



Projet JEE

Équipe : JavaWeeb



Membres :

- Élisabeth Barré
- Hugo Bernat
- Mathieu Laumonier
- Philippe Martin
- Théo Mingorance

Date : 27 Mars 2020



Dans le cadre de ce projet, nous souhaitons fournir à un administrateur de site de jeux en ligne une solution qui permette à des joueurs de se connecter et à lui-même ou d'autres administrateurs de suivre l'activité des joueurs. La solution est donc une application web qui propose des fonctionnalités pour des joueurs comme l'inscription, le login ou le choix d'un jeu, mais aussi pour des administrateurs comme l'affichage de la liste des jeux, liste des joueurs ou des parties en cours. L'équipe de développeurs JavaWeeb a mené ce projet dans une optique fonctionnelle avant tout, mais s'est aussi attachée à rendre l'esthétique de l'interface suffisamment ludique pour être en adéquation avec l'univers des sites de jeux en ligne.

Diagramme de classes	3
Modèle de données	4
Explication des implémentations techniques des différentes fonctionnalités	5
Fonctionnalités pour les joueurs	5
Fonctionnalités pour les administrateurs	6
Conclusion	7



Diagramme de classes

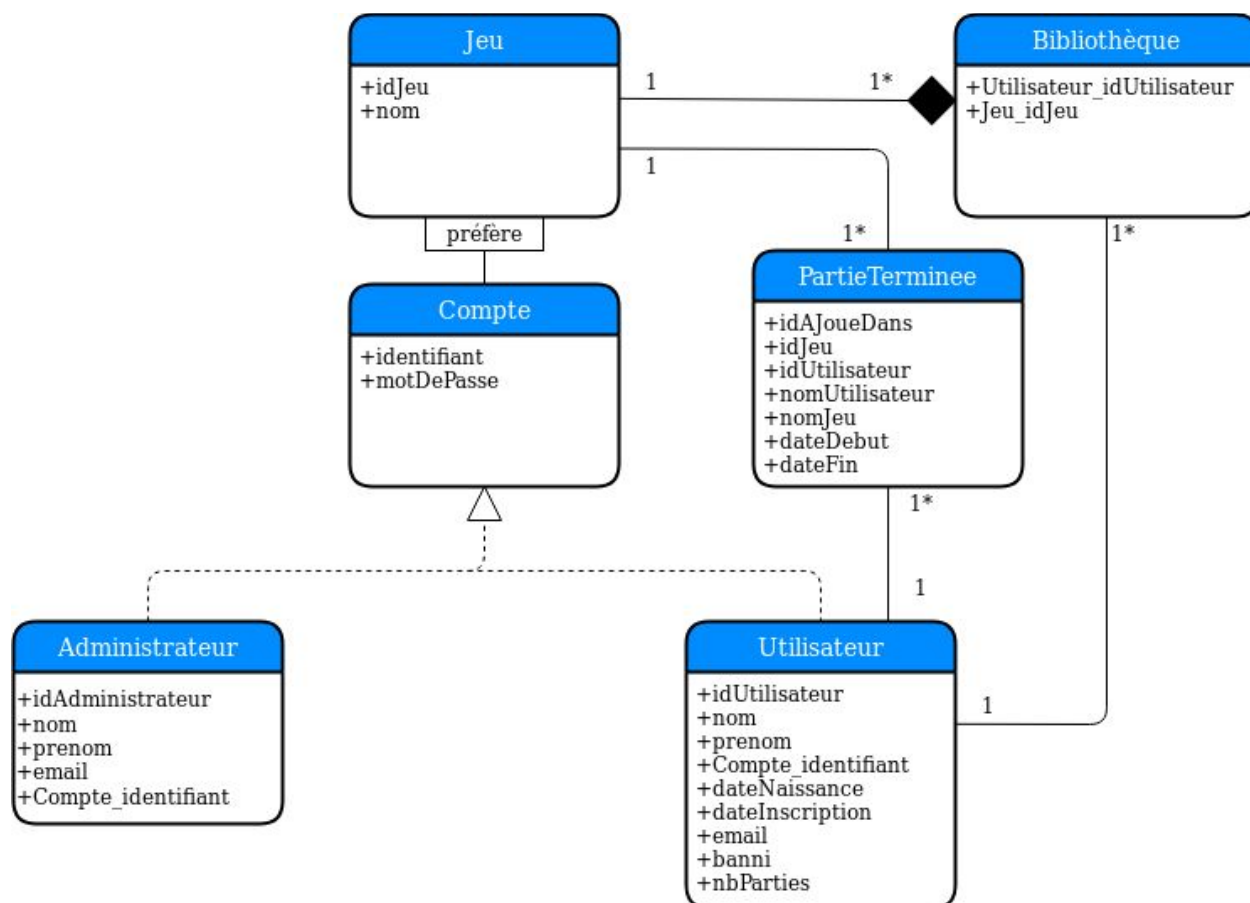


Fig1 : Diagramme de classes simplifié de l'application web

La figure Fig1 représente le diagramme de classes de notre application. Dans la classe **Compte**, les attributs sont "identifiant" et "motDePasse", qui servent indifféremment à la connexion de l'utilisateur ou de l'administrateur. Ensuite, on a posé la contrainte suivante pour un administrateur : il ne peut pas jouer avec ce compte-ci. Un utilisateur peut être banni, c'est représenté par l'attribut **banni**.

