



Documentation technique

On The Road A Game

- 1) Suivi et informations concernant ce document
- 2) Analyse du projet
- 3) Avancement du projet
- 4) Technologies web
- 5) Base de données

Sommaire :

I.	Suivi et informations concernant ce document.....	4
1.	Présentation	4
2.	Diffusion du document	4
3.	Historique des modifications	4
4.	Glossaire	5
5.	Documents applicables	5
6.	Document de référence	5
II.	Analyse du projet.....	6
7.	Les enjeux	6
8.	Les objectifs	6
9.	Les moyens	6
10.	Les rôles et les responsabilités de chacun	7
11.	Analyse des risques	7
12.	Technologies de partage de fichiers	7
13.	Technologies de communication	8
14.	Backlog	8
III.	Avancement du projet	9
15.	Tableau d'avancement des tâches	9
16.	GANTT	11
IV.	Technologies web.....	12
17.	Technologies utilisées	12
18.	Hébergement	12
19.	Convention de nommage	12
20.	Chemin utilisateur chef ou membre	15
21.	Chemin administrateur	15
22.	Chemin visiteur	15
23.	Arborescence générale	16
24.	Authentification	16
25.	Formulaire d'inscription	16
26.	Gestion des équipes	17
27.	Gestion paiement en ligne	17
28.	Cas d'utilisation UML	18
V.	Base de données.....	23
29.	Technologies utilisées	23
30.	Structure générale	23

31.	Table Admin	23
32.	Table User	24
33.	Table Session	24
34.	Table Defi	24
35.	Table team	25
36.	Table Article	25
37.	MCD	26
38.	Sources	26

Table des figures :

Figure 1, analyse des risques.....	7
Figure 2, backlog.....	8
Figure 3, GANTT dates.....	11
Figure 4, GANTT graphique.....	11
Figure 5, arborescence.....	16
Figure 7, UML création compte user.....	18
Figure 6, UML administrateur général.....	18
Figure 8, UML envoyer preuve défi chef team.....	19
Figure 9, UML chef team general.....	19
Figure 10, UML gestion administrateur.....	20
Figure 11, UML ajouter membre chef team.....	20
Figure 12, UML utilisateur general.....	21
Figure 13, UML utilisateur.....	22
Figure 14, MCD.....	26

I. Suivi et informations concernant ce document

1. Présentation

Depuis plus de sept ans, l'entreprise On The Road A Game organise des voyages mêlant mystères, jeux, aventures et rencontres. Les participants découvrent au dernier moment leur destination, chaque équipe est de plus assurée de terminer l'aventure, car il n'y pas d'élimination possible. Les participants recevront également une somme d'argent qu'ils devront le moins dépenser possible, la somme restante sera redistribuée à des associations caritatives.

Cependant la crise sanitaire du COVID-19 ayant rendu inenvisageable les voyages à l'étranger. L'entreprise doit pouvoir trouver des solutions afin de poursuivre son développement malgré la situation. Il faut donc trouver de nouvelles sources de revenus et impliquer les anciens et futurs voyageurs dans une nouvelle forme d'aventures.

2. Diffusion du document

Diffusion	Statut	Nom
	Lecture	GUIBOUD RIBAUD Arnaud
	Présentation	///

3. Historique des modifications

Version	Pages	Description de la modification - Auteur	Date
0.1	1-12	Mise en page, titres, sous titres, structure générale	2021/01/04
0.5	8-28	Wireframes, backlog, avancement projet, analyse des risques	2021/01/09
0.7	1-32	Relecture, modification wireframes, gestion des équipes, tables base de données	2021/01/04

4. **Glossaire**

Abréviation	Signification
Général	
OTRAG	On The Road A Game.
Technologies Web	
FrameWork	Environnement de travail facilitant le développement d'une solution technique.
Base de données	
Table	Ensemble homogène de données agencées sous forme de tableaux.
Infrastructures et systèmes d'informations	
PacketTracer	Logiciel développé par Cisco permettant de représenter graphiquement une installation réseau.

5. **Documents applicables**

Description	Identification
Analyse_des_risques.xlsx	REF [0]
ArborescenceGenral.pptx	REF [1]
Backlog_V2.xlsx	REF [2]
Fonctionnalités.docx	REF [3]

6. **Document de référence**

Description	Identification
PROJET UF B2 2020 2021 - OnTheRoadAGame vf2.pdf	REF [0]

II. Analyse du projet

7. Les enjeux

Ce projet permettra d'évaluer nos compétences acquises durant nos années scolaires en Bachelor 1 et Bachelor 2. Ce projet montre nos capacités à travailler en groupe et nous adapter aux méthodes de travaux de chacun. Notre réussite dépend des solutions envisagées pour permettre à l'entreprise On The Road A Game, de continuer son développement, malgré la crise sanitaire rendant tout voyage très compliqué.

8. Les objectifs

Les objectifs de ce projet sont multiples. En effet nous devons tout d'abord concevoir une infrastructure système et réseau permettant de soutenir les besoins web, applicatifs et le stockage des données. La partie Frontend du site web doit présenter le concept de OTRAG, permettre l'inscription à une session, afficher le classement général en temps réel ainsi que des articles d'actualités mis à jour automatiquement. L'espace jeu comprend un espace de gestion de ses données personnelles, une liste des défis et leur présentation, un système d'envoi des preuves de la réalisation des défis et un espace informant les équipes de la validation du défis ou non. Depuis le site, les administrateurs pourront également gérer les sessions et les paramétrer, créer les défis à relever et également valider la réussite du défis grâce aux preuves envoyées par le participant.

9. Les moyens

Chaque acteur du projet est équipé de son propre matériel, à savoir au minimum un ordinateur par personne ainsi qu'une connexion à internet. Nous avons à notre disposition de nombreux outils fournis par Microsoft tel que la suite Office 365 et les systèmes d'exploitation Windows 10 Education. Nous pouvons également demander de l'aide à nos intervenants, joignables par mail ou durant les quatre heures consacrées au projet par semaine en cas de difficultés.

10. Les rôles et les responsabilités de chacun

Chacun des membres du projet est responsable de son travail et doit en assumer les conséquences en cas de retard ou de perte de celui-ci. Malgré tout, si l'un des acteurs éprouve des difficultés, le reste de l'équipe se doit de l'aider et de lui expliquer ce qu'il est possible de faire pour résoudre le problème rencontré.

