

# Rapport du projet Plateforme de recueil d'expérience vidéoludique



© Samantha Whitten

*Réalisé par*

Paul AMBLARD, Davide DE JESUS MARTINS, Corentin GRARD,  
Morgane PHUN-VONG

*Sous la direction de*  
Antoine CHOLLET

*Pour l'obtention du*  
DUT Informatique

*Année Universitaire 2017-2018*

# Remerciements

Nous adressons nos remerciements aux personnes qui nous ont aidées dans la réalisation de ce projet.

En premier lieu, nous remercions M. Chollet, professeur à l'IUT de Montpellier. En tant que tuteur, il nous a conseillé et aidé tout le long de ce projet.

Nous remercions également Mme Messaoui, professeur à l'IUT de Montpellier, pour ses conseils, son aide à la relecture et à la correction de notre rapport.

Nous souhaitons remercier M. Lebreton, professeur à l'IUT de Montpellier, pour son aide lorsque nous avons des problèmes techniques pendant ce projet.

Nous tenons à remercier M. Mesmoudi, élève à l'IUT de Montpellier, pour son aide et ses conseils.

Nous remercions Mme Monteil, une proche, pour son aide à la relecture et à la correction de notre rapport.

# Sommaire

<b>Remerciements</b>	II
<b>Sommaire</b>	III
<b>Table des figures</b>	V
<b>Glossaire</b>	VI
<b>Introduction</b>	VI
<b>Cahier des charges</b>	2
Présentation du sujet et analyse de l'existant	2
Présentation du sujet	2
Analyse de l'existant	2
Analyse des besoins fonctionnels	4
Analyse des besoins non fonctionnels	6
Spécifications techniques	6
Contraintes ergonomiques	6
<b>Rapport technique</b>	8
Conception	8
Réalisation	9
Architecture interne	9
Extraits de code	11
Technologies utilisées	14
<b>Résultats</b>	16
Test et validation	16
Manuel d'utilisation pour un utilisateur	16
Inscription	16
Connexion/Déconnexion	17
Son Profil	17
Poster une anecdote	17
Ses Brouillons	17
Interface d'une anecdote	18
Signalement	19
Filtrer les anecdotes	20
Rechercher un jeu	20
Manuel d'utilisation pour un administrateur	21
Liste des utilisateurs	21
Générer Rapport	21
Validation des jeux	22

Gérer les signalements	23
Interface d'une anecdote	24
<b>Gestion de projet</b>	25
Démarche personnelle	25
Planification des tâches	26
Bilan critique	27
<b>Conclusion</b>	28
<b>Bibliographie</b>	28
<b>Annexes techniques</b>	V
Annexe 1 : diagramme des cas d'utilisation	V
Annexe 2 : schéma du modèle relationnel	VI
Annexe 3 : diagrammes de séquence	VII
Annexe 4 : diagramme enregistrement d'une anecdote	VIII
Annexe 5 : ancien digramme de classe	IX

# Table des figures

Figure 1 - Une anecdote sur l'accueil de SeCoucherMoinsBete <sup>1</sup>	2
Figure 2 - Une anecdote sur l'accueil de SuisJeNormal <sup>2</sup>	3
Figure 3 - Diagramme des cas d'utilisation (zoom sur l'acteur Lambda)	4
Figure 4 - Diagramme des cas d'utilisation (zoom sur l'acteur Utilisateur)	5
Figure 5 - Diagramme des cas d'utilisation (zoom sur l'acteur Administrateur)	5
Figure 6 - Diagramme de classe	8
Figure 7 - Arborescence du site avec architecture MVC <sup>4</sup>	10
Figure 8 - Code PHP pour sauvegarder une anecdote	11
Figure 9 - Suite du code PHP pour sauvegarder une anecdote	12
Figure 10 - Code JS pour ajouter une réaction	13
Figure 11 - Code PHP pour ajouter une réaction	14
Figure 12 - Capture d'écran d'une erreur PHP	16
Figure 13 - Capture d'écran des inscriptions	16
Figure 14 - Capture d'écran du profil	18
Figure 15 - Capture d'écran d'une anecdote	19
Figure 16 - Capture d'écran du pop-up signalement	19
Figure 17 - Capture d'écran des filtres des anecdotes	20
Figure 18 - Capture d'écran de la page "Jeux"	20
Figure 19 - Capture d'écran de la liste des utilisateurs	21
Figure 20 - Capture d'écran de "Générer Rapport"	22
Figure 21 - Capture d'écran de la page "Validation des jeux"	23
Figure 22 - Capture d'écran du formulaire d'un jeu	23
Figure 23 - Capture d'écran de la liste des signalements	24
Figure 24 - Capture d'écran du détail des signalements	24
Figure 25 - Capture d'écran d'une anecdote côté administrateur	24
Figure 26 - Capture d'écran de l'outil Trello	26

# Glossaire

**Framework** : «ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).» (Framework [en ligne]. Wikipédia, 1 août 2003, 24 novembre 2017. <https://fr.wikipedia.org/w/index.php?title=Framework>)

**API** : «ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.» (API [en ligne]. Wikipédia, 19 mai 2003, 6 décembre 2017. [https://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation))

# Introduction

Les jeux vidéo prennent une place de plus en plus importante dans le marché du divertissement. Ayant même dépassés l'industrie du cinéma en 2015, en termes de chiffre d'affaires. Mais plus que du divertissement, le jeu vidéo commence à se faire reconnaître en tant qu'art. Et comme toute forme d'art, celui-ci véhicule des émotions.

Afin de mener des recherches dans ce domaine, le commanditaire de notre projet souhaite réaliser une plateforme de recueil d'expérience vidéoludique. Ayant pour but d'étudier la façon dont les gens expriment leurs émotions à travers leurs expériences vidéoludiques. Cette plateforme devra donc permettre aux utilisateurs de poster des anecdotes et à l'administrateur de récupérer les données relatives à ces anecdotes afin de pouvoir les utiliser dans ses recherches. De plus en dehors de la portée scientifique, M. Chollet souhaite que ce projet ait une dimension d'échange et de partage et pour cela les utilisateurs doivent pouvoir consulter les anecdotes et échanger à propos de ces dernières.

Pour cela nous allons tout d'abord présenter la conception de notre cahier des charges, suivi d'un rapport technique de notre solution. Il sera ensuite présenté en détail le résultat de notre travail. Et avant de conclure, les différentes méthodes de gestion de projet mise en place durant notre projet seront expliquées.

# 1. Cahier des charges

## 1.1. Présentation du sujet et analyse de l'existant

### 1.1.1. Présentation du sujet

Dans le but de mener à bien notre projet, nous allons réaliser un site internet de partage d'anecdotes. Il s'agit d'une plateforme simple, ne nécessitant pas d'installation préalable. Ce site permettra à des utilisateurs de poster des anecdotes sur des expériences vidéoludiques. Une base de données va recueillir ces anecdotes. Grâce à ce site, il sera aussi possible d'extraire des données liées aux anecdotes. Ces données pourront être ensuite traitées par le site afin d'en récupérer des informations.

### 1.1.2. Analyse de l'existant

Pour réaliser ce projet nous avons tout d'abord analysé des plateformes de partage d'anecdotes déjà existantes. Ainsi nous avons pu retenir différents points sur les besoins fonctionnel et ergonomique de ce type de plateforme.

Premièrement, nous avons aimé la possibilité de permettre à l'auteur de modifier son anecdote, de la commenter ou de mettre en valeur un commentaire pertinent d'un autre utilisateur.

Deuxièmement, le site SeCoucherMoinsBete<sup>1</sup> propose un système de résumé pour afficher le début de l'anecdote sur la page d'accueil (cf. figure 1). Nous avons ensuite la possibilité de cliquer sur "En savoir plus" pour pouvoir lire l'intégralité de l'histoire. Cela évite une certaine lourdeur visuelle. Ce qui permet d'avoir une harmonie entre chaque anecdote, elles ont donc toute la même apparence. Ainsi, si le résumé ne nous intéresse pas cela permet de choisir plus rapidement ce que nous allons lire.

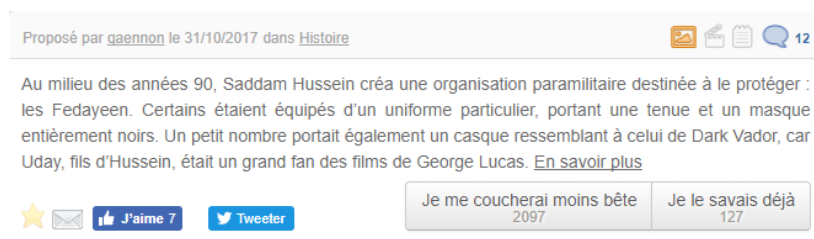


Figure 1 - Une anecdote sur l'accueil de SeCoucherMoinsBete<sup>1</sup>

<sup>1</sup> <http://secouchermoinsbete.fr/>



Sur ce même site, si nous possédons un compte nous pouvons mettre des anecdotes en favori. Cependant nous ne sommes pas dans l'obligation de s'inscrire. Sans compte, nous avons toujours accès au contenu du site mais certaines fonctionnalités comme "mettre en favori" ne sont pas disponibles. Nous avons pensé à intégrer cette option si nous avons le temps à la fin de notre projet. C'est une option qui n'est pas primordiale pour le site mais qui peut être intéressante pour le confort de l'utilisateur.

De plus, nous avons trouvé intéressant de pouvoir partager des anecdotes sur les réseaux sociaux. Cela permet de faire connaître une anecdote à ses amis ou d'en parler avec eux. Cela reste également une option qui n'est pas nécessaire pour le fonctionnement du site mais utile pour le confort de l'internaute.

D'une part, nous avons remarqué que la plupart des sites possédait une charte du site. Un point auquel nous ne pensons pas forcément mais qui est important dans un site. Nous avons alors retenu cette idée et souhaitons l'ajouter sur notre site.

D'autre part, nous avons apprécié les sites qui mettaient à la disposition des internautes une foire aux questions. Cela permet aux personnes de trouver les réponses à leurs questions plus rapidement. Ainsi, l'administrateur aura moins de questions via le formulaire de contact. Un gain de temps est donc gagné du côté de l'administrateur et des utilisateurs.

Puis, nous avons estimé qu'il fallait faire attention au design et à l'ergonomie de notre site. Certains sites présentent un mauvais design, des éléments trop compacts, une mauvaise visibilité des différentes informations. Ce qui pénalisent le site car le contenu peut être intéressant. Il faut donc soigner le design, faire quelque chose d'épuré et d'attractif. Cependant il ne faut pas non plus faire un design trop simple, trouver le juste milieu.

Ensuite, nous avons trouvé que la possibilité de signaler une anecdote est bien. Cela permet d'aider les administrateurs dans leur travail de modération.

De plus nous avons aimé le système de filtres qui permet de classer les anecdotes selon leur popularité, leur date de publication, leurs différentes catégories.

D'ailleurs, nous avons vu sur le site SuisJeNormal<sup>2</sup> qu'en cliquant sur une anecdote nous avons accès au détail des réactions des utilisateurs (cf. figure 2). Ce détail peut-être original pour les lecteurs ou l'auteur de l'anecdote. Un élément à garder de côté si nous avons le temps de le développer par la suite.

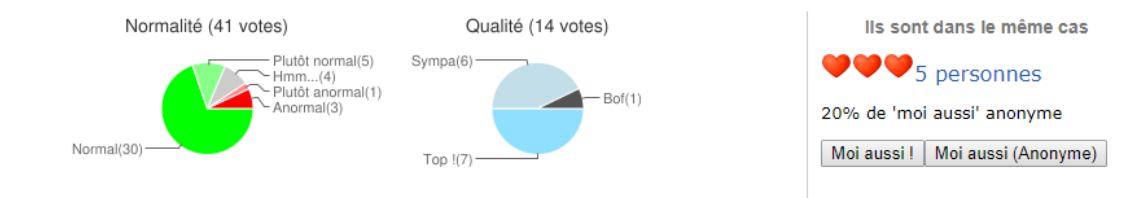


Figure 2 - Une anecdote sur l'accueil de SuisJeNormal<sup>2</sup>

<sup>2</sup> <http://www.suisjenormal.fr/>

Toutefois nous avons décidé qu'il est préférable d'éviter de mettre des publicités invasives car cela gêne le confort du lecteur. De plus dans notre cas, nous ne comptons pas mettre de publicité sauf si notre client change d'avis.

## 1.2. Analyse des besoins fonctionnels

Actuellement, les sites de partages sont de plus en plus présents sur internet. C'est dans ce contexte que notre site va s'insérer. Le principe étant de permettre à un utilisateur de partager une anecdote sur une de ses expériences vidéoludiques. Grâce à l'étude de ces sites communautaires, nous avons pu concevoir le nôtre. Nous allons donc étudier le diagramme des cas d'utilisation, en trois parties. Dans un premier temps, nous verrons la partie liée aux personnes lambdas. Dans second temps, nous étudierons les fonctionnalités des utilisateurs et enfin, nous verrons côté administrateur. Pour faciliter ces études, nous avons découpé et effectué des zooms sur le diagramme ("voir diagramme des cas d'utilisation, placé en annexe").

