

# 151 / Intégrer des éléments de base de données dans des applications Web

## Rapport personnel

Date de création : 09.02.2024  
Version 2 du 10.05.2024

---

Joye Eric



Module du 09.02.2024 au 10.05.2024

---

# Table des matières

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>1.1</b>	<b>Analyse .....</b>	<b>4</b>
1.1.1	Présentation du projet .....	4
1.1.2	Uses Case .....	5
1.1.3	Maquettes .....	5
1.1.4	Diagramme activité .....	7
1.1.5	Diagramme de séquences systèmes .....	8
1.1.6	Schéma ER .....	10
<b>1.2</b>	<b>Conception .....</b>	<b>11</b>
1.2.1	Diagrammes de classe .....	11
	a) Client .....	11
	b) Serveur .....	12
1.2.2	Schéma relationnel .....	13
1.2.3	Diagramme séquence interactions .....	14
1.2.4	Conception des tests .....	14
<b>1.3</b>	<b>Implémentation .....</b>	<b>15</b>
1.3.1	Descente de code .....	15
1.3.2	Problèmes rencontrés .....	15
1.3.3	Tests fonctionnels .....	15
1.3.4	Hébergement .....	15
<b>1.4</b>	<b>Synthèse .....</b>	<b>15</b>
1.4.1	Présentation réalisation .....	15
1.4.2	Différences entre planning et réalisation .....	15
1.4.3	Conclusion .....	16



## 1 Introduction

### 1.1 Analyse

Voici les consignes pour le projet :

- Le projet se fait individuellement.
- Le projet devra être publié sur serveur emf-informatique.
- Le projet doit comporter du côté client au moins deux zones différentes (visiteur et utilisateur connecté).
- L'application cliente HTML (JS-AJAX-JQuery) et celle serveur PHPo doivent être réalisées en Objet.
- Les données seront enregistrées dans une base de données mariaDB.
- La réalisation du projet doit être documentée. La documentation (exempte de fautes d'orthographe) devra contenir tous les éléments demandés dans le RP.

Nous avons aussi des consignes pour l'organisation / fonctionnement du projet : **Côté**

**client :**

- Il y a une séparation des fichiers images, html, js, css
  - Il ne doit y avoir de scripts js dans les fichiers html
  - Le projet fonctionne parfaitement avec tous les navigateurs
  - L'application cliente doit s'initier par un login-password
- Côté serveur :**
- A part les fichiers appelés par le client, les autres PHP sont des objets
  - Une séparation doit être faite entre les contrôleurs et les worker
  - Le serveur doit être appelé avec les bons paramètres (GET, POST, DELETE, PUT) et doit retourner les bons codes de réussite ou d'erreur
  - Le serveur doit retourner des données au format XML ou JSON
  - Le mot de passe doit être enregistré haché dans la base de données
  - L'information que l'utilisateur est bien logué doit être sauvée dans la session de l'utilisateur
  - Le projet doit être robuste face aux injections HTML, JS et SQL

#### 1.1.1 Présentation du projet

##### a) Description

Bienvenue sur "League of Creations", la plateforme interactive dédiée aux joueurs de League of Legends et/ou de création artistique ! Notre site web offre une expérience unique permettant aux utilisateurs de découvrir, créer et partager leurs propres champions LoL à travers des fan arts.

##### b) Fonctionnalités

- Accédez à une galerie complète des champions de League of Legends avec leur informations détaillées.
- Connectez-vous pour personnaliser votre expérience sur "League of Creations".

- Une fois connecté, exprimez votre créativité en créant de nouveaux champions LoL sous forme de fan arts.
- Ajoutez des détails tels que le nom, l'histoire (la description), la régions... • Parcourez les champions créés par d'autres membres de la communauté.
- Exprimez votre appréciation en "likant" les créations que vous adorez.

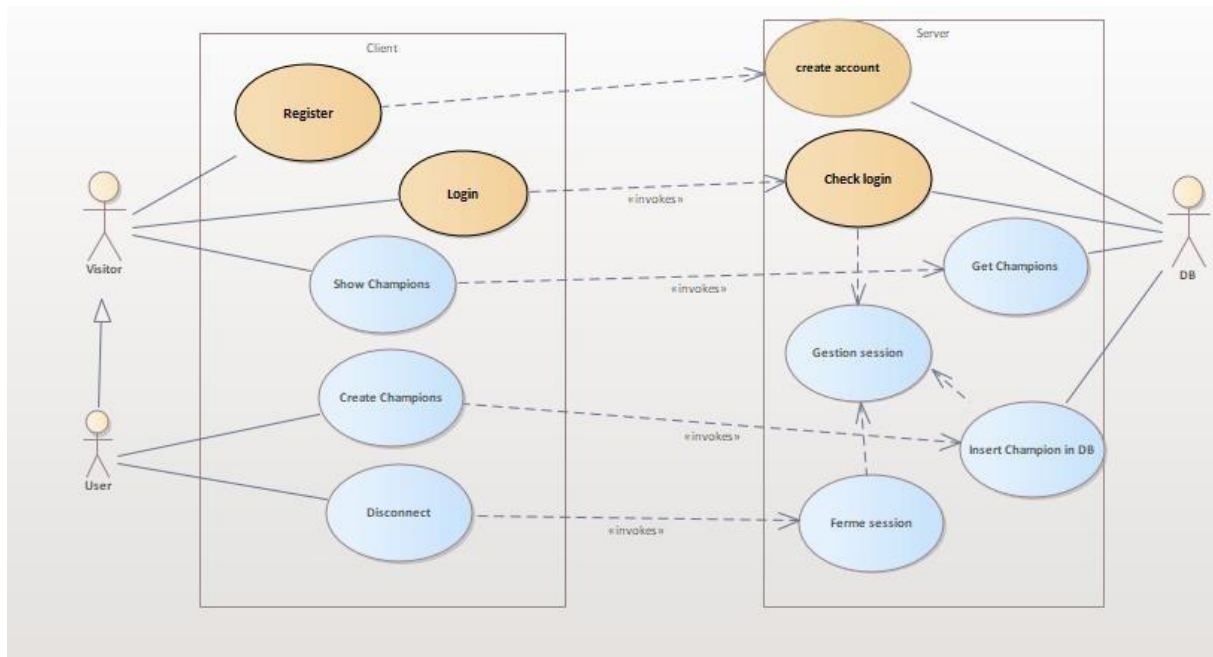
### c) Objectif du projet

"League of Creations" vise à créer une communauté dynamique et créative où les joueurs de League of Legends peuvent partager leur passion pour le jeu à travers l'art et l'imagination. Nous encourageons la collaboration, l'inspiration et le respect mutuel au sein de notre communauté.

