

Application Method to my Business

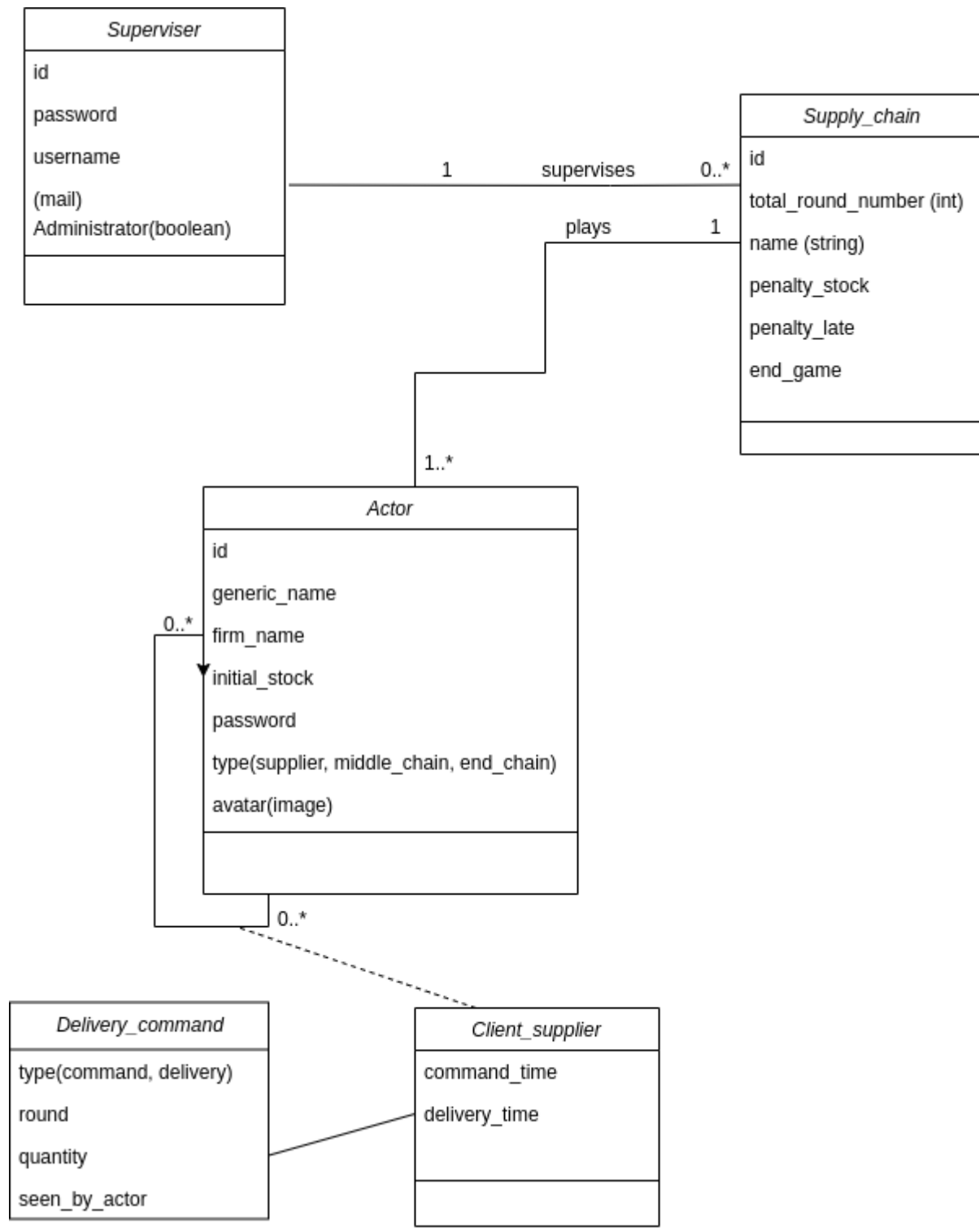
- *Spécifications Techniques* -

Le Modèle Conceptuel de Données :	3
Le Modèle	3
Dictionnaire de données	4
Les Vues	6
Accueil :	6
Création d'un compte :	6
Connexion :	7
Création de partie :	7
Rejoindre une partie :	7
Salon d'une partie vue par le superviseur :	8
Paramétrage de l'état initial de la chaîne logistique :	8
Salon d'une partie vue par un joueur :	9
Salle d'attente :	9
Tableau de bord d'une partie vue par le superviseur :	9
Tableau de bord d'une partie vue par un joueur :	10
Affichage de la fiche de suivi :	11
Vue Règles du jeu :	11
Graphiques de fin de partie :	11
Les Routes :	12
Accueil :	12
Création d'un compte :	12
Connexion :	12
Rejoindre une partie :	13
Création de partie :	15
Salon d'une partie vue par le créateur de la partie:	16
Salon d'une partie vue par un joueur de la partie :	16
Logout :	17
Paramétrage Chaîne Logistique :	17
Tableau de bord d'une partie vue par le superviseur (Graphes pénalités vues par le superviseur) :	17
Tableau de bord d'une partie vue par le joueur	18
Terminer la partie (automatique) - voir méthode post du tableau de bord vue par un superviseur :	20
Affichage de la fiche de suivi :	20
Graphes de fin de partie :	20
Estimation charges	22
Les vues :	22

Les routes :	23
Total :	24

Le Modèle Conceptuel de Données :

1. Le Modèle



2. Dictionnaire de données

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

username = Nom d'utilisateur du superviseur

password = Chaîne de caractères confidentielle choisi par le superviseur lors de la création de son compte

administrator = booléen qui détermine si le superviseur est administrateur de l'application

Supply_chain = Entité associée à une chaîne logistique 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 = Nom d'une partie (unique afin d'identifier la partie)

penalty_stock = La valeur de la pénalité pour un produit en stock à la fin d'un round

penalty_late = La valeur de la pénalité pour un produit non livré

end_game = Un booléen qui permet d'arrêter une partie avant d'atteindre un nombre de round égal à total_round_number

