

MA-004-157-1

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saâd Dahlab, Blida
USDB.
Faculté des sciences.
Département informatique.



Mémoire pour l'obtention du diplôme de Master 2 en
Informatique
Option : Ingénierie de Logiciels

Sujet

**Conception et Réalisation D'un Système
D'encaissement Pour L'Hypermarché
UNO**

Organisme d'accueil :
NUMIDIS

Réalisé par :

ALILAT Wissam

Encadré par :

➤ Dr. N. BOUSTIA

➤ Mr. BELKALEM Karim

Devant le jury :

➤ Mme BENSTITI

➤ Mme MANCER

➤ Mme GUASMIA

Promotion : 2012/2013

MA-004-157-1

Resumé

NUMIDIS filiale du groupe CEVITAL est spécialisée dans la grande distribution, sa mission principale est de lancer et développer sa chaîne d'hypermarché et supermarché et de devenir l'enseigne de référence pour la famille algérienne en offrant à la fois un large choix de produits, de qualité aux meilleurs prix et de services irréprochables.

Pour ce faire, NUMIDIS a décidé de mettre en place un système d'encaissement fiable qui répond aux besoins des clients et de l'entreprise.

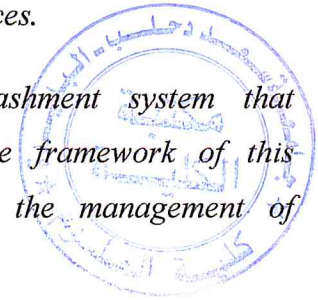
Dans le cadre de ce projet, notre travail consiste à mettre en œuvre un système d'encaissement pour la gestion des ventes, des clients, des employés, des articles, des remises, de la fidélité...

Mot clé : Encaissement, Vente, Clients, Remise , Fidélité

Abstract

NUMIDIS, subsidiary of CEVITAL group, specialized in big delivery, its main mission is to promote and develop its hypermarket and supermarket chain , and to become the reference brand for the Algerian family by offering at the same time a large choice of products, from quality to best prices and faultless services.

For this sake, NUMIDIS decided to install a reliable encashment system that responds to the needs of the enterprise's customers, within the framework of this project, our work is to implement an encashment system for the management of selling, of customers, employees, items, price cutting, fidelity,...



Key words: encashment, selling, price cutting, fidelity

Remerciements

Tout d'abord j'aimerais remercier DIEU de m'avoir offert le courage de réaliser ce projet et de m'avoir éclairé ma lanterne tout au long des années de mes études.

J'aimerais adresser mes remerciements les plus sincères à mon Encadreur Mr. BELKALEM Karim (Responsable de développement à NUMIDIS) pour sa patience, sa disponibilité, ses sacrifices, sa compréhension et ses conseils bénéfiques.

J'aimerais aussi adresser mes remerciements à ma promotrice Dr N. BOUSTIA d'avoir accepté d'encadrer mon projet et de m'avoir guidée dans la conduite de ce travail.

Aussi je voudrais remercier tous les enseignants du département d'informatique commençant par son chef Mr Massied

Mr BALA, Mme ABED, Mme ELGHERS, Mr CHERIF ZAHAR, Mr HAMOUDA, Mr HADJ YAHIA, Mme OOUAHRANI

Je tiens aussi à remercier tous mes collègues de la DSIO du NUMIDIS pour leurs aides, conseils et l'ambiance du travail

Enfin, je remercie toute personne ayant contribué de près ou de loin à la progression de ce projet.

Dédicace

Je dédie mon travail

*A ma chère et tendre grand-mère paternelle qui a précipité au-delà par la mort.
Qu'elle soit immortalisée par ce travail et que ses empreintes restent à jamais
marquer dans les souvenirs de mes pensées.*

lah yarhimek mani

A mes chers parents,

*Que nulle dédicace ne puisse exprimer ce que je leurs dois, pour leurs
bienveillance, leurs sacrifices, leurs affection et leurs soutien.*

« Que Dieu vous garde pour moi ».

A mes chers sœurs et frère Amira, Farah et Abderahmene

*À ma grand-mère que j'aime beaucoup et qui souhaite me voir réussir dans mes
études et dans ma vie, que dieu la garde pour nous.*

*A ma chère amie, sœur, mon âme Ben Bellal Rachda que dieu nous garde l'une
pour l'autre*

A mes amies MALEG Djaouida , AZZOUZ Imène et Manel

À mes tantes, mes oncles, mes cousins et cousines

*A tous mes amis de l'université : B.Amine , Imene , Yacine, Ramzy, Hanan,
Amine , Adel, Maria , Khalil, Doria, Djihad , Bouchra , Krimo....*



Wissam

Table des matières

| | |
|--|------------|
| <i>Resumé</i> | <i>i</i> |
| <i>Abstract</i> | <i>ii</i> |
| <i>Remerciements</i> | <i>iii</i> |
| <i>Dédicaces</i> | <i>iv</i> |
| <i>Table des matieres</i> | <i>v</i> |
| <i>Liste des figures</i> | <i>vii</i> |
| <i>Introduction Générale</i> | <i>1</i> |
| CHAPITRE 01 : ETUDE DE L'EXISTANT | |
| 1- DESCRIPTION DE LA STRUCTURE D'ACCUEIL | 3 |
| 2- ETUDE DE L'EXISTANT | 7 |
| 2-1 les fonctionnalités de WINSTORE..... | 7 |
| 2-2 Diagnostique..... | 11 |
| 2-3 Suggestion | 11 |
| CONCLUSION | 11 |
| CHAPITRE 02 : CONCEPTION | |
| 1- DEMARCHE ADOPTEE | 12 |
| 2- LANGAGE DE MODELISATION | 14 |
| 3- Capture des besoins fonctionnels | 15 |
| 3-1 Identification des acteurs | 15 |
| 3-2 identification des cas d'utilisations fonctionnels | 16 |
| 4- Capture des besoins technique | 26 |
| 4-1 <i>Identification des cas d'utilisation techniques</i> | 26 |
| 5- ANALYSE | 32 |
| 5-1 Conception du modèle statique..... | 32 |
| 5-1-1 diagramme de classe de base de données centrale..... | 32 |
| 5-1-2 Description des classes..... | 33 |
| 5-1-3 diagramme de classe de base de données caisse | 36 |
| 5-1-4 description des classes..... | 37 |
| 5-1-5 schéma de l'entrepôt..... | 40 |
| 5-2 Diagramme de séquence | 41 |
| 5-2-1 Diagramme de séquence du cas d'utilisation « créer un article » | 41 |
| 5-2-2 Diagramme de séquence du cas d'utilisation « modifier un article »..... | 42 |
| 5-2-3 Diagramme de séquence du cas d'utilisation « supprimer un article » | 42 |
| 5-2-4 Diagramme de séquence du cas d'utilisation « rechercher un article »..... | 43 |
| 5-2-5 Diagramme de séquence du cas d'utilisation « ouvrir une caisse » | 44 |
| 5-2-6 Diagramme de séquence du cas d'utilisation « payer un ticket par un bon d'achat »..... | 45 |
| 5-2-7 Diagramme de séquence du cas d'utilisation « créer une carte de fidélité » | 46 |
| 5-2-8 Diagramme de séquence du cas d'utilisation « créer un employé »..... | 47 |

| | |
|--|-----------|
| 6- Conception Générique | 49 |
| 6-1 Authentification..... | 49 |
| 6-2 Archivage des tickets..... | 50 |
| 6-3 La recherche | 50 |
| 7- Conception préliminaire | 51 |
| 7-1 Architecture de l’outil..... | 51 |
| 7-2 Architecture physique..... | 51 |
| 7-3 Développement du modèle de déploiement | 53 |
| 7-4 Critère ergonomique adoptés..... | 54 |
| 8- Conception détaillée | 56 |
| 8-1 Application des règles de passage sur le diagramme de classe | 56 |
| 8-1-1 Diagramme de classe de la base de données centrale | 57 |
| 8-1-2 Diagramme de classe de la base de données de la caisse..... | 60 |
| <i>conclusion</i> | 63 |
| CHAPITRE 03: REALISATION | |
| Introduction | 64 |
| 1- Choix des outils de développement..... | 64 |
| 2- L’environnement du travail..... | 66 |
| 3- Sécurité du nouveau système | 67 |
| 4- Présentation du prototype réalisé | 67 |
| <i>conclusion</i> | 69 |
| CONCLUSION GENERALE | 70 |
| ANNEXE1 :Etude comparative entre les processus de développement | 71 |
| Conclusion | 77 |
| ANNEXE2 : Benschmarking | 78 |
| Conclusion | 79 |
| BIBLIOGRAPHIE | 80 |

Table des Figures

| | |
|---|----|
| Figure 01 : Organigramme du groupe CEVITAL..... | 5 |
| Figure 02 : Diagramme du cas d'utilisation « Gestion d'articles» | 7 |
| Figure 03: Le processus de développement en Y adapté | 11 |
| Figure 04 : Diagramme des acteurs | 15 |
| Figure 05 : Diagramme du cas d'utilisation « Gestion d'articles» | 18 |
| Figure 06 : Diagramme du cas d'utilisation « rechercher un article» | 19 |
| Figure 07 : Diagramme du cas d'utilisation « gestion de la caisse » | 20 |
| Figure 08 : Diagramme du cas d'utilisation « Gestion des tickets » | 21 |
| Figure 09: Diagramme du cas d'utilisation « payer un ticket » | 22 |
| Figure 10 : Diagramme du cas d'utilisation « gestion de tiroir » | 23 |
| Figure 11: Diagramme du cas d'utilisation « gestion du magasin » | 23 |
| Figure 12 : Diagramme du cas d'utilisation « gestion de la fidélité » | 24 |
| Figure 13: Diagramme du cas d'utilisation « gestion des remises» | 25 |
| Figure 14 : Diagramme du cas d'utilisation « gestion des employés» | 27 |
| Figure 15 : Diagramme du cas d'utilisation « gestion des rayons» | 28 |
| Figure 16 : Diagramme du cas d'utilisation « gestion des modes de paiement» | 29 |
| Figure 17 : Diagramme du cas d'utilisation « gestion des clients» | 30 |
| Figure 18 : Diagramme du cas d'utilisation « gestion des ventes » | 31 |
| Figure 19: «Diagramme de classe de la base de données centrale» | 32 |
| Figure 20: «Diagramme de classe de la base de données de la caisse » | 37 |
| Figure 21 : Diagramme de séquence du cas « créer un article » | 41 |
| Figure 22 : Diagramme de séquence du cas « modifier un article » | 42 |
| Figure 23 : Diagramme de séquence du cas « supprimer un article ». | 43 |
| Figure 24 : Diagramme de séquence du cas « rechercher un article » | 44 |
| Figure 25 : Diagramme de séquence du cas « créer un article » | 45 |
| Figure 26 : Diagramme de séquence du cas « payer un ticket par un chèque cadeau/Bon d'achat » ... | 46 |
| Figure 27 : Diagramme de séquence du cas « créer une carte de fidélité ». | 47 |
| Figure 28 : Diagramme de séquence du cas « créer un employé ». | 48 |
| Figure 29 : Architecture 3tiers..... | 53 |
| Figure 30 : Diagramme de déploiement..... | 54 |
| Figure 31 : Interface de la page d'accueil | 68 |
| Figure 32 : Interface du compte du responsable de magasin | 68 |
| Figure 33 : Interface du compte de l'arrière caissier..... | 69 |
| Figure 34 : Interface de la liste des remises sur des articles..... | 69 |
| Figure 1: Le processus en cascade..... | 71 |
| Figure 2: Cycle en V | 72 |
| Figure 3: Le processus XP..... | 72 |
| Figure 38: Le processus 2TUP | 73 |
| Figure 4: Le processus UP..... | 74 |
| Figure 40: Le processus RUP | 75 |
| Figure 5: Le cycle en spirale | 75 |
| Figure 42: Le modèle incrémental..... | 76 |

Introduction générale

La gestion des espèces est un enjeu majeur pour NUMIDIS. En effet, les spécificités du marché Algérien font que la quasi-totalité des règlements sont effectués en espèces. Cette situation particulière a des impacts non négligeables pour NUMIDIS :

- ✚ Le temps passé à la gestion des espèces est important
- ✚ La gestion des espèces empiète fortement sur les aspects organisationnels de la gestion du magasin.
- ✚ Les diverses manipulations d'espèces réalisées au cours d'une journée type ont un coût financier.
- ✚ L'omniprésence des espèces rend attrayante la fraude interne
- ✚ La sécurité des personnes n'est pas garantie, aussi bien dans le magasin que pendant le transport de la recette à la banque.

Ces considérations font que l'implémentation d'une solution de gestion des encaissements peut être un atout majeur pour NUMIDIS

- ❖ La mise en place d'un système d'encaissement et de fidélité fiable est sans aucun doute un processus qui pourrait grandement simplifier l'organisation du travail dans les magasins NUMIDIS tout en améliorant la productivité des employés, la sécurité de la ligne de caisses et la satisfaction de la clientèle.

La solution doit permettre l'utilisation de différents moyens de paiement, tels que : des cartes bancaires, des cartes distributeur, des cartes prépayées, un porte-monnaie électronique etc.

Notre document est structuré comme suit :

Chapitre 1 : *Etude de l'Existant*

Ce chapitre contient on présente la structure d'accueil de notre projet ainsi une analyse du système existant au niveau de l'entreprise.

Chapitre 2 : Conception

Ce chapitre s'occupe de la conception de notre système commençant par la démarche adoptée, capture des besoins technique et fonctionnels, modèle statique, modèle dynamique, architecture du système et modèle de déploiement.

Chapitre 3 : Réalisation

Ce chapitre comporte la partie de la mise en œuvre du système ; il aborde la description de l'outil ainsi que le développement de l'application.



Chapitre 01 :

ETUDE DE L'EXISTANT

Chapitre 01 : ETUDE DE L'EXISTANT

1- Description de la structure d'accueil :

❖ Présentation du Groupe CEVITAL :

Le groupe CEVITAL a été créé en 1971 par son fondateur Mr ISSAD REBRAB avec la création et la reprise de plusieurs sociétés de construction métallique.

- CEVITAL est une Holding à croissance soutenue.
- Le groupe CEVITAL c'est aussi un chiffre d'affaire de 3 Milliards USD en 2012.
- CEVITAL est un groupe présent sur plus de dix métiers.

❖ Historique du Groupe CEVITAL :

1971-1988 - Création et reprise de plusieurs sociétés de construction métallique : SOCOMEG, PROFILOR, SOTECOM, SACM, ENALUX, NORD METAL, METALLOR, METAL SIDER

1991 - Création de JBM - Reprise des activités IBM en Algérie. - Création du journal « Liberté » – Quotidien d'information

1992 - Création de C.B.S. - Reprise des activités de Rank Xerox

1995 - Création de Agro-grain - Importation et distribution de produits agro-alimentaires

1997 - Création de Hyundai Motors Algérie - Distribution de véhicules de Hyundai Corporation

1998 - Création de Cevital SPA – Industries agroalimentaires - Bejaia

2005 - Unité d'eau minérale plate et gazeuse et de sodas - Deux unités de fabrication de bâtiment préfabriqué en béton

2006- Reprise de l'unité COJEK - Jus et conserves - Lancement du projet NUMIDIS - Grande distribution

2007 – Reprise de BATICOMPOS – Matériaux de construction préfabriqués - Création de SAMHA - Distribution de produits électroniques et électroménagers de

Chapitre 01 : ETUDE DE L'EXISTANT

marque SAMSUNG Electronics en Algérie => *En 2013, La Filiale SAMHA se voit accorder le statut d'Opérateur Economique Agrée (OEA) par la Direction Générale des Douanes, grâce à la gestion irréprochable de ses activités.*

2008 - Création de MFG - Commercialisation de verre plat en Europe => *En 2013, La Filiale MFG se voit accorder le statut d'Opérateur Economique Agrée (OEA) par la Direction Générale des Douanes, en raison de sa gestion transparente, la clarté de ses opérations de dédouanement et un passif sans failles dans la gestion et le déroulement de ses activités.*

2008 - Création de COGETP - Véhicules de travaux publics VOLVO

2009 - Usine de production de produits électroniques et électroménagers de SAMSUNG Electronics - Nouvelle raffinerie de sucre de 1 million de tonnes/an - Deux centrales électriques en cogénération de 2X25 MW

2011 - **Création de SODI Automotive** pour représenter les marques de véhicules italiens: Fiat, Alfa Romeo & Lancia. SIERRA Cevital- Un joint-venture spécialisé dans le développement et la gestion des centres commerciaux.

2013 – **L'acquisition de l'entreprise OXXO** : le groupe Cevital rachète l'entreprise de menuiseries en PVC, implantée en France, dans le cadre de son développement à l'international.

2013 - **Reprise des activités de Michelin Algérie:** Le groupe Cevital a signé un partenariat commercial avec Michelin Algérie où il a racheté pas moins de 67 % du capital de la filiale française. le but à plus long terme étant de reprendre intégralement cette société d'ici fin 2013.

❖ **Les Cinq Pôles D'activités Du CEVITAL :**

Chapitre 01 : ETUDE DE L'EXISTANT

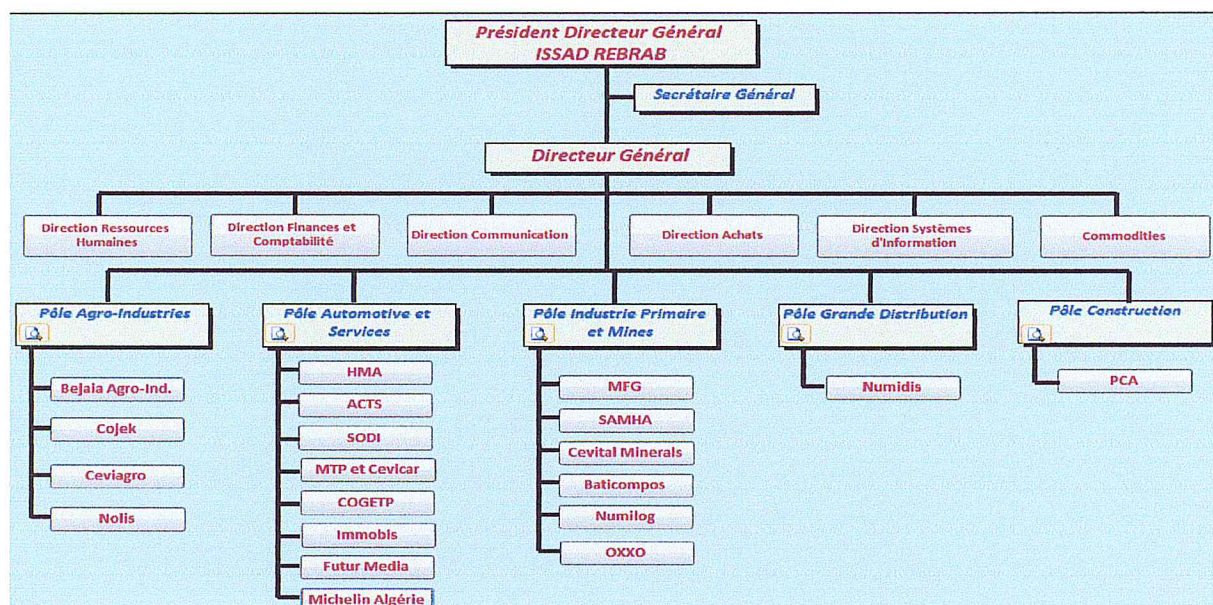


Figure 01 : Organigramme du groupe CEVITAL

❖ Présentation de NUMIDIS :

NUMIDIS est une filiale spécialisée dans la grande distribution. Elle s'est fixée comme objectif de développer une chaîne de supermarchés et d'hypermarchés sous l'enseigne Unocity et Uno pour satisfaire les attentes de ses clients d'abord à l'échelle régionale ensuite à l'échelle nationale, en leur vendant tout sous le même toit et à des prix très concurrentiels.

Son principal objectif est de répondre aux attentes des consommateurs en présentant les produits en libre-service dans le respect des règles du merchandising.

NUMIDIS est aussi l'entreprise qui incarne :

Nouveauté , **U**nité , **M**otivation , **I**nitiative, **D**éfis **I**dées , **S**uccès

❖ Historique de NUMIDIS :

16 janvier 2007: création de NUMIDIS , société par action et filiale du groupe CEVITAL.

Numidis compte plus de 1.300 collaborateurs à son actifs, avec une prévision de recrutement de plus de 700 collaborateurs en 2012.

Chapitre 01 : ETUDE DE L'EXISTANT

10 novembre 2007: ouverture du 1^{er} supermarché à ROUIBA de 500 m² de surface de vente.

10 juin 2008: ouverture du 2^{ème} supermarché à GARIDI de 2 000 m² de surface de vente

05 aout 2010: NUMIDIS s'inscrit dans l'histoire de la grande distribution en Algérie avec l'ouverture de son premier hypermarché à Bab Ezzouar de 4 950 m² de surface de vente.

13 Septembre 2011: Ouverture du second hypermarché à Bouira de 5 900 m² de surface de vente.

17 Avril 2012: Ouverture du troisième hypermarché à Ain Defla de 5 122 m² de surface de vente.

26 Juin 2012: Ouverture du quatrième hypermarché à Mostaganem de 6 150 m² de surface de vente.

01 Avril 2012: Ouverture Uno Relais- Station-service Yellel (Relizane)

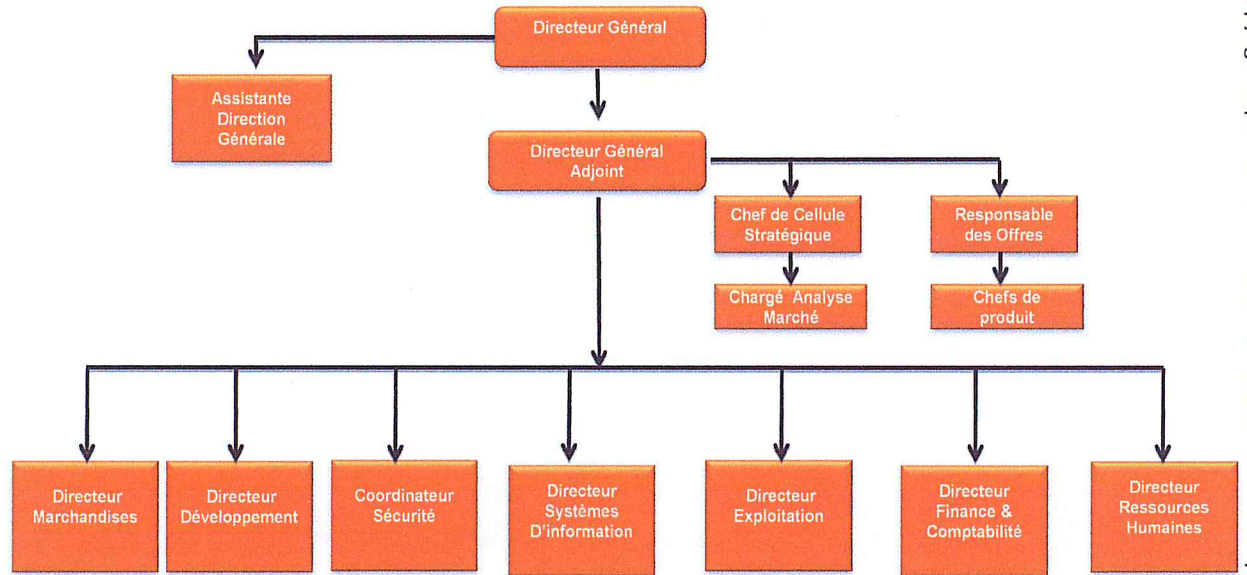
01 Juillet 2012: Ouverture Uno Relais- Station-service Ain Arnat (Setif)

01 Novembre 2012: Ouverture Uno Relais- Station-service Sidi Zitouni (Bordj Bouariridj)

11 Décembre 2012: Ouverture Uno Relais- Station-service Sidi Yakoub (Sidi Belabès)

20 Février 2013: Prévision d'Ouverture Uno Relais- Station-service Tiberkanine (Ain Defla).

Chapitre 01 : ETUDE DE L'EXISTANT



Copyright Département Ressources Humaines Numidis SPA/2011

❖ PROJETS de NUMIDIS :

Figure 02 : Organigramme de NUMIDIS

Numidis projette de cumuler à l'horizon 2012 :

- + Une surface de vente de 80 000 m².
- + Plus de 3 000 collaborateurs.
- + Ouverture de trois (3) hypermarchés par an sur l'ensemble du territoire national et devenir ainsi le N° 1 de la Grande Distribution.