La classe **Bibliothèque** permet de stocker tous les jeux choisis comme favori par un utilisateur. La classe **PartieTerminee** nous permet de donner la liste des parties terminées à l'administrateur.

Les parties en cours ne sont pas stockées dans la base de données pour éviter des appels trop fréquents à celle-ci.



Modèle de données

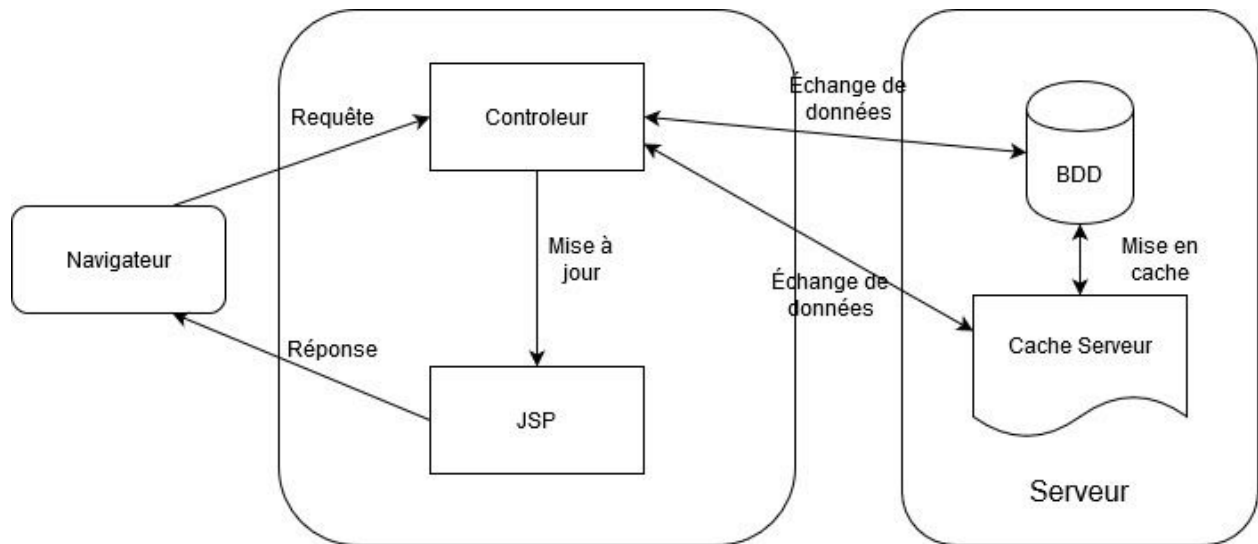


Fig2 : Modèle de données de l'application web

Le modèle de données est le suivant. Nous passons par des contrôleurs pour faire la gestion de l'affichage des pages internet (JSP). Les contrôleurs prélèvent les données soit directement de la base de données, soit du cache du serveur. Les données stockées dans le cache du serveur sont celles qui sont susceptibles d'être appelées le plus souvent ou qui peuvent subir le plus de modifications. Le choix de mettre ces données dans le cache est dans le but d'éviter de faire un nombre important de requêtes à la base de données. Les contrôleurs reçoivent aussi les requêtes des navigateurs et renvoient les pages JSP correspondantes. L'API mysql-connector est utilisée pour assurer depuis le DAO la connexion avec la base de données Mysql. De plus, l'API Servlet et son extension JSP sont utilisées pour mettre en place le protocole de communication HTTP entre le client (depuis les pages jsp) et les contrôleurs (servlets).



Explication des implémentations techniques des différentes fonctionnalités

Fonctionnalités pour les joueurs

1. Login / logout

Nous avons fait une page qui permet de se connecter. Elle est accessible depuis la première page du site, en cliquant sur le lien “Connexion” dans le header.

On peut se déconnecter par un lien dans le header au même emplacement.

Lors d’une connexion, un contrôleur récupère les données entrées par l’utilisateur et les compare avec celles de la base de données : il recherche dans la base de données l’identifiant, puis il fait la comparaison des hashes des mots de passe et enfin il regarde si le joueur est banni. Une fois le joueur connecté, il peut modifier les informations de son profil. Un contrôleur remplace alors simplement les données de la base de données par celles entrées par l’utilisateur.

