

Projet WEB/ACOL

BERGONZOLI Maud, LURI VAÑÓ Jorge, PAULY Cédric, WILLEM Logan

Résumé

Dossier regroupant les rendus demandés concernant la partie ACOL, du projet "Loup-garou" joint entre les matière CAWEB et ACOL.
Projet effectué dans le cadre de l'enseignement de seconde année, second semestre à l'Ensimag en filière ISI, année 2022-2023.

Table des matières

1	Introduction	3
2	Analyse du projet	3
2.1	Description des acteurs	3
2.2	Cas d'utilisation	3
2.3	Diagrammes de séquence	5
2.3.1	Inscription avec cas d'échec	5
2.3.2	Connexion avec cas d'échec	5
2.3.3	Création d'une partie	6
2.3.4	Vote	6
2.4	Diagramme de classe d'analyse	7
2.5	Diagramme d'états-transitions	7
3	Conception du projet	8
3.1	Architecture MVC	8
3.2	Diagrammes de classes logicielles	9
3.3	Diagrammes de séquence détaillés	10
3.3.1	Inscription	10
3.3.2	Connexion et création de partie	10
3.3.3	Vote	11
4	Manuel utilisateur	12
4.1	Utilisateurs visés	12
4.2	Lancement de l'application	12
4.2.1	Scallingo	12
4.2.2	Version locale	12
4.3	Interface utilisateur	13
4.3.1	Arrivée dans l'interface	13
4.3.2	Page d'accueil	13
4.3.3	Page d'attente	13
4.3.4	Page de jeu	13
4.3.4.1	Page d'informations	14
4.3.4.2	Page de discussion	14
4.3.4.3	Page de vote	14

4.3.4.4	Page des joueurs	14
4.4	Fonctionnalités	14
4.4.1	Inscription et connexion	14
4.4.2	Création d'une partie	15
4.4.3	Joindre une partie	15
4.4.4	Fonctionnalités en jeu	15
4.4.4.1	Utiliser son pouvoir	15
4.4.4.2	Choisir un salon de discussion et discuter	16
4.4.4.3	Voter	16
4.5	Faiblesses de l'application	16
4.5.1	Choix des dates	16
4.5.2	Retour en arrière	16
4.5.3	Base de données	16
5	Bilan du projet	17
5.1	Gestion de projet	17
5.1.1	Analyse et conception	17
5.1.2	Développement	17
5.1.2.1	Frontend	17
5.1.2.2	Backend	17
5.1.3	Mise en place des tests	18
5.2	Résultats obtenus	18
5.3	Bilans personnels	18
5.3.1	Cédric Pauly	18
5.3.2	Jorge Luri Vaño	19
5.3.3	Logan Willem	19
5.3.4	Maud Bergonzoli	19

1 Introduction

Ce document a pour but de présenter les différents diagrammes d'analyse et de conception réalisés pour exprimer l'organisation du projet commun WEB/ACOL.

Il comportera également le manuel utilisateur ainsi qu'un bilan du projet.

Dans la suite de ce document, on distinguera conception du projet, conception logicielle et du développement. Plus précisément, le mot conception (de projet) fera référence à l'ensemble des éléments suivants :

- L'analyse du projet en aval,
- La conception logicielle du projet via les diagrammes,
- Le développement documenté de l'application,
- La mise en place de tests,
- La gestion d'un GitLab mettant en place du CI/CD.

2 Analyse du projet

La section analyse est relative au travail en amont et portera sur les premières réflexions du projet et sa compréhension.

On précisera lorsque ce sera nécessaire, les libertés prises au vu du cahier des charges proposé et expliciterons certains aspects si besoin.

2.1 Description des acteurs

Nous allons décrire ici la liste des acteurs qui rentrent en jeu dans l'application et seront visibles sur le diagramme qui suit.

- **Temps** : Le temps est une composante essentielle de l'application car il va permettre de lancer 3 événements majeurs : lancer ou annuler une partie lorsque la date limite pour une création est atteinte et changer entre le jour et la nuit pendant une partie.
- **Utilisateur** : On définira par **Utilisateur**, n'importe quelle personne utilisant l'application possédant un compte avec un login et un mot de passe. On supposera donc que pour chacune de ses actions, elle se sera connectée et inscrite en amont.
- **Joueur** : Un **Joueur** est simplement un **Utilisateur** auquel on a associé une partie en cours.
- **Mort** : Il s'agit d'un **Joueur** qui est mort. Il a donc accès en lecture à l'intégralité de la partie mais ne peut discuter qu'avec le spiritiste si celui-ci l'a choisi pendant la nuit.
- **Vivant** : C'est un **Joueur** vivant qui peut donc participer à la discussion ainsi qu'aux votes se déroulant sur la place du village. Il peut également utiliser son pouvoir de voyance ou de spiritisme la nuit.
- **Humain** : C'est un **Vivant** qui n'est pas loup-garou. Bien qu'il n'ait pas de caractéristiques supplémentaires que le **Vivant**, on le laissera dans les schémas pour permettre une meilleure compréhension.
- **Insomniaque** : Il s'agit d'un **Humain** qui peut lire la discussion du repère des loups-garous.
- **Loup-garou** : C'est un **Vivant** qui peut de plus voter de nuit et écrire dans le salon du repère.
- **Contamination** : Il s'agit d'un **Loup-garou** qui a en plus le pouvoir de contamination.

Puisque la voyance et le spiritisme sont des pouvoirs partagés par des loup-garous et des humains, ils seront accessibles par le biais du vivant. Cela aurait sinon nécessité un duplicata pour chaque combinaison rôle/pouvoir possible.

2.2 Cas d'utilisation

Le diagramme est plutôt explicite au vu des acteurs décrits précédemment. On rajoutera simplement que pour améliorer la lisibilité, de gros blocs de type "Vote" ont été rajoutés pour montrer qu'il s'agissait

d'actions liées entre elles, mais ne changent pas le sens du diagramme.

De plus, pour chaque cas d'utilisation des acteurs héritant de **Utilisateur**, une flèche "include" devrait aller vers la connexion. Par soucis de lisibilité, elles auront été retirées.

