

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

CENTRE UNIVERSITAIRE DE MILA

INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de

LICENCE ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Informatique**

Thème

**CONCEPTION ET REALISATION D'UN
SYSTEME D'INFORMATION DE GESTION DE
STOCK DU CENTRE UNIVERSITAIRE DE MILA**

Présenté par : - Fekraoui Abdennour

- Bouhennache Nouredine

Sous la direction de : M^{elle} Khalfi Souheila

Année universitaire 2010-2011

DEDICACES

*A mes très chers parents qui m'ont beaucoup
soutenu, et encouragé ;*

A mes très chers sœurs Rima, Samiha, Madiha,

A ma petite belle-sœur Roumiassa

A mon beau-frère Karim ;

A tous mes amis de l'CUV ;

A toute la famille Fekraoui;

A toute la famille Merabet;

*A mon binôme et amie Noureddine qui j'ai
partagé ce modeste travail;*

*A tous ceux que j'aime tant et que je n'ai
pas cités.*

Je dédie ce mémoire ...

ABDENNOUR

DEDICACES

A mes très chers parents qui m'ont beaucoup soutenu, encouragé et qui ont fait de moi ce que je suis aujourd'hui ;

A mes chères sœurs

Hanane et Lamia ;

A mes beau frères

Abdelhamide, Abdesslame, SalahEdine ;

A mes oncles et tantes, ainsi que mes cousins et cousin ;

A tous mes amis en particulier

Sami, Riad, Abdelatife, Badis, Zaki ;

A tous mes amies en particulier

Wafia, Hamida et ces amies ;

A tous mes amis (es) 3^{em} année informatique en particulier

Abdenour ; Basma ; Imane ; Hayate ; Somai ;

A toute les familles

Bouhenache et Baouta;

A tous ceux que j'aime tant et que je n'ai pas cités ;

Je dédie ce mémoire ...

Nouredine bouhenache

REMERCIEMENTS

Nous désirons remercier des personnes importantes qui nous ont fortement aidés à concrétiser ce projet, que ce soit de près ou de loin.

Un grand merci pour :

- ❖ Toute l'équipe du Service des moyens, de l'inventaire et des archives.*
- ❖ Tous nos enseignants pour toutes les connaissances qu'ils nous ont inculquées.*
- ❖ A Melle S. KHALFI notre encadreur pour sa Disponibilité, son aide précieuse et ses conseils qui nous ont été d'une Utilité indéniable*

RÉSUMÉ

L'importance des flux de stock, pousse les entreprises à rechercher de nouveaux outils pouvant les aider à avoir une meilleure gestion de leur stock. Le centre universitaire de Mila est parmi les entreprises qui possèdent un stock important d'articles. Cependant, les activités qui contrôlent et traitent les mouvements du stock au niveau de ce magasin sont manuelles, ce qui rend la tâche du magasinier fastidieuse et complexe et ne peut entièrement les assumer. La fatigue, l'oubli et l'erreur ont de grandes répercussions sur la qualité du travail et par conséquent, un effet négatif sur les résultats attendus de l'organisation. Par conséquent, il est indispensable de développer un système d'information qui couvre les exigences réelles du magasin. Notre objectif alors, vise à concevoir une application qui prend en considération ces derniers. Cette application a été développée en utilisant les différents outils nécessaires aussi bien pour la modélisation que pour l'implémentation. En ce qui concerne la modélisation, nous avons utilisé le langage UML, quant à l'implémentation nous avons adopté l'environnement DELPHI.

Mots-clés : UML, ACCESS, DELPHI, METHODE DE CONSTRUCTION, GESTION DE STOCK.

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE

1 Contexte de travail	1
2 Problématique et motivations	1
3 Objectifs de travail	2
4 Organisation du mémoire	2

CHAPITRE I

LE LANGAGE DE MODELISATION UNIFIE (UML)

1.1 Introduction	3
1.2 Le langage de modélisation unifié (UML)	3
1.2.1 UML est une unification des méthodes	3
1.2.2 Les Diagrammes UML	4
1.2.2.1 Diagramme de cas d'utilisation	5
1.2.2.1.1 Elément des diagrammes de cas d'utilisation	5
1.2.2.1.2 Les relations entre acteurs	5
1.2.2.1.3 Les relations entre cas d'utilisation	5
1.2.2.2 Diagramme de classes	6
1.2.2.3 Diagramme d'objets	6
1.2.2.4 Diagramme de séquence	7
1.2.2.5 Diagramme d'états-transitions	8
1.2.2.6 Diagramme d'activités	8
1.2.2.7 Diagramme de composant	9
1.2.2.8 Diagramme de déploiement	10
1.3 Mise en œuvre d'UML	10
1.3.1 Identification des besoins et spécification des fonctionnalités	11
1.3.1.1 Identification et représentation des besoins : diagramme de cas d'utilisation	11
1.3.1.2 Spécification détaillée des besoins : diagrammes de séquence système	11
1.3.3 Maquette de l'IHM de l'application (non couvert par UML)	12
1.3.2 Phases d'analyse	13
1.3.2.1 Analyse du domaine : modèle du domaine	13
1.3.2.2 Diagramme de classes participantes	14
1.3.2.3 Diagrammes d'activités de navigation	15
1.3.3 Phase de conception	17
1.3.3.1 Diagramme d'interaction	17
1.3.3.2 Diagramme de classes de conception	19
1.4 Conclusion	20

CHAPITRE II

ETUDE DE L'EXISTANT

2.1	Introduction	21
2.2	Présentation de l'organisme d'accueil (CENTRE UNIVERSITAIRE DE MILA)	21
2.3	Cadre de l'étude	23
2.3.1	Description des tâches effectuées au niveau du magasin	23
2.3.1.1	La réception des articles	23
2.3.1.2	Le traitement des bons de commandes internes	24
2.3.2	Etudes des postes de travail.....	24
2.3.3	Étude des documents	26
2.4	Conclusion.....	34

CHAPITRE III

ETUDE DE CAS

3.1	Introduction	35
3.2	Identification des cas d'utilisations	35
3.3	Description des cas d'utilisation	35
3.3.1	Cas d'utilisation authentification	35
3.3.2	Cas d'utilisation réception des articles.....	36
3.3.3	Cas d'utilisation consommation des articles	36
3.3.4	Cas d'utilisation Fiche de stock d'un article	37
3.3.5	Cas d'utilisation édition des articles en stock alerte	37
3.3.6	Cas d'utilisation Ajouter article	37
3.3.7	Cas d'utilisation Mise à jour des articles	38
3.3.7.1	Modifier article.....	38
3.3.7.2	Supprimer article	38
3.3.8	Cas d'utilisation Ajouter famille.....	39
3.3.9	Cas d'utilisation MAJ des familles	39
3.3.9.1	Modifier famille	39
3.3.9.2	Supprimer famille	40
3.3.10	Cas d'utilisation Ajouter fournisseur	40
3.3.11	Cas d'utilisation MAJ des fournisseurs	40
3.3.11.1	Modifier fournisseurs	40
3.3.11.2	Supprimer fournisseurs	41
3.3.11	Cas d'utilisation Ajouter service.....	41
3.3.12	Cas d'utilisationMAJ des services	42
3.3.12.1	Modifier service	42
3.3.12.2	Supprimer service	42
3.4	Diagramme de cas d'utilisation.....	43
3.5	Les diagrammes de séquence	44
3.5.1	Authentification.....	44
3.5.2	Réception des articles.....	45
3.5.3	Consommation des articles	46

3.5.4	Fiche de stock d'un article	47
3.5.5	Édition des articles en stock alerte	48
3.5.6	Ajouter article.....	49
3.5.7	Mise à jour des articles.....	50
3.5.7.1	Modifier article.....	50
3.5.7.2	Supprimer article	51
3.5.8	Ajouter famille	52
3.5.9	Mise à jour des familles	53
3.5.9.1	Modifier famille	53
3.5.9.2	Supprimer famille.....	54
3.5.10	Ajouter fournisseur.....	55
3.5.11	Mise à jour des fournisseurs.....	56
3.5.11.1	Modifier fournisseur.....	56
3.5.11.2	Supprimer fournisseur	57
3.5.12	Ajouter service	58
3.5.13	Mise à jour des services	59
3.5.13.1	Modifier service	59
3.5.13.2	Supprimer service.....	60
3.6	Le diagramme de classe	61
3.7	Conclusion.....	62

CHAPITRE IV

DÉVELOPPEMENT DE L'APPLICATION

4.1	Introduction	63
4.2	Le passage du diagramme de classe au modèle relationnel	63
4.3	Environnement de développement de l'application	65
4.3.1	Pourquoi Delphi ?.....	65
4.3.2	Implémentation de la base de données sous Access	65
4.4	Contrôle est sécurité	65
4.4.1	Contrôle.....	65
4.4.2	Sécurité.....	66
4.5	Interfaces de l'application.....	66
4.5.1	Interface authentification.....	66
4.5.2	Interface du menu principale (Gestion de stock)	67
4.5.3	Interface Famille d'articles.....	67
4.5.4	Interface article.....	68
4.5.5	Interface service	69
4.5.6	Interface fournisseur.....	69
4.5.7	Interface livraison.....	70

CONCLUSION GÉNÉRALE ET PERSPECTIVES

1	Bilan	71
2	Perspectives	71
	BIBLIOGRAPHIE	72

LISTE DES FIGURES

Figure 1.1	Quelle méthode pour passer de l'expression des besoins au code de l'application?
Figure 1.2	Les besoins sont modélisés par un diagramme de cas d'utilisation.....
Figure 1.3	Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.....
Figure 1.4	Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.....
Figure 1.5	La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.
Figure 1.6	Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.....
Figure 1.7	Les diagrammes d'activités de navigation représentent graphiquement les activités de navigation dans l'interface.
Figure1.8	Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.
Figure 1.9	Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d'objets en collaboration.....
Figure 1.10	Chaîne complète de la démarche de modélisation du besoin jusqu'au code.
Figure 2.1	Organigramme Du Centre Universitaire de Mila.....
Figure 2.2	Hiérarchie du SMIA.....
Figure 3.1	Diagramme de cas d'utilisation
Figure 3.2	Diagramme de séquence du cas d'utilisation Authentification.....
Figure 3.3	Diagramme de séquence du cas d'utilisation Réception des articles.....
Figure 3.4	Diagramme de séquence du cas d'utilisation Consommation des articles
Figure 3.5	Diagramme de séquence du cas d'utilisation Fiche de stock d'un article
Figure 3.6	Diagramme de séquence du cas d'utilisation Edition des articles en stock alerte
Figure 3.7	Diagramme de séquence du cas d'utilisation Ajouter article.....
Figure 3.8	Diagramme de séquence du cas d'utilisation Modifier article.....
Figure 3.9	Diagramme de séquence du cas d'utilisation Supprimer article
Figure 3.10	Diagramme de séquence du cas d'utilisation Ajouter famille
Figure 3.11	Diagramme de séquence du cas d'utilisation Modifier famille
Figure 3.12	Diagramme de séquence du cas d'utilisation Supprimer famille
Figure 3.13	Diagramme de séquence du cas d'utilisation Ajouter fournisseur.....
Figure 3.14	Diagramme de séquence du cas d'utilisation Modifier fournisseur.....
Figure 3.15	Diagramme de séquence du cas d'utilisation Supprimer fournisseur.....
Figure 3.16	Diagramme de séquence du cas d'utilisation Ajouter service
Figure 3.17	Diagramme de séquence du cas d'utilisation Modifier service
Figure 3.18	Diagramme de séquence du cas d'utilisation Supprimer service.....
Figure3.19	Diagramme de classe

Figure 4.1	Interface authentication
Figure 4.2	Interface menu principale (Gestion de stocke).
Figure 4.3	Interface Famille d'articles.
Figure 4.4	Interface article.
Figure 4.5	Interface service.
Figure 4.6	Interface fournisseur.
Figure 4.7	Interface livraison

LISTE DES TABLEAUX

Tableau 2.1: Fiche d'étude du poste de travail : Magasin.....	
Tableau 2.2: Documents manipulés	
Tableau 2.3: Fiche d'étude du document : Bon de commande.....	
Tableau 2.4: Fiche d'étude du document : Bon de livraison.....	
Tableau 2.5: Fiche d'étude du document : Facture.....	
Tableau 2.6: Fiche d'étude du document : liste des articles à commander.....	
Tableau 2.7: Fiche d'étude du document : Bon commande interne.....	
Tableau 2.8: Fiche d'étude du document : Bon de livraison de magasin.....	
Tableau 2.9: Fiche d'étude du document : Décharge.....	
Tableau 2.10: Fiche d'étude du document : Affectation Matériel.....	
Tableau 2.11: Fiche d'étude du document : fiche de stock.....	
Tableau 2.12: Fiche d'étude du document : registre d'entrée.....	
Tableau 3.1 : Les cas d'utilisations.....	

LISTE DES ACRONYMES

UML	Unified ModelingLanguage
UP	Unified Process
XP	eXtreme Programming
SMIA	Service des Moyens, d'Inventaire et des Archives
CSMIA	chef Service des Moyens, d'Inventaire et des Archives
RAD	Rapide Application Développement
ODBC	Open Data Base Connectivity

INTRODUCTION GÉNÉRALE

1 Contexte de travail

Durant ces dernières années l'informatique s'est imposée d'une manière très impressionnante dans les entreprises. Elle se développe de jour en jour afin de nous faciliter la tâche dans tous les domaines d'activités y compris celui de la gestion de flux de stock. L'importance de cette dernière, pousse les entreprises à rechercher de nouveaux outils pouvant les aider à avoir une meilleure gestion des différents traitements exigés par cette activité.

Un système de gestion de stock constitue l'âme de l'entrepôt et sans lui l'entrepôt ne serait plus un centre de distribution efficace. Il permet de gérer en temps réel les quantités de marchandises en se basant sur les entrées / sorties d'un produit. Les fonctionnalités de ce système sont nombreuses, on peut citer à titre d'exemple : l'enregistrement des commandes des clients, la demande d'approvisionnement auprès du fournisseur, le suivi des commandes livrées ainsi que la comptabilité qui consiste à enregistrer les factures et de suivre les paiements des fournisseurs et des clients... Etc.

2 Problématique et motivation

Pendant notre stage au niveau du magasin du Centre Universitaire De Mila, nous avons pu constater que l'ensemble des traitements se fait manuellement, ce qui engendre un certain nombre de problèmes entre-autre :

- La difficulté de suivre le mouvement des articles stockés dans le magasin,
- La lenteur dans l'accès aux informations en temps réel parce que ces dernières sont stockées dans les registres,
- La vérification des articles en stock alerte est effectuée manuellement,
- La difficulté d'établir les bons de sortie qu'il faut affecter aux services demandeurs internes (affectation matériel, décharge, bonlivraison de magasin),
- Et le risque de perte d'informations.

Au cœur de ces problèmes se situe la nécessité de construire un système d'information pour la gestion du stock afin de faciliter la tâche du magasinier et d'assurer l'accès instantané aux données et une sécurisation de ces dernières.

Malheureusement, le magasinier ne dispose pas d'une telle application qui l'assiste dans son travail, de ce fait, nous allons nous occuper de sa réalisation.

3 Objectif de travail

Notre objectif consiste alors à la conception et la réalisation d'un système d'information de gestion de stock permettant de répondre aux besoins du magasinier. Plus précisément, il prend en charge la gestion des articles dès leur réception depuis les fournisseurs à leur livraison aux services internes tout en assurant une gestion des seuils qui permet de contrôler la quantité des articles en stock.

Pour atteindre cet objectif, nous avons utilisé le langage UML (Unifier Modeling Language) dans la phase de conception tout en étant encadrés par la démarche proposée par Laurent AUDIBERT[3]. Pour l'implémentation, notre choix s'est porté sur l'environnement de développement DELPHI et la base de données est implémentée avec ACCESS.

4 Organisation du mémoire

Ce mémoire est structuré de la manière suivante :

- ◆ *Le chapitre I* s'intéresse à présenter le langage de modélisation UML et à décrire la démarche de développement utilisée.
- ◆ *Le chapitre II* est dédié à présenter l'organisme d'accueil qui est le centre Universitaire De Mila dans lequel nous avons effectué notre stage. Il consiste aussi à étudier les différents postes de travail et les divers documents manipulés au sein du magasin, pour y arriver enfin à construire un cahier de charge et ce, dans le but de préparer la conception et la réalisation du système.
- ◆ *Le chapitre III* est consacré à l'analyse de l'étude de cas.
- ◆ *Le chapitre IV* sert à présenter les outils dont nous nous sommes servis pour le développement de l'application ainsi qu'une brève description de quelques interfaces de cette application.
- ◆ Ce modeste mémoire s'achève par une conclusion générale en présentant un récapitulatif de tout ce que nous avons réalisé tout en planifiant les perspectives que nous envisageons pour compléter ce travail.

CHAPITRE I

LE LANGAGE DE MODELISATION UNIFIE (UML)

1.1 Introduction

Dans le cadre de ce chapitre, nous ferons tout d'abord un survol sur quelques généralités du langage de modélisation unifié UML (Unified Modeling Language). Par la suite, nous allons présenter la méthode de modélisation explicitant et encadrant toutes les étapes suivies pour la réalisation de notre projet.

1.2 Le langage de modélisation unifié (UML)

UML est une notion graphique conçue pour représenter, spécifier, construire et documenter les systèmes logiciels. Ses deux principaux objectifs sont la modélisation de système utilisant les techniques de l'orienté objet, depuis la conception jusqu'à la maintenance, et la création du langage abstrait compréhensible par l'homme et intraitable par les machines. Il permet de construire plusieurs model d'un système, chacun d'eux met en valeur des aspects différentes : fonctionnels, statiques, dynamiques, organisationnels. UML est devenu un langage incontournable dans les projets de développement. [1]

1.2.1 UML est une unification des méthodes

UML est né de la fusion des trois méthodes qui ont le plus influencées la modélisation objet au milieu des années 90 :

- OMT de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statiques, dynamiques et fonctionnels d'un système ;
- OOD de GradyBooch, définie pour le Department of Defense, introduit le concept de paquetage (package) ;
- OOSE d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

1.2.2 Les Diagrammes UML

UML dans sa version 2 s'articule autour de treize diagrammes, chacun d'entre eux est appliqué à la représentation d'un système logiciel suivant un point de vue particulier. Ces diagrammes sont regroupés dans deux grands ensembles: les diagrammes structurels et les diagrammes de comportement.

Diagrammes structurels ou diagrammes statiques (UML Structure)

- diagramme de classes (Class diagram)
- diagramme d'objets (Object diagram)
- diagramme de composants (Component diagram)
- diagramme de déploiement (Deployment diagram)
- diagramme de paquetages (Package diagram)
- diagramme de structures composites (Composite structure diagram)

➤ **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)**

- diagramme de cas d'utilisation (Use case diagram)
- diagramme d'activités (Activity diagram)
- diagramme d'états-transitions (State machine diagram)
- **Diagrammes d'interaction (Interaction diagram)**
- diagramme de séquence (Sequence diagram)
- diagramme de communication (Communication diagram)
- diagramme global d'interaction (Interaction overview diagram)
- diagramme de temps (Timing diagram) maîtrise

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions.

Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique. [3]

1.2.2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

1.2.2.1.1 Élément des diagrammes de cas d'utilisation

-Acteur : Représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. [2]

-Cas d'utilisation (use case) : Est une manière spécifique d'utiliser un système. Les acteurs sont à l'extérieur du système, ils modélisent tout qui interagit avec lui. Un cas d'utilisation réalise un service bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'a initié.[1]

1.2.2.1.2 Les relations entre acteurs

La seule relation entre acteur est la relation de **généralisation**. Quand un acteur fils hérite d'un acteur père, il hérite en réalité de toutes les associations du père. [2]

1.2.2.1.3 Les relations entre cas d'utilisation

-Relation d'inclusion : Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B. [2]

-Relation d'extension : Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation A signifie qu'une instance de A peut être étendue par le comportement décrit dans B. [2]

-Relation de généralisation: Les cas d'utilisation descendants héritent de la description de leurs parents communs. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires. [2]

1.2.2.2 Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. [3]

-Une classe : Représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type. [2]

-Un objet: Est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe. [2]

-Un attribut : Représente un type d'information contenu dans une classe. [2]

-Une opération: Représente un élément de comportement (un service) contenu dans une classe. [2]

-Une association: Représente une relation sémantique durable entre deux classes. [2]

-Une superclasse : Est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (**sous-classes**) par une relation de **généralisation**. Les sous-classes « Héritent » des propriétés de leur **superclasse** et peuvent comporter des propriétés spécifiques supplémentaires. [2]

-une méthode : un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions (appelées opérations) que l'objet est à même de réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (ou d'agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier. [3]

1.2.2.3 Diagramme d'objets

Un diagramme d'objets représente des objets (*i.e.* instances de classes) et leurs liens (*i.e.* instances de relations) pour donner une vue figée de l'état d'un système à un instant donné. Un diagramme d'objets peut être utilisé pour :

- illustrer le modèle de classes en montrant un exemple qui explique le modèle ;
- préciser certains aspects du système en mettant en évidence des détails invisibles dans le diagramme de classes ;

- exprimer une exception en modélisant des cas particuliers ou des connaissances non généralisables qui ne sont pas modélisés dans un diagramme de classe ; prendre une image (*snapshot*) d'un système à un moment donné.

1.2.2.4 Diagramme de séquence

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation. Et représenter des interactions entre des lignes de vie, présentés dans un ordre chronologique. [1]

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

-Message: Un message définit une communication particulière entre des lignes de vie. Plusieurs types de messages existent, les plus communs sont :

1. **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions, et message peut être suivi d'une réponse qui se représente par une flèche en pointillé. [3]
2. **Message asynchrone** : Une interruption ou un événement sont de bons exemples de signaux. Ils n'attendent pas de réponse et ne bloquent pas l'émetteur qui ne sait pas si le message arrivera à destination. Un signal est, par définition, un message asynchrone. [3]
3. **Messages de création et destruction d'instance** :
 - La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.
 - La destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet.

