



PROJET WEB/ACOL

Jeu du loup garou

Rapport

- *Auteurs* -

Naima AMALOU
Nabil BENSRHIER
Hamza BENJELLOUN
Nadir AIT LAHMOUCH

30 avril 2020

Table des matières

1	Introduction	3
2	Document d'analyse	4
2.1	Diagramme de cas d'utilisations :	4
2.2	Description des cas d'utilisations :	6
2.2.1	1er cas d'utilisation :	6
2.2.2	2ème cas d'utilisation :	6
2.2.3	3ème cas d'utilisation :	7
2.2.4	4ème cas d'utilisation :	8
2.2.5	5ème cas d'utilisation :	8
2.2.6	6ème cas d'utilisation :	8
2.2.7	7ème cas d'utilisation :	9
2.2.8	8ème cas d'utilisation :	9
2.3	Diagrammes de séquences :	10
2.3.1	1er diagramme : Inscrit/Authentification	10
2.3.2	2ème diagramme : Configuration et lancement d'une partie	11
2.3.3	3ème diagramme : Discussion/Chat	13
2.3.4	4ème diagramme : Vote	14
2.4	Diagramme de classes :	15
2.5	Diagramme d'objet :	16
3	Document de conception	17
3.1	Architecture logique du système :	17
3.2	Description détaillée de l'incrément choisi	18
3.2.1	Prérequis :	18
3.2.2	S'inscrire	18
3.2.3	Se connecter et se déconnecter de l'application :	18
3.2.4	Compte personnel :	18
3.2.5	Jouer dans une partie :	18
3.2.6	Jeu	19
3.3	Conception détaillée de l'incrément choisi	21
3.3.1	Diagramme de classes logicielles	21
3.3.2	Conception de la base de donnée	24
3.3.3	Diagrammes de séquence	25
3.4	Diagrammes d'états-transition :	30
3.5	Validation :	31
3.5.1	Peuplement de la base de données :	31
3.5.2	Description détaillée de la base de données :	31

4 Manuel utilisateur	33
4.1 Prérequis :	33
4.2 Inscription/Connexion :	34
4.2.1 Connexion :	34
4.2.2 Inscription :	35
4.3 Page principale :	36
4.4 Configuration d'une nouvelle partie :	37
4.5 Jeu :	38
4.5.1 Maître du jeu :	38
4.5.2 Joueur :	39
4.6 Pouvoirs :	42
4.6.1 Voyance chez les humains :	42
4.6.2 Contamination chez les loups garous :	43
4.7 Vote :	44
4.8 Fin de partie :	45
4.9 Déconnexion :	45
5 Bilan sur les outils de modélisation	46

Chapitre 1

Introduction

Le jeu du loup garou est un jeu de société très populaire trouvant ses origines dans un ancien jeu soviétique des années 80. Il est sorti en 2001 et rencontre depuis un ample succès.

L'application Web que nous proposons reprend les règles de ce jeu célèbre et sa manière d'être joué. Nous automatisons la plupart des procédures mais gardons quand même un GM pour plus d'authenticité.

Ce rapport a pour vocation de suivre les étapes de création de cette application ainsi qu'une évaluation des outils de modélisation utilisés dans ce projet. Aussi, nous établirons un bilan général sur les outils que nous avons utilisé dans le processus de création et d'implémentation.

Chapitre 2

Document d'analyse

Commençons ce document par la première étape nécessaire pour le bon fonctionnement de chaque projet : **L'Analyse**.

En effet, cette étape est primordiale pour une meilleure compréhension des besoins ainsi qu'établir une première version structurée de notre système. Et pour ce faire, nous userons de différents diagrammes.

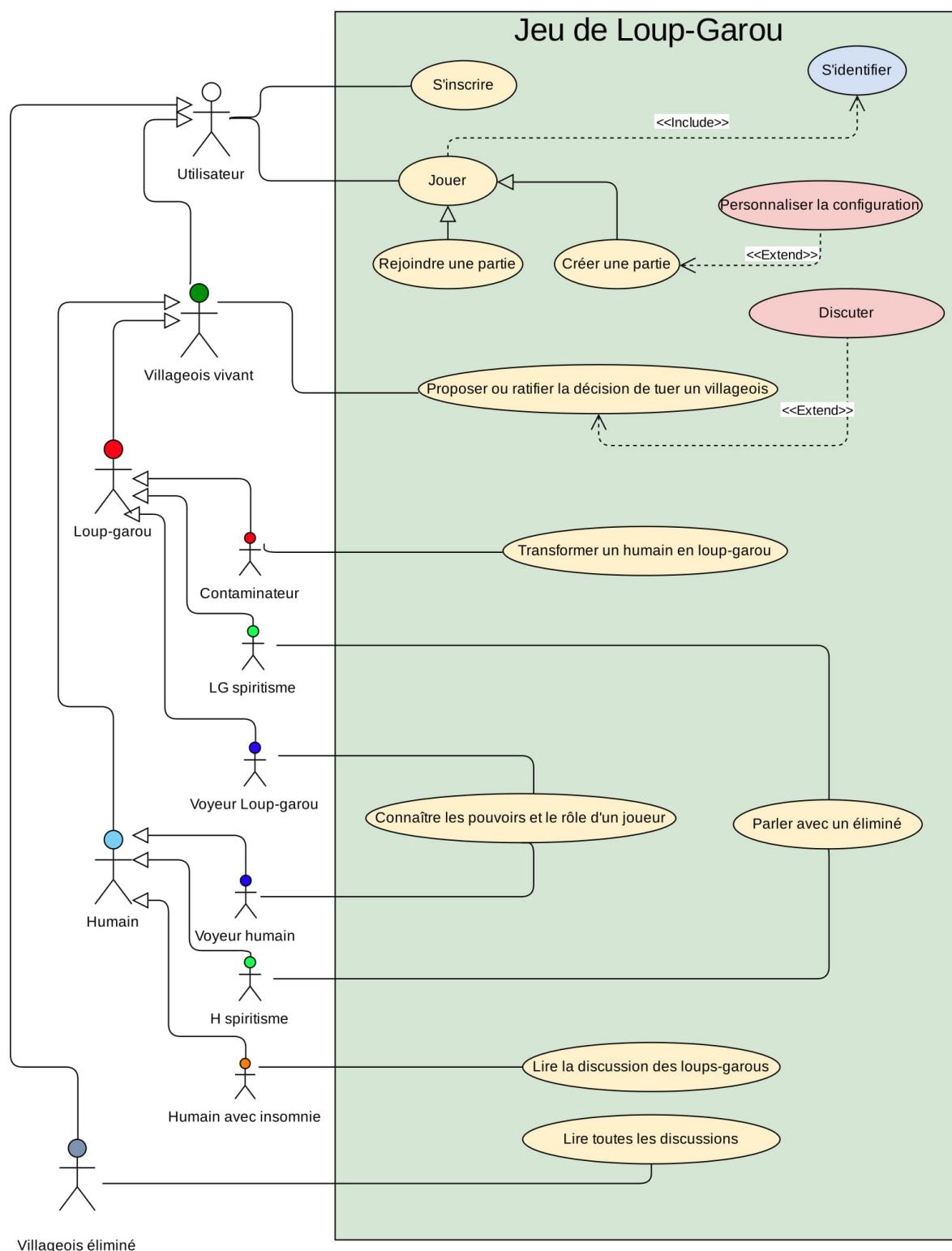
2.1 Diagramme de cas d'utilisations :

Un diagramme de cas d'utilisation dans sa forme la plus simple est une représentation de l'interaction d'un utilisateur avec le système qui montre la relation entre l'utilisateur et les différents cas d'utilisation dans lesquels il est impliqué. Cela nous donnera une meilleure visualisation du projet.

Le diagramme ci-dessous présente notre jeu dans sa globalité, il nous permet d'exprimer les besoins utilisateurs par rapport aux principales fonctionnalités du système (inscription, connexion ...), et en particulier les besoins des joueurs dans la phase du jeu (voter, écrire des messages ...)

Voici donc le diagramme en question :

Visual Paradigm Online Diagrams Express Edition



Visual Paradigm Online Diagrams Express Edition

FIGURE 2.1 – Diagramme de cas d'utilisations

2.2 Description des cas d'utilisations :

Voici une description détaillée des différents cas d'utilisations de notre applications. Nous illustrerons une partie de ceux ci par la suite avec des diagrammes de séquences adéquats.

2.2.1 1er cas d'utilisation :

- **Cas** : S'inscrire
- **Acteur** : Utilisateur

Le cas d'utilisation commence si l'utilisateur clique sur le bouton **S'inscrire**.

Scénario principal :

1. Le système demande à l'utilisateur de saisir :
 - Une adresse mail.
 - Un pseudonyme.
 - Un mot de passe.
2. L'utilisateur saisit ses données et valide.
3. Le système informe l'utilisateur que l'inscription s'est effectuée et un e-mail de confirmation est envoyé à l'utilisateur.

Scénario d'erreur : Pseudonyme existe déjà.

Le système affiche un message d'erreur et propose à l'utilisateur de choisir un autre pseudonyme. Retour à l'étape 1.

2.2.2 2ème cas d'utilisation :

- **Cas** : Jouer
- **Acteur** : Utilisateur

Le cas d'utilisation commence lorsque l'utilisateur clique sur le bouton **Jouer**.

Scénario principal :

1. Le système demande à l'utilisateur de saisir son pseudonyme et son mot de passe.
2. L'utilisateur saisit son pseudonyme et son mot de passe et valide
3. Le système demande à l'utilisateur de choisir une partie parmi la liste des parties disponibles ou d'en créer une nouvelle.
4. L'utilisateur choisit une partie et valide.
5. Le système affiche le rôle de l'utilisateur et son pouvoir, ainsi que la liste des utilisateurs et le nombre des loups-garous

Scénario d'erreur : Utilisateur inconnu.

Utilisateur n'est pas connu dans le système. Le système affiche un message d'erreur et propose à l'utilisateur de choisir entre créer un compte ou revenir à l'étape 1.

Scénario alternatif : Création d'une nouvelle partie.

Le système propose deux types de parties :

- **Création rapide** : la configuration par défaut.
- **Création personnalisée** : une configuration donnée par l'utilisateur.

L'utilisateur choisit alors un type de création et valide ses choix. Le scénario reprend à l'étape 5.

2.2.3 3ème cas d'utilisation :

- **Cas** : Proposer ou ratifier la décision de tuer un villageois
- **Acteur** : Villageois vivant

Ce cas débute en cliquant soir sur le bouton **Proposer** pour proposer un villageois ou le bouton **Ratifier** pour voter pour une proposition.

Pré-condition : Le manque de propositions ou lorsque aucune décision n'a été validée par le système.

Scénario principal :

1. Le système demande au villageois vivant de proposer ou de ratifier une proposition existante.
2. Le vivant propose un villageois.
3. Le système affiche la liste de propositions à tous les utilisateurs avec le nombre de vote de chacune. Retour au Mode Chat.
4. La majorité ratifient telle proposition en discutant ou pas.
5. Le système valide la décision et affiche la victime.

Scénario alternatif : Ratifier une proposition.

- **2.a)** Le vivant choisit de ratifier une proposition.
- **2.b)** Le système affiche les propositions.
- **2.c)** Le villageois vote pour une des proposition. Le scénario reprend à l'étape 3.

Scénario d'erreur : Décision déjà validée.

- **2.a)** Proposition et ratification impossible, car il existe une décision déjà validée. Retour en mode Chat.

2.2.4 4ème cas d'utilisation :

- Cas : Transformer un humain en loup-garou.
- Acteur : Contaminateur.

Ce cas commence lorsque le contaminateur clique sur le bouton **Contaminer**.

Scénario principal :

1. Le système demande au contaminateur de choisir un humain parmi la liste des humains.
2. Le contaminateur choisit un humain.
3. Le système actualise le nombre des loups-garous et ce nouveau loup-garou contaminé rejoint leur repaire.

Scénario d'erreur : Pas de contamination pendant le tour.

- 1.a) Retour au mode Chat.

2.2.5 5ème cas d'utilisation :

- Cas : Connaître les pouvoirs et le rôle d'un joueur.
- Acteur : Voyant.

Ce cas commence lorsque le voyant clique sur le bouton **Observer**.

Scénario principal :

1. Le système demande au voyant de choisir un villageois parmi la liste des villageois.
2. Le voyant en choisit un.
3. Le système affiche son rôle et ses pouvoirs.

Scénario d'erreur : Pas de voyance pendant le tour.

- 1.a) Retour au mode Chat.

2.2.6 6ème cas d'utilisation :

- Cas : Parler avec un villageois déjà éliminé.
- Acteur : Spiritualiste.

Ce cas commence lorsque le spiritualiste clique sur le bouton **Parler avec un mort**.