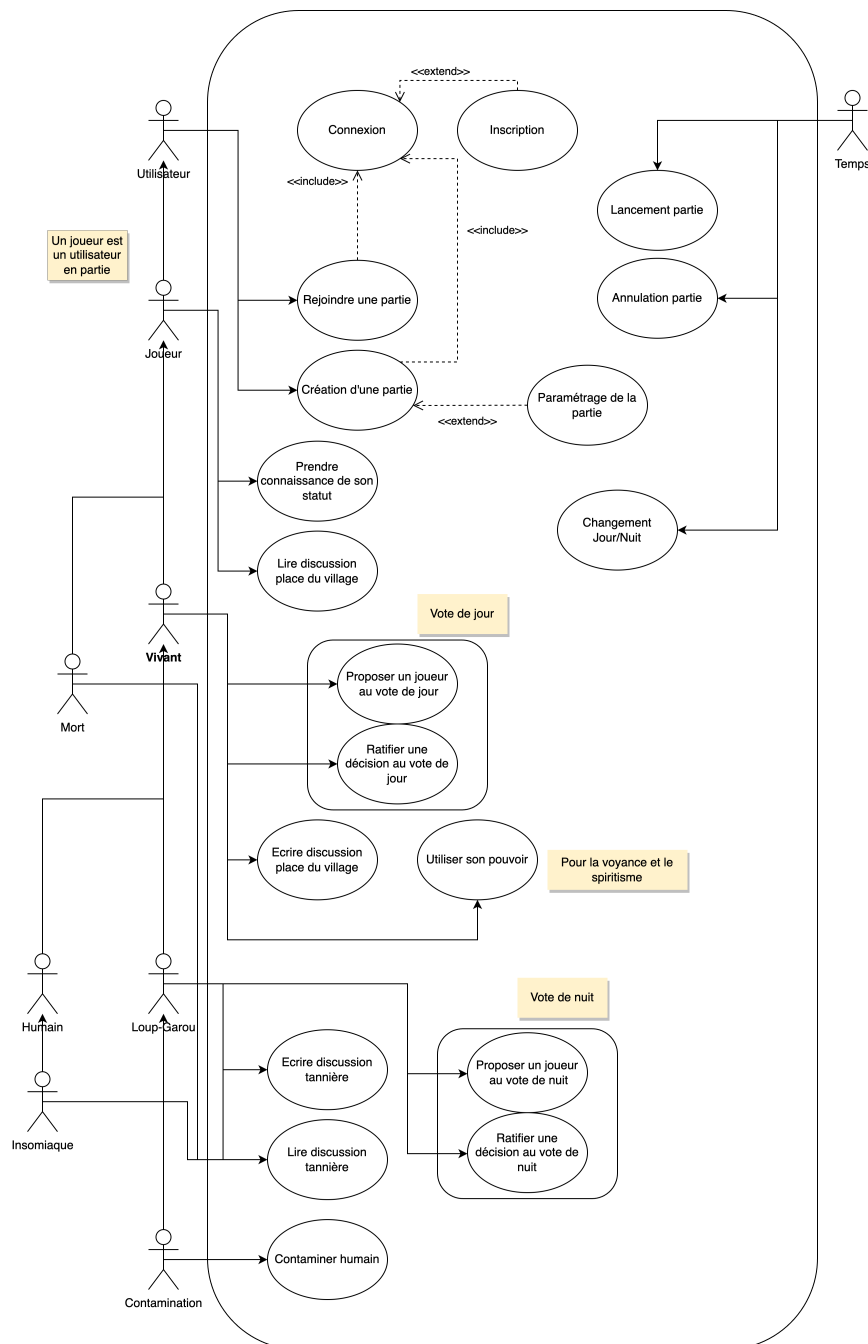


FIGURE 1 – Diagramme de cas d'utilisation

2.3 Diagrammes de séquence

Nous allons décrire dans cette section quatre cas d'utilisation les plus importants au bon fonctionnement de l'application.

D'autres cas auraient pu être ajoutés mais nous considérons suffisamment explicites, intuitifs ou redondants pour ne pas les avoir traduits sous forme de diagramme de séquence.

2.3.1 Inscription avec cas d'échec

Nous avons ici choisi de montrer comment allait se dérouler une inscription pour montrer quelles vérifications étaient faites pendant le processus.

On supposera que l'utilisateur a ouvert l'application et est dans la page d'inscription.

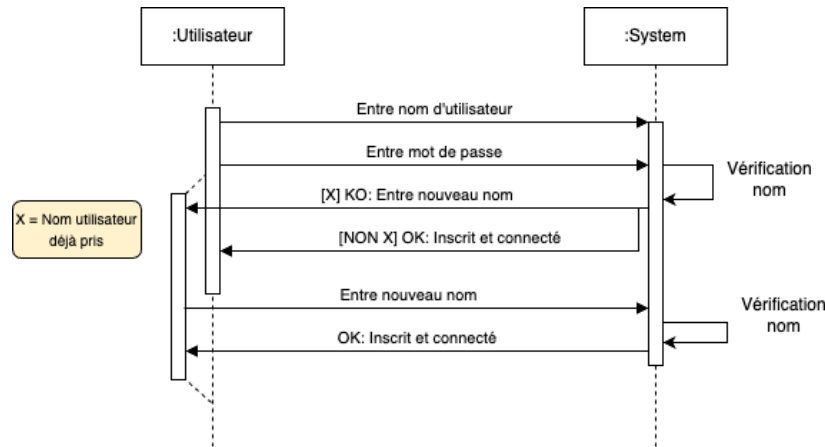


FIGURE 2 – Diagramme de séquence - inscription

Une fois l'utilisateur inscrit, il est donc présent dans la base de donnée et directement connecté.

2.3.2 Connexion avec cas d'échec

Pour les mêmes raisons que précédemment, nous allons montrer comment se passe la connexion. On supposera que l'application est ouverte sur la page de connexion et que l'on a été inscrit en amont.

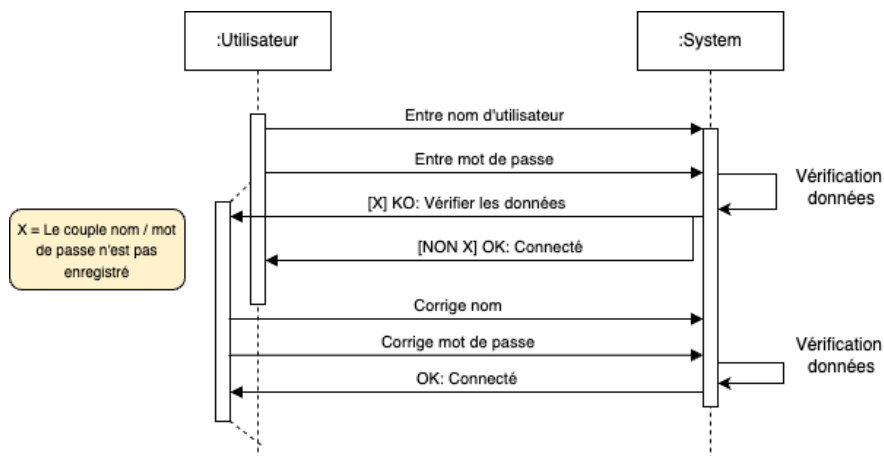


FIGURE 3 – Diagramme de séquence - connexion

A l'issue de la connexion, il arrive donc dans la page d'accueil qui est destiné à l'utilisateur si les données ont été effectivement vérifiées (i.e. que la paire login/mot de passe existe).

2.3.3 Création d'une partie

Voyons ensuite comment se passe la création d'une partie et son lancement en fonction du nombre de personnes qui vont la rejoindre.

On suppose donc que l'utilisateur est connecté sur l'application et a choisi de créer une partie (plutôt que d'en rejoindre une).

