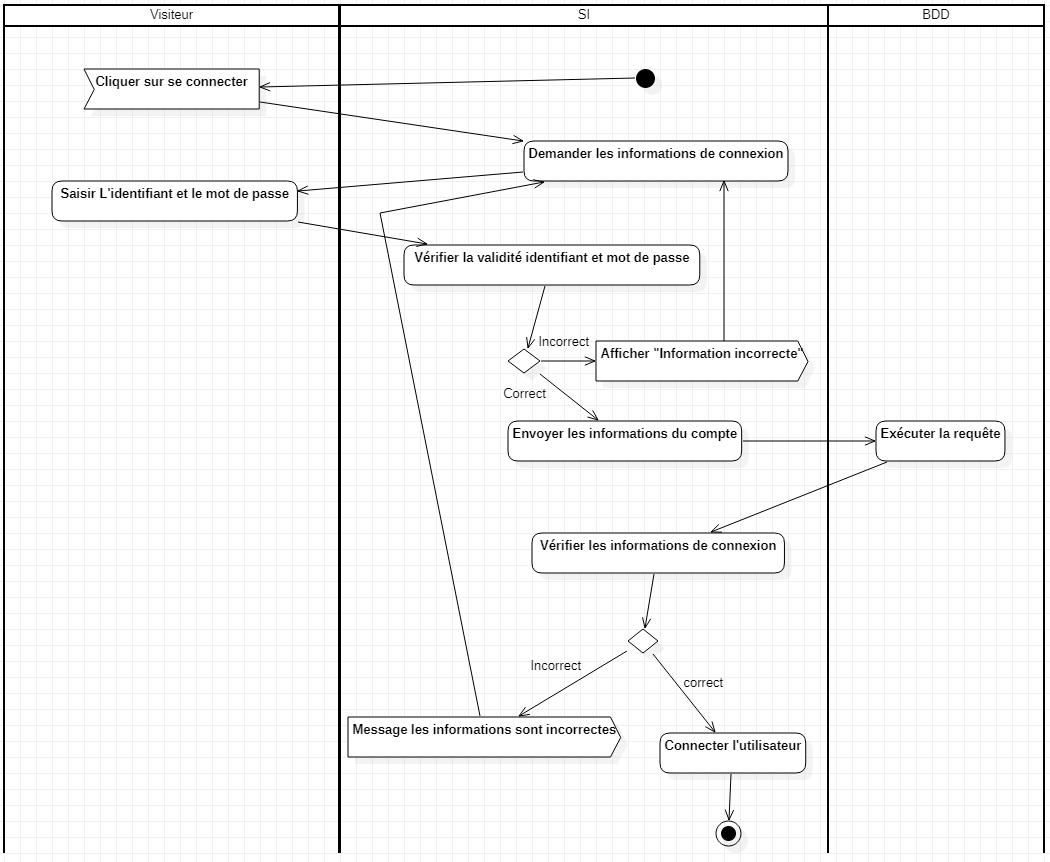
Diagrammes d’activité et de séquence

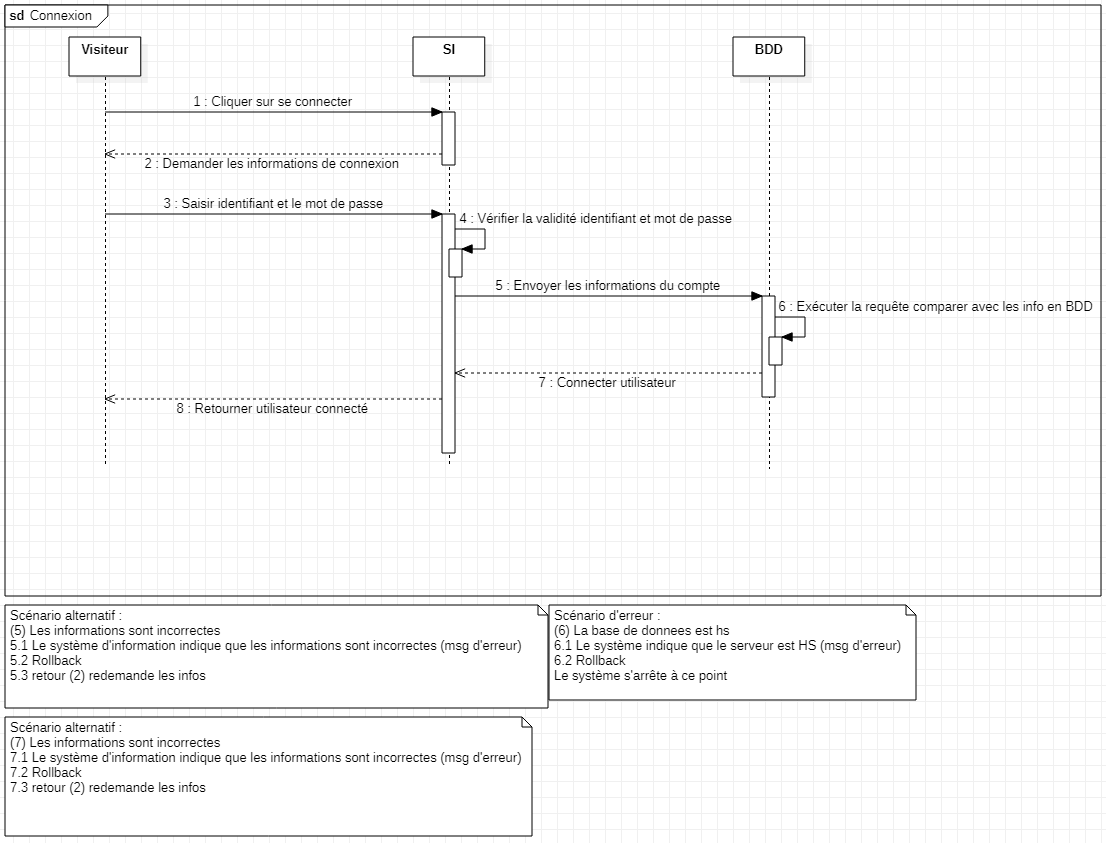
Création de compte



