

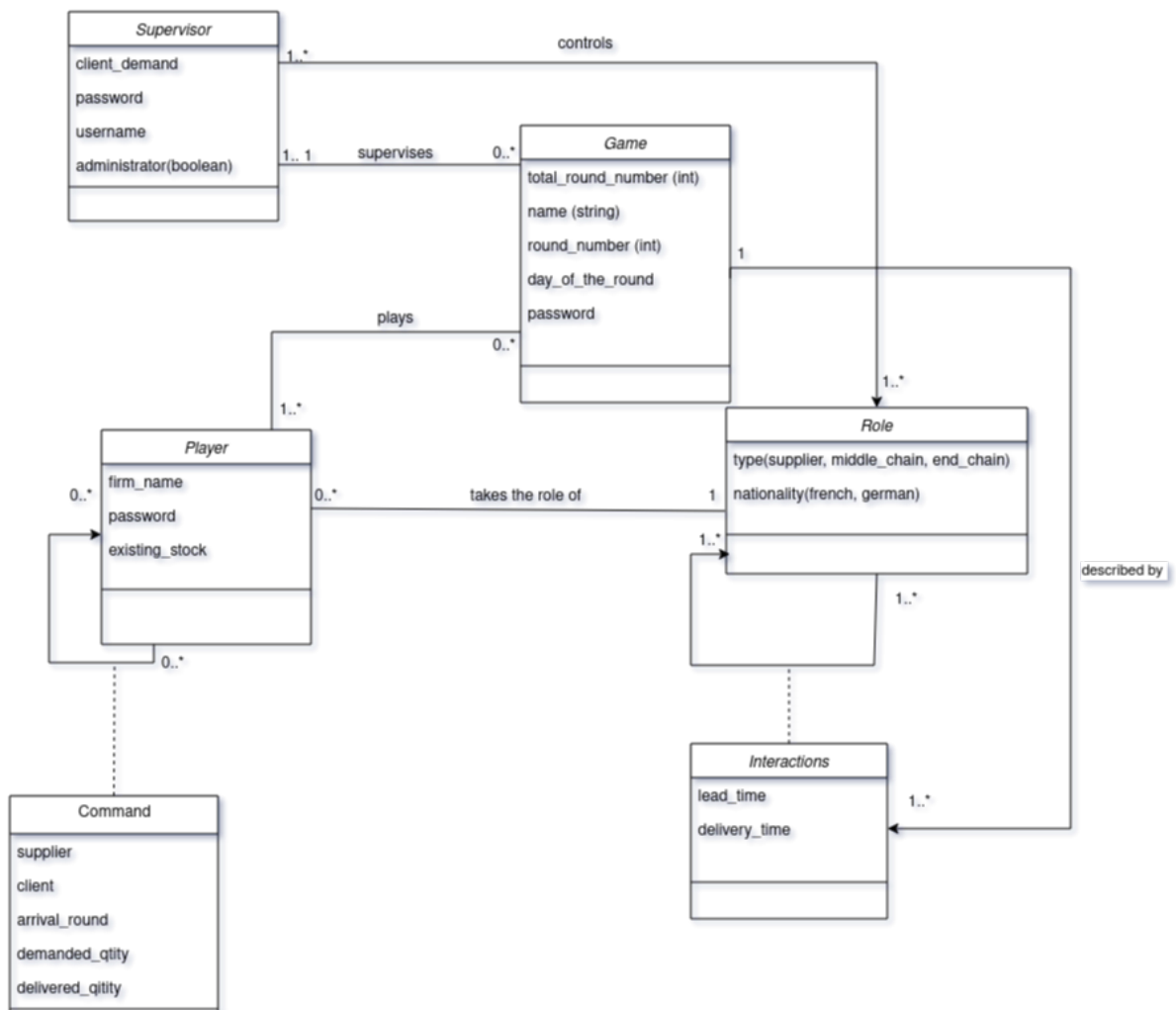
# Application BEER GAME

## - *Spécifications Techniques* -

<b>Le Modèle Conceptuel de Données :</b>	<b>2</b>
Le Modèle	2
Dictionnaire de données	3
<b>Les Vues</b>	<b>5</b>
Accueil :	5
Création d'un compte :	5
Connexion :	5
Création de partie :	6
Rejoindre une partie :	6
Salon d'une partie vue par le superviseur :	6
Paramétrage de l'état initial de la chaîne logistique :	7
Salon d'une partie vue par un joueur :	7
Tableau de bord d'une partie vue par le superviseur :	7
Tableau de bord d'une partie vue par un joueur :	7
Affichage de la fiche de suivi :	8
Vue Règles du jeu :	8
Graphiques de fin de partie :	8
<b>Les Routes :</b>	<b>9</b>
Accueil :	9
Création d'un compte :	9
Connexion :	9
Rejoindre une partie :	10
Création de partie :	11
Salon d'une partie vue par le créateur de la partie:	12
Salon d'une partie vue par un joueur de la partie :	12
LogOut :	13
Paramétrage Chaîne Logistique :	13
Tableau de bord d'une partie vue par le superviseur :	13
Tableau de bord d'une partie vue par un joueur :	14
Terminer la partie :	15
Affichage de la fiche de suivi :	15
Graphes de fin de partie :	16
<b>Estimation charges</b>	<b>17</b>
Les vues :	17
Les routes :	18
Total :	19

# Le Modèle Conceptuel de Données :

## 1. Le Modèle



## 2. Dictionnaire de données

**Player =** Entité associée à un joueur et créée lors de la création d'une partie

**firm\_name =** Nom de l'entreprise représentant un maillon de la chaîne logistique et choisi par le joueur

**password =** Chaîne de caractères confidentielle choisi par le joueur lors de la connexion qui permet de sécuriser son compte

**existing\_stock =** Le stock par round (fonction de temps)

**Game =** Entité associée à une partie et instanciée lors de la création d'une partie

**total\_round\_number =** Numéro total de périodes qui composent la partie et qui est choisi par le superviseur avant la création de la partie dans les paramètres initiaux

**name (string) =** Nom d'une partie (unique afin d'identifier la partie)

**round\_number (int) =** Le numéro du round actuel (fonction de temps)

**day\_of\_the\_round =** Permet d'identifier l'étape du round

**Supervisor =** Entité associée aux utilisateurs qui ont des droits de superviser/créer une partie : il s'agit des utilisateurs qui se créent un compte

**client\_demand =** C'est le superviseur qui détermine la demande finale (la demande des clients finaux, c'est à dire le marché)

**password =** Chaîne de caractères confidentielle choisi par le joueur lors de la création de son compte qui permet de sécuriser son compte

**username =** Identifiant unique des utilisateurs créant un compte

**administrator (boolean) =** Boolean