CONCEPTION ET RÉALISATION D'UNE PLATEFORME WEB ET MOBILE POUR LE JEU INTITULÉ "LES LOUPS GAROUS DE THIERCELIEU"

Mehdi Frikha

Jixuan Yang

Samuel VANIÉ

Emmanuel FEZEU

Mouahé

TABLE DES MATIERES

1. Analyse

1.1 Description des acteurs

- **Système** : gestion globale du jeu. Il se charge de lancer le jeu, le terminer, de changer son état, l'archivage continue des fils de discussions, la gestion des votes.
- **Temps** : le timer qui gère les signaux, le passage entre le jour et la nuit, qui gère aussi le temps d'attente avant de débuter une partie.
- **Hôte** : Celui qui a créé la partie.
- **Joueur** : une personne enregistrée dans la base de données.
- **Villageois** : L'un des participants d'une partie donnée.
- **Villageois mort** : un villageois tué durant la partie.
- **Loup Garou** : une des deux catégories de villageois.
- **Loup garou alpha** : un loup capable de contaminer un villageois.
- **Humain** : un villageois qui ne possède pas les pouvoirs du loup garou.
- **Petite fille** : une humaine qui peut voir qui discute la nuit chez les loups garou.
- **Voyante** : peut voir chaque nuit le rôle d'un villageois encore vivant
- **Sorcière** : Peut discuter avec un joueur mort. Une sorcière peut être une humaine ou un loup garou

A la page suivante on pourrait voir un diagramme qui représente l'héritage.

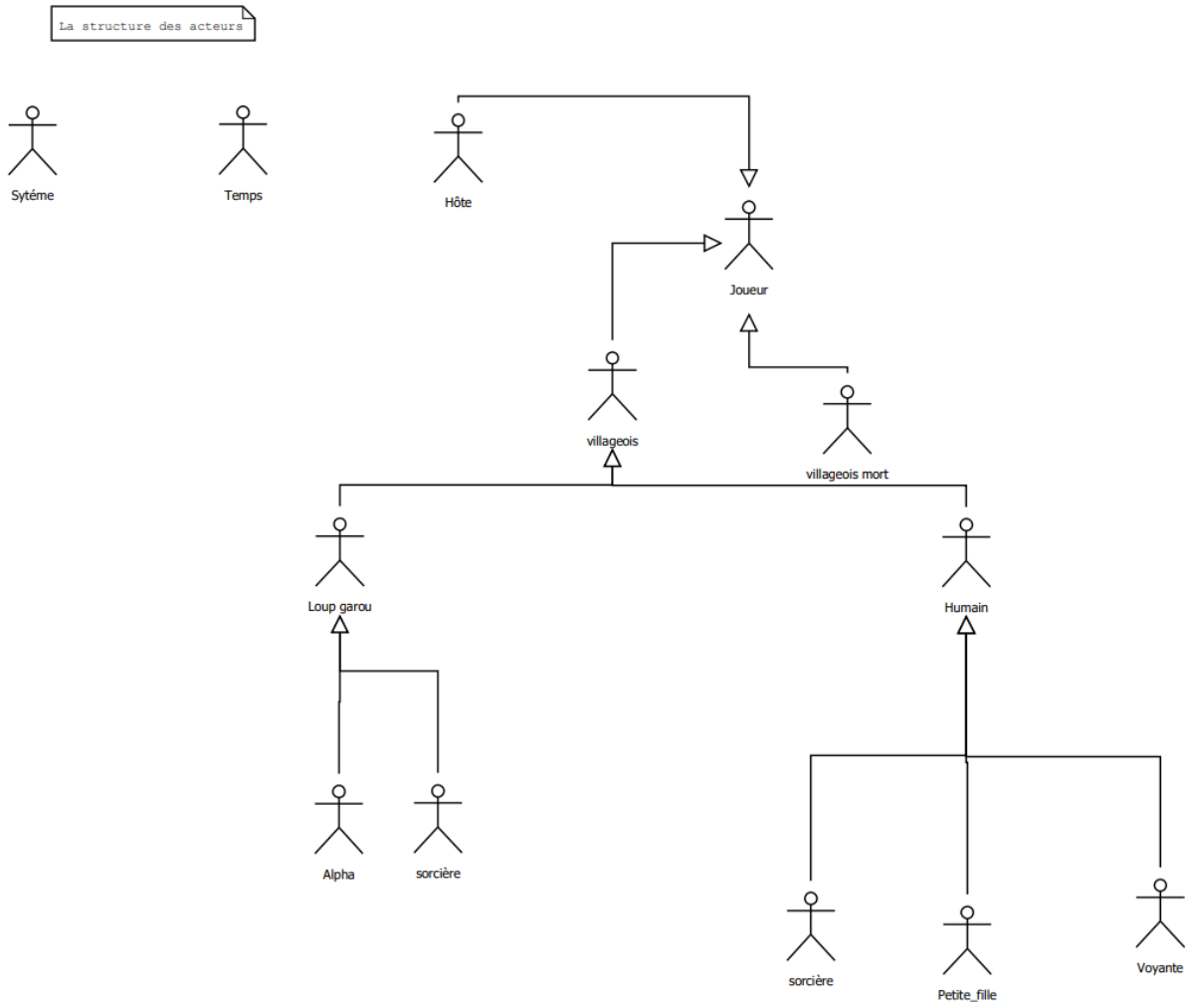


Figure 1.1 : les différents acteurs du systèmes et les relations.

1.2 Cas d’utilisation

Maintenant nous passons à l'élaboration des différents cas d'utilisations.

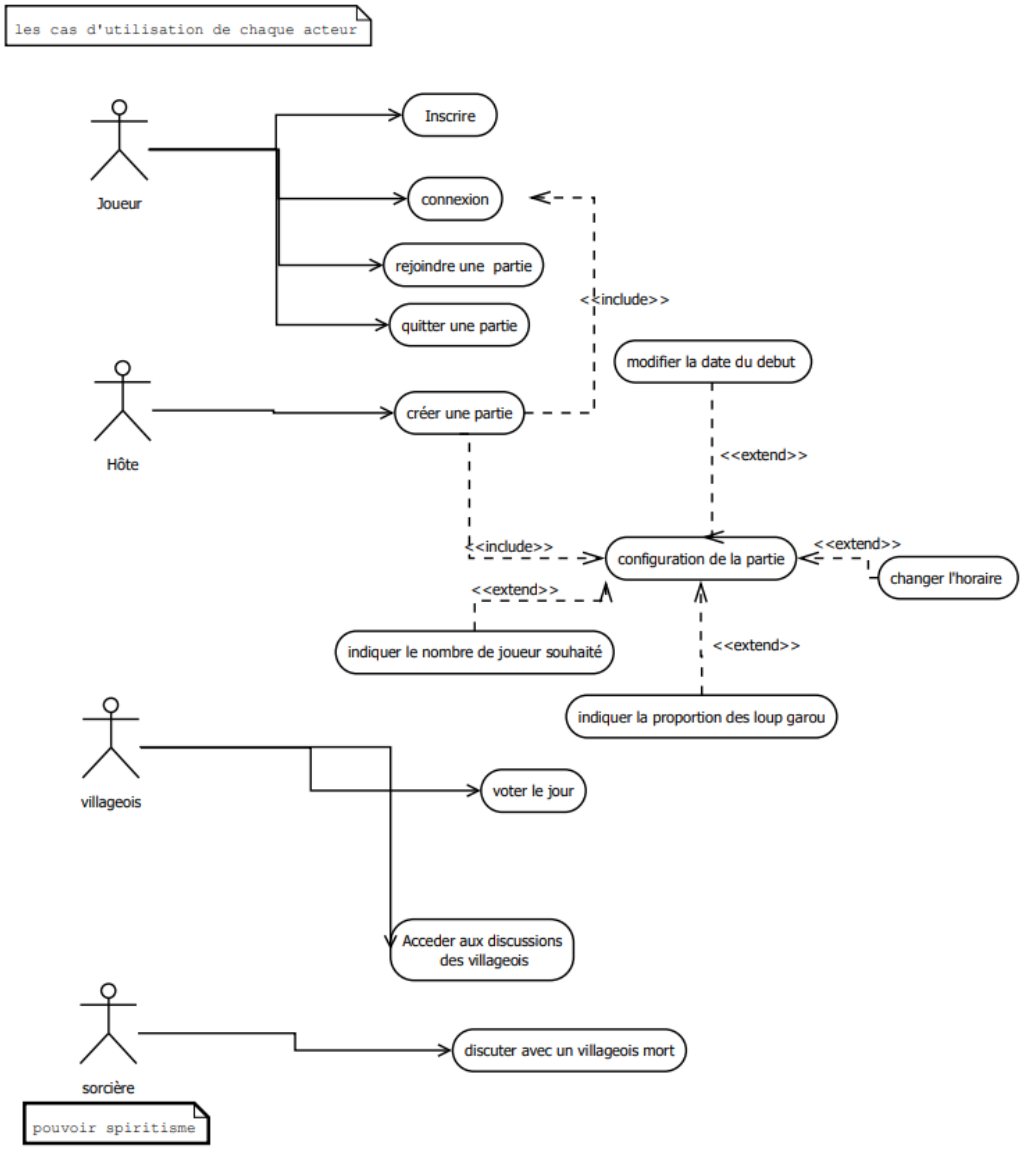


Figure 1.2 : les cas d’utilisation des acteurs Joueur, Hôte, Villageois et Sorcière

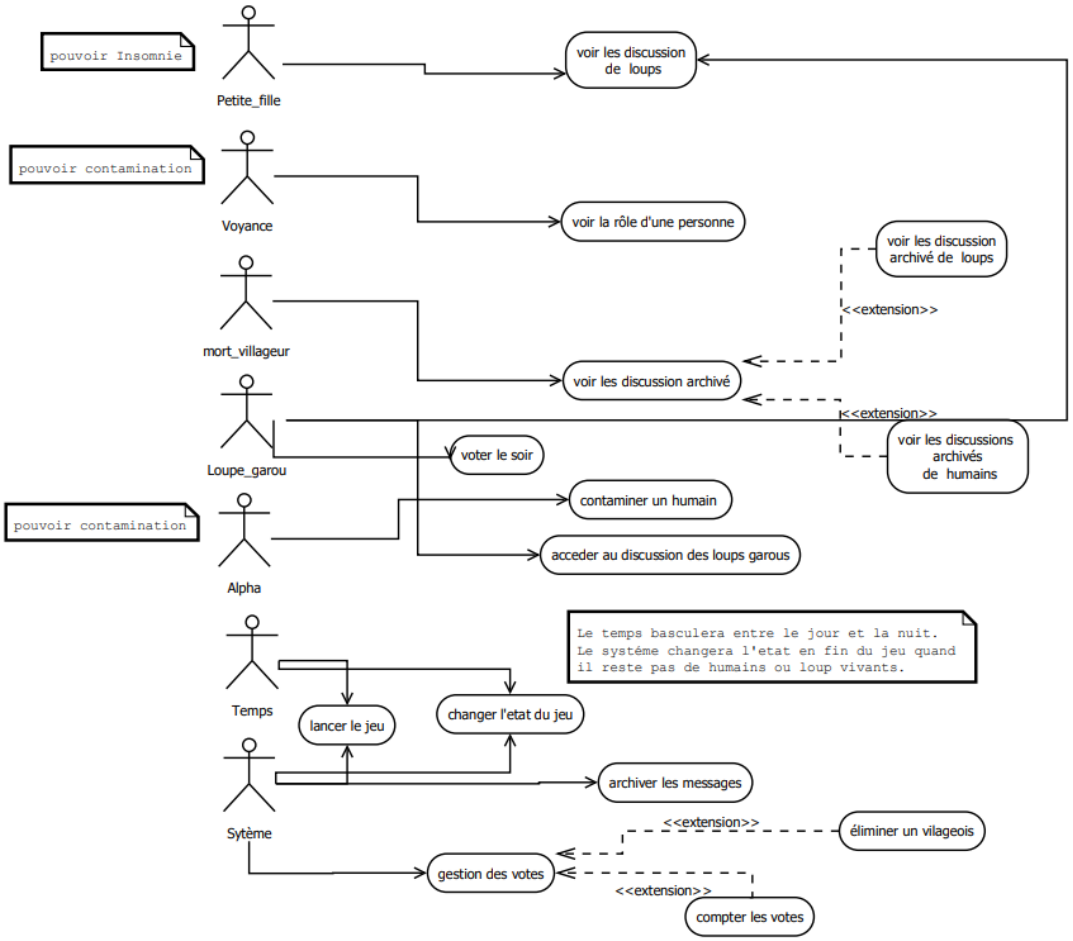


Figure 1.3 : les cas d'utilisation des acteurs (petite fille, le villageois mort, le loup garou et le loup garou alpha)

Cette figure montre aussi les actions des réalisées par le temps et le système.

