



PLAYERS TO PLAYERS

UNE APPLICATION WEB D' ALEXANDRE METGY, MORVAN LASSAUZAY ET VICTOR NEA

PROJET POUR L'UE DEVELOPPEMENT D'APPLICATIONS RETICULAIRES 2016 - 2017

SOMMAIRE

I)	Description de l'application.....	3
a.	Présentation.....	3
b.	Cas d'utilisation.....	4
	i. Créer un compte.....	4
	ii. Se connecter.....	4
	iii. Modifier ses préférences.....	4
	iv. Créer une partie.....	5
	v. Supprimer une partie créée.....	6
	vi. Rechercher un jeu.....	6
	vii. Rechercher une partie par rapport à un jeu cherché.....	7
	viii. Rejoindre une partie en Ile-de-France.....	8
	ix. Consulter le profil d'un autre joueur.....	9
	x. Devenir ami avec un autre joueur.....	10
	xi. Supprimer un profil de sa liste d'amis.....	11
II)	Développement de l'application.....	12
a.	Choix technologiques et APIs.....	12
	i. Hébergement Amazon Web Services.....	12
	ii. GiantBomb.....	12
	iii. Google Maps.....	11
	iv. Design.....	12
	v. Outils et environnement de développement.....	12
b.	Architecture.....	14
	i. Organisation des données.....	14
	ii. Schéma d'application.....	14
c.	Implémentation des servlets.....	16
	i. Communication avec l'API GiantBomb.....	19
	ii. Implantation de l'API Google Maps.....	20
	iii. Implantation de Hibernate.....	21

I) DESCRIPTION DE L'APPLICATION

a. Présentation



Qu'est-ce que cette application ?

Players To Players est un nouveau réseau social de joueurs de jeux vidéo dont le but est de les réunir entre eux afin qu'ils puissent jouer ensemble. Vous n'avez pas envie de jouer seul le soir après votre journée de travail ? Vous avez un super jeu de combat mais vous trouvez que ce n'est pas drôle de jouer seul ? Rejoignez cette nouvelle communauté de joueurs franciliens, invitez vos amis, rejoignez des joueurs autour de vous, ou organisez vous-même la prochaine soirée jeux vidéo inoubliable de l'Île-de-France.



D'où vient cette idée ?

En tant que passionnés de jeux vidéo, nous voulions créer une application dans ce thème. Les contraintes imposées par le cours nous ont naturellement amenés vers cette idée : il fallait réunir les gens pour qu'ils puissent jouer ensemble. De plus, après quelques petites recherches, il s'est avéré qu'il n'existait pas aujourd'hui de site qui propose ce type de service.



Quelles sont ses fonctionnalités ?

Players To Players vous permet :

- de trouver des personnes autour de vous qui partagent les mêmes consoles ou goûts
- de trouver des parties selon vos critères et préférences
- de rejoindre une partie créée autour de vous
- d'héberger une partie chez vous
- de lier des amitiés avec d'autres joueurs, pour rester en contact plus facilement



Quel est le contenu de ce rapport ?

Dans ce rapport, nous vous présenterons d'abord les fonctionnalités de l'application avec des cas d'utilisation. Ils vous expliqueront comment l'utiliser, de la création d'un compte jusqu'à la création d'un hébergement de partie.

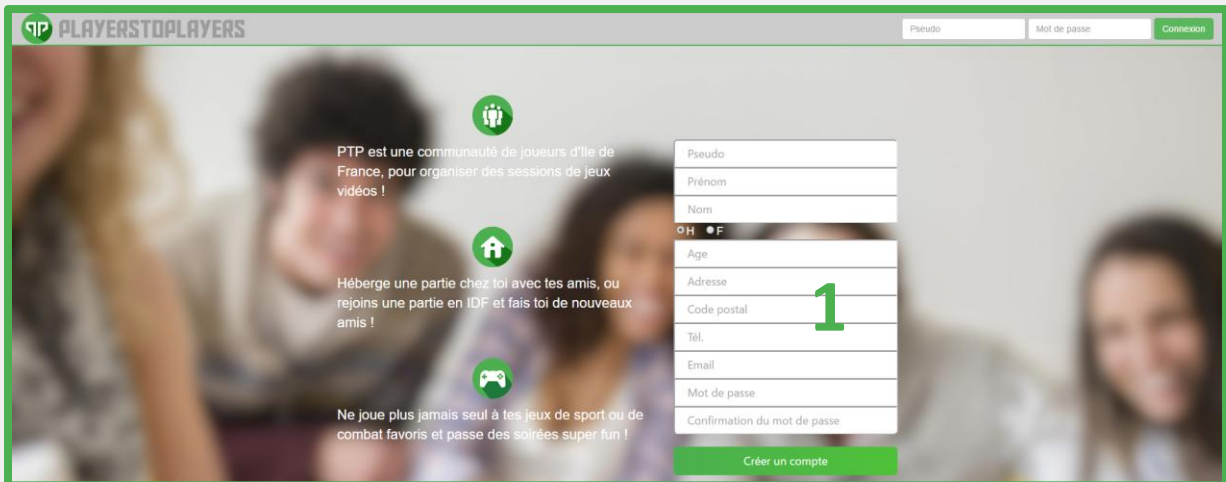
Nous entrerons ensuite dans les détails techniques du projet. Dans un premier temps, nous vous indiquerons les choix que nous avons faits concernant les outils et les API. Dans un second temps, nous nous intéresserons à l'architecture, du schéma de l'application jusqu'au détail des servlets en passant par l'organisation des données.

b. Cas d'utilisation

i. Comment crée-t-on un compte sur Players To Players ?

Remarque : Un compte est nécessaire sur Players To Players pour utiliser ses services.

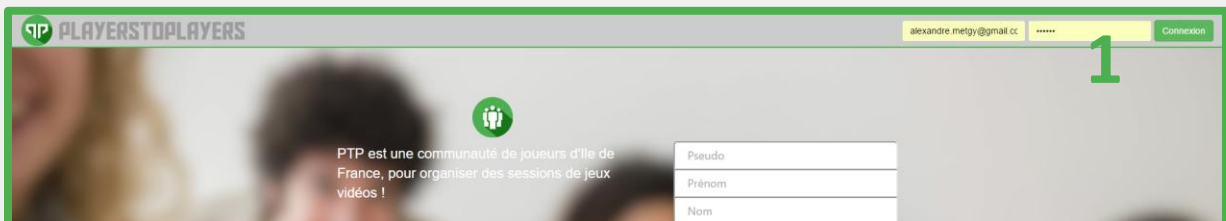
Utilisation : Morvan accède à Players To Players via son navigateur et l'URL pour la première fois. Il remplit le formulaire d'inscription (1) sur la page d'accueil de l'application. En particulier, il veille à indiquer une adresse email syntaxiquement valide ainsi qu'un mot de passe qui coïncide avec sa confirmation.



ii. Comment se connecte-t-on ?

