

**Faculté des sciences exactes  
Département d'informatique**



**جامعة بجاية**  
**Tasdawit n Bgayet**  
**Université de Béjaïa**

**Mémoire du projet de fin de cycle licence en vue de l'obtention du  
diplôme de licence académique en informatique spécialisé en  
systèmes informatiques :**

---

**Intitulé du projet :**

Développement, conception et mise en place d'une application desktop sous Java en vue de gérer des fonctionnalités administratives nécessaires dans le cas de gestion d'un magasin de vente de produits.

---

**Encadré par :  
Mme BACHIRI Lina**

**Présenté par :  
Amine Hamouche**

**L3 Section C groupe 1**

**Promotion 2019-2020.**

## Remerciements :

**Je tiens à remercier mes parents et ma famille qui m'ont soutenu tout au long de mon cursus universitaire.**

**Aussi les membres du jury pour la peine qu'ils se donneront pour lire mon travail ainsi que les critiques qu'ils y apporteront.**

**Finalement mes très chers ami(e)s aussi.**

# Table des matières

Introduction générale et structure du modèle .....	5
I. Introduction et étude préparatoire du projet .....	7
I.1 Introduction .....	7
I.2 Présentation du projet .....	7
I.3 Rédaction du cahier des charges.....	7
I.3.1 Structure du modèle .....	7
I.3.2 Problématique à faire face.....	8
I.3.3 Objectif de l'étude.....	8
I.3.4 Dresser la liste des fonctionnalités désirées.....	9
I.4 Cycle de vie choisi.....	9
I.5 Modélisation .....	10
II. Plan et détails de conception par fonction .....	11
II.1 Introduction .....	11
II.2 Diagramme de cas d'utilisation général.....	11
II.2.1 Diagramme de cas d'utilisation de l'acteur « Employé » .....	13
II.2.2 Diagramme de cas d'utilisation de l'acteur « Client » .....	14
II.3 Diagramme de classes .....	14
II.4 Diagrammes de séquence des cas d'utilisation .....	15
II.4.1 Diagramme de séquence du cas d'utilisation « d'authentification »...	16
II.4.2 Diagramme de séquence du cas d'utilisation « d'ajout d'acheteur » .	17
II.4.3 Diagramme de séquence du cas d'utilisation « Mise à jour acheteur » .....	18
II.4.4 Diagramme de séquence du cas « Consulter les détails des acheteurs » .....	19
II.4.5 Diagramme de séquence du cas « Supprimer acheteur ».....	20
II.4.6 Diagramme de séquence du cas « Ajouter produit » .....	21
II.4.7 Diagramme de séquence du cas « Mise à jour produit » .....	22
II.4.8 Diagramme de séquence du cas « consultation des détails des produits » .....	23
II.4.9 Diagramme de séquence du cas « Supprimer produit ».....	24
II.4.10 Diagramme de séquence du cas « Créer facture » .....	25
II.4.11 Diagramme de séquence du cas « Consulter fonctions disponibles »	26
II.5 Le modèle relationnel .....	26
II.6 Conclusion .....	27

III. Réalisation du système .....	28
III.1 Introduction.....	28
III.2 Environnement de réalisation.....	28
III.2.1 Lucidchart .....	28
III.3 Environnement de développement.....	28
III.3.1 NetBeans .....	28
III.3.2 SQLite .....	29
III.4 Outils de développement.....	30
III.4.1 La machine virtuelle Java (JVM).....	30
III.4.2 Environnement d'exécution Java (JRE).....	30
III.4.3 Le kit de développement Java (JDK) .....	30
III.5 Langages de programmation utilisés .....	30
III.5.1 Java .....	30
III.5.2 Structured Query Language (SQL).....	30
III.6 Présentation de l'application .....	31
III.6.1 Nom de l'application .....	31
III.6.2 Logo de l'application .....	31
III.7 Présentation des interfaces d'utilisation .....	31
III.7.1 Interface d'identification .....	31
III.7.2 Interface d'accueil principale .....	32
III.7.3 Interface d'ajout (Client, Produit).....	34
III.7.4 Interface de mise à jour (Client, Produit).....	35
III.7.5 Interface consultation des données (Client, Produit).....	37
III.7.6 Interface suppression des données (Client, Produit) .....	38
III.7.7 Interface de facturation (1 Client, x Produits).....	39
III.8 Conclusion .....	42
Conclusion générale .....	42
Bibliographie.....	43
Table des figures .....	<b>Error! Bookmark not defined.</b>

# Introduction générale et structure du modèle

Certes avec les différentes avancées technologiques que sont faites d'année en année, instinctivement on se dirait qu'une application web ou une application mobile seraient les meilleures options pour créer une application dans le but de répondre aux besoins d'un client.

Le but de ce travail est de faciliter l'activité d'une personne, au maximum un petit groupe de personnes qui s'occuperont de la gestion d'une boutique sur un nombre limité de machine, et donc prioriser la stabilité de l'application et les performances serait un bon moyen de maximiser les gains de cette boutique en question d'où le choix que j'ai entrepris vers l'application desktop.

Encore une fois, ce travail offrira l'opportunité à l'équipe de gestion d'être indépendant et ceci est dû à l'environnement de travail qui est interne, Au vu des innovations technologiques et de la digitalisation des entreprises le choix d'une application doit dépendre de son utilité. Pour le moment, il semble que les applications mobiles soient la nouvelle tendance. Néanmoins, à ne pas négliger l'importance des applications web et desktop !

Pour la réalisation du projet, j'ai suivi le modèle du Processus agile/Feature Driven Développement avec lequel j'ai commencé à construire mon application selon un modèle général puis une liste de fonctionnalités que je vais citer un peu plus bas et qui me semblait le plus adéquat en prenant en compte un cahier des charges de taille relativement petit pour la modélisation du système, le langage UML (Unified Modeling Language) est utilisé.

Quant à l'architecture respectée dans la programmation du système c'est bien évidemment JAVA avec l'utilisation de Swing, qui est une bibliothèque graphique pour ce langage de programmation.

Par ailleurs, j'ai utilisé le SGBD (Système de Gestion de Bases de données) SQLite qui propose un moteur de base de données relationnelle accessible par le langage SQL, contrairement aux serveurs de BDDs

traditionnels comme MYSQL ou WampServer sa particularité est de ne pas reproduire le même schéma habituel du modèle client-serveur mais d'être plus au moins intégré au programme en question, ma base de données intégrale est enregistrée dans un fichier indépendant de la plateforme sous extension .db ou bien .sql.

En effet, puisque mon choix s'est porté vers les performances et la stabilité du système et donc ne pas centraliser les données c'était à mon goût plus judicieux de choisir ce type de base de données embarquée.

Ce rapport va détailler toutes les nombreuses phases de conception et est structuré en trois chapitres :

- Le premier chapitre va se permettre de définir le contexte et le modèle dans lequel une étude préalable du projet se fera, ainsi traiter la problématique essentielle, l'identification d'acteurs suivi de la rédaction du cahier des charges avec l'élaboration d'un diagramme de contexte statique et finalement on verra la méthodologie et la modélisation choisie pour la conception.
- Dans ce deuxième chapitre on s'attardera sur l'analyse des besoins avec un diagramme de cas d'utilisation général, de classes ainsi que les diagrammes de séquences les plus essentiels pour objectif de détailler la conception du projet.
- Dans ce troisième chapitre je vais présenter la mise en place du projet, ses interfaces graphiques ainsi que les outils utilisés lors de la conception détaillée.
- Pour terminer, une conclusion permettant de faire une synthèse relative au travail accompli.

# I. Introduction et étude préparatoire du projet

## I.1 Introduction

Dans ce chapitre, je vais mettre l'accent sur l'analyse préliminaire du projet ainsi commencer à traiter la problématique de création d'une application à but administrative.

## I.2 Présentation du projet

Cette application desktop sous java aura pour but d'organiser le travail administratif des employés d'une boutique (supposons qu'elle soit spécialisée dans la vente exclusif de vêtements).

Elle accompagnera donc les employés dans leur tâche de gestion ainsi que de récupérer les données indispensables sur les acheteurs pour créer une facturation et réussir une vente dans les plus brefs délais.

## I.3 Rédaction du cahier des charges

### **I.3.1 Structure du modèle**

Pour déterminer le modèle/domaine d'étude ainsi que le contexte du système, la priorité est d'identifier les acteurs de ce dernier.

Ici on aura que deux acteurs qui vont interagir avec mon système ; l'employé/administrateur et bien évidemment le client, après ça le but est de cerner avec quel genre d'environnement communique ce système.

Ci-dessous mon diagramme de contexte système résume à peu près la communication entre ces acteurs avec le système :

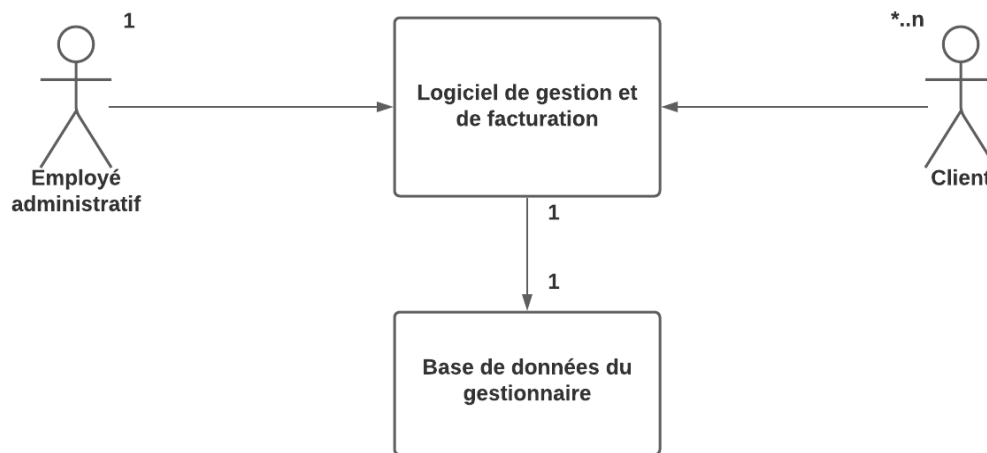


Diagramme de contexte système

**Figure I.1 Diagramme de contexte système**

### I.3.2 Problématique à faire face

Dans mon exemple de boutique de vente de produits j'imaginai qu'effectuer les tâches manuellement allait demeurer très complexe et fastidieux (surtout avec un nombre de produits en boutique dépassant la centaine et plus encore !), j'en ai donc déduit quelques points importants au cours de ses tâches « manuelles » :

- Manque d'informations sur les produits et donc se fier à la mémoire humaine ce qui peut en résulter un effet catastrophique sur le système au complet.
- Difficulté de mise en place d'un inventaire et si la boutique évolue ça causera des répercussions sur le travail en général.
- Environnement de travail stressant pour le ou les employés, causé par toute cette masse de données à prendre en considération lors de n'importe quel achat effectué par un client.

### I.3.3 Objectif de l'étude

Pour essayer de répondre à la problématique dernièrement soulevée avec ces contraintes exprimées, le système doit donc changer et doit répondre à quelques buts précis :

- Être capable d'avoir accès à tous les détails sur les produits en stock, ainsi que les détails des acheteurs saisis dans la base de données.



- De même pouvoir ajouter des acheteurs, et faire la liaison avec les produits achetés.
- Finalement, pouvoir réussir à générer une facture.

#### **I.3.4 Dresser la liste des fonctionnalités désirées**

Selon mon modèle établi je vais facilement reconnaître les fonctionnalités que va comporter mon application qu'on retrouve habituellement dans la plupart des programmes de gestion :

- L'ajout, la modification et la suppression d'acheteurs potentiels dans la base de données.
- L'ajout, la modification et la suppression de produits qui seront insérés dans la base de données au vus d'une facturation.
- La possibilité de visualiser ce qui a été inséré dans la base de données pour une meilleure gestion de ces dernières informations.

### **I.4 Cycle de vie choisi**

Allons droit au but, certes il existe d'autres cycles de vie et de conception peut être meilleurs mais selon moi ils ont approximativement le même scénario un cahier des charges de la taille d'une encyclopédie aussi fermés aux changements qu'une tête de pioche...

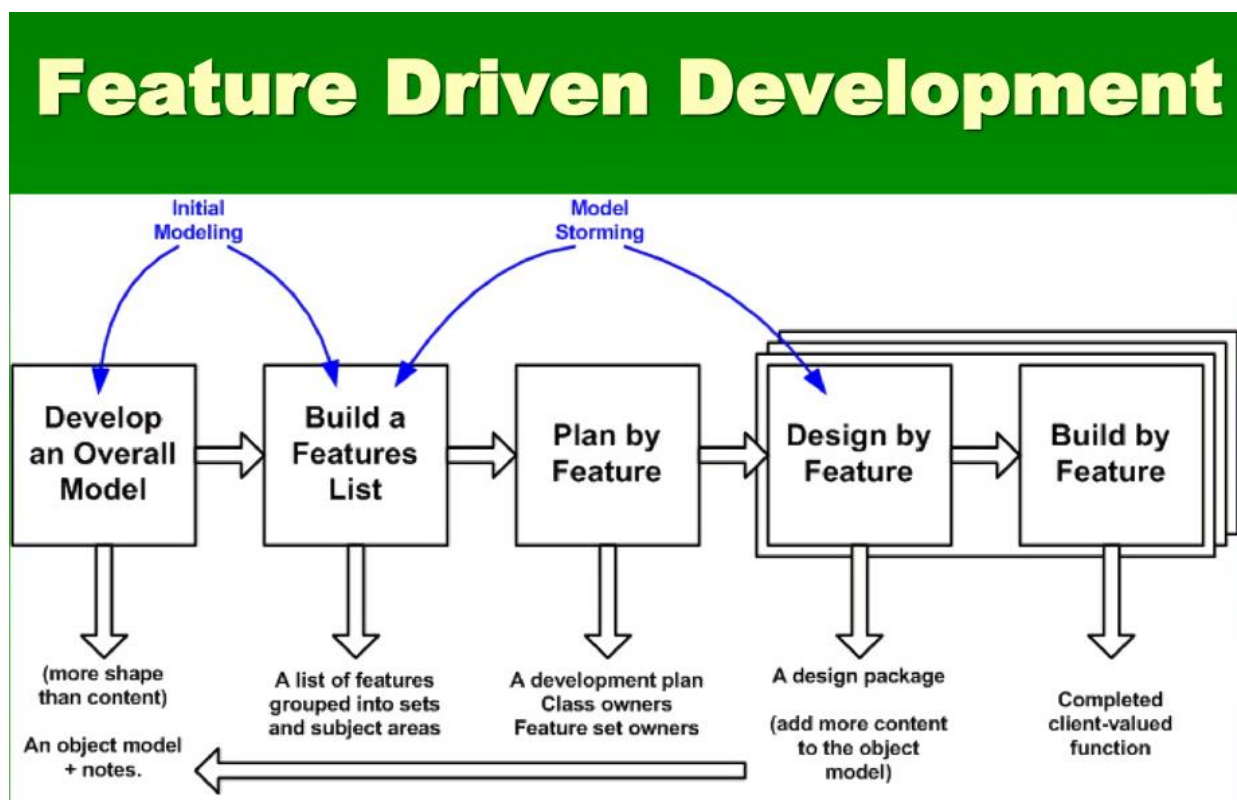
La méthode choisie qui est la conception agile est, elle centrée sur la satisfaction du client et non à la quasi conformité au cahier des charges et puisque la situation actuelle ne m'a pas laissé l'occasion d'interagir avec un vrai client à qui j'offrirai mes capacités de programmeur, je ferai office de celui-ci.