-Scénario: Représente une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions. [2]

-Ligne de vie : Une ligne de vie se dessine par un rectangle, auquel est accrochée une ligne verticale pointillée, Représente l'ensemble des opérations exécutées par un objet. [3]

1.2.2.5 Diagramme d'états-transitions

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets. Un diagramme d'états-transitions rassemble et organise les états et les transitions d'un classeur donné. [3]

-État : Un état représente une période dans la vie d'un objet pendant laquelle ce dernier attend un événement ou accomplit une activité, se représente graphiquement dans un diagramme d'états-transitions par un rectangles aux coins arrondis. [3]

- **État initial** : L'état initial est un pseudo état qui indique l'état de départ, par défaut, lorsque le diagramme d'états-transitions, ou l'état enveloppant, est invoqué. Lorsqu'un objet est créé, il entre dans l'état initial.
- **État final** : L'état final est un pseudo état qui indique que le diagramme d'états-Transitions, ou l'état enveloppant, est terminé.

-Événement : Un événement de changement est généré par la satisfaction (i.e. passage de faux à vrai) d'une expression booléenne sur des valeurs d'attributs.

-Transition : Une transition définit la réponse d'un objet à l'occurrence d'un événement. Elle lie, généralement, deux états E1 et E2 et indique qu'un objet dans un état E1 peut entrer dans l'état E2 et exécuter certaines activités, si un événement déclencheur se produit et que la condition de garde est vérifiée.

-Condition de garde : La condition de garde est évaluée uniquement lorsque l'événement déclencheur se produit. Si l'expression est fausse à ce moment-là, la transition ne se déclenche pas, si elle est vraie, la transition se déclenche et ses effets se produisent.[3]

1.2.2.6 Diagramme d'activités

Les diagrammes d'activités permettent de spécifier des traitements très proche des langages de programmation objet : spécifier des actions de base, structure de contrôle (condition, boucle), Ils sont donc bien adapter à la spécification détaillée des traitements en phase de réalisation. On peut aussi de utiliser de façon plus informelle pour décrire des enchainements d'actions de haut niveau, en particulier pour la description détaillé des cas d'utilisation. [1]

-Activité : Une activité définit un comportement décrit par un séquençement organisé d'unités dont les éléments simples sont les actions. Une activité est un comportement (behavior en anglais) et à ce titre peut être associée à des paramètres. De la gauche vers la droite, on trouve: le nœud représentant une action, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.

-Transition : Le passage d'une activité vers une autre est matérialisé par une transition. Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles. Les transitions spécifient l'enchaînement des traitements et définissent le flot de contrôle.

-La synchronisation Les flots de contrôle parallèles sont séparés ou réunis par des barres de synchronisation qui peuvent être des :

- **Débranchements:** les transitions qui partent d'un débranchement ont lieu en même temps.
- **Jonctions:** on ne franchit une jonction qu'après la réalisation de toutes les transitions qui s'y rattachent.

1.2.2.7 Diagramme de composant

Les diagrammes de composant décrivent les composants et leurs dépendances dans l'environnement de réalisation, ils montrent le choix de réalisation. En général, ils ne sont utilisés que pour des systèmes complexes. Un composant est une vue physique qui représente une partie implantable d'un système. Un composant peut être un script, fichier de demande, un fichier de donnée, table d'un composant peut être réalisé par d'autre composants, etc... Les éléments utilisés dans un diagramme de déploiement sont :

- Composant

Un composant représente une entité logicielle d'un système. Un composant doit fournir un service bien précis, les fonctionnalités qu'il encapsule doivent être cohérentes entre elle et générique.

-Dépendance

Une dépendance est utilisée pour modéliser la relation entre deux composants qui indique qu'un composant fait référence aux services offerts par autre composant.

1.2.2.8 Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds. Les diagrammes de déploiement existent sous deux formes : spécification et instance. [3]

-Composant : Un composant représente une entité logicielle du système. Sur un diagramme de déploiement, les composants sont placés dans des nœuds pour identifier l'endroit de leur déploiement.

-Nœud : Un nœud représente un ensemble d'élément matériel du système. Cette entité est représentée par un cube tridimensionnel.

1.3 Mise en œuvre d'UML [3]

UML n'est pas une méthode et ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet. Il n'est qu'un langage de modélisation, ce qui justifie le besoin de faire appel à une méthode qui permet le passage de l'expression des besoins au code de l'application (voir figure1.2). Nous allons donc présenter une des méthodes proposées dans la littérature. Il s'agit d'une méthode simple et générique proposée par LAURENTAUDIBERT [3]. Elle se situe à mi-chemin entre *UP (Unified Process)*, qui constitue un cadre général très complet de processus de développement, et *XP (eXtremeProgramming)* qui est une approche minimaliste à la mode centrée sur le code.

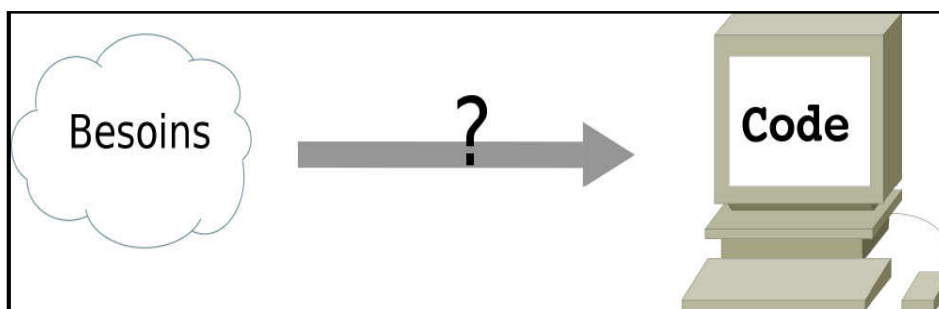


Figure 1.1:Quelle méthode pour passer de l'expression des besoins au code de l'application?

1.3.1 Identification des besoins et spécification des fonctionnalités

1.3.1.1 Identification et représentation des besoins : diagramme de cas d'utilisation

Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs (voir figure1.2). Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Durant cette étape, il faut déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation.

Les interactions entre les acteurs et le système (au sein des cas d'utilisation) seront explicitées sous forme textuelle et sous forme graphique au moyen de diagrammes de séquence.

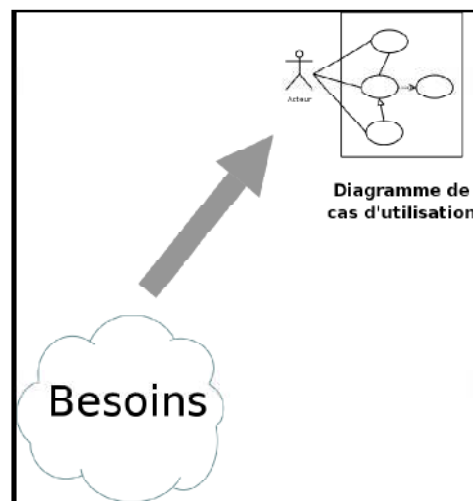


Figure 1.2 : Les besoins sont modélisés par un diagramme de cas d'utilisation.

1.3.1.2 Spécification détaillée des besoins : diagrammes de séquence système

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation, et la production de diagrammes de séquence système illustrant cette description textuelle (voir figure 1.3). Cette étape conduit souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins.

Les scénarios de la description textuelle des cas d'utilisation sont illustrés par des diagrammes de séquence système. Il faut, au minimum, représenter le scénario nominal de chacun des cas d'utilisation par un diagramme de séquence qui spécifier l'interaction entre l'acteur, ou les

acteurs, et le système. Le système est ici considéré comme un tout et est représenté par une ligne de vie. Chaque acteur est également associé à une ligne de vie.

1.3.1.3 Maquette de l'IHM de l'application (non couvert par UML)

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application (voir figure 1.4). La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs.

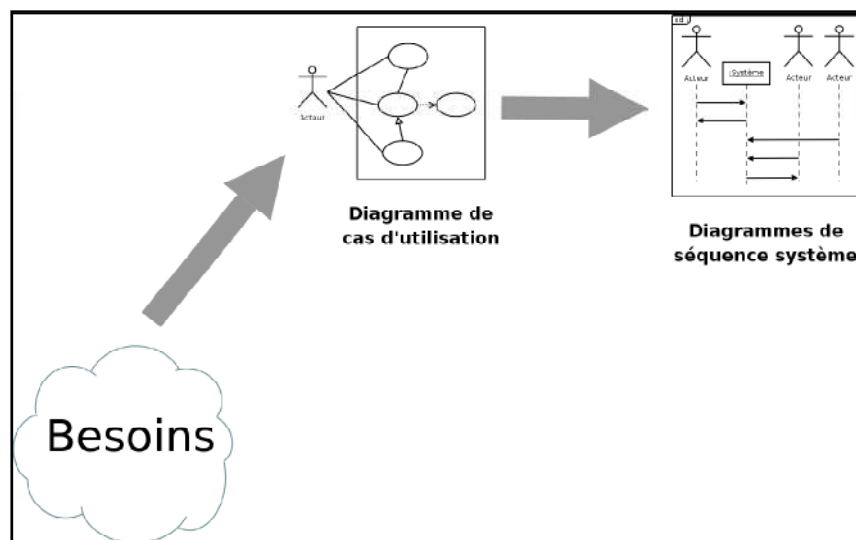


Figure 1.3: Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.

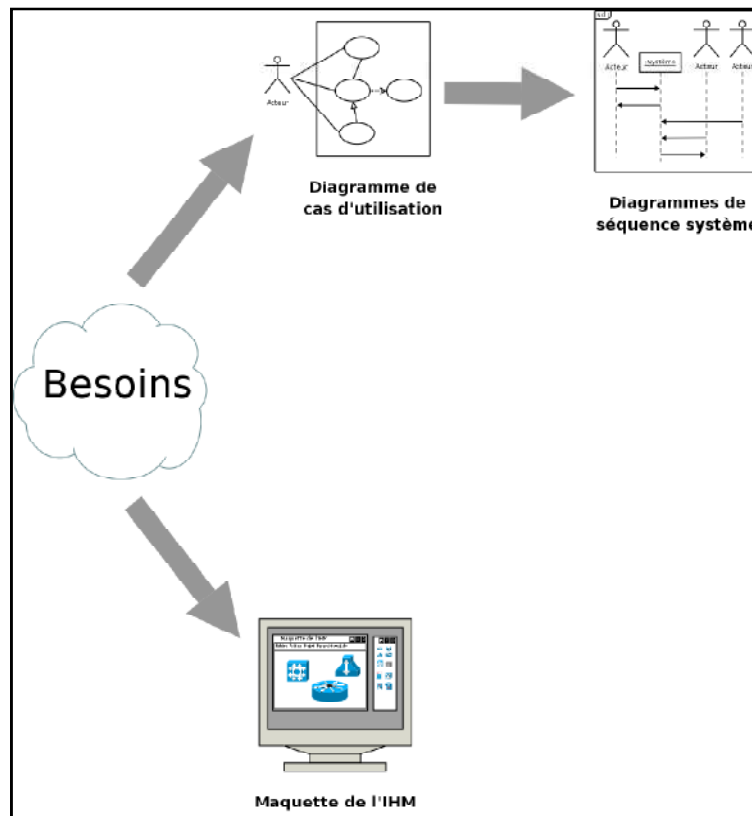


Figure 1.4 : Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.

1.3.2 Phases d'analyse

1.3.2.1 Analyse du domaine : modèle du domaine

L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine (voir figure 1.5). Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts peuvent être identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine.

Les classes du modèle du domaine ne doivent pas contenir d'opération, mais seulement les attributs.

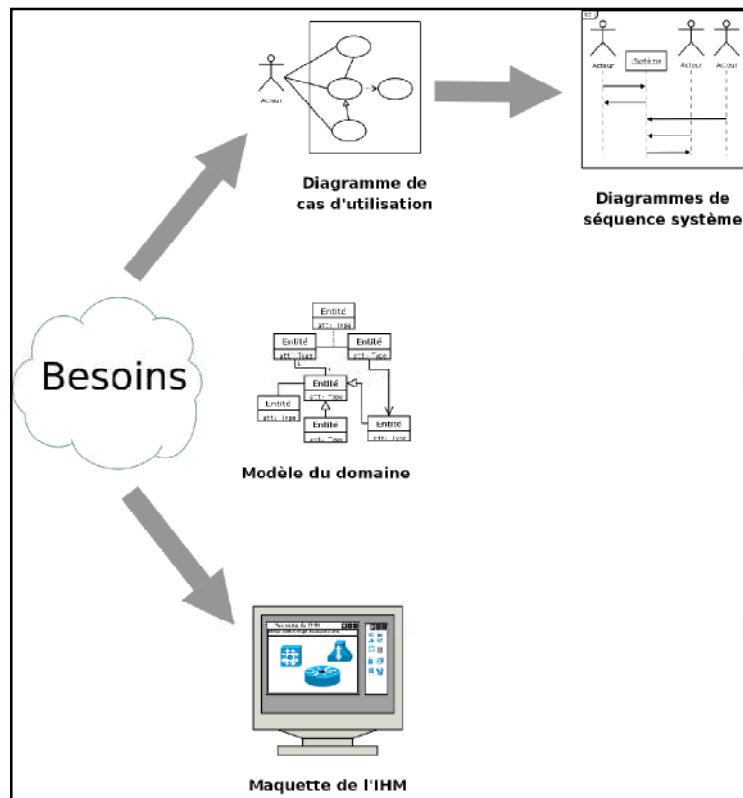


Figure 1.5 : La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.

1.3.2.2 Diagramme de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction et le diagramme de classes de conception (voir figure 1.6). Le diagramme de classes participantes modélise trois types de classes d'analyse sont :

- **Les classes de dialogues :** Les classes qui permettent les interactions entre l'IHM et les utilisateurs sont qualifiées de *dialogues*. Ces classes sont directement issues de l'analyse de la maquette. Il y a au moins un dialogue pour chaque association entre un acteur et un cas d'utilisation. En général, les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.
- **Les classes de contrôles :** Les classes qui modélisent la cinématique de l'application sont appelées *contrôles*. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues

par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.

- **Les classes entités :** Les classes métier, qui provient directement du modèle du domaine, sont qualifiées *d'entités*. Ces classes sont généralement persistantes, c'est-à-dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles.

1.3.2.3 Diagrammes d'activités de navigation

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant toute une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc. Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

UML offre la possibilité de représenter graphiquement les activités de navigation dans l'interface en produisant des diagrammes dynamiques. On appelle ces diagrammes des diagrammes de navigation (voir figure 1.7).

Les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : «fenêtre», «menu», «menu contextuel», «dialogue», etc.

La modélisation de la navigation à intérêt à être structurée par acteur.

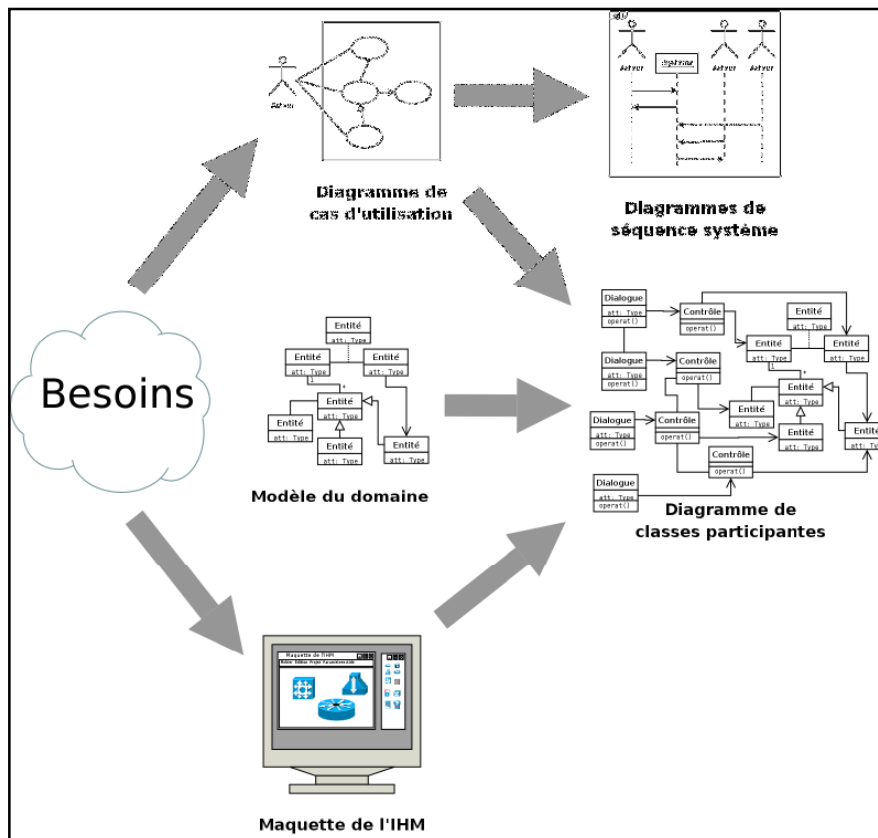


Figure 1.6 : Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.

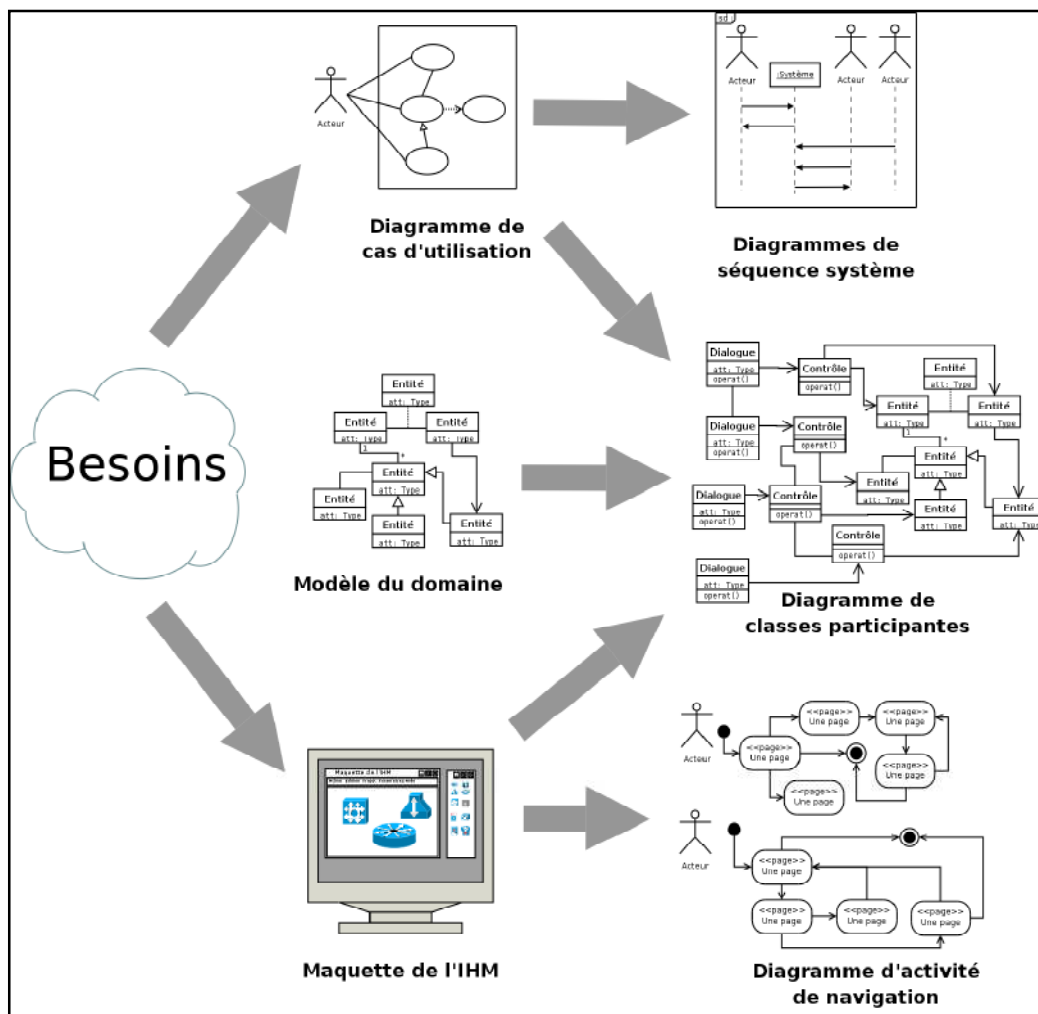


Figure 1.7 : Les diagrammes d’activités de navigation représentent graphiquement les activités de navigation dans l’interface.

1.3.3 Phase de conception

1.3.3.1 Diagrammes d’interaction

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par le diagramme de séquence système, aux classes d’analyse du diagramme de classes participantes. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d’interaction UML (voir figure 1.8). Parallèlement, une première ébauche de la vue statique de conception, c’est-à-dire du diagramme de classes de conception, est construite et complétée.

Durant cette phase, l’ébauche du diagramme de classes de conception reste indépendante des choix technologiques qui seront faits ultérieurement.

Les diagrammes d'interactions sont utiles au concepteur pour décider quelle est la classe qui va contenir chaque service ou fonction. Dans les diagrammes d'interaction, les objets communiquent en s'envoyant des messages qui invoquent des opérations sur les objets récepteurs. Il est ainsi possible de mettre en œuvre l'allocation des responsabilités à partir des diagrammes d'interaction.