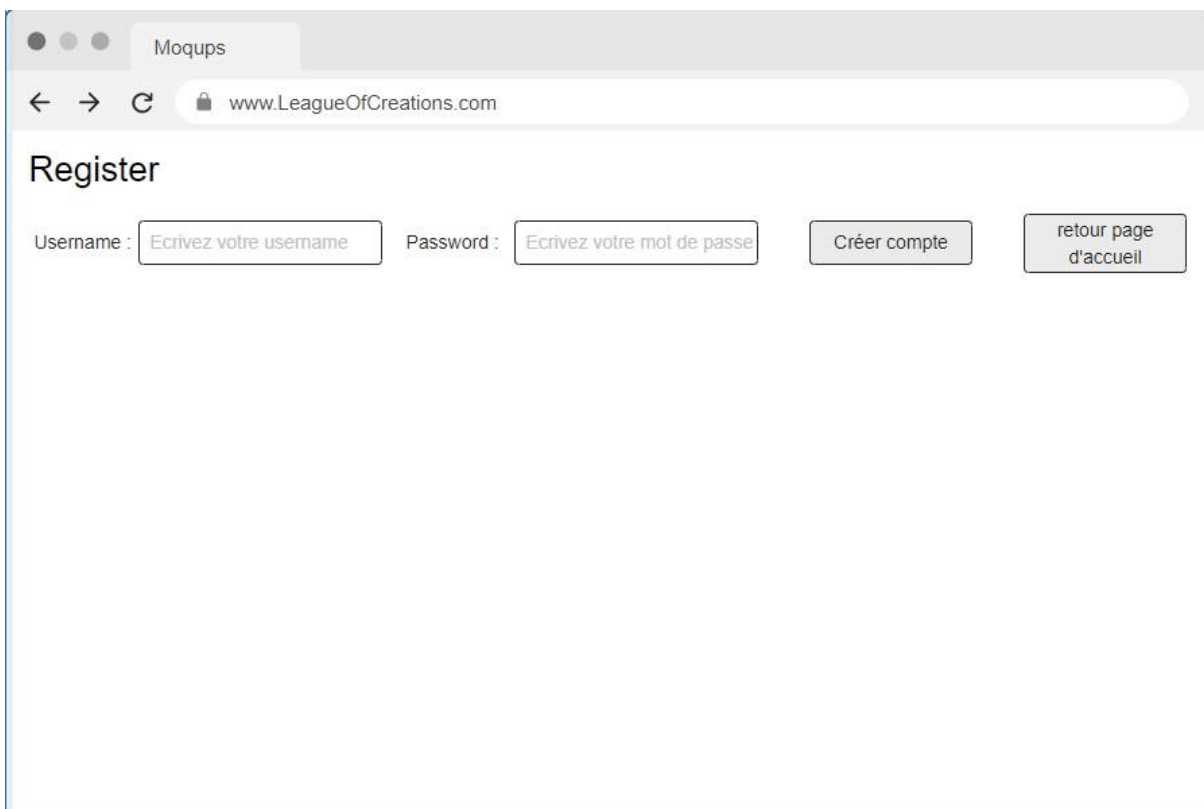
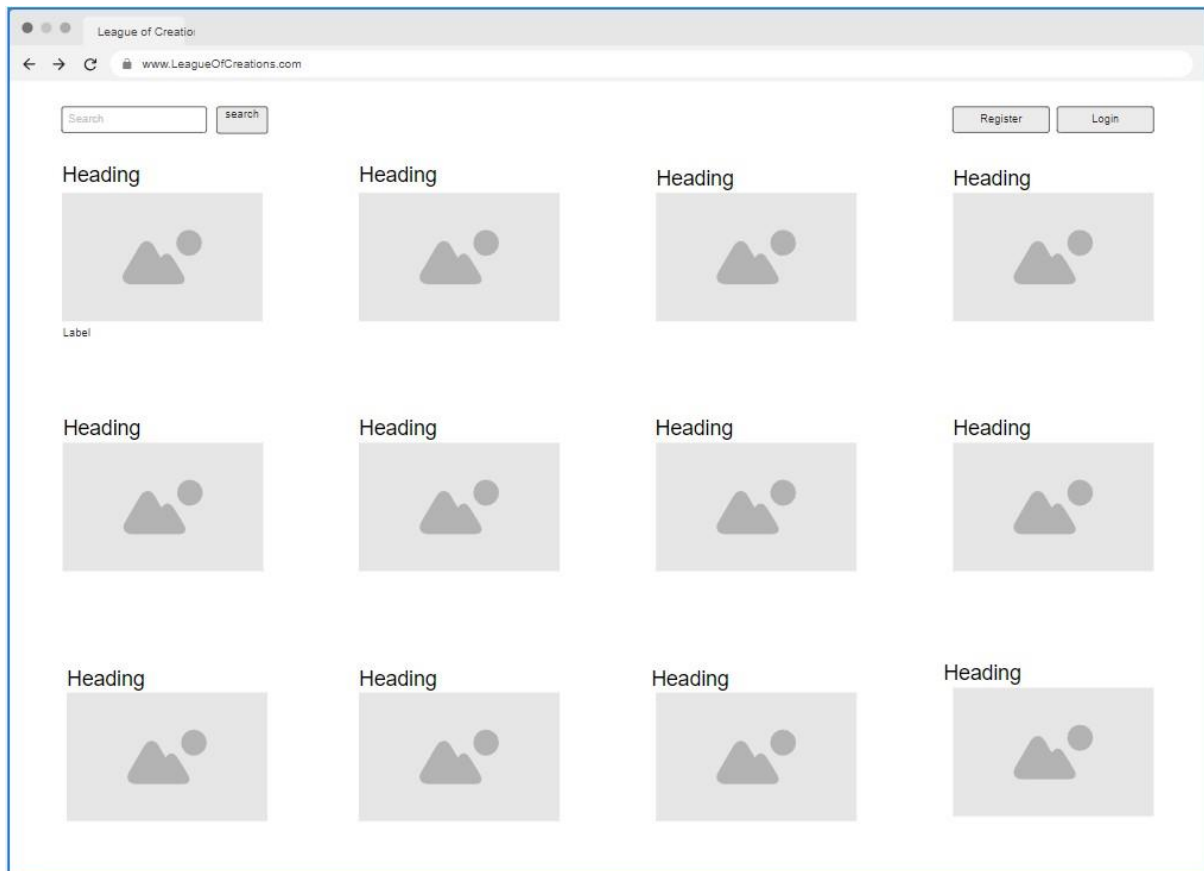
Rejoignez-nous sur "League of Creations" et donnez vie à vos champions préférés sous une nouvelle perspective artistique !

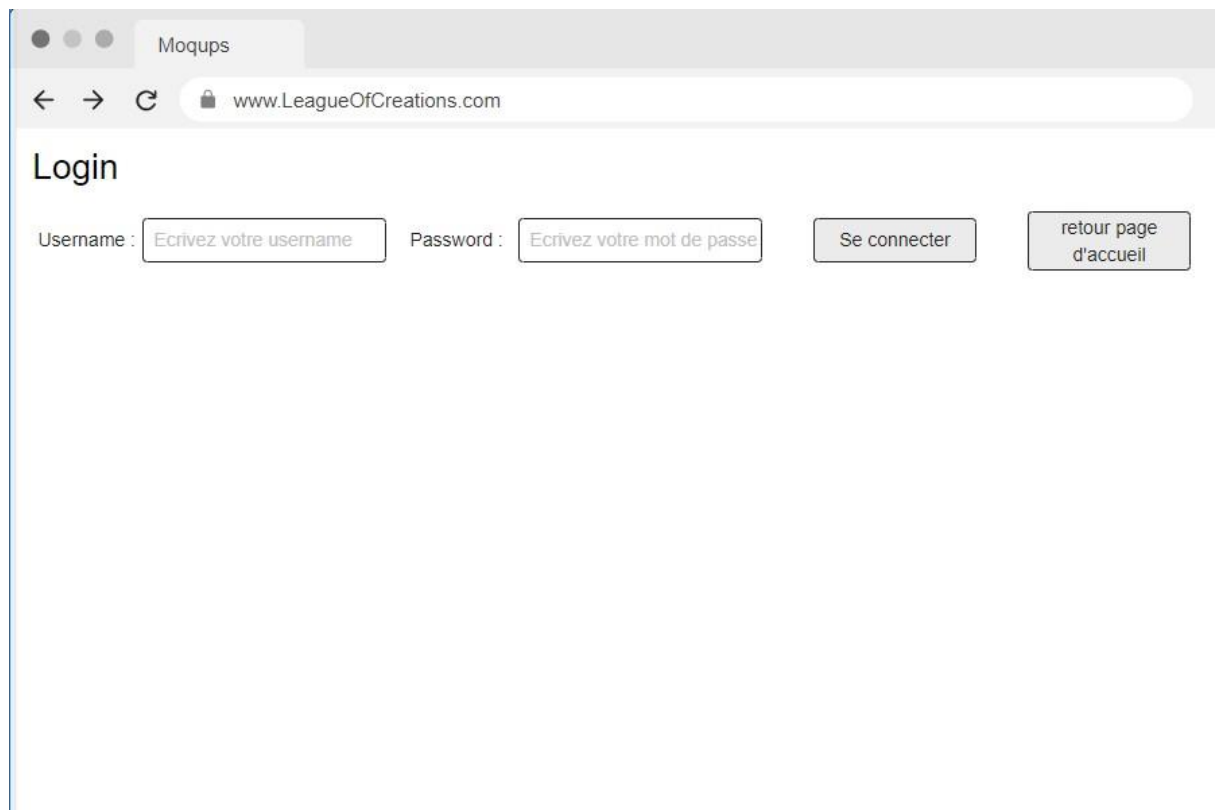
### 1.1.2 Uses Case

Voici le l'Use Case de mon projet :



### 1.1.3 Maquettes





Moqups

← → ↻ www.LeagueOfCreations.com

## Login

Username :  Password :