Actor = Entité associée joueurs et aux acteurs de la chaîne logistique

generic_name = Le nom générique de l'acteur de la chaîne logistique

username = Identifiant du joueur défini lorsqu'il prend le contrôle d'un acteur

password = Chaîne de caractères confidentielle choisi par le joueur lors de la prise de contrôle d'un acteur

initial_stock = Stock de l'acteur en début de simulation de la chaîne logistique

actor_type = Chaîne de caractères permettant de différencier les acteurs en bout de chaîne (le marché et les fournisseurs avec un stock infini) des acteurs pouvant être sélectionné par un joueur

Client_supplier = Entité représentant les liens de client et fournisseurs de deux acteurs

command_time = Le délai entre l'envoi de la demande d'approvisionnement par le client et la réception de cette demande par le fournisseur

delivery_time = Le délai entre l'envoi de la livraison par le fournisseur et la réception de cette livraison par le client

Delivery_Command = Entité représentant les envois de bons de commandes et les livraisons de matières

quantity = La quantité demandée par le client ou livrée par le fournisseur

seen_by_actor = Booléen utilisé pour identifier les commandes à afficher pendant la première étape de chaque round

Règles de gestion :

- Les mots-de-passe sont salés et stockés après hachage.
- firm_name (string) doit différer pour tous les joueurs d'une même chaîne logistique
- 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.

Depuis cette vue il y a plusieurs bouton :

- "Créer une partie"

Si l'utilisateur est déjà connecté le bouton envoie une requête GET sur la route "Création de partie" (create)

Si l'utilisateur n'est pas connecté le bouton envoie une requête GET sur la route "Connexion"

- "Rejoindre une partie"

Ce bouton envoie une requête GET sur la route "Rejoindre une partie"(join). L'utilisateur n'a pas besoin d'être connecté pour accéder à cette vue.

- "Liste parties"

Ce bouton n'est visible que si l'utilisateur est connecté à l'application. Il redirige l'utilisateur vers la vue "Liste de mes parties".

