

Projet de programmation Web

Rapport technique

Équipe PDB :

Sébastien Hert

Alex Jobard

Thomas Lepercq

Dejan Paris

Adam Rivière

27 mars 2020

Table des matières

1	L'équipe technique	3
2	Spécifications techniques	3
2.1	Base de données	3
2.2	Tomcat	3
2.3	Java/JEE	3
3	Organisation de la plateforme	4
3.1	Utilisateurs	4
3.1.1	Joueurs	4
3.1.2	Administrateurs	4
3.2	Interface Web	5
3.2.1	Connexion	5
3.2.2	Inscription	5
3.2.3	Choix du jeu	5
3.2.4	Liste des jeux	6
3.2.5	Liste des joueurs	6
3.2.6	Détails du joueur	6
3.2.7	Parties en cours	6
3.2.8	Parties terminées	6
3.3	Application	7
3.3.1	Package <i>launch</i>	7
3.3.2	Package <i>user</i>	7
3.3.3	Package <i>servlet</i>	7
3.3.4	Package <i>tools</i>	7
3.4	Base de données	8
3.4.1	Table <i>User</i>	8
3.4.2	Table <i>Game</i>	9
3.4.3	Table <i>UsualGames</i>	9
3.4.4	Table <i>Session</i>	9
4	Annexe	9

Le but de ce projet est la mise au point d'une plateforme de jeux en ligne, permettant de centraliser l'accès à plusieurs jeux auxquels peuvent jouer les utilisateurs inscrits. Elle doit posséder un système de connexion, d'enregistrement des parties en cours et achevées, de gestion des jeux et des utilisateurs.

1 L'équipe technique

L'équipe qui s'est chargée de la mise en oeuvre de ce projet est composée des personnes suivantes :

- Dejan PARIS, coordinateur du projet.
- Thomas LEPERCQ, responsable du développement Front-End.
- Sébastien HERT, responsable de la partie application.
- Adam RIVIÈRE, responsable de la base de données.
- Alex JOBARD, chargé de relier l'interface web et la partie application.

2 Spécifications techniques

2.1 Base de données

La base de donnée est implémentée en MySQL et est hébergée en ligne sur le serveur de Tony Chouteau. La liaison est ensuite effectuée grâce à l'interface JDBC, et sa structure est détaillée dans la section *Base de Données*.

2.2 Tomcat

Nous avons choisi d'utiliser une implémentation déjà rédigée qui permet de simuler un serveur Tomcat sur nos machine. Cette version ne requiert que l'installation de Maven et permet de faire fonctionner l'application sur n'importe quel OS, sans machine virtuelle et surtout sans l'utilisation d'un environnement de développement spécifique.

2.3 Java/JEE

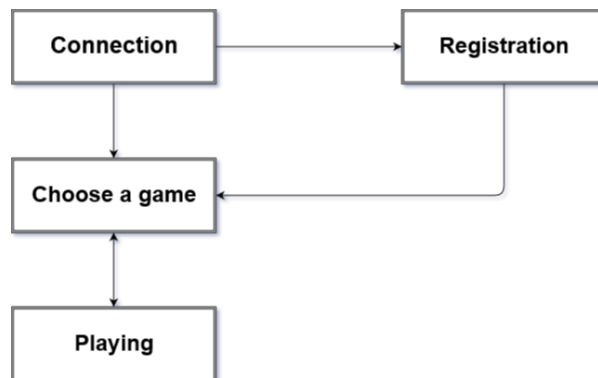
Ce projet a utilisé la technologie JEE, et plus particulièrement les deux composantes *JSP* et *Servlet*. La technologie JSP est une extension de la notion de Servlet permettant de simplifier la génération de pages web dynamiques. Un servlet est un composant coté serveur, écrit en Java facilitant l'exécution orientée "question / réponse".

3 Organisation de la plateforme

3.1 Utilisateurs

3.1.1 Joueurs

Les joueurs sont les utilisateurs classiques, aux droits limités sur la plateforme. Après connexion / création d'un nouveau compte, le joueur est renvoyé vers sa liste de jeux enregistrés. En sélectionnant un jeu, il commence immédiatement une partie.