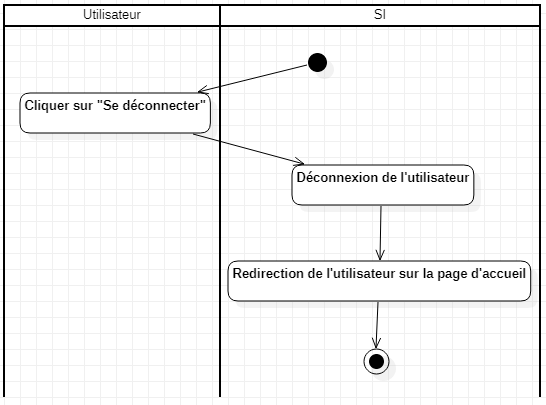


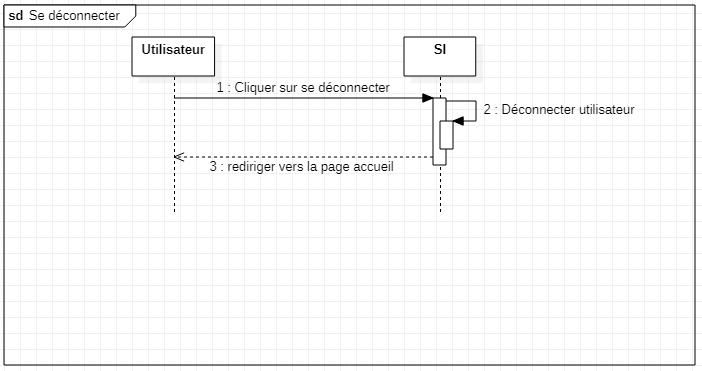
Connexion



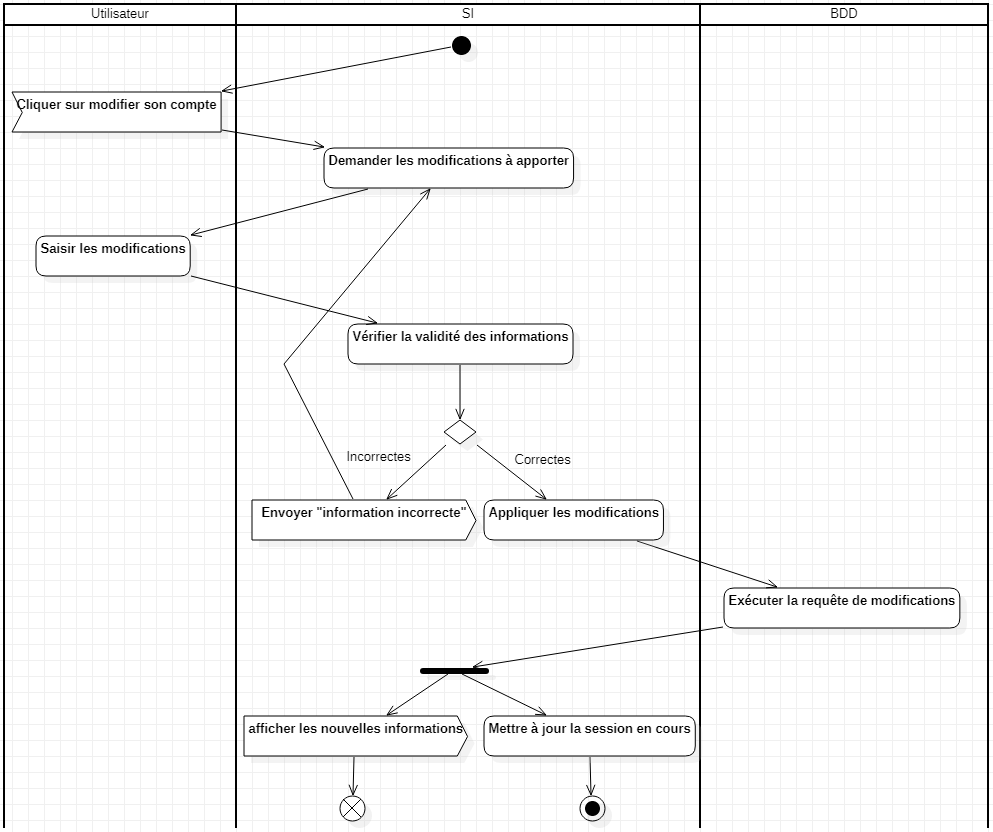