11. Analyse des risques

N°	Liste des risques	Effet sur les objectifs	Impact	Probabilité	Détection	Criticité	Mesures préventives	Mesures curatives
1	Dépassement des délais	Retard de livraison	4	2	4	32	Organisation et répartition des tâches cohérente, demander du temps supplémentaire ou augmenter les effectifs pour tenir les délais.	Livraison retardée ou non complète
2	Attentat terroriste	Retard de livraison, arrêt du projet	5	1	5	25	Rien	Livraison retardée voir annulée
3	Risque épidémique	Retard de livraison	1	1	5	5	Télé-travail possible pour tous les employés	peu de perturbations puisqu'instaurer le travail à distance en informatique n'est pas très compliqué
4	Mauvaise répartition des tâches entre les membres du projet	Retard de livraison	3	2	3	18	Utilisation de logiciels comme GANTT et GIT pour avoir un suivi du travail de chacun et	Livraison retardée ou non complète
5	Manque de connaissances et d'expérience	Retard de livraison, certaines livrables peuvent être non rendues ou fausses	3	3	2	18	Apprendre uniquement des choses qui nous serviront aux tâches que chaque membre doit effectuer	Livraison retardée ou non complète
6	Livrable rendue en retard, voir non rendue par un des membres du projet (ou plusieurs)	Retard de livraison	3	3	2	18	Utilisation de GANTT pour anticiper les retards et prendre des mesures	Livraison retardée ou non complète
7	Ordinateur en panne	Retard de livraison	2	1	5	10	Chaque membre du projet doit posséder deux ordinateurs, ou nous devons avoir un ordinateur de secours en plus	Livraison retardée ou non complète
8	Effacement des données	Retard de livraison	4	1	5	20	Le travail doit être stocké dans plusieurs disques et en ligne, d'où l'intérêt d'utiliser git	Livraison retardée ou non complète
9	Piratage informatique	Livraison retardée	4	1	3	12	Utilisation de Virtual Machines et de plusieurs OS, Linux étant moins facilement "hackable" que Windows, avoir des ordinateurs linux de secours peut permettre de continuer à travailler en cas de piratage	Livraison retardée ou non complète voir impossible si les ordinateurs sont inutilisables
10	Connexion mal orientée	Livraison retardée	3	2	1	6	S'assurer que nous répondons bien aux attentes du prestataire, notamment avec une documentation technique complète	Livraison retardée ou non complète, voir non en accord avec la demande du prestataire
11	Virtual Machines inopérantes (réseaux)	Livraison retardée	4	1	4	16	Faire régulièrement des sauvegardes des VM et s'assurer qu'il est d'autres VM identiques sur plusieurs ordinateurs	Livraison retardée ou non complète
12	Ajout de fonctionnalités	Livraison retardée	2	3	4	24	Sortir une première version sans les modifications pour mettre en place une mise à jour pour ajouter ces fonctionnalités	Livraison retardée ou non complète

Figure 1, analyse des risques

12. Technologies de partage de fichiers

Grâce aux cours Git de l'année précédente, nous avons décidé en toute logique d'utiliser cet outil pour partager nos fichiers et les sauvegarder. Git a l'avantage d'être gratuit, facile d'utilisation et fiable. Git est également utilisé par de nombreuses personnes à travers le monde, ainsi que par de grands groupes comme Google, Microsoft ou encore Amazon. Le lien de notre Git est le suivant :

https://github.com/carolusDev/ProjetUF2020_2021

13. Technologies de communication

L'utilisation de Teams, développé par Microsoft, a été un choix évident. Ce logiciel de communication nous a permis de nous entraider, de nous tenir informé de l'avancée du projet et de travailler simultanément à plusieurs, comme pour la réalisation de la présentation par exemple.

14. Backlog

User story	User story créée	Feature	importance	difficulté	Statut	Assignée au sprint
avoir une interface pratique	Non	Contenu	100	50	A faire	3
espace de jeu	Non	Compte	100	80	A faire	4
le site doit communiquer avec un serveur	Non	?	100	60	A faire	3
l'utilisateur doit arriver sur une page publique d'accueil, qu'il soit connecté ou non	Non	Contenu	100	40	A faire	3
l'utilisateur doit pouvoir accéder à une description du jeu et du site	Non	Contenu	70	40	A faire	5
l'utilisateur doit pouvoir voir le classement général	Non	Contenu	60	70	A faire	5
l'utilisateur doit pouvoir voir un défi type afin de se préparer à ceux qu'il devra faire durant l'aventure	Non	Contenu	60	60	A faire	4
l'utilisateur doit pouvoir s'inscrire et/ou se connecter de partout à tout moment	Non	Compte	100	50	A faire	4
l'utilisateur doit pouvoir payer sa participation	Non	Compte	100	100	A faire	6
l'utilisateur doit pouvoir accéder à son espace	Non	Compte	100	50	A faire	4
l'utilisateur doit pouvoir déposer les vidéos de ses défis réalisés	Non	Compte	90	70	A faire	6
le client doit pouvoir voir toutes ses souscriptions en cours ou passées	Non	Compte	70	80	A faire	5
l'utilisateur doit pouvoir accéder et gérer ses informations personnelles	Non	Compte	100	50	A faire	4
l'utilisateur doit pouvoir suivre la validation de ses défis en temps réel, et le total de ses points	Non	Compte	100	70	A faire	6
les administrateurs doivent pouvoir accéder à un espace qui leur est propre	Non	Compte admin	100	50	A faire	4
les administrateurs doivent pouvoir suivre les défis et les valider, avec l'attribution de points qui convient	Non	Compte admin	100	70	A faire	6
les administrateurs doivent pouvoir gérer les utilisateurs et leurs accès, par exemple aux différents défis	Non	Compte admin	100	80	A faire	5

Figure 2, backlog

III. Avancement du projet

15. Tableau d'avancement des tâches

Tâches	Avancement	BERENGUER Jonathan	CHARLEMAGNE Clément	LEVY DE MAREUIL Hélie
Documentation				
Document	95%			
Base de données				
Document d'analyse	95%			
Mise en place	0%			
Technologies web				
Document d'analyse	95%			
Espace public				
Accueil	0%			
Présentation du site	0%			
Classement général	0%			
Défi démo	0%			
Formulaire d'inscription	0%			
Paieement en ligne	0%			
Connexion	0%			
Espace client/utilisateur				
Dépose des vidéos de défis	0%			
Listing des défis souscrits et description	0%			
Gestion des données et coordonnées	0%			
Suivi de validation des défis	0%			
Espace administrateurs				
Suivi des dépôts & validation	0%			
Gestion des utilisateurs (CRUD) et des accès	0%			
Attribution des défis et des points	0%			
Infrastructure et systèmes d'informations				
Virtualiser et faire communiquer des serveurs	0%			
Sécuriser l'ensemble de l'infrastructure et du réseau	0%			

Permettre la résolution des noms	0%			
Permettre à tout utilisateur d'accéder au site et aux applicatifs	0%			
Permettre à tout utilisateur authentifié d'accéder à ses données	0%			
Déployer une politique de sécurité cohérente aux besoins	0%			

