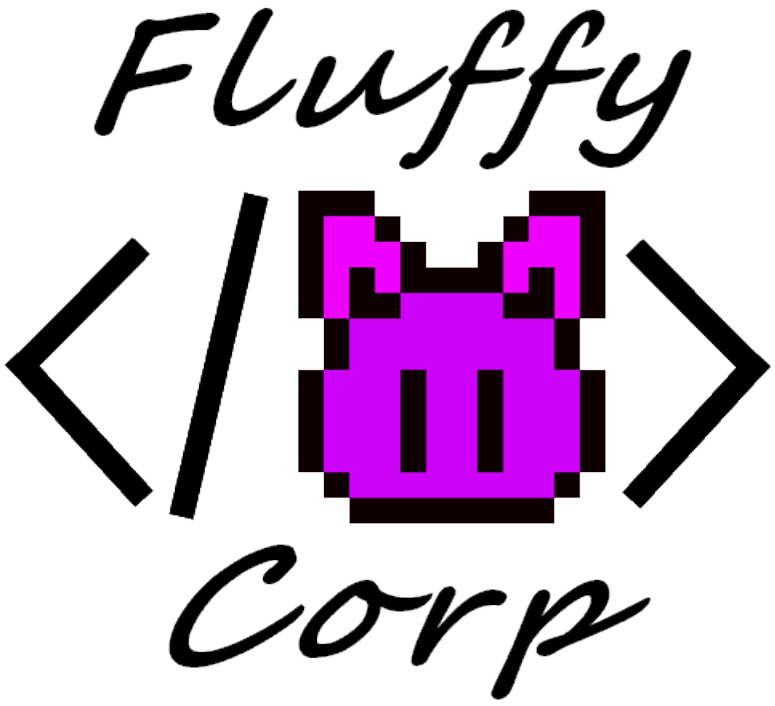
- COMPTE-RENDU -
Projet Programmation Web JEE

Tony Chouteau
Divi De Lacour
Thibault Gaudier
Marc Vacquant
Simon Lerin

27 mars 2020



Sommaire

1 Introduction

Projet de programmation web pour le 27 mars. Nous regroupons ici les différents livrables (hors application packagée en war) dans ce document :

- procédure pour installer l'application
- les sources de l'application
- rapport technique

  diagramme de classes

  modèle de données : structure des tables et relations entre tables

  explication des implémentations techniques des différentes fonctionnalités

1.1 Sources

Les sources complètes peuvent être téléchargées à l'adresse :
https://github.com/TonyChouteau/JEE-WebApp

2 Installation

On détaille ici la procédure d'installation de l'application sur un environnement vierge contenant un tomcat et mysql.

Pour installer l'application, il suffit théoriquement de copier l'archive (.war) fournie dans le répertoire "webapps" de tomcat et de démarrer le serveur, avant de se rendre à l'adresse localhost:8080. Toutefois il est important de noter que lors des tests de déploiement réalisés, cette méthode n'a pas été fonctionnelle. L'équipe a rencontré l'erreur suivante :

```
   ECHEC - L'application pour le chemin de contexte [/embedded Tomcat Sample]
n'a pas pu être démarrée
ECHEC - L'exception [org.apache.catalina.LifecycleException:
Echec de démarrage du composant
[StandardEngine[Catalina].StandardHost[localhost].StandardContext[/embeddedTomcatSample]
a été rencontrée.
```

3 Explication des implémentations techniques des différentes fonctionnalités

On détaille ici les explications des implémentations des différentes fonctionnalités :

Le code source en java utilisé par le serveur est constitué d'un Main s'occupant de la création et de l'initialisation du serveur et de deux packages :

- **data** : Fournit les outils permettant l'accès aux données

- **servlet** : Fournit les différents servlets utilisés par le serveur

Le serveur utilise aussi des fichiers `jsp` qui fournissent les templates des pages web que les servlets remplissent.
Un fichier `web.xml` associe les urls des pages web et les servlets.

3.1 Le package data

Ce package contient toutes les méthodes pour gérer les données internes au serveur. On a donc une représentation des valeurs le BDD sous forme d'objet JAVA dans les classes User, Jeu, Partie et PartieFinie. On possède deux bases de données :

- Une base de données extérieure `MySQL`
- une base de données interne au serveur Java `CurrentGame`

La liste des parties en cours est stockée à l'intérieur du serveur Java pour en accélérer l'accès, il faut cependant noter qu'un crash du serveur la supprimerait. Le reste des données qui sont la liste des utilisateurs, des jeux, des favoris et des parties terminées est stockée sur un serveur SQL exterieur pour être rendu plus "permanent".
La connexion à la base de données SQL se fait en utilisant `JDBC`. On crée une unique connexion au serveur pendant toute la durée de fonctionnement du serveur Java. Nous avons pour cela mis en place un design pattern Singleton dans la classe DB. Les requêtes sont faites en `preparedStatement` pour éviter les injections SQL. La classe `DB` fournit des méthodes Java permettant d'opérer sur la base de données.