Scénario principal :

1. Le système demande au spiritualiste de choisir un éliminé parmi la liste des éliminés.
2. le spiritualiste choisit un mort.
3. Le système lance une discussion entre les deux.

Scénario d'erreur : Pas de spiritisme pendant le tour.

- 1.a) Retour à la salle de discussion principale.

Scénario d'erreur : Absence d'éliminés.

- 2.a) Retour à la salle de discussion principale.

2.2.7 7ème cas d'utilisation :

- **Cas** : Lire la discussion des loups-garous.
- **Acteur** : Humain avec insomnie.

Ce cas commence lorsque l'humain clique sur le bouton **Espionner**.

Scénario principal :

1. Le système ajoute l'humain dans la liste des lecteurs de la discussion des loups-garous.

Scénario d'erreur : Pas d'espionnage pendant le tour.

- 1.a) Retour à la salle de discussion principale.

2.2.8 8ème cas d'utilisation :

- **Cas** : Lire toutes les discussions.
- **Acteur** : Villageois éliminé.

Ce cas commence lorsque le villageois est éliminé.

Scénario principal :

1. Le système ajoute le villageois dans la liste des lecteurs de toutes les discussions.

2.3 Diagrammes de séquences :

Un diagramme de séquence montre les interactions d'objets organisées en séquence temporelle. Il décrit les objets et les classes impliqués dans le scénario et la séquence de messages échangés entre les objets nécessaires pour réaliser sa fonctionnalité.

Dans cette section, nous allons illustrer les plus importants cas d'utilisations pour mieux visualiser les cas et erreurs à prendre en considération pendant l'implémentation.

2.3.1 1er diagramme : Inscrit/Authentification

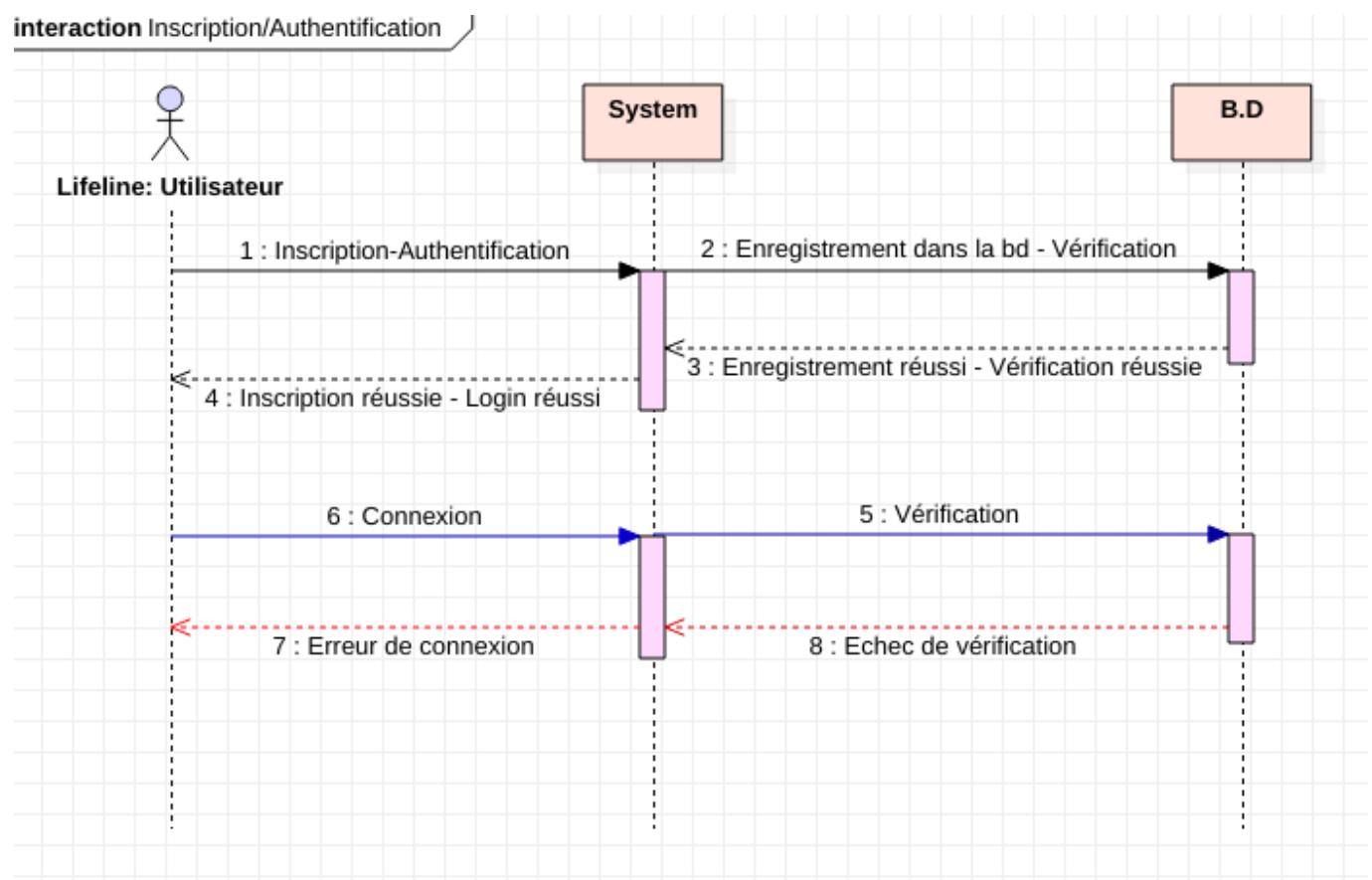


FIGURE 2.2 – Diagramme de séquence : inscription/authentification

Le diagramme représente les séquences présentes dans l'inscription à l'application en ligne, ainsi que la connexion si jamais l'utilisateur a déjà un compte. Si c'est sa première visite, les informations entrées sont enregistrées dans une base de données et serviront à son authentification plus tard. Des messages d'erreur seront envoyés selon si les informations entrées pendant l'inscription sont conformes ou si la vérification pendant la connexion a échoué.

2.3.2 2ème diagramme : Configuration et lancement d'une partie

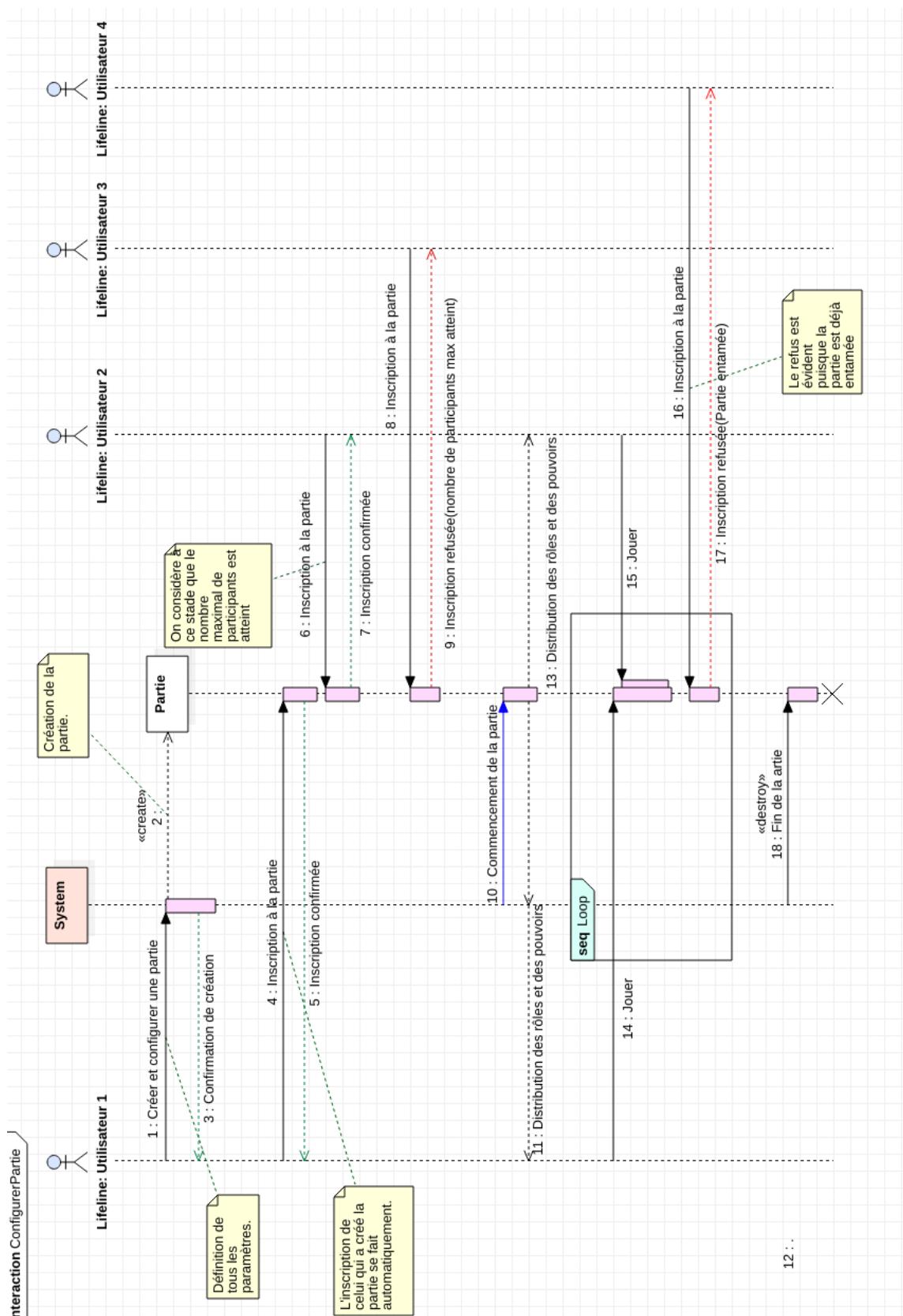


FIGURE 2.3 – Diagramme de s quence : Configuration et lancement d'une partie

La configuration de la partie se fait quand un utilisateur décide d'en créer une nouvelle. Durant cette action, il lui est demandé de rentrer les paramètres de la partie qu'il veut créer. A savoir :

- le nombre de participants souhaité (ou une fourchette), par défaut entre 5 et 20 ;
- la durée du jour et de la nuit (voir ci-dessus les valeurs par défaut) ;
- l'horaire de début, par défaut le lendemain matin 8h00 ;
- la probabilité que chaque pouvoir spécial soit attribué (par défaut 0 : partie sans pouvoirs spéciaux) ;
- la proportion initiale de loups-garous (par défaut 1/3).

Si il respecte toutes les formes demandées dans chaque champ, la partie est créée et bascule dans une autre page où il attendra d'autres joueurs.

Ces joueurs devront s'inscrire pour rejoindre la partie. Cependant, l'inscription est fermée une fois l'horaire de début ou si le nombre maximal de joueurs sont atteints.

Le lancement de la partie se fait automatiquement après la fermeture d'inscription et les joueurs sont redirigés vers la salle de discussion principale, et la partie peut alors commencer !

