Gestion de Stock

Cahier de charges

Sommaire :

1 - Contexte du projet :

Le présent projet **gestion de stock** s'inscrit dans le programme de formation de l'élève **Lotfi FATHI** au sein de l'école ESIC en vue de la préparation d'un diplôme RNCP N°6 de **"Concepteur Développeur Applications" -** Session 2020-2021. Ce projet constitue une première mis-en-pratique de différentes connaissances acquises durant une année de formation sur le développement web backend, le développement web frontend ainsi que la conception et la gestion des bases de données. Le projet n'a pas atteint à ce jour la maturité d'une mise-en-production.

2 - L'idée du projet :

L'objectif du projet est de développer et déployer sur le web une solution informatique qui permettra à des entreprises (individuelle ou collective) de vente (physique ou en ligne) de produits commerciaux de tout genre de gérer leurs stocks de produits, de faciliter les interactions de l'entreprise avec ses clients et ses fournisseurs et de structurer le travail au sein de l'entreprise même.

A ce jour, le projet gestion de stock permet à ses clients de précisément pouvoir :

- Créer un profil principal de l'entreprise.
- Créer et gérer des profils utilisateur de l'application au nom de l'entreprise.
- Créer et gérer des catégories d'articles ou de services à vendre.
- Créer et gérer des articles, des produits...
- Créer et gérer des profils clients.
- Créer et gérer des fournisseurs.
- Passer et gérer les commandes des clients.
- Passer et gérer les commandes des fournisseurs.
- Enregistrer des ventes directes au magasin.
- Consulter l'historique et l'état du stock des article en temps réel et pouvoir apporter des corrections (si besoin).

Précisions:

-Le verbe 'gérer' utilisé dans le document fait référence à l'ensemble des opérations CRUD(Create, Read, Update, Delete).

-Le pluriel utilisé fait également référence à l'unité.

3 - Technologies utilisées :

Le projet gestion de stock se compose principalement de deux projets. Un projet backend de type **API Rest** qui peut être consommé par d'autres projets backend ou d'autres interfaces terminales et un projet **frontend** qui consomme cet API et donne l'illustration sur tout navigateur web connecté à internet.

- La conception de la solution a été réalisé sur : https://www.diagrams.net/
- L'API Rest du projet Gestion de stock a été créé avec le Framework **Spring Boot 2.4.8** évoluant dans l'écosystème **Java 11**.
- Le frontend du projet gestion de stock a été créé avec le Framwork **Angular 10.**
- Les éditeurs de codes utilisés au début de projet sont **Spring Tools Suits 4.11.0** et **VS Code 1.56.2.** Par la suite, on a opté pour un éditeur de code en ligne qui encapsule les deux projets : **GitPod**.
- Pour la gestion relationnelle des données **SGBD-R**, le choix s'est porté sur **MySQL.** Une base de données **"gestionstock"** a été généré dans le but.
- Pour le stockage du code de l'application le versionning, nous avons opté pour Git/gitHub.
- **Maven** est le gestionnaire des packages du projet backend.
- **NPM** est le gestionnaire des packages du projet frontend.
- Les Framework **Junit** and **Karma** ont été utilisé pour la tests unitaires backend et front end respectivement.
- **Postman** pour les tests d'intégration des API.

4 - Considérations importantes :

Le coté sécurité a été dument implémenté avec Spring Security. La gestion de l'authentification se fait via un Protocol JWT. Aucun accès n'est autorisé sans possession d'un Token valide. Le premier canal d'accès est le remplissage du formulaire d'inscription d'une entreprise. Une fois cela est fait, un profil utilisateur est généré pour le demandeur avec un code d'accès qui lui sera envoyé par email. L'accès et la gestion de la session grace à un couple (email, password) permettront par la suite à l'utilisateur principale d'accéder à l'applications et de créer d'autres profils utilisateurs.

Une documentation automatisée de l'API a été implémenté avec l'utilisation de **Swagger**. Cet outil permet aussi de tester dans la volée les endpoint des API.

La gestion des images a été débugué à l'API Flickr de Twitter. Un espace dans le cloud a été alloué pour le fait. Plus d'informations sur: https://www.flickr.com/services/api/

Grace à l'inversion de contrôle et l'injection de dépendances, nous avons une première garantie concernant la maintenabilité et la réutilisation du code.

5 - Ressources du projet :

Le code source du projet est hébergé sur le répertoire suivant :

https://github.com/LotfiF/projet_gds

Un accès rapide au workspace gitpod peut être configuré comme expliqué ici : https://www.gitpod.io/docs/getting-started

Le modèle conceptuel de donnée (**Annexe I**) et le modèle physique de données (**Annexe II**) sont également accessibles via le répertoire du projet.

La documentation de l'API est accessible via le lien suivant :

http://localhost:8081/swagger-ui.html ou https://8081-coffee-echidna-37t0zian.ws-eu17.gitpod.io/swagger-ui.html