Prérequis : Posséder un compte (i)

Remarque : On se connecte à l'application via la page d'accueil de création d'un nouveau compte.



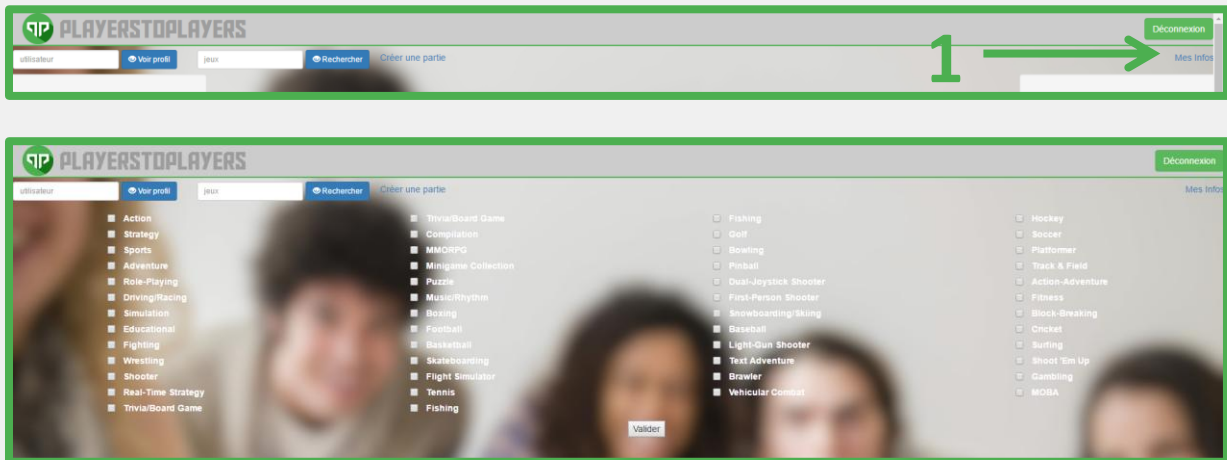
Utilisation : Depuis la page d'accueil, Alexandre entre ses identifiants dans le formulaire de connexion de la barre de menu (1), puis valide en cliquant sur le bouton « Connexion ».

iii. Comment édite-t-on ses préférences ?

Prérequis : Être connecté (ii)

Remarque : On peut modifier sa liste de jeux et les consoles possédées, ainsi que les genres de jeux auxquels on joue.

Utilisation : Depuis la page d'accueil, Victor accède aux informations qu'il souhaite éditer à partir du lien « Mes infos » (1). Sur la page MyGameTypes, il coche/décoche les genres de jeux auxquels il joue/ne joue pas dans le formulaire des genres (2), puis valide via le bouton « Valider ».



iv. Comment crée-t-on une partie ?

Prérequis : Être connecté (ii)

Remarque : On peut héberger une partie à l'adresse désirée, renseigner les consoles et jeux possédés

Utilisation : Depuis la page d'accueil, Victor accède au formulaire de création d'une partie à partir du lien « Créer une partie » (1). Sur la page CreateGameSession, il remplit le formulaire (2). Pour cela, il renseigne un nom qu'il veut donner à la partie, une description. Il choisit aussi la(les) plateforme(s) qu'il possède, ainsi que les jeux. Enfin il renseigne l'adresse de la partie ainsi que la date prévue. Un message de confirmation indique que la partie a bien été créée (3). Victor accède ensuite aux parties qu'il a créées depuis le menu « Mes infos » rubrique « Mes parties » (4). Dans la nouvelle page MyGameSessions, il vérifie que la partie qu'il a créée est bien enregistrée (5).

Créer une partie

Label
Tekken Party

Description
Jouer à Tekken 2 sur PS2, n'hésitez pas à ramener des manettes !

Plateformes
PlayStation 3
Recherche terminée.
PlayStation 3

Jeux
Tekken 6
Recherche terminée.
Tekken 6

Adresse
4 rue de la grande truanderie

Code postal
75001

Date
12-11-2016 19:00

Créer partie



v. Comment supprime-t-on une partie que l'on a créée ?

Prérequis : Avoir créé une partie (iv)

Remarque : On peut supprimer une partie que l'on a déjà créée : toutes les informations la concernant seront supprimées

Utilisation : Alexandre est dans la liste des parties qu'il a créées. Il sélectionne la partie à supprimer, « tekken party », en cliquant sur le bouton « Détails » (1). Dans la nouvelle fenêtre qui recense les détails de la partie, il clique sur le bouton « Supprimer » (2). Un message de confirmation apparaît pour indiquer que l'opération de suppression a bien abouti (3).




vi. Comment recherche-t-on un jeu ?


Prérequis : Être connecté (ii)

Remarque : On peut rechercher n'importe quel jeu auquel on souhaite jouer grâce à l'API GiantBomb


Utilisation : Depuis la page d'accueil, Victor entre le jeu « tekken » dans le formulaire de recherche de jeux puis clique sur le bouton « Rechercher » (1). Dans la liste des jeux contenant le mot « tekken » dans la nouvelle page SearchGameResult, il accède au jeu « Tekken 7 » en cliquant sur le lien correspondant (2). Il arrive sur la page Game avec les informations du jeu « Tekken 7 ».




Nom : [Galaga: Tekken 20th Anniversary Edition](#)
Date de sortie : "2015-04-30 00:00:00"



1 Nom : [Tekken 7](#)
Date de sortie : "2015-02-18 00:00:00"



Nom : [Pachi-Slot Tekken 2nd](#)
Date de sortie : "2013-05-21 00:00:00"



Nom : "Tekken 7: Fated Retribution"
Note :
Description : "In addition to the obligatory balance tweaks, this upgrade to Tekken 7 adds additional characters - including Akuma from the Street Fighter series."
Date de sortie : "2016-07-05 00:00:00"
Genres :
Plateformes : Arcade, PC, Xbox One, PlayStation 4
Développeurs :
Éditeurs :
Jeux similaires :

2


Ajouter à mes jeux

vii. Comment rejoint-on une partie pour un jeu que l'on a cherché ?

Prérequis : Avoir cherché un jeu (v) et une partie avec ce jeu a été créée (iv)

Remarque : Une liste des parties apparaît pour un jeu que l'on a cherché

Utilisation : Victor est sur la page du jeu « Tekken 7 : Fated Retribution ». Dans le menu de gauche des parties pour ce jeu, il clique sur le bouton « Détails » de la partie qu'il va rejoindre (1). Il rejoint ensuite la partie comme expliqué dans (vii).