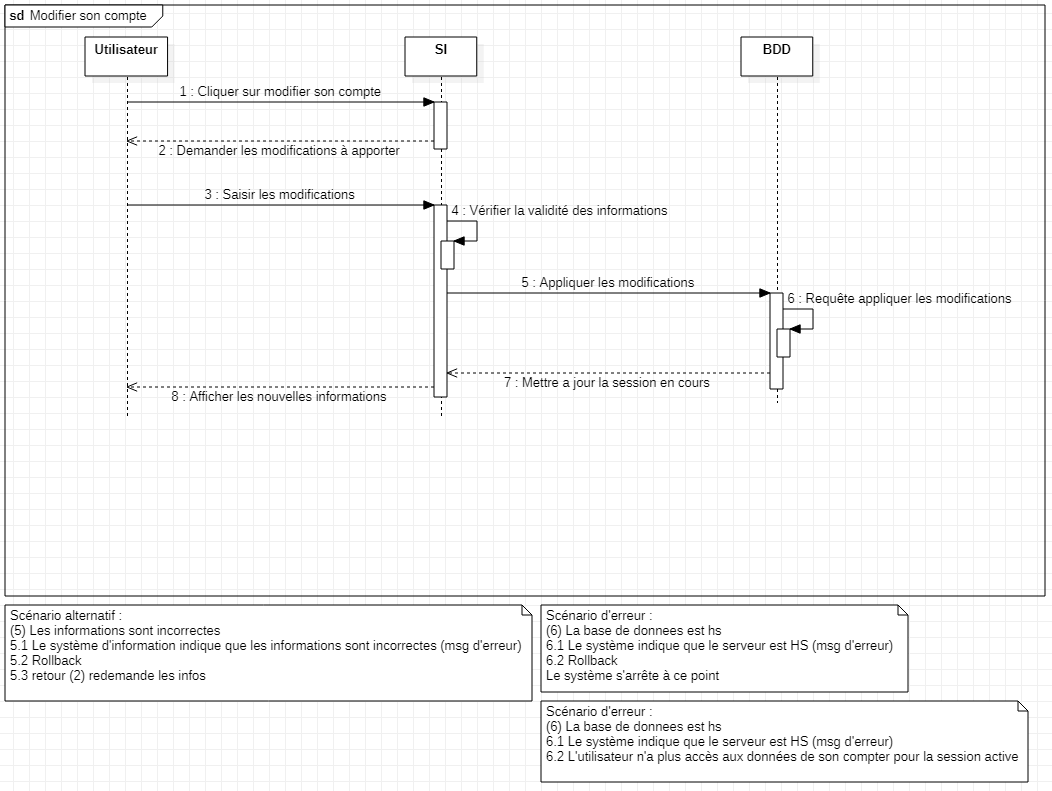
Déconnexion



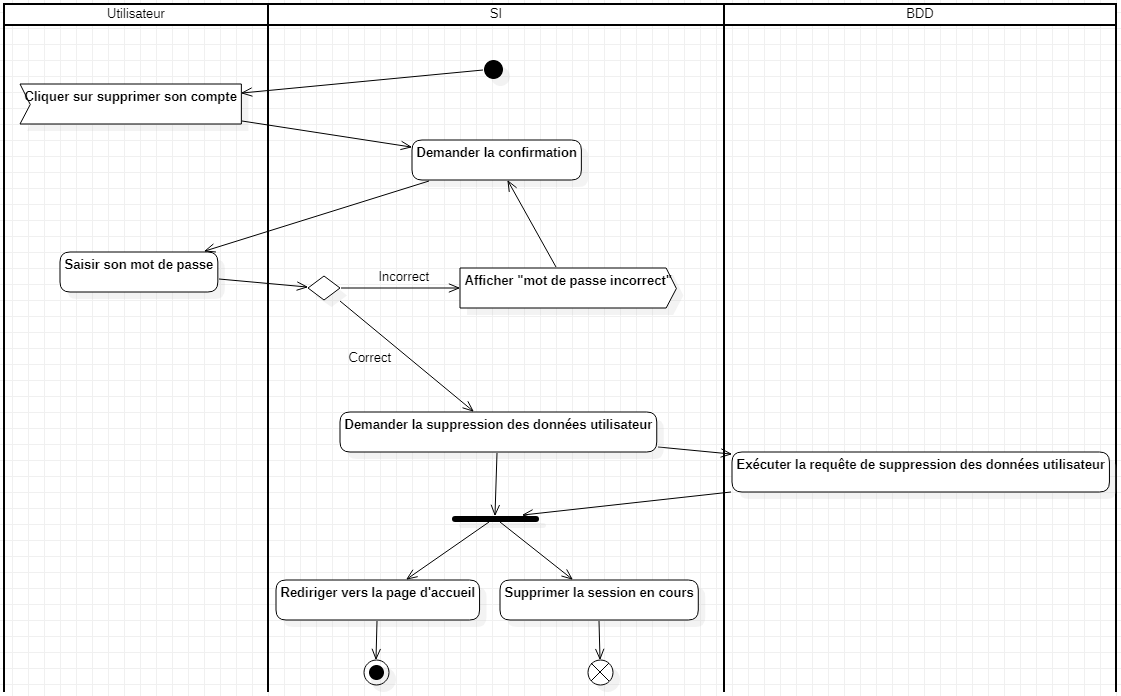


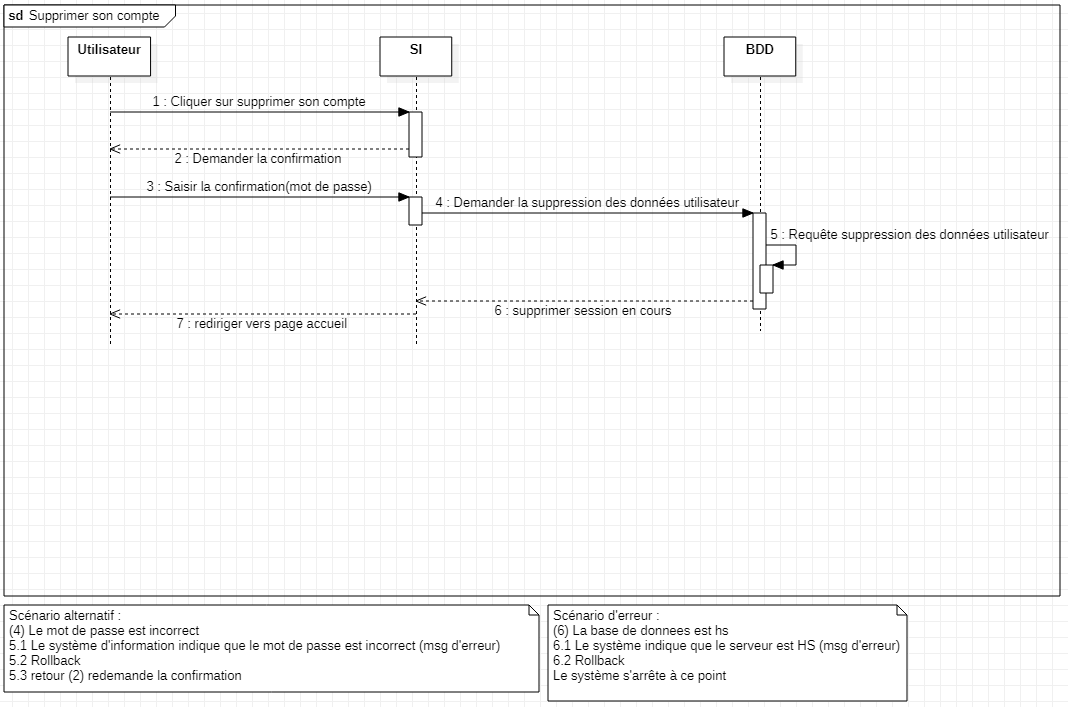
