

Application BEER GAME

- *cahier des charges* -

Chapitre 1 : Contexte et périmètre du projet	2
Contexte	2
Périmètre	2
Chapitre 2 : Système existant et objectifs du projet	3
Système existant	3
Objectifs du projet	4
Règle du Jeu	5
Explication du jeu	5
Détail d'un Round	5
Chapitre 3 : Caractéristiques fonctionnelles du futur système	8
Les données	8
Les acteurs et cas d'utilisation du système cible	10
Hypothèses	10
Définition des processus	10
Fonctionnement souhaité	11
Fonctionnement dégradé :	13
Chapitre 4 : Caractéristiques non fonctionnelles du futur système	15
Interface Homme-Machine	15
Contraintes d'exploitations	15
Contraintes de développement	16
Délais de réalisation	16
Glossaire:	17

Chapitre 1 : Contexte et périmètre du projet

1. Contexte

Dans l'idée d'une pédagogie plus dynamique, le projet d'aujourd'hui a pour but de rendre un cours sur le fonctionnement d'une chaîne logistique plus interactif. Pour ce faire, nous souhaitons utiliser le jeu inventé par le MIT nommé: "Beer Game". C'est un jeu qui simule une chaîne d'approvisionnement où chacun des maillons est le client de la personne précédente et le fournisseur de la personne suivante (sauf pour les premiers et derniers maillons qui sont respectivement les fournisseurs initiaux et les clients). Ce jeu devra être lancé par des enseignants à destination d'étudiants qui se forment à la supply chain et doivent leur permettre de comprendre l'effet coup de fouet que peuvent rencontrer les chaînes logistiques principalement. Ils vont devoir essayer d'adapter leur manière de jouer et de prendre des décisions de manière autonome afin d'assurer un approvisionnement qui soit adapté aux commandes futures et au stock. Il ne faudra donc pas avoir d'indices concernant les futures commandes de chaque joueur dans le jeu.

L'informatisation du jeu facilitera la mise en place et le déroulement du jeu. De plus, l'application sera disponible et adaptée au travail à distance

Nous souhaitons ainsi numériser ce jeu et donc développer une application web permettant aux élèves de pouvoir jouer au "Beer Game".

2. Périmètre

- Les objectifs

Les buts du projet sont donc de créer une application web accessible sur tous les navigateurs qui simule une chaîne logistique. Cette application devra être open-source et le code sera à disposition du client. Enfin, l'application devra pouvoir être déployée facilement et sur tout serveur.

- Les livrables

Premièrement le développeur devra fournir, au bout de deux semaines, des propositions d'interfaces, une modélisation métier et un début de spécification d'implémentation.

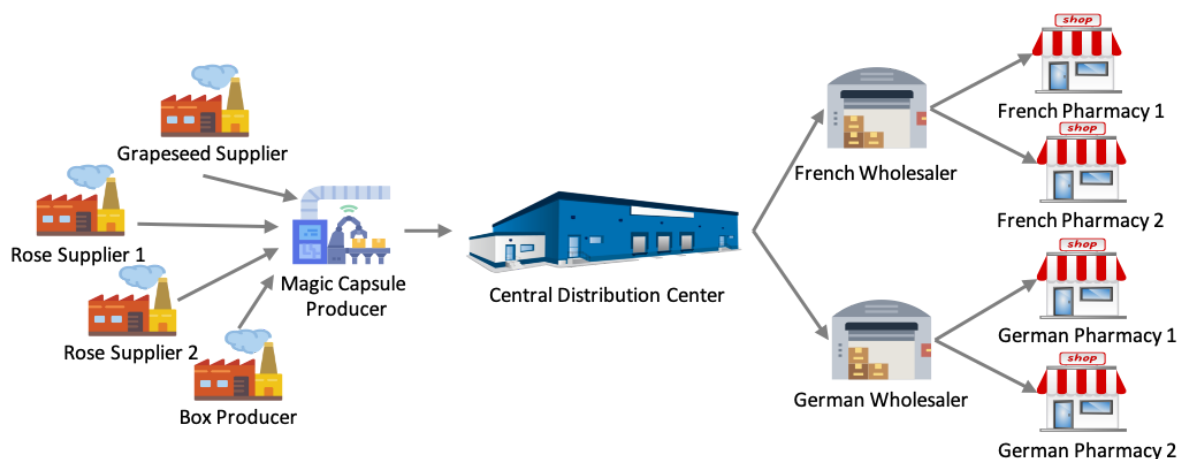
Puis, au terme des 3 mois du projet, le livrable finale exigé sera une application web permettant de répondre aux différents besoins qui seront spécifiés plus tard et le code de l'application afin de pouvoir permettre au client une autonomie sur la maintenance de l'application.