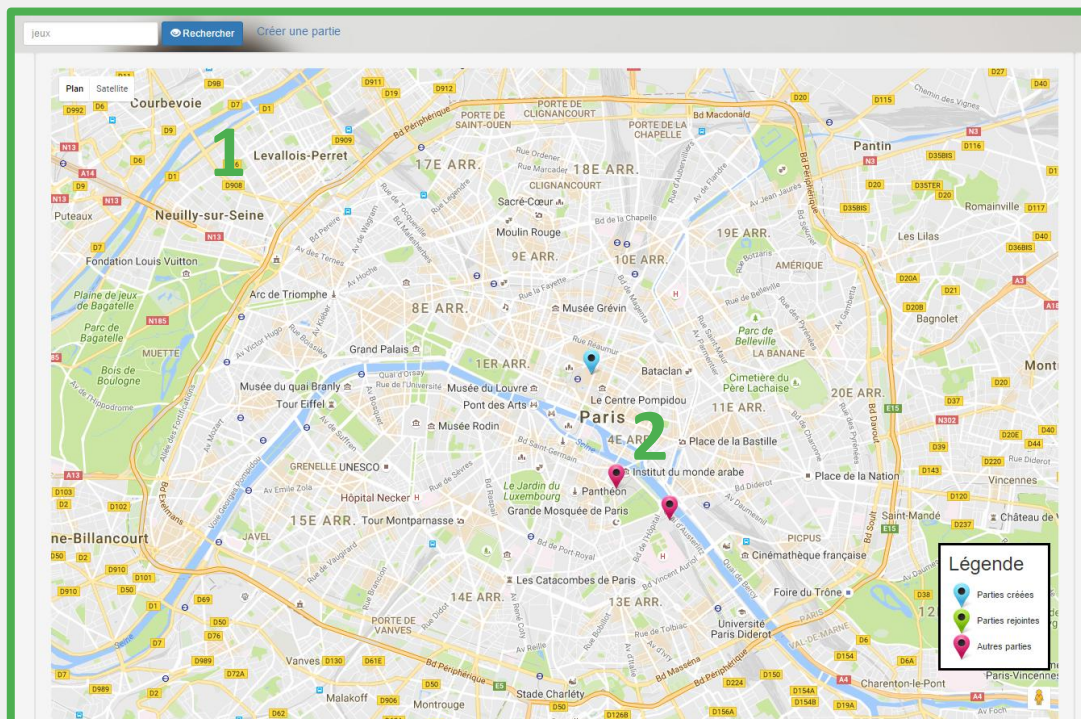
<h3>Parties pour ce jeu</h3> <p>tekken party in paris</p> <hr/> <p>Hébergé par: alerran 2016-11-12 19:00:00</p> <p>Détails</p>		<p>Nom : Tekken 7: Fated Retribution</p> <p>Note :</p> <p>Description : In addition to the ob... including Akuma from the Street Fi...</p> <p>Date de sortie : 2016-07-05 00:00</p> <p>Genres :</p> <p>Plateformes : Arcade, PC, Xbox On...</p> <p>Développeurs :</p> <p>Editteurs :</p> <p>Jeux similaires :</p> <p style="text-align: right;">Ajouter à mes jeux</p>
--	---	---

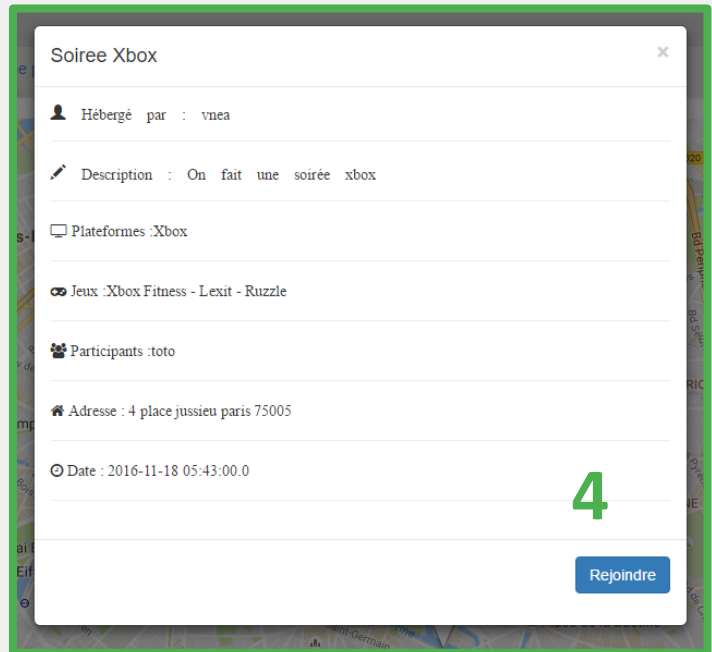
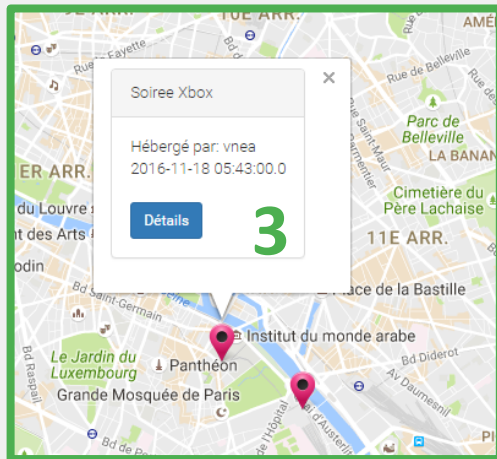
viii. *Comment rejoindre une partie autour de soi ?*

Prérequis : Être connecté (ii)

Remarque : On peut localiser les parties autour de soi grâce à l'API Google Maps, des marqueurs de couleur rose indiquent des parties hébergées par d'autres joueurs, des marqueurs de couleur bleue ses propres parties et des marqueurs de couleurs vertes les parties rejointes par le joueur.

Utilisation : Sur la page d'accueil, une carte Google Maps indique les parties en Île-de-France (1). Il clique sur le marqueur de la partie hébergée par Victor (2). Cela ouvre une fenêtre avec la date et l'adresse (3). Il clique sur le bouton « Détails ». Cela ouvre une autre fenêtre avec les détails de la partie (4). Il clique sur le bouton « Rejoindre ».