Étant donné que certaines fonctionnalités du site sont bloquées à une personne lambda, il lui est nécessaire d'avoir un compte pour s'exprimer. Cela permet entre autres de garder une trace de ce que font les utilisateurs. Cependant, il peut quand même accéder à la base du site, c'est-à-dire, voir des anecdotes. Toute autre fonctionnalité est donc bloquée. (Cf. figure 3.)

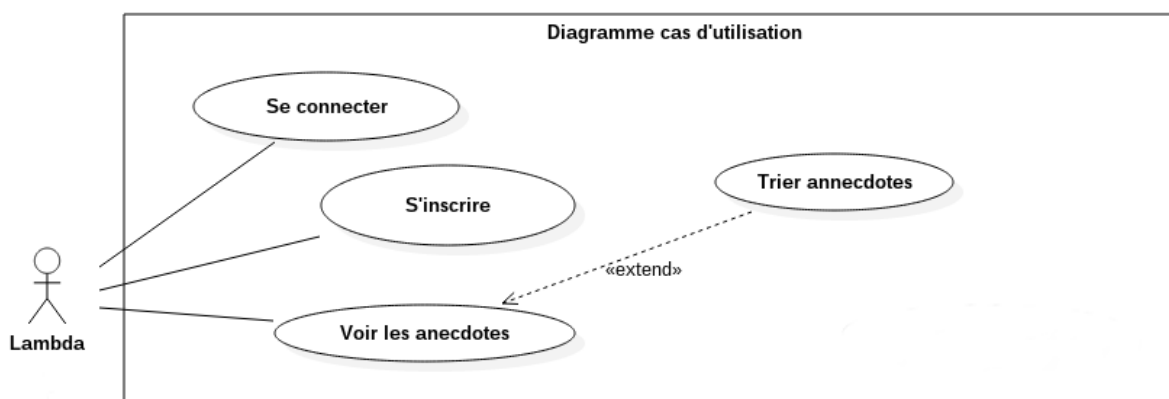


Figure 3 - Diagramme des cas d'utilisation (zoom sur l'acteur Lambda)

Une fois connecté, un utilisateur a accès à d'autres fonctionnalités. Il s'agit d'un site communautaire, donc il va donner vie au site en ajoutant du contenu en postant des anecdotes. Cependant, certains utilisateurs potentiellement mal intentionnés peuvent mettre du contenu inapproprié. C'est en ce sens que nous permettons aux utilisateurs de faire remonter ces événements à l'administrateur en signalant ce contenu. (Cf. figure 4.)

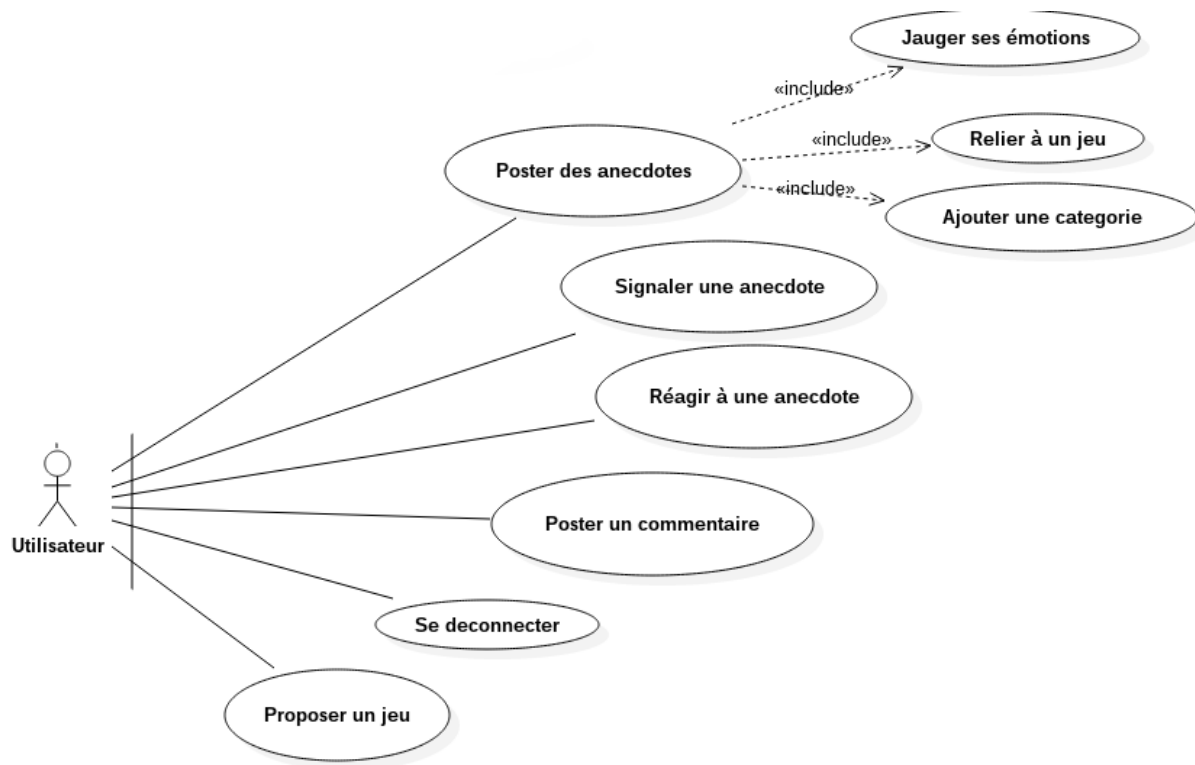


Figure 4 - Diagramme des cas d'utilisation (zoom sur l'acteur Utilisateur)

Le rôle principal de l'administrateur est de gérer le site, afin d'éviter tout débordement. Il peut donc supprimer des anecdotes qui auront été signalées par les utilisateurs. Si un utilisateur se fait remarquer par sa mauvaise conduite, il sera banni par l'administrateur. Le monde des jeux vidéo évolue constamment. Il y a toujours de nouveaux jeux qui sortent. L'administrateur devra maintenir la base de données des jeux à jour en les ajoutant. Le but de ce site étant scientifique, M. Chollet aura la possibilité via des filtres (jeux, émotions...) de générer un rapport sur des éléments spécifiques. Ainsi, il pourra étudier ce rapport dans le cadre de ses recherches. (Cf. figure 5.)

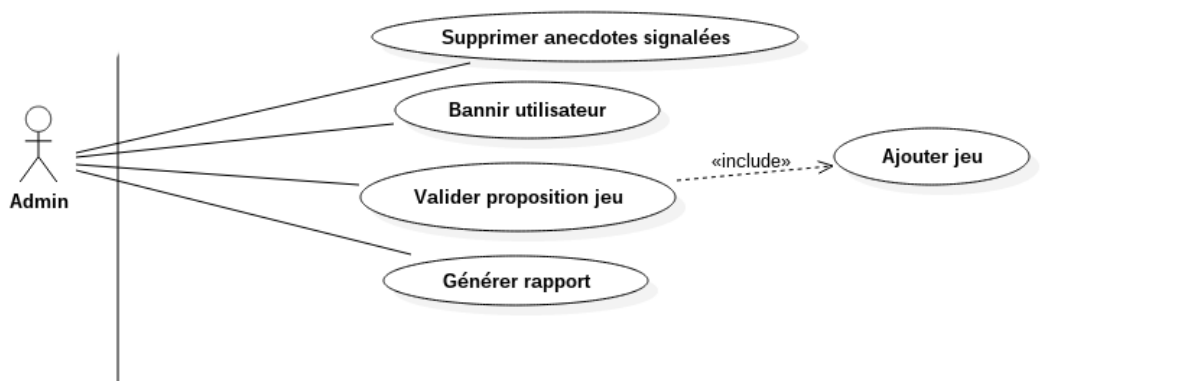


Figure 5 - Diagramme des cas d'utilisation (zoom sur l'acteur Administrateur)

## 1.3. Analyse des besoins non fonctionnels

### 1.3.1. Spécifications techniques

Les spécifications techniques se découpent, dans notre cas, en deux catégories. Les spécifications explicitement précisées par le client et celles implicites qui ont été déduites à l'aide de l'analyse de l'existant que nous avons mené, de ce qui est sous-entendu par M. Chollet. Tout ceci ayant servi à réaliser un cahier des charges sous la forme d'une user map story.

Au cours de nos entretiens avec notre commanditaire, les principales contraintes explicites, développées précédemment, étaient que la plateforme de recueil d'expérience vidéoludique doit permettre de récupérer les données nécessaires pour ses recherches.

De plus il était explicitement demandé que la plateforme se présente sous la forme d'un site internet PHP. Un autre souhait de notre client était la dimension sociale et émotionnelle que devait avoir ce projet, décrit comme une plateforme de partage. Cela a donc défini les spécifications et les contraintes régissant le partage et la réaction aux anecdotes. Par exemple, pour permettre à un utilisateur d'exprimer ses émotions, il était nécessaire de ne pas le brider par une limite de caractère trop faible. Et il fallait également mettre à disposition un outil lui permettant d'écrire de manière libre, pouvant varier sa taille de police d'écriture ou autre.

De ces différentes spécifications, il a été déduit de multiples outils et langages pour développer ce projet. Le PHP a été choisi afin de réaliser le corps du projet car fortement conseillé par notre tuteur. Afin de remplir les demandes concernant l'ergonomie et l'esthétique de la plateforme l'outil CSS ainsi que le JavaScript, langage utile afin de remplir les contraintes ergonomiques qui seront expliquées plus tard.

Pour récupérer, traiter et stocker les données nécessaires non seulement au fonctionnement primaire du site mais également aux recherches de M. Chollet. Un serveur HTTP Apache a été utilisé avec une base de donnée PHPMyAdmin qui utilise le langage MySQL.

Chacun de ces outils ont apporté leurs propres contraintes en fonction de ses limites d'utilisation.

Toutes les spécifications résultant des différentes contraintes, aussi bien matériel que choisi par le client ont dû être pris en compte dans la conception du projet.

### 1.3.2. Contraintes ergonomiques

En raison des objectifs de notre projet, le site est destiné à être accessible à tous et doit attirer le plus d'utilisateurs possibles. Pour cela le site doit remplir plusieurs contraintes ergonomiques afin de faciliter son usage. L'objectif étant de rendre la navigation sur le site le plus fluide et agréable possible.

Pour remplir ces contraintes nous nous sommes servis des dix principes de Jacob Nielsen<sup>3</sup> sur l'utilisabilité des sites internet comme ligne directrice de l'ergonomie de notre site. Ces dix principes représentent chacun une contrainte, certain n'ayant pas pu être rempli à cause du manque de temps, que notre site doit remplir.

Le premier étant la visibilité du statut du système, l'utilisateur ne doit jamais se perdre sur le site et doit toujours avoir un feedback sur ses principales actions. Pour cela nous avons veillé à garder une construction des pages simple et intuitive. Ainsi qu'à ajouter de nombreuses notifications afin de garder l'utilisateur au courant sur le traitement de ses actions.

La seconde contrainte étant que le site doit parler le même langage que l'utilisateur, dans le sens où le site doit se mettre au niveau de l'utilisateur. Dans notre cas, ce fut une certaine contrainte du fait de la portée scientifique du site, la création d'une anecdote et d'un profil doit rester simple et rapide et des compromis ont dû être faits entre l'utilisabilité et la précision des données récoltées.

Le troisième point consistant à permettre à l'utilisateur de personnaliser son expérience du site qui n'a pas été réalisée car cela été jugé trop long à mettre en œuvre pour le bénéfice que cela apporte.

Le quatrième principe de Nielsen, en lien avec le contenu textuel du site ne dépend pas entièrement de nous du fait qu'il sera livré un produit dépourvu de certain texte.

La cinquième et neuvième contraintes définissent le besoin d'une gestion d'erreur efficace et compréhensible par l'utilisateur, ce qui a été pris en compte dans notre projet.

Le sixième et septième principes respectivement pour le besoin d'établir un code visuel relatif à chaque action et pour la création de raccourcis afin de simplifier l'expérience des habitués ont été réalisés dans la mesure de l'utile.

Le huitième critère définissant un design esthétique et minimaliste comme permettre de maximiser l'utilisabilité a été pris en compte dans la réalisation des maquettes ayant servi à construire le site.

Le dixième point, quant à lui est la présence d'un manuel d'utilisation ce qui est le cas dans ce rapport et sur le site.

Ces contraintes permettent d'augmenter l'utilisabilité du site et donc de rendre sa navigation plus simple et agréable.

---

<sup>3</sup> <http://tecfa.unige.ch/tecfa/teaching/LMRI41/PrincipesNielsen.html>

## 2. Rapport technique

### 2.1. Conception

Avant de commencer notre projet, nous avons fait toute une partie de conception grâce principalement à des diagrammes UML qui nous ont permis d’avoir une idée plus claire du futur fonctionnement de notre site.

Nous avons donc commencé par réaliser un diagramme de classe (cf. figure 6) pour voir les différentes données qui se trouveront sur notre site et voir comment ses données seront liées aux autres. Cependant avant de produire ce diagramme de classe, nous avons fait avant un premier diagramme qui a été modifié (“voir ancien diagramme de classe, placé en annexe”). À partir de là nous avons créé le modèle relationnel (“voir schéma du modèle relationnel, placé en annexe”) qui correspond à notre future base de données.