1.3 Diagrammes de séquence

A partir du diagramme des cas d'utilisation nous obtenons les diagrammes de séquences ci dessous:

Dans le cas d'une connexion, l'utilisateur s'authentifie et le système vérifie les identifiants entrés. Après cela deux cas sont possibles, soit les données sont valides et l'utilisateur se connecte, soit elles sont invalides et l'on demande à l'utilisateur de recommencer.

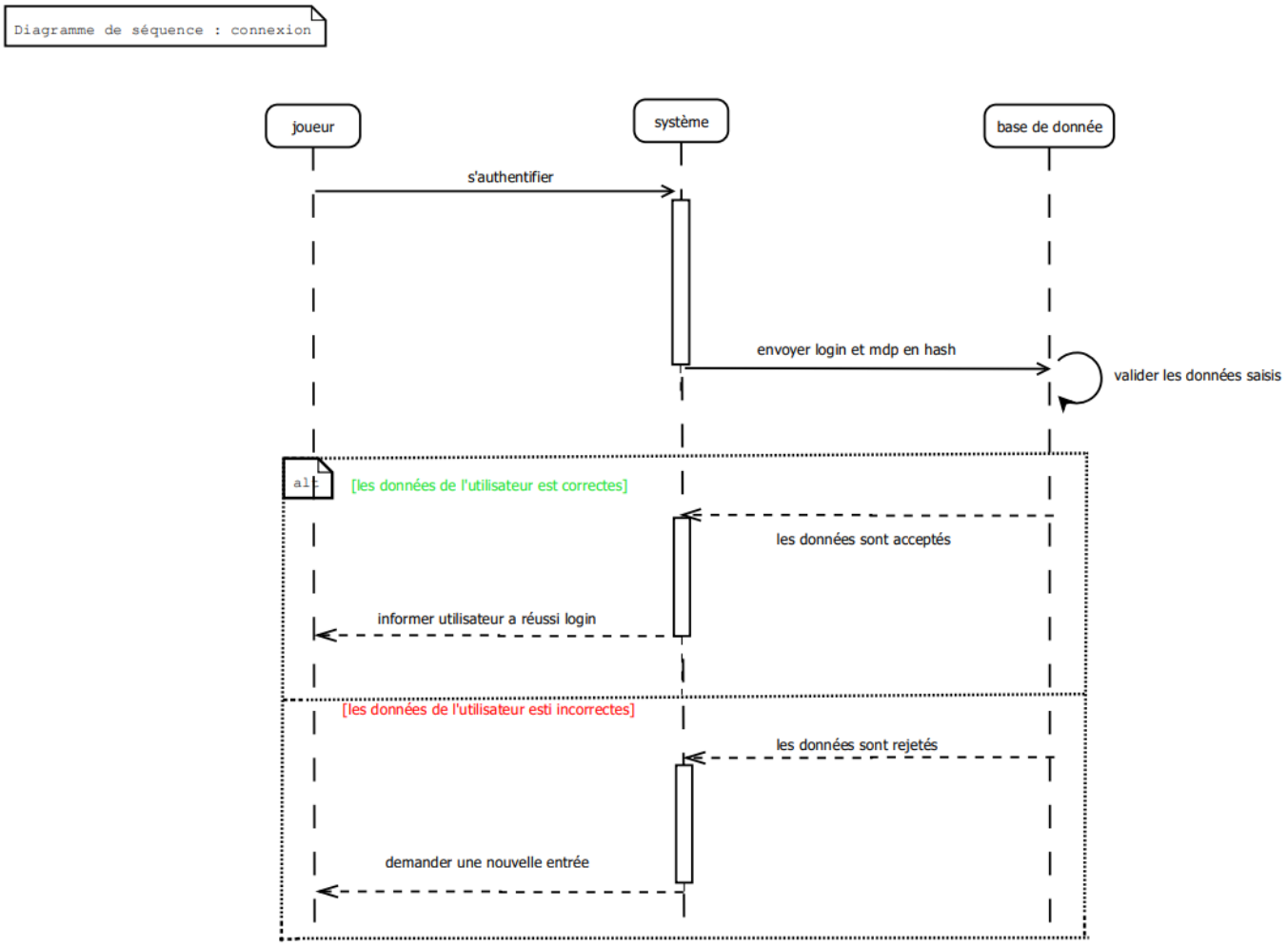


Figure 1.4 : Le diagramme de séquence pour la connexion.

Le cas de l'inscription et de la création est similaire théoriquement à celui de la connexion, à des détails près.

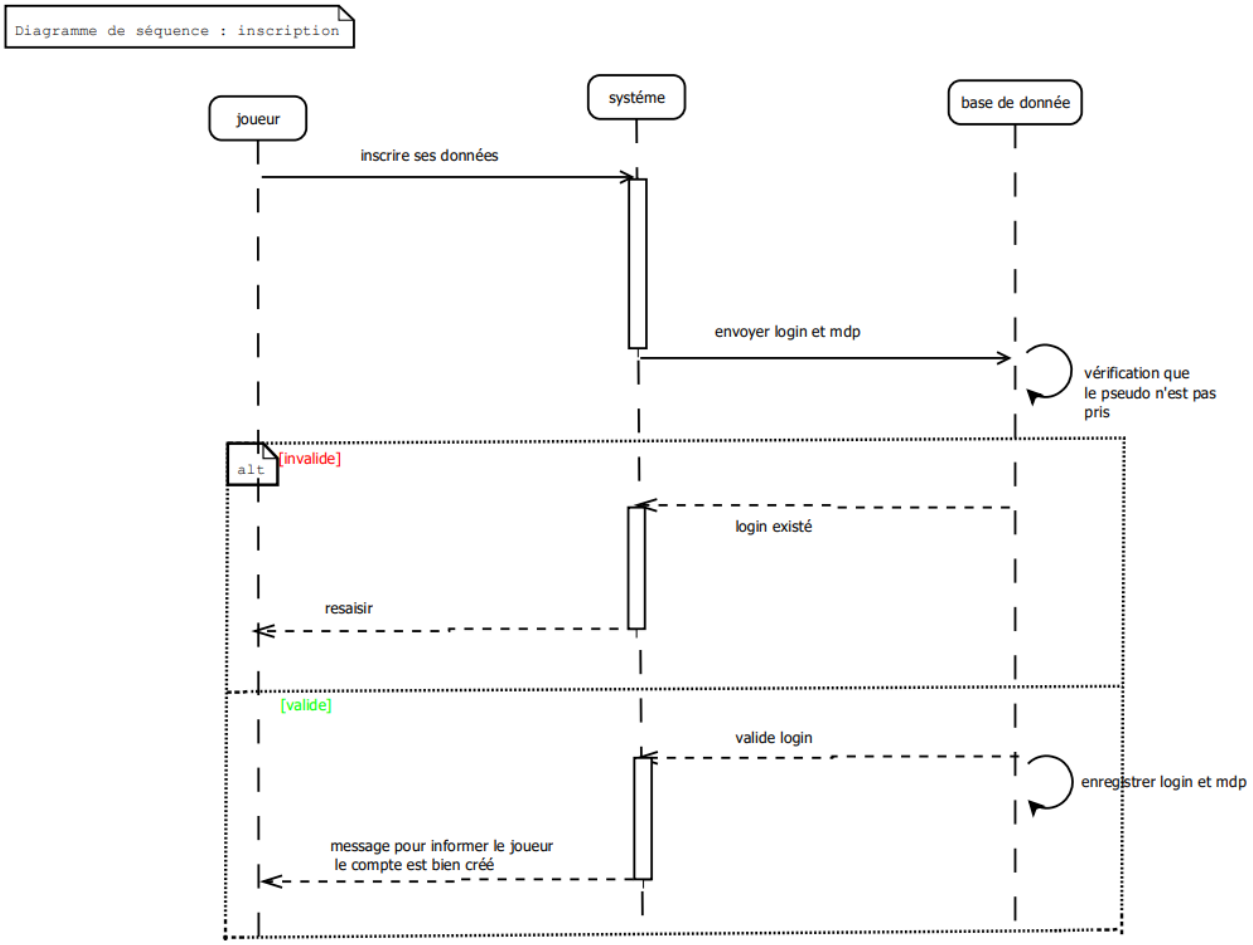


Figure 1.5 : Le diagramme de séquence pour l'inscription.

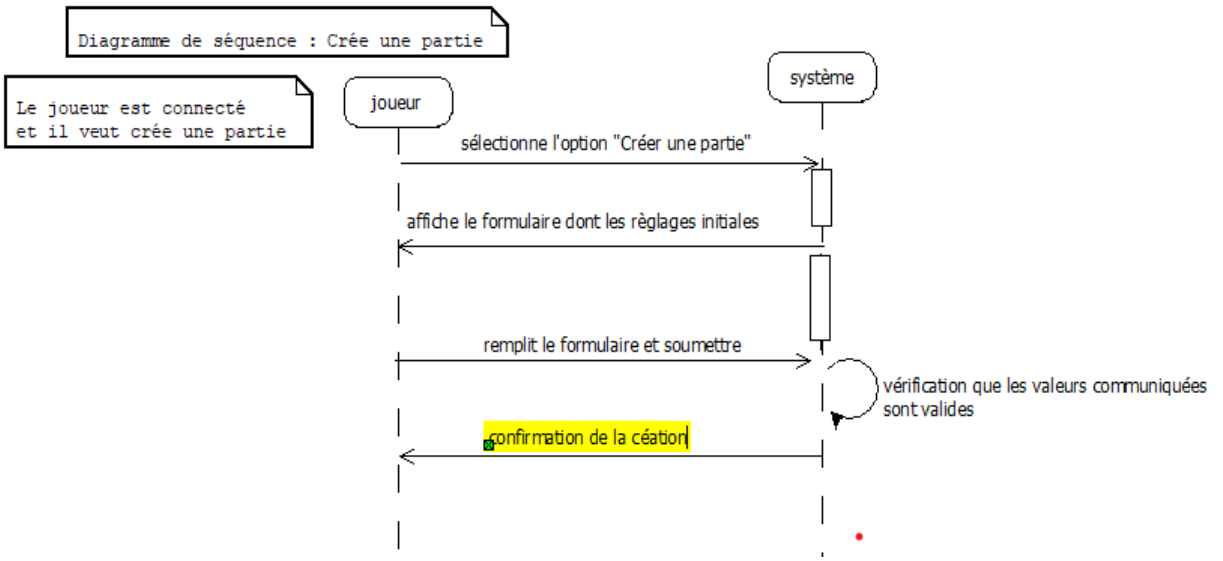


Figure 1.6 : Le diagramme de séquence pour la création de partie

Pour le cas où l'utilisateur veut rejoindre une partie, le système vérifie si la partie existe encore, et si le nombre maximum de joueurs n’est pas atteint, si toutes ces conditions sont réunies alors le joueur est ajouté à la partie, sinon la demande est invalidée et on le signale au joueur.

