

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la Communication (STIC)

Conception et réalisation d'une application Web de gestion des achats e-purchasing basée sur l'architecture JEE

Préparé par : Bouanane Zina
Sedrati Samira

Soutenue devant le jury

Encadré par Hedjaz Sabrine
Président Boubakir Mohamed
Examineur Boufaghes Hamida

Grade M.A.B
Grade M.A.A
Grade M.A.B

Année Universitaire : 2015/2016

Remerciement



*Tout d'abord nous tenions à remercier Allah de nous avoir donné le courage
la force et la volonté, de terminer ce modeste travail.
durant les années d'études.*

*Nous exprimons nos remerciements à notre encadreur Madame « HEDJAZ SEBRINE »
pour l'assistance qu'elle nous a témoignée, pour sa disponibilité, pour ces orientations et
conseils sans lesquels ce travail ne verrait pas le jour, qu'elle trouve ici l'expression de
notre gratitude.*

*Nous remercions également les membres de notre jury monsieur « BOBAKIR
MOHEMED » et madame « BOUFAGHASE HAMIDA » pour
l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre
travail Et de l'enrichir par leurs propositions.*

*Ainsi, nous adressons nos vifs remerciements à Monsieur DJEMEL ET Ishak pour son
aide et ses précieux conseils.*

*Enfin, une pensée particulière à toute personne ayant de près ou de loin poussé à donner
le coup de rein final pour boucler ce mémoire et tourner cette page de notre vie...*

Zina & Samira





Dédicace

A mes très chers parents « Zebaida & Boukhemis »

*Pour tout l'amour d'ont vous m'avez entouré, pour tout ce que vous avez fait
pour moi.*

*Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec
l'espoir de ne jamais vous décevoir.*

*Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de Vos
prier quotidiennes.*

A mes très chers frères, sœurs et mes tantes

« Wassila , Naima, Farah, Fadila , Boubker, Amine , Youcef , Rafia , Saida »

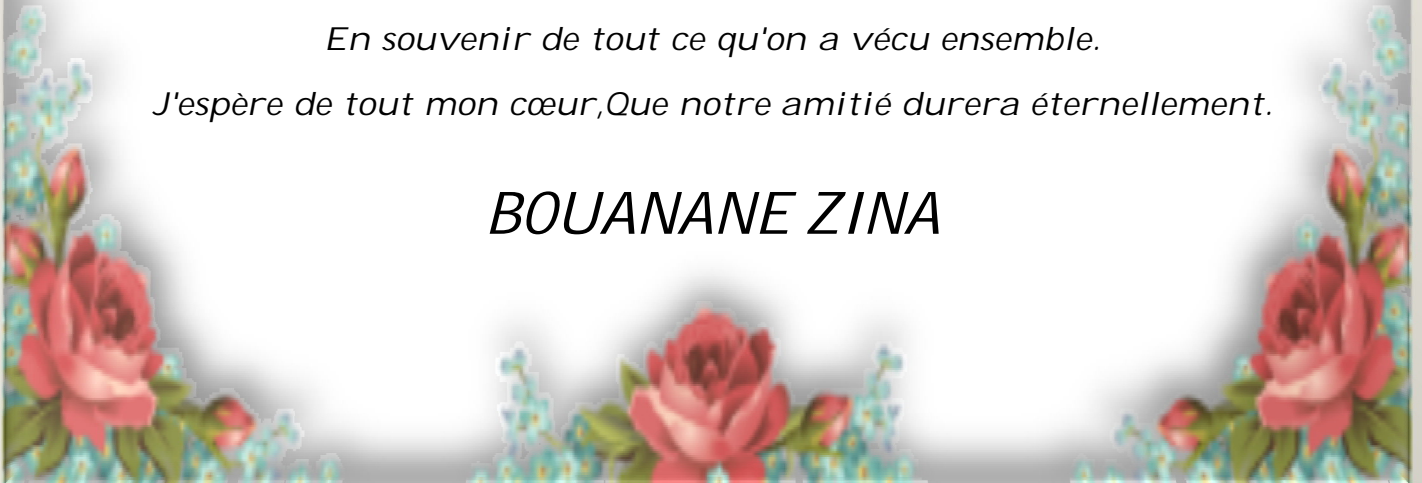
*Vous occupez une place particulière dans mon cœur. Je vous dédie ce
travail en vous souhaitant un avenir radieux, plein de
bonheur et de succès. Surtout Bouhami Widad*

En souvenir de nos éclats de rire, des bons moments et nuits blanches.

En souvenir de tout ce qu'on a vécu ensemble.

J'espère de tout mon cœur, Que notre amitié durera éternellement.

BOUANANE ZINA



Dédicace



A mes très chers parents «Atika & Khalil»

Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices.

Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait

d'incroyable pour moi.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

A mes très chers frère, sœurs «Aïssam ,Adel, Aziza , Lamia »

Aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous.

Je vous dirais tout simplement, un grand merci, je vous aime.

A mes très chers ami(e)s «Asma, Zina, Farah»

En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble.

SEDRATI SAMIRA



Table de matières

Titre	Page
Introduction générale	2
CHAPITRE I : GENERALITES	
1. Introduction	5
2. Le processus 2TUP	5
2.1. Définition	5
2.2. Schéma général de processus 2TUP	5
3. Les applications web	6
3.1. Définition d'une application web	6
3.2. Pourquoi créer une application Web ?	6
4. java EE	7
4.1. Définition	7
4.2. Les couches de base d'une architecture java EE	8
4.3. La plate-forme java EE	8
4.3.1. Les composants Web	9
4.3.2. Les composants métiers	10
4.3.3. Les services d'infrastructures et de communication	10
4.4. Les conteneurs	11
4.5. Modèle MVC pour java EE	11
4.5.1. Modèle MVC de type1	12
4.5.2. Modèle MVC de type2	13
4.6. Modèle DAO (Data Access Object)	14
5. Les Frameworks	15
5.1. Framework Spring	15
5.2. Framework Hibernate	16
6. Les avantages des applications java EE	17
7. Conclusion	17
CHAPITRE II : ETUDE PRELEMINAIRE	
1. Introduction	19
2. Elaboration du cahier de charge	20
2.1. Présentation du projet	20
2.2. Grands choix techniques	20
2.3. Recueil des besoins fonctionnels	21
2.4. Recueil des besoins opérationnels	21
3. Description de contexte du système	21
3.1. Identification des acteurs	21
3.2. Identification des messages	23
3.3. Modélisation de contexte	24
4. Conclusion	26
CHAPITRE III: CAPTURE DES BESOINS FONCTIONNELS ET TECHNIQUE	
1. Introduction	28
2. Capture des besoins fonctionnels	28
2.1. Déterminer les cas d'utilisations	29
2.2. Description détaillée des cas d'utilisation	33
2.2.1. Description textuelle des cas d'utilisation	33
2.2.2. Description graphique des cas d'utilisation	46
2.3. Identification des classes candidates	65
2.3.1. Définition	65
2.3.2. La liste des classes candidates	66
2.3.3. Responsabilités des classes	67
2.4. Diagramme de classes participantes	69
3. Capture des besoins techniques	80

