

SYSTEME DE GESTION DES STOCKS POUR UNE ENTREPRISE DE VENTE AU DETAIL

Design de la Base de Données

Licence 3 - CSC 394 | GL - DWM

iPNet Institute of Technology | 2022-2023

Table des matières

- Analyse
- Conception
- Implémentation

Équipe projet

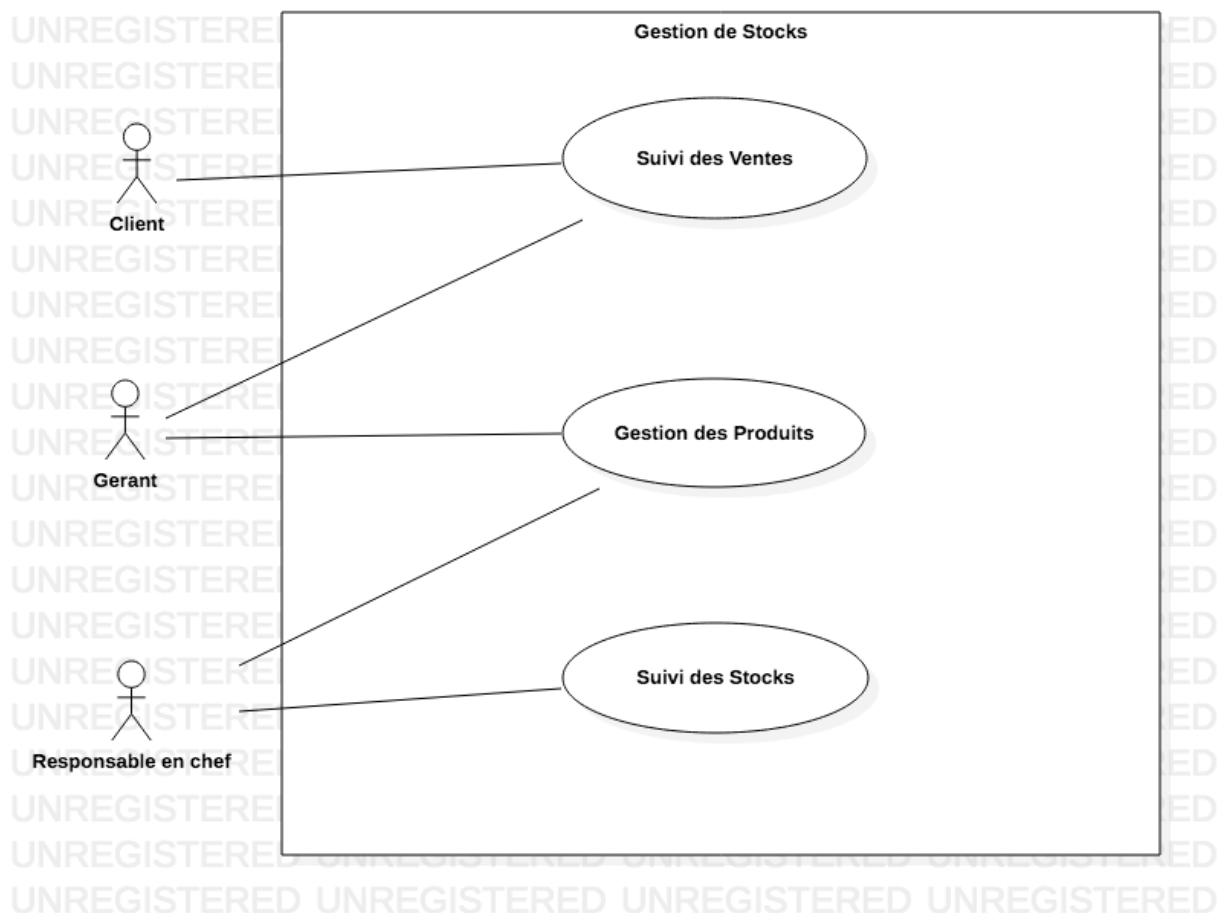
- DJANTA Samuel
- DJAOURA Dessabatouma
- GABLA Aimé Giscard
- GOGO Kossi Daniel
- GOMINA Samah Clavaire
- KAMBIA Méyébinesso Irenée
- ODOLA Gédéon Roland
- SESSOU-ADISSEH Jean-Josué Bright
- SIROUH Ismahil
- SOUSSOUKPO Yao David
- TEKOU Ayélé Clarisse Emmanuella

