

# Projet Duel sur la Toile Rapport

AUBRY Erwan  
BEAUV AIS Antoine  
BIZOT Jérôme  
CHRISTOPHE Marie  
CIZEAU Valentin  
COLLET Axel  
FRUCHART Théodore



INFORMATIQUE

Année universitaire : 2017 – 2018

## Table des matières

I.Introduction.....	4
II.Analyse de la base de données.....	5
1.Dictionnaire de données.....	5
2.Modèle Conceptuel de Données.....	8
3.Script de la base de données.....	13
4.Requêtes.....	17
III.Module “Joueur et Administrateur”.....	19
1.Joueur.....	19
a.Diagramme de cas d'utilisation.....	20
b.Description.....	20
i.Inscription.....	20
ii.Connexion.....	22
iii.Création d'une partie.....	24
iv.Rejoindre partie.....	25
v.Répondre invitation à jouer.....	26
vi.Communication.....	27
vii.Demande d'amis.....	28
viii.Répondre demande d'amis.....	29
2.Administrateur.....	30
a.Diagramme de cas d'utilisation .....	31
b.Description.....	32
i.Activer utilisateur.....	32
ii.Désactiver utilisateur.....	33
iii.Consulter statistiques.....	34
iv.Gérer les jeux.....	34
v.Ajouter un jeu.....	37
vi.Mettre à jour un jeu.....	39
IV.Module jeu.....	40
1.Puissance 4.....	41
a.Diagramme de cas d'utilisation.....	42
b.Description.....	42
2.Mastermind.....	44
a.Diagramme de cas d'utilisation.....	45
b.Description.....	45
i.Rejoindre une partie.....	46
ii.Lancer une manche.....	47
iii.Quitter une partie .....	48
V.Conclusion.....	50
VI.Annexes.....	51
1.Annexes I.H.M.....	52
2.Fiches individuelles.....	56
a.Deuxième phase.....	56
b.Troisième phase.....	61
c.Quatrième phase.....	64

## I. Introduction

Le travail *Duel sur la Toile* est un projet de développement informatique organisé au sein de l'I.U.T. d'Orléans.

En effet, le client fictif *Duel sur la Toile* réalise un appel d'offres pour le développement d'une plateforme en ligne permettant à des joueurs du monde entier de s'affronter exclusivement en 1 contre 1 sur des jeux dédiés. La réalisation de l'application est soumise à plusieurs contraintes :

- Le langage JAVA sera utilisé pour la programmation ;
- Les programmes Java seront mis en relation via une base de données MySQL.

Des sociétés de jeux extérieures pourront proposer des jeux au client *Duel sur la Toile*. Ainsi, ils pourront être ajoutés sur la plateforme par les gestionnaires assignés. Les jeux se joueront soit :

- en différentes manches où le joueur devra obtenir le maximum de points face à son adversaire pour les remporter.
- en différentes manches où le joueur doit être plus rapide que l'ennemi, comme par exemple le jeu du *Mastermind* qui sera implémenté.
- en tour par tour, comme le *Puissance 4*.

La plateforme nécessitera l'enregistrement préalable de l'utilisateur. Celui-ci disposera d'une liste d'amis et pourra ainsi défier ceux-ci ou demander un adversaire sur l'application sur un jeu en particulier. Dans le deuxième cas, l'adversaire sera choisi s'il a fait la même demande. Les joueurs recevront des notifications sur l'état de leurs parties en cours et pourront consulter leurs statistiques. De plus, les utilisateurs seront en mesure de communiquer entre eux via un système de messagerie.

Les différents aspects suivants doivent être réalisés dans le développement de l'application :

- une interface d'administration de la plateforme, qui permettra l'ajout, la mise à jour et la désactivation des jeux ainsi que la gestion des utilisateurs. Cette interface devra aussi être capable de générer des statistiques relatives aux jeux disponibles, comme le nombre de parties effectuées pendant un temps, ou le classement des meilleurs joueurs, et caetera...
- une interface joueur qui permettra l'accès à la plateforme pour les utilisateurs, notamment les jeux ainsi que la messagerie, ainsi qu'un profil résumant les parties en cours, les invitations reçues, les statistiques, et cetera...

Le rapport est divisé en quatre chapitres ; l'analyse de la base de données suivie par le module Administrateur et Utilisateur, puis du module Jeu et enfin de la conclusion.

## II. Analyse de la base de données

Cette partie est l'étape d'analyse préliminaire. C'est une partie très importante car elle nous a permis de prendre connaissance avec le projet et créer par nous-mêmes une solution adaptée aux différents problèmes que soulève le sujet. C'est durant ce chapitre que la base de données devait être constituée dans le système.

Pour cela, un dictionnaire de données a été réalisé. Un dictionnaire de données est une collection des informations nécessaires à la conception d'une base de donnée relationnelle.

Le MCD (Modèle Conceptuel des Données) a ainsi pu être créé. Il s'agit un schéma représentant les différentes tables ainsi que les associations constituées des données que nous avons préalablement définies dans le dictionnaire, ce qui permet de les mettre en relation dans notre système.

De fait, la réalisation des scripts de cette base de données permet sa création au niveau informatique et non seulement à un niveau abstrait tel le MCD.

Pour le tester nous devions aussi faire des requêtes précisées dans le sujet afin de voir si la base de données fonctionnait correctement.

### 1. Dictionnaire de données

Le dictionnaire de données a été organisée et discuté par tout le groupe supervisé par notre chef de projet.

Pour la mise en commun du dictionnaire des données, notre chef de projet a décidé pour notre groupe de l'utilisation de Google Drive de plus pour la création du dictionnaire des données notre chef de projet a pour cela choisi l'utilisation de Google Sheets.

Un dictionnaire de données permet de préparer le MCD que nous aurons par la suite. Ce dernier se compose des noms pour chaque donnée (le nom compris par le client), de la définition (le nom compris par le développeur), la structure (si c'est un entier, une chaîne de caractère,...), le type (élémentaire ou calculable), la quantification, un exemple pour chaque donnée et un commentaire. Ce tableau de données met sur la table toutes les données que l'on utilisera pour le projet.

L'utilisateur contient un pseudo comme clef primaire ainsi chaque pseudo sera unique.

Un utilisateur peut :

- correspondre à une partie et lui même correspond à un jeu.
- peut envoyer une invitation qui appartient à une partie qui correspond donc à l'utilisateur qui a envoyé l'invitation.
- appartenir à plusieurs conversation et composer plusieurs message par conversation.
- accéder à plusieurs statistique d'utilisateur.
- rédiger plusieurs rapport.
- consulter plusieurs statistique de la plateforme.
- être amis avec plusieurs autres utilisateurs.

Dictionnaire des Données						
Utilisateur						
Nom	Définition	Structure	Type	Quantification	Exemple	Commentaire
Pseudonyme de l'Utilisateur	pseudoUtilisateur	Varchar (50)	E	-	"UserDu45"	Le pseudo qui permet d'identifier une seule personne
Email de l'Utilisateur	mailUtilisateur	Varchar (50)	E	-	"jean.dupres@gmail.com"	Une adresse email pour vérifier l'utilisateur
Mot de passe de l'Utilisateur	passwdUtilisateur	Varchar (50)	E	-	"ze@sdqqaZ"	Permet de sécuriser l'accès au compte
Compte activé	activeUtilisateur	Bool	E	2	true.	Information si l'utilisateur est activé ou non
Rôle de l'Utilisateur	roleUtilisateur	Varchar (50)	E	2	"Admin"	Rôle de l'utilisateur
Jeu						
Identifiant d'un Jeu	idJeu	int	E	-	1 pour Puissance 4	L'identifiant d'un jeu
Nom du jeu	nomJeu	Varchar (50)	E	-	"Puissance 4"	Le nom d'un jeu en particulier
Description du Jeu	descJeu	Varchar (150)	E	-	"Ce jeux est un jeu de plateforme..."	La description d'un jeu en particulier
FichierJAR du Jeu	sourceJeu	blob	E	-	Puissance_4.jar	Le fichier source (en jar) d'un jeu
Type du Jeu	typeJeu	Varchar (50)	E	3	"Stratégie"	Le type d'un jeux
Disponibilité du Jeu	dispoJeu	Bool	C	2	True.	Information si le jeux est présent ou non
Invitation						
Identifiant de l'Invitation	idInvite	int	E	-	2156	Identifiant de l'invitation
Date de l'Invitation	dateInvite	date	E	-	"02/04/2018"	Date d'une invitation
Type de l'Invitation	typeInvite	Varchar (25)	E	2	"Jeux"	Type d'une invitation
Partie						
Identifiant de la partie	idPartie	int	E	-	6578959	Identifiant d'une partie
DatePartie	datePartie	date	E	-	"01/02/2018"	Date d'une partie jouer
Numéro Etape de la Partie	numEtapePartie	int	C	-	1	Le numéro d'étape d'une partie
Description de la Partie	infoJSONPartie	Varchar (50)	C	-	[id: 45, jeux: "puissance 4"]	Enregistrement des informations d'une partie
Rapport						
Identifiant du Rapport	idRapport	int	E	-	46574984	Identifiant d'un rapport
Champ Texte du Rapport	champTexteRapport	Varchar (125)	E	-	"Voici le nième rapport ...."	L'endroit où l'admin écrit son rapport
Message						
Identifiant du message	idMessage	int	E	-	256	L'identifiant du message
Destinataire du Message	DestMessage	Varchar (125)	E	-	Joueur1	Le destinataire du message
Corp du Message	corpsMessage	Varchar (50)	E	-	"Salut comment tu vas ?"	Le contenu du message
Message lu	estLuMessage	bool	E	-	true.	Information de si le message est lu ou non
Nombre de caractère	nbCaractMessage	int	C	-	200	Le nombre de caractère dans le corp du message
Statistique Utilisateur						
Identifiant de la statistique	idStat	int	E	-	656598741	Identifiant d'une statistique
Moyenne des points	avgPtsStat	float	C	-	256.25	Moyenne des points d'un joueur
Fréquentation d'un jeu	frecJeuStat	int	C	-	22	Nombre de fois que le joueur a jouer à un certain jeu
Conversation						
Identifiant de la conversation	idConv	int	E	-	25	Identifiant d'une conversation
Envoyer						
Destinataire d'un message	destinataire	Varchar (50)	E	-	Joueur1	Destinataire d'un message
Correspondre						
Score de l'utilisateur	score	int	C	-	45	Le score d'un utilisateur

Figure II.1 : Dictionnaire des données

La table "Conversation" n'est pas présente dans le nouveau MCD ainsi que le "Rapport".

Description :

- Nom : nom complet de l'attribut du MCD ;
- Définition : nom de l'attribut tel que vu dans le MCD ;
- Structure : le type de l'attribut du MCD ;

- Type : si l'attribut est calculable ou élémentaire ;
- Quantification : nombre maximum du même attribut en même temps ;
- Exemple : exemple d'entrée que nous pouvons trouver dans le MCD ;
- Commentaire : explication sur l'attribut.

## 2. Modèle Conceptuel de Données

Lors de la première phase, un M.C.D. (Modèle Conceptuel de Données) a été réalisé pour permettre la conception du script MySQL. Ce M.C.D. se base sur les différents points à respecter sur le sujet.

Un M.C.D. est un diagramme permettant d'illustrer la base de données d'un système. Ce diagramme se compose d'entités (correspondant aux tables) et de liaisons, qui lient deux voire davantage d'entités. Ce diagramme met en avant les clés primaires et étrangères, mais aussi les données et leurs types.

Ce M.C.D. a été conçu avec le logiciel Jmerise, un outil en Java™ permettant de réaliser facilement un M.C.D. et d'en générer un M.L.D. ainsi que le script MySQL. Le logiciel Jmerise a été utilisé car il est particulièrement simple à utiliser, en plus d'être très complet.

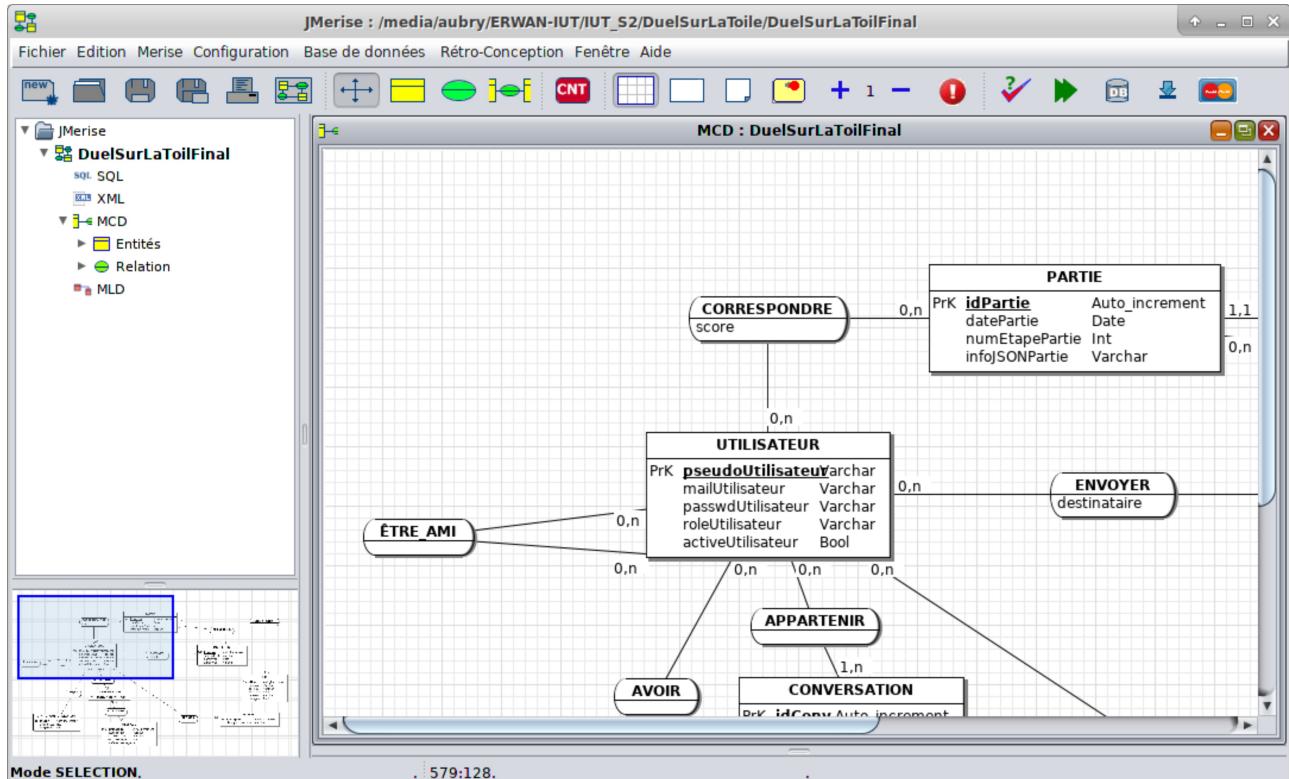


Figure II.2: JMerise

Le M.C.D. a pour noyau les utilisateurs couplés avec des fonctions différentes, comme avoir des amis, avoir des conversations, correspondre à des parties, envoyer des messages, etc...

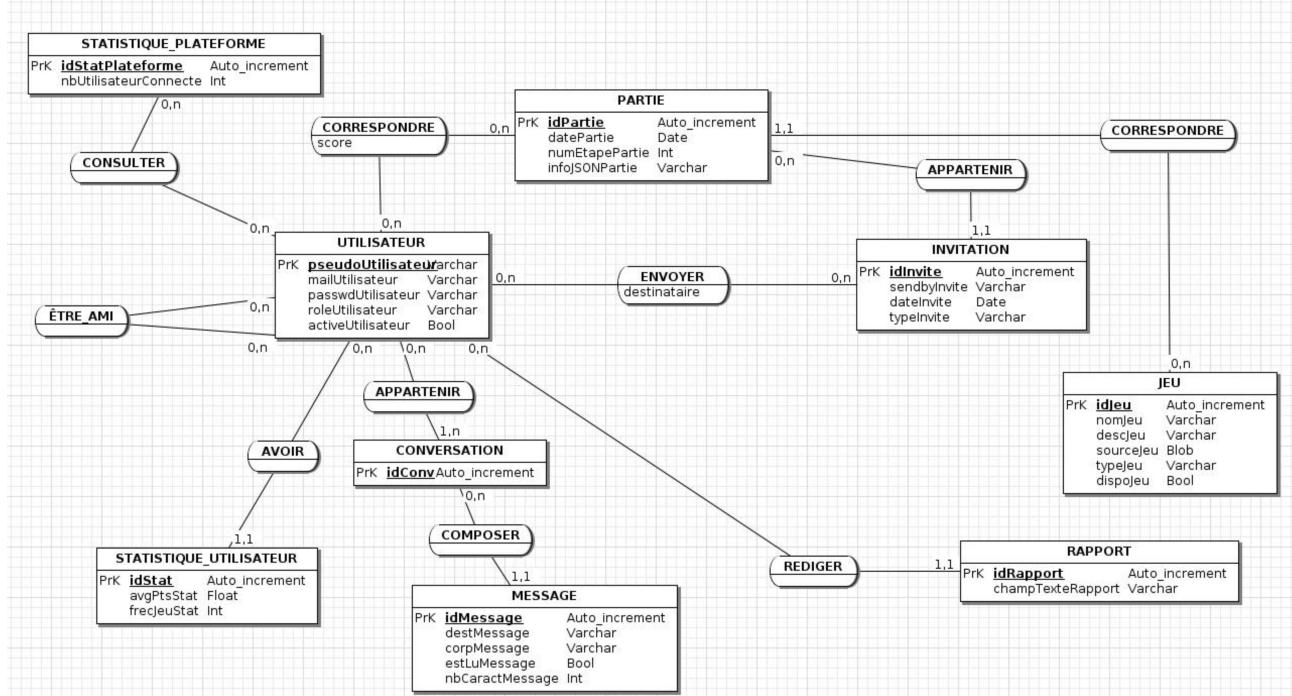


Figure II.3: Modèle Conceptuel de Données

L'utilisateur possède un pseudonyme comme clé primaire. Ainsi, chaque pseudonyme sera unique.

Un utilisateur peut :

- correspondre à une partie qui elle-même appartient à un jeu ;
- envoyer une invitation qui appartient à une partie, et qui correspond donc avec l'expéditeur de l'invitation ;
- appartenir à plusieurs conversations et composer plusieurs messages par conversation ;
- accéder à plusieurs statistiques utilisateur ;
- rédiger plusieurs rapports ;
- consulter plusieurs statistiques de la plateforme ;
- être ami avec plusieurs autres utilisateurs.

Pendant la réalisation de ce M.C.D., quelques problèmes ont été rencontrés tels que d'illustrer les rôles utilisateurs ou le fait de rejoindre une partie via une invitation.