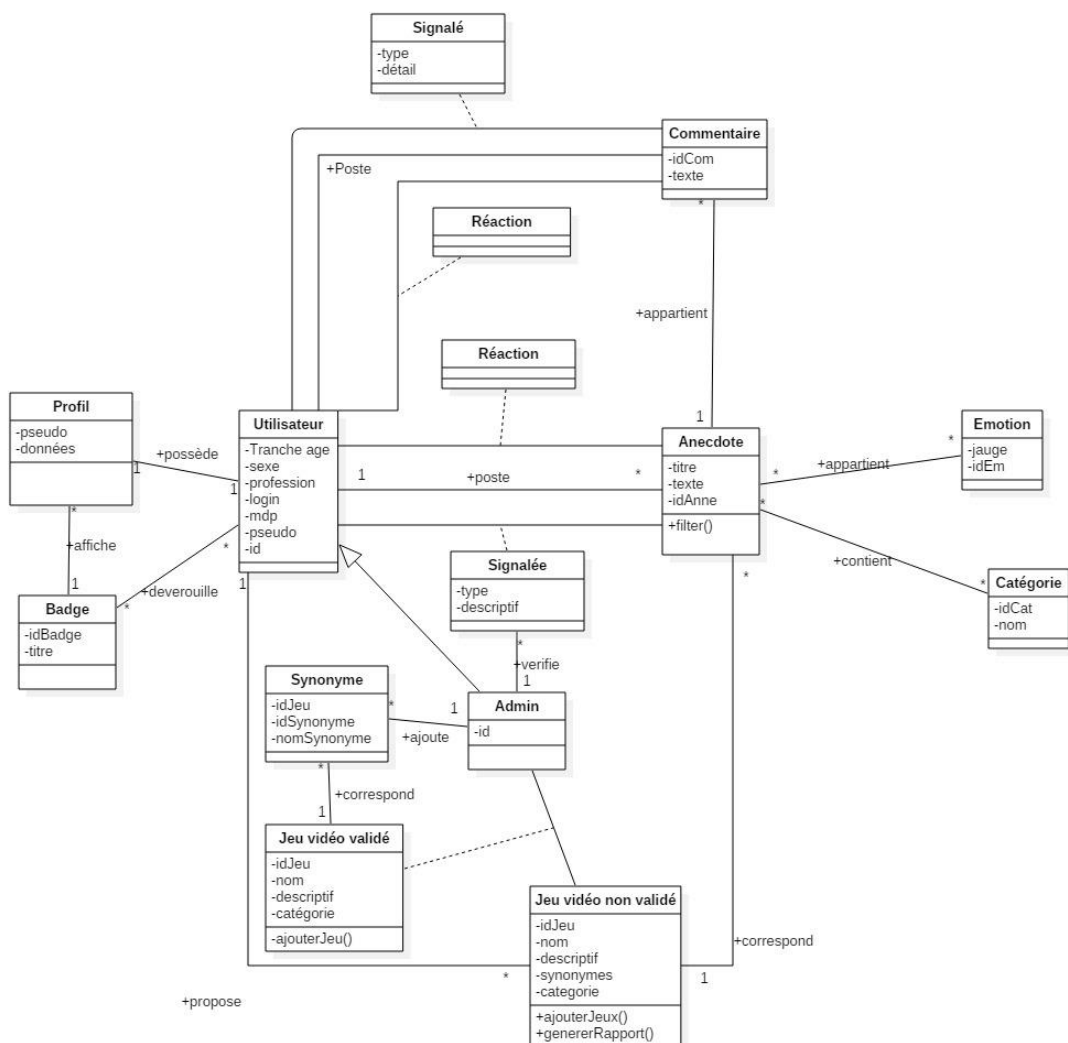


Figure 6 - Diagramme de classe

Ensuite nous avons réalisé deux diagrammes de séquence (“voir diagramme de séquences, placé en annexe”) pour comprendre comment un utilisateur va naviguer sur notre site. Pour cela nous avons réalisé un premier diagramme qui nous montre comment un utilisateur pourra poster une anecdote sur notre site et un autre montrant comment un utilisateur pourra soumettre un jeu à un administrateur. Malheureusement ces diagrammes ont été réalisés en début de projet et ne sont pas représentatifs du fonctionnement final du site.

Nous avons donc réalisé un autre diagramme qui représente l’enregistrement d’une anecdote (“voir diagramme enregistrement d’une anecdote, placé en annexe”). Il explique comment le système va enregistrer une anecdote entrée par un utilisateur. Il regroupe donc les deux diagrammes de séquence prévus initialement.

## 2.2. Réalisation

### 2.2.1. Architecture interne

Pour réaliser ce projet, nous avons dû trouver un hébergeur pour notre site qui a été infolimon, à l’IUT. Nous nous sommes ensuite attaqués à des questions comme l’architecture de site. Pour cela, nous avons choisi une architecture Modèle, Vue, Contrôleur. Elle permet de bien séparer les différentes actions internes du site que ce soit de l’accès à la base de données jusqu’à l’affichage des données.

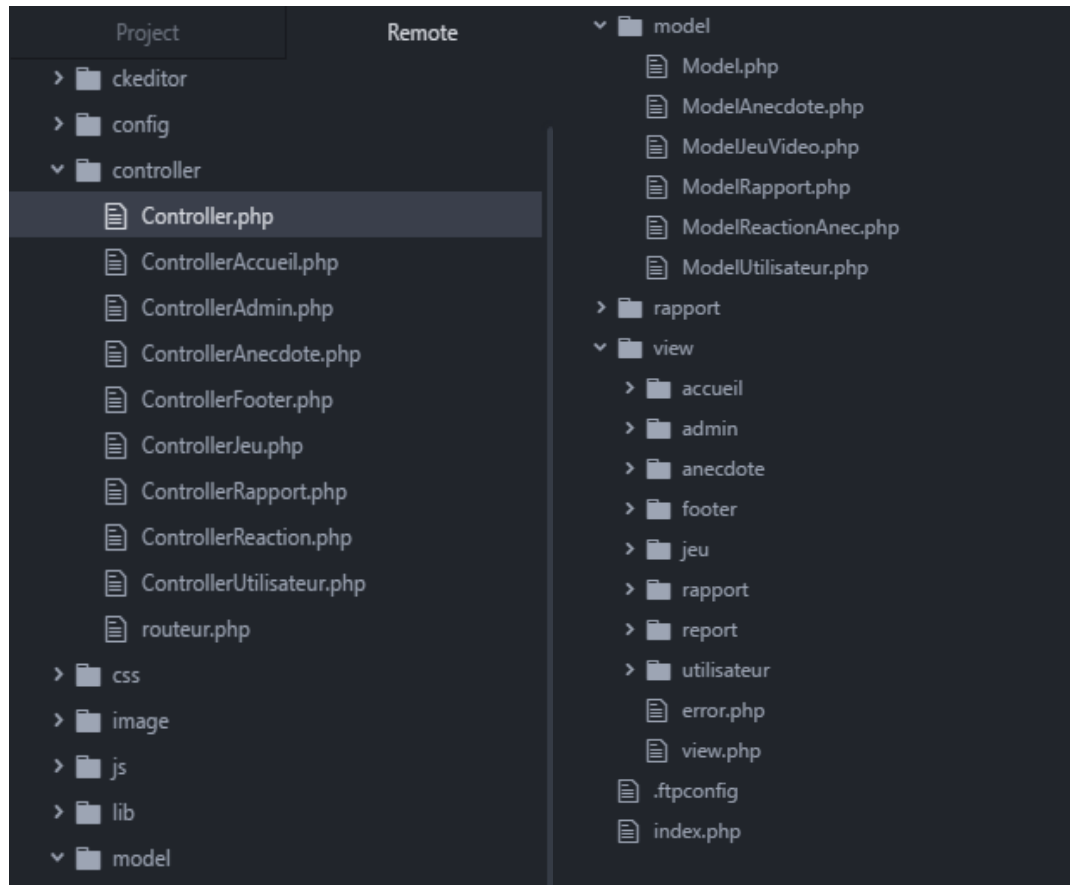


Figure 7 - Arborescence du site avec architecture MVC<sup>4</sup>

Il a aussi été nécessaire de séparer le travail en différentes fonctions. Nous avons la gestion des utilisateurs, qui va de leur inscription à leur gestion par l'administrateur (modification, suppression d'un utilisateur) en passant par les réactions des utilisateurs. Nous avons également la gestion des anecdotes. Ceci regroupe les fonctions permettant de créer les anecdotes, jusqu'à leur affichage. Ces anecdotes peuvent être signalées les mettant dans une liste. Cette liste peut-être vue par l'administrateur afin qu'il supprime les anecdotes si nécessaire. Nous devons aussi gérer les jeux reliés aux anecdotes, de leur proposition par un utilisateur à leur insertion dans la base de données. De plus, une anecdote pouvant être commentée, nous avons dû implémenter un système permettant aux utilisateurs d'écrire un commentaire, qui peut aussi être signalé. Enfin, il y a la fonctionnalité "scientifique" permettant de générer des rapports en fonction de filtres rentrés au préalable.



### 2.2.2. Extraits de code

Nous allons ici détailler quelques-unes de nos fonctionnalités importantes.

#### Enregistrer une anecdote :

Cette fonction created correspond au moment où un utilisateur a rempli une anecdote et cliqué sur le bouton “Publier” ou “Ajouter aux brouillons”. Ce que réalise cette fonction est l’algorithme présenté plus haut (“voir diagramme sur le fonctionnement du site, placé en annexe”). Cette fonction a donc pour rôle d’enregistrer une anecdote. (Cf. figure 8 et 9.)

```
67 //Fonction permettant d'enregistrer une anecdote dans la BD
68 public static function created(){
69     if(isset($_SESSION['login'])){ //Si le user est connecté
70         $jeu=ModelJeuVideo::selectNom(Util::myGet('jeu')); //Sélection du jeu
71         // entré par l'utilisateur si il existe
72         $publie=0;
73         if($jeu==false){ //Si il n'existe pas alors
74             if(!ModelJeuVideo::save(array( //On enregistre un nouveau jeu dans la
75                 //base de donnée, jeu non valide
76                 "nomJeu"=>Util::myGet('jeu'),
77                 "valide"=>0,
78             ))){
79                 //Affichage page d'erreur si enregistrement fail
80                 $view=array("view", static::$object, "errorSave.php");
81                 $pagetitle='Jeu non sauvegardé';
82                 require(File::build_path(array("view","view.php")));
83                 die();
84             }else{
85                 //On sélectionne le jeu qui vient d'être créé
86                 $jeu=ModelJeuVideo::selectNom(Util::myGet('jeu'));
87             }
88             //Si l'utilisateur a rentré un jeu valide alors on respecte
89             //son choix de soit publier l'anecdote, soit la mettre dans les brouillons
90         }elseif ($jeu->get('valide')==1){
91             $publie=Util::myGet('publie');
92         } //Sinon on garde publie à 0 et donc on enregistre son jeu dans les brouillons
93         $idJeu=$jeu->get('idJeu'); //Sélection de l'id du jeu
94         $id=uniqid(); //Génération de l'ID de l'anecdote
95         if(ModelAnecdote::save(array( //Enregistrement de l'anecdote
96             "idAnecdote"=>$id,
97             "titre"=>Util::myGet('titre'),
98             "texte"=>Util::myGet('texte'),
99             "categorie"=>Util::myGet('categorie'),
100             "idJeu" => $idJeu,
101             "idLogin"=>$_SESSION['login'],
```

Figure 8 - Code PHP pour sauvegarder une anecdote

```

98         "texte"=>Util::myGet('texte'),
99         "categorie"=>Util::myGet('categorie'),
100        "idJeu" => $idJeu,
101        "idLogin"=>$_SESSION['login'],
102        "publie" => $publie,
103        "joie"=>Util::myGet('joie'),
104        "peur"=>Util::myGet('peur'),
105        "colere"=>Util::myGet('colere'),
106        "degout"=>Util::myGet('degout'),
107        "tristesse"=>Util::myGet('tristesse'),
108        "surprise"=>Util::myGet('surprise'),
109    )==false){
110        //Affichage page erreur enregistrement
111        $tab_u = ModelAnecdote::selectAll();
112        $view=array("view", static::$object, "errorSave.php");
113        $pagetitle='Anecdote non sauvegardée';
114        require(File::build_path(array ("view","view.php")));
115    }elseif(Util::myGet('publie')==0){
116        //Affichage page si enregistré dans les brouillons
117        header("Location: index.php?controller=anecdote&action=readBrouillon");
118    }elseif($jeu->get('valide')==0){
119        //Affichage page si jeu non validé donc on ne peut pas publier
120        $tab_a=ModelAnecdote::selectBrouillon($_SESSION['login']);
121        $view=array("view", static::$object, "nonPublie.php");
122        $pagetitle='Anecdote non publiée';
123        require(File::build_path(array ("view","view.php")));
124    }else{
125        //Affichage de l'anecdote si publié
126        header("Location: index.php?controller=anecdote&action=readAll&idAnecdote=$id");
127    }
128 }else{
129     //Si l'utilisateur n'est pas connecté retour à la page d'accueil
130     header('Location: index.php');
131 }
132 }

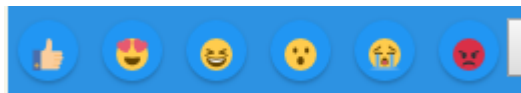
```

Figure 9 - Suite du code PHP pour sauvegarder une anecdote

## Ajouter une réaction :

Cette fonction n'est pas essentielle à notre site mais elle montre bien toutes les différentes technologies que nous avons pu utiliser durant le projet tel que le JavaScript avec JQuery<sup>4</sup>. Grâce à JQuery et AJAX nous avons pu modifier le contenu de la page avec l'action d'un utilisateur. Ici nous allons voir comment avec un clic, un utilisateur peut ajouter une réaction à une anecdote et avoir un retour visuel du succès de son clic.