qui permet de donner des droits de superuser (administrator ou personne en charge de la maintenance de l'application)

**Role =** Entité associée à une maille de la chaîne logistique dénommée aussi acteur

**type(supplier, middle\_chain, end\_chain) =** Un descriptif qui donne le type de l'acteur en fonction de sa position dans la chaîne

**nationality(german, french) =** Descriptif qui permet de savoir la nationalité de l'acteur ce qui va permettre de lui associer un icône avec le drapeau concerné

Interactions = Entité qui permet de relier des joueurs qui jouent différents acteurs de la chaîne logistique

lead\_time = Intervalle de temps écoulé entre le moment où la commande est passée et où la livraison est effectuée. Il s'agit d'un entier qui donne un nombre de rounds

delivery\_time = Intervalle de temps qui s'écoule entre le moment où les produits partent de l'acteur qui livre (acteur fournisseur) et le moment quand ils arrivent à destination (chez l'acteur client) - voir command

Command = Entité associée à une commande particulière qui permet de stocker l'historique des bilans

supplier = L'acteur qui livre la commande

client = L'acteur demandeur de la livraison qui reçoit la commande

arrival\_round = Le round quand la livraison arrive au client

demand\_qty = La quantité souhaitée (marquée sur le bon de commande) par le client

delivered\_qty = La quantité livrée réellement par le fournisseur

Règles de gestion :

- Les mots-de-passe sont salés et stockés après hachage.
- firm\_name (string) = différent pour tous les acteurs
- Départ de la livraison = Command.arrival\_round - Interactions.lead\_time
- Les Player.existing\_stock sont des tableaux d'entiers qui donne l'état de stocks d'un player à l'instant t
- Supervisor.client\_demand est un tableau d'entiers rempli par le superviseur qui représente la demande (faite par le marché) pour chaque maille finale de chaîne
- Les joueurs incarnent des acteurs : l'attribution de rôles se fait selon la règle "premier venu premier servi". Ainsi une fois qu'un acteur a été choisi par un joueur, il n'est plus disponible dans la liste de la partie concernée.

# Les Vues

## Accueil :

La vue d'accueil présente l'application et le descriptif du jeu.

Tout utilisateur arrivant sur l'application est redirigé sur cette vue.

Cette vue est aussi la vue par défaut si un utilisateur tente d'accéder à une vue qui n'existe pas ou dont il n'a pas les droits d'accès.

Un utilisateur connecté peut voir la liste de ses parties en cours. Pour chaque partie il a accès à un bouton "Rejoindre la partie" qui le redirige sur la vue du salon si la partie n'est pas lancée ou sur le tableau de bord sinon.