L'idée est donc de découper le projet en petit bloc puis les hiérarchiser selon les besoins et la valeur de ces derniers.

Le développement est organisé par itérations courtes autour des fonctionnalités mentionnées en impliquant le plus possible l'utilisateur dans les tests, ça m'a permis d'éviter les dérives observées quand des

développeurs sont restreints par une entité, le changement est donc encouragé puisque si mon projet a une bâtisse agile alors je ne vais pas me priver (au nom du client) des bonnes idées en cours de route !

Je vais conclure avec une analogie qui va je l'espère aider à expliquer mon choix ; si selon moi je pars en voyage la semaine prochaine je ne vais sûrement pas planifier chaque coin de rues par lesquels je vais passer mais seulement les grandes lignes, puisqu'on sait tous que les imprévus font parties intégrantes de la vie.



**Figure I.2 Déroulement de la méthode agile FDD**

## I.5 Modélisation

Pour modéliser mon projet, j'ai choisi le langage de modélisation unifié (UML) vu plus haut avec le diagramme de contexte statique, UML est une approche orientée objet à base de pictogrammes, elle me sera

utile pour donner une vision globale du système et clarifier quelques aspects techniques de certaines fonctionnalités.

Il faut bien noter que UML n'est pas un langage de réalisation mais bien celui de modélisation souvent on confond les deux, d'abord on énonce la problématique puis bien après on fait une modélisation de ce dernier pour s'y référer lors du développement.

L'inconvénient majeur avec UML est qu'il est un peu complexe et qu'il nécessite un certain temps d'apprentissage et d'adaptation mais heureusement pour nous ce n'est pas vraiment un problème puisque j'ai eu l'occasion de l'avoir deux années de suites lors de mon parcours universitaire.

## II. Plan et détails de conception par fonction

### II.1 Introduction

Selon mon cycle de vie (FDD) choisit établir les diagrammes de séquence et de cas d'utilisation est en théorie la troisième étape à réaliser en effet les deux étapes avant elle ont bien été effectuées c'est-à-dire ; Développer le modèle général en faisant une description du système ainsi que son contexte.

Et selon les connaissances acquises lors du développement du modèle on dresse une liste de fonctionnalités et leurs caractéristiques désirées être présentes ceci en créant une base solide comme deuxième étape.

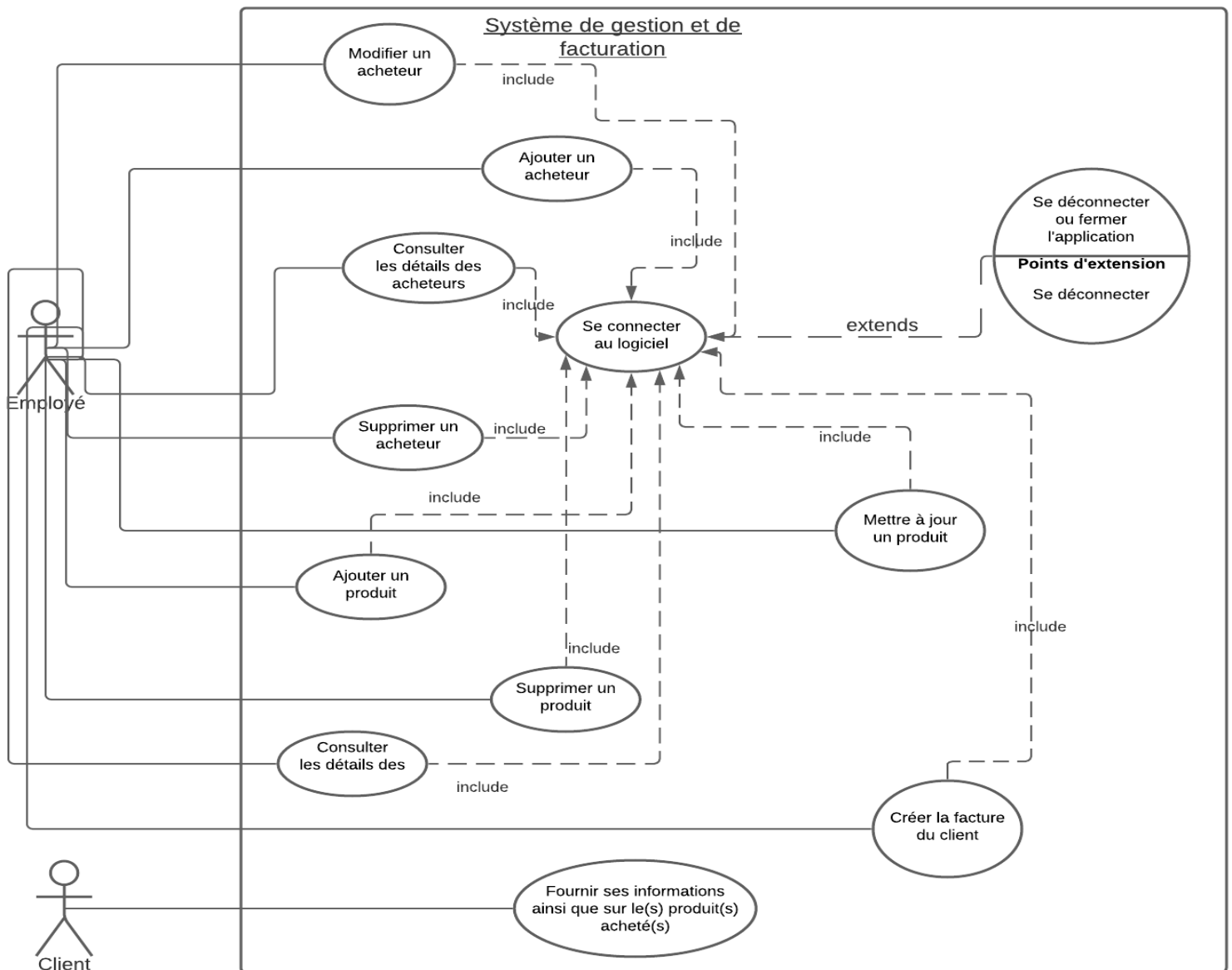
### II.2 Diagramme de cas d'utilisation général

Bien souvent, la maîtrise d'ouvrage et les utilisateurs ne sont pas informaticiens il faut donc un moyen simple pour modéliser les besoins c'est là qu'entre en jeu le diagramme de cas d'utilisation.

Un diagramme de cas d'utilisation est généralement utile pour recueillir, analyser et organiser les besoins du client. Dans les figures suivantes je

vous présenterai le diagramme général ainsi que les diagrammes de cas d'utilisation relatifs à chaque acteur du système.

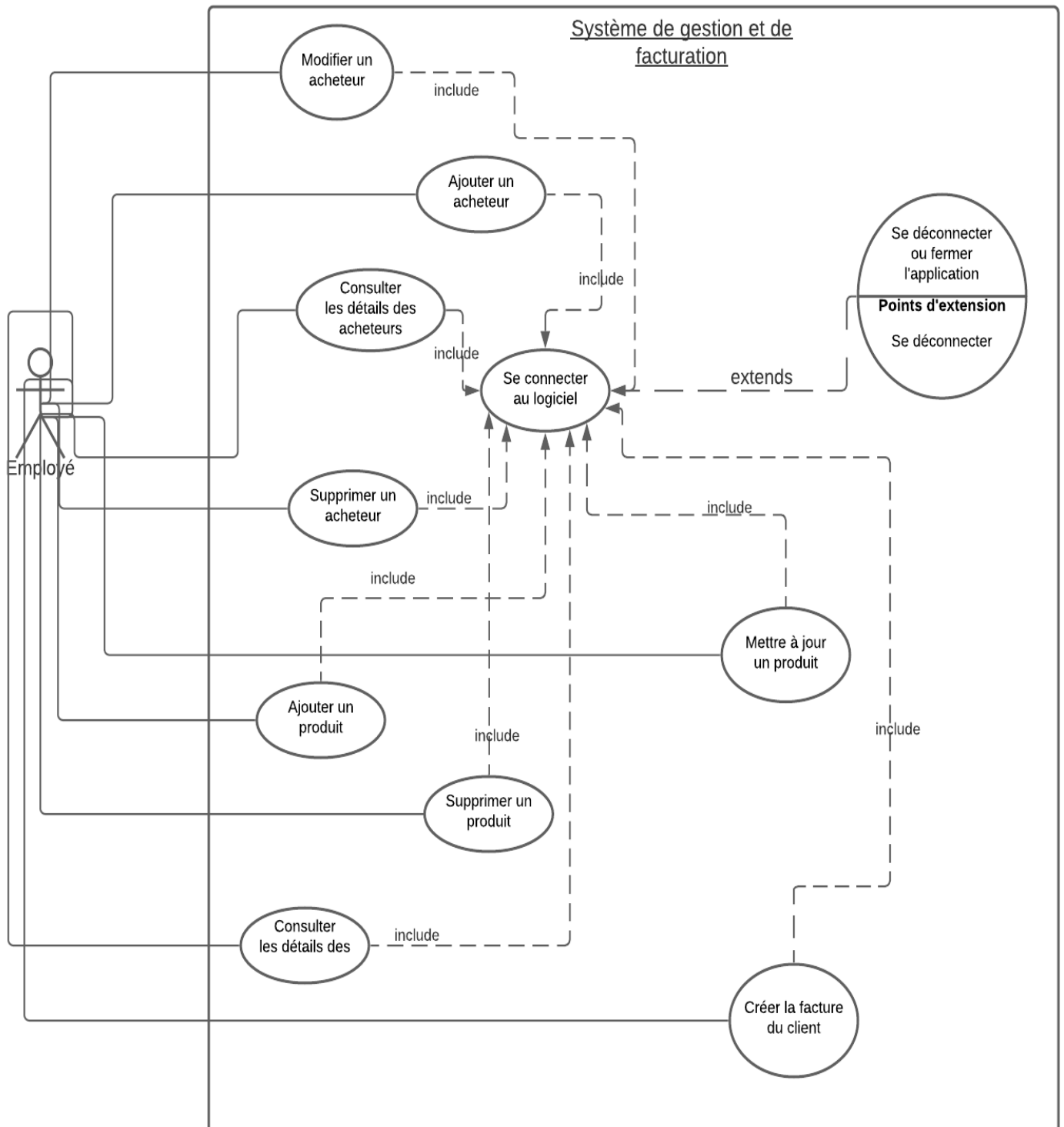
**Figure II.1 Diagramme de cas d'utilisation général**



**Diagramme de cas d'utilisation général**

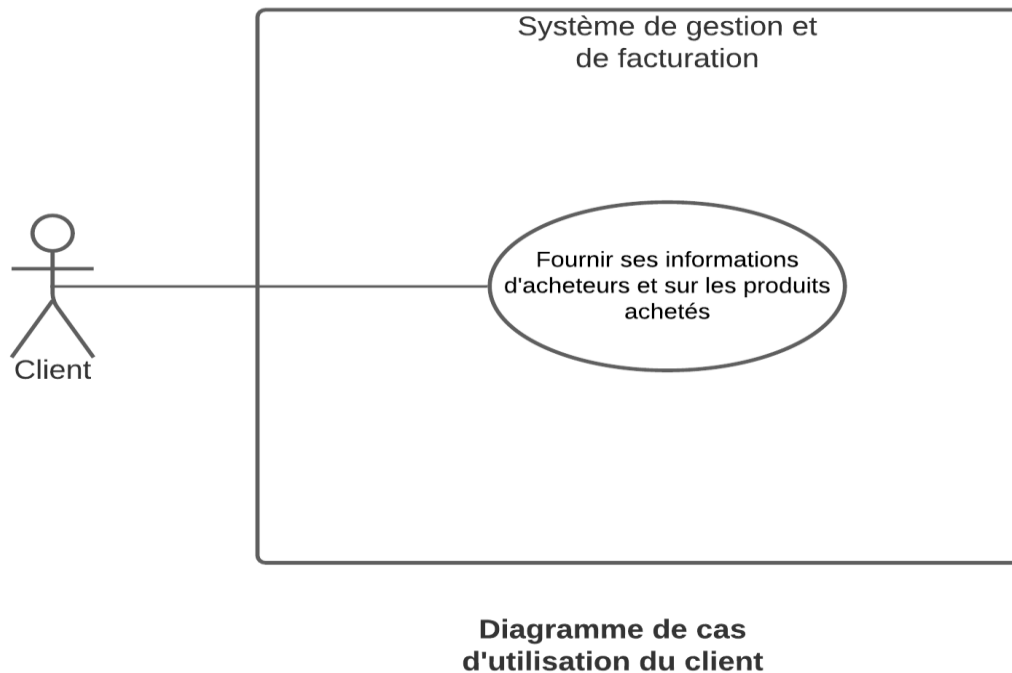
## II.2.1 Diagramme de cas d'utilisation de l'acteur « Employé »