Enfin, la partie déploiement de l'application n'est pas de la responsabilité du développeur, il devra simplement s'assurer que l'application est déployable sur tout serveur.

Chapitre 2 : Système existant et objectifs du projet

1. Système existant

Le SI actuellement en place est un jeu en présentiel qui utilise des jetons pour modéliser les mouvements de matières entre les différents joueurs et où les données sont enregistrées à la main sur des feuilles de papier. La chaîne logistique utilisée comme support est la suivante :



Comme indiqué sur ce schéma un joueur peut prendre un rôle parmi ceux sur la chaîne logistique à l'exception de Box Producer, Grapeseed Supplier et Rose Supplier.

Ensuite a lieu le déroulement de la partie qui se divise en plusieurs semaines fictives réparties de la manière suivante:

- Lundi : les joueurs découvrent les commandes auxquelles ils doivent répondre cette semaine.
- Mardi : les joueurs décident de la quantité qu'ils vont effectivement livrer pour toutes les commandes qu'ils ont reçues. Ils placent alors les jetons dans des gobelets. Les quantités livrées et les restes à livrer sont alors enregistrés sur une fiche de suivi.
- Mercredi : les joueurs réceptionnent les produits que leur fournisseur leur a livré. Les quantités reçues sont enregistrées sur leur fiche de suivi et dans leur stock.
- Jeudi : les joueurs passent une ou plusieurs commandes à un ou plusieurs fournisseurs. Les quantités commandées sont enregistrées sur la fiche de suivi.
- Vendredi : les commandes effectuées mardi sont envoyées (les joueurs font respectivement passer le gobelet préparé mardi au joueur qui représente l'entité suivante de la chaîne logistique modélisée par le jeu).

De plus il existe différentes restrictions dans les règles du jeu :

- Ne pas regarder les actions des autres joueurs
- Ne pas effectuer une action à un jour qui est différent de celui prévu (il est interdit par exemple de commander un autre jour que jeudi)

Malheureusement cette manière de procéder présente plusieurs inconvénients. Premièrement le jeu est difficile à mettre en place et nécessite de l'organisation. Aussi il est

difficile pour le superviseur de la partie de pouvoir suivre parfaitement toutes les actions de chacun des groupes. Enfin dans cette version il est facile de tricher (même sans le vouloir) en jetant un coup d'œil sur les gobelets des autres joueurs par exemple et donc en pouvant anticiper les commandes qui arriveront ensuite.

Il est donc envisagé, pour pallier ces problèmes, de créer une application web.

