

ENSIIE

RAPPORT DE PROJET

Projet web en groupe ENSIIE 1A 2018

Guangyue CHEN
Louis LAFUMA
Baptiste LAMBERT
Romain PEREIRA

Encadrants :
Thomas COMES
Nassim KIROUANE
Rémi PARPAILLON

22/05/2018

Projet WEB de groupe

Gangyue CHEN
Louis LAFUMA
Baptiste LAMBERT
Romain PEREIRA

22/05/2018

Table des matières

1	La problématique	2
2	Solution technique	3
2.1	Organisation du travail	3
2.1.1	Organisation général	3
2.1.2	Répartition des rôles	3
2.2	Back-end	3
2.2.1	Base de données	3
2.2.2	L'architecture du site	5
2.2.3	API 'REST'	6
2.3	Front-end	7
2.3.1	Le 'layout'	7
2.3.2	Discord	8
2.3.3	Notification	8
3	Conclusion	8
4	Références	9

Préambule

Ce projet est réalisé dans le cadre de nos études à l'ENSIIE. Les objectifs sont d'apprendre à concevoir et développer des applications web utilisant un serveur de bases de données, et prendre conscience des problématiques d'organisations d'équipes et de répartition des tâches.

1 La problématique

2 Solution technique

2.1 Organisation du travail

2.1.1 Organisation général

Un 'fork' du dépôt Github a été effectué, puis mis en privé. Guangyue travaille sous MacOSX, les autres sous Linux. Nous n'avons pas rencontré de problèmes de compatibilités, seulement de légères difficultés les 1ers jours pour configurer Docker. Nous avons développé sous PHP-Storm, Eclipse et Sublime-text selon les préférences.

2.1.2 Répartition des rôles

Romain PEREIRA et Guangyue CHEN se sont occupés du Back-end : Romain a construit l'architecture MVC du site (et l'API), et Guangyue s'est spécialisé sur l'intégration du jeu PUBG.

Louis LAFUMA et Baptiste LAMBERT ont développé certaines parties visuelles du site.

2.2 Back-end

2.2.1 Base de données

La base de données relationnelle peut être représentée par le diagramme UML suivant. (La transcription SQL du modèle est disponible dans le fichier 'data/db.sql') :

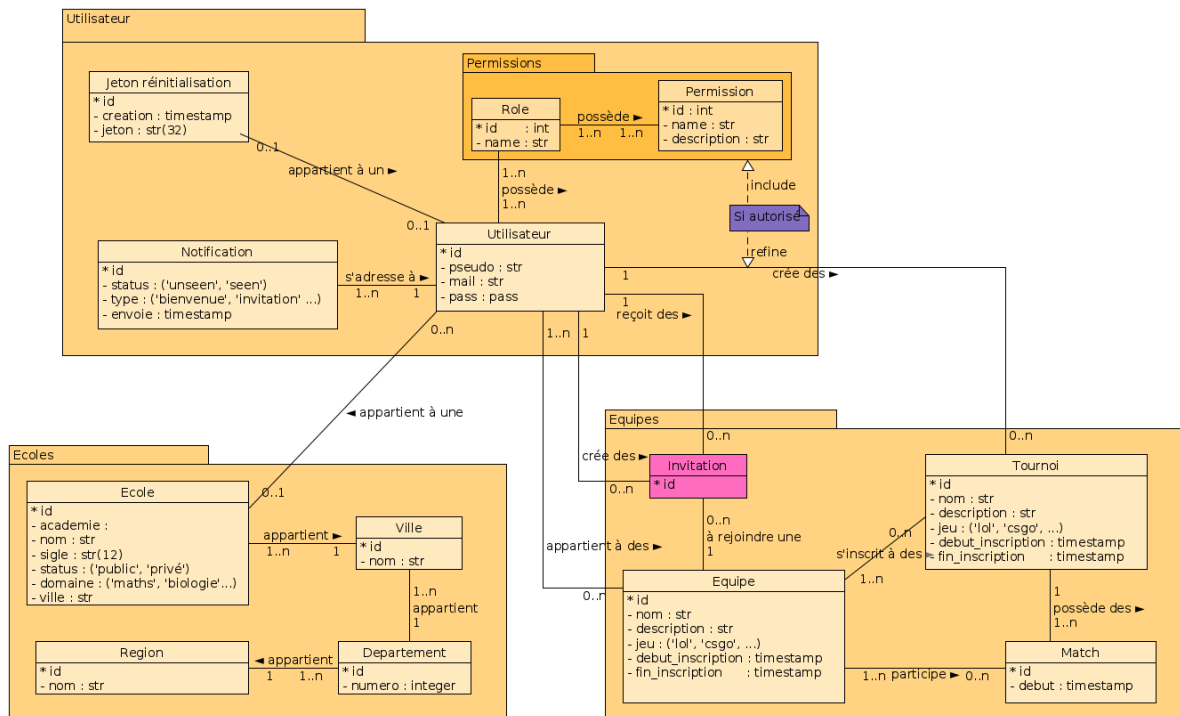


FIGURE 1 – Schéma UML de la base de données

Permission Représente une permission ponctuelle (créer une équipe, créer un tournoi, bannir un joueur, inscrire son équipe...)

