Projet intégrateur en Gestion de Données (BDi)
M1 MIAGE - **Groupe 7**

BONNIN Alice - BOU SERHAL Jean - CHOLVY Jordan - IERMOLI Quentin

Lien GitHub Front : https://github.com/CHOLVY-M1-Miage/projectBdiGroupe7ProdFront
Lien GitHub Back : https://github.com/CHOLVY-M1-Miage/projectBdiGroupe7ProdBack
Serveur FRONT: http://129.88.210.39:4200/
Serveur BACK: http://129.88.210.39:8080/
(La VM semble s'éteindre vers minuit, tous les jours.)

Si vous cherchez à lancer l'application via le code sur GitHub, il est essentiel de set une variable d'environnement `GOOGLE_APPLICATION_CREDENTIALS` avec la valeur du path pointant vers `src\main\resources\gromedg7-firebase-adminsdk-uina0-445b07b14e.json`

Il est également indispensable de configurer Oracle pour l'accès à la base ;
Le fichier .env doit contenir :

```
JDBC_DATABASE_URL=jdbc:oracle:thin:@im2ag-oracle.univ-grenoble-alpes.fr:1521:IM2AG
JDBC_DATABASE_USER=cholvyj
JDBC_DATABASE_PWD=c93788da2f

GOOGLE_APPLICATION_CREDENTIALS="src\main\resources\gromedg7-firebase-adminsdk-uina0-445b07b14e.json"
```

(Ce dernier n'est pas push sur Git pour des soucis de sécurité.)

Identifiants d'un utilisateur de l'application GroMed :
E-mail U1 : robert.morane@outlook.com
Mot de passe U1 : leVra!Her0

Email U2 : toto@tata.fr
Mot de passe U2 : tititi

[Objectifs]
L'objectif du groupe pour ce projet intégrateur, était de réussir le mieux possible la première itération, et ensuite, s'il restait du temps, d'entamer la seconde.
Les objectifs peuvent être résumés ainsi :

- Avoir une base de données solide

- Faire communiquer le back et le front afin de pouvoir réaliser les fonctions suivantes :
    - S'authentifier pour accéder à l'application
    - Rechercher un article à travers l'utilisation des filtres "Nom de médicament", "Molécule", "Fournisseur", "Générique" et "Agrément collectivités".
    - Afficher les informations que nous avons considérées comme importantes : Nom médicament, présentation, substance, fournisseur et prix.
    - Ajouter un article au panier
    - Valider le panier
- Gérer l'aspect transactionnel
- Avoir une IHM respectant la charte graphique et qui soit complète

[BD]

Parmi tout cela, la base de données a bien été établie, l'authentification est disponible et fonctionnelle. Les filtres sont tous fonctionnels, mais les informations ne sont pas toutes affichées : il n'y a que la présentation ainsi que le prix. L'affichage n'est certes pas parfait, mais le fond - c'est-à-dire la recherche de la base et l'envoi des résultats au front - est présent, et nous avons donc préféré investir le reste de notre temps sur d'autres fonctionnalités.

L'ajout d'article est censé produire : lorsque l'utilisateur clique sur un élément du tableau affichant les résultats de la recherche, il est directement ajouté au panier et met à jour le prix total. A cause de plusieurs défauts de communication entre le back et le front, l'ajout d'article n'est fonctionnel que dans le back (via la route, ex : postman).
Valider le panier permet non seulement de gérer le changement d'état d'une commande, mais aussi de modifier le stock logique correspondant au médicament.

En effet, le système de concurrence a été pensé en anticipation de la possible itération 2 :

- Le stock logique diminue lors de la validation du panier (première itération)
- Après l'attente des 30 minutes, la commande est envoyée et le stock physique est mis à jour à ce moment là (itération 2)

Ainsi, si deux utilisateurs ajoutent un même article qui n'existe qu'en un seul exemplaire, c'est le premier utilisateur à valider son panier qui l'obtient. Le deuxième utilisateur devra choisir de changer de présentation, ou d'annuler sa commande.
Dans la théorie, s'il y avait eu le temps ou l'opportunité, un autre système aurait été mis en place, un système avec non pas 2, mais 3 stocks.
Il y aurait eu :