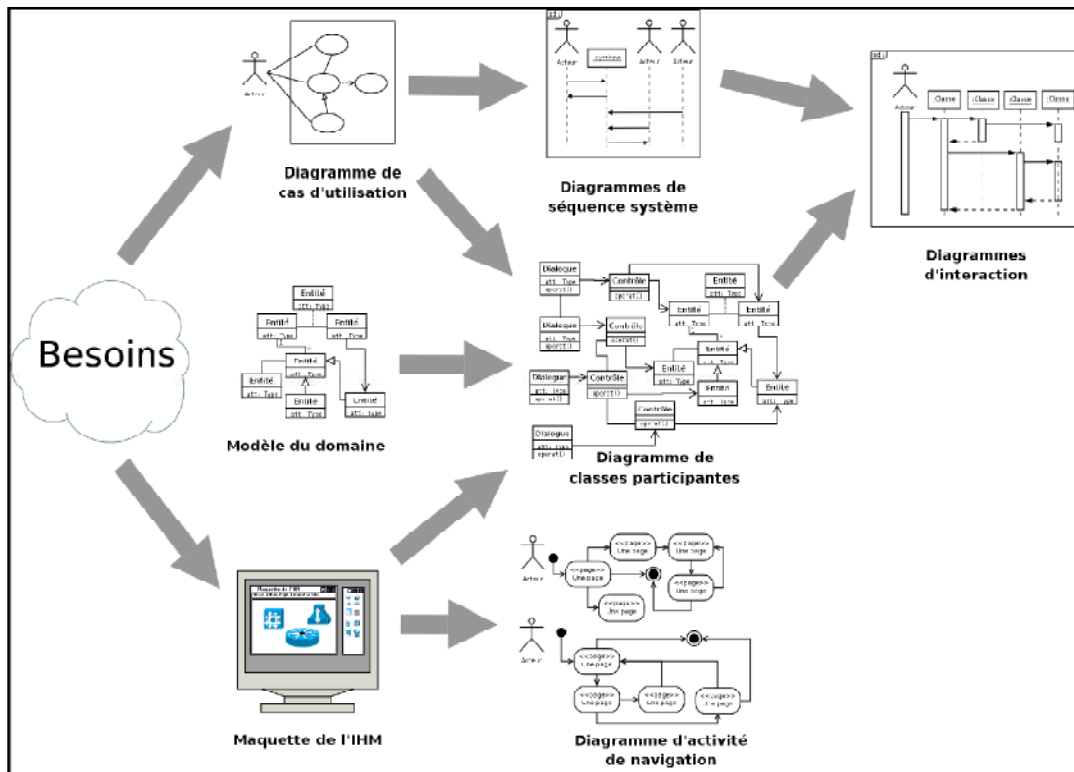


Figure 1.8 : Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

Par rapport aux diagrammes de séquences système, nous remplaçons ici le système, vu comme une boîte noire, par un ensemble d'objets en collaboration (voir figure 1.9). Ces objets sont des instances des trois types de classes d'analyse du diagramme de classes participantes, à savoir des dialogues, des contrôles et des entités.

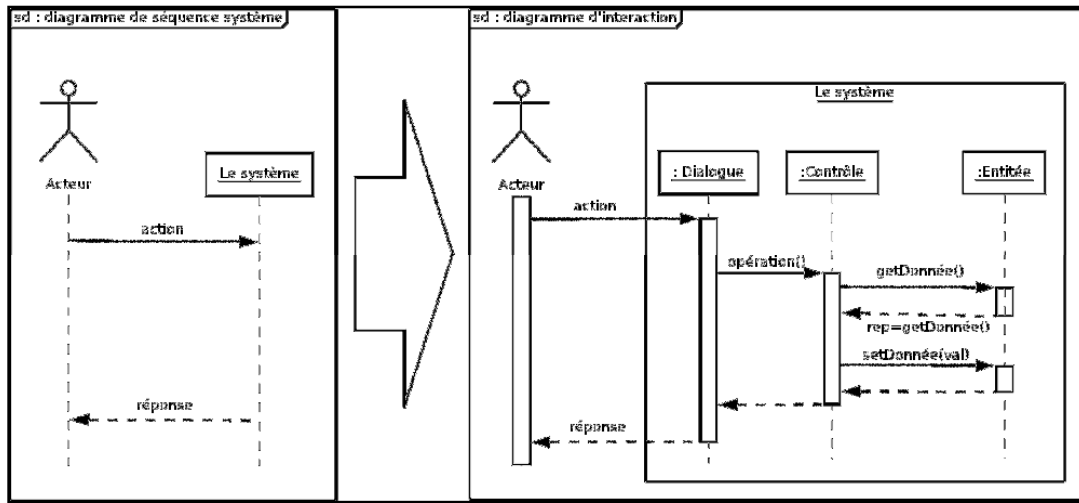


Figure 1.9 : Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d'objets en collaboration.

1.3.3.2 Diagramme de classes de conception

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation (voir figure 1.10). Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle du diagramme d'interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation.

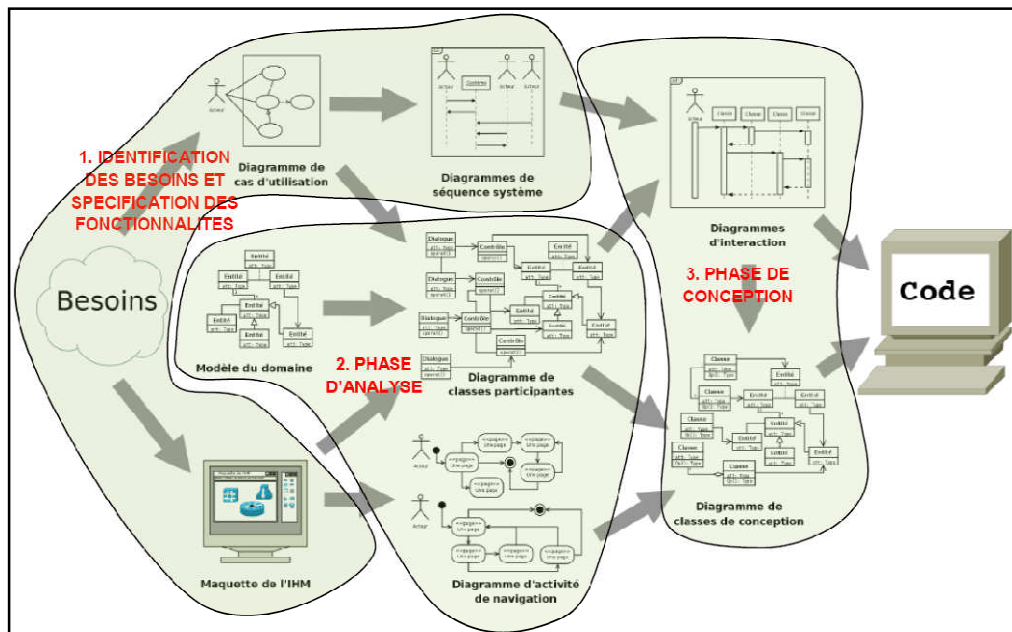


Figure 1.10 : Chaîne complète de la démarche de modélisation du besoin jusqu'au code.

1.4 Conclusion

A l'issu de ce chapitre, nous nous sommes adaptés avec l'outil de modélisation UML d'une part et la méthode proposée par Laurent AUDIBERT d'autre part, ce qui va nous guider pour la réalisation des prochaines étapes de notre projet.

CHAPITRE II

ETUDE DE L'EXISTANT

2.1 Introduction

L'étude de l'existant est une étape indispensable dans tout projet informatique, elle consiste en un travail de terrain au terme duquel on pourra bien connaître la situation actuelle de l'organisation. Pour ce faire, nous procédant tout d'abord par l'étude des différents postes de travaux puis l'analyse des documents véhiculant d'un acteur à un autre.

En ce qui nous concerne, notre étude de l'existant portera sur le magasin du Centre Universitaire de Mila.

2.2 Présentation de l'organisme d'accueil (CENTRE UNIVERSITAIRE DE MILA)

Le Centre Universitaire de Mila¹ est un établissement public à caractère scientifique, culturel et professionnel doté de la personnalité morale et de l'autonomie financière. Il a été créé par le décret exécutif N°204-08 du 09 juillet 2008. Donnant ainsi naissance à l'enseignement supérieur dans la wilaya.

Il est situé à 5km du centre-ville de Mila sur la route nationale N°79 allant vers les communes de Zeghaia et de Ferdjioua et s'étend sur une superficie d'environ 87 hectares.

Le Centre Universitaire de Mila propose durant l'année 2009-2010 une formation supérieure dans les domaines suivants :

- Mathématiques et Informatique.
- Science de la Nature et de la Vie.
- Sciences Economiques, Commerciales et Sciences de Gestion.
- Langues et Littérature Arabe.
- Anglais.

Le diagramme suivant présente l'organigramme du centre universitaire de Mila :

¹ <http://www.centre-univ-mila.dz>

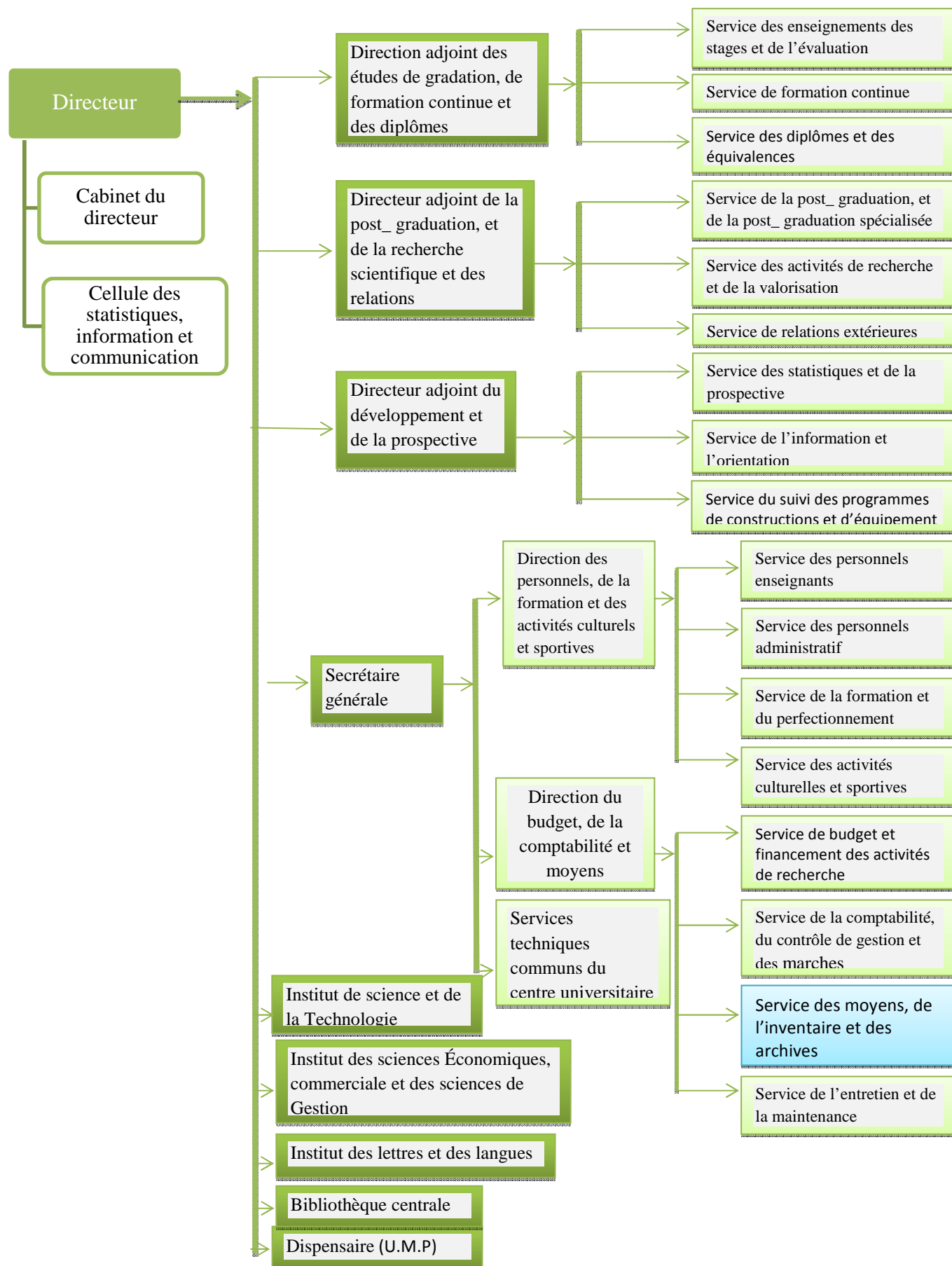


Figure 2.1 : Organigramme Du Centre Universitaire de Mila.

2.3 Cadre de l'étude

Notre stage est effectué exactement au niveau du magasin attaché au Service des Moyens, d'Inventaire et des Archives (SMIA) du Centre Universitaire de Mila (voir figure 2.2). Le magasinier est la seule personne qui gère le magasin. Il dispose d'un nombre importants d'articles consommables et non consommables gérés en fonction des besoins des différents services du centre.

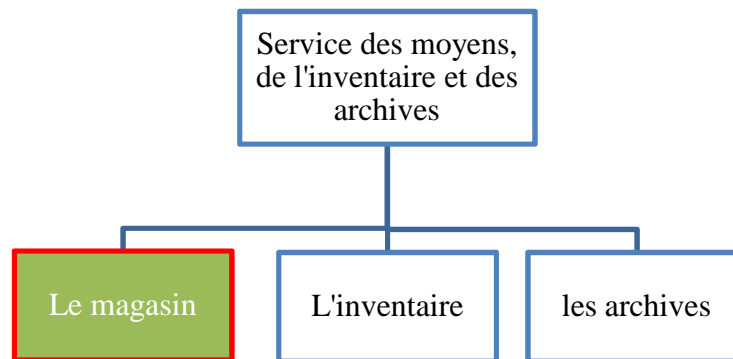


Figure 2.2 : hiérarchie du SMIA.

2.3.1 Description des tâches effectuées au niveau du magasin

Nous avons remarqué que les opérations les plus importantes qui se déroulent au niveau du magasin sont :

- L'arrivée des produits livrés par le fournisseur ;
- Le traitement des bons de commandes internes établis par les services du centre.

2.3.1.1 La réception des articles

- Dès la réception des articles, le magasinier décharge la livraison dans la salle de réception puis procède à la vérification de la conformité de la marchandise par rapport au bon de commande externe et au bon de livraison remis par le fournisseur.
- Par la suite, le bon de livraison est signé par le CSMIA.
- Enfin, le magasinier range les articles dans le magasin et inscrit ces derniers dans un registre appelé registre d'entrée.
- Pour le suivi de l'état du stock, il procède à la mise à jour de la fiche de stock (Quantité entrée, prix, date d'entrée...etc.) de chaque article.
- En cas de non-conformité, le magasinier contacte son supérieur pour qu'il prenne la décision adéquate à cette situation.

2.3.1.2 Le traitement des bons de commandes internes

- Les services appartenant au centre peuvent commander des produits dont ils ont besoin au magasinier en établissant un bon de commande interne.
- Lors de la réception de ces derniers, le magasinier les traite selon la disponibilité du stock. Pour chaque produit, il vérifie sa quantité en stock par rapport à la quantité commandée pour voir la possibilité de satisfaire la commande.
- Pour les articles consommables, le magasinier établit un bon de livraison de magasin (en 2 exemplaires : une copie pour lui et la deuxième pour le service demandeur). Pour les articles non consommables, il établit une affectation matérielle (en 3 exemplaires : une copie pour lui, la deuxième pour le CSMA et la dernière pour le service demandeur) ou une décharge (en 2 exemplaires : une copie pour lui et la deuxième pour le service demandeur).

Pour éviter une rupture en stock, le magasinier informe son responsable dans le cas où un ou plusieurs produits atteignent leur seuil minimum.

2.3.2 Études des postes de travail

Pour une compréhension du système existant, les postes de travail entrant dans notre champ d'étude doivent être étudiés et examinés minutieusement. Cela nous permettra d'identifier les besoins en informations des différents utilisateurs.

Dans cette étape, une fiche d'étude de poste sera élaborée pour chaque poste de travail et contiendra les informations suivantes :

- Désignation du poste ;
- Effectif du poste ;
- Mission du poste ;
- Les moyens utilisés ;
- La structure de rattachement ;
- Les tâches à accomplir par le poste, en précisant pour chaque tâche :
 - Sa fréquence.
 - Sa durée.
- Les documents ou les registres manipulés par le poste et pour chaque document en désignant:

- Son origine (cas d'une entrée) et sa destination (cas d'une sortie).
- Sa fréquence.

Dans notre cas, nous allons réaliser une seule fiche parce qu'il s'agit d'un seul poste qui est le magasin :

Fiche d'étude du poste de travail : Magasin

Fiche d'étude du poste de travail	
Désignation : Magasin	
Structure de rattachement : Service des moyens, de l'inventaire et des archives	
Responsable : Magasinier	
Responsable hiérarchique : Chef de Service des moyens, de l'inventaire et des archives	
Effectif du poste : 1	
Responsabilité : la gestion du magasin	
Moyens utilisés : ordinateur, téléphone, imprimante, photocopieur.	
Tâches à accomplir	
Tâches	Fréquence
- La réception des articles et la vérification (control et compte) de ces derniers avec le Bon de livraison.	Variable
-Rangement des articles dans le magasin.	Variable
-la mis à jour des fiches de stock.	Variable
-Satisfaction des commandes internes des services et établissement des bons de sorties.	Variable
-Consultation du stock.	variable
-Faire la statistique des articles.	1/ mois
-Etablissement d'une liste des produits qu'il faut commander.	Variable

Tableau 2.1: Fiche d'étude du poste de travail : Magasin

Documents manipulés :

Désignation	Nature			Origine	Destinataire	Fréquence
	Entrée	Sortie	Interne			
bon de commande				Chef service	Fournisseur	1
Bon de livraison	×			fournisseur	magasinier	2

Facture				fournisseur	Chef service	3
Bon commande interne	×			Service demandeur	magasinier	1
Bon livraison de magasin		×		magasinier	Service demandeur, magasin, responsable	3
décharge		×		magasinier	Service demandeur	2
Affectation du matériel		×		magasinier	Service demandeur	2
Fiche de stock			×	magasinier	magasinier	1
Registre d'entrée			×	magasinier	magasinier	1
Liste des produits qu'il faut commander		×		magasinier	Chef service	1

Tableau 2.2: Documents manipulés

2.3.3 Étude des documents

Après avoir étudié les différents postes de travail concernés, nous devons procéder à une étude détaillée de tous les documents et registres afin de définir de manière précise les informations qui circulent entre les différents acteurs d'en faire sortir toutes les données manipulées.

Les documents recensés durant notre étude sont les suivants :

- Bon de commande.
- Bon de livraison.
- Facture.
- Bon de commande interne.
- Bon de livraison de magasin.
- Décharge.
- Affectation matériel.
- Fiche de stock.
- Registre d'entrée.
- Demande d'approvisionnement.

Pour mettre en évidence le volume d'informations que contiennent les documents, leurs structurations ainsi que leur contenu en information, une fiche d'étude été établie pour chaque document et elle contiendra :

- Code du document.
- Désignation du document.
- Rôle du document.
- Nature du document (Interne, ou Externe).
- Source du document.
- Destination du document.
- Nombre d'exemplaires.
- Le contenu du document.
- Et d'éventuelles observations sur le document.

Légende des types :

A : Alphabétique.

N : Numérique.

AN : Alpha Numérique.

Date : Date.

- **Fiche d'étude du document** : Bon de commande

Code et rôle du document				
Code: BC				
Désignation: bon de commande				
Rôle: demander des articles au fournisseur.				
Caractéristiques du document				
Origine: chef service				
Destinataire: magasinier				
Nombre d'exemplaires: 1				
Nature: entrée				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N° BC	Numéro de bon de commande	AN	10	
adresse	Adresse du fournisseur	AN	50	
DATE	Date du bon de commande	Date	10	

NOM	Nom du fournisseur	A	50	
PRENOM	Prénom du fournisseur	A	50	
Ville	Ville du fournisseur	N	50	
articles	Liste des articles	AN	50	
QTE	Quantité entrée de chaque article	N	4	
P U	Prix unité de chaque article	N	8	
Somme	La somme	N	10	
M T	Montant totale	N	10	

Tableau 2.3: Fiche d'étude du document : Bon de commande

- **Fiche d'étude du document : Bon de livraison**

Code et rôle du document				
Code: BL				
Désignation: bon de livraison				
Rôle: Pour confirmer la livraison.				
Caractéristiques du document				
Origine: fournisseur				
Destinataire: magasinier				
Nombre d'exemplaires: 2				
Nature : entrée				
Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
Adresse	Adresse du fournisseur	AN	50	
NTél	Numéro Téléphone du fournisseur	AN	10	
FAX	Fax du fournisseur	AN	10	
RC	Registre commerce du fournisseur	AN	20	
Code article	Code de l'article	N	10	
NIS	Numéro d'identifiant statistique	N	15	
CB	Compte bancaire	AN	30	
DES	Désignation de l'article	AN	50	

QTS	Quantité entrée d'un article	N	4	
PRT	Prix totale	N	12	
MOT	Montant totale	N	12	
DAT	Date du bon de livraison	date	10	

Tableau 2.4: Fiche d'étude du document : Bon de livraison

- **Fiche d'étude du document : Facture**

Code et rôle du document				
Code: Facture				
Désignation: facture				
Rôle: pour déterminer la quantité et le montant total de la facture.				
Caractéristiques du document				
Origine: fournisseur				
Destinataire: chef service				
Nombre d'exemplaires: 3				
Nature: entrée				
Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N°facture	Numéro de la facture	AN	8	
Nom	Nom de client	AN	30	
Addressee	Adresse de client	AN	30	
TELC	Numéro téléphone de client	AN	15	
NIFC	Numéro identifiant fiscale de client	N	14	
NISC	Numéro identifiant fiscale du client	N	14	
NOM	Nom du fournisseur	AN	20	
PRENOM	Prénom du fournisseur			
RC	Registre de commerce	N	10	
NIF	Numéro identifiant fiscale de fournisseur	AN	18	
NIS	Numéro identifiant fiscale de fournisseur	AN	17	
NB	Numéro bancaire	AN	22	
AB	Agence bancaire	AN	20	
TEL	Numéro téléphone de fournisseur	AN	15	

FAX	Fax du fournisseur	AN	15	
DES	Désignation de l'article	AN	50	
QTS	Quantité entrée de chaque article	N	4	
PRT	Prix unité	N	7	
MOT	Montant totale	N	10	
DATE	Date de la facture	DATE	10	

Tableau 2.5: Fiche d'étude du document : Facture

- **Fiche d'étude du document:** liste des articles à commander.