16. GANTT

Spint	Date de début	durée	date de fin
0	11/18/2020	14	12/2/2020
1	12/2/2020	14	12/16/2020
2	12/16/2020	14	12/30/2020
3	12/30/2020	14	1/13/2021
4	1/13/2021	14	1/27/2021
5	1/27/2021	14	2/10/2021
6	2/10/2021	14	2/24/2021
7	2/24/2021	14	3/10/2021
8	3/10/2021	14	3/24/2021
9	3/24/2021	14	4/7/2021
10	4/7/2021	14	4/21/2021
11	4/21/2021	14	5/5/2021
12	5/5/2021	14	5/19/2021

Figure 3, GANTT dates

Dates clés sprints

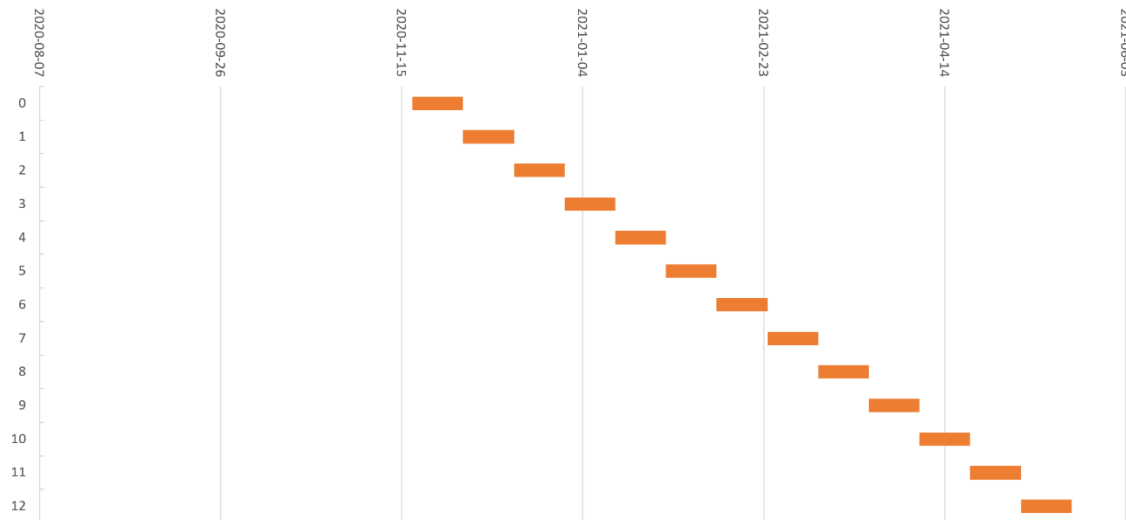


Figure 4, GANTT graphique

IV. Technologies web

17. Technologies utilisées

Pour réaliser ce projet nous avons décidé d'utiliser le Framework Laravel. L'utilisation de ce Framework nous semblait évident étant donné que nous l'avions vu en cours. Ce Framework reste le plus simple d'utilisation pour réaliser la partie back demandée par ce projet. Nous avons choisi d'utiliser Xampp comme technologie de serveur local.

18. Hébergement

Notre site sera hébergé sous Microsoft Azure.

19. Convention de nommage

Les contrôleurs

Les contrôleurs doivent être au singulier, sans espacement entre les mots et se terminent par Controller. C'est ce qu'on appelle le format PascalCase.

Exemples : UserController / HomeController

Les méthodes dans les contrôleurs

Les méthodes devront suivre la règle du camelCase, c'est à dire avec la première lettre en minuscule et les mots suivants avec une majuscule.

Exemples : index() / create()

Les tables de la base de données

Le nom des tables doit être en minuscule et au pluriel. Si elle est composée de plusieurs mots, il faut les séparer par des _

Exemples : users / posts / picture

Les tables pivot

Tout comme le nom des tables, les tables pivot doivent être en minuscule mais par contre au singulier. Elle reprendra le nom des 2

tables à lier avec un `_` pour séparation. Il conviendra de les mettre par ordre alphabétique.

Exemples : `category_post` / `role_user`

Le nommage des colonnes

Les noms doivent être en minuscule et séparés par un `_`. Il n'est pas nécessaire de rappeler le nom de la table dans le nom du champ.

Exemples : `id` / `name` / `content` / `created_at`

Les clés primaires des tables

Par convention, il faut les nommer simplement `id`

Les clés étrangères

Ces clés, également appelées en anglais `foreign keys` doivent être formées du nom de la table en référence suivant de `_id`

Exemples : `user_id` (fait référence à l'id de la table `users`) / `post_id` (fait référence à l'id de la table `posts`)

Les variables PHP

Celles-ci doivent être écrites en `camelCase`, c'est à dire avec le premier caractère en minuscule et les mots suivants avec une majuscule.

Une variable sera notée au pluriel si le résultat attendu contiendra plusieurs éléments sinon on la mettra au singulier.

Exemples : `$users` / `$countPosts`

Les modèles

Les modèles commenceront par une majuscule, sans espace et une majuscule pour chaque mot. C'est au format `PascalCase`.

Exemples : `User` / `Post`

Les propriétés des modèles

Les propriétés seront notées en `snake_case`, en minuscule. C'est la même convention de nommage que les noms des colonnes de la base de données.

Exemples : `$this->name` / `$this->params`

Les méthodes dans les modèles

Dans les modèles, les méthodes doivent être appelées en camelCase

Exemples : `getUser()` / `getName()`

Les relations entre les modèles

Pour les relations un vers un (one to one)

Elles doivent être au singulier et suivre les mêmes conventions d'appellation que les modèles normaux (camelCase, mais avec la première lettre en minuscule)

Exemples : `public function phone()` / `public function post()`

Pour les relations un vers plusieurs (one to many)

Elles doivent être également en camelCase mais au pluriel

Exemples : `public function posts()` / `public function users()`

Les relations polymorphiques

```
public function category()
{
    return $this->morphMany('App\Category', 'categoryable');
}
```

Laravel considère que les champs `categoryable_id` et `categoryable_type` existent

Les traits

Le nom d'un trait devra être un adjectif. Les fichiers seront placés dans le dossier `app\Traits`

Exemples : `Notifiable` / `Dispatchable`

Le nom des fichiers sous Blade

Ils devront être au format `snake_case`

Exemples : `index.blade.php` / `post_users.blade.php`

20. Chemin utilisateur chef ou membre

Depuis l'accueil affichant des informations sur son équipe, l'utilisateur connecté pourra accéder aux pages suivantes :

- ➔ Actualités
- ➔ Classement
- ➔ Mes défis
- ➔ Mon compte

Me déconnecter

21. Chemin administrateur

Le compte administrateur étant uniquement conçu pour superviser le jeu, de nombreuses fonctionnalités secondaires, comme le classement ne sont plus accessibles. Cependant des outils de gestion ont été ajoutés. Une fois connecté l'administrateur pourra accéder aux pages suivantes :