L'équipe enseignante nous a, par la suite, fourni une nouvelle version, corrigée, du M.C.D. du projet.

Cette autre version ne contient pas les statistiques contrairement à notre M.C.D.. De plus un utilisateur peut participer à plusieurs parties, mais seulement deux parties différentes qui correspondent à n'importe quel jeu ayant n'importe quel type.

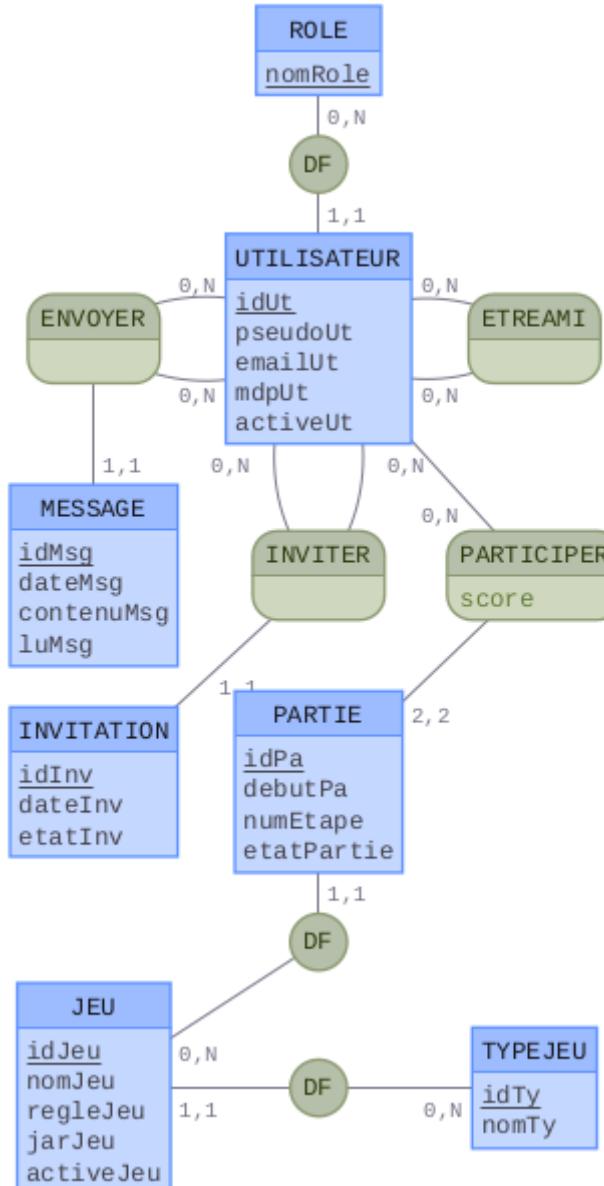


Illustration II.1: Version corrigée du M.C.D.

Le nouveau M.C.D. contient une table “Rôle” permettant à l’utilisateur d’être reconnu comme simple joueur ou administrateur. De plus, contrairement à notre M.C.D. le rôle n’est pas inclus directement dans la table “UTILISATEUR”.

La table “conversation” n’est pas présente dans le nouveau M.C.D. ainsi que le “rapport”.

### 3. Script de la base de données

Le script de la base de données sert à implémenter la base de données. Il a été réalisé via le logiciel de développement Atom. *Voir annexe SQL VI.24.*

Nous avons réalisé ce script à partir du MCD décrit plus haut. Nous n’avons pas gardé les tables suivantes :

- AVOIR
- APPARTENIR
- CORRESPONDRE
- RÉDIGER

En effet, il s’agissait de Contraintes d’Intégrités Fonctionnelles, il était donc plus efficace de retirer ces tables. Cependant, nous avons donc dû ajouter l’identifiant de l’une des tables dépendantes de la CIF dans l’autre. Ainsi, nous avons mis :

- l’ID de JEU dans PARTIE ;
- l’ID de STATISTIQUE\_UTILISATEUR dans UTILISATEUR ;
- l’ID de INVITATION dans PARTIE ;
- l’ID de UTILISATEUR dans RAPPORT ;

Pour les autres tables et attributs, nous les avons simplement implémentés dans le script sans les modifier. *Voir annexe SQL VI.25.*

Un nouveau script nous a été donné en correction. *Voir annexe SQL VI.26.*

On peut constater des différences avec le nôtre. En effet, au lieu d’un attribut pour indiquer le rôle, l’équipe enseignante a créé une table à part. Ainsi, on ne peut pas être autre chose qu’administrateur ou joueur. On constate aussi l’absence de la table CONVERSATION, le tri des messages par date, par destinataire et par expéditeur sera donc géré par les requêtes ou par le code Java.

De même, il n’y a pas de table STATISTIQUE (utilisateur et joueur), les informations présentes dans ces tables seront donc obtenues via des requêtes (pour : nbUtilisateurCo, avgPtsStat et freqJeuStat). Comme les entités CONVERSATION et STATISTIQUE\_PLATEFORME et les liaisons APPARTENIR et CONSULTER sont absentes. La table CORRESPONDRE, qui permettait le lien entre UTILISATEUR et PARTIE, est également absente. Les informations de CORRESPONDRE ont été déplacées dans PARTIE.

## 4. Requêtes

Pour nous assurer du bon fonctionnement de notre système, nous avons du faire les requêtes :

- La liste des partie en cours de l'utilisateur iuto ;
- La liste des message non lus de l'utilisateur iuto ;
- Le nombre de parties gagnées par le joueur iuto contre le joueur iutc pour le jeu Puissance 4.

Pour les créer , nous nous sommes évidemment aider du MCD,du script et d'une base de donnée que nous avons fait précédemment.

Pour commencer, voici la requête « La liste des parties en cours de l'utilisateur iuto » :

Un utilisateur est désigné par la donnée pseudoUtilisateur et une partie est désigné par la donnée idPartie. Ces deux données sont contenu dans l'association Correspondre. En sachant que l'on cherche La liste des parties en cours de l'utilisateur iuto, on sélectionne donc l'idPartie quand cette partie est associé au pseudoUtilisateur iuto.

Et voici la même requête en fonction du MCD que nous a fourni le corps enseignant :

```
select idPartie from CORRESPONDRE where pseudoUtilisateur='iuto';
```

Ensuite, voici la requête « La liste des message non lus de l'utilisateur iuto » :

```
select corpMessage, DestMessage from APPARTENIR natural join CONVERSATION  
natural join MESSAGE where pseudoUtilisateur='iuto' and estLuMessage=false;
```

Pour cette requête, nous voulons sélectionner les messages de l'utilisateur iuto qui ne sont pas lu. Pour cela, nous sélectionnons le corps du message désigné par la donnée corpMessage et le destinataire du message qui est désigné par la donnée DestMessage . Toute deux contenue dans la table MESSAGE que nous allons joindre aux table CONVERSATION et APPARTENIR car nous avons aussi besoin de la donnée pseudoUtilisateur. Ensuite, on les sélectionne quand le pseudoUtilisateur est iuto et quand la donnée estLuMessage est égal à false.

Enfin, voici la requête «Le nombre de parties gagnées par le joueur iuto contre le joueur iutc pour le jeu Puissance 4 » :

```
select count(idPartie) nbPartieGagnee from PARTIE natural join JEU where
numEtapePartie=-1 and nomJeu='Puissance 4' and idPartie in ( select j1.idPartie
    from CORRESPONDRE j1, CORRESPONDRE j2 where j1.pseudoUtilisateur='iuto' and
j2.pseudoUtilisateur='iutc' and j1.score > j2.score and j1.idPartie=j2.idPartie );
```

Pour cette requête, on sélectionne le nombre de partie(count(idPartie)) qui sont finis (quand numEtapePartie est égal à -1) et qui sont associés au jeu Puissance 4 (nomJeu='Puissance 4'). Il faut aussi que les joueurs dans les parties soient 'iuto' (j1.pseudoUtilisateur='iuto') et iutc (j2.pseudoUtilisateur='iutc') et que le score du joueur iuto soit supérieur à iutc puisqu'il a gagné (j1.score>j2.score) et qu'évidemment les 2 joueurs soit dans la même partie (j1.idPartie=j2.idPartie).

## III. Module “Joueur et Administrateur”

### 1. Joueur

Ce module implémente les fonctionnalités auxquelles auront accès les joueurs : utilisateurs lambdas de la plateforme.

Lorsque le joueur arrivera sur la plateforme, il aura le choix entre “se connecter” ou “s’inscrire” ; s’il n’a pas encore de compte, il devra s’inscrire. Pour ce faire, il devra choisir un pseudonyme non utilisé, un mot de passe qu’il devra confirmer et donner une adresse email valide. Une fois le compte créé, il pourra se connecter en utilisant son pseudonyme et son mot de passe.

Une fois connecté, les utilisateurs tout comme les administrateurs peuvent librement jouer à un jeu présent sur le logiciel et effectuer les actions suivantes :

- Consulter ses messages et en envoyer
- Répondre aux invitations, si un joueur accepte une invitation, une partie est immédiatement lancée entre le joueur ayant envoyé l’invitation et celui l’ayant acceptée.
- Consulter une partie en cours et éventuellement y jouer (un tour si c’est une partie de puissance 4, une manche si c’est une partie de Mastermind,...). Lorsqu’il joue, un joueur peut consulter les statistiques de son adversaire.
- Consulter ses statistiques ;
- Historique des parties ;
- Ratio de victoires (nombre de victoire / nombre de partie) pour chaque jeu et au total ;
- nombre de partie jouées pour chaque jeu et au total ;
- Consulter sa liste d’amis et leurs statistiques ;
- Envoyer et accepter une demande d’amis ;
- Créer une partie en libre accès ou en privé (il devra alors envoyé une invitation à un autre joueur pour jouer) ;
- Rejoindre une partie en libre accès ;

- Envoyer une invitation pour une partie, qui sera considéré comme obsolète au bout de 24h sans aucune réponse du destinataire.

### a. Diagramme de cas d'utilisation

Ce diagramme a pour but de décrire toutes les fonctionnalités dont les joueurs de la plateforme "Duel Sur La Toile".

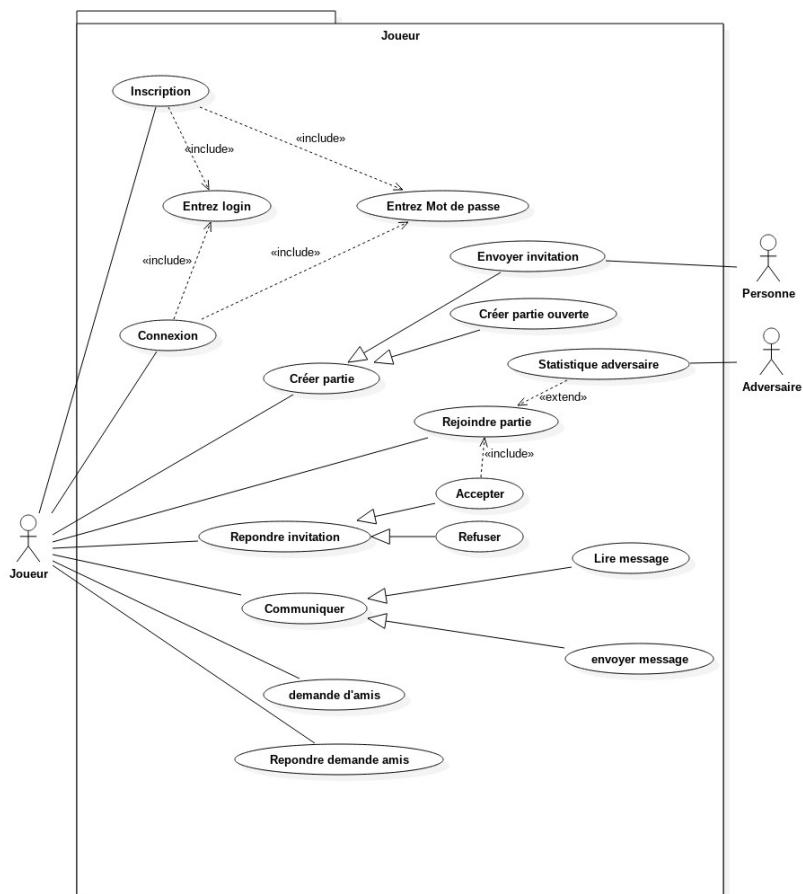


Diagramme des cas d'utilisation pour le module Joueur.

### b. Description

La description textuelle de ce diagramme a pour but de définir précisément tout ce que peut faire un joueur et surtout comment il peut le faire.

#### i. *Inscription*

Tout utilisateur doit préalablement créer un compte pour utiliser la plateforme Duel Sur La Toile.

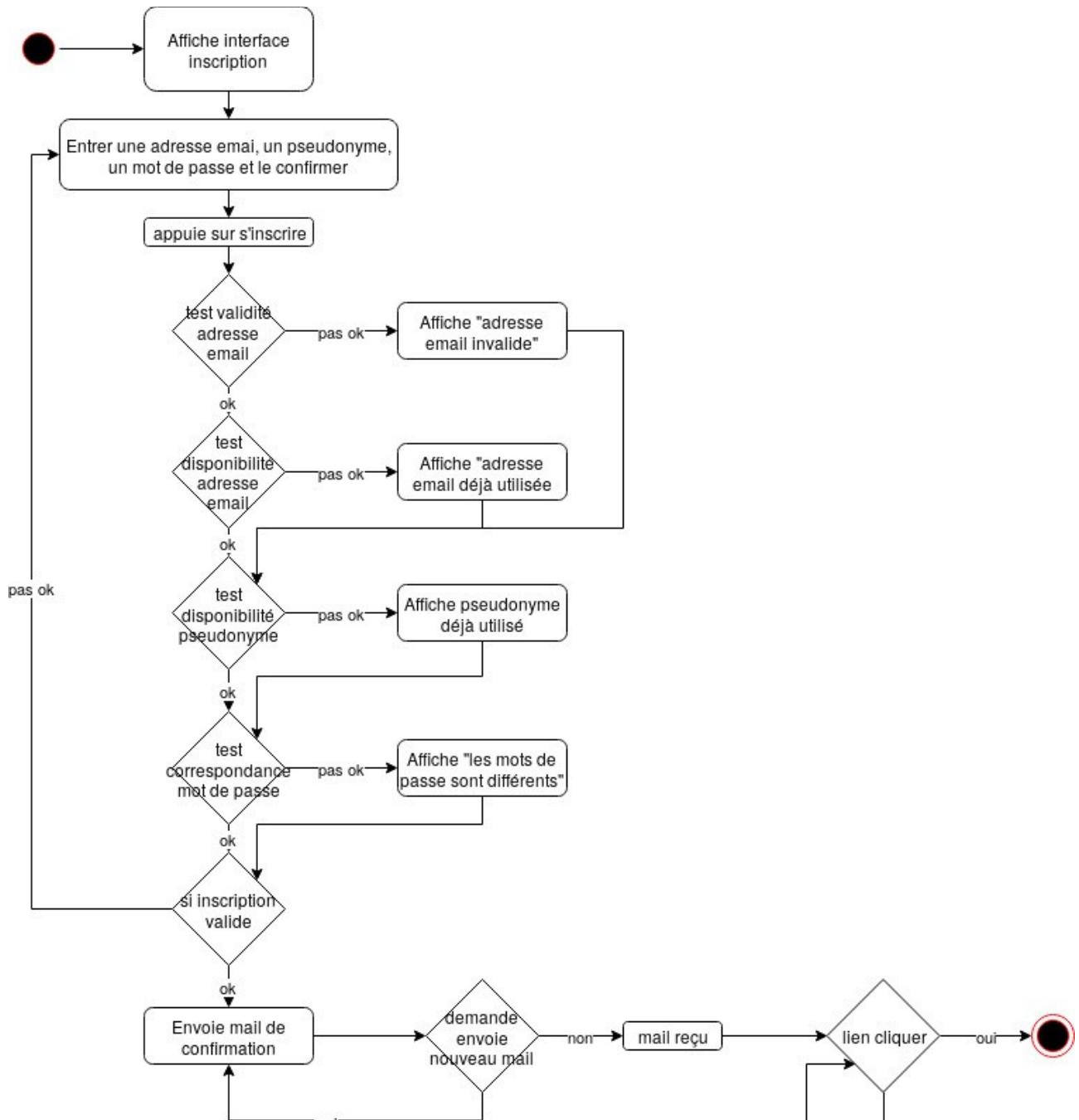


Diagramme de séquence de l'inscription

Le scénario débute lorsqu'un utilisateur clique sur le bouton "Inscription". Il est alors accueilli par un affichage de l'interface d'inscription constituée des champs suivants :

- pseudonyme ;
- email ;
- mot de passe, et

- confirmation du mot de passe.

L'utilisateur accepte les conditions légales de son pays en choisissant de s'inscrire. Une fois que l'utilisateur a rempli les champs requis et clique sur le bouton de confirmation, un mail de confirmation d'inscription est envoyé à l'adresse entrée par l'utilisateur. Une fois que l'utilisateur clique sur le lien du mail, il peut utiliser la page de connexion.

Il est possible que lorsque l'utilisateur tente de s'inscrire, il entre des informations invalides. De ce fait, le système affiche quels champs doivent être remplis à nouveau et les conditions de remplissage pour permettre à l'utilisateur d'entrer des informations valides.

Une fois inscrit, il peut se connecter et accéder aux fonctionnalités de la plateforme. Une nouvelle instance d'utilisateur est créée dans la base de données.

## ***ii. Connexion***

Un joueur (qui peut être un administrateur) doit se connecter pour accéder aux autres fonctionnalités de la plateforme.

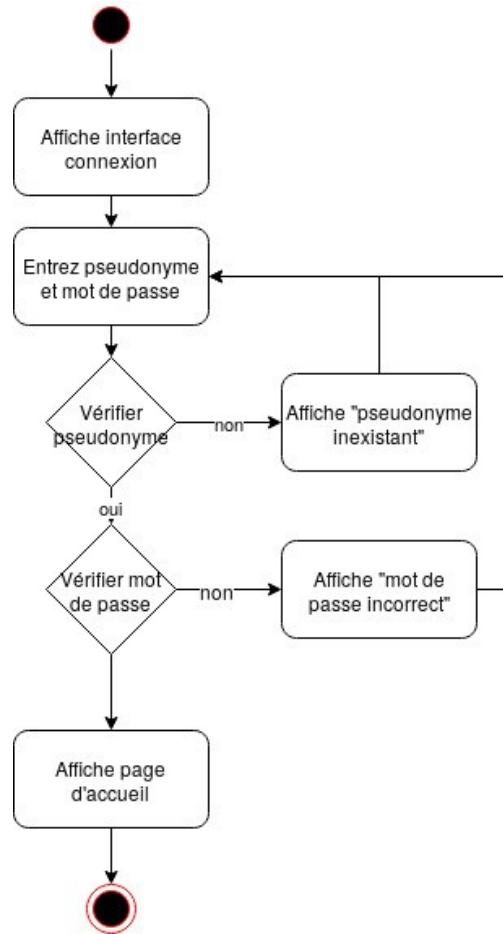


Figure III.1 : Diagramme de séquence connexion

Pour pouvoir se connecter, il doit déjà avoir créé un compte, c'est-à-dire s'être inscrit (l'inscription est unique par compte).