Enfin il y a un bouton "Déconnexion" (log_out) qui permet de se déconnecter pour un utilisateur authentifié et qui renvoie sur la vue Accueil

Charge estimée : Difficile

Création d'un compte :

Cette vue est accessible après être passé par la route "Créer une partie". Alors une fois passée par là depuis la page index, si l'on est pas connecté la

On accède à cette vue depuis la route "Créer un compte". Route qui est accessible depuis la vue de connexion.

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 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 depuis la route "Créer une partie" si l'utilisateur n'est pas connecté. 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.

Un bouton "Créer un compte" envoie une requête GET sur la route de création de compte.

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 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 Supply_chain.

Charge estimée : Facile

Liste des parties :

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

Cette vue permet de visualiser l'ensemble des parties créées par le superviseur. Pour chacune de ces parties, le superviseur a un bouton "Rejoindre le salon de la partie" qui lui permet d'accéder au salon de la partie.

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

Charge estimée : Moyenne

Rejoindre une partie :

Cette route permet de rejoindre un salon de partie existant. Elle est accessible à tout utilisateur authentifié ou non. Elle a également sa propre vue.

L'utilisateur doit remplir un formulaire contenant le nom de la partie, et dans le cas où l'utilisateur aurait déjà choisi un acteur de la chaîne logistique, le nom et le 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 de la partie qu'il souhaite rejoindre n'existe pas dans l'attribut "name" de la table supply_chain.

Structure du code :

Nous avons mis cette route dans le fichier play.py qui contient les routes spécifiques aux joueurs d'une partie. Elle possède ainsi le blueprint play et affiche des messages d'erreurs lorsque le nom de la partie n'a pas été entré ou qu'elle n'existe pas.

Si le joueur rentre un nom d'entreprise et un mot de passe qui n'existe pas dans la base de donnée, un message d'erreur s'affiche. Dans tous les cas, il est ramené à la vue "Rejoindre la Partie" grâce à la fonction render_template.

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.

Le superviseur voit également la liste des acteurs et tous les acteurs qui ont été sélectionnés par un joueur. Pour chaque acteur sélectionné par un joueur, il peut utiliser le bouton "Réinitialiser" pour retirer les attributs firm_name et password de l'acteur. Cela permet de réinitialiser un acteur si un joueur oublie son mot de passe.

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 et est requise pour se connecter grâce à un wrapper.

Un bouton “Rejoindre la Chaîne” envoie une requête sur la route “Lobby” des joueurs. Depuis cette vue, il est possible de revenir sur la vue d’accueil en appuyant sur “Annuler”.

La méthode POST qui permet de sélectionner l’acteur vérifiera que l’utilisateur n’a pas rentré le même nom d’entreprise ou le même acteur. Dans ces deux cas, un message d’erreur s’affiche et le joueur est redirigé vers la route “Lobby” du joueur.

Le joueur est redirigé sur la route d’accès au tableau de bord d’un joueur si tous les joueurs sont prêts sinon le joueur est redirigé vers une page d’attente en attendant que les autres joueurs soient prêts. Pour cela, on vérifiera que le nombre de lignes dans la base de données qui fait référence à la même partie est égale au nombre de lignes ayant un nom d’entreprise rempli (attribut `firm_name` de la table `Actor`) et faisant référence à la même partie dans la base de données.

Structure du code:

Cette route est également dans le fichier `play.py` et possède ainsi le blueprint `play` pour spécifier que cette route est pour un joueur.

Charge estimée : Moyenne

Salle d’attente :

Cette vue permet d’avoir un intermédiaire avant le salon de la partie et après le lobby. Il s’agit d’une vue qui est uniquement un page de type “chargement en cours”. Cela demande au joueur d’attendre jusqu’à ce que les autres joueurs aient choisi leur propres acteurs et appuyé sur le bouton ‘Prêt’. Elle est revue aussi lors d’un passage entre 2 étapes lorsque tous les acteurs n’ont pas fini de remplir leur tâches(par exemple calculer le stock etc). Ainsi cette page s’assure que tous les joueurs sont redirigés vers leur tableau de bord en même temps.

