

Construction d'applications web – Analyse, conception et validation de logiciels

L'objectif de ce travail est de réaliser l'analyse, la conception et l'implémentation d'une application Web correspondant au cahier des charges ci-dessous.

L'application doit être écrite en Java (J2EE) et doit être déployée sur un serveur Tomcat. Ceci doit être possible via maven, soit en ligne de commande soit en passant par netbeans. Elle doit respecter une architecture MVC. Les accès à la base de données seront effectués par l'API JDBC. L'application devra utiliser un pool de connexions.

1 À rendre

Vous devez déposer sur Teide, au plus tard le 11 avril, une archive `.tar.gz` contenant :

1. un unique fichier PDF `rapport.pdf` comportant 4 parties :
 - (a) Document d'analyse :
 - acteurs, diagramme de cas d'utilisations et description de ces cas d'utilisations, illustrées par des diagrammes de séquence système pertinents.
 - diagramme de classes d'analyse.
 - (b) Document de conception :
 - L'architecture générale modèle-vue-contrôleur est imposée, mais indiquez comment vous la mettez en œuvre : quels sont les contrôleurs et les vues, comment tout s'articule-t-il.
 - Conception détaillée : diagramme de classes logicielles, diagrammes de séquence, diagrammes d'états-transitions si cela est pertinent. Ces diagrammes doivent être cohérents entre eux et avec l'implémentation. Ils doivent être expliqués et commentés. Évitez d'inclure des diagrammes illisibles ou qui n'apportent aucune information intéressante.
 - (c) Manuel utilisateur.
 - (d) Bilan sur les outils de modélisation utilisés, en particulier les problèmes rencontrés, ainsi que les solutions trouvées. Il vous est demandé dans cette partie de bien préciser les logiciels, en particulier les modeleurs UML que vous avez utilisés.
2. un fichier `install_bd.sql` contenant une série de requêtes pour créer la base de données dont l'application a besoin et éventuellement la pré-peupler avec des données pour faciliter les tests. Il est recommandé (mais pas obligatoire) de laisser **en plus** une base de données opérationnelle sur ensioracle1 en indiquant les identifiants de connexion, en cas de problème avec le script.
3. un fichier `readme.txt` indiquant comment déployer votre application pour qu'elle fonctionne. En principe les manipulations doivent se limiter à modifier des noms d'utilisateurs et des mots de passe dans des fichiers de configuration à préciser, exécuter le script `install_bd.sql` puis compiler et déployer via maven ou netbeans.
4. un dossier contenant le projet maven proprement dit, de préférence sans les fichiers compilés.

2 Cahier des charges

On souhaite réaliser une application permettant de jouer en ligne à une variante du jeu de Loup-Garou.

Principe du jeu : 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 de 8h00 à 22h00 et la nuit de 22h00 à 8h00. Ces horaires sont configurables avant le lancement d'une partie.

Le jeu comporte un certain nombre de *votes* visant à désigner un villageois qui sera tué. Les joueurs peuvent s'organiser comme ils l'entendent dans la salle de discussion appropriée pour définir des modes de scrutin, on ne demande pas au logiciel de gérer cet aspect. Le logiciel doit en revanche permettre d'enregistrer le résultat de la décision collective de la façon suivante :

1. L'un des participants au vote propose un joueur à la décision (cela peut être lui-même).
2. Les autres participants doivent alors ratifier explicitement cette décision.
3. Si la majorité absolue des participants (en comptant le joueur ayant fait la proposition) a ratifié une décision, celle-ci est immédiatement validée et il n'est plus possible d'en changer.
4. Il est possible de proposer une décision même s'il y a déjà une autre décision en attente de ratification (mais pas si une autre décision a déjà été validée).
5. Si aucune décision n'a été ratifiée à la fin de la période (jour ou nuit), personne n'est tué.

Déroulement de la partie : Le jour, tous les villageois encore vivants peuvent discuter librement sur la place du village. Celle-ci comprend une liste chronologique de messages depuis le matin. Chaque joueur peut lire tous ces messages et, s'il est encore vivant, poster un nouveau message qui s'ajoutera à la liste. Les messages identifient clairement leur auteur : il n'est possible ni de poster de façon anonyme ni d'usurper une identité.

Les villageois peuvent décider de lyncher l'un des leurs, accusé d'être un loup-garou, chaque jour, selon la procédure décrite dans le paragraphe « votes ». Ce villageois est alors tué.

Lorsque la nuit tombe, les discussions du jour sont archivées et ne sont plus visibles par les joueurs encore en jeu.

Pendant la nuit, les loups-garous se réunissent dans leur repaire et peuvent décider de dévorer l'un des villageois, à nouveau selon la procédure du paragraphe « votes ». Ce villageois est alors tué. Le repaire des loups-garous fonctionne comme la place du village, mais seuls les loups-garous y ont normalement accès (voir « pouvoirs spéciaux » pour les exceptions).

Le matin, les discussions de la nuit sont archivées et ne sont plus visibles par les joueurs encore en jeu. Le logiciel indique sur la place du village qui a été retrouvé mort...

Fin de la partie : Une partie se termine dès qu'il ne reste plus en jeu que des loups-garous ou que des humains. Une fois la partie terminée, les archives des discussions sont rendues visibles à tous.

Pouvoirs spéciaux : On souhaite pouvoir configurer des parties où certains joueurs seront dotés de pouvoirs spéciaux. Ces pouvoirs spéciaux sont :

- 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 pouvoirs sont secrets, c'est-à-dire que le logiciel ne doit jamais indiquer à un joueur quels pouvoirs ont les autres (les joueurs peuvent en revanche en parler et il est par exemple envisageable pour un loup-garou de prouver aux autres qu'il dispose du pouvoir de contamination en utilisant ce pouvoir).

Inscription des utilisateurs : Il suffit pour s'inscrire sur l'application de spécifier un pseudonyme et un mot de passe. Plusieurs utilisateurs ne peuvent pas avoir le même pseudonyme.

Configuration d'une partie : Tout utilisateur peut créer une partie. Il doit indiquer :

- 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).

Tous ces paramètres sont publics. Chaque utilisateur peut alors s'inscrire à cette partie, jusqu'à l'horaire de début ou jusqu'à ce que le nombre maximum de participants soit atteint. Si, à l'horaire de début prévu, le nombre minimal de participants n'est pas atteint, la partie est annulée. Sinon, les rôles et les pouvoirs sont tirés au sort et la partie commence.

Le nombre de loups-garous à créer est égal à la proportion indiquée multipliée par le nombre de joueurs et arrondie au supérieur. Cas particulier : le minimum est de 1 même si la proportion indiquée est 0. Pour les pouvoirs, on décide d'abord si le pouvoir doit exister, conformément à la probabilité demandée, puis aléatoirement à qui l'attribuer parmi les joueurs n'ayant pas déjà un pouvoir.

Le nombre initial de loups-garous est indiqué aux joueurs, mais pas l'existence ou non des pouvoirs : ils connaissent uniquement la probabilité initiale que ces pouvoirs existent.

La liste des joueurs est affichée sur la place du village. Chaque joueur peut prendre connaissance de son rôle et de ses pouvoirs. La partie se déroule ensuite comme indiqué dans le paragraphe « déroulement ».