Pour se connecter, il doit cliquer sur le bouton "Connexion" qui permettra l'affichage de la page de connexion ; celle-ci se compose de deux champs de texte à remplir ("pseudonyme" et "mot de passe") ainsi qu'un bouton de validation. Si les informations entrées sont inexactes, un message s'affiche selon l'erreur détectée ("pseudonyme inexistant", "compte désactivé" ou "mot de passe incorrect") et l'utilisateur est invité à réessayer.

Si les informations entrées sont exactes (que le pseudonyme existe, que le compte est activé et que le mot de passe correspond au pseudonyme), l'utilisateur est redirigé sur la page d'accueil et obtient l'accès aux fonctionnalités de la plateforme.

### iii. Création d'une partie

Ce cas d'utilisation décrit la création d'une partie pour l'un des jeux présents sur la plateforme.

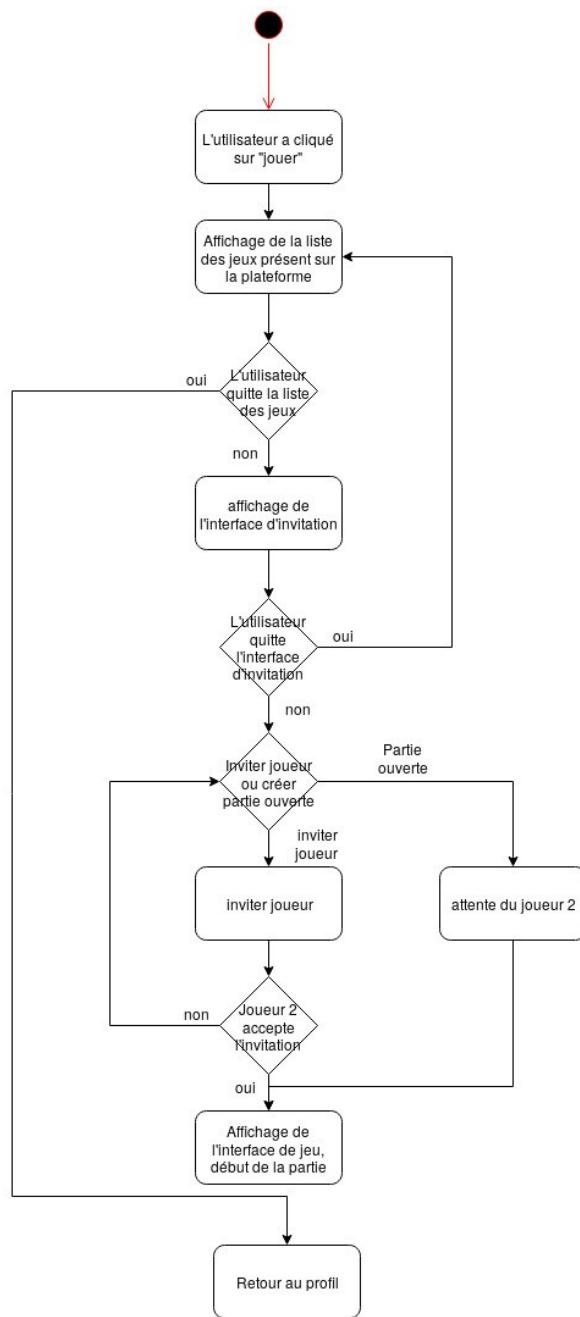


Figure III.2 :

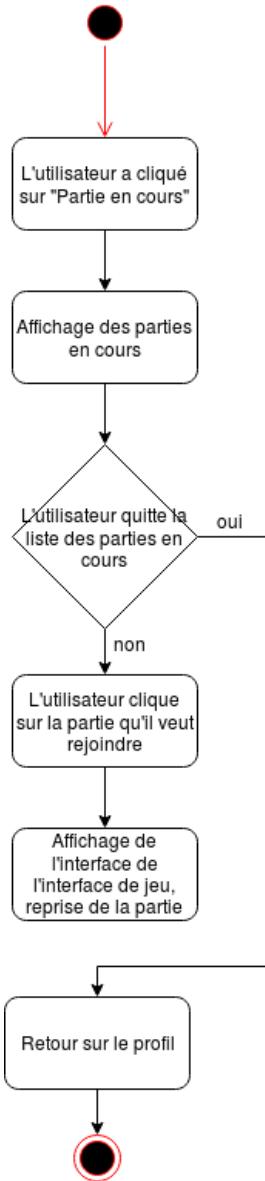
Pour pouvoir créer une partie, un joueur doit être connecté.

Le joueur clique sur le bouton “Jouer”. Il se retrouve alors avec la liste des jeux présents sur la plateforme. Il pourra alors choisir le jeu qui l'intéresse, le redirigeant sur

l'interface d'invitation. Le joueur peut alors choisir d'inviter l'un de ses amis ou de créer une partie ouverte que n'importe quel membre peut rejoindre.

À partir de ce moment, le joueur attend qu'un joueur accepte son invitation ou rejoigne sa partie. Il peut accéder aux autres fonctionnalités de la plateforme pendant ce temps.

#### iv. Rejoindre partie



Rejoindre une partie permet à une personne de jouer avec un autre membre sur la plateforme. Le joueur peut rejoindre une partie dans les cas où :

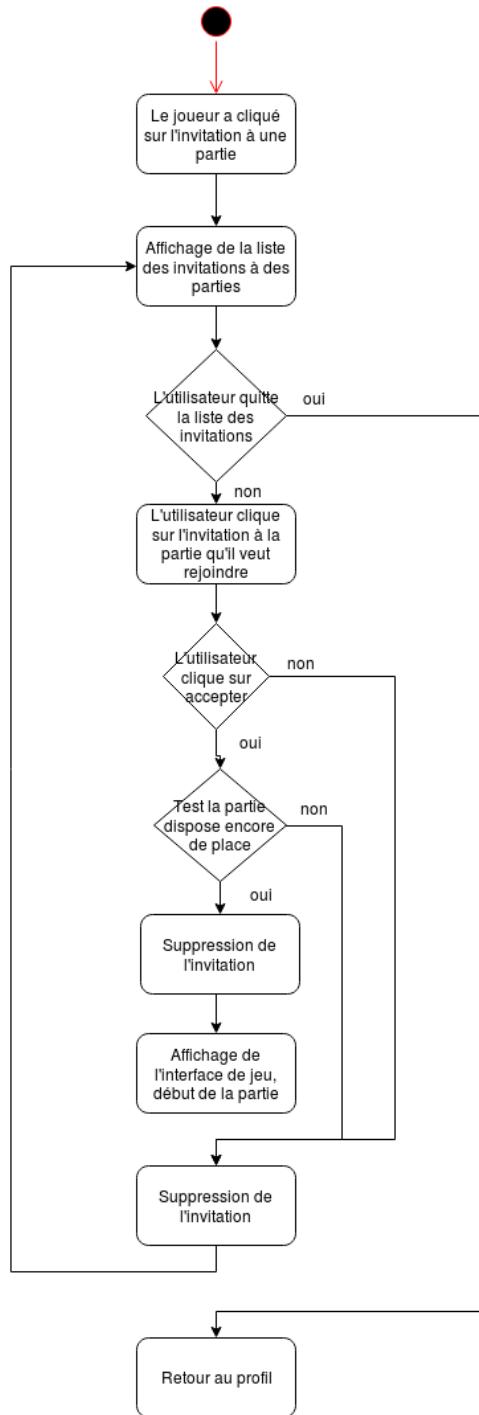
- la partie choisie n'est pas commencée, et la place est libre, ou
- le joueur reprend une ancienne partie.

Pour rejoindre une partie, le joueur doit cliquer sur le bouton "Partie en cours".

Après cela, l'utilisateur obtient l'affichage des parties en cours. Il clique alors sur la partie qu'il veut rejoindre et son appareil affiche l'interface du jeu et la partie reprend là où elle avait été arrêté.

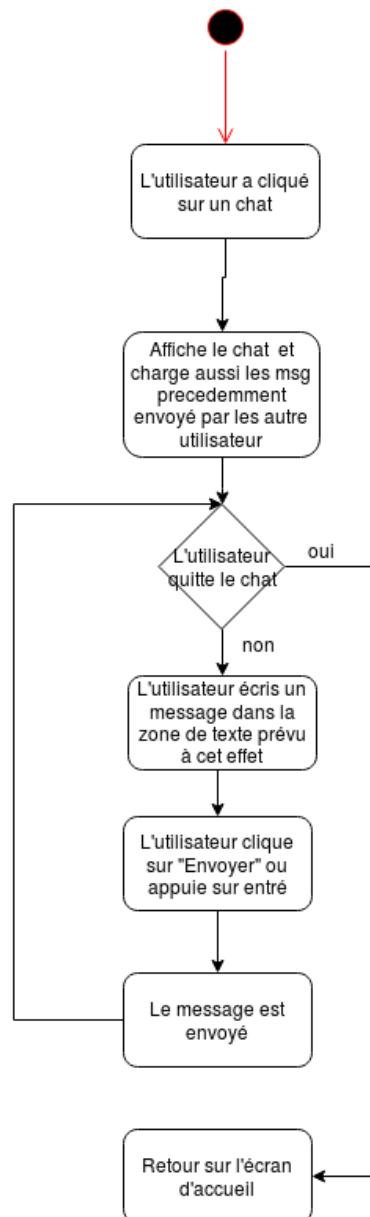
Il est aussi possible pour l'utilisateur de quitter l'affichage des parties en cours.

#### v. Répondre invitation à jouer



Répondre à une invitation sert à rejoindre la partie à laquelle on a été invité ou à indiquer au joueur qui nous l'a envoyée que nous ne souhaitons pas rejoindre sa partie. Pour répondre à une invitation, le joueur doit être connecté et avoir reçu une invitation. Il doit cliquer sur invitation, ce qui affichera la liste de nos invitations, et peut ensuite choisir d'accepter une invitation et de la refuser à l'aide des boutons appropriés. Après avoir répondu à une invitation, celle-ci est supprimée. Si le joueur l'a accepté, une partie est immédiatement lancée entre le joueur qui a accepté l'invitation et celui qui l'a envoyée.

## vi. Communication



Il est possible pour l'utilisateur d'envoyer et recevoir des messages de ses amis ; il doit bien sûr être connecté pour utiliser cette fonction.

Cette option est disponible en cliquant sur un tchat.

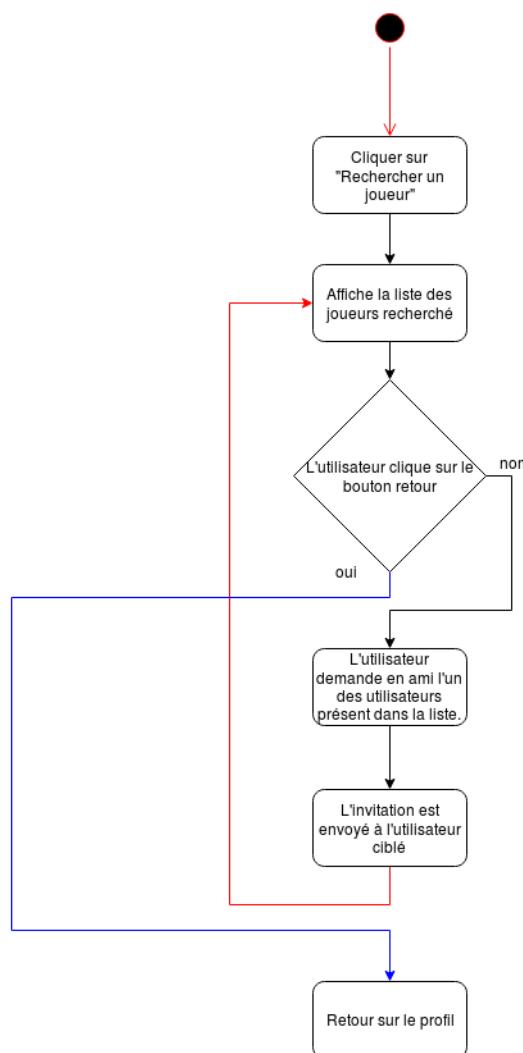
En ce cas, le système affiche les messages du tchat. L'utilisateur saisit un message et clique sur le bouton "envoyer" ou appuie sur la touche "Entrée" pour l'expédier.

Il est possible aussi pour l'utilisateur de quitter le chat à tout moment.

De ce fait, un ou plusieurs messages sont envoyés ; les messages reçus sont marqués comme lu dès lors que le destinataire ouvre la fenêtre de tchat.

#### vii. Demande d'amis

La demande d'amis sert à inviter un autre utilisateur à être son ami.



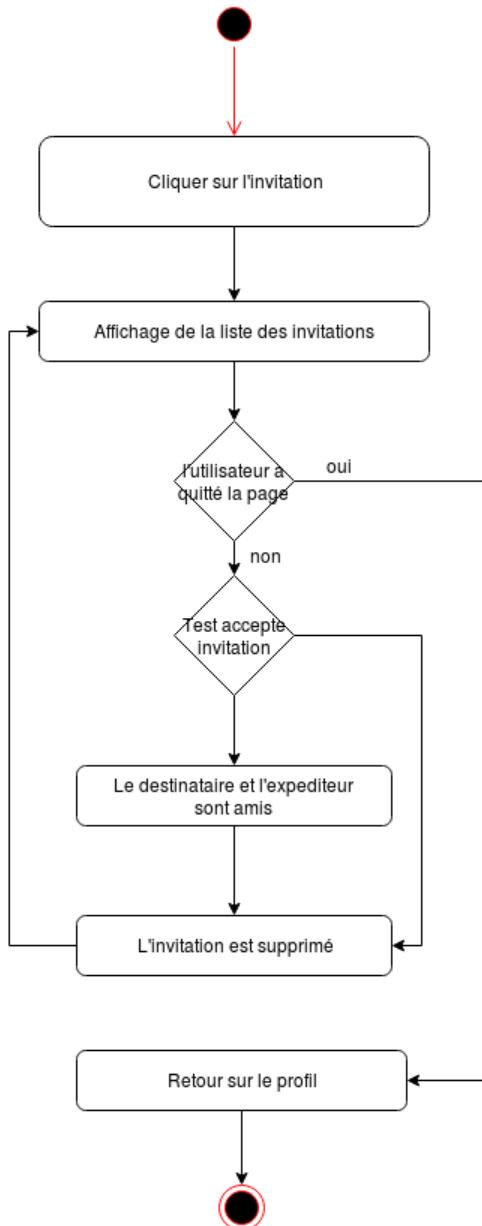
L'utilisateur doit d'abord être connecté.

Afin de rechercher un joueur, l'utilisateur clique sur le bouton "Rechercher un joueur". La liste des joueurs s'affiche alors ; le joueur peut alors envoyer une demande d'ami en cliquant sur le bouton "envoyer une demande d'amis" à côté du pseudonyme concerné.

Le joueur cible reçoit une demande d'ami.

### viii. Répondre demande d'amis

Répondre à une demande d'amis sert à accepter une demande d'amis venant d'un autre utilisateur de la plateforme.



Pour répondre à une demande d'amis, le joueur doit être connecté et avoir reçu une demande d'ami.

Pour répondre à une demande d'amis, il doit cliquer sur le bouton « Demande d'amis », ce qui affichera la liste de ses demande d'amis. Il peut ensuite choisir d'accepter et de refuser autant de demande d'amis qu'il le souhaite en cliquant sur les boutons "Accepter" et "Refuser".

Après avoir répondu à une demande d'ami, celle-ci est supprimée. Si le joueur l'a accepté, les deux joueurs sont désormais amis et peuvent s'envoyer des invitations et des messages.

## 2. Administrateur

Ce module a pour objectif de permettre aux administrateurs de gérer la plateforme le plus facilement possible. Les administrateurs sont responsables de la gestion des utilisateurs et des jeux présents sur le logiciel.

Un administrateur devra pouvoir activer et désactiver un jeu (C'est-à-dire le rendre accessible ou inaccessible aux joueurs). Il devra aussi pouvoir ajouter un jeu à la plateforme ou en supprimer (dans ce cas, les modifications sont définitives c'est pourquoi on évitera de supprimer les jeux. De plus cela stopperait toutes les parties en cours.).

L'administrateur devra aussi pouvoir gérer les joueurs, c'est-à-dire les activer ou les désactiver. Désactiver un joueur supprime toutes ses invitations en cours. Il abandonne aussi automatiquement toutes ses parties en cours. De plus, il n'aura plus accès aux fonctionnalités auquel ont normalement accès les joueurs.

Enfin, un administrateur doit aussi pouvoir consulter les statistiques de la plateforme et en faire un rapport, c'est-à-dire :

Le nombre de parties en cours ; Cette partie est l'étape d'analyse préliminaire. C'est une partie très importante car elle nous a permis de prendre connaissance avec le projet et créer par nous même une solution adapté aux différents problèmes que soulève le sujet. C'est durant ce chapitre que la **base de données** devait être constituée du système.

Pour cela, le dictionnaire de données a été défini. Un dictionnaire de données est une collection de données nécessaire à la conception d'une base de donnée relationnelle.

Le MCD (Modèle Conceptuel des Données) a ainsi pu être créé. C'est une sorte de schéma représentant les différentes tables ainsi que les associations constitué des données que nous avons préalablement défini dans le dictionnaire .