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

Fin de Partie :

Cette route ne possède qu'une méthode POST et permet de mettre à jour la base de données pour savoir si la partie est terminée. L'attribut `end_game` sera à True si le superviseur clique sur le bouton "Terminer la Partie" et sera à False si le superviseur clique sur le bouton "Reprendre la Partie". Dans le cas d'une partie finie, les joueurs ne peuvent plus continuer les étapes et ils sont dirigés vers les graphiques de fin de partie.

Structure du code :

La route associée à cette vue est dans le fichier `supervise.py` et est muni d'un blueprint `supervise` car seul le superviseur de la partie peut terminer la partie manuellement.

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

Cette vue n'est accessible que par un joueur de la partie via la méthode GET de la route "`playre_game`".

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" et vers "règles du jeu".

L'utilisateur voit le numéro du round en cours. Il voit aussi les quantités qu'il a commandées et qui sont en cours, et son stock actuel. 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 qui est une méthode POST de la route "`player_game`".
- Pour la deuxième étape, le formulaire contient, pour chaque client du joueur, la quantité qu'il souhaite livrer et le calcul des restes à livrer laissé au joueur.
On lui affichera également les restes à livrer du joueur avant qu'il ne lance ses livraisons. Cette étape sera muni d'une méthode POST pour vérifier que les quantités rentrées dans le formulaire sont cohérentes.
- Pour la troisième étape, la vue est composée, pour chaque fournisseur, de la quantité de matière reçue et d'un formulaire lui demandant de rentrer son stock actuel. Une méthode POST via le bouton "Valider" permettra de mettre à jour la base

de données afin de signaler que le joueur a bien vu les livraisons et que le calcul du stock actuel est juste.

- Pour la quatrième étape, le formulaire contient, pour chaque fournisseur, la quantité de matière à commander.

Pour l'étape 2, un message d'erreur s'affiche lorsque :

- *Le joueur n'a pas assez de stock par rapport à la quantité qu'il veut livrer.*
- *Le joueur envoie une livraison plus importante que ce que le client souhaite et que ce qu'il reste à livrer.*
- *Le joueur insère une valeur négative.*

Pour l'étape 3, un message d'erreur s'affiche lorsque :

- *Le joueur ne rentre pas la bonne valeur du stock actuel.*

Structure du code :

La route concernant cette vue se trouve dans le fichier play.py et possède un blueprint play. Nous avons rédigé un autre fichier appelé step.by afin d'écrire des fonctions plus claires et permettant de réduire la taille du fichier play.py

Charge estimée : Difficile

Affichage de la fiche de suivi :

Cette vue n'est accessible que par un joueur de la partie. De plus, le joueur ne peut voir que la fiche de suivi lui appartenant. Elle contient le stock actuel, les commandes en cours, les livraisons en cours ainsi que le reste à livrer.

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. Il s'agit d'un bouton "fiche de suivi" qui se présente sous la forme d'un onglet en haut de la page. La vue sera statique et présentera un texte qui rappelle les règles du jeu.

Charge estimée : Facile

Graphiques de fin de partie :

Une route est créée pour que le joueur puisse accéder au graphes de fin de partie via une méthode GET.

La vue affiche des graphiques d'analyse de fin de partie qui représentent les estimations et les pénalités en fonction du round. Les graphiques concernent tous les acteurs de la chaîne logistique.

Structure du code :

La route associée se trouve dans le fichier play.py et possède un blueprint play.

Nous avons rédigé un autre fichier appelé step.py afin d'écrire les fonctions qu'on appelle par la suite et permettant donc de réduire la taille du fichier play.py

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. Un bouton "Accueil" est également présent sur chacune des routes pour pouvoir revenir à l'accueil à tout moment.

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.