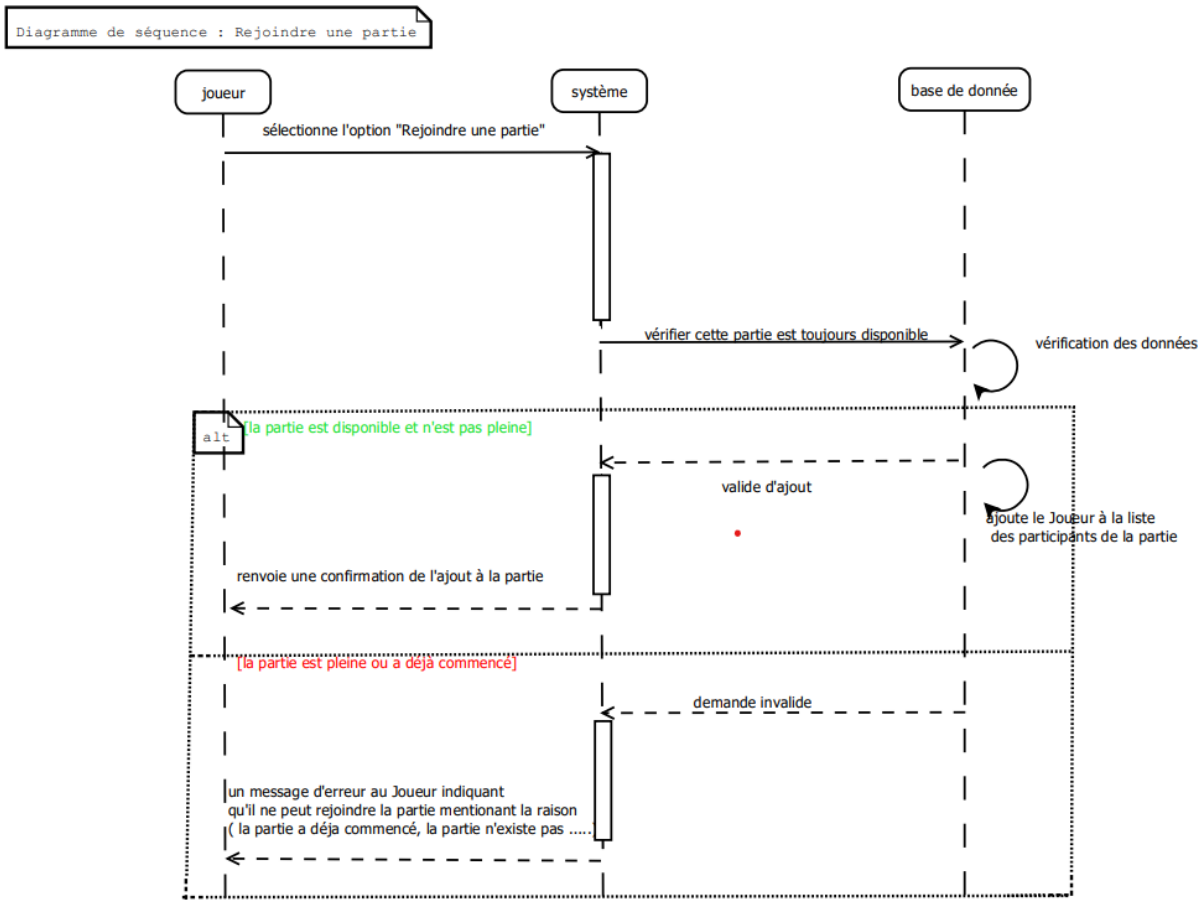


Figure 1.7 : Le diagramme de séquence pour rejoindre une partie.

Lorsqu'un joueur quitte une partie, on vérifie si la partie peut continuer sans lui car il est signalé dans la partie comme mort et dans le même temps on le déconnecte. si la partie peut continuer pas soucis. Sinon le jeu est arrêté et la victoire est déclarée pour l'un des camps.

Ces conséquences sont les mêmes appliquées dans les cas du vote, où, le début de chaque vote est envoyé au système et les résultats sont mis à jour continuellement (chaque joueur est informé du nombre de voix contre lui) et à la fin des votes le résultat final est affiché dans le fil de discussion.

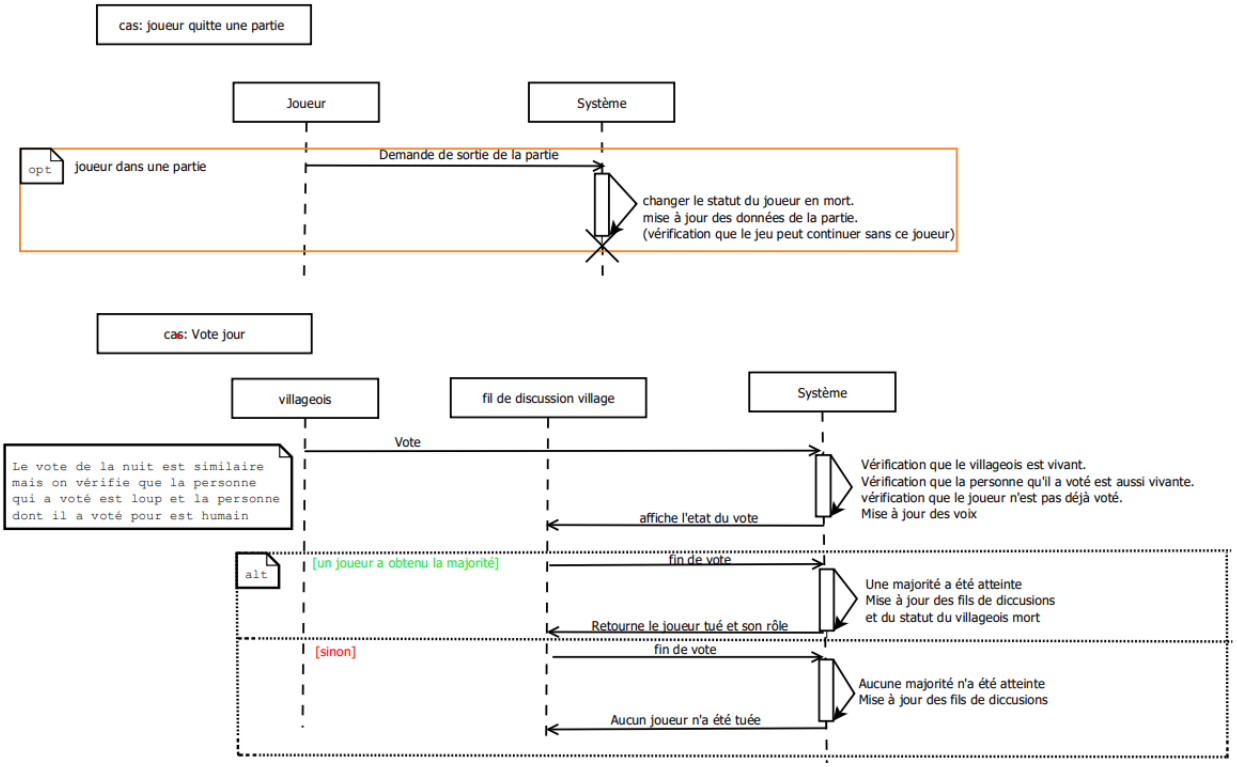


Figure 1.8 : Le diagramme de séquence pour quitter une partie.

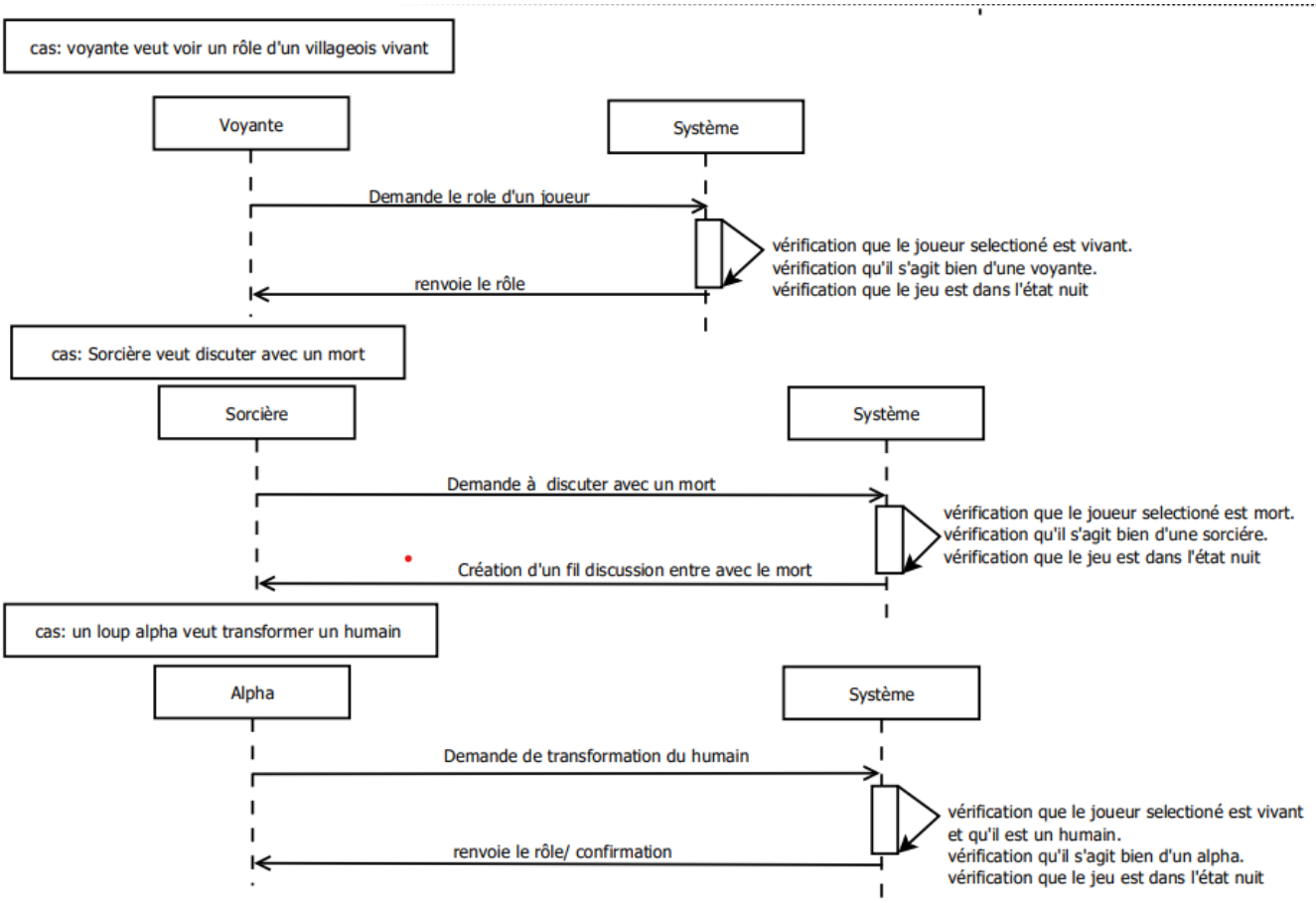


Figure 1.9 : Le diagramme séquence des trois pouvoirs qu'on peut utiliser.