Depuis cette vue, il est possible en cliquant sur le bouton "Connexion" d'accéder à la route de connexion, à la route de création de partie (bouton "Créer une partie") si l'utilisateur est connecté et à la route permettant de rejoindre une partie (bouton "Rejoindre une partie"). Il y a un bouton déconnexion qui permet de se déconnecter pour un utilisateur authentifié. Il est également possible d'accéder à la vue décrite ci-dessous appelée "création de compte" en cliquant sur le bouton ayant le même nom.

*Charge estimée :* Difficile

## Création d'un compte :

Cette vue est accessible à tout utilisateur non connecté à l'application depuis la page d'accueil.

L'utilisateur doit remplir un nom d'utilisateur, un mot de passe et une confirmation du mot de passe. Le champ nom d'utilisateur correspond à l'attribut username de la table Superviseur, cet attribut doit être unique dans la base de données.

Un bouton "Créer le compte" envoie une requête POST sur la route de création de compte.

Un bouton "Annuler" envoie une requête GET sur la route d'accès à la page d'accueil.

L'utilisateur voit un message d'erreur si le nom d'utilisateur entré est déjà utilisé.

L'utilisateur voit un message d'erreur si les deux mots de passe sont différents.

*Charge estimée :* Facile

## Connexion :

Cette vue est accessible à tout utilisateur non connecté à l'application et permet de se connecter à son compte personnel.

L'utilisateur doit fournir son nom d'utilisateur et son mot de passe.

Un bouton "Se connecter" envoie une requête POST sur la route de connexion avec les informations remplies dans le formulaire.

Un bouton "Annuler" envoie une requête GET sur la route d'accès à la page d'accueil.

L'utilisateur voit un message d'erreur si le nom d'utilisateur n'existe pas dans la base de données dans l'attribut "username" de la table Superviseur.

L'utilisateur voit un message d'erreur si le couple (nom d'utilisateur, mot de passe) n'existe pas dans la base de données dans les attributs ("username", "password") de la table Superviseur.

*Charge estimée* : Facile

### **Création de partie :**

Cette vue n'est accessible que par un superviseur connecté (authentifié) à l'application.

Cette vue permet de configurer les informations d'accès à une nouvelle partie.

L'utilisateur doit renseigner un formulaire contenant le nom et le mot de passe de la partie.

Un bouton "Créer la partie" et un bouton "Annuler" sont aussi présents sur la vue.

Le bouton "Créer la partie" envoie le formulaire avec une méthode POST sur la route de création de partie.

Le bouton annuler envoie une méthode GET sur la route d'accès à la page d'accueil.

L'utilisateur voit un message d'erreur si le nom de la partie qu'il souhaite créer existe déjà dans l'attribut "name" de la table Game.

*Charge estimée* : Facile

### **Rejoindre une partie :**

Cette vue permet de rejoindre un salon de partie existant. Elle est accessible à tout utilisateur authentifié ou non.

L'utilisateur doit remplir un formulaire contenant le nom, le mot de passe d'une partie, le nom et un mot de passe pour l'entreprise qu'il va jouer pendant la partie.

Un bouton "Rejoindre la partie" envoie ce formulaire sur la route "Rejoindre une partie" avec une requête POST.

Un bouton "Annuler" envoie une requête GET sur la route d'accès à la page d'accueil.

L'utilisateur voit un message d'erreur si le nom et le mot de passe de la partie qu'il souhaite rejoindre n'existe pas dans les attributs "name" et "password" de la table Game.

*Charge estimée* : Moyenne

### **Salon d'une partie vue par le superviseur :**

Cette vue n'est accessible que par le superviseur de la partie.

Sur cette vue, le superviseur peut voir la chaîne logistique, le nombre de joueurs qui sont prêts à commencer la partie, le nom et le mot de passe de la partie.

Un bouton "Modifier les paramètres de la chaîne logistique" envoie une requête GET sur la route de paramétrage de la chaîne logistique.

Un bouton "Lancer la partie" envoie une requête GET sur la route "Tableau de bord d'une partie vue par le superviseur".

Un bouton "Quitter le salon" envoie une requête GET sur la route d'accès à la page d'accueil.

*Charge estimée : Difficile*

### **Paramétrage de l'état initial de la chaîne logistique :**

Cette vue n'est accessible que par le superviseur de la partie.

Le superviseur peut définir les valeurs initiales de la chaîne logistique pour les paramètres suivants : stocks, matières en cours de livraison, temps de livraison et temps de communication entre les acteurs. Il peut sauvegarder ces valeurs afin qu'elles soient appliquées lors de la partie avec le bouton "Enregistrer" ou revenir à la vue du salon d'une partie vue par le superviseur sans enregistrer les modifications avec le bouton "Annuler".