3.2 Le package servlet

Les servlets s'occupent de gérer une tentative de connexion à une url particulière du site. Une servlet s'occupe de plusieurs urls du même thème.

- **Admin** : Fournit toutes les fonctionnalités qui ne sont accessibles qu'à un administrateur.
- **Connect** : Gère le profil utilisateur avec les connexions, déconnexions, inscription et édition du profil.
- **Highscores** : Fournit l'historique des parties jouées et les scores obtenus.
- **Play** : Permet au joueur de choisir le jeu auquel jouer.
- **Players** : Offre l'accès à la liste des joueurs.

Les servlets s'occupent de vérifier que l'utilisateur a le droit d'accéder à la page. Si c'est le cas elles génèrent la page web à afficher en complétant un fichier JSP déjà écrit. Elles utilisent des méthodes prédéfinies pour accéder à la base de données.

3.3 Sécurité

Nous avons durant ce projet essayé de tenir compte des questions de sécurité qui se posent lors de création d'un site web. Nous avons donc essayé d'appliquer certaines mesures de sécurité de base d'un site web : Les envois d'informations dans la base de données SQL se font avec des `PreparedStatement` pour éviter les injections SQL. Les mots de passes des utilisateurs sont hachés en `sha-256` sans sel.
Au sujet de l'identification de l'utilisateur, la session de l'utilisateur est maintenue en stockant son numéro d'identification dans son navigateur. Il est important de noter que ce numéro reste identique pour deux connexions d'un même utilisateur.
Un premier contrôle des entrées est fait en frontend avec du javascript pour faciliter l'utilisation mais il est toujours doublés d'un contrôle backend.
Il est aussi important de noter qu'il n'y a pas eu de système de "log" de mis en place.

4 Annexes

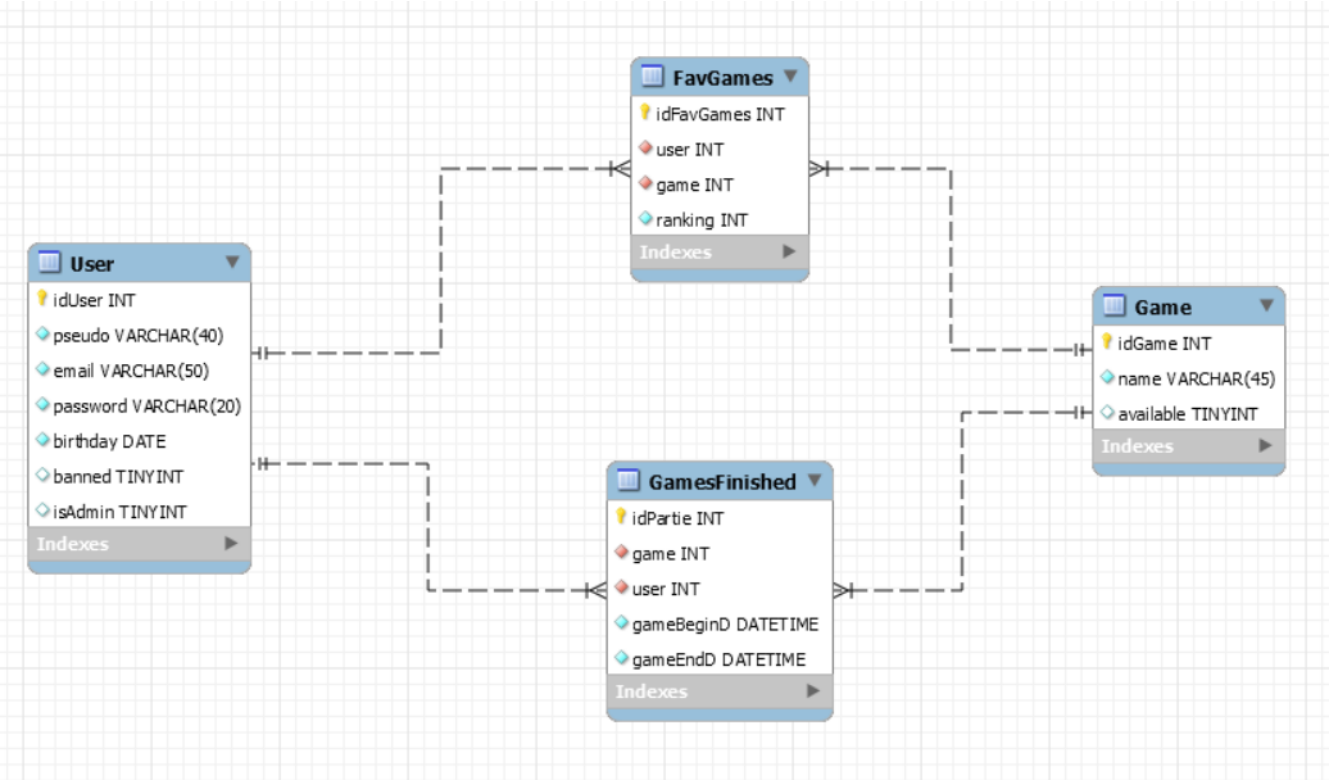


Figure 1: Modèle de données