3.1.2 Administrateurs

L'administrateur a le contrôle total sur la plateforme. Il a accès à cinq pages différentes, sur lesquelles il peut respectivement : ajouter / supprimer des jeux, terminer les parties en cours, accéder aux parties archivées, voir la liste des joueurs, consulter les informations d'un joueur et le bannir.



3.2 Interface Web

Les pages de l'application ont quelques fonctionnalités en commun :

- la possibilité de se déconnecter en haut à gauche des pages,
- une sécurité empêchant un utilisateur non autorisé (non connecté, banni ou n'ayant pas les privilèges requis) d'afficher la page
- pour les pages accessibles aux administrateurs, des boutons permettant de naviguer entre les pages.

3.2.1 Connexion

Elle correspond à la page d'accueil de l'application. Sur celle-ci, un utilisateur pourra s'identifier en entrant son login et son mot de passe. Un nouvel utilisateur aura également la possibilité de créer un nouveau compte en cliquant "Create a new account".

La vérification de la correspondance du mot de passe et du login est alors effectuée par le *HomeServlet* qui lui est associée via un appel à notre base de données en ligne.

En cas de succès de connexion, un utilisateur sera redirigé vers une page où il pourra choisir le jeu auquel il souhaite jouer, tandis qu'un administrateur verra apparaître une liste de jeux. Un joueur banni ne peut pas se connecter à la plateforme.

3.2.2 Inscription

La page *register.jsp* permet à un nouvel utilisateur de créer un compte, en renseignant :

- un pseudonyme (qui doit être unique),
- un mot de passe (à écrire deux fois pour confirmation),
- une adresse email (elle aussi propre à l'utilisateur),
- sa date de naissance (créer un compte nécessite d'avoir au moins 13 ans).

Si les données entrées sont valides, un nouveau compte "joueur" sera enregistré et l'utilisateur sera redirigé vers le choix du jeu, comme après une connexion. En cas de donnée erronée, un message d'erreur approprié s'affiche.

Cette page est associée au *RegisterServlet* qui gère la vérification des données saisies et l'enregistrement dans la base de données du nouvel utilisateur.

3.2.3 Choix du jeu

Après connexion ou inscription, le joueur est accueilli par le menu de choix de jeu *gamechoice.jsp*. Il se présente comme un tableau répertoriant les jeux disponibles, ainsi que le nombre de joueurs entrain de jouer à chaque jeu. En cliquant sur l'une des lignes, le joueur arrive directement sur la page du jeu correspondant : *playing.jsp*.

La recherche des données et la redirection sont gérées par *GameChoiceServlet*.

3.2.4 Liste des jeux

La liste des jeux est actualisée par un échange entre la page *gameslist.jsp* et le *servlet GamesListServlet*. Cette page n'est accessible que par connexion d'un administrateur, et elle fournit le détails des jeux disponibles sur le site et le nombre de sessions en cours pour chaque jeu. La page fournit aussi la possibilité d'ajouter ou supprimer un jeu selon la volonté de l'administrateur : l'ajout se fait en entrant le nouveau jeu dans la première ligne du tableau ; la suppression s'effectue en cochant les cases à droite des jeux à effacer.

3.2.5 Liste des joueurs

La liste des joueurs est actualisée par un échange entre la page *playerslist.jsp* et le *servlet PlayersListServlet*. De même que pour la liste des jeux, elle n'est accessible que par un administrateur. On peut y trouver la liste des joueurs avec leurs pseudonymes, la date de leurs inscriptions, ainsi que le nombre de sessions jouées. Une fonctionnalité non implémentée est la possibilité de bannir des joueurs à l'aide du bouton *Ban*.

3.2.6 Détails du joueur

Non fonctionnelle

La page *playerdetails.jsp* devrait afficher les informations d'un joueur sélectionné dans la liste des joueurs (pseudo, adresse mail, date d'inscription, date de naissance), mais l'implémentation de cette fonctionnalité n'est pas terminée. Cette page est associée à *PlayerServlet*.

3.2.7 Parties en cours

Dans *activesessions.jsp* s'affichent les sessions de jeu non terminées, sous forme d'un tableau contenant pour chaque partie : le nom du jeu, le pseudo du joueur, la date et l'heure de début. La page est gérée par *ActiveSessionsServlet* et actualisée automatiquement toutes les deux minutes. Un bouton *Refresh* permet l'actualisation manuelle.