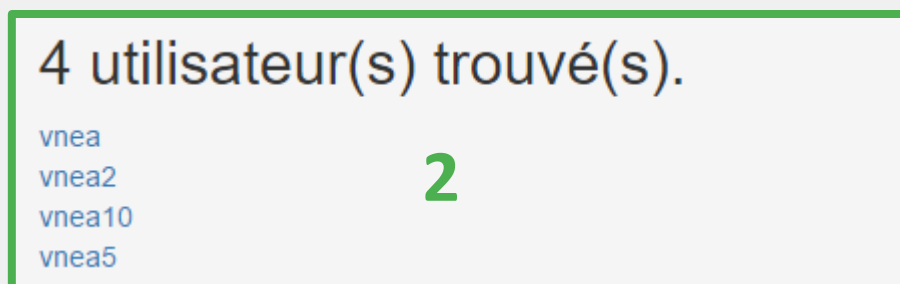
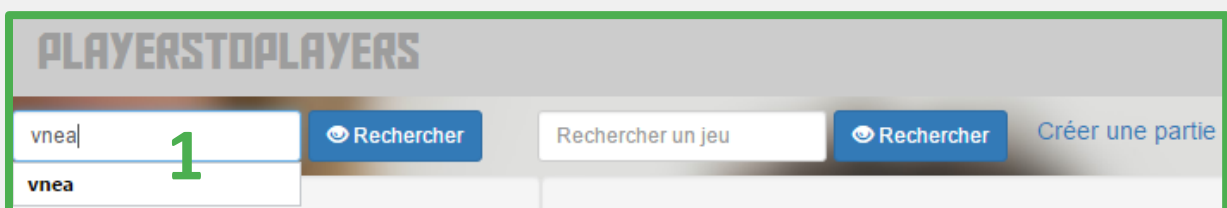


ix. Comment consulte-t-on le profil d'un autre joueur ?

Prérequis : Être connecté (ii)

Remarque : On peut consulter le profil d'un autre joueur, consulter ses informations et devenir ami avec lui (ix).

Utilisation : Sur la page d'accueil, Alexandre entre le pseudo du profil de Victor dans le formulaire de recherche de profil et clique sur le bouton « Voir profil » (1). Dans la page SearchUserResult qui recense les profils avec le même préfixe (2), il sélectionne le profil désiré. Dans la nouvelle page Profile, Alexandre a accès aux informations concernant le profil de Victor (3).



Nom d'utilisateur : vnea

Prénom

Vicooo

Nom

Neea

Nom

F

Age

23

Adresse

Code postal

91260

Numéro de téléphone

0606060606

E-mail

vico@hotmail.fr

Ajouter à ma liste d'amis

x. Comment devient-on ami avec un joueur ?

Prérequis : Être sur le profil d'un autre joueur (viii)

Remarque : On peut lier une amitié avec n'importe quel profil afin de rester en contact plus facilement pour consulter la liste des parties créées et rejointes par cet ami (voir image (1) de (xi))

Utilisation : Alexandre est sur le profil de Victor. Afin de l'ajouter en tant qu'ami, il clique sur le bouton « Ajouter à ma liste d'amis » (1). Un message de confirmation apparaît ensuite en haut de la fenêtre du profil de Victor afin d'indiquer que l'opération a bien eu lieu (2). Alexandre accède ensuite à sa liste d'amis depuis le menu « Mes infos » rubrique « Mes amis » (3). Dans la nouvelle page MyFriends, il a la confirmation qu'il est devenu ami avec Victor (4).

E-mail

vico@hotmail.fr

Ajouter à ma liste d'amis

Vous êtes désormais ami avec vnea.

Nom d'utilisateur : vnea

Prénom

Vicooo

Déconnexion

Mes Infos

Mon profil

Mes parties

Mes jeux

Mes plateformes

Mes genres

Mes amis

Mes amis

vnea

xi. Comment supprime-t-on un profil de sa liste d'ami ?

Prérequis : Être ami avec un joueur (ix) et être sur son profil (viii)

Remarques : On peut supprimer un ami de sa liste d'amis : ce résultat sera visible sur les deux profils de joueurs

Utilisation : Alexandre est sur le profil de Victor. Il clique sur le bouton « Supprimer de ma liste d'amis » (1). Il peut comme (ix) vérifier que Victor n'est plus dans liste d'amis.

The screenshot shows a user profile interface. At the top, there is an 'E-mail' field with the value 'vico@hotmail.fr'. A large green number '1' is overlaid on the right side of this field. Below the email field is a section titled 'Liste des parties'. This section contains two entries: 'Partie chez toto' and 'Soiree Play Xbox'. Each entry has a sub-section 'Hébergé par:' followed by a name and a timestamp. The first entry is 'toto' with timestamp '2016-11-15 02:53:00.0' and the second is 'vnea' with timestamp '2016-11-16 20:52:00.0'. Each entry has a blue 'Détails' button. At the bottom of the page, there is a button labeled 'Supprimer de ma liste d'amis'.

II) DEVELOPPEMENT DE L'APPLICATION

a. Choix technologiques et APIs

i. Hébergement Amazon Web Services

Notre choix d'hébergement pour notre application web Java s'est tourné vers Amazon Web Services (AWS), un service de cloud computing du groupe américain Amazon.

Pourquoi ne pas avoir choisi l'hébergement de Google, l'autre service proposé dans l'UE ?

On peut utiliser aussi bien utiliser gratuitement le service de Google que celui d'Amazon dans le cadre de notre projet. Néanmoins, concernant la persistance, il faut utiliser NoSQL du côté du Google App Engine, alors que AWS nous propose du SQL. Trois autres raisons nous ont poussés à opter pour la solution d'Amazon :

- Il nous a semblé que les bases de données NoSQL était plutôt adaptées pour traiter de grosses quantités de données, ce qui n'est pas notre cas a priori.
- Nous connaissons assez bien les bases de données SQL, choisir les bases de données NoSQL aurait peut être demandé un effort d'apprentissage de plus à fournir, ce qui semble un peu compliqué étant donné les délais et les autres projets à réaliser dans les autres UEs.
- Au tout début du projet, nous avons déjà modélisé les données avec des tables relationnelles, ce qui est parfaitement adapté pour notre application de réseau social.

Pourquoi ne pas avoir choisi un hébergement autre que ceux proposés dans l'UE ?

Nous avons effectué quelques recherches qui se sont révélées infructueuses : les solutions pour héberger des applications web Java sont en général payantes, et assez chères.

Quelles sont les fonctionnalités de AWS utilisées dans le développement de ce projet ?