Rôle Représente une ensemble de permissions (administrateur : toutes les permissions, modérateur : bannir un joueur, ...).

Utilisateur Possèdent différents rôles, et donc toutes les permissions liées aux rôles. Contient les informations relative à un utilisateur enregistré sur le site : mail, pseudo, mot de passe...

Le mot de passe est hashé à l'aide de la fonction PHP 'password_hash'. On protège ainsi efficacement l'accès par un tier au compte de l'utilisateur. (les détails techniques ne seront pas présentés ici, voir documentation [4](#) et [5](#))

Notification Un message destiné à attirer l'attention de l'utilisateur du site (e.x : le notifie d'une invitation à rejoindre une équipe)

Jeton de réinitialisation Une chaîne de 32 caractères générées aléatoirement. Ce jeton est généré et envoyé par mail à un utilisateur s'il a oublié son mot de passe. Il a alors 15 minutes pour changer son mot de passe en accédant au service consacré à cet effet.

Une base de données a été générée à partir de la page Wikipédia [6](#). Les données ont été extraite de la page, au format csv, à l'aide d'un programme Python (et de la bibliothèque BeautifulSoup [7](#)).

Cette base de donnée aura 2 utilités principales : auto-complétion pour la recherche, et organisation de tournois par région. Elle permettra également d'avoir des statistiques par région (fait t'on plus d'e-sport à Angers ou bien à Paris).

Equipe Les joueurs peuvent créer des équipes, inviter d'autres joueurs dans leur équipes, et rejoindre lorsqu'une invitation est reçu.

Tournoi Les utilisateurs possédant les permissions peuvent rejoindre et créer des tournois

Match Les utilisateurs ayant la permission peuvent créer des matchs dans un tournoi. Ils peuvent également être généré aléatoirement lors de la création d'un tournoi.

2.2.2 L'architecture du site

Le site suit une architecture MVC classique. La distinction entre les dossiers 'public' et 'src' est clair :

- 'public' : contenu auxquelles l'utilisateur à accès par son navigateur. La lecture des sources de ce dossier n'a aucunes conséquences sur la sécurité du site. Il contient l'API 'REST' public, et une page d'index qui fait appelle au Controller.
- 'src' : contenu 'privé', auxquelles seuls les développeurs ont accès. Il est séparé en 3 sous-dossiers (Model/View/Controller). L'architecture a été conçu de sorte à scale efficacement, et à pouvoir travailler à plusieurs en parallèle sur le projet en minimisant les conflits.

Le code est amplement commenté, et une attention particulière a été porté à sa clarté. Ainsi, nous vous invitons à regarder directement les sources pour comprendre l'architecture en profondeur.

```
public:
total 16
drwxrwxr-x 6 rpereira rpereira 4096 mai 15 17:29 api
drwxrwxr-x 4 rpereira rpereira 4096 mai 15 22:59 doc
-rw-rw-r-- 1 rpereira rpereira 949 mai 15 18:50 index.php
drwxrwxr-x 6 rpereira rpereira 4096 mai 2 12:09 res

src:
total 12
drwxrwxr-x 3 rpereira rpereira 4096 avril 27 19:49 Controller
drwxrwxr-x 3 rpereira rpereira 4096 mai 11 22:04 Model
drwxrwxr-x 3 rpereira rpereira 4096 mai 6 17:48 View

src/Controller:
total 36
-rw-rw-r-- 1 rpereira rpereira 246 mai 11 17:49 Aside.php
drwxrwxr-x 2 rpereira rpereira 4096 mai 15 17:33 Content
-rw-rw-r-- 1 rpereira rpereira 1601 mai 15 17:32 Content.php
-rw-rw-r-- 1 rpereira rpereira 253 avril 27 19:49 Footer.php
-rw-rw-r-- 1 rpereira rpereira 245 avril 24 18:38 Header.php
-rw-rw-r-- 1 rpereira rpereira 259 avril 27 19:50 Navbar.php
-rw-rw-r-- 1 rpereira rpereira 814 mai 11 18:24 PageElement.php
-rw-rw-r-- 1 rpereira rpereira 264 avril 27 19:51 Sidebar.php
-rw-rw-r-- 1 rpereira rpereira 259 avril 27 19:51 Toastbar.php

src/Model/ULC:
total 20
drwxrwxr-x 2 rpereira rpereira 4096 mai 11 18:04 BDD
drwxrwxr-x 2 rpereira rpereira 4096 mai 11 22:00 Discord
drwxrwxr-x 2 rpereira rpereira 4096 mai 11 21:58 LOL
drwxrwxr-x 2 rpereira rpereira 4096 mai 14 23:40 Utilisateur
-rw-rw-r-- 1 rpereira rpereira 857 mai 11 18:11 Utils.php

src/View:
total 48
-rw-rw-r-- 1 rpereira rpereira 15639 mai 15 18:38 aside.phtml
drwxrwxr-x 2 rpereira rpereira 4096 mai 15 17:32 Content
-rw-rw-r-- 1 rpereira rpereira 1290 mai 12 01:26 discord.phtml
-rw-rw-r-- 1 rpereira rpereira 19 avril 24 16:31 footer.phtml
-rw-rw-r-- 1 rpereira rpereira 1966 mai 15 16:48 header.phtml
-rw-rw-r-- 1 rpereira rpereira 6429 mai 15 23:24 navbar.phtml
-rw-rw-r-- 1 rpereira rpereira 2805 mai 15 18:32 sidebar.phtml
-rw-rw-r-- 1 rpereira rpereira 3701 mai 10 22:26 toastbar.phtml
```