*Charge estimée : Difficile*

### **Salon d'une partie vue par un joueur :**

Sur cette vue, un joueur peut choisir l'acteur qu'il veut incarner pendant la partie parmi les acteurs disponibles.

Le nom de la partie et la chaîne logistique sont visibles au joueur.

Un bouton "Prêt" envoie une requête sur la route de sélection d'un acteur. Depuis cette vue, il est possible de revenir sur la vue d'accueil en appuyant sur "Quitter le salon".

Quand le superviseur lance la partie, le joueur est redirigé sur la route d'accès au tableau de bord d'un joueur.

*Charge estimée : Moyenne*

### **Tableau de bord d'une partie vue par le superviseur :**

Cette vue n'est accessible que par le superviseur de la partie.

Il voit le numéro sur le nombre total de rounds et de l'étape actuellement jouée par les joueurs sur le nombre d'étapes par round. Il a aussi accès à la liste des joueurs qui n'ont pas validé l'étape actuelle.

Il a aussi accès à un onglet "Fiches de suivi" où le superviseur peut voir la fiche de suivi d'un acteur sélectionnable avec un menu déroulant.

De plus, le superviseur voit les graphiques d'évolution des pénalités de chaque joueur.

Il peut depuis cette vue passer la partie à l'étape de jeu suivante.

*Charge estimée : Difficile*

### **Tableau de bord d'une partie vue par un joueur :**

Cette vue n'est accessible que par un joueur de la partie.

Un joueur ne peut voir que le tableau de bord lui appartenant. Sur la vue il y a 2 onglets supplémentaires vers "fiche de suivi" (voir la partie suivante) et vers "règles du jeu".

Il voit le numéro sur le nombre total de rounds et de l'étape actuellement jouée par les joueurs sur le nombre d'étapes par round. Il voit aussi les quantités qu'il a commandées pour ce round, son stock actuel et ses restes à livrer. S'il a reçu des commandes pendant ce round, le montant total de produit commandé est aussi affiché.

Cette vue possède les informations et un formulaire à remplir correspondant à l'étape actuellement joué :

- Pour la première étape, les commandes reçues et un bouton de validation sont affichés.
- Pour la deuxième étape, le formulaire contient, pour chaque client du joueur, la quantité qu'il souhaite livrer.
- Pour la troisième étape, la vue est composée, pour chaque fournisseur, de la quantité de matière reçue et d'un formulaire "Combien avez-vous de stock maintenant ?"
- Pour la quatrième étape, le formulaire contient, pour chaque fournisseur, la quantité de matière à commander

*Charge estimée* : Difficile

### **Affichage de la fiche de suivi :**

Cette vue n'est accessible que par un joueur de la partie.

Un joueur ne peut voir que la fiche de suivi lui appartenant.

Sur cette vue se trouve un tableau avec les informations remplies dans les rounds précédents.

*Charge estimée* : Moyenne

### **Vue Règles du jeu :**

On accède à cette vue depuis le tableau de bord. Elle sera statique et présentera un texte qui rappelle les règles du jeu.

*Charge estimée* : Facile

### **Graphiques de fin de partie :**

Affichage de graphiques d'analyse de fin de partie qui présente les estimations et les pénalités en fonction du round.

*Charge estimée* : Difficile



# Les Routes :

## Accueil :

La route permettra d'accéder à la page d'accueil.

Pour arriver sur cette route, on pourra passer par la route LogOut, la route Accueil, la route de Connexion ou lorsqu'on entre sur l'application.

Cette route peut nous amener sur la route de création de compte, la route rejoindre la partie, ou la route de connexion.

Sur la vue Accueil d'un superviseur, le superviseur pourra créer une partie et aura des boutons supplémentaires visibles uniquement pour lui. Ces boutons lui permettront de rejoindre les parties en train d'être jouées ou qui sont encore dans le salon de création.

*Charge estimée : Facile*

## Création d'un compte :

La route devra avoir une méthode GET afin de pouvoir afficher la vue de création de compte. Il faudra qu'on réalise une méthode POST qui prendra en paramètre un nom d'utilisateur et un mot de passe pour créer un nouvel utilisateur. Ce nom et ce mot de passe seront enregistrés dans les clés "username" et "password" respectivement de la table "Supervisor". Une confirmation de mot de passe permet la vérification de la syntaxe correcte du mot de passe de l'utilisateur.