AWS comprend plusieurs services destinés aux développeurs. Nous avons utilisé deux d'entre eux :

- AWS Elastic Beanstalk, qui est le service de déploiement et de gestion de notre application
- Amazon RDS, qui est le service de gestion de la base de données sur le cloud Amazon

ii. GiantBomb

GiantBomb est une des API que nous utilisons dans ce projet. Il s'agit d'une API REST qui consiste en une bibliothèque pour quasiment tous les jeux vidéo. En particulier, elle nous permet d'extraire des informations comme :

- les consoles sur lesquelles le jeu est paru
- des images de différents formats : icônes, petites, moyennes, grandes... que nous utilisons dans notre application
- les genres du jeu

Ci-dessous, un exemple de réponse XML de l'API après avoir recherché un jeu dans l'API. Néanmoins, le format des réponses que nous avons décidé de garder est le format JSON, que nous manipulons avec un peu plus d'aisance.

```
<?xml version="1.0" encoding="UTF-8" ?>
<game>
  <image>
    <icon_url>
      <![CDATA[ http://www.giantbomb.com/api/image/square_avatar/249745-zelda_adventure_box.jpg ]]>
    </icon_url>
    <medium_url>
      <![CDATA[ http://www.giantbomb.com/api/image/scale_medium/249745-zelda_adventure_box.jpg ]]>
    </medium_url>
    <screen_url>
      <![CDATA[ http://www.giantbomb.com/api/image/screen_medium/249745-zelda_adventure_box.jpg ]]>
    </screen_url>
    <small_url>
      <![CDATA[ http://www.giantbomb.com/api/image/scale_small/249745-zelda_adventure_box.jpg ]]>
    </small_url>
    <super_url>
      <![CDATA[ http://www.giantbomb.com/api/image/scale_large/249745-zelda_adventure_box.jpg ]]>
    </super_url>
    <thumb_url>
      <![CDATA[ http://www.giantbomb.com/api/image/scale_avatar/249745-zelda_adventure_box.jpg ]]>
    </thumb_url>
    <tiny_url>
      <![CDATA[ http://www.giantbomb.com/api/image/square_mini/249745-zelda_adventure_box.jpg ]]>
    </tiny_url>
  </image>
  <name>
    <![CDATA[ Zelda's Adventure ]]>
  </name>
  <platforms>
    <platform>
      <api_detail_url>
        <![CDATA[ http://www.giantbomb.com/api/platform/3045-27/ ]]>
      </api_detail_url>
      <id>27</id>
      <name>
        <![CDATA[ CD-i ]]>
      </name>
      <site_detail_url>
        <![CDATA[ http://www.giantbomb.com/cd-i/3045-27/ ]]>
      </site_detail_url>
      <abbreviation>
        <![CDATA[ CDI ]]>
      </abbreviation>
    </platform>
  </platforms>
</game>
```



Pourquoi avoir choisi cette API ? En existe-t-il d'autres ?

Nous nous sommes intéressés à deux API : IGDB, notre premier choix, puis GiantBomb. A titre de comparaison, ci-dessous, un tableau avec trois API, dont IGDB et GiantBomb, tiré du site de l'API IGDB.

	IGDB.com	GiantBomb	thegamesdb.net
Total unique games	+24,673	+48,000	+25,000
Allowed to use commercially	✓		
Requests per day	7,000 - 700,000	4,800	
Allowed to build competing product	✓		
SSL support	✓		✓
Protocol / Formats	JSON, REST	JSON, JSONP, XML, REST	XML, REST

Première raison qui nous a amené à choisir GiantBomb : le nombre de jeux présents dans leur base. Bien que les jeux en plus dans GiantBomb soient pour la plupart des jeux anciens, on a imaginé que certains joueurs pouvaient affectionner les jeux old-school. Ainsi, cela leur permettrait de trouver facilement d'autres personnes avec les mêmes goûts grâce à l'application.

Le nombre de requêtes n'étant pas déterminant pour le moment, ce n'est pas pénalisant pour nous. Néanmoins, ce sera un élément à prendre en compte pour un éventuel déploiement futur de l'application, hors du cadre de l'UE.

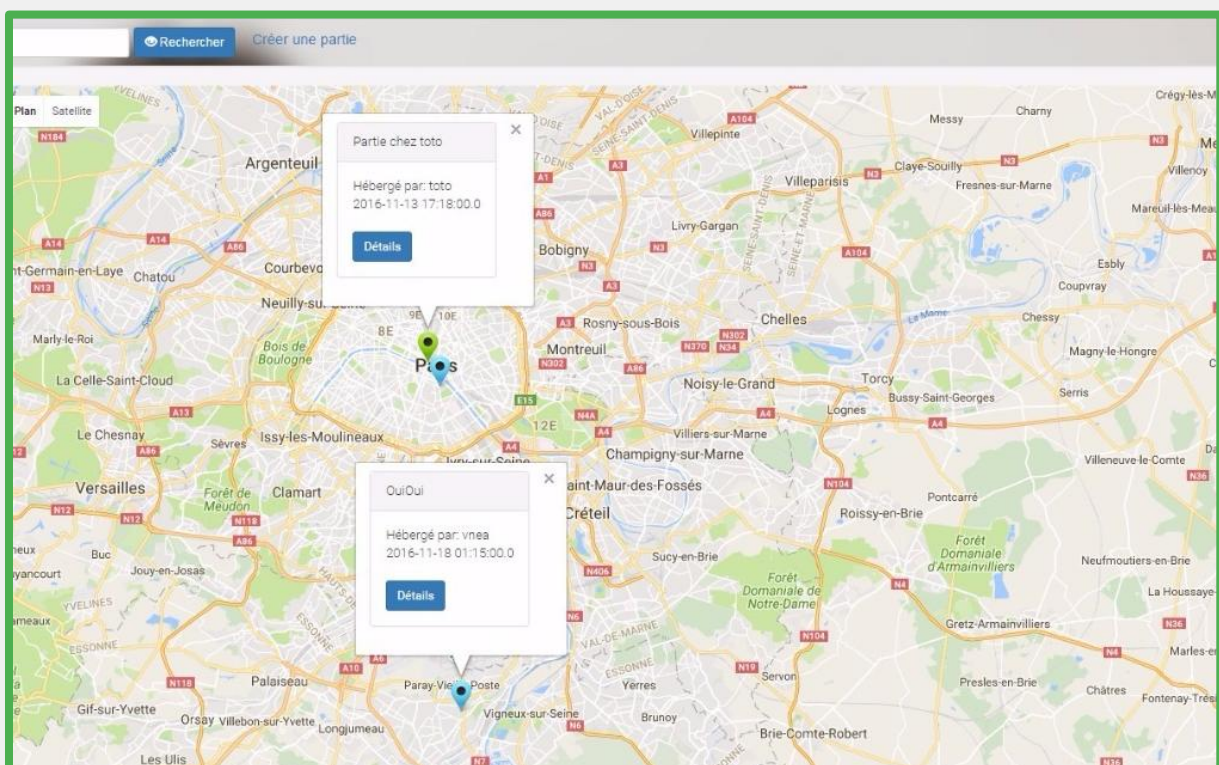
Enfin, un élément qui n'apparaît pas dans le tableau mais qui a été primordial pour nous, est que l'API GiantBomb a une documentation plus complète et claire de son utilisation. Après avoir essayé les deux, c'est cette raison qui nous a poussé à faire notre choix.

iii. Google Maps

Google Maps API est une API de Google, qui permet aux développeurs d'implanter des fonctionnalités de localisation dans leur application et en y intégrant l'outil Google Maps. Dans le cadre de notre projet, il répond tout d'abord à une des contraintes qui est l'utilisation dynamique d'une API. Ensuite, en tant que réseau social qui veut se faire réunir les joueurs, il est indispensable d'intégrer à notre application un outil qui leur permet de se localiser et de localiser les parties de jeux vidéo en cours ou en préparation autour d'eux.