Avant l'ajout d'une réaction :



Après l'ajout d'une réaction :



<sup>4</sup> Voir bibliographie [20]

Pour faire cela nous avons donc utilisé du JavaScript, on appelle la fonction `addReact` (cf. figure 10) quand l'utilisateur clique sur un bouton. Ensuite JQuery va faire une requête AJAX sur le site. Ici on peut voir (l. 76/77) que nous allons envoyer une requête à l'adresse "index.php?controller=reaction&action=addReact&valeur=valeur&idAnecdote=idAnecdote". Si la requête réussit et le site nous renvoie une réponse alors il ne nous reste qu'à supprimer la possible réaction sur l'anecdote et ajouter la nouvelle.

```
73 //Envoie requete ajout réaction.
74 function addReact(idAnec,val){
75     $.ajax({ //Requête AJAX
76         type: "POST",
77         url: "index.php?controller=reaction&action=addReact",//URL
78         data: "valeur="+val+"&idAnecdote="+idAnec,
79         dataType : 'html', // On désire recevoir du HTML
80         success : function(data){ //data contient le possible "true"
81             $("#"+idAnec+" .reaction").removeClass("selected");
82             if(data){//Si on a ajouté une réaction
83                 $("#"+idAnec+val).addClass("selected");
84             }
85         }
86     });
87 }
```

Figure 10 - Code JS pour ajouter une réaction

Maintenant nous allons voir comment le serveur récupère la requête envoyée par JQuery et la traite. Avec la requête envoyée, la fonction appelée dans le PHP sera addReact du controller reaction (cf. figure 11).

```
11 //Fonction permettant d'ajouter une réaction à une anecdote
12 public static function addReact(){
13     if(isset($_SESSION['login'])){//On peut ajouté une réaction que en étant connecté
14         //On regarde si l'utilisateur avait déjà mis une réaction pour cette anecdote
15         $res=ModelReactionAnec::select(array("idLogin" => $_SESSION['login'],
16         "idAnecReac" => $_POST['idAnecdote']));
17         if($res!=false){//Si il y en avait déjà une
18             ModelReactionAnec::delete(array("idLogin" => $_SESSION['login'],//On la supprime
19             "idAnecReac" => $_POST['idAnecdote']));
20             //On décrémente le compteur de réaction de 1
21             self::addNbReact($res->get('typeReacAnec'),$_POST['idAnecdote'],-1);
22             //Si la réaction est différente que celle qui était déjà là alors
23             if($res->get('typeReacAnec')!= $_POST['valeur']){
24                 ModelReactionAnec::save(array{//On enregistre la nouvelle réaction
25                     "idLogin" => $_SESSION['login'],
26                     "idAnecReac" => $_POST['idAnecdote'],
27                     "typeReacAnec" => $_POST['valeur'],
28                 ));
29                 //On incrémente le compteur de réaction
30                 self::addNbReact($_POST['valeur'],$_POST['idAnecdote'],1);
31                 print "true";//On renvoie true au Javascript
32             }
33         }else{//Si c'est la première réaction qu'il ajoute à l'anecdote alors
34             ModelReactionAnec::save(array{//On l'enregistre
35                 "idLogin" => $_SESSION['login'],
36                 "idAnecReac" => $_POST['idAnecdote'],
37                 "typeReacAnec" => $_POST['valeur'],
38             ));
39             //On incrémente le compteur de réaction
40             self::addNbReact($_POST['valeur'],$_POST['idAnecdote'],1);
41             print "true";//On renvoie true au Javascript
42         }
43     }
44 }
```

Figure 11 - Code PHP pour ajouter une réaction

## 2.3. Technologies utilisées

Pour faciliter le développement du site, nous avons utilisé des bibliothèques, Framework et éditeur de texte.

Nous avons utilisé Bootstrap<sup>5</sup>, un Framework\*, pour permettre au site d'avoir rapidement un front-end élaboré. Bootstrap offre un responsive design tout fait. Ce qui permet de ne pas perdre du temps à adapter notre site pour chaque appareil. Ce Framework permet l'utilisation de boutons, de tables, de formulaire ayant un design très esthétique. Alors nous n'avons pas besoin de nous attarder sur le côté graphique de notre site. Ainsi nous avons pu nous concentrer sur le développement des principales fonctionnalités du site.

Nous avons aussi utilisé la bibliothèque CKeditor<sup>6</sup> qui nous a permis d'avoir un éditeur de texte simple à mettre en place et complet.

Nous avons également installé le plugin “spoiler” venant de CKeditor. Nous souhaitons permettre aux utilisateurs de pouvoir cacher une partie de leur rédaction. Car il arrive que pour exprimer ses expériences, l'internaute doit dévoiler les événements qui se produisent dans un jeu. Les lecteurs ne veulent pas être au courant de ses événements, c'est pourquoi le système de “spoiler” est important.

De plus nous avons utilisé le Framework MUI<sup>7</sup> pour faire les boutons des réactions. MUI semble être un Framework très complet mais nous ne l'utilisons que pour des boutons ce qui est très réducteur. Il faudrait idéalement isoler le code des boutons pour alléger la taille des fichiers envoyés à l'utilisateur et donc accélérer le chargement de la page.

Enfin nous avons utilisé la bibliothèque jQuery. Celle-ci permet de faciliter l'écriture du JavaScript. Grâce à elle nous avons réalisé quelques éléments animés sur notre site ainsi que des fonctions Ajax qui nous permettent de récupérer des données du serveur.

---

<sup>5</sup> Voir bibliographie [4]

<sup>6</sup> Voir bibliographie [8]

<sup>7</sup> Voir bibliographie [29]

## 3. Résultats

### 3.1. Test et validation

Nous avons testé continuellement notre site, afin de s'assurer qu'il était bien fonctionnel. Pour les tests de fonctionnalités, nous avons dans un premier temps fait des tests "personnels". Pour cela nous nous sommes servis des erreurs PHP pour réparer les plus grosses erreurs (cf. figure 12).

**Fatal error: Uncaught Error: Call to undefined method**

*Figure 12 - Capture d'écran d'une erreur PHP*

Nous avons ensuite utilisé l'aspect visuel du site, les données affichées et toute autre méthode visible directement sur le site afin d'arranger d'autres erreurs. Par exemple, lors de l'implémentation de la liste d'anecdotes, en voyant que toutes les anecdotes étaient pareilles, nous avons compris qu'il y avait une erreur. Nous avons ainsi pu la corriger et avoir un site fonctionnel.

Pour la validation des stories, nous commençons par regarder si elle correspondait aux attentes du client. Si c'était le cas, nous la passons en "terminer à valider". Ensuite, lors de nos réunions avec le M. Chollet, nous lui proposons notre solution et il la validait ou nous disait ce qui n'allait pas.

### 3.2. Manuel d'utilisation pour un utilisateur

Vous pouvez accéder au site via le lien suivant : <http://webinfo.iutmontp.univ-montp2.fr/~phunvongm/videoludique/index.php?>

#### 3.2.1. Inscription

Pour vous inscrire sur le site, veuillez cliquer sur le bouton "Inscription" dans le menu de navigation en haut de votre écran, indiquée par le rectangle rouge. Ceci vous redirigera sur un formulaire d'inscription. Il faudra alors remplir chaque champ demandé. Attention à bien remplir les champs avec un astérisque. Ces champs sont obligatoires. Il vous restera plus qu'à valider en cliquant sur "Confirmer". (Cf. figure 13.)



*Figure 13 - Capture d'écran des inscriptions*

### 3.2.2. Connexion/Déconnexion

Pour pouvoir vous connecter, il vous suffit de cliquer sur le bouton “Connexion” se situant sur le menu de navigation. Vous aurez à remplir les champs et à cliquer sur “Connexion”. Pour pouvoir vous déconnecter, il suffit de cliquer sur le bouton “Déconnexion” se trouvant sur le menu de navigation à droite.

### 3.2.3. Son Profil

Vous pouvez accéder à votre profil dans le menu de navigation à droite après vous être connecté(e). À cet endroit vous pourrez voir vos informations personnelles. Mais vous avez également la possibilité de modifier certaines de vos données en confirmant avec votre mot de passe (le bouton bleu avec un crayon). Vous pouvez supprimer votre compte en cliquant sur le bouton rouge illustré d’une poubelle. Vous pouvez accéder à vos brouillons (fonctionnalité expliquée plus tard).

### 3.2.4. Poster une anecdote

Pour poster une anecdote, vous devez posséder un compte et être connecté(e).

Cliquez sur le bouton “Anecdotes” situé dans le menu. Un mini menu s'affiche, il faut ensuite cliquer sur “Poster une anecdote”. Vous aurez accès à un formulaire. Sachez que publier une anecdote est irréversible. Vous ne pourrez pas la modifier ou la supprimer. Il vous faudra contacter l’administrateur si besoin.

Vous avez la possibilité de continuer votre anecdote plus tard. Il suffit de cliquer en bas du formulaire sur “Enregistrer en brouillon”. Ainsi dans vos brouillons se trouvera l’anecdote que vous étiez en train d’écrire.

Attention, prenez connaissance que si vous souhaitez publier une anecdote avec un jeu qui n'est pas dans la base, celle-ci sera sauvegardée automatiquement dans vos brouillons. Vous devez attendre l’approbation du jeu avant de pouvoir la publier.

### 3.2.5. Ses Brouillons

Vous avez un espace pour vos brouillons. Celui-ci se trouve dans votre profil à l’endroit indiqué par la flèche rouge (cf. figure 14).

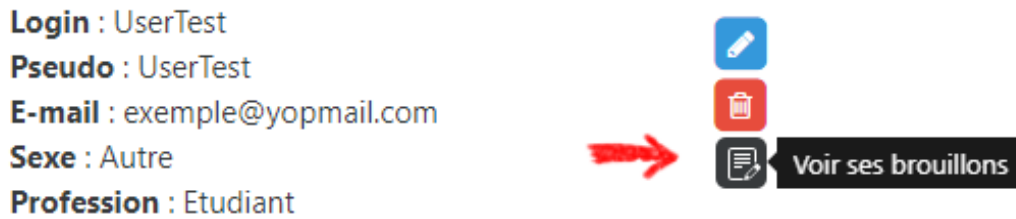


Figure 14 - Capture d'écran du profil

Cela vous mène vers une liste où sont stockées vos anecdotes en brouillon. Vous pouvez les modifier. Par la suite vous avez le choix entre les sauvegarder dans les brouillons ou les publier.

Vous pouvez également supprimer vos brouillons.

### 3.2.6. Interface d'une anecdote

Une anecdote se découpe en deux parties : la partie principale de l'anecdote (rectangle A) et la partie complémentaire de l'anecdote (rectangle B). (Cf. figure 15.)

La partie principale de l'anecdote se compose de :

- 1 - Le titre de l'anecdote
- 2 - L'auteur de l'anecdote
- 3 - Le texte de l'anecdote
- 4 - Les réactions de l'anecdote émis par les utilisateurs : lorsque vous réagissez en cliquant sur une émotion, celle-ci voit son fond virer à l'orange. En survolant une émotion vous verrez le nombre de réaction.

**Flèche rouge** - "Lire la suite" : si le texte de l'anecdote est trop long, seulement le début est visible. Il vous faudra cliquer sur ce bouton pour voir la suite de l'anecdote.

**Flèche violette** - "Partager" : permet de générer un lien de partage sur l'anecdote en question.

Dans la partie complémentaire de l'anecdote se trouve les différentes jauges d'émotions que l'auteur a ressenties lors de son expérience.



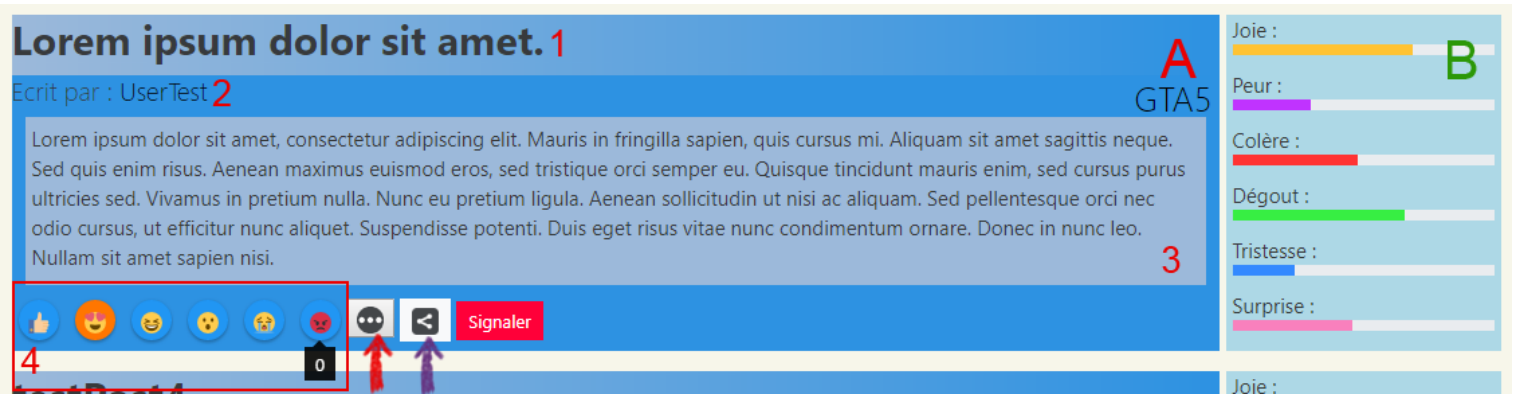


Figure 15 - Capture d'écran d'une anecdote

### 3.2.7. Signalement

Pour pouvoir signaler une anecdote vous devez être un membre du site.