Moqups

← → ↻ www.LeagueOfCreations.com

## Création de champion

Nom du champion

Description du champion

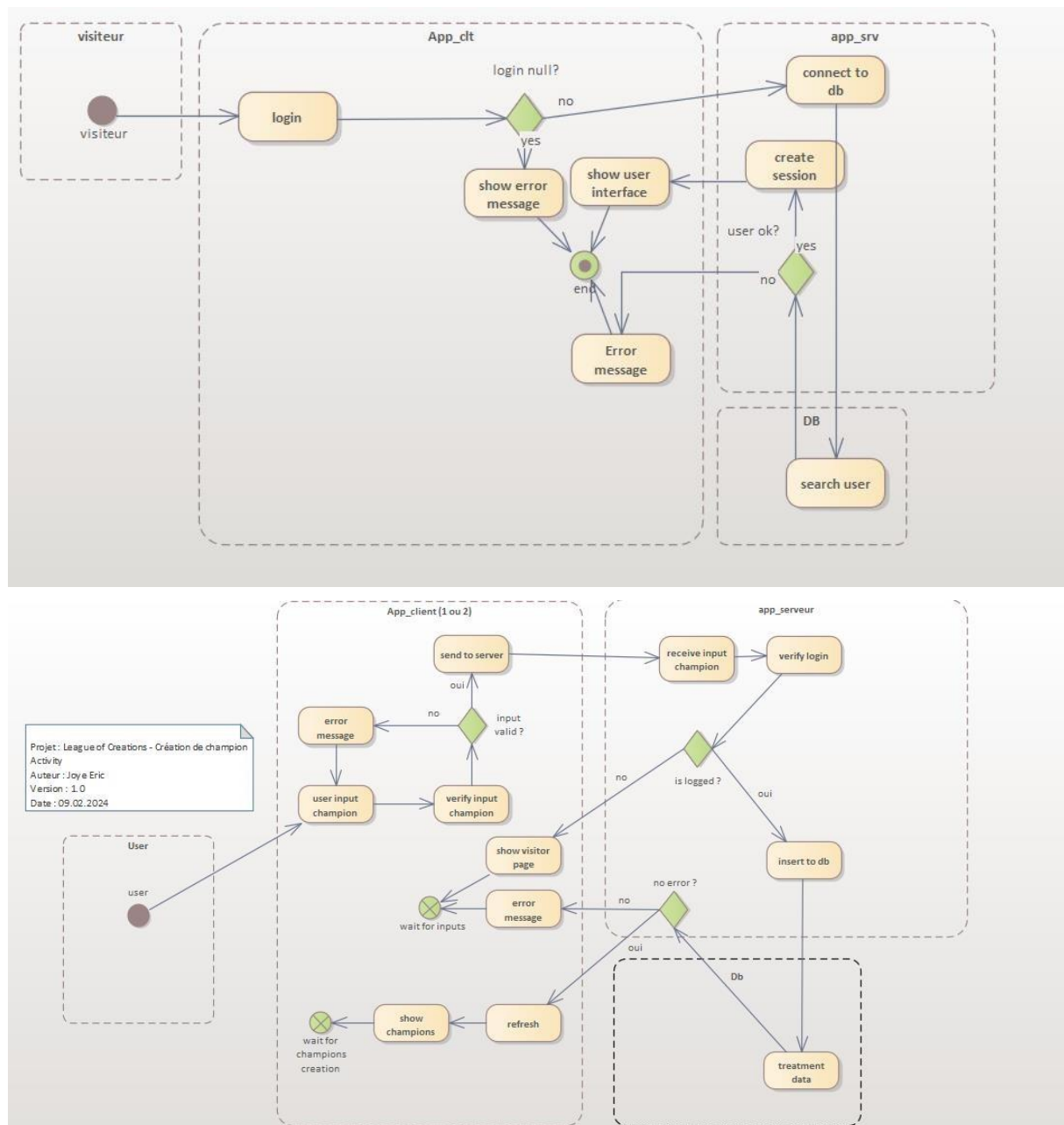
☒ Mana

Region

Rôle ☒ Top ☒ Mid ☒ Jungle ☒ Ad carry ☒ Support