I. Analyse

1. Approche descendante : Le modèle de fonctionnement du système d'information

Formalisme : Diagramme de cas d'utilisation | Notation UML

✓ Composante graphique



✓ Composante textuelle : *Scenarios de cas d'utilisation*

Scenarios	Cas d'utilisation : <i>Gestion des Produits</i>
Principal	<p>Le Responsable en chef <i>enregistre</i> des Produits.</p> <p>Il peut par la suite, <i>effectuer un listage, un filtrage et une recherche</i> de Produit(s) parmi tous les Produits enregistrés.</p> <p>S'il y a lieu de <i>modifier</i> des informations relatives à un ou plusieurs de ces Produits (voire de les <i>supprimer</i>), il le fait.</p> <p>Le Gérant pour sa part ne dispose que d'un <i>droit en lecture (listage, filtrage, recherche)</i> sur ces Produits.</p>
Alternatif	N/A

Scenarios	Cas d'utilisation : <i>Suivi des Ventes</i>
Principal	<p>Le Client est le déclencheur de ce cas d'utilisation ; sans client on ne peut parler de Vente.</p> <p>Suite à une Vente, le Gérant <i>enregistre</i> celle-ci en conformité avec les informations suivantes : date, numéro de la vente, nom du client, prix total de la vente ; et les informations suivantes sur les Produits achetés : désignation, quantité, prix unitaire, prix total.</p> <p>Après chaque <i>enregistrement</i>, le système met à jour automatiquement</p>

	la quantité en stock de chaque Produit .
Alternatif	N/A

Scenarios	Cas d'utilisation : <i>Suivi des Stocks</i>
Principal	Le Responsable en chef définit pour les Stocks de Produits , un seuil de rupture . Suite aux mises à jour apportées par le système sur la quantité de chaque Produit , lorsque le seuil de rupture est franchi le système envoie une alerte et génère un email qui est directement expédié au Responsable en chef .
Alternatif	N/A

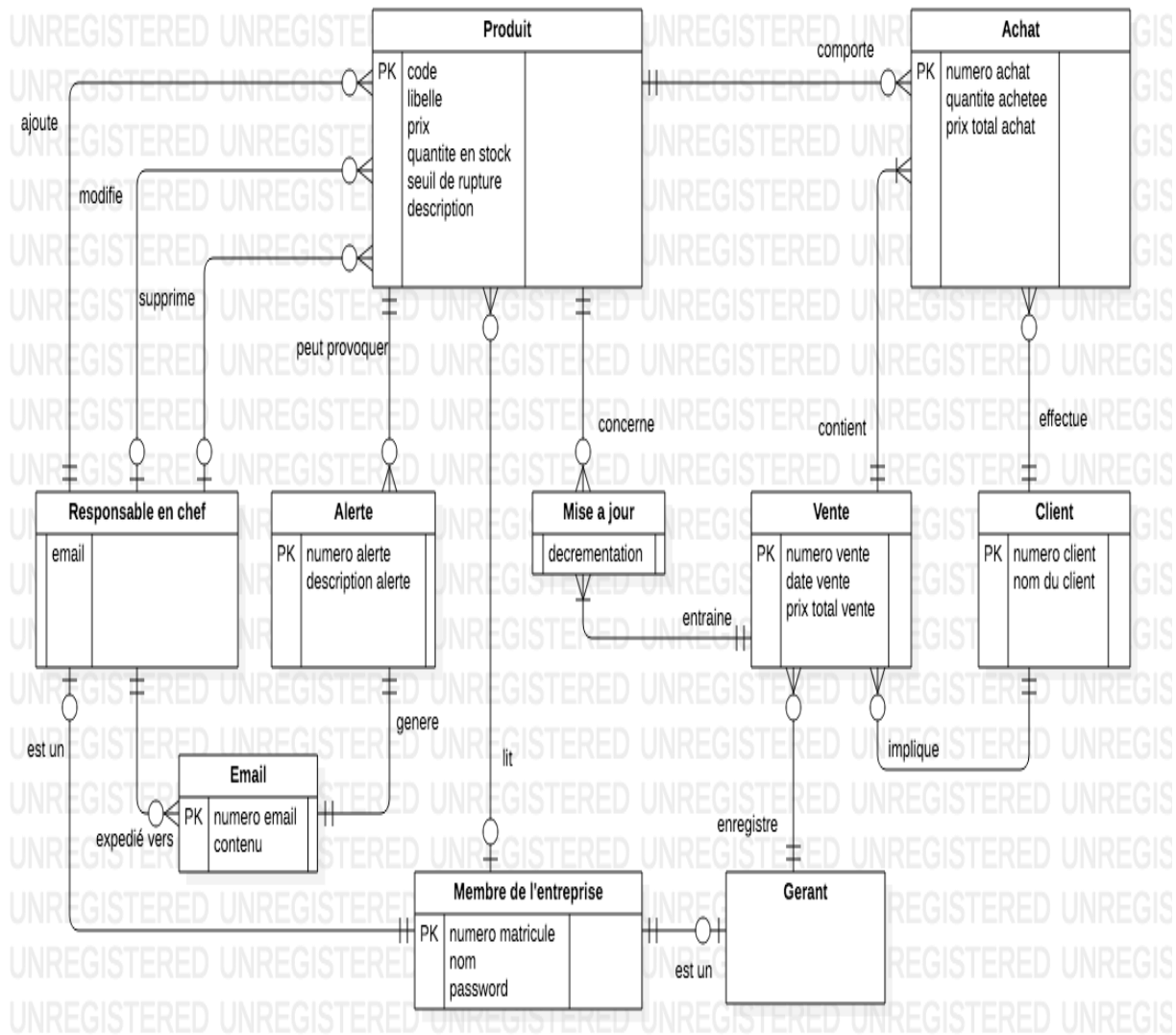
2. Modèle Conceptuel de Données (Formalisme : Entité – Association | Notation de Chen)