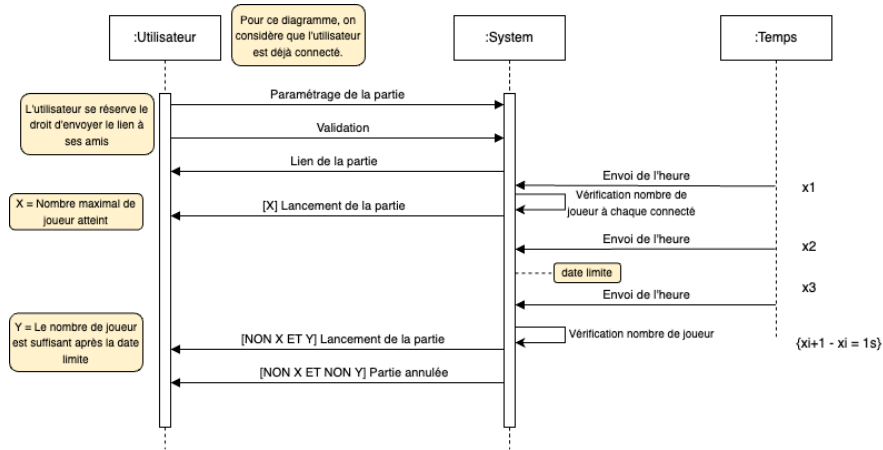


FIGURE 4 – Diagramme de séquence - création de partie

Selon les limites demandées et le nombre de joueur atteint, on est donc ou bien lancé dans une partie ou bien la partie est annulée et on retourne à l'écran d'accueil.

2.3.4 Vote

Enfin, le vote présenté ici se déroulera de la même manière de jour comme de nuit.

On supposera cette fois que les joueurs sont dans une partie, en capacité de voter et sur la page de vote.

On ne montre ici que deux joueurs pour montrer le système de proposition et de ratification.

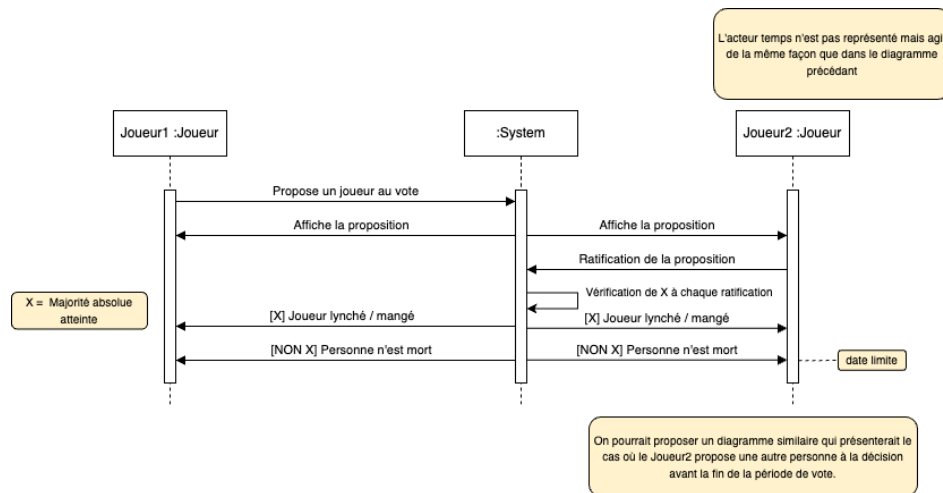


FIGURE 5 – Diagramme de séquence - vote

Selon l'issue du vote, on a donc ou bien quelqu'un qui est mort ou bien personne ne l'est, selon si la majorité absolue a été atteinte avant la date limite.

2.4 Diagramme de classe d'analyse

Le diagramme d'analyse de classe est le suivant :

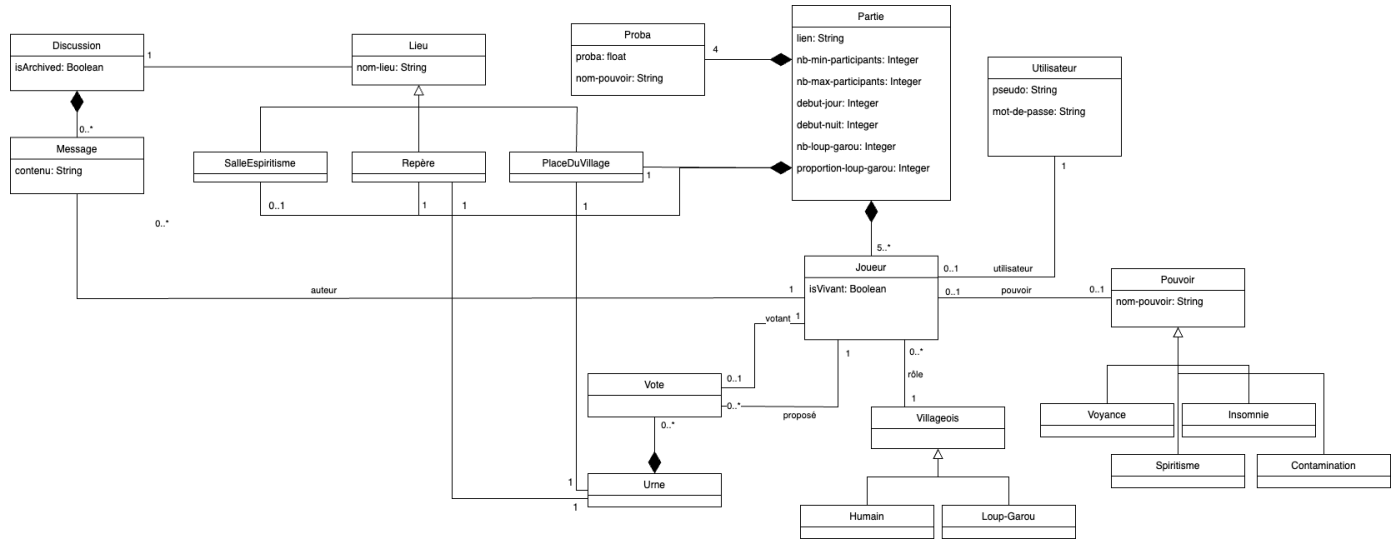


FIGURE 6 – Diagramme d'analyse de classe

On notera que les probabilités des pouvoirs ne sont pas directement liées aux pouvoirs : cela vient du fait que l'on souhaite garder l'accès à ces probabilités au long du jeu, qu'elles aient été associées à un joueur ou pas.

D'autre part, on notera que pour faciliter la gestion de l'application et de la base de données, un utilisateur peut au plus n'avoir qu'une seule partie en cours ou en attente de lancement.

2.5 Diagramme d'états-transitions

Enfin, on présente ici le diagramme d'état transition pour l'utilisation "globale" de l'application. Pour davantage de détails, on se référera au manuel utilisateur, présent plus loin.

Bien qu'assez intuitif, il permet de montrer l'utilisation générale de l'application.

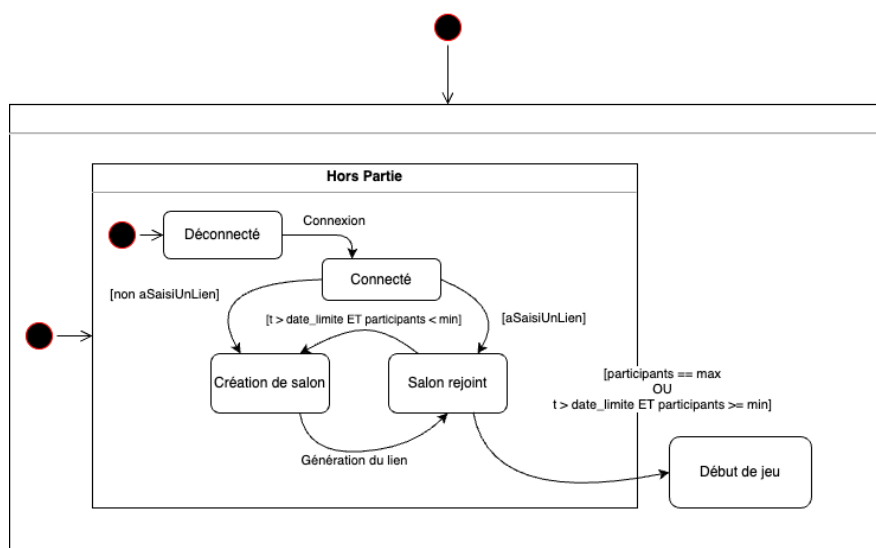


FIGURE 7 – Diagramme d'état/transition

3 Conception du projet

La conception aura été complétée pendant et à l'issue du projet. Evoluant au fil de la redéfinition du cahier des charges. On présentera ici le rendu final.