- Le stock physique, qui se met à jour quand la commande est envoyée
- Le stock logique, qui est mis à jour lors de la validation du panier
- Le stock logique intermédiaire, qui est mis à jour lors de l'ajout au panier

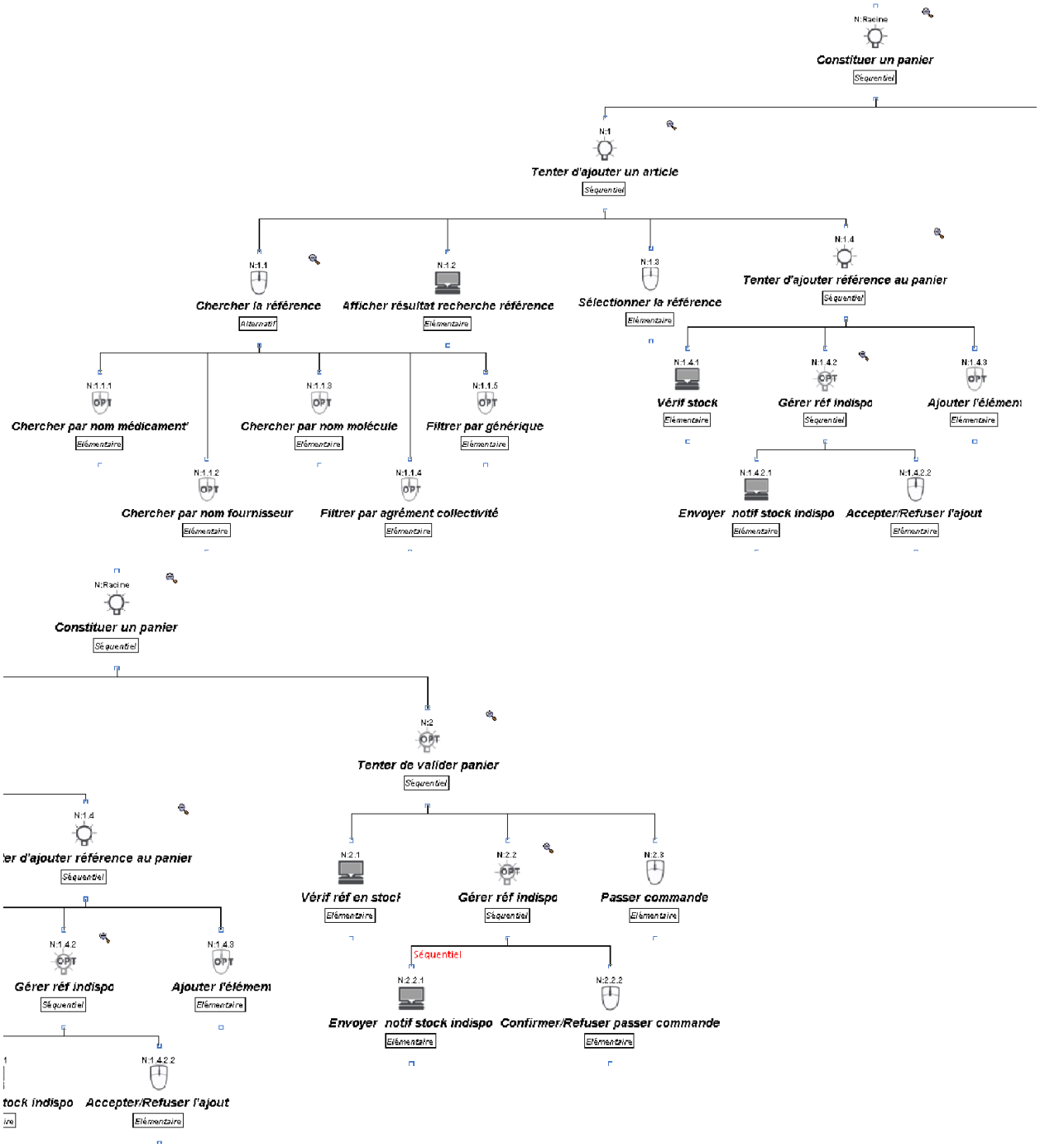
De ce fait, si le stock logique intermédiaire (= stock logique moins somme de l'article dans les différents paniers) est nul, il ne sera pas possible d'en commander, et les utilisateurs ayant déjà le médicament dans leur panier n'auront pas de problèmes pour le commander : il sera considéré comme réservé.