Si le contenu d'une anecdote est offensant, vide ou autre chose qui nécessite d'être effacé, vous pouvez signaler cette anecdote. Il suffit de cliquer sur le bouton rouge "Signaler" se trouvant en bas de l'anecdote.

Un pop-up s'affiche par la suite (cf. figure 16). Il vous suffit de choisir le type du signalement, d'expliquer pourquoi vous signaler. Ce dernier n'est pas obligatoire mais recommandé pour aider l'administrateur à mieux gérer les signalements.

Faites attention car dans les explications vous ne pouvez pas aller à la ligne. En appuyant sur la touche "entrée" vous allez envoyer votre signalement, même si celle-ci est en cours de rédaction.

Vous ne pouvez que signaler une seule fois une anecdote.

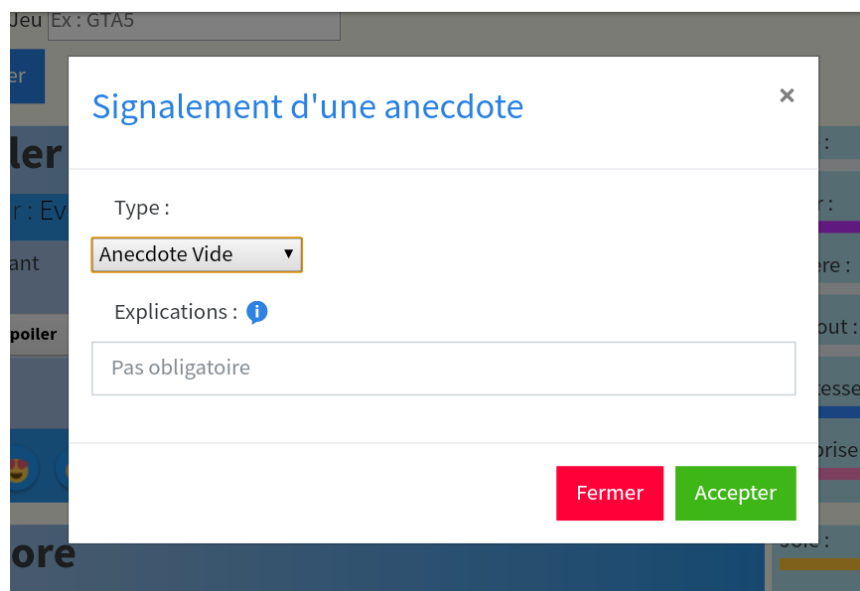


Figure 16 - Capture d'écran du pop-up signalement

### 3.2.8. Filtrer les anecdotes

Vous pouvez trier les anecdotes selon les filtres proposés par le site. Cette fonctionnalité se trouve en haut de la page “Anecdotes”, en bas de la navigation.

Vous avez un menu déroulant vous permettant de choisir entre les différents filtres. Sinon vous pouvez faire une recherche par jeu. (Cf. figure 17.)

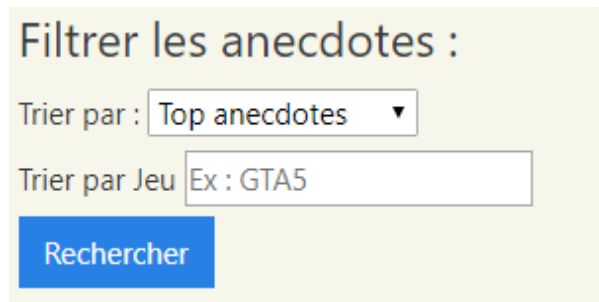


Figure 17 - Capture d'écran des filtres des anecdotes

### 3.2.9. Rechercher un jeu

Si vous souhaitez rechercher les jeux qui sont enregistrés dans le site, vous trouverez ces informations dans la page “Jeux” dans le menu.

Vous avez sur cette page, la liste de tous les jeux et vous pouvez faire une recherche via la barre de recherche. (Cf. figure 18.)

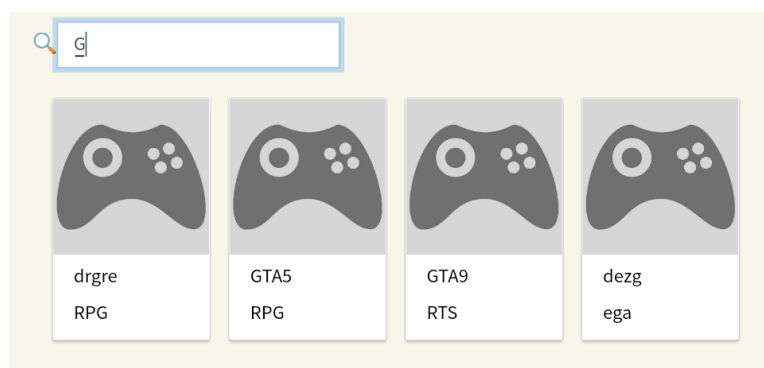


Figure 18 - Capture d'écran de la page “Jeux”

### 3.3. Manuel d'utilisation pour un administrateur

En tant qu'administrateur vous avez accès au panneau d'administration se trouvant à gauche dans le menu. Vous y trouverez une grande partie des fonctionnalités qui seront détaillées par la suite.

#### 3.3.1. Liste des utilisateurs

Dans le panneau d'administration, vous pouvez voir la liste des utilisateurs. Ainsi vous aurez la possibilité de visualiser tous les membres du site.

À partir de cette interface (cf. figure 19), vous voyez les différentes données des utilisateurs. Vous pouvez accéder à leur profil, modifier leur données mais également supprimer des comptes. Cette action est irréversible.



PSEUDO	LOGIN	SEXE	EMAIL	PROFESSION	MODÉRATION
 dieu Membre	dieu	Homme	corentin.grard@gmail.com	mangeur	  
 EviloxTheGod Admin	eviloX		corentin.grard@gmail.com	Dieu	  
 MorganeuhLaMeilleure Membre	Morganeuh	Femme(à prouver)	mornychu@gmail.com	Chômeur	  
 NightYano Admin	NightYano		davide.martins@hotmail.fr	Jesus	  
 user Membre	user	user	paul.amblard325@gmail.com	user	  
 UserTest Membre	UserTest	Autre	exemple@yopmail.com	Etudiant	  

Figure 19 - Capture d'écran de la liste des utilisateurs

#### 3.3.2. Générer Rapport

Vous pouvez générer un rapport avec différentes options par rapport au contenu du site tel que les anecdotes. Pour y accéder, aller sur le panneau d'administration et cliquer sur "Générer Rapport". (Cf. figure 20.)

Si vous souhaitez extraire toutes les anecdotes du site, ne touchez à rien et cliquez seulement sur le bouton "Télécharger". Vous allez télécharger un fichier Excel contenant toutes les anecdotes avec les différentes informations reliées à celles-ci.

Vous pouvez extraire des anecdotes selon leur jeu, leur catégorie, leur date de publication mais aussi par rapport aux émotions.

Pour les jauges des émotions, sachez que les anecdotes générées correspondent à une valeur des émotions supérieures à celle sélectionnée. De base, les anecdotes sélectionnées sont supérieures à zéro.

Un conseil, ouvrez le fichier téléchargé avec libre office. Cela vous confère une bonne visibilité des données.

Choisir les options pour le rapport :

Jeu  
Ex : GTAS

Catégorie Tous ▼

Date début  
jj/mm/aaaa

Date fin  
jj/mm/aaaa

Joie 0    Peur 0    Colère 0    Dégoût 0    Tristesse 0    Surprise 0

Annuler Télécharger

Figure 20 - Capture d'écran de "Générer Rapport"

### 3.3.3. Validation des jeux

Les utilisateurs vont proposer de nouveaux jeux pour alimenter la base de données. Vous pourrez les valider ou les refuser via la page "Validation des jeux" qui se trouve dans le panneau d'administration.

Pour traiter une demande, il suffit de survoler un jeu (cf. figure 21). Si vous souhaitez refuser une proposition, cliquez sur le bouton rouge. Si vous voulez valider une proposition, cliquez sur le bouton vert. La validation n'est pas finie, vous devez remplir un formulaire pour compléter les informations sur le jeu.

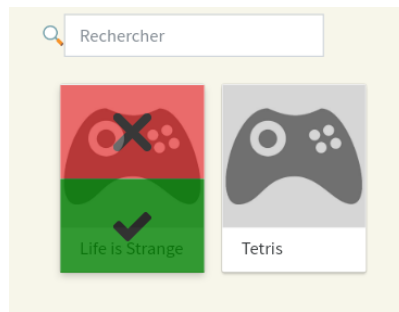


Figure 21 - Capture d'écran de la page "Validation des jeux"

Vous devez impérativement cliquer sur la case "Jeu validé" indiquée par la flèche rouge (cf. figure 22). Pour valider le jeu, il vous faut cliquer sur le bouton "Accepter".


Modifier un jeu :

Nom du jeu

Life is Strange

Catégorie

Ex : RPG

Jeu validé ☐ 

Description

Rich text editor toolbar with icons for undo, redo, bold, italic, strikethrough, link, unlink, list, indent, outdent, quote, source, and others.

Annuler Accepter

Figure 22 - Capture d'écran du formulaire d'un jeu

### 3.3.4. Gérer les signalements

Pour accéder à la gestion des signalements des anecdotes émis par les utilisateurs, il faut aller dans le panneau d'administration et cliquer sur "Gérer les signalements".

Vous trouverez une liste des signalements par anecdote. Vous pouvez voir le détail des signalements en cliquant sur "Détail" indiqué par la flèche verte (cf. figure 23). Ainsi vous avez

la liste des différents signalements émis sur l'anecdote en question (cf. figure 23). Si jamais le ou les signalements ne sont pas justifiés, vous pouvez à partir du “Détail” supprimer ce signalement en cliquant sur le bouton rouge “Supprimer signalement”.

Si les signalements sont justifiés et donc que l'anecdote doit être supprimée, il faut cliquer sur l'identifiant de l'anecdote indiqué par la flèche rouge (cf. figure 24). Vous aurez accès à l'anecdote, vous pourrez ainsi voir son contenu. Pour supprimer l'anecdote, il suffit de cliquer sur “Supprimer” en bas de l'anecdote.

IDENTIFIANT DE L'ANECDOTE	TITRE DE L'ANECDOTE	NOMBRE DE SIGNALEMENTS	
5a539176947a1	Lorem ipsum dolor sit amet.	2	 <a href="#">Détail</a>
5a5393429a8b3	Lorem ipsum dolor sit amet.	1	<a href="#">Détail</a>

Figure 23 - Capture d'écran de la liste des signalements

LOGIN	TYPE DE SIGNALEMENT	DESCRIPTIF DU SIGNALEMENT
Morganeuh	ContenuOff	Lorem ipsum dolor sit amet.
UserTest	Lorem	
<a href="#">Retour à la liste des signalements</a>		<a href="#">Supprimer signalement</a>

Figure 24 - Capture d'écran du détail des signalements

### 3.3.5. Interface d'une anecdote

Sur l'anecdote, vous avez accès à deux autres boutons : “Modifier” et “Supprimer” (cf. figure 25).

“Modifier” vous dirige vers un formulaire où vous pourrez modifier l'anecdote.









“Supprimer” vous affiche un pop-up pour vous prévenir que supprimer cette anecdote est irréversible.

## Lorem ipsum dolor sit amet.

Ecrit par : Morgane

GTA9

Lorem ipsum dolor sit amet.

[Signaler](#)
[Modifier](#)
[Supprimer](#)

Joie :

Peur :

Colère :

Dégout :

Tristesse :

Surprise :

Figure 25 - Capture d'écran d'une anecdote côté administrateur

## 4. Gestion de projet

### 4.1. Démarche personnelle

Pour réaliser ce projet nous avons dû fournir une grande quantité de travail. Nous avons donc essayé de rester un maximum structuré pour réaliser le projet. C'est pour cela que nous avons travaillé avec les méthodes agiles. Nous avons donc commencé par voir avec notre tuteur quelles sont ses attentes et quel est le but du projet. À partir de là, nous en avons extrait des user stories. Ainsi nous avons réalisé cinq sprints durant le projet, les sprints n'avaient pas tous la même durée car certaines semaines nous ne pouvions pas autant nous impliquer dans le projet que d'autres. À la fin de chaque sprint nous faisons une réunion avec le tuteur où nous lui présentons ce que nous avons réalisé et par la suite il validait ou refusait nos stories. Malheureusement certaines semaines comme pendant les vacances, faire des réunions était impossible donc nous lui communiquions notre travail par email afin qu'il le vérifie. Durant ce projet nous avons essayé de tenir le plus au courant possible notre tuteur pour être sûr que le projet avançait dans la bonne direction.