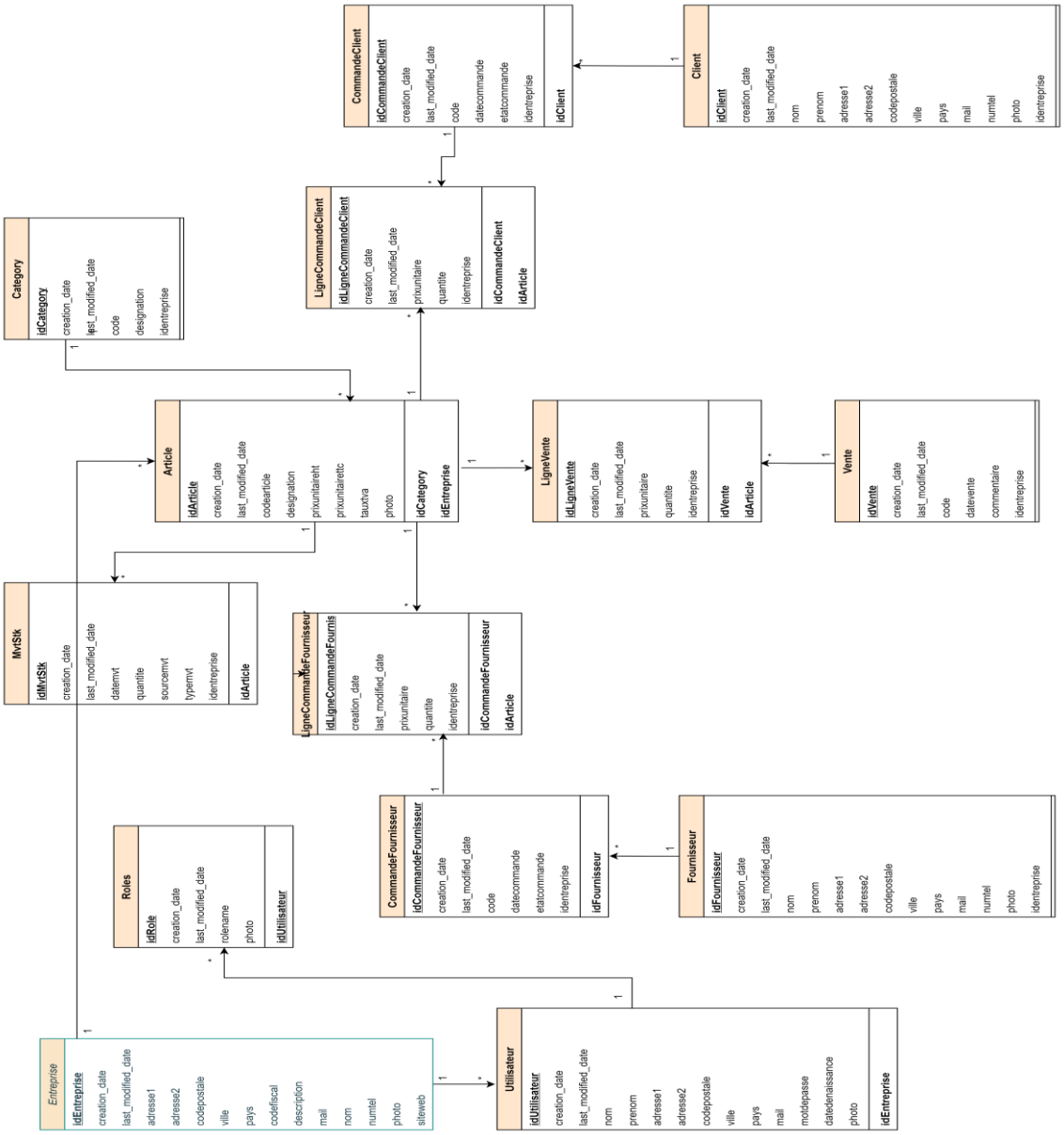
Le paquet de tests d'intégration est disponible sur l'application web de Postman (code d'accès à solliciter).

https://web.postman.co/

6 - Futures fonctionnalités :

- Dashboard + statistiques.
- Gestion du mailing.
- Contraintes sur la saisie et les validateurs.
- Internationalisation.
- Hébergement dans le Cloud.
- Système de Ci/Cd.
- Déploiement.
- Optimisation des performances.
- Solution de payement en ligne.
- Export/import des fichiers.
- La pagination.

Annexe I


CommandeClient
idCommandeClient
creation_date
last_modified_date
code
datecommande
etatcommande
identreprise
idClient
Client
idClient
creation_date
last_modified_date
nom
prenom
adresse1
adresse2
codepostale
ville
pays
mail
numtel
photo
identreprise
LigneCommandeClient
idLigneCommandeClient
creation_date
last_modified_date
prixunitaire
quantite
identreprise
idCommandeClient
idArticle
Category
idCategory
creation_date
last_modified_date
code
designation
identreprise
Article
idArticle
creation_date
last_modified_date
codearticle
designation
prixunitairetf
prixunitairettc
tauxtva
photo
idCategory
idEntreprise
LigneVente
idLigneVente
creation_date
last_modified_date
prixunitaire
quantite
identreprise
idVente
idArticle
Vente
idVente
creation_date
last_modified_date
code
datevente
commentaire
identreprise
MvtStk
idMvtStk
creation_date
last_modified_date
datemvt
quantite
sourcemvt
typemvt
identreprise
idArticle
LigneCommandeFournisseur
idLigneCommandeFournis
creation_date
last_modified_date
prixunitaire
quantite
identreprise
idCommandeFournisseur
idArticle
CommandeFournisseur
idCommandeFournisseur
creation_date
last_modified_date
code
datecommande
etatcommande
identreprise
idFournisseur
Fournisseur
idFournisseur
creation_date
last_modified_date
nom
prenom
adresse1
adresse2
codepostale
ville
pays
mail
numtel
photo
identreprise
Roles
idRole
creation_date
last_modified_date
rolename
photo
idUtilisateur
Entreprise
idEntreprise
creation_date
last_modified_date
adresse1
adresse2
codepostale
ville
pays
codefiscal
description
mail
nom
numtel
photo
siteweb
Utilisateur
idUtilisateur
creation_date
last_modified_date
nom
prenom
adresse1
adresse2
codepostale
ville
pays
mail
motdepasse
datedenaissance
photo
idEntreprise

Annexe II