3.1. Spécification techniques du point de vue matériel	81
3.2. Spécification d'architecture	82
3.2.1. Spécification d'organisation du modèle déploiement SGAE	83
3.3. Elaboration du modèle de spécification logicielle	83
3.3.1. Modèle de spécification logicielle	84
3.3.2. Description textuelle des cas d'utilisation technique	85
3.3.3. Organisation en couche du modèle de spécification	85
4. Conclusion	87
CHAPITRE IV : ANALYSE	
1. Introduction	89
2. Découpage en catégorie	89
2.1. Répartition des classes candidates en catégories	90
2.2. Elaboration de diagramme de classe préliminaires pour chaque catégorie	91
2.3. Dépendance entre les catégories	92
3. Développement du modèle statique	92
3.1. Affiner les classes	93
3.2. Affiner les associations	93
3.3. Ajouter les attributs	93
3.4. Ajouter les opérations	93
3.5. Diagramme de classe détaillé pour chaque catégorie	94
4. Développement de modèle dynamique	95
4.1. Diagramme de séquence	95
5. Conclusion	110
CHAPITRE V : CONCEPTION	
1. Introduction	112
2. Conception générique	112
2.1. Définition	112
2.2. Elaboration du modèle logique	113
2.2.1. Définition d'un Framework technique	113
2.2.2. Organisation des Frameworks techniques	113
3. Conception préliminaire	115
3.1. Définition	115
3.2. Développement du modèle de déploiement	115
4. Conception détaillé	116
4.1. Définition	116
4.1.1. Concevoir les attributs et les opérations	116
4.1.2. Le diagramme de classe	118
4.1.3. Concevoir les Associations	118
4.2. Les classes modifiées	121
4.3. Le Modèle de base de données	122
5. Conclusion	122
Chapitre VI : IMPLEMENTATION	
1. Introduction	124
2. Langage de programmation et choix techniques	124
2.1. Java	124
2.2. Les API java EE	124
2.3. Les standards web	125
3. Outils et environnement de développement	126
4. Présentation de quelque interface de notre application	127
5. Conclusion	130
Conclusion générale	132

Liste des figures

CHAPITRE I : GENERALITES		
Figure I-1	schéma général de processus 2TUP	5
Figure I-2	L'architecture java EE	7
Figure I-3	Les composants de la plate-forme java EE	9
Figure I-4	le modèle MVC	12
Figure I-5	le modèle MVC I	13
Figure I-6	le modèle MVC II	14
Figure I-7	La couche DAO	14
CHAPITRE II : ETUDE PRELEMINAIRE		
Figure II-1	La phase d'étude préliminaire	19
Figure II-2	Le résumé des activités et des produits de l'étude préliminaire	20
Figure II-3	Le diagramme de contexte dynamique du système SGAE	24
CHAPITRE III: CAPTURE DES BESOINS FONCTIONNELS ET TECHNIQUE		
Figure III-1	L'étape de capture des besoins	28
Figure III-2	La démarche de capture des besoins fonctionnels	29
Figure III-3	Le diagramme de cas d'utilisation	32
Figure III-4	La démarche de capture des besoins techniques	81
Figure III-5	Configuration matérielle de système SGAE	82
Figure III-6	Architecture 3 tiers	83
Figure III-7	Modèle de spécification logicielle initial	84
Figure III-8	Organisation du modèle de spécification logicielle	86
CHAPITRE IV : L'ANALYSE		
Figure IV-1	L'étape d'analyse	89
Figure IV-2	Découpage en catégories	90
Figure IV-3	Modèle structurel d'analyse	92
Figure IV-4	La démarche d'élaboration du modèle statique	92
Figure IV-5	La démarche d'élaboration du modèle dynamique	95
Figure IV-6	Les diagrammes de séquences du système SGAE	110
CHAPITRE V : CONCEPTION		
Figure V-1	L'étape de la conception	112
Figure V-2	Organisation du modèle logique de SGAE	114
Figure V-3	Le modèle de déploiement du SGAE	115
Figure V-4	La conception des attributs et des opérations	117
Figure V-5	Le diagramme de classe	118
Figure V-6	La transformation des associations	120
Figure V-7	Les classes modifiées	121
CHAPITRE VI : IMPLEMENTATION		
Figure VI-1	Fenêtre d'accueil	127
Figure VI-2	Fenêtre d'inscription	127
Figure VI-3	Fenêtre de panier	128
Figure VI-4	Fenêtre liste des commandes	128
Figure VI-5	Fenêtre d'authentification de l'administrateur	129
Figure VI-6	Fenêtre de gestion des articles	129
Figure VI-7	Fenêtre d'ajout d'un article	130
Figure VI-8	Fenêtre voir les statistiques	130

Liste des tableaux

CHAPITRE II : ETUDE PRELEMINAIRE		
Tableau II-1	Les acteurs et leurs rôles	22
Tableau II-2	Légende des messages de diagramme de contexte	25
CHAPITRE III: CAPTURE DES BESOINS FONCTIONNELS ET TECHNIQUE		
Tableau III-1	La liste des cas d'utilisation du système SGAE	30
Tableau III-2	La liste des acteurs et des messages par cas d'utilisation du système SGAE	31
Tableau III-3	Description textuelle des cas d'utilisation fonctionnels du système SGAE	46
Tableau III-4	Diagramme de séquence système des cas d'utilisation fonctionnels du système SGAE	56
Tableau III-5	Diagramme d'activité des cas d'utilisation fonctionnels du système SGAE	65
Tableau III-6	Les classes candidates et ses responsabilités	69
Tableau III-7	Les diagrammes de classes participantes pour tous les cas d'utilisation	80
Tableau III-8	Description textuelle des cas d'utilisation techniques	85
CHAPITRE IV : ANALYSE		
Tableau IV-1	Les diagrammes de classe préliminaire des catégories	91
Tableau IV-2	Les diagrammes de classe détaillée des catégories	94

Introduction générale

Actuellement, le monde connaît une avance technologique considérable dans tous les domaines, grâce à l'informatique qui s'est imposée d'une manière très impressionnante dans les entreprises, cela est dû à son apport extraordinaire dans le développement du travail et le domaine de la gestion des bases de données.