2.3.3 3ème diagramme : Discussion/Chat

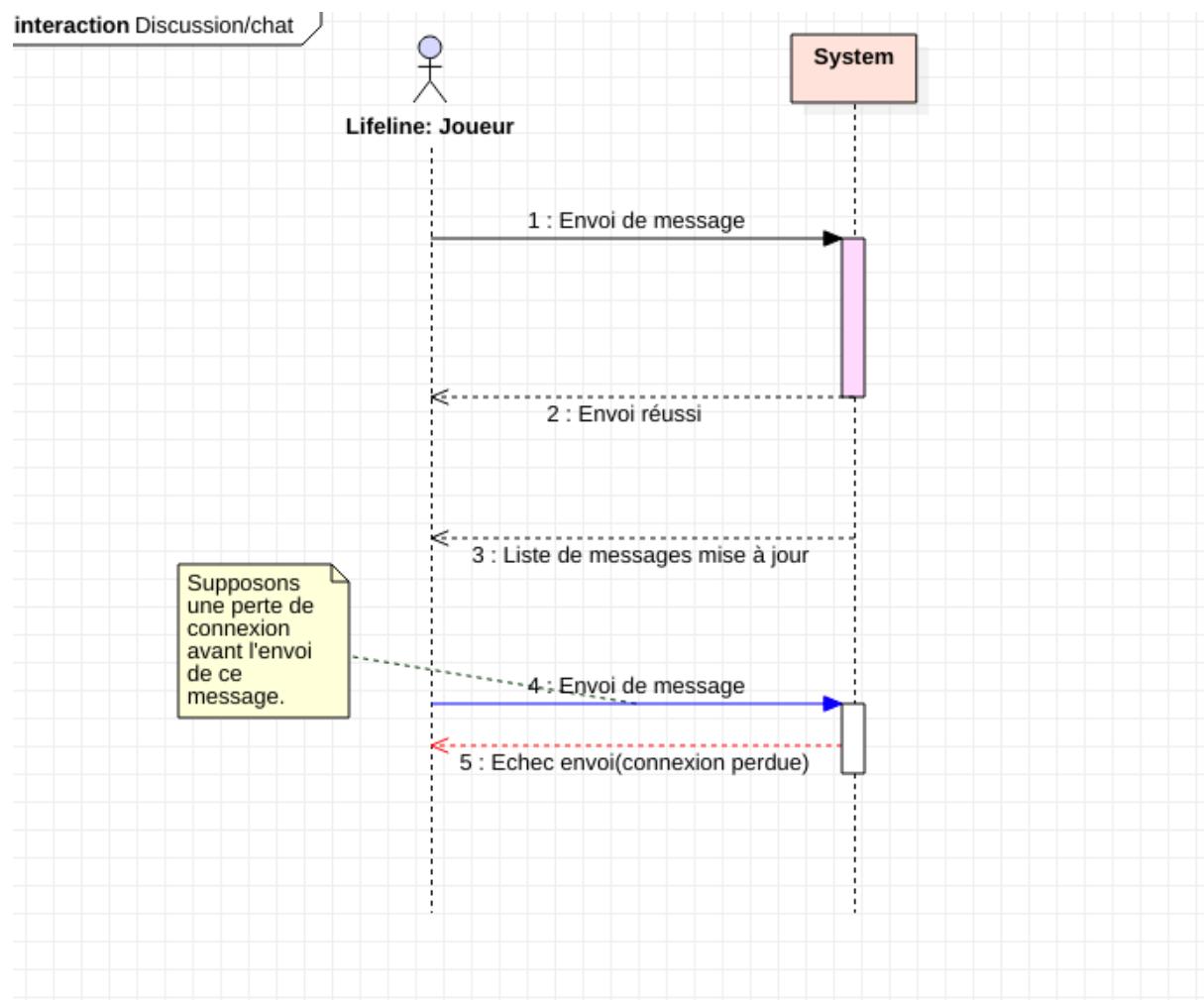


FIGURE 2.4 – Diagramme de séquence : Discussion/Chat

Le diagramme ci dessus représente les séquences d'envoi et d'accès aux messages présents dans la salle de discussion d'une partie. Il est valable aussi pour les discussions privées des Loup-garous dans leur repère. Ce "Chat" est le moyen de discuter les votes et ainsi éliminer au fur et à mesure des villageois.

2.3.4 4ème diagramme : Vote

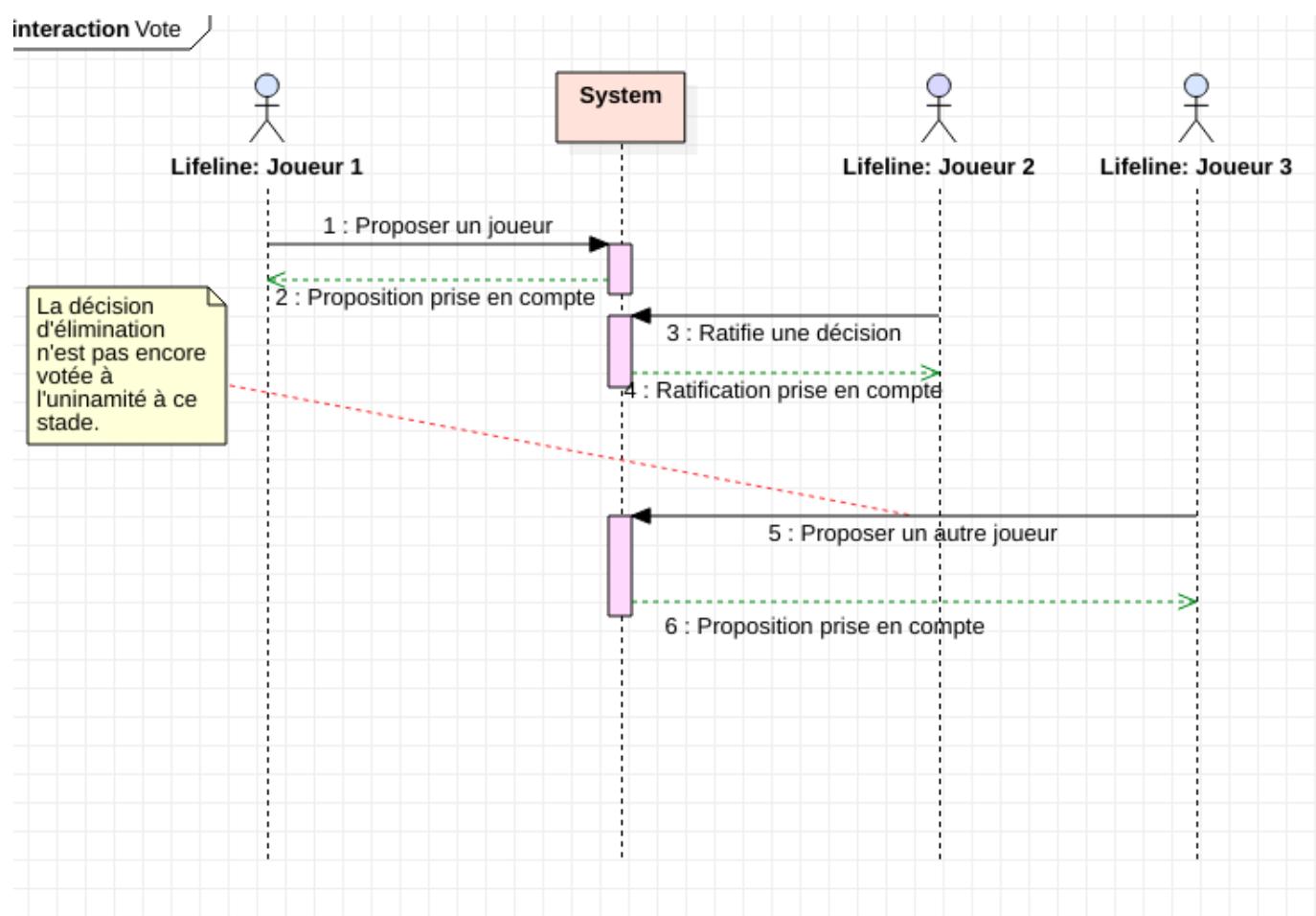


FIGURE 2.5 – Diagramme de séquence : Vote

Ce diagramme illustre une étape très importante du jeu : le vote. Pour éliminer un joueur, une proposition à la décision le concernant doit faire l'unanimité (ratification). Aussi, plusieurs propositions peuvent exister en même temps. Si aucune décision n'a été ratifiée à la fin de la journée, personne n'est tué. En revanche, les loups garous finissent toujours par se mettre d'accord à éliminer.

2.4 Diagramme de classes :

En génie logiciel, un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en affichant les classes du système, leurs attributs, opérations (ou méthodes) et les relations entre les objets. Ce diagramme nous permettra donc de schématiser notre application pour un meilleur déploiement.

Voici le diagramme de classes d'analyse de notre application :

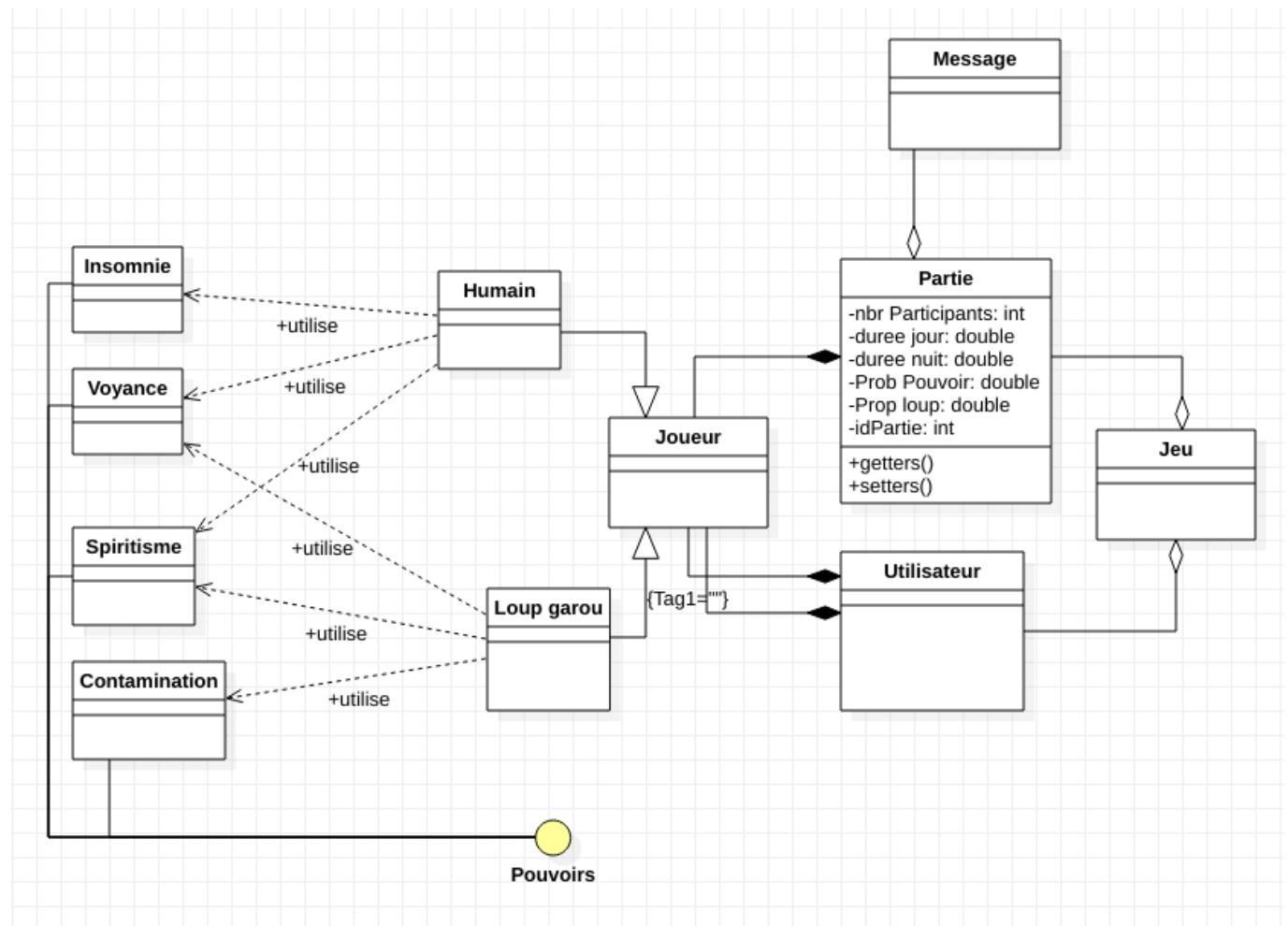


FIGURE 2.6 – Diagramme de classes d'analyse

Le diagramme ci-dessus représente une première structure de notre application. Celle ci va évoluer une fois l'implémentation commencée.

2.5 Diagramme d'objet :

Un diagramme d'objet dans le langage de modélisation unifié (UML) est un diagramme qui montre une vue complète ou partielle de la structure d'un système modélisé à un moment spécifique.

On propose ci-dessous un diagramme d'objet pour donner un exemple plus clair de ce dont ressemblera une partie de ce jeu et pour montrer les acteurs qui y contribueront avec les interactions nécessaires.