2- Etude de l'existant :

Le présent système d'encaissement au niveau du Numidis concerne la version 3.0. de Winstore adapté depuis trois ans [Winstore 3.0].

2-12-1 Les fonctionnalités de Winstore :

VENTE :

- OUVERTURE CAISSE : L'ouverture de caisse se fait une fois dans la journée à l'ouverture du magasin. Elle consiste à s'identifier et à renseigner le fonds de caisse.

Chapitre 01 : ETUDE DE L'EXISTANT

- SORTIE DE CAISSE : L'objectif est de revenir au menu général de Winstore

ARTICLE :

- Rechercher un article par son libellé : Il s'agit de saisir directement le libellé de l'article.
- Rechercher un article par sa catégorie : Il s'agit de sélectionner la catégorie à laquelle appartient l'article et de descendre dans les différents niveaux de sa hiérarchie.
- Rechercher un article par la liste des articles voisins : La liste des articles voisins permet, à partir d'un article scanné en caisse, de retrouver l'ensemble des articles appartenant au modèle de l'article scanné.
- RETOUR ARTICLE : L'objectif est de reprendre un article retourné par une cliente et de choisir d'accorder à la cliente :
 - un échange
 - un remboursement (fonction «Espèces»)

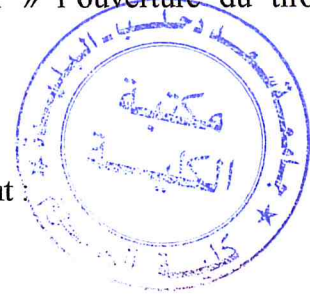
OPERATIONS SUR TICKET :

- Mettre un ticket en attente : L'objectif est de mémoriser le ticket en cours pour le reprendre ultérieurement et ainsi libérer l'écran de vente. Un seul ticket peut être mis en attente.
- Reprendre un ticket : L'objectif est de reprendre un ticket mis en attente.
- Editer une facture sur ticket : L'objectif est d'imprimer une facture à partir d'un ticket.
- Faire un retour ticket : L'objectif est d'annuler un ticket suite, par exemple, à un changement d'avis de la cliente.
- Changer le mode de paiement : L'objectif est de revenir sur un ticket enregistré pour en modifier le mode de paiement suite, par exemple, à une erreur de manipulation ou à un changement d'avis de la cliente.
- Editer un duplicata de ticket : L'objectif est d'imprimer un exemplaire du ticket.

Chapitre 01 : ETUDE DE L'EXISTANT

TIROIR-CAISSE :

- Faire une remise au coffre : L'objectif est de retirer de l'argent du tiroir-caisse. La monnaie retirée de la caisse est remise au coffre dans une enveloppe
- Faire une insertion de monnaie : L'objectif est de confirmer en caisse l'ajout de monnaie retirée du coffre.
- Ouvrir le tiroir-caisse : L'objectif est de « forcer » l'ouverture du tiroir-caisse.



MODE DE PAIEMENT : il s'agit de choisir le mode de paiement

- Paiement par espèces
- Paiement par chèque
- Paiement par chèque cadeau
- Paiement par bon d'achat

CLOTURE :

- Clôturer la caisse en fin de journée L'objectif de la clôture est de :
 - vérifier la journée de caisse
 - réaliser les remises en banque
 - éditer les états de clôture
 - et déclencher les transferts quotidiens avec la Centrale.
- Clôturer le magasin : Pour réaliser cette opération, toutes les caisses du magasin doivent avoir été clôturées.

EDITIONS :

Les éditions de caisse concernent généralement les opérations d'encaissement du jour sur l'ensemble des caisses. Elles peuvent être demandées en cours de journée via les éditions. Toutes les éditions affichent en en-tête les date et heure d'édition ainsi que l'intitulé du magasin.

- Analyse détaillée des remises.
- Journal des ventes : L'objectif est d'obtenir le détail des ventes sur le magasin ou sur une caisse.

Chapitre 01 : ETUDE DE L'EXISTANT

- Analyse par vendeur : L'objectif est d'obtenir des statistiques de vente par vendeur.
- Analyse par caissière : L'objectif est d'obtenir des statistiques de vente par caissière
- Etat des encaissements : L'objectif est de visualiser les montants encaissés par mode de règlement

OUTILS :

- CONSULTATION FICHE ARTICLE :
L'objectif est de visualiser la fiche article qui regroupe les caractéristiques de l'article : le code article, le code EAN, le code modèle, la période de promotion...
- CLIENTS :
 - Accéder à la liste clients : L'objectif est de consulter la liste clients sans entrer en encaissement et visualiser par exemple le nombre de points de la cliente.
- COFFRE :
 - REB enveloppe : L'objectif est de remettre en banque les enveloppes dans le coffre.
 - Edition du contenu : L'objectif est de consulter et d'imprimer le contenu du coffre.
 - Ouverture d'enveloppe : L'objectif est d'ouvrir une enveloppe dans le coffre pour la transformer en vrac ; le contenu reste dans le coffre.
 - Ajout de monnaie en caisse : L'objectif est de retirer de la monnaie du coffre et de l'ajouter en caisse.
 - Entrée : L'objectif est de remettre au coffre de l'argent qui ne provient pas de la caisse (exemple : monnaie remise par la banque pour alimenter le fonds de caisse)
 - Sortie : L'objectif est de sortir des espèces (en vrac ou sous enveloppe) pour une autre destination qu'une caisse ou que la banque.

PARAMETRES DU PERSONNEL :

- Créer un nouveau membre du personnel
- Modifier la fiche d'un membre du personnel
- Supprimer la fiche d'un membre du personnel

2-2 Diagnostique :

Cette étude nous a permis de faire ressortir les insuffisances énumérées comme suit:

1. Les rapports (chiffre d'affaire, nombre de pause) et l'historique des ventes sont stockés dans la même base de données dans des tables apparts ce qui mène à la saturation de la base de données.
2. absence de la gestion de fidélité.
3. absence de la gestion des remises.

2-3 Suggestion :

Afin de répondre aux carences citées précédemment, nous proposons la série des suggestions suivantes :

- Concevoir un système d'encaissement qui comporte tous les fonctionnalités cité en dessus et aussi implémenter la gestion de la fidélité, gestion des remises.
- Concevoir une nouvelle base de données afin de garder l'historique des ventes.
- Améliorer la qualité des rapports, en introduisant les graphes par exemple.

Conclusion

Globalement, il a été présenté dans cette étude de ce premier chapitre une brève étude de ce qui fera par la suite une base informationnelle pour la conception et la mise en œuvre de notre futur application. Le deuxième chapitre de ce mémoire concerne la conception de notre système.

Chapitre 02 :

CONCEPTION

1-Démarche adoptée :

2-1 La démarche adoptée :

La conception et la réalisation d'un projet s'appuie sur une approche ou démarche à suivre, qui doit être choisie minutieusement, afin d'assurer le succès du projet. Or, le choix est très difficile car il existe plusieurs approches, et la décision d'adopter telle ou telle approche doit se faire selon plusieurs critères concernant le projet, comme le type, la nature, les buts ... etc.

Pour la conception et la réalisation du système d'encaissement ; la démarche adoptée, appelée autrement le **processus de développement** qui répond aux besoins du projet sera **2TUP « TwoTrackUnifiedProcess »**

Le choix de ce processus est basé sur les avantages qu'il offre à notre système par rapport aux autres processus (voir annexe 1), en effet, le processus 2TUP est caractérisé par les points suivants [Roque 2002]:

- 1- C'est un processus UP¹ qui suit deux chemins (fonctionnel et technique).
- 2- C'est un processus itératif qui permet la compréhension progressive du problème à travers des raffinements successifs.
- 3- C'est un processus piloté par les risques qui représentent les causes majeures d'échec d'un projet logiciel et qu'il faut écarter en priorité.
- 4- C'est un processus orienté par les besoins d'utilisateurs représentés par les cas d'utilisation.
- 5- C'est un processus centré sur l'architecture donc il permet le découpage en module afin de faciliter la maintenance.

L'approche 2TUP consiste à fusionner les résultats obtenus des deux branches (fonctionnel et technique) pour donner un processus de développement en forme de Y comme indique la figure suivante :

¹UnifiedProcess : le processus unifié est un processus de développement logiciel construit sur UML ; il est itératif, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques. Pré-études (inception), élaboration, construction, et le déploiement. [Roque 2002]

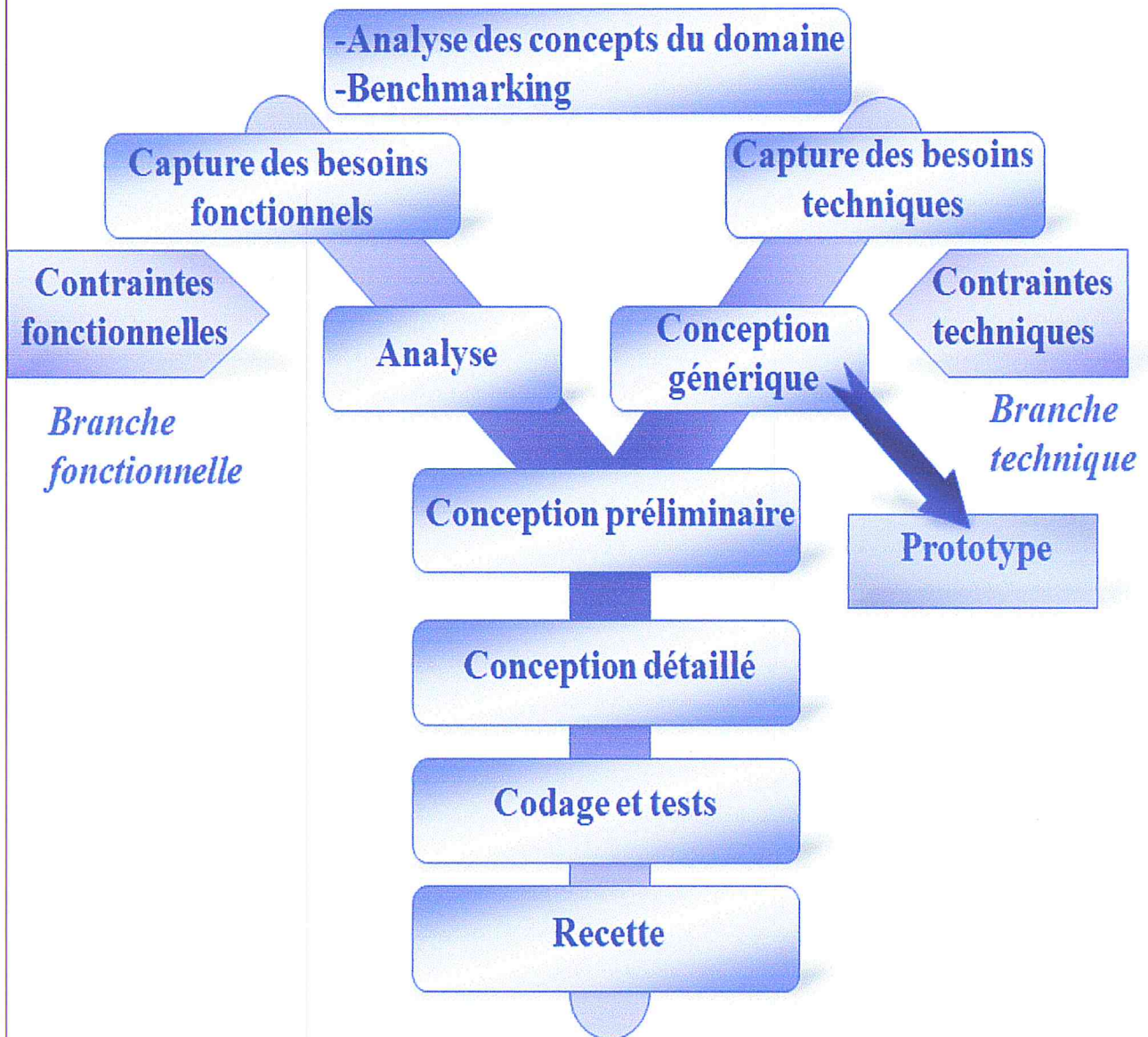


Figure 3: Le processus de développement en Y adapté

2- Langage de modélisation UML :

Pour faire face à la complexité croissante des systèmes d'information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années réside dans la programmation orientée objet (P.O.O.).[Roque 2002]

Face à ce nouveau mode de programmation, les méthodes de modélisation classique telle que MERISE ont rapidement montré certaines limites et ont dû s'adapter

(cf.MERISE/2). De très nombreuses méthodes ont également vu le jour comme Booch, OMT ... Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu UML (Unified Modelling Language « langage de modélisation objet unifié »), issu de la fusion des méthodes Booch, OMT (Object Modelling Technique) et OOSE (Object Oriented Software Engineering).

UML nous servent à formaliser nos besoins, c'est-à-dire à les représenter sous une forme graphique suffisamment simple pour être compréhensible par toutes les personnes impliquées dans le projet. N'oublions pas que bien souvent, le maître d'ouvrage et les utilisateurs ne sont pas des informaticiens. Ils leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes UML, Ils permettent de recenser les grandes fonctionnalités d'un système.

3- Capture des besoins fonctionnels :

Tout d'abord, nous entamons notre travail de recueil des besoins par les aspects fonctionnels, par la suite, nous la compléterons par le recueil des besoins technique.

Cette étape consiste à identifier ce que notre système va offrir aux utilisateurs.

3-1 Identification des acteurs :

- **Qu'est-ce qu'un acteur ?**

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié [Roques 2002].

Le système d'encaissement est conçu pour quelques employés du magasin, les acteurs seront classés de la manière suivante :

| Acteur | Rôle |
|---------------------|------------------------------------|
| Responsable magasin | du Gestion du magasin |
| Arrière caissier | Gestion des caisses (backoffice) |
| Caissier | Gestion de la caisse (Frontoffice) |

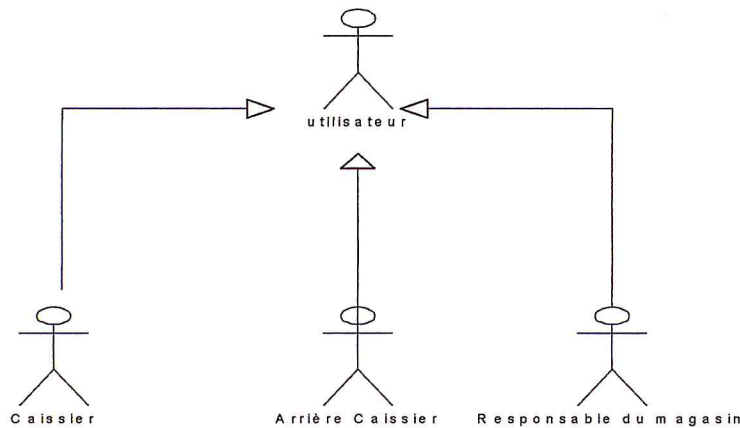


Figure 04 : Diagramme des acteurs

3-2 Identification des cas d'utilisation fonctionnels:

| N° du cas d'utilisation | Cas d'utilisation | Acteur | Description du cas d'utilisation |
|-------------------------|------------------------|---------------------------------------|---|
| 01 | Gestion des articles | Caissier Et Arrière Caissier | -Le caissier ajoute , annule et recherche un articles - l'arrière caissier crée , modifie supprime, et retourne un article |
| 02 | Gestions de la caisse | Caissier et Arrière Caissier | -le caissier met la caisse en pause -l'arrière caissier ouvre et ferme |
| 03 | Gestion de tiroir | Arrière caissier | -l'arrière caissier ajoute la monnaie au tiroir, prélève la monnaie du tiroir pour le remettre au coffre |
| 04 | Gestion des tickets | Caissier et arrière caissier | - Le caissier met en attente un ticket , reprend un ticket , paye un ticket , édite un duplicata d'un ticket, associer un ticket a une carte de fidélité. - l'arrière caissier retourne un |
| 05 | Gestion du magasin | Responsable de magasin | -Le responsable du magasin clôture le magasin |
| 06 | Gestion de la fidélité | Arrière caissier | L'arrière caissier crée une carte de fidélité. Le caissier consulte le nombre de points |
| 07 | Gestion des remises | Arrière Caissier | L'arrière caissier crée, modifie, supprime une remise |

Chapitre 02 : Conception

Cas d'utilisation N°01 : Gestion des articles

| Cas d'utilisation | Description |
|----------------------------------|--|
| Ajouter un article | Ajouter un article dans un ticket après l'avoir scanner |
| Annuler un article | Enlever un article d'un ticket non encore payé |
| Associer un article à un vendeur | Associer l'article vendu à un vendeur afin qu'on puisse faire des analyses de nombre de vente de chaque vendeur. |
| Rechercher un article | Rechercher soit par son code soit par sa famille... |
| Créer un article | D'ajouter un nouvel article dans la base de données. |
| Modifier un article | Modifier les informations de l'article (prix, nom ...) |
| Supprimer un article | Supprimer un article de la base de données |

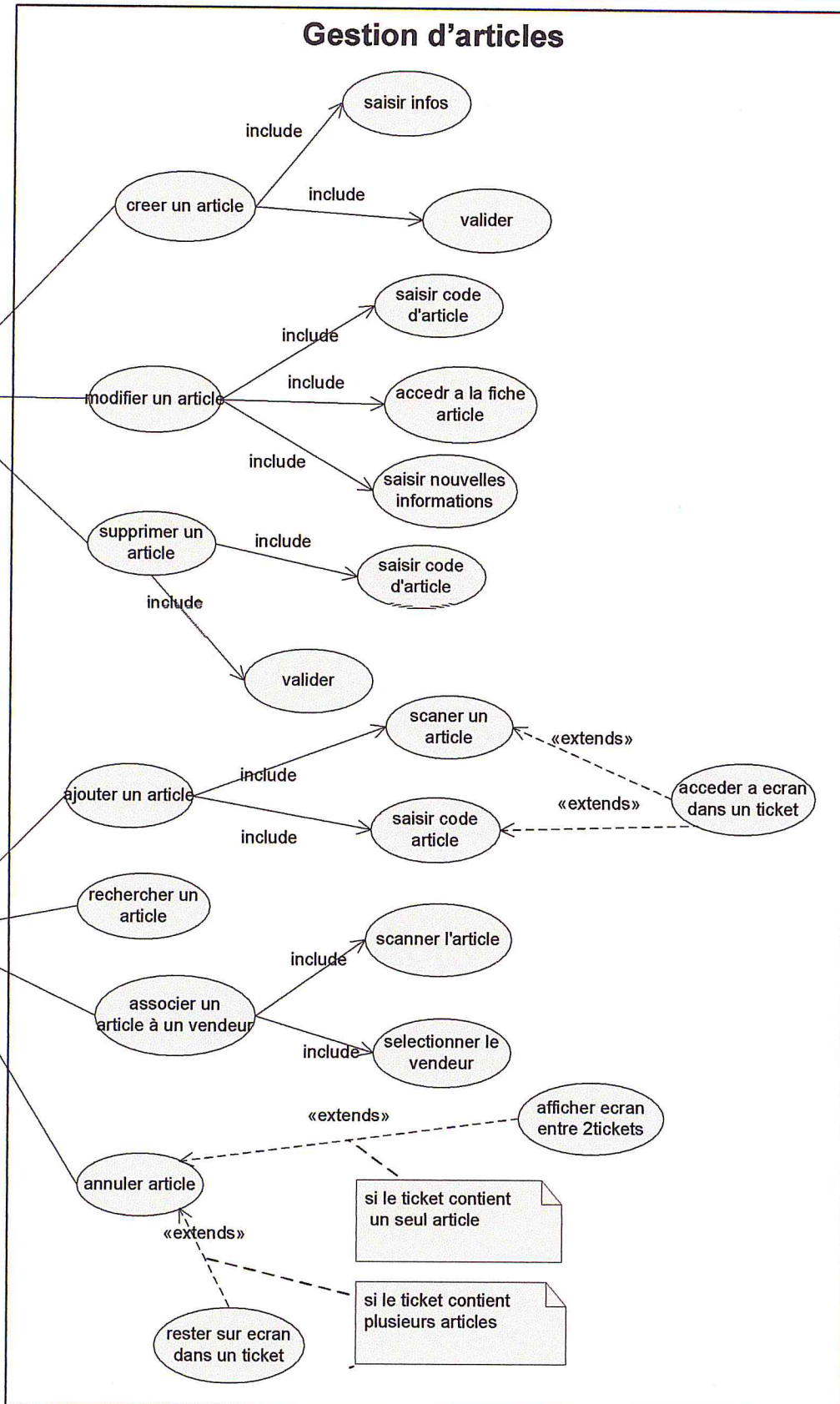


Figure 05 : Diagramme du cas d'utilisation « Gestion d'articles »

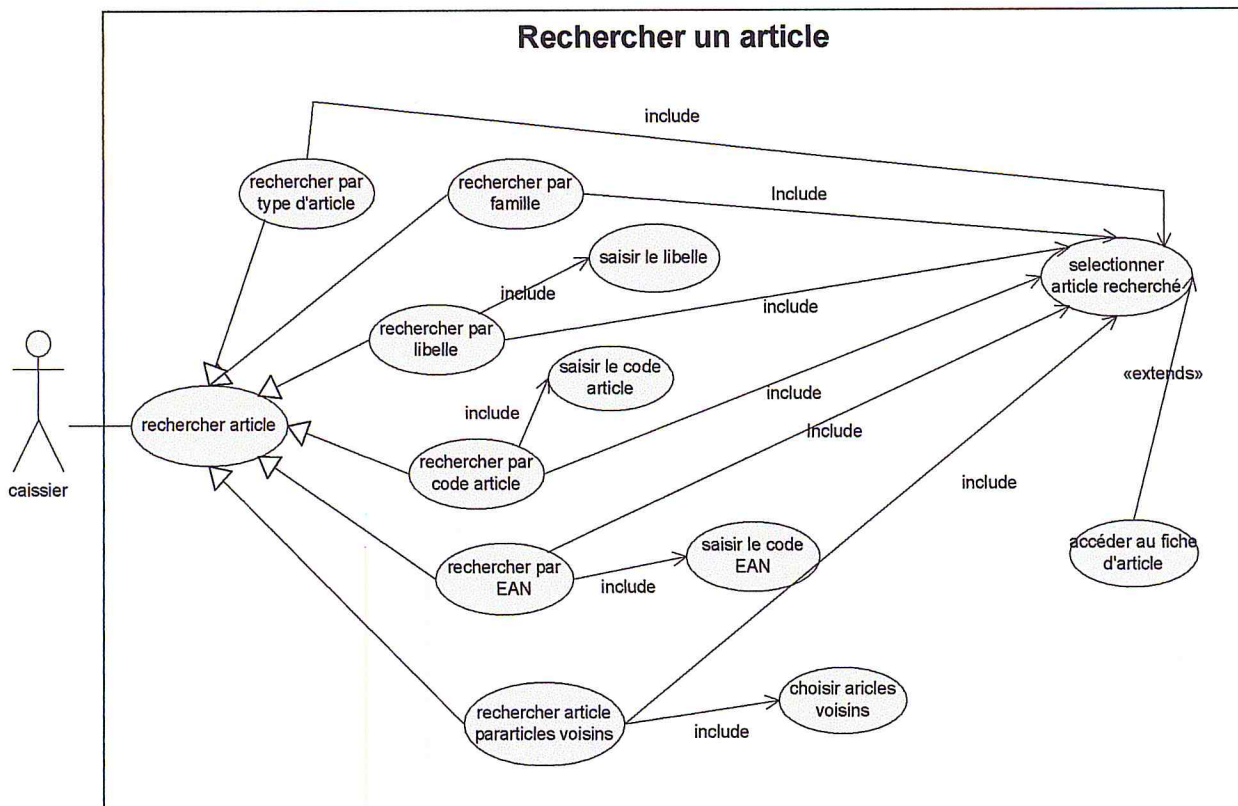


Figure 06 : Diagramme du cas d'utilisation « rechercher un article »