L'informatique est de plus en plus utilisée dans tous les domaines d'activité y compris celui de la gestion des achats en ligne auquel nous rattacherons d'ailleurs notre étude de master. L'achat en ligne est une solution informatique destinée à optimiser et à rationaliser les fonctions d'achat, et cela pour des meilleurs services comme la gestion des commandes d'achats et des fonctions administratives avancées aussi bien aux clients qu'aux vendeurs, permettant ainsi d'améliorer l'efficacité des opérations et la réduction des coûts.

L'objectif de notre projet est la conception et la réalisation d'une application web pour la gestion des achats en ligne, et en même temps l'adoption d'une architecture orientée entreprise appelée JEE (Java Enterprise Edition).

Java EE est une architecture indépendante qui permet de développer, déployer et gérer des applications d'entreprise multi tiers en ajoutant la possibilité de fournir une architecture complète, stable, sécurisée et rapide. **L'architecture JEE** permet un découpage de l'application et donc une séparation des rôles lors du développement, ainsi que la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés, afin de permettre par la suite que le produit final puisse être évolutif et facilement maintenable.

Pour réaliser ce travail, nous allons utiliser le processus 2TUP basé sur la notation UML comme méthode de conception. Pour l'implémentation, on utilise le langage JAVA. pour la programmation Web (Jsp, Servlet, Java Beans, le modèle MVC 2), avec le SGBD MYSQL.

Notre projet est réparti en six chapitres et une conclusion générale:

Le chapitre 01 : « Généralités » Ce chapitre sera consacré à la présentation du processus 2TUP, des applications Web, de l'architecture JEE, ainsi que les modèles et les techniques utilisés.

Le chapitre 02 : « Etude préliminaire » Ce chapitre consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en donnant une version textuelle préliminaire du

cahier des charges. Après avoir identifié les acteurs qui interagiront avec le système, nous développerons un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

Le chapitre 03 : « Capture des besoins » Ce chapitre comporte deux étapes : la capture des besoins fonctionnels et celle des besoins techniques. La phase de capture des besoins fonctionnels formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire, en donnant une description textuelle et une autre graphique pour chaque cas d'utilisation. La capture des besoins techniques 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 chapitre 04 : « Analyse » Ce chapitre comporte les étapes de découpage en catégories, de développement du modèle statique et développement du modèle dynamique. Le découpage en catégories consiste de découper le modèle UML en blocs logiques les plus indépendants possibles. Le développement du modèle statique va nous permettre d'illustrer les principales constructions du diagramme de classes UML durant l'étape d'analyse. Le développement du modèle dynamique va nous permettre d'illustrer comment décrire des scénarios mettant en jeu un ensemble d'objets échangeant des messages.

Le chapitre 05 : « Conception » Ce chapitre comporte la conception générique, la conception préliminaire et la conception détaillée. La conception générique consiste à développer la solution qui répond aux spécifications techniques que nous avons présentées au chapitre 3. Dans la phase de conception préliminaire on s'effectue la fusion des études fonctionnelles et techniques. La conception détaillée consiste à construire et à documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution.

Le chapitre 06 : « Implémentation » Dans ce chapitre on donne une description de l'application, les technologies de programmation utilisées, ainsi que les interfaces graphiques de l'application.

Conclusion générale : La conclusion générale résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Chapitre I

Généralités

1. Introduction

Dans le cadre de ce chapitre, nous allons définir quelques généralités portant sur les méthodes et les techniques mettant en évidence la réalisation de notre projet. Nous allons commencer par présenter le processus 2TUP (2 Track Unified Process) pour la modélisation unifiée UML (Unified Modeling Language), définir les applications web et enfin nous allons présenter les principaux concepts de la technologie java EE (Java Enterprise Edition).

2. Le processus 2TUP

2.1. Définition :

2TUP est un processus de développement logiciel qui implémente le processus unifié. Il propose un cycle de développement en Y, qui dissocie les aspects techniques et les aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire et les messages échangés entre les deux. Cette étape consiste aussi à modéliser le contexte et de générer le cahier des charges. [W0]

2.2. Schéma général de processus 2TUP :

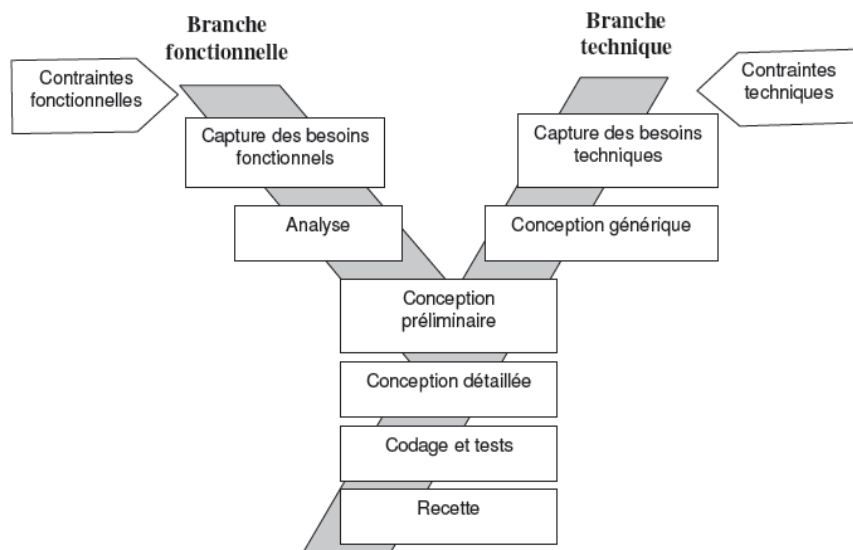


Figure I-1 : schéma général de processus 2TUP

✓ Branche fonctionnelle (gauche) :

➤ **Capture des besoins fonctionnels** : recensement des besoins liés au métier que les

utilisateurs veulent qu'ils soient pris en compte dans le système. Ces besoins sont capturés à l'aide des diagrammes de cas d'utilisation.