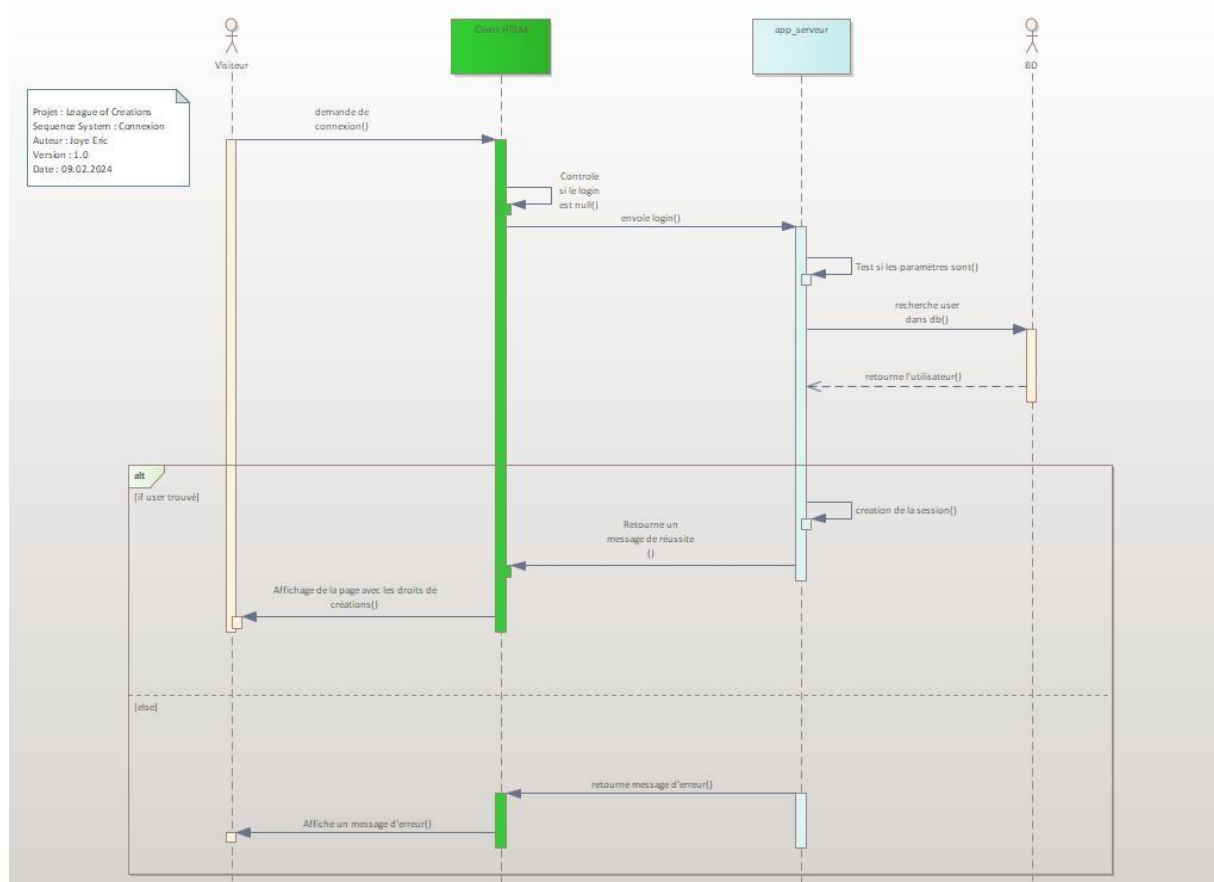
Type

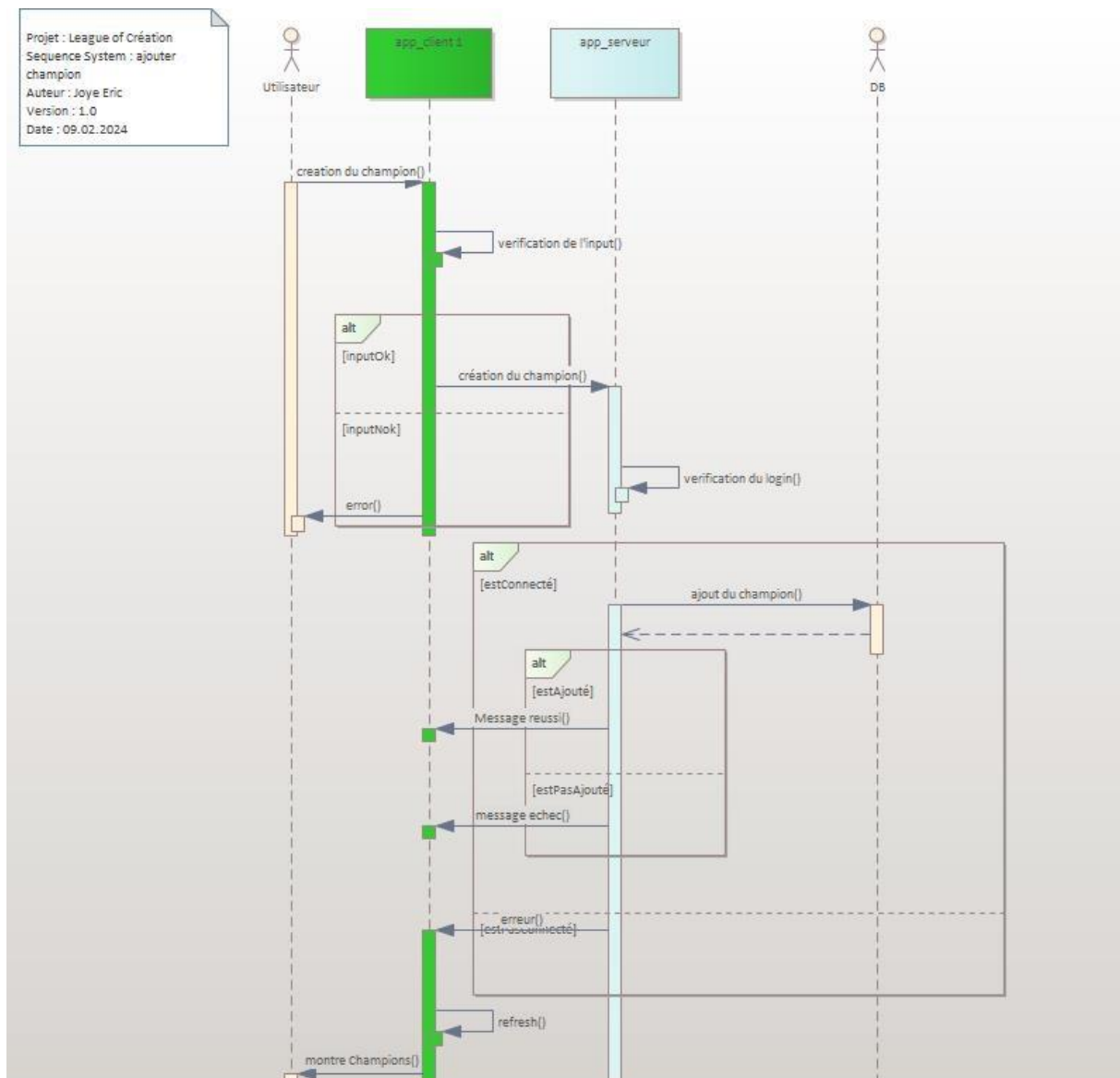
## 1.1.4 Diagramme activité



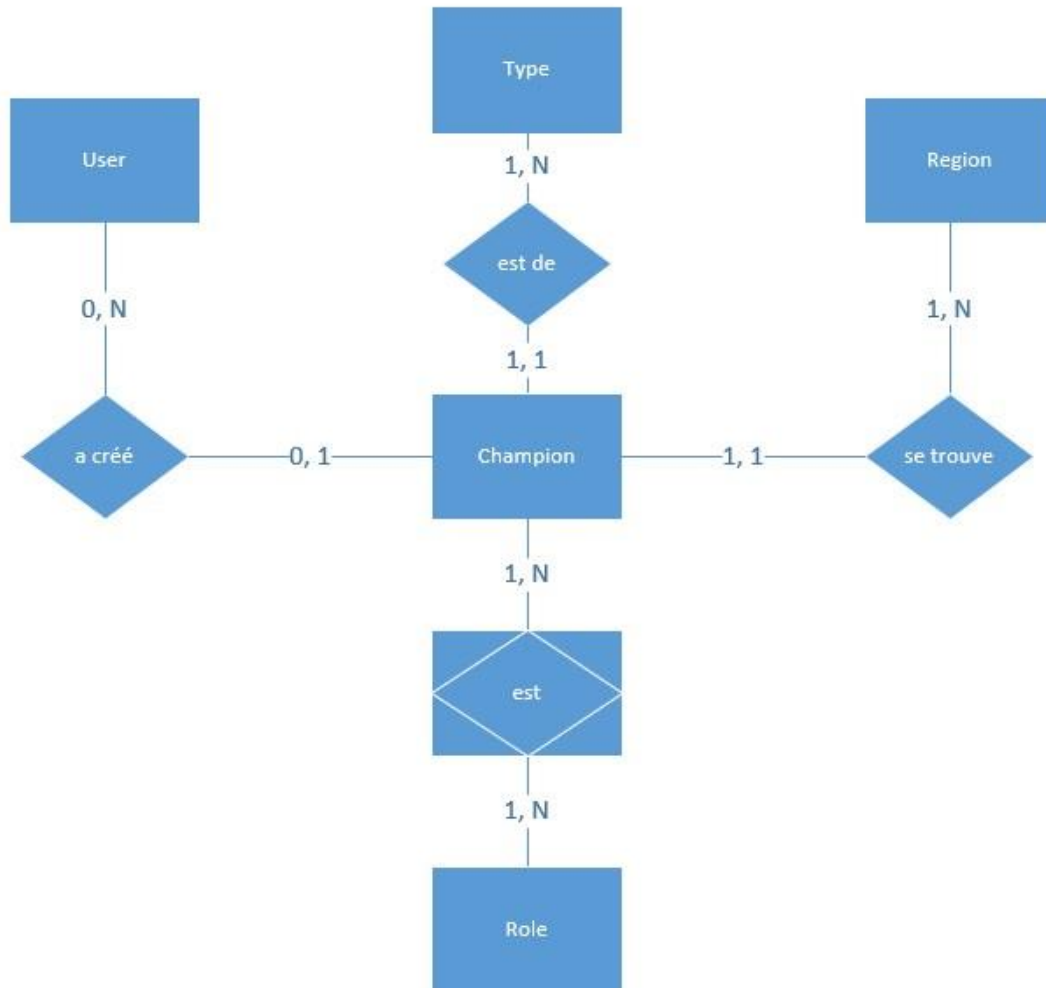


## 1.1.5 Diagramme de séquences systèmes