3.1 Architecture MVC

Nous avons choisi de garder ce diagramme le plus simple possible pour comprendre son fonctionnement au mieux.

Nous avons pensé aux différentes vues en réfléchissant aux utilisations de manière chronologiques : on s'inscrit, puis on se connecte, puis on crée ou rejoint une partie ...

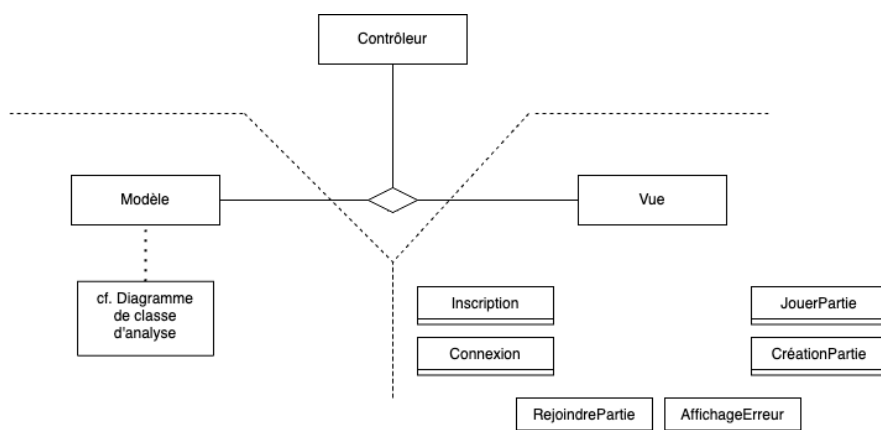


FIGURE 8 – Diagramme MVC

3.2 Diagrammes de classes logicielles

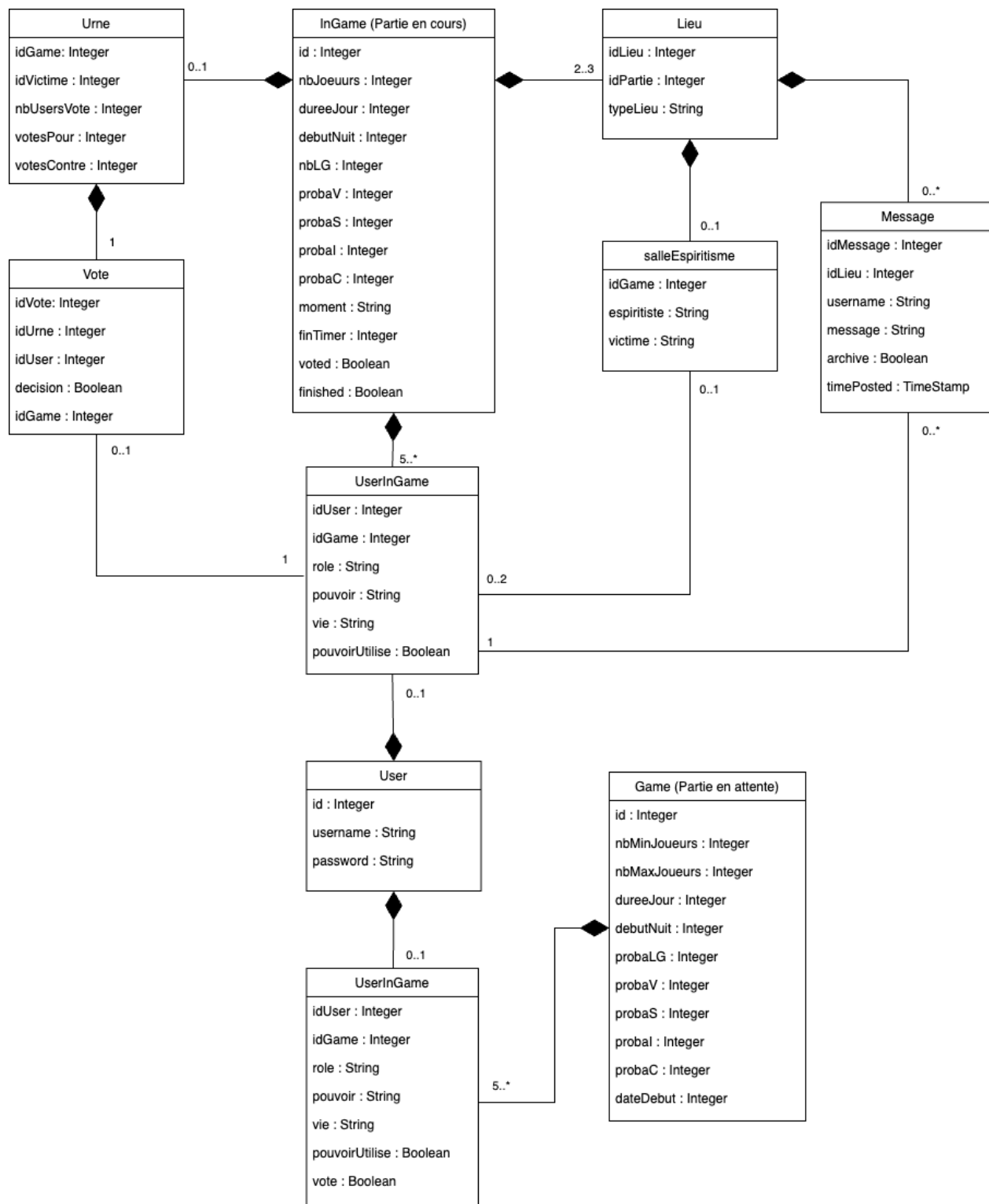


FIGURE 9 – Diagramme de classe logicielle

3.3 Diagrammes de séquence détaillés

Nous avons choisis de refaire les diagrammes précédents mais de manière plus détaillée avec les appels aux fonctions effectivement implémentées.

Le jaune correspondra aux fonctions utilisées par le front-end et pour le rouge, les fonctions du back-end. Le détail des sous-fonctions ne sera pas développé ici pour des raisons de lisibilité.

3.3.1 Inscription

Par comparaison au diagramme précédent, nous avons ici précisé quelles fonctions seraient appelées. On remarque toutefois que la structure du diagramme est identique.