Pour ce qui est de la répartition et de la réalisation du travail dans le groupe nous nous sommes organisés de la manière suivante. Au début de chaque sprint nous faisons une petite réunion où nous décidions de ce qui devait être fait durant le sprint. Ensuite si nous étions seuls sur une story alors nous réalisons le travail et lorsque celui-ci était fini on le présentait aux autres membres pour confirmation. Mais la plupart du temps nous travaillions ensemble car nous avons organisé pleins de petite réunion à distance ou on se regroupait pour travailler chacun sur sa partie. Cela nous permettait de nous motiver à travailler.

Nous allons maintenant voir les outils utilisés pour communiquer durant ce projet. Comme énoncé plus tôt nous avons utilisé les emails pour communiquer avec notre tuteur. Nous avons fait usage des emails pour faire les comptes rendus de sprint ainsi que pour poser des questions sur le projet. Pour la communication interne du groupe nous utilisons Messenger<sup>8</sup> pour parler du projet, de notre avancement, de la planification des réunions... Nous avons aussi beaucoup utilisé des logiciels de communication vocale tels que Discord ou TeamSpeak pour faire nos réunions de travail ou de planification de sprint. Enfin pour le partage de fichier nous avons utilisé Google Drive.

Pour conclure durant le projet nous avons dû chercher beaucoup d'informations car nous avons travaillé sur des technologies que nous ne connaissions pas encore tel que le JavaScript/JQuery. Pour trouver toutes les données nécessaires à la réalisation de notre projet nous avons utilisé principalement les recherches internet. Notre manque de connaissance étant uniquement par rapport à des langages informatiques, internet suffisait amplement à trouver les informations nécessaires. Vous pourrez retrouver les informations utilisés dans notre site dans

---

<sup>8</sup> Un système de messagerie instantanée créé par la société Facebook (Wikipédia, 4 mai 2016)

la bibliographie. Nous avons aussi dû aussi demander de l'aide à des gens, vous pourrez retrouver ces personnes dans les remerciements.

## 4.2. Planification des tâches

Pour réaliser nos sprints nous nous sommes aidé de l'outil Trello (cf. figure 26), nous avons créé un tableau pour chaque sprint avec à l'intérieur ce que l'on veut faire, ce que l'on a fait, et ce que le client nous a validé.

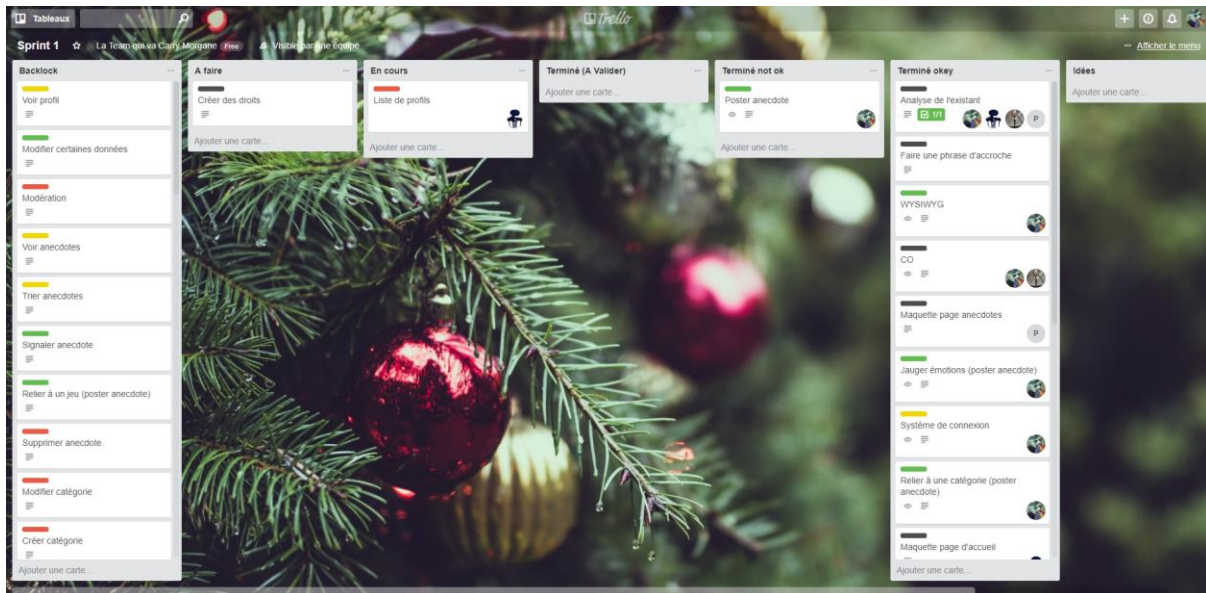


Figure 26 - Capture d'écran de l'outil Trello

Au début du projet nous avons déterminé plusieurs fonctionnalités telles qu'un système de niveau de d'expérience ou bien un système de commentaire. Mais en réalisant le projet nous nous sommes rendu compte que nous ne pourrions pas tout réaliser. Mais grâce aux méthodes agiles nous avons réalisé les fonctionnalités les plus importantes et avons abandonné les stories secondaires.

Durant ce projet nous avons perdu du temps sur des problèmes. Par exemple quand nous avons commencé à réaliser les jeux nous nous sommes dit qu'utiliser une base de données de jeux vidéo via une API\* pourrait être une bonne idée nous avons mis en place tous le système et cela fonctionnait bien. Mais un problème subsistait, l'API nous permettait de faire seulement trois mille requêtes par mois. Donc lors de la réunion de fin de sprint, le client nous a refusé cette story et nous nous sommes donc rabattus sur des jeux entrés manuellement et stockés dans notre base de données.

Un autre problème que nous avons rencontré durant ce projet fut par rapport aux triggers<sup>9</sup> de MySQL. Nous avons voulu implémenter des triggers dans la base de données pour

<sup>9</sup> Appelés également "déclencheurs"



mettre à jour automatiquement les compteurs de réactions. Malheureusement après beaucoup d'essais et de recherches internet, nous avons demandé de l'aide à M. Mesmoudi, un élève de l'IUT ayant eu le même problème que nous. Il nous a répondu qu'il valait mieux ne pas toucher aux triggers MySQL et qu'il n'a pas aussi réussi à les faire marcher. Nous nous sommes donc rabattus sur une mise à jour des tuples via PHP.

### 4.3. Bilan critique

Par rapport aux stories que nous avons prévu de réaliser, nous avons dû abandonner un certain nombre d'entre elles car nous n'avons pas assez de temps pour les faire.

Nous avons donc dû faire impasse des fonctionnalités assez importantes telles que les commentaires car cela nous aurait demandé beaucoup de temps à réaliser.

Une autre fonctionnalité que nous avons dû abandonner est "traiter les données" c'est-à-dire donner à l'administrateur une page où il pourrait voir de nombreuses statistiques par rapport aux anecdotes que les utilisateurs ont postées. Le fait que cette story ne soit pas réalisée n'est pas très grave car le client pourra accéder à ces différentes données via la fonction "Générer un rapport" où toutes les données sont extraites sous format CSV<sup>10</sup>.

De plus nous n'avons pas réalisé la fonction "créer un synonyme pour un jeu vidéo". Cette fonctionnalité n'était pas très importante et pour la réaliser il aurait fallu fournir un travail assez conséquent.

Et enfin nous n'avons pas réalisé pleins d'idées que nous avons pu avoir tel que le système de badge pour un profil avec des niveaux et de une expérience pour chaque utilisateurs. Mais ces fonctionnalités étaient plus des idées que nous avons évoquées avec le client et ne se trouvaient pas dans notre backlog.

Concernant la modification du code du site, le fait d'utiliser un modèle MVC, qui est un modèle répandu, facilite la reprise ou la modification des différentes parties du site par un tiers. Celui-ci saura où et comment ajouter ou modifier une fonctionnalité. Nous avons aussi commenté les fonctions importantes notre code pour une reprise en main facile du site.

---

<sup>10</sup> "Format informatique ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules" (Wikipédia, 17 octobre 201124)

## 5. Conclusion

Après avoir mené à terme ce projet, une rétrospective est importante afin de pouvoir prendre du recul vis-à-vis de ce qu'on a effectué.

Tout d'abord, analysons notre taux de complétion du projet. La plupart des principales fonctionnalités demandées par le client ont été terminées. En effet, comme expliqué précédemment, seul le système de commentaire, une partie de la modération et du traitement des données n'ont pas pu être réalisés par manque de temps. Notre efficacité est au final de quarante-cinq stories réalisées sur soixante-et-un ce qui représente 74% des stories et ce nombre peut-être expliqué de manière simple. La plupart des stories non réalisées sont en lien avec le système de commentaires qui a été découpé en de nombreuses stories d'effort faible. Donc nous avons fait six-cent-huit sur sept-cent-dix-huit points d'effort répartis sur les différentes stories ce qui représente 85% de l'effort.

On remarque que ce qui n'a pas été fini n'a vraiment pas été mis sur le côté par difficulté ou manque de travail mais en grande partie à cause du manque de temps.

Dans le futur, on pourrait améliorer plusieurs points. Tout d'abord la finition des différentes stories initialement prévues. Ensuite il est possible de compléter le projet par l'ajout de plusieurs fonctionnalités dont le client avait fait part, comme par exemple un système de badge pour les utilisateurs afin de les fidéliser.

Durant ce projet, afin de répondre aux attentes du client il a été nécessaire d'utiliser des technologies que nous ne connaissions pas, et au vu des contraintes de temps, cela nous a appris à rapidement s'approprier de nouvelles technologies. Ce projet a, de par ces conditions, également renforcé notre autonomie. De plus cela nous a aussi permis d'aborder le travail de groupe de manière plus concrète que durant les projets précédents, nous permettant ainsi de compléter notre expérience du travail en équipe.

## 6. Bibliographie

- [1]  
jQuery F.- jquery.org, :« contains() Selector | jQuery API Documentation ». .
- [2]:  
« nth-child », *Mozilla Developer Network*. [En ligne]. Disponible sur:  
<https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child>. [Consulté le: 08-janv-2018].
- [3]  
jQuery F.- jquery.org, .« parents() | jQuery API Documentation ». .
- [4]  
M. O. contributors Jacob Thornton, and Bootstrap, « Bootstrap ». [En ligne].  
Disponible sur: <https://getbootstrap.com/>. [Consulté le: 09-janv-2018].
- [5]  
« Bootstrap snippet e commerce product list ». [En ligne]. Disponible sur:  
<https://bootdey.com/snippets/view/e-commerce-product-list>. [Consulté le: 08-janv-2018].
- [6]  
« Bootstrap snippet table user list ». [En ligne]. Disponible sur:  
<https://bootdey.com/snippets/view/table-user-list>. [Consulté le: 08-janv-2018].
- [7]  
« Bootswatch: Cosmo ». [En ligne]. Disponible sur: <https://bootswatch.com/cosmo/>.  
[Consulté le: 08-janv-2018].
- [8]  
« CKEditor Ecosystem | CKEditor.com ». [En ligne]. Disponible sur:  
<https://ckeditor.com/>. [Consulté le: 09-janv-2018].
- [9]  
« Créer un scroll infini en jQuery - Buzut ». [En ligne]. Disponible sur:  
<https://buzut.fr/creer-un-scroll-infini-en-jquery/>. [Consulté le: 08-janv-2018].
- [10]  
« CSS “Always on the bottom” Footer », *CodePen*. [En ligne]. Disponible sur:  
<https://codepen.io/cbracco/details/zekgx>. [Consulté le: 08-janv-2018].
- [11]  
« cURL : Dialogue avec une API en PHP », *Max-Koder*, 05-mai-2017. .
- [12]  
« Delete free vector icons designed by Freepik », *Flaticon*. [En ligne]. Disponible sur:  
[https://www.flaticon.com/free-icon/delete\\_32178](https://www.flaticon.com/free-icon/delete_32178). [Consulté le: 08-janv-2018].
- [13]

« Get Header from PHP cURL response - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/41978957/get-header-from-php-curl-response>. [Consulté le: 08-janv-2018].

[14]

« How to Style Input Type Range in Chrome, Firefox, and IE | Brenna O'Brien ». [En ligne]. Disponible sur: <http://brennaobrien.com/blog/2014/05/style-input-type-range-in-every-browser.html>. [Consulté le: 08-janv-2018].

[15]

« HTML Form Elements ». [En ligne]. Disponible sur: [https://www.w3schools.com/html/html\\_form\\_elements.asp](https://www.w3schools.com/html/html_form_elements.asp). [Consulté le: 08-janv-2018].

[16]

« http - Can PHP cURL retrieve response headers AND body in a single request? - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/9183178/can-php-curl-retrieve-response-headers-and-body-in-a-single-request>. [Consulté le: 08-janv-2018].

[17]

« javascript - Ajax get a return value from php? - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/15126600/ajax-get-a-return-value-from-php>. [Consulté le: 08-janv-2018].

[18]

« javascript - Disabling the button after once click - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/2323948/disabling-the-button-after-once-click>. [Consulté le: 08-janv-2018].

[19]

« javascript - How to remove an element slowly with jQuery? - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/1807187/how-to-remove-an-element-slowly-with-jquery>. [Consulté le: 08-janv-2018].