- ➔ Gestion générale

Gestion défis

Création sessions

Création défis

- ➔ Gestion équipes
- ➔ Gestion comptes

22. Chemin visiteur

Le visiteur depuis l'accueil où sont affichées des informations sur l'entreprise OTRAG ainsi que son jeu, pourra avoir accès aux pages suivantes :

- ➔ Classement
- ➔ Actualités
- ➔ Me connecter
- ➔ Participer à l'aventure

23. Arborescence générale

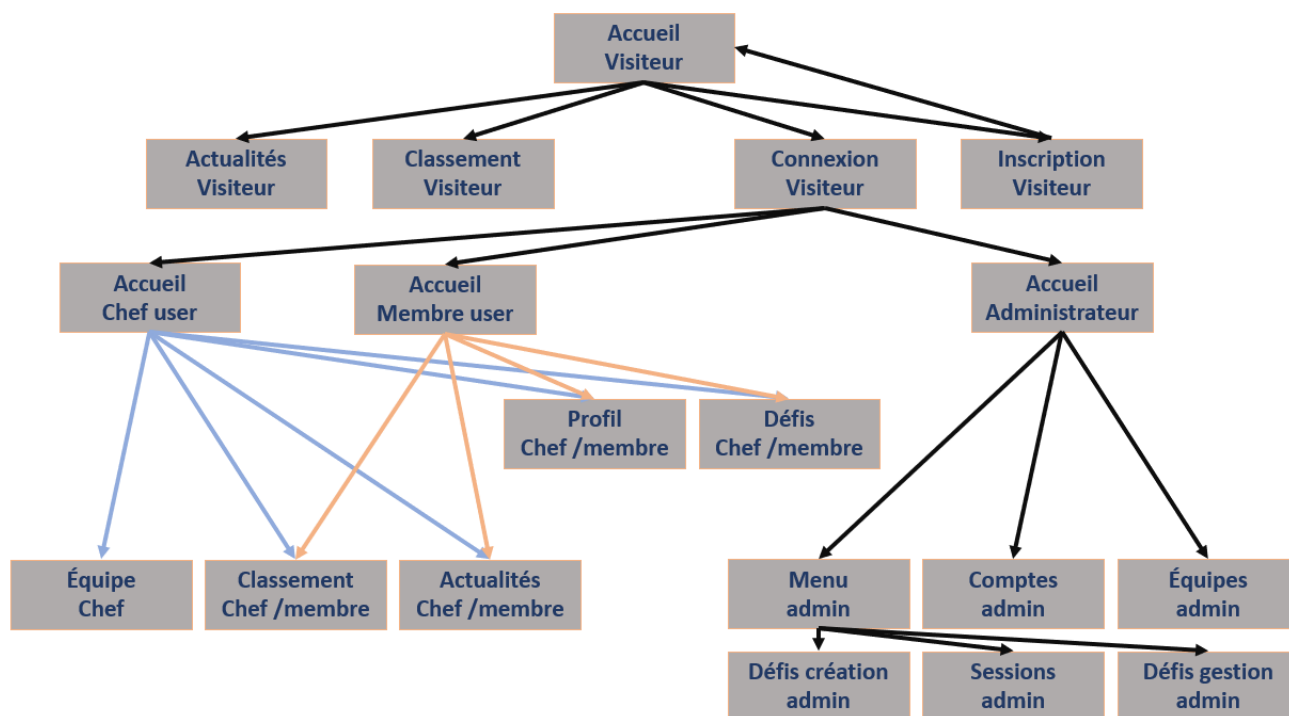


Figure 5, arborescence

24. Authentification

Pour qu'un utilisateur puisse s'authentifier, il lui suffit de se rendre sur la page de connexion et d'entrer son email ainsi que son mot de passe. Il sera redirigé automatiquement vers l'accueil chef d'équipe, membre d'équipe ou administrateur en fonction de son statut.

25. Formulaire d'inscription

Le formulaire d'inscription permet à une personne de devenir candidat à une session de jeu OTRAG. Les champs à remplir sont les suivants :

- ➔ Civilité
- ➔ Nom
- ➔ Prénom
- ➔ Mot de passe
- ➔ Date de naissance
- ➔ Pays

- ➔ Code postal
- ➔ Ville
- ➔ Numéro de téléphone
- ➔ Adresse email

26. Gestion des équipes

Chaque joueur a pour obligation d'appartenir à une équipe, celle-ci peut être composée de 1 à 4 joueurs. Un joueur peut donc être simplement membre de l'équipe ou chef d'équipe, un joueur seul sera automatiquement considéré comme chef d'équipe.

Pour devenir chef d'équipe il faut cocher la case « être chef d'équipe » au moment de l'inscription, si une famille souhaite jouer ensemble ils devront se mettre d'accord au préalable. Mais si un joueur est seul il peut à la fois accueillir des joueurs inconnus dans son équipe en se proposant chef d'équipe, ou bien accepter l'invitation d'un autre joueur. Le chef d'équipe a accès à une page supplémentaire sur son espace personnel, cette page lui permet de superviser son équipe, ajouter un membre et les voir dans une liste. L'administration a la possibilité de supprimer une équipe.

27. Gestion paiement en ligne

Pour notre solution de paiement en ligne, nous avons décidé d'utiliser une API Paypal. A ce jour, Paypal est le seul système de paiement vraiment international qui fonctionne. Il est possible, depuis la France, de vendre à un indien, un sénégalais, un canadien ou un japonais sans installer de module autre que Paypal. Indispensable donc pour ceux qui font de l'export. Paypal est simple à installer (pour les marchands) et à utiliser (pour les clients). Les plugins sont utilisés sur des milliers de sites, avec toutes les versions de presque n'importe quel solution e-commerce. Peu importe votre choix de plateforme ([Open SSource ou SaaS](#)) il y a de fortes chances pour que la solution soit disponible.