Cette route nous amènera :

- soit vers la vue d'accueil si le nom et le mot de passe n'existe pas dans la base de données. (Comprend le bouton annulé)
- soit vers la page de création de compte si le nom du compte existe déjà dans la clé "username" de la table "Supervisor" ou bien si l'utilisateur n'a pas entré le nom ou le mot de passe ou si le mot de passe ne correspond à celle de la confirmation de mot de passe. La page de création de compte affichera également la raison de cette redirection, c'est-à-dire l'erreur de manière assez vague afin que l'utilisateur ne puisse deviner le nom et le mot de passe d'un autre compte comme "le mot de passe et le nom ne sont pas valident".

*Charge estimée : Moyenne*

## Connexion :

La route aura la méthode GET pour afficher la page de connexion et la méthode POST qui permet de vérifier que l'utilisateur fournit des informations de connexion correspondant à celles d'un utilisateur existant dans la base donnée. La table et les clés utilisées pour vérifier l'existence du compte seront la table "Superviseur" et les clés username et password .

Pour arriver sur cette route, il faut être sur la page d'accueil ou bien se tromper dans les données que l'on rentre dans la page de connexion.

Cette route nous amènera :

- soit vers la page d'accueil du compte associé au nom et au mot de passe entrés ou bien la vue d'accueil général si l'utilisateur va sur la route Accueil avec le bouton "annuler".
- soit vers la page de connexion si le nom du compte ou le mot de passe n'existe pas dans la base donnée ou bien si l'utilisateur n'a pas rentré l'un des deux paramètres. La page de connexion affichera également la raison de cette redirection, c'est-à-dire l'erreur de manière assez vague afin que l'utilisateur ne puisse deviner le nom et le mot de passe d'une autre partie.

*Charge estimée : Moyenne*

### **Rejoindre une partie :**

La route aura une méthode GET afin d'accéder à la vue rejoindre la partie et une méthode POST qui permettra de se rendre sur une partie et de vérifier que le nom de la partie et le mot de passe associé existent. On entrera également le nom de l'entreprise et le mot de passe du joueur ainsi qu'une case pour savoir si le nom de l'entreprise et le mot de passe existent déjà dans la base, si c'est le cas, on restera sur la vue rejoindre une partie avec les raisons de cette redirection.

Si le nom de l'entreprise et le mot de passe sont correctes, on a 4 issues possibles :

-Si la partie n'a pas encore été lancée et que les acteurs n'ont pas tous été occupés, le joueur est dirigé vers la route "Salon vue par un joueur". Cela se traduit dans la base de donnée par vérifier que la somme des lignes dans la table Player ne sont pas égales à la somme du nombre de lignes dans la table Actor.

-Si la partie n'a pas encore été lancée et que les acteurs sont tous affectés ou si la partie est lancée et que le joueur rentre un nom d'entreprise et mot de passe inexistant dans la partie, le joueur est redirigé vers une vue d'erreur expliquant que le nombre d'acteurs est complet. Cela se traduit dans la base de donnée par vérifier que la somme des lignes dans la table Player sont égales à la somme du nombre de lignes dans la table Actor.

-Si la partie est lancée et que le joueur rentre un nom d'entreprise et mot de passe existant dans la partie, le joueur est redirigé vers le tableau de bord du joueur associée au compte. Cela se traduit dans la base de donnée par vérifier sur chaque ligne de la table "Player" qu'il existe une ligne telle que la clé "id\_game" associée au nom de la partie et mot de passe rentrée, que la clé "firm\_name" et "password" de la table Player correspondent nom d'entreprise et mot de passe rentrée par l'utilisateur.

Si le nom et le mot de passe ne convient ou que l'un des paramètres est manquant:

- La route nous redirigera vers la vue permettant de rejoindre la partie. Pour accéder à la partie depuis la route "Rejoindre la Partie", le nom de la partie rentré est vérifié en regardant la clé "name" de la table Game

Les erreurs nous renvoyant sur la vue "Rejoindre la Partie" sont :

- un mot de passe ou un nom de partie vide
- une combinaison mot de passe et nom de la partie non existant

Pour accéder à cette route, il faut passer par la page d'accueil qui possède un bouton rejoindre la partie.

Si le nom et le mot de passe de la partie existe dans la base données, cette route nous amènera :

- soit vers la page du salon superviseur si l'utilisateur est le créateur de la partie.
- soit vers la page du salon joueur si l'utilisateur n'est pas le créateur de la partie.

*Charge estimée* : Difficile

### **Création de partie :**

La route permettra par une méthode GET d'accéder à la page de création de la partie et elle aura une méthode POST pour créer une partie et vérifier que le nom de la partie n'existe pas déjà. Pour vérifier l'existence de la partie, on vérifiera l'égalité entre les lignes de la clé "name" de la Partie de la table Game. Dans le cas où la partie ne peut être créer, l'utilisateur qui aura créé la partie ne pourra pas

La méthode POST permettra également de se rediriger vers la page permettant d'accéder à la première page de modification de la partie du jeu. Le nom de la Partie sera stockée dans la table Game.