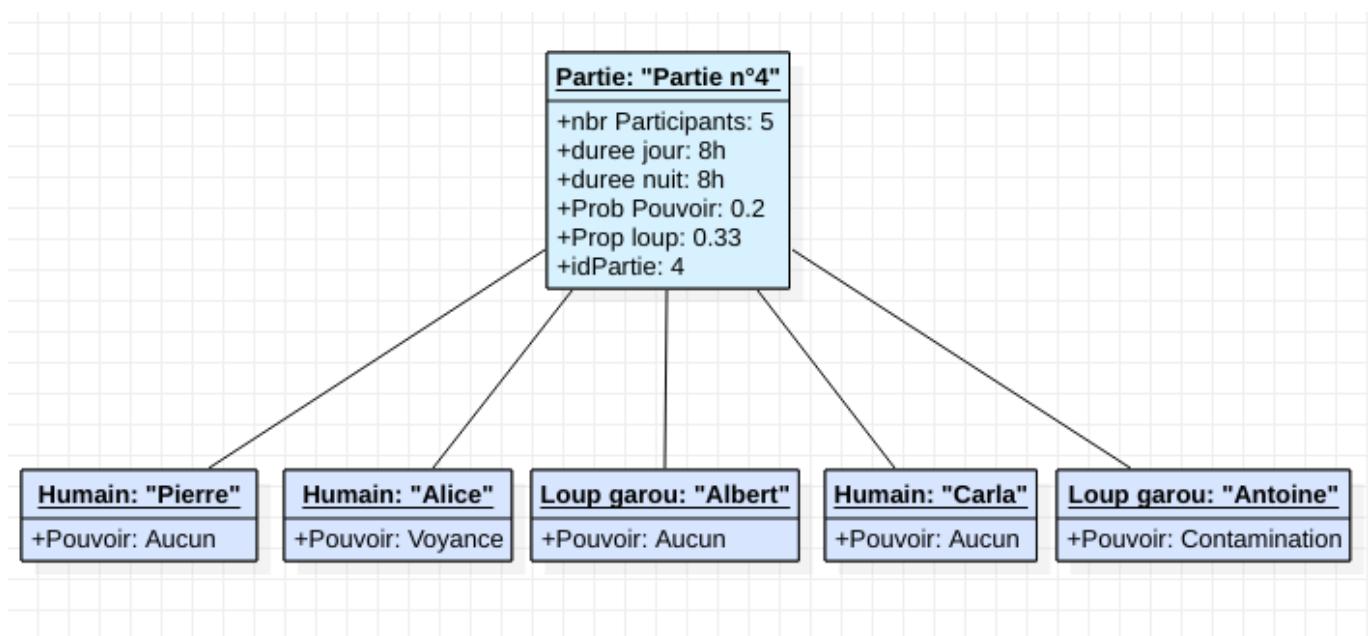


FIGURE 2.7 – Diagramme d'objets

Notre vue d'ensemble sur l'application est maintenant complète avec cet exemple. On peut désormais commencer la réflexion sur l'architecture globale de l'application ainsi que son implémentation.

Chapitre 3

Document de conception

Après le début de l'implémentation, en se basant notamment sur le document d'**Analyse**, vient l'étape de **Conception** où on va expliquer les choix pris avec des versions évolués des diagrammes précédents ainsi que d'autres nouveaux.

3.1 Architecture logique du système :

Vu la demande d'une interface Utilisateur/Graphique, nous avons décidé d'opter pour une architecture Modèle-Vue-Contrôleur.

Ce choix se justifie par le besoin de contrôler l'interface (**VUE**) dans sa relation avec une base de données (**MODELE**). Certaines actions sur les vues nécessiteront une récupération/modification du modèle, ce qui engendrera une modification de la vue.

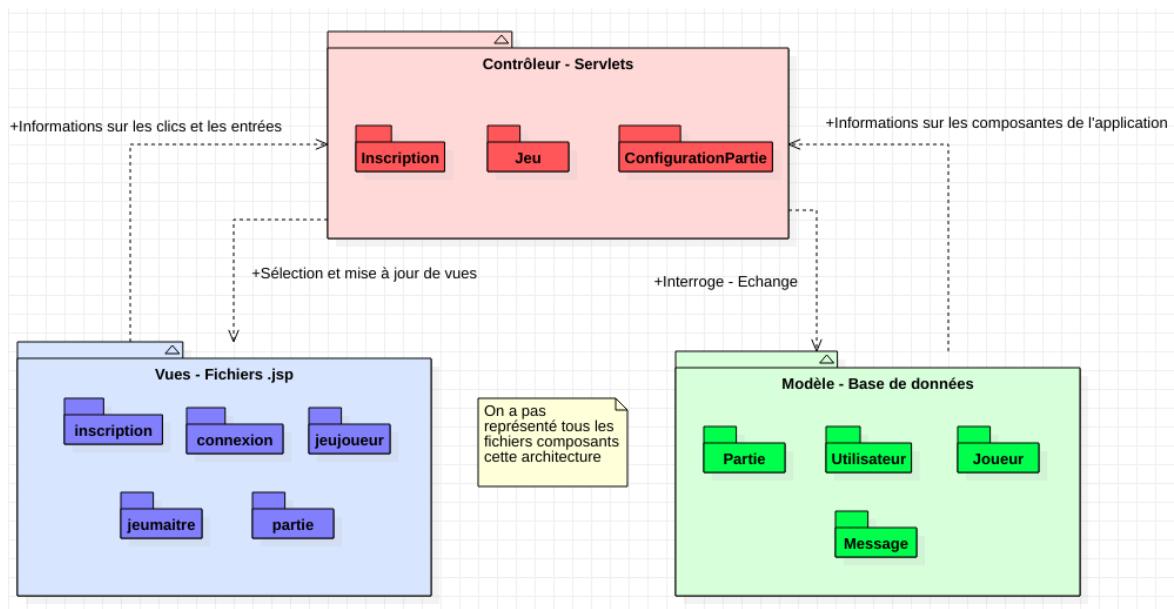


FIGURE 3.1 – Schéma de l'architecture logique du logiciel : architecture MVC

3.2 Description détaillée de l'incrément choisi

Cette section reviendra sur les fonctionnalités essentielles que propose notre application. Ces dernières suivent en majeure partie les cas d'utilisation dans le document d'analyse que vous pouvez trouver ici [2.2](#).

Commençons d'abord par les prérequis nécessaire pour le déploiement de l'application.

3.2.1 Prérequis :

Pour faire fonctionner l'application dans sa première version, il faut :

- Une connexion au VPN de Grenoble INP ;
- Maven et le serveur Apache Tomcat.

3.2.2 S'inscrire

L'utilisateur peut s'inscrire à l'application en fournissant un pseudonyme non déjà existant, un email et un mot de passe. Si les informations entrées sont valides, elles sont directement sauvegardées dans la base de données (voir [3.3.2](#)). Toutes les informations sont enregistrées en clair sauf pour le mot de passe.

3.2.3 Se connecter et se déconnecter de l'application :

L'utilisateur peut se connecter à l'application à l'aide d'un pseudonyme et d'un mot de passe.

L'utilisateur peut aussi se déconnecter à tout moment grâce à un bouton dédié à ceci.

3.2.4 Compte personnel :

Les utilisateurs de l'application (ceux qui se sont inscrits) ont chacun leur compte personnel qui est régulièrement mis à jour selon les parties de jeu qui sont en cours. L'interface principale dispose de boutons permettant à l'utilisateur soit d'accéder aux **Parties en cours** pour tenter de s'inscrire à l'une d'entre elles, soit d'en créer une nouvelle.

3.2.5 Jouer dans une partie :

Comme susmentionné, un utilisateur a le choix entre rejoindre une partie déjà créée mais non entamée, ou en créer une nouvelle.

Rejoindre une partie déjà existante

Les conditions pour rejoindre une partie déjà existante sont :

- Le nombre maximale de joueurs n'est pas encore atteint.
- l'horaire de début n'est pas encore atteinte.

Configurer une nouvelle partie

Un utilisateur peut à tout moment lancer une nouvelle partie, un bouton est dédié à ceci dans l'interface principale du compte de l'utilisateur.

une fois la configuration entamée, l'utilisateur est prié de rentrer un multitude d'information dans des champs associés à ceci. Bien entendu, ces champs sont soumis à certaines contraintes, mais l'utilisateur est notifié par des messages d'erreurs explicites si erreur il commet. Ce champs sont :

- le nombre de participants souhaité (ou une fourchette), par défaut entre 5 et 20 ;
- la durée du jour et de la nuit (les valeurs par défaut sont de 8h à 22h pour le jour et de 22h à 8h pour la nuit) ;
- l'horaire de début, par défaut le lendemain matin 8h00 ;
- la probabilité que chaque pouvoir spécial soit attribué (par défaut 0 : partie sans pouvoirs spéciaux) ;
- la proportion initiale de loups-garous (par défaut 1/3).

Une fois la configuration terminée, l'utilisateur qui a crée la partie reste en attente de nouveaux joueurs jusqu'à l'horaire de début. Si le nombre de participants souhaité n'est pas atteint, la partie est supprimée de la base de données.

3.2.6 Jeu

Le jeu suit dans les grandes lignes le cahier de charge et les cas d'utilisations spécifiés dans le document d'analyse.

Salle de discussion/Chat

Un Chat est mis à disposition de tous les joueurs pour discuter pendant le jour et débattre sur l'identité des loups garous. Cette même salle de discussion se transforme la nuit en repaire pour les loups garous. Ces derniers peuvent alors se mettre d'accord sur le villageois à éliminer. Les humains, ne pouvant pas accéder au chat pendant la nuit, la discussion des loups garous est alors privée.

Chaque passage à la nuit suivante est accompagné par l'archivage de la discussion de la journée.

Éliminer un joueur

Un joueur peut être éliminé dans deux cas de figures :

- Pendant la nuit, tué par les loups garous.
- Pendant le jour, soupçonné d'être un loup garou, après un vote à l'unanimité de tous les villageois.

Vote

Le vote dans l'application pour éliminer une personne se déroule de la manière suivante, on peut soit :

- Ratifier une décision déjà présente dans le tableau de votes.
- Proposer une nouvelle personne pour qu'elle soit soumise au vote.

Le jour arrivant à sa fin, la personne avec le plus de votes est alors tuée.

Pouvoirs spéciaux

Les joueurs dans une partie peuvent être dotés de pouvoirs spéciaux, ils sont au nombre de 4 :

- **Contamination** (disponible uniquement pour les loups-garous) : permet au joueur de transformer chaque nuit un humain en loup-garou.
- **Insomnie** (disponible uniquement pour les humains) : permet au joueur d'assister aux discussions du repaire des loups-garous (mais pas de participer en se faisant passer pour un loup-garou).
- **Voyance** : permet au joueur de connaître le rôle et les pouvoirs d'un joueur de son choix chaque nuit.
- **Spiritisme** : permet au joueur de parler avec un joueur éliminé de son choix chaque nuit. Il s'agit d'une salle de discussion similaire à la place du village mais avec seulement deux participants.

Les joueurs dotés d'un pouvoir ne peuvent l'exercer qu'une seule fois par nuit. La distribution des pouvoirs est dépendante de la probabilité de distribution entrée par le créateur de la partie (voir [3.2.5](#)).

Fin de la partie

La partie se termine quand il ne reste que des humains ou que des loups garous. Une fois le gagnant annoncé, tous les joueurs ont accès aux archives de discussions.

3.3 Conception détaillée de l'incrément choisi

3.3.1 Diagramme de classes logicielles

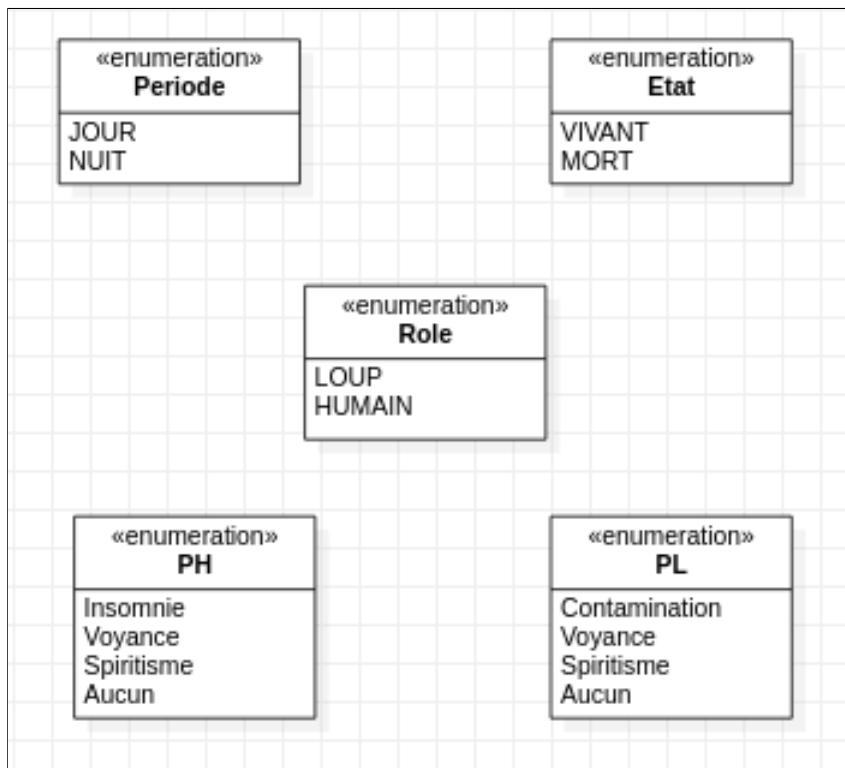


FIGURE 3.2 – Les différentes énumérations utilisées pour la représentation de notre diagramme de classes

Avantage des énumérations :

Il s'agit d'une vue statique des énumérations utilisées qui modélisent certains concepts de notre système. La raison pour laquelle on a choisi de telles énumérations est de fournir un diagramme de classes clair et lisible. D'autre part, On peut utiliser les énumérations comme attributs, ceci nous permet d'éviter des associations et des agrégations supplémentaires.