La fermeture des parties en cours se fait à l'aide des cases à droite du tableau et du bouton *Close session*.

3.2.8 Parties terminées

La page *archivedsessions.jsp* permet simplement de consulter l'historique des parties terminées. Le tableau contient les noms des jeux, pseudos des joueurs, date et heure de début et fin ; ces données sont envoyées par *ArchivedSessionsServlet*.

3.3 Application

Dans cette section, nous allons décrire le code java utilisé en Back-End et qui permet le bon fonctionnement de l'application. Nous allons présenter les différentes parties du code en java regroupées en packages.

3.3.1 Package *launch*

Dans ce premier package se trouve le *Main* et le *Manager* de l'application.

Nous n'avons pas édité le fichier *Main*, en effet celui-ci permet le lancement complet de l'application qui lance son propre serveur Tomcat relié à l'extension MySQL nécessaire au bon fonctionnement de l'ensemble.

Le *Manager* quant à lui gère les interactions en stockant certaines données utiles comme l'utilisateur courant le jeu auquel il joue ou encore le joueur qu'un administrateur observe.

3.3.2 Package *user*

Dans ce package sont stockées les classes *User* et *Game*.

La classe *User* définit un utilisateur quelconque, qu'il soit administrateur, joueur ou banni. Nous avons décidé de ne pas faire de sous-classes de *User* qui pourraient contenir le "rôle" cité ci-dessus. La raison est simple, il faut dans tous les cas vérifier si l'utilisateur est bien administrateur.

La classe *Game* regroupe les informations concernant un jeu donné.

3.3.3 Package *servlet*

Dans ce package est stocké l'intégralité des *Servlets* qui permettent la jonction entre la partie web et l'implémentation Java et MySQL. Chaque *Servlet* est associé à une page *.jsp* et contient l'ensemble des méthodes propres à la page correspondante. Leur fonctionnement est décrit plus haut dans la section *Interface Web*.

3.3.4 Package *tools*

Dans ce package sont stockées les classes utiles comme les accès à la base de données ou le hashage des mots de passe.

La classe *Hasher* gère le chiffage des mots de passe des utilisateurs. Il vérifie également si un mot de passe est juste.

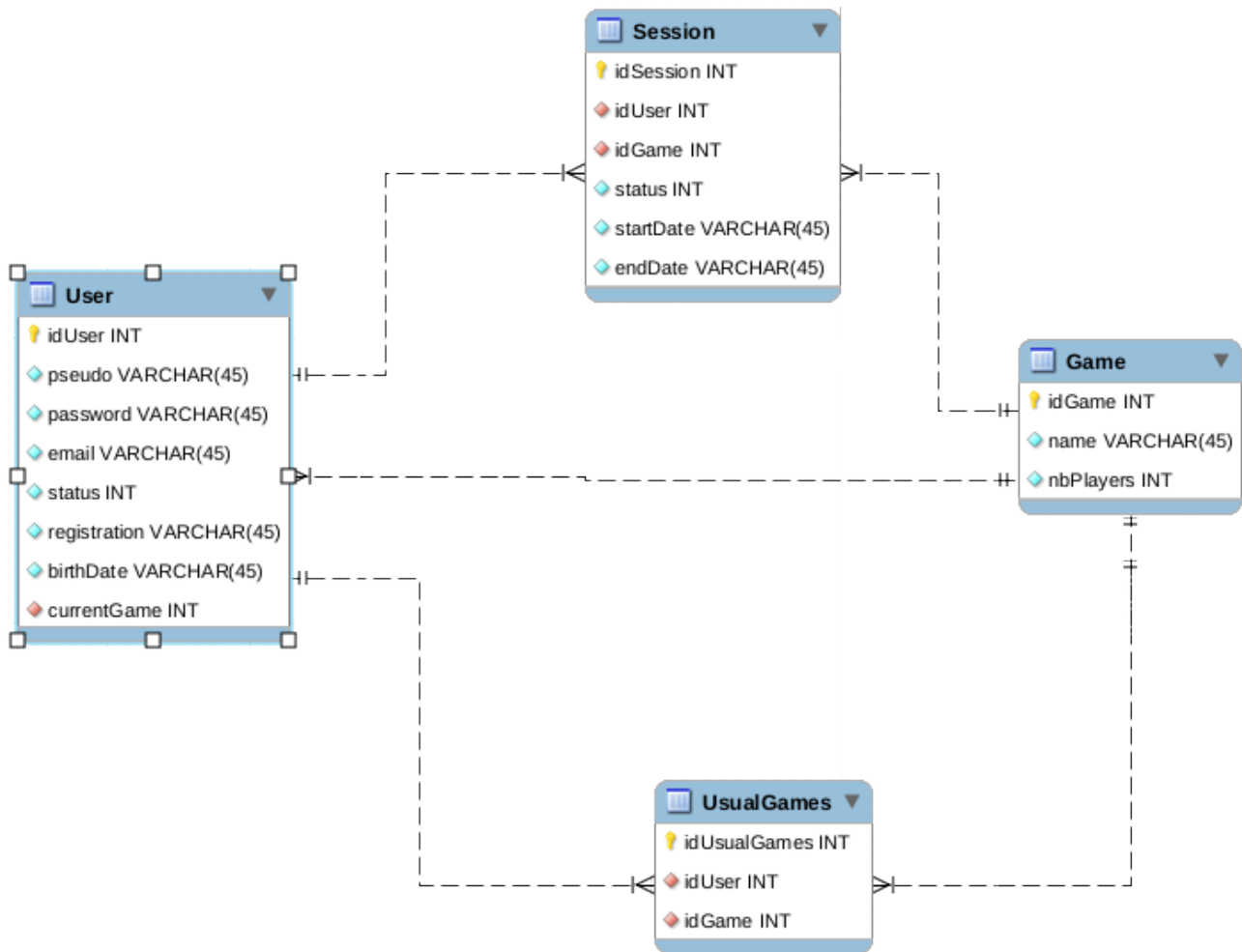
La classe *SDate* est une version *non-deprecated* de la classe *Date* de java, remise à jour. Elle gère correctement tout un ensemble de méthodes classiques attendues par l'utilisateur et non maintenues dans les versions actuelles de Java.