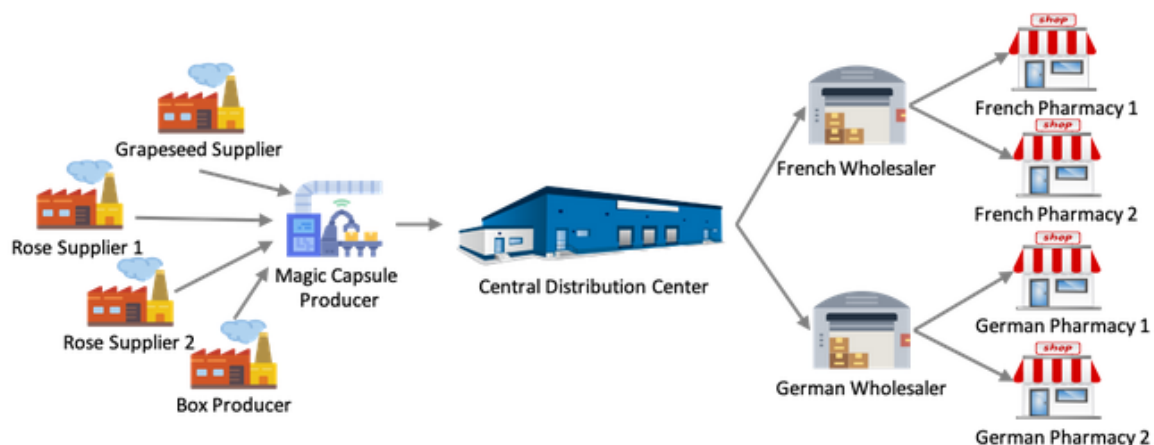
Pour rentrer un peu plus dans les détails, le projet a pour objectif de simuler une chaîne logistique et d'apporter une vue globale sur cette simulation. L'objectif principal de l'application est pédagogique, elle doit permettre à des étudiants ou à des professionnels en formation de comprendre le principe de chaîne logistique. Ajouté à cela, cette application devrait permettre de faciliter la réalisation de la simulation. En effet, au lieu de se déplacer et récupérer les gobelets de tous les joueurs, les élèves n'auront plus qu'à rentrer leurs données dans le logiciel qui transmettra les informations instantanément aux enseignants

2. Objectifs du projet

De plus, cela facilitera également l'organisation, puisque seuls des ordinateurs seront nécessaires contrairement au cas il faudrait utiliser des post-it.

Le client souhaiterait avoir une application Web permettant la simulation d'un jeu de Beer Game et les spécifications techniques de l'application développée. Un guide d'utilisation ou des indications d'aide seront disponibles sur l'application. Ce jeu devra être open-source.

Tout d'abord commençons par introduire la chaîne logistique qui sera modélisée dans le jeu: c'est la même chaîne que présentée précédemment. Elle doit pouvoir permettre de faire jouer 8 joueurs en même temps dont les rôles sont répartis de "Magic Capsule Producer" jusqu'aux "French/German Pharmacy". C'est une chaîne logistique qui modélise l'approvisionnement en médicaments de pharmacie.



3. Règle du Jeu

a. Explication du jeu

Plusieurs rounds vont être réalisés représentant ainsi la notion du temps. Ces rounds vont permettre aux joueurs de réaliser leurs commandes et de transmettre leurs ventes aux clients.

Les joueurs auront le rôle à la fois de fournisseur et de clients. En effet, ils vont devoir commander des ressources pour les revendre. La transformation de tous les produits étant considérée instantanée on ne considérera aucune contrainte de temps liés à une quelconque manipulation du produit. Un retard dans la transformation du produit sera donc impossible. Les produits seront donc forcément disponibles le jour de la livraison.

Ensuite, les clients en fin de chaîne seront représentés par le superviseur: c'est lui qui transmettra les commandes aux joueurs : il symbolise donc la demande du marché.

Les joueurs quant à eux, n'auront la possibilité de transmettre leurs commandes qu'aux joueurs qui leur seront associés dans la chaîne logistique et qui seront leurs clients.

De plus, le superviseur aura la possibilité de visualiser les graphiques d'évolution des états de chaque joueur. Ces graphiques permettent de visualiser les commandes restantes à livrer et les pénalités liées aux excès de stock. Les joueurs subissent des pénalités quand ils n'arrivent pas à répondre entièrement au besoin de leurs clients ou s'ils ont un excès de stock. Ces pénalités sont proportionnelles au nombre de produits stockés/non livrés. La valeur de chacune des pénalités est à définir à la création de la partie.

Les joueurs n'ont pas accès au montant des pénalités qu'ils auront accumulées ni au montant des pénalités des autres joueurs pendant le déroulement de la partie. Ils auront accès à leurs informations personnelles à la fin de la partie. ainsi que les actions réalisées par les autres joueurs pendant la partie.