Les joueurs peuvent posséder des pouvoirs. Dans le cas de la voyante, elle peut voir le rôle d'un autre joueur. Pour cela, elle clique sur l'action correspondante dans son menu qui envoie une requête au système, qui lui vérifie qu'elle est bien la voyante et que cette action se déroule pendant la nuit avant de répondre avec le rôle de l'utilisateur en question.

La sorcière, elle peut discuter avec un mort. Ce qui est très pratique pour connaître les soupçons de ceux-ci. Le fonctionnement dans ce cas est presque identique au précédent dans la mesure où il s'agit d'une demande au système et d'une réponse. Les mêmes vérifications sont effectuées, en plus de la vérification que le joueur soit bien mort en lieu et place de la vérification que le joueur est vivant.

Le fonctionnement est encore le même pour le rôle de loup Alpha.

1.4 Diagramme de classe d'analyse

Le diagramme suivant illustre le résultat de notre analyse. La partie supérieure du diagramme montre comment les données seront stockées dans une base de données. Les informations stockées sont les plus importantes pour gérer le déroulement du jeu. Cependant, certaines informations ne seront pas stockées, comme les booléens indiquant si un utilisateur a voté ou non, ou s'il a utilisé son pouvoir…

Nous avons choisi d'utiliser le patron de conception "State" pour représenter les quatre états possibles du jeu : "EnAttente" (le jeu n'a pas encore commencé, mais a été créé), "Jour" (le jeu a commencé et nous sommes en train de jouer), "Nuit" et enfin "FinJeu" (l'état final du jeu).

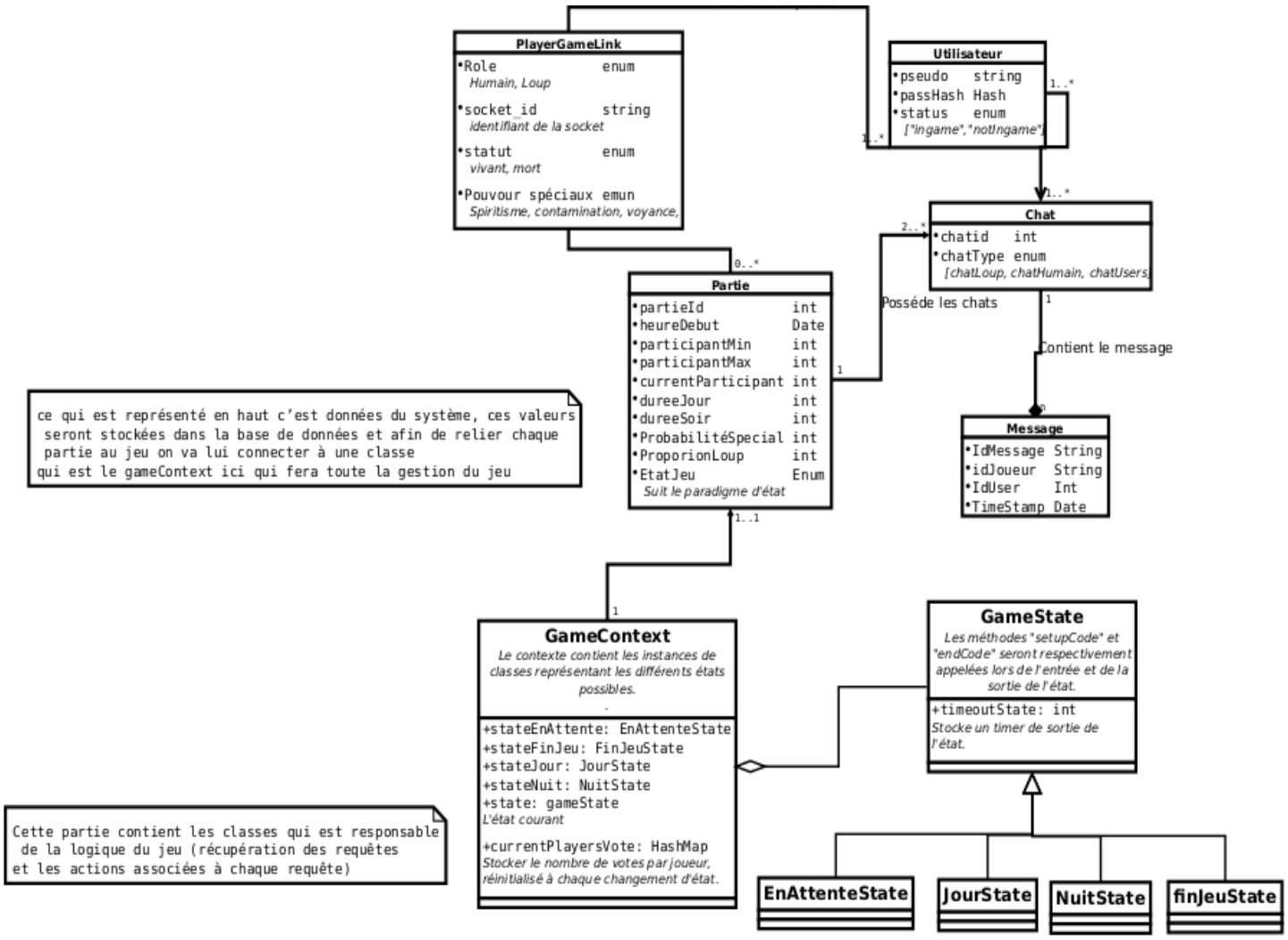


Figure 1.10 : Diagramme de classe d'analyse

L'idée de cette conception est que le serveur recevra des messages de socket, qu'il transmettra ensuite au "GameContext". Ce dernier appellera la méthode de

"CurrentState" qui traitera de cet événement. l'état actuel du jeu sera représenté par “CurrentState”.

2. Conception

2.1 Architecture MVC

Dans ce projet, nous avons été contraints de suivre une architecture MVC. L'application mobile que nous avons développée comportait un front-end et un back-end. Notre objectif était de suivre ce modèle. Ainsi nous avons mis en place une base de données (qui représente notre modèle), nos vues qui sont les pages de front-end et enfin notre API qui représente les différents contrôleurs mis en place.

Dans notre cas, le front-end est une application React Native, tandis que le back-end est un serveur Node.js que nous avons utilisé pour créer cette application. Comme décrit tantôt le front-end constitue la vue du modèle mvc car elle contient toutes les pages de l'application qui sont affichées à l'utilisateur.

Lorsque l'utilisateur souhaite effectuer une action qui nécessite une intervention du serveur, nous envoyons une requête au serveur, qui consulte ensuite la base de données via le modèle, grâce à l'outil de communication Mongoose.

Si la base de données vérifie que les données sont valides et que le contrôleur vérifie une logique spécifique, il enverra une réponse à la vue, qui mettra à jour l'affichage en fonction des données reçues du serveur. Vous

pouvez voir un diagramme illustrant le modèle mvc réalisée dans la Figure 2.

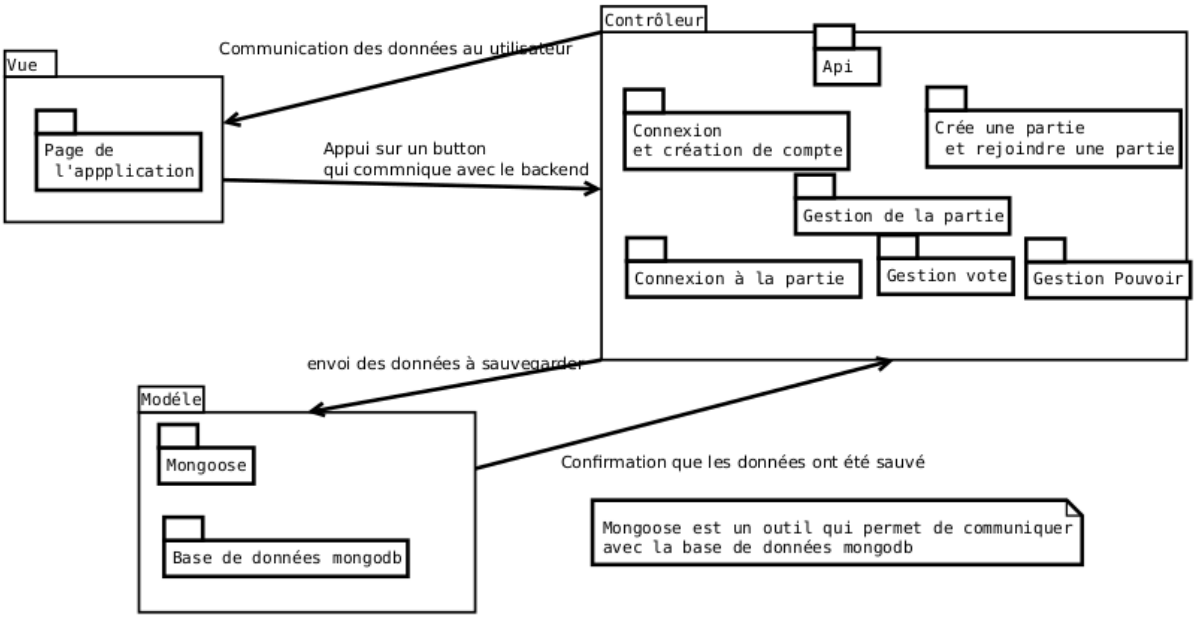


Figure 2.1 : Architecture MVC

2.2 Conception détaillées

2.2.1. Diagramme de classe logiciel

Le diagramme de classes logiciel (Figure 2.2) est très similaire au diagramme de classe d'analyse. Nous avons simplement ajouté les différentes méthodes des classes. Dans la classe "GameContext", nous avons ajouté les différentes méthodes qui feront le traitement des différents événements envoyés par la socket, comme rejoindre une partie, voter ou envoyer un message. Ces méthodes appelleront les gestionnaires de l'état courant. Par exemple, si l'état courant est "Nuit" et qu'on envoie un message, nous appellerons la méthode qui traite les messages de la nuit.

Nous avons également ajouté deux méthodes "setupCode" et "endCode", qui seront appelées respectivement à l'entrée de l'état et lorsqu'on quitte l'état.