[20]

jQuery F.- jquery.org, « jQuery ». [En ligne]. Disponible sur: <https://jquery.com/>.

[21]

« jquery - Close Bootstrap Modal - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/16493280/close-bootstrap-modal>. [Consulté le: 08-janv-2018].

[22]

« jQuery AJAX Autocomplete – Country Example ». [En ligne]. Disponible sur:  
<http://phpspot.com/jquery/jquery-ajax-autocomplete-country-example/>. [Consulté le: 08-janv-2018].

[23]

« jQuery attr() Method ». [En ligne]. Disponible sur:  
[https://www.w3schools.com/jquery/html\\_attr.asp](https://www.w3schools.com/jquery/html_attr.asp). [Consulté le: 08-janv-2018].

[24]

« JSON.parse() », *Mozilla Developer Network*. [En ligne]. Disponible sur:  
[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets\\_globaux/JSON/parse](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/JSON/parse). [Consulté le: 08-janv-2018].

[25]

« Le contenu généré, le numérotage automatique et les listes ». [En ligne]. Disponible sur:  
<http://www.yoyodesign.org/doc/w3c/css2/generate.html#lists>. [Consulté le: 08-janv-2018].

[26]

« Le Pattern MVC », *OpenClassrooms*. [En ligne]. Disponible sur:  
<https://openclassrooms.com/courses/apprendre-asp-net-mvc/le-pattern-mvc>. [Consulté le: 08-janv-2018].

[27]

« Make jQuery :contains Case-Insensitive », *CSS-Tricks*. [En ligne]. Disponible sur:  
<https://css-tricks.com/snippets/jquery/make-jquery-contains-case-insensitive/>. [Consulté le: 08-janv-2018].

[28]

« MUI - Material Design Buttons ». [En ligne]. Disponible sur:  
<https://www.muicss.com/docs/v1/css-js/buttons>. [Consulté le: 08-janv-2018].

[29]

« MUI - Material Design CSS Framework ». [En ligne]. Disponible sur:  
<https://www.muicss.com/>. [Consulté le: 09-janv-2018].

[30]

« Pagination ». [En ligne]. Disponible sur:  
<https://igdb.github.io/api/references/pagination/>. [Consulté le: 08-janv-2018].

[31]

« php - echo inside a while loop? - Stack Overflow ». [En ligne]. Disponible sur:  
<https://stackoverflow.com/questions/22124235/echo-inside-a-while-loop>. [Consulté le: 08-janv-2018].

[32]

« php ckeditor can't set php data - Stack Overflow ». [En ligne]. Disponible sur: <https://stackoverflow.com/questions/18186274/php-ckeditor-cant-set-php-data>. [Consulté le: 08-janv-2018].

[33]

« PHP: Exemples - Manual ». [En ligne]. Disponible sur: <http://php.net/manual/fr/curl.examples.php>. [Consulté le: 08-janv-2018].

[34]

« PHP: readfile - Manual ». [En ligne]. Disponible sur: <http://php.net/manual/fr/function.readfile.php>. [Consulté le: 08-janv-2018].

[35]

« [Résolu] Afficher la valeur d'un input type range en temps réel », *Alsacrérations*. [En ligne]. Disponible sur: <https://forum.alsacreations.com/topic-5-75012-1-Afficher-la-valeur-dun-input-type-range-en-temps-reel.html>. [Consulté le: 08-janv-2018].

[36]

« Show non-public members with json\_encode », *Jonathan D. Johnson*. [En ligne]. Disponible sur: [http://localhost:4000/show-non-public-members-with-json\\_encode/](http://localhost:4000/show-non-public-members-with-json_encode/). [Consulté le: 08-janv-2018].

[37]

« Simple Share Buttons ». .

[38]

Paomedia, « “small-n-flat” by Paomedia », *Iconfinder*. [En ligne]. Disponible sur: [https://www.iconfinder.com/icons/285648/group\\_user\\_icon](https://www.iconfinder.com/icons/285648/group_user_icon). [Consulté le: 08-janv-2018].

[39]

« Tag HTML - Balise SPAN ». [En ligne]. Disponible sur: <http://www.startyourdev.com/html/tag-html-balise-span>. [Consulté le: 08-janv-2018].

[40]

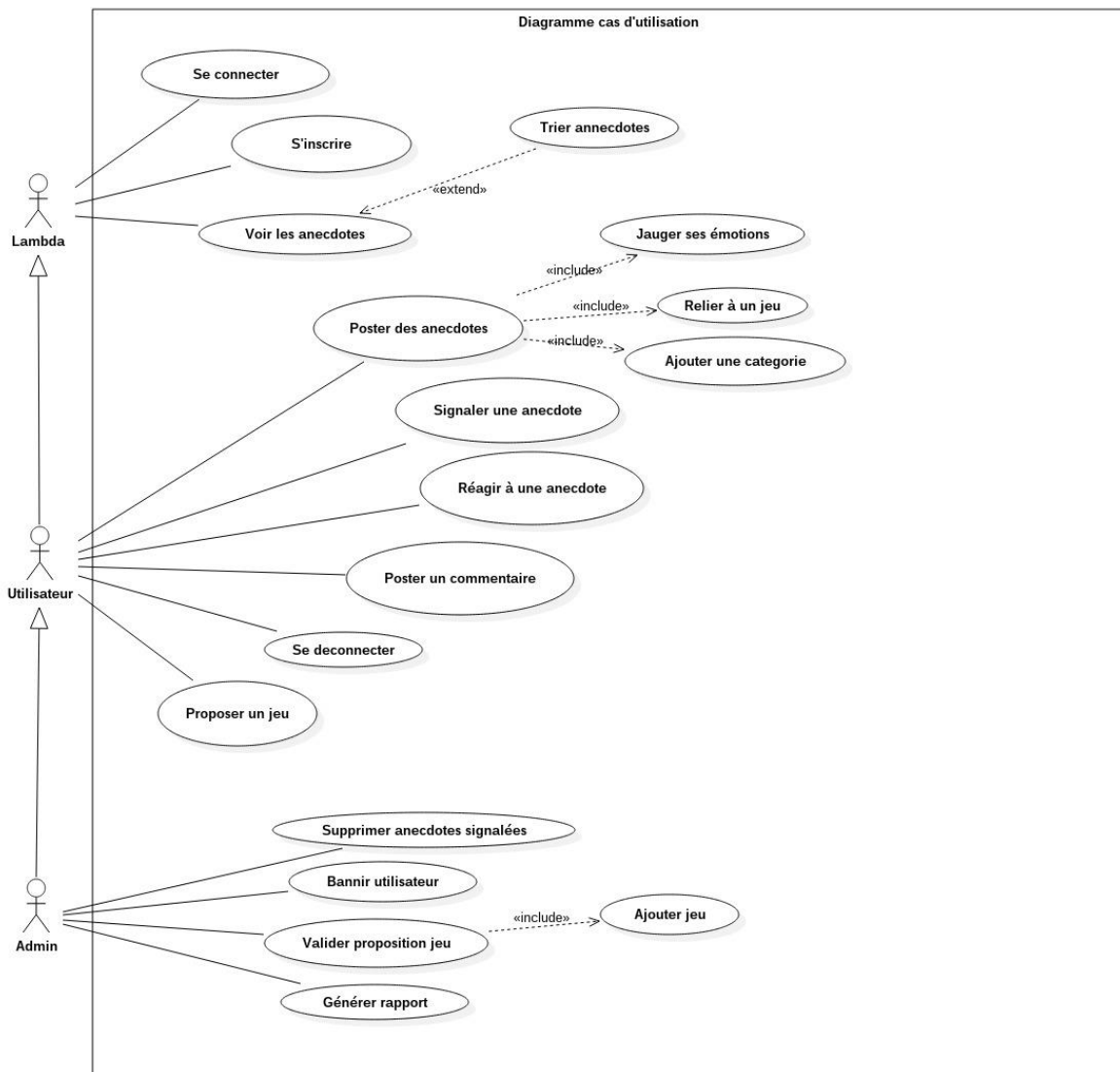
M. O. contributors Jacob Thornton, and Bootstrap, « Tooltips ». [En ligne]. Disponible sur: <https://getbootstrap.com/docs/4.0/components/tooltips/>. [Consulté le: 08-janv-2018].

[41]

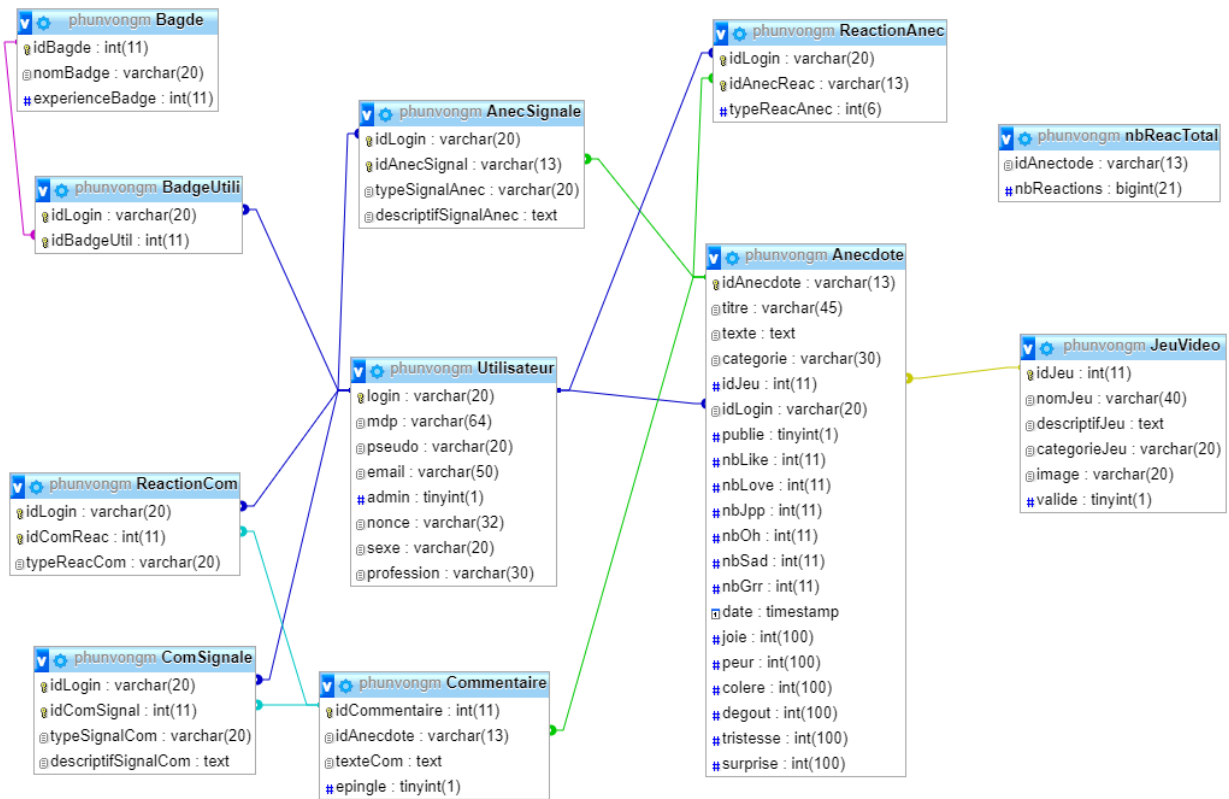
« Tryit Editor v3.5 ». [En ligne]. Disponible sur: [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_validation\\_js](https://www.w3schools.com/js/tryit.asp?filename=tryjs_validation_js). [Consulté le: 08-janv-2018].