Pour permettre aux joueurs d'avoir une prise de recul sur leurs choix stratégiques, ils doivent remplir une fiche de suivi contenant les commandes à livrer, la livraison apportée aux clients, le reste à livrer pour visualiser les pénalités, la quantité reçue par le client, le stock restant, et la commande que le joueur va demander au client.

Une fois que les joueurs ont fini de remplir la fiche de suivi et donc finaliser toutes leurs actions, le round suivant est lancé.

A la fin de la partie, le superviseur a la possibilité d'afficher le graphe d'évolution sur les écrans associé à chaque joueur.

b. Détail d'un Round

Avant de commencer, il faudra que le superviseur choisisse le délai de transport, le délai de passage des commandes, les quantités de matières en cours de livraison et les stocks initiaux.

Round 1

Ici, les joueurs pourront passer une commande à leurs fournisseurs mais n'auront pas de commande à livrer.

Étant donné le contexte du jeu, l'ordre des tâches réalisables devra suivre un certain ordre selon l'ordre des actions dans les jours de la semaine. Pour ce round, les joueurs n'auront qu'à commander ce dont ils ont besoin pour la semaine suivante

Rounds $n \geq 1$

Dans ce cas, l'ordre temporel des actions, dépendant de chaque jour de la semaine, va être important.

La première étape du round consiste à prendre connaissance des quantités qu'ils devront livrer à leurs clients. Ils reçoivent un bon de commande de chaque client qui leur est associé sur la chaîne logistique. Ce bon de commande contient la quantité souhaitée par un client. Afin de passer au round suivant, les joueurs doivent remplir la fiche de suivi avec les montant de chaque bon de commande.

Ensuite, les joueurs vont répartir leurs stocks à livrer aux différents clients en notant la quantité qu'ils vont leur fournir sur la fiche de suivi.

De plus, les joueurs vont recevoir leurs commandes : ce qui va se refléter par une mise à jour automatique de la fiche de suivi dans la colonne "des réceptions de commandes" mais également dans l'interface de visualisation des stocks.

Ajoutée à cela, les joueurs vont pouvoir réaliser leurs commandes pour la semaine suivante. Étant donné qu'il n'y a aucun indice ici sur les commandes de la semaine suivante, cette action pourra se faire en même temps que les autres actions qu'elles soient automatiques ou manuelles sur la fiche de suivi.

Enfin, la dernière action à réaliser consister à envoyer les commandes à leurs clients. Cette action doit pouvoir se faire automatiquement lorsque tous les joueurs auront finalisé leurs actions.

Une fois les actions terminées, ils pourront voir les nouvelles données de la semaine suivante.

Entre les Rounds :

La mise à jour des graphiques de pénalités, de quantités à livrer et de stock restant devra être faite, ainsi que les stocks des joueurs.

La fiche de suivi de tous les joueurs devra être mise à jour.

Dans cette partie, il est nécessaire de décrire en détail le besoin et ce qu'on attend :

→ Qu'est-ce qu'on attend pour l'interface?

Une interface devrait permettre de pouvoir accéder facilement aux différentes commandes nécessaires au bon fonctionnement du jeu. Le jeu doit pouvoir s'utiliser de manière aisée, de manière intuitive pour permettre aux utilisateurs de mieux se concentrer sur la capacité à gérer leurs entreprises et leurs commandes. La forme est ainsi laissée cours à l'imagination du Maître d'Oeuvre, mais doit mettre en évidence les fonctionnalités.

L'interface utilisateur doit donner accès au fiche de suivi, à la réalisation de commande et de livraison. L'interface Administrateur doit pouvoir visualiser les graphiques et la fiche de suivi des joueurs.

→ Qu'est-ce qu'on attend pour les fonctionnalités de chacun des joueurs

L'utilisateur pourra signaler à l'ensemble des joueurs qu'il a finalisé sa tâche. Une fois que l'ensemble des joueurs auront finalisé leurs tâches , l'application devra mettre à jour les fiches de suivi, les graphiques d'évolution de l'administrateur, et empêcher les joueurs de remplir les lignes des round précédents de la fiche de suivi.