Cas d'utilisation N°02 : Gestions de la caisse

| Cas d'utilisation | Description |
|---------------------------|---|
| Mettre la caisse en pause | Quitter la page d'encaissement sans fermer l'application. |
| Ouvrir la caisse | Ouvrir la caisse et ajouter son fond. |
| Fermer la caisse | Fermer la caisse et de faire la remise au coffre. |

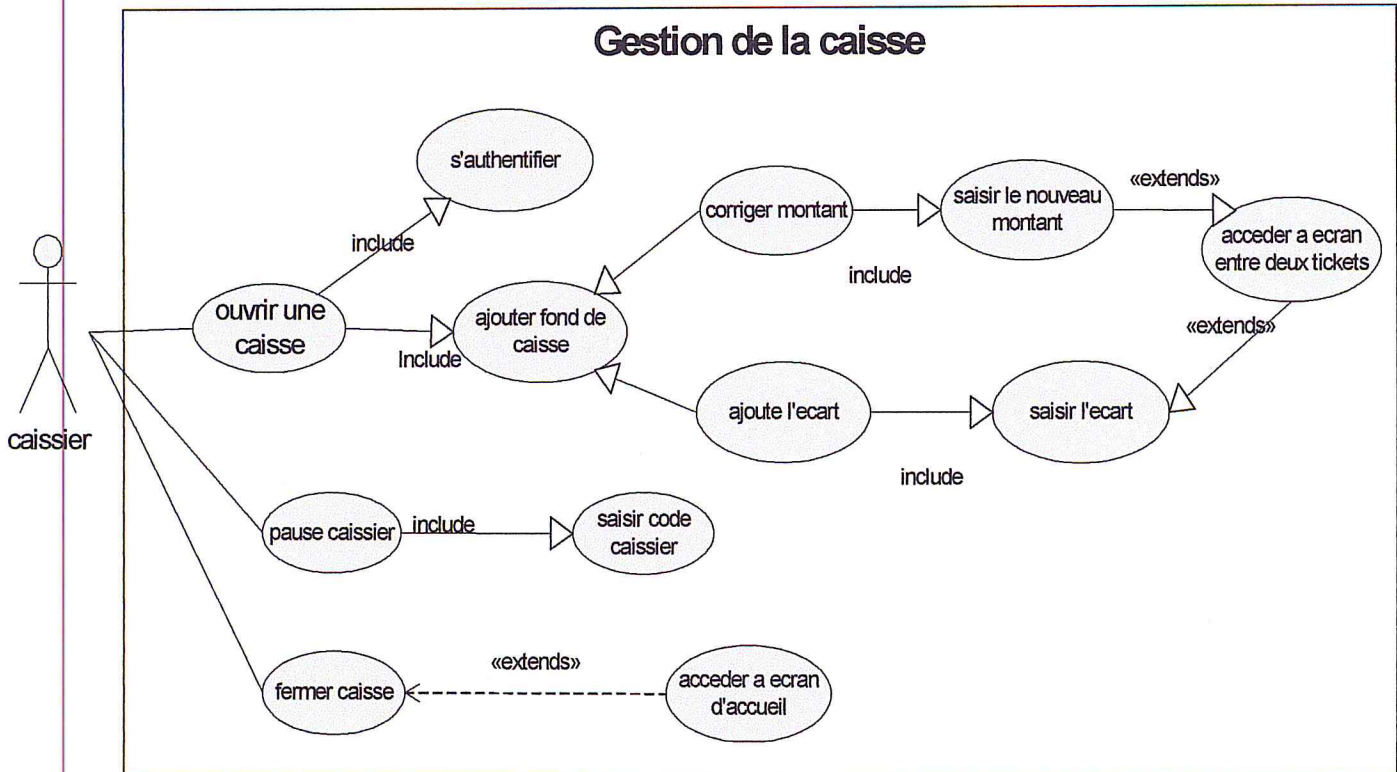


Figure 07 : Diagramme du cas d'utilisation « gestion de la caisse »

Cas d'utilisation N°03 : Gestion des tickets

| Cas d'utilisation | Description |
|--|--|
| Mettre un ticket en attente | Laisser un ticket en attente pour des raisons et prendre un nouveau ticket pour ne que le client suivant ne s'ennuie pas |
| Reprendre un ticket | Reprendre un ticket qui a été mis en attente |
| Retourner un ticket | Retourner un ou plusieurs articles ou tous les articles du ticket au magasin. |
| Payer un ticket | Choisir le type de paiement et suivre la procédure par rapport à chaque type de paiement. |
| Editer un duplicata d'un ticket | Imprimer un duplicata d'un ticket après avoir scanné son code barre. |
| Associer un ticket a une carte de fidélité | Affecter un nombre de point au client qui a une carte de fidélité. |

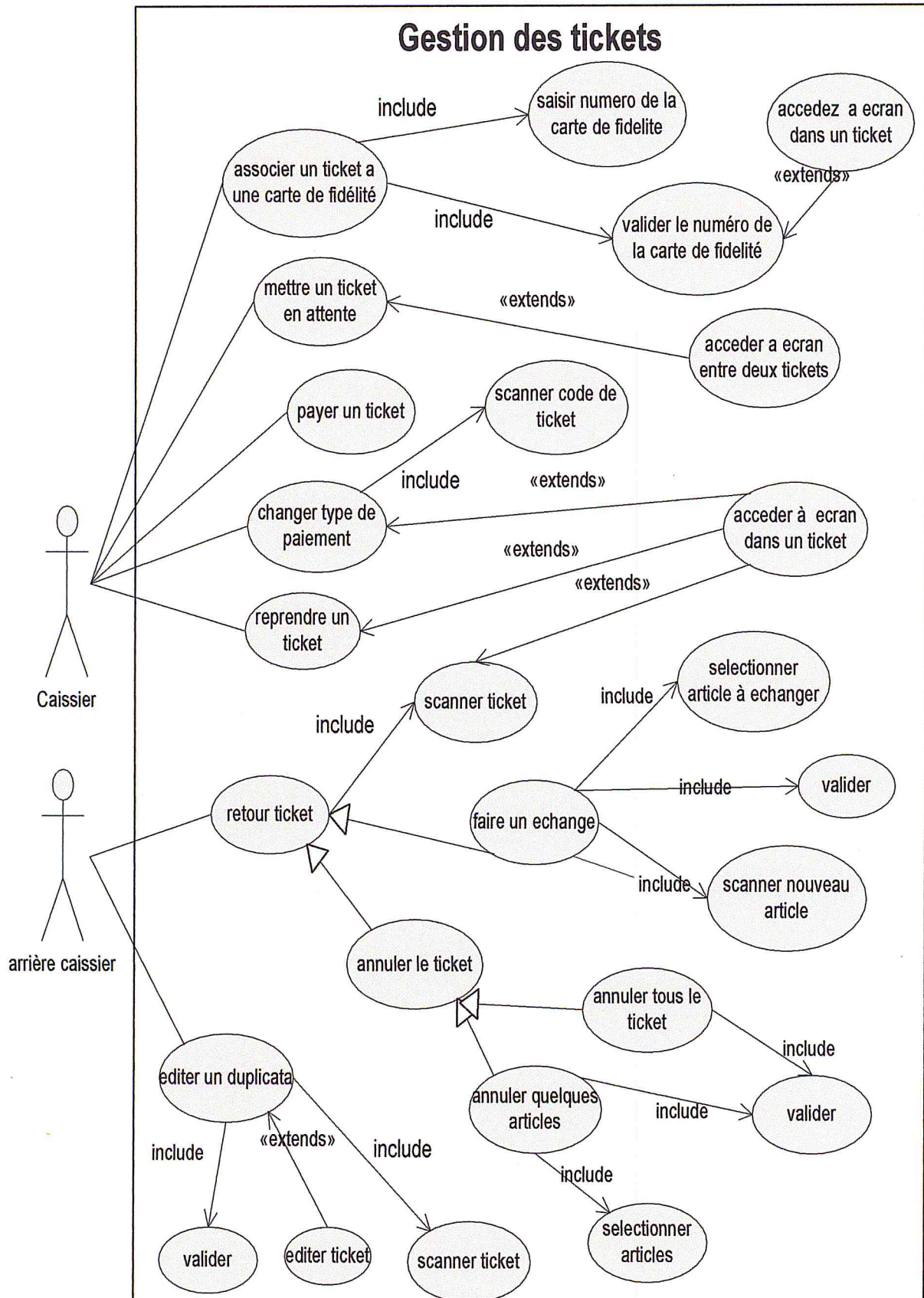


Figure 08 : Diagramme du cas d'utilisation « Gestion des tickets »

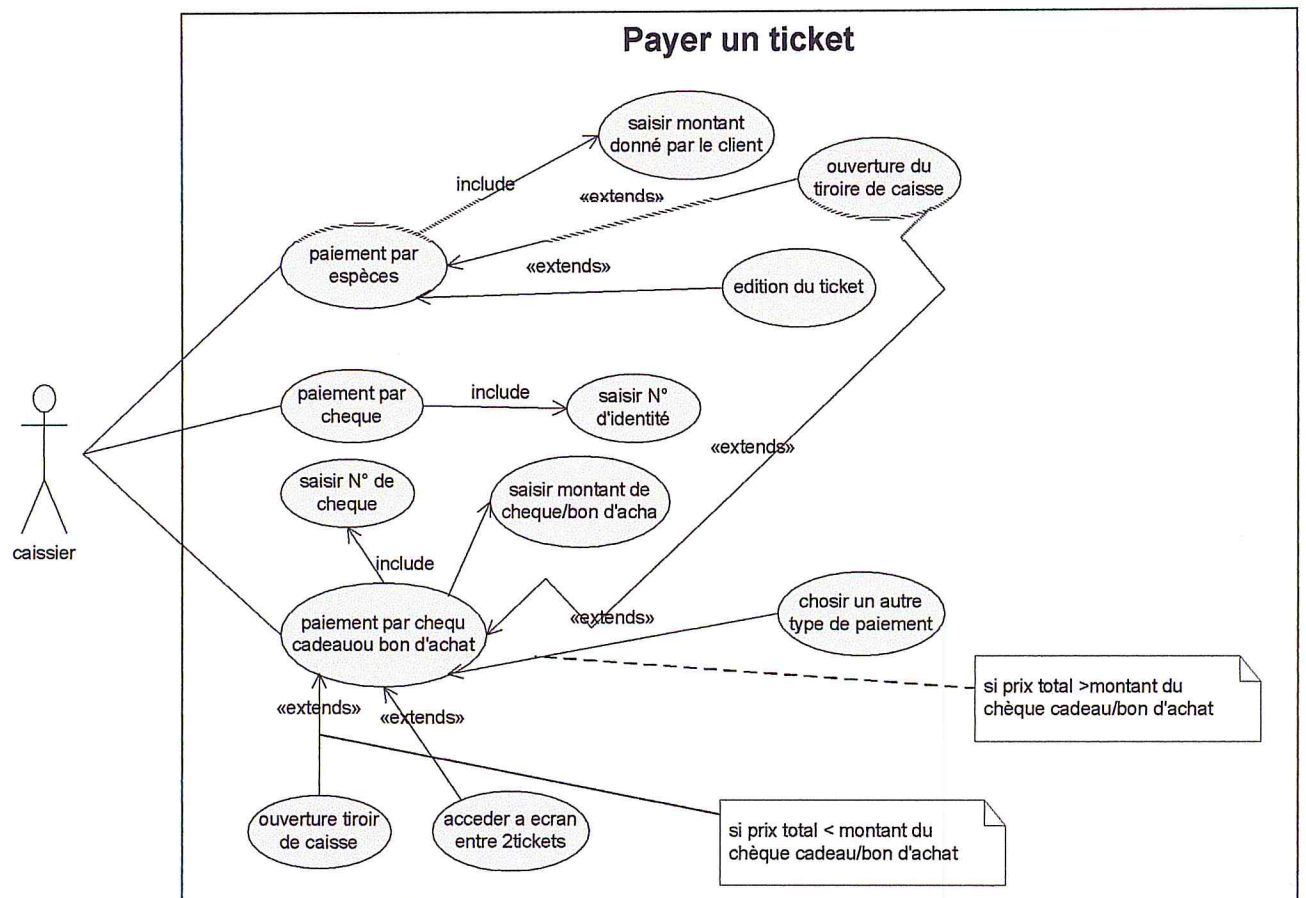


Figure 09: Diagramme du cas d'utilisation « payer un ticket »

Cas d'utilisation N°04 : Gestion de tiroir

| Cas d'utilisation | Description |
|---------------------|---|
| Ajouter la monnaie | Ajouter la monnaie au fond de la caisse |
| Prélever la monnaie | Prélever la monnaie de la caisse et la remettre au coffre |

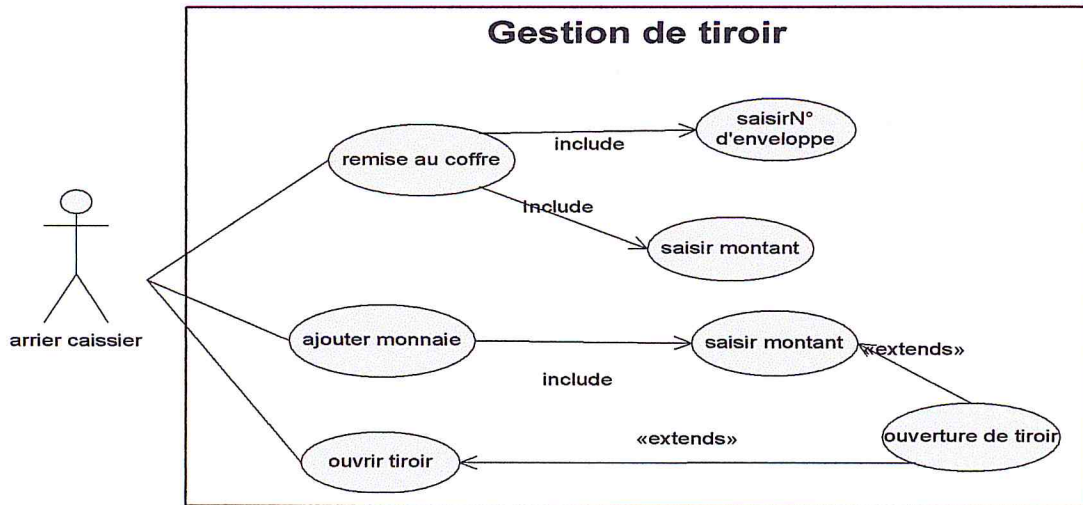


Figure 10 : Diagramme du cas d'utilisation « gestion de tiroir »

Cas d'utilisation N°05 : Gestion du magasin

| Cas d'utilisation | Description |
|-------------------|---|
| Fermer le magasin | Clôturer le magasin et remettre les enveloppes en banque. |

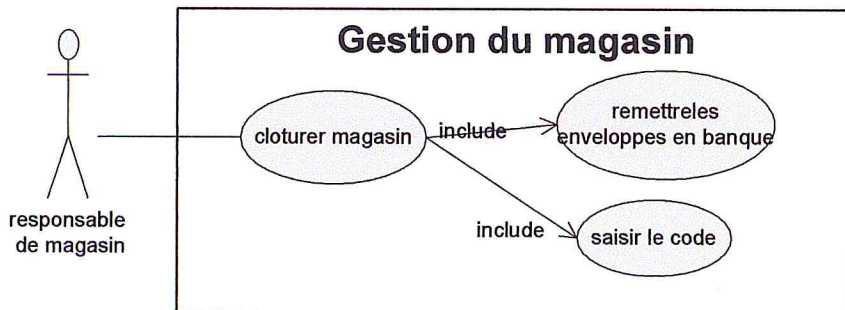


Figure 11: Diagramme du cas d'utilisation « gestion du magasin »

Cas d'utilisation N°06 : Gestion de fidélité

| Cas d'utilisation | Description |
|--|--|
| Créer une carte de fidélité | Créer une carte de fidélité et l'associer à un client. |
| Consulter nombre de points de fidélité | Accéder au compte dans l'espace fidélité après avoir saisi le numéro de la carte afin de consulter le nombre de points acquis par le client. |

Chapitre 02 : Conception

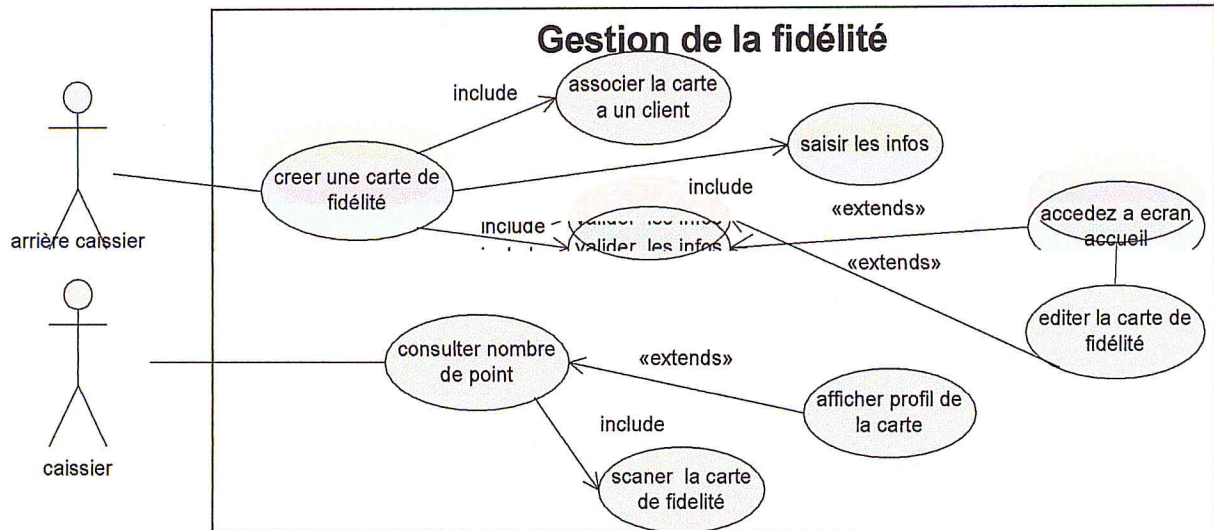


Figure 12 : Diagramme du cas d'utilisation « gestion de la fidélité »

Cas d'utilisation N°07 : Gestion des remises

| Cas d'utilisation | Description |
|----------------------|--|
| Créer une remise | créer une remise de prix pour un ou plusieurs articles et une durée précise. |
| Modifier une remise | modifier les paramètres de la remise (articles, délai, prix) |
| Supprimer une remise | supprimer une remise et remettre les prix originaux des articles. |

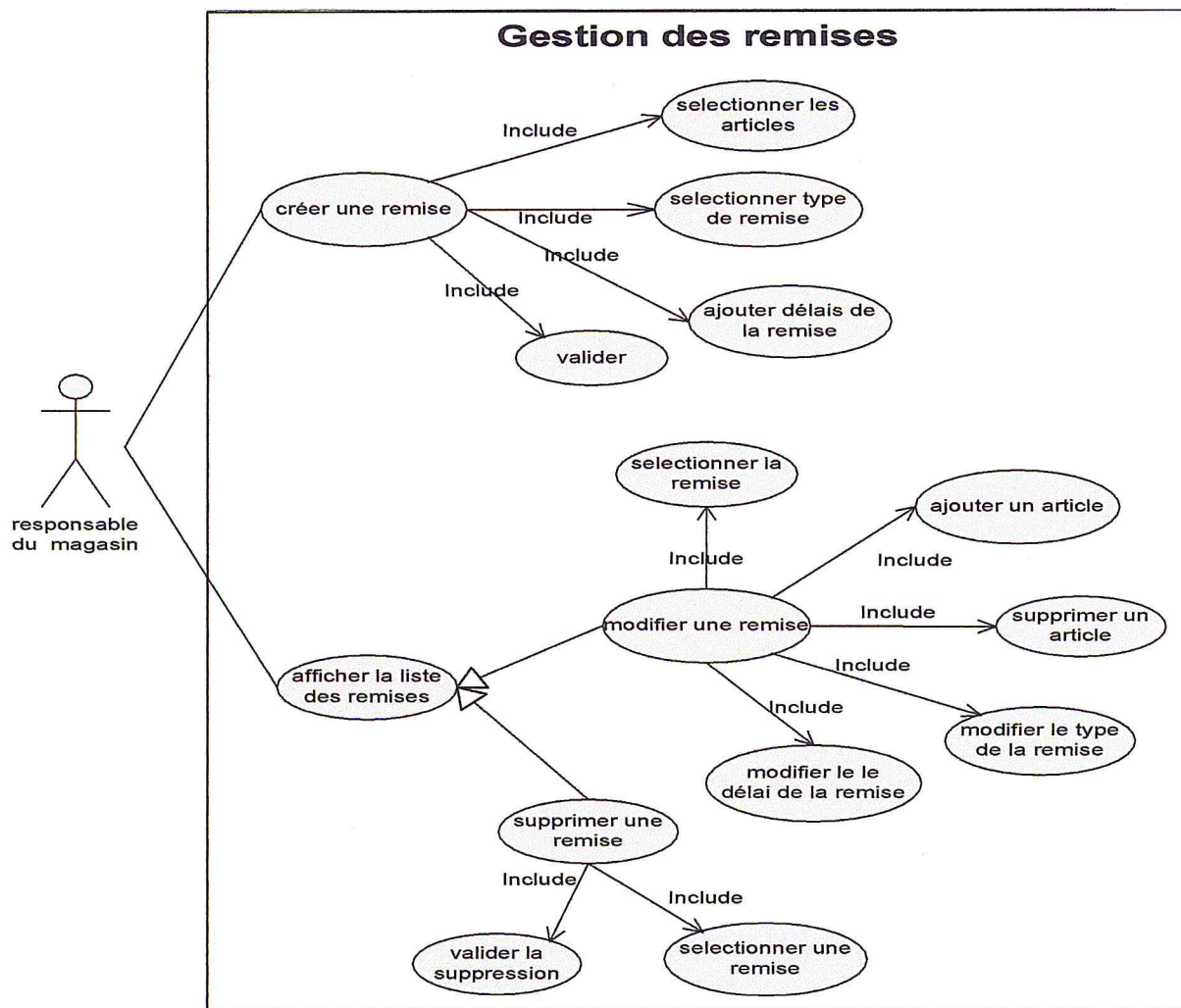


Figure 13: Diagramme du cas d'utilisation « gestion des remises »

A la fin de cette étape, nous sommes arrivés à identifier le côté fonctionnel du système, l'étape suivante concerne l'identification des besoins techniques du système.