Code et rôle du document				
Code: DA				
Désignation: demande d'approvisionnement				
Rôle: préparer la liste des articles en stock alerte.				
Caractéristiques du document				
Origine: magasinier				
Destinataire: chef service				
Nombre d'exemplaires: 1				
Nature: sortie				
Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N° BDA	Numéro de demande d'approvisionnement	AN	10	
Articles	La liste des articles	AN	50	
DATE	Date du bon	DATE	10	

Tableau 2.6: Fiche d'étude du document : liste des articles à commander.

- **Fiche d'étude du document :** Bon commande interne

Code et rôle du document				
Code: BCI				
Désignation: bon de commande interne				
Rôle: demander les articles au magasinier				
Caractéristiques du document				
Origine: service bénéficié				
Destinataire: magasinier				

Nombre d'exemplaires: 1				
Nature: entrée				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N°BCI	Numéro de Bon commande interne	AN	9	
Nom	Nom d'article	AN	70	
QTS	Quantité commandé	N	4	
OBS	observation	AN	20	
DAT	Date du bon de commande interne	date	10	

Tableau 2.7: Fiche d'étude du document : Bon commande interne.

- **Fiche d'étude du document :** Bon de livraison de magasin

Code et rôle du document				
Code: BLM				
Désignation: bon livraison de magasin				
Rôle: confirmer que le magasinier répond au besoin du service demandeur.				
Caractéristiques du document				
Origine: magasinier				
Destinataire: service bénéficiaire				
Nombre d'exemplaires: 2				
Nature: sortie				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N° BLM	Numéro de bon livraison de magasin	AN	9	
DAT	Date du bon	Date	10	
DES	Désignation	AN	70	
QTS	Quantité sortie d'un article	N	4	
OBS	Observation	AN	20	
Nom	Nom du bénéficiaire	A	20	
Prénom	Prénom du bénéficiaire			

Tableau 2.8: Fiche d'étude du document : Bon de livraison de magasin

- **Fiche d'étude du document : Décharge**

Code et rôle du document				
Code: D				
Désignation: décharge				
Rôle: pour confirmer que le magasinier a répondu favorablement à la demande.				
Caractéristiques du document				
Origine: magasinier				
Destinataire: service demandeur				
Nombre d'exemplaires: 1				
Nature: sortie				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N° DE	Numéro décharge	AN	10	
DAT	Date de la décharge	DATE	10	
DES	Désignation de l'article	AN	70	
MARQUE	Marque de l'article	AN	10	
QTS	Quantité sortie	N	3	
RE	Référence de l'article	AN	30	
N IN	Numéro inventaire	AN	8	
Nom	Nom du service Bénéficiaire	A	20	
OBS	observation	AN	20	

Tableau 2.9: Fiche d'étude du document : Décharge.

- **Fiche d'étude du document : Affectation Matériel**

Code et rôle du document
Code: AM
Désignation: Affectation Matériel
Rôle: pour confirmer que le magasinier a répondu favorablement à la demande.
Caractéristiques du document
Origine: magasinier
Destinataire: service bénéficiaire
Nombre d'exemplaires: 2
Nature: sortie

Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
N AFF	Numéro d'affectation matériel	AN	8	
DAT	Date de l'affectation	DATE	10	
DES	Désignation	AN	70	
MA	Marque de l'article	AN	10	
QTS	Quantité de l'article	N	3	
RE	Référence de l'article	AN	30	
NUM_ Inventaire	Numéro inventaire	AN	8	
Nom	Nom du service	A	20	
OBS	observation	AN	20	

Tableau 2.10: Fiche d'étude du document : Affectation Matériel.

- **Fiche d'étude du document** : fiche de stock

Code et rôle du document				
Code: FS				
Désignation: fiche de stock				
Rôle: enregistrer toutes les informations de chaque article réceptionné.				
Caractéristiques du document				
Origine: magasinier				
Destinataire: magasinier				
Nombre d'exemplaires: 1				
Nature: Position				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
DES	Désignation de l'article	AN	70	
DATE	Date d'entrée de l'article	DATE	10	
N° BN	Numéro du bon de livraison	N	3	
QTEE	Quantité entrée	N	4	
QTES	Quantité Sortie	N	4	
QTES	Quantité en stock	N	4	

Tableau 2.11: Fiche d'étude du document : fiche de stock.

- **Fiche d'étude du document** : registre d'entrée.

Code et rôle du document				
Code: RG				
Désignation: registre d'entrée				
Rôle: inscription de tous les articles entrant				
Caractéristiques du document				
Origine: magasinier				
Destinataire: magasinier				
Nombre d'exemplaires: 1				
Nature: Position				
Eléments d'information				
Rubrique	Désignation	Type	Taille	Observation
DES	Désignation de l'article	AN	70	
DATE	Date d'entrée de l'article	DATE	10	
N° BL	Numéro du bon de livraison	AN	8	
Nom	Nom du Fournisseur	A	20	
QTEE	Quantité entrée	N	3	

Tableau 2.12: Fiche d'étude du document : registre d'entrée.

2.4 Conclusion

Durant l'analyse de l'existant nous avons pu recenser toutes les informations nécessaires et indispensables à notre projet à savoir la conception et la réalisation d'un système de gestion de stock. Elle nous a permis la compréhension des besoins de l'organisation et ce à travers l'analyse détaillée des postes de travail et les documents manipulés. Par conséquent, elle nous a aidé à bien fixer les objectifs afin qu'elle puisse être la base de la conception de ce système.

Dans le prochain chapitre nous aborderons l'étude conceptuelle de notre système.

CHAPITRE III

ETUDE DE CAS

3.1 Introduction

L'objectif de cette étape est de déterminer de façon détaillée et précise ce que le système devra faire, afin de répondre aux objectifs établis lors de l'étude de l'existant, tout en respectant les contraintes établies préalablement.

3.2 Identification des cas d'utilisations

Le tableau suivant englobe les différents CU de ce système :

N°	Cas d'utilisation	Acteur
1	Authentification	Magasinier
2	Réception des articles	Magasinier
3	Consommation des articles	Magasinier
4	Fiche de stock d'un article	Magasinier
5	Articles en stock alerte	Magasinier
6	Ajouter article	Magasinier
7	Mise à jour des articles	Magasinier
8	Ajouter famille	Magasinier
9	Mise à jour des familles	Magasinier
10	Ajouter fournisseur	Magasinier
11	Mise à jour des fournisseurs	Magasinier
12	Ajouter service	Magasinier
13	Mise à jour des services	Magasinier

Tableau 3.1 : Les cas d'utilisations.

3.3 Description des cas d'utilisation

Nous allons maintenant donner une description de chaque cas d'utilisation.

3.3.1 Cas d'utilisation «authentification»

Titre : Authentification

Finalité : Ce cas permet d'utiliser le système par le magasinier.

Acteur : Le magasinier

Pré condition : Le magasinier possède un compte.

Enchaînements nominaux:

1. le magasinier lance l'application
2. Le système affiche une fenêtre pour qu'il s'identifie.
3. Le magasinier saisie son login et son mot de passe.
4. Le système vérifie leur validité puis lance le menu principal de l'application.

Enchaînements alternatifs :

- Login ou mot de passe erroné, l'authentification est demandée à nouveau au maximum trois fois.

Enchaînements exceptionnels:

- Le magasinier ne saisit pas le bon mot de passe. l'application se ferme.

Post conditions :

- Le menu de l'application est accessible.

3.3.2 Cas d'utilisation « Réception des articles»

Titre : Réception des articles

Finalité : Ce cas permet de traiter les articles livrés

Acteur principal : Le magasinier

Pré condition : -Le magasinier s'est authentifié.

-La commande est livrée.

Enchaînements nominaux:

1. le magasinier demande d'introduire les informations de la nouvelle livraison.
2. Le système lui affiche le formulaire.
3. Le magasinier remplit ce formulaire : s'il s'agit d'un nouvel article et/ou d'un nouveau fournisseur, le magasinier doit ajouter ces derniers.
4. Le système contrôle les informations saisies.
5. Le magasinier valide la saisie du formulaire.
6. Le système enregistre les données de la livraison.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronées.

Post conditions :

- La nouvelle livraison a été enregistrée.

3.3.3 Cas d'utilisation « consommation des articles»

Titre : Consommation des articles

Finalité : Ce cas permet satisfaire les demandes internes des services demandeurs.

Acteur principal : Le magasinier

Pré condition : -Le magasinier s'est authentifié.

-La commande est reçue depuis le service.

Enchaînements nominaux:

1. le magasinier demande d'ajouter un nouveau bon de sortie selon le type des articles commandés. Pour les articles consommables, le magasinier choisit un bon de livraison de magasin. Pour les articles non consommables, il choisit une affectation ou une décharge.
2. Le système lui affiche le formulaire selon le type du bon de sortie.
3. Le magasinier remplit ce formulaire par les données nécessaires (désignation article, quantité sortie, date, ...etc.).

4. Le système contrôle les informations saisies.
5. Le magasinier valide la saisie du formulaire.
6. Le système enregistre les informations.
7. Le magasinier lance l'impression des bons de sorties.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronées.

Post conditions :

- La commande est livrée.

3.3.4 Cas d'utilisation «Fiche de stock»

Titre : Fiche de stock

Finalité : Ce cas permet de consulter l'état de chaque article

Acteur : Le magasinier

Pré condition : Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande de visualiser la liste de tous les articles.
2. Le système affiche pour chaque article sélectionné sa quantité entrée globale, sa quantité sortie globale et sa quantité en stock

3.3.5 Cas d'utilisation «articles en stock alerte»

Titre : articles en stock alerte

Finalité ce cas permet d'afficher la liste des articles qui sont en stock alerte.

Acteur principal : Le magasinier

Pré condition : - Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande au système d'afficher la liste des articles qui ont atteint le seuil minimum en stock.
2. le système affiche tous les articles dont la quantité en stock est inférieur ou égale au seuil minimal.
3. Le magasinier lance l'impression de cette liste.

Post conditions : -La liste est imprimée.

3.3.6 Cas d'utilisation Ajouter article

Titre : ajouter un article.

Finalité : Ce cas permet d'ajouter un article

Acteur : Le magasinier

Pré condition : Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande l'ajout d'un nouvel article.
2. Le système affiche le formulaire pour saisir les informations nécessaires.
3. Le magasinier remplit ce formulaire (Désignation, marque ...etc.)
4. Le système contrôle les informations.
5. Le magasinier valide la saisie.

6. Le système enregistre l'ajout.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronées.

Post conditions :

- L'article est ajouté.

3.3.7 Cas d'utilisation «Mise à jour des articles»

3.3.7.1 Modifier article

Titre : Modifier Article

Finalité : Ce cas permet de modifier les informations d'un article existant

Acteur : Le magasinier

Pré condition : -Le magasinier s'authentifie.

-L'article concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des articles.
2. Le système affiche la liste de tous les articles existants.
3. Le magasinier sélectionne un article pour modification
4. Le système affiche les informations de l'article sélectionné.
5. le magasinier saisit les modifications.
6. Le système contrôle la saisie des informations modifiées.
7. Le magasinier valide la modification.
8. Le système enregistre la modification.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronées.

Post conditions :

- L'article est modifié.

3.3.7.2 Supprimer article

Titre : Supprimer Article

Finalité : Ce cas permet de supprimer un article existant

Acteur : Le magasinier

Pré condition : Le magasinier s'authentifie.

L'article concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des articles.
2. Le système affiche la liste de tous les articles.
3. Le magasinier sélectionne l'article qu'il veut supprimer.
4. Le système informe le magasinier s'il veut vraiment le supprimer.
5. Le magasinier valide l'opération.

Post conditions :

- L'article est supprimé.

3.3.8 Cas d'utilisation Ajouter famille

Titre : Ajouter une famille.

Finalité : Ce cas permet d'ajouter une famille

Acteur : Le magasinier

Pré condition : Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande l'ajout d'une nouvelle famille.
2. Le système affiche le formulaire pour saisir les informations nécessaires.
3. Le magasinier remplit ce formulaire (Désignation ...etc.)
4. Le système contrôle les informations saisies.
5. Le magasinier valide la saisie du formulaire.
6. Le système enregistre les informations de la famille.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronés.

Post conditions :

- La nouvelle famille est ajoutée.

3.3.9 Cas d'utilisation « MAJ des familles »

3.3.9.1 Modifier famille

Titre : Modifier une famille

Finalité : Ce cas permet de modifier les informations d'une famille existante

Acteur : Le magasinier

Pré condition : - Le magasinier s'est authentifié.

- La famille concernée existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des familles.
2. Le système affiche la liste de tous les familles existantes.
3. Le magasinier sélectionne une famille pour modification
4. Le système affiche les informations de la famille sélectionnée.
5. le magasinier saisie les modifications.
6. Le système contrôle la saisie des informations modifiées.
7. Le magasinier valide la modification.
8. Le système enregistre la modification.

Enchaînements alternatifs :

1. Login ou mot de passe erroné ; l'authentification est demandée à nouveau.
2. Les informations sont incomplètes ou erronés.

Post conditions :

- La famille est modifiée.

3.3.9.2 Supprimer famille

Titre : Supprimer une famille

Finalité : Ce cas permet de supprimer une famille existante

Acteur : Le magasinier

Pré condition : Le magasinier s'est authentifié.
La famille concernée existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des familles existantes.
2. Le système affiche la liste de tous les familles.
3. Le magasinier sélectionne la famille qu'il veut supprimer.
4. Le système informe le magasinier s'il veut vraiment la supprimer.
5. Le magasinier valide l'opération.

Post conditions :

- La famille est supprimée.

3.3.10 Cas d'utilisation Ajouter fournisseur

Titre : ajouter un fournisseur.

Finalité : Ce cas permet d'ajouter un fournisseur.

Acteur principal: Le magasinier

Pré condition : Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande l'ajout d'un nouveau fournisseur.
2. Le système affiche le formulaire pour saisir les informations nécessaires.
3. Le magasinier remplit ce formulaire (Nom, TEL, ...etc.)
4. Le système contrôle les informations saisies.
5. Le magasinier valide la saisie.
6. Le système enregistre l'ajout.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronés.

Post conditions :

- Le nouveau fournisseur est ajouté.

3.3.11 Cas d'utilisation MAJ des fournisseurs

3.3.11.1 Modifier fournisseur

Titre : Modifier un fournisseur

Finalité : Ce cas permet de modifier les informations d'un fournisseur existant

Acteur principal: Le magasinier

Pré condition : - Le magasinier s'est authentifié.

-Le fournisseur concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des fournisseurs.

2. Le système affiche la liste de tous les fournisseurs existants.
3. Le magasinier sélectionne un fournisseur pour modification
4. Le système affiche les informations du fournisseur sélectionné.
5. le magasinier saisie les modifications.
6. Le système contrôle les informations modifiées.
7. Le magasinier valide la modification.
8. Le système enregistre la modification.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronés.

Post conditions :

- Le fournisseur est modifié.

3.3.11.2 Supprimer fournisseur

Titre : Supprimer un fournisseur

Finalité : Ce cas permet de supprimer un fournisseur existant

Acteur : Le magasinier

Pré condition : - Le magasinier s'est authentifié.

-Le fournisseur concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des fournisseurs existants.
2. Le système affiche la liste de tous les fournisseurs.
3. Le magasinier sélectionne le fournisseur qu'il veut supprimer.
4. Le système informe le magasinier s'il veut vraiment la supprimer.
5. Le magasinier valide l'opération.

Post conditions :

-Le fournisseur est supprimé.

3.3.12 Cas d'utilisation Ajouter service

Titre : ajouter service

Finalité : Ce cas permet d'ajouter un nouveau service.

Acteur : Le magasinier

Pré condition : Le magasinier s'est authentifié.

Enchaînements nominaux:

1. Le magasinier demande l'ajout d'un nouveau service.
2. Le système affiche le formulaire pour saisir les informations nécessaires.
3. Le magasinier remplit ce formulaire (Nom, TEL, ...etc.)
4. Le système contrôle les informations saisies.
5. Le magasinier valide les informations.
6. Le système enregistre les informations.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronés.

Post conditions :

- Le nouveau service est ajouté.

3.3.13 Cas d'utilisation MAJ des services

3.3.13.1 Modifier service

Titre : Modifier un service

Finalité : Ce cas permet de modifier les informations d'un service existant

Acteur : Le magasinier

Pré condition : - Le magasinier s'est authentifié.

-Le service concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des services.
2. Le système affiche la liste de tous les services existants.
3. Le magasinier sélectionne un service pour modification
4. Le système affiche les informations du service sélectionné.
5. le magasinier saisit les modifications.
6. Le système contrôle les modifications modifiées.
7. Le magasinier valide la modification.
8. Le système enregistre la modification.

Enchaînements alternatifs :

- Les informations sont incomplètes ou erronés.

Post conditions :

- Le service est modifié.

3.3.13.2 Supprimer service

Titre : Supprimer un service

Finalité : Ce cas permet de supprimer un service existant

Acteur : Le magasinier

Pré condition : - Le magasinier s'est authentifié.

-Le service concerné existe.

Enchaînements nominaux:

1. Le magasinier demande la visualisation de la liste des services existants.
2. Le système affiche la liste de tous les services.
3. Le magasinier sélectionne le service qu'il veut supprimer.
4. Le système informe le magasinier s'il veut vraiment la supprimer.
5. Le magasinier valide l'opération.

Post conditions :

-Le service est supprimé.