Chapitre 3 : Caractéristiques fonctionnelles du futur système

1. Les données

Fichier de suivi des données logistiques :

	étape 1	étape 2	étape 2	étape 3	étape 4	étape 5
Date (d)	Demande (D)	Livraison Client (LC)	Reste à Livrer (RaL)	Reception (R)	Stock (S) $=S(-1)-LC+R$	Commande Fournisseur (CF)
0	30	30	0	30	90	30
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

Ajouter une colonne « cumul des commandes fournisseur attendues » (CCA)
 Initialement $CCA = 3*CF$
 Ensuite $CCA = CCA(-1) + CF - R$

Les données stockées pour chaque acteur, pour chaque semaine et pour chaque partie sont :

- Le stock après réception des quantités envoyées par les fournisseurs :
 - Cette donnée est définie au début de la partie par le superviseur puis mise à jour quand des matières sont livrées par un fournisseur ou envoyées à un client.
 - Le stock de produit d'un acteur est un nombre entier positif ou nul.
 - Cette donnée est accessible par l'acteur et le superviseur.
 - Cette donnée sera calculée et remplie par les joueurs, puis validée/vérifiée par le système.
- Les quantités de produit commandés à ses fournisseurs :
 - Cette donnée est initialisée à 0 au début de la partie puis mise à jour quand des produits sont commandés à un fournisseur.
 - La quantité de produit commandé d'un acteur est un nombre entier positif ou nul.
 - Cette donnée est accessible par l'acteur, le fournisseur cible de la commande et le superviseur.
 - Cette donnée sera rentrée par le joueur pour qu'il puisse choisir les quantités qu'il voudrait recevoir de chaque fournisseur auquel le joueur est lié
- Les quantités en cours de livraison pour chacun de ses clients :

- Cette donnée est initialisée par le superviseur au début de la partie puis mise à jour quand une commande est envoyée à un client.
 - La quantité en cours de livraison d'un acteur est un nombre entier positif ou nul.
 - Cette donnée est accessible par l'acteur, et le superviseur.
 - Le client cible des commandes verra les quantités envoyées par chacun de ses fournisseurs dès lors qu'il les aura reçues.
 - Cette donnée sera calculée et remplie par les joueurs, puis validée/vérifiée par le système.
- Les quantités qui vont être envoyées à la fin de la semaine pour chacun de ses clients :
 - Cette donnée est initialisée par le superviseur au début de la partie puis mise à jour quand une commande est envoyée à un client.
 - La quantité en cours de livraison d'un acteur est un nombre entier positif ou nul.
 - Cette donnée est accessible par l'acteur, le client cible de la commande et le superviseur.
 - Cette donnée remplie par les joueurs, puis validée/vérifiée par le système pour voir la cohérence avec les quantités en stock
- Les restes à livrer pour chacun de ses clients :
 - Cette donnée est initialement à 0 au début de la partie puis mise à jour quand une commande est envoyée à un client.
 - Le reste à livrer d'un acteur est un nombre entier positif ou nul.
 - Cette donnée est accessible par l'acteur et le superviseur.
 - Cette donnée sera calculée et remplie par les joueurs, puis validée/vérifiée par le système.
- Le cumul des pénalités de stocks et de ruptures :
 - Cette donnée est initialisée à 0 au début de la partie puis mise à jour à chaque fin de semaine de jeu.
 - Le cumul des pénalités d'un acteur est un nombre positif ou nul.
 - Cette donnée est accessible par le superviseur.
 - Cette donnée sera calculée automatiquement par le système car les joueurs n'ont pas accès aux pénalités de stock et de ruptures qu'ils ont reçues

Les données stockées pour chaque partie sont :

- Le type d'acteur :
 - Cette donnée est une chaîne de caractères permettant d'identifier si l'acteur est contrôlé par un joueur ou une intelligence artificielle.
 - Les valeurs acceptées sont 'joueur' ou 'ia'. La valeur par défaut est 'ia'.
 - Cette donnée est modifiée quand un joueur prend le contrôle d'un acteur de la chaîne logistique.
- Le nom de la partie :
 - Cette donnée est déterminée par le superviseur quand il crée la partie.
 - L'identifiant d'un acteur est une chaîne de caractères.