4- Capture des besoins techniques

La capture des besoins technique couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification logicielle concerne donc les contraintes techniques [Roques 2002].

4-1 Identification des cas d'utilisation technique :

| N° du cas d'utilisation | Cas d'utilisation | Acteur | Description du cas d'utilisation |
|-------------------------|------------------------------|------------------------|---|
| 01 | Gestion des employés | Responsable de magasin | -Le responsable de magasin ajoute, supprime, rechercher et modifie le profil des employés. |
| 02 | Gestion des niveaux | Arrière caissier | L'arrière caissier crée, supprime et modifie les informations d'un niveau. |
| 03 | Gestion de modes de paiement | Arrière caissier | -l'arrière caissier ajoute et supprime le paramètres des modes de paiement. |
| 04 | Gestion des clients | Arrière caissier | -l'arrière caissier crée, recherche et supprime un client, consulte et modifie le profil d'un client. |
| 05 | Gestion des ventes | Responsable de magasin | -le responsable de magasin consulte l'historique des ventes, tickets, chiffres d'affaires, nombre de pause de caissier... |

Chapitre 02 : Conception

Cas d'utilisation N°01 : Gestion des employés

| Cas d'utilisation | Description |
|------------------------------|--|
| Ajouter un employé | Ajouter un employé dans la base de données |
| Supprimer un employé | Supprimer un employé de la base de données |
| Modifier profil d'un employé | Modifier les informations personnelles de l'employés (adresse, numéro de téléphone, salaire..) |
| Rechercher un employé | Rechercher un employé par son nom , code, poste... |

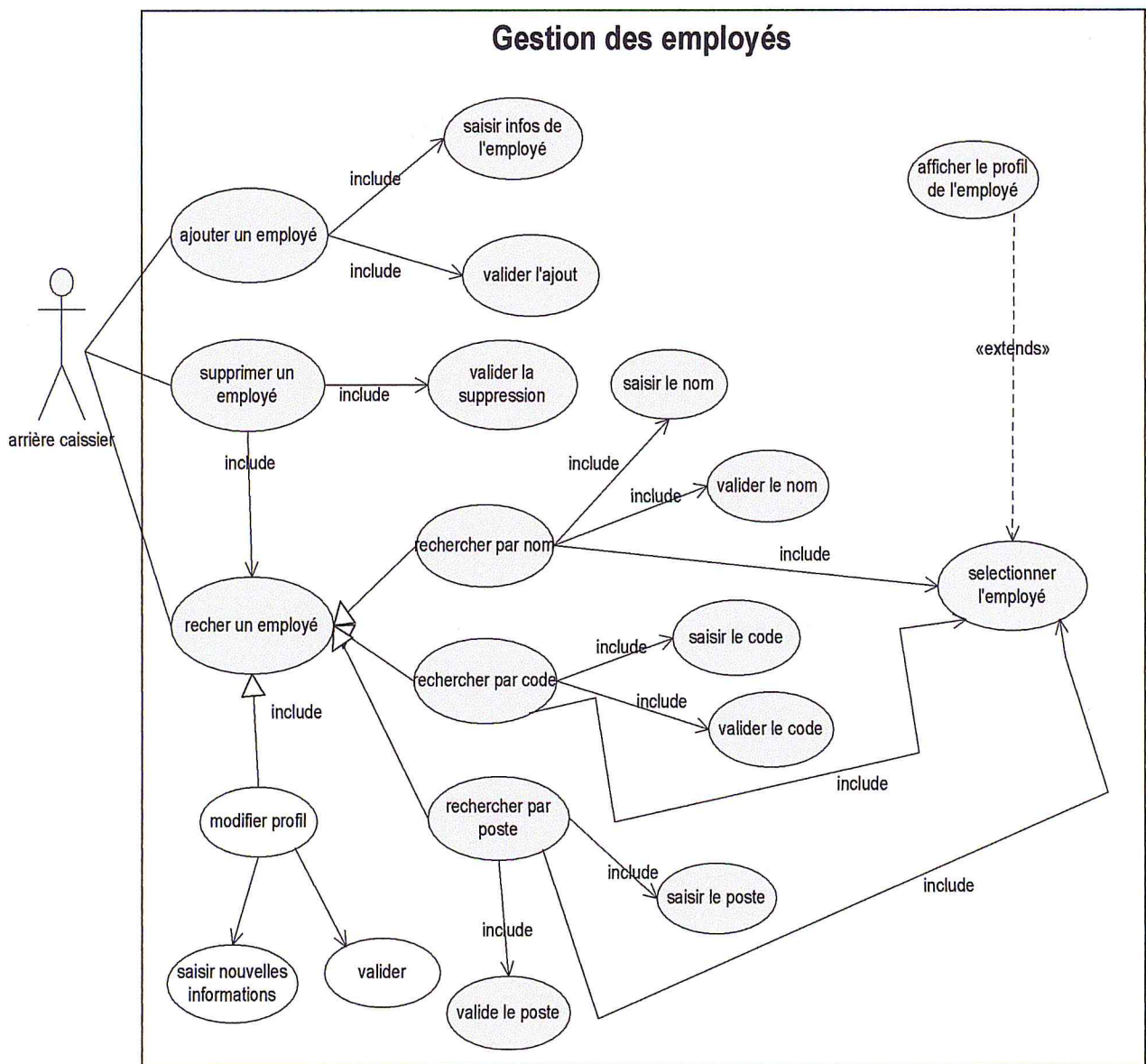


Figure 14 : Diagramme du cas d'utilisation « gestion des employés»

Chapitre 02 : Conception

Cas d'utilisation N°02 : Gestion des niveaux (rayon)

| Cas d'utilisation | Description |
|-------------------|--|
| Ajouter un niveau | Ajouter un nouveau rayon |
| Supprimer rayon | Supprimer un rayon donné |
| Modifier un rayon | Modifier les informations d'un rayon (code, articles) |

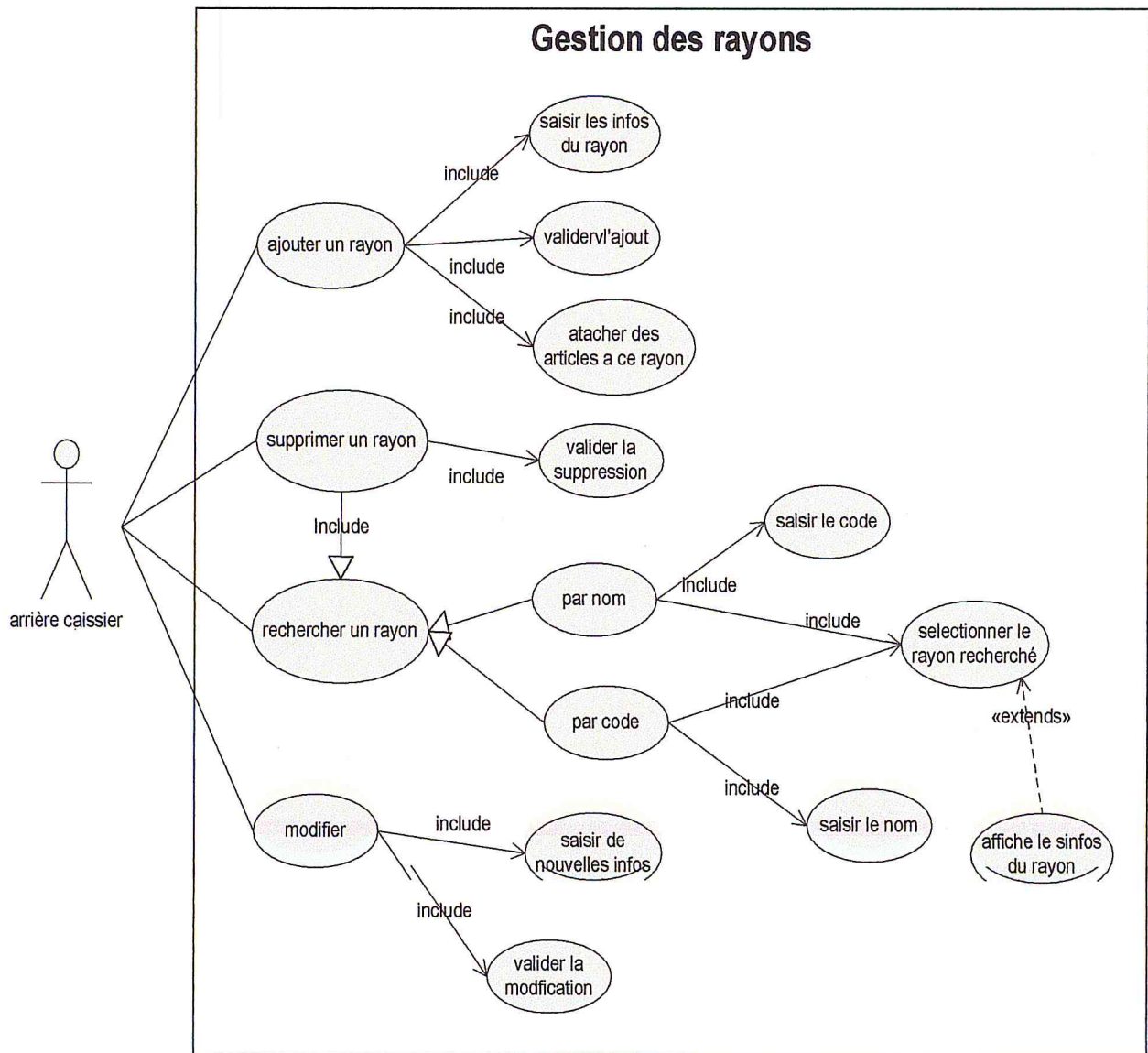


Figure 15 : Diagramme du cas d'utilisation « gestion des rayons»

Cas d'utilisation N°04 : Gestion de modes de paiements

| Cas d'utilisation | Description |
|-------------------------------|---|
| Ajouter un mode de paiement | il s'agit d'ajouter un mode de paiement |
| Supprimer un mode de paiement | Il s'agit de supprimer mode de paiement |

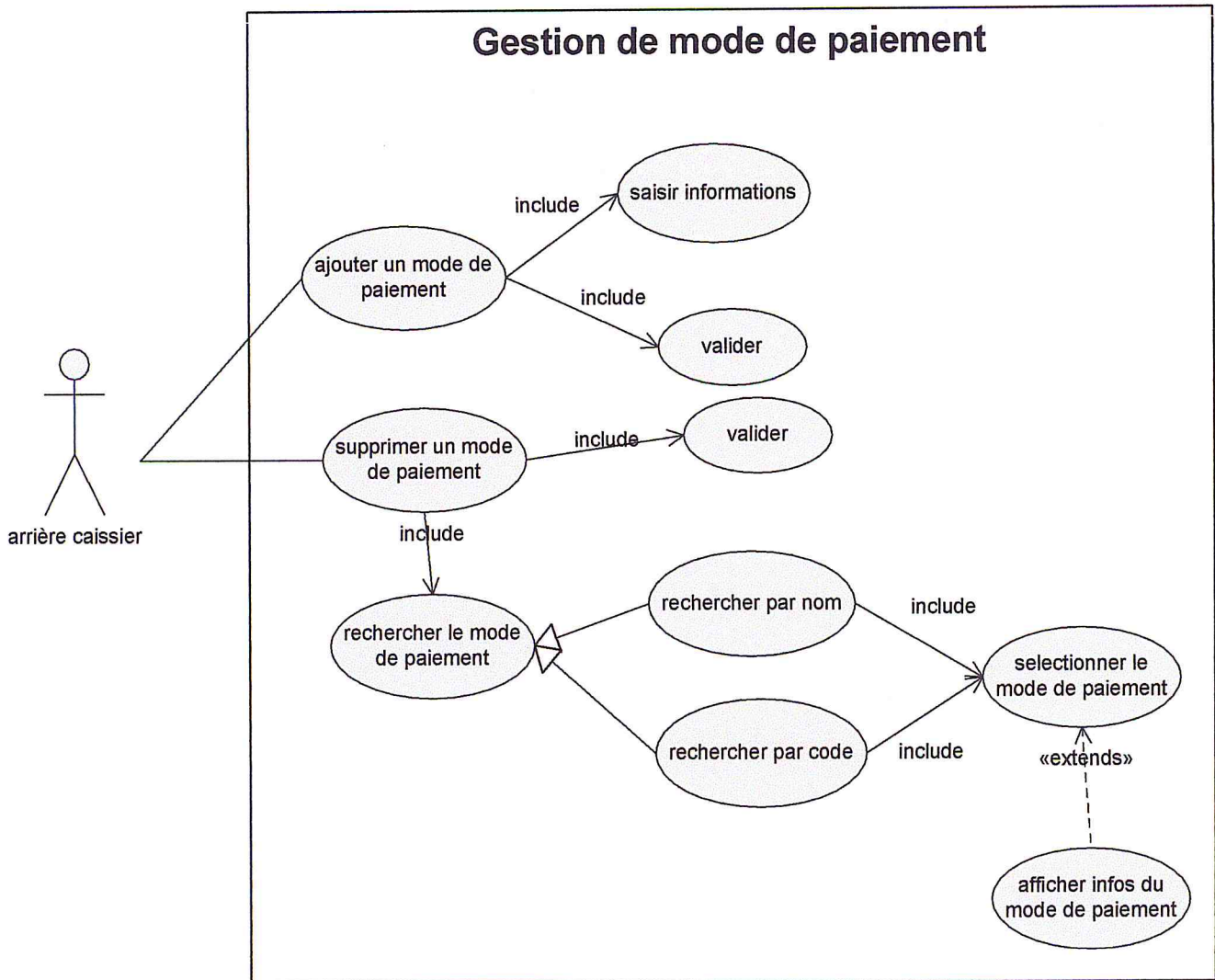


Figure 16 : Diagramme du cas d'utilisation « gestion des modes de paiement »

Cas d'utilisation N°05: Gestion des clients

| Cas d'utilisation | Description |
|------------------------------|--|
| Ajouter un client | Ajouter un client (remplir sa fiche personnelle : nom prénom, N° d'identité, type de client...). |
| Consulter profil d'un client | Accéder à ses informations personnelles |
| Modifier profil d'un client | Modifier profil d'un client (type, adresse, ...) |

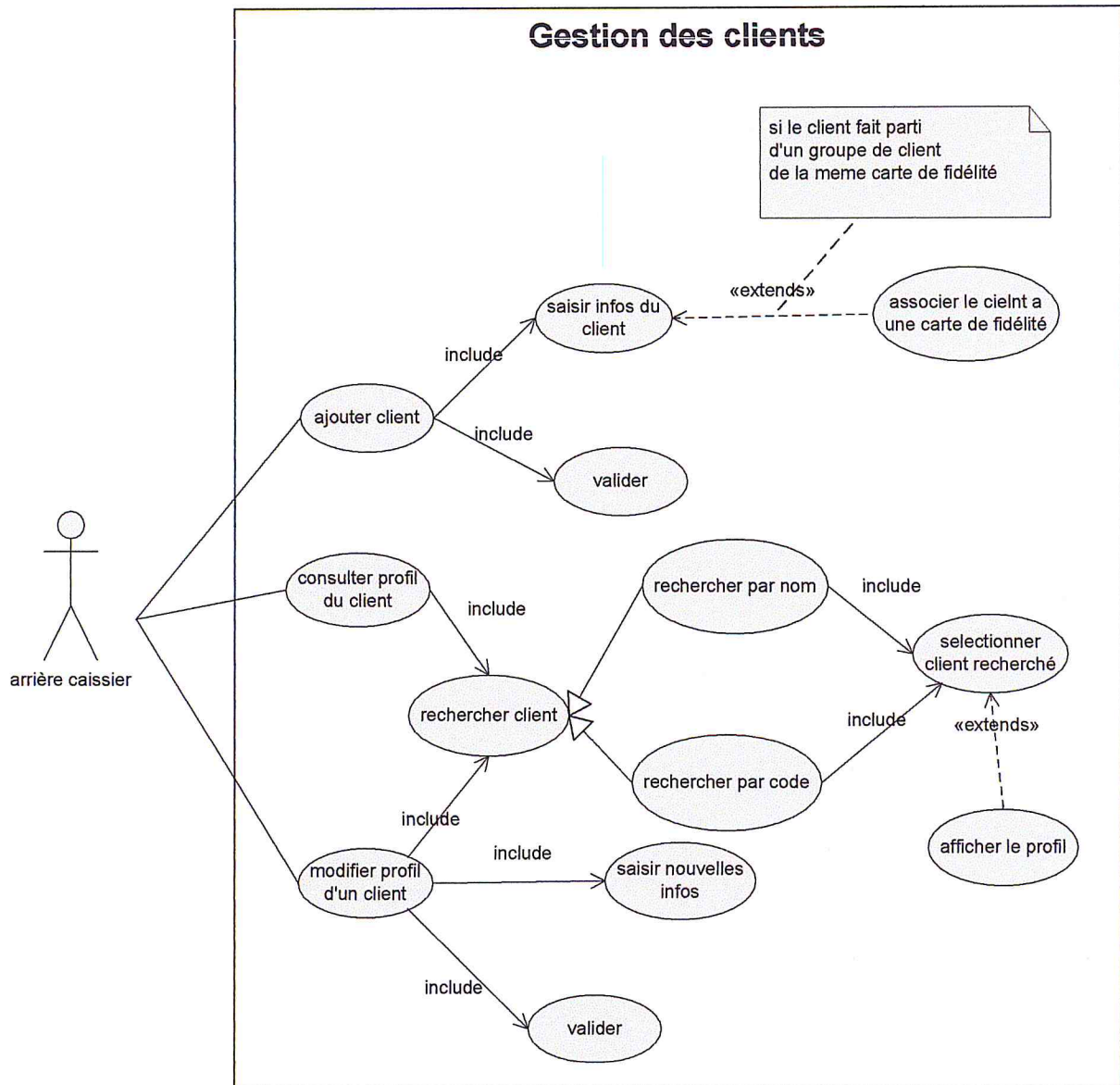


Figure 17 : Diagramme du cas d'utilisation « gestion des clients»

Cas d'utilisation N°05: Gestion des ventes

| Cas d'utilisation | Description |
|------------------------|--|
| Consulter le reporting | Consulter l'historique des ventes, tickets, chiffres d'affaires , indicateurs de performance ... |

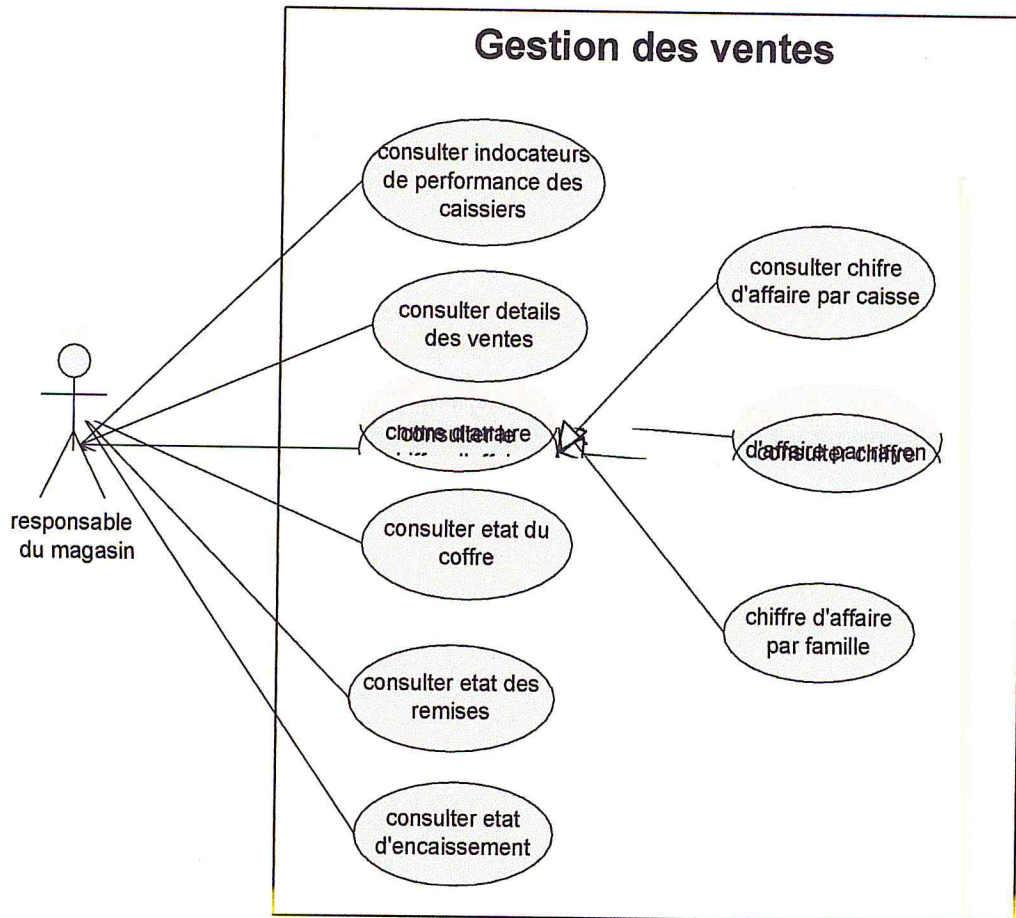


Figure 18 : Diagramme du cas d'utilisation « gestion des ventes »

A la fin de cette partie, nous sommes arrivés à déterminer les besoins techniques du système, et en addition à la partie « capture des besoins fonctionnels », nous pouvons faire une introduction à la partie analyse.

5- Analyse

5-1 Conception du modèle statique « diagramme de classe » :

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités utilisées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet, ou au mieux les modules du langage de développement.