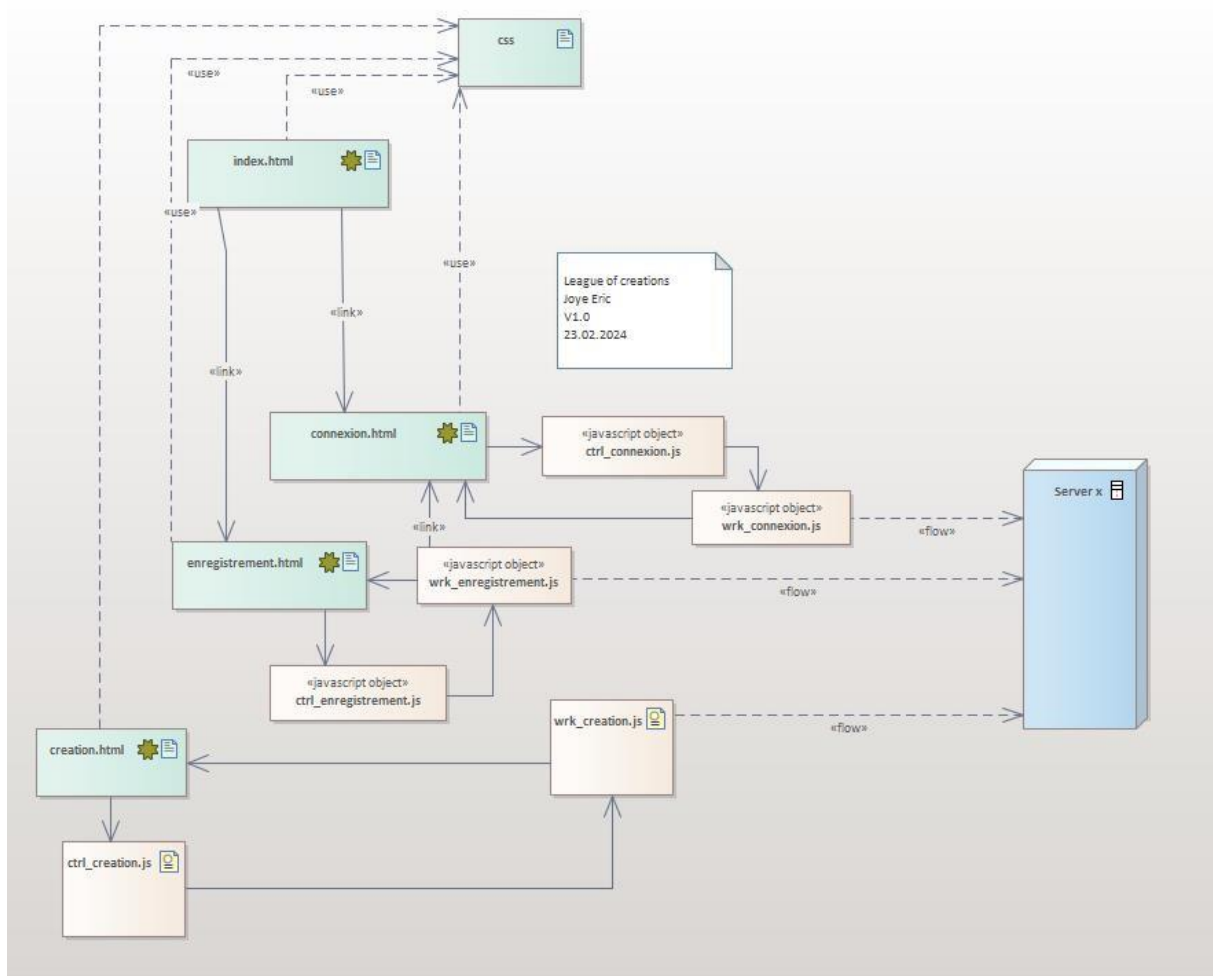
### 1.1.6 Schéma ER



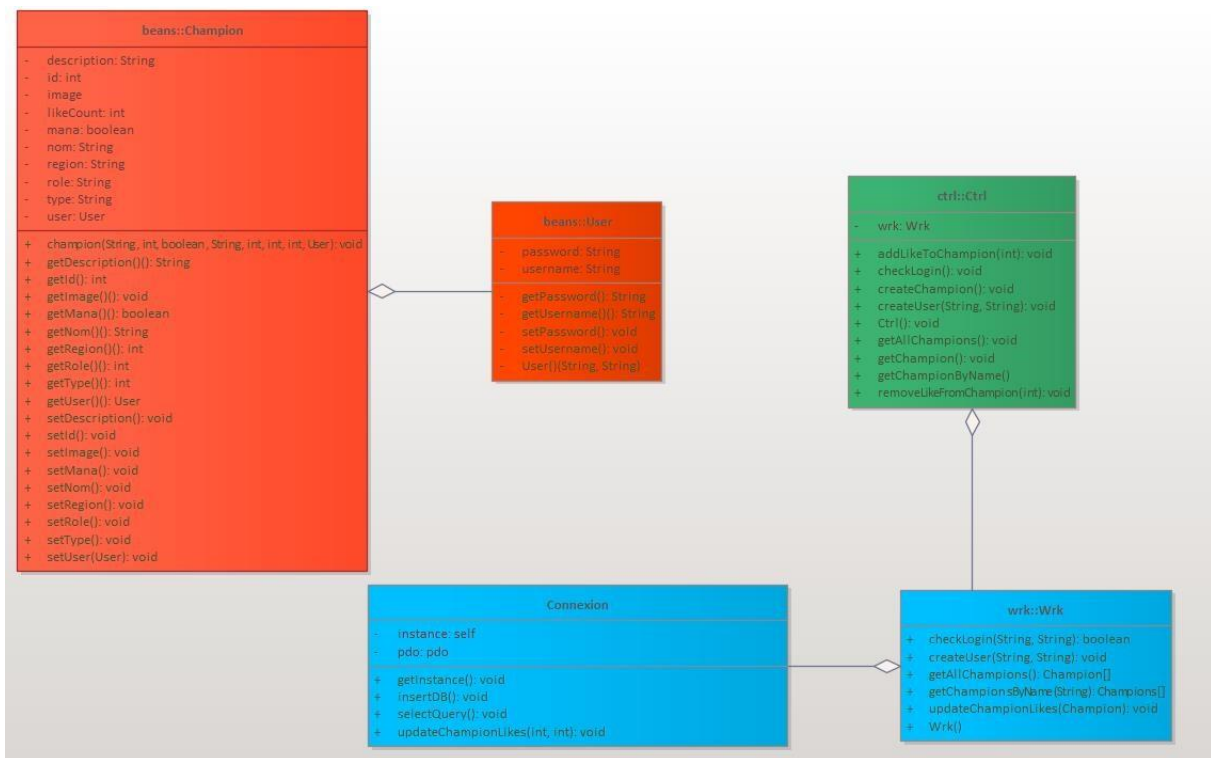
## 1.2 Conception

### 1.2.1 Diagrammes de classe

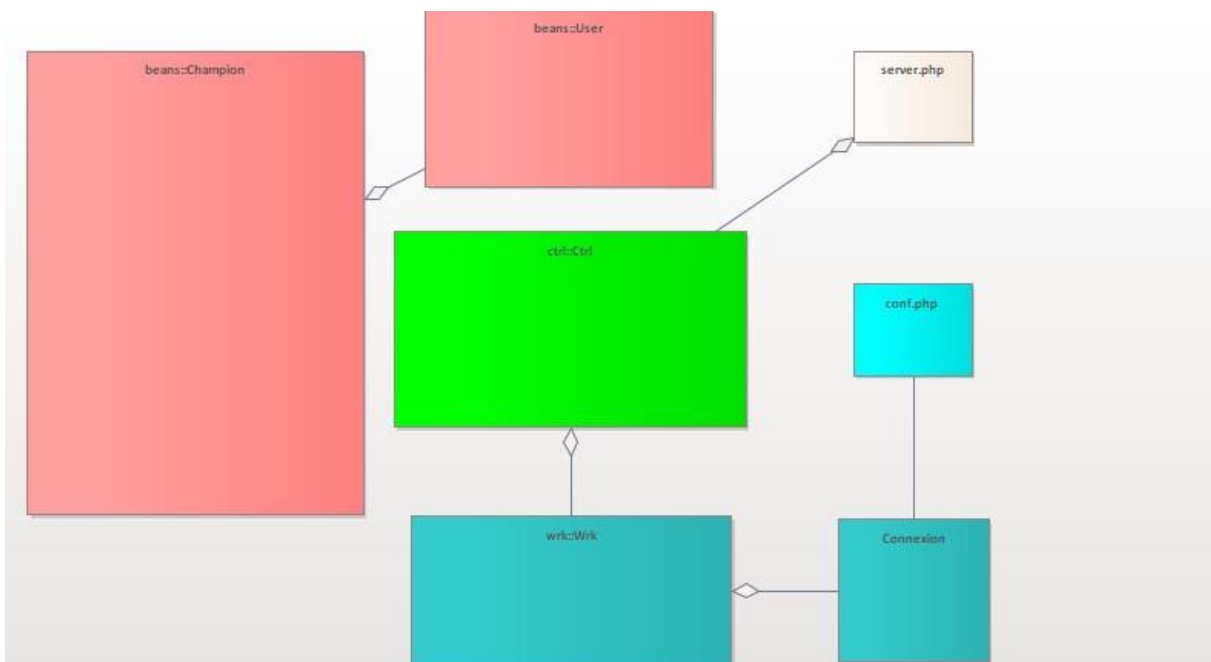