- Cette donnée est accessible par le superviseur et les joueurs.
- La date de création de la partie :
 - Cette donnée est déterminée automatiquement à la création de la partie.
 - Cette donnée est accessible par le superviseur.
- La date de fin de la partie :
 - Cette donnée est déterminée automatiquement à la création de la partie.
 - Cette donnée est accessible par le superviseur.

2. Les acteurs et cas d'utilisation du système cible

Les joueurs seront des élèves en école d'ingénieur ou des professionnels en formation. Ils jouent un acteur de la chaîne logistique. Néanmoins, cette application pourra être utilisée hors de ce cadre également. En effet, elle pourra être destinée à des individus qui aimeraient se former à la supply chain ou à des étudiants provenant de filière leur apprenant à la maîtriser.

Le superviseur d'une partie de jeu est un enseignant. Il doit avoir accès aux données des résultats ainsi que des graphiques résumant les actions des utilisateurs tout au long de la partie.

Cette application pourra être étendue par exemple à des nouveaux entrepreneurs d'industries pour qu'il puisse comprendre les dangers qu'ils peuvent rencontrer lors de la commande de stock ou bien lors des retards de livraison.

3. Hypothèses

- Les acteurs en début de chaîne peuvent commander une infinité de produits.
- Les matières fournies au premier acteur de la chaîne sont vendues en fin de chaîne sans transformation par les acteurs.
- Il n'y a pas de communication entre les joueurs.
- Les flux monétaires ne sont pas pris en compte.
- Les pénalités de stock et de rupture ne sont pas accessibles aux acteurs pendant la partie.
- Rejoindre une partie ne nécessite pas la création d'un compte.
- Les données de chaque partie sont conservées 3 mois ou jusqu'à suppression par un administrateur de l'application.

4. Définition des processus

Le but du projet est de produire une application permettant de simuler le jeu du Beer Game. Pour cela, nous avons divisé le déroulement d'une partie de Beer Game en 3 stades:

- La création

- La partie en elle-même
- L'analyse de fin de partie

Pour chacun de ces stades, nous avons identifié les fonctions nécessaires au déroulement d'une partie et des fonctions optionnelles demandées par le client.

a. Fonctionnement souhaité

Fonctions Principales (nécessaire au déroulement d'une partie) :

Stade 1 : Création d'une partie

- Créer un salon de jeu avec une chaîne logistique fixe :

En tant que superviseur, je souhaite créer une partie de Beer Game afin de simuler une chaîne logistique.

- Inviter un joueur dans le salon de jeu :

En tant que superviseur, je souhaite inviter un joueur dans le salon de jeu afin qu'il puisse participer à la chaîne logistique.

- Choisir un acteur de la chaîne logistique :

En tant que joueur, je souhaite choisir un acteur de la chaîne logistique afin de participer à la partie.

- Lancer la partie :

En tant que superviseur, je souhaite pouvoir lancer la partie afin de débiter la simulation.

Stade 2 : Partie

- Visualiser les commandes reçues :

En tant que joueur, je souhaite voir les commandes que je reçois afin de pouvoir choisir la quantité à livrer.

- Passer une commande au fournisseur :

En tant que joueur, je souhaite pouvoir passer une commande à un de mes fournisseurs afin de pouvoir réapprovisionner les stocks.

- Effectuer une livraison :

En tant que joueur, je souhaite pouvoir livrer un de mes clients afin de satisfaire la commande qu'il a passée.

- Passer à l'étape de jeu suivante :

En tant que superviseur, je souhaite pouvoir passer à l'étape de jeu suivante afin d'avancer dans la simulation.

- Remplir les restes à livrer :

En tant que joueur, je souhaite pouvoir remplir la colonne des commandes restantes à livrer sur la fiche de suivi afin d'estimer les pénalités de rupture qui me seront appliquées.

- Rentrer les stocks :

En tant que joueur, je souhaite pouvoir remplir la colonne des stocks après les réceptions des commandes fournisseurs sur la fiche de suivi afin d'estimer les pénalités de stocks qui me seront appliquées.