Pour gérer la cohérence des données, nous avons pensé à faire de la validation d'un panier, une seule et unique transaction, qui comprend : la transformation du panier en commande validée, enlever les

références indisponibles et les mettre dans un nouveau panier créé, ainsi que la décrémentation des stocks logiques. Une erreur au niveau de toutes ces requêtes impliquerait un rollback et “ne validerait pas” le panier. La validation d’un panier fonctionne parfaitement dans le back (via la route, ex : postman), mais nous n’avons pas pu mettre en place cette gestion de cohérence par manque de temps, et surtout parce qu’il est difficile de maîtriser des flux de données si disjoints dans si peu de temps, avec tout ce qu’il y a à gérer à côté.
La validation du panier ne marche pas en communication avec le front, à cause d’un défaut de communication de l’ID du panier entre le back et le front.

[IHM]
Concernant l’aspect IHM, nous avons cherché à respecter les couleurs de la **charte graphique** : tous les éléments graphiques de notre interface ont été modifiés pour correspondre aux exigences de couleurs et de polices (et la taille de la police).
Nous avons respecté notre maquette figma, réalisée dans le cadre d’une Conception Centrée Utilisateur (sans forcément pouvoir suivre tout un processus CCU, étant restreints par le temps).
Respect du **modèle de tâches** :
Le modèle de tâches réalisé comprend plus de fonctionnalités que celles implémentées pendant ces 2 semaines (notre 1ère itération).
Nous ne considérons pas l’ajout d’une commande-type au panier *(Ajouter une commande-type)* (2e itération).
Nous ne déterminons pas la quantité *(Déterminer la quantité)* pour l'ajout d’un article au panier *(Ajouter un article)*, puisque nous considérons que cela fait partie de *Modifier un panier*. Modifier un panier n’est pas possible dans notre implémentation puisque la tâche relève de la 2e itération.
Ajouter article au panier a été renommée : *Tenter d’ajouter article au panie*r.
La validation du panier implémentée ne nous permet pas d’en faire une commande-type *(Choisir d’en faire une commande type)*, ni de *Générer une facturation*. Cependant, *Valider panier (*renommée : *Tenter de valider panier)* comprend désormais une vérification des références en stock, une gestion des références indisponibles (avec un envoi de notification “stock indisponible”, et une confirmation/refus de vouloir passer la commande malgré indisponibilité), et *Passer la command*e.

Ce qui donne le modèle de tâches suivant :