5-1-1 Diagramme de classe la base de données centrale :

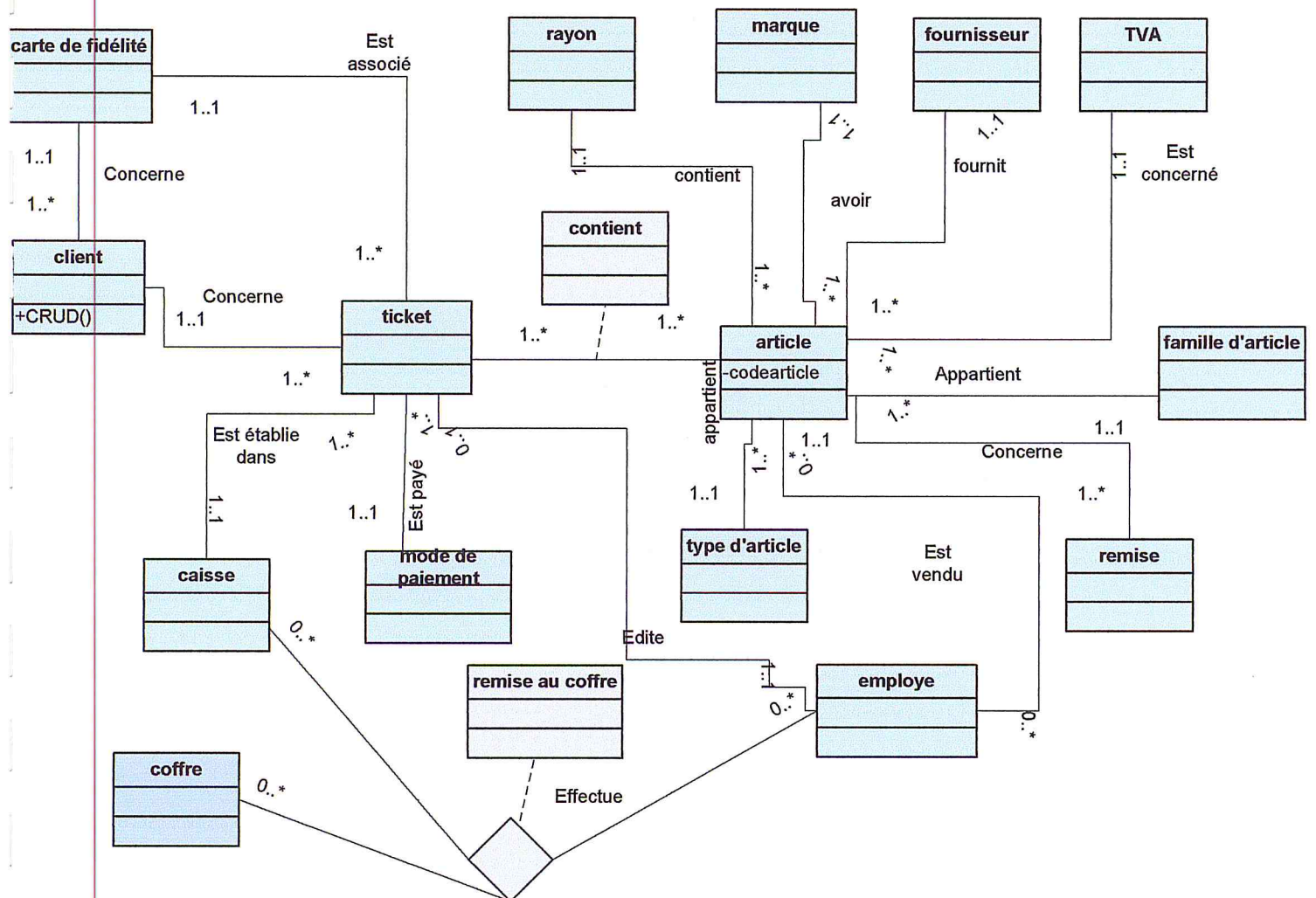


Figure 19: «Diagramme de classe de la base de données centrale»

Chapitre 02 : Conception

5-1-2 Description des classes :

| Article | |
|-------------|---|
| Attributs | Désignation |
| Code_art | Code article interne |
| Code_EAN | Code barre de l'article |
| Lib_art | Libelle de l'article |
| Desc_art | Description de l'article |
| Prix_achat | Prix d'achat de l'article |
| Prix_vente | Prix de vente de l'article |
| Quant_art | La quantité de l'article dans le stock. |
| Date_fab | Date de fabrication de l'article |
| Date_peremp | Date de péremption de l'article |
| Methodes | <ul style="list-style-type: none">- CRUD()- Annuler un article ()- Associer un article a un vendeur() |

| Rayon | |
|-----------|--|
| Attributs | Désignation |
| Code_ray | Code du rayon |
| Nom_ray | Nom du rayon |
| Méthodes | <ul style="list-style-type: none">- CRUD() |

| Marque | |
|-----------|--|
| Attributs | Désignation |
| Code_marq | Code de la marque |
| Nom_marq | Nom de la marque |
| Méthodes | <ul style="list-style-type: none">- CRUD() |

| TVA | |
|-----------|--|
| Attributs | Désignation |
| Code_tva | Code de la TVA |
| Type_tva | Type de la TVA |
| Méthodes | <ul style="list-style-type: none">- CRUD() |

| Fournisseur | |
|-------------|--|
| Attributs | Désignation |
| Code_fourn | Code fournisseur |
| Nom_fourn | Nom du fournisseur |
| Adr_fourn | Adresse du fournisseur |
| Num_fourn | Numéro du fournisseur |
| Methods | <ul style="list-style-type: none">- CRUD() |

Chapitre 02 : Conception

| Famille d'article | |
|-------------------|----------------------------|
| Attributs | Désignation |
| Code_fam | Code de famille d'articles |
| Nom_fam | Nom de famille d'articles |
| Méthodes | - CRUD() |

| Ticket | |
|--------------|---|
| Attributs | Désignation |
| Id_tick | Code du ticket |
| Montant_tick | Montant du ticket |
| Date_tick | Date du ticket |
| Heure_tick | Heure du ticket |
| Methodes | <ul style="list-style-type: none"> - CRUD() - Annuler un ticket - Retourner un ticket() - Payer un ticket() - Associer un ticket a une carte de fidélité() - Editer un duplicata de ticket(). |

| Mode de paiement | |
|------------------|--------------------------|
| Attributs | Désignation |
| Id_MDP | Code du mode de paiement |
| Type_MDP | Type du mode de paiement |
| Méthodes | - CRUD(). |

| Type d'article | |
|----------------|------------------------|
| Attributs | Désignation |
| Id_type_art | Code du type d'article |
| Type_article | Type d'article |
| Méthodes | - CRUD() |

| Client | |
|------------------|-------------------------------|
| Attributs | Désignation |
| Id_client | Code client |
| Nom_client | Nom du client |
| Prénom_client | Prénom du client |
| Sex_client | Sexe du client |
| Adr_client | Adresse du client |
| N°_de_tel_client | Numéro de téléphone du client |
| | Adresse mail du client |

Chapitre 02 : Conception

| | |
|-----------------|----------|
| Adr_mail_client | |
| Méthodes | - CRU(). |

| Carte de fidélité | |
|-------------------|---|
| Attributs | Désignation |
| Id_carte | Code de la carte |
| Date-début | Date début de la carte de fidélité |
| Date_exp | Date d'expiration de la carte de fidélité |
| Nbr_point | Nombre de points acquis pour |
| Méthodes | - CRUD() |

| Caisse | |
|-------------|--|
| Attributs | Désignation |
| Id_caisse | Code de la caisse |
| Fond_caisse | Montant du fond de caisse |
| Méthodes | <ul style="list-style-type: none"> - ouvrir la caisse(). - Fermer la caisse(). - Mettre la caisse en pause(). |

| Employé | |
|------------|---|
| Attributs | Désignation |
| Id_emp | Code utilisateur |
| Nom_emp | Nom de l'utilisateur |
| Prénom_emp | Prénom de l'utilisateur |
| Date_début | Date de début d'embauche |
| N°_tel | Numéro de téléphone de |
| Adr_emp | Adresse de l'employé |
| Adr_mail | Adresse mail de l'employé |
| Mot_passe | Mot de passe de l'employé |
| Sexe-emp | Sexe de l'employé |
| Type-emp | Type de l'employé (caissier ,arrière caissier, vendeur, et responsable du magasin) |
| Méthodes | - CRUD() |

| Remise au coffre | |
|------------------|------------------------|
| Attributs | Désignation |
| N°_env | Numéro d'enveloppe |
| Montant_env | Montant de l'enveloppe |
| Date-env | Date de l'enveloppe |

Chapitre 02 : Conception

| | |
|------------|-----------------------------------|
| Heure_env | Heure de l'enveloppe |
| Détail_env | Détails du contenu de l'enveloppe |

| Remise | |
|-------------|--------------------------------|
| Attributs | Désignation |
| N°_remise | Numéro de la remise |
| Date_début | Date début de la remise |
| Date_exp | Date d'expiration de la remise |
| Type_remise | Type de la remise |
| Méthodes | - CRUD(). |

| Coffre | |
|-----------|-----------------------|
| Attributs | Désignation |
| Id_coffre | Identifiant du coffre |
| Fond_coff | Fond de coffre |

5-1-3 Diagramme de classe de la base de données de la caisse :

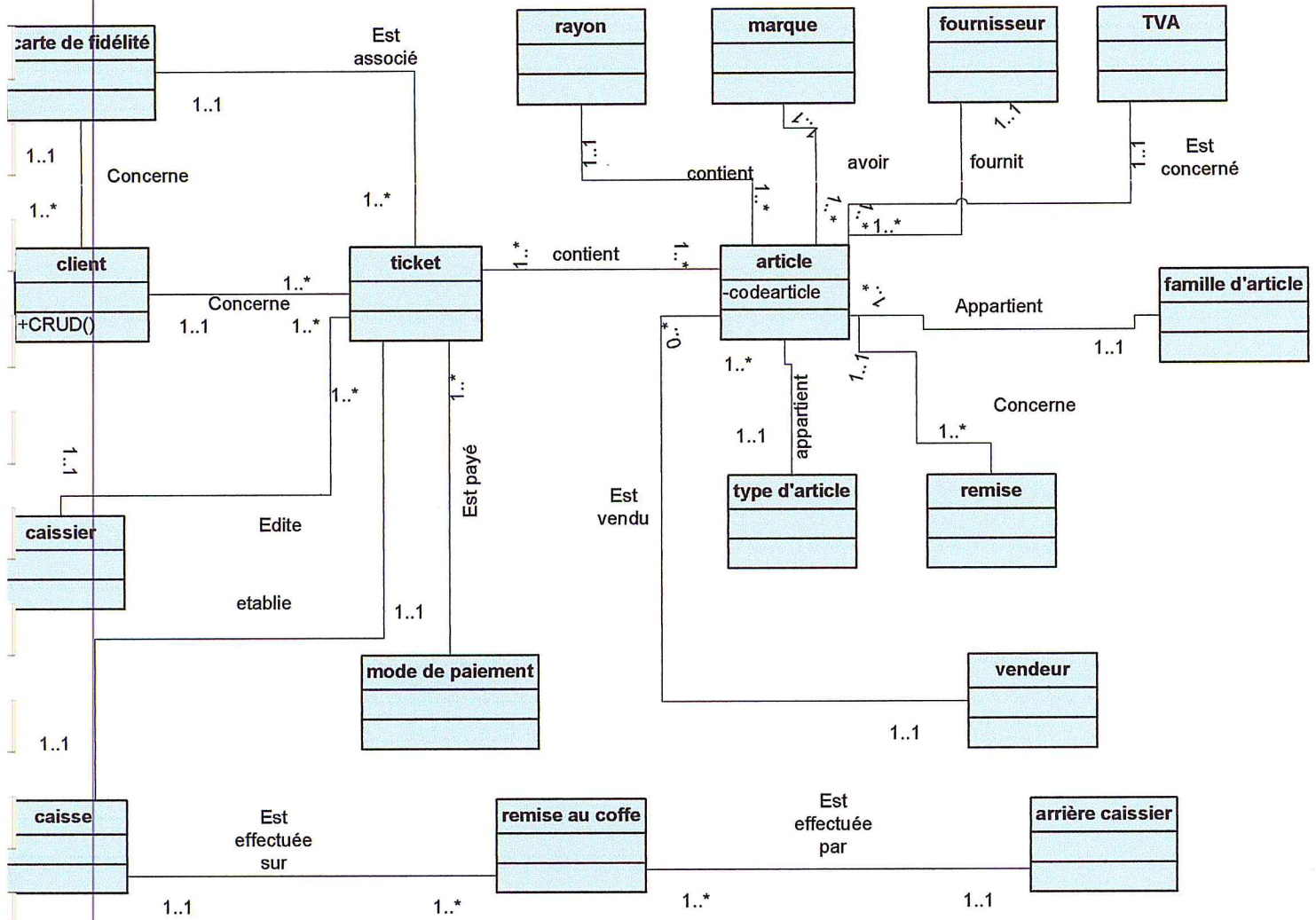


Figure 20: «Diagramme de classe de la base de données de la caisse »

5-1-4 Description des classes :

| Article | |
|------------|----------------------------|
| Attributs | Désignation |
| Code_art | Code article interne |
| Code_EAN | Code barre de l'article |
| Lib-art | Libelle article |
| Prix_vente | Prix de vente de l'article |

| Rayon | |
|-----------|---------------|
| Attributs | Désignation |
| Code_ray | Code du rayon |
| Nom_ray | Nom du rayon |

Chapitre 02 : Conception

| TVA | |
|-----------|----------------|
| Attributs | Désignation |
| Code_tva | Code de la TVA |
| Type_tva | Type de la TVA |

| Famille d'article | |
|-------------------|----------------------------|
| Attributs | Désignation |
| Code_fam | Code de famille d'articles |
| Nom_fam | Nom de famille d'articles |

| Ticket | |
|--------------|-------------------|
| Attributs | Désignation |
| Id_tick | Code du ticket |
| Montant_tick | Montant du ticket |
| Date_tick | Date du ticket |
| Heure_tick | Heure du ticket |

| Mode de paiement | |
|------------------|--------------------------|
| Attributs | Désignation |
| Id_MDP | Code du mode de paiement |
| Type_MDP | Type du mode de paiement |

| Type d'article | |
|----------------|------------------------|
| Attributs | Désignation |
| Id_type_art | Code du type d'article |
| Type_article | Type d'article |

| Client | |
|---------------|------------------|
| Attributs | Désignation |
| Id_client | Code client |
| Nom_client | Nom du client |
| Prénom_client | Prénom du client |

| Carte de fidélité | |
|-------------------|---|
| Attributs | Désignation |
| Id_carte | Code de la carte |
| Date-début | Date début de la carte de fidélité |
| Date_exp | Date d'expiration de la carte de fidélité |
| Nbr_point | Nombre de points acquis pour |

Chapitre 02 : Conception

| Caisse | |
|-------------|---------------------------|
| Attributs | Désignation |
| Id_caisse | Code de la caisse |
| Fond_caisse | Montant du fond de caisse |

| Vendeur | |
|-------------|-------------------|
| Attributs | Désignation |
| Id_vend | Code vendeur |
| Nom_vend | Nom du vendeur |
| Prénom_vend | Prénom du vendeur |

| Caissier | |
|-----------------|--------------------|
| Attributs | Désignation |
| Id_caissier | Code caissier |
| Nom_caissier | Nom du caissier |
| Prénom_caissier | Prénom du caissier |

| Remise au coffre | |
|------------------|-----------------------------------|
| Attributs | Désignation |
| N°_env | Numéro d'enveloppe |
| Montant_env | Montant de l'enveloppe |
| Date-env | Date de l'enveloppe |
| Heure_env | Heure de l'enveloppe |
| Détail_env | Détails du contenu de l'enveloppe |

| Remise | |
|-------------|--------------------------------|
| Attributs | Désignation |
| N°_remise | Numéro de la remise |
| Date_début | Date début de la remise |
| Date_exp | Date d'expiration de la remise |
| Type_remise | Type de la remise |

5-1-5 Schéma de l'entrepôt :

L'entrepot va contenir l'historique des tickets d'une manière plus juste il compte le contenu de la table ticket qu'on la vide chaque fin de journée afin d'éviter la saturation de notre base de données centrale d'où notre entrepot aura le schéma suivant :

| Ticket | |
|--------------|---------------------------------|
| Attributs | Désignation |
| Id_tick | Code du ticket |
| Montant_tick | Montant du ticket |
| Date_tick | Date du ticket |
| Heure_tick | Heure du ticket |
| Id_emp | Identifiant de l'employés de la |
| Id_mdp | caisse. |
| Id_caisse | Identifiant du mode de paiement |
| Id-client | Identifiant de la caisse. |
| | Identifiant du client |

Le développement du modèle statique étant achevé, nous procéderons aux spécifications des états de transition.

5-2 Diagrammes de séquence :

Après avoir vu les diagrammes des cas d'utilisation techniques et fonctionnels et le diagramme de classe où on a déterminé les objets de notre système ; nous allons voir les diagrammes de séquence des cas d'utilisation principaux .

5-2-1 Diagramme de séquence du cas d'utilisation « Créer un article » :

| Description du cas « Créer article » | |
|--|---|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à créer un article | 1) Saisir Code article interne 2) Saisir Code barre de l'article 3) Saisir Libelle de l'article 4) Saisir Description de l'article 5) Saisir Prix d'achat de l'article 6) Saisir Prix de vente de l'article 7) Sélectionner type d'article 8) Sélectionner famille d'article 9) Sélectionner TVA 10) Sélectionner fournisseur 11) Sélectionner marque d'article |
| Post condition | ✓ Ajout terminé avec succès |

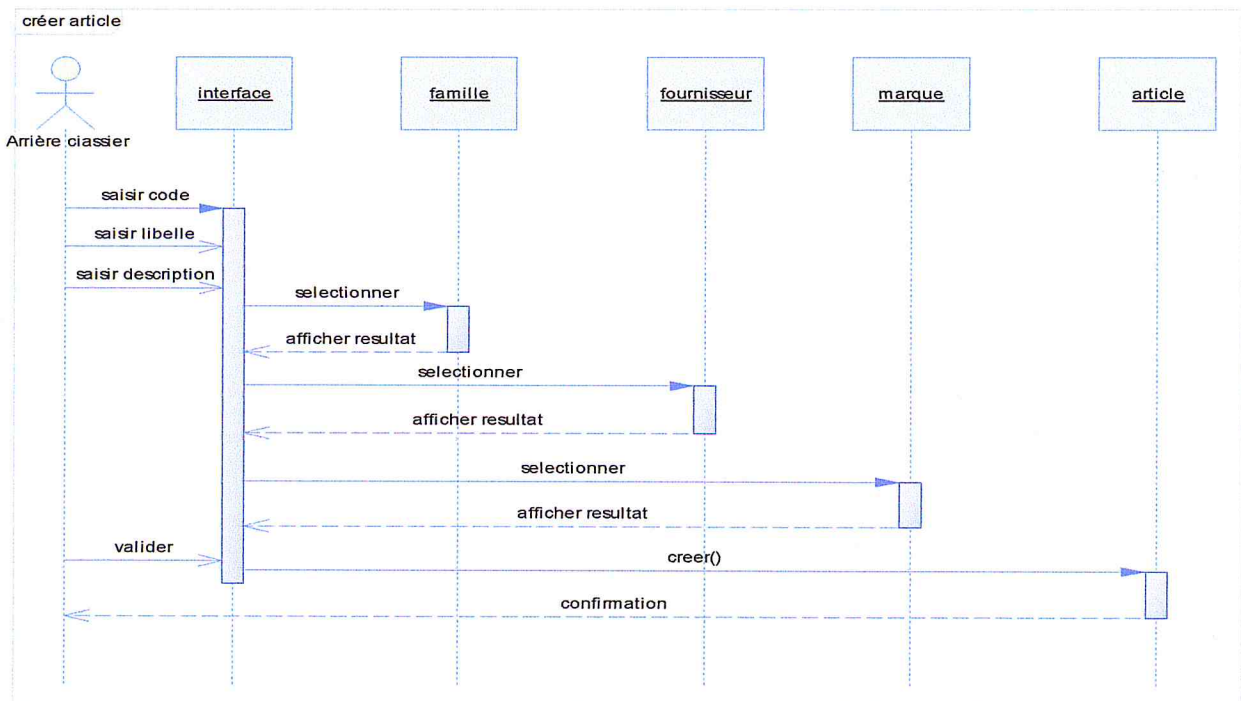


Figure 21 : Diagramme de séquence du cas « créer un article »

Chapitre 02 : Conception

5-2-2 Diagramme de séquence du cas d'utilisation «modifier article»

| Description du cas « Modification d'article » | |
|---|--|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à modifier un | 1) Saisir nouvelles valeurs de l'article 2) Valider |
| Post condition | ✓ Modification terminée avec succès |

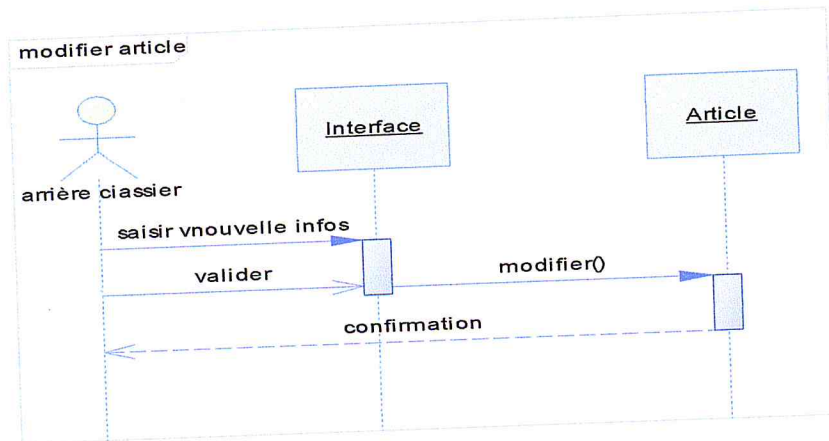


Figure 22 : Diagramme de séquence du cas « modifier un article »

5-2-3 Diagramme de séquence du cas d'utilisation «supprimer un article» :

| Description du cas « supprimer article » | |
|--|--|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à supprimer un | 1) Saisir Code article interne 2) Valider |
| Post condition | ✓ Ajout terminée avec succès |

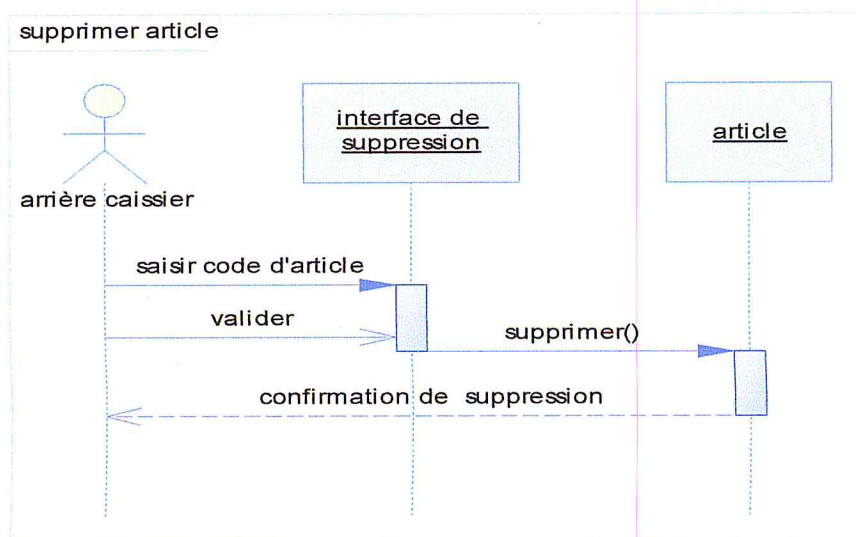


Figure 23 : Diagramme de séquence du cas « supprimer un article ».