Modification





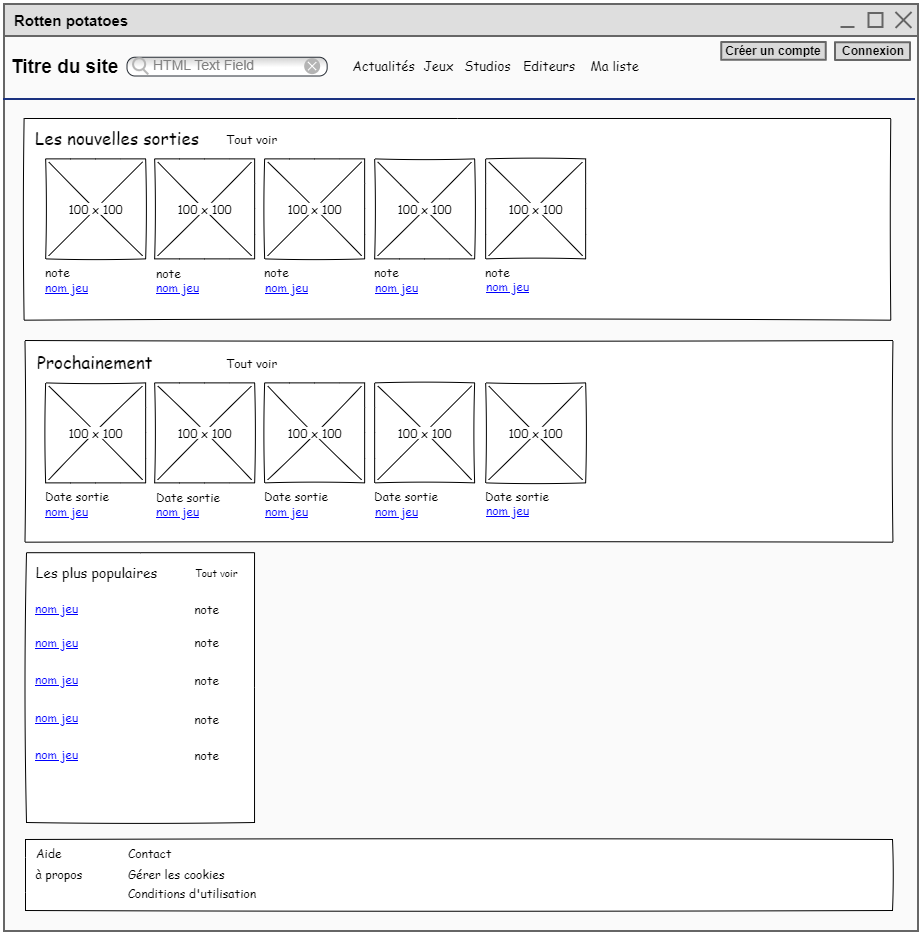
Suppression

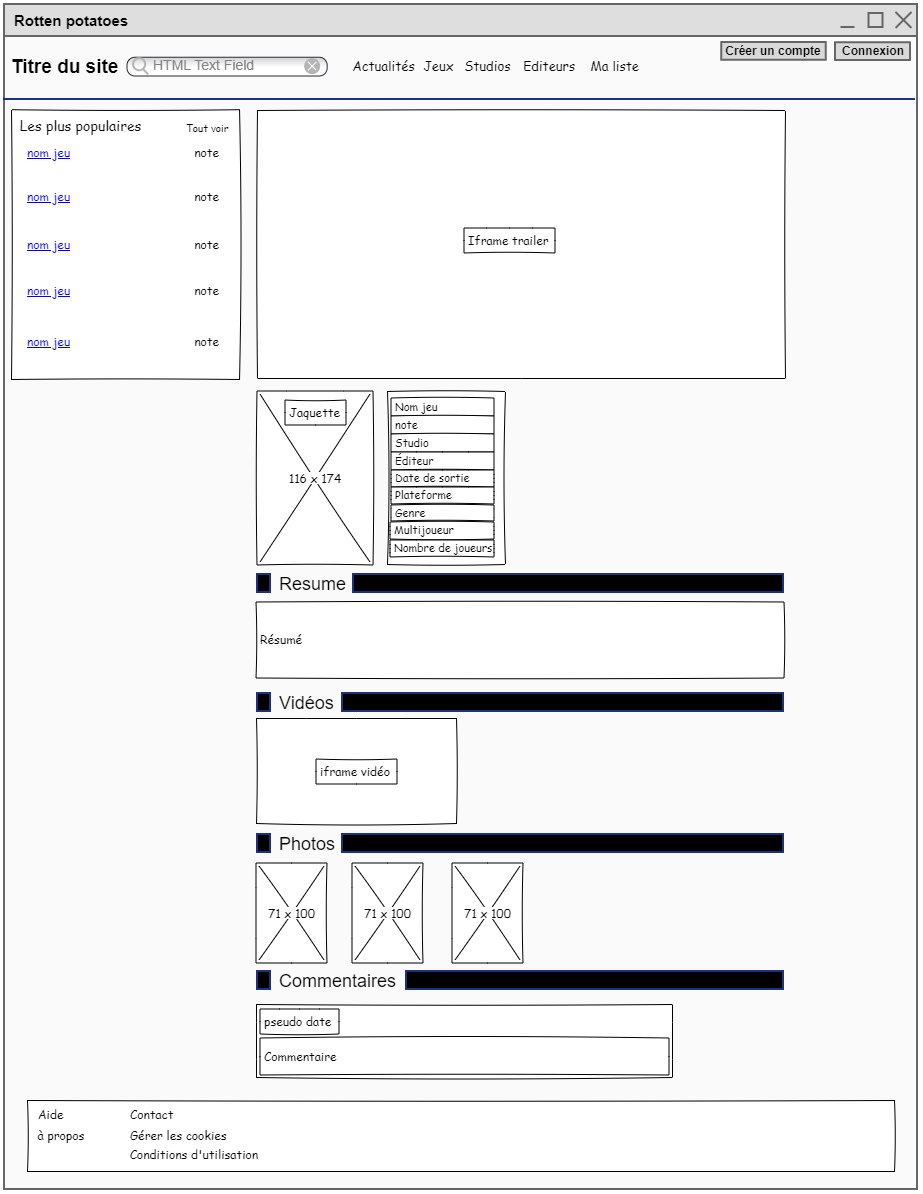




Maquettes

Par soucis de pertinence j’ai choisi de montrer les maquettes des pages accueil et jeu, qui sont pour moi les deux pages et plus importantes du site.