**Figure II.2 Diagramme des cas d'utilisation de l'employé**



**Diagramme de cas d'utilisation de l'employé**

## II.2.2 Diagramme de cas d'utilisation de l'acteur « Client »



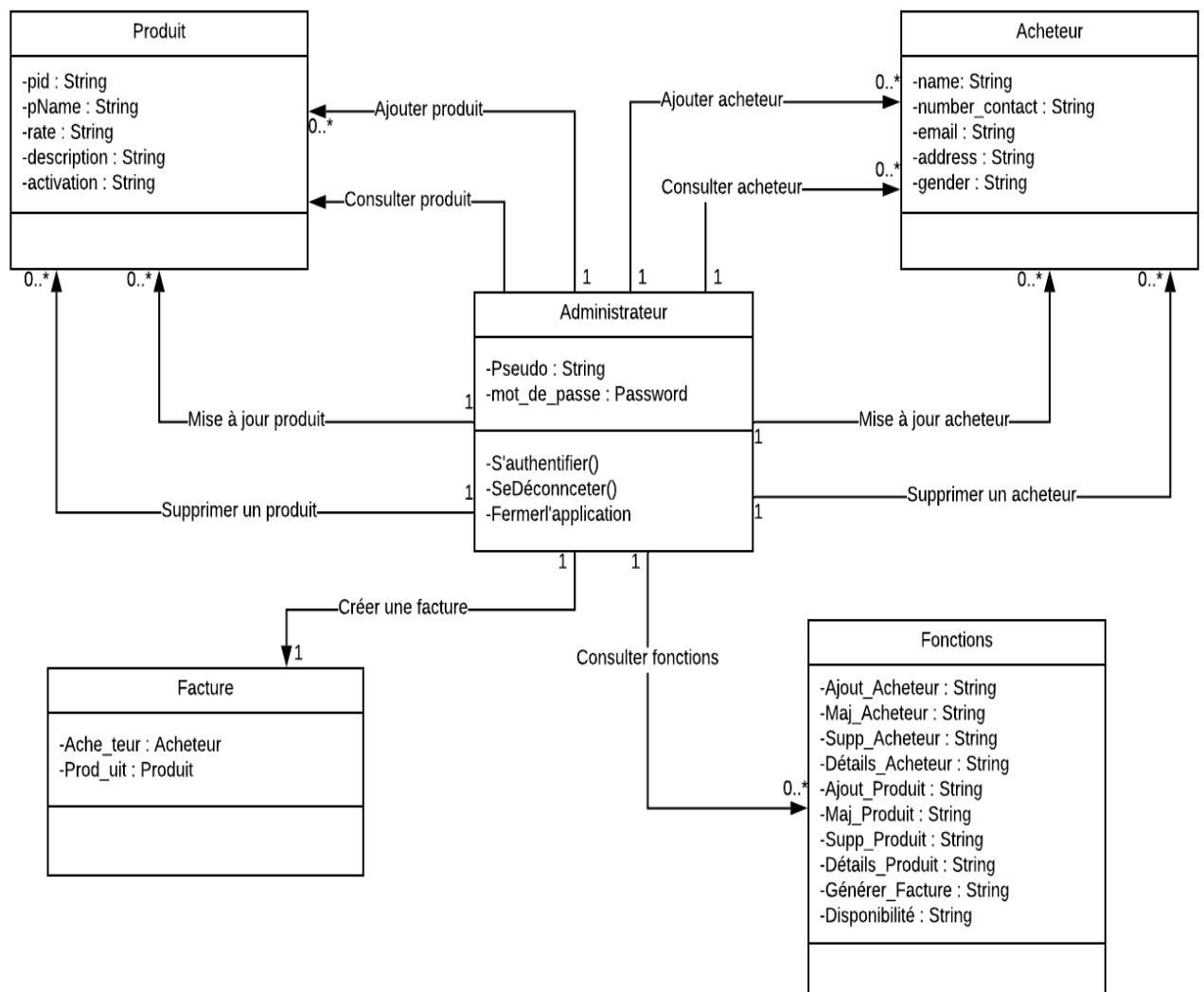
**Figure II.3 Diagramme de cas d'utilisation du client**

**NB : comme vous l'avez vu la liste des fonctionnalités tourne autour de l'employé puisque l'application est à but administrative le client n'a pas de cas d'utilisation marquant.**

## II.3 Diagramme de classes

Le diagramme de classes est l'un des plus importants diagrammes lors de la modélisation d'un projet même s'il ne montre qu'une vue statique du système, il montre néanmoins une vue interne du système ainsi qu'une représentation abstraite de ce dernier.

Les principaux éléments constituant ce diagramme sont les classes et les différentes relations incluses telles que ; l'association, la généralisation et d'autres aussi.



**Figure II.4 Diagramme de classes**

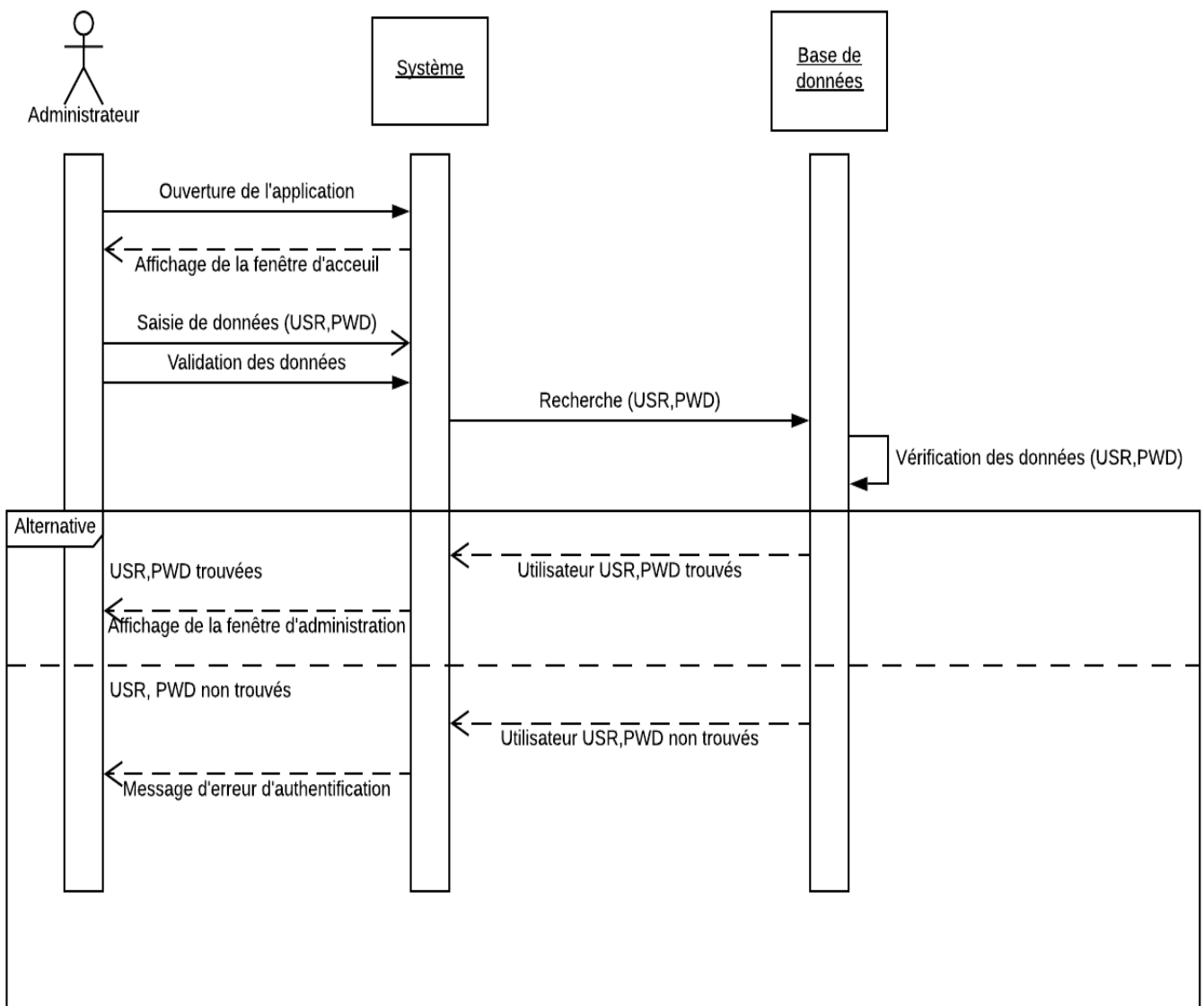
## II.4 Diagrammes de séquence des cas d'utilisation

L'objectif d'un diagramme de séquence d'un cas d'utilisation spécifique, est de représenter la temporalité d'interaction entre objets, avec une chronologie des messages échangés entre les acteurs.

Dans la rubrique qui suit, on parlera en ordre de séquence des acheteurs puis des produits selon leur cas d'utilisation.

## II.4.1 Diagramme de séquence du cas d'utilisation « d'authentification »

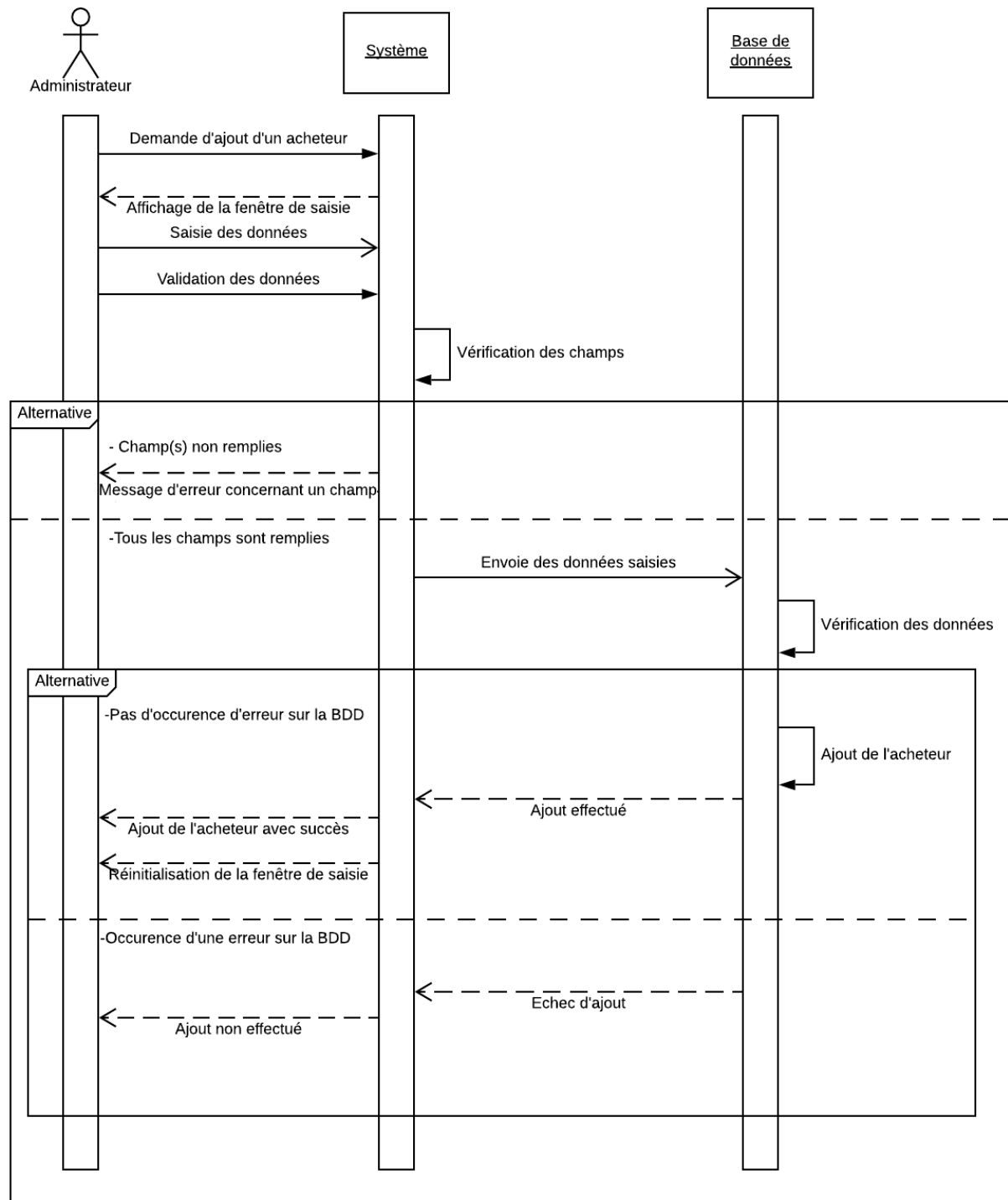
Pour commencer à utiliser l'application et n'importe quelle de ces fonctionnalités on doit débiter par l'opération d'authentification de l'administrateur.