Quelles sont les fonctionnalités du site qui implantent cette API ?



L'application fait des appels dynamiques à Google Maps lorsqu'il est question de chercher une partie ou d'en localiser une (image ci-dessus).

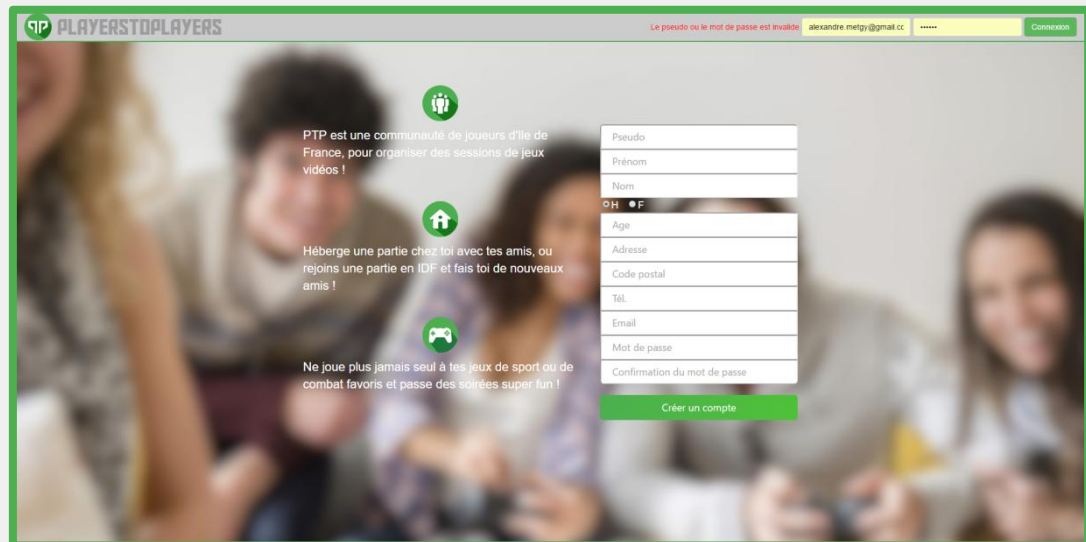
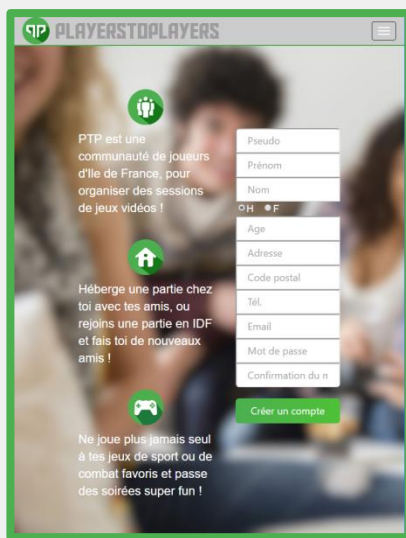
iv. Design

L'un des problèmes que nous avons rencontrés dans le développement de notre application a été l'apparence graphique. Bien que nous connaissions le fonctionnement des feuilles de style CSS, nous n'avons pas la fibre artistique. De plus, on voulait que notre site soit adapté à différentes tailles d'écran, sans casser le design à chaque redimensionnement. C'est pourquoi après plusieurs semaines de développement, nous avons décidé d'intégrer le framework Bootstrap à notre application.



Qu'est-ce que Bootstrap et comment a-t-il été utilisé ?

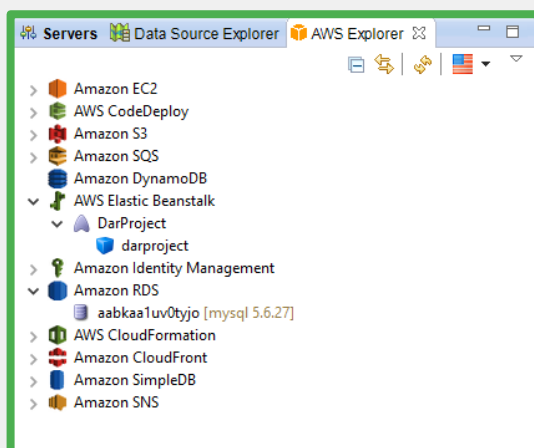
Bootstrap est un framework HTML, CSS et Javascript, qui permet en particulier de construire des applications web réactives qui s'adaptent aux différents formats d'écran : mobile, tablette, PC (voir figure ci-dessous). Il consiste en des fichiers CSS et Javascript qui contiennent des éléments de design, et que nous avons copiés au sein de notre projet. Nous avons aussi basée une partie de notre interface sur un template Bootstrap, en particulier pour le menu du haut. Cela nous a permis de nous concentrer davantage sur le fond de l'application et son architecture que sur la forme.



Adaptation du menu haut en fonction de la taille de l'écran

v. Outils et environnement de développement

Nous avons développé notre application avec l'EDI Eclipse, auquel nous avons ajouté les plugins nécessaires au développement et à la coopération.



Plugin AWS pour Eclipse

Ce plugin nous permet d'accéder aux services de AWS, en particulier les services Elastic Beanstalk et RDS, qui permettent de gérer, déployer et d'accéder à la base de données depuis Eclipse.

Plugin Git pour Eclipse

Ce plugin nous a permis de travailler sur Git depuis notre environnement de développement. Nous avons par contre remarqué que ce plugin était parfois capricieux et sources de bug, c'est pourquoi nous l'avons laissé au profit de la ligne de commande vers la fin de notre projet.

Trello

Aussi, concernant le travail en équipe, en plus de Git, nous avons utilisé l'outil Trello. C'est un outil particulièrement utilisé au sein de la méthode Agile, que des enseignants nous ont présenté dans des UEs de gestion de projet. Il nous a permis de suivre efficacement l'avancement du projet et la répartition des tâches.



Hibernate

Nous avons utilisé dans notre projet l'ORM Hibernate, qui nous permet de gérer les données de notre base sous forme d'objets et les utiliser avec des méthodes Java de haut-niveau. Nous allons parler plus en détail de ce framework dans la partie suivante consacrée aux détails d'implémentation.

b. Architecture

i. Organisation des données



Quelles données sont stockées dans notre base de données ?