✓ Recensement des données persistantes

Nom donnée	P	Type	Mode	Contrainte d'intégrité	Règle de calcul
Produit					
Code	X	Entier	Mémorisé		
Libelle	X	Texte	Mémorisé		
Prix	X	Monnaie	Mémorisé		
Quantité en stock	X	Entier	Mémorisé		
Seuil de rupture	X	Entier	Mémorisé		
Description	X	Texte	Mémorisé		
Vente					
Date	X	Date	Mémorisé		
Numéro de la vente	X	Entier	Mémorisé		

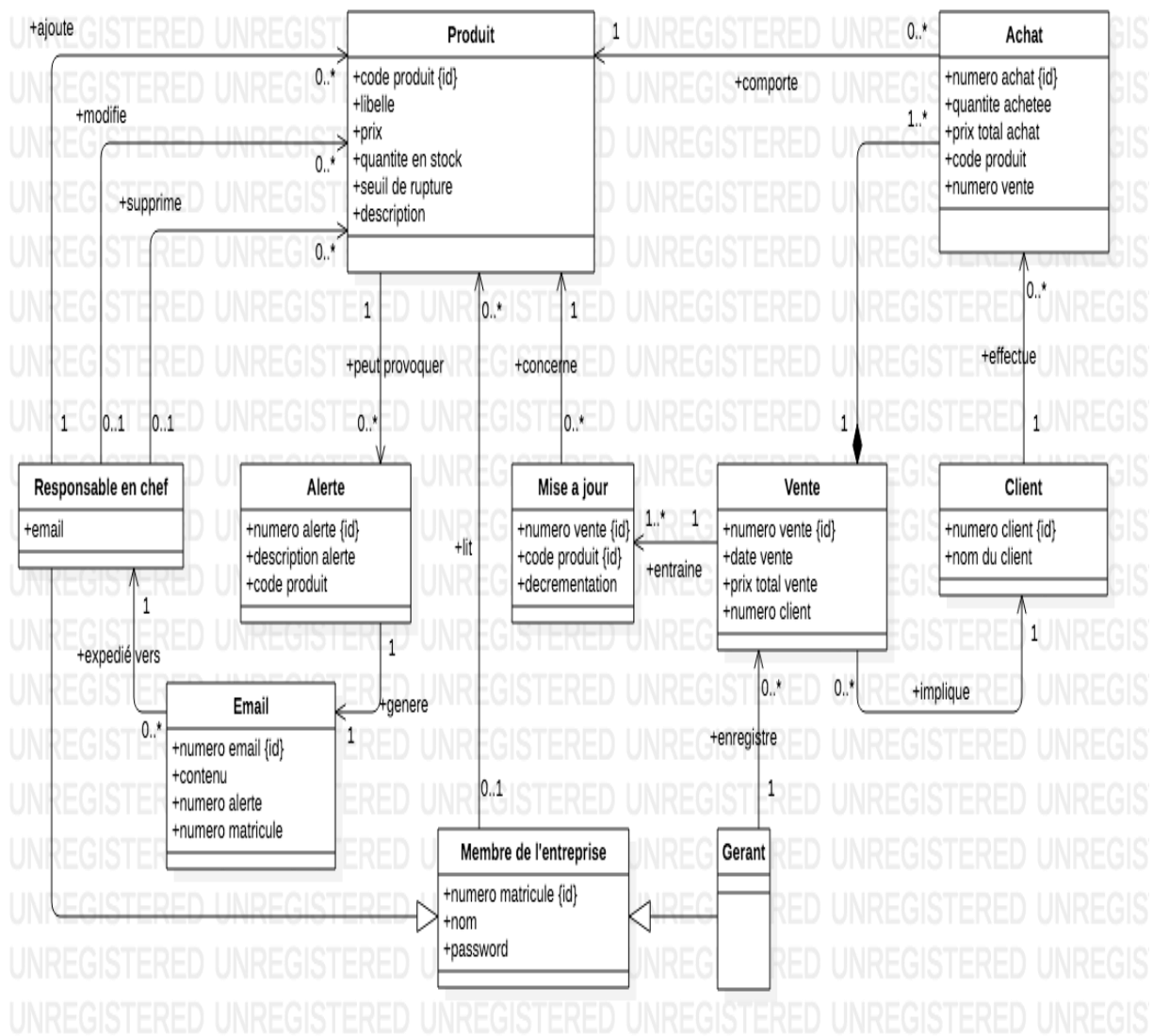
Prix total de la vente	X	Monnaie	Calculé		
Achat					
Numéro achat	X	Entier	Mémorisé		
Quantité achetée	X	Entier	Mémorisé		
Prix total achat	X	Monnaie	Calculé		
Alerte					
Numéro de l'alerte	X	Entier	Mémorisé		
Description de l'alerte	X	Texte	Mémorisé		
Email					
Numéro de l'email	X	Entier	Mémorisé		
Contenu de l'email	X	Texte	Mémorisé		