Stade 3 : Analyse

- Afficher les graphiques d'évolution des pénalités et des stocks de tous les acteurs :

En tant que superviseur, je souhaite pouvoir afficher les graphiques de fin de partie afin de montrer le fonctionnement global de la chaîne logistique aux joueurs.

Autres fonctions :

Stade 1 : Création de la partie

- Choisir l'état initial des stocks, livraison, délais de livraison et délais de communication :

En tant que superviseur, je souhaite pouvoir modifier les paramètres globaux de la chaîne logistique afin de voir l'impact de ces paramètres sur le déroulement du jeu.

- Attribuer une Intelligence Artificielle aux acteurs qui n'ont pas de joueurs :

En tant que superviseur, je souhaite affecter une intelligence artificielle aux acteurs qui ne sont pas contrôlés par un joueur afin de ne pas avoir d'acteur de la chaîne logistique sans contrôleur.

- Choisir le nombre de semaines à jouer :

En tant que superviseur, je souhaite pouvoir définir le nombre de semaines à jouer afin de pouvoir adapter le jeu aux durées d'enseignement disponible.

Stade 2 : Partie

- Visualiser les stocks :

En tant que joueur, je souhaite pouvoir consulter l'état de mes stocks afin de pouvoir passer des commandes à mes fournisseurs adaptées.

- Afficher la fiche de suivi :

En tant que joueur, je souhaite pouvoir visualiser la fiche de suivi afin de pouvoir connaître les produits qu'il me reste à livrer et les commandes que j'ai passé à mes fournisseurs.

- Remplir la fiche de suivi :

En tant que joueur, je souhaite pouvoir remplir la fiche de suivi afin de conserver un historique des commandes que j'ai livrées et que j'ai reçues.

- Voir toutes les fiches de suivi :

En tant que superviseur, je souhaite avoir accès à toutes les fiches de suivi afin de pouvoir suivre l'avancement de la partie.

- Voir les commandes en cours de livraison :

En tant que superviseur, je souhaite pouvoir visualiser les commandes en cours de livraison afin d'avoir accès aux quantités de produits en cours de livraison.

- Voir les graphiques d'analyses :

En tant que superviseur, je souhaite avoir accès aux graphiques d'analyse en cours de partie afin d'alerter un groupe si ses pénalités deviennent trop importantes.

- Pouvoir prendre le contrôle d'un acteur :

En tant que superviseur, je souhaite pouvoir prendre le contrôle d'un acteur afin de faire avancer la partie si un groupe n'arrive plus à accéder à l'application.

b. Fonctionnement dégradé :

On appelle fonctionnement dégradé, le minimum acceptable souhaité dans le cas où il serait impossible de mener à bien le projet souhaité initialement (et énoncé précédemment).

Fonctions Principales (nécessaire au déroulement d'une partie) :

Dans ce modèle-ci, nous ferons appel tout d'abord à toutes les fonctions principales définies dans la définition des processus.

Excepté la fonction : *"Permettre au superviseur de pouvoir jouer un ou plusieurs rôles en cas de manque de joueurs"* qui est remplacée par :

- Le nombre de joueur est fixé à 8 :

En tant que superviseur je ne peux assurer le rôle d'un ou plusieurs joueurs et je dois m'assurer que tous les acteurs de la chaîne participent, donc qu'il y a 8 joueurs.

Autres fonctions :

Stade 1 : Création de la partie

- L'état initial des stocks, livraison, délais de livraison et délais de communication sont fixes

En tant que superviseur, je ne peux pas modifier les constantes de partie.

- Le nombre de semaines/rounds est fixé à 10 : pour toutes les parties

En tant que superviseur, je souhaite pouvoir définir le nombre de semaines à jouer afin de pouvoir adapter le jeu aux durées d'enseignement disponible.

Stade 2 : Partie

- Visualiser les stocks :

En tant que joueur, je souhaite pouvoir consulter l'état de mes stocks afin de pouvoir passer des commandes à mes fournisseurs adaptées.