La classe *SQL* regroupe les méthodes qui accèdent à la base de données, que cela soit en lecture ou en écriture.

La classe *TestSQL* est juste une liste d'instructions utilisant les méthodes de la classe *SQL* permettant de remplir la base de donnée pour les tests.

3.4 Base de données

Les données des utilisateurs, jeux, et parties en cours sont stockées dans une base de données dont voici le schéma :



Plus précisément, la base de données est composée de quatre tables : *User* qui décrit un utilisateur, *Game* qui décrit un jeu, *UsualGames* qui relie un jeu à un utilisateur et *Session* qui décrit une partie d'un utilisateur sur un jeu.

3.4.1 Table *User*

La table *User* est composée de huit attributs. Premièrement, la clé primaire *idUser* est générée automatiquement à chaque création d'utilisateur. Viennent ensuite les attributs qui servent à décrire un utilisateur : son pseudo, son mot de passe hashé au préalable, son adresse mail, sa date de naissance, la date de son inscription et enfin son statut. Celui-ci a trois possibilités : 0, l'utilisateur est un administrateur, 1 il est joueur et 2 il est banni. Enfin, le dernier attribut permet de savoir à quel jeu l'utilisateur est en train de jouer à travers l'utilisation d'une clé étrangère. Il existe également un utilisateur par défaut *root* qui est utilisé dans certaines requêtes mais qui est inaccessible depuis l'application.

3.4.2 Table *Game*

La table *Game* est composée de trois attributs. Tout d'abord, comme pour la table *User*, la clé primaire *idGame* est générée automatiquement. Les deux autres attributs sont le nom du jeu et le nombre de joueurs en train d'y jouer. Comme pour la table *User*, il existe un jeu par défaut pour le cas où un utilisateur n'est pas en train de jouer ou n'a pas de jeu préféré. Dans ce cas, on affiche le jeu "None".

3.4.3 Table *UsualGames*

La table *UsualGames* est composée de trois attributs, étant tous trois des clés. Le premier est automatiquement généré, c'est la clé primaire, et les deux autres sont des clés étrangères permettant de lier des jeux à un utilisateur à travers ses jeux habituels.

3.4.4 Table *Session*

La table *Session* est composée de six attributs. Les trois premiers sont les mêmes que pour la table *UsualGames*, ils sont utilisés pour lier un joueur à un jeu. Les trois autres servent à décrire une partie du joueur sur le jeu : le statut du joueur, la date et l'heure de début et de fin de partie.