2. Formulaire d’inscription

La page d’inscription est accessible à partir de la page de connexion. Il y a un formulaire pour récupérer les données dans les champs. Il y a seulement la vérification pour les champs spéciaux, par exemple, la date mais aussi pour vérifier l’égalité entre les deux mots de passe donnés. Une fois l’inscription faite, l’utilisateur est redirigé vers la page d’accueil et doit se connecter. Lorsqu’il est connecté, il peut accéder à une page d’informations sur son profil avec entre autres la liste de ses jeux favoris, sur laquelle il peut effectuer des suppressions.

La bibliothèque JSTL et le groupe de tags *core* ont été utilisés pour notamment envoyer dans le flux de sortie de la JSP le résultat de l’évaluation d’expressions fournies dans des paramètres ou encore pour implémenter des structures itératives ou conditionnelles.

3. Choix du jeu

La page des jeux est accessible via le lien dans le header “Jeux”.

C’est une page comportant plusieurs images de jeux dans un tableau. En cliquant sur l’un d’eux, on est redirigés vers une page “en jeu” et la partie est commencée. C’est sur la page des jeux que le joueur peut faire l’ajout de jeux à ses favoris. Un joueur peut jouer seulement si il est connecté, autrement une page pour se connecter s’ouvrira.

4. Début / fin du jeu

Au début d’une partie, on recherche le jeu dans la base de données, puis on crée un objet *PartieEnCours*, avec comme information supplémentaire la date précise du lancement du jeu et on stocke cet objet dans le cache du serveur.

Une fois que l’on ferme le jeu, on crée un nouvel objet *PartieTerminee* à partir de l’objet *PartieEnCours*, auquel on ajoute la date de fin de partie. On termine en envoyant l’objet *PartieTerminée* dans la base de données pour le sauvegarder.



Fonctionnalités pour les administrateurs

1. Login/logout

La connexion pour l'administrateur est exactement la même que pour un utilisateur lambda.

2. Liste des jeux disponibles sur le site

Une page qui contient la liste des jeux est disponible à partir du lien "Jeux" dans le header. Il y a une demande dans la base de données pour obtenir la liste des jeux. Il y a la possibilité de supprimer des jeux en cliquant sur le bouton correspondant. Les données sont alors supprimées de la base de données en commençant par la table Bibliothèque puis la table Jeu.

On peut aussi ajouter un jeu, le système le crée alors via une requête à la base de données.

On observe la modification de la page bibliothèque de jeux seulement pour l'administrateur et pas pour l'utilisateur.

3. Page sur les parties en cours

Cette page contient un tableau listant toutes les parties en cours, ayant pour colonnes : nom du jeu, pseudo du joueur, date/heure de début de partie. Un bouton « fin de jeu » sur chaque ligne permet à l'administrateur de terminer une partie. Afin de mettre à jour l'affichage pour refléter les nouvelles parties en cours, nous avons choisi l'option d'un bouton « rafraîchir » pour ne rafraîchir que le tableau sans recharger les autres éléments de la page. De plus, un rafraîchissement automatique du tableau s'effectue toutes les 15 secondes. Cela a été implémenté avec notamment l'architecture Ajax et la bibliothèque jQuery.

4. Page sur les parties terminées

On affiche la liste des parties terminées avec le nom du jeu, le pseudo du joueur, la date de début et de fin et la durée. L'affichage de cette liste n'est pas actualisé automatiquement, il faut rafraîchir la page pour que cela soit fait.

5. Page sur la liste des joueurs

Pour gérer la liste des joueurs, on affiche un tableau avec les attributs demandés. On a associé un bouton bannir à chaque ligne, lorsque l'on clique sur ce dernier, le bouton est remplacé par un bouton débannir. Une requête est alors effectuée à la base de données pour changer la valeur correspondante. On peut également cliquer sur le pseudo d'un joueur, ce qui entraîne une requête pour récupérer les informations du joueur et les afficher dans une page.



Conclusion

Ainsi, l'application satisfait plusieurs fonctionnalités attendues par un administrateur et un joueur. Dans le futur, nous pourrions mettre à jour la liste des jeux disponibles dans l'interface joueur à chaque création d'un nouveau jeu par l'administrateur. En outre, malgré la fonction de hashage qui sécurise le mot de passe des utilisateurs, l'application web mériterait à gagner en sécurité. Enfin, il faudrait empêcher un joueur d'accéder aux ressources de `games_user.jsp` (page de la liste des jeux disponibles) pendant qu'il est dans une partie en cours et l'obliger à terminer cette partie avant d'accéder à la liste des jeux disponibles. En effet, un utilisateur peut lancer plusieurs jeux différents s'il ouvre les jeux dans un nouvel onglet, cela peut engendrer des incohérences.