De plus, la méthode POST de cette route permet de stocker le "password" hashé et salé et garantit l'unicité du "username". De plus, s'il y a une erreur (username déjà utilisé, un des champs non rempli ou bien mauvaise confirmation de mot de passe) l'utilisateur est renvoyé sur la vue de création de compte avec un message d'erreur.

Enfin si tout se passe bien l'utilisateur est redirigé vers la vue "Connexion"

Charge estimée : Moyenne

Connexion :

La route aura une méthode GET pour afficher la page de connexion

La route aura également une méthode POST qui prend en paramètres les noms d'utilisateur et mot de passe et qui redirige vers la page d'accueil. Elle gère également les erreurs (mauvais identifiants, mauvais mot de passe ou bien données manquantes) et renvoie sur la page de connexion dans le cas où il y en aurait eu une.

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.

Méthode POST

Pour cette méthode POST, on va vérifier dans la table "Supply_chain" qu'il existe une ligne contenant le nom de la partie et le mot de passe associés respectivement aux clés "name" et "password". Si le nom de la partie et le mot de passe ne sont pas corrects, on renverra dans la route "Rejoindre la Partie" un message d'erreur expliquant que le nom de la partie et le mot de passe ne sont pas justes.

1- Vérification utilisateur qui veut participer à la partie

Si l'utilisateur ne rentre que le nom de la partie et le mot de passe de la partie, deux options s'offrent à nous.

Soit tous les rôles ont déjà été affectés et donc l'utilisateur est redirigé vers la route Rejoindre la Partie avec un message d'erreur tel que "Il n'y a plus de places disponibles". Pour cela, on va vérifier que la somme des lignes de la table "Actor" qui ont la clé "name_supply_chain" de la table "Supply_chain" associée au nom de la partie auquel il veut se connecter, soit égale à la somme des lignes de la table "Actor".

Dans le cas contraire, où il y a encore des acteurs disponibles, l'utilisateur sera redirigé vers la route "Salon vue par le joueur".

On vérifiera que le nombre de personnes connectées sur le salon est bien inférieure ou égal au nombre d'acteurs possibles.

2- Vérification existence d'un joueur de la partie

On entrera également la possibilité de rentrer le nom de l'entreprise et le mot de passe du joueur si l'on veut se connecter à un compte déjà existant de la partie.

Pour cela, on va vérifier qu'il existe une ligne qui contient le nom d'entreprise et le mot de passe entrés par l'utilisateur dans les clés "firm_name" et "password" de la table "Actor" respectivement, avec la bonne clé "name_supply_chain", associée au nom de la partie et au mot de passe, de la table "Supply_chain".

Si le nom du joueur et le mot de passe entrés ne correspondent pas on le ramènera vers la route Rejoindre la Partie avec un message d'erreur 'Vous n'avez pas entré correctement un point'.

On vérifiera qu'il n'y aura pas déjà un utilisateur connecté sur le joueur.

2.a - Vérification du lancement de la partie pour le joueur déjà créé

Une fois que l'on a rentré le bon nom d'entreprise et le bon mot de passe, il faut que l'on vérifie si la partie a commencé.

Pour vérifier cela, on va sélectionner la clé "round" de la table "supply_chain" et voir si elle est égale à 0.

Si round = 0 , cela signifie que la partie n'a pas commencé. Le joueur est ainsi redirigé vers la route "Salon du joueur" dans lequel il aura les lignes nom d'utilisateur, mot de passe et confirmation préremplie par les valeurs qui l'avaient mis au préalable.

Si round > 0, cela signifie que la partie a déjà commencé , le joueur est redirigé vers la route "Tableau de bord vue par un joueur" de ce joueur.

On vérifiera qu'il n'y aura pas déjà un utilisateur connecté sur le joueur.

3- Bilan des erreurs possibles

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 existante dans la table "Supply_chain".
- Un mot de passe ou nom du joueur vide ou non présent dans la table "Actor"
- Une personne déjà connectée sur le joueur auquel on souhaite se connecter.
- Un nombre d'utilisateurs connectés supérieurs aux personnes dans le salon alors que nous n'utilisons pas encore de compte.