4 Annexe

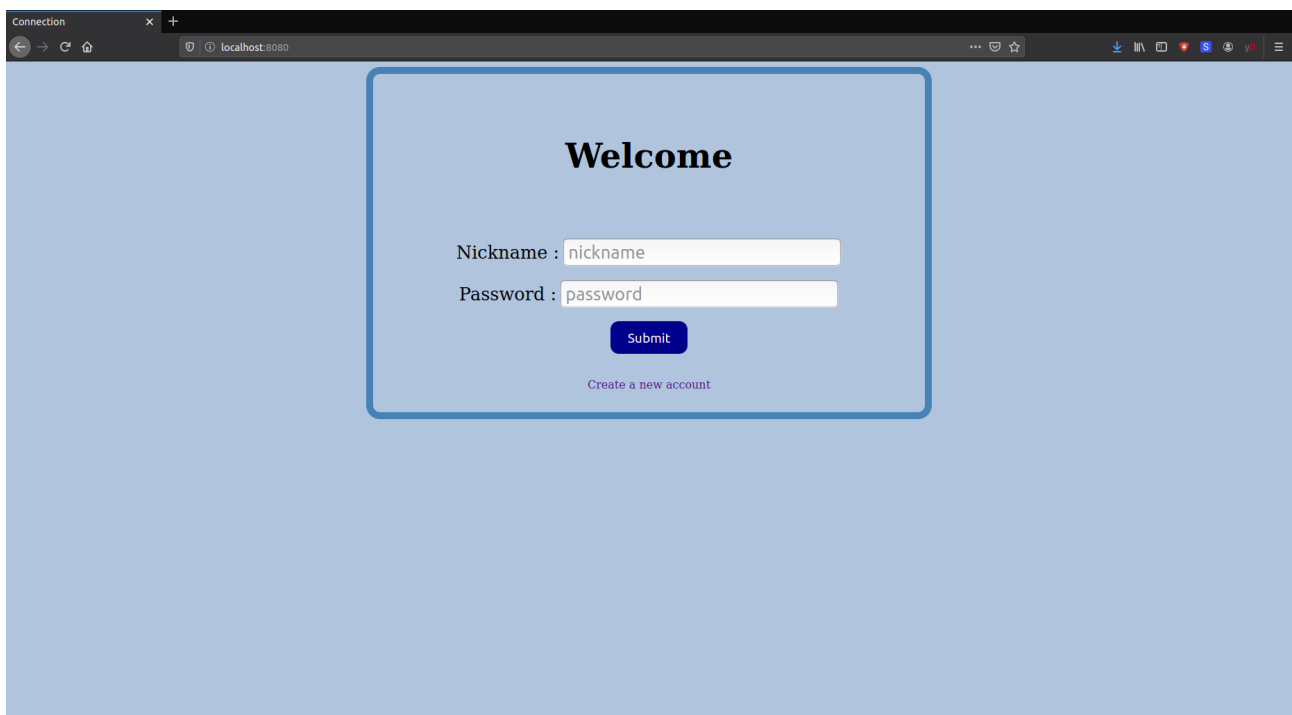


FIGURE 1 – Page d'accueil de l'application

Register

Back to login

Nickname :

Password :

Confirm password :

E-mail :

Birthday :

FIGURE 2 – Page de nouvel utilisateur

Choose a game

Game	Number of players
Fortnite	2
LOTRO	0
Minecraft	1
SWTOR	0
Oblivion	0
Total War Warhammer	1
Spellcaster University	0
Dragon Quest	0