## a) Client



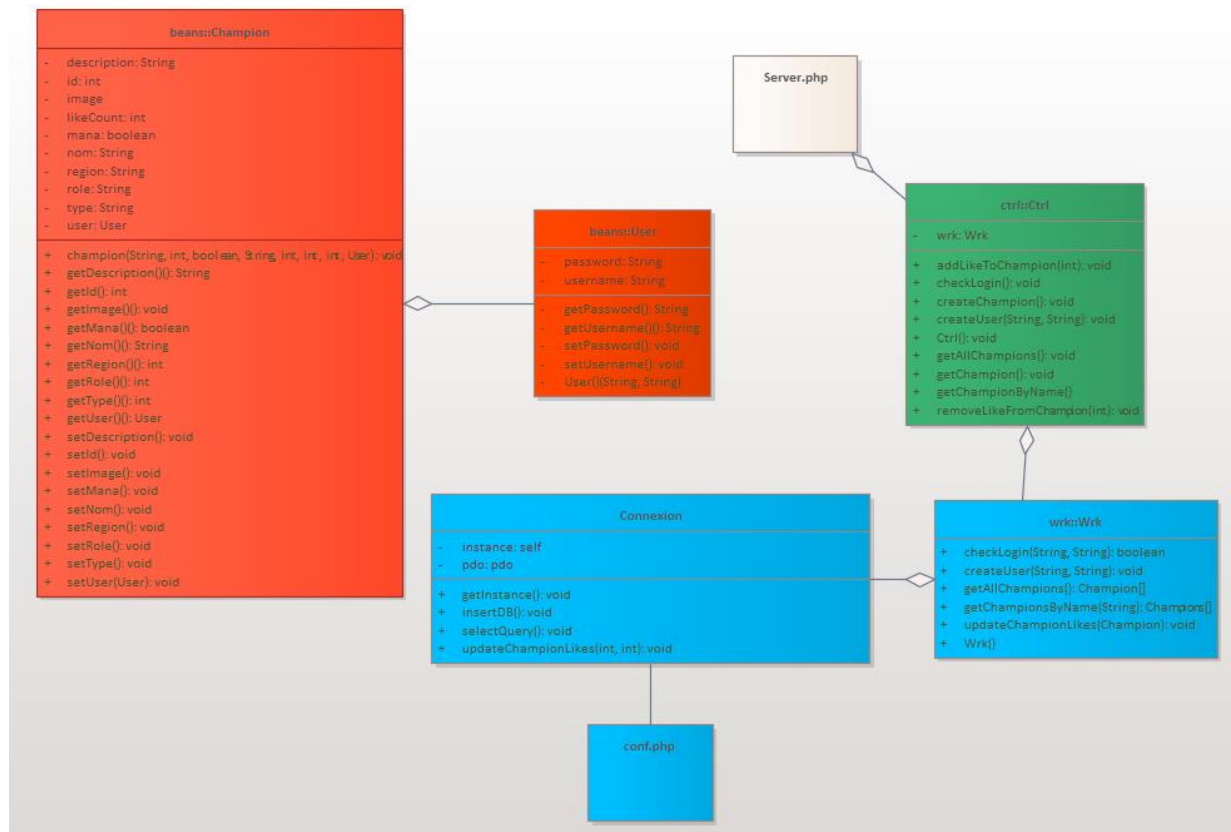
## b) Serveur



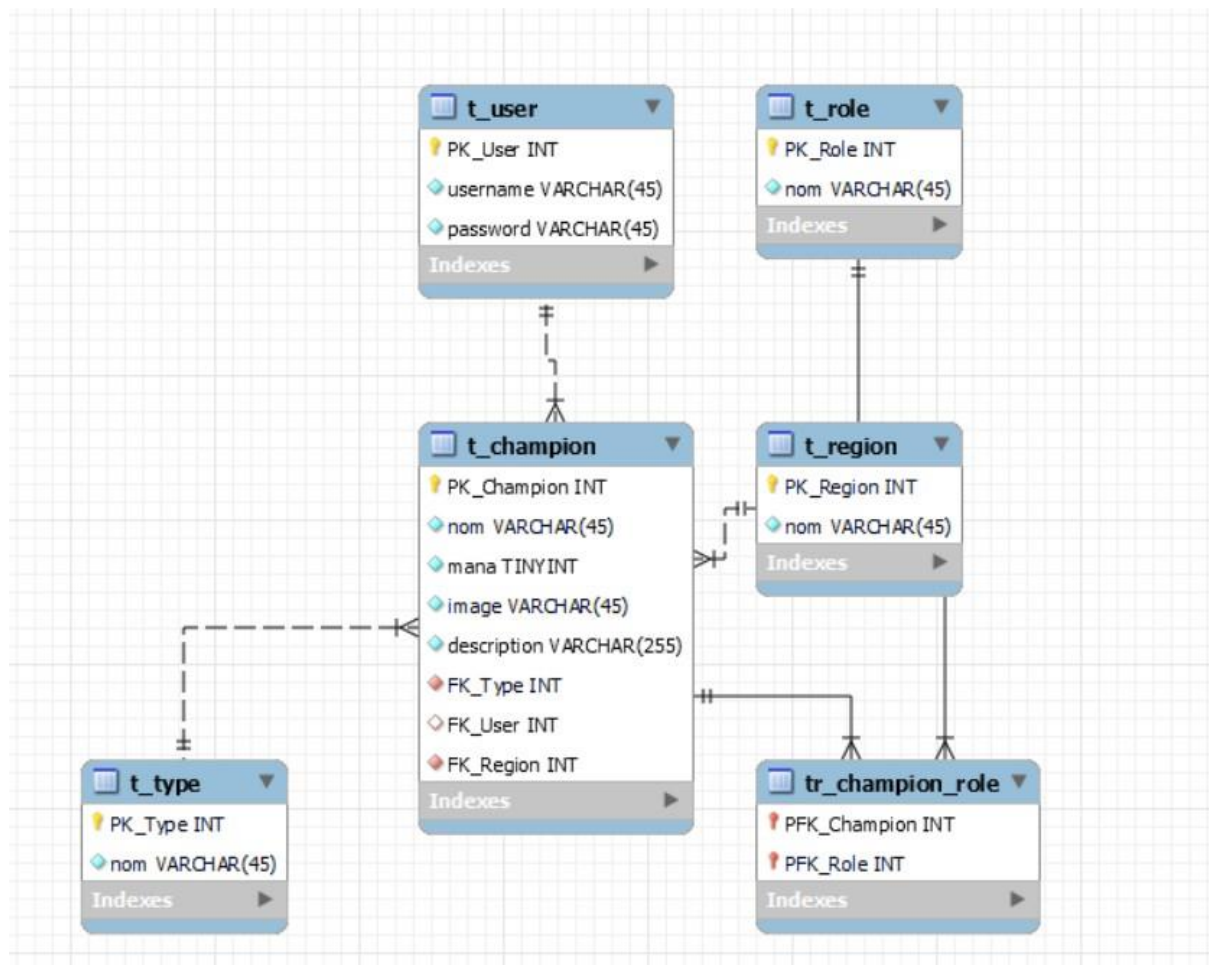
Voici le diagramme corrigé juste je n'ai pas les attributs et méthodes dedans :



Voici la dernière version :