- **Analyse** : étude des besoins des utilisateurs et leur modélisation. On utilise essentiellement les diagrammes de classes et le diagramme de séquence. [W01]

✓ Branche technique (droite) :

- **Capture des besoins techniques** : recensement des besoins à caractère technique non lié au métier des utilisateurs. Ces besoins sont capturés à l'aide des diagrammes de composants et de déploiement.
- **Conception générique** : proposition de solution technique indépendante du métier. [W01]

✓ Phase de réalisation :

- **Conception préliminaire** : croisement de l'analyse et de l'architecture générique du système.
- **Conception détaillée** : approfondissement de la conception (typage de données, détail des algorithmes, etc.)
- **Codage et tests** : création des bases de données, implémentation des programmes, dans des environnements.
- **Recette** : remise du système aux utilisateurs finaux. [W01]

3. Les applications web

3.1. Définition d'une application web :

Une application web est un logiciel hébergé sur un serveur, accessible depuis n'importe quel navigateur web via un réseau informatique (internet, etc.) d'une façon de faciliter l'accomplissement d'une tâche précise sur le Web.

3.2. Pourquoi créer une application Web?

La création d'application web présente ainsi de nombreux avantages :

- ✓ L'accès est universel : à partir d'un navigateur web, tel que FireFox , Chrome, Internet Explorer. Plus besoin donc d'assurer un développement et un déploiement sur de multiples systèmes d'exploitation.
- ✓ Il n'est plus nécessaire d'installer un logiciel sur chaque poste, ce qui diminue en grande partie les frais de maintenance et élimine certaines incompatibilités.
- ✓ Une application bien conçue peut être utilisée par différents types de terminaux (ordinateurs, laptops, tablettes, smartphones, etc.).

- ✓ Ces applications sont capables de gérer des données multimédias (textes, photos, vidéos, etc.).
- ✓ Certaines parties de ces applications sont ouvertes aux clients et fournisseurs et facilitent ainsi les échanges d'information entre les partenaires
- ✓ Pour mettre à jour l'application, il suffit de modifier l'application sur le serveur et tous les postes accèdent instantanément à la nouvelle version.

4. java EE

4.1. Définition :

Java EE est une norme élaborée par la société SUN plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveau, basé sur des composants, s'appuyant sur le langage JAVA.

La première version des spécifications de java EE fut publiée en 1999, la version 1.3 apparut en 2001, puis la version 1.4 en 2003 et la version 1.5 (renommée Java EE 5) en 2007. Depuis septembre 2014, la dernière version en cours est Java EE 8.

Java EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Les applications java EE possèdent une architecture où plusieurs composants applicatifs interviennent, une telle application est organisée sous forme de plusieurs couches interconnectées que l'on appelle « tiers ». L'architecture d'une application se découpe en au moins trois tiers. La partie cliente, la partie métier et la partie de données.

La figure suivante résume l'architecture java EE.

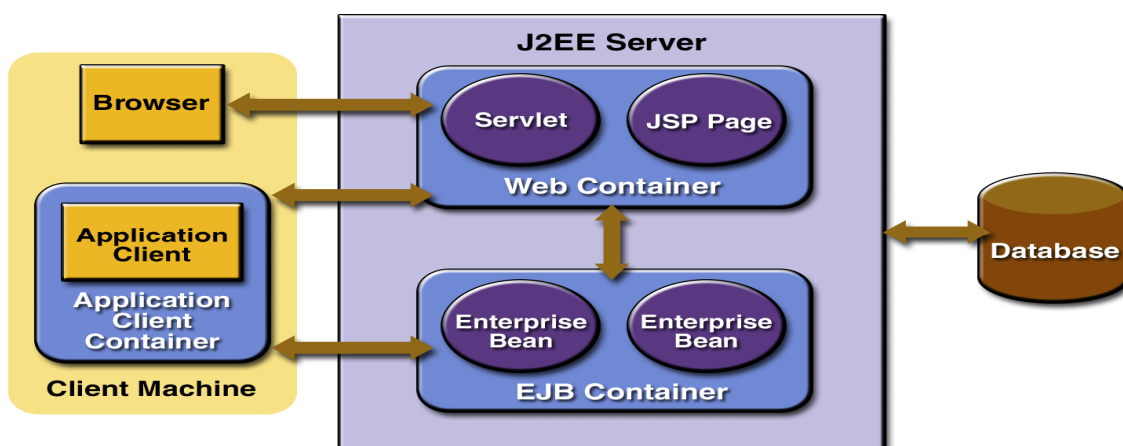


Figure I -2 : L'architecture java EE

4.2. Les couches de base d'une architecture java EE:

Java EE est utilisé principalement dans le cadre du développement de grosses applications distribuées. Ces applications sont développées dans une architecture de type client/serveur. L'application est structurée en couches, chacune des couches correspondant à une fonctionnalité propre et étant implémentée sur l'un ou l'autre des postes (client ou serveurs).

- **Couche présentation** : correspond à la partie de l'application visible et interactive avec les utilisateurs. Elle relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour, lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche métier. [W3]
- **Couche métier** : correspond à la partie fonctionnelle de l'application, décrivant les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, aussi elle offre des services applicatifs et métiers à la couche présentation. Pour fournir ces services, elle s'appuie sur les données du système, accessibles au travers des services de la couche d'accès aux données. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés. [W3]
- **Couche accès aux données** : elle consiste de la partie gérant l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système. [W3]

4.3. La plate-forme java EE:

Java EE est une norme qui va spécifier à la fois l'infrastructure de gestion des applications et les APIs (Application Program Interfaces) des services utilisés pour concevoir ces applications. Les spécifications du serveur d'applications, c'est-à-dire de l'environnement d'exécution : java EE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'applications conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services. [W02]

Ces services sont des extensions JAVA indépendants, permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée java EE SDK (J2EE Software Development Kit). Les différents services et/ou fonctionnalités associées à J2EE sont classés comme suit :

- Les composants Web.
- Les composants métiers.

- Les services d'infrastructure et de communication. [W02]