Disconnect

FIGURE 3 – Liste des choix de jeux pour un joueur

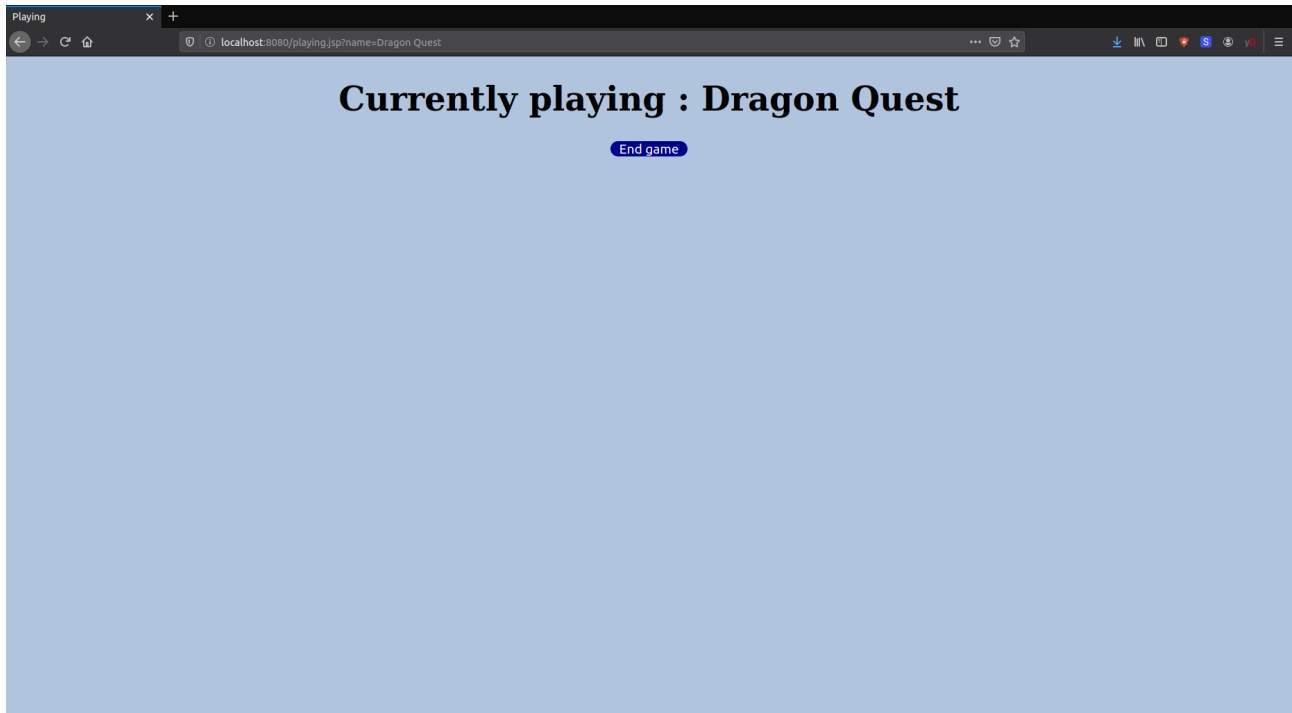


FIGURE 4 – Exemple d’une page de jeu

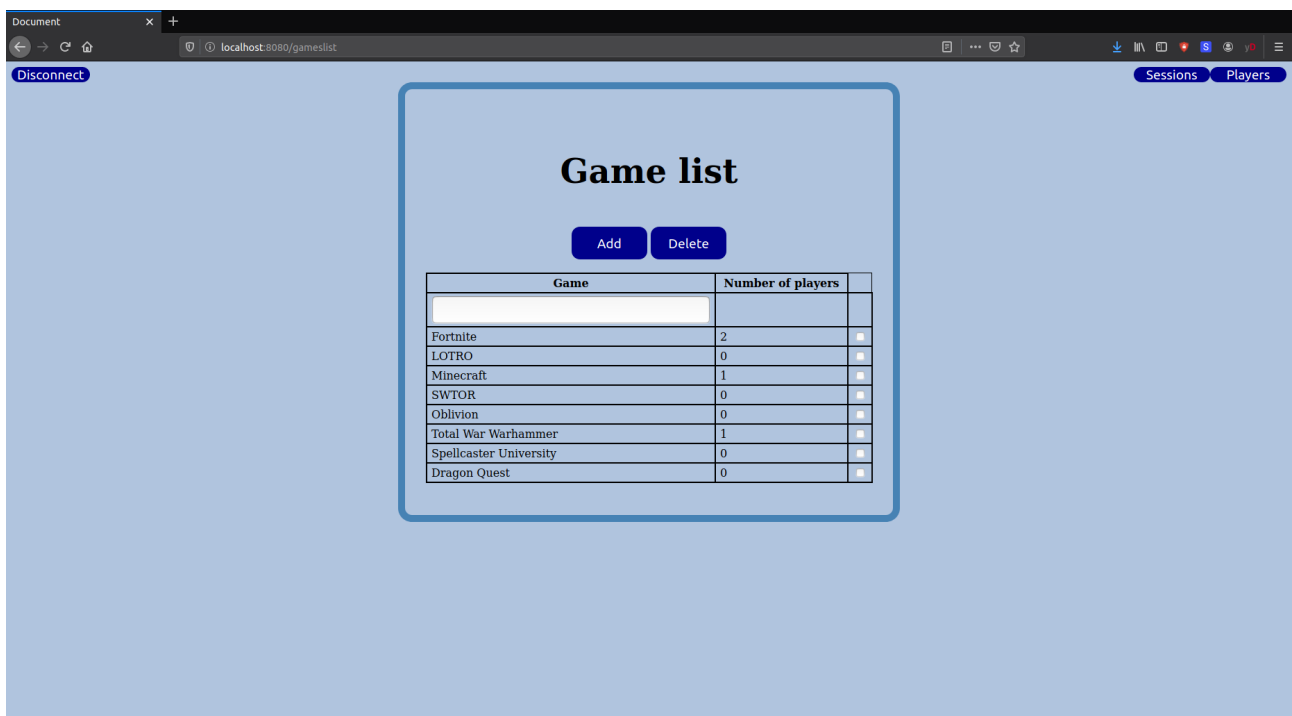


FIGURE 5 – Liste des choix de jeux pour un administrateur

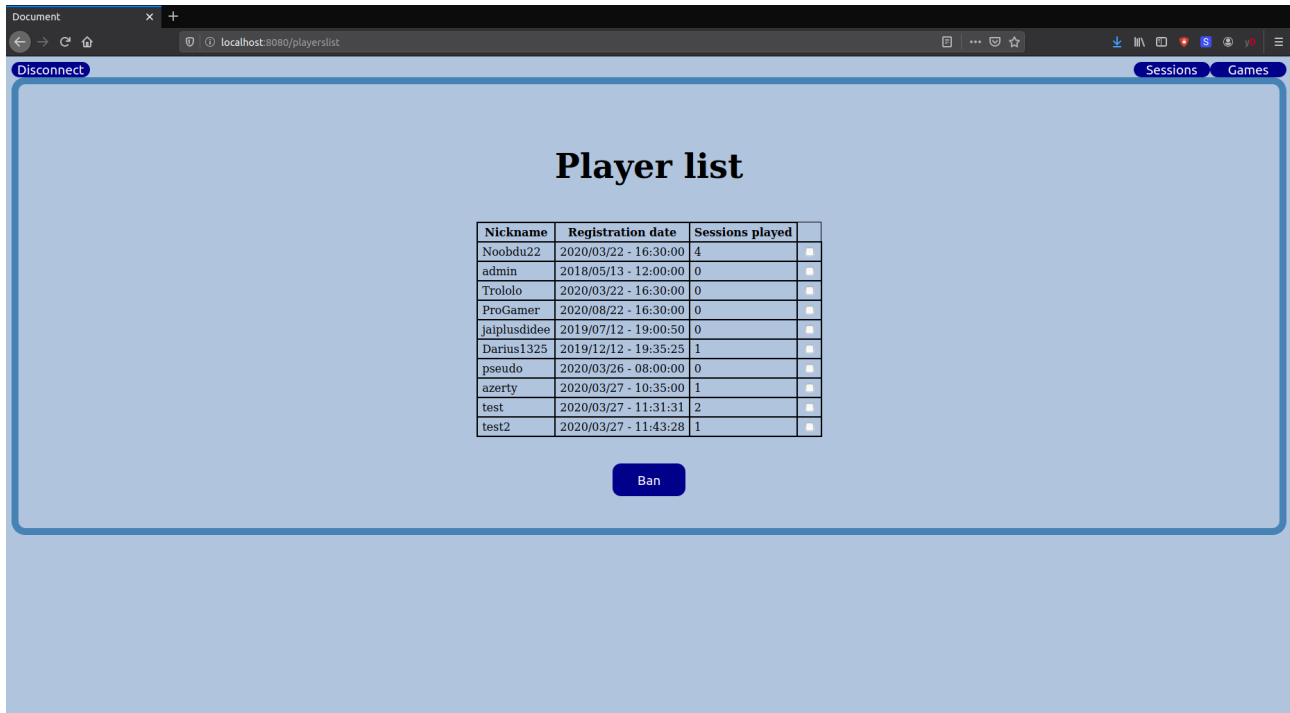


FIGURE 6 – Liste de tous les utilisateurs de la plateforme

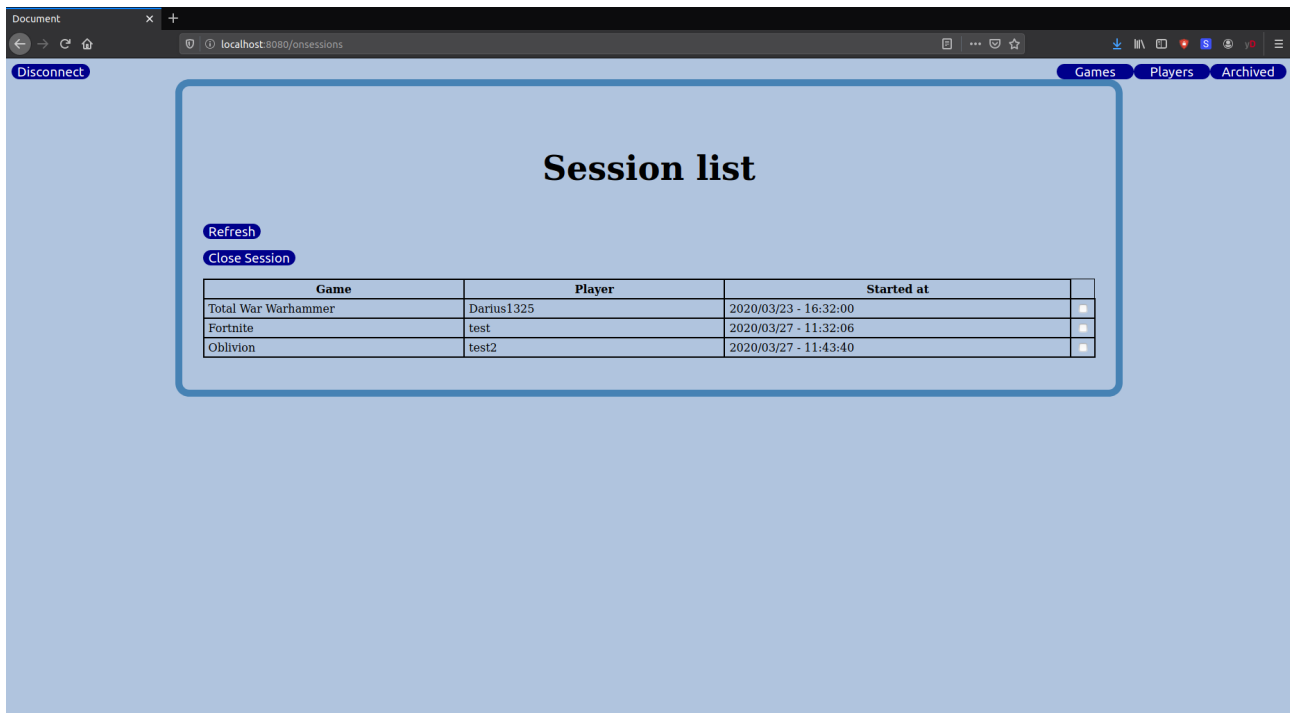


FIGURE 7 – Liste des sessions en cours

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/offsessions'. The page has a navigation bar with 'Disconnect', 'Sessions', 'Games', and 'Players'. The main content area is titled 'Finished sessions' and contains a table with the following data:

Game	Player	Started at	Ended at
Minecraft	Noobdu22	2020/03/24 - 02:00:50	2020/03/27 - 11:45:41
Total War Warhammer	azerty	2020/03/27 - 10:35:16	2020/01/01 - 00:00:00
Fortnite	Noobdu22	2020/03/27 - 11:21:40	2020/03/27 - 11:45:55
Oblivion	test	2020/03/27 - 11:31:44	2020/03/27 - 11:31:51
Minecraft	Noobdu22	2020/03/27 - 11:45:31	2020/03/27 - 11:45:41
Fortnite	Noobdu22	2020/03/27 - 11:45:45	2020/03/27 - 11:45:55

FIGURE 8 – Liste des sessions terminées