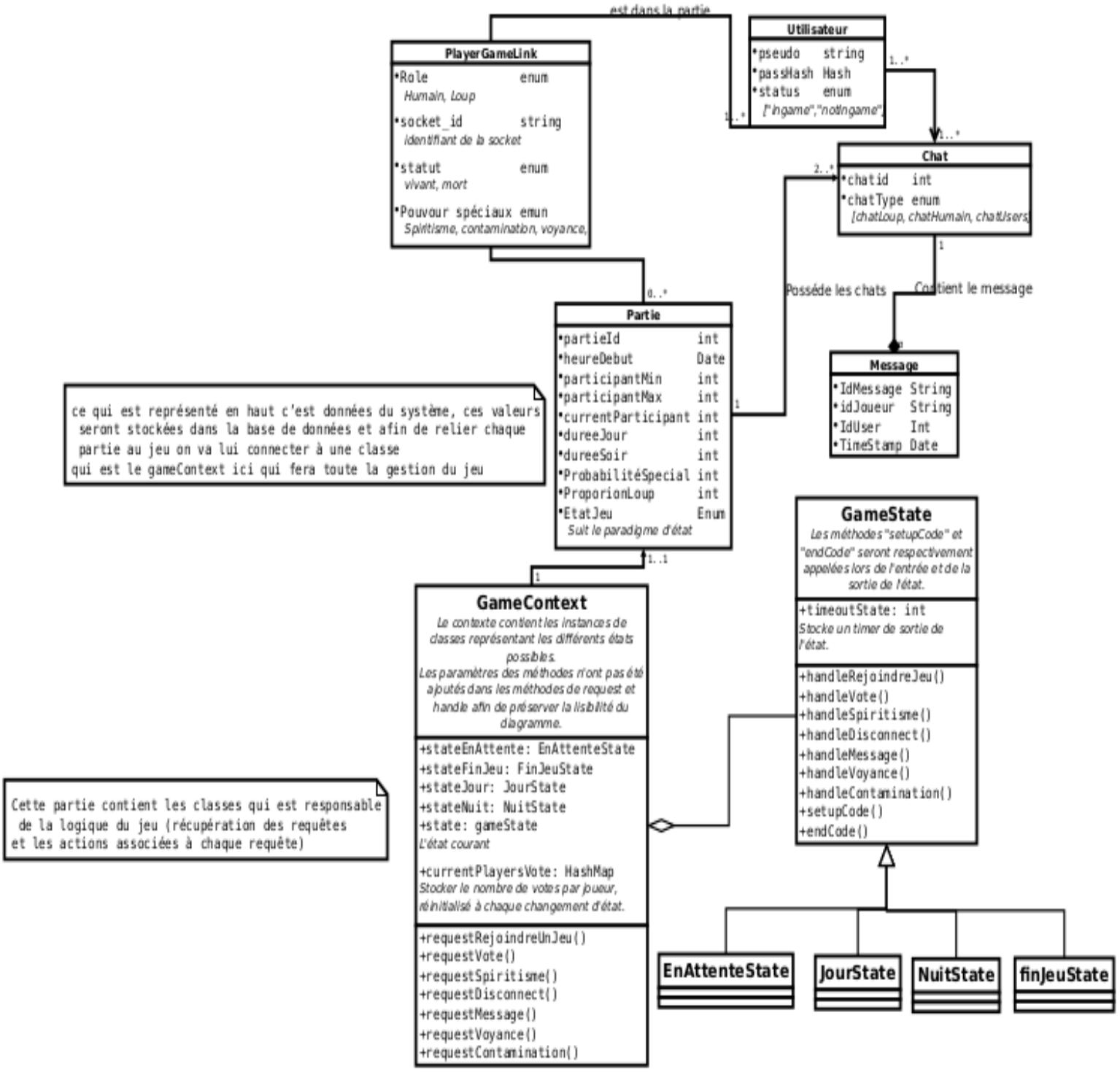


Figure 2.2 Diagramme de classes logicielles

2.2.2. Diagrammes de séquence détaillés

Le diagramme de séquence suivant illustre les différentes transitions d'états. Initialement, nous sommes dans l'état d'attente, en attendant que le minuteur arrive à sa fin. Ensuite, nous informons le système, qui vérifiera la base de données pour déterminer le nombre de joueurs connectés et savoir s'il est possible de commencer le jeu ou non.

Le système entre alors dans une boucle de nuit/jour jusqu'à ce qu'il ne soit plus possible de continuer. Lorsque le jeu ne peut plus continuer, ou si le jeu ne peut pas commencer, nous passons à l'état "Fin de jeu" et nous lançons également un minuteur qui arrêtera le jeu après une période de temps spécifique.

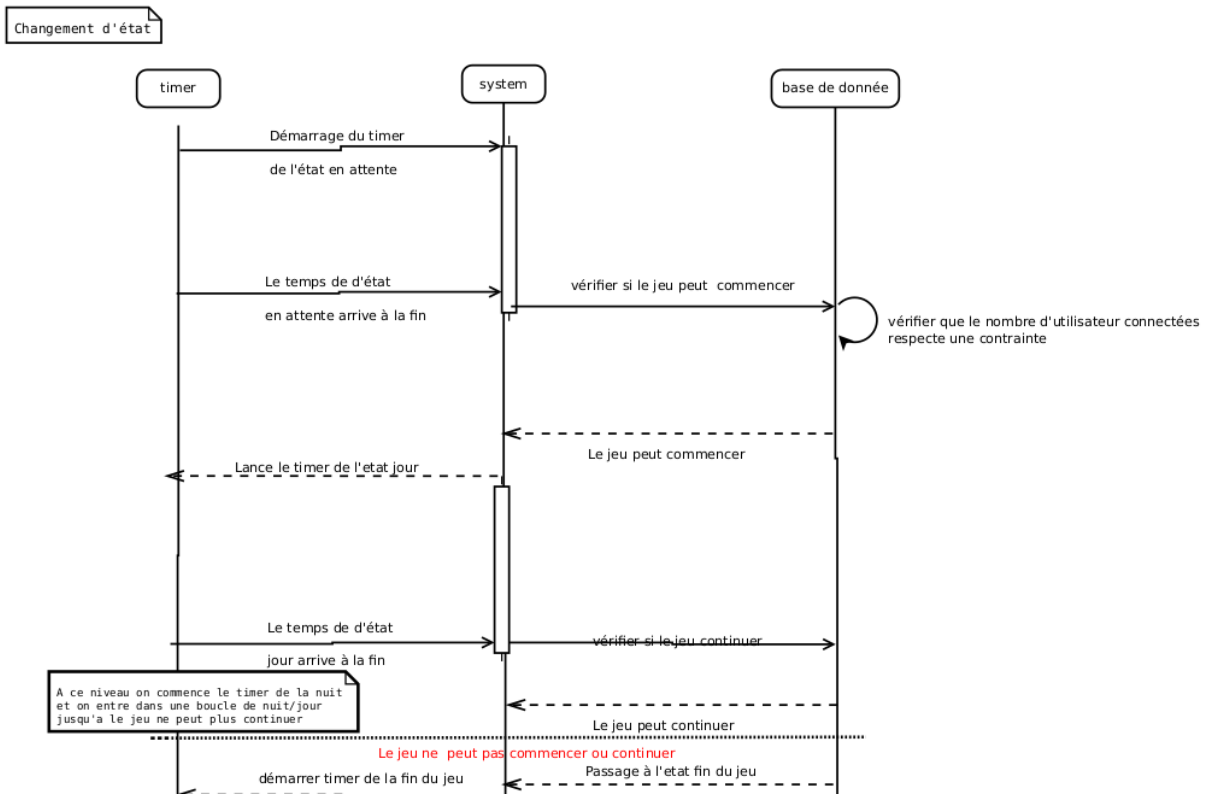


Figure 2.3 : Diagramme de séquence changements d’états

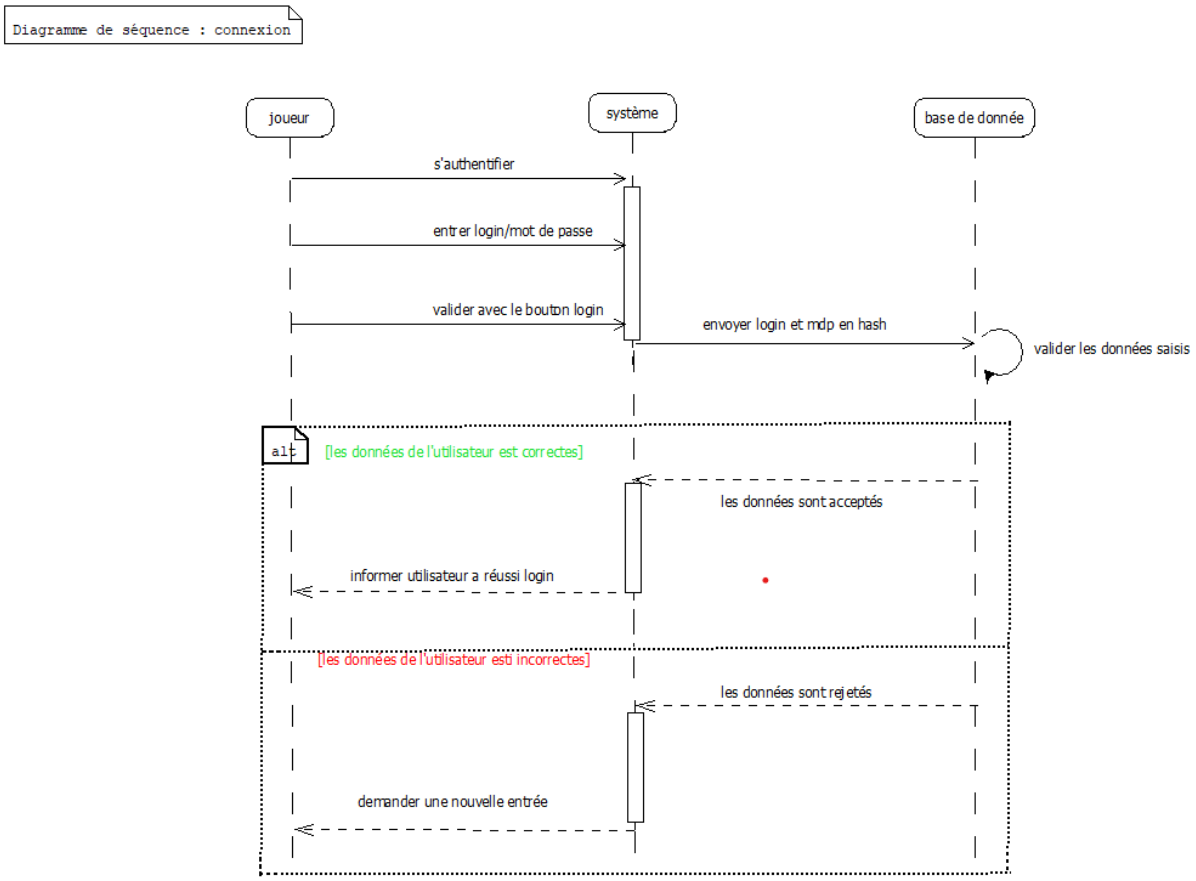


Figure 2.4 : Diagramme de séquence de connexion

Pour la création d'une partie (figure 2.5), l'utilisateur choisit l'option correspondante dans son menu. Le système lui répond alors avec un formulaire lui permettant de régler les différents paramètres demandés pour le jeu (temps de début de partie, proportion de loup, proportion de pouvoir spéciaux…), L'utilisateur renseigne et valide ces informations qui sont ensuite vérifiées par le système et celui-ci ensuite ajoute la nouvelle partie à la base de données. L'utilisateur hôte est automatiquement ajouté à la partie par le système, qui ensuite lui envoie un popup de confirmation de création de la partie avec l'id de celle-ci.