- Afficher la fiche de suivi :

En tant que joueur, je souhaite pouvoir visualiser la fiche de suivi afin de pouvoir connaître les produits qu'il me reste à livrer et les commandes que j'ai passé à mes fournisseurs.

- Remplir la fiche de suivi :

En tant que joueur, je souhaite pouvoir remplir la fiche de suivi afin de conserver un historique des commandes que j'ai livrées et que j'ai reçues.

- Voir toutes les fiches de suivi :

En tant que superviseur, je souhaite avoir accès à toutes les fiches de suivi afin de pouvoir suivre l'avancement de la partie.

- Voir les commandes en cours de livraison :

En tant que superviseur, je souhaite pouvoir visualiser les commandes en cours de livraison afin d'avoir accès aux quantités de produits en cours de livraison.

- Voir les graphiques d'analyses :

En tant que superviseur, je souhaite avoir accès aux graphiques d'analyse en cours de partie afin d'alerter un groupe si ses pénalités deviennent trop importantes.

Chapitre 4 : Caractéristiques non fonctionnelles du futur système

1. Interface Homme-Machine

Ce support permet d'inviter des acteurs à participer au jeu. Une fonction d'export des résultats des parties précédentes au format .csv doit être présente sur ce support visuel. Enfin, l'utilisateur peut voir l'ensemble des parties qu'il a créées et qui sont en cours d'exécution.

Un support de visualisation des commandes. Ce support affiche l'ensemble des commandes de l'acteur (bons de commandes passés à ses fournisseurs et commandes en cours de livraison pour ses clients). C'est sur ce support visuel que sont effectuées les commandes décrites dans la partie 2.1 de ce document. Il doit permettre la sélection d'un fournisseur et la quantité de produits à commander.

La fiche de suivi doit être visualisable et remplissable par un joueur. Le remplissage de la fiche de suivi doit être vérifié par le système et doit indiquer à l'utilisateur s'il a fait une faute dans le remplissage.

Une analyse de fin de partie doit être accessible par le superviseur. Cette analyse contient le graphique d'analyse des pénalités et d'évolution des stocks pour chaque équipe.

2. Contraintes d'exploitations

- Volumétrie :

Le système d'information devra pouvoir accueillir au minimum l'ensemble des élèves d'une classe de génie industriel simultanément soit environ 60 personnes. Aussi, par mesure de sécurité et car en L3 certains cours portent sur le génie industriel, il sera demandé que le système d'information puisse supporter un nombre minimum de 100 connexions simultanées. La volumétrie des données stockées et échangées reste, quant à elle, à être définie.

- Performances :

Le temps de réponse envisagé doit être inférieur ou égal à 1 seconde.
Le temps de création d'une partie doit être inférieur à 5 secondes.

- Disponibilité :

L'application devra être disponible 24/7

3. Contraintes de développement

Le projet sera dans un premier temps hébergé sur une machine de l'équipe via un serveur local. Dans un second temps, il est prévu que cet outil soit déployé sur les serveurs de l'école une fois son prototype validé mais la partie déploiement ne fait pas partie des exigences demandées et est hors scope du présent projet.

Concernant le développement de l'application, les langages utilisés seront :

- Systèmes de gestion de base de données : PostgreSQL
- Développement BackEnd : Python avec l'utilisation du framework Flask
- Développement FrontEnd : HTML, CSS, Javascript

4. Délais de réalisation

Le projet durera 3 mois et devra être livré avant le 25/02/2022.

Glossaire:

Fiche de suivi : tableau qui recense les données des commandes clients et fournisseurs et l'état des stocks

Open source: Se dit d'un logiciel dont le code source est libre d'accès, réutilisable et modifiable

Pénalités : Les pénalités sont des malus appliqués au joueur qui est visible par le superviseur

Superviseur: créateur de la partie. Il est par extension dans le cadre de ce cahier de charges l'enseignant ou bien le formateur.

Administrateur: Personne en charge de la maintenance de l'application

Supply Chain: chaîne d'approvisionnement également appelée chaîne logistique

Acteur: représente un maillon de la chaîne logistique

Joueur: entité qui prend le contrôle d'un acteur (personne, groupe de personne ou intelligence artificielle)