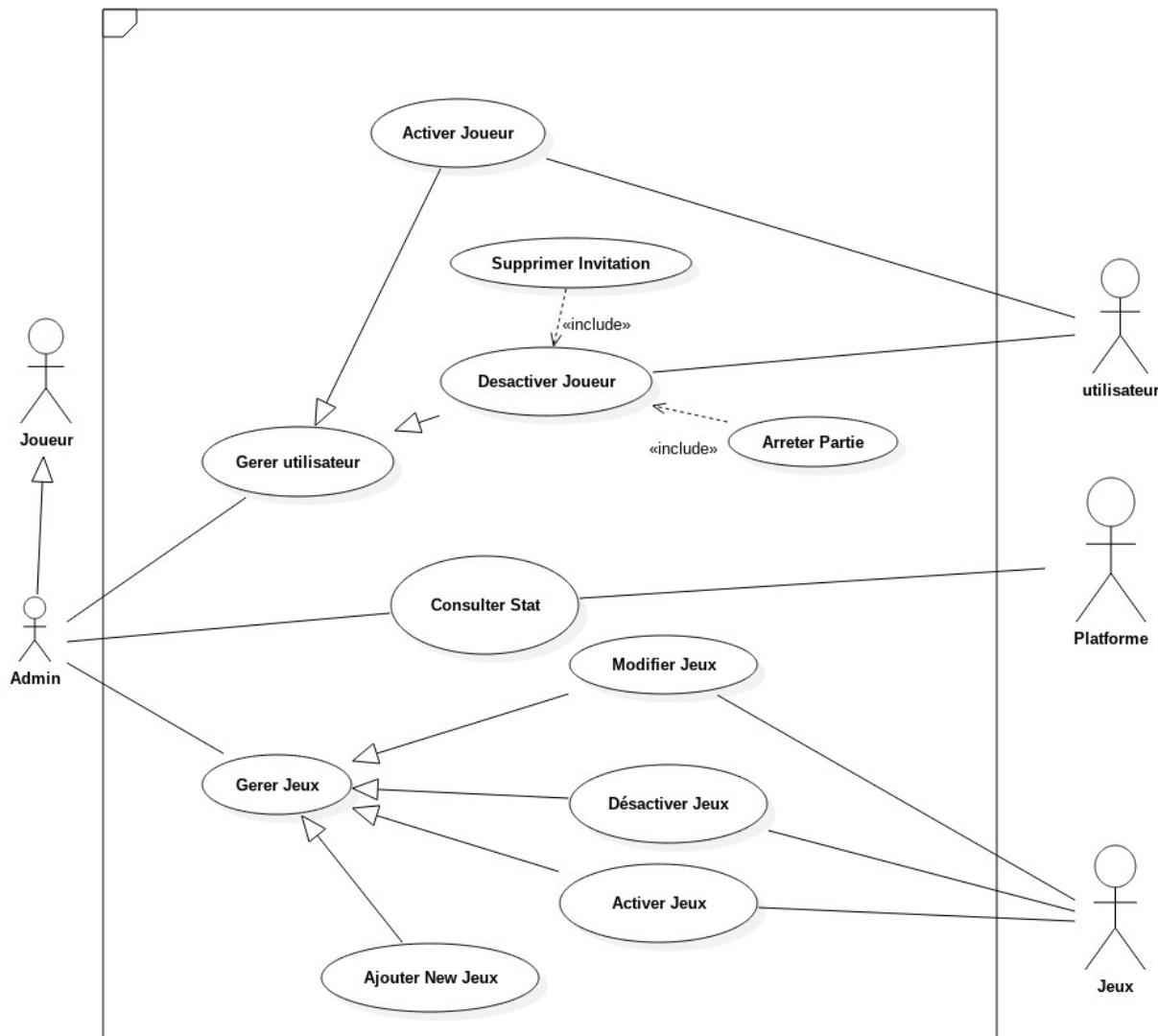
De fait, la réalisation des scripts de cette base de données permet sa création au niveau informatique et non seulement à un niveau abstrait tel le MCD.

Pour le tester nous devions aussi faire des requêtes précisées dans le sujet afin de voir si la base de données fonctionne bien.

- Le nombre de parties jouées pour chaque jeu sur une période donnée ;
- Le nombre de joueurs connectés sur une période donnée, ou actuellement ;

### a. Diagramme de cas d'utilisation :

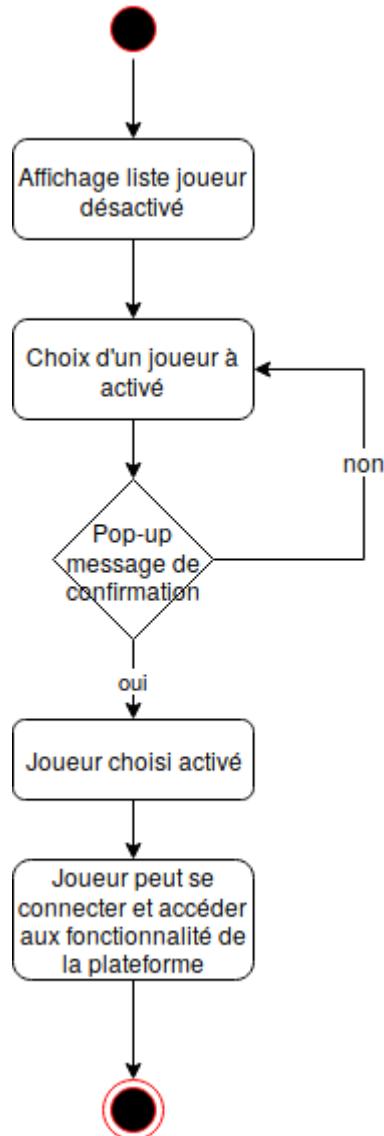
Ce diagramme a pour but de décrire toutes les fonctionnalités auxquelles ont accès les administrateurs de la plateforme "Duel sur la toile".



La description textuelle a pour but de définir précisément tout ce que peut faire un administrateur et surtout comment il peut le faire.

## b. Description

### i. Activer utilisateur



Ce cas d'utilisation a pour but de permettre à l'administrateur d'activer un utilisateur qui n'a pas le rôle administrateur. En effet, un administrateur ne peut exercer ses droits sur un autre utilisateur ayant les mêmes droits que lui.

Les conditions pour effectuer ce cas d'utilisation sont :

- L'activateur doit avoir le rôle administrateur.
- L'utilisateur cible doit exister dans la base de données. En effet, il doit être inscrit sur la plateforme.
- L'utilisateur cible ne doit pas avoir le rôle administrateur.

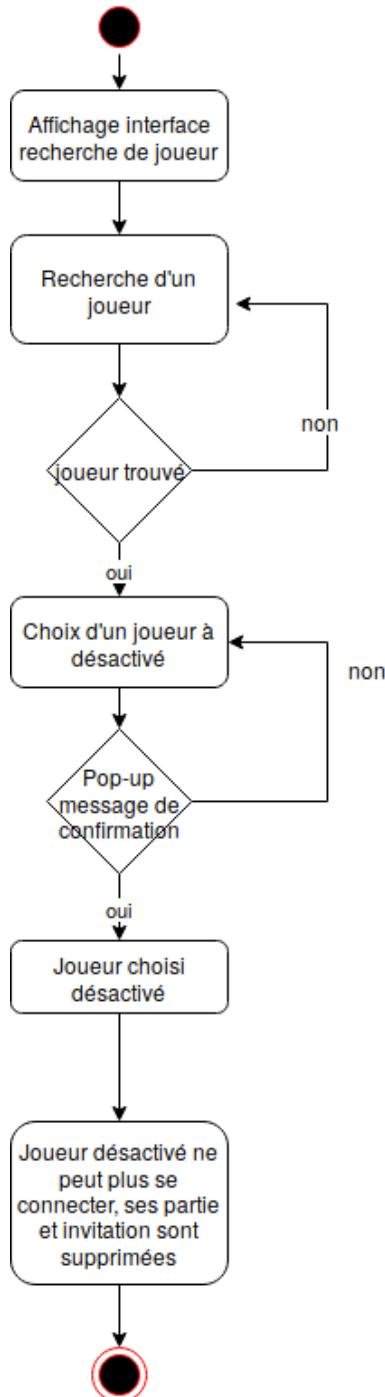
- L'utilisateur cible doit être désactivé. Sinon, l'activer n'est pas nécessaire.

Le scénario commence lorsque l'administrateur clique sur le bouton "Activer joueur".

Le système va alors afficher une liste de tous les joueurs désactivés. L'administrateur choisit un utilisateur à activer, et clique sur le bouton d'activation associé. Ainsi, l'utilisateur est activé.

Cependant, l'administrateur peut choisir de revenir en arrière au lieu de choisir un utilisateur à activer à l'aide d'un bouton retour. Il revient alors sur l'écran d'accueil.

## ii. Désactiver utilisateur



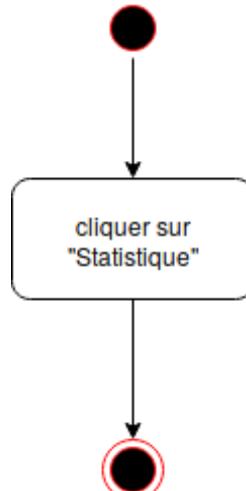
Le processus de désactivation d'un joueur est essentiellement le même que celui d'activation. L'administrateur fait usage d'une fonction "Désactiver joueur" de fonctionnement similaire. On retrouve les conditions suivantes :

- L'activateur doit avoir le rôle administrateur.
- L'utilisateur cible doit exister dans la base de données. En effet, il doit être inscrit sur la plateforme.
- L'utilisateur cible ne doit pas avoir le rôle administrateur.

- L'utilisateur cible doit être activé. Sinon, le désactiver n'est pas nécessaire.

L'IHM est autrement similaire à celle du point précédent.

### ***iii. Consulter statistiques***



Ce cas d'utilisation a pour but de permettre à l'administrateur de voir les différentes statistiques de la plateforme telles que :

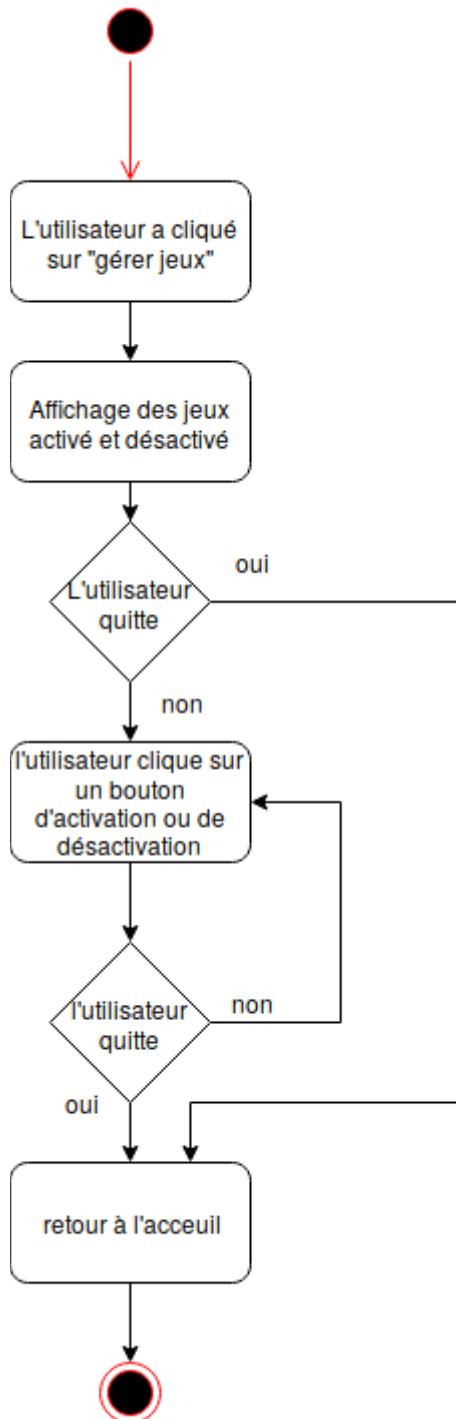
- le nombre de messages envoyés ;
- le nombre d'utilisateurs désactivés ;
- le nombre d'utilisateurs connectés par jour ;
- le nombre d'utilisateurs inscrits, et
- le nombre de joueurs en partie dans un jeu donné ou en général.

L'utilisateur doit avoir le rôle d'administrateur pour user de ce cas d'utilisation.

Le scénario commence lorsque l'administrateur clique sur le bouton "Consulter statistiques". La plateforme affiche alors toutes les données décrites ci-dessus. L'administrateur a ainsi accès aux statistiques de la plateforme.

### ***iv. Gérer les jeux***

Ce cas d'utilisation a pour but de permettre à l'administrateur d'activer et désactiver un jeu. Un jeu désactivé est inaccessible aux utilisateurs.



Les conditions pour **activer** un jeu dans ce cas d'utilisation sont :

- L'utilisateur doit être un administrateur ;
- Le jeu que l'on veut activer doit être dans la base de données ;
- Le jeu doit être désactivé, sinon il n'est pas nécessaire de l'activer.

Le scénario commence lorsque l'administrateur clique sur le bouton "Activer jeu". Le système affiche ainsi la liste des jeux désactivés. L'administrateur choisit un jeu à activer et clique sur le bouton d'activation associé à ce jeu. Une nouvelle fenêtre

demandant à l'administrateur si il veut véritablement effectuer cette action s'affiche alors.

S'il clique sur le bouton "Oui", le jeu est activé et devient disponible aux utilisateurs sur la plateforme. Cependant, l'administrateur peut choisir de revenir en arrière au lieu de choisir un jeu à désactiver. En appuyant sur un bouton retour, il revient alors sur l'écran d'accueil.

Il peut aussi choisir de répondre "Non" lorsque le système lui demande s'il veut vraiment effectuer l'action. Il revient alors sur la liste des jeu désactivés.

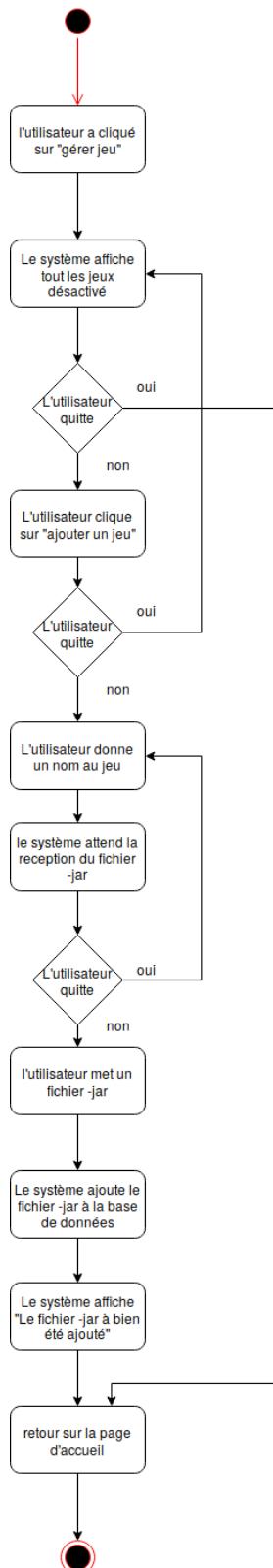
Il est également possible de désactiver un jeu.

Les conditions pour **désactiver** un jeu dans ce cas d'utilisation sont :

- L'utilisateur doit être un administrateur ;
- Le jeu que l'on veut désactiver doit être dans la base de données ;
- Le jeu doit être activé, sinon il n'est pas nécessaire de le désactiver.

L'interface est alors similaire à celle de l'activation d'un jeu.

## v. Ajouter un jeu



Ce cas d'utilisation a pour but de permettre à l'administrateur d'ajouter un nouveau jeu.

Les conditions pour effectuer ce cas d'utilisation sont :

- L'utilisateur doit être administrateur ;
- Le jeu que l'on veut ajouter ne doit pas être dans la base de données.

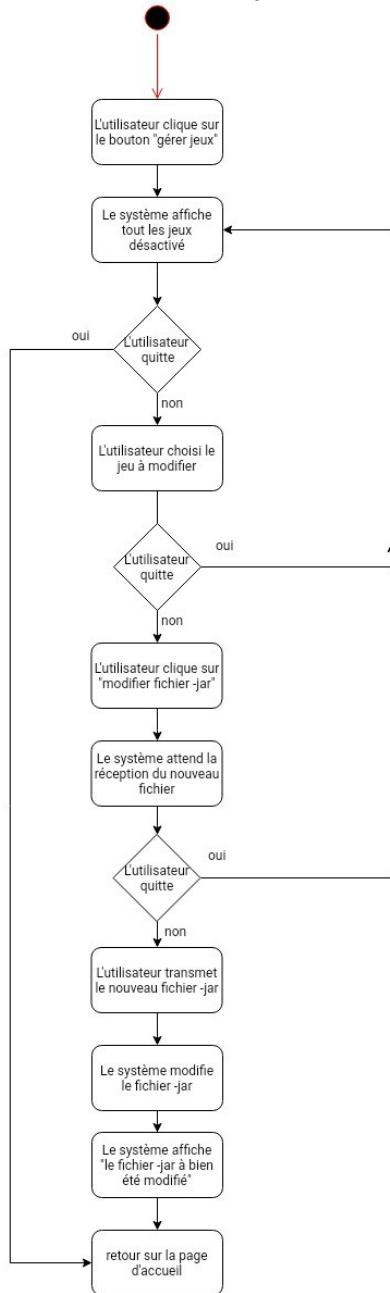
Le scénario commence lorsque l'administrateur clique sur "Gérer jeux". Le système affiche alors tout les jeux désactivés. L'administrateur clique alors sur un bouton "Ajouter un jeu". Une fenêtre s'ouvre et on doit y indiquer le nom du jeu.

Ensuite, le système ouvre l'explorateur de fichiers du système et l'utilisateur doit sélectionner l'exécutable Java JAR du jeu pour l'envoyer sur le site. Le jeu est alors ajouté à la base de données et le message "Le fichier JAR a bien été ajouté." s'affiche si l'opération a réussi. L'administrateur est alors renvoyé à l'accueil. A la suite de ses évènements, le jeu est ajouté sur la plateforme.

Cependant, si on quitte l'interface au lieu de cliquer sur le bouton "Ajouter un jeu", l'administrateur est directement renvoyé à l'accueil. S'il quitte l'interface de sélection de fichier ou de nommage, il est renvoyé sur l'écran des jeux désactivés et aucun jeu n'est ajouté.

### vi. Mettre à jour un jeu

Ce cas d'utilisation a pour but de permettre à l'administrateur de mettre à jour les jeux existants en modifiant l'exécutable Java associé.



Les conditions à remplir pour effectuer ce cas d'utilisation sont:

- L'utilisateur doit être administrateur ;
- Le jeu à mettre à jour doit être désactivé ;
- Le jeu à mettre à jour doit être présent dans la base de données.

Le scénario commence lorsque l'administrateur clique sur le bouton "Gérer jeux". Le système affiche alors la liste des jeux présents sur la plateforme. L'utilisateur choisit le jeu à mettre à jour (il doit être désactivé). Il clique alors sur le bouton "Mettre à jour jeu". Le système lui propose ainsi un explorateur de fichiers pour sélectionner le nouvel exécutable Java qui constituera la nouvelle version du jeu.

Le système prend le nouveau fichier en compte et affiche le message "Le fichier JAR a été modifié.". Il renvoie ensuite l'utilisateur sur la page d'accueil. Si l'on quitte l'interface sans avoir envoyé un nouveau fichier JAR, aucune modification n'est prise en compte et l'utilisateur est renvoyé sur la liste des jeux. Il peut aussi quitter la liste des jeux et retourner directement sur la page d'accueil.

## IV. Module jeu

Deux jeux seront livrés avec l'application : un jeu Puissance 4 en anglais et un jeu Mastermind en français. La plateforme et les jeux seront programmés dans le langage Java.