**Figure II.5 Diagramme de séquence du cas « s'authentifier »**

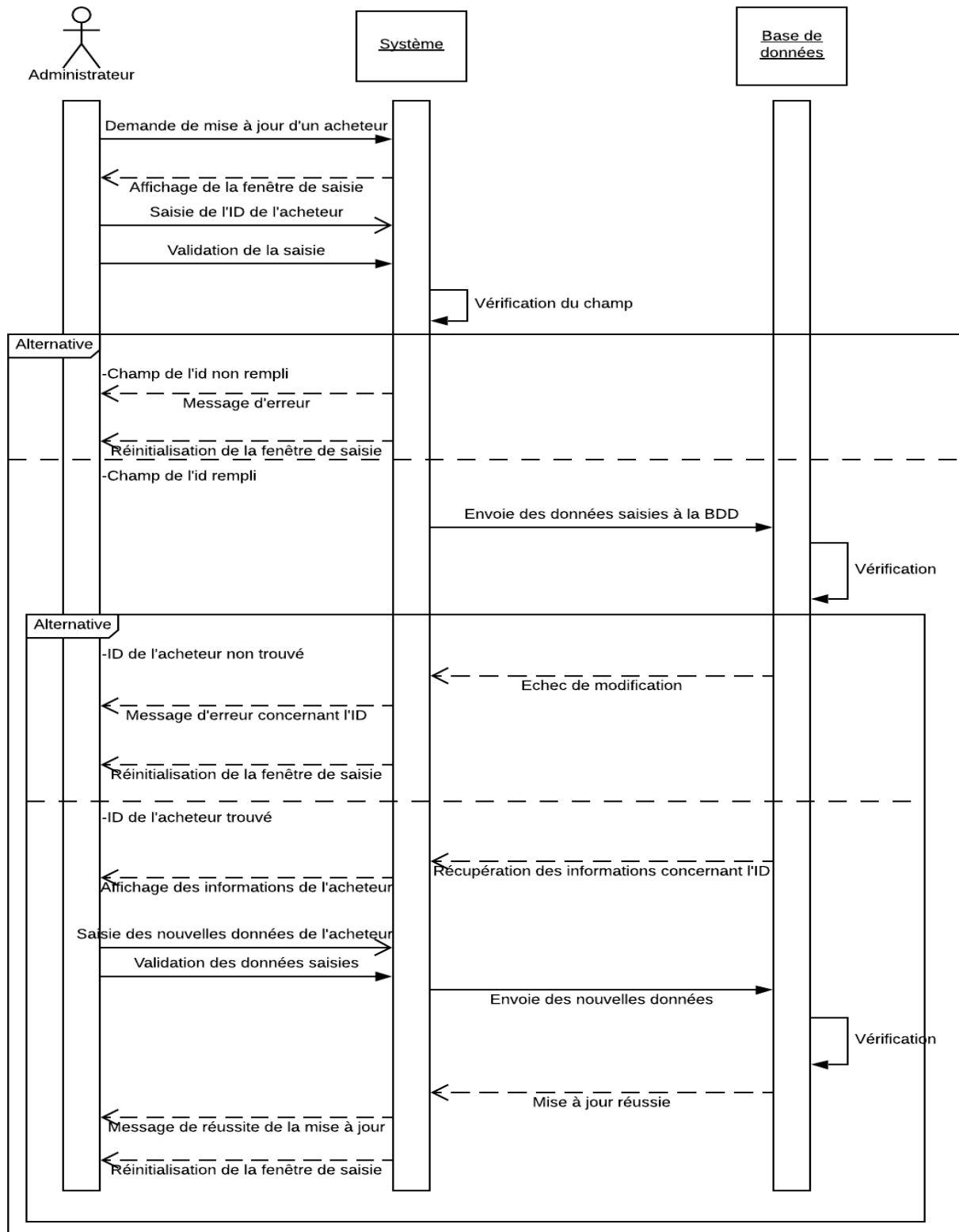


## II.4.2 Diagramme de séquence du cas d'utilisation « d'ajout d'acheteur »



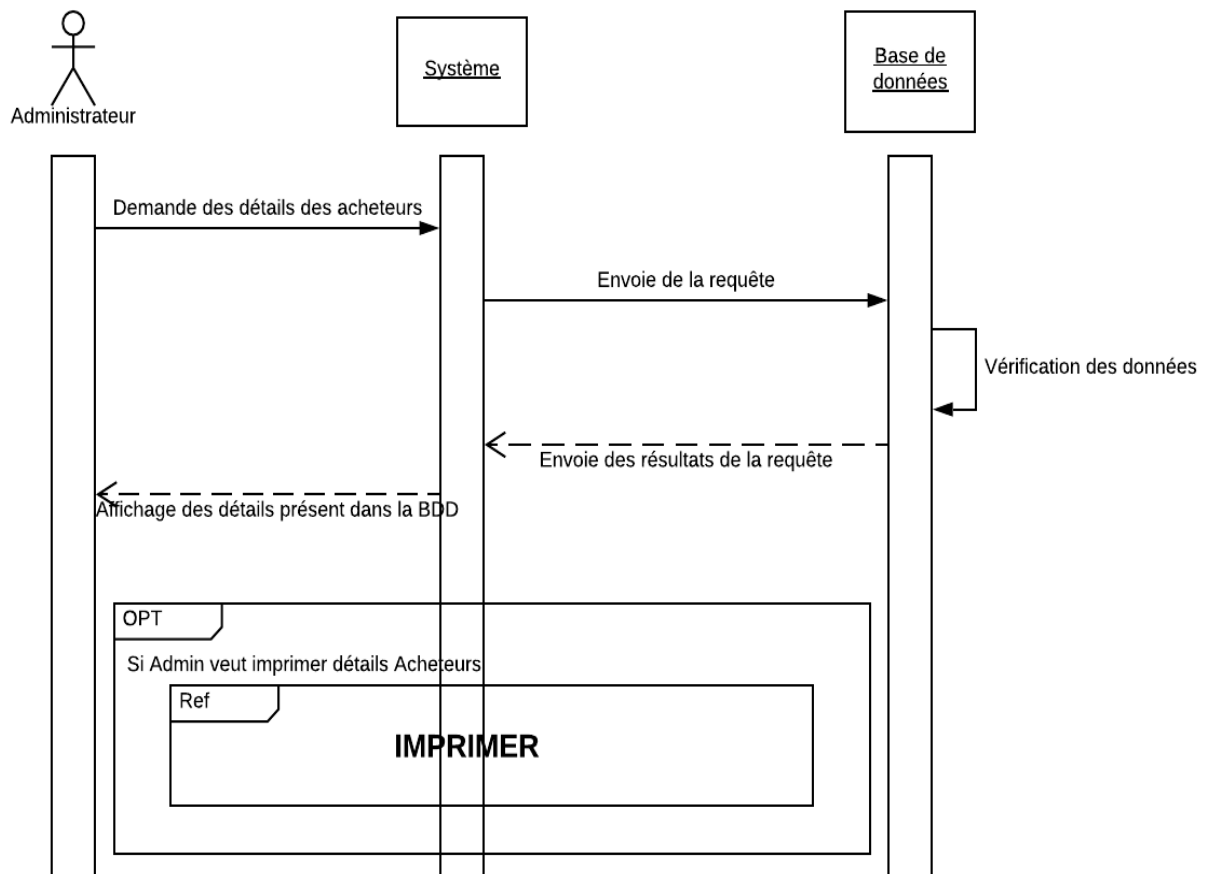
**Figure II.6 Diagramme de séquence du cas « Ajout acheteur »**

### II.4.3 Diagramme de séquence du cas d'utilisation « Mise à jour acheteur »



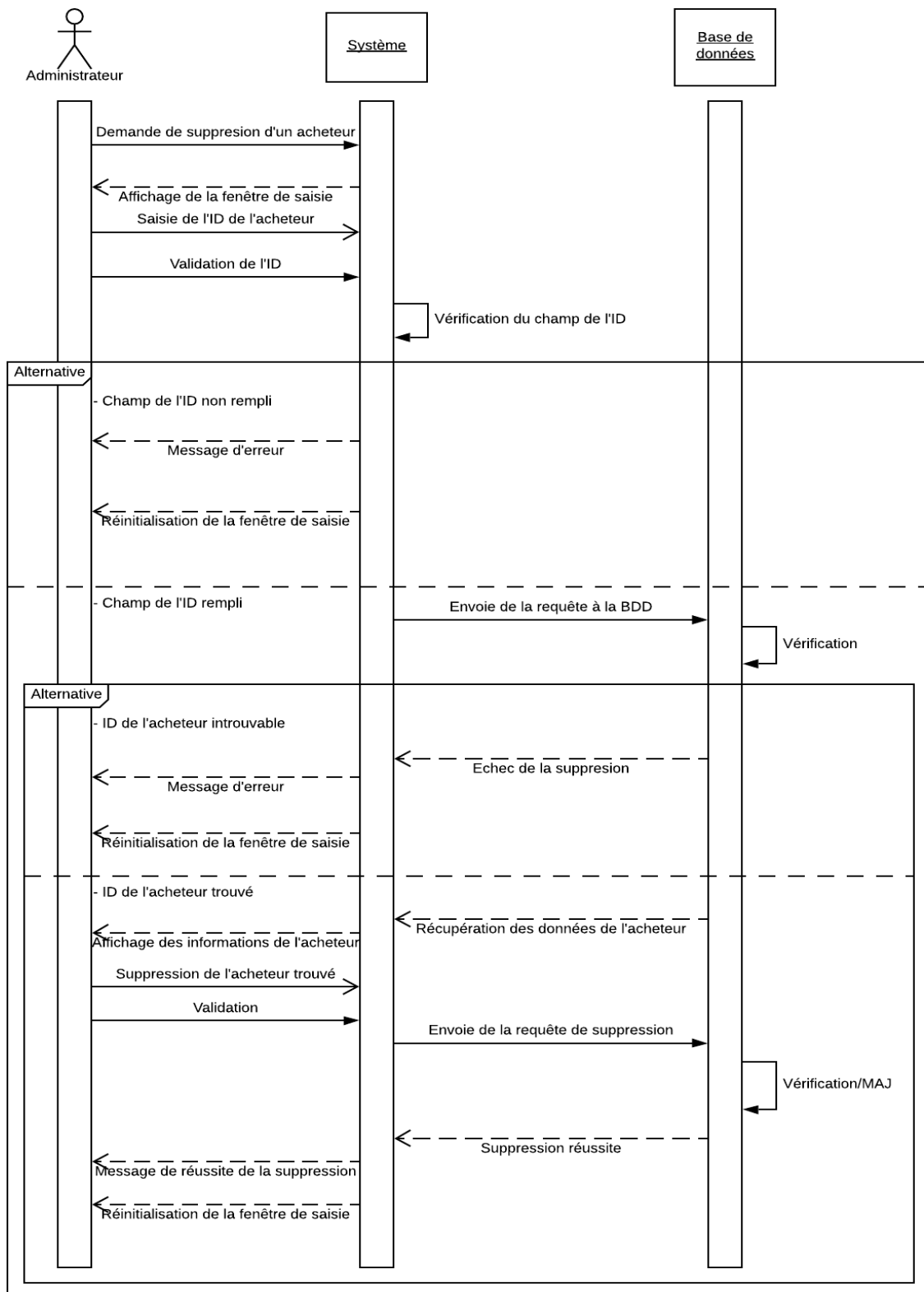
**Figure II.7 Diagramme de séquence du cas « Mise à jour acheteur »**

## II.4.4 Diagramme de séquence du cas « Consulter les détails des acheteurs »



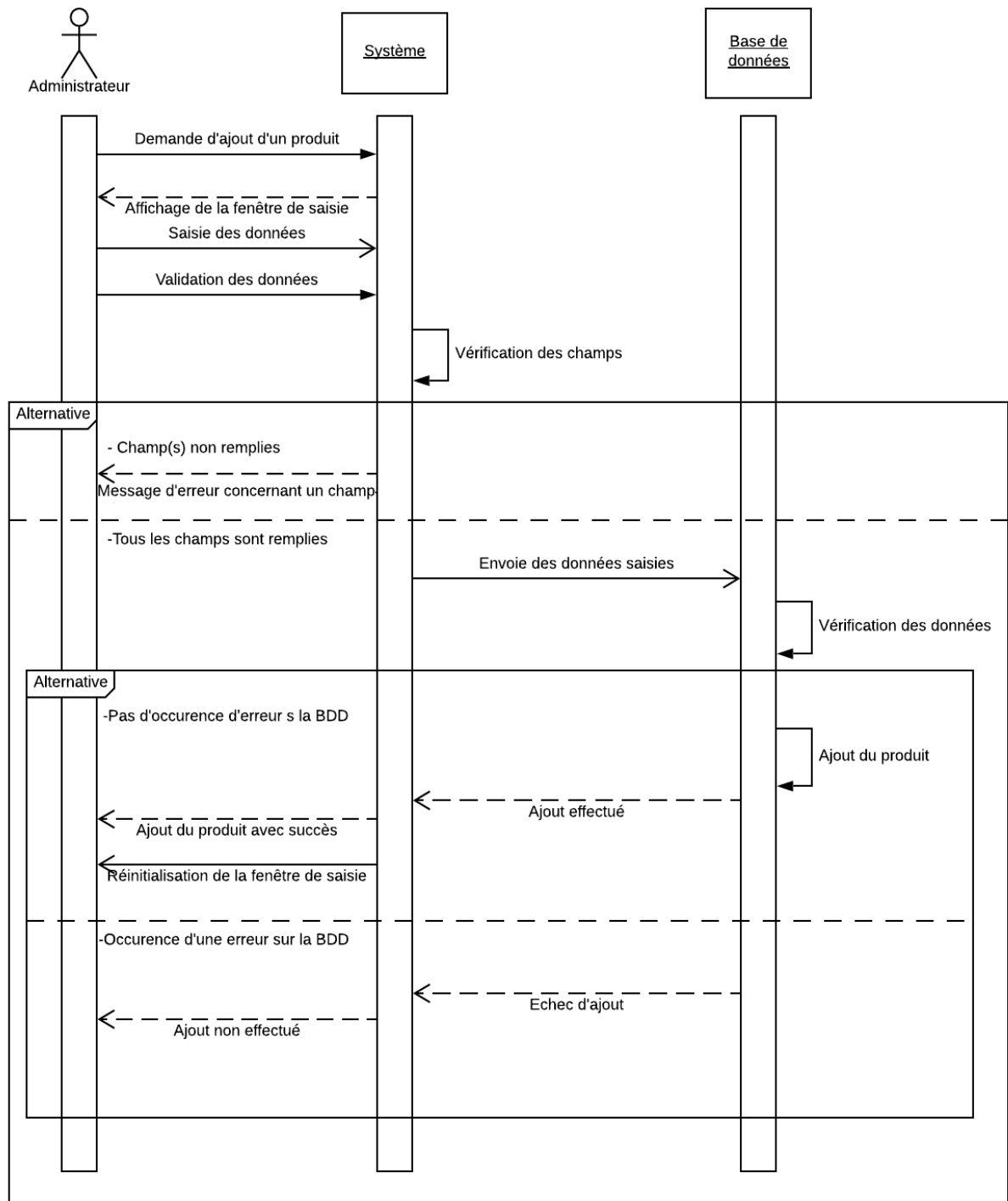
**Figure II.8 Diagramme de séquence du cas « Consultation des détails des acheteurs »**