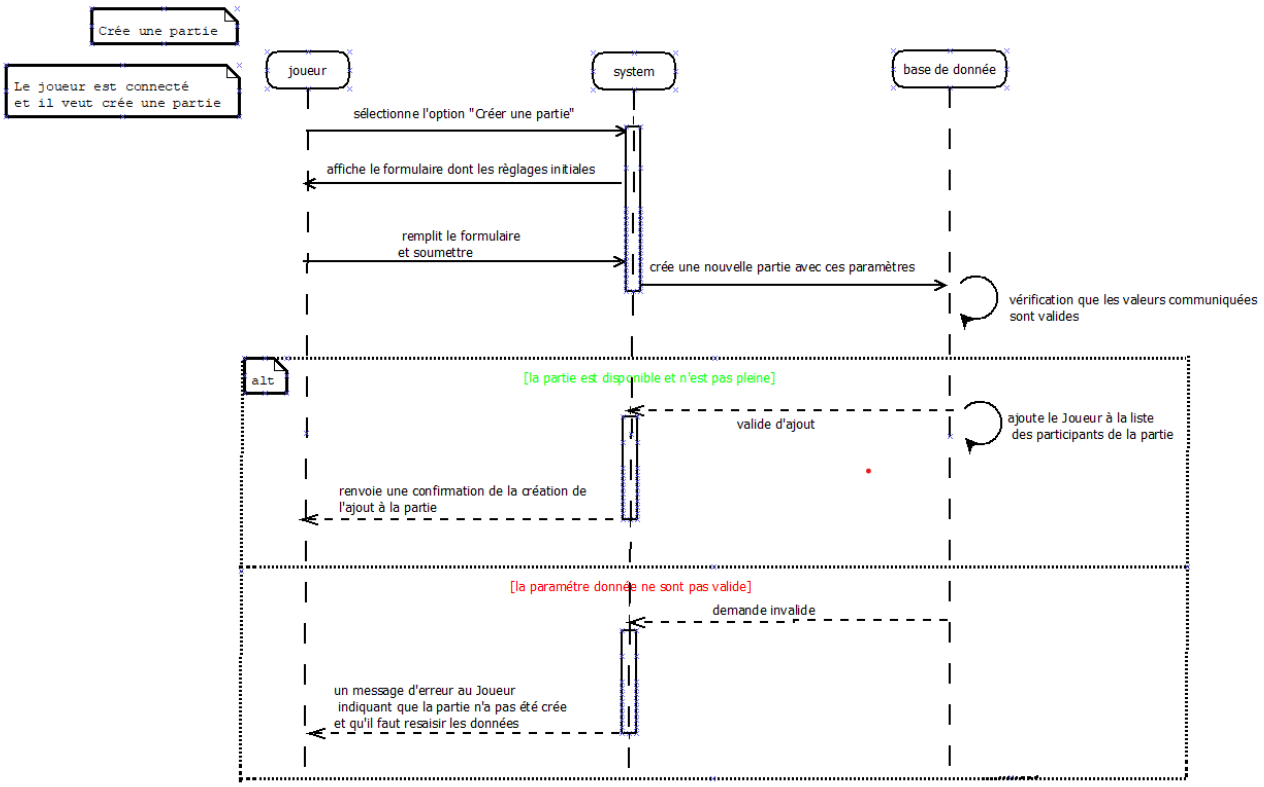


Figure 2.5 : Diagramme de séquence de création d'une partie

2.2.3 Diagramme d'état-transitions

Pour une meilleure compréhension on décide d'expliciter le diagramme d'état-transitions concernant la mise en place d'une partie. Le diagramme de séquence résume les différents états par lesquels doit passer l'utilisateur pour participer au jeu et comment la partie dans l'ensemble se déroule jusqu'à la fin.

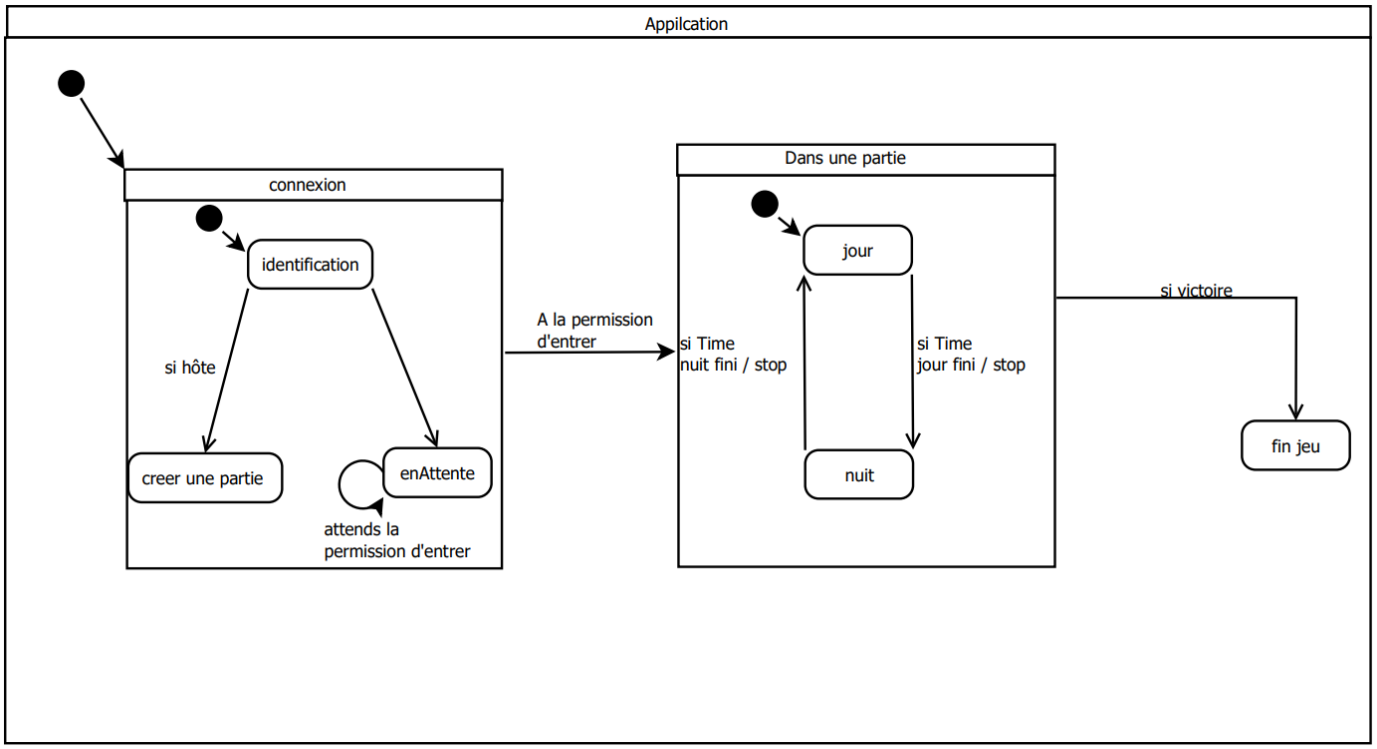


Figure 2.6 : Diagramme d’état/transitions pour une partie

3. Manuel d’utilisateur

3.1 Introduction

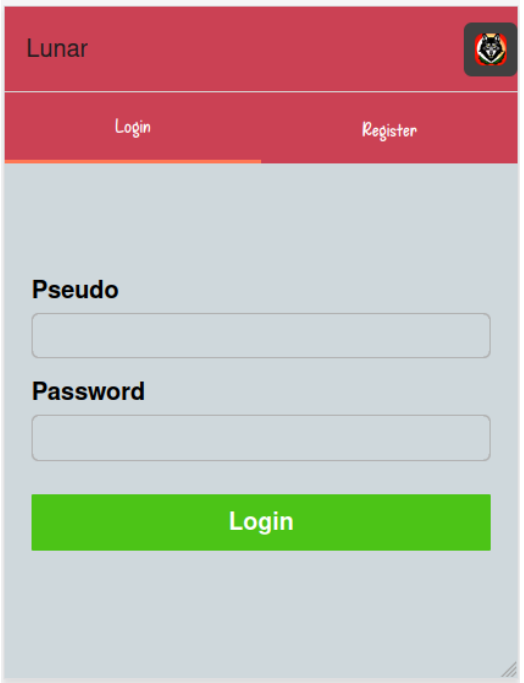
Le manuel d'utilisation comportera des captures d'écran décrivant chaque partie de l'application, accompagnées d'une explication détaillée de chaque élément de l'interface.

3.2 Connexion et création de compte

Dans la page illustrée dans la figure 3.1, nous avons une interface qui nous permet d'ouvrir deux sous-fenêtres. La première fenêtre, qui est ouverte par défaut, contient en haut à gauche le nom du jeu, et en haut à droite un bouton représentant un loup. En cliquant sur cette image, la page actuelle sera rafraîchie.

On peut également remarquer la présence de deux champs de saisie permettant d'entrer un nom d'utilisateur et un mot de passe. Lorsque les données sont entrées, il suffit d'appuyer sur le bouton "login" pour accéder au jeu.

Si les informations saisies sont correctes, nous pourrons accéder au jeu, sinon un message d'erreur s'affiche pour signaler le problème. Cependant, ce message d'erreur ne sera pas visible sur le navigateur.



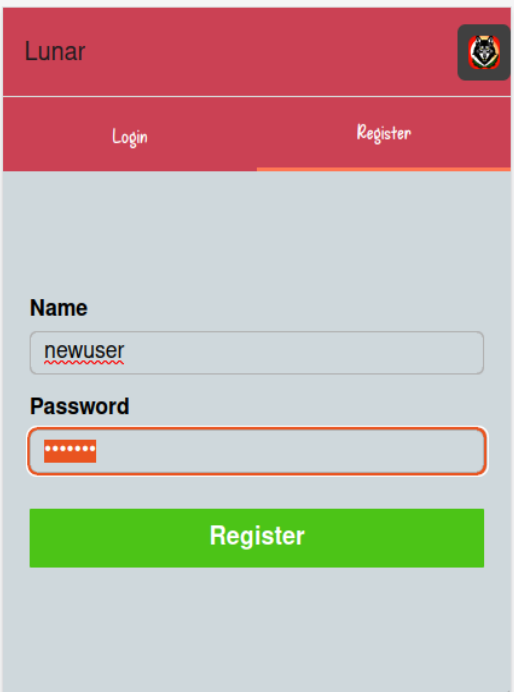


Figure 3.1. Page connexion **Figure 3.2. Page création de compte**

La figure 3.2 présente la page de création de compte, sur laquelle l'utilisateur peut saisir les détails du nouveau compte qu'il souhaite créer. Une fois les informations renseignées, il suffit d'appuyer sur le bouton "register".

3.3 Créer et rejoindre une partie

Une fois que l'utilisateur se connecte avec succès, il est automatiquement redirigé vers une nouvelle page comme illustré dans la figure 3.3. Le nom de l'application est affiché en haut à gauche, accompagné d'un bouton de rafraîchissement de page en haut à droite. De plus, un bouton de déconnexion est disponible pour retourner à l'écran d'accueil. En dessous, deux boutons permettent à l'utilisateur de rejoindre ou de créer un nouveau jeu en toute simplicité.

Figure 3.3 Page d'entrée du jeu

3.3.1 Création de la partie

Si l'utilisateur souhaite créer une partie, il sera redirigé vers une nouvelle page, illustrée dans la figure 3.4, où plusieurs paramètres du jeu peuvent être modifiés.

Ces paramètres incluent le nombre de joueurs souhaité, la durée du jour et de la nuit, la date de début, la proportion des loups dans le jeu et la probabilité des pouvoirs spéciaux.