3.4 Diagramme de cas d'utilisation

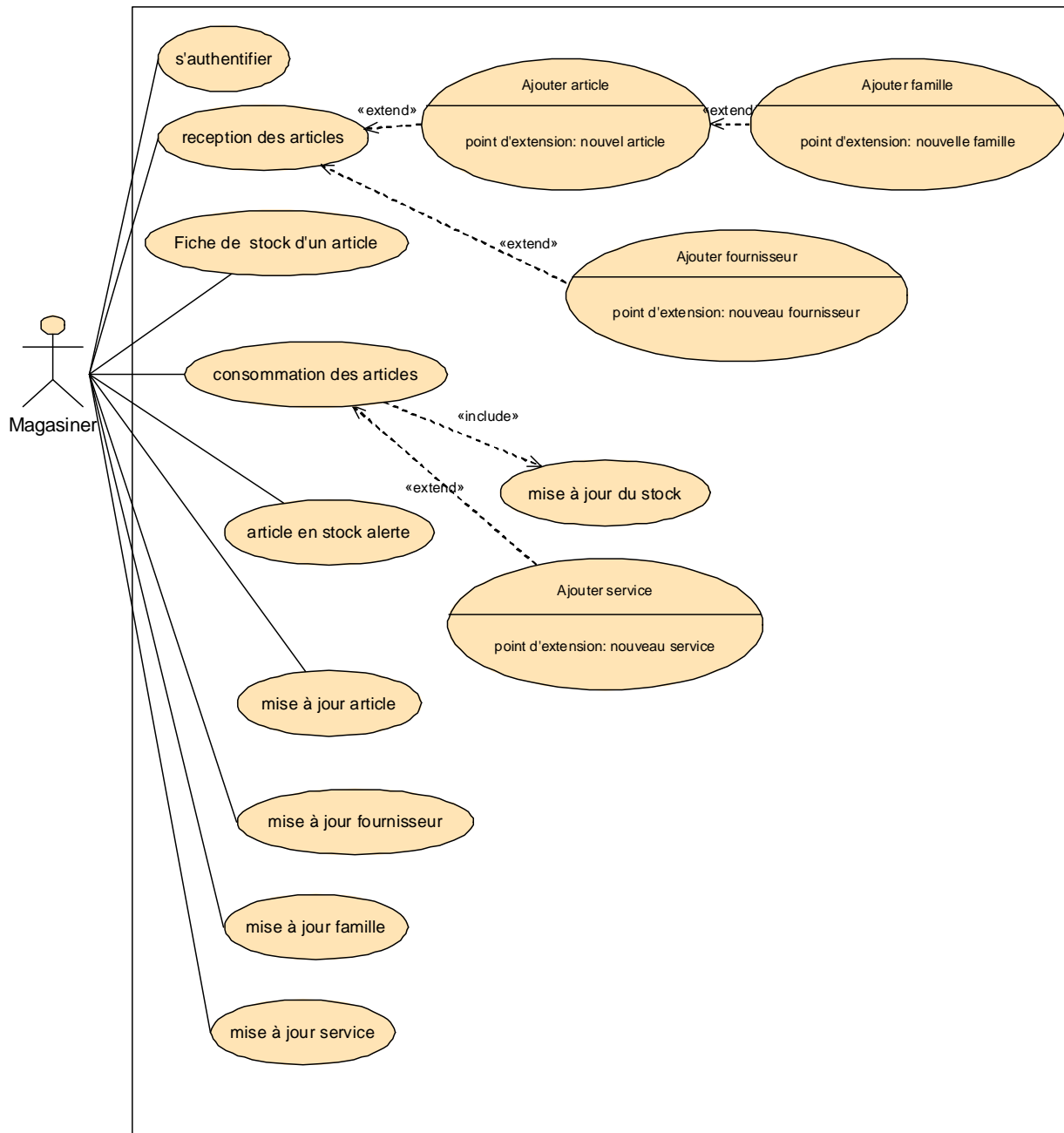


Figure 3.1 : Diagramme de cas d'utilisation

3.5 Les diagrammes de séquence

3.5.1 Authentification

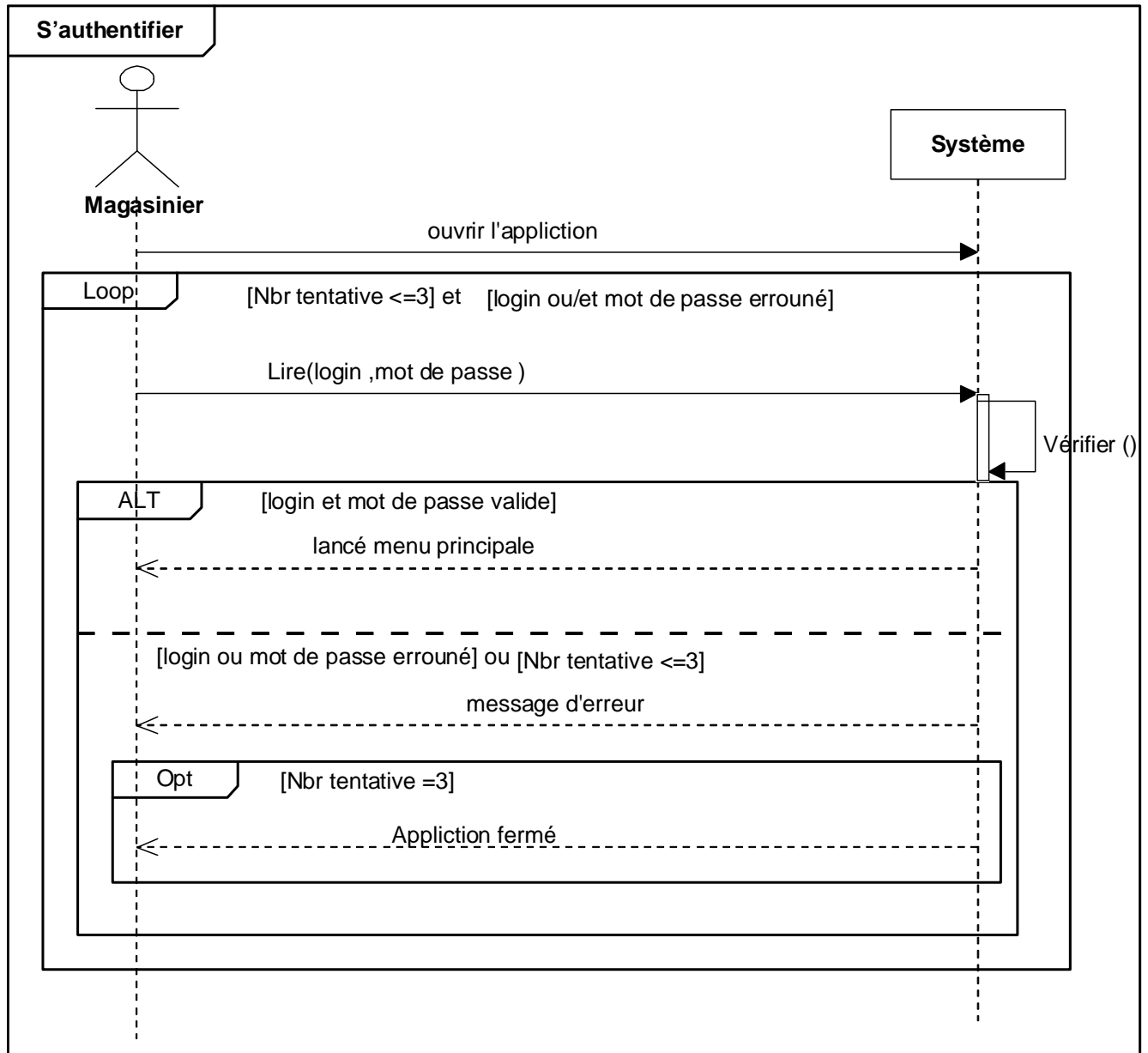


Figure3.2 : Diagramme de séquence du cas d'utilisation Authentification

3.5.2 Réception des articles

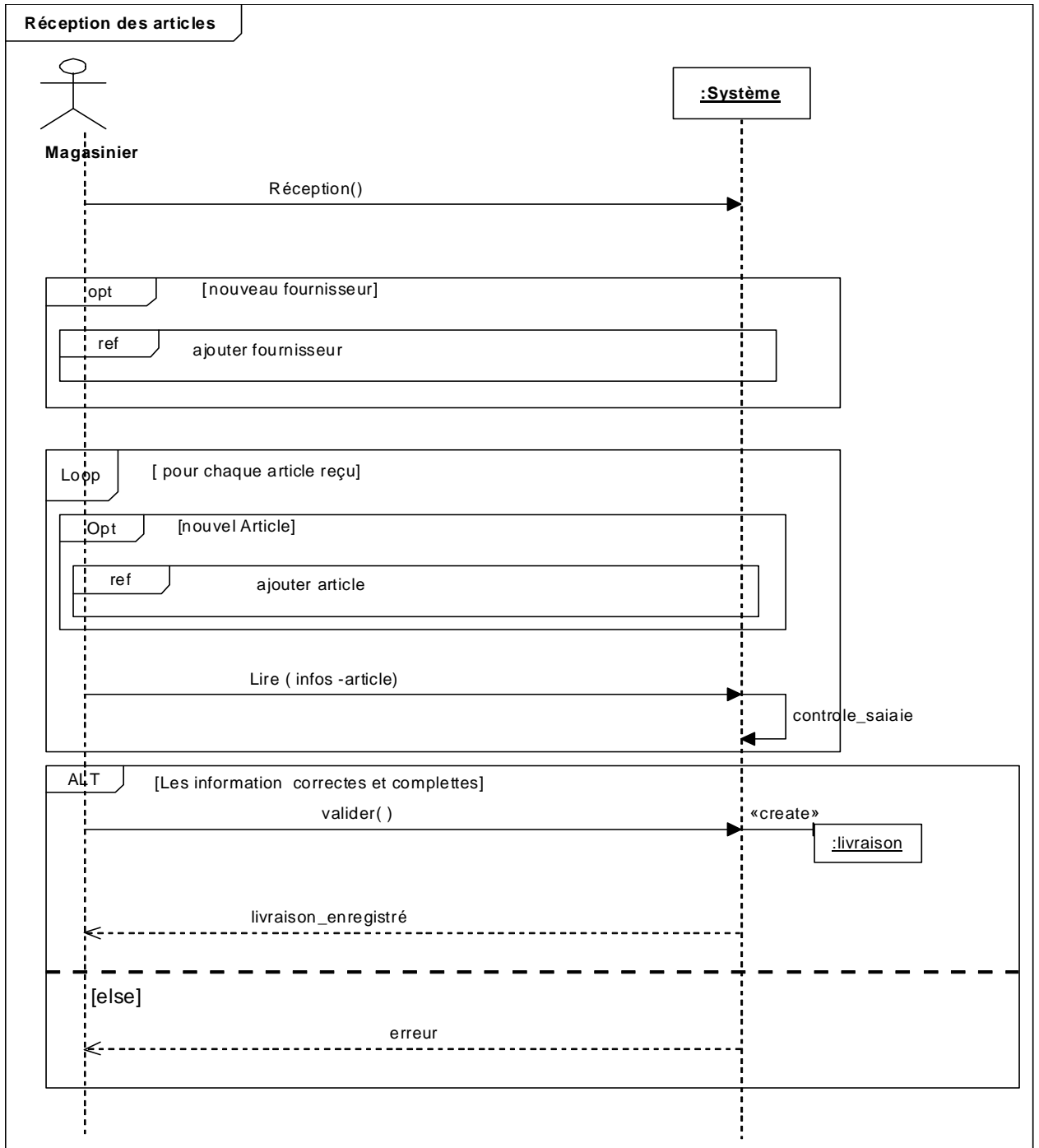


Figure 3.3 : Diagramme de séquence du cas d'utilisation Réception des articles.

3.5.3 Consommation des articles

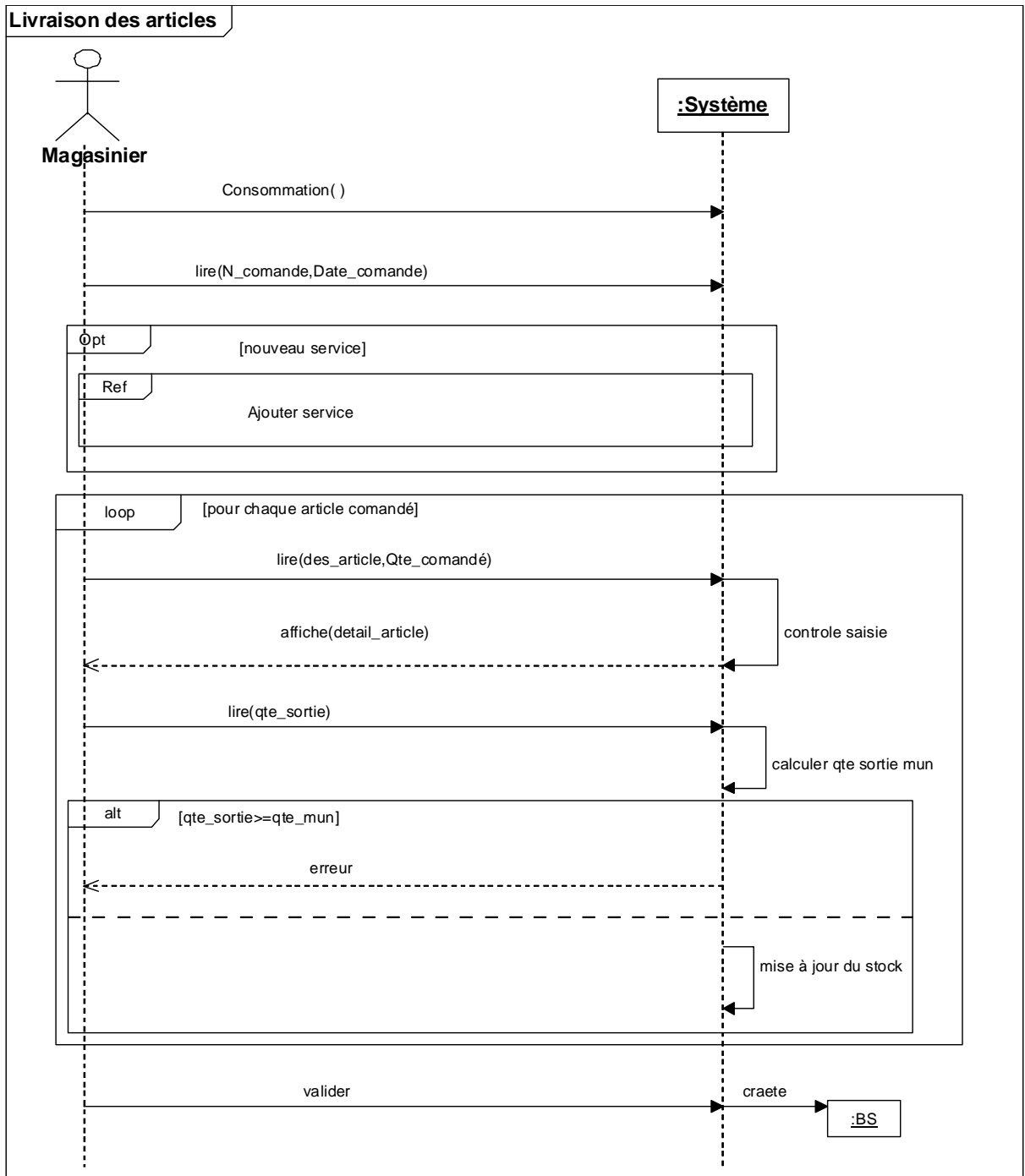


Figure 3.4 : Diagramme de séquence du cas d'utilisation Consommation des articles.

3.5.4 Fiche de stock d'un article

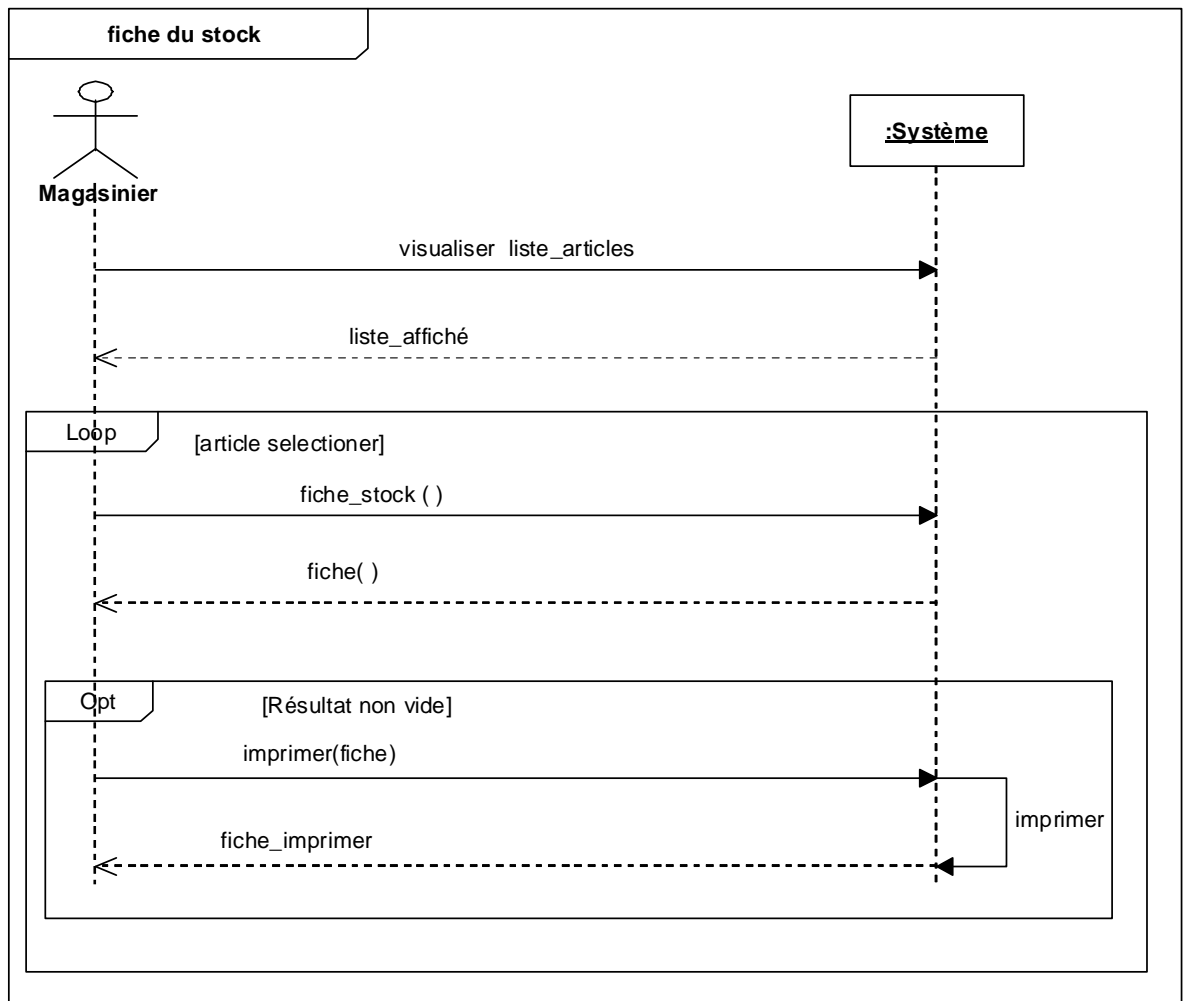


Figure3.5 : Diagramme de séquence du cas d'utilisation fiche de stock d'un article.

3.5.5 Édition des articles en stock alerte

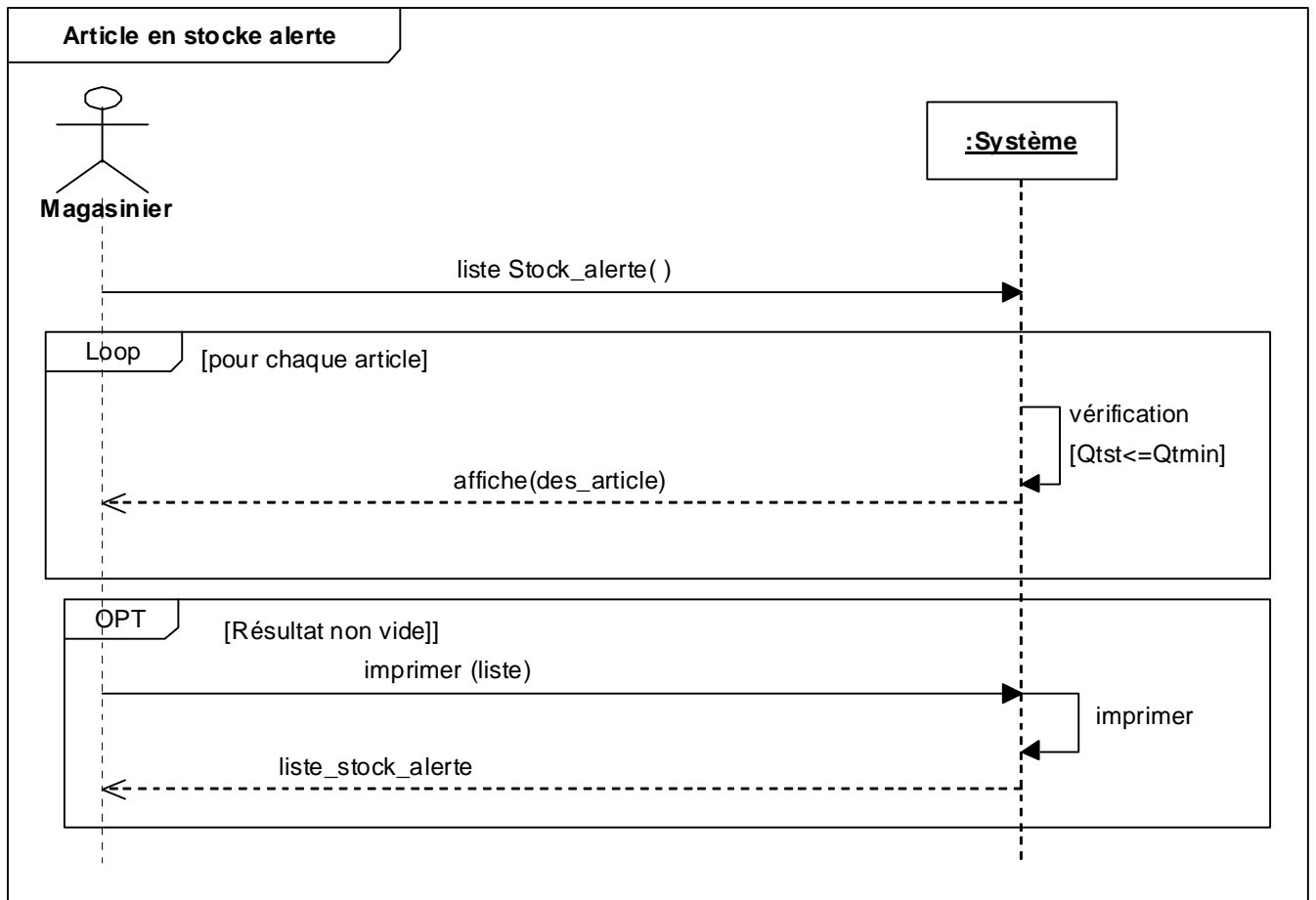


Figure3.6 : Diagramme de séquence du cas d'utilisation édition des articles en stock alerte

3.5.6 Ajouter article

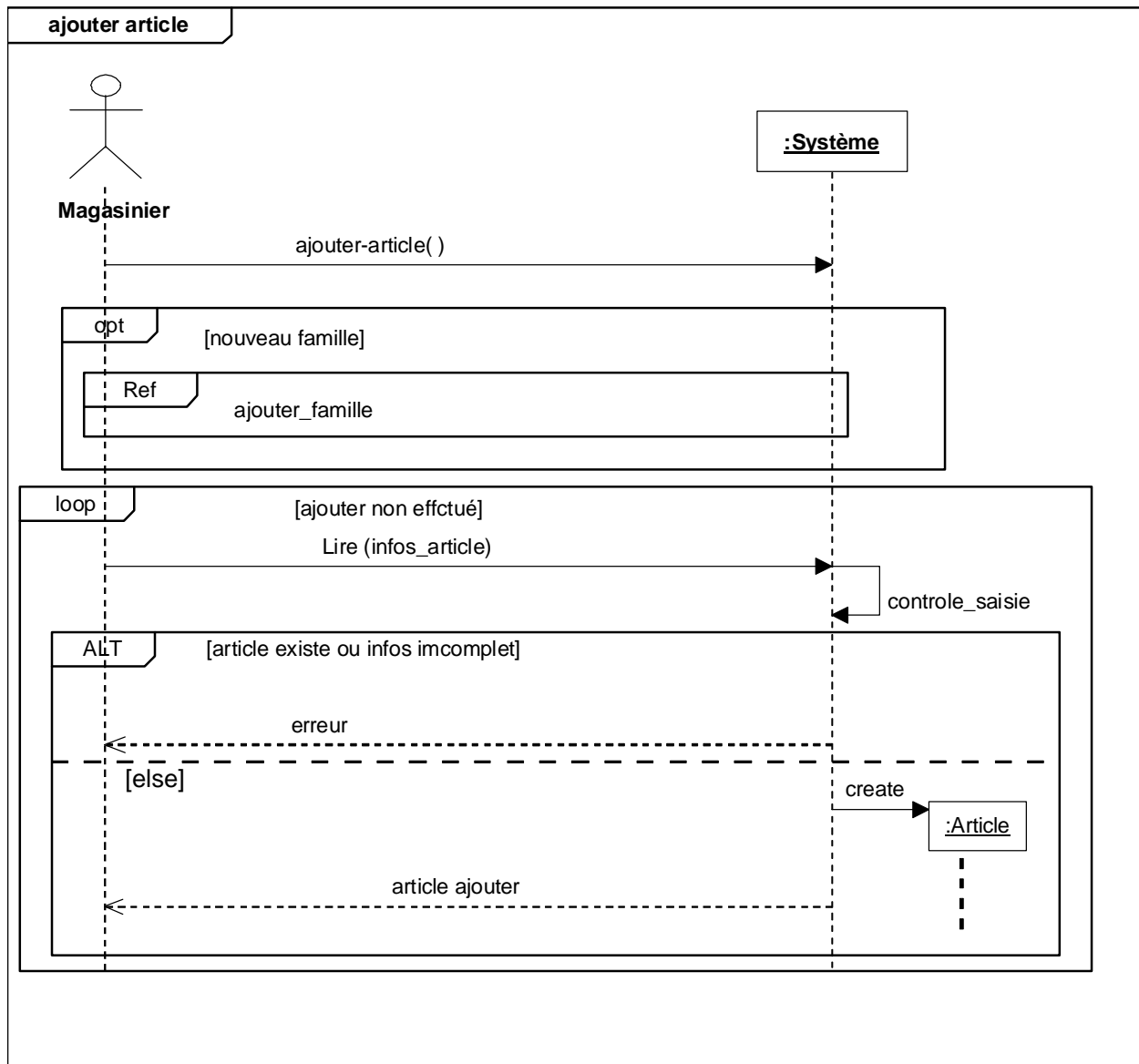


Figure3.7 : Diagramme de séquence du cas d'utilisation ajouter article.