## II.4.5 Diagramme de séquence du cas « Supprimer acheteur »



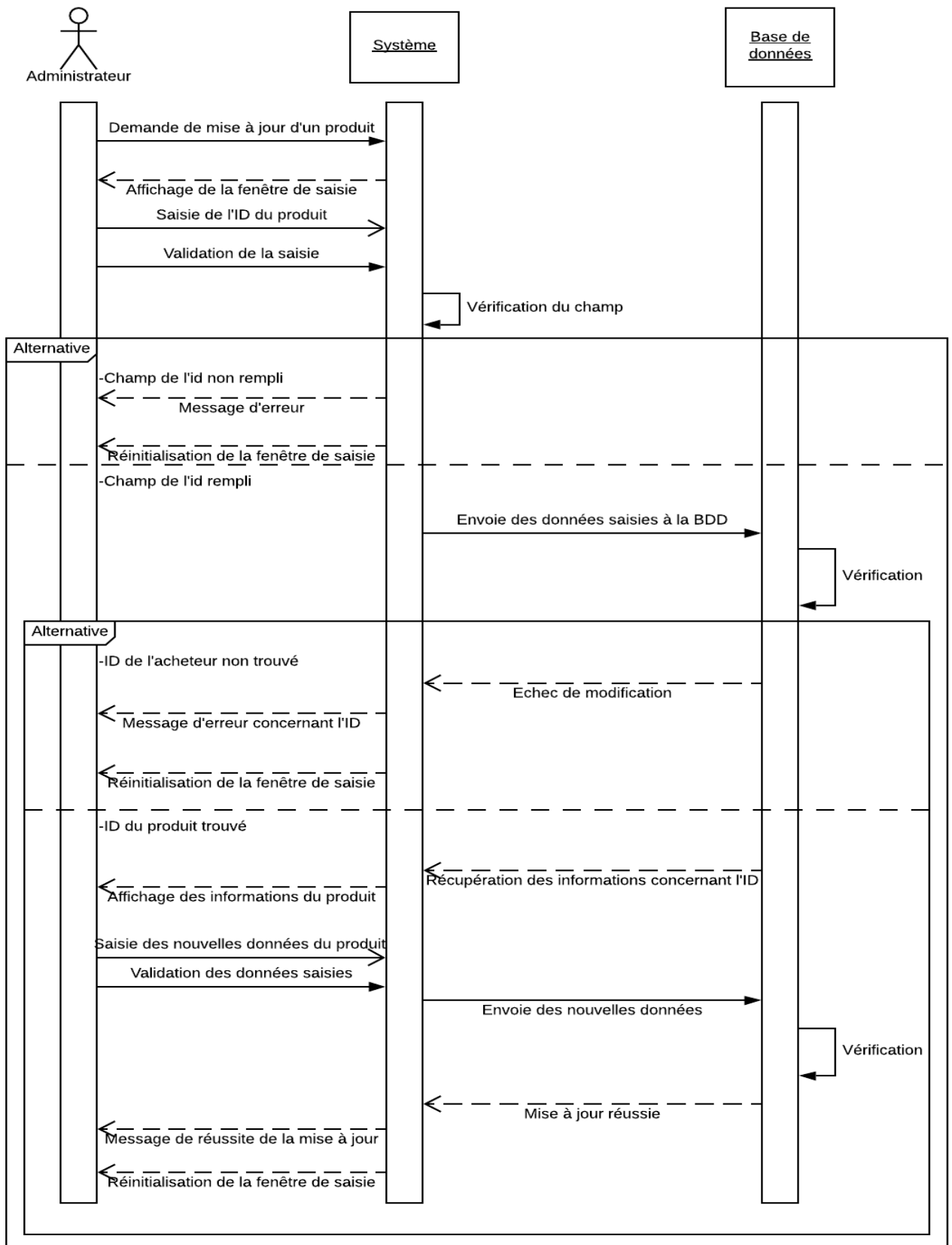
**Figure II.9 Diagramme de séquence du cas « Supprimer acheteur »**

## II.4.6 Diagramme de séquence du cas « Ajouter produit »



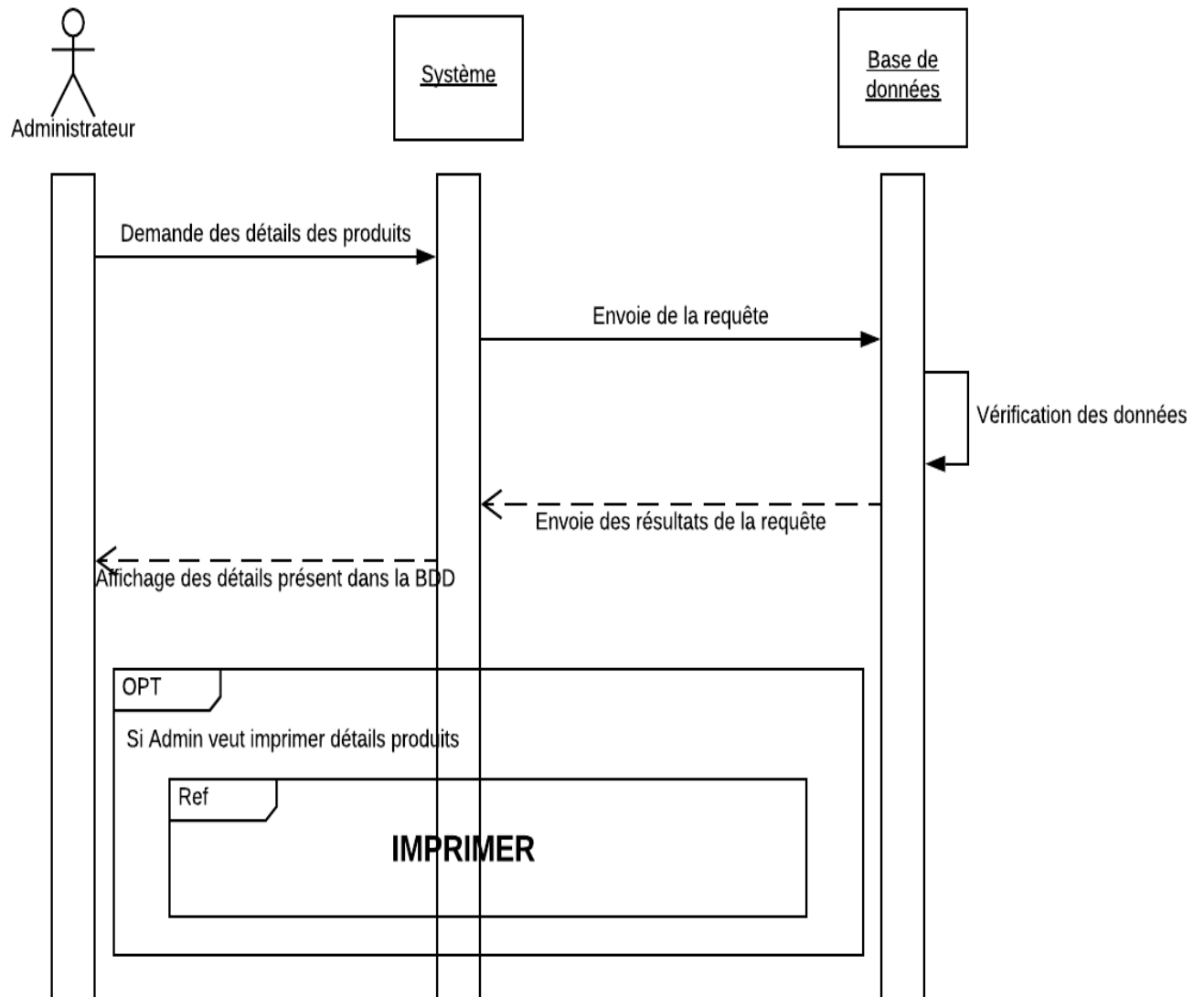
**Figure II.10 Diagramme de séquence du cas « Ajouter produit »**

## II.4.7 Diagramme de séquence du cas « Mise à jour produit »



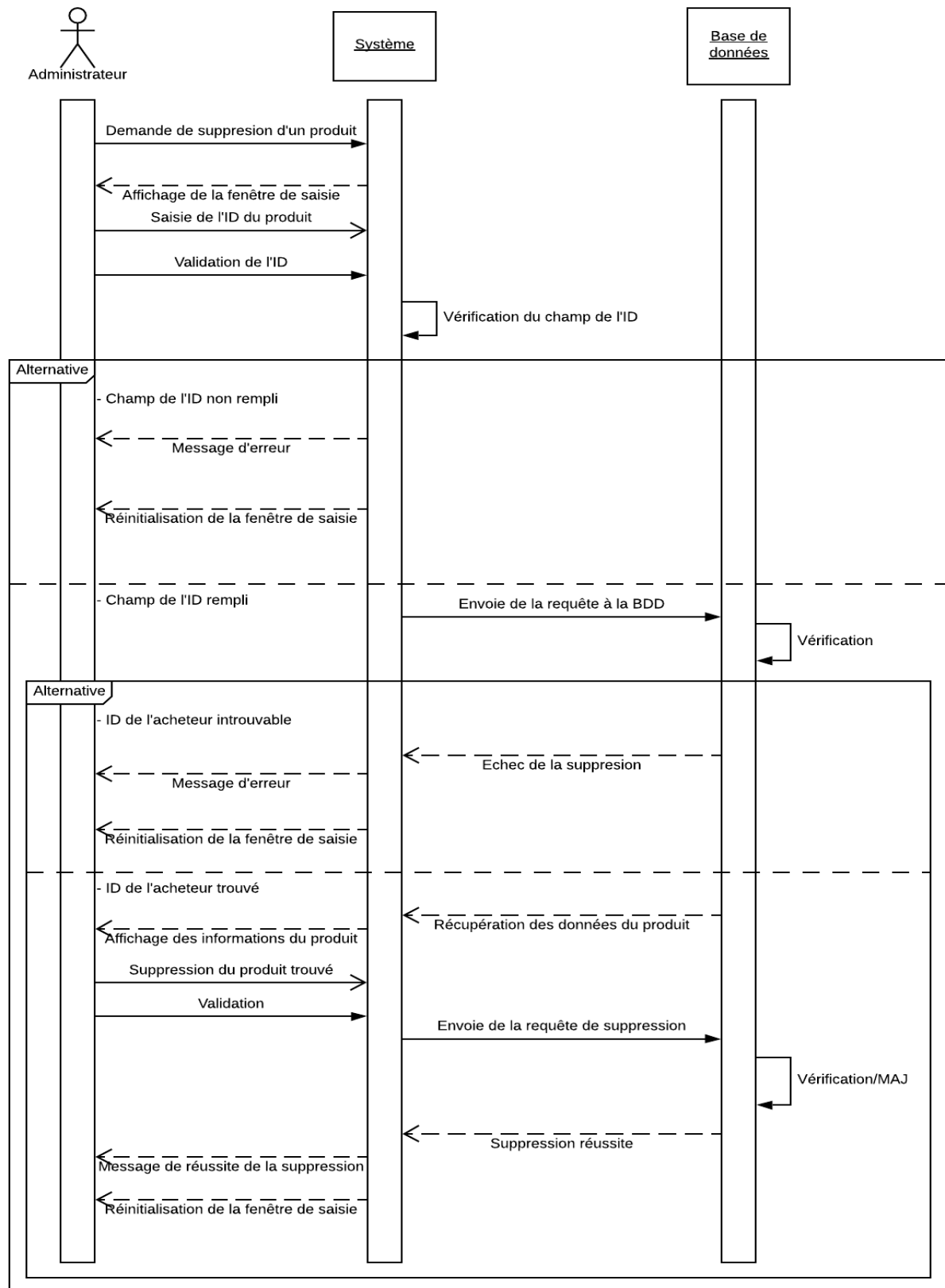
**Figure II.11 Diagramme de séquence du cas « MAJ produit »**

#### II.4.8 Diagramme de séquence du cas « consultation des détails des produits »



**Figure II.12 Diagramme de séquence du cas « Consultation des détails du produit »**

## II.4.9 Diagramme de séquence du cas « Supprimer produit »

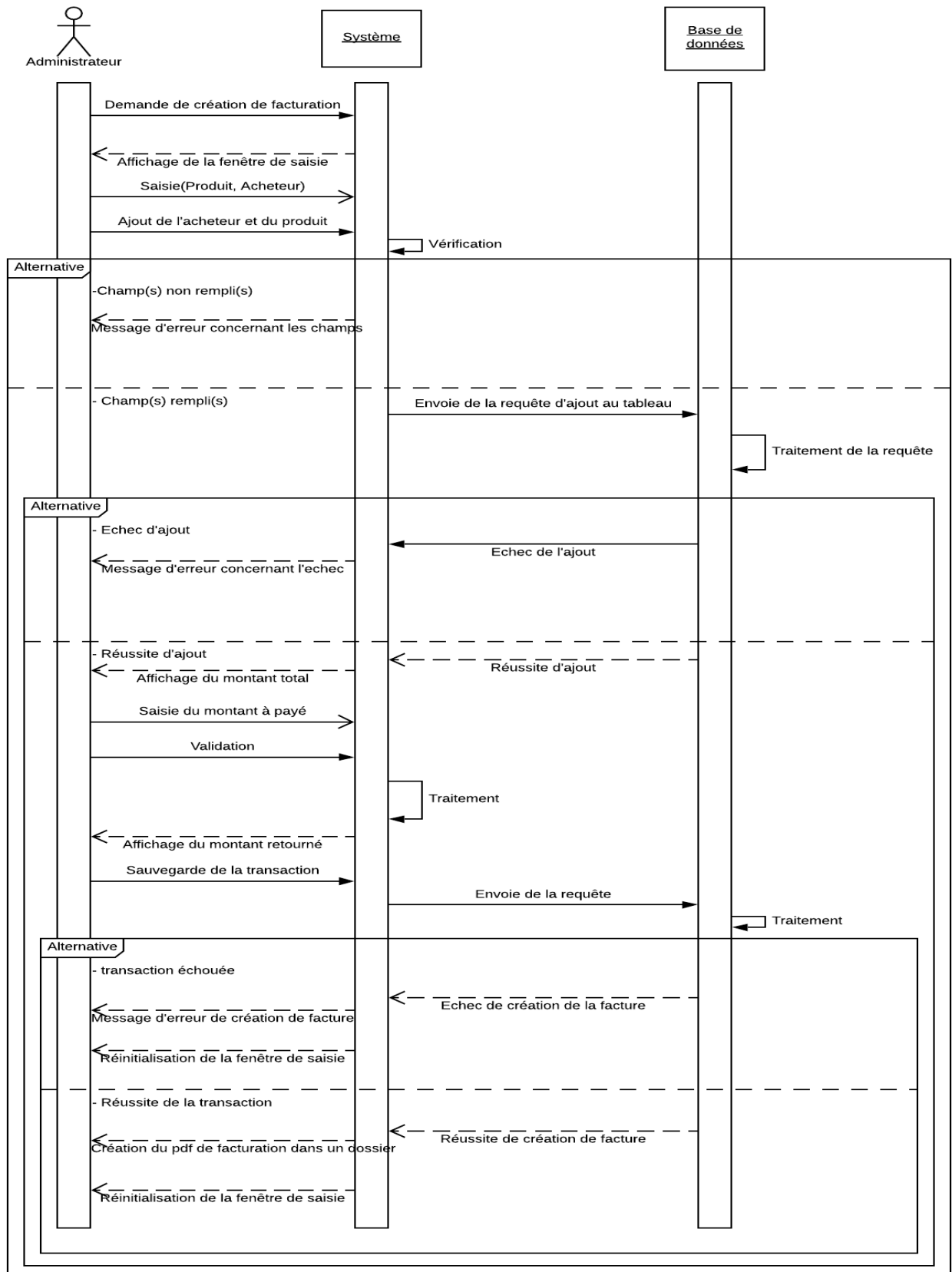


**Figure II.13 Diagramme de séquence du cas « Supprimer produit »**

»