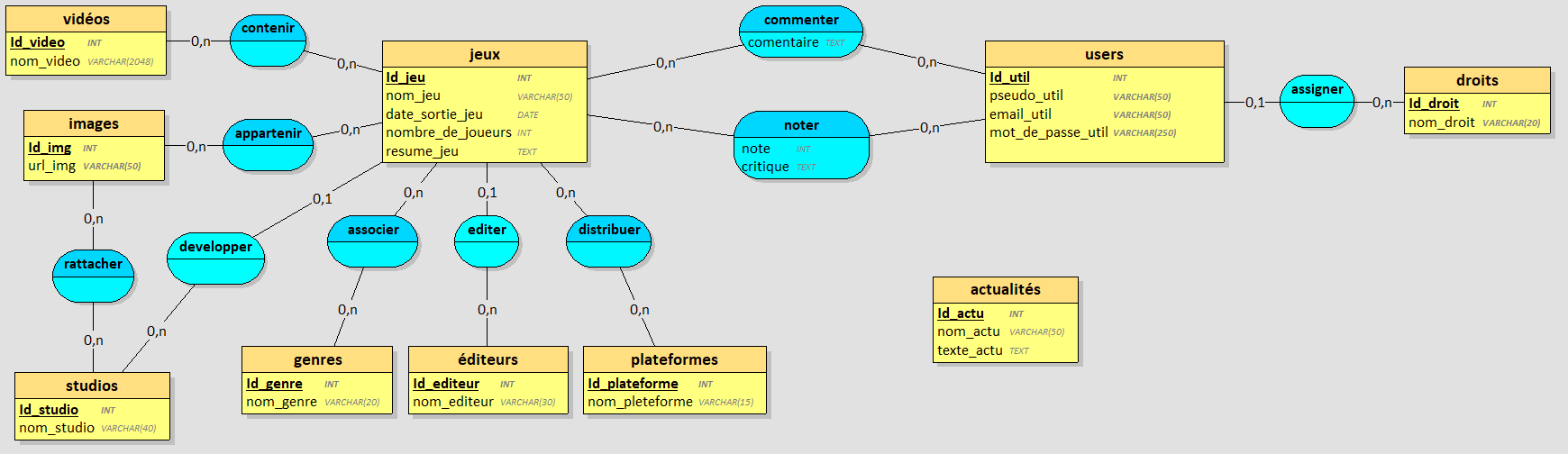




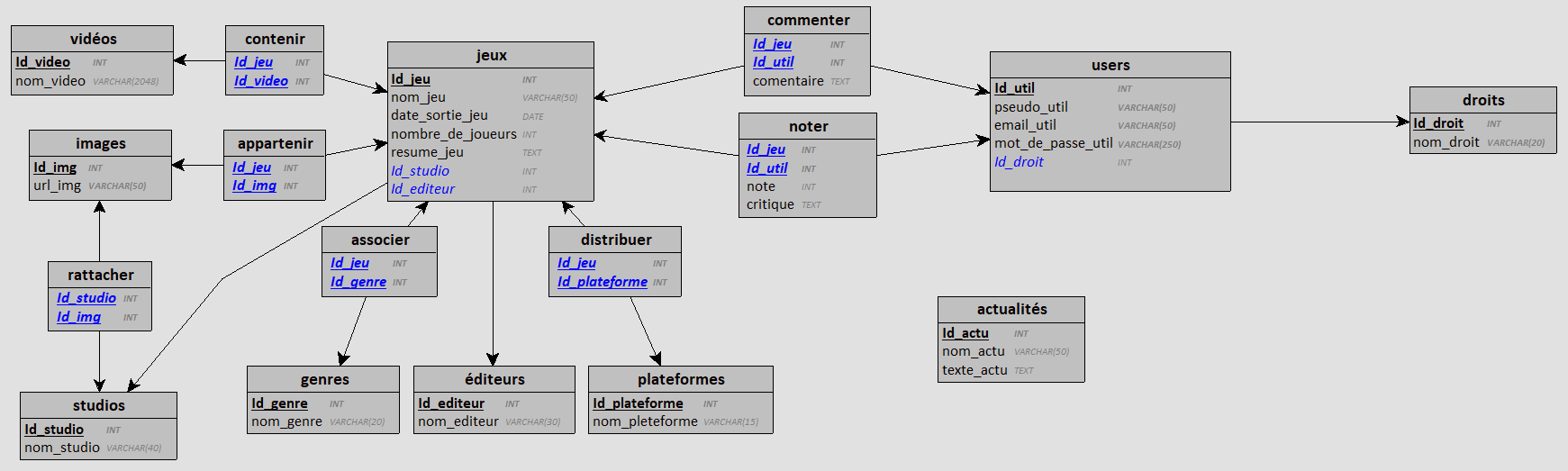
## **Conception**

Voici le modèle logique de la base de donnée de Rotten potatoes

MCD



MLD



Arborescence

Outils techniques utilisés

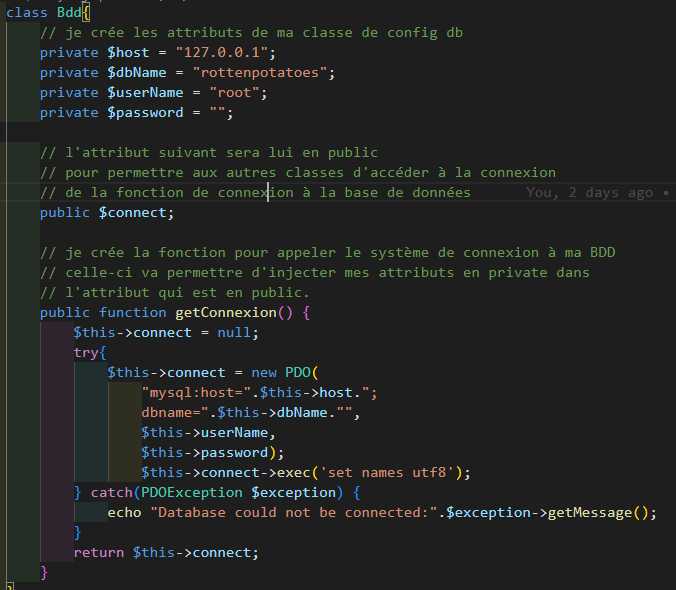
Pour mettre en œuvre ce projet, la partie front-end sera réalisée en HTML CSS, avec des éléments en Javascript pour gérer certains éléments dynamiques. La partie back-end sera quant à elle réalisé avec une structure suivant le modèle MVC en PHP. Le tout sera connecté à une base de donnée SQL accessible via des requêtes en MySQL et administrable grâce à phpMyAdmin.

Le code sera développé via Visual Studio Code, sauvegardé et versionné avec Git. Les requêtes de création de la base de donnée seront faites via MySQLWorkbench. Le site sera hébergé sur un serveur local créé via Wamp.

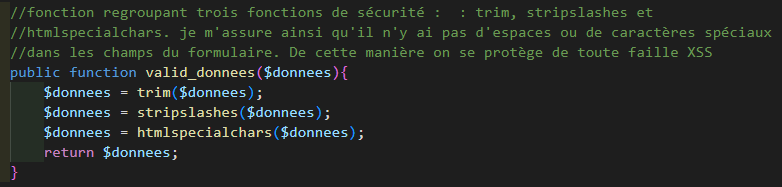
Fonctionnalités

Les fonctionnalités choisies correspondent aux fonctions présentées précédemment dans les diagrammes, mais il faut au préalable présenter deux fonctions utilitaires nécessaires au fonctionnement de reste du projet : la connexion à la base de donnée et la validation des données.

La connexion à la base de donnée est dans un fichier à part puis intégrée aux modèles pour éviter la répétition du code :