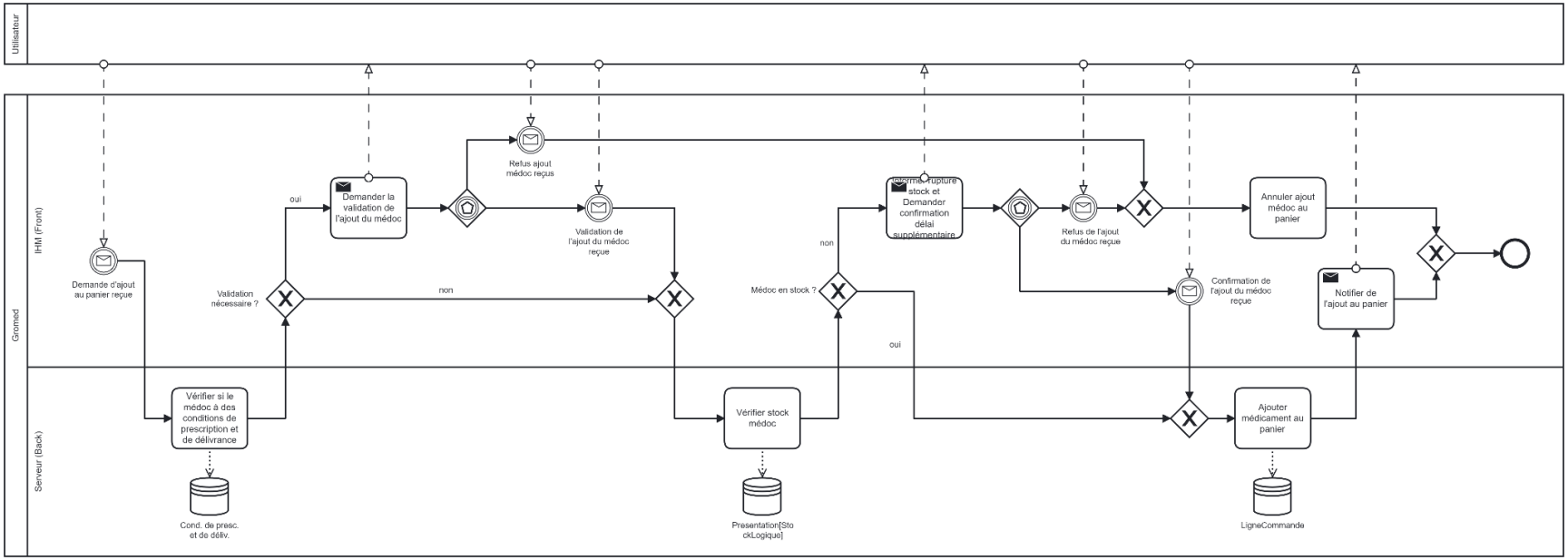
Responsivité :

Côté responsivité, le CSS a été fait avec non pas des tailles paramétrées en pixels, mais en viewport, afin de s'adapter à toutes tailles d'écran en format paysage : tablettes, écrans d'ordinateur classique. Il n'y a cependant pas d'interface considérée comme destinée aux mobiles.

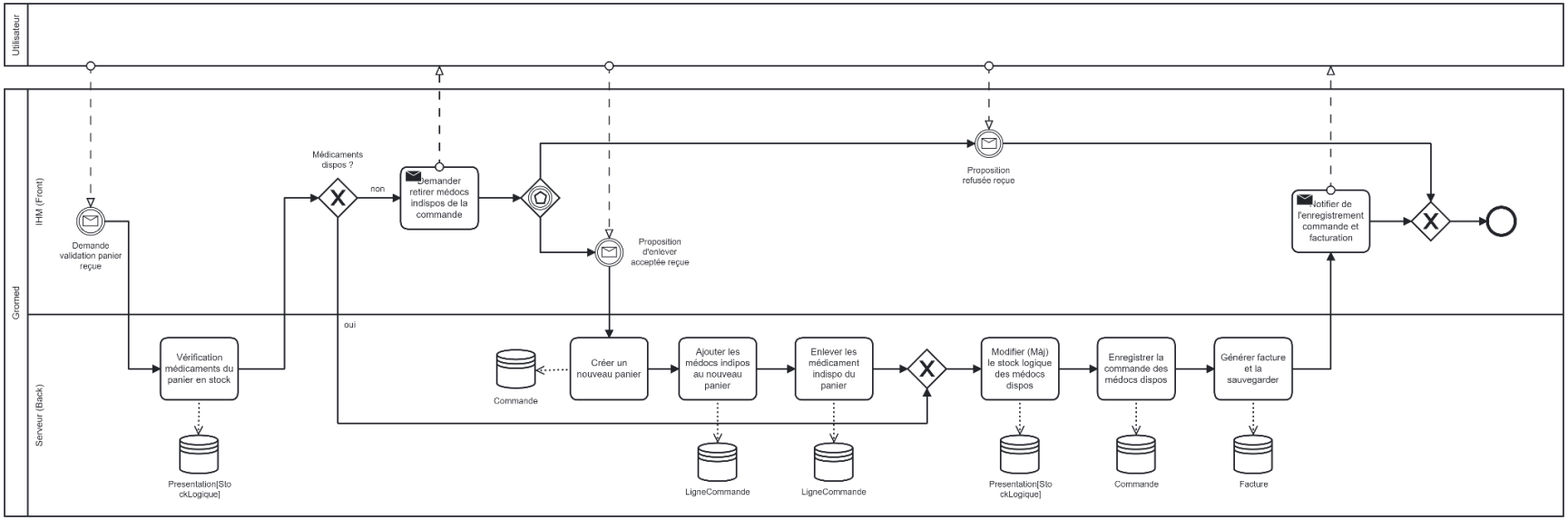
Il manque des retours utilisateurs dans l'interface, par exemple : mot de passe incorrect, utilisateur non reconnu, ajout d'une référence, validation d'un panier… et nous sommes conscients de l'importance de ces retours dans une IHM.