Les jeux présents sur l'interface sont accessibles via la base de données où le fichier source du jeu est directement stocké dans un objet appelé BLOB. Le jeu sera en format JAR, soit un fichier Java exécutable directement sur l'interface graphique. Un jeu peut être activé ou non ; si désactivé aucun joueur ne peut y accéder.

Une partie se fait sur un même jeu et entre deux joueurs. Lorsque la partie est lancée, les utilisateurs devront faire plusieurs manches qui sera au nombre de trois. Une partie aura trois états, en cours (0), fini (-1) ou abandonné (-2). On conservera l'état d'une partie en cours dans un fichier JSON où l'on trouvera toutes les informations nécessaires (score, état de la partie, ...). Chaque joueur durant une partie aura un score qui, dans le cas où il est parmi les plus élevés, sera ajouté dans le classement des meilleurs résultats pour le jeu sur l'application. Cette partie est l'étape d'analyse préliminaire. C'est une partie très importante car elle nous a permis de prendre connaissance avec le projet et créer par nous même une solution adapté aux différents problèmes que soulève le sujet. C'est durant ce chapitre que la **base de données** devait être constituée du système.

Pour cela, le dictionnaire de données a été défini. Un dictionnaire de données est une collection de données nécessaire à la conception d'une base de donnée relationnelle.

Le MCD (Modèle Conceptuel des Données) a ainsi pu être créé. C'est une sorte de schéma représentant les différentes tables ainsi que les associations constitué des données que nous avons préalablement défini dans le dictionnaire.

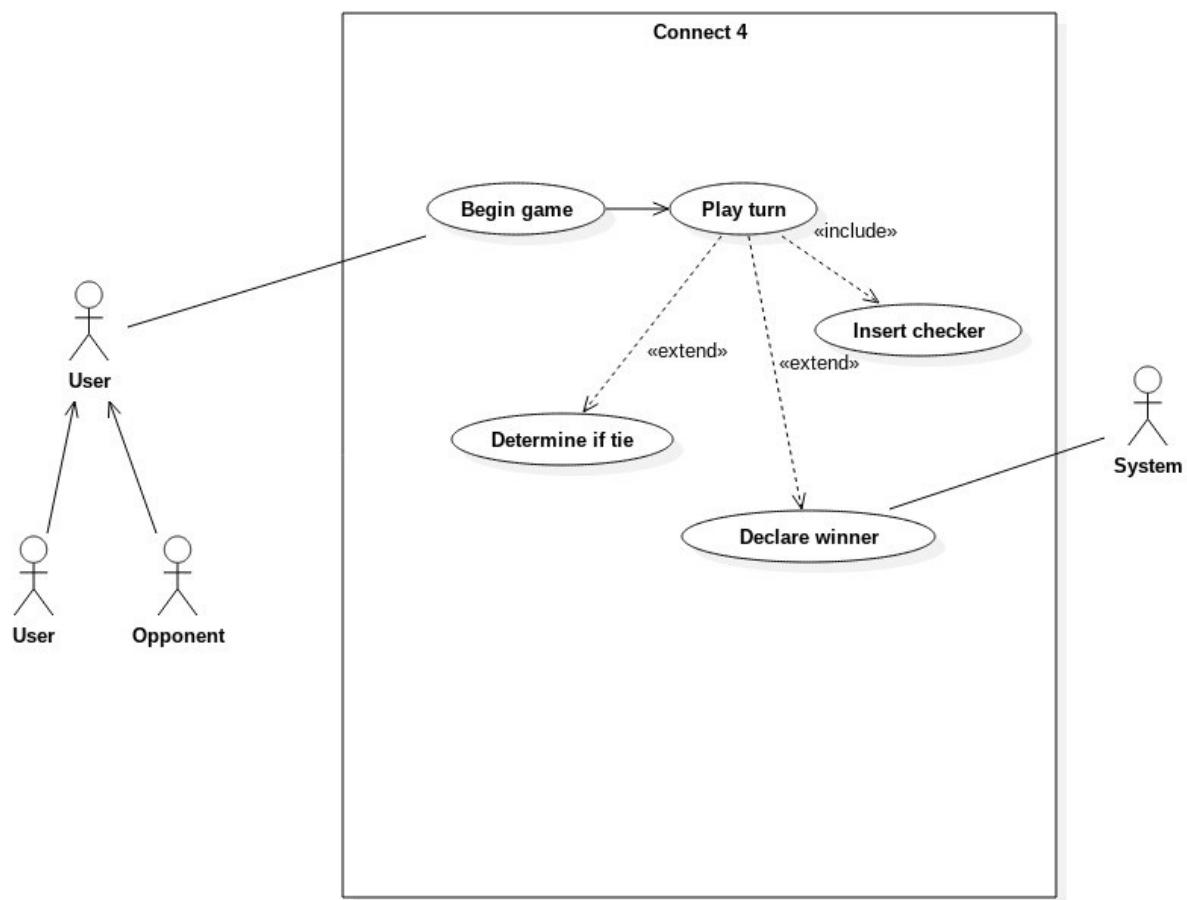
De fait, la réalisation des scripts de cette base de données permet sa création au niveau informatique et non seulement à un niveau abstrait tel le MCD.

Pour le tester, nous devions aussi faire des requêtes précisées dans le sujet afin de voir si la base de données fonctionne bien.

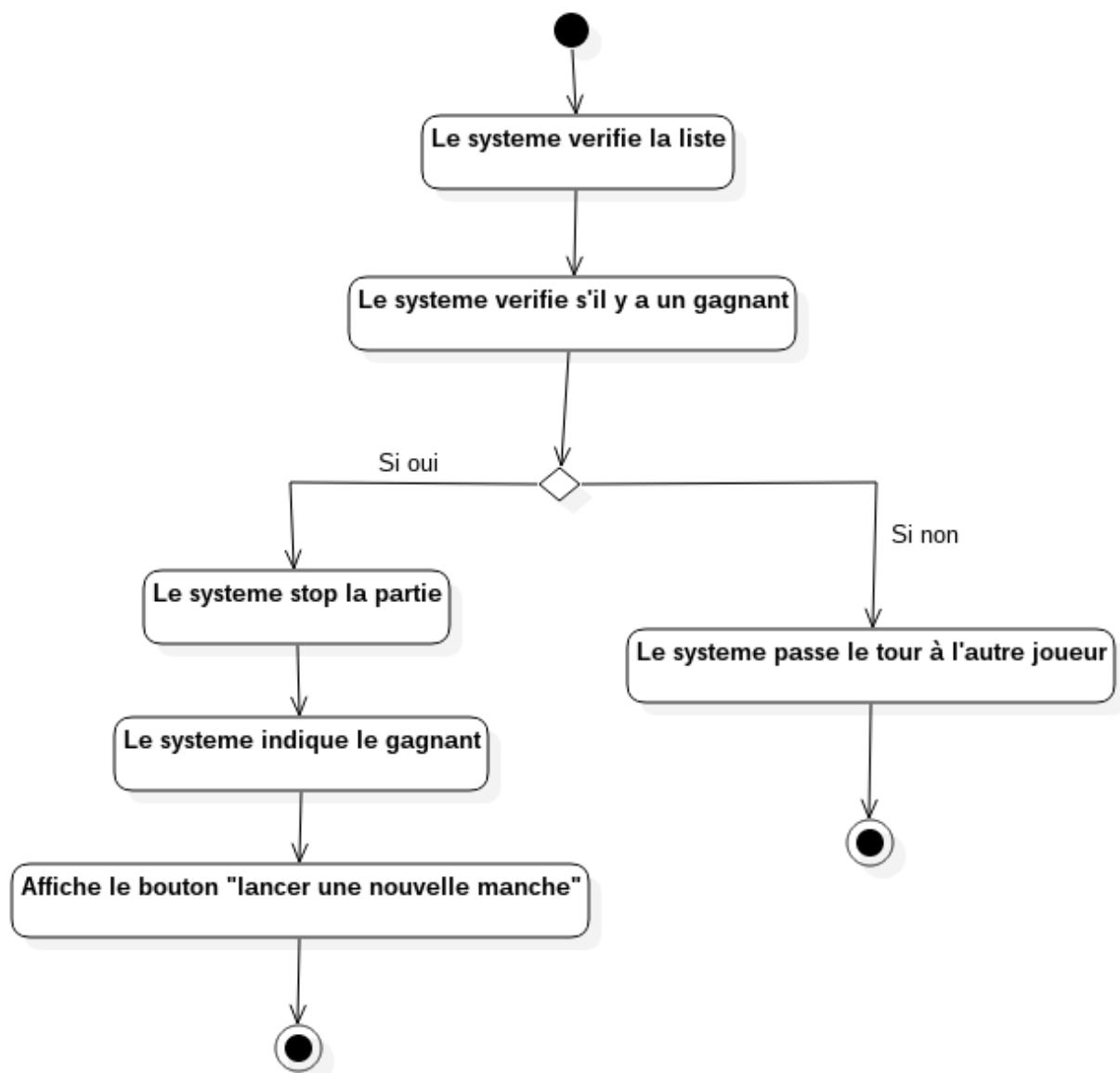
## 1. Puissance 4

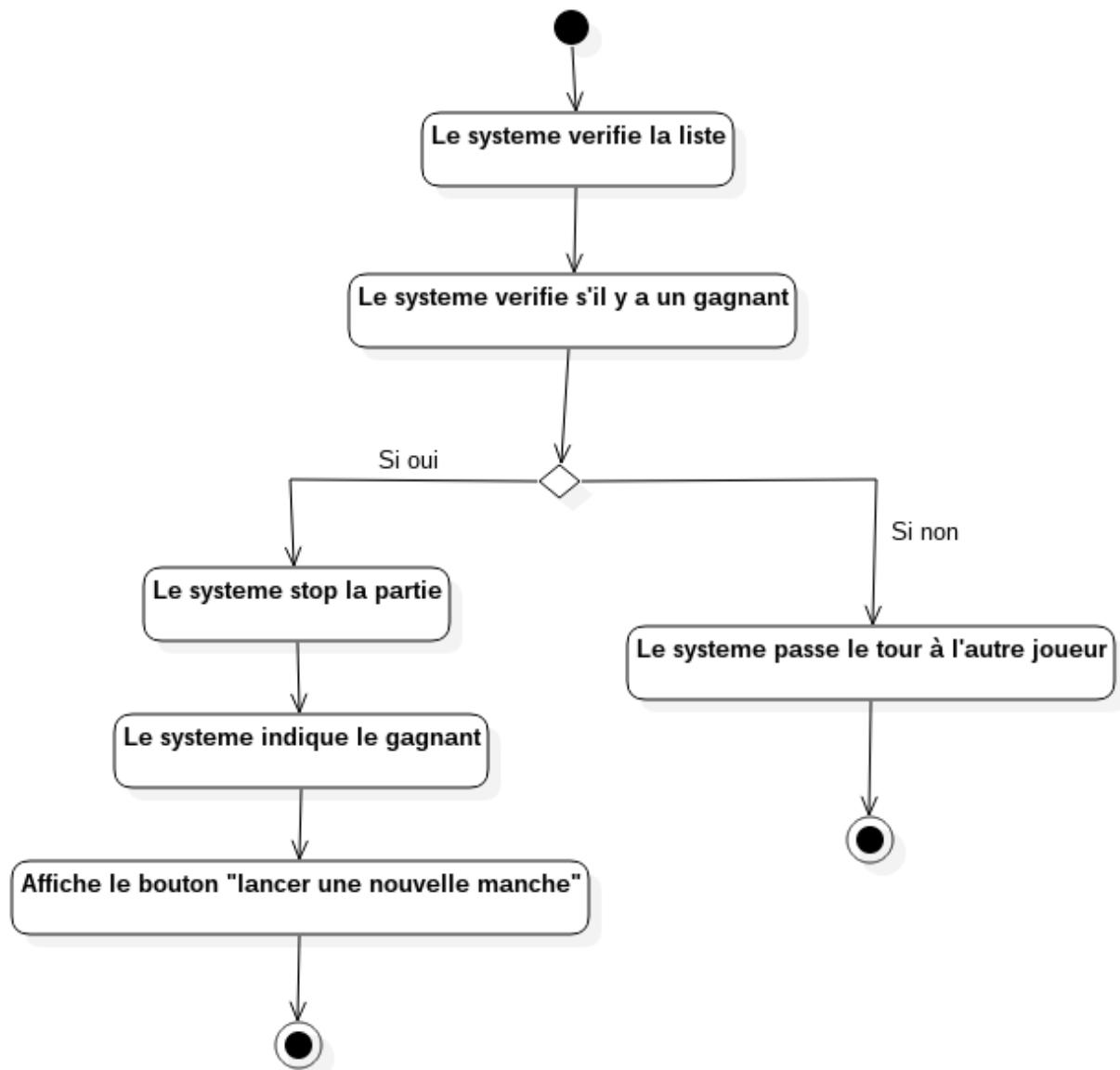
Le puissance 4 est un jeu en tour par tour. Le but est de réussir à aligner quatre pions de sa couleur (rouge ou jaune) à l'horizontale, à la verticale, ou en diagonale dans un tableau de 7 colonnes sur 6 lignes. Les joueurs placent, chacun leur tour, un jeton dans la colonne de leur choix ; le pion descendra dans la case libre la plus basse de la colonne. Les jetons d'une même colonne vont donc s'empiler au cours du jeu. La partie se termine lorsqu'un joueur est parvenu à aligner quatre pions de la même couleur. Ce jeu sera implémenté en anglais.

a.     Diagramme de cas d'utilisation



b. Description





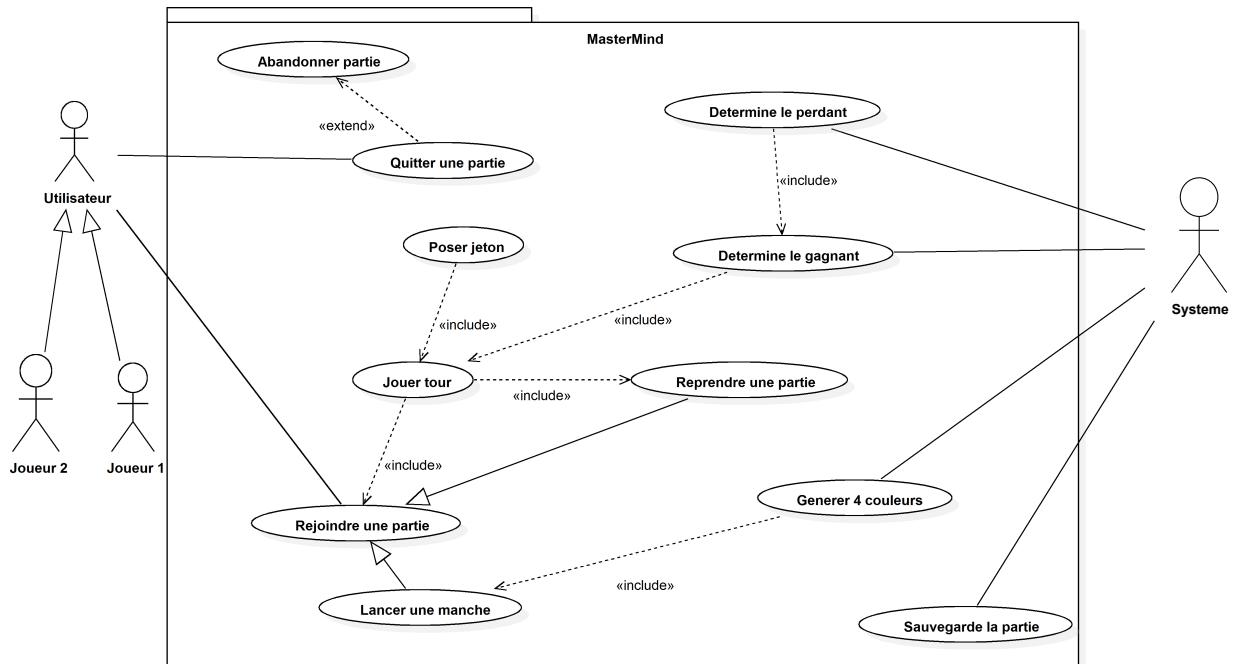
## 2. Mastermind

Le Mastermind est un jeu sans interaction avec son adversaire, un des joueurs sélectionne au choix une combinaison de quatre couleurs. Les joueurs devront trouver cette combinaison en un nombre minimal d'essais. A chaque essai, le jeu informe le joueur du nombre de couleurs bien placées et du nombre de couleurs mal placées afin que le joueur puisse petit à petit définir la combinaison de couleurs à trouver. Les joueurs ont dix coup chacun pour trouver la combinaison.

Une partie se déroule sur trois manches. À chaque fois qu'un joueur trouve la solution, il gagne un nombre de points égal au nombre de coup qu'il lui reste. À la fin de

la partie, on fait le total des points de chaque joueur, et celui qui en a le plus remporte la victoire.

### a. Diagramme de cas d'utilisation

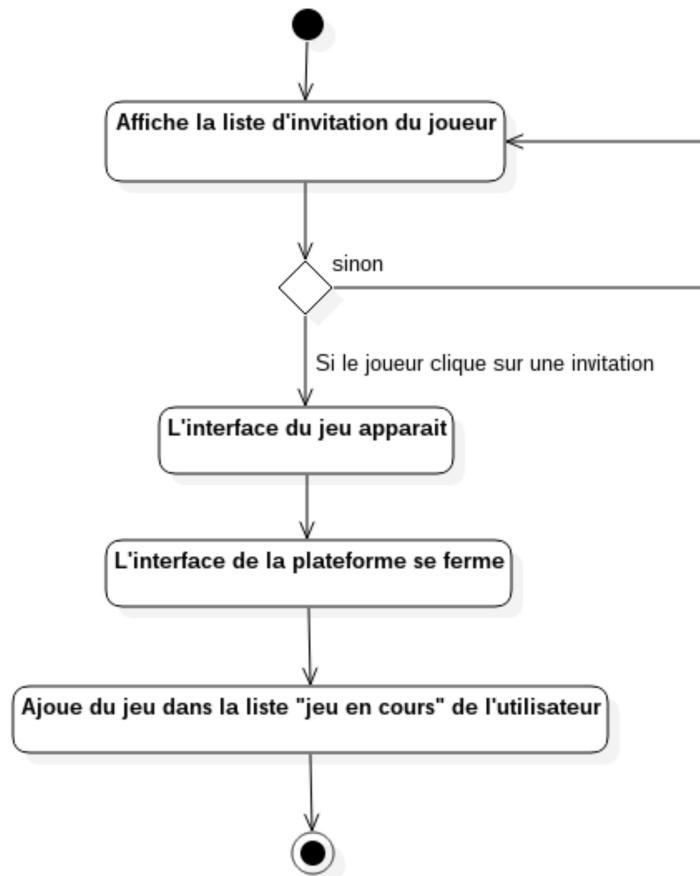


### b. Description

Un utilisateur peut faire différentes actions :

- décider de lancer une manche, ce qui implique que le système génère les 4 couleurs que les deux joueurs devront deviner. Dès qu'un utilisateur lance une manche son adversaire pourra rejoindre la partie s'il le quitte.
- rejoindre une partie en cours. Lorsque l'utilisateur est dans une partie il peut jouer
- son tour ce qui sous entend qu'il peut placer un jeton sur le plateau du mastermind.
- jouer son tours ce qui implique que le système va déterminer le gagnant et le perdant
- de la partie après qu'il ai placé un jeton sur le plateau.
- quitter à tout moment la partie ce qui sous entend qu'il peut également
- l'abandonner, la différence entre ces deux actions est que l'abandonnement d'une
- partie implique qu'il ne peut pas en aucun cas la reprendre.