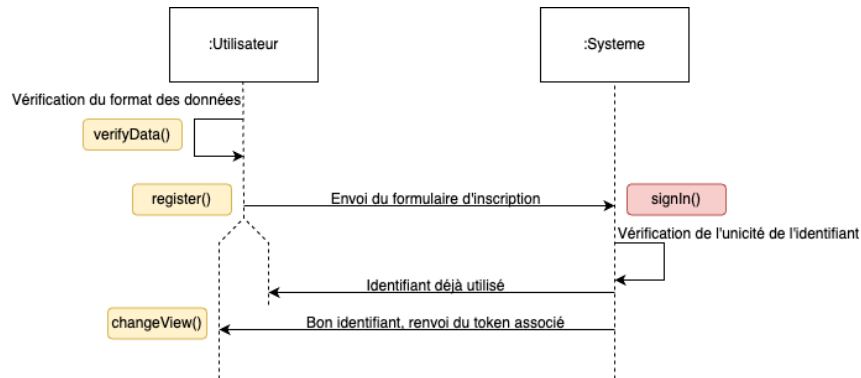


FIGURE 10 – Diagramme de séquence - inscription

3.3.2 Connexion et création de partie

Ici se déroule la séquence de connexion en puis de création d'une partie. Comme pour l'inscription, la structure est sensiblement similaire à celles proposées dans l'analyse.

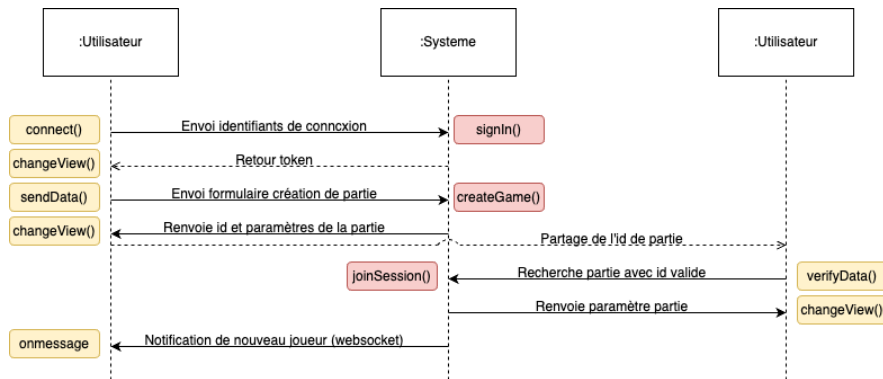


FIGURE 11 – Diagramme de séquence - connexion et création de partie

3.3.3 Vote

pour ce diagramme, nous présenterons les trois vues possibles pendant un vote.

Le `setVoteState(2)` permet d'afficher une vue où l'on n'a pas accès à la vue des votes actuels (si on est un humain et qu'on souhaiterait voir le vote des loups-garous par exemple).

Le cas `setVoteState(1)` permet de voir les votes mais pas de pouvoir ratifier ou proposer une décision.

Quant à `setVoteState(0)` c'est la vue que permet de ratifier ou faire une proposition car on n'a pas encore voté alors qu'on est en droit de le faire.

Dans notre diagramme, on a choisi ici de faire une nouvelle proposition.

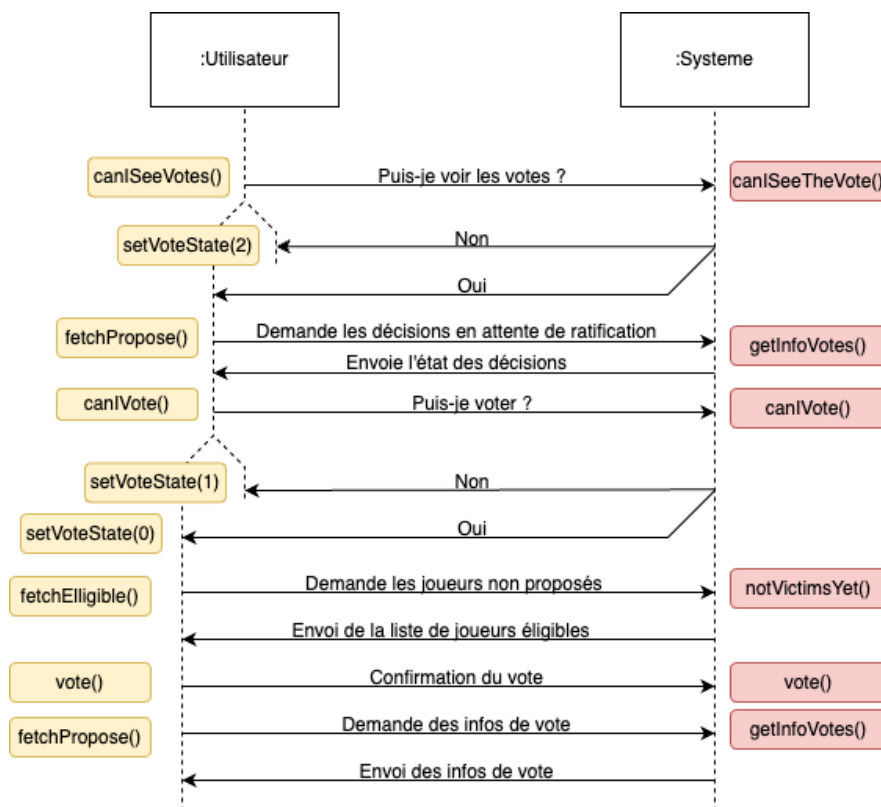


FIGURE 12 – Diagramme de séquence - vote

4 Manuel utilisateur

L'application est une variante mobile et en ligne du jeu de Loup-Garou.

L'action se déroule dans un village dont certains habitants sont humains et d'autres des loups-garous. Chaque nuit, les loups-garous se transforment et tuent un villageois. Pour se défendre, chaque jour, les villageois éliminent l'un d'entre eux. Le but de chaque faction est d'éliminer l'autre, mais les humains ne savent pas qui sont les loups-garous avant la fin de la partie.

Au début d'une partie, chaque joueur se voit attribuer un rôle : loup-garou ou humain. Ces rôles sont cachés aux autres joueurs. Les loups-garous et les humains sont collectivement appelés « villageois » (en effet les loups-garous sont aussi des habitants du village).

Lorsque le villageois d'un joueur est tué, ce joueur est éliminé. Les joueurs éliminés ne peuvent plus intervenir mais ont accès en lecture à toutes les discussions du jeu (sur la place du village comme dans le repaire des loups-garous), y compris les discussions archivées.

Le jeu fonctionne différemment le jour et la nuit. Par défaut le jour dure 14h et la nuit de 10h. La durée des journées est configurable lors de la création d'une session.

4.1 Utilisateurs visés

Cette application est un jeu mobile. Elle fonctionne sur Android mais que partiellement sur iOS. De ce fait, l'application est destinée à un public de joueurs mobile. Elle convient à des joueurs de loup-garou passionnés, voulant trouver une version du jeu en ligne gratuite et accessible. Elle convient plus généralement aux amateurs de jeux en ligne mais aussi à tout adolescents et adultes aptes à jouer à des jeux de stratégie et souhaitant passer du bon temps avec ses proches.

4.2 Lancement de l'application

Notre application a été testée via Expo Go. Quelle que soit la base de données utilisée, il faudra lancer depuis un ordinateur les commandes suivantes :

1. `cd frontend`
2. `npm i`
3. `npx expo start --tunnel`

Il suffira ensuite d'installer et de lancer l'application Expo Go sur son téléphone puis de scanner le QR-Code apparaissant sur le terminal.