3.5.7 Mise à jour des articles

3.5.7.1 Modifier article

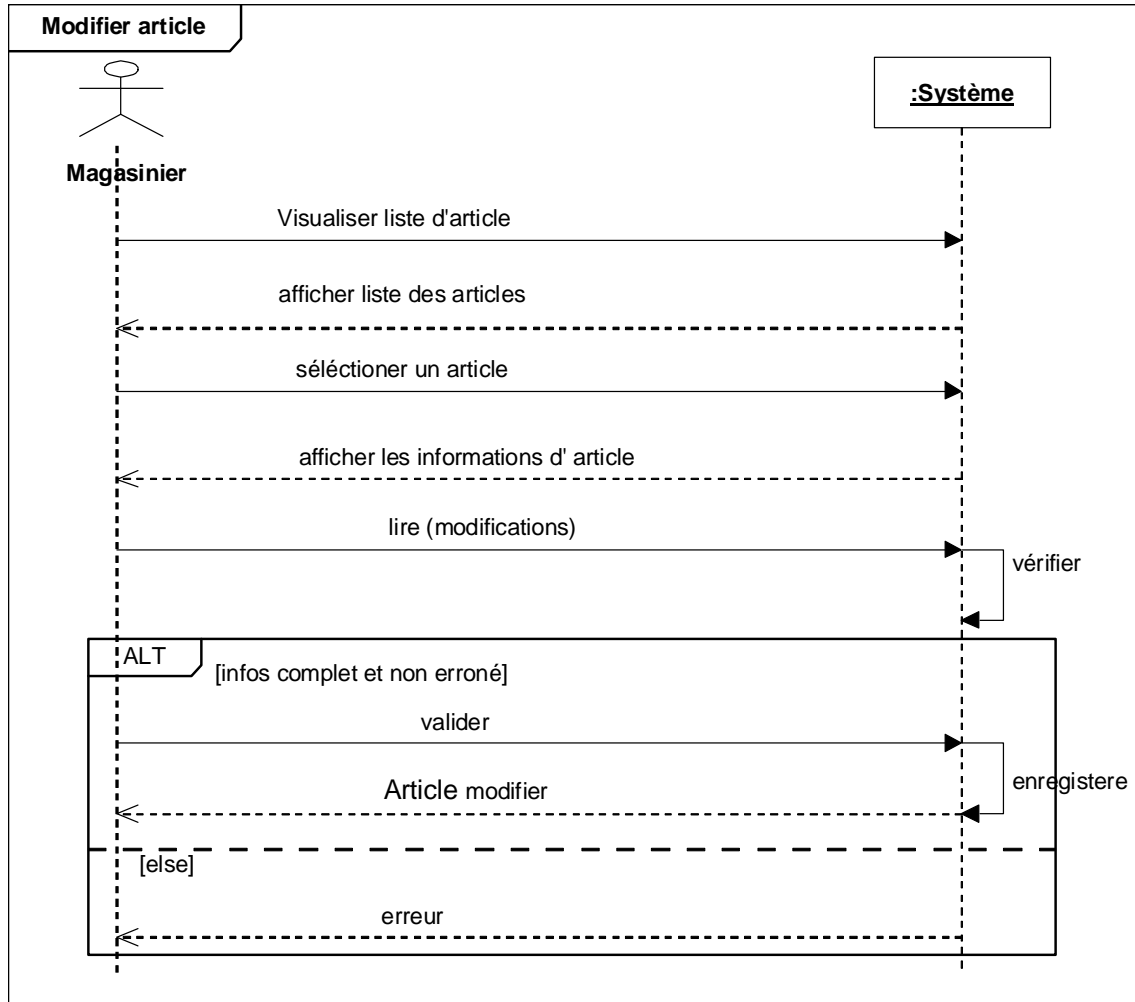


Figure 3.8 : Diagramme de séquence du cas d'utilisation modifier article.

3.5.7.2 Supprimer article

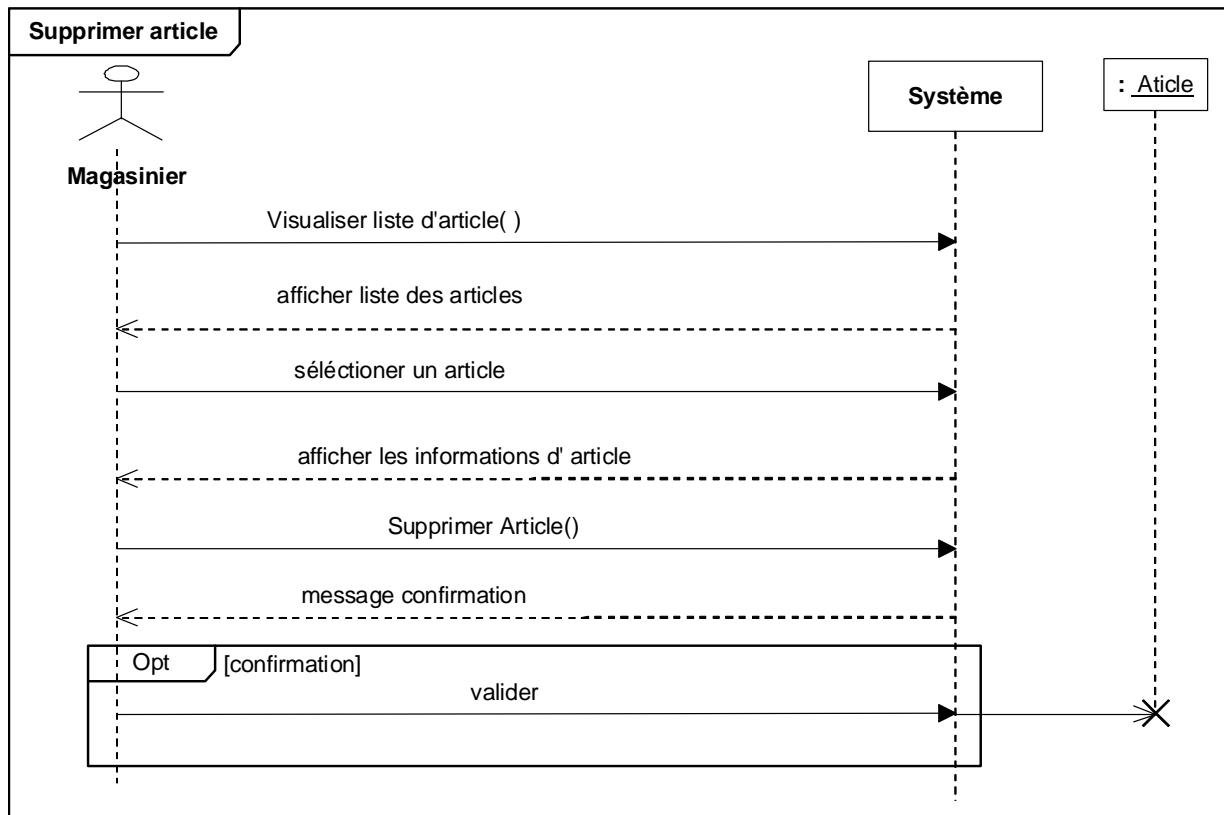


Figure 3.9: Diagramme de séquence du cas d'utilisation supprimer article.

3.5.8 Ajouter famille

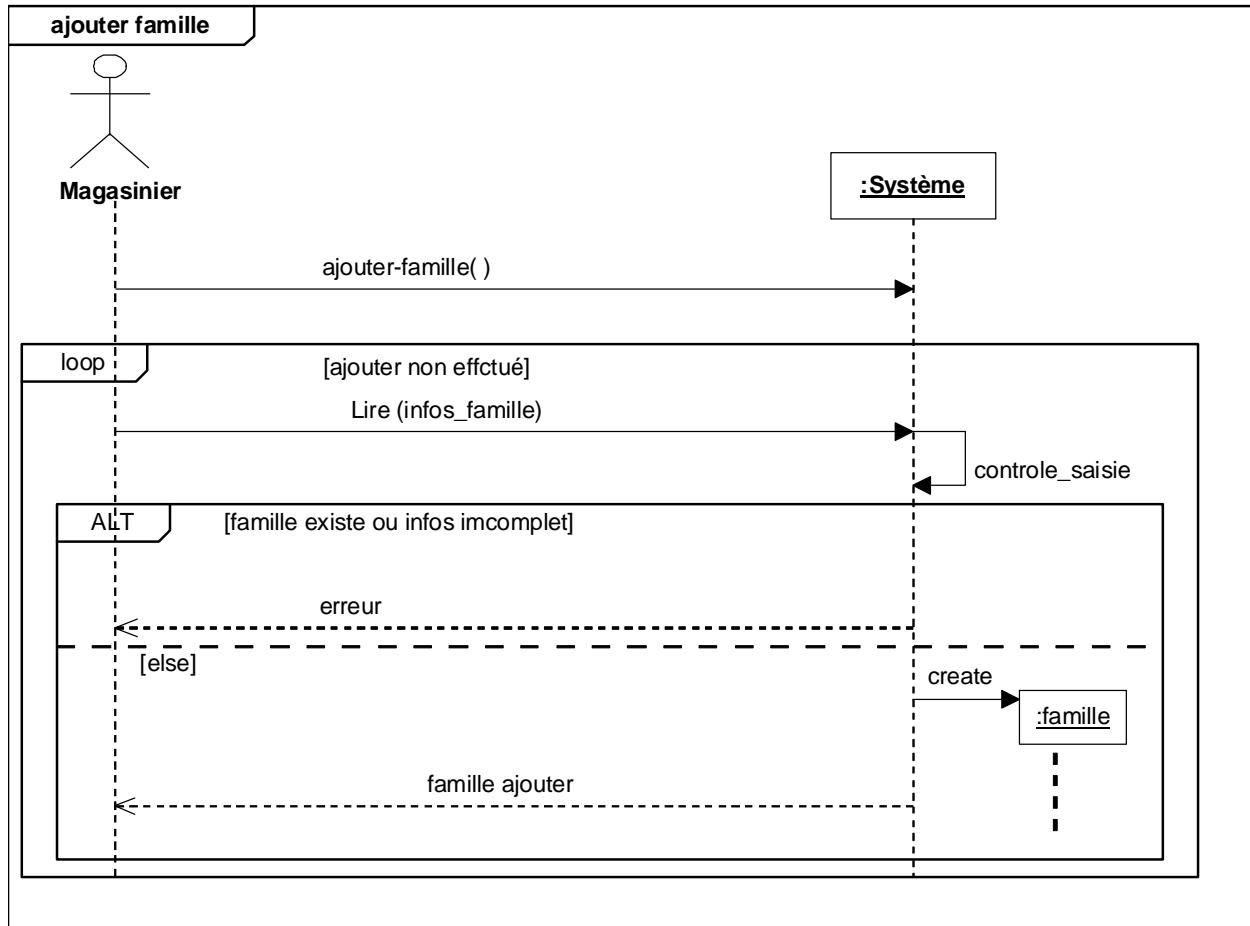


Figure3.10 : Diagramme de séquence du cas d'utilisation ajouter famille.

3.5.9 Mise à jour des familles

3.5.9.1 Modifier famille

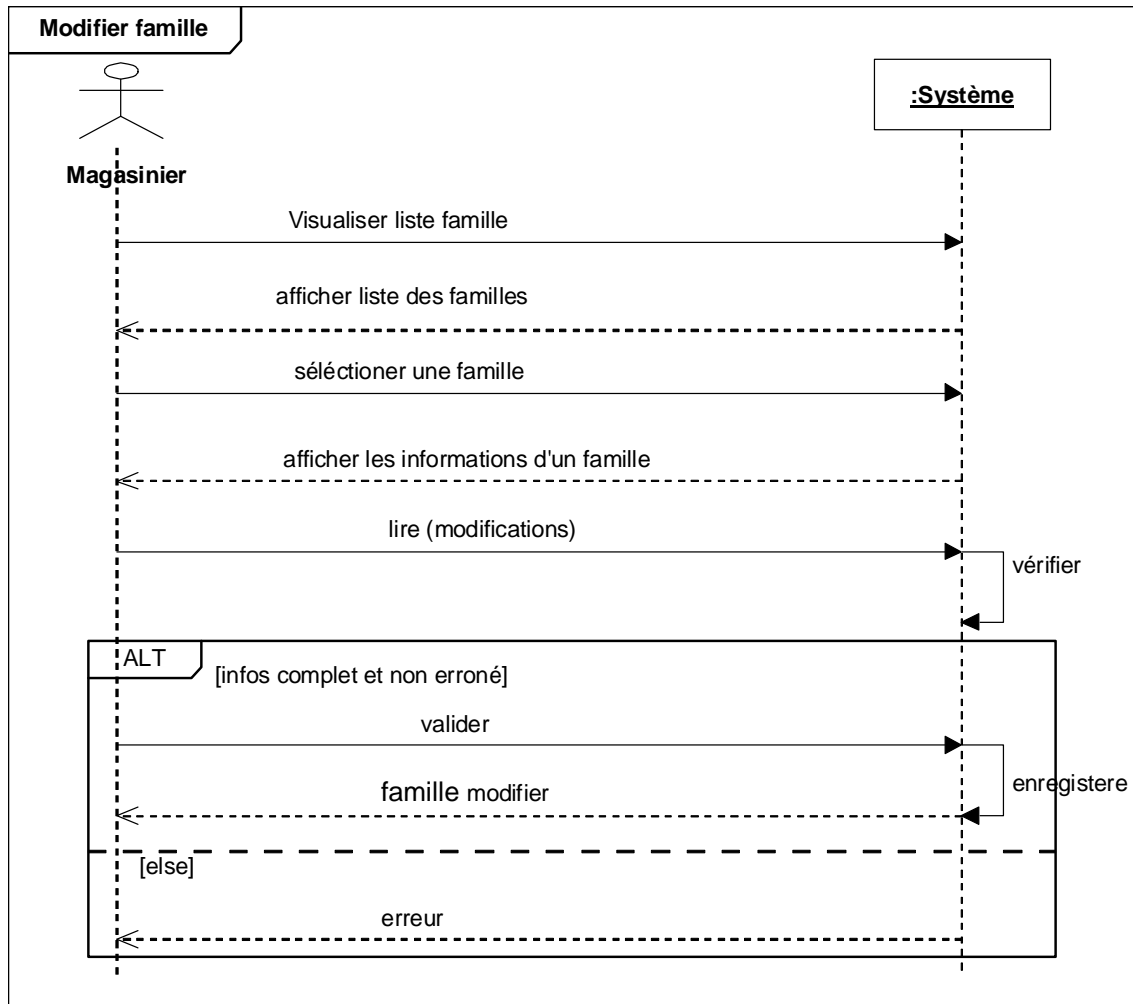


Figure 3.11 : Diagramme de séquence du cas d'utilisation modifier famille.

3.5.9.2 Supprimer famille

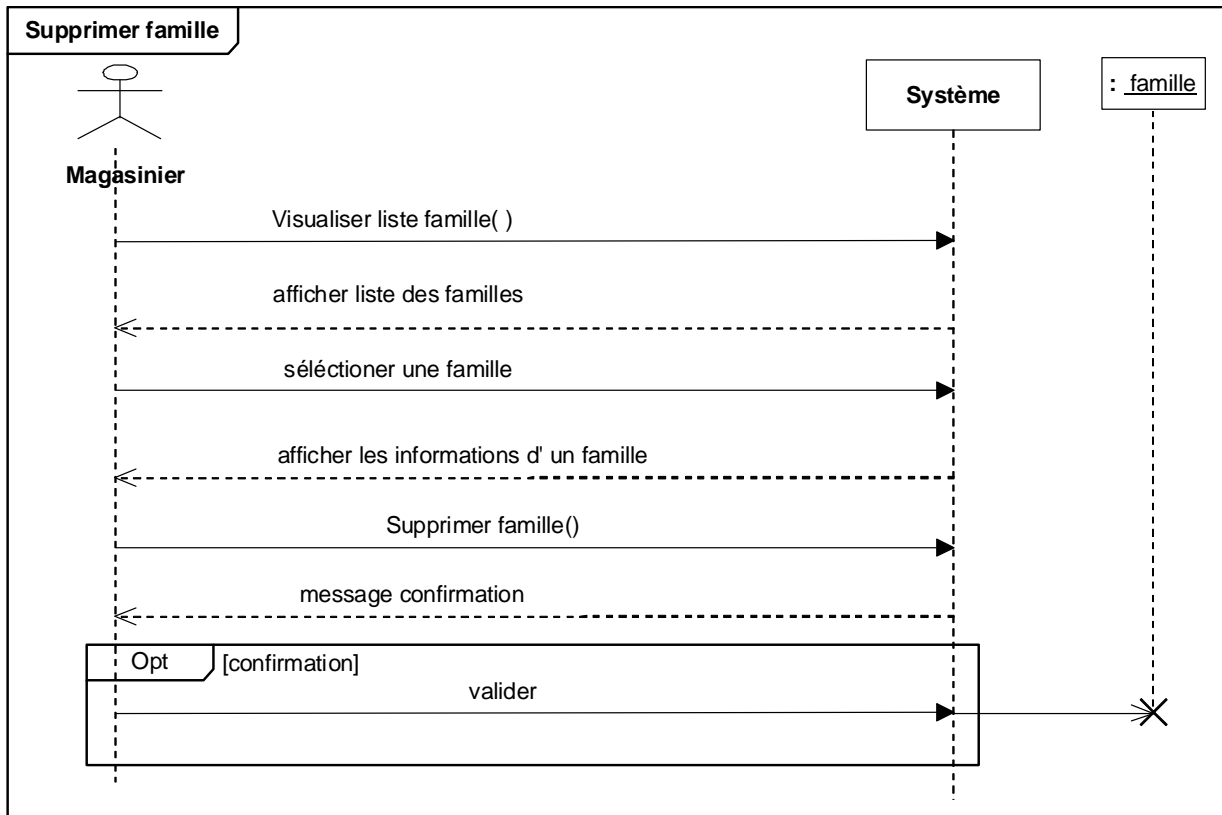


Figure 3.12: Diagramme de séquence du cas d'utilisation supprimer famille.

3.5.10 Ajouter fournisseur

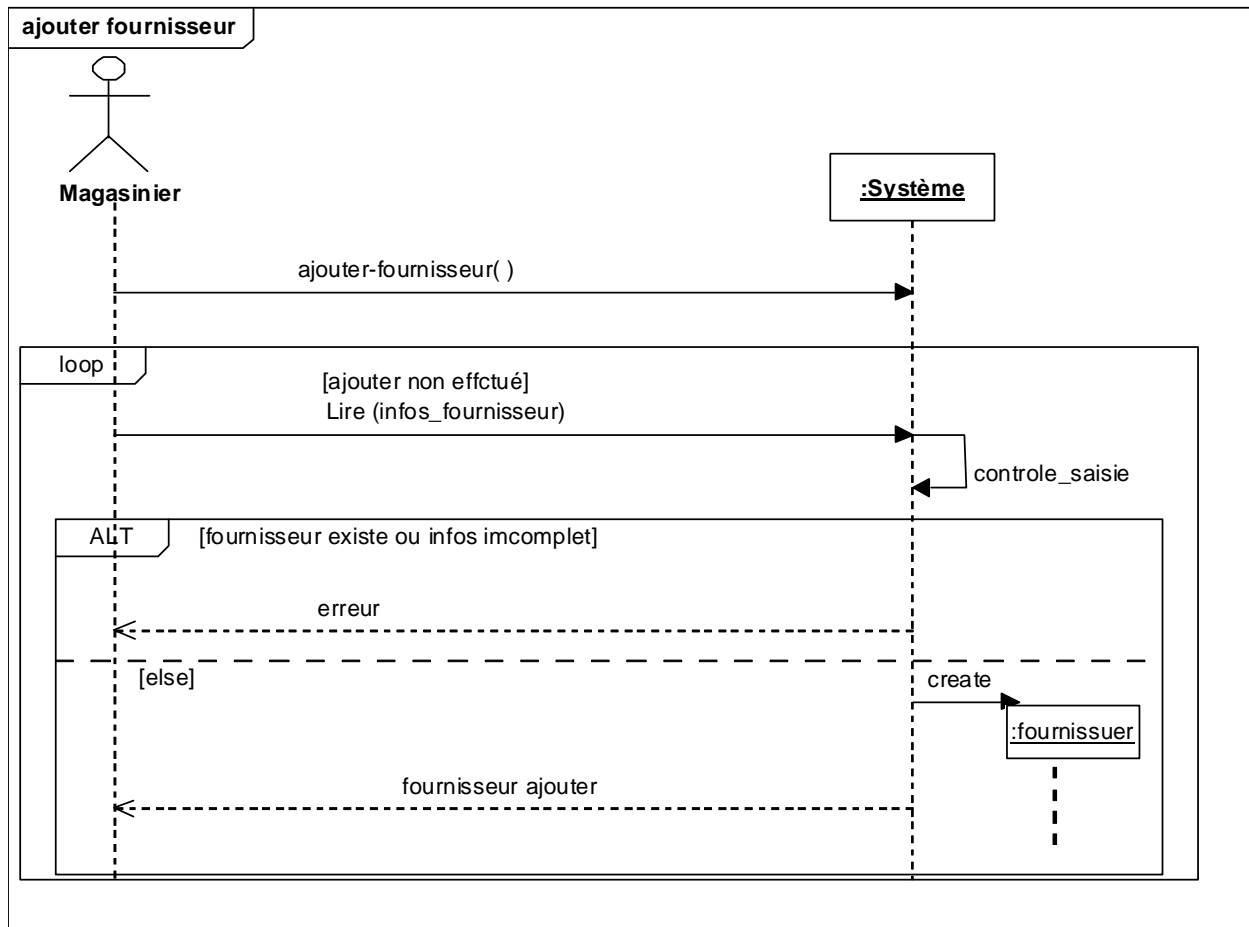


Figure 3.13 : Diagramme de séquence du cas d'utilisation ajouter fournisseur.