## 1.2.2 Schéma relationnel



## Diagramme séquence interactions

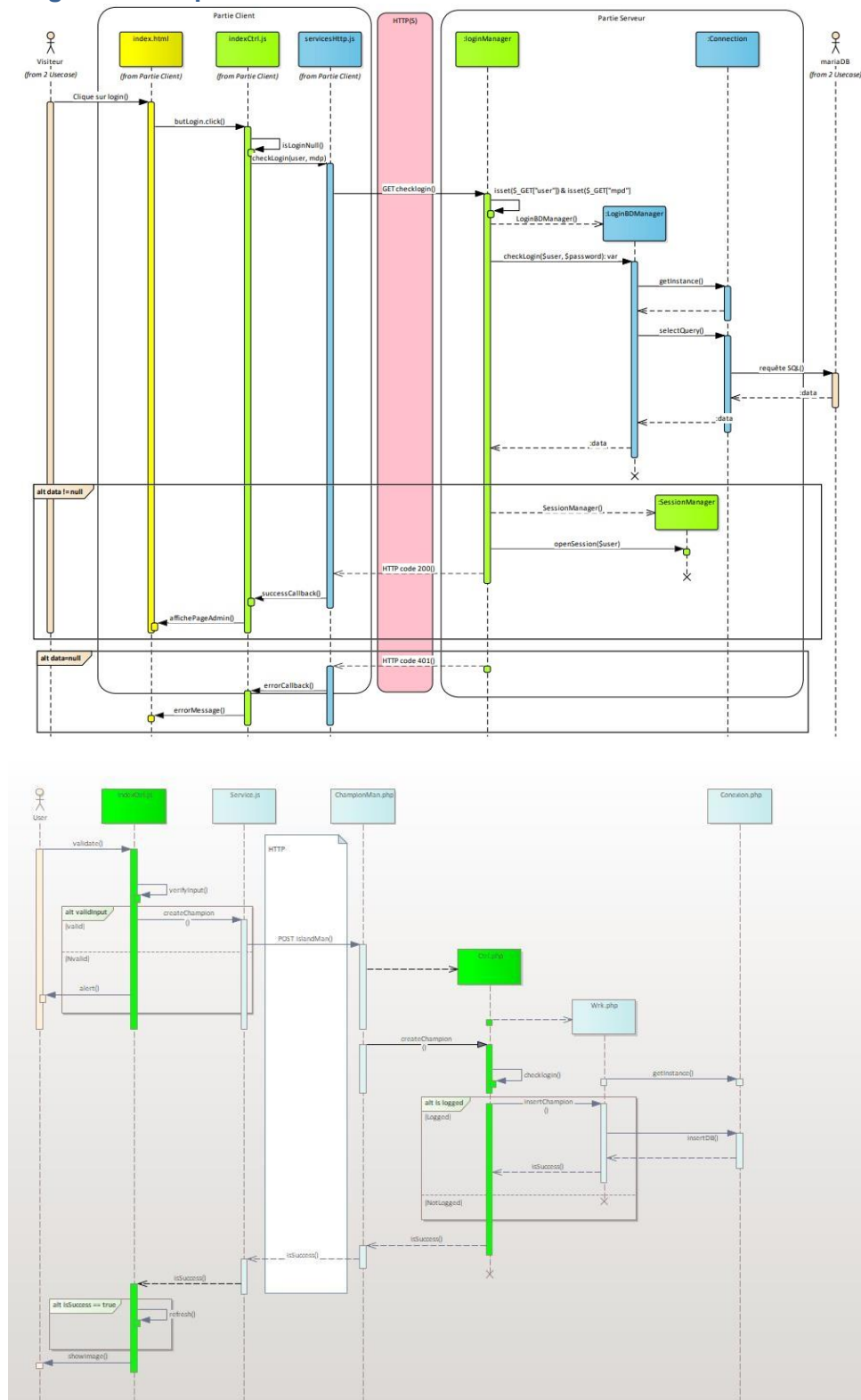
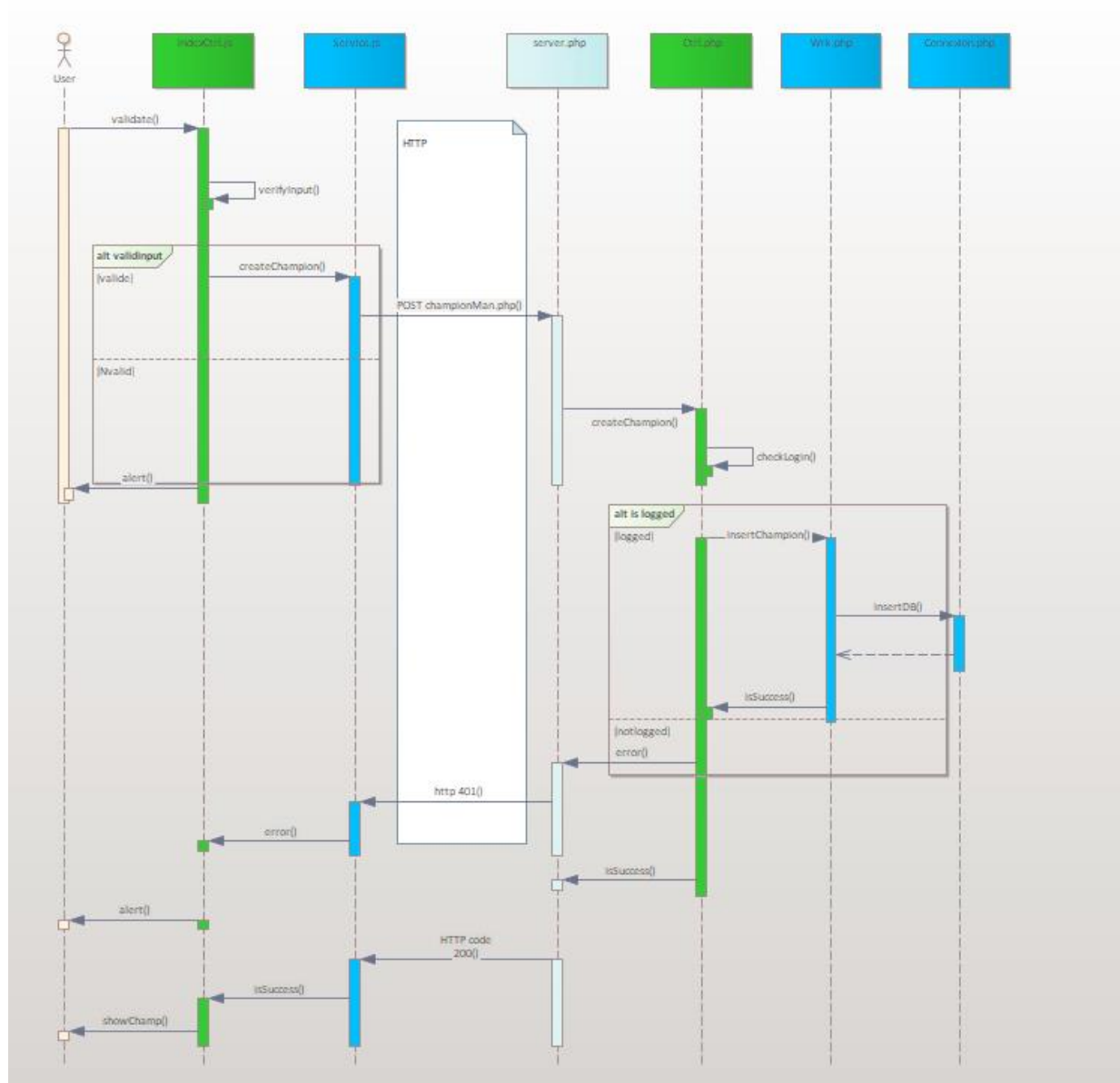


Schéma modifié avec les codes https :





### Conception des tests

Description du test	Résultat Attendu	Résultat Obtenu	Test OK/NOK
Ajout d'un champion étant connecté	Ajout du champion dans la DB et affichage de la page d'accueil.	Retour sur la page d'accueil avec le nouveau champion ajouté	OK
Ajout d'un champion n'étant pas connecté	Alerte d'une erreur 401	Erreur 401 User Not Authenticated	OK

Ajout d'un champion étant connecté, sans nom	Message qui apparait en bas de la page spécifiant de remplir tous les champs	Message en bas de la page indiquant que les champs requis ne sont pas tout remplis	OK
Ajout d'un champion étant connecté, sans image	Message qui apparait en bas de la page spécifiant de remplir tous les champs	Message en bas de la page indiquant que les champs requis ne sont pas tout remplis	OK
Ajout d'un champion étant connecté, avec un nom existant	Un message d'erreur 403.	Message d'alerte Erreur Refused 403	OK
Ajout d'un champion étant connecté, Injection SQL Image : salut','0','0','0','0',' ','4'&quot;;drop table;--	La requête est transformée en texte.	Un champion est ajouté mais avec ce qui a été entré dans le champ reste du texte	OK
Ajout d'un champion étant connecté, Injection HTML Nom : <a href="google.com">Salut</a>	Les balises se font transformer en texte.	Une erreur 403 Refused apparait	NOK
Ajout d'un champion étant connecté, Injection HTML Nom : <h1>Salut</h1>	Les balises se font transformer en texte.	Les balises s'affichent bien en texte	OK

## 1.3 Implémentation

### 1.3.1 Descente de code pour créer un champion

#### IndexCtrl.js

```
createChampion() {
  var name = $("#name").val();
  var image = $("#image").val();
  var description = $("#description").val();
  var mana = document.getElementById("mana").checked ? 1 : 0;
  var i = document.getElementById("region");
  var region = i.options[i.selectedIndex].value;
  var j = document.getElementById("type");
  var type = j.options[j.selectedIndex].value;
  var roles = [];
  $("#roles input[type='checkbox']:checked").each(function () {
    roles.push($(this).attr("value"));
  });
  if (name && image && description && region != "default" && type != "default" && roles.length > 0) {
```

```

        wrk.createChampion(name, image,description,
mana,roles,region,type,ctrl.createSuccess, ctrl.createError);
    }else{
        $("#Cresult").text("Required Field Not Full");
    }
}

createError(){
    alert("Error Creating champion");
}
createSuccess(){
    ctrl.loadAccueilCo();
}

```

On prend la valeur de chaque champ que l'on va donner au Wrk pour faire la requête ajax. Attention on vérifie que les champs de texte obligatoire ne soient pas vides.

### server.php

```

<?php
// header('Access-Control-Allow-Origin: http://localhost:8080');
// header('Access-Control-Allow-Credentials: true');
// header('Access-Control-Allow-Headers: Content-Type');
require_once("beans/Champions.php");
require_once("beans/Region.php");
require_once("beans/Type.php");
require_once("beans/Users.php");
require_once("ctrl/Ctrl.php");
require_once("wrk/Wrk.php");
require_once("wrk/Connexion.php");
session_start();
$Ctrl = new Ctrl();

// Suppression de cette ligne car elle semble redondante avec la suite du
code

if ($_SERVER['REQUEST_METHOD'] == "GET") {
    if (isset($_GET['action'])) {
        if ($_GET['action'] == "champion") {
            echo $Ctrl->getChampion();
        } else {
            phpinfo();
        }
    }
}

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (isset($_POST['action'])) {
        if ($_POST['action'] == "create") {
            $Ctrl->createChampion($_POST['name'], $_POST['image'],
$_POST['description'], $_POST['mana'], $_POST['type'], $_POST['roles'],
$_POST['region']);
        }
        if ($_POST['action'] == "register") {
            echo $Ctrl->registerUser(new Users($_POST['username'],
$_POST['password']));
        }
    }
}

```

```

        if ($_POST['action'] == "checkLogin") {

            if (isset($_SESSION['nom']) && isset($_SESSION['id'])) {
                http_response_code(200);
            } else {
                http_response_code(401);
            }

        }

        if ($_POST['action'] == "login") {
            $username = $_POST['username'];
            $password = $_POST['password'];
            $Ctrl->login($username, $password);

        }

        if ($_POST['action'] == "deconnexion") {
            session_destroy();
        }

    }
}
$Ctrl = null;

```

La requête va passer dans les différents tests et va aller appeler la méthode d'ajout dans le Ctrl en lui passant tous les paramètres.