La base de donnée associée peut être stockée, ou bien à distance avec Scallingo, ou bien en local. Cela peut être modifié en commentant ou bien les deux premières lignes ou bien les deux dernières du fichier `frontend/src/constants/backend.js`.

4.2.1 Scallingo

La base de données était initialement pensée pour être via Scallingo. Cependant, les quatre membres ayant épuisé leur essai, il ne sera pas fonctionnel.

Cependant, si on souhaite on pourra joindre son propre compte et remplacer avec les adresses qui conviennent.

4.2.2 Version locale

Dans le cas où la version avec Scallingo ne serait pas fonctionnelle, il est possible de gérer la base de données localement. Avant de lancer les commandes précédentes, on effectuera d'abord les manipulation qui suivent.

Il faudra dans un premier temps commenter les deux premières lignes et décommenter les deux suivantes. Ensuite on modifiera les deux lignes en remplaçant `localhost` par l'adresse IP de l'ordinateur.

Il faudra ensuite ouvrir un terminal et lancer les commandes suivantes :

1. `cd backend/`
2. `npm i`
3. `npm run updatedb`
4. `npm run startdev`

On ouvrira ensuite un nouveau terminal pour lancer les commandes précédentes.

4.3 Interface utilisateur

Cette application est naturellement organisée en plusieurs vues qui permettent chacune des actions bien distinctes listées dans la section Fonctionnalités.

4.3.1 Arrivée dans l'interface

Lors de notre arrivée dans l'application, on arrive dans la vue de connexion où l'on peut rentrer son login et mot de passe afin de se connecter.

Il est également possible depuis cette page de s'inscrire via le lien en bas de la page. L'inscription va alors demander de choisir un identifiant non utilisé ainsi que de rentrer deux fois le mot de passe.

On rappelle que chaque utilisateur a au plus une partie en cours, cela signifie qu'un joueur peut-être dans trois états :

1. Dans aucune partie en cours, il peut alors créer ou rejoindre une partie
2. Dans l'attente d'une partie
3. Dans une partie en cours

Ainsi, une fois l'inscription validée (ou la connexion faite) on est alors envoyé à l'accueil, en salle d'attente ou directement dans notre partie en cours selon si on est déjà associé à une partie en cours de déroulement (on ne pourra donc pas re-accéder à une partie archivée une fois déconnecté de cette partie).

4.3.2 Page d'accueil

Cette page propose d'aller soit dans la création de partie, soit d'en rejoindre une déjà créée.

Que ce soit parce que l'on a créé une partie ou parce que l'on l'a rejointe, on est envoyé en salle d'attente si elle n'est pas encore lancée, sinon directement dans le jeu si jamais on atteint le nombre maximal de joueurs.

4.3.3 Page d'attente

La page d'attente permet de consulter les paramètres du jeu en cours de création (durée des journées, probabilités ...) ainsi que de voir la liste des participants qui s'actualisera en temps réel.

4.3.4 Page de jeu

Lorsque l'on arrivera dans une partie, on pourra naviguer entre quatre écrans ayant chacun leur spécificités propre. Ils sont accessibles en swipant horizontalement sur l'écran.

4.3.4.1 Page d'informations

La page d'information est un récapitulatif du joueur et de la partie en cours.

On peut notamment y voir son rôle, son pouvoir et s'il est vivant ou mort. C'est depuis cet écran que le joueur pourra utiliser son pouvoir s'il en possède un.

On a également accès aux informations de la partie dont le temps actuel (jour ou nuit) ainsi que le temps restant avant de changer d'état.

4.3.4.2 Page de discussion

La page de discussion regroupe l'ensemble des discussions actuellement accessibles par le joueur. On notera que l'accès en question est la lecture des discussions, l'envoi de message sera possible si la barre d'entrée de message est apparente.

Si le joueur entre dans une discussion, il pourra retourner à l'écran de choix en utilisant le retour en arrière du téléphone.

Les messages seront actualisés en temps réel sur le téléphone.

4.3.4.3 Page de vote

La page de vote permettra de voter de jour comme de nuit, que ce soit en temps qu'humain ou loup-garou. Les morts et l'insomniaque pourront observer le déroulé du vote pendant la nuit (sans pouvoir toutefois intervenir).

Du moment où l'on a accès au visuel des votes, on peut uniquement voir quel joueur a été proposé, et combien de voies il possède pour l'instant. Il ne sera pas possible de savoir qui a voté pour qui.

4.3.4.4 Page des joueurs

La page des joueurs récapitule les pseudos des joueurs dans la partie, morts (de couleur rouge) comme vivants (de couleur verte). Si l'on est loup-garou, les autres loup-garous vivants seront affichés en orange.

On notera que depuis cette page, ainsi que de toutes les autres, on n'indique jamais les rôles et pouvoirs des joueurs morts. On pourra simplement faire des déductions avec le nombre de villageois et de loups-garous restant (visibles sur la page d'information).

4.4 Fonctionnalités

Seront ici décrites les fonctionnalités de l'application et comment les utiliser plus précisément.

4.4.1 Inscription et connexion

Le fonctionnement pour s'inscrire et se connecter est assez simple : on entre un login ainsi qu'un mot de passe.

Puisqu'il s'agit d'une entrée de texte, le login devra suivre les contraintes suivantes pour éviter les injections de code :

- Il doit être uniquement composé de lettres (minuscules ou majuscules) ou des chiffres.
- Il doit être d'une longueur de 5 à 16 caractères.

Pour le mot de passe, la seule contrainte est : qu'il doit être de longueur entre 8 et 32 caractères.

4.4.2 Création d'une partie

Le cahier des charges étant peu précis, nous avons choisi de sélectionner une durée (en heures et en minutes) plutôt que des horaires pour la partie pour ne pas avoir la contrainte implicite qu'un cycle dure 24h.

Pour créer une partie, tous les paramètres proposés par défaut sont modifiables dans la limite de certaines contraintes :

- Les durées en minutes doivent être entre 0 et 60.
- Une partie a au moins de 5 personnes.
- Une partie a au plus 100 personnes.
- Les probabilités doivent être comprises entre 0 et 1, en utilisant des "." pour noter les décimales.
- Le nombre minimal de joueur doit être inférieur ou égal au nombre maximal de joueur.
- La date de début doit être ultérieure à celle de l'envoi du formulaire.

Une fois le formulaire envoyé, le joueur est envoyé directement en salle d'attente où il aura accès aux vues décrites précédemment. Il aura notamment en bas de la page le numéro associé à la partie qu'il pourra partager à ses amis.

4.4.3 Joindre une partie

Pour rejoindre une partie déjà créée par un autre joueur, il suffit simplement de cliquer sur "Rejoindre une partie" puis d'entrer le code à 6 chiffres communiqué dans la salle d'attente.

D'autre part en cliquant sur l'icône en bas à gauche, on pourra changer entre les deux vues de la salle d'attente. Les deux vues accessibles pendant ce temps sont : un affichage figé des paramètres choisis pour la partie avec le code associé, et la liste des joueurs ayant rejoint la salle d'attente qui s'actualise de manière dynamique.