Les erreurs nous renvoyant sur la vue "Création de la Partie" sont :

- un mot de passe ou un nom de partie vide
- une combinaison mot de passe et nom de la partie non existant

Pour arriver sur cette route, on doit passer par la page d'Accueil, ou bien par la page de connexion .

Cette route nous amènera sur la page Salon du superviseur de la partie, page qui permettra de configurer les paramètres du jeu.

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien un utilisateur qui possède un compte donc dans notre base donnée donnée un élément contenu dans la table Superviseur.

*Charge estimée* : Moyenne

## **Salon d'une partie vue par le créateur de la partie:**

La route aura pour but d'accéder à la page dite "salon" de la partie vue par le créateur de la partie.

Cette aura une méthode GET qui permet d'accéder à la page de salon de la partie en tant que superviseur.

Pour accéder à cette partie, il faut passer par la route Création de la partie ou bien par la route rejoindre une Partie dès lors que l'utilisateur qui rejoint la partie utilise le compte du créateur de la Partie.

Cette route peut amener sur la route :

- permettant de modifier les paramètres.
- "Page principale partie superviseur". Cependant pour pouvoir lancer la partie, la route prendra en paramètre le nombre d'utilisateurs qui sera dans la chaîne logistique et vérifiera que tous les rôles d'acteurs de la chaîne ont bien été assignés.
- Cancel permettant de revenir sur la page d'accueil.

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien le superviseur.

*Charge estimée : Facile*

## **Salon d'une partie vue par un joueur de la partie :**

La route aura pour but d'accéder à la page dite "salon" de la partie vue par un joueur de la partie.

Cette méthode aura une méthode GET qui permet d'accéder à la page de salon de la partie en tant que joueur et une méthode POST permettant à l'utilisateur de rentrer son rôle dans la chaîne logistique. Le choix du rôle se fait par "premier arrivé premier servi".

La méthode POST vérifiera que dans la table player, il n'y a pas de ligne avec une clé "Role" ayant le même rôle choisi présent pour l'id\_game associé à la clé "name" et "password" de la table "Game". Cela n'est normalement pas possible puisque l'on supprimera les rôles d'acteur déjà pris. Cela est possible si deux joueurs choisissent le même rôle en même temps.

Pour accéder à cette partie, il faut passer par la route Rejoindre la Partie.

Cette route peut amener sur la route :

- Tableau qui permettra de modifier les données dans la base de données et de figer la ligne sur la vue "Salon d'une partie vue par un joueur" pour empêcher le changement de choux d'un acteur.
- Accueil permettant de revenir sur la page d'accueil via le bouton "Quitter le salon".

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien le joueur associée à cet acteur et non pas un autre joueur.

*Charge estimée : Difficile*

## **LogOut :**

La route LogOut permettra de déconnecter un utilisateur de sa session ainsi que de se rendre sur la route Accueil de l'application. Pour accéder à la route LogOut, on peut passer :

- par la route de la page d'Accueil du compte d'un utilisateur.

*Charge estimée :* Facile

## **Paramétrage Chaîne Logistique :**

La route aura une méthode GET permettant d'accéder à la page de paramétrage et une méthode POST qui permettra au superviseur de rentrer les paramètres du jeu dans les tables Game et Interactions associées à la partie du jeu. Les données rentrées seront ajoutées dans la table "Game" en ce qui concerne les paramètres de la chaîne logistique et pour ce qui est des acteurs, on les mettra dans la table "Interactions" ainsi que dans la table "Player" pour ce qui du stock initial.

Cette route ne sera accédé qu'à partir de la route salon d'une partie vue par le créateur du jeu.

Cette route peut amener sur le salon d'une partie vue par le créateur du jeu ou bien la route de la page principal du superviseur.

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien le superviseur.

*Charge estimée :* Difficile

## **Tableau de bord d'une partie vue par le superviseur :**

Le superviseur peut accéder à et visualiser les fiches de suivi de chaque acteur de la chaîne (onglet supplémentaire).

La route aura une méthode GET permettant d'accéder à la vue du tableau de bord. La méthode GET récupérera les éléments des clés "demanded\_qty", et "delivered\_qty" de la table Command et la clé "stock\_existant" de la table "Player" pour calculer les pénalités de stock restant de reste à livrer de chaque round et les afficher sous forme de graphiques de chaque acteur. On aura 4 graphiques qui affichent pour tous les joueurs en même temps : les pénalités en cumulée et par round pour les deux types de pénalités. Cette actualisation des graphiques devra se faire à chaque round.