## 7. Annexes techniques

### 7.1. Annexe 1 : diagramme des cas d'utilisation

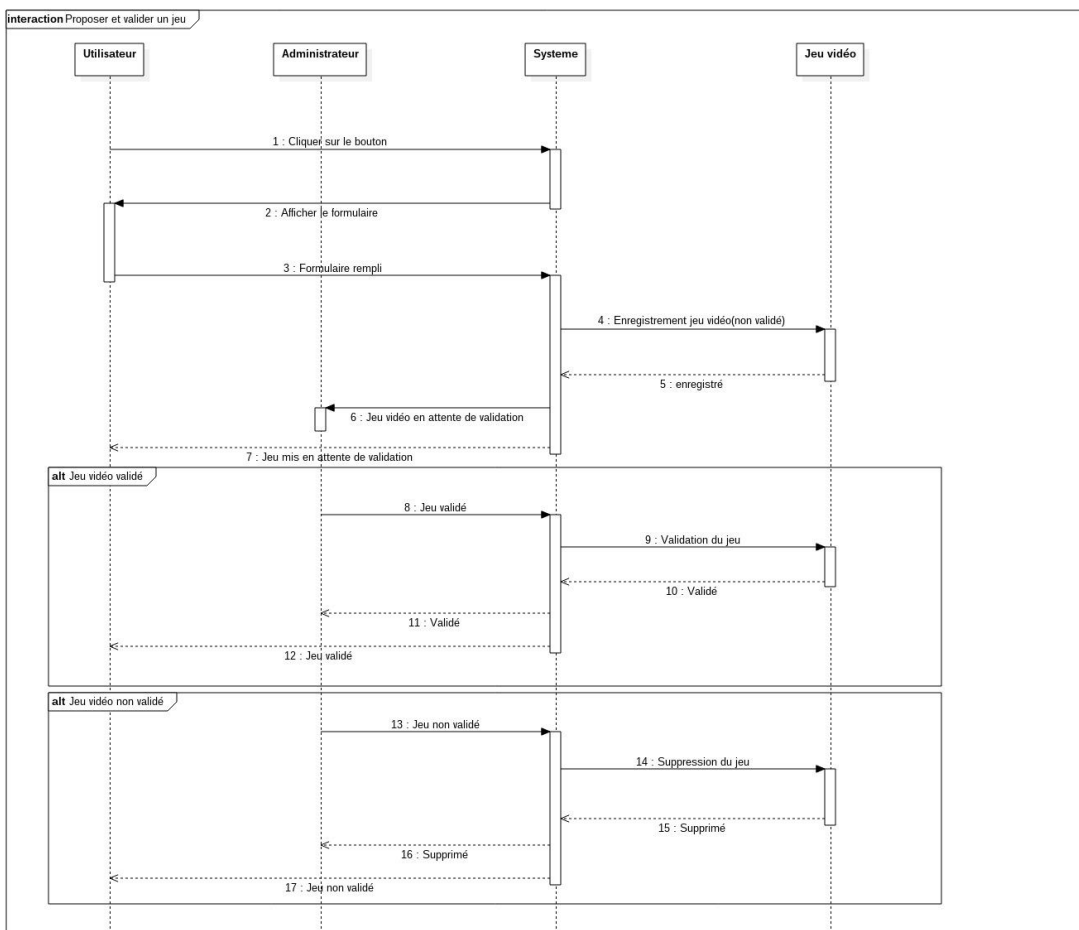
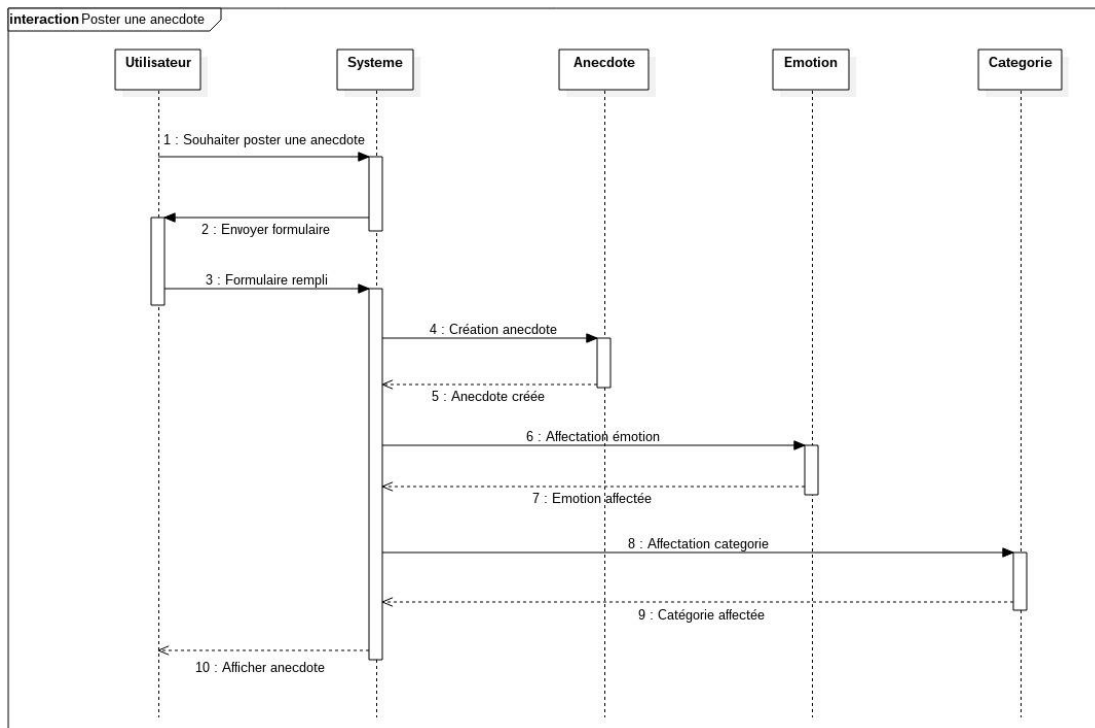


## 7.2. Annexe 2 : schéma du modèle relationnel

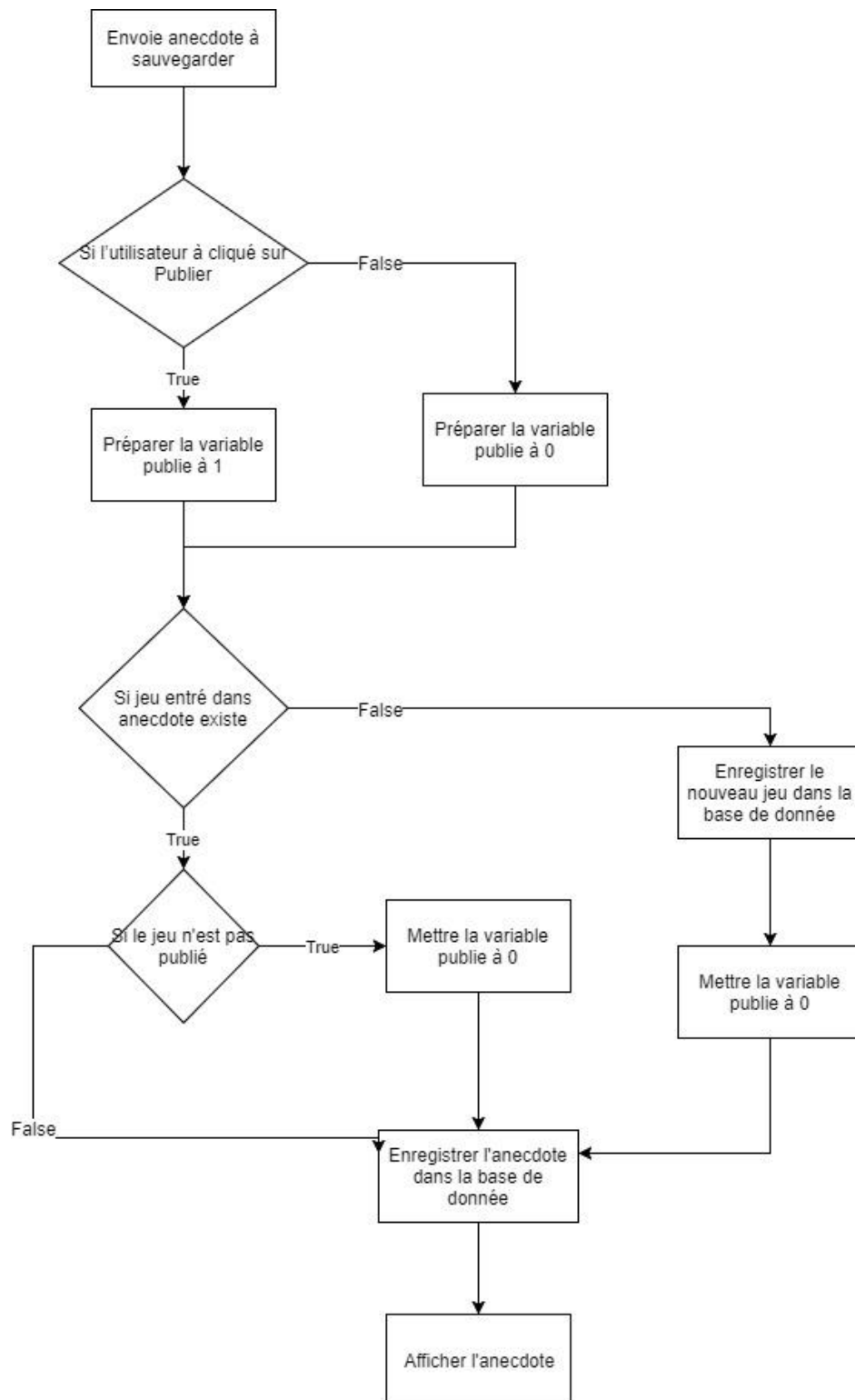




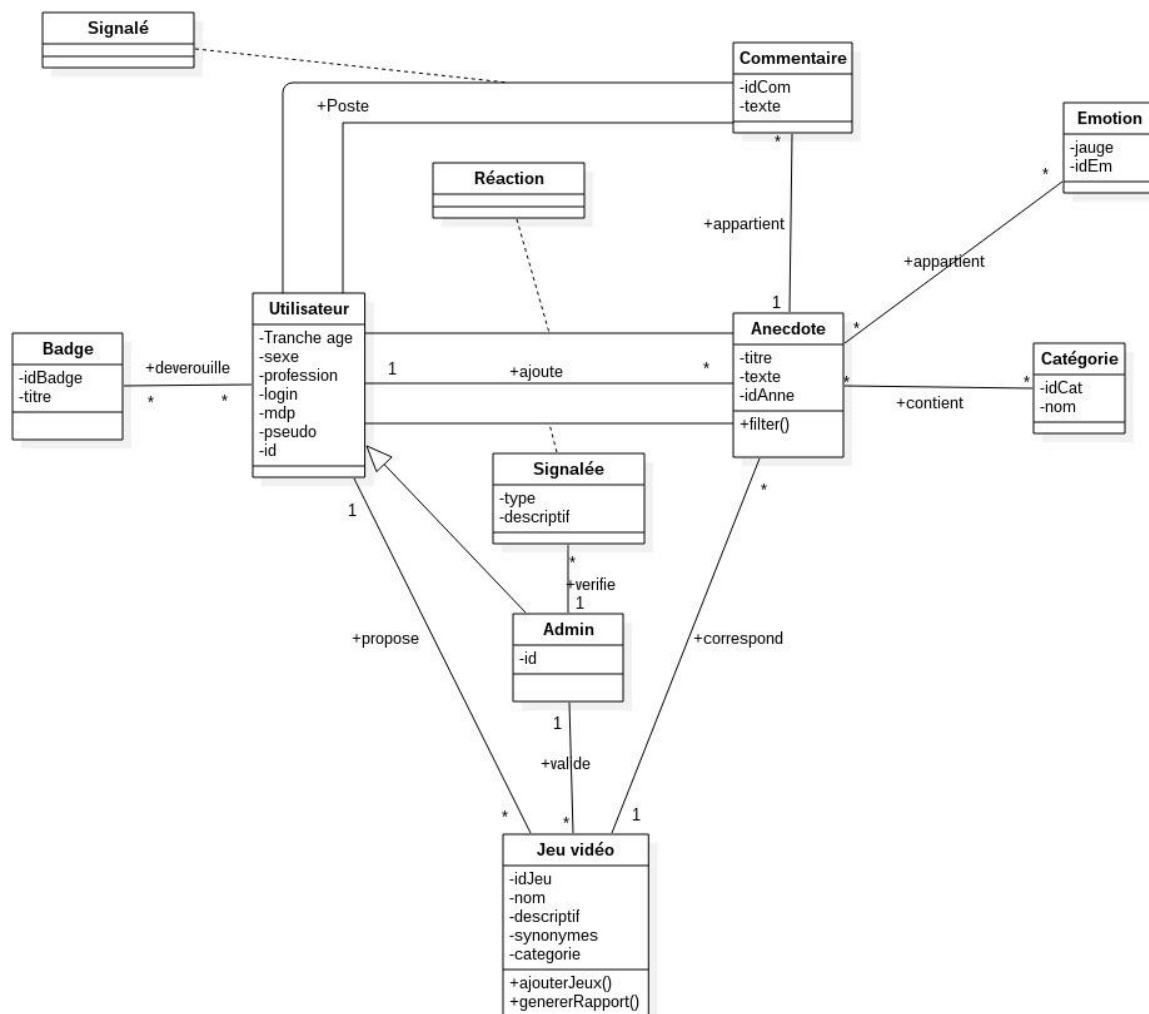
### 7.3. Annexe 3 : diagrammes de séquence



#### 7.4. Annexe 4 : diagramme enregistrement d'une anecdote



## 7.5. Annexe 5 : ancien diagramme de classe



**Résumé :**

Le projet “Recueil d’expérience vidéoludique” est un site où un utilisateur pourra exprimer le ressenti qu’il a eu en jouant à un jeu vidéo ou bien une anecdote qu’il a eu en jouant à un jeu vidéo. Le site comprend 3 rôles : un utilisateur non connecté qui pourra voir les anecdotes, un utilisateur connecté pourra poster des anecdotes et enfin un administrateur qui pourra faire de la modération et télécharger des rapports sur les anecdotes.

Le site a été développé en PHP pour la partie serveur, la base de données est un serveur MySQL et enfin le site utilise la bibliothèque JQuery pour le coté client.

**Mots-clés :** Site Web, Anecdote, Partage, MySQL, PHP, JQuery

**Abstract:**

The “Recueil d’expériences vidéoludique” is a website where a user can express the feeling he had while playing a video game or an anecdote he had while playing a video game. The website has 3 different roles: a not logged in user who can see the anecdotes, a logged in user who can post anecdotes and finally an administrator who can do moderation and download reports about the anecdotes.

The website uses PHP for the server part, MySQL for the database server and finally the JQuery library for the client side.

**Keywords :** Website, Anecdote, Sharing, MySQL, PHP, JQuery