28. Cas d'utilisation UML

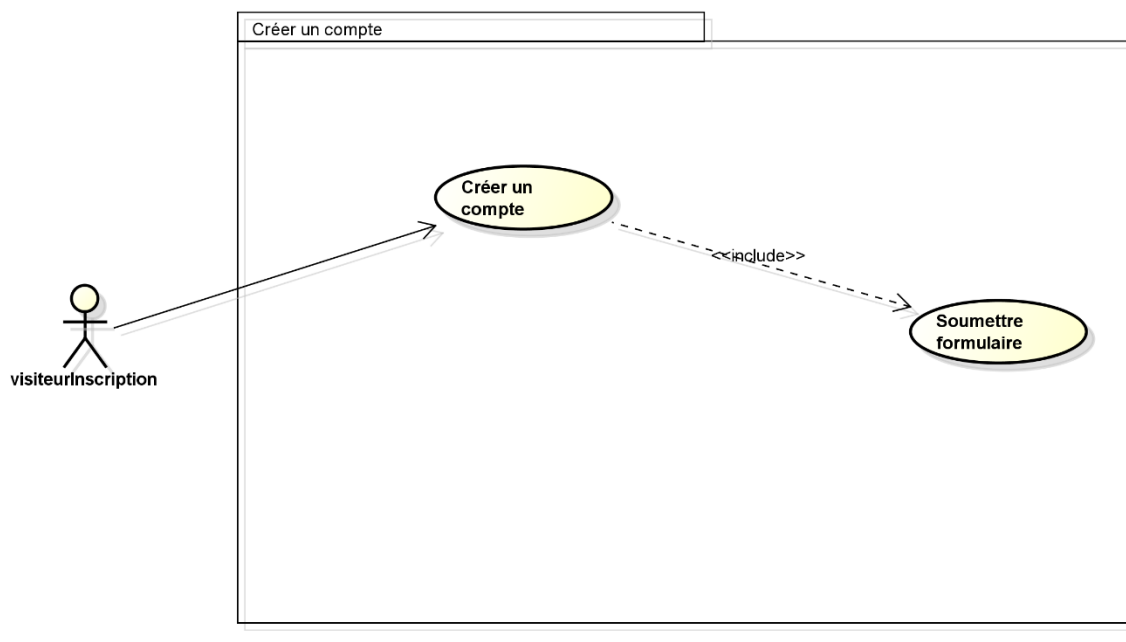


Figure 7, UML création compte user

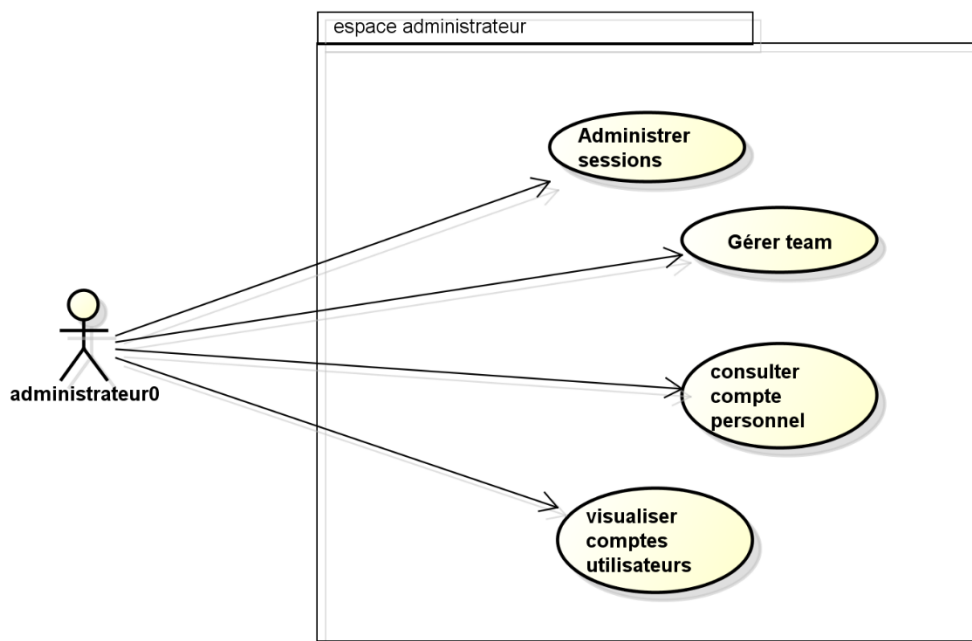


Figure 6, UML administrateur général

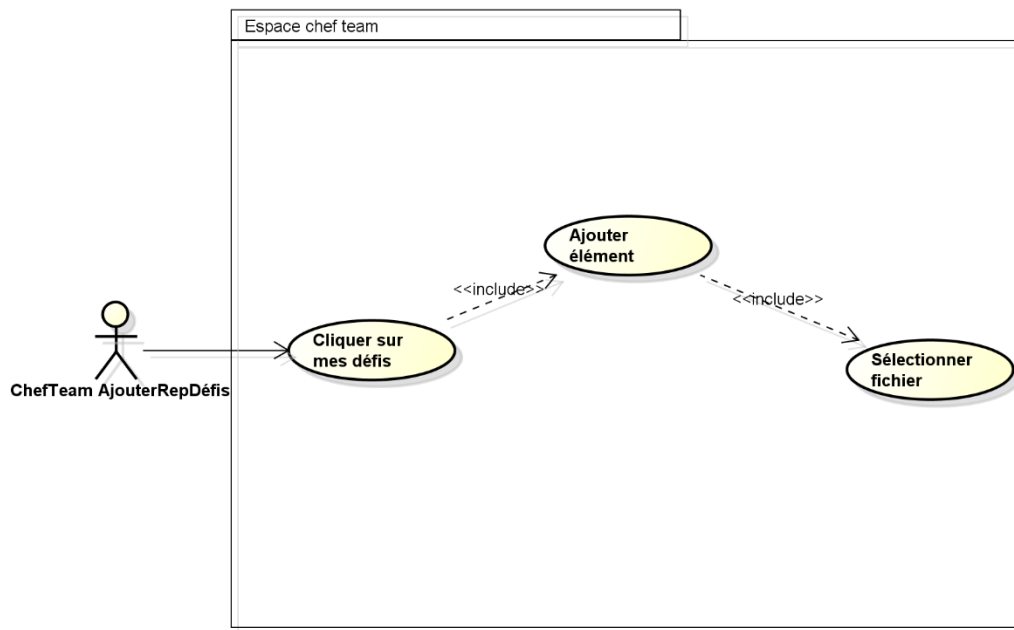


Figure 8, UML envoyer preuve défi chef team

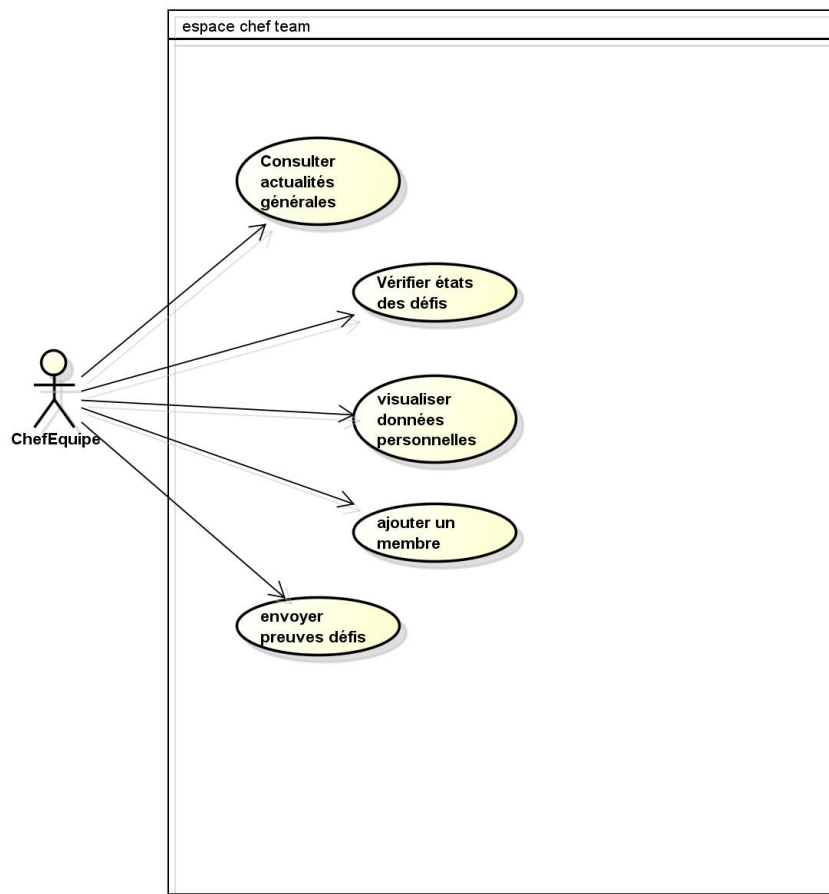


Figure 9, UML chef team general