Figure 3.4 Page configuration du jeu

Lorsque l'utilisateur souhaite modifier un paramètre, un menu contenant une zone de saisie sera affiché comme illustré dans la figure 3.5. Si l'utilisateur entre une valeur non valide, un message d'erreur s'affiche et la valeur ne sera pas enregistrée, comme montré dans la figure 3.6.

Les restrictions sur les valeurs sont les suivantes : le nombre de joueurs souhaité doit être compris entre 3 et 20, la durée de la nuit et du jour doivent être comprises entre 1 et 23, leur somme doit être égale à 24, la proportion et la probabilité des loup-garous doivent être comprises entre 0 et 1. Enfin, la date de début doit être ultérieure à la date actuelle. Si l'utilisateur reste sur

cette page après la date de début, la partie ne sera pas créée tant que la date n'est pas modifiée.

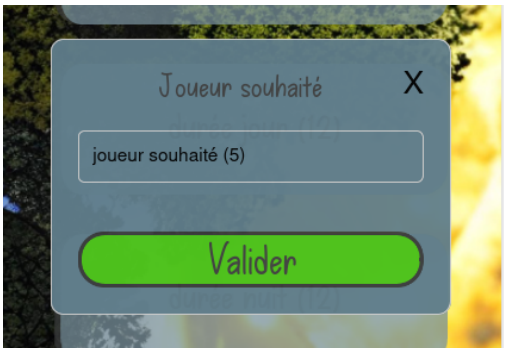


Figure 3.5. Modification paramètre **Figure 3.6. Erreur modification**

Enfin, pour créer le jeu, il suffit d'appuyer sur le bouton situé en bas de la page, comme illustré dans la figure 3.7.

Figure 3.7 Button création du jeu

3.3.2 Rejoindre une partie

Pour rejoindre une partie, l'utilisateur doit cliquer sur le bouton "Rejoindre un jeu" présent sur la page d'accueil, comme illustré dans la figure 3.3. Ensuite, un menu s'affiche comme montré dans la figure 3.8 où l'utilisateur devra entrer l'identifiant de la partie fournie par le créateur du jeu.

Après avoir entré l'identifiant, l'utilisateur devra appuyer sur le bouton "Rejoindre le jeu" du menu. Si l'identifiant est valide et qu'il reste de la place dans la partie, l'utilisateur sera redirigé vers la page du jeu. Sinon, un message d'erreur sera affiché en rouge sous l'inputbox, comme montré dans la figure 3.9.

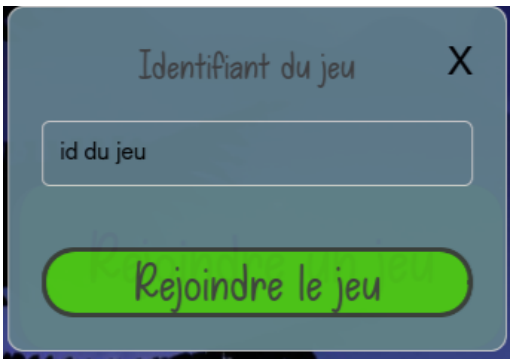


Figure 3.8. Menu rejoindre jeu **Figure 3.9. Erreur lors de la connexion au jeu**

3.4 Partie

3.4.1 Salle d'attente

Une fois que l'utilisateur a rejoint une partie, il sera redirigé vers une salle d'accueil comme illustrée dans la figure 3.10. Cette page affichera des informations sur l'état du jeu. En haut à gauche on aura accès à l'état du jeu qui va varier entre ce qui est affiché actuellement et Jour, Nuit et Fin du jeu.

En haut à droite, un bouton sera présent pour permettre à l'utilisateur de quitter la partie. Lorsqu'il appuie sur ce bouton, il sera dirigé vers le menu illustré dans la figure 3.11, où il pourra confirmer ou annuler son choix.

Au centre de la page, des informations sur le jeu seront affichées. L'identifiant du jeu, qui sera partagé avec d'autres joueurs pour qu'ils puissent rejoindre, sera affiché en premier. Ensuite, le dernier message du serveur indiquant la dernière mise à jour, le nombre de joueurs connectés et le nombre de joueurs souhaités seront affichés. Enfin, le temps de début de jeu en secondes sera également affiché. Si le nombre de joueurs souhaité est atteint ou si le compteur arrive à sa fin, le jeu commencera.

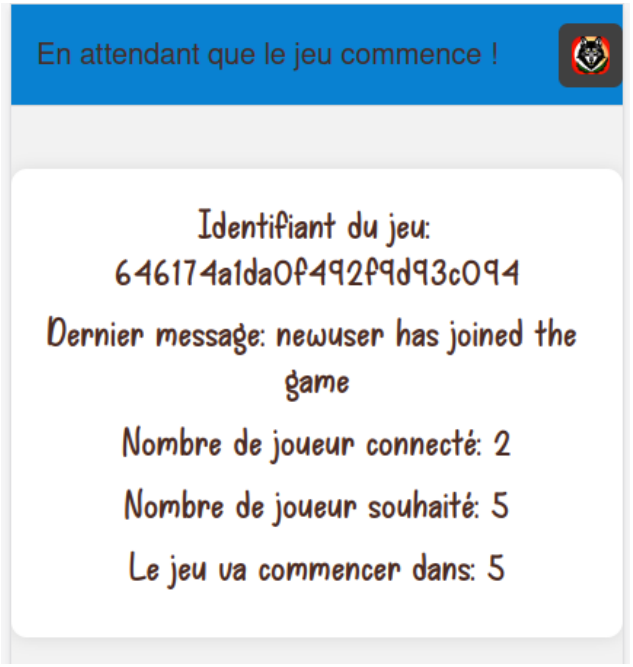


Figure 3.10 Salle d'attente

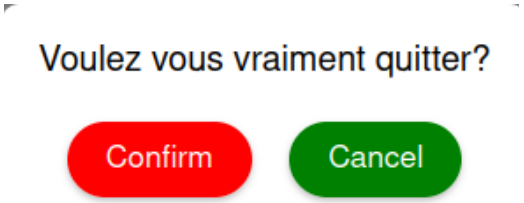


Figure 3.11 Menu confirmation

3.4.2 Jour

Une fois le jeu commencé, on sera redirigé vers la page du jour qui est illustrée dans la figure 3.12. En haut à gauche, on trouve l'indicateur de l'état actuel qui est "jour". Ensuite, il y a deux fenêtres : une pour le jeu et une pour le chat. Dans la fenêtre du jeu, on peut voir les informations du joueur, telles que son pseudo et son rôle qui est "loup" dans ce cas. On peut également voir qu'il possède un pouvoir spécial, qui est "voyance" mais qui n'est pas très visible sur l'image.

Après cela, on peut voir les actions qu'il peut faire pendant le jour, qui ne peuvent être que de voter, car on est dans la journée.

Enfin, on peut voir une liste des joueurs vivants avec leur pseudo et le nombre de votes qui ont été enregistrés pour eux durant le vote en cours. Cette valeur change dynamiquement lorsqu'un vote est réalisé pour une personne.

Si on appuie sur le bouton "Vote", un menu s'affiche comme illustré dans la figure 3.13. Ce menu contient la liste des joueurs pour lesquels on peut voter.

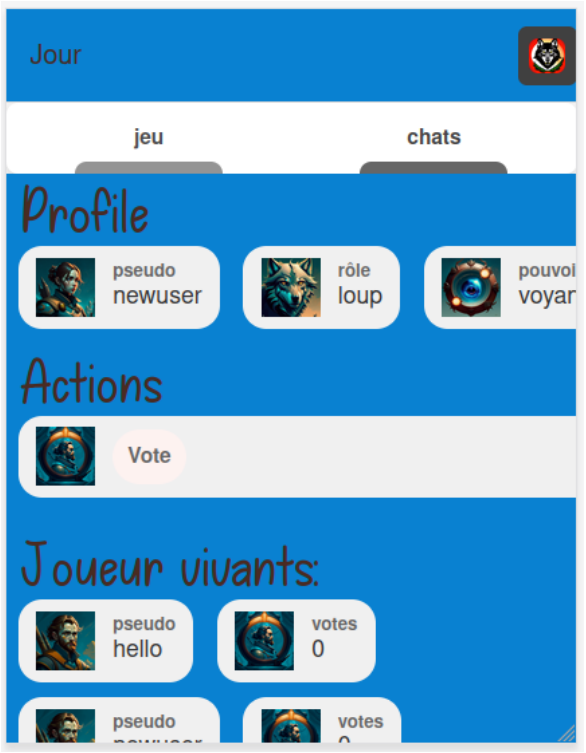


Figure 3.12 Page jour action vote

Figure 3.13 Liste joueur

Quand on appuie sur la fenêtre de chat, on peut voir qu'il existe un chat appelé place de village pendant la journée, comme illustré dans la figure 3.14. Dans cette page, nous pouvons voir les messages envoyés sur ce fil durant la journée et écrire sur ce chat uniquement pendant la journée.

Les messages envoyés par l'utilisateur sur ce jeu sont affichés en vert sur la droite, tandis que ceux envoyés par les autres joueurs sont affichés en blanc sur la gauche. Chaque message contient également le nom de la personne qui l'a envoyé. Dans le bas de la page, il y a un champ de saisie pour écrire le message et un bouton "envoyer" pour l'envoyer.

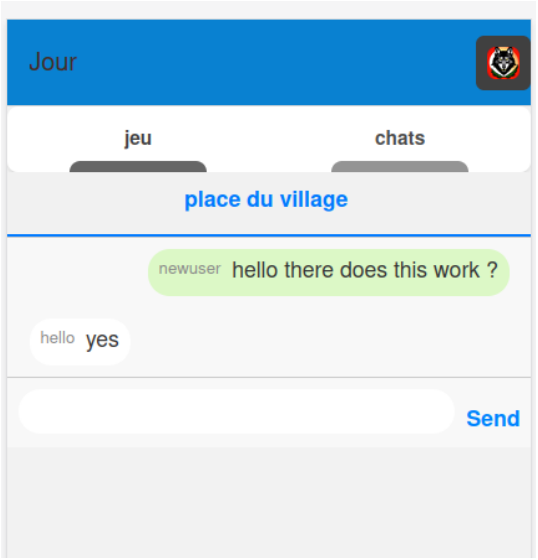


Figure 3.14 Fenêtré chats

Quand l'état de jour et de nuit arrive à la fin, on montrera un message indiquant ce qui c'est passé qui ne peut être que deux choses: un joueur a été tué ou aucun joueur n'a été tué. La figure 3.15 montre un exemple de cela

3.4.3 Nuit

Pendant la nuit, l'affichage est similaire à celui du jour, mais avec des informations spécifiques à chaque rôle. Les loups verront la liste des loups vivants ainsi que celle des humains vivants avec l'indicateur de vote comme illustré dans la figure 3.15. Seuls les loups peuvent voter et ils ne peuvent voter que pour des humains.