### Ctrl.php

```

public function createChampion($name, $image, $description, $mana, $type,
$roles, $region){
    if (isset($_SESSION['nom']) && isset($_SESSION['id'])) {
        $champion = new Champions(htmlspecialchars($name), $mana
,htmlspecialchars($image), htmlspecialchars($description) , $type, null,
$region, $roles);
        $champion->setUser($_SESSION['id']);
        $result = $this->wrk->insertChampion($champion);
        if ($result) {
            http_response_code(200);
        } else {
            http_response_code(500);
        }
    } else {
        http_response_code(401);
    }
}

```

Création d'un objet champion ainsi que test de la session pour pouvoir envoyer l'objet au Wrk.

### Wrk.php

```

public function insertChampion($champ)
{
    try {
        $this->con->startTransaction();
        $res = $this->con->executeQuery("INSERT INTO t_champion (`nom`,
`mana`, `image`, `description`, `FK_Type`, `FK_User`, `FK_Region`) VALUES
(?, ?, ?, ?, ?, ?);", array($champ->getNom(), $champ->getMana(), $champ-
>getImage(), $champ->getDescription(), $champ->getType(), $champ->getUser(),
$champ->getRegion()));
        $var = $this->con->getLastId("t_champion");
        $role = $champ->getRole();
    }
}

```

```

        foreach (explode(",", $role) as $item) {
            $this->con->selectSingleQuery("INSERT INTO tr_champion_role
(`PFK_Champion`,`PFK_Role`) VALUES (?,?)", array($var, $item));
        }
        $this->con->commitTransaction();
        if ($res == 1) {
            return true;
        } else {
            return false;
        }
    } catch (PDOException $e) {
        $this->con->rollbackTransaction();
        print "Erreur !: " . $e->getMessage() . "<br/>";
        die;
    }
}

```

Insertion du champion dans la db.

### 1.3.2 Problèmes rencontrés

Le seul problème que j'ai eu, c'était de comprendre comment la session fonctionner. J'ai pris beaucoup de temps pour réussir à rendre un projet ok.

### 1.3.3 Tests fonctionnels

Ajout d'un champion étant connecté	Ajout du champion dans la DB et affichage de la page d'accueil.	Retour sur la page d'accueil avec le nouveau champion ajouté	OK
Ajout d'un champion n'étant pas connecté	Pas possible	Pas possible	OK
Ajout d'un champion étant connecté, sans nom	Message qui apparait en bas de la page spécifiant de remplir tous les champs	Message en bas de la page indiquant que les champs requis ne sont pas tout remplis	OK
Ajout d'un champion étant connecté, sans image	Message qui apparait en bas de la page spécifiant de remplir tous les champs	Message en bas de la page indiquant que les champs requis ne sont pas tout remplis	OK
Ajout d'un champion étant connecté, avec un nom existant	Ok, on peut avoir plusieurs fois le meme nom	Champion créé	OK

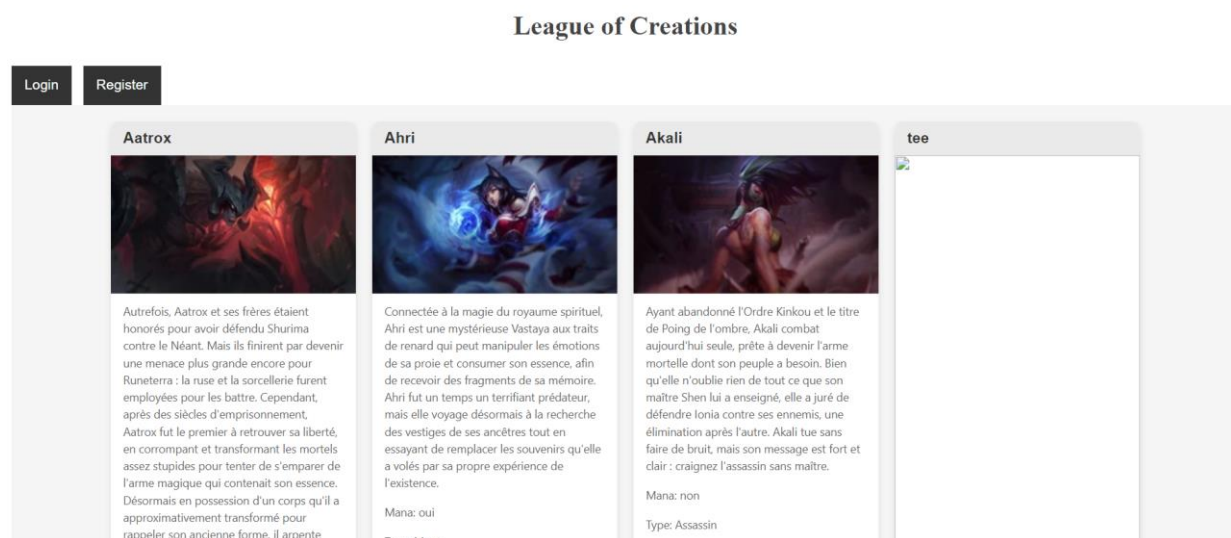
Ajout d'un champion étant connecté, Injection SQL Image : salut','0','0','0','0',' ','4'&quot;;drop table;--	La requête est transformée en texte.	Un champion est ajouté mais avec ce qui a été entré dans le champ reste du texte	OK
--	--------------------------------------	--	----

### 1.3.4 Hébergement

Pour l'hébergement du site sur EMFProd il a fallu ajouter la base de données en utilisant phpmyadmin. Il a fallu créer un utilisateur pour la base de données ainsi qu'une base de données en donnant les permissions à l'utilisateur. Ensuite j'ai ouvert une connexion FTP pour transférer tout le projet sur le site.

## 1.5 Synthèse Présentation réalisation

Page d'accueil :



Page d'inscription :

**Inscription**

eric

\*\*\*\*

S'inscrire

retour

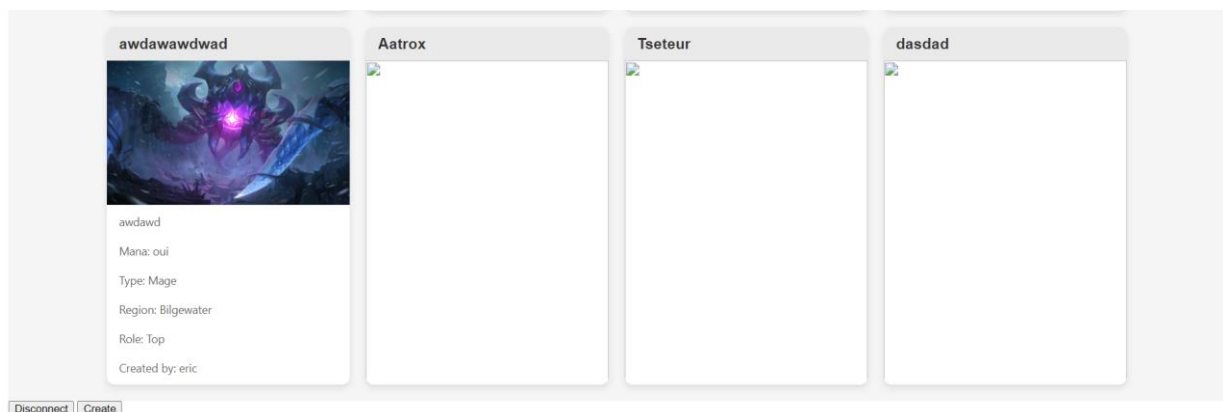
Page de connexion :

## Connexion

Se connecter

retour

Page d'accueil une fois connecté :



## Différences entre planning et réalisation

Les différences sont qu'il était prévu d'avoir un système de like pour voter pour sortir un champion dans le jeu. J'aurais du faire deux listes différentes pour les champions dont une ou on voit que les champions de league of legend pour la page quand on est pas connecté et une ou on voit tous les champions créé quand on se connecte.

### 1.4.3 Conclusion

Ce que j'ai aimé durant ce module c'est le fait que ça soit un projet et qu'on puisse avancer à notre rythme. J'ai pu beaucoup apprendre dans le développement web. Malheureusement je n'ai pas pu faire ce que je voulais durant ce module.