[AFMP]
Pour l'analyse des processus (et leur modélisation), certaines modifications ont été apportées aux BPMN d'ajout au panier et de validation du panier. Ci-dessous les nouvelles versions :

Ajout d’une référence au panier :

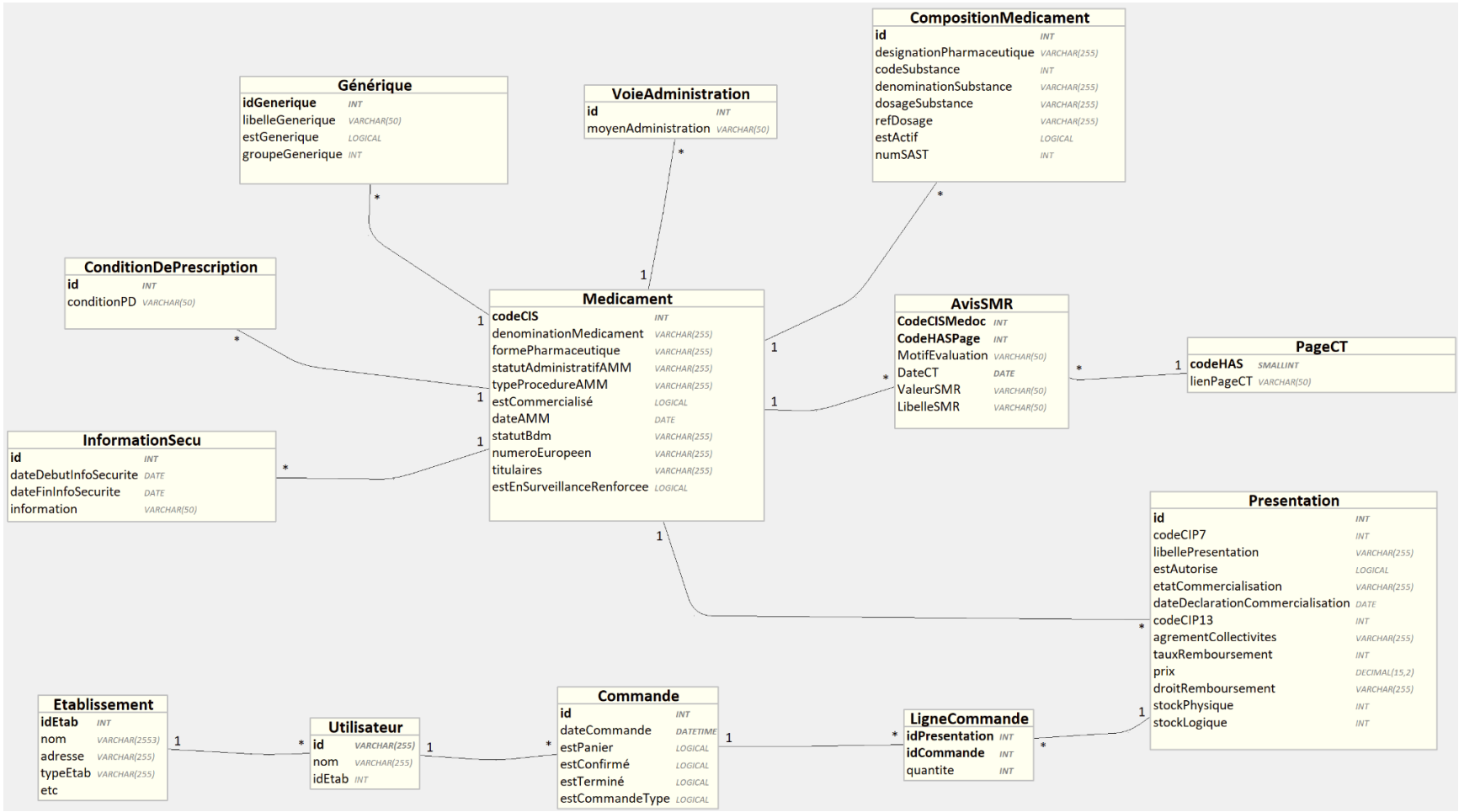


Validation du panier :