Nous stockons dans notre base toutes les données relatives aux profils de joueurs, aux parties et aux liens d'amitié sous forme de tables relationnelles avec MySQL. A titre d'exemple, voici comment nous nous représentons les parties :

GameSession							
idSession	address	root_idPlayer	label	postCode	meetingDate	timestampDate	description
INT CP*	VARCHAR	INT CE*	VARCHAR	VARCHAR	DATETIME	DATETIME	VARCHAR

GameSession_players		GameSession_platforms		GameSession_games	
GameSession_idSession	players_idPlayer	GameSession_idSession	platforms	GameSession_idSession	games
INT FK	INT FK	INT FK	VARCHAR	INT FK	VARCHAR

*CP : clé primaire

*CE : clé étrangère



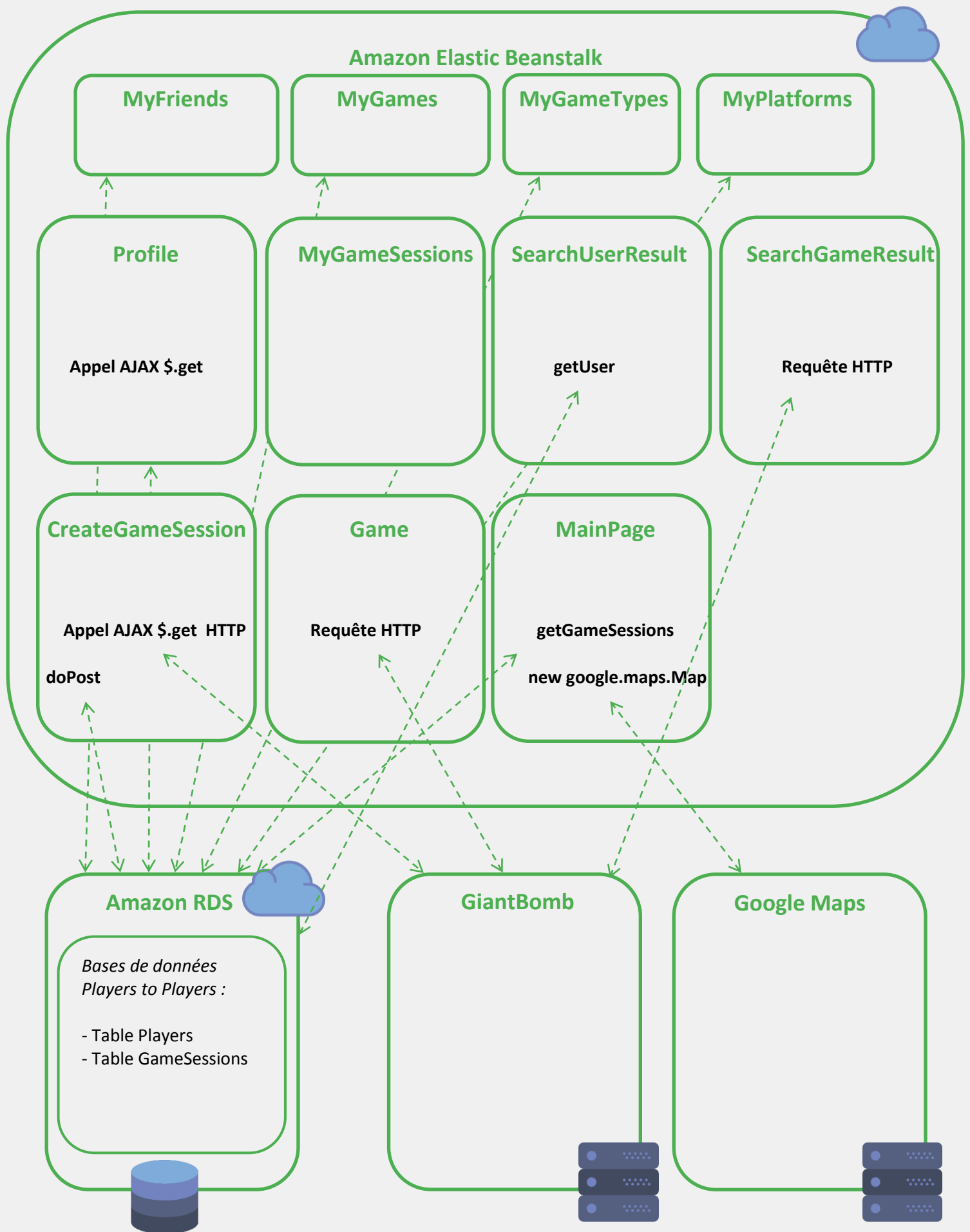
Quelles données sont stockées dans notre base de données ?

Nous avons fait le choix dans ce projet de ne pas utiliser l'API GiantBomb de manière statique, c'est-à-dire que nous faisons appel à chaque fois qu'il est question de chercher un jeu ou d'autres informations, sans les conserver dans notre propre base de données. Plusieurs raisons à cela :

- la taille de la base de GiantBomb est plutôt conséquente : environ 50.000 jeux, à cela il faut ajouter toutes les images pour les jeux, conservées en différentes tailles
 - l'API est plutôt réactive, nous n'avons pas noté de problème de lenteur en l'utilisant de manière dynamique
 - la base de donnée de l'API évolue constamment avec les sorties de jeux et de consoles.
- Nous préférons ne pas devoir mettre à jour notre base de jeux et de consoles tous les jours ou toutes les semaines par exemple, en risquant parfois de louper un nouveau jeu qui vient de sortir et qui est déjà disponible dans GiantBomb et pas dans notre base.

ii. Schéma d'application

En dessous, le schéma de notre application. Il lie les servlets, les API GiantBomb et Google Maps, ainsi que notre base de données.



c. Implémentation des servlets

Dans cette section, nous allons nous intéresser aux détails d'implémentation que nous jugeons utiles de rapporter, notamment lorsqu'il s'agit des communications avec les API externes et le traitement des données, et aussi quand ils répondent aux contraintes imposées dans l'UE.

i. Communication avec l'API GiantBomb

A quels moments fait-on appel à l'API GiantBomb ?

- Quand un joueur recherche un jeu et souhaite obtenir une liste des parties avec ce jeu. Dans ce cas, on récupère diverses informations liées au jeu et qui seront affichées dans l'application à titre informatif : images, description, consoles...
- L'API est aussi appelée quand un joueur crée une partie, pour obtenir les jeux qu'il souhaite ajouter dans la partie qu'il va héberger.

Comment sont faits ces appels ?