Dans cet ordre d'idées, voici le diagramme de classes illustrant notre implémentation :

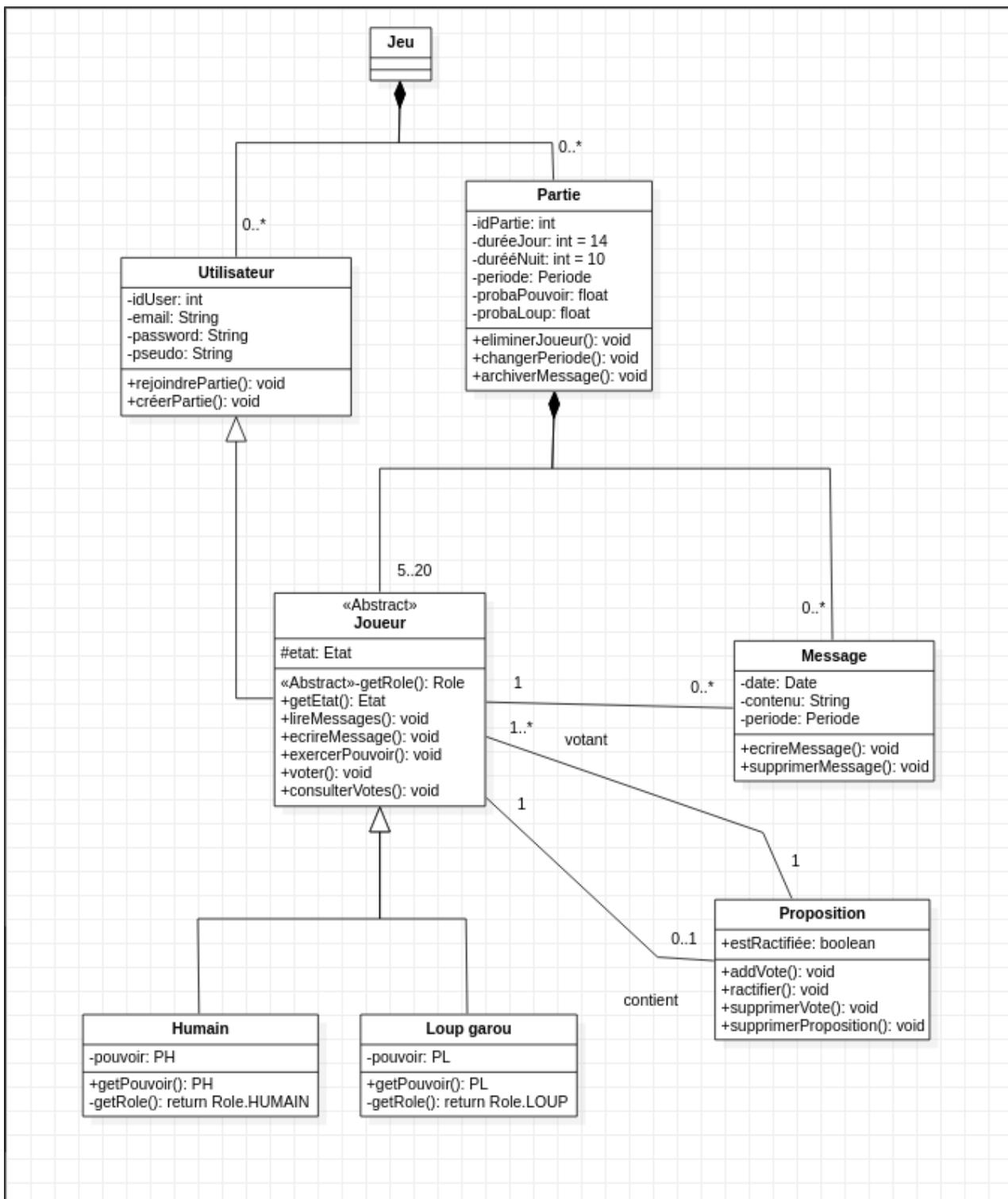


FIGURE 3.3 – Diagramme de classe de conception

Le diagramme de classe de conception ci-dessus vient en premier par rapport aux documents de conceptions. Il met la lumière sur certaines classes d'analyse : **Partie, Utilisateur, Joueur, Message**.

On a donné une version différente adaptée au langages Orientés Objets au terme d'implantation, elle traite aussi la phase de vote ainsi que la précision des attributs et des méthodes.

Les accès à la base de données sont gérés par les DAO (Date Access Object) mis en place pour certaines des classes représentées dans le schéma précédent. En voilà une partie :

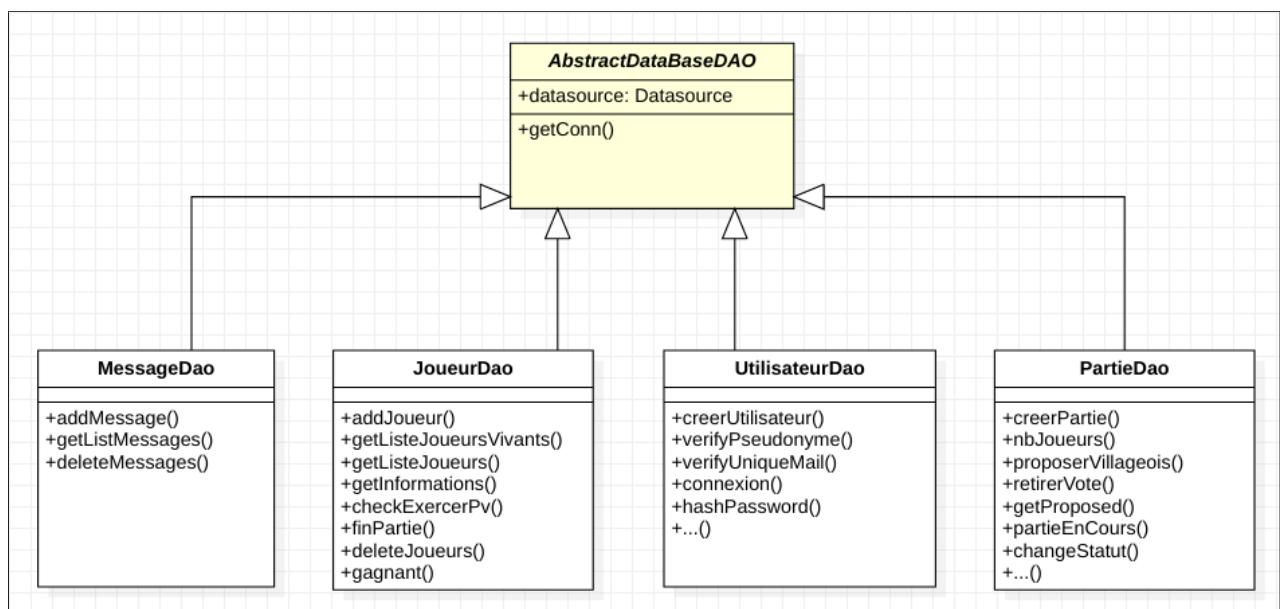


FIGURE 3.4 – Dao relatif à quelques classes de l'application

Ces classes étendent une abstract class **AbstractDataBaseDAO** pour une meilleure factorisation du code

3.3.2 Conception de la base de donnée

Nous avons implémenté la base de données en se basant sur le schéma entité association ci dessous :

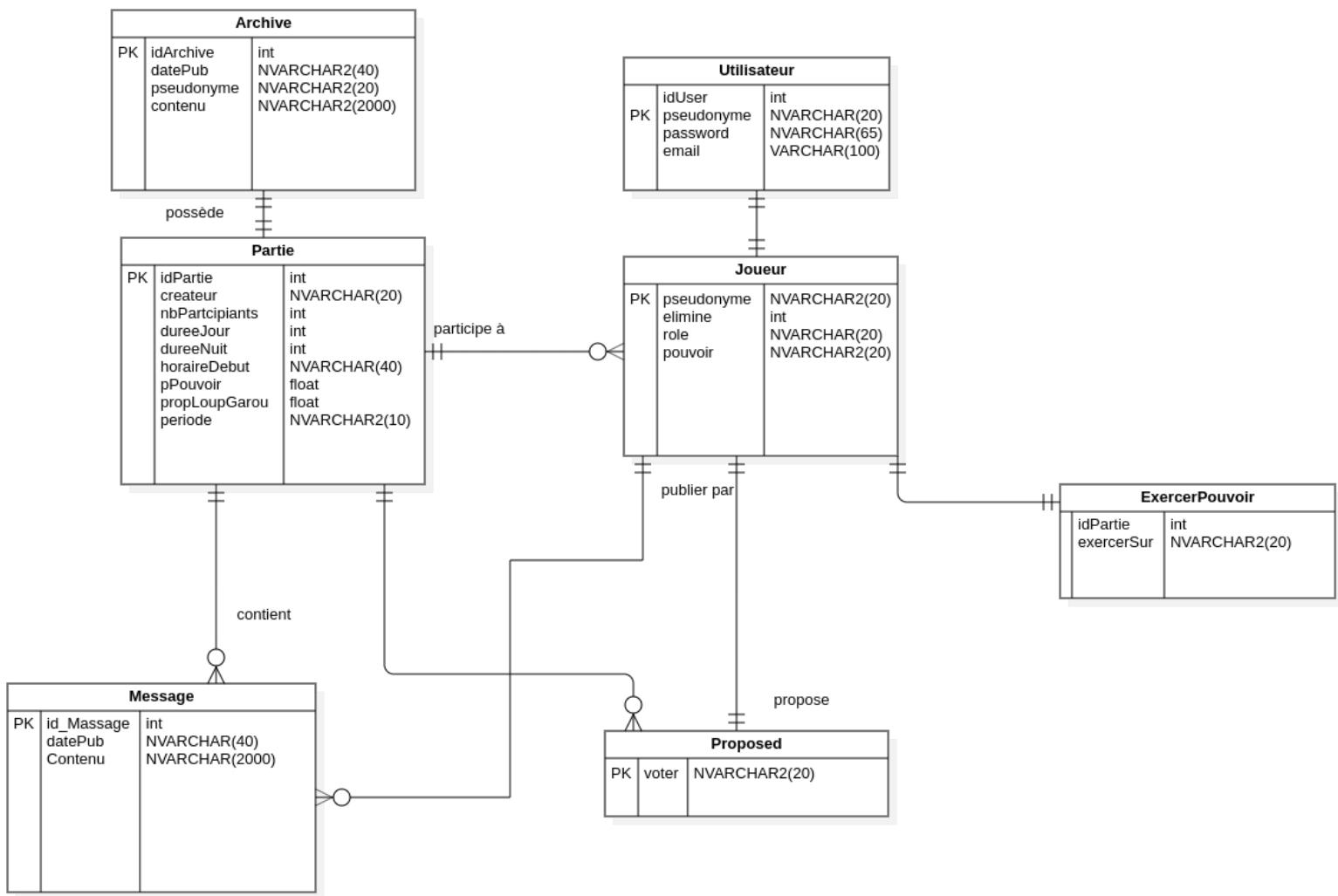


FIGURE 3.5 – Schéma entité-association de la base de données

Le schéma relationnel associé au schéma entité-association ci dessus est le suivant :

- **Utilisateur**([idUser](#), pseudonyme, password, email).
- **Joueur**([idPartie](#),pseudonyme, elimine, role, pouvoir)
- **Partie**([idPartie](#), createur, nbParticipants, dureeJour, dureeNuit, horaireDebut, pPouvoir, propLoupGarou, periode)
- **MassageJour**([idPartie](#),[id_MessageJour](#), datePub, pseudonyme, contenu)
- **MassageNuit**([idPartie](#),[id_MessageNuit](#), datePub, pseudonyme, contenu)
- **Archive**([idPartie](#), [idArchive](#), datePub, pseudonyme , contenu)
- **ExercerPouvoir**([idPartie](#), [exercerPar](#), exercerSur)
- **Proposed**([idPartie](#), pseudonyme, voter)
- **Removed**([idPartie](#), pseudonyme)