4- Atterrir sur la route Rejoindre la Partie

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

""

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 qu'il n'existe aucune ligne portant le nom de la partie dans la clé "name" de la table Supply_chain.

Dans le cas où la partie ne peut être créée, l'utilisateur sera redirigé vers la route "Création de partie" avec un message d'erreur.

D"

ans le cas contraire, on insérera une ligne dans la table "Supply_chain" avec le bon nom de la partie dans la clé "name_supply_chain", le mot de passe entré par le superviseur dans la clé "password", l'id sera incrémentée de 1 à chaque ajout de ligne, et l'attribut "total_round_number" sera initialisé à 5.

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 existante dans la table "Supply_chain". Pour cela, on vérifiera que le nom de la partie ne fait référence à aucune ligne de la table "Supply_chain" à partir de la clé "name". Cela se résume en termes de requête à renvoyer rien (None).

Pour arriver sur cette route, on doit passer par la route Accueil d'un superviseur.

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

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien avec le compte du superviseur et qu'il n'y a pas quelqu'un déjà connecté sur le compte du 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 :

- "Paramétrage initial".
- "Tableau de bord vu par le superviseur". Cependant pour pouvoir arriver sur cette route, 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. Pour cela, on va vérifier que la somme des lignes de la table acteur est bien égale à la somme des lignes dans la table "Actor". Dans le cas contraire, la route sera redirigée vers la route "Salon d'une partie vue par le créateur de la partie".
- Accueil à l'aide du bouton "Accueil".

La route vérifie que l'utilisateur qui tente de rentrer dans cette route-ci est bien le superviseur et que deux utilisateurs ne peuvent accéder au même compte du 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 vue "Salon de la partie vue par un joueur" et une méthode POST permettant à l'utilisateur de rentrer son rôle, son nom d'entreprise et son mot de passe 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 "Actor" ayant le même rôle choisi présent pour la partie de la table Actor. De même pour le nom d'entreprise avec la clé "firm_name", "name_supply_chain en clé étrangère dans la table "Actor". Cela n'est normalement pas possible d'avoir deux mêmes rôles puisque l'on supprimera les rôles d'acteur déjà pris dans les choix possibles. Cela est possible si deux joueurs choisissent le même rôle en même temps.

Si le joueur peut sélectionner cet acteur, on remplit les attributs firm_name et password par les valeurs remplies par le joueur.

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

Cette route peut amener sur la route :

- "Salon vue par un joueur" si le nom du joueur ou si l'acteur est déjà pris dans la méthode POST.
- "Tableau de bord vue par un joueur" si la table "Actor" a tous les rôles des acteurs dans sa table. Cela se traduit par vérifier qu'aucun acteur de la chaîne qui a pour type "player" n'a de nom d'entreprise vide.
- "Accueil" permettant de revenir sur la page d'accueil via le bouton "Quitter le salon".

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 Supply_chain et Client_supplier associées à la partie du jeu. Le superviseur peut choisir de modifier les délais de livraison et de commande (stocké dans la table Client_supplier), les stocks initiaux des joueurs (stocké dans la table Actor) et les quantités en cours de livraison lors de la création de la chaîne logistique.

Cette route crée ou supprime des livraisons pour que les joueurs reçoivent des matières tous les rounds.

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".

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 (Graphes pénalités vues 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 "quantity" de la table Command et la clé "stock_initial" de la table "Player" pour calculer les pénalités de stock restant de reste à livrer de chaque round (à l'aide de la clé "round" de la table delivery_command et les clés "command_time" et "delivered_time" de la table "Interactions") 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 d'accéder à la route "Fiches de suivi"

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 "Tableau de bord vue par le superviseur"
- la route Graphes de fin de partie si la clé "round" est égale à la clé "total_round_number" de la table Supply_chain.
- la route "Fiches de suivi"

Cette route permettra d'accéder à:

- les routes "Fiches de suivi superviseur" qui permettent de visualiser les fiches de suivi de chaque joueur en fonction d'un paramètre dans la vue qui est le rôle du joueur dans la chaîne logistique.
- la route du "Tableau de bord vue par le superviseur" et on mettra à jour les données sur les tableaux de bords lorsque tous les individus auront finalisé leurs tâches sinon il sera redirigé vers le même tableau de bord. Cet actualisation passera également par une incrémentation de la clé "round" de la table "Game" afin de passer au round suivant.
- la route "Graphique de fin des pénalités" lorsque l'on sera au dernier round

Charge estimée : Difficile

Fiches de suivi superviseur

- Cette route aura une méthode GET pour afficher la vue de la "fiche de suivi superviseur" aura une méthode POST afin d'aller récupérer la route "Tableau de bord vue par un joueur" de tous les joueurs.
- Pour cela, on demandera au superviseur de rentrer un paramètre qui sera le rôle de l'acteur que veut regarder le superviseur et d'aller chercher la route "Fiche de suivi" du joueur qu'il aura choisi.
- Cette route sera accessible via la route "Tableau de bord du superviseur" et amènera sur les routes "Fiche de suivi" des joueurs.

Tableau de bord d'une partie vue par le joueur

La route aura une méthode GET permettant de visualiser le tableau de bord du joueur ainsi que d'afficher la vue du formulaire de l'étape en cours. Elle renvoie également toutes les informations nécessaires à l'affichage de la fiche de suivi du joueur.

La méthode POST permet d'envoyer les informations remplies par l'utilisateur dans le formulaire. Le formulaire change selon l'étape de la partie :

- Pendant la première étape, il permet de changer l'attribut seen_by_actor des commandes reçues par le joueur.
- Pendant la deuxième étape, il crée, pour chaque client du joueur, une instance de Delivery_command de type "delivery" avec la quantité livrée.
- Pendant la troisième étape, la route vérifie que le joueur a bien renseigné la bonne valeur de stock.

- Enfin, pendant la quatrième étape, la route crée une instance de Delivery_command de type "command" pour chaque fournisseur du joueur.

Une vérification est effectuée à chaque formulaire pour que les données de la base de données restent cohérentes (aucune commande supérieure au stock du joueur, ...).

Une fois un formulaire validé, la route affiche au joueur le formulaire de l'étape suivante ou une page d'attente des autres joueurs.

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.

Charge estimée : Difficile

Terminer la partie (automatique) - voir méthode post du tableau de bord vue par un superviseur :

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.

La méthode GET permet également de remettre l'état du joueur associée à False grâce à la clé "is_ready" de la table Player associée à la clé "firm_name" du joueur.

Pour cela, il y aura deux conditions nécessaires :

- Pour les joueurs de la partie, il faudra que la clé "round" et la clé "total_round" de la table "Game" pour name_supply_chain ou bien que le superviseur termine la partie. A ce moment, le joueur sera amené 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. Les valeurs seront déjà actualisées grâce à la méthode POST de la route "Tableau de bord vue par un joueur".

Cette route pourra être accédée par la route "Tableau de bord vue par un joueur".

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 avec les mises à jour grâce à la méthode POST de "Tableau de bord vue par le joueur" qui permet de récupérer les commandes envoyés et possiblement de mettre à jour la table de suivi en même temps.

Pour accéder à cette route, il faut passer soit par la route “Tableau de bord vue par le joueur”, soit par la route passer une commande, soit par la route envoyer une commande.

Cette route peut amener sur :

- la route “tableau de bord du joueur”
- la route “Règles du jeu”

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 “quantity”, et “id_client” de la table “Command” .En effet, pour calculer les pénalités de chaque joueur, on va faire la somme sur chaque clé “quantity” et ‘quantity” avec le type “command” des lignes qui possèdent les clés "id game","id_client" ayant un “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_initial” 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” et la clé “total_round” de la table “Supply_chain” pour le “name_supply_chain” soient égales.
- 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 “Terminer la partie” pour les joueurs et la route “Tableau de bord vue par le superviseur” via la méthode POST.

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