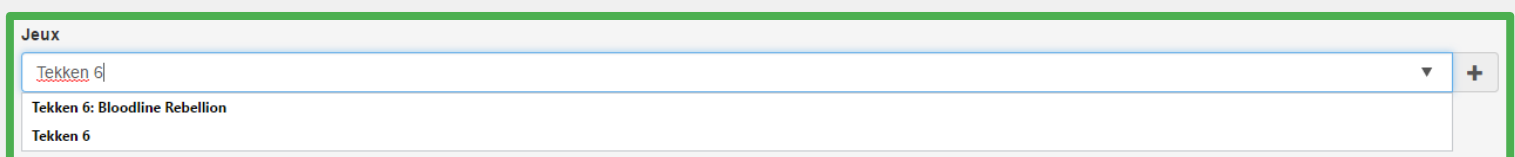
Nous avons une classe utilitaire qui contient des méthodes pour travailler avec l'API, GiantBombUtils. Par exemple, la méthode `getGamesAsJsonString(String gameName)`, qui renvoie les informations d'un jeu en format JSON. C'est cette méthode qui est appelée par exemple quand un joueur a recherché un jeu, depuis la servlet Game.

Code Java

```
public static String getGamesAsJsonString(String gameName) {
    return ResultRequester.resultRequest(
        // Début de l'URL, on précise que l'on recherche un jeu
        createRequestResourceUrl(GiantBombResource.GAMES) +
        // Paramètres de l'URL envoyés à l'API pour la recherche
        URLEncoder.encode("&filter=name:" + gameName +
"&sort=date_added:desc&field_list=name,original_release_date,image"),
        USER_AGENT,
        ACCEPT
    );
}
```

Remarque : La méthode `resultRequest` de la classe `ResultRequester` permet de lancer la requête et d'obtenir la réponse de l'API.

Dans le formulaire de création de partie, nous effectuons des appels AJAX afin de récupérer dynamiquement les jeux.



The screenshot shows a web form with the title "Jeux". It features a search input field containing the text "Tekken 6". Below the input field is a dropdown menu that is open, showing two options: "Tekken 6: Bloodline Rebellion" and "Tekken 6". The entire form is enclosed in a light blue border.

Code Javascript 1

```
$(CGS.games.elements.inputSearch).keyup(function() {
    clearTimeout(timerGames);
    var value = $(this).val().trim();

    ...

    // Input has a value, we do search
    if (value.length) {
        timerGames = setTimeout(function() {
            // Remove all games found
            $(CGS.games.elements.datalist).empty();

            // Show that we are searching for games
            CGS.games.showSearchingText();

            // Call server API to get games
            GiantBomb.getGames(
                value,
                CGS.games.handlers.success,
                CGS.games.handlers.error
            );
        }, CGS.timerKeyUp);
    }
}
```

Code Javascript 2

```
getGames: function(gameName, handlerSuccess, handlerError) {
    $.get({
        url : GiantBomb.url,
        data : {
            resource: GiantBomb.resource.games,
            game : gameName.trim()
        },
        success : function(data, textStatus, jqXHR) {
            handlerSuccess(data, textStatus, jqXHR);
        },
        error : function(jqXHR, textStatus, errorThrown) {
            handlerError(jqXHR, textStatus, errorThrown);
        }
    });
}
```

Dans le code 1, nous appelons une fonction qui se charge de récupérer des jeux quand l'utilisateur tape dans l'input des jeux (événement keyup). Plus précisément, cette fonction appelle la méthode get de jQuery, qui permet de récupérer des données en utilisant une requête HTTP, en passant par la méthode getGames du code 2.

ii. Implantation de l'API Google Maps

Nous implantons une carte Google Maps dans la servlet MainPage, la page d'accueil de l'application. Nous ajoutons dans cette carte des marqueurs de couleurs qui correspondent aux parties.

Code Javascript 1

```
new google.maps.Map($("#gmaps")[0], mapOptions);
```

Code Java et Javascript 2

```
scriptMarkers += "$(document).ready(function() {\n";
// Create markers for each game session
for (GameSession gameSession : gameSessions) {
    Location location = GoogleMapsUtils.geocode(gameSession.getAddress() + " " +
gameSession.getPostCode());
    if (location != null) {
        // Marker
        String makerVar = "marker" + gameSession.getIdSession();
        scriptMarkers += "var " + makerVar + " = new google.maps.Marker({\n";
        scriptMarkers += "position: " + location.toJsonApprox() + ",\n";
        scriptMarkers += "map: " + GMAPS + ",\n";

        // Normal case
        String icon = IMG_MARKER_OTHERS_GS;

        // Created game session
        if (gameSession.getRoot().getUsername().equals(username)) {
            icon = IMG_MARKER_CREATED_GS;
        }
        // Joined game session
        else {
            List<Player> players = gameSession.getPlayers();
            for (Player player : players) {
                // Joined game session
                if (player.getUsername().equals(username)) {
                    icon = IMG_MARKER_JOINED_GS;
                    break;
                }
            }
        }

        ...

        // Add open tooltip when clicking on marker
        scriptMarkers += "google.maps.event.addListener(" + makerVar + ",
\"click\", function() {\n";
        scriptMarkers += tooltipVar + ".open(" + GMAPS + ", " + makerVar +
");\n";
        scriptMarkers += "});\n";
    }
}
```

Dans le code 1, nous créons la carte que nous plaçons dans la div d'id « gmaps ».

Dans le code 2, pour chaque partie précédemment récupérée, nous récupérons le code de géolocalisation à partir de l'adresse. On donne une icône au marqueur selon que c'est une partie créée (Created game session), une partie rejointe (Joined game session) ou une autre partie non-rejointe (Normal case). Enfin, nous créons un listener qui permettra d'ouvrir une info bulle lorsque l'on clique sur le marqueur.

iii. Implantation de Hibernate

Les classes Player et GameSession représentent les joueurs et les parties. Nous utilisons le design pattern DAO (Data Access Object) pour gérer les données des joueurs et des parties, avec les classes GameSessionDaoImpl et PlayerDaoImpl qui implémentent respectivement les interfaces GameSessionDao et PlayerDao.

Code Java 1

```
@Entity
@Table(name="Players")
public class Player {
    public Player(){}

    // Properties
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Integer idPlayer;

    @Basic(optional=false)
    @Column(unique = true)
    private String email;
    @Basic(optional=false)
    @Column(unique = true)
    private String username;

    ...

    @ElementCollection
    @LazyCollection(LazyCollectionOption.FALSE)
    private List<String> gamesType = new ArrayList<>();

    @ElementCollection
    @LazyCollection(LazyCollectionOption.FALSE)
    private List<String> games = new ArrayList<>();

    ...
}
```

Code Java 2

```
public class PlayerDaoImpl implements PlayerDao {

    public PlayerDaoImpl() {
    }

    @Override
    public void insert(Player p) {
        Session s = HibernateUtils.getSession();
        Transaction t = s.beginTransaction();

        s.save(p);
        t.commit();
        s.close();
    }

    ...
}
```

Le code 1 contient un aperçu de la classe Player qui représente un joueur dans le code Java. Le code 2 contient un aperçu du DAO de Player, avec la méthode insert. On fait appel dans le DAO à Hibernate afin de manipuler les données comme des objets : nous créons une session Hibernate puis nous enregistrons le nouveau joueur dans la base de données.