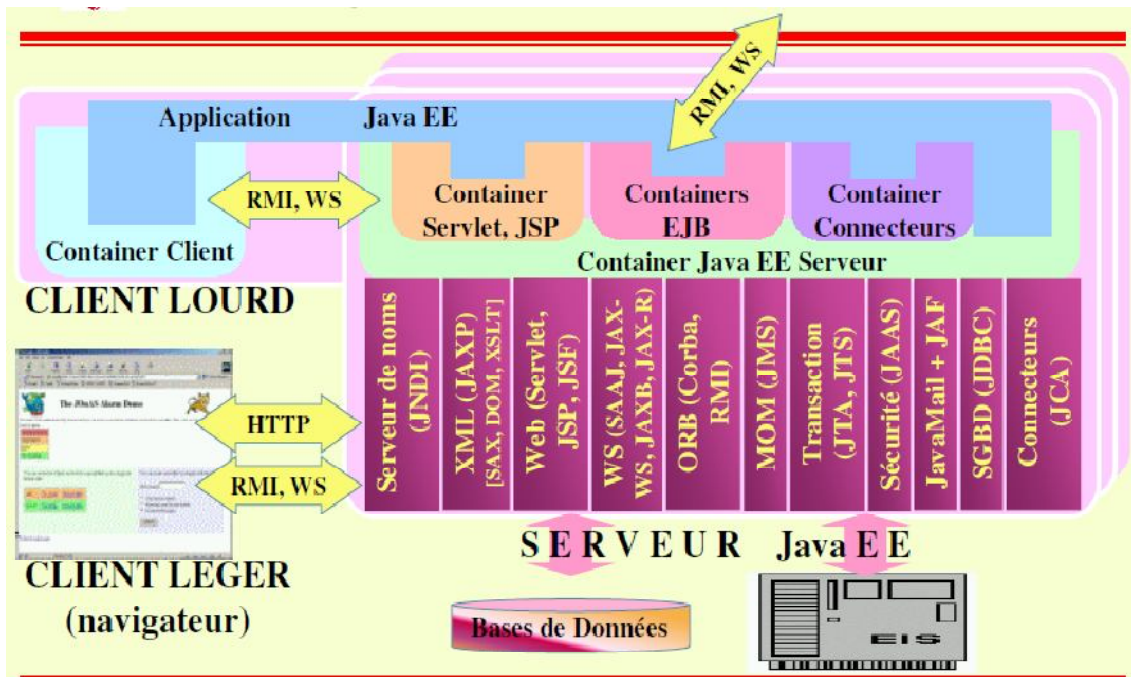


Figure I -3 : Les composants de la plate-forme java EE

4.3.1. Les composants Web :

Le serveur web d'une application java EE rend disponible la logique d'une application sur le web, donc c'est le serveur web qui gère la communication avec les clients web et qui répond à leurs requêtes.

- **Les servlets :**

La Servlet est écrit en langage Java, ce sont des programmes qui aident à la construction d'application générant des pages Web dynamiques. Il conçoit sous la forme d'une classe java et utilise une API spécifique. Cependant, le développement doit toujours mixer le code Java et HTML. De plus, la moindre modification oblige à recompiler la Servlet et à la recharger.

- **JSP (Java Server Page) :**

Les JSP viennent résoudre les problèmes de recompilation. Ici, c'est le code Java que l'on incorpore dans la page HTML avec technique de Scripting. Le serveur compile automatiquement la page en une Servlet et l'exécute ensuite. Ces techniques limitent aussi la réutilisation de code. Pour ce qui est du monde java, il a donc été proposé de faire collaborer les servlets et les JSP dans les applications. Les Servlets sont utilisées par les développeurs pour gérer les aspects programmation d'une application Web et JSP sont utilisées par les infographistes pour effectuer l'affichage.

4.3.2. Les composants métiers :

Les parties liées au domaine d'application, qui s'occupent de traiter la logique métier.

- **Les Java Beans :**

Les JavaBeans sont des composants écrits en Java(classe java) qui peuvent être employés dans tous les tiers d'une application. Ils sont souvent considérés comme un complément des servlets ou comme un composant des interfaces utilisateur. La spécification JavaBeans de Sun Microsystems définit les composants du type JavaBeans comme des composants logiciels réutilisables manipulables visuellement dans un outil de conception.

- **Les EJB (Entreprise Java Beans) :**

Les EJB sont des composants spéciaux, fonctionnant sur un serveur et utilisés pour construire la logique applicative et permet d'accéder aux données des applications java EE. Ils possèdent certaines caractéristiques comme la réutilisabilité, la possibilité de s'assembler pour construire une application, etc.

4.3.3. Les services d'infrastructures et de communication :

Les services principaux sont :

- ✓ Services d'infrastructure :

- **JDBC** : (Java Data Base Connectivity), permet l'accès aux bases de données. JDBC est une API d'accès aux systèmes de gestion de bases de données relationnelles qui permet d'exécuter des requêtes au sein d'un programme Java et de récupérer les résultats, ce qui représente une alternative aux solutions propriétaires. C'est de plus une tentative de standardiser l'accès aux bases de données car L'API est indépendant du SGBD choisi.
- **JTA/JTS** : (Java Transaction API/ Java Transaction), est un API définissant des interfaces standard avec un gestionnaire de transactions.
- **JCA** : (java EE Connector Architecture), est un API de connexion au système d'information de l'entreprise.
- **JMX** :(Java Management Extension), fournit des extensions permettant de développer des applications web de supervision d'application.
- **JNDI** : (Java Naming and Directory Interface), API d'accès aux annuaires d'entreprises

(DNS, LDAP, etc.).

✓ Services de communication :

- **JAAS** : (Java Authentication Autorisation Service), service de gestion des droits d'accès.
- **RMI** : (Remote Méthode Invocation), qui permet la communication entre objets distants.
- **JavaMail** : permettant l'envoi de courrier électronique.
- **JMS** : (Java Message Service), fournit des fonctionnalités de communication asynchrone (appelées MOM pour Middleware Object Message) entre applications. [W03]

4.4. Les conteneurs :

Les conteneurs assurent la gestion du cycle de vie des composants qui s'exécutent entre eux, et fournissent des services qui peuvent être utilisés par les applications lors de leur exécution.

Il existe plusieurs conteneurs définis par java EE:

- conteneur web : pour exécuter les servlets et les JSP
- conteneur d'EJB : pour exécuter les EJB
- conteneur client : pour exécuter des applications sur les postes qui utilisent des composants java EE.

Les serveurs d'applications peuvent fournir un conteneur web uniquement (exemple : Tomcat) ou un conteneur d'EJB uniquement (exemple : JBoss, Jonas, etc.) ou les deux (exemple : Websphère, Weblogic, etc.).[W04]

4.5. Modèle MVC pour java EE:

Le modèle MVC signifie Modèle-Vue-Contrôleur est un modèle de conception logicielle largement répandu, fort et utile. Le MVC (*design pattern* en anglais) très utilisé pour réaliser des sites web, il est donc indépendant du langage de programmation. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données. Ainsi l'application se retrouve segmentée en trois composants essentiels : [W5]

- Le modèle : représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données. [W5]
- La vue : correspond à l'IHM. Elle présente les données et interagit avec l'utilisateur. Dans le

cadre des applications Web, il s'agit d'une interface HTML/JSP, mais n'importe quel composant graphique peut jouer ce rôle. [W5]

- Le contrôleur : fait le lien entre la vue et le modèle et gère les interactions avec l'utilisateur il va utiliser les données du modèle, les traiter en fonction de l'action de l'utilisateur, et les envoyer à la vue afin qu'elle les affiche. .