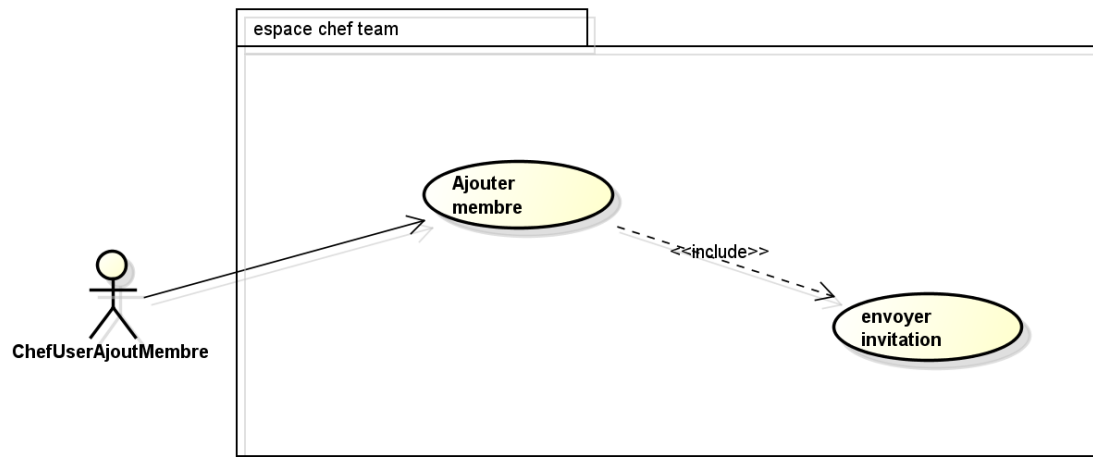


Figure 11, UML ajouter membre chef team

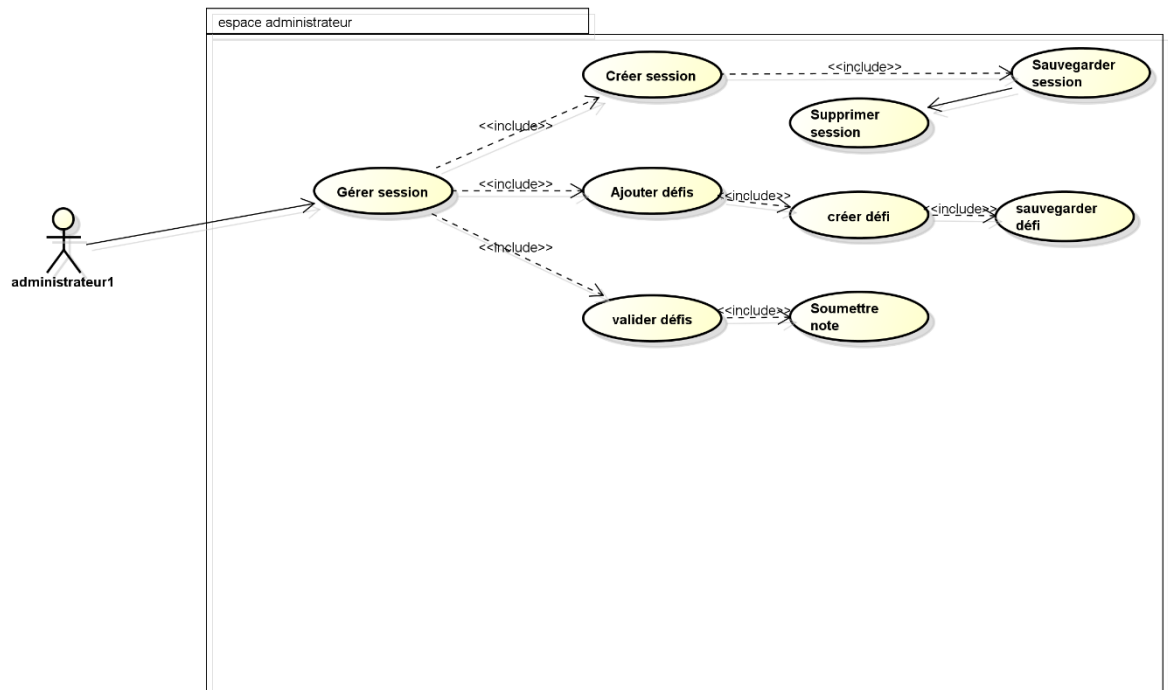


Figure 10, UML gestion administrateur

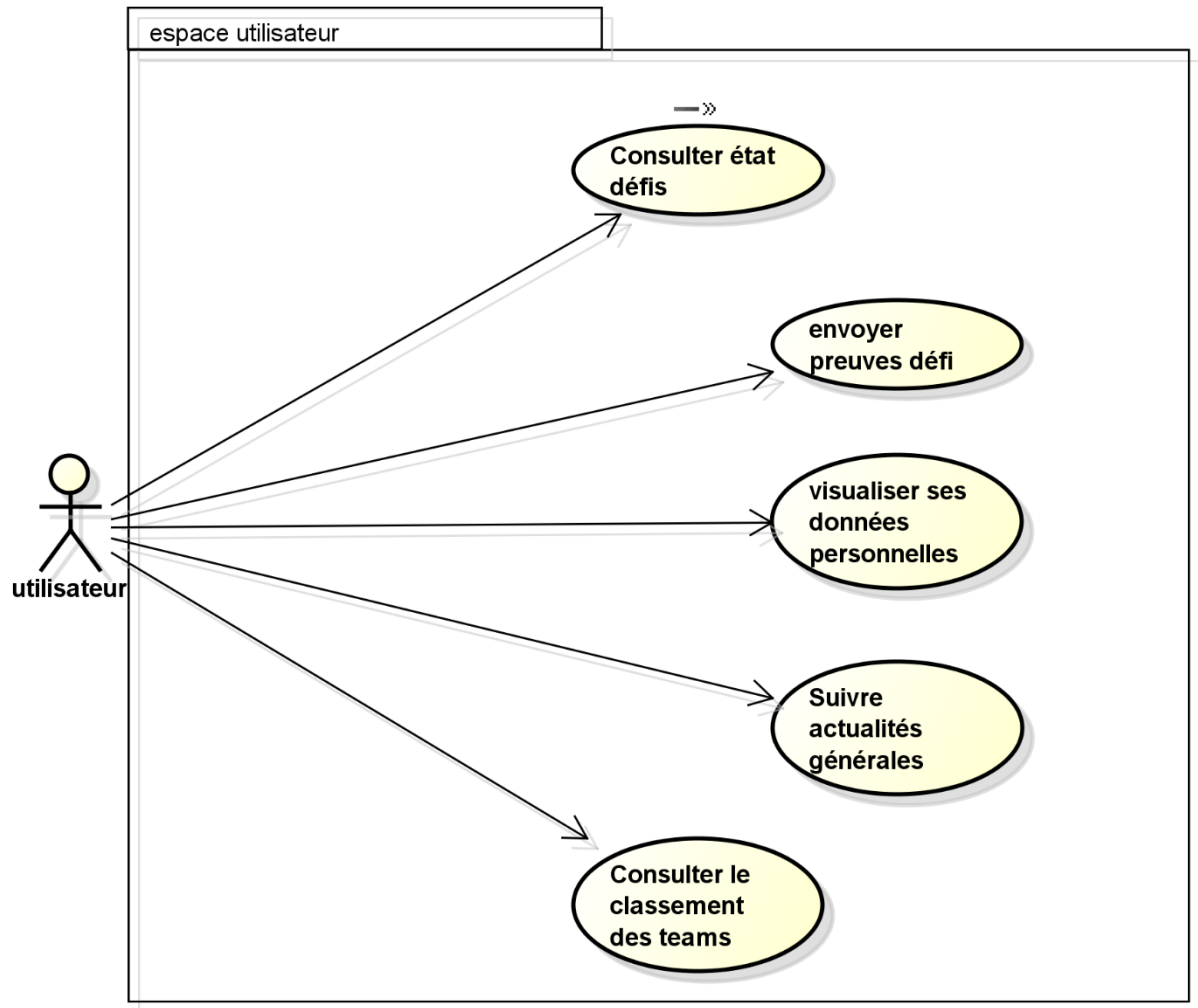


Figure 12, UML utilisateur general

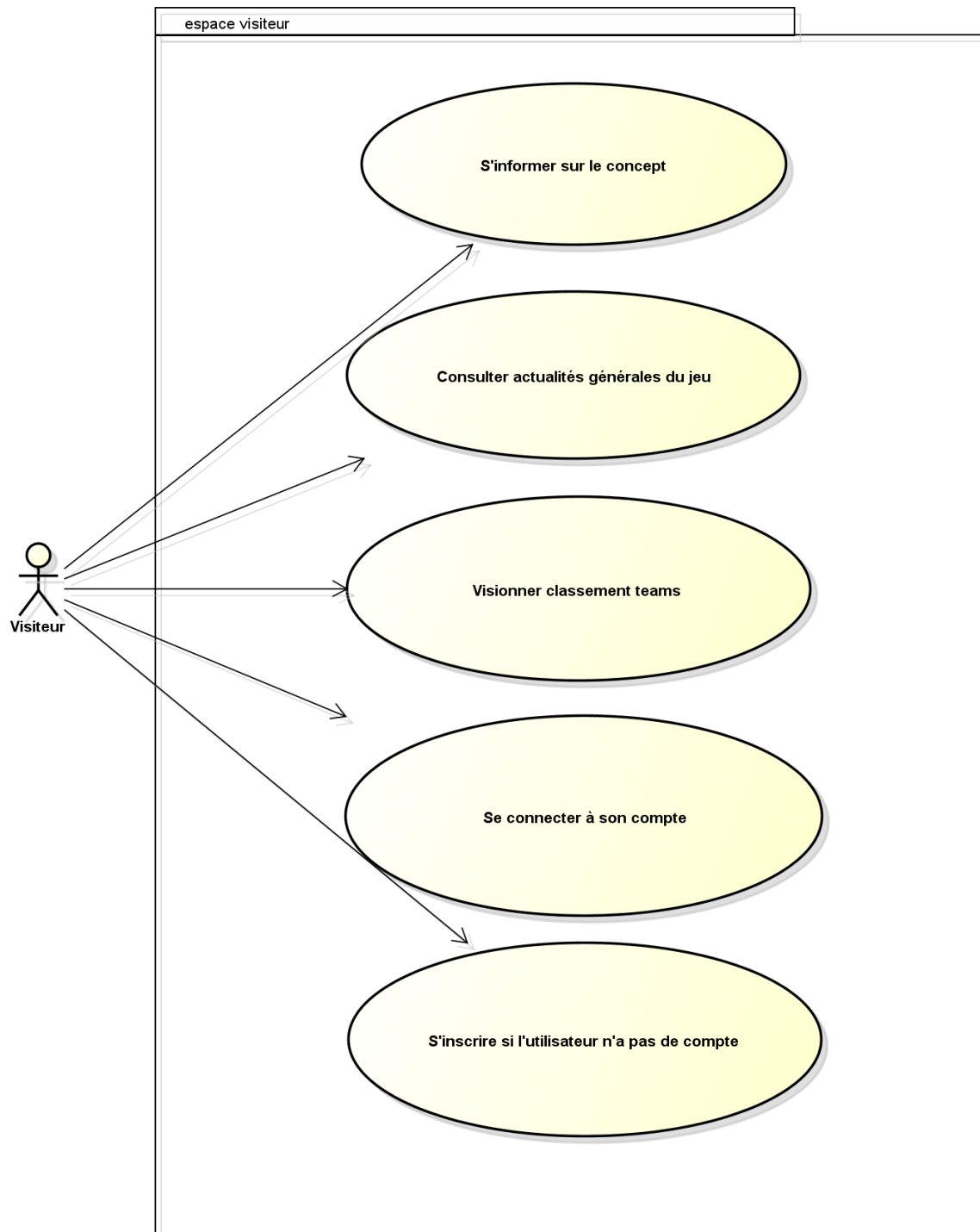


Figure 13, UML utilisateur

V. Base de données

29. Technologies utilisées

Concernant la base de données nous avons décidé d'utiliser MySQL, dans un premier temps parce que nous avons déjà beaucoup utilisé cette technologie. Mais également parce que MySQL est le système de base de données le plus approprié pour stocker le nombre important d'informations que demande ce projet.