3.5.11 Mise à jour des fournisseurs

3.5.11.1 Modifier fournisseur

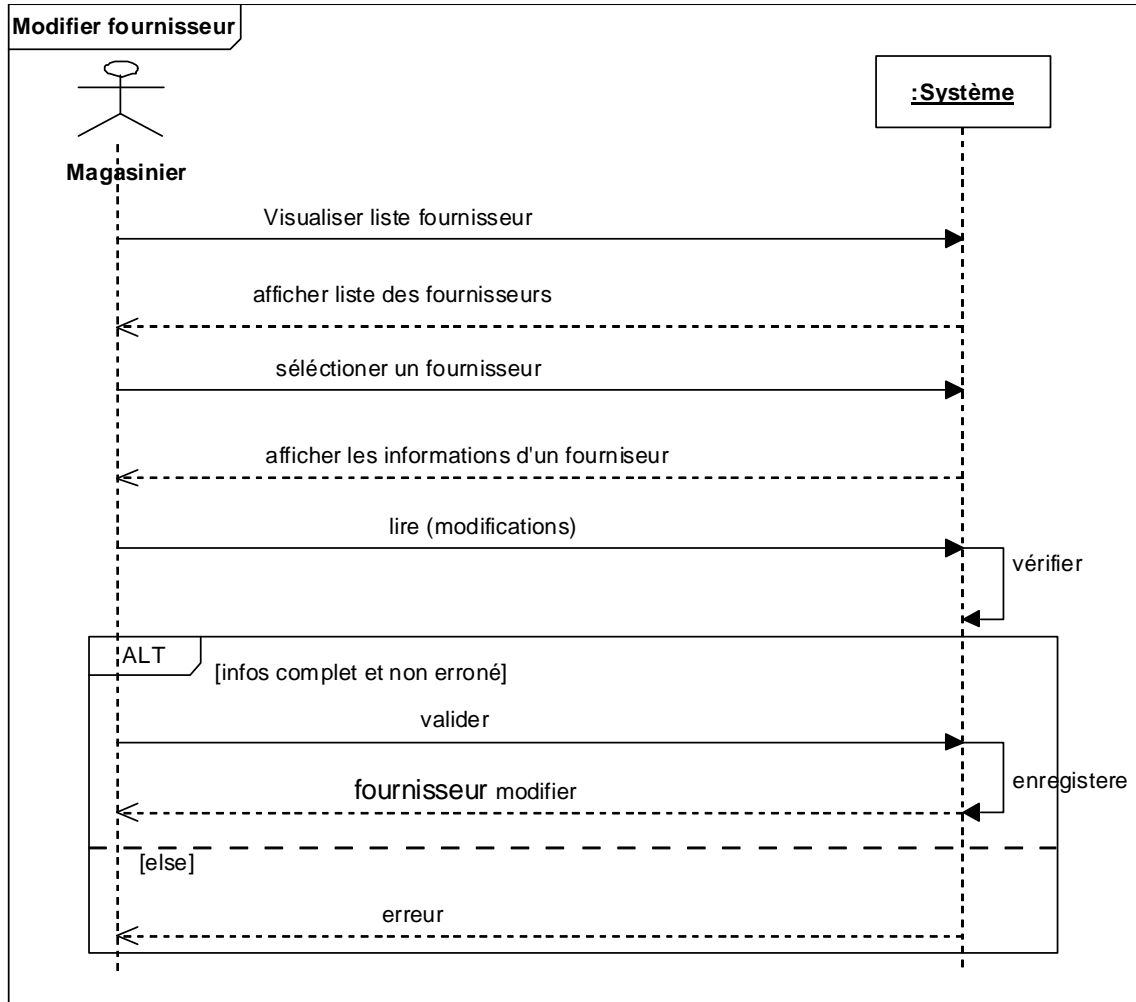


Figure 3.14: Diagramme de séquence du cas d'utilisation modifier fournisseur.

3.5.11.2 Supprimer fournisseur

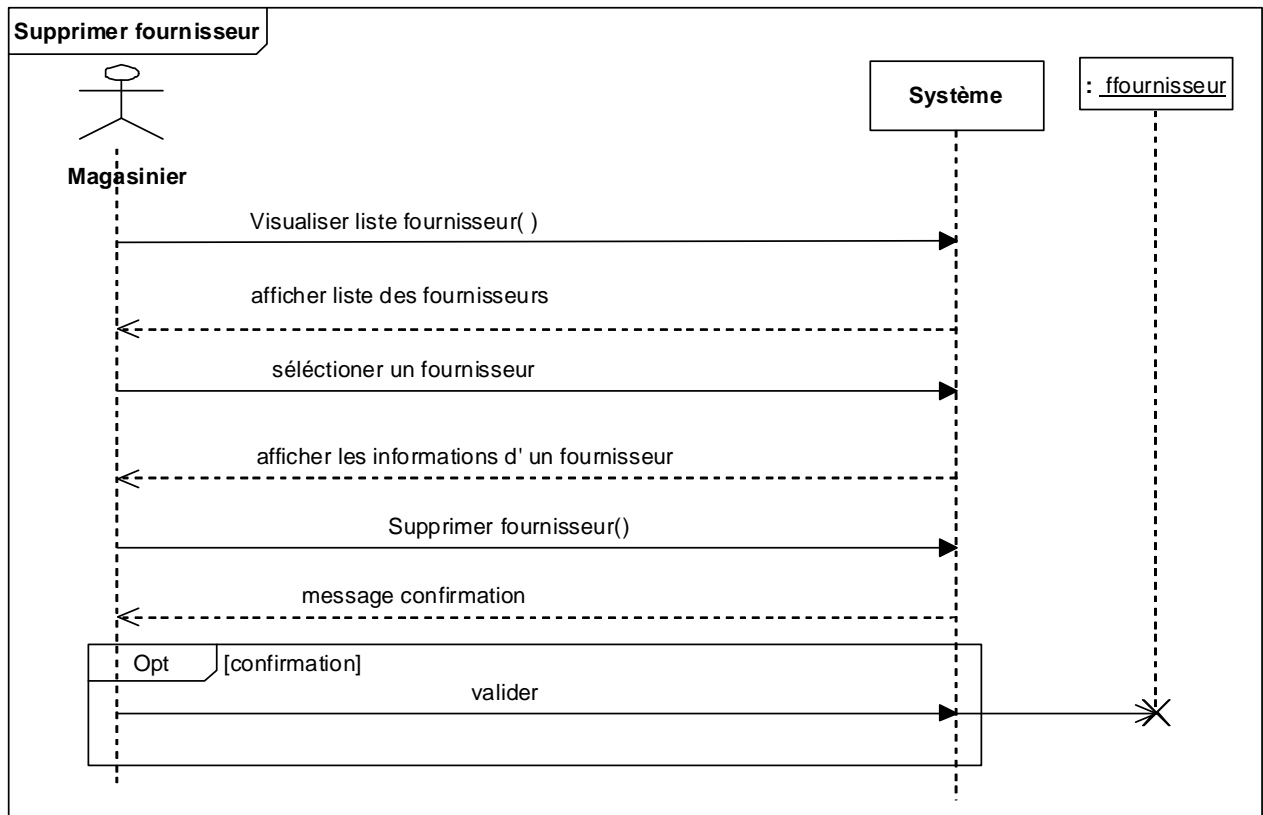


Figure 3.15 : Diagramme de séquence du cas d'utilisation supprimer fournisseur.

3.5.12 Ajouter service

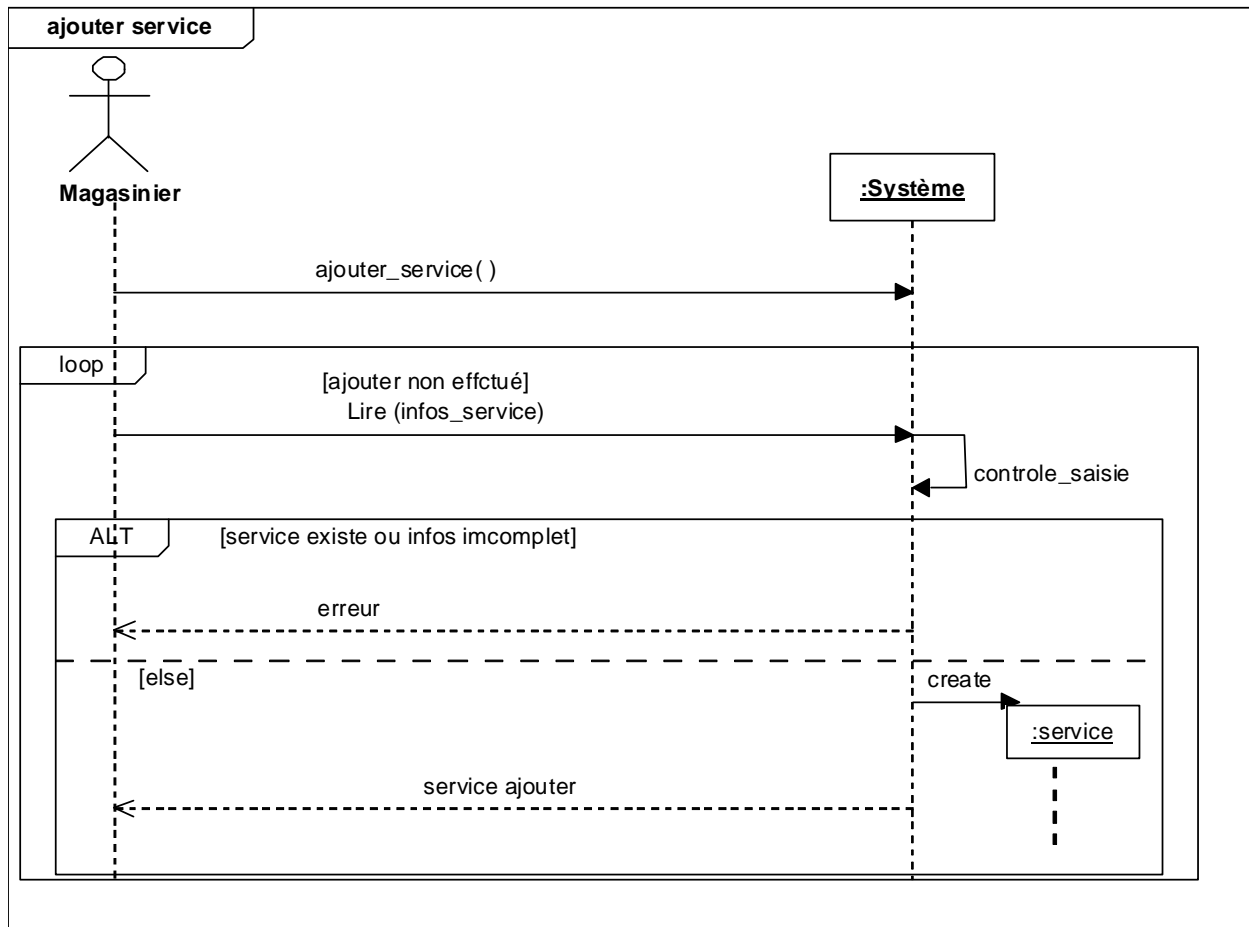


Figure 3.16 : Diagramme de séquence du cas d'utilisation ajouter service.

3.5.13 Mise à jour des services

3.5.13.1 Modifier service

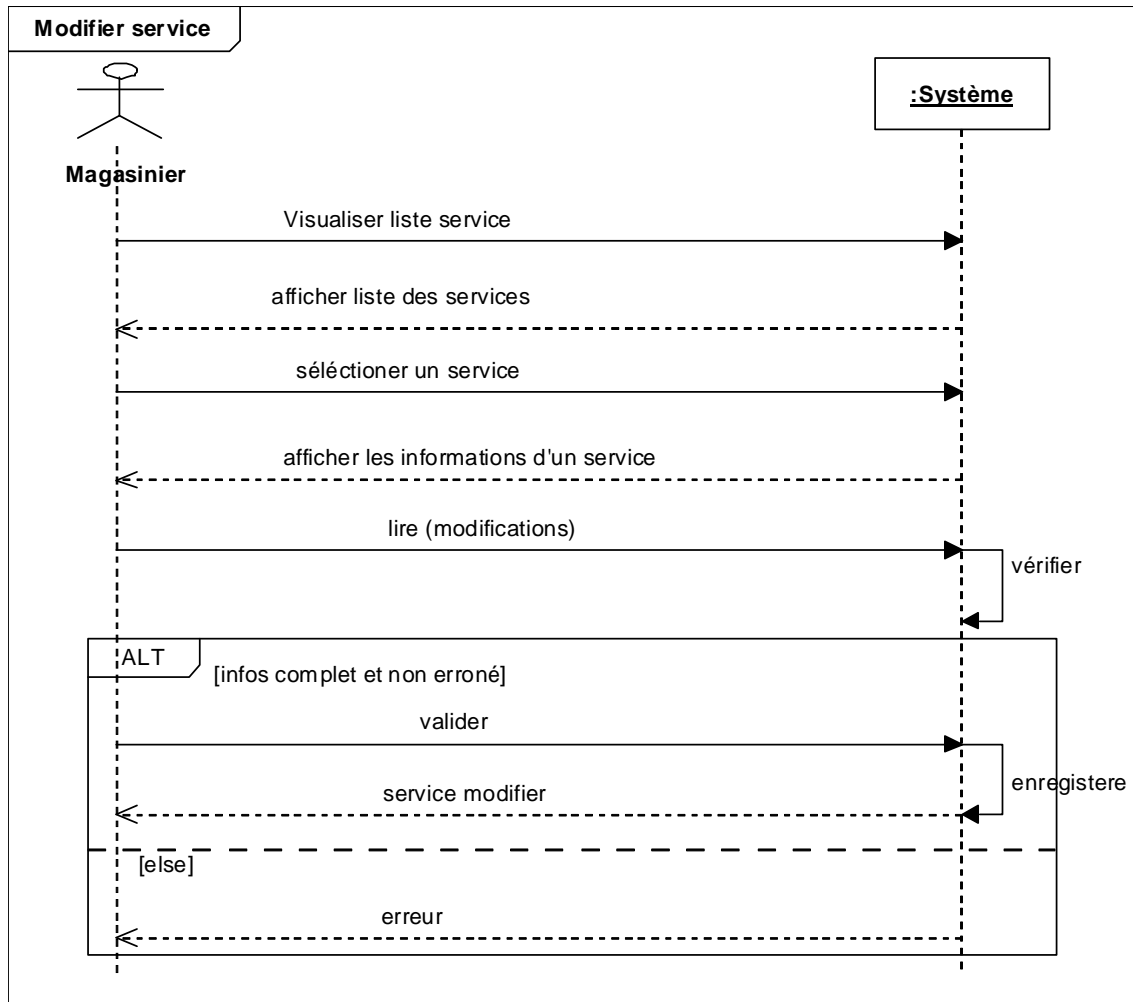


Figure 3.17 : Diagramme de séquence du cas d'utilisation modifier service.

3.5.13.2 Supprimer service

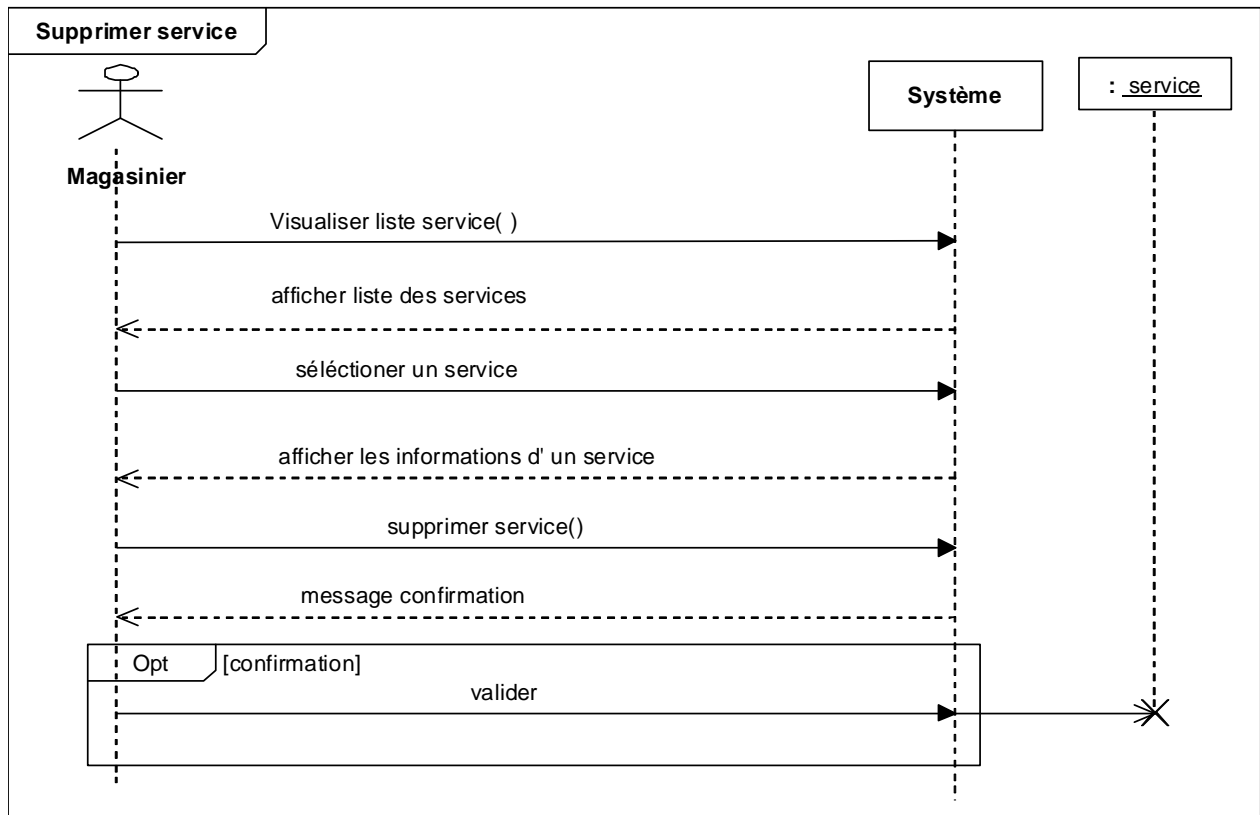


Figure 3.18 : Diagramme de séquence du cas d'utilisation supprimer service.

3.6 Le diagramme de classe

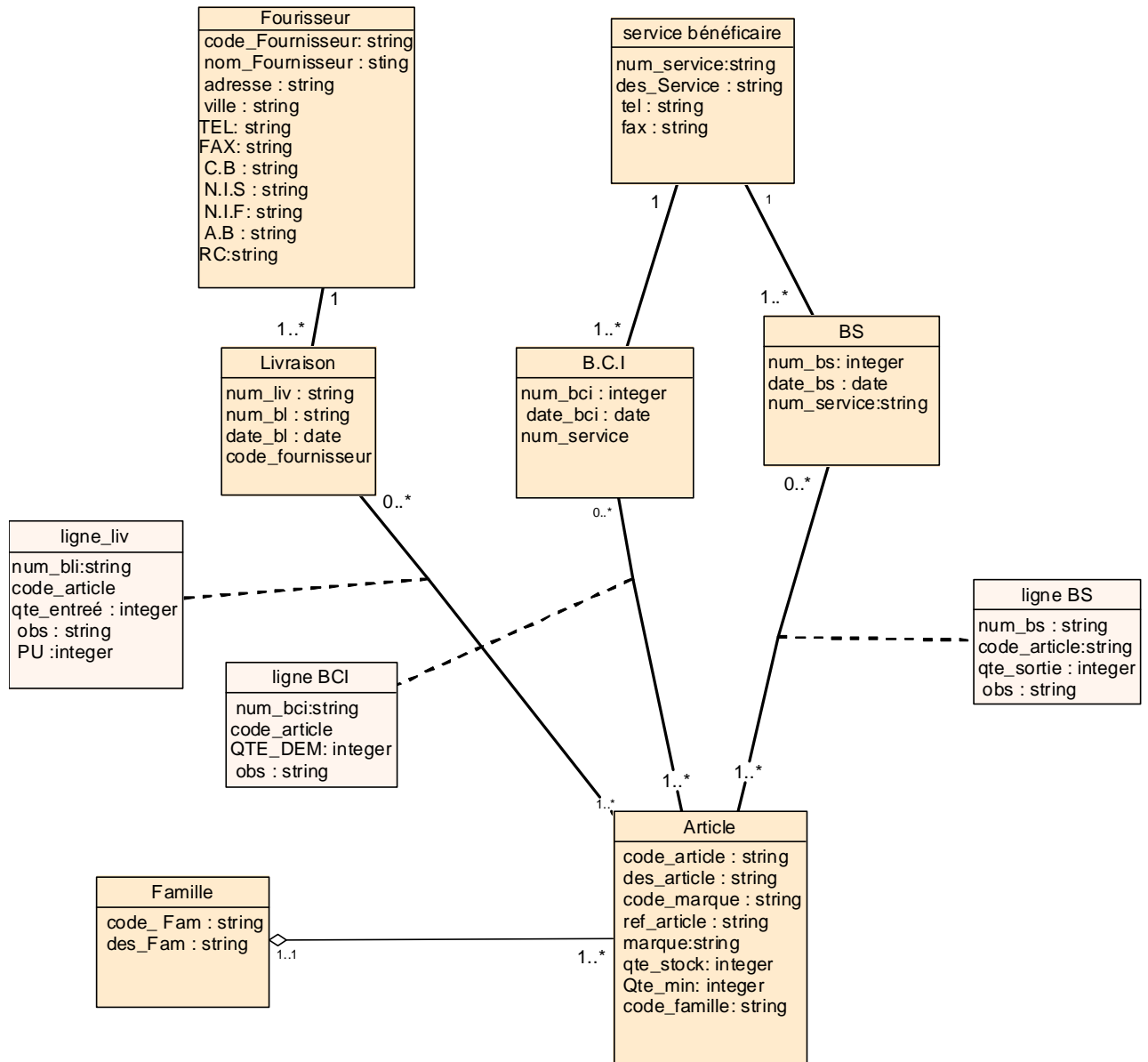


Figure 3.19 : Diagramme de classe

3.7 Conclusion :

Arrivant à cette étape, on peut dire que nous avons analysé le système qu'on veut construire.