3.3.3 Diagrammes de séquence

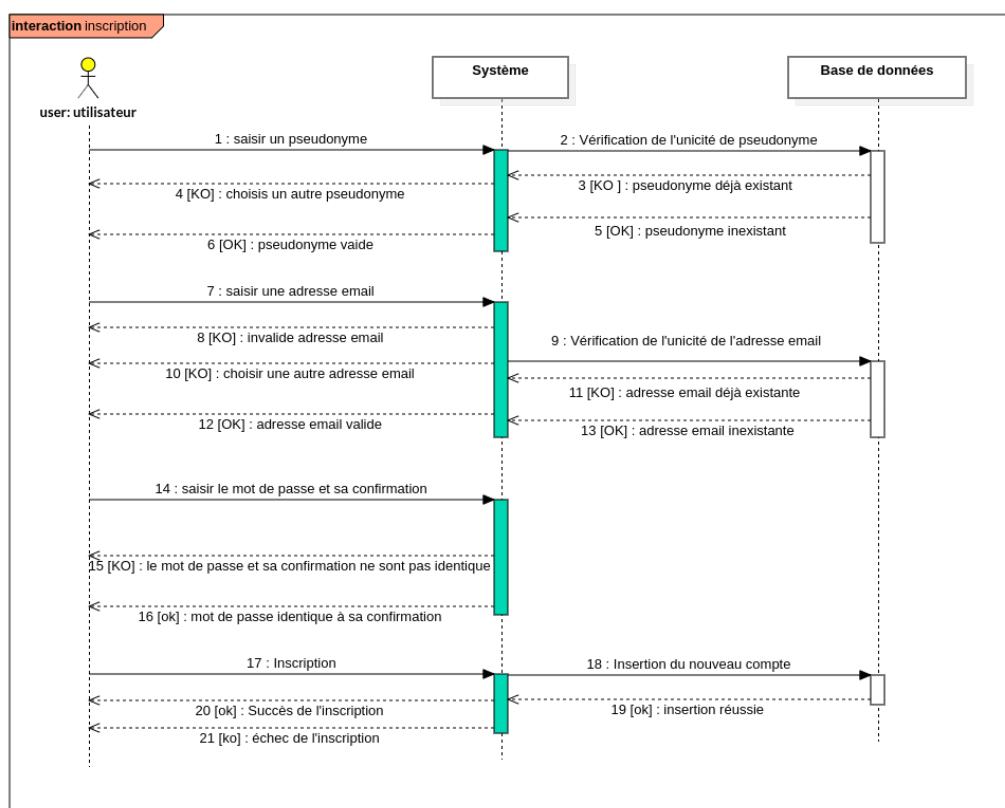


FIGURE 3.6 – Inscription

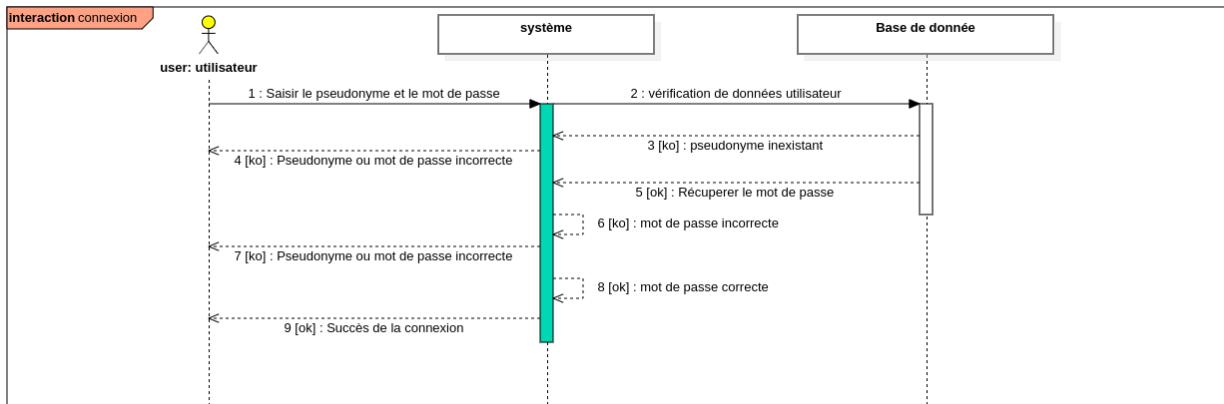


FIGURE 3.7 – Connexion

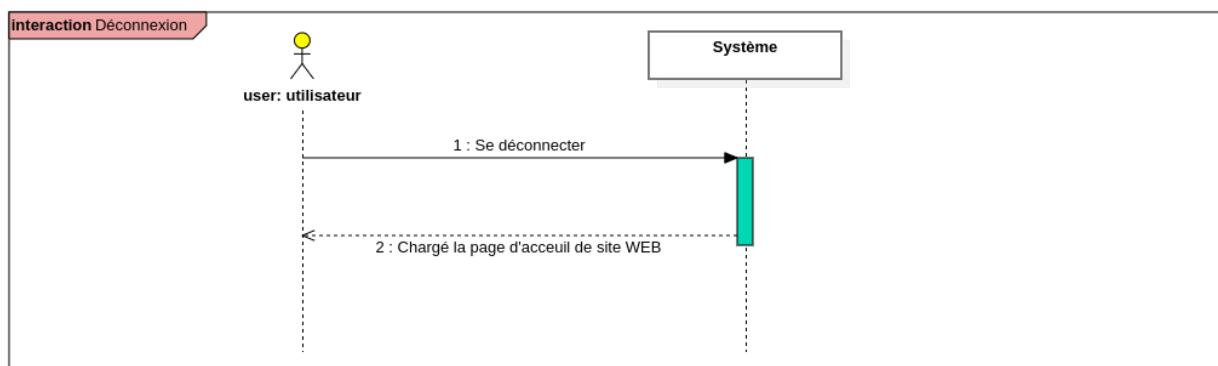


FIGURE 3.8 – Déconnexion de l'application

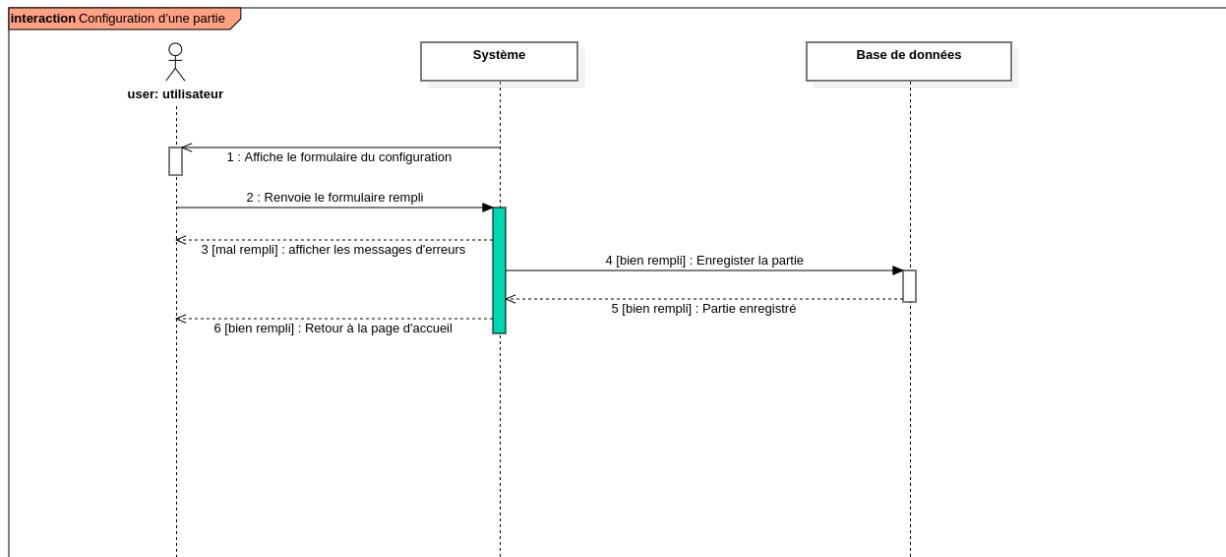


FIGURE 3.9 – Configuration d'une partie

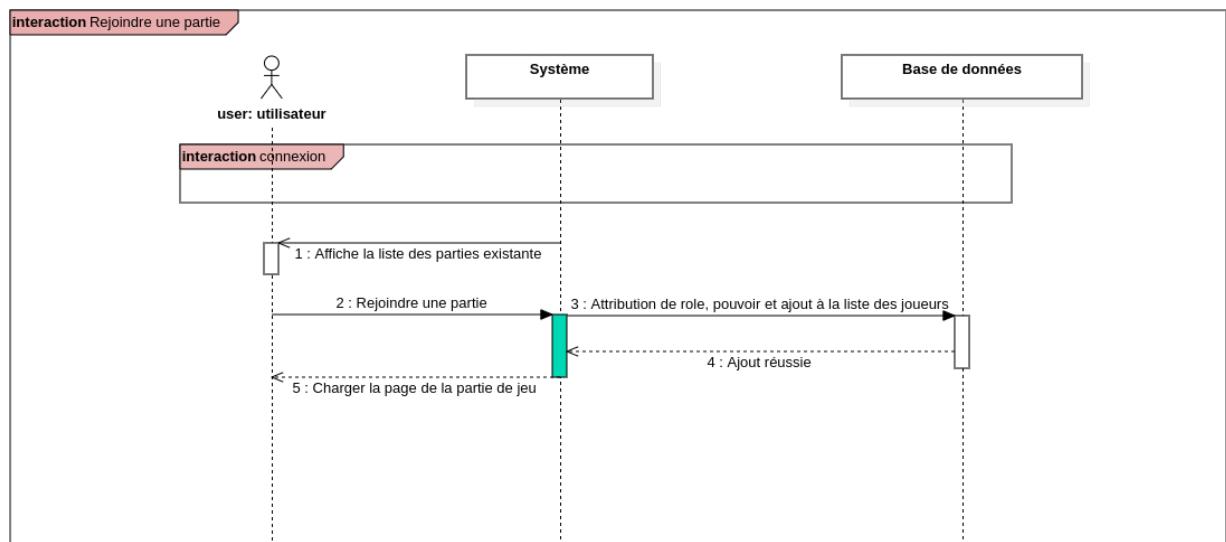


FIGURE 3.10 – Rejoindre une partie

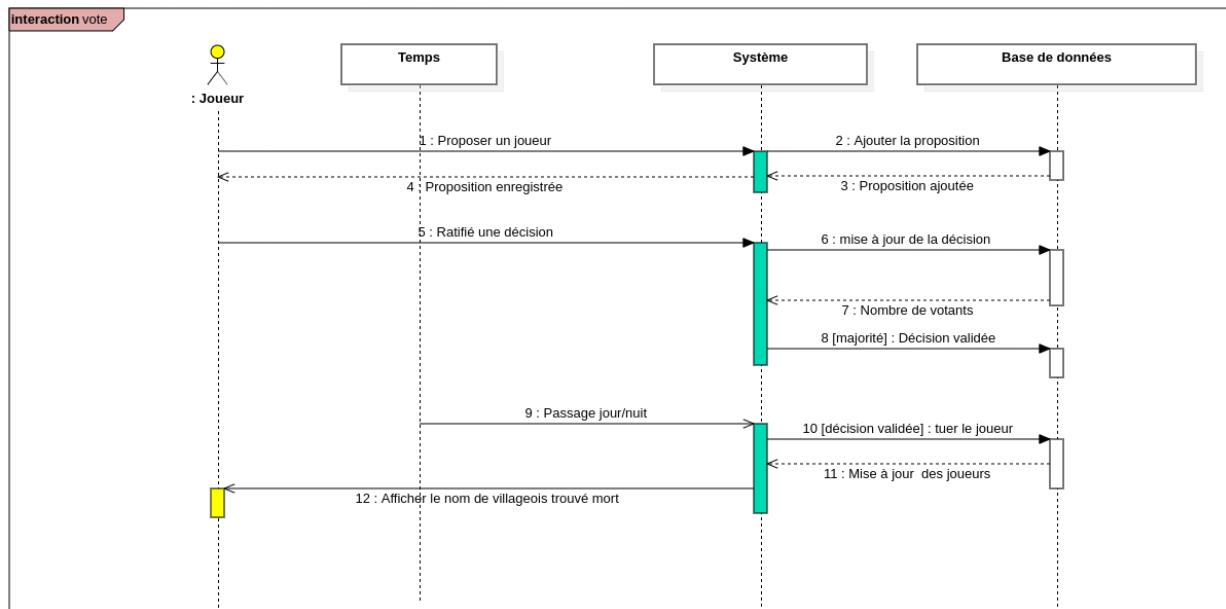


FIGURE 3.11 – Vote

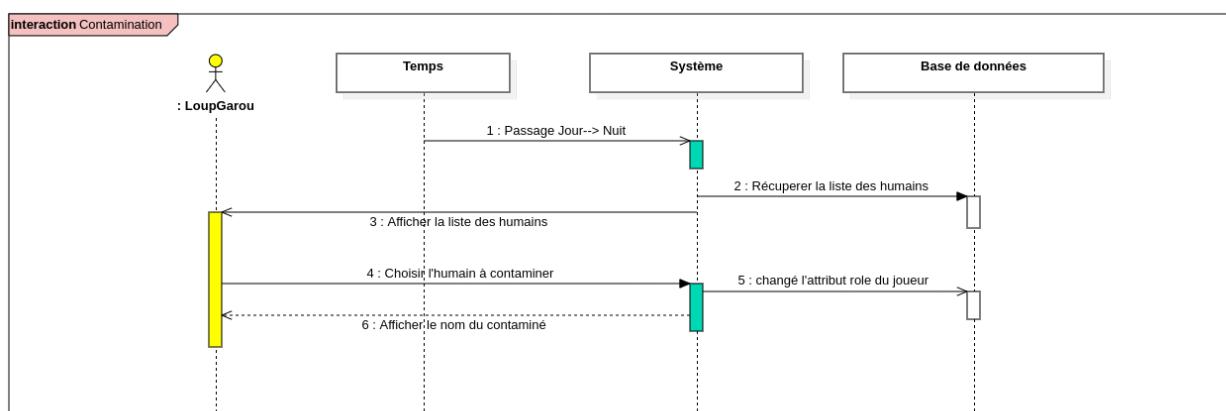


FIGURE 3.12 – Contamination

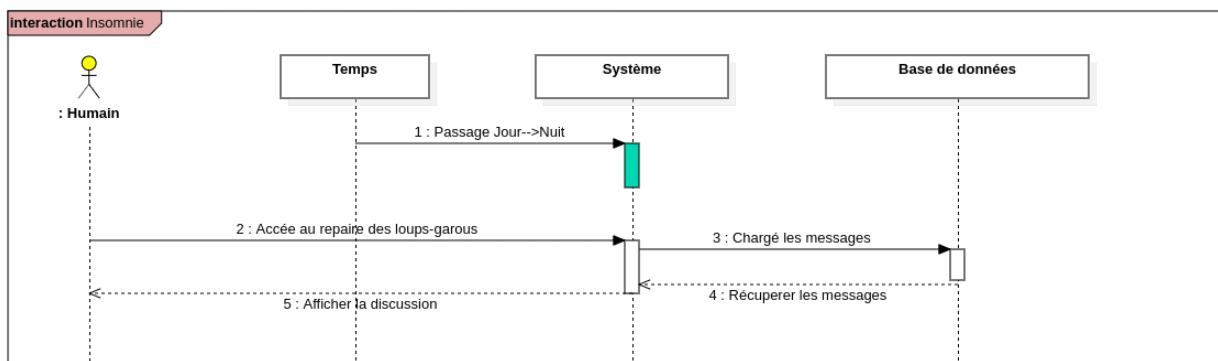


FIGURE 3.13 – Insomnie

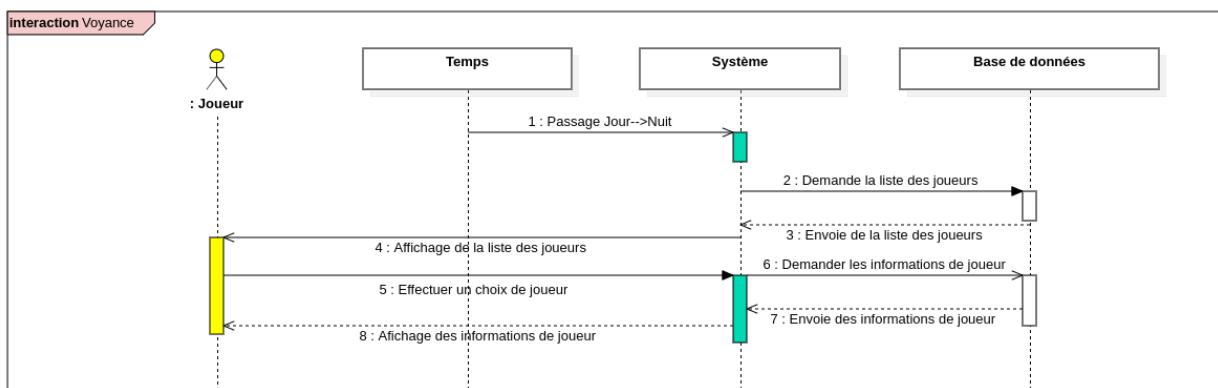


FIGURE 3.14 – Voyance

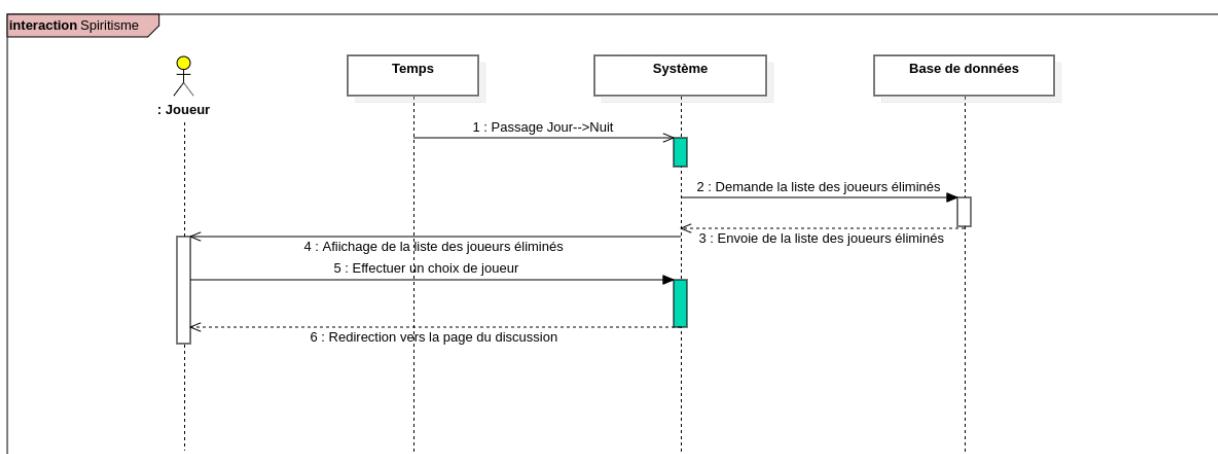


FIGURE 3.15 – Spiritisme

3.4 Diagrammes d'états-transition :

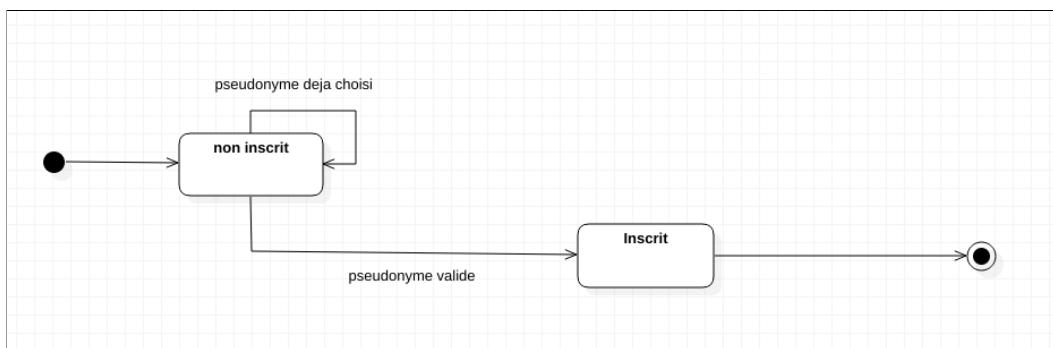


FIGURE 3.16 – diagramme état transition pour l'inscription

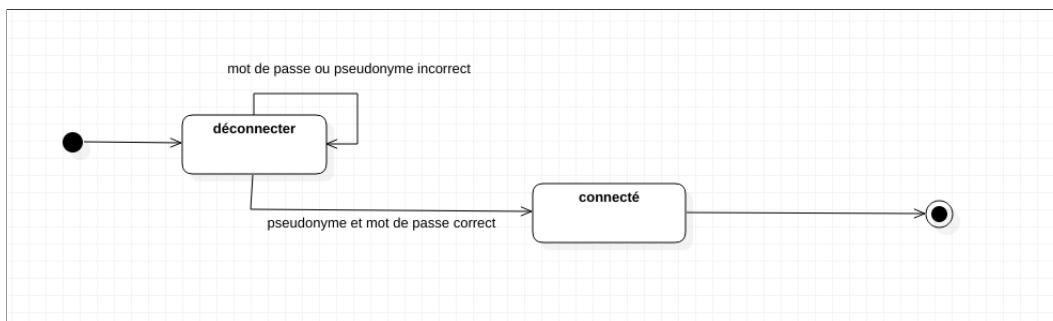


FIGURE 3.17 – Diagramme état transition pour la connexion

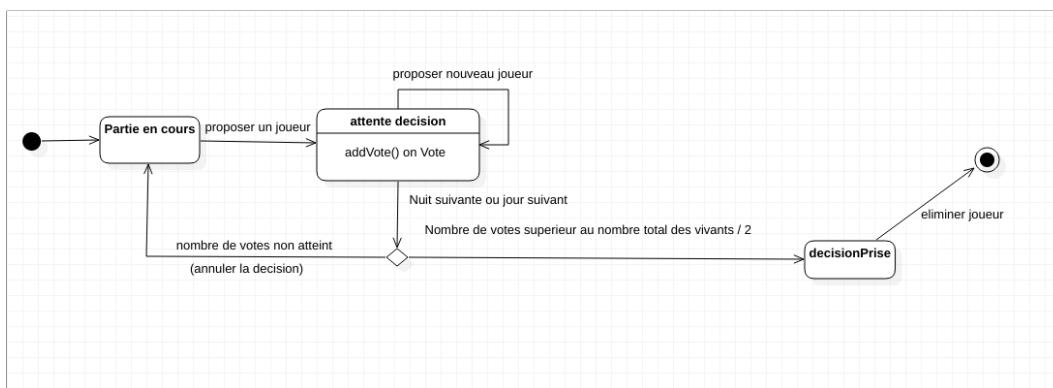


FIGURE 3.18 – Diagramme état transition pour les votes

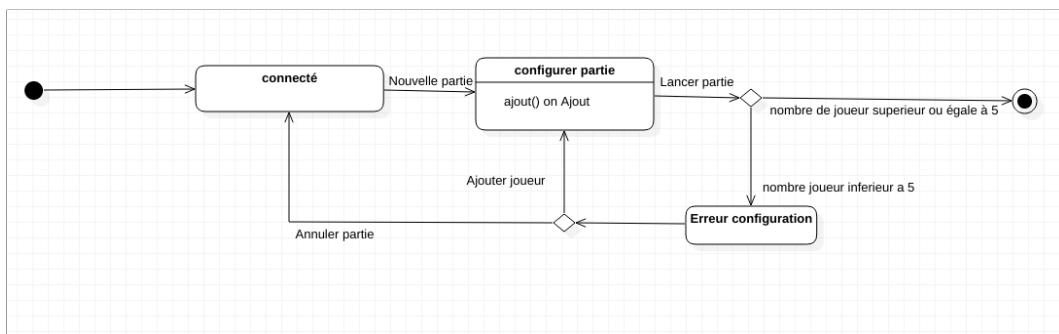


FIGURE 3.19 – Diagramme état transition pour la configuration d'une partie

3.5 Validation :

3.5.1 Peuplement de la base de données :

Dans la version finale de notre application, elle sera connectée à une base de données centrale très performante. Ce qui permettra notamment la gestion d'un très grand nombre de sessions à la fois ainsi que le traitement d'un plus grand nombre d'informations. Mais avant le déploiement, il est crucial de pouvoir réaliser des tests sur notre outil. Et pour ce faire, on a utilisé une génération fictive permettant de peupler notre base de données.