5-2-4 Diagramme de séquence du cas d'utilisation «rechercher un article»

| Description du cas « rechercher un article » | |
|---|--|
| Pré-condition | ✓ Etre un caissier |
| Scénario nominal associé à rechercher un article par famille | 1) Sélectionner famille d'article 2) Le système affiche liste des articles 3) Sélectionner l'article 4) Le système affiche le profil d'article |
| Scénario nominal associé à rechercher un article par type | 1) Sélectionner type d'article 2) Le système affiche liste des articles 3) Sélectionner l'article 4) Le système affiche le profil d'article |
| Scénario nominal associé à rechercher un article par code | 1) Saisir code de l'article 2) Valider 3) Le système affiche ligne de l'article 4) Sélectionner la ligne . 5) Le système affiche le profil d'article |
| Scénario nominal associé à rechercher un article par EAN | 1) Saisir EAN de l'article 2) Valider 3) Le système affiche ligne de l'article 4) Sélectionner la ligne. 5) Le système affiche le profil d'article |

| | |
|-----------------------|---------------------|
| Post condition | ✓ Sortir du système |
|-----------------------|---------------------|

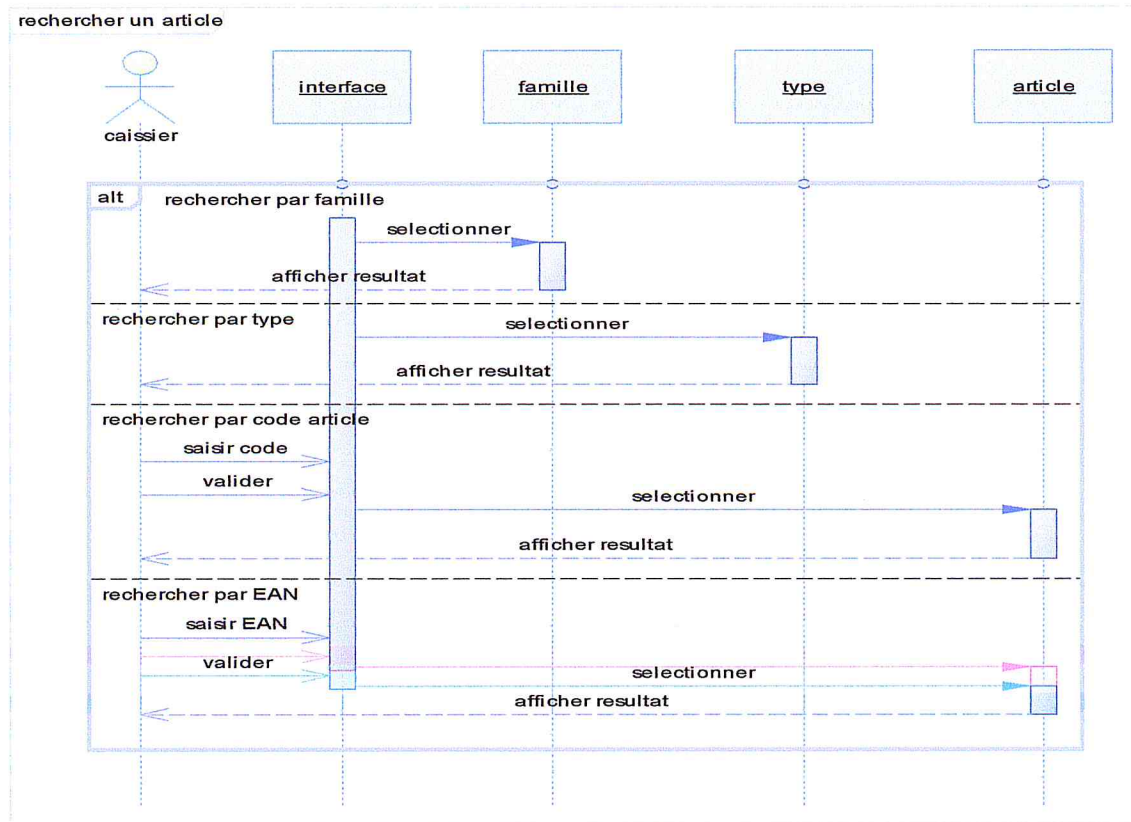


Figure 24 : Diagramme de séquence du cas « rechercher un article »

5-2-5 Diagramme de séquence du cas d'utilisation «ouvrir une caisse»

| Description du cas « ouvrir une caisse » | |
|---|---|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à ouvrir une caisse | 1) Ajouter fond de caisse 2) Valider |
| Post condition | ✓ Ajout terminé avec succès |

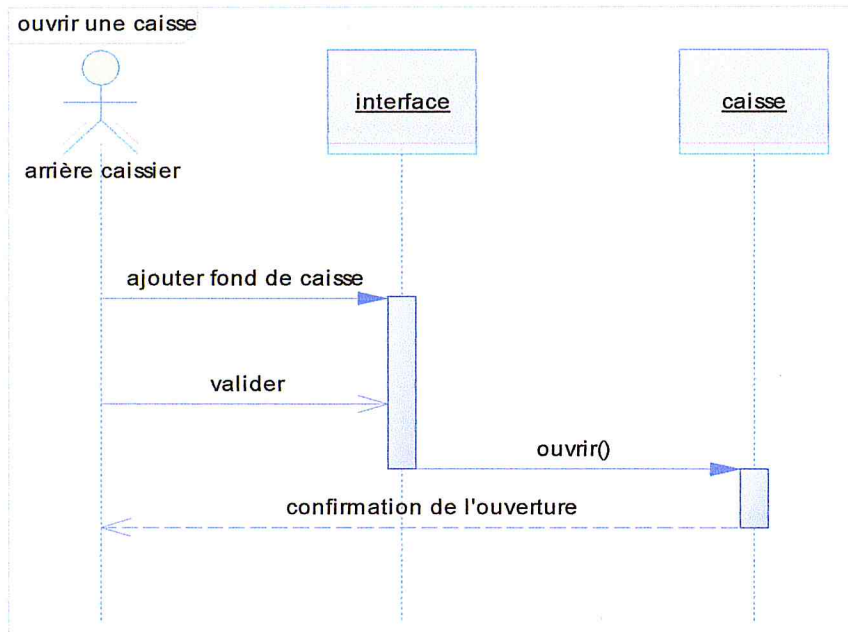


Figure 25 : Diagramme de séquence du cas « créer un article »

5-2-6 Diagramme de séquence du cas d'utilisation «payer un ticket par bon d'achat/chèque cadeau» :

| Description du cas « payer un ticket » | |
|--|---|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à payer un ticket par un chèque cadeau | 1) Saisir montant du chèque cadeau 2) Valider 3) Le système effectue des vérifications de montant |
| Post condition | ✓ Edition du ticket |

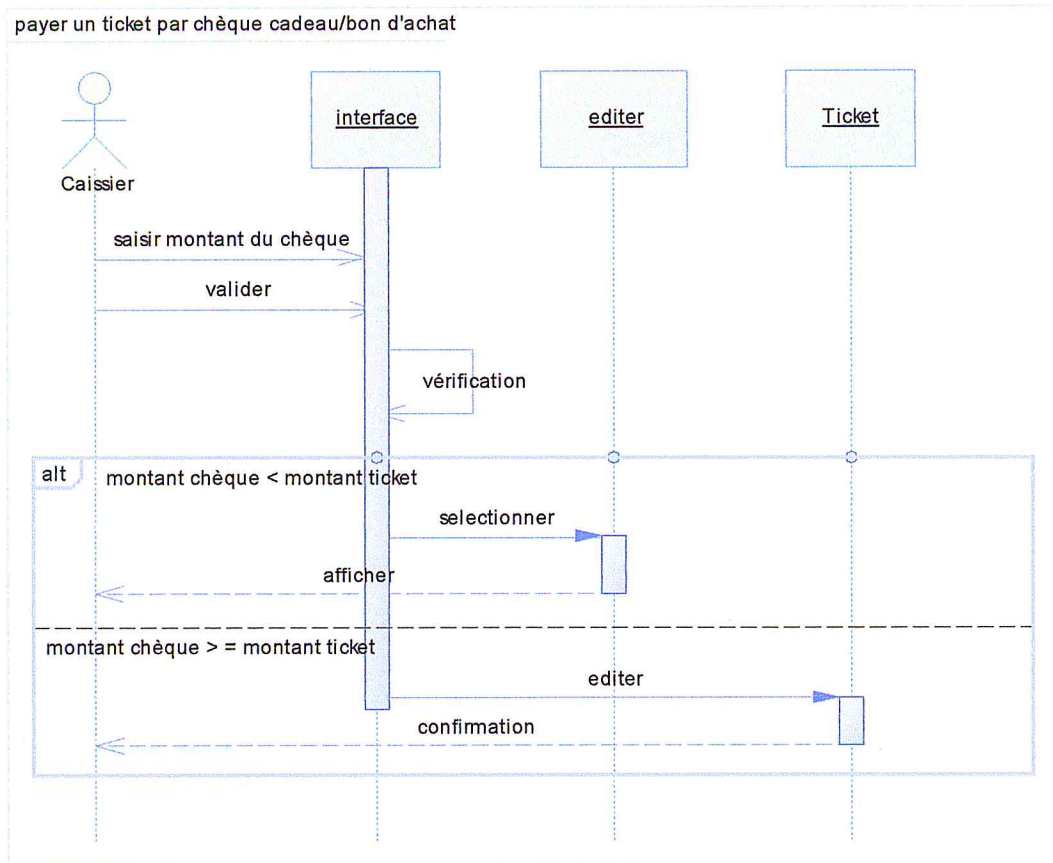


Figure 26 : Diagramme de séquence du cas « payer un ticket par un chèque cadeau/Bon d'achat »

5-2-7 Diagramme de séquence du cas d'utilisation «créer une carte de fidélité»

| Description du cas « créer une carte de fidélité » | |
|---|--|
| Pré-condition | ✓ Etre un arriére caissier |
| Scénario nominal associé à créer une carte de fidélité | 1) Saisir nom de l'employé 2) Saisir prénom de l'employé 3) Saisir date de naissance de l'employé 4) Sélectionner le sexe de l'employé 5) Saisir l'adresse de l'employé 6) Saisir N° d'identité 7) Saisir numéro de téléphone de l'employé 8) Saisir l'adresse mail de l'employé 9) Sélectionner le type de l'employé 10) Saisir mot de passe de l'employé 11) Valider |
| Post condition | ✓ Ajout terminé avec succès |

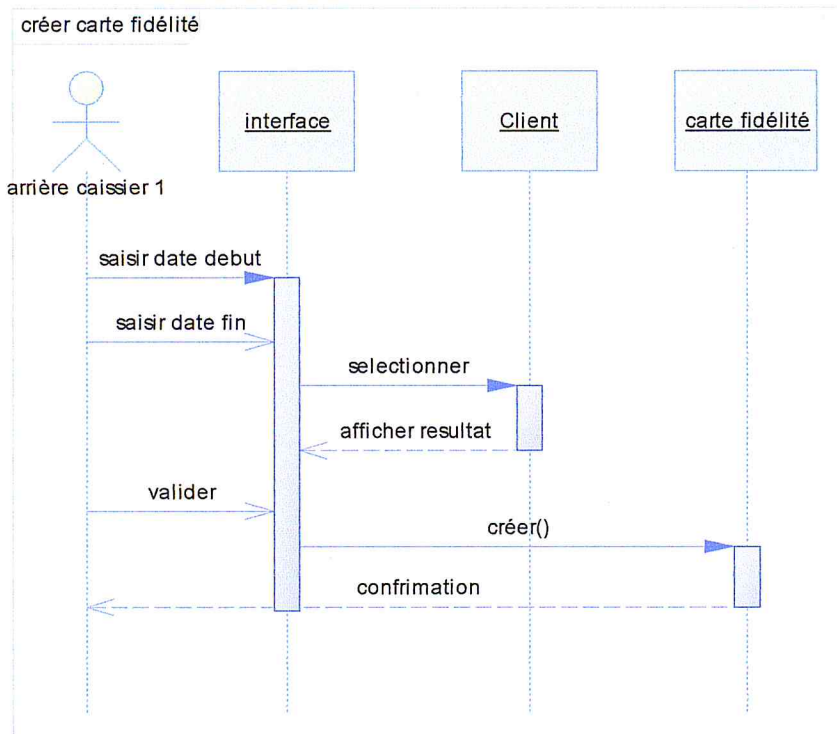


Figure 27 : Diagramme de séquence du cas « créer une carte de fidélité ».

5-2-8 Diagramme de séquence du cas d'utilisation « Créer employés »

| Description du cas « créer employé » | |
|--|--|
| Pré-condition | ✓ Etre un arrière caissier |
| Scénario nominal associé à créer un employé | 1) Saisir nom de l'employé 2) Saisir prénom de l'employé 3) Saisir date de naissance de l'employé 4) Sélectionner le sexe de l'employé 5) Saisir l'adresse de l'employé 6) Saisir N° d'identité 7) Saisir numéro de téléphone de l'employé 8) Saisir l'adresse mail de l'employé 9) Sélectionner le type de l'employé 10) Saisir mot de passe de l'employé 11) Valider |
| Post condition | ✓ Ajout terminé avec succès |

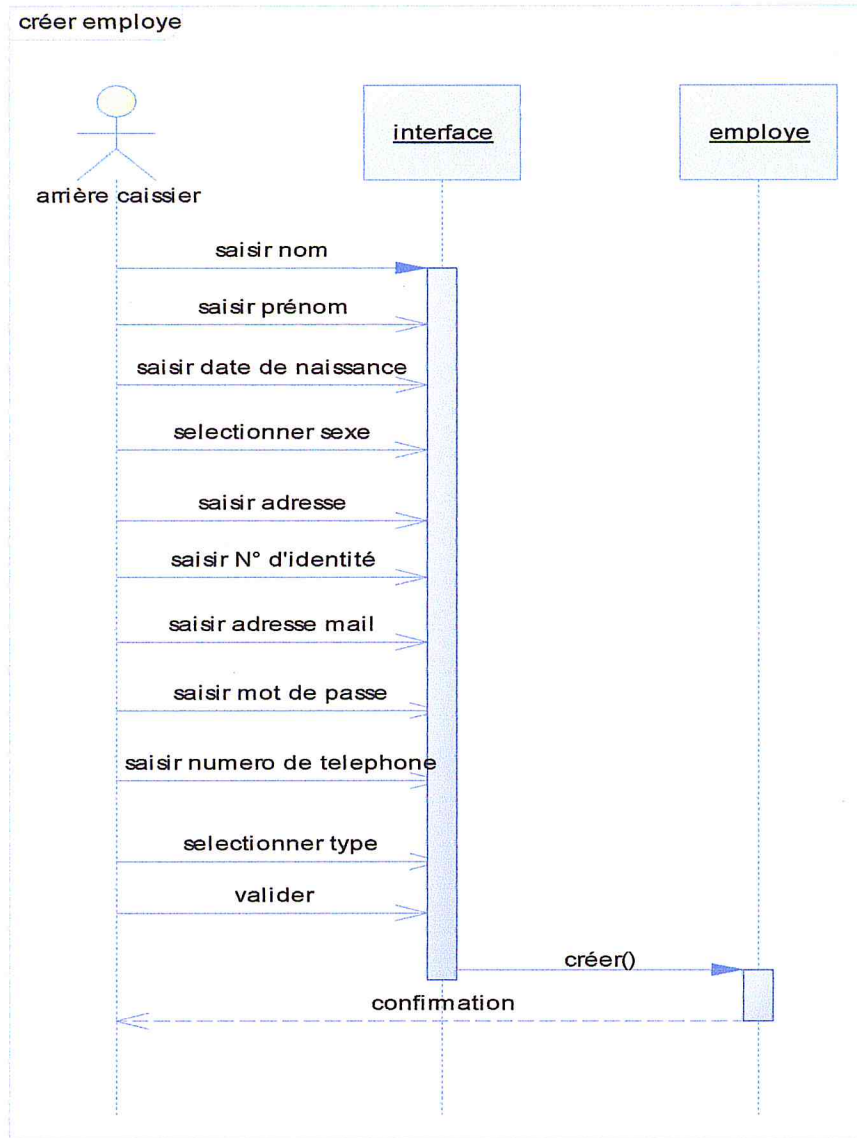


Figure 28 : Diagramme de séquence du cas « créer un employé ».

6- Conception générique :

La conception générique consiste à développer la solution qui répond aux spécifications techniques. Cette conception est qualifiée de générique car elle est entièrement indépendante des aspects fonctionnels spécifiés en branche gauche du modèle en Y. La conception générique reste donc une activité de la branche droite [Roques 2002].

De ce fait, la conception générique se préoccupe de détailler les aspects technique de notre système.

6-1 Authentification :

L'accès à l'application est basé sur les privilèges attribués aux utilisateurs selon leurs niveaux de qualification, et ils sont répartis en 3 niveaux :

- ◆ Premier niveau,
- ◆ Second niveau,
- ◆ Troisième niveau.

| Niveau | Statut | Privilèges |
|------------------|---------------------------|---|
| Premier niveau | Caissier | L'accès permet le caissier de : Gestion de la caisse (Front) |
| Second niveau | Arrière Caissier | L'accès permet l'arrière caissier de : ✓ Gestion de la caisse (Back) |
| Troisième niveau | Responsable Du magasin | L'accès permet le responsable du magasin de : ✓ Gestion du magasin |

6-2 Archivage des tickets :

Dans le but de garder la trace des données des tickets, le responsable doit archiver les anciennes données afin d'étudier la progression des ventes.

L'archivage des tickets de ventes dans l'entrepôt de données se fait chaque fin de journée automatiquement.

6-3 La Recherche :

Le système doit faciliter aux employés la tâche de recherche d'un article pour cette raison, nous devons leur fournir plusieurs critères de recherche (par famille , par libelle, par code article , par code EAN).

Nous devons attribuer un moteur de recherche comprenant les quatres critères cités précédemment.

La conception générique étant terminée, nous procéderons maintenant à la conception préliminaire.

7- Conception préliminaire :

Selon[Roques 2002], la conception préliminaire est l'étape la plus délicate du processus 2TUP, car elle en représente le cœur. C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques.

7-1 Architecture de l'outil :

Le serveur central de NUMIDIS envoie la descente des mises à jour aux serveurs des magasins, chaque serveur de magasin génère un fichier de mise à jour et le met à la disposition des caisses. Les caisses importent le fichier et l'intègrent dans leur base, étant donné que chaque caisse a une image vierge de la base de données centrale de son magasin.

Les caisses travaillent simultanément sur la base de données centrale, au cas de déconnexion avec le serveur, les caisses passent en mode dégradée, d'où la caisse effectue ses tâches dans la petite base propre à elle et elle essaiera de se connecter au serveur. Au moment de la reconnexion la caisse doit repasser au mode connecté manuellement et effectue ses opérations sur la base centrale. En fin de journée on fait la remontée des données des petites bases de chaque caisse vers la base centrale après on envoie toutes les ventes de la base centrale à l'entrepôt et on vide la base centrale.

7-2 Architecture physique :

L'architecture à trois niveaux, dite aussi architecture 3-tiers [iFranche], est devenue un grand classique. Elle permet de distinguer rigoureusement la couche « présentation », la couche « métier » et la couche « données » d'une application (Figure 29) Le tiers présentation, qui réalise l'interface graphique de l'application, gère les interactions avec l'utilisateur et la logique de navigation.

- Le tiers métier contient des composants qui collaborent pour régler des problèmes métiers ou de logique applicative relatifs au domaine d'application du système d'information.
- Le tiers persistance est utilisé par la couche métier pour conserver des informations sur un support de persistance (base de données, fichiers, ...)

Chapitre 02 : Conception

Cette notion de couches n'émet aucune hypothèse quant à la distribution physique de l'application. Les développeurs sont ainsi affranchis de la localisation du code, que celui-ci soit dans un autre processus ou sur une autre machine. Mais cette séparation en couches a aussi d'autres objectifs :

- L'évolutivité : ce découpage offre la garantie de la meilleure évolutivité possible pour chaque couche. La mise à jour d'une couche n'affecte pas ou peu les autres couches, dans la mesure de ses relations avec les couches voisines. Un changement dans la partie métier n'oblige par exemple pas à redéployer tous les clients.
- La sécurité : les données sont traitées par le tiers métier et ne sont pas visibles directement par les clients. Pour répondre à des exigences de sécurité (par exemple celles demandées par des applications prenant en charge des achats ou transactions bancaires depuis le domicile), on peut intégrer des niveaux d'isolation entre les couches comme des protocoles d'authentification (qui peuvent représenter eux-mêmes un niveau de l'architecture).
- L'interopérabilité : l'uniformisation des interfaces entre les couches, et/ou la mise en place de protocoles standardisés d'échange permet de faire communiquer des services réalisés avec des moyens différents (langage de programmation, protocoles, ...) et sur des supports hétérogènes (système d'exploitation, matériel, ...).
- La réutilisabilité : les entreprises ont besoin d'accéder et de distribuer des informations de leurs systèmes d'application antérieurs à Internet. Cette architecture permet, grâce à des protocoles de communication, d'utiliser l'existant. D'autre part, grâce à la séparation logique, l'entreprise pourra réutiliser des composants dans d'autres applications.
- La maîtrise de la montée en charge : on peut notamment modifier le déploiement physique selon les besoins d'exploitation et utiliser des « serveurs d'application » qui optimisent l'utilisation des ressources pour des utilisateurs nombreux.

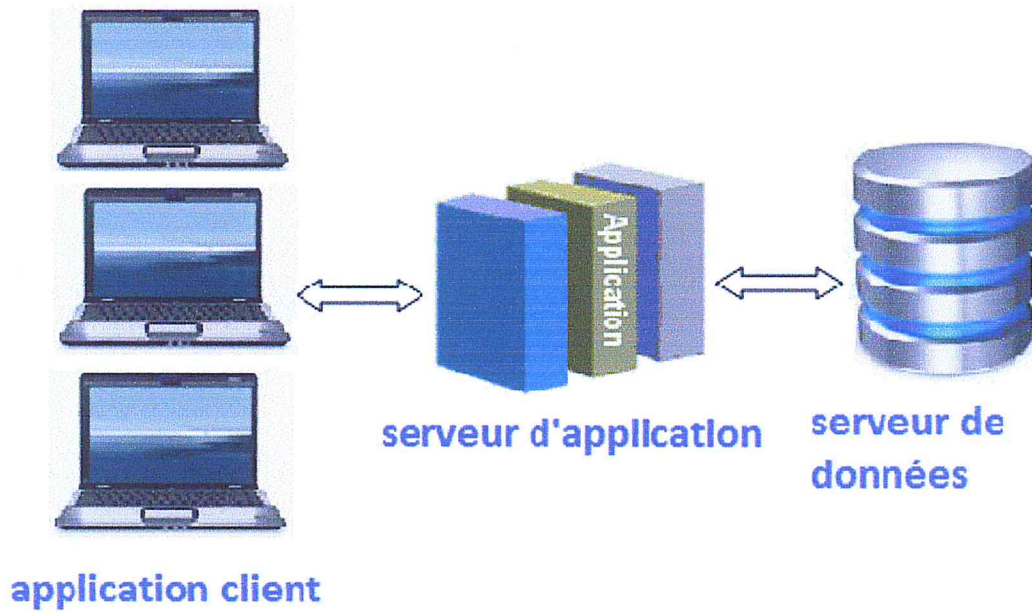


Figure 29 : Architecture 3tiers

7-3 Développement du modèle de déploiement :

Le déploiement d'une solution client/serveur se construit sur la définition des postes de travail. Les modèles de déploiement et de configuration matérielle s'expriment tous deux à l'aide d'un diagramme de déploiement. Cependant, ils n'expriment pas tout à fait le même niveau de description.

Le modèle de déploiement considère plutôt chaque nœud comme un poste de travail. Il exprime la répartition physique des fonctions métier du système et permet de justifier la localisation des bases de données et des environnements de travail.

Le modèle de déploiement aide à préciser la qualification des postes client, des réseaux et de leur sécurité physique, par rapport à des critères fonctionnels [Roque 2002].