### i. Rejoindre une partie



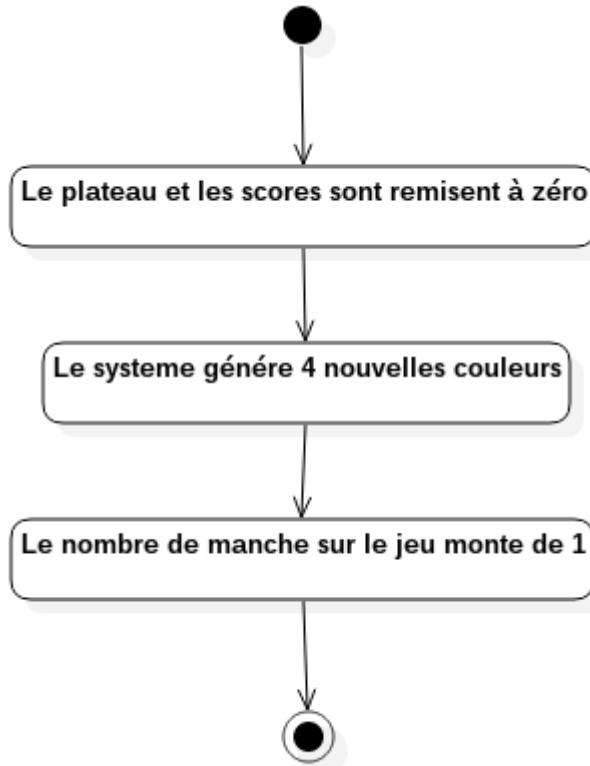
Pour qu'un utilisateur rejoint une partie, il doit être présent dans la base de données et que le jeu doit être actif. L'utilisateur , pour rejoindre une partie doit cliquer sur l'icône du jeu de son choix sur l'interface de la plateforme ou rejoindre via une invitation.

Rejoindre une partie provoque l'affichage graphique du jeu puis de la fermeture de la fenêtre de liste des jeux et de l'ajout du jeu dans la liste des “jeu en cours”. Lorsque le joueur est dans une partie il peut jouer un tour, ou alors lancer une manche ou reprendre la partie s'il l'a quitté.

Rejoindre une partie implique que le joueur joue son tour et donc qu'il pose un jeton .

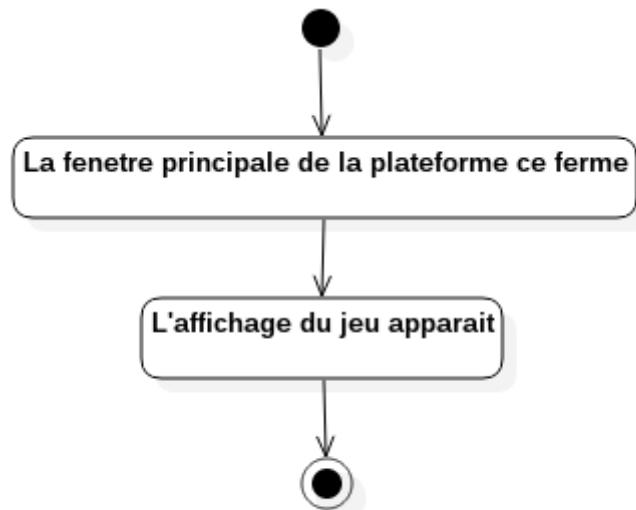
Sur le plateau et que le système détermine le gagnant et le perdant.

## ii. Lancer une manche



Dans le cas où l'utilisateur lance une manche, il doit appartenir à la base de données et appartenir à une partie. Pour que l'utilisateur puisse lancer une manche, il doit cliquer sur le bouton “Lancer la manche” sur l'interface graphique de la partie en cours.

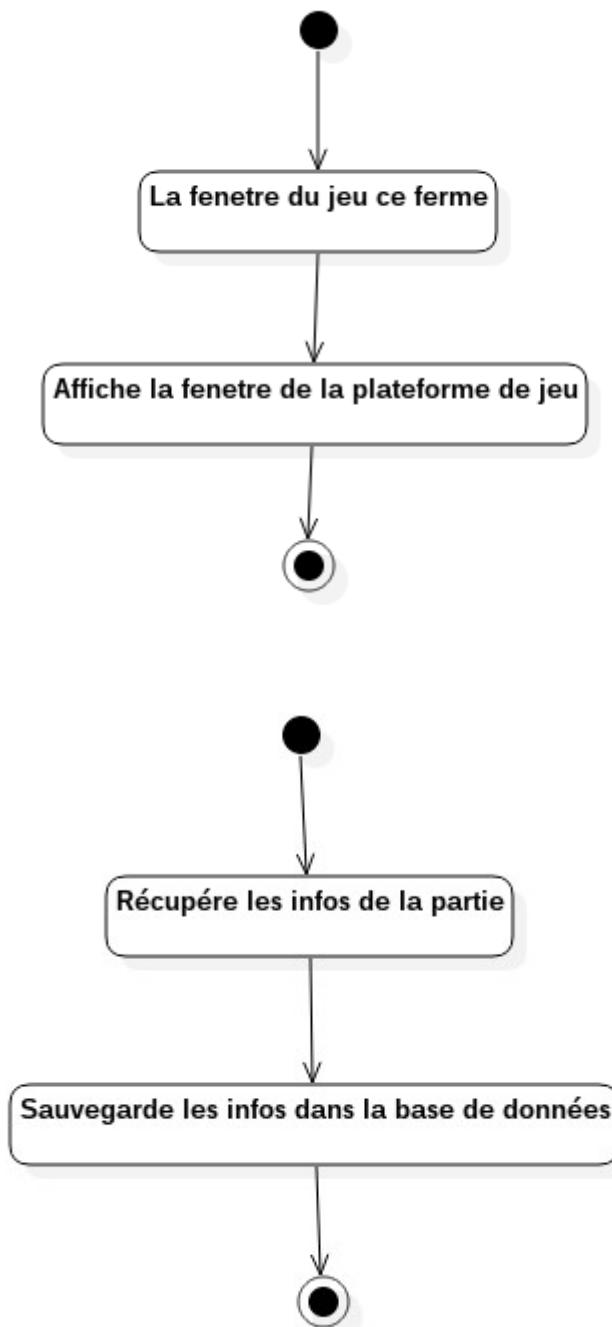
Lancer une manche provoque alors le lancement d'une nouvelle manche sur le jeu choisi par l'utilisateur. Lorsque le joueur a lancé une manche, un autre joueur peut rejoindre la partie et le système génère les quatre couleurs que les deux joueurs devront deviner.



### ***iii. Quitter une partie***

Dans le cas où l'utilisateur quitte une partie , l'utilisateur doit appartenir à la base de données, le jeu qu'il souhaite quitter doit être actif et la partie qu'il souhaite quitter doit être lancée. Pour que l'utilisateur puisse quitter la partie, il doit cliquer sur le bouton “Quitter la partie” sur l'interface graphique du jeu.

Le cas de quitter une partie implique que le joueur puisse l'abandonner, sous -entendant donc qu'il ne pourra jamais reprendre la partie qu'il a abandonnée. Lorsque l'utilisateur quitte une partie, la fenêtre du jeu est fermée, puis que l'interface graphique de la liste des jeux s'affiche et que la partie en cours s'affiche dans la liste des parties en cours de l'utilisateur.



## V. Conclusion

En premier lieu, plusieurs problèmes lors de ce projet ont eu lieu :

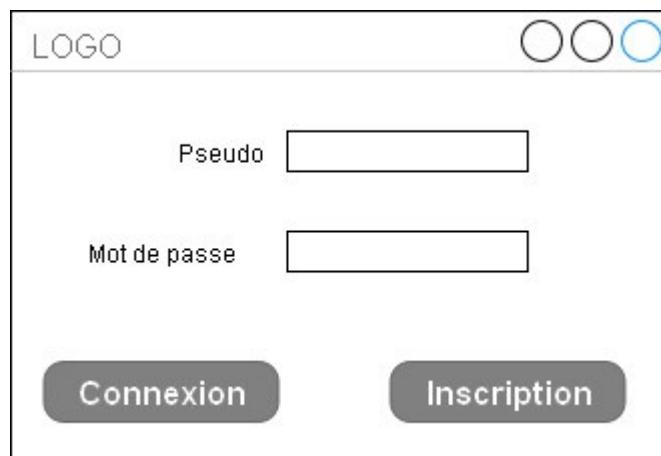
- Mauvaise configuration avec .gitignore en début de semaine de projet.
  - Le problème a été réglé dès le premier mardi.
- Difficultés pour l'implémentation multijoueur des jeux.
  - Mastermind opérationnel.
  - Puissance 4 fonctionnel localement, mais inutilisable en multijoueur.
- Développement des jeux comme des applications à part entière.
  - Intégration difficile des jeux sur la plateforme.
  - Nécessité d'utiliser plusieurs fils d'exécution (*threads*).

Ensuite, ce projet a permis, d'une part, un apprentissage au niveau humain. En effet, celui-ci a souligné l'importance du travail de groupe et a donné une belle opportunité de son apprentissage, qui pourra ainsi servir par la suite dans la vie professionnelle. Le projet a également montré que le partage et la communication dans le groupe sont des valeurs considérables pour réaliser et accomplir le travail. D'autre part, ce projet a permis de s'approprier de nouvelles connaissances au niveau technologique, comme par exemple les outils de travail collaboratif. Ainsi, ce travail a fourni de l'expérience pour de futurs projets dans l'avenir, espérons-le, proche.

Enfin, le projet est toujours améliorable. En effet, l'IHM n'est pas satisfaisante pour l'ensemble du groupe, et constitue donc un point d'amélioration possible ; par exemple, rendre la plateforme plus attrayante et moderne au lieu de rester dans le gris classique des applications. De plus, le Puissance 4 n'est pas terminé et doit pouvoir être joué en multijoueur car celui-ci doit être livré avec l'application. De plus, nous devons mettre en place le système de parties ouvertes comme ce qui était demandé par l'entreprise et, pour finir, le serveur distant utilisé par l'application est assez lent, ralentissant de surcroît l'application.

## **VI. Annexes**

## 1. Annexes I.H.M.



LOGO 

Pseudo :

Adresse email :

Confirmation adresse email :

Mot de passe :

Confirmation mot de passe :

**Valider** **Déjà inscrit ?**

LOGO 

JEUX	PROFIL	CONVERSATION	PARTIE
PHOTO DU JEU	PHOTO DU JEU		Photo du jeux
Nom du jeu	Nom du jeu		Nom du jeu
Description du jeu	Description du jeu		Description complete du jeu
PHOTO DU JEU	PHOTO DU JEU		<b>Jouer</b>
Nom du jeu	Nom du jeu		
Description du jeu	Description du jeu		

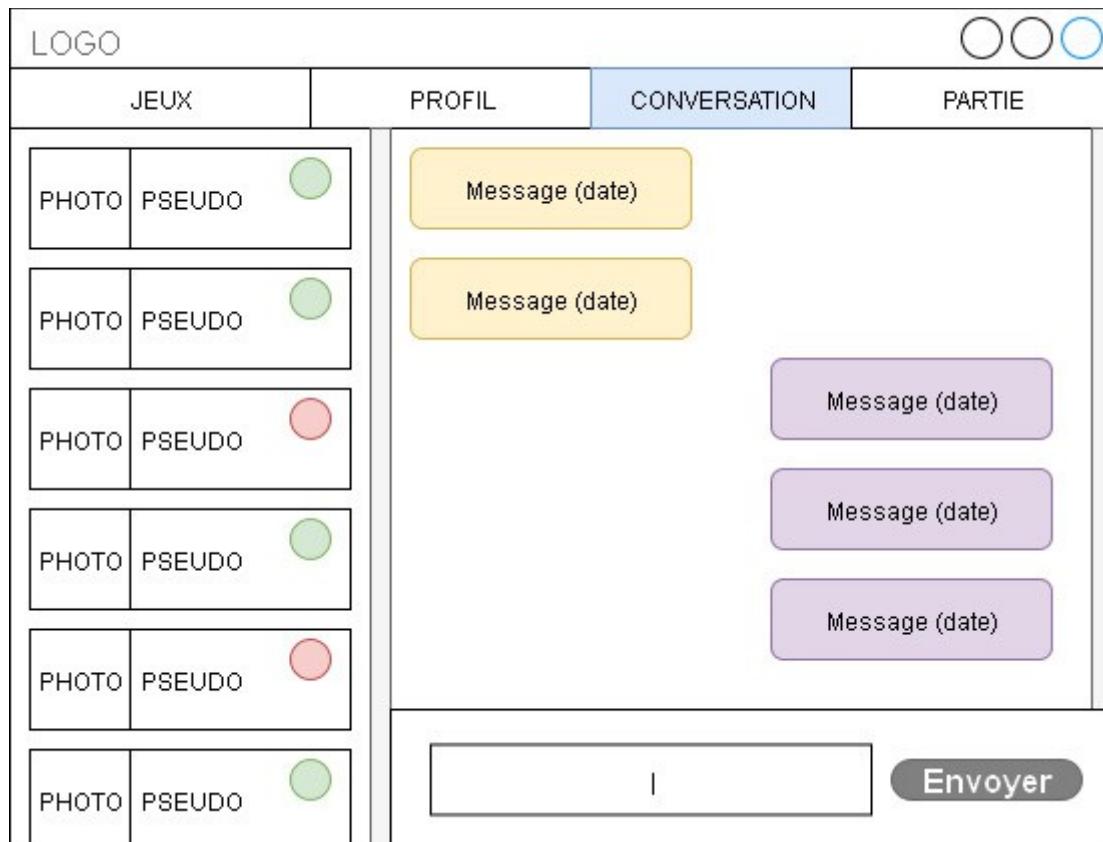
LOGO		OOO																		
JEUX	PROFIL	CONVERSATION	PARTIE																	
	<b>PSEUDO</b> EMAIL: RÔLE: RATIO : 45%																			
LISTE D'AMIS (nbAmis) <span style="float: right;">A-Z En ligne</span> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text"/> chercher       </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">PHOTO</td> <td style="width: 25%;">PSEUDO</td> <td style="width: 25%;">PHOTO</td> <td style="width: 25%;">PSEUDO</td> </tr> <tr> <td>supprimer</td> <td>supprimer</td> <td>supprimer</td> <td>supprimer</td> </tr> <tr> <td style="background-color: #f0f0f0;">PHOTO</td> <td style="background-color: #f0f0f0;">PSEUDO</td> <td style="background-color: #f0f0f0;">PHOTO</td> <td style="background-color: #f0f0f0;">PSEUDO</td> </tr> <tr> <td>supprimer</td> <td>supprimer</td> <td>supprimer</td> <td>supprimer</td> </tr> </table>					PHOTO	PSEUDO	PHOTO	PSEUDO	supprimer	supprimer	supprimer	supprimer	PHOTO	PSEUDO	PHOTO	PSEUDO	supprimer	supprimer	supprimer	supprimer
PHOTO	PSEUDO	PHOTO	PSEUDO																	
supprimer	supprimer	supprimer	supprimer																	
PHOTO	PSEUDO	PHOTO	PSEUDO																	
supprimer	supprimer	supprimer	supprimer																	

LOGO		OOO		
JEUX	PROFIL	CONVERSATION	PARTIE	
	<a href="#">STATISTIQUE</a> <a href="#">INVITATION</a> <a href="#">AJOUTER UN AMI</a> <b>EDITER</b> <a href="#">DECONNEXION</a>			
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <b>Changer de mot de passe</b>          Ancien mot de passe : <input type="text"/>          Nouveau mot de passe : <input type="text"/>          Confirmation du Nouveau mot de passe : <input type="text"/> <div style="text-align: right; margin-top: 10px;"> <b>Valider</b> </div> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <b>Changer d'adresse email</b>          Nouvelle adresse email : <input type="text"/>          Confirmation du Nouvelle adresse email : <input type="text"/> <div style="text-align: right; margin-top: 10px;"> <b>Valider</b> </div> </div>				

LOGO		OOO											
JEUX	PROFIL	CONVERSATION	PARTIE										
	<a href="#">STATISTIQUE</a> <a href="#">INVITATION</a> <a href="#">AJOUTER UN AMI</a> <a href="#">EDITER</a> <a href="#">DECONNEXION</a>												
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <span style="float: right;">INVITATION A-Z date</span>  <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text"/> Recherche ▾       </div> <p>Total resultat: 10</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Nom du Jeu</td> <td style="width: 25%;">PSEUDO VS PSEUDO</td> <td style="width: 25%;">Nombre de manche</td> </tr> <tr> <td></td> <td></td> <td>Score du joueur 1</td> </tr> <tr> <td></td> <td></td> <td>Score du joueur 2</td> </tr> </table> </div>					Nom du Jeu	PSEUDO VS PSEUDO	Nombre de manche			Score du joueur 1			Score du joueur 2
Nom du Jeu	PSEUDO VS PSEUDO	Nombre de manche											
		Score du joueur 1											
		Score du joueur 2											

LOGO		OOO																		
JEUX	PROFIL	CONVERSATION	PARTIE																	
	<a href="#">STATISTIQUE</a> <a href="#">INVITATION</a> <a href="#">AJOUTER UN AMI</a> <a href="#">EDITER</a> <a href="#">DECONNEXION</a>																			
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <span style="float: right;">RECHERCHER UN AMI A-Z En ligne</span>  <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text"/> chercher       </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">PHOTO</td> <td style="width: 25%;">PSEUDO</td> <td style="width: 25%;">PHOTO</td> <td style="width: 25%;">PSEUDO</td> </tr> <tr> <td>ajouter</td> <td>ajouter</td> <td>ajouter</td> <td>ajouter</td> </tr> <tr> <td style="background-color: #f0f0f0;">PHOTO</td> <td style="background-color: #f0f0f0;">PSEUDO</td> <td style="background-color: #f0f0f0;">PHOTO</td> <td style="background-color: #f0f0f0;">PSEUDO</td> </tr> <tr> <td>ajouter</td> <td>ajouter</td> <td>ajouter</td> <td>ajouter</td> </tr> </table> </div>					PHOTO	PSEUDO	PHOTO	PSEUDO	ajouter	ajouter	ajouter	ajouter	PHOTO	PSEUDO	PHOTO	PSEUDO	ajouter	ajouter	ajouter	ajouter
PHOTO	PSEUDO	PHOTO	PSEUDO																	
ajouter	ajouter	ajouter	ajouter																	
PHOTO	PSEUDO	PHOTO	PSEUDO																	
ajouter	ajouter	ajouter	ajouter																	

LOGO		OOO					
JEUX	PROFIL	CONVERSATION	PARTIE				
	<a href="#">STATISTIQUE</a> <a href="#">INVITATION</a> <a href="#">AJOUTER UN AMI</a> <a href="#">EDITER</a> <a href="#">DECONNEXION</a>						
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <span style="float: right;">STATISTIQUES A-Z %</span>  <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text"/> Recherche ▾       </div> <p>Total resultat: 10</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">PHOTO</td> <td style="width: 25%;">PSEUDO</td> <td style="width: 50%;"> <b>Ratio : 100%</b>            Nombre de partie gagné : 10            Nombre de partie jouer : 10         </td> </tr> </table> </div>					PHOTO	PSEUDO	<b>Ratio : 100%</b> Nombre de partie gagné : 10 Nombre de partie jouer : 10
PHOTO	PSEUDO	<b>Ratio : 100%</b> Nombre de partie gagné : 10 Nombre de partie jouer : 10					



## 2. Fiches individuelles

### a. Deuxième phase

#### I- Tâches réalisés

Durant la phase 2 j'ai été responsable du diagramme de GANTT du projet. Par la suite je me suis occupé du module jeux et plus précisément de la partie Mastermind de ce module. Dans la partie Mastermind j'ai été acteur de la description textuelle du diagramme de cas d'utilisation, de la réalisation du diagramme de cas d'utilisation et du diagramme de séquence.