Deux méthodes peuvent être utilisées dans ce dessein :

- Génération de sessions à travers une série d'inscriptions. En effet, cela ajoute directement les informations concernant cette personne dans la base de données. On peut faire ça aussi pour une Partie.
- Un script SQL avec une série de requêtes peuplant la base de données.

3.5.2 Description détaillée de la base de données :

- **Utilisateur** : les utilisateurs de l'application, ceux qui ont un compte.
- **Joueur** : un utilisateur participant à une partie devient joueur durant la durée de celle-ci.
- **Partie** : une partie de jeu du loup garou.
- **MessageJour** : les messages échangés entre les joueurs pendant le jour.
- **MessageNuit** : les messages échangés entre les joueurs pendant la nuit.
- **Archive** : où sont stockés les messages archivés d'une partie.

- **ExercerPouvoir** : cette table sert à stocker les fois où un pouvoir a été exercé.
- **Proposed** : contient les personnes proposées comme candidats à l'élimination pendant un tour.
- **Removed** : contient les personnes éliminées pendant une partie.

Chapitre 4

Manuel utilisateur

Nous donnerons dans cette partie quelques directives nécessaires concernant l'utilisation de cette application. Une attention particulière a été portée à l'ergonomie de ce jeu pour faciliter davantage son utilisation.

Dans les vues concernant le jeu, une liste des joueurs ayant comme fond d'écran une image (voir 4.1) est affiché et mise à jour chaque tour. Par souci de clarté, les prise de vues des différentes pages du jeu présentes dans ce manuel ne l'incluent pas.