30. Structure générale

La base de données du projet est composée des 7 tables suivantes :

- ➔ Admin
- ➔ ChefUser
- ➔ MembreUser
- ➔ Session
- ➔ Defi
- ➔ Team
- ➔ Article

Les points gagnés par les équipes après chaque défi seront stockés dans la table des teams. Après chaque défi, les points y seront cumulés jusqu'à la fin de la session. Le classement général sera donc basé sur ce champ.

31. Table Admin

Dans la table Admin on retrouve les champs suivants :

Id (AutoIncrement)

Nom (varchar)

Prénom (varchar)

Date de naissance (date)

Civilité (varchar)

Mot de passe (text)

Pays (varchar)

Code postal (varchar)

Ville (varchar)

Numéro de téléphone (varchar)

Adresse email (varchar)

Statut (varchar)

32. Table User

Dans la table MembreUser on retrouve les champs suivants :

Id (AutoIncrement)

Nom (varchar)

Prénom (varchar)

Date de naissance (date)

Civilité (varchar)

Mot de passe (text)

Pays (varchar)

Code postal (varchar)

Ville (varchar)

Numéro de téléphone (varchar)

Adresse email (varchar)

Statut (varchar)

Team (varchar)

33. Table Session

Dans la table Session on retrouve les champs suivants :

Id (AutoIncrement)

Date de début (date)

Date de fin (date)

Dotation (varchar)

Nom (varchar)

34. Table Defi

Dans la table Defi on retrouve les champs suivants :

Id (AutoIncrement)
Nombre de points (double)
Titre (varchar)
Description (text)
Session (varchar)

35. Table team

Dans la table Team on retrouve les champs suivants :

Id (AutoIncrement)
Nom (varchar)
Session (varchar)
Chef (varchar)
membreUn (varchar)
membreDeux (varchar)
membreTrois (varchar)
points (double)
argent (double)
DefiResponseVideo (blob)
DefiResponseText (text)
DefiResponsePicture (blob)

36. Table Article

Dans la table Article on retrouve les champs suivants :

Id (AutoIncrement)
Titre article (varchar)
Texte (longtext)
Images (blob)

37. MCD

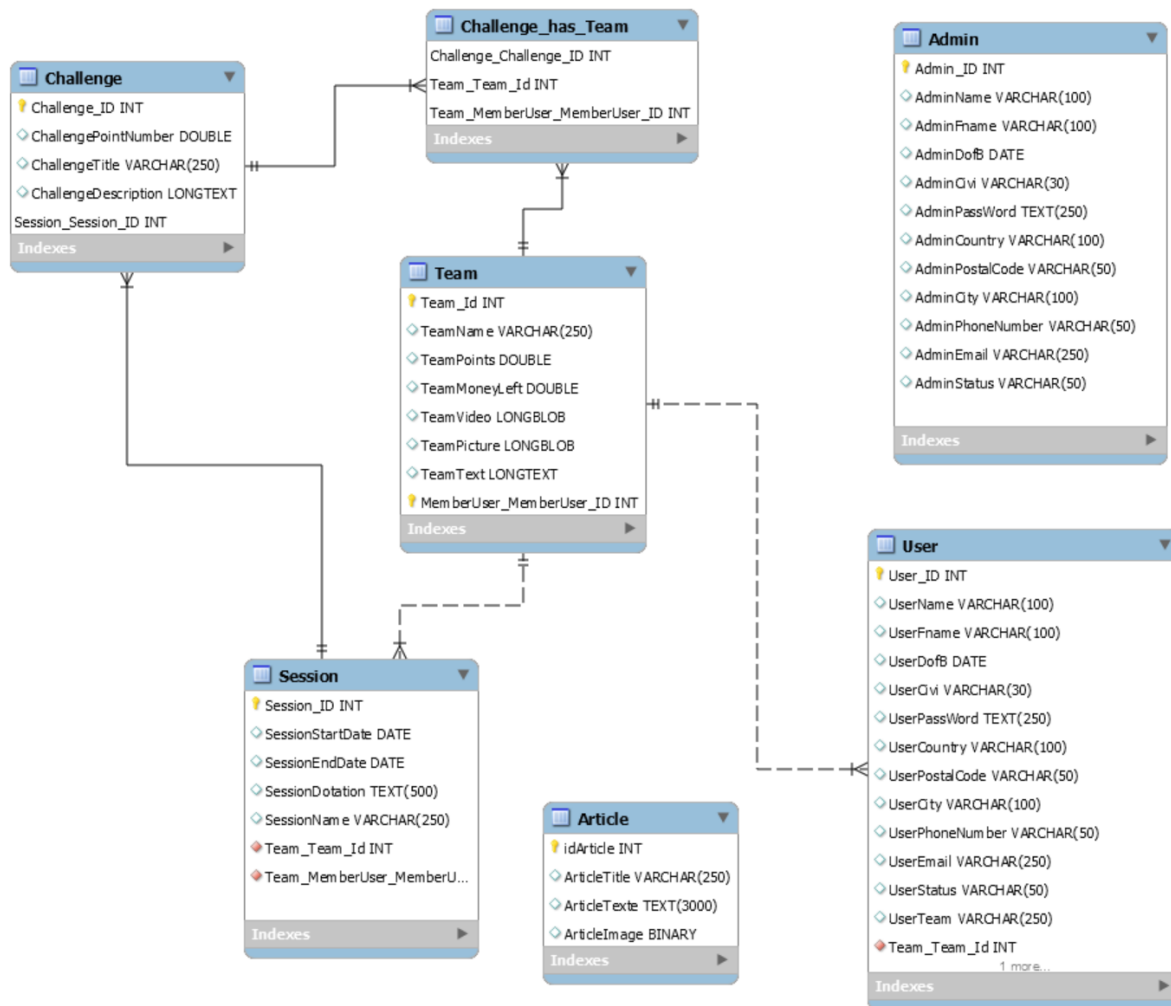


Figure 14, MCD

38. Sources

Paypal:

<https://developer.paypal.com/home>

<https://medium.com/in-laravel/how-to-integrate-paypal-into-laravel-977bf508c13>

Laravel Documentation:

<https://laravel.com/docs/8.x/installation>

Microsoft Azure:

https://azure.microsoft.com/fr-fr/free/search/?ef_id=EAIaIQobChMIgNT7_Z7gIVqejtCh0hLAMZEAAAYiAAE

[gKAs D BwE:G:s&OCID=AID2100046 SEM EAiaIQobChMIgN T7 Z7gIVqejtCh0hL AMZEAAyAiAAEgKAs D BwE:G:s&dcclid=CPLkyoGA2u4CFapmFQgdi9EImA](#)

MySQL Workbench:

<https://www.mysql.com/fr/products/workbench/>

Xampp:

<https://www.apachefriends.org/fr/index.html>

Adobe XD:

<https://www.adobe.com/fr/products/xd.html>

On the Road A Game:

<https://www.ontheroadagame.fr/>

Git du projet:

https://github.com/carolusDev/ProjetUF2020_2021.git