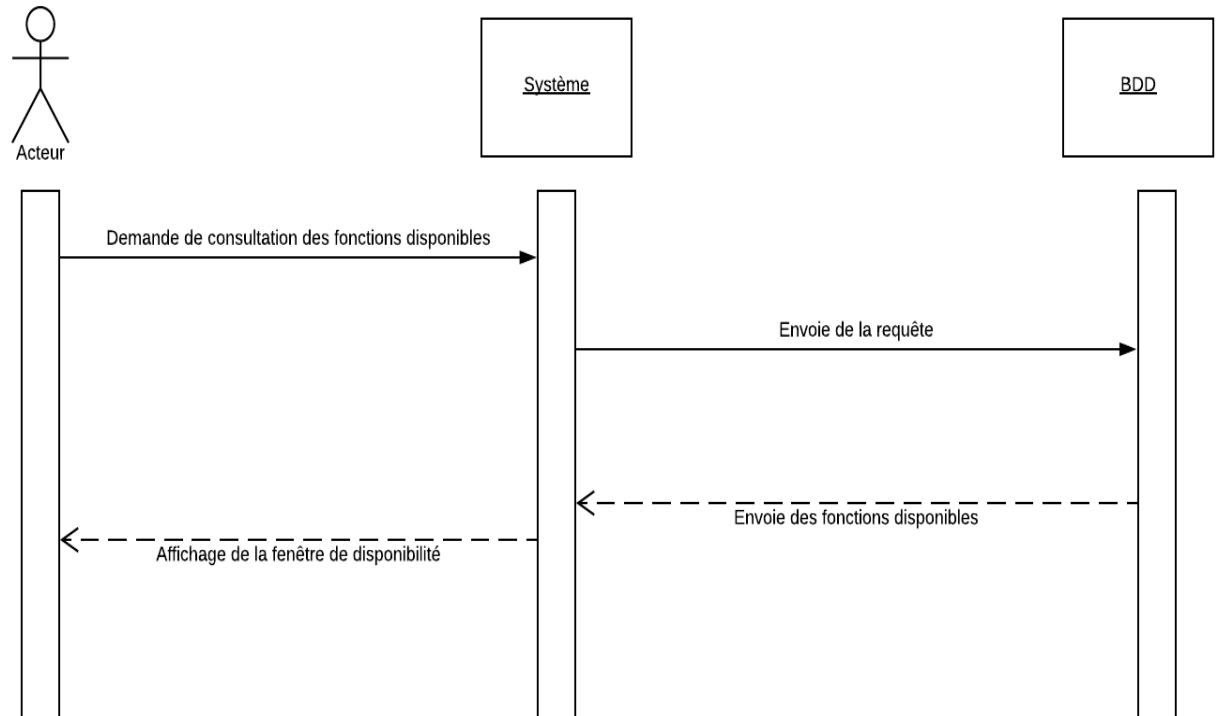


## II.4.10 Diagramme de séquence du cas « Créer facture »



**Figure II.14 Diagramme de séquence du cas « Créer facture »**

### II.4.11 Diagramme de séquence du cas « Consulter fonctions disponibles »



**Figure II.15 Diagramme de séquence du cas « Consulter fonctionnalités disponibles »**

## II.5 Le modèle relationnel

Dans ce qui suit je vous présenterai le modèle relationnel, il va nous aider à schématiser les informations présentes dans la base de données grâce à la puissance de sa simplicité.

**User** (Id\_user, user\_name, password).

**Buyer** (contactnumber, name, e-mail, address, gender).

**Product** (P\_id, P\_name, rate, description, activation).

**Fonction** (A\_ach, M\_ach, S\_ach, D\_ach, A\_prod, M\_prod, S\_prod, D\_prod).

**Facture** (#contactnumber, #P\_id).

## II.6 Conclusion

Au cours de ce deuxième chapitre nous avons vu en détails le plan de conception ainsi que la modélisation des échanges entre utilisateurs et système avec la base de données dans chaque fonctionnalité, à travers plusieurs diagrammes importants avec en dernier une présentation de cette base de données par le biais du modèle relationnel.

Grâce à ça on gagne un certain aperçu et description de la structure qu'a l'application, la prochaine partie se concentrera sur la réalisation du projet c'est-à-dire les outils ainsi que les fondements des technologies utilisées par ailleurs on passera vers les interfaces de l'application.

## III. Réalisation du système

### III.1 Introduction

Ce dernier chapitre va traiter de la dernière partie de ce rapport qui a pour but de décrire les outils utilisés pendant la conception, ensuite une présentation de quelques interfaces avec lesquelles l'administrateur aura à interagir pendant l'utilisation de l'application.

### III.2 Environnement de réalisation

#### III.2.1 Lucidchart

Lucidchart est une plateforme de collaboration en ligne permettant la création de diagrammes et une bonne visualisation des données, l'interface est accessible via n'importe quel navigateurs web, elle permet de créer des diagrammes via glissement et dépôt de conteneurs.

C'est une excellente alternative gratuite pour quelques programmes de création payants.



**Figure III.1 Logo de Lucidchart**

### III.3 Environnement de développement

#### III.3.1 NetBeans

NetBeans est un environnement de développement intégré développé par Apache qui est aussi open source, il supporte une large variété de langage de programmation.



**Figure III.2 Logo de NetBeans**

### III.3.2 SQLite

SQLite est un moteur de bases de données relationnelles utilisant le langage SQL (Structured Query Language), son avantage est qu'il est intégré à l'application avec un fichier de base de données propre à lui-même et évite de reproduire le schéma client-serveur inutile dans ce cas-là.



**Figure III.3 Logo de SQLite**

## III.4 Outils de développement

### III.4.1 La machine virtuelle Java (JVM)

La JVM est un environnement d'exécution dans lequel le bytecode est exécuté suivant certaines instructions ce qui permet de produire un même résultat d'exécution du code quel que soit la plateforme sur lequel il opère.

### III.4.2 Environnement d'exécution Java (JRE)

Java Runtime Environment est un ensemble d'outils qui permettent le développement d'applications java, il contient des libraires et des bibliothèques que la JVM utilise pendant l'exécution

### III.4.3 Le kit de développement Java (JDK)

Java development kit désigne un ensemble de bibliothèques du langage de programmation java ainsi que les outils nécessaires à la compilation du code java pour être transformé en bytecode et envoyé à la machine virtuelle.

## III.5 Langages de programmation utilisés

Ici je présente les langages de programmations utilisés lors de la réalisation de mon application

### III.5.1 Java

Java est un langage de programmation orienté objet dont la syntaxe ressemble à celle du langage C, il est très utilisé pour la conception d'applications d'entreprise ainsi que les applications mobiles.

### III.5.2 Structured Query Language (SQL)

SQL est un langage de requête structuré, il a été utilisé dans mon projet pour exploiter pleinement les bases de données relationnelles.

## III.6 Présentation de l'application

### III.6.1 Nom de l'application

Mon application porte le nom de « **AdminHelper** », « **Admin** » pour administrateur et « **Helper** » en anglais signifiant aide en français, je pense que le sens saute aux yeux c'est une application qui aidera l'administrateur dans son travail.

### III.6.2 Logo de l'application

Dans la figure qui suivra je vous présenterai le logo de mon application :



**Figure III.4 Logo**

## III.7 Présentation des interfaces d'utilisation

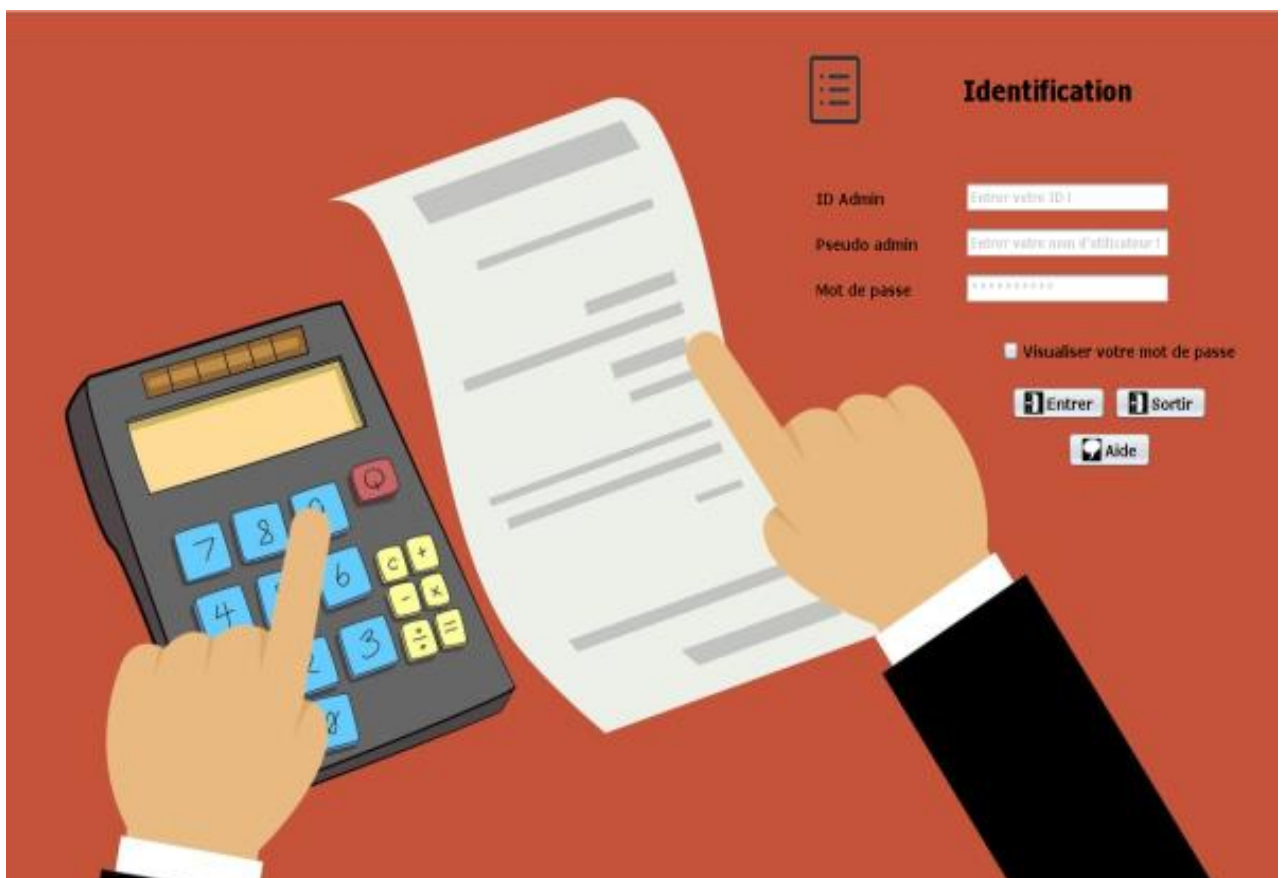
Dans cette section, je vais principalement vous présenter les interfaces importantes de mon application.

### III.7.1 Interface d'identification

Lors de l'ouverture de la session l'administrateur doit passer par une identification et ceci pour le bien-être de l'application, il doit saisir son ID d'admin, son pseudo d'admin et bien évidemment son mot de passe unique.

Si c'est sa première utilisation, il peut demander de l'aide en cliquant sur le bouton d'« **aide** », si les informations saisies sont correctes et qu'il clique sur « **entrer** » il accèdera à l'interface principale par contre si les informations sont fausses un message d'erreur apparaîtra, s'il est confus il peut redemander de l'aide ou bien sortir complètement de l'application en cliquant sur le bouton « **sortir** ».

L'identification est illustrée dans la figure qui suivra :



**Figure III.5 Interface d'identification**

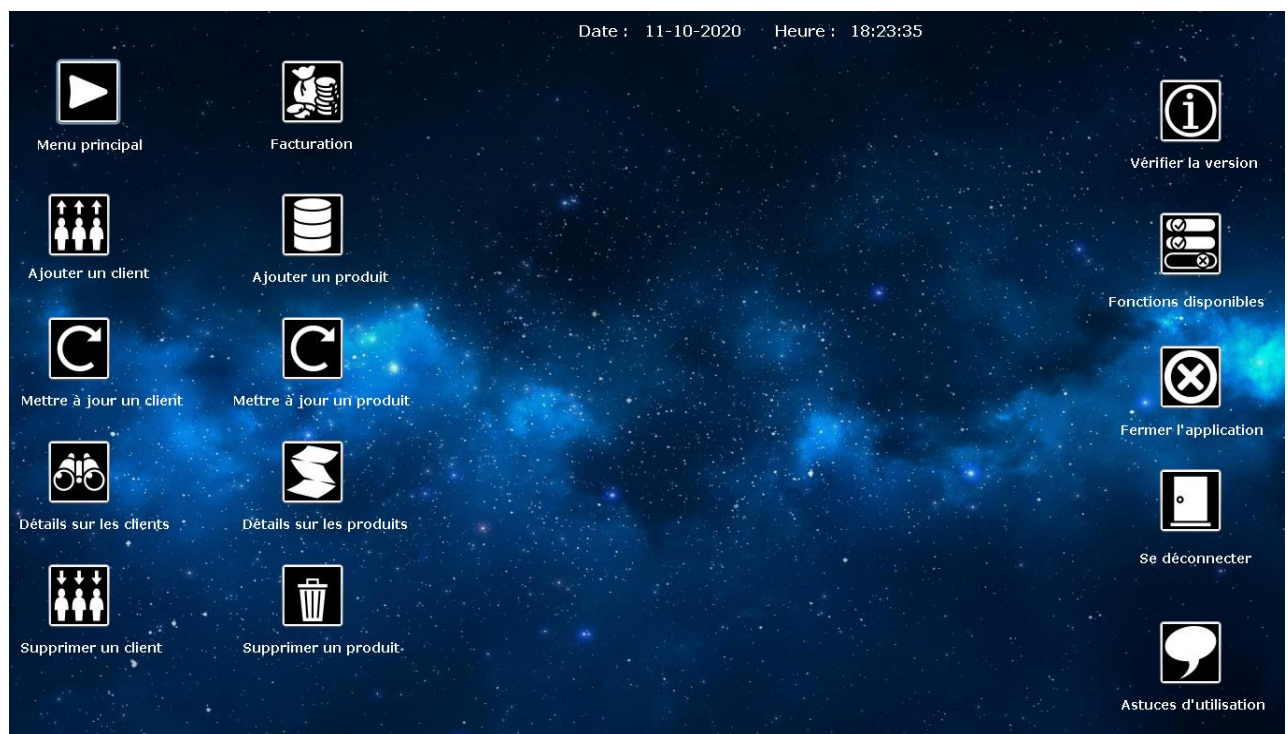
### III.7.2 Interface d'accueil principale



Comme illustré bien avant quand l'administrateur arrive à s'identifier il accède à une interface d'accueil, bien évidemment l'administrateur peut consulter des bulles d'aide en cliquant sur « **Aide à l'utilisation** ».