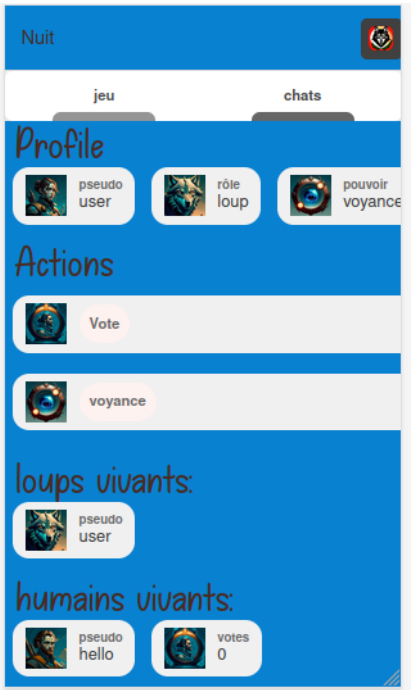


Figure 3.15 Page nuit pour un loup

Les chats vont dépendre du rôle, pour les humains ils vont voir le chat de place de village mais aucun message ne peut être envoyé dans ce chat pendant la nuit. Les loups-garous, pendant la nuit, auront un chat uniquement à eux. La figure 3.16 montre un exemple de la sous fenêtre chat pour un loup.

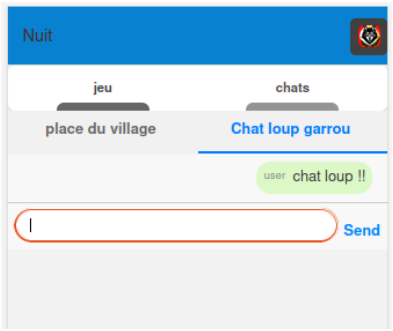


Figure 3.16 Fenêtre chat loup garou dans la nuit

Pendant la nuit, les pouvoirs spéciaux sont également utilisés. Le pouvoir voyance est accessible via un bouton similaire à celui du vote, comme illustré dans la figure 3.15. Lorsque ce bouton est cliqué, un menu similaire à celui de la figure 3.13 apparaît, permettant de sélectionner le joueur sur lequel le pouvoir sera utilisé. Une fois le pouvoir utilisé, une pop-up apparaît pour afficher les informations sur le rôle et le pouvoir de la personne sélectionnée, comme illustré dans la figure 3.17.

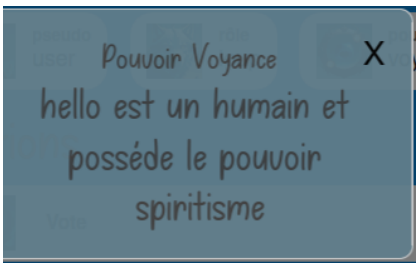


Figure 3.17 Pouvoir voyance

Le pouvoir de **spiritisme** est accessible via un bouton, tout comme le pouvoir de voyance. Il nous permet de créer une salle de discussion avec un joueur mort. Lorsque nous cliquons sur ce bouton, un menu similaire à celui de la figure 3.13 apparaît, affichant la liste des joueurs morts. Une fois qu'un joueur est sélectionné, un chat est créé, qui sera accessible uniquement par la personne ayant utilisé le pouvoir et le joueur mort. Ce chat sera ajouté comme un nouveau chat dans la fenêtre de discussion. Cette fonctionnalité est illustrée dans la figure 3.18.

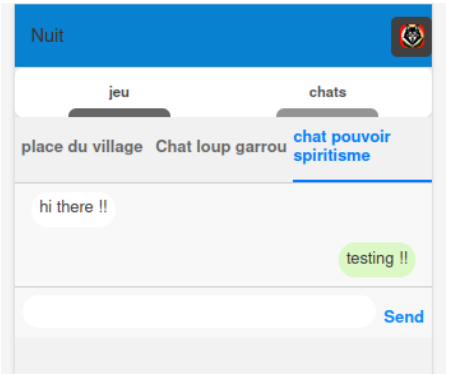


Figure 3.18 Chat room créé avec le pouvoir spiritisme.

Le pouvoir insomnie est donné à un seul humain dans le jeu cela lui permet de lire les chats des loups garou mais il ne pourra pas voir qui a envoyé les messages mais uniquement le contenu des messages. Cette fonctionnalité est illustrée dans la figure 3.19(image d'un loup qui envoie un message) et 3.20(l'utilisateur avec le pouvoir insomnie qui peut lire contenu de message mais ne peut pas voir qui a envoyé le message).

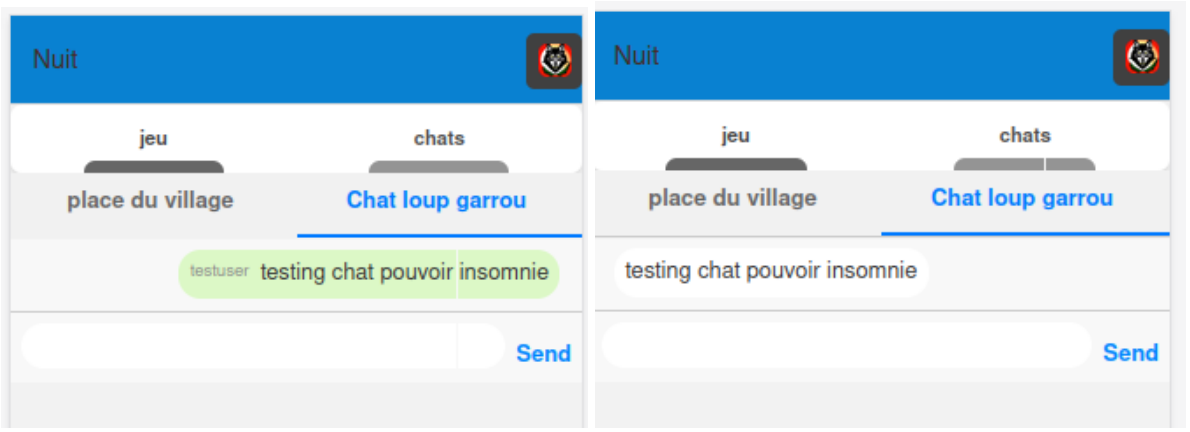


Figure 3.19 Message envoyé par un loup & Figure 3.20 Reception message avec le pouvoir insomnie.

Le pouvoir de contamination, accessible aux loups-garous, est également représenté par un bouton. Lorsqu'on l'appuie, une liste des humains encore vivants s'affiche. En sélectionnant un humain dans cette liste, celui-ci sera converti en loup-garou et fera partie de l'équipe des loups-garous dès le début du jour suivant.

3.4.4 Joueur mort

Lorsqu'un joueur meurt, son interface utilisateur sera modifiée et affichera ce qui est illustré dans la figure 3.21. Il aura la possibilité de voir les joueurs qui sont encore en vie et ceux qui sont morts, mais il ne pourra plus interagir avec eux via le chat ni voter.

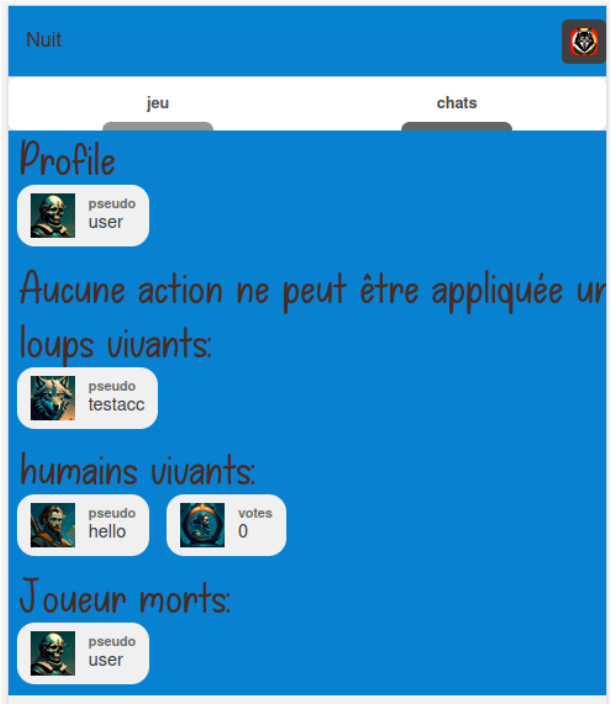
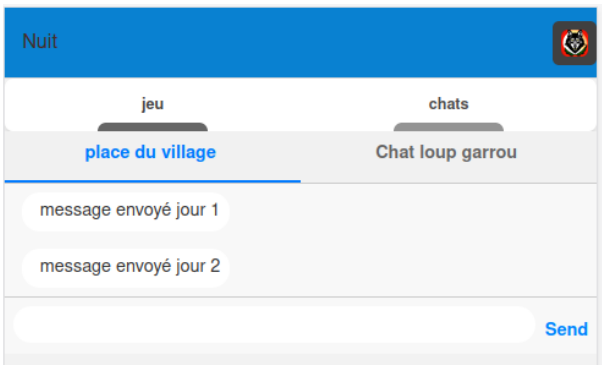


Figure 3.21 Page jeu pour un joueur mort

Quand un joueur est mort, il ne peut plus envoyer de messages dans les chats, mais il peut toujours rejoindre les deux chats et voir tous les messages précédents ainsi que les nouveaux messages qui y sont postés. Les figures 3.22 et 3.23 montrent l'historique complet des messages affichés pour le joueur décédé.



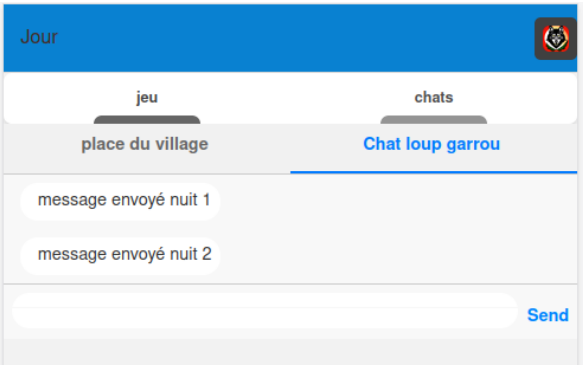


Figure 3.22 Archive chat jour

Figure 3.23 Archive chat nuit

3.4.5 Fin du jeu

Finalement, lorsque le jeu se termine, un pop-up apparaîtra pour indiquer le résultat final du jeu, comme illustré dans la figure 3.24. Chaque joueur pourra également accéder à toutes les archives des messages de la partie, comme présenté dans les figures 3.22 et 3.23.

Figure 3.24 Résultat du jeu

Le jeu se termine lorsqu'il n'y a plus de loups ou d'humains.

4. Bilan du projet

Le sujet du projet était très intéressant et nous avons pu acquérir de nouvelles connaissances, notamment sur React, React Native, les sockets et la gestion des événements d'un jeu. Bien que ces sujets ne soient pas difficiles en soi, il aurait été préférable de les aborder davantage dans le cours afin de réduire considérablement la quantité de travail nécessaire. Certains points prenaient pas mal de temps pour les manipuler correctement, en particulier l'utilisation des hooks de React comme useEffect, surtout lorsque des données sont modifiées via une API liée à cet effet. Les sockets n'étaient pas très intuitifs aussi dans des cas complexes.

En termes de conception, nous avons utilisé un patron de conception qui s'est avéré très utile et applicable dans un cas réel. Nous avons beaucoup apprécié l'architecture organisée et propre que nous avons mise en place.

En ce qui concerne la gestion d'équipe, le projet n'a pas été un succès complet . La répartition des tâches ainsi que le suivi des objectifs se faisait via les issues gitlab cela nous a permis de garder nos objectifs en vue , de savoir comment découper le travail. La répartition des tâches n 'était pas optimale car certains membres ayant travaillé plus que d'autres. Cependant, cela a été une expérience enrichissante d'où nous pouvons apprendre pour l'avenir.