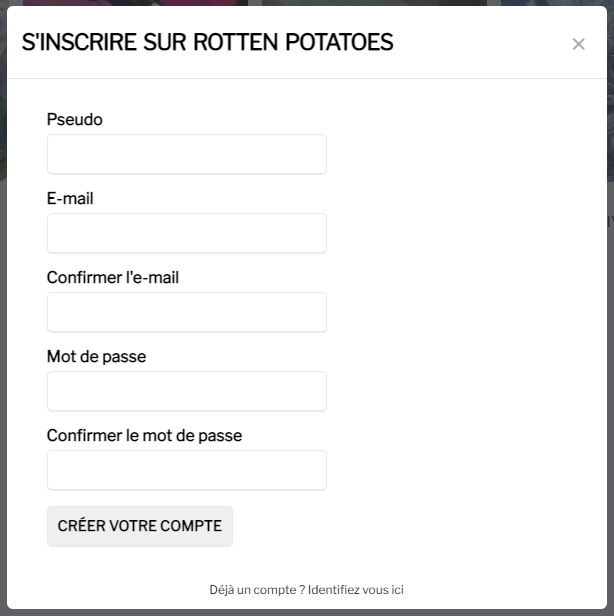


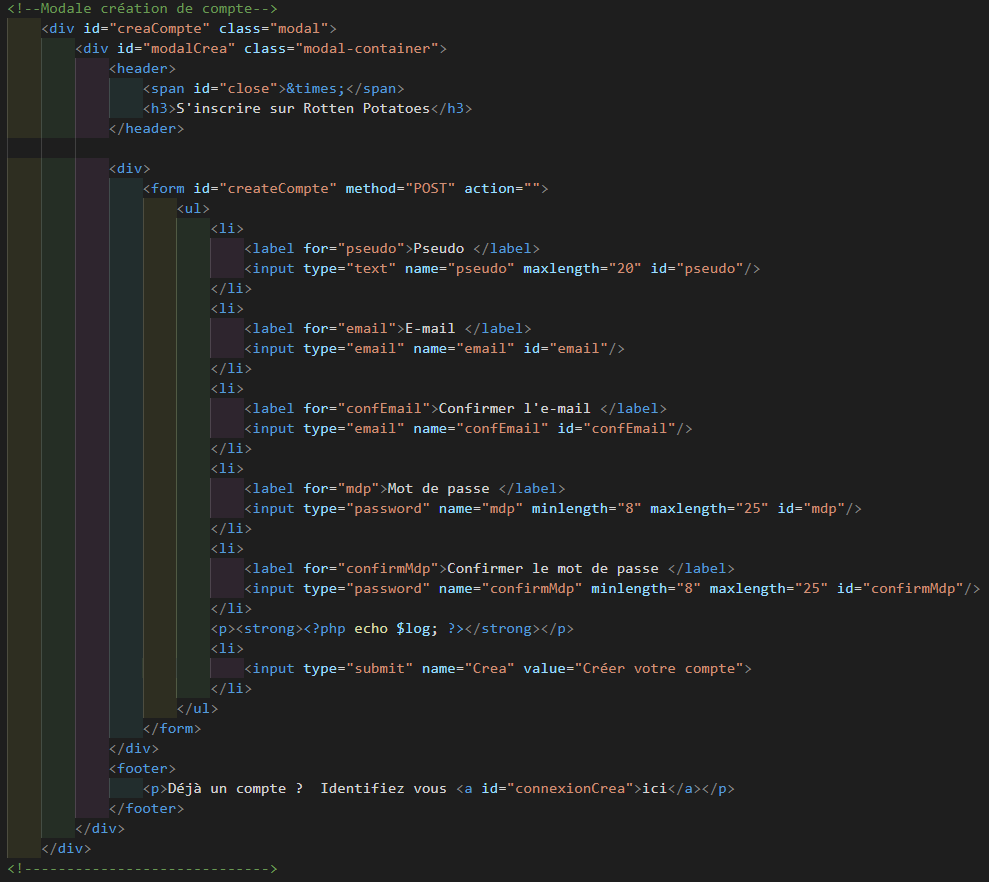
La fonction valid\_donnees quant à elle regroupe plusieurs fonctions de sécurité nécéssaires au traitement des données envoyé par les utilisateurs au site, on y retrouve trim, pour enlever les espaces, stripslashes pour enlever les slashs et htmlspecialchars pour convertir les caractères spéciaux comme les quotes ou les chevrons en entités html.



Création de compte

La création de compte passe tout d’abord par un formulaire disponible dans une modale depuis le header du site





Qui va envoyer les données au controller, qui une fois la présence des champs obligatoires vérifié, va les faire passer dans la fonction valid\_données pour les sécuriser.

Ensuite il va comparer le champ mot de passe avec sa confirmation et renvoyer une erreur s’ils sont différents, le même processus sera appliqué à l’email et sa confirmation avec en plus la fonction FILTER\_VALIDATE\_EMAIL qui vérifie que le format de l’email est correct.

Si ces étages de vérifications se passe correctement, le controller va ensuite utiliser les setters de la classe User pour affecter les valeurs des variables aux attributs de la classe, tout en utilisant la fonction password\_hash avec PASSWORD\_BCRYPT sur le mot de passe pour le hacher et ainsi éviter l’insertion en clair dans la base de donnée.

Une fois fait, le controller va pouvoir vérifier si le pseudo et le mail sont déjà présents en BDD, et renvoyer une erreur si c’est le cas. Pour cela les fonctions verifyMail et verifyPseudo seront utilisées en combinaison avec un rowCount pour vérifier la présence de champs dans le retour de la requête.

Pour finir le controller va exécuter la fonction createUser pour créer l’utilisateur en BDD grâce à des requêtes préparées. Si la création se passe comme prévu, il va vérifier qu’il y a bien un seul utilisateur avec le pseudo choisi dans la BDD avec la fonction getSingleUser de User et rowCount, et renvoyer une erreur si ce n’est pas le cas.

Une fois la création terminée, l’utilisateur est renvoyué sur la page d’accueil.