Vous pouvez remarquer tous les différents boutons certains complètement à gauche et les autres à droite, ceux de gauche sont les boutons utilitaires qui interviennent lors du travail de l'administrateur, il pourra alors faire toutes sortes d'opération telles qu'« **Ajouter** » soit des clients ou des produits, « **Mettre à jour** » ou « **supprimer** » ou « **consulter** » les données des clients et des produits de même pour la **facturation**.

Tant dis que les boutons de droite serviront plutôt à l'entretien de l'application en consultant une potentielle mise à jour aussi vérifier que si les fonctionnalités précédemment établies seront bien opérantes et non en maintenance, et évidemment deux boutons qui serviront à se déconnecter de la session ou sortir complètement de l'application !



**Figure III.6 Interface d'accueil principale**

### III.7.3 Interface d'ajout (Client, Produit)

Ici je présente deux interfaces qui ont du moins la même fonction c'est-à-dire l'ajout en base de données de potentiels clients de même que des produits d'achats.

Dans les deux interfaces vous remarquerez qu'elles présentent les mêmes caractéristiques.

Une partie saisie de données relative soit au client ou au produit et une deuxième partie, où on distingue trois boutons un pour ajouter un client/produit, un deuxième pour réinitialiser à zéro la fenêtre et enfin un dernier bouton pour fermer cette interface même.

The screenshot displays a software interface for adding a client. The main window is titled "Ajouter un client" and features a dark blue background with a starry pattern. At the top, it shows the date "Date : 11-10-2020" and the time "Heure : 18:23:35". The central form contains the following fields and labels:

- Nom et prénom du client: Enter le nom
- Numéro du client: Enter le numéro du contact
- E-mail: Enter l'e-mail
- Adresse: Enter l'adresse
- Sexe: Homme (dropdown menu)

Below the form are three buttons: "Ajouter", "Réinitialiser", and "Fermer". The interface is surrounded by a grid of icons for various functions:

- Menu principal
- Facturation
- Ajouter un client
- Ajouter un produit
- Mettre à jour un client
- Mettre à jour un produit
- Détails sur les clients
- Détails sur les produits
- Supprimer un client
- Supprimer un produit
- Vérifier la version
- Fonctions disponibles
- Fermer l'application
- Se déconnecter
- Astuces d'utilisation

**Figure III.7 Interface d'ajout client**



**Figure III.8 Interface d'ajout produit**

#### III.7.4 Interface de mise à jour (Client, Produit)

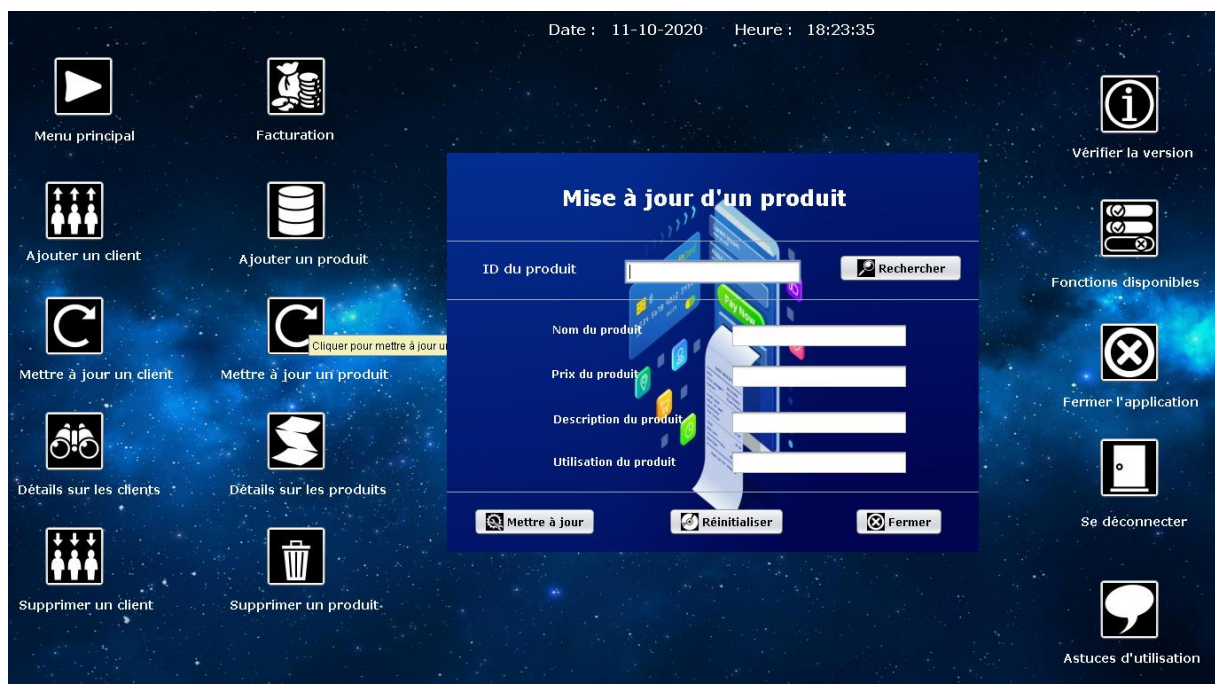
Comme pour les deux interfaces d'ajout décrites précédemment, ces deux interfaces ont aussi la même fonction qui est de mettre à jour les données saisies après ajout soit d'un client ou d'un produit et comme les précédentes il y a deux parties.

Une pour la saisie et recherche du produit/client en précisant soit l'ID ou un numéro de contact respectivement et une autre partie contenant des boutons, les deux boutons de réinitialisation et de la fermeture de la fenêtre n'ont pas changé mais au lieu de l'ajout c'est celui de la « **mise à jour** » qui a pris sa place.





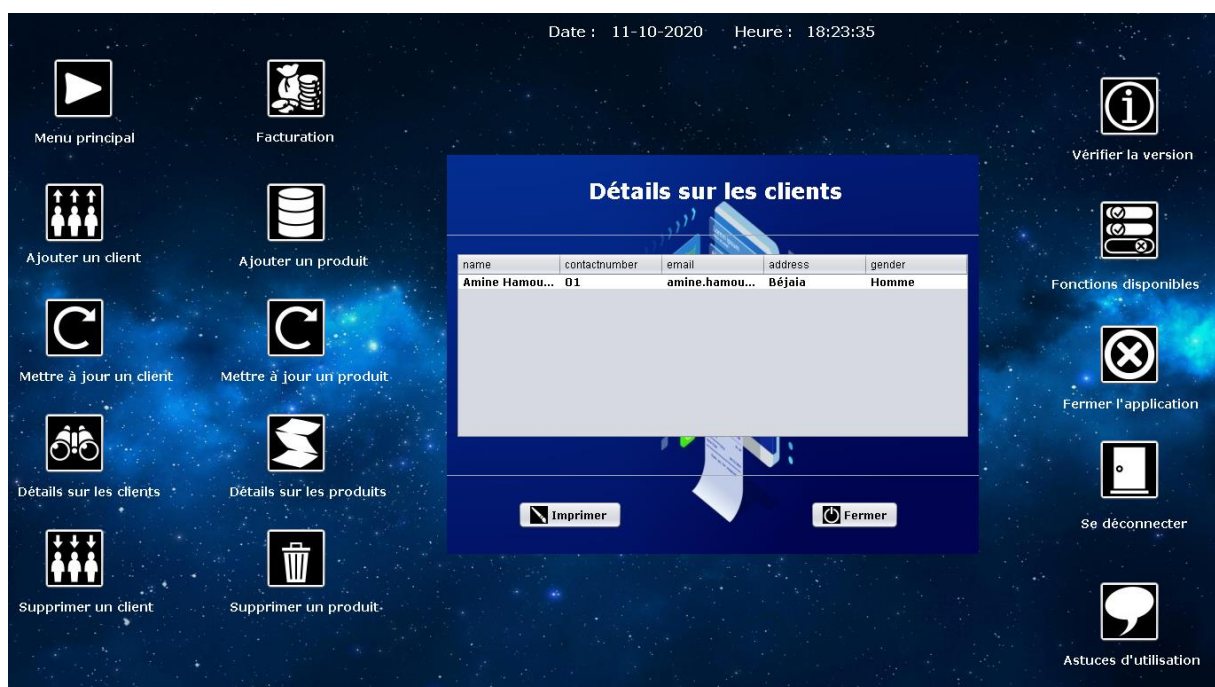
**Figure III.9 Interface de mise à jour d'un client/acheteur**



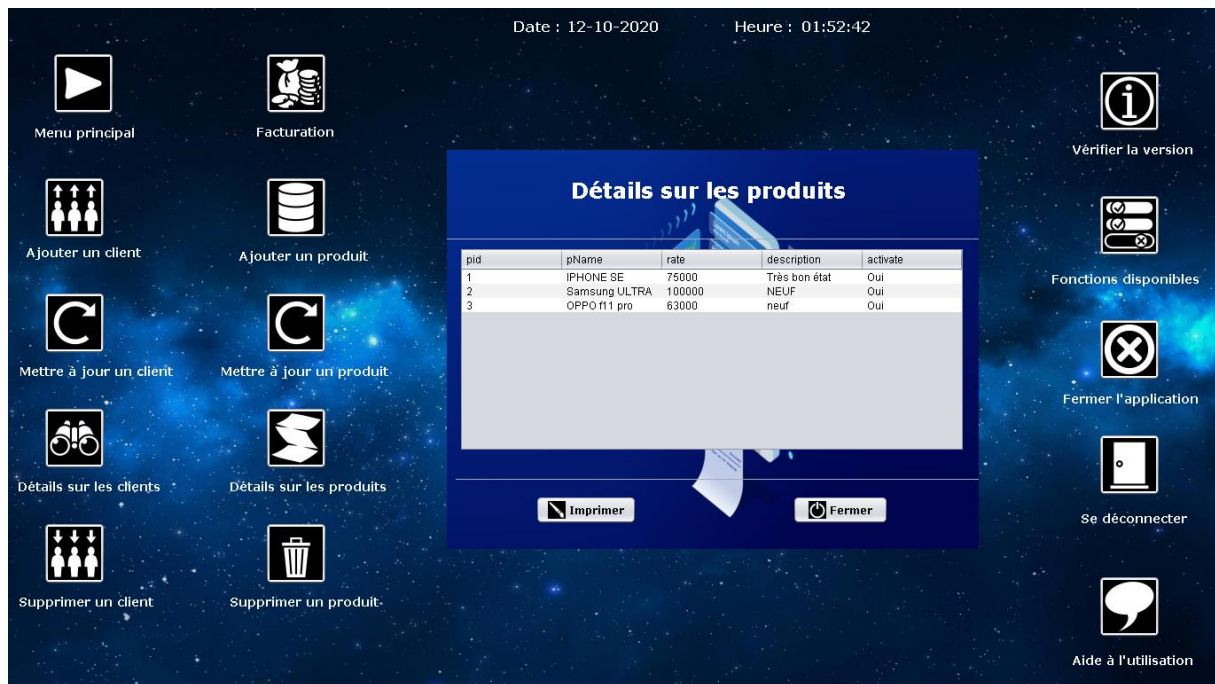
**Figure III.10 Interface de mise à jour d'un produit**

### III.7.5 Interface consultation des données (Client, Produit)

Contrairement aux autres types d'interface celle-ci n'est constituée que d'un tableau où les informations sont affichées selon les différents attributs les caractérisant, avec en option d'imprimer ces dernières en suivant une démarche avec bien entendu l'imprimante branchée avec le pc ou dans un réseau.



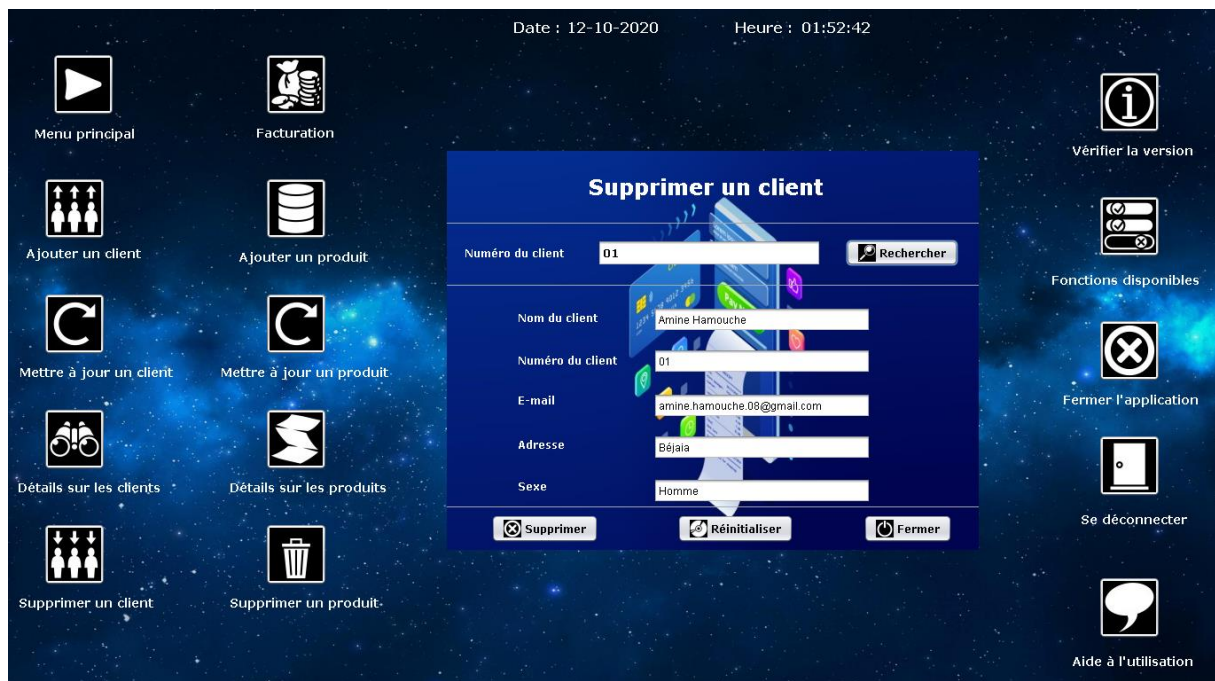
**Figure III.11 Interface de consultation des informations client**