FIGURE 4.1 – Liste des joueurs

4.1 Prérequis :

Pour pouvoir assurer le bon fonctionnement de l'application, assurez-vous d'être connecté au VPN de Grenoble INP (vpn.grenet.fr). Ceci est nécessaire pour la connexion à la base de données où toutes les informations concernant les joueurs/parties seront stockées.

Aussi, pour lancer le projet, le serveur Apache Tomcat ainsi que Maven sont nécessaires.

4.2 Inscription/Connexion :

4.2.1 Connexion :



FIGURE 4.2 – Page de connexion

Lorsque l’application est lancée, l’utilisateur est face à une page où il lui est demandé de se connecter (si jamais il avait déjà un compte). Et ceci en renseignant ses identifiants de connexion indispensables à l’utilisation de l’application.

Après avoir renseigné ses identifiants, l’utilisateur peut cliquer sur le bouton **Connexion**.

Si les identifiants sont corrects, il est alors redirigé vers l’interface principale de son compte ; sinon, un message d’erreur s’affiche et l’utilisateur peut alors réessayer.

Si l'utilisateur n'a pas de compte, il peut commencer son inscription en cliquant sur le bouton **Inscription**. Il sera alors redirigé vers une autre page.

4.2.2 Inscription :

You can register via this form.

Pseudonyme*

Adresse email *

Mot de passe *

Confirmation du mot de passe *

Inscription

Retour à la page de connexion

FIGURE 4.3 – Page d’inscription

Pour finaliser son inscription, il est demandé de renseigner :

- Un pseudonyme (qu'il devra changer si jamais ce pseudonyme existe déjà).
- Une adresse mail valide.
- Un mot de passe et sa confirmation.

Si l'inscription échoue, un message explicite s'affiche expliquant les erreurs commises. Sinon, le message "*Inscription réussie*" s'affiche. L'utilisateur peut dès lors revenir à la page de connexion (bouton **Retour à la page de connexion**) et se connecter pour accéder à son compte personnel.

4.3 Page principale :



FIGURE 4.4 – Page de bienvenue

Si la connexion aboutit, l'utilisateur est redirigé vers la page principale de son compte.

Deux cas de figures existent pour cette page :

- Si aucune partie de jeu n'est en cours, une page avec un message d'accueil et un bouton **Nouvelle partie** s'affiche (figure 4.3). Si on clique sur ce bouton, on est redirigé vers la page de configuration (figure 4.4).
- Sinon, une page montrant des informations sur la partie en cours s'affiche. En effet, seule une partie peut avoir lieu en même temps. Celle-ci doit alors s'achever avant qu'un autre maître de jeu puisse commencer une autre (figure 4.3).



FIGURE 4.5 – Page montrant la partie en cours

4.4 Configuration d'une nouvelle partie :

Commencez la configuration de la partie

Selectionner les utilisateurs à jouer dans la partie ,au minimum 5 joueurs

bensrhin [ajouter](#)
 Maria [ajouter](#)
 Antoine [ajouter](#)
 Albert [ajouter](#)
 Carla [ajouter](#)
 Hamza [ajouter](#)

Probabilité d'attribution des pouvoirs *

Proportion des loups garous *

Lancez la partie

FIGURE 4.6 – Page de configuration

Pour configurer une partie, l'utilisateur qui deviendra maître du jeu est prié d'indiquer en premier lieu les personnes qui pourront jouer (en cliquant sur **ajouter** devant le pseudo de la personne), avec un minimum de 5 personnes. Si manque il y a dans le nombre de personnes ayant une session ouverte, le maître du jeu devra attendre l'arrivée d'autre potentiels joueurs.

Ensuite, l'utilisateur devra renseigner les champs suivants :

- La probabilité que chaque pouvoir spécial soit attribué (par défaut 0 : partie sans pouvoirs spéciaux).
- La proportion initiale de loups-garous (par défaut 1/3).

Pour lancer la partie, il suffit ensuite de cliquer sur le bouton **Lancez la partie**.

4.5 Jeu :

4.5.1 Maître du jeu :

Maintenant que la partie est entamée, les participants auront des pages consacrées à la partie. Nous parlerons en premier lieu du Game Master, le maître du jeu.

Rappelant d'abord son rôle :

- Seul le maître du jeu peut créer une partie.
- C'est au maître du jeu d'enclencher manuellement les passages jour/nuit.

Le reste des fonctionnalités du jeu sont gérées automatiquement par l'application.



FIGURE 4.7 – Page principale du maître du jeu

La page principale du maître du jeu après le lancement de la partie présente un seul bouton **Gérer la partie en cours**. Ce bouton permet d'accéder à la partie qu'il vient de créer.

Voici la page qui suit le click sur ce bouton :



FIGURE 4.8 – Page de la partie du maître du jeu

La page comporte :

- Un message d'accueil
 - Deux boutons **Nuit** et **Jour**.
 - Un Chat (salle de discussion).
- Les boutons servent de bascule du **jour** à la **nuit** et vice versa.

Le maître ne peut participer à la discussion.

4.5.2 Joueur :

Parlons maintenant du déroulement de la partie pour un joueur normal. Si ils sont sélectionnés pour participer dans une partie, et après leurs connexions, ils sont directement redirigés vers une page d'accueil avec un seul bouton **Accéder à la partie en cours** les invitant à rejoindre la partie dans laquelle ils sont inscrits.



FIGURE 4.9 – Page principale d'un joueur

Il y a deux types de joueurs, **humain** et **loup garou**.

Humain :

Les humains dans le jeu sont des villageois normaux, à moins qu'ils soient dotés du pouvoir de "**Voyance**". Dans ce cas, ils peuvent connaître le rôle et les pouvoirs d'un joueur de leur choix chaque nuit.

Voici une vue de l'écran principal d'un humain dans une partie :



FIGURE 4.10 – Page de la partie d'un humain

On peut voir dans la figure ci-dessus que le joueur a un petit rectangle récapitulant toutes les informations le concernant lui et l'état de la partie (nuit ou jour).

Sur la gauche, des informations concernant les précédentes éliminations sont données.

Le joueur, pendant le jour, peut proposer un villageois à la décision d'élimination. Cette décision est à débattre avec le reste des joueurs dans la salle de discussion dédiée (en bas à gauche). Pour proposer quelqu'un au vote, il suffit de choisir un joueur parmi ceux présents dans le menu déroulable et cliquer ensuite sur **Valider**.

Loup garou :

Le rôle des loups garous dans la partie est d'éliminer un à un les humains. La partie est gagnée si ils y parviennent.

Ils peuvent aussi les contaminer si ils sont dotés de ce pouvoir.

L'écran principal d'un loup garou est similaire à celui d'un humain sauf bien sûr dans le rectangle d'information où il est stipulé cette fois ci que la personne est bel et bien un loup garou.

Les loups garous peuvent eux aussi proposer des personnes pour qu'elles soient éliminées et participer à la discussion dans l'espace dédié à ceci.

Cependant, pendant la nuit, les loups garous se rassemblent dans leurs repaires (la salle de discussion se transforme en repaire la nuit) pour se mettre d'accord sur la personne à éliminer.

L'élimination d'un joueur se passe exactement comme une élimination par votes pendant le jour (voir section 4.7).

Voici une vue illustrant ceci :



FIGURE 4.11 – Repaire des loups garous

Pour éliminer une personne, les loups garous se mettent d'accord dans leur repaire sur qui tuer. Les humains ne pouvant pas accéder au Chat le soir, la discussion des loups garous est entièrement privée.

4.6 Pouvoirs :

4.6.1 Voyance chez les humains :

Comme déjà expliqué, ce pouvoir permet au joueur de connaître le rôle et les pouvoirs d'un joueur de son choix chaque nuit.

Voici une vue illustrant ceci :



FIGURE 4.12 – Pouvoir de voyance

Pour voir les informations concernant un joueur, il suffit d'en choisir un parmi ceux présents dans le menu déroulable et cliquer ensuite sur **Regarder des informations sur ce joueur**. Une fois que le pouvoir est exercé, un message s'affiche à l'écran avec le rôle et le pouvoir (si jamais il y a) de la personne choisie.

4.6.2 Contamination chez les loups garous :

Les loups garous peuvent à leur tour être dotés d'un pouvoir bien utile, la **Contamination**.

Ce pouvoir permet au joueur de transformer chaque nuit un humain en loup garou.



FIGURE 4.13 – Pouvoir de contamination

Pour contaminer un joueur, il suffit d'en choisir un parmi ceux présents dans le menu déroulable et cliquer ensuite sur **Contaminer cet humain**. Une fois que le pouvoir est exercé, un message s'affiche à l'écran confirmant cette contamination.

4.7 Vote :

Cette application ne gère pas les votes automatiquement. Au moment où le maître du jeu déclenche le passage **jour→nuit** ou **nuit→jour**, il indique qui a été tué pendant ce tour, ce dernier étant celui qui a reçu le plus de votes. Pour voter contre quelqu'un, on peut soit :

- Ratifier une décision déjà présente dans le tableau de votes, en cliquant sur **Voter** à côté du pseudo de la personne.
- Proposer une nouvelle personne (comme déjà expliqué dans la rubrique **Humain**, voir 4.10).

The screenshot shows a game interface titled "les loups-garous vs les humains". At the top, there is a blue box containing player information: "Nom du joueur : 'Albert'", "Rôle : Loup Garou.", "Pouvoir : Contamination.", and "Période : Jour". Below this, a message box displays a proposal from Albert: "Les gens, Votez contre Carla, elle est très évasive". On the left, there is a dropdown menu "Proposer un villageois : Choisir un villageois" with a "Valider" button. In the center, a table lists proposed villagers and their status: "Villageois Proposés" (Carla, Albert), "Ratifier la proposition" (links to "voter"), and "Nombre de votes" (1 for both). A message box at the bottom right says "Your message" and "Send".

Villageois Proposés	Ratifier la proposition	Nombre de votes
Carla	voter	1
Albert	voter	1

FIGURE 4.14 – Votes

4.8 Fin de partie :

Une fois la partie terminé, l'interface principale de leur compte change, un bouton **Accéder à l'archive de la partie précédente** est ajouté leur permettant d'accéder à l'archive de la dernière partie jouée.



FIGURE 4.15 – Compte d'un joueur après avoir joué une partie

4.9 Déconnexion :

Pour se déconnecter, rien de plus simple, il suffit de cliquer sur le bouton **Déconnexion** présent en haut à droite dans la page d'accueil de tout joueur (voir figure 4.3).

Chapitre 5

Bilan sur les outils de modélisation

Pendant ce projet, pour la partie modélisation, nous avons utilisé en majeure partie **Staruml**.

Ce logiciel open source est simple d'utilisation et très intuitif pour un débutant. Et un de ses avantages est qu'il possède une documentation très détaillée : <https://docs.staruml.io/>.

Ce choix est bien réfléchi, et a été décidé après le test de plusieurs autres modeleurs tel que **draw.io** ou encore **Visual paradigm for uml**. En effet, vu l'importance de la modélisation UML dans l'analyse et la conception de tout projet, l'offre de tels outils est abondante et le choix de l'un d'entre-eux est difficile car ils semblent tous offrir les mêmes services.

Staruml avait un gros avantage dans l'organisation de fichiers uml. En effet tous les diagrammes, en phase de construction, peuvent être enregistrés dans un seul fichier d'extension **".mdj"**. Cependant, la navigation entre les différentes fenêtres (différents diagrammes UML) est fastidieuse au début et demande un certain temps d'adaptation. Malheureusement, les diagrammes ne sont pas tous affichés à l'écran en même temps.

L'ergonomie du logiciel est très agréable. le logiciel est divisé en 5 zones principales :

- Le "model explorer", représenté sous la forme d'une arborescence, ce qui facilite (après avoir pris l'habitude) la navigation entre les différents diagrammes d'un projet et leurs entités.
- La partie "toolbox" qui fournit les différentes entités qu'on ajoute à nos diagrammes.
- La partie "properties" qui permet de modifier les propriétés d'une entité.
- L'espace de travail où on édite les diagrammes.

Un autre point positif est la possibilité de modifier le styles de nos entités. La couleur de fond et police d'écriture sont donc personnalisables. On peut aussi choisir la forme des associations (oblique ou rectiligne).

De manière général, ce projet nous a permis de prendre conscience de l'importance de la modélisation UML et des différentes étapes que sont l'analyse et la conception.

En effet, en faisant des recherches sur des outils de modélisation appropriés à utiliser, on est tombé sur une multitude de forums de professionnels débattant sur tel ou tel outil. Ceci s'explique entre autre par le fait qu'il existe des outils qui permettent une génération de code en C++ et Java, ce qui peut être très utile pour poser un premier squelette du projet à partir d'un diagramme de classes par exemple.