FIGURE 2 – Liste des sources

2.2.3 API 'REST'

Une API a été implémenté sur le modèle REST.

La documentation a été générée via Doxygen, et est accessible à l'adresse <http://localhost:8080/doc/html/files.html>.

Cependant, elle ne peut pas être entièrement considéré comme totalement 'REST', car le serveur enregistre des informations dans la session PHP, ce qui entre en contradiction avec le principe 8. La satisfaction d'autres principes du 'REST' est également sujet à débat.

Mais elle reste pratique et ouverte : le client du site a été conçu en utilisant très largement cette API. (voir './res/js/api.js')

▼ game	
▼ lol	
▼ accounts	
▼ delete	
index.php	Supprime la liaison entre un compte League of Legends et l'utilisateur du site
▼ get	
index.php	Renvoie la liste des comptes League of Legend associé à un compte
▼ third-party-code	
▼ generate	
index.php	Génère un <i>third-party-code</i> permettant de lier un compte League of Legend à un compte utilisateur du site
▼ link	
index.php	Lis un compte League of Legends à l'utilisateur connecté du site. api/user/game/lol/third-party-code/generate/index.php

FIGURE 3 – *Partie de la page d'accueil de la documentation*

Lis un compte League of Legends à l'utilisateur connecté du site. api/user/game/lol/third-party-code/generate/index.php . Plus de détails...
Description détaillée
Lis un compte League of Legends à l'utilisateur connecté du site. api/user/game/lol/third-party-code/generate/index.php .
Relis l'utilisateur de la session courante au compte League of Legend passé en paramètre.
Arguments:
• summonerName : nom de l'invocateur (e.x: 'PCF toss')
Code erreur:
• 200: la liaison a bien été enregistré. • 400: requête mal formatté. (paramètre manquant, utilisateur non connecté, ou summonerID inexistant) • 408: aucun third-party-code n'est défini. • 409: le compte est déjà lié à un autre utilisateur du site. • 424: le code entré dans League of Legend n'est pas bon
Paramètres
• COOKIE PHPSESSID : le cookie de session PHP api/user/account/connect/index.php • POST summonerName : le nom d'invocateur du compte League of Legends à associer. (e.x: 'PCF toss')
Renvoie
- le code généré
• code réponse: • 200 : le code a été généré avec succès • 400 : erreur de la requête (paramètre(s) manquant(s) ou invalide(s)) • 408: aucun third-party-code n'est défini. • 409: le compte est déjà lié à un autre utilisateur du site. • 424: le code entré dans League of Legend n'est pas bon

FIGURE 4 – *Détail d'une requête (lié un compte Lol à l'utilisateur du site)*

2.3 Front-end

2.3.1 Le 'layout'

Le site a été construit en suivant cette leçon de *W3Schools*.

La barre de navigation (navbar), la barre sur le coté gauche (sidebar), et la barre à droite (asidebar) sont fixes, et leur contenu varient selon l'état du client (connecté ou non, rôles...)

Le centre du site contient le contenu de la page, et varient selon la page que l'utilisateur visite.