Le modèle de déploiement de notre système est représenté dans le diagramme suivant :

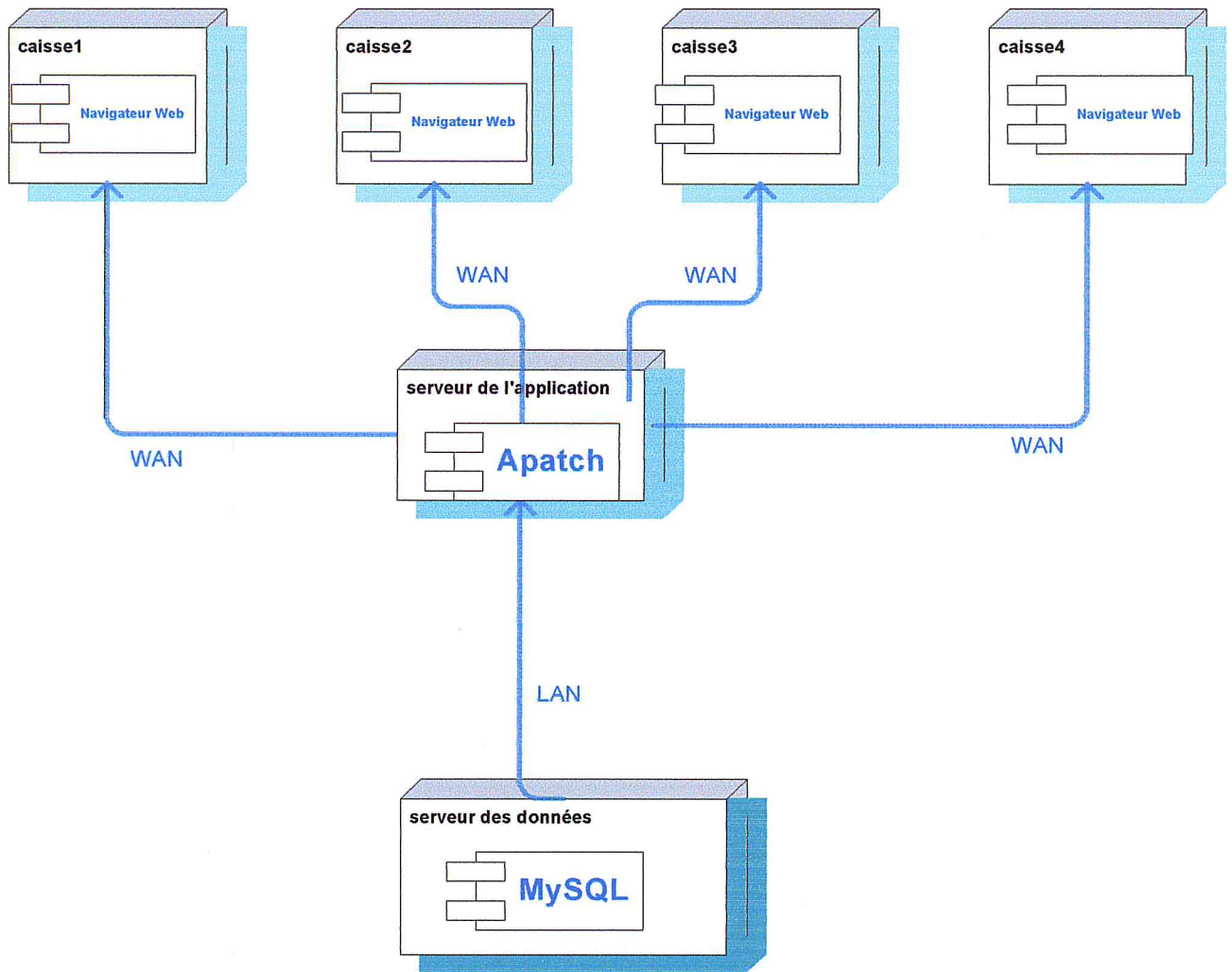


Figure 30 : Diagramme de déploiement

7-4 Critères ergonomiques adoptés :

L'ergonomie du système représente une clé très importante pour attirer l'utilisateur à notre système, pour cette raison, nous avons donné une très grande importance aux critères ergonomique pour attirer le maximum des utilisateurs, il ne faut pas oublier que les visiteurs de notre système sont variés (de différent domaine, niveaux intellectuels etc.).

Les critères ergonomiques que nous avons utilisés se résument en :

Chapitre 02 : Conception

- ✓ L'utilisation des interfaces cohérentes (même couleurs, même forme etc.).
- ✓ L'utilisation des indications pour faciliter l'utilisation et orienter l'utilisateur
- ✓ La simplicité des interfaces en accès et en consultation (minimiser les tâches au visiteur).

La conception préliminaire étant terminée, nous entamerons maintenant la conception détaillée.

8- Conception détaillé :

Une fois le diagramme de classe élaboré, celui-ci peut être traduit en un modèle relationnel, c'est-à-dire en un ensemble de relation en troisième forme normale.

Cet ensemble de relations normalisées peut être alors implémenté directement, sous forme de base de données, moyennant un système de gestion de base de données relationnelle (SGBDR) tels que SQL, Access, Oracle ... [Graïne 2009].

Le passage du diagramme de classe vers le modèle relationnel se fait selon six règles :

- 1) Une classe du diagramme de classes se transforme en une relation dans le modèle relationnel. Une clé sera ajoutée à la relation,
- 2) Une association ayant une multiplicité (0,*) ou (1,*) dans le diagramme de classe, se traduit par l'ajout en tant qu'attribut simple de la clé de la classe située de côté de multiplicité 1 ou 0, dans la classe située du côté de multiplicité *,
- 3) Une association ayant une multiplicité (*, *) dans le diagramme de classe se traduit par une relation supplémentaire dans le modèle relationnel ayant pour clé, les clés des relations issues des classes participante à l'association,
- 4) Une classe d'association se transforme en une relation ayant pour clé, les clés des classes participante à cette association,
- 5) Dans une relation d'héritage, la clé de la classe mère est ajoutée en tant que clé dans la classe fille,
- 6) Dans le cas d'une association réflexive, la clé ajoutée à la relation correspondante est dupliquée dans la relation après avoir été renommée.

8- 1 - Application des règles de passage sur le diagramme de classe :

En premier lieu, nous allons faire le passage des classes en tables MySQL, ensuite le passage des relations en tables MySQL en tenant compte des clés primaires et étrangères ainsi que les -index.

Chapitre 02 : Conception

8-1-1 Diagramme de classe de la base de données centrale :

| Article | |
|-------------|--------------|
| Attributs | Type |
| Code_art | Int(4) |
| Code_EAN | Int(16) |
| Lib_art | Varchar(50) |
| Desc_art | Varchar(100) |
| Prix_achat | Float(8) |
| Prix_vente | Float(8) |
| Quant_art | Int(20) |
| Date_fab | Date |
| Date_peremp | Date |

| Clé | Type |
|-------------------------|---------------|
| <u>Code art</u> | Clé |
| <u>Code ray</u> | Clé |
| <u>Code mar</u> | Clé |
| <u>Code four</u> | Clé |
| <u>Code-fam</u> | Clé |
| <u>Code tva</u> | Clé |
| <u>Id type a rt</u> | Clé étrangère |

| Rayon | |
|-----------|--------------|
| Attributs | Type |
| Code_ray | Int(2) |
| Nom_ray | Varchar (50) |

| Clé | Type |
|-----------------|--------------|
| <u>Code ray</u> | Clé primaire |

| Marque | |
|-----------|-------------|
| Attributs | Désignation |
| Code_marq | Int(4) |
| Nom_marq | Varchar(50) |

| Clé | Type |
|-----------------|--------------|
| <u>Code mar</u> | Clé primaire |

| Fournisseur | |
|-------------|--------------|
| Attributs | Type |
| Code_fourn | Int(30) |
| Nom_fourn | Varchar(50) |
| Adr_fourn | Varchar(100) |
| Num_fourn | Varchar(21) |

| Clé | Type |
|------------------|--------------|
| <u>Code four</u> | Clé primaire |

| Famille d'article | |
|-------------------|-------------|
| Attributs | Type |
| Code_fam | Int(3) |
| Nom_fam | Varchar(50) |

| Clé | Type |
|-----------------|--------------|
| <u>Code fam</u> | Clé primaire |

Chapitre 02 : Conception

| Ticket | |
|----------------|----------|
| Attributs | Type |
| <u>Id_tick</u> | Int(4) |
| Montant_tick | Float(8) |
| Date_tick | Date |
| Heure_tick | Datetime |

| Mode de paiement | |
|------------------|-------------|
| Attributs | Type |
| <u>Id_mdp</u> | Int(2) |
| Type_mdp | Varchar(50) |

| Type d'article | |
|--------------------|-------------|
| Attributs | Type |
| <u>Id_type_art</u> | Int(3) |
| Type_art | Varchar(50) |

| Client | |
|-------------------|--------------|
| Attributs | Type |
| <u>Id_client</u> | Int(10) |
| Nom_client | Varchar(50) |
| Prénom_client | Varchar(50) |
| Sex_client | Varchar(15) |
| Adr_client | Varchar(100) |
| Num_de_tel_client | Varchar(21) |
| Adr_mail_client | Varchar(100) |

| Carte de fidélité | |
|-------------------|---------|
| Attributs | Type |
| <u>Id_carte</u> | Int(10) |
| Date-début | Date |
| Date_exp | Date |
| Nbr_point | Int(4) |

| Clé | Type |
|--------------------|---------------|
| <u>Id_tick</u> | Clé primaire |
| <u>Id_caissier</u> | Clé étrangère |
| <u>Id_mdp</u> | Clé étrangère |
| <u>Id_carte</u> | Clé étrangère |
| <u>Id_client</u> | Clé étrangère |

| Clé | Type |
|---------------|--------------|
| <u>Id_mdp</u> | Clé primaire |

| Clé | Type |
|------------------|--------------|
| <u>Id_type_a</u> | Clé primaire |

| Clé | Type |
|------------------|---------------|
| <u>Id_client</u> | Clé primaire |
| <u>Id_carte</u> | Clé étrangère |

| Clé | Type |
|-----------------|--------------|
| <u>Id_carte</u> | Clé primaire |

Chapitre 02 : Conception

| Caisse | |
|-------------|------------|
| Attributs | Type |
| Id_caisse | Int(2) |
| Fond_caisse | Float (10) |

| Clé | Type |
|------------------|--------------|
| <u>Id_caisse</u> | Clé primaire |

| TVA | |
|-----------|-------------|
| Attributs | Type |
| Code_tva | Int(1) |
| Type_tva | Varchar(10) |

| Clé | Type |
|-----------------|--------------|
| <u>Code_tva</u> | Clé primaire |



| Remise au coffre | |
|------------------|-------------|
| Attributs | Type |
| Num_env | Int(5) |
| Montant_env | Float(6) |
| Date-env | Date |
| Heure_env | Datetime |
| Type | Varchar(20) |

| Clé | Type |
|------------------|--------------|
| <u>Num_env</u> | Clé primaire |
| <u>Id_caisse</u> | Clé |
| <u>Id_emp</u> | Clé |

| Employé | |
|---------------|--------------|
| Attributs | Type |
| Id_emp | Int(2) |
| Nom_emp | Varchar(50) |
| Prenom_emp | Varchar(50) |
| Date_d'embauc | Date |
| he | Varchar(21) |
| N°_tel | Varchar(100) |
| Adr_emp | Varchar(50) |
| Adr_mail | Varchar(50) |
| Mot_passe | Varchar(20) |
| Sex-emp | Varchar(30) |
| Type_emp | |

| Clé | Type |
|---------------|--------------|
| <u>Id_emp</u> | Clé primaire |

Chapitre 02 : Conception

| Remise | |
|-------------|-------------|
| Attributs | Type |
| Num_remise | Int(5) |
| Date_début | Date |
| Date_exp | Date |
| Type_remise | Varchar(10) |

| Clé | Type |
|-----------------|--------------|
| <u>Num remi</u> | Clé primaire |
| <u>Code art</u> | Clé |

| Est_vendu | |
|-----------|--------|
| Attributs | Type |
| Id_emp | Int(2) |
| Code_art | Int(4) |

| Clé | Type |
|-----------------|-------|
| <u>Id emp</u> | Index |
| <u>Code-art</u> | Index |

| Ticket_contient_article | |
|-------------------------|---------|
| Attributs | Type |
| Id_tick | Int(4) |
| Code_art | Int(4) |
| Quantite | Int (3) |

| Clé | Type |
|-----------------|-------|
| <u>Id tick</u> | Index |
| <u>Code-art</u> | Index |

8-1-2 Diagramme de classe de la base de données caisse :

| Article | |
|------------|-------------|
| Attributs | Désignation |
| Code_art | Int(4) |
| Code_EAN | Int(16) |
| Lib-art | Varchar(50) |
| Prix_vente | Float(8) |

| Clé | Type |
|-------------------|--------------|
| <u>Code-art</u> | Clé primaire |
| <u>Code ray</u> | Clé |
| <u>Code-fam</u> | Clé |
| <u>Code tva</u> | Clé |
| <u>Id type ar</u> | Clé |

| Rayon | |
|-----------|--------------|
| Attributs | Désignation |
| Code_ray | Int(2) |
| Nom_ray | Varchar (50) |

| Clé | Type |
|-----------------|--------------|
| <u>Code ray</u> | Clé primaire |

Chapitre 02 : Conception

| TVA | |
|----------------------|-----------------------|
| Attributs | Désignation |
| Code_tva Type_tva | Int(1) Varchar(10) |

| Clé | Type |
|-----------------|--------------|
| <u>Code_tva</u> | Clé primaire |

| Famille d'article | |
|---------------------|-----------------------|
| Attributs | Désignation |
| Code_fam Nom_fam | Int(3) Varchar(50) |

| Clé | Type |
|-----------------|--------------|
| <u>Code_fam</u> | Clé primaire |

| Ticket | |
|--|--|
| Attributs | Désignation |
| Id_tick Montant_tick Date_tick Heure_tick | Int(4) Float(8) Date Datetime |

| Clé | Type |
|--------------------|--------------|
| <u>Id_tick</u> | Clé primaire |
| <u>Id caissier</u> | Clé |
| <u>Id mdp</u> | Clé |
| <u>Id carte</u> | Clé |
| <u>Id client</u> | Clé |

| Mode de paiement | |
|--------------------|-----------------------|
| Attributs | Désignation |
| Id_MDP Type_MDP | Int(2) Varchar(50) |

| Clé | Type |
|---------------|--------------|
| <u>Id_mdp</u> | Clé primaire |

| Type d'article | |
|-----------------------------|-----------------------|
| Attributs | Désignation |
| Id_type_art Type_article | Int(3) Varchar(50) |

| Clé | Type |
|--------------------|--------------|
| <u>Id_type_art</u> | Clé primaire |

Chapitre 02 : Conception

| Client | |
|--|---------------------------------------|
| Attributs | Désignation |
| Id_client Nom_client Prénom_client | Int(10) Varchar(50) Varchar(50) |

| Clé | Type |
|------------------|--------------|
| <u>Id_client</u> | Clé primaire |
| <u>Id_carte</u> | Clé |

| Carte de fidélité | |
|---|-----------------------------------|
| Attributs | Désignation |
| Id_carte Date-début Date_exp Nbr_point | Int(10) Date Date Int(4) |

| Clé | Type |
|-----------------|--------------|
| <u>Id_carte</u> | Clé primaire |

| Caisse | |
|--------------------------|---------------------|
| Attributs | Désignation |
| Id_caisse Fond_caisse | Int(2) Float (6) |

| Clé | Type |
|------------------|--------------|
| <u>Id_caisse</u> | Clé primaire |

| Vendeur | |
|------------------------------------|--------------------------------------|
| Attributs | Désignation |
| Id_vend Nom_vend Prénom_vend | Int(2) Varchar(50) Varchar(50) |

| Clé | Type |
|----------------|------|
| <u>Id_vend</u> | Clé |

| Caissier | |
|--|--------------------------------------|
| Attributs | Désignation |
| Id_caissier Nom_caissier Prénom_caissier | Int(2) Varchar(50) Varchar(50) |

| Clé | Type |
|--------------------|--------------|
| <u>Id_caissier</u> | Clé primaire |

Chapitre 02 : Conception

| Remise au coffre | |
|------------------|-------------|
| Attributs | Désignation |
| Num_env | Int(5) |
| Montant_env | Float(6) |
| Date-env | Date |
| Heure_env | Datetime |
| Type_env | Varchar(20) |

| Clé | Type |
|--------------------|--------------|
| <u>Num_env</u> | Clé primaire |
| <u>Id caisse</u> | Clé |
| <u>Id caissier</u> | Clé |

| Remise | |
|-------------|-------------|
| Attributs | Désignation |
| Num_remise | Int(5) |
| Date_début | Date |
| Date_exp | Date |
| Type_remise | Varchar(10) |

| Clé | Type |
|-----------------|---------------|
| <u>Num remi</u> | Clé primaire |
| <u>Code art</u> | Clé étrangère |

| Est_vendu | |
|-----------|--------|
| Attributs | Type |
| Id_emp | Int(2) |
| Code_art | Int(4) |

| Clé | Type |
|-----------------|-------|
| <u>Id vend</u> | Index |
| <u>Code-art</u> | Index |

| Ticket_contient_article | |
|-------------------------|--------|
| Attributs | Type |
| Id_tick | Int(4) |
| Code_art | Int(4) |

| Clé | Type |
|-----------------|-------|
| <u>Id tick</u> | Index |
| <u>Code-art</u> | Index |

Conclusion

Nous avons exposé dans ce chapitre notre travail d'analyse et de conception de l'application, en faisant appel aux divers outils de modélisation UML. Notre cycle de vie se trouve à ce stade au niveau « codage ». Nous allons utiliser les informations générées dans les phases précédentes pour développer la source de cette application. Le prochain chapitre est réservé au reste des phases du cycle de vie de développement.

Chapitre 03 :

REALISATION

Introduction :

Après avoir présenté la conception de notre solution, nous entamons l'étape de la réalisation de notre Système d'encaissement au sein du NUMIDIS.

Etant la dernière étape dans le cycle de développement des systèmes d'informations, la réalisation consiste à couronner les phases précédentes, donnant un aspect tangible aux axes d'amélioration présentés lors de l'étude de l'existant et une forme concrète à la conception cela en procédant comme suit :

- Choix des outils du développement ;
- Description des différents facteurs de risques pouvant nuire le bon fonctionnement du système. Ainsi, la politique de sécurité adoptée pour faire face à ces risques ;
- Présentation de l'application via des prises d'écrans afin de figurer le travail fait et d'illustrer les grandes et principales fonctionnalités réalisées ;

1- Choix des outils de développement :

Le choix des éléments de développement nous a été proposé par les responsables du Numidis :

a) Le langage de programmation

Pour le développement de notre application, nous avons besoin d'un langage dédié aux applications web, il existe plusieurs langages script utilisés dans le développement web comme (ASP, JSP, etc.), pour le développement de notre application, nous avons choisi le PHP et Java Script pour les raisons suivantes :

➤ PHP

PHP (Hypertext PreProcessor) initialement appelé (Personnel Home Page) est un langage de programmation, sa principale application se situe au niveau de la gestion des sites web dynamiques. Les capacités de PHP ne s'arrêtent pas à la création de pages Web. Il est aussi possible de manipuler des images, de créer des fichiers PDF, de se connecter à des bases de données ou des serveurs LDAP¹, et même d'instancier des objets Java. Un module annexe lui permet également de

¹ Lightweight Directory Access Protocol.

Chapitre 03 : Réalisation

fournir des interfaces graphiques classiques (client riche, sans navigateur ou serveur Web), via GTK². PHP dispose de près de 3 000 fonctions utilisables dans des applications très variées et couvre pratiquement tous les domaines en rapport avec les applications Web. Presque tous les SGBD du marché (Systèmes de gestion de bases de données) peuvent s'interfacer avec PHP, qu'ils soient commerciaux ou qu'ils viennent du monde du logiciel libre.

PHP intègre des possibilités de connexion à la majorité des bases de données (Oracle, SQL Serveur, MySQL, dBase, ODBC, etc.), annuaires (LDAP, etc.) et systèmes de paiement en ligne (Verisign, Cybercash, Crédit Mutuel, etc.). PHP est distribué via une licence propre qui permet sa rediffusion, son utilisation et sa modification librement et gratuitement. Il peut être téléchargé depuis le site Web officiel de PHP³ ou un de ses miroirs⁴ [Daspet 2004].

Parmi les différents langages existants, PHP est le plus simple à apprendre et aussi le plus rapide à exécuter. De plus, il est portable sur de multiples plates-formes : un même script peut fonctionner sur différents serveurs et avec divers systèmes d'exploitation. Contrairement à certains langages généraux (Perl, C, Java...), PHP est uniquement dédié au développement de pages Web dynamiques. Ses applications sont donc parfaitement adaptées aux besoins d'Internet. Le code PHP est directement intégré dans les pages HTML : il est donc rapide à mettre en place et très facile à créer. Si on ajoute à ces qualités le fait que PHP soit gratuit, on comprend aisément son succès et comment il est devenu si rapidement très populaire [Defrance 2006].

➤ Java Script

JavaScript, est un langage de script très populaire, offre la possibilité d'implémenter des traitements élaborés dans des pages Web. Il peut être mis en œuvre dans toute application disposant d'un interpréteur pour ce langage.

Une fois la page web chargée en mémoire, il est possible de la manipuler avec JavaScript en se fondant sur la technologie DOM⁵. Cette dernière permet d'influer sur la structure de la page en ajoutant, modifiant ou supprimant des éléments.

² GTK est une bibliothèque permettant de créer des interfaces graphiques (GUI) très facilement [Developpez]

³ <http://www.php.net/>

⁴ <http://fr.php.net/>

⁵ Document Object Model

Chapitre 03 : Réalisation

JavaScript permet de réaliser des requêtes HTTP en recourant aux mécanismes fournis par le langage JavaScript. JavaScript n'étant pas un langage compilé, il doit être exécuté par le biais d'un interpréteur, dont plusieurs sont disponibles gratuitement tandis que d'autres sont intégrés aux navigateurs Web.

Pour constituer une application robuste, il faut faire appel au langage Ajax, Ajax (Asynchronous JavaScript And XML) permet d'exécuter des requêtes HTTP sans pour autant avoir à recharger la page qui l'exécute [Templier T 2007].

b) Le SGBD

MySQL est probablement la base de données disponible en freeware la plus utilisée. Réputé pour sa rapidité, MySQL est une base fondée sur le langage SQL.

Elle est implémentée sur un mode client-serveur, avec du côté serveur un démon MySQL et du côté client toute une variété de bibliothèques et de programmes.

MySQL est disponible en version binaire ou compilable à partir des sources, et ce pour différentes plateformes (y compris Linux et Windows). Toutes les versions sont téléchargeables à partir du site de MySQL⁶.

c) L'éditeur de texte :

L'éditeur de texte que nous avons choisi est un éditeur de texte de type (WYSIWYG), l'abréviation de (what you see is what you get), l'avantage de ce genre d'éditeur est de donner la possibilité aux développeurs de voir le design de son site avant de voir le code source.

2- L'environnement du travail

L'environnement du travail se compose de :

Système d'exploitation : nous avons travaillé sur Microsoft Windows 7 professionnel version 2009 service pack 1.

WAMP Server : c'est une plateforme utilisée pour le développement Web, elle contient le serveur Apache et le serveur MySQL.

⁶ <http://www.mysql.com/download.html>

Chapitre 03 : Réalisation

Sybase Power AMC 15 : pour dessiner les diagrammes UML des étapes « Analyse » et « Conception » nous avons utilisé le logiciel « Sybase Power AMC » à cause de sa compatibilité avec la norme UML 2 .0 que nous avons utilisé dans l'élaboration de notre projet.

Adobe DreamWeaver CS5 : c'est un logiciel dédié pour la conception des applications Web.