Il est important de noter que les données sont indépendantes de la présentation. En d'autres termes, le modèle ne réalise aucune mise en forme. Ces données pourront être affichées par plusieurs vues. De ce fait le code du modèle est factorisé : il est écrit une seule et unique fois puis réutilisé par chaque vue. [W5]

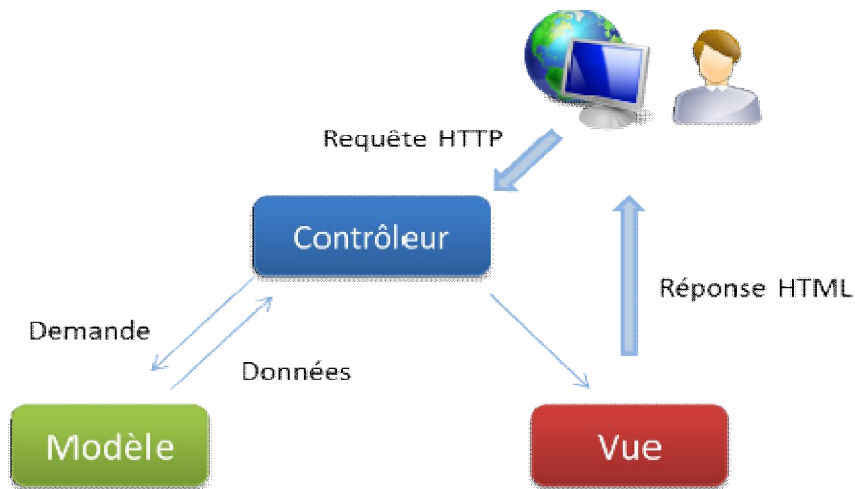


Figure I -4 : le modèle MVC

4.5.1 Modèle MVC de type1 :

Dans ce modèle, chaque requête est traitée par un contrôleur sous la forme d'une Servlet. Celle-ci traite la requête, fait appel aux éléments du modèle si nécessaire et redirige la requête vers une JSP qui se charge de créer la réponse à l'utilisateur. [W6]

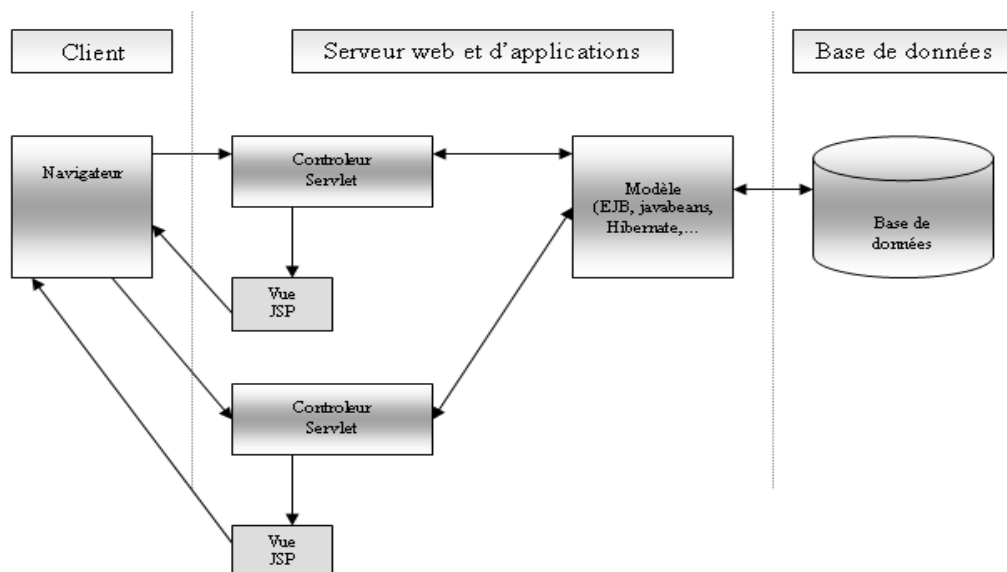


Figure I -5 : Le modèle MVC I

L'inconvénient est donc une multiplication du nombre de servlets nécessaire à l'application, l'implémentation de plusieurs servlets nécessite beaucoup de code à produire d'autant chaque Servlet doit être déclarée dans le fichier web.xml. [W6]

4.5.2 Modèle MVC de type2 :

Dans l'organisation logicielle MVC2, le Servlet est unique, en classe et en instance. Il garantit l'unicité du point d'entrée de l'application. Il prend en charge une partie du contrôle de l'application. Les contrôleurs MVC se retrouvent alors partiellement déportés dans l'entité « dynamique de l'application » qui assure le contrôle du dynamisme de l'application et qui gère les relations entre les objets métier et la présentation. Les contrôleurs deviennent essentiellement des contrôleurs du dialogue entre l'utilisateur et les objets métiers. [W5]

Pour cela des Framework ont été développés, qui sont composés d'une seule Servlet. Les Framework les plus connus, sont : (Struts, Spring, etc.).

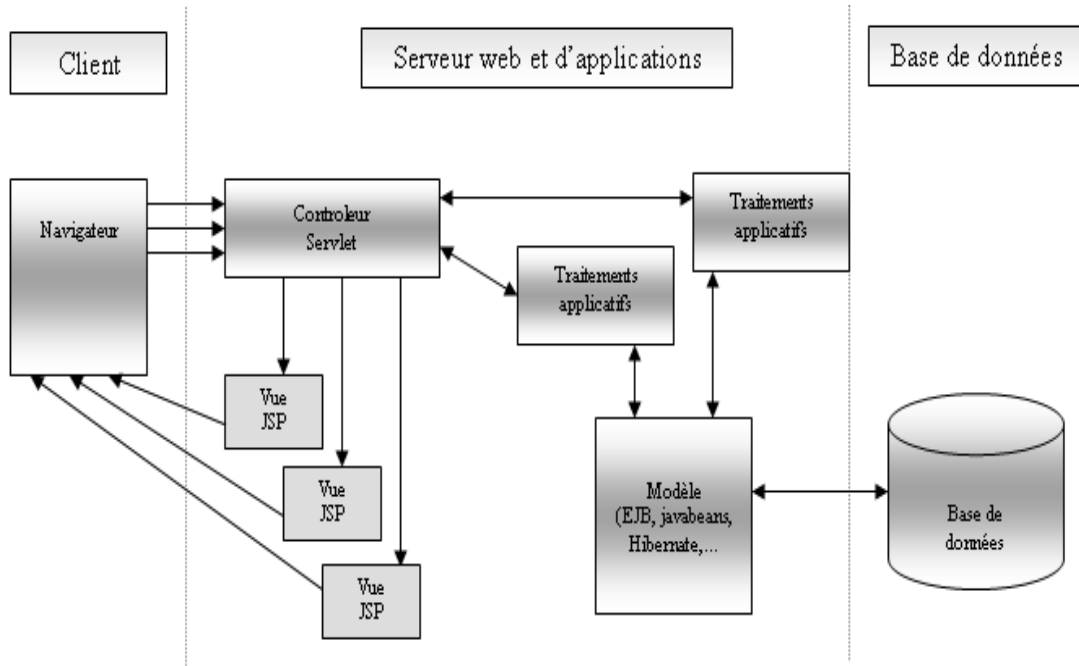


Figure I -6 : Le modèle MVC II

Le modèle MCV II hérite des propriétés du modèle MVC. Son organisation cependant capitalise sur la longue expérience acquise avec MVC et s'adapte au contexte des applications internet et de la plate-forme J2EE.