#### II- Tâche à réaliser

Ce qu'il me reste à faire et de réaliser l'IHM de la plateforme, de la communication entre la plateforme et la base de donnée, réaliser le code ainsi que l'interface graphique des jeux "puissance 4" et "Mastermind".

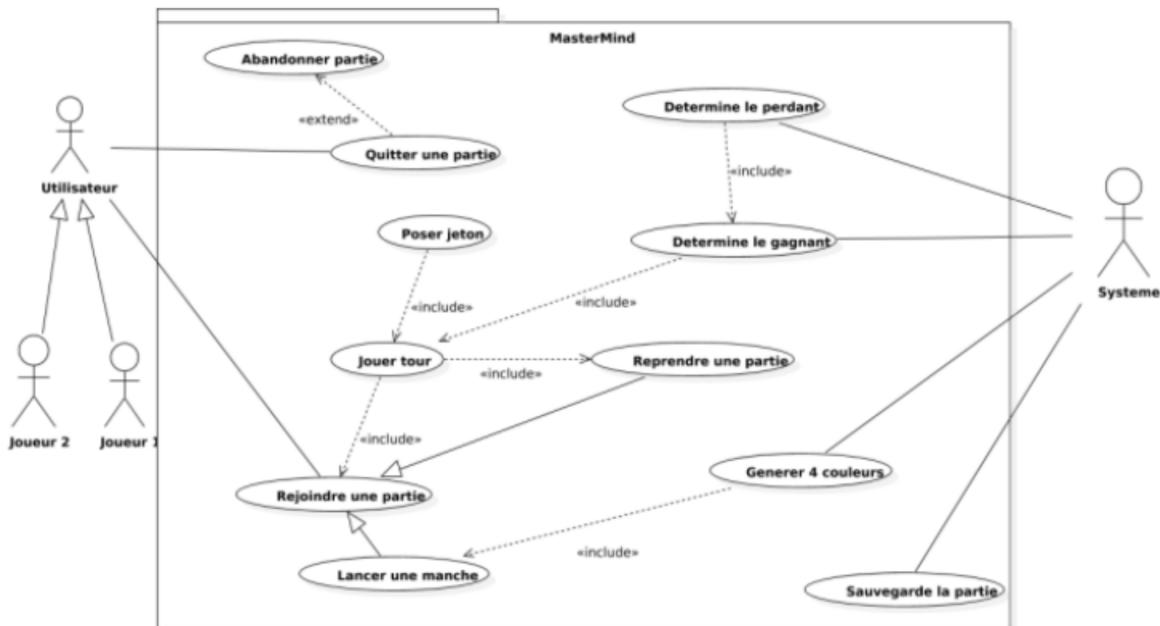


Illustration VI.1: AUBRY Erwan

## I – Tâches réalisées

Pour cette phase, j'ai été désigné chef du module « Jeu », qui comprend les deux jeux Puissance 4 et Mastermind avec notamment leurs réalisations et leurs modélisations.

Pour cette phase, j'ai été spécifiquement chargé de la réalisation du Diagramme de Cas d'Utilisation avec CIZEAU Valentin ainsi que de son adaptation sous forme de diagramme textuel.

## II – Les tâches restantes

L'analyse UML et le modèle des tâches des jeux devront donc être réalisés pour la troisième phase, ainsi que les IHM de ceux-ci. Une nouvelle fiche individuelle sera également rédigée.

*Illustration VI.2: BEAUVASIS Antoine*

### I- Tâches réalisés

Durant la phase 2, le groupe a été séparé en 2 groupe en fonction des modules. Le groupe se chargeant du module "Joueur et Administrateur" est composé de Axel Collet, Marie Christophe, Théodore Fruchart et moi-même. Donc concernant ce module, j'ai été acteur de la création des diagrammes de cas d'utilisation. Par la suite, avec l'aide de Théodore je me suis occupé des descriptions textuelles des cas d'utilisation et des diagrammes d'activité concernant le module Joueur. Les diagrammes d'activité ont été compliqué à faire étant donné que nous n'avons pas encore fais de cours concernant ce sujet.

### II- Tâche à réaliser

Ce qu'il me reste à faire , ce sont les diagramme de séquences, l'analyse UML, le modèle des tâches et les IHM concernant le module "Joueur et Administrateur". Je devrais aussi faire une fiche individuel sur le travail que j'aurais fait et qu'il me reste à faire.

*Illustration VI.3: BIZOT Jérôme*

#### **1. Le travail effectué :**

## Diagramme des cas d'utilisation Puissance 4 :

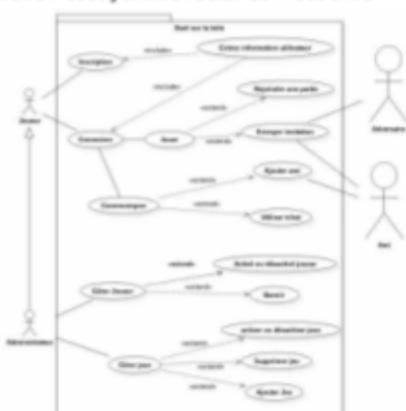
Ce diagramme a été fait par mes soin avec l'aide d'Antoine Beauvais en anglais, nous avons essayé de faire un diagramme le plus simple possible pour que les demandeurs puissent comprendre l'essentiel



## Diagramme des cas d'utilisation Détailé

Pour ce diagramme j'ai été acteur avec Erwan Aubry et supervisé par Antoine Beauvais

Nous avons essayé de détailler le plus possible les fonctionnalités qui sera présente sur la plateforme tout en essayant de rester sur l'essentiel



Description textuelle du diagramme des cas d'utilisation du puissance 4

Cette description a été réalisé avec Antoine Beauvais en anglais pour décrire les différents scénario nominal et alternatif en cohérence avec le Diagramme des cas d'utilisation

## 2. Le travail a réalisé :

Pour la phase 3 je devrai m'occuper de l'analyse UML, de la conception de l'IHM avec Antoine Beauvais et Erwan Aubry

### *Illustration VI.4: CIZEAU Valentin*

Tout d'abord, durant cette phase, j'ai attribué à chaque membre de l'équipe plusieurs choses à faire. J'ai donné à Erwan Aubry la charge de faire le planning prévisionnel GANTT et j'ai divisé l'équipe en 2 groupes : Erwan, Antoine et Valentin qui doivent s'occuper du module "Jeux" et Théodore, Jérôme, Axel et moi sur le module "Joueur et Administrateur". Ensuite, j'ai fait le modèle des fiches individuelles et j'ai complété la matrice RACI. Puis, avec ma sous-équipe, on a d'abord fait le diagramme des cas d'utilisation détaillé à partir de celui de la phase 1. Nous avons décidé de partager le diagramme de cas d'utilisation du joueur et de l'administrateur car il serait illisible et ce n'est pas le but de ce diagramme. En outre, avec Axel, nous avons fait les descriptions textuelles de ces diagrammes. Enfin, j'ai supervisé le sous-groupe du module "Jeux" pour savoir s'il y avancé et tous les diagrammes d'activité du module "Joueur et Administrateurs".

Il me restera à faire, durant la phase 3, tout ce qui comporte le module "Joueur et Administrateur" avec toujours le même sous-groupe. Il s'agit de :

- Les diagrammes de séquences des cas d'utilisations qui ont été reporté suite au retard des cours.
- L'analyse UML du module.
- Les maquettes de l'IHM, qui devront être très précise car elles ont vocation à être réalisées.
- Une nouvelle fiche individuelle où je devrais présenter les différences entre ce que j'avais prévu de faire et ce que j'ai fait.
- Le modèle des tâches du module Joueur et Administrateur.
- Une partie du rapport de PPP.

A la phase 4, il me restera à attribuer à chacun le travail à faire avec les souhaits de chacun.

*Illustration VI.5: CHRISTOPHE Marie*

### **I- Tâche réalisés**

Lors de cette phase 2 j'ai travailler sur le module administrateur pour cela j'ai coopérer avec Bizot Jérôme, Christophe Marie, Collet Axel, Fruchart Théodore.

Pour le module administrateur de la phase 2 j'ai été acteur sur le diagramme de cas d'utilisation détailler et du diagramme de cas d'utilisation textuelle

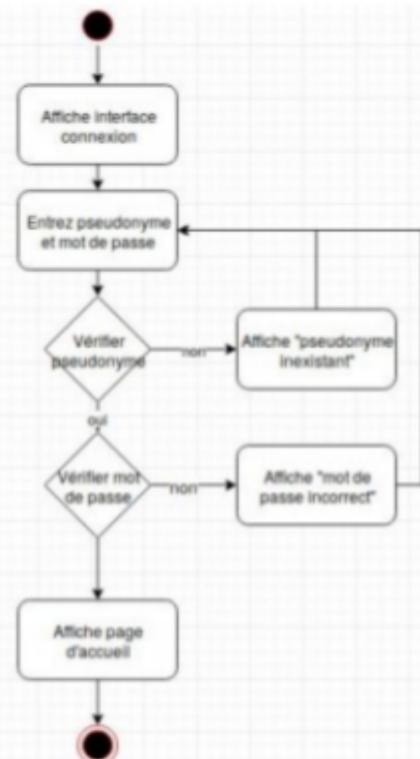
### **II-Tâche réalisés**

Lors de la phase 3 et 4 je travaillera sur le module administrateur et joueur notamment sur l'analyse Uml de joueur et administrateur ainsi que les IHM et une nouvelle fiche personnelle.

*Illustration VI.6: COLLET Axel*

### Tâches réalisées :

Au cours de la phase 2, j'ai travaillé sur le module Joueur et Administration. J'ai donc créé, avec Marie, Axel et Jérôme, un diagramme des cas d'utilisations décrivant les fonctionnalités auquel auront accès les Joueurs. Cependant, le diagramme de cas d'utilisation seul est insuffisant pour décrire précisément ces fonctionnalités. Ainsi, avec Jérôme, nous avons rédigé une description textuelle de chacun des cas d'utilisations et des diagrammes activité qui décrivent les cas d'utilisation de manière plus visuelle. J'ai aussi écrit la fiche individuelle que vous êtes en train de lire et la description précise de ce que j'ai fait dans le rapport de PPP



### Tâches à réaliser :

Il nous reste encore de nombreuses tâches à accomplir. En effet, lors de la phase 3, je travaillerai toujours sur le module Joueur et Administrateur. Je devrais donc réaliser, avec Marie, Jérôme et Axel :

- Les diagrammes de séquences des cas d'utilisations qui ont été reporté suite au retard des cours.
- L'analyse UML du module.
- Les maquettes de l'IHM, qui devront être très précise car elles ont vocation à être réalisées.
- Une nouvelle fiche individuelle où je devrais présenter les différences entre ce que j'avais prévu de faire et ce que j'ai fait.
- Le modèle des tâches du module Joueur et Administrateur.
- Une partie du rapport de PPP.

En phase 4, j'implémerai le module Joueur et Administrateur et me chargerai essentiellement d'implémenter les fonctions java et JDBC.

*Illustration VI.7: FRUCHART Théodore*

## b. Troisième phase

### I- Tâches réalisés

Durant la phase 3 je me suis occupé de des diagrammes de tâches pour le mastermind et la partie connexion et inscription à la plateforme. Par la suite je me suis occupé de l'IHM de la plateforme. Je me suis également occupé d'une partie de l'état de l'avancement du projet.

### II- Tâche à réaliser

Ce qu'il me reste à faire c'est de coder les IHM de la plateforme ainsi que le mastermind.

*Illustration VI.8: AUBRY Erwan*

## Écriture du rapport

Au cours de cette troisième phase, j'ai notamment été chargé de l'écriture du rapport qui sera rendu à l'issue du projet. Cela comprend notamment l'incorporation des maquettes de l'Interface Homme Machine.

## Description textuelle de l'IHM

J'ai également contribué à la réalisation de la description textuelle de l'I.H.M. avec Jérôme BIZOT.

*Illustration VI.9: BEAUV AIS Antoine*

### I- Tâches réalisés

Durant la phase 3, j'ai travaillé sur le modèle des tâches des parties joueur et administrateur avec l'aide de Théodore Fruchart. Je me suis aussi occupé de la conception de l'IHM en m'appuyant justement sur le modèle des tâches que nous avions précédemment créé.

### II- Tâche à réaliser

Ce qu'il me reste à faire , c'est le codage de la plateforme que je vais faire avec Théodore Fruchart et notre chef de projet , Marie Christophe.

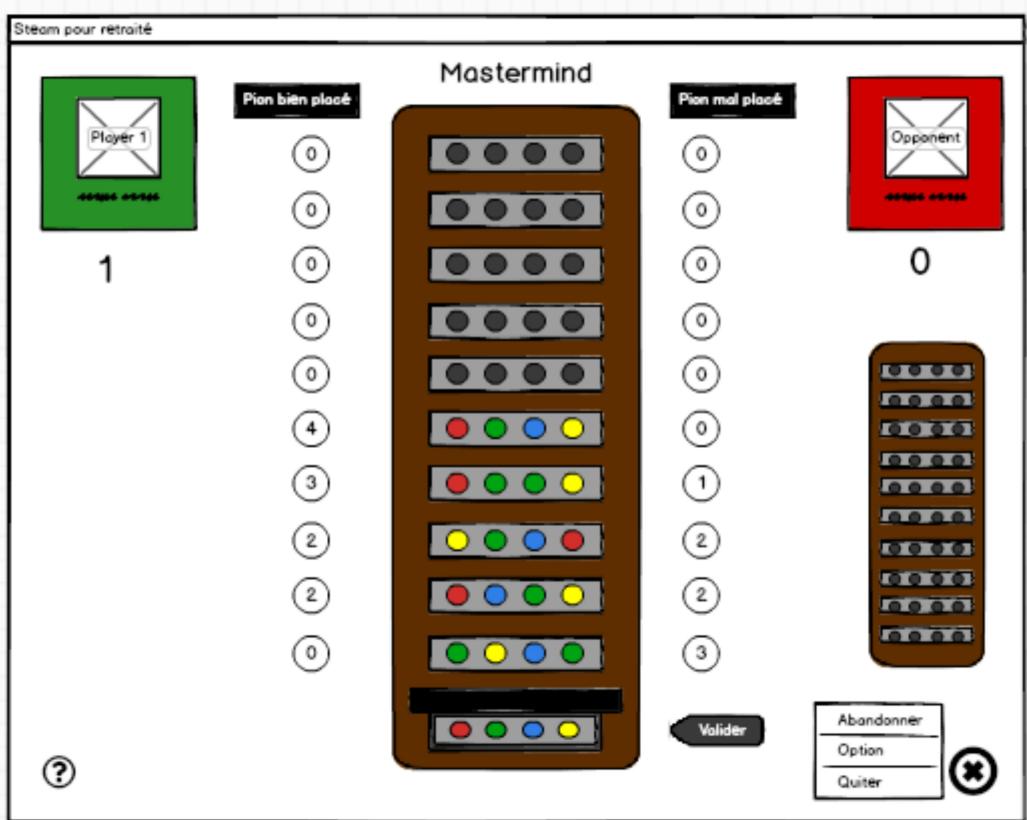
Ensuite, je contribuerais à finaliser le rapport une fois que le code sera fini .

*Illustration VI.10: BIZOT Jérôme*

### 1. Le travail effectué :

#### IHM Mastermind :

Cette a été fait par mes soin, j'ai essayé de faire une IHM la plus simple possible pour que l'utilisateur puissent naviguer facilement et jouer sans se perdre par une avalanche d'information, pour quelle soit la plus simple et belle possible j'ai demandé à chaque étape les avis des membres du projet pour pouvoir l'améliorer au mieux



#### IHM Mastermind - description textuel :

Cette description a été réalisé pour expliquer l'IHM et les choix qui été fait au demandeur

#### Analyse UML:

j'ai aidé les autres membre du groupe pour l'analyse UML et certain diagramme qui était inachevé

### 2. Le travail a réalisé :

Pour la suite du projet je devrai m'occuper de la programmation du Mastermind donc le jeu et l'IHM avec Erwan Aubry

Illustration VI.11: CIZEAU Valentin

Tout d'abord, durant cette phase, j'ai remis à jour le gantt qui a été plutôt bien respecté. Ensuite, j'ai réalisé presque toutes les IHM de la plateforme vue par un joueur seulement avec l'aide de Erwan et Jérôme a fait les dernières IHM (surtout de la partie de l'administrateur). Enfin, avec Théodore, nous avons fait le diagramme de classe de la plateforme global.

A la phase 4, j'ai attribué à chacun une partie du code et donné les règles pour coder correctement sur le git.

*Illustration VI.12: CHRISTOPHE Marie*

Lors de la phase 3 j'ai été donné de faire certains diagrammes de séquence, de tâche et faire l'IHM du connect 4.