3- Sécurité du nouveau système :

La sécurité des systèmes d'informations est l'ensemble des moyens techniques et organisationnels mis en place pour garantir essentiellement la sécurité de l'information et du système d'information dans son ensemble. Nous devons considérer les aspects suivants :

- ✚ **Confidentialité et protection des données** : les données ne seront pas accessibles à tout le monde.
- ✚ **Sécurité des accès** : chaque utilisateur sera muni de son propre nom d'utilisateur et de son mot de passe. On veillera à ce que ce dernier soit difficile à deviner.
- ✚ **Les procédures de sauvegarde et gestion des sauvegardes** : les procédures de sauvegarde doivent être lancées périodiquement. Ainsi en cas de sinistre ou de perte de données, nous devons être à la mesure de récupérer l'ensemble des données.
- ✚ **La disponibilité** : permettant de maintenir le bon fonctionnement du système d'information et une diffusion continue de l'information en temps voulu.
- ✚ **L'Intégrité** : L'intégrité garantit que les données sont protégées contre toute modification accidentelle ou délibérée (malveillante).

4- Présentation du prototype réalisé :

Dans ce chapitre, nous présenterons quelques prises d'écran de notre Système afin de montrer une des procédures informatisées. :

- Page d'authentification (accueil) : on a prévu une authentification typique pour les utilisateurs de la caisse où ils doivent mentionner leurs postes : Responsable, arrière caissier ou caissier « figure 31 » afin de gérer les permissions de fonctionnalités ; d'où le caissier, le responsable du magasin

Chapitre 03 : Réalisation

ainsi l'arrière caissier auront dans leurs session que les fonctionnalités permises comme on le voit dans les figures 32, 33et 34 .



The screenshot shows the login interface for UNO hypermarché. At the top, there is a logo for 'UNO hypermarché' with the text 'UNO hypermarché' below it. Below the logo, there is a form with the following fields and buttons:

- Type d'utilisateur**: A dropdown menu.
- Identifiant**: A text input field.
- Mot de passe**: A text input field.
- Envoyer** and **Annuler**: Two buttons at the bottom of the form.

Figure 31 :Interface de la page d'accueil



The screenshot shows the management interface for a store manager. At the top, there is a logo for 'UNO hypermarché' and the text 'UNO Hypermarché'. Below the logo, there is a sidebar menu with the following categories and items:

- Gestion Du Magasin**
 - Magasin
 - Employes
 - Repport
- Gestion Des Caisnes**
 - Fidelité
 - Remise
 - Rayons
 - Articles
 - Caisnes
 - Tirroirs
 - Tickets
 - Clients
- modes de paiement**
 - TVA
 - Marque
 - Fournisseur
 - Type d'articles
 - famille d'article
- Gestion Du Front**
 - Encaissement

Figure 32 : interface du compte du responsable de magasin.

Chapitre 03 : Réalisation



Figure 33 : interface du compte de l'arrière caissier



Figure 34 : interface de la liste des remises sur des articles

Conclusion

Nous avons vu dans ce chapitre les outils de développements et notre environnement de travail pour réaliser notre système ainsi les aspects de sécurités pris en considération dans notre système.

Conclusion générale

A la fin de ce projet, nous sommes arrivés à concevoir et réaliser l'outil qui nous a permis d'atteindre les objectifs principaux que nous avons posés au début du projet et qui se résument dans les deux points suivants:

- ❖ Concevoir un système d'encaissement qui comporte toutes les fonctionnalités nécessaires au bon fonctionnement des hypermarchés et supermarchés de NUMIDIS particulièrement la gestion de la fidélité, gestion des remises.
- ❖ Concevoir une nouvelle base de données afin de garder l'historique des ventes.
- ❖ Améliorer la qualité des rapports, en introduisant les graphes par exemple.

Tout au début, nous avons commencé l'élaboration de notre travail par l'application d'une technique appelée le « Benchmarking » qui s'occupe de l'analyse des systèmes similaires existants dans le monde entier afin de profiter de leurs fonctionnalités, pour notre cas on a analysé le présent système de NUMIDIS, ensuite nous avons procédé à la conception et la réalisation de notre outil qui gère les acteurs du domaine de travail.

Nous avons essayé de construire un système simple en termes d'IHM pour satisfaire les employés non informaticiens.

Notre système peut être amélioré en lui intégrant par la suite, les éléments suivants :

- 1- Le passage vers le mode dégradé afin d'éviter l'arrêt de travail au cas d'un passage en panne.
- 2- Intégrer les périphériques de caisses (scanner, imprimantes, tiroir).
- 3- Relier notre système avec le système central de NUMIDIS.

ANNEXE

Annexe 1 : Etude comparative entre les processus de développement

Introduction

Méthodologie = Formalisme + Processus+ Outil

Un processus définit une séquence d'étapes, en partie ordonné, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

Pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles, On découpe le processus en deux axes :

- ✓ L'axe de développement technique, qui se concentre principalement sur la qualité de production,
- ✓ L'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais.

1) Processus en cascade

Propose de citer les phases du projet de façon séquentielle suivant le schéma suivant :

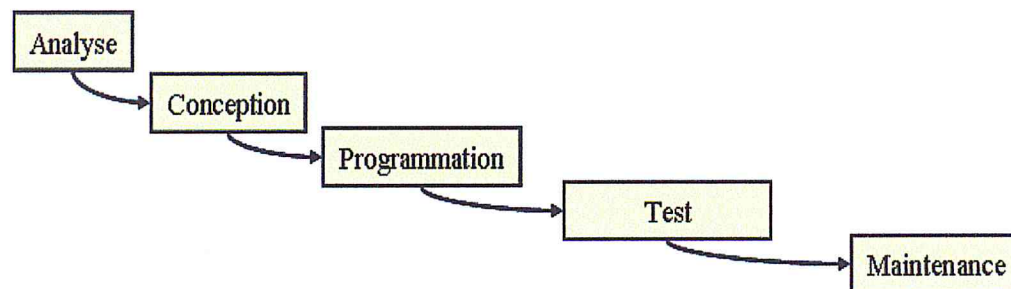


Figure 35: Le processus en cascade

Points forts

- ✓ Distingue clairement les phases du projet,
- ✓ Bien adapté lorsque les besoins sont clairement identifiés et stables,
- ✓ Enchaînement depuis le cahier des charges jusqu'à la réalisation.

Points faibles

- ✓ Non itératif,
- ✓ Ne propose pas de modèles de document,
- ✓ Retour en arrière limité,
- ✓ Mauvaise gestion des modifications et des erreurs.

2) Le processus en V

Ce modèle est adapté aux projets de taille et de complexité moyenne. C'est une amélioration du modèle en cascade traditionnel. Il permet d'identifier et d'anticiper très tôt les éventuelles évolutions des besoins. C'est aussi un moyen de vérifier de la maturité des utilisateurs, car s'il en était autrement, ils se trouveraient dans l'incapacité de fournir des tests de recettes dès la phase de spécification.

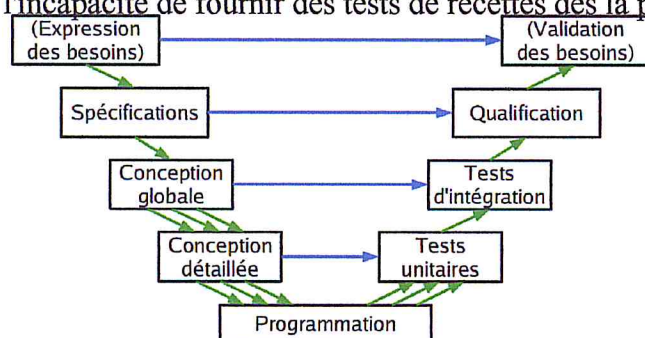


Figure 36: Cycle en V

Points fort

- ✓ Modèle avantageux pour une maîtrise d'œuvre,
- ✓ Enchaînement depuis le cahier des charges jusqu'à la réalisation,
- ✓ Bien adapté lorsque les besoins sont clairement identifiés et stables,
- ✓ Rassurant pour une maîtrise d'ouvrage.

Points faible

- ✓ Il doit s'engager significativement,
- ✓ Pas assez de résultats intermédiaires pour valider la version finale du produit,
- ✓ Trop de choses reportées à l'étape de programmation (l'interface utilisateur),
- ✓ difficile de prendre en compte des changements importants dans les spécifications dans une phase avancée du projet,
- ✓ durée parfois trop longue pour les produits compétitifs,
- ✓ Retour limité à une phase en amont.

3) Le processus XP (extrême programming)

Ensemble de « Bests Practices » de développement (travail en équipes, transfert de compétences...), Cible des projets de moins de 10 personnes.

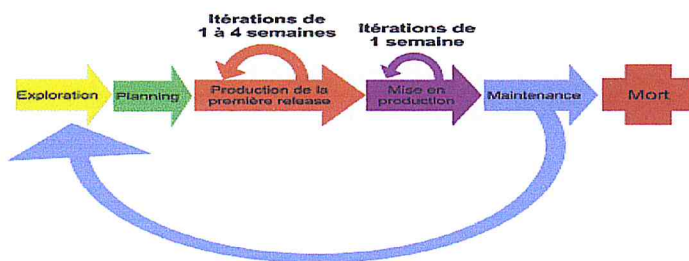


Figure 37: Le processus XP

Points forts

- ✓ Itératif,
- ✓ Simple à mettre en œuvre,
- ✓ Fait une large place aux aspects techniques : prototypes, règles de développement, tests etc.,
- ✓ Innovant: programmation en duo, kick-off matinal debout etc.

Points faibles

- ✓ Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration,
- ✓ Élude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire,
- ✓ Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ?,
- ✓ Risque d'avoir un code pas assez documenté.

4) Le processus 2TUP (two track unified process)

Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. Ce processus suit deux chemins, (Architecture fonctionnelle, Architecture technique.), Il s'articule autour de l'architecture, et il propose un cycle de développement en Y, il est bien détaillé dans « UML en action », et il cible des projets de toutes tailles.

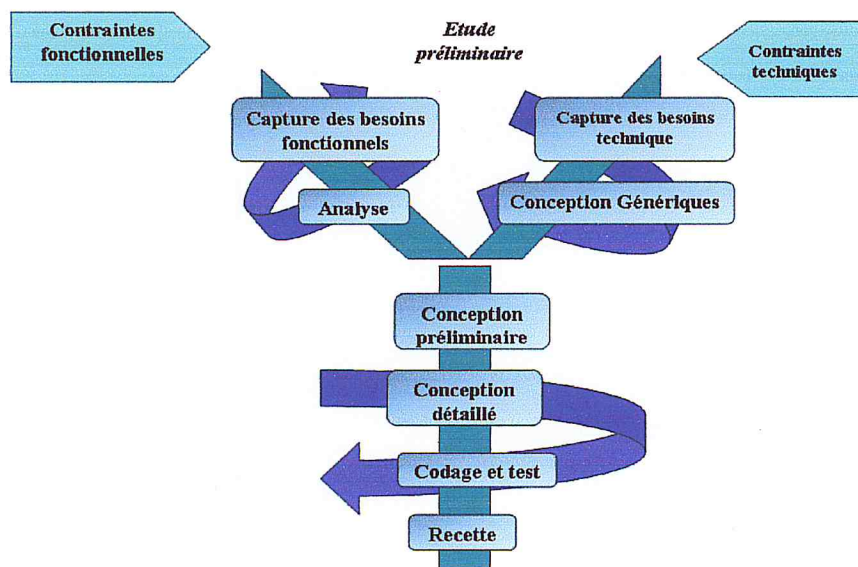


Figure 38: Le processus 2TUP

Points forts

- ✓ Itératif,
- ✓ Fait une large place à la technologie et à la gestion du risque,
- ✓ Définit les profils des intervenants, les livrables, les plannings, les prototypes.

Points faibles

- ✓ Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement etc.,
- ✓ Ne propose pas de documents types.

5) Le processus UP (processus unifier)

Un processus unifié est un processus de développement logiciel construit sur UML. Il est :

« Itératif » (le processus de développement est appliqué plusieurs fois) et « Incrémental » (chaque itération augmente la quantité d'information), il est centré sur l'architecture, il est conduit par les cas d'utilisation et piloté par les risques.

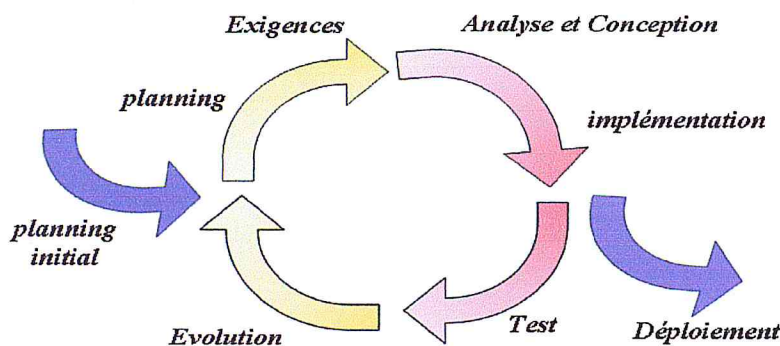


Figure 39: Le processus UP

Points forts

- ✓ Laisse tenir compte des changements d'exigence très courants en gestion de projet,
- ✓ Permet d'accélérer le rythme de développement grâce à des objectifs clairs à court terme,
- ✓ Les éléments sont intégrés progressivement et non pas en fin de cycle (longue étape, incertaine décomposée en 6 à 9 plus petites intégrations impliquant peu d'éléments),
- ✓ Permet de limiter les risques de retard car identification des problèmes dès les premiers stades de développement (c.à.d. les premières itérations sont déroulées et vous examinez tous les composants du processus au fur et à mesure).

Points faibles

- ✓ Destiné pour les projets de très grande taille.

6) Le processus RUP (Rational UnifiedProcess)

Ce processus est promu par rational, il est considéré à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web), il cible des projets de plus de 10 personnes.

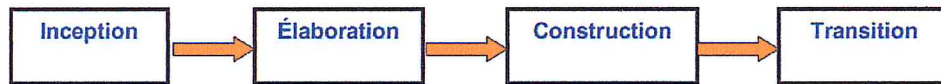


Figure 40: Le processus RUP

Points forts

- ✓ Itératif,
- ✓ Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes etc.,
- ✓ Propose des modèles de documents, et des canevas pour des projets types.

Points faibles

- ✓ Coûteux à personnaliser : batterie de consultants,
- ✓ Très axé processus, au détriment du développement : peu de place pour le code et la technologie,
- ✓ Fait tout, mais lourd,
- ✓ Parfois difficile à mettre en œuvre de façon spécifique,
- ✓ Vision non évidente ni immédiate.

7) Le cycle en spirale

Ce modèle de cycle de vie tient compte de la possibilité de réévaluer les risques en cours de développement, il emprunte au prototypage incrémental mais lui adjoint une dimension relevant de la prise de décision managériale et non purement technique. Il couvre l'ensemble du cycle de développement d'un produit. Il met l'accent sur l'activité d'analyse des risques.

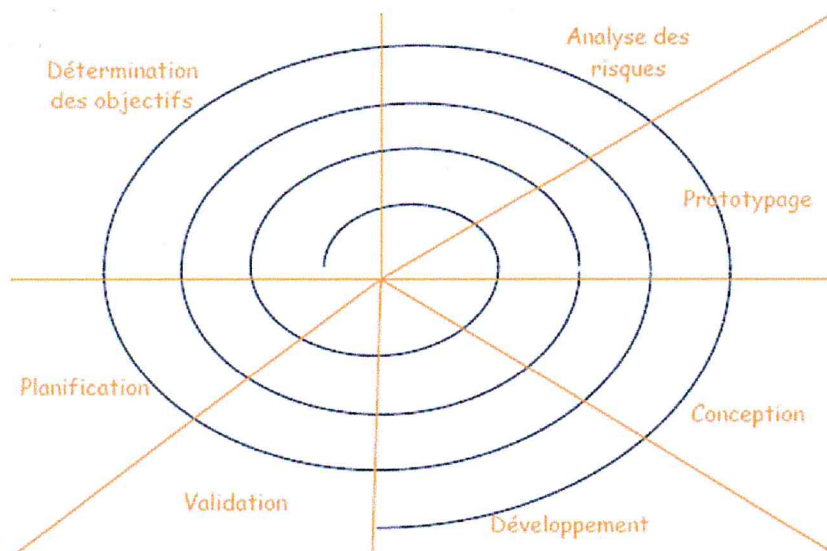


Figure 41: Le cycle en spirale

Points forts

- ✓ Utilisation du prototypage,
- ✓ Analyse (progressive) des risques,
- ✓ taux de défaut susceptible d'être généré.

Points faibles

- ✓ Il ne peut pas être soumis à un planning bien défini,
- ✓ Il ne donne pas la possibilité de modification.



8) Le cycle incrémental

Dans les modèles par incrément un seul ensemble de composants est développé à la fois : des incréments viennent s'intégrer à un noyau de logiciel développé au préalable. Chaque incrément est développé selon l'un des modèles précédents.

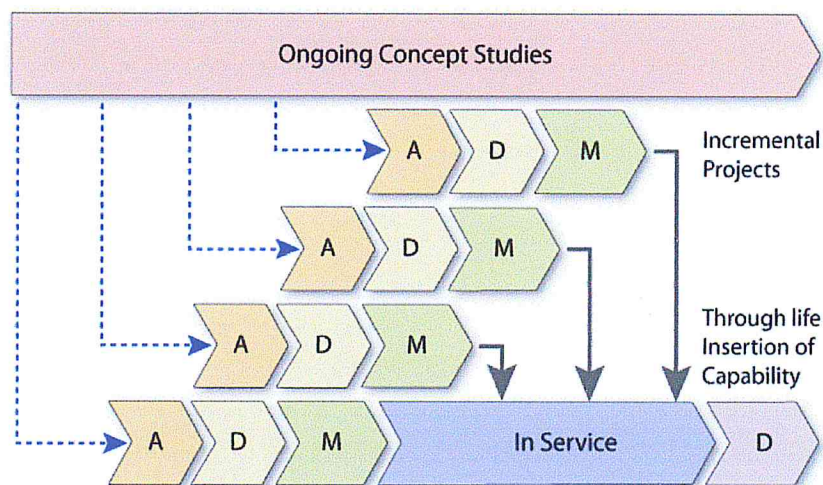


Figure 42: Le modèle incrémental

Points forts

- ✓ Chaque développement est moins complexe,
- ✓ Les intégrations sont progressives,
- ✓ Il est ainsi possible de livrer et de mettre en service chaque incrément,
- ✓ Il permet un meilleur lissage du temps et de l'effort de développement grâce à la possibilité de recouvrement (parallélisations) des différentes phases.

Points faibles

- ✓ Remettre en cause les incréments précédents ou pires le noyau,
- ✓ Ne pas pouvoir intégrer de nouveaux incréments.

Conclusion

Notre projet est de taille moyenne (élimination de la méthode UP), et il ne demande pas de nombreux développeurs pour le réaliser (élimination de la méthode XP), de plus, il ne contient pas beaucoup de risques qui peuvent être la source de son échec (élimination des méthodes incrémentale et spirale), mais il demande des fois le retour arrière pour effectuer certaines modifications et corrections (élimination des méthodes en cascade et en V).

En outre, il n'est pas complexe, ses besoins sont bien définis et ses objectifs sont très clairs, mais il demande une technologie web élevée (élimination de la méthode RUP). Pour ces raisons, la démarche ou le cycle de développement le plus adéquat pour notre projet est le modèle 2TUP, car c'est le plus avantageux, le moins coûteux et ses inconvénients n'influencent pas sur les résultats grâce à la nature des besoins et des objectifs du projet.

Annexe 2 : Benchmarking

Première définition

Le terme de « Benchmarking » signifie « Trouver, au niveau mondial, l'entreprise ou les entreprises qui réalisent de la manière la plus performante un processus ou une tâche donnée, d'aller l'étudier (« benchmarker ces entreprises ») et d'adapter ensuite ce processus à sa propre entreprise.

Pour les entreprises, il s'agit de se comparer aux « leaders » qui se positionnent sur le marché, de s'inspirer de leurs idées, de leurs pratiques, de leurs fonctionnements et de leurs expériences afin que les pratiques en internes s'améliorent [Bench 2003].

Deuxième définition

Le Benchmarking est un formidable levier d'innovation dans l'entreprise. C'est à la fois une méthode d'analyse permettant de s'étalonner en s'inspirant des meilleurs points, et à la fois un état d'esprit, un style de management. Le Benchmarking peut s'effectuer en interne entre les services, ou les entités ; comme en externe, avec ses différents fournisseurs, distributeurs, concurrents, produits du marché.

Le principe est de comparer différents critères objectifs ou subjectifs, mais du moins quantifiables avec des règles que l'on peut établir afin d'avoir les mêmes systèmes d'analyse entre les différents membres qui auront à réaliser le benchmark. Le but étant de déterminer les critères sur lesquels on n'a pas besoin d'investir car on les satisfait mieux que les autres, ainsi que ceux sur lesquels un travail d'amélioration est nécessaire pour arriver à l'optimum¹ [Erwan 98].

Les types du Benchmarking

Il existe plusieurs types de Benchmarking utilisés selon les besoins de l'utilisateur, ces types sont :

Benchmarking interne

¹(D'Après A. Evans, "Why you should Benchmark - Your Way to Business Best Practice" , Benchmarking Link-Up Australia, <http://www.ozemail.com.au>, 1998)

Il vise à comparer les processus, produits ou services appartenant à la même entreprise.

Benchmarking concurrentiel

Il vise à comparer une entreprise au meilleur de ses concurrents sur le marché.

Benchmarking générique

Il vise à comparer des entreprises appartenant à des secteurs d'activités différents mais qui ont des processus similaires.

Benchmarking fonctionnel

Il vise à comparer une fonction générique de valeur ajoutée et commune à des entreprises non concurrentes mais appartenant à un même secteur d'activités.

Benchmarking processus

Il vise à mettre en évidence, pour chaque entreprise engagée et généralement reconnue comme leader dans son secteur d'activités, la spécificité de certaines opérations de son processus critique.

Benchmarking stratégique

Il vise à recueillir les meilleures pratiques des entreprises le plus souvent concurrentes avec une mise en évidence des objectifs stratégiques associés à ces pratiques.

Benchmarking organisationnel

Il vise à faire évoluer certaines activités qui ont une grande incidence sur l'organisation de manière à rendre celle-ci mieux adaptée à un contexte de compétitivité[Vaisman 1999].

Conclusion

D'après notre étude, nous avons réussi à comprendre le sens du mot « Benchmarking » ainsi que ses différents types.

Nous pouvons déduire en analysant les différents types du Benchmarking, que nous allons appliquer les Benchmarking fonctionnel à notre projet, car il s'agit de voir des projets similaires dans un contexte non concurrentiel.

Bibliographie :

| | |
|-------------------|---|
| [Winstore3.0] | Manuelle d'utilisation de Winstore 3.0 « Document interne » |
| [Roque 2002] | Roque P., Vallée F., « UML 2 en action, de l'analyse des besoins à la conception », 4 ^{ème} édition, EYROLLES, février 2002. |
| [Graïne 2009] | GRAÏNE S., « UML 2 pour une modélisation orientée objet », les éditions l'Abeille, 2009. |
| [Daspet 2004] | Daspet E., De Geyer C-P., « PHP5 avancé », édition EYROLLES 2004. |
| [Templier T 2007] | Templier T., Gougeon A., « JavaScript pour le web 2.0 », édition, EYROLLES, 2007. |
| [Defrance 2006] | DEFRANCE J-M., « PHP/MySQL avec dreamweaver 8 », les éditions EYROLLES 2006. |
| [Bench 2003] | www.3ie.fr |
| [Erwan 1998] | http://erwan.neau.free.fr/innovation_benchmarking.htm |
| [Vaisman 1999] | http://ovaisman.online.fr |
| [iFranche] | http://ctresbon.ifranche.com |
| [Developpez] | http://www.developpez.com |