D'autre part, nous signalons que la solution a été mise à jour tout au long de sa réalisation, car nous avons suivi une démarche itérative et incrémentale.

Enfin, la phase de conception abordé, permette de bien cerner la solution proposée et mieux comprendre le fonctionnement du système et ses différentes fonctionnalités, mais surtout permettent de préparer la phase de réalisation qui concrétisera tout ce qui a été présenté jusque-là.

CHAPITRE IV

IMPLEMENTATION

4.1 Introduction

Après avoir terminé l'étude conceptuelle, nous pouvons alors entreprendre la phase implémentation, ayant comme objectif d'aboutir à un produit final exploitable par les utilisateurs. Dans cette phase nous présentons les différents outils que nous avons utilisés pour le développement, puis nous décrivons quelques interfaces de l'application que nous avons extraites via des prises d'écrans (*screenprints*), afin d'illustrer les grandes et principales fonctionnalités réalisées.

4.2 Le passage du diagramme de classe au modèle relationnel

Nous donnons ci – après quatre règles (de R1 à R4) pour traduire le model du domaine en un schéma relationnel équivalent. Il existe d'autres solutions de transformation mais ces règles sont les plus simples et les plus opérationnelles :

Transformation des entités/ classes : La règle est simple :

R1 : * Chaque entité devient une relation, identifiant de l'entité devient clé primaire pour la relation.

* Chaque classe du diagramme UML devient une relation. Il faut choisir les attributs de la classe pouvant jouer le rôle d'identifiant.

Transformation des associations

Les règles de transformation que nous allons voir dépendant des cardinalités / multiplicités maximale des associations. Nous distinguons trois familles d'association :

Association 1..* : La règle est la suivante :

R2 : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

Association *.* : La règle est la suivante :

R3 : L'association / classe- association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connectées à l'association. Chaque attribut devient clé étrangère si l'entité / classe connectée dont il devient une relation en vertu de la règle R1. Les attributs de l'association / classe- association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Association 1..1 : La règle est la suivante :

R4 : Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimal égale à zéro. Dans le cas du diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité / classe connectée à l'association. Si les deux cardinalités minimales égale à zéro, le choix est donné entre les deux relations dérivées de la R1. Si les deux cardinalités minimales égale à un, il est préférable de fusionner les deux entités / classe en une seule.

Article (code_article, des_article, code_marque, ref_article, marque, qte_min,qte_stock,#
code_famille)

Famille (code_famille, des_famille)

Fournisseur (code_fournisseur, nom_fournisseur, adresse, ville, TEL, FAX, RIB, NIS, NIF,
RC, AB)

Service (num_service, des_service, tel, fax)

Livraison (num_liv, num_bl, date_bl,#code_fournisseur)

Ligne_liv (num_bli, code_article, qte_entreé, pu, obs)

BCI (num_bci, date_bci,# num_service)

Ligne_BCI(num_bci,code_article, QTE_DEM,obs)

BS(num_bs, date_bs,# num_service)

Ligne_BS (num_bs, code_article, qte_sortie, obs)

4.3 Environnement de développement de l'application

4.3.1 Pourquoi Delphi ?

Pour réaliser notre application, notre choix a été centré sur DELPHI pour les différentes raisons :

- Delphi est un environnement de programmation visuelle, orienté objet pour le RAD (Rapid Application Development). Il permet de développer des applications pour Windows 98, Windows NT, Windows XP.
- Delphi est un outil moderne, qui fait appel à une conception objet.
- Il prend en charge le maintien automatique d'une partie du code source.
- Il permet de créer facilement de nouveaux composants qui peuvent être intégrés dans la palette des composants déjà existants.
- Son compilateur intégré permet une application rapide et efficace car les erreurs éventuelles du code sont immédiatement détectées. L'utilisateur est alors informé précisément des erreurs de son programme.
- Il permet également d'utiliser des formats images, textes, sons, grâce à certains composants.

Delphi n'est pas lié à un format de données spécifiques. Il peut en effet utiliser des tables, DBASE, Access, ou Paradox et accéder à des bases de données SQL serveur à travers un ODBC (Open Data Base Connectivity).

4.3.2 Implémentation de la base de données sous Access

Pour implémenter notre base de données « Gstock », nous avons utilisé Microsoft Access. Il s'agit d'un système qui sert à créer, et à gérer des bases de données relationnelles. Cette base de données a une extension en.mdb . Il s'agit de bases de données accessibles à travers le moteur qui porte le nom de JET.1-4-b utilisé nativement par Access.

4.4 Contrôle et sécurité

4.4.1 Contrôle

L'erreur peut naître en de nombreux points du circuit de l'information, elle peut être provoquée par un facteur humain ou par une défaillance du matériel.

Il est important de détecter et rectifier l'erreur dès son apparition, car une information non correcte risque de rendre incorrectes de nombreuses autres informations. Par conséquent, différents contrôles s'imposent :

- Vérifier les informations à saisir ;
- Contrôler le type de l'information ;
- Contrôler la présence de l'information ;
- Utiliser les messages d'aides ;
- Des contrôles sont réalisés grâce à des relations de comparaisons.

4.4.2 Sécurité

Le concept de sécurité en informatique est très large, il s'agit de la sécurité de tous les dangers qui menacent tous les systèmes d'informations. Pour assurer le bon fonctionnement du système mis en place et pour éviter les destructions dues imprévues, il est indispensable de prévoir des mesures de sécurité pour protéger l'ensemble des informations contre toutes actions frauduleuses.

Pour cela nous avons distingué plusieurs mesures de sécurité :

- Il faut sauvegarder l'application sur des supports fiables.
- Utiliser les mots de passe.
- Utiliser les anti-virus.

4.5 Interfaces de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de notre application de gestion de stock.

4.5.1 Interface authentification

L'interface d'authentification permet au magasinier d'accéder à sa session par le biais de son login et son mot de passe.

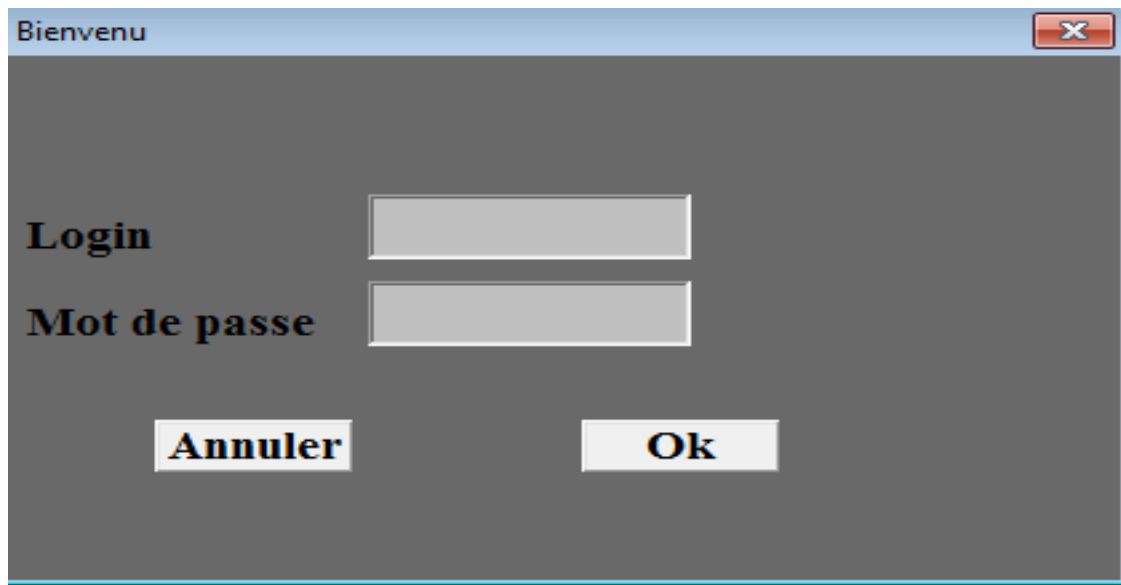


Figure 4.1 Interface authentification.

4.5.2 Interface du menu principale (Gestion de stock)

Cette interface permet d'avoir une vue globale sur l'application et l'accès à toutes ses fonctionnalités.

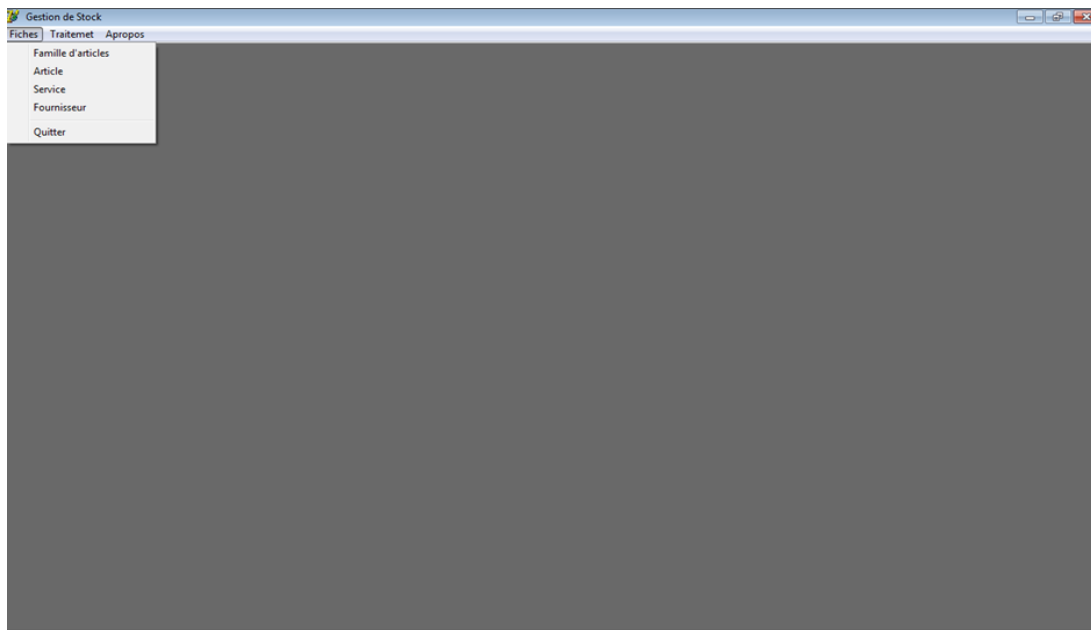


Figure 4.2 Interface menu principale (Gestion de stocke).

4.5.3 Interface Famille d'articles

Cette interface permet d'ajouter, supprimer et modifier une famille d'articles. Ainsi, elle permet de visualiser toutes les familles des articles existantes.

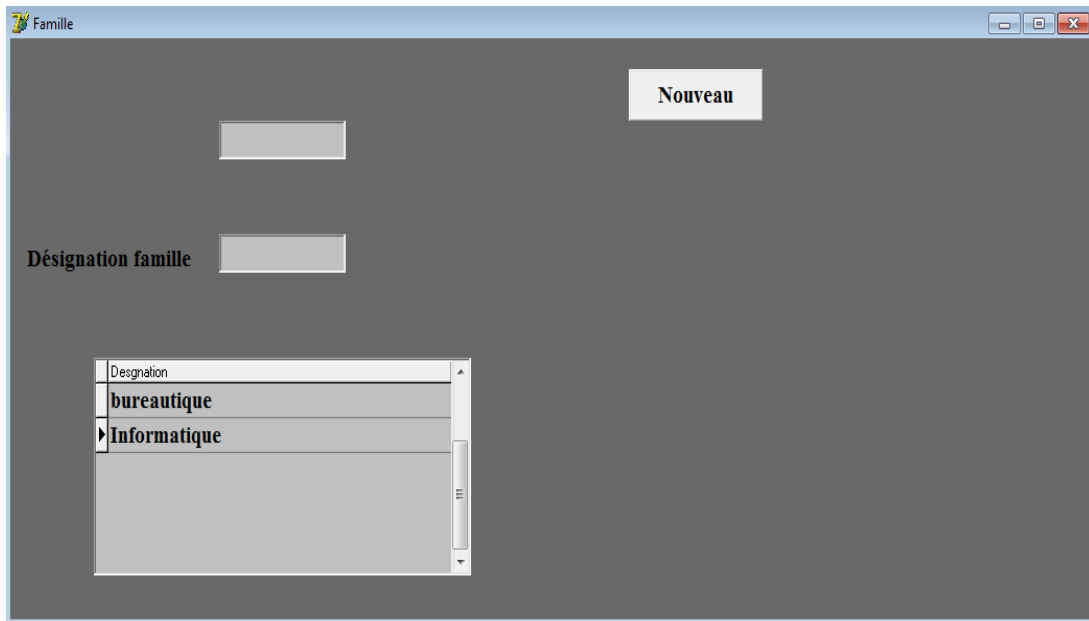


Figure 4.3 Interface Famille d'articles.

4.5.4 Interface article

Cette interface permet d'ajouter, supprimer et modifier un article. Ainsi, elle permet de visualiser la liste de tous les articles existants.

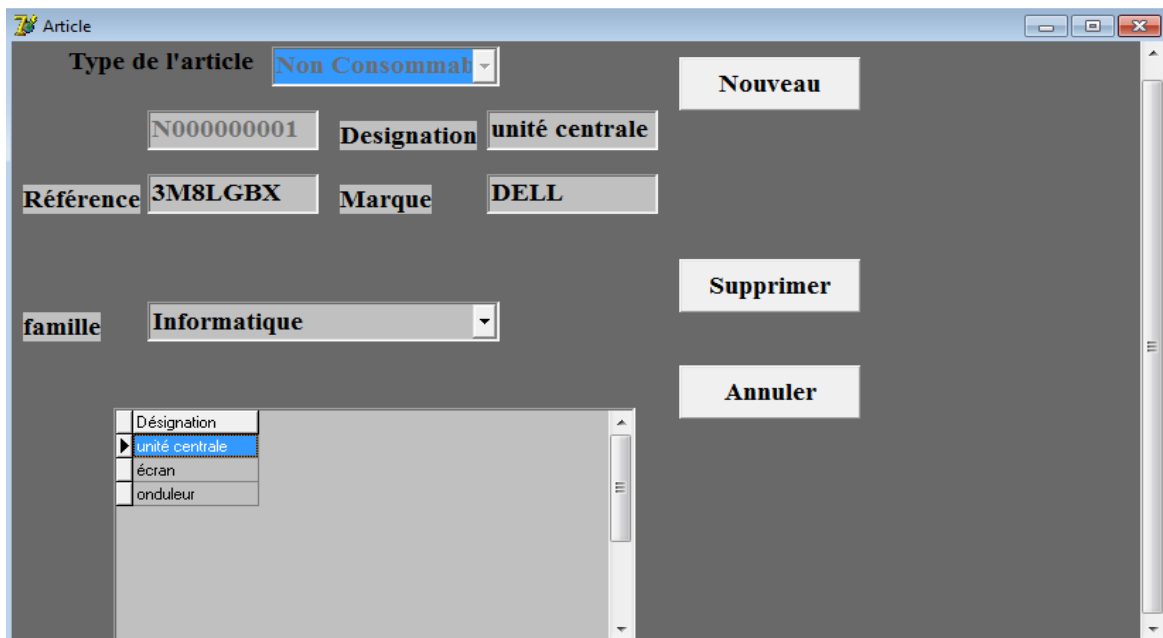


Figure 4.4 Interface article.

4.5.5 Interface service

Cette interface permet d'ajouter, supprimer et modifier un service. Elle permet aussi de visualiser la liste de tous les services existants.

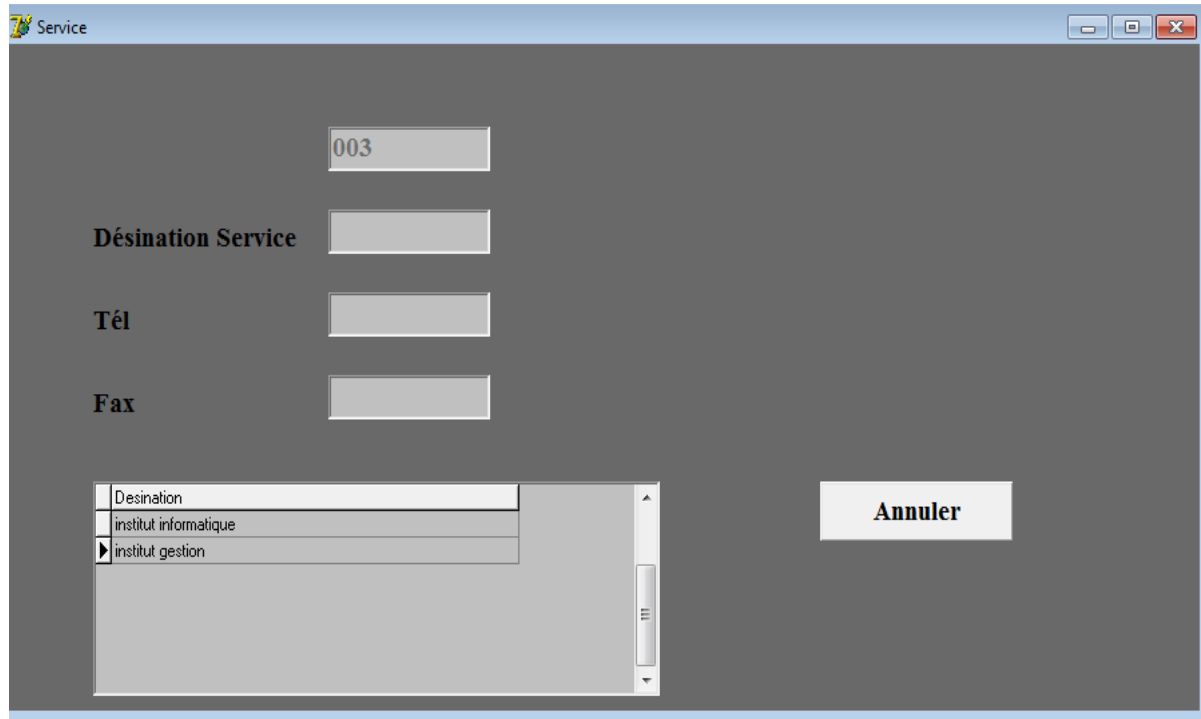


Figure 4.5 Interface service.

4.5.6 Interface fournisseur

Cette interface permet d'ajouter, supprimer et modifier un fournisseur. Elle permet aussi de visualiser la liste de tous les fournisseurs existants.

0000000002

Nom: Fouad RIB: 333400221111-26

Adresse: Loutr Zrghaia Mil NIS: 19744312000041

Ville: Mila NIF: 19744312000041

Tel: 0550801199 RC: 430100507

Fax: 031560115 AB: C.P.A.MILA

Enregistrer

Annuler

NOM
sami

Figure 4.6 Interface fournisseur.

4.5.7 Interface livraison

Cette interface permet d'ajouter, chercher, consulter et modifier une livraison, comme elle peut permettre la visualisation du détail de chaque livraison.

Bon de reception

N de livraison

N Bon de livraison Date de livraison: 04/05/20

Fournisseur: sami Téléphone

Nouveau

Consulter

--

Figure 4.7 Interface livraison.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

1 Bilan

Ce projet qui s'inscrit dans le cadre de la gestion du stock, était à l'objectif de répondre aux deux principaux processus de cette activité, à savoir : la gestion des Entrées/Sorties de la marchandise. Pour ce faire, nous sommes passés par plusieurs phases, allant de la collecte d'informations acquises de l'étude de l'existant, passant par la suite par la phase d'analyse et de conception. Durant cette étape, nous avons eu recours au langage de modélisation UML tout en suivant la méthode proposée par LAURENT AUDIBERT. Pour finir nous avons opté pour la mise en œuvre de l'application, l'environnement de développement DELPHI.

Au terme de notre travail, nous espérons avoir mis en place un système d'information qui offre une solution qui prend en charge les principaux besoins du magasin même si des mises à jour doivent être apportées au système développé.

En fait, à la fin de la réalisation de ce mémoire il est important de dire que ce projet nous a permis d'exploiter nos connaissances théoriques acquises pendant le cycle de notre formation. En outre, nous nous sommes mieux rapprochés et familiarisés avec un environnement dynamique. Par conséquent, nous avons pu avoir une idée plus approfondie sur l'importance des systèmes d'informations dans les entreprises.

2 Perspectives

Malgré les efforts consentis pour la mise en œuvre de ce système, il reste néanmoins primordial de lui appliquer de probables correctifs et améliorations nécessaires pour qu'il soit correctement exploitable.

Pour finir, nous prévoyons comme perspectives du travail réalisé dans ce mémoire, l'enrichissement du système actuel par l'implémentation des fonctionnalités suivantes :

- La prise en charge des statistiques ;
- Le suivi de l'opération d'approvisionnement ;
- La gestion de la facturation ;
- Et la gestion d'inventaire.

BIBLIOGRAPHIE

- [1]: *Benoît Charroux, Aomar Osmani, and Yann Thierry-Mieg. UML2. Pearson Education France, 2005*
- [2]: *Pascal ROQUES, UML 2 par la pratique étude de cas et exercices corrigés, ÉDITIONS EYROLLES, Septembre 2006.*
- [3]: *Laurent AUDIBERT, UML 2-de l'apprentissage à la pratique (cour et exercice).*