Lors de la phase 4, nous devrons commencer le codage de l'application et je travaillerai sur le code du puissance 4.

*Illustration VI.13: COLLET Axel*

### **Tâches réalisées :**

Lors de la phase 3, j'ai travaillé, comme indiqué dans la fiche précédente, sur le module "Joueur et Administrateur". Ainsi, avec Jérôme, j'ai réalisé les modèles des tâches du module, qui ont pour rôle de définir le déroulement des actions des utilisateurs de manière à pouvoir s'en servir pour créer les IHM. Nous avons aussi entamé l'organisation du code. En effet, toujours avec Jérôme, nous avons réalisé un diagramme de séquence pour chaque cas d'utilisation, afin de définir de quelle classe nous aurons besoin. Cependant, lorsque j'ai fait le diagramme de classe avec Marie, nous nous sommes aperçue que la manière dont nous avions pensé le code lors de l'élaboration des diagrammes de séquences n'était pas la meilleure et nous en avons choisi une autre. C'est pourquoi les diagrammes de séquences ne sont plus à jour.

### **Tâches à réaliser :**

En phase 4, je m'occuperai du code du module Joueur et Administrateur.

*Illustration VI.14: FRUCHART Théodore*

### c. Quatrième phase

Durant la phase 4 je me suis occupé principalement de la programmation du Mastermind. Le mastermind est capable d'être jouable en multijoueur via la base de donnée et avertir un adversaire de l'état de la partie (si elle est abandonné, gagné, perdu...).

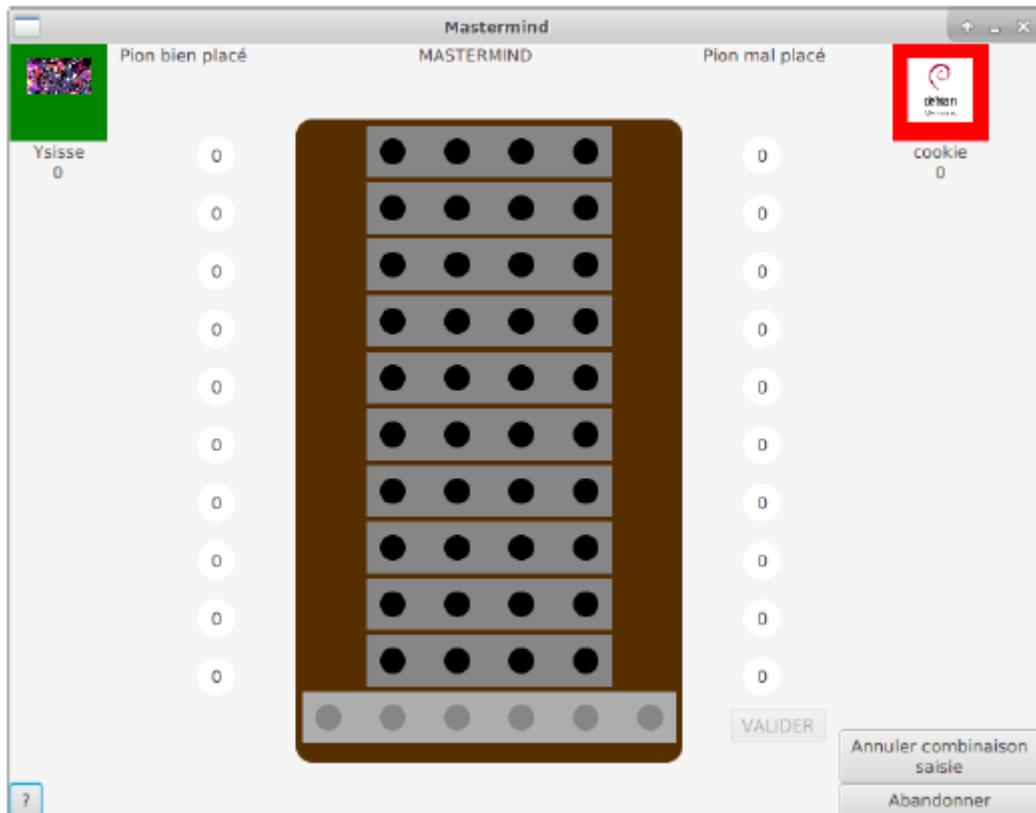
De plus je me suis également occupé de la gestion des invitation de parties et des parties en cours, c'est à dire, de pouvoir créer une partie, inviter à une partie, la rejoindre et l'abandonner directement via la plateforme.

Durant le développement du projet je me suis également occupé de la partie administration des jeux, où si l'on est admin nous pouvons changer le status d'un jeu, ajouter un jeu, et modifier un jeu.

J'ai améliorer l'interface de la page conversation de la plateforme.

Pour finir je me suis occuper de la correction de certain bug présent sur le mastermind et la plateforme en general, comme le menu de la plateforme, l'inscription d'un joueur, ....

*Illustration VI.15: AUBRY Erwan*



*Illustration VI.16: AUBRY Erwan (image 1)*



*Illustration VI.17: AUBRY Erwan (image 2)*

### Illustration VI.18: AUBRY Erwan (image 3)

J'ai travaillé sur le code de Puissance 4 avec Axel. La version local fonctionne mais la version multi ne fonctionne pas. Ainsi, il restera cela à faire fonctionner.

De plus, j'ai confectionné le manuel d'utilisateur et finit le rapport.

Tout au long des 2 semaines de projet, j'ai aidé mes camarades pour leurs fonctions et aidé Marie pour son bug avec ses images.

*Illustration VI.19: BEAUV AIS Antoine*

Le travail effectué :

Administration utilisateur :

J'ai réalisé l'affichage et les fonctionnalité de l'administration des joueurs, les administrateur pourront sélectionné les différent utilisateur de la plateforme et changer certaines de leurs données.

Vérification Mastermind :

J'ai aidé Erwan Aubry sur la conception du mastermind et revue le code pour qu'il soit le moins lourd possible pour la plateforme.

Débogage de l'application :

J'ai débogué certaine fonctionnalité de l'application comme par exemple le conflit entre les requêtes du profil et les requêtes de l'administration en les modifiants.

Non effectué :

J'ai un regret sur le projet, par manque de temps, dans le tableau des utilisateurs via la partie administrateur, changer la colonne "Etat" qui contient les "True" et "False" par des CheckButton ce qui peut rendre l'affichage un peu plus lisible.

Un mot sur le projet:

Ce projet m'a permis de découvrir de nouvelle fonctionnalité sur javaFX mais également de créer une meilleur complicité avec les membre du groupe et d'avoir pus m'enrichir personnellement en terme de méthode de programmation et de conception pour les années à venir.

*Illustration VI.20: CIZEAU Valentin*

Durant ces 2 semaines de projet, je me suis occupée de donner à chacun une partie du code que je pense que la personne pouvait réaliser. J'ai donc donné le code de Puissance 4 à Axel et Antoine, le code de Mastermind à Erwan, le code de la partie profil (et les sous menu) et la partie de la page des jeux à Théodore, la partie code de l'administrateur est pour Jérôme et Valentin et donc je me suis occupée de la partie de l'application en générale donc de switch les fenêtres et de la partie Partie de l'application. La partie Connexion et tout ce qui va avec c'est tout le monde qui s'en est occupé.

D'une part, grâce au code de M.Limet lors du dernier TP de BD, j'ai repris son système de switch les fenêtres pour l'application.

D'autre part, je me suis occupée de la partie du code de partie pour afficher la page des parties en cours et des invitations de parties. J'ai eu un problème avec les images pour faire l'IHM mais j'ai réussi à régler le problème. Et grâce à Erwan, nous avons fait les codes pour rejoindre une partie en cours et accepter une invitation de partie.

Le projet m'a fait réaliser que le poste de Chef de projet était assez dur à faire mais j'ai à chaque fois essayé de bien répartir les tâches à chacun. De plus, j'ai appris à mieux faire une application java et les erreurs qui ne faut pas faire et les bonnes pratiques. Enfin, j'ai compris que le travail de groupe était très important pour bien faire un développement car il y a certaine erreur qu'on arrive à rejoindre que grâce à nos camarades .

#### *Illustration VI.21: CHRISTOPHE Marie*

Durant cette 4 phase j'ai travaillé sur le puissance 4 accompagné principalement par Antoine Beauvais.

Je travaille en priorité sur l'IHM du puissance 4 ainsi que sur la gestion des warning et de quelques éléments de la documentation.

J'ai aussi aidé sur d'autre module tel que le module administrateur et le module conversation.]

#### *Illustration VI.22: COLLET Axel*

**Tâches réalisées :**

Lors de la phase 4, j'ai travaillé sur la plateforme "Duel Sur La Toile" et plus particulièrement sur le profil (Statistique, Édition, Invitation, Information et Historique) avec l'aide d'Erwan. J'ai aussi aider Valentin et Jérôme sur leur module Administrateur et Connexion.

J'ai bien travaillé sur ce que j'avais prévu dans ma fiche personnelle précédente, à savoir la plateforme.

**Rétrospective :**

Au cours de ce projet, j'ai beaucoup progressé en organisation en communication. De plus, je maîtrise à présent plusieurs outils qui m'étaient inconnus tel que mybalsamic ou netbeans. Ensuite, je suis maintenant très rapide dans la réalisation de diagramme d'activité, de séquence ou encore de cas d'utilisation. Enfin, je suis plus efficace dans la création d'IHM et dans l'utilisation des bibliothèques javafx et JDBC.

*Illustration VI.23: FRUCHART Théodore*

### 3. Annexes SQL

```
CREATE TABLE UTILISATEUR(
    pseudoUtilisateur Varchar (50) NOT NULL ,
    mailUtilisateur Varchar (50) NOT NULL ,
    passwdUtilisateur Varchar (50) NOT NULL ,
    roleUtilisateur Varchar (50) DEFAULT 'Utilisateur',
    activeUtilisateur Bool DEFAULT 1,
    PRIMARY KEY (pseudoUtilisateur );
```

```
CREATE TABLE CORRESPONDRE(
    pseudoUtilisateur Varchar (50) NOT NULL ,
    idPartie      int NOT NULL ,
    score         int NOT NULL ,
    PRIMARY KEY (idPartie, pseudoUtilisateur ));
```

```
CREATE TABLE PARTIE(
    idPartie      int Auto_increment NOT NULL ,
    datePartie    Date NOT NULL ,
    numEtapePartie int NOT NULL ,
    infoJSONPartie Varchar (50) NOT NULL ,
    idJeu         int NOT NULL ,
    PRIMARY KEY (idPartie ));
```

```
CREATE TABLE STATISTIQUE_UTILISATEUR(
    idStat        int Auto_increment NOT NULL ,
    avgPtsStat   Float NOT NULL DEFAULT 0.0,
    freqJeuStat   Int NOT NULL DEFAULT 0,
    pseudoUtilisateur Varchar (50) NOT NULL ,
    PRIMARY KEY (idStat ) );
```

```
CREATE TABLE MESSAGE(
    idMessage     int Auto_increment NOT NULL ,
    destMessage   Varchar (125) NOT NULL ,
    corpMessage   Varchar (50) NOT NULL ,
    estLuMessage  Bool NOT NULL DEFAULT 0,
    nbCaractMessage Int NOT NULL DEFAULT 125,
    idConv       int NOT NULL ,
    PRIMARY KEY (idMessage ));
```

```
CREATE TABLE APPARTENIR(
    idConv        Int NOT NULL ,
    pseudoUtilisateur Varchar (50) NOT NULL ,
    PRIMARY KEY (idConv ,pseudoUtilisateur ) );
```

```
CREATE TABLE JEU(
    idJeu        int Auto_increment NOT NULL ,
    nomJeu       Varchar (50) NOT NULL ,
    descJeu      Varchar (150) NOT NULL ,
    sourceJeu    Blob NOT NULL ,
    typeJeu      Varchar (50) NOT NULL ,
    dispoJeu     Bool NOT NULL ,
    PRIMARY KEY (idJeu ));
```

```
CREATE TABLE INVITATION(
    idInvite     int Auto_increment NOT NULL ,
    dateInvite   Date NOT NULL ,
    typeInvite   Varchar (25) NOT NULL ,
    idPartie    int NOT NULL ,
    PRIMARY KEY (idInvite ));
```

```
CREATE TABLE RAPPORT(
    idRapport    int Auto_increment NOT NULL ,
    champTexteRapport Varchar (125) NOT NULL ,
    pseudoUtilisateur Varchar (50) NOT NULL ,
    PRIMARY KEY (idRapport ));
```

```
CREATE TABLE CONSULTER(
    idStatPlateforme  int NOT NULL ,
    pseudoUtilisateur Varchar(50) NOT NULL ,
    PRIMARY KEY (idStatPlateforme, pseudoUtilisateur ));
```

```
CREATE TABLE CONVERSATION(
    idConv        int Auto_increment NOT NULL ,
    PRIMARY KEY (idConv ));
```

```
CREATE TABLE ETRE_AMI(
    pseudoUtilisateur Varchar (50) NOT NULL ,
    pseudoUtilisateur_1 Varchar (50) NOT NULL ,
    PRIMARY KEY (pseudoUtilisateur ,pseudoUtilisateur_1 ));
```

```
CREATE TABLE STATISTIQUE_PLATEFORME(
    idStatPlateforme    Int Auto_increment NOT NULL ,
    NbUtilisateur connecte int NOT NULL ,
    PRIMARY KEY (idStatPlateforme ));
```

*Illustration VI.24: Script de création de la base de données.*

ALTER TABLE **INVITATION** ADD CONSTRAINT FK\_INVITATION\_idPartie FOREIGN KEY (idPartie) REFERENCES **PARTIE**(idPartie);

ALTER TABLE **PARTIE** ADD CONSTRAINT FK\_PARTIE\_idJeu FOREIGN KEY (idJeu) REFERENCES **JEU**(idJeu);

ALTER TABLE **RAPPORT** ADD CONSTRAINT FK\_RAPPORT\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **STATISTIQUE\_UTILISATEUR** ADD CONSTRAINT FK\_STATISTIQUE\_UTILISATEUR\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **CONSULTER** ADD CONSTRAINT FK\_CONSULTER\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **CONSULTER** ADD CONSTRAINT FK\_CONSULTER\_idStatPlateforme FOREIGN KEY (idStatPlateforme) REFERENCES **STATISTIQUE\_PLATEFORME**(idStatPlateforme);

ALTER TABLE **MESSAGE** ADD CONSTRAINT FK\_MESSAGE\_idConv FOREIGN KEY (idConv) REFERENCES **CONVERSATION**(idConv);

ALTER TABLE **ENVOYER** ADD CONSTRAINT FK\_ENVOYER\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **ENVOYER** ADD CONSTRAINT FK\_ENVOYER\_idInvite FOREIGN KEY (idInvite) REFERENCES **INVITATION**(idInvite);

ALTER TABLE **ENVOYER** ADD CONSTRAINT FK\_ENVOYER\_destinataire FOREIGN KEY (destinataire) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **ETRE\_AMI** ADD CONSTRAINT FK\_ETRE\_AMI\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **ETRE\_AMI** ADD CONSTRAINT FK\_ETRE\_AMI\_pseudoUtilisateur\_1 FOREIGN KEY (pseudoUtilisateur\_1) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **APPARTENIR** ADD CONSTRAINT FK\_APPARTENIR\_idConv FOREIGN KEY (idConv) REFERENCES **CONVERSATION**(idConv);

ALTER TABLE **APPARTENIR** ADD CONSTRAINT FK\_APPARTENIR\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **CORRESPONDRE** ADD CONSTRAINT FK\_CORRESPONDRE\_pseudoUtilisateur FOREIGN KEY (pseudoUtilisateur) REFERENCES **UTILISATEUR**(pseudoUtilisateur);

ALTER TABLE **CORRESPONDRE** ADD CONSTRAINT FK\_CORRESPONDRE\_idPartie FOREIGN KEY (idPartie) REFERENCES **PARTIE**(idPartie);

*Illustration VI.25: Clés étrangères de la base de données.*

```
CREATE TABLE ROLE (
    nomRole varchar(10),
    PRIMARY KEY (nomRole) );
```

```
CREATE TABLE MESSAGE (
    idMsg decimal(6,0),
    dateMsg datetime,
    contenuMsg text,
    luMsg char(1),
    idUt decimal(6,0),
    idUt1 decimal(6,0),
    PRIMARY KEY (idMsg) );
```

```
CREATE TABLE INVITATION (
    idInv decimal(6,0),
    dateInv datetime,
    etatInv char(1),
    idUt decimal(6,0),
    idUt1 decimal(6,0),
    PRIMARY KEY (idInv) );
```

```
CREATE TABLE UTILISATEUR (
    idUt decimal(6,0),
    pseudoUt varchar(10) unique,
    emailUt varchar(100),
    mdpUt varchar(100),
    activeUt char(1),
    nomRole varchar(10),
    PRIMARY KEY (idUt) );
```

```
CREATE TABLE PARTIE (
    idPa decimal(6,0),
    debutPa datetime,
    numEtape char(1),
    etatPartie text,
    idJeu decimal(6,0),
    idUt1 decimal(6,0),
    score1 VARCHAR(42),
    idUt2 decimal(6,0),
    score2 VARCHAR(42),
    PRIMARY KEY (idPa) );
```

```
CREATE TABLE ETREAMI (
    idUt decimal(6,0),
    idUt1 decimal(6,0),
    PRIMARY KEY (idUt, idUt1) );
```

```
CREATE TABLE JEU (
    idJeu decimal(6,0),
    nomJeu varchar(20),
    regleJeu text,
    jarJeu blob,
    activeJeu char(1),
    idTy decimal(6,0),
    PRIMARY KEY (idJeu) );
```

```
CREATE TABLE TYPEJEU (
    idTy decimal(6,0),
    nomTy varchar(20),
    PRIMARY KEY (idTy) );
```

```
ALTER TABLE UTILISATEUR ADD FOREIGN KEY (nomRole) REFERENCES ROLE (nomRole);
```

```
ALTER TABLE ETREAMI ADD FOREIGN KEY (idUt1) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE ETREAMI ADD FOREIGN KEY (idUt) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE MESSAGE ADD FOREIGN KEY (idUt1) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE MESSAGE ADD FOREIGN KEY (idUt) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE PARTIE ADD FOREIGN KEY (idUt1) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE PARTIE ADD FOREIGN KEY (idUt2) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE PARTIE ADD FOREIGN KEY (idJeu) REFERENCES JEU (idJeu);
```

```
ALTER TABLE INVITATION ADD FOREIGN KEY (idUt1) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE INVITATION ADD FOREIGN KEY (idUt) REFERENCES UTILISATEUR (idUt);
```

```
ALTER TABLE JEU ADD FOREIGN KEY (idTy) REFERENCES TYPEJEU (idTy);
```

*Illustration VI.26: Base de données corrigée.*