4.4.4 Fonctionnalités en jeu

Une fois que l'on est dans une partie (que l'on s'est donc connecté dans l'application et qu'on a été associé à une partie qui est en cours), on accède donc aux nombreuses fonctionnalités du jeu.

On notera que le début de la partie commencera la nuit, quelles que soient les durées et la date de début rentrées.

4.4.4.1 Utiliser son pouvoir

Comme indiqué précédemment dans la section 4.3.4.1, la vue d'information permet d'accéder aux informations concernant l'état de la partie et du joueur.

Si le joueur possède un pouvoir, un bouton "Utiliser le pouvoir" sera affiché et cliquer dessus permettra de l'utiliser.

Plus précisément,

- Si le joueur est voyant, il aura alors accès, chaque nuit, à la liste des joueurs encore en vie et pourra sélectionner un joueur dont il souhaite connaître les informations. On affichera alors le rôle et pouvoir de la cible.
- Si le joueur est contaminateur, il aura accès chaque nuit à la liste des humains qu'il peut contaminer. Il pourra alors en sélectionner un par nuit pour le contaminer.
- Si le joueur est spiritiste, il aura accès, chaque nuit, au choix de avec quel mort il souhaite discuter.

Dans le cas de l'insomniaque, il n'aura pas de bouton "Utiliser le pouvoir" car il aura directement accès à la discussion du repère et pourra voir le déroulé des votes la nuit.

4.4.4.2 Choisir un salon de discussion et discuter

Pour choisir le salon de discussion, une fois dans la page de discussion, le joueur verra s'afficher une liste de boutons avec le nom des conversations auxquelles il peut accéder pour l'instant.

Une fois qu'il aura cliqué sur celle qui l'intéresse, il pourra alors faire défiler les derniers messages ainsi que éventuellement écrire et envoyer des messages en cliquant sur la barre d'entrée de texte, entrant son message qu'il pourra envoyer en cliquant sur l'icône d'envoi.

4.4.4.3 Voter

La page de vote permettra de voter de jour comme de nuit (pour la place du village comme le repère), son fonctionnement sera strictement identique, la seule différence sera sur les droits de qui peut faire quoi. Un joueur peut être dans l'un des trois cas suivants :

1. S'il n'a pas encore voté alors qu'il est en droit de le faire, il peut alors ratifier une décision en cliquant sur l'un des pseudos proposés. Le pseudo sera alors mis en gras et on pourra cliquer sur le bouton "Voter" pour valider la décision.
2. S'il n'a pas encore voté alors qu'il est en droit de le faire, il peut sinon proposer un autre joueur via "Proposer". Il aura alors accès à la liste des joueurs éligibles qui n'ont pas encore été proposés et pourra sélectionner celui de son choix. Son vote sera alors automatiquement pris en compte.
3. S'il a déjà voté ou s'il n'en a pas le droit (joueur mort ou insomniaque), il peut alors simplement voir la liste des joueurs proposés ainsi que le nombre de votes pour cette personne.

4.5 Faiblesses de l'application

Bien que nous avons testé au mieux notre application au fur et à mesure du projet, certaines faiblesses sont restées par manque de temps. Des corrections ultérieures pourront être faites pour améliorer les points suivants.

4.5.1 Choix des dates

Le principe de choix d'une date et d'une heure de début à l'aide d'un selecteur de date et d'heure n'est fonctionnel que sur Android. Malheureusement, le principe utilisé n'est pas compatible avec les appareils iOS.

4.5.2 Retour en arrière

La possibilité de retourner en arrière en fonction des vues n'est disponible que sur certains appareils Android, où le bouton de retour est affiché en bas de l'écran. De ce fait, les utilisateurs iOS ne disposant pas de ce bouton ne peuvent retourner en arrière de vues en vues.

4.5.3 Base de données

La session Scallingo ayant expiré peu avant la date de rendu, nous avons donc fini notre application en version locale. Ainsi l'application fonctionne actuellement en version locale. La version Scallingo ne prend pas en charge proprement les websockets définies.

5 Bilan du projet

Nous allons à présent faire un récapitulatif de ce projet sur son organisation et le ressenti que nous en avons eu.

5.1 Gestion de projet

Afin d'être le plus efficace possible, nous nous sommes réparti le travail de la manière suivante :

5.1.1 Analyse et conception

Dans un premier temps nous avons commencé par analyser le sujet au travers des diagrammes proposés. Pour cet aspect tous les membres du groupe y ont participé à leur échelle. Cela a permis d'avoir la meilleure compréhension possible du cahier des charges et de se mettre d'accord sur les éventuelles imprécisions vues.

La conception a quant à elle été alimentée au fur et à mesure pendant le développement du projet, pendant que l'on comprenait plus en détail le projet et précisions ainsi le cahier des charges.

5.1.2 Développement

Pour le développement en lui-même du projet, nous avons fait le choix de nous répartir en deux équipes de deux : une sur le frontend et une sur le backend. Ainsi le modèle d'architecture MVC était bien respecté. Les deux équipes communiquaient verbalement et mettaient sur un fichier partagé les routes demandées et créées. De plus il arrivait que des membres travaillent sur les deux aspects lorsque nécessaire.

5.1.2.1 Frontend

La subsection frontend était développée par Logan et Maud. Le développement s'est déroulé en deux temps : d'abord la mise en place des pages d'inscription, de connexion ainsi que la création et le partage des parties, puis la gestion interne des parties.

La mise en place de l'application a été faite de manière assez efficace : chaque membre travaillait sur les vues qu'il lui avait été attribuées en s'entraidant parfois. De plus, des composants ont été créés au fur et à mesure pour pouvoir les partager et factoriser le code.

Pour la seconde partie, puisque nous avons décidé de répartir l'affichage pendant la partie en 4 vues (Informations, Discussions, Votes, Joueurs), Logan s'est chargé de la vue Informations et Joueurs tandis que Maud s'est occupée de Discussions, Votes et l'usage des pouvoirs. Une fois les vues mises en place (aspect purement visuel et interactif), Logan s'est également chargé de mettre en place des websockets afin que l'actualisation des votes et messages se fasse en temps réel (plutôt que de rafraîchir manuellement ou de faire un fetch périodique).

5.1.2.2 Backend

La section Backend a été mise en place par Jorge et Cédric. Comme précédemment, le développement s'est fait en deux temps.

Au début, la grande difficulté a été celle de comprendre comment on allait structurer et développer le Backend. Par exemple, on savait comment coder avec un Backend déjà "initialisé" avec npm comme nous avons vu dans le TP Backend, mais nous ignorions comment le lancer. Nous avons donc appris à le faire entièrement de zéro.

Puis, la structure. Nous devons concevoir la structure qui régirait notre application par derrière. Nous avons donc passer du temps à parler pour comprendre tous les points du projet.

Une fois lancés, le développement était assez intuitif et direct. Durant les vacances, nous avons passé toute la semaine à développer le Backend, ce qui nous a donné un code entièrement fonctionnel à la fin de la semaine. Nous savions que nous n'aurions pas du temps pendant les semaines avant les partiels pour travailler la structure du Backend et que l'on aurait beaucoup de bugs à résoudre, nous avons alors préféré d'être en avance.