✓ Modélisation graphique



II. Conception

1. Modèle Relationnelle/Logique de Données (MRD/MLD)



2. Modèle Physique de Données : Script de définition de données

✓ Produit

```
create table Produit (  
    codeProduit int auto_increment,  
    libelle varchar (255) not null,  
    prix int not null,  
    quantiteEnStock int not null,  
    seuilDeRupture int not null,  
    descriptionProduit text not null,  
    primary key (codeProduit)  
);
```

✓ Client

```
create table Client (  
    numeroClient int auto_increment,  
    nomClient varchar (255) not null,  
    primary key (numeroClient)  
);
```

✓ Responsable

```
create table Responsable (  
    numeroMatricule int auto_increment,  
    nom varchar (255) not null,  
    passwordd char (16) not null,  
    email varchar (255) not null,  
    primary key (numeroMatricule)  
);
```

✓ Gérant

```
create table Gerant (  

```

```
numeroMatricule int auto_increment,  
nom varchar (255) not null,  
passwordd char (16) not null,  
primary key (numeroMatricule)  
);
```

✓ Vente

```
create table Vente (  
    numeroVente int auto_increment,  
    dateVente date not null,  
    prixTotalVente int not null,  
    client int not null,  
    primary key (numeroVente),  
    foreign key (client) references Client(numeroClient)  
);
```

✓ Achat

```
create table Achat (  
    numeroAchat int auto_increment,  
    quantiteAchete int not null,  
    prixTotalAchat int not null,  
    produit int not null,  
    vente int not null,  
    primary key (numeroAchat),  
    foreign key (produit) references Produit(codeProduit),  
    foreign key (vente) references Vente(numeroVente)  
);
```

✓ Alerte

```
create table Alerte (  
    numeroAlerte int auto_increment,  
    descriptionAlerte text not null,  
    produit int not null,  
    primary key (numeroAlerte),
```

```
foreign key (produit) references Produit(codeProduit)
);
```

✓ Email

```
create table Email (
    numeroEmail int auto_increment,
    contenu text not null,
    alerte int not null,
    responsable int not null,
    primary key (numeroEmail),
    foreign key (alerte) references Alerte(numeroAlerte),
    foreign key (responsable) references Responsable(numeroMatricule)
);
```

III. Implémentation

Création de la base de données : *create database GestionStocks ;*

Choix de la base de données : *use GestionStocks ;*

Création des tables : *[saisir dans l'ordre les scripts de définition de données écrits précédemment]*