4.6. Modèle DAO (Data Access Object) :

Le modèle DAO propose de regrouper les accès aux données persistantes dans des classes à part, plutôt que de les disperser. Il s'agit surtout de ne pas écrire ces accès dans les classes métier, qui ne seront modifiées que si les règles de gestion métier changent. [W7]

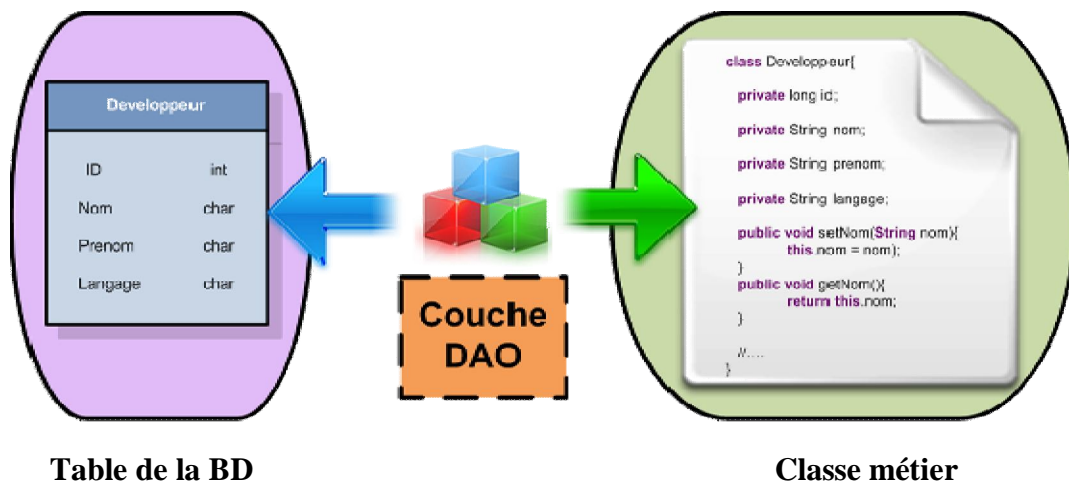


Figure I -7 : La couche DAO

5. Les Frameworks

Les frameworks sont des structures logicielles qui définissent des cadres dans lesquels viennent s'insérer les objets et concepts spécifiques à une application. En pratique, un framework (traduction littérale: squelette d'application) est un ensemble de classes et de mécanismes associés à une architecture logicielle qui fournit un ou plusieurs services techniques ou métiers aux applications pour facilitant la création de tout ou d'une partie d'un système logiciel. [W8]

Les principaux avantages de ces frameworks sont la réutilisation de leur code, la standardisation du cycle de vie du logiciel (Spécification, développement, maintenance, évolution), ils permettent de formaliser une architecture adaptée aux besoins de l'entreprise. Ils tirent parti de l'expérience des développements antérieurs. Pour cela plusieurs Frameworks ont été développés comme : (Struts, Spring, Hibernate, etc.). Pour notre projet on s'intéresse au Framework **Spring et Hibernate**.

5.1. Framework Spring :

Spring est l'un des Framework les plus répandu dans le monde Java pour le développement des applications J2EE, initialement il a développé par Rod Johnson et Juergen Holler, sa popularité a grandie au profit de la complexité de Java EE notamment pour ses versions antérieures à la version 5, mais aussi grâce à la qualité et la richesse des fonctionnalités. Spring permet une grande flexibilité dans les fonctionnalités et les projets utilisés dans une application. [W9]

➤ Les fonctionnalités proposées par Spring :

Spring propose de nombreuses fonctionnalités de base pour le développement des applications :

- Un conteneur léger implémentant le design pattern IoC (Injection of Control) pour la gestion des objets et de leur dépendance en offrant des fonctionnalités avancées concernant la configuration et l'injection automatique. Un de ces points forts est d'être non intrusif dans le code de l'application tout en permettant l'assemblage d'objets faiblement couplés.
- Une gestion des transactions par déclaration offrant une abstraction du gestionnaire de transactions sous-jacentes.
- Faciliter le développement des DAO de la couche de persistance en utilisant JDBC (Java Data Base Connectivity), JPA(Java Percistence Api), ou une solution open source comme (Hibernate).
- Faciliter la testabilité de l'application.

Une application typique utilisant Spring est généralement structurée en trois couches :

- la couche présentation : interface homme machine
- la couche service : interface métier avec mise en œuvre de certaines fonctionnalités (transactions, sécurité, etc.).
- La couche accès aux données : recherche et persistance des objets du domaine. [W9]

➤ Les avantages de Spring :

- ✓ Spring est un framework open source majoritairement développé par SpringSource.
- ✓ Il est très largement utilisé dans le monde Java, ce qui en fait un standard de facto et constitue une certaine garantie sur la pérennité du framework.
- ✓ Spring propose une très bonne intégration avec des frameworks open source (Struts, Hibernate, etc.) ou des standards de Java (Servlets, JMS (Java Messaging Service)).
- ✓ Toutes les fonctionnalités de Spring peuvent s'utiliser dans un serveur Java EE et pour la plupart dans un simple conteneur web ou une application autonome.
- ✓ Les fonctionnalités offertes par Spring sont très nombreuses et les sujets couverts ne cessent pas d'augmenter au fur et mesure des nouvelles versions et des nouveaux projets ajoutés au portfolio.
- ✓ La documentation de Spring est complète et régulièrement mise à jour lors de la diffusion de chaque nouvelle version. [W9]

➤ Spring et JAVA EE :

Spring est né de l'idée de fournir une solution plus simple et plus légère que celle proposée par Java 2 EE. C'est pour cette raison que Spring a été initialement désigné comme un conteneur léger. L'idée principale de Spring est de proposer un framework qui utilise une solution simple pour développer des applications plutôt que d'utiliser des EJB complexes dans un conteneur. [W9]

5.2. Framework Hibernate :

Hibernate est une solution open source du type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa.

Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant, il est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données. (Oracle, MySQL).[W9]

➤ Le fonctionnement d'Hibernate :

Hibernate a besoin de plusieurs éléments pour fonctionner :

- une classe de type javabean qui encapsule les données d'une occurrence d'une table.
- un fichier de configuration qui assure la correspondance entre la classe et la table (mapping).
- des propriétés de configuration notamment des informations concernant la connexion à la base de données

Une fois ces éléments correctement définis, il est possible d'utiliser Hibernate dans le code des traitements à réaliser. [W9]

6. Les avantages des applications java EE

Java EE s'appuie entièrement sur le langage Java, il bénéficie de nombreux avantages de ce langage, tels que la bonne portabilité sur les différents systèmes d'exploitation, la maintenabilité, l'indépendance du code, et la sécurité

Le développement d'applications d'entreprise nécessite la mise en œuvre d'une infrastructure importante. Beaucoup de fonctionnalités sont utilisées et développées, le but étant de produire des applications sûres, robustes et faciles à maintenir. [W10]

7. Conclusion

Dans ce chapitre, nous avons introduit le processus 2TUP qui sera la base des chapitres suivants, nous avons rappelé la notion des applications web avec ses avantages, et enfin nous avons également présenté la technologie java EE qui est une bonne solution pour les applications d'entreprise.

Chapitre II

Etude Préliminaire

1. Introduction

Dans ce chapitre nous allons déterminer la phase d'étude préliminaire qui permet de recueillir les informations initiales sur le système. Il s'agit dans cette phase de définir le contour du système, les différents acteurs, ainsi que les messages d'interaction avec le système.

L'étude préliminaire est la toute première étape du processus 2TUP. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques.

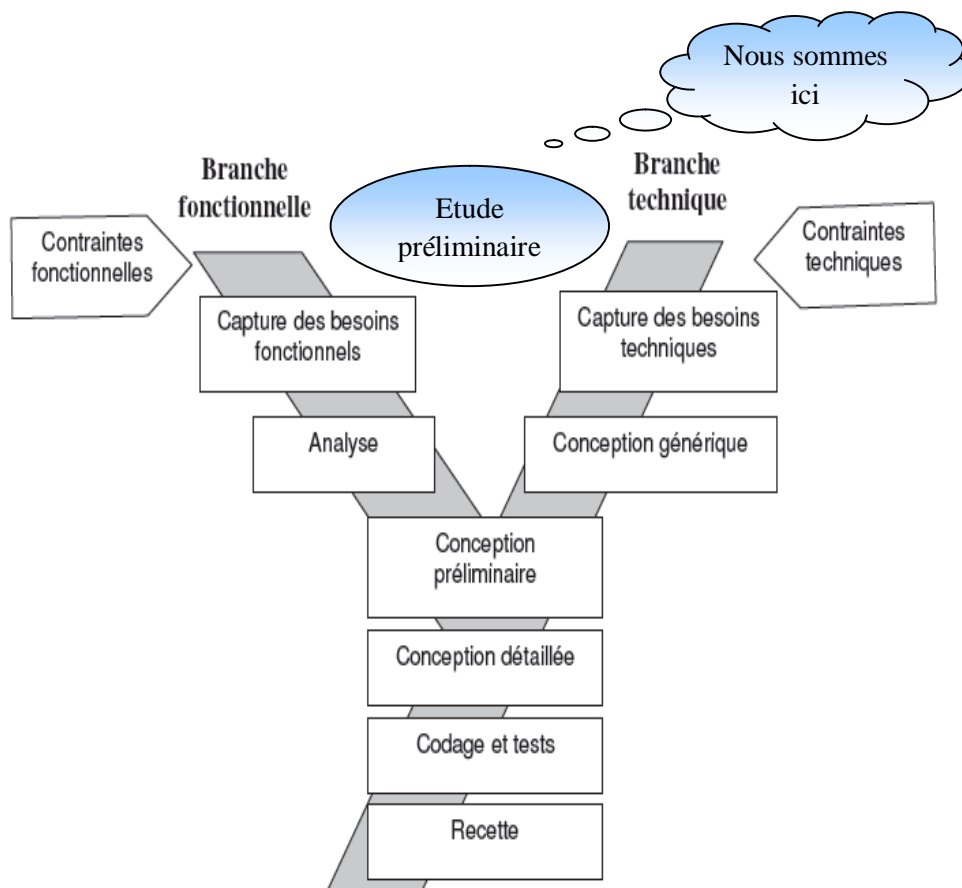


Figure II-1: La phase d'étude préliminaire

La figure suivante illustre le résumé des activités et les produits de l'étude préliminaire :

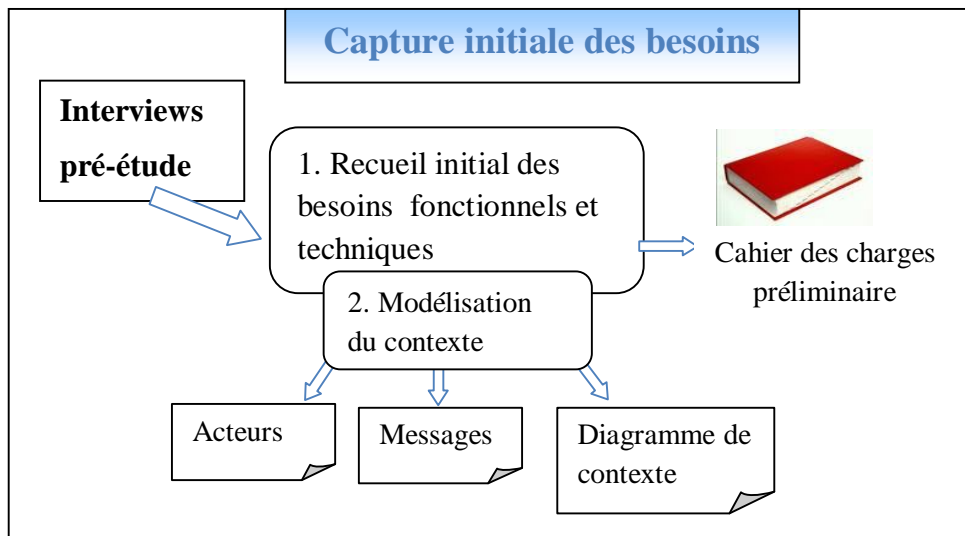


Figure II-2: Le résumé des