[PC]
Par manque de temps, nous n’avons pas de couverture de tests, et nous en sommes conscients (ainsi que de l’importance des tests dans un projet).

Diagramme de classe UML :
Nous avons tout d'abord voulu lier une commande à l'établissement, en pensant que cela nous permettrait de créer plus de scénarios de concurrence de plusieurs utilisateurs (d'un même établissement) sur la même commande. Mais cela ne paraissait pas cohérent vis-à-vis du sujet et du fonctionnement désiré de l'application. Alors nous nous contentons de bien penser la concurrence sur le stock en priorité.


Générique
idGenerique INT
libelleGenerique VARCHAR(50)
estGenerique LOGICAL
groupeGenerique INT
VoieAdministration
id INT
moyenAdministration VARCHAR(50)
CompositionMedicament
id INT
designationPharmaceutique VARCHAR(255)
codeSubstance INT
denominationSubstance VARCHAR(255)
dosageSubstance VARCHAR(255)
refDosage VARCHAR(255)
estActif LOGICAL
numSAST INT
ConditionDePrescription
id INT
conditionPD VARCHAR(50)
Medicament
codeCIS INT
denominationMedicament VARCHAR(255)
formePharmaceutique VARCHAR(255)
statutAdministratifAMM VARCHAR(255)
typeProcedureAMM VARCHAR(255)
estCommercialisé LOGICAL
dateAMM DATE
statutBdm VARCHAR(255)
numeroEuropeen VARCHAR(255)
titulaires VARCHAR(255)
estEnSurveillanceRenforcee LOGICAL
AvisSMR
CodeCISMedoc INT
CodeHASPage INT
MotifEvaluation VARCHAR(50)
DateCT DATE
ValeurSMR VARCHAR(50)
LibelleSMR VARCHAR(50)
PageCT
codeHAS SMALLINT
lienPageCT VARCHAR(50)
InformationSecu
id INT
dateDebutInfoSecurite DATE
dateFinInfoSecurite DATE
information VARCHAR(50)
Presentation
id INT
codeCIP7 INT
libellePresentation VARCHAR(255)
estAutorise LOGICAL
etatCommercialisation VARCHAR(255)
dateDeclarationCommercialisation DATE
codeCIP13 INT
agrementCollectivites VARCHAR(255)
tauxRemboursement INT
prix DECIMAL(15,2)
droitRemboursement VARCHAR(255)
stockPhysique INT
stockLogique INT
Etablissement
idEtab INT
nom VARCHAR(2553)
adresse VARCHAR(255)
typeEtab VARCHAR(255)
etc
Utilisateur
id VARCHAR(255)
nom VARCHAR(255)
idEtab INT
Commande
id INT
dateCommande DATETIME
estPanier LOGICAL
estConfirmé LOGICAL
estTerminé LOGICAL
estCommandeType LOGICAL
LigneCommande
idPresentation INT
idCommande INT
quantite INT

[Conclusion]

1 - Nous avons discuté et pensé aux scénarios de concurrence qui peuvent survenir sur les données de la base, et durant notre travail, nous avons cherché à maintenir le plus possible une cohérence des données. Mais malgré ces efforts, nous n'avons pas pu aller plus loin sur cet aspect, à cause des difficultés auxquelles nous avons fait face.

2 - Ce qui manque dans notre projet pour aboutir à une itération 1 présentable : régler les défauts de communication entre le back et le front et ceci n'est qu'une question de gestion des flux d'objets JSON que nous n'avons pas pu perfectionner par manque de temps.

3 - Nous avons quand même bien avancé sur différents aspects, mais pas suffisamment sur tous (les aspects), pour avoir un ensemble cohérent.