Durant le reste du temps, nous nous sommes séparés les tâches. Cédric est passé en conception de tests et Jorge est resté dans la maintenance du Backend pendant qu'il finissait la partie ACOL du projet. Nous sommes passés en fonctionnement sous forme de tickets : dès que le frontend détectait une erreur, un bug ou lui manquait des requêtes, nous réalisions les modifications nécessaires pour résoudre le ticket.

Ce mode de fonctionnement a été très efficace et opératif, car ça permettait de décharger le travail du frontend, être efficace dans le développement en divisant les tâches dès qu'une difficulté est survenue et d'avancer dans les autres aspects du projet. Toute cette organisation a aussi permis de pouvoir avancer dans les possibles erreurs du backend pendant que nous révisions nos partiels.

5.1.3 Mise en place des tests

Le projet étant testé au fur et à mesure à la main, les interfaces évoluant constamment, et le temps nous manquant, nous ne nous sommes penchés sur les tests que tard.

Dans le manque de temps, nous avons donc pris le parti de finir l'application et la rendre fonctionnelle, au détriment éventuel de l'automatisation des tests et du CI/CD, qui pourraient être ajoutés plus facilement par nous-même ou une autre équipe.

5.2 Résultats obtenus

Ainsi comme demandé par le cahier des charges, nous avons pu concevoir au sens large l'intégralité de l'application : l'analyse, la conception et le développement ainsi que la documentation ont été entièrement complétés.

Cependant, pour cause du manque de temps, les tests ont été peu nombreux et le CI/CD donc incomplet. On notera toutefois qu'il s'agit d'un projet conséquent qui malgré les difficultés rencontrées est fonctionnel et que les tests "à la main" passent, il ne faut plus que les automatiser, ce qui ne devrait pas prendre plus de quelques jours à un développeur pour avoir une batterie de tests complète.

5.3 Bilans personnels

Cette dernière section conclura notre ressenti personnel au vu de ce qui a été fait individuellement.

5.3.1 Cédric Pauly

Ce projet fut pour moi une très bonne expérience et un très bon moyen de consolider les notions vu en cours durant ce semestre. Sa difficulté assez élevée m'a aussi permis d'aller plus loin sur certains aspects de la construction d'applications web, notamment concernant le backend avec des notions plus développées et d'autres plus compliquées.

Contrairement à certains de mes camarades, la plus grosse difficulté que j'ai rencontré durant ce projet est la gestion de mon temps. J'ai sous estimé la charge de travail de ce projet et me suis fait rattraper par les

révisions des partiels. J'ai ainsi dû mettre en pause la gestion de ce projet. Si j'avais anticipé cette période de partiels, j'aurais pu terminer mon travail dans sa globalité. La partie test du backend et du frontend ne sont donc pas complètes par ma faute dans ce projet. C'est une leçon que je n'oublierai pas et que j'emporterai avec moi dans de futurs projets.

5.3.2 Jorge Luri Vaño

Pour ma part, je considère ce projet très formateur et je considère que j'ai beaucoup appris. Malheureusement, je n'ai pas pu participer activement dans la partie frontend, mais mon travail sur le backend m'a permis de très bien comprendre le fonctionnement en interne de l'application et de comprendre donc la structuration du développement d'applications.

Je considère, d'autre part, que la bonne organisation de mon temps a été un atout pour vivre positivement ce projet. En m'avançant sur la surcharge de travail que nous aurions postérieurement avec les partiels, j'avais prévu de m'avancer au maximum pendant les vacances afin de finir au plus possible le backend avant le début des révisions. Ainsi, même si j'ai pu passer beaucoup de temps dessus, j'ai pu travailler dans la continuité, ce qui a permis beaucoup moins de surcharge mentale.

Ainsi, j'ai un bilan très positif du projet, même si la taille reste très conséquente.

5.3.3 Logan Willem

J'ai tendance à penser que ce projet est celui qui m'a le plus plu à l'ENSIMAG. Non seulement il est ludique, dans le sens où on a l'envie de s'y mettre et de faire le plus possible, mais il permet également de toucher à beaucoup de points techniques ce qui est en fait un sujet assez large.

Que ce soit la partie frontend, en utilisant React, React Native ou encore l'API fetch, tout comme la partie backend avec l'ORM Sequelize, ou le framework Express, j'ai pu approfondir toutes ces notions dont on avait pour la plupart vues les bases en TP.

Le fait qu'il s'agisse d'un projet comprenant plus ou moins toutes les étapes d'un cycle de vie d'un projet, de l'analyse, à la conception, jusqu'à l'intégration en passant par le codage et le test est quelque chose qui a pu être découvert lors du Projet GL, mais c'est réellement grâce à ce projet que ces étapes ont pu être approfondies.

J'ai cependant deux regrets par rapport à ce projet. Le premier étant la négligence des tests. En effet, on n'a pas donné une grande importance aux tests lors de ce projet, mais cela s'explique selon moi en 2 raisons, la première étant qu'il s'agit d'un processus plus complexe et plus chronophage que les tests qu'on pouvait effectuer lors du projet GL par exemple. La seconde raison réside dans le peu d'expérience que l'on a avec Cypress et Jest. On a certes pu les voir lors de TP, mais pour ma part, j'avais l'impression de faire du bricolage pour réussir à faire des tests plutôt que réellement comprendre ce qu'il se passe, de ce fait je savais que j'allais devoir passer du temps à apprendre ces outils, mais je ne l'ai jamais pris... Mon second regret vient de mon incapacité à pouvoir travailler sur le projet de manière efficace juste avant et pendant les partiels. En effet, sans compter les partiels, j'ai dû chercher activement un stage, puis un logement ainsi que gérer mon départ à l'étranger l'année prochaine, ce qui a grandement impacté mon emploi du temps. Je reste cependant très satisfait du travail que l'on a fourni, bien qu'il ne soit pas testé ni même complété à 100%.

Au delà de tout ce qui a déjà été dit, je tenais à dire que grâce à ce projet, j'ai pu découvrir les joies du développement web, et que l'idée même de travailler dans ce domaine est une idée qui me plaît plutôt bien !

5.3.4 Maud Bergonzoli

Pour ma part, mon ressenti sur ce projet est globalement positif : j'ai beaucoup appris en développement web, plus particulièrement sur le frontend sur lequel j'ai travaillé majoritairement.

En effet, réaliser les différentes vues et composants du projet, m'ont permis d'acquérir une bonne compréhension de React Native, notamment sur les différents modules déjà disponibles mais aussi sur le fonctionnement des fonctions natives, comme `useEffect` par exemple.

Cependant, ce projet a été également frustrant par sa taille : bien trop conséquent au vu du temps à disposition. Malgré le repoussement des dates de rendu par deux fois, nous avons été obligé de mettre ce projet en pause le temps des révisions des partiels mais cela nous a juste demandé d'être encore plus investi sur le temps en amont et en aval de cette pause.

J'ai investi beaucoup de temps sur ce projet que ce soit pour son développement ou la rédaction des documents. Et je suis fière de ce que l'on a fait, j'aurais simplement souhaité avoir le temps de le finir proprement, sans avoir à laisser un pan entier de côté.

Je maintiens toutefois qu'il s'agit d'un très bon projet, très complet qui m'a beaucoup enseigné. Tout comme le projet GL, je pense qu'il est très formatteur et prometteur, sous réserve de le redimensionner pour les prochaines années.