**Figure III.12 Interface de consultation des informations produit**

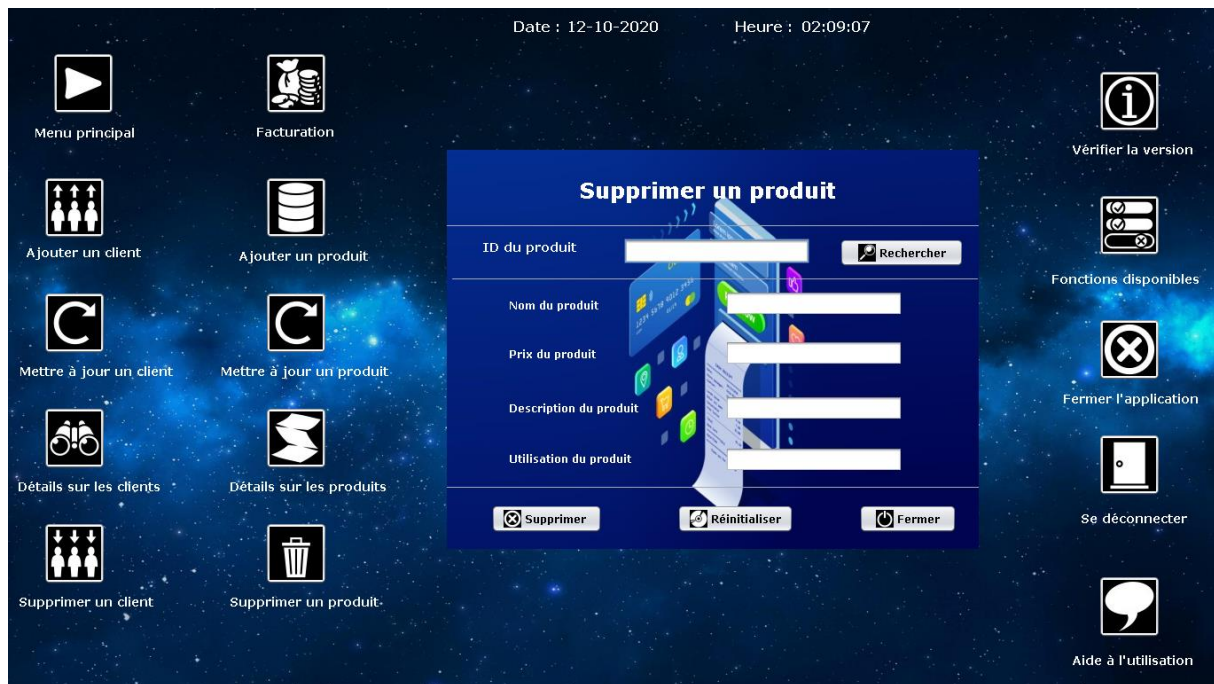
### III.7.6 Interface suppression des données (Client, Produit)

Interface très semblable à celle de la mise à jour en cherchant soit un client ou un produit, l'administrateur peut entamer la suppression définitive de l'objet en cliquant sur le bouton « **supprimer** ».



**Figure III.13 Interface de suppression d'un client**





**Figure III.14 Interface de suppression d'un produit**

### III.7.7 Interface de facturation (1 Client, x Produits)

Cette dernière interface présente dans la partie gauche du menu est l'aboutissement des différentes fonctions qu'établira l'administrateur. En effet, en tapant le numéro de l'acheteur et en récupérant ses détails l'admin va pouvoir procéder à l'ajout des différents produits dans un panier choisis par le client, la liste des produits finalisée un montant à payer s'affichera ainsi qu'un espace où le client communiquera le montant qu'il déboursa.

Ceci dans un but précis qui est la création d'une facture où tous les détails de la transaction seront stipulés je rappelle qu'une facture est indispensable dans n'importe quel projet touchant à la vente ou à la gestion d'entreprise, la facture doit indiquer les renseignements concernant le revendeur avec la date et l'heure ainsi que la quantité avec le prix payé.

**NB : La facture créée est sous extension .PDF, l'administrateur pourra alors procéder à son impression.**

# Facturation

Date : 12-10-2020  
Heure : 23:03:44

## Détails de l'acheteur :

Nom du client :    
Numéro du client :   
E-mail :   
Adresse :   
Sexe :

## Détails du produit :

ID du produit :    
Nom du produit :   
Prix :   
Quantité :   
Description :

## Détails de la facture :

Montant total à payé :   
Montant payé :   
  
Montant retourné :

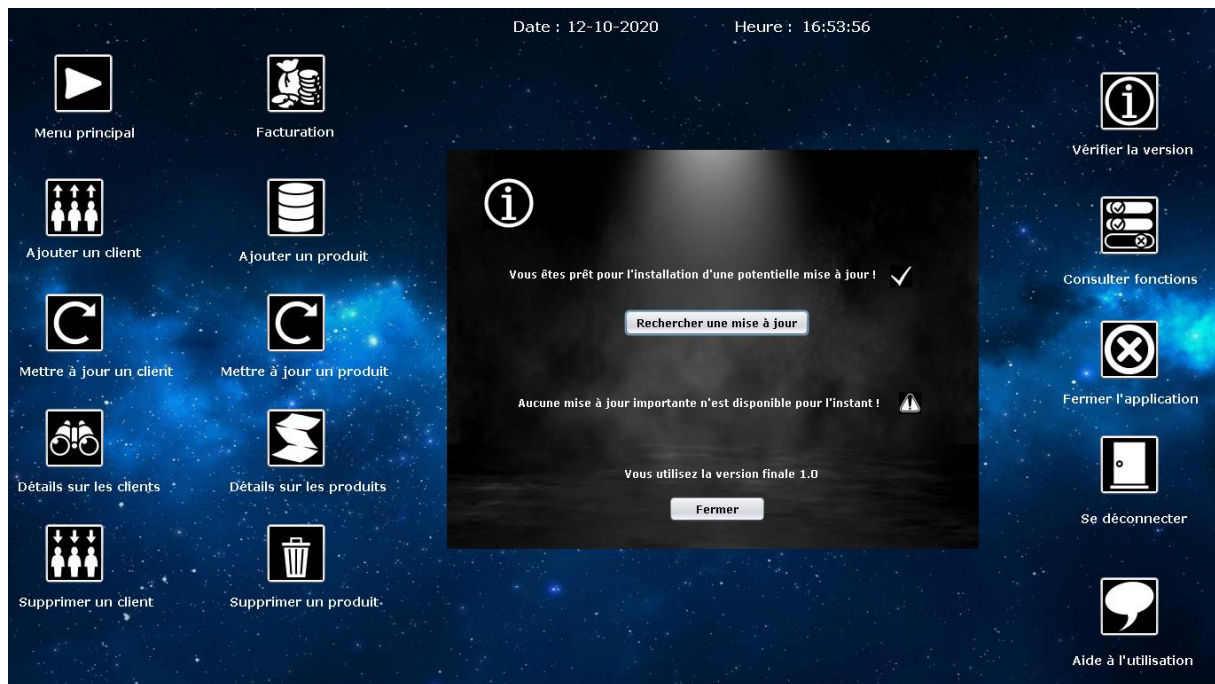
Nom	Prix	Quantité	Description	Total
IPHONE SE	75000	1	Très bon état	75000
Samsung ULTRA	100000	1	NEUF	100000
OPPO f11 pro	63000	1	neuf	63000

**Figure III.15 Interface de facturation**

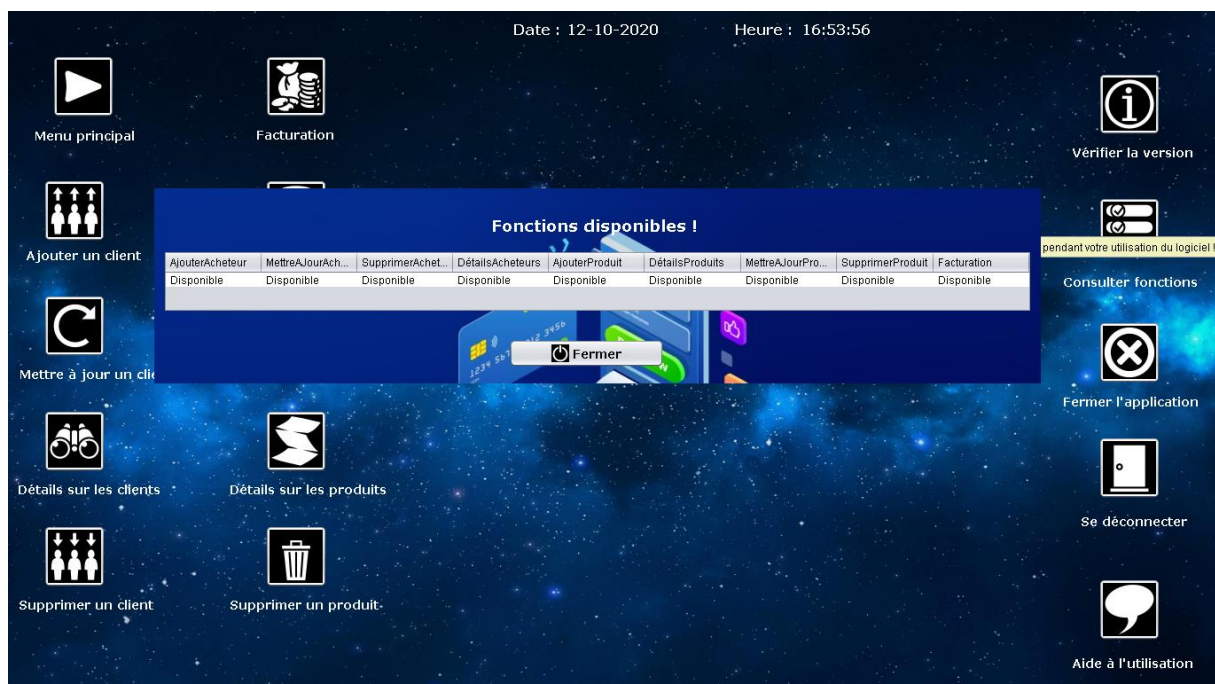
On a aussi quelques boutons à droite de l'interface principale dont on a évoqué le sujet un peu plus haut et que j'ai décrit comme étant des boutons complémentaires à l'application, en effet on peut « **vérifier** » la version d'utilisation ou bien en recherchant une nouvelle mise à jour de l'application.

Encore « **consulter** » les fonctions disponibles au cas où quelques fonctions à gauche seraient en maintenance temporairement, bien évidemment des boutons pour soit « **se déconnecter** » de la session ou bien « **fermer** » l'application directement.





**Figure III.16 Interface de vérification de mises à jour**



**Figure III.17 Interface de vérification de disponibilité de fonctions**

### III.8 Conclusion

Dans ce qui a précédé, j'ai présenté les divers outils avec lesquels j'ai pu entreprendre mon projet de l'outil de réalisation jusqu'à la présentation du nom de l'application ainsi que son logo en passant par les langages de programmation utilisés avec les outils de développement exploités pour finalement présenter les interfaces les plus importantes de celle-ci.

## Conclusion générale

À travers ce travail, j'ai pu réaliser et concevoir une application sur Windows qui sert à faciliter le travail d'un administrateur chargé de la gestion d'un stock de produits ainsi que de la vente et facturation de ceux-ci.

J'ai commencé ce mémoire par une brève introduction générale avec la description du modèle que j'allais suivre selon mon cycle de vie agile choisi, avec un aperçu de la problématique avec laquelle je ferai face et j'ai ainsi dressé la liste des fonctionnalités dont j'aurai à développer pendant l'étude de ce projet.

Ensuite, dans le chapitre suivant j'ai adressé le plan de conception et de réalisation modélisé par le langage de modélisation qui est UML ceci par le biais de divers diagrammes mais seulement les plus nécessaires tel que le diagramme de contexte système, ceux des cas d'utilisation avec les diagrammes de séquences de ces derniers ainsi qu'un diagramme de classes et finalement un croquis du modèle relationnel de la base de données utilisée.

Puis, j'ai commencé à réaliser le tout en présentant les technologies utilisées (langages de programmation et autres) ainsi que les interfaces les plus importantes liées à l'utilisation de mon application.

En effet, je peux dire que ce travail a accentué les connaissances que j'avais déjà en UML et par ailleurs me former dans de nouveaux secteurs que je n'ai pas eu l'occasion d'explorer que sont le design et la gestion de base de données.

En guise d'aspects futurs et de perspectives, l'application « **AdminHelper** » peut très bien être enrichie avec de nouvelles options d'utilisation comme par exemple :

- Un système de recouvrement de mots de passes au cas où un administrateur aurait oublié le sien.

- L'étude d'une importation de la plateforme en ligne pour faciliter l'envoi de facturation aux désireux d'acheter sans pour autant être obligé de se déplacer.
- Un système de messagerie pour créer une communication entre administrateurs.

## Bibliographie

[1] : Wikipédia, Développement basé sur les fonctionnalités  
<https://www.productplan.com/glossary/feature-driven-development>.

[2] : LucidChart, Qu'est-ce que l'UML ?  
<https://www.lucidchart.com/pages/fr/langage-uml>.

[3] : Différence entre JDK, JRE et JVM,  
<https://waytolearnx.com/2018/11/difference-entrejdk-jre-jvm.html>.

[4] : Les diagrammes de classes, Laurent Audibert  
<https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes>.

[5] : Les diagrammes de séquences, Laurent Audibert  
<https://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-interaction>.

[6] : Les bases de données relationnelles, Laurent Audibert  
<https://laurent-audibert.developpez.com/Cours-BD/?page=bases-de-donnees-relationnelles>.

[7] : Interfaces graphiques en Java – introduction, Université de Nice – Sophia Antipolis, Richard Grin.

[8] : Rédiger un cahier des charges, openclassroom  
<https://openclassrooms.com/fr/courses/6739646-realisez-un-cahier-des-charges-fonctionnel?archived-source=3676841>.