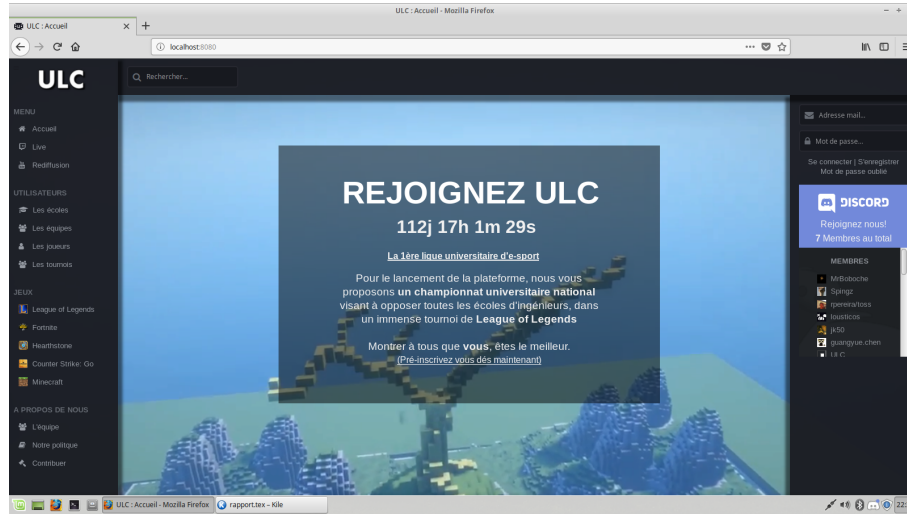


FIGURE 5 – Page d'accueil du site (non connecté)

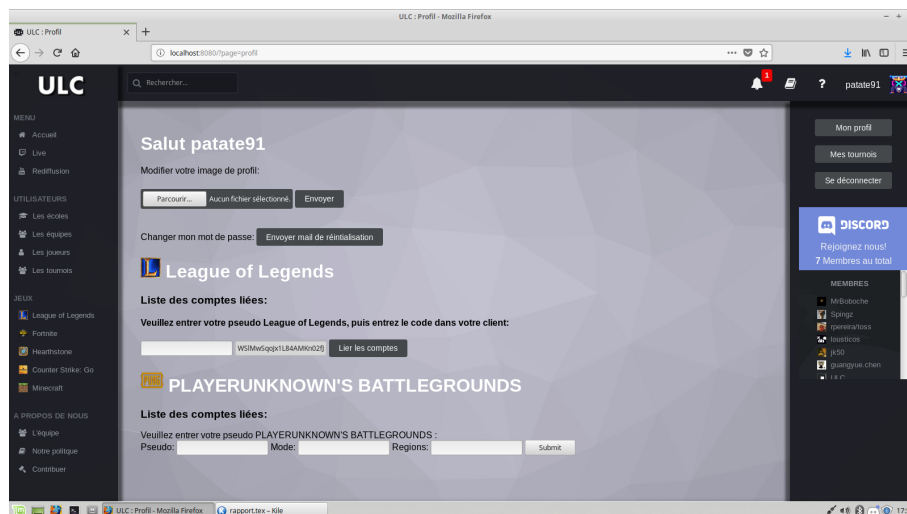


FIGURE 6 – Page de profil d'un joueur (connecté)

2.3.2 Discord

Le widget dans la 'sidebar' a été programmé à l'aide de l'API Discord ([3](#)), et d'un wrapper PHP (voir 'discord.phtml'). Cette API se configure simplement : chaque administrateur du Discord à accès à une clef d'API sans restrictions.

2.3.3 Notification

Le système de notification 'Facebook-like' fonctionne en temps réel. Le client récupère ces notifications toutes les 5 secondes (par l'intermédiaire de l'API). (voir cloche sur la figure [6](#))

3 Conclusion

4 Références

- [1] 'PHP The Right Way' - *+200 authors*
<http://www.phptherightway.com/>
- [2] API officiel de Riot Games
<https://developer.riotgames.com/>
- [3] API officiel de Discord
<https://discordapp.com/developers/docs/intro>
- [4] php.net - password_hash documentation
<http://php.net/manual/fr/function.password-hash.php>
- [5] Salage de mot de passe
[https://en.wikipedia.org/wiki/Salt_\(cryptography\)](https://en.wikipedia.org/wiki/Salt_(cryptography))
- [6] Liste écoles d'ingénieur en France - Wikipédia
https://fr.wikipedia.org/wiki/Liste_des_écoles_d'ingénieurs_en_France
- [7] BeautifulSoup - Documentation
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [8] API REST - Wikipédia
https://fr.wikipedia.org/wiki/Representational_state_transfer#Sans_état