La route aura également une méthode POST qui permettra

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien le superviseur.

Cette route pourra être accessible via :

- la route de la page principale de la partie vue par le superviseur
- la route des graphiques vues par le superviseur

Cette route permettra d'accéder à:

- la route de la page principale de la partie vue par le superviseur
- la route de la page principale de la partie vue par le superviseur après passage au round suivant
- la route des graphiques vues par le superviseur

*Charge estimée : Difficile*

### **Tableau de bord d'une partie vue par un joueur :**

La route aura une méthode GET permettant de visualiser le tableau de bord du joueur ainsi que d'afficher la vue du formulaire pour passer une commande, le formulaire d'envoi des commandes et d'afficher via un formulaire les commandes reçues. La méthode GET permettra de calculer les stocks restants de chaque joueur à partir des clés "delivered\_qtity" de la table "Command" et la clé "stock\_existant" de la table "Command". Cette méthode GET permet également d'avoir les commandes reçues au round. Pour cela, la méthode GET récupérera dans la table "COMMAND" toutes les clés "delivered\_qtity" tels que la clé arrival\_round correspond au round actuel, la clé id\_clients correspondent à celles du joueur qui reçoit les commandes, et la clé id\_fournisseur séparer les différentes livraisons dans l'affichage de la vue.

Pour accéder à cette route, il faut passer soit par la route du tableau de suivi, soit par la route du tableau de bord ou bien la route salon d'une partie vue par un joueur.

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien un joueur qui s'est connecté avec l'acteur associée à ce tableau de bord.

Les méthodes POST peuvent être séparées en 2 catégories :

– Passer une commande

La méthode POST permettra de rentrer les données du formulaire et de les mettre dans les bases de données du joueur. Cependant, cette méthode ne sera possible que si les données rentrées dans la fiche de suivi sont justes. Dans le cas contraire, si le joueur valide, un message d'erreur s'affiche. La Méthode POST ira mettre dans la table "Command", l'id du joueur de celui qui passe les commandes dans la clé "id\_client", l'id de celui à qui l'on passe les commandes dans la clé "id\_fournisseur", le round auquel le joueur qui a passé la commande recevra la livraison, dans la clé "arrival round", et la quantité demandé dans la clé "demanded\_qtity". Pour le round d'arrivé, on récupère les clés Tant que la quantité délivrée, n'aura pas été envoyé on mettra 0 dans la clé "delivered\_qtity".

– Envoyer une commande

La méthode POST permettra de rentrer les données du formulaire et de les mettre dans les bases de données du joueur. Cependant, cette méthode ne sera possible que si les données rentrées sont justes. Dans le cas contraire, si le joueur valide, un message d'erreur s'affiche. Une méthode PATCH permettra d'afficher les commandes reçues. La Méthode POST ira vérifier les valeurs à partir de la clé "stock\_existant" de la table "Player", la clé "arrival\_round" et "delivered\_qtity" de la table "Command" pour vérifier que le stock\_existant

permet de fournir les quantités à envoyer. Cette méthode POST ira également mettre à jour la quantité à envoyer dans la clé "delivered\_qty" de la table "Command". Un code python permettra de vérifier la cohérence du remplissage de la fiche de suivi.

Il faut que la somme des commandes envoyées durant le round ne soit pas supérieur au stock disponible.

Cette route peut amener sur :

- la route de la fiche de suivi
- la route terminer la partie

*Charge estimée : Difficile*

### **Terminer la partie :**

La route aura une méthode GET qui permettra de passer à la route "Tableau de bord vue par un joueur" du round suivant ou bien aux graphiques si la clé "Finalisé" de la table Player est à True pour tous les joueurs au round actuel.

Pour cela, il y aura deux conditions nécessaires :

- Pour les joueurs et le superviseur de la partie, il faudra que la clé "round\_number" et la clé "total\_round" de la table "Game" pour l'id\_game donné soit égales et que la clé "day\_of\_round" de la table "Games" soit égale à 4 (finir la semaine en résumé). A ce moment, le joueur sera amenée sur la route Graphiques de fin de partie
- Sinon ils seront amenés sur la route "Tableau de bord vue par le joueur" du round suivant. Il faudra actualiser toutes les valeurs.

Cette route pourra être accédée par la route "Tableau de bord vue par un joueur" et "tableau de bord vue par un superviseur"

Cette route pourra rediriger les joueurs et le superviseur vers la route Accueil.

*Charge estimée : Difficile*

### **Affichage de la fiche de suivi :**

La route aura une méthode GET permettant à la vue de la fiche de suivie et une méthode POST permettant de rentrer les données de la fiche de suivi dans les bases de données du joueur. Cependant, cette méthode ne sera possible que si les données rentrées dans la fiche de suivi sont justes et une fois que tous les autres joueurs ont validé leur fiche de suivi. Dans le cas contraire, si le joueur valide, un message d'erreur s'affiche. Une méthode PATCH permettra d'afficher les commandes reçues. La Méthode POST ira vérifier les valeurs à partir de la clé "stock\_existant" de la table "Player", la clé "arrival\_round" et "delivered\_qty" de la table "Command" pour vérifier les valeurs rentrés correspondent bien

à celles de la route Passer les commandes. Un code python permettra de vérifier la cohérence du remplissage de la fiche de suivi.

Pour accéder à cette route, il faut passer soit par la route de la page principale vue par le joueur de la partie, soit par la route passer une commande, soit par la route envoyer une commande.

Cette route peut amener sur :

- la route des graphiques du joueur lorsque la partie est terminée
- la route tableau de bord du joueur
- la route de la fiche de suivi si les données rentrés ne sont pas correctes

*Charge estimée : Moyenne*

### **Graphes de fin de partie :**

Une méthode GET sera implémentée sur la route afin d'afficher les courbes des pénalités de chaque joueur. Cette méthode GET ira :

- récupérer les données des clés "demanded\_qty" et "delivered\_qty", et "id\_client" de la table "Command". En effet, pour calculer les pénalités de chaque joueur, on va faire la somme sur chaque clé "delivered\_qty" et "demanded\_qty" des lignes qui possèdent les clés "id\_game", "id\_client" ayant un "arrival\_round" inférieur ou égal à n dans les clés puis la différence entre les sommes calculées pour obtenir la pénalité des restes à livrer au round n.
- Pour les pénalités de stock au round n, il faudra faire la même chose que le point précédent auquel il faudra ajouter le stock existant de la clé "stock\_existant" dans la table "Player". Si le calcul donne un nombre négatif (car on n'a pas suffisamment de stocks pour répondre à la demande client), il faudra mettre zéro : le client n'aura plus de stocks tout simplement pour répondre aux commandes.
- Pour les joueurs et le superviseur de la partie, il faudra que la clé "round\_number" et la clé "total\_round" de la table "Game" pour l'id\_game donné soit égales et que la clé "day\_of\_round" de la table "Games" soit égale à 4 (finir la semaine en résumé).
- Pour accéder à cette route, il faut que tous les joueurs soient sur la route "fiche de suivi" et qu'ils aient tous cliqué sur le bouton prêt. Le superviseur sera directement redirigé vers cette route.

Cette route pourra être accédée par la route

Cette route pourra rediriger les joueurs et le superviseur vers la route Accueil.

*Charge estimée : Difficile*



## Estimation charges

### Les vues :

Vues	Catégories	Unité de tps	temps réel
Accueil	Difficile	4	16
Création d'un compte	Facile	1	4
Connexion	Facile	1	4
Création de partie	Facile	1	4
Rejoindre une partie	Moyenne	2	8
Salon d'une partie vue par le superviseur	Difficile	4	16
Paramétrage de l'état initial de la chaîne logistique	Difficile	4	16
Salon d'une partie vue par un joueur	Moyenne	2	8
Tableau de bord d'une partie vue par le superviseur	Difficile	4	16
Tableau de bord vue par un joueur	Difficile	4	16
Affichage de la fiche de suivie	Moyenne	2	8
Règles du jeu	Facile	1	4
Graphiques de fin de partie	Difficile	4	16

## Les routes :

Routes	Catégories	Unité de tps	temps réel
Accueil	Facile	1	4
Création d'un compte	Moyenne	2	8
Connexion	Moyenne	2	16
Création de partie	Difficile	4	16
Rejoindre une partie	Difficile	4	16
Salon d'une partie vue par le superviseur	Difficile	4	16
Paramétrage de la chaîne logistique (route qui sert dans le cas où chaîne modulaire)	Difficile	4	16
Salon d'une partie vue par un joueur	Difficile	4	16
Tableau de bord d'une partie vue par le superviseur	Difficile	4	16
Tableau de bord vue par un joueur	Difficile	4	16
Affichage de la fiche de suivie	Moyenne	2	
Règles du jeu	Facile	1	4
Graphiques de fin de partie	Difficile	4	16
LogOut	Facile	1	4
Terminer la partie	Difficile	4	16

Total :

On multiplie le tout par 1,5 pour l'intégration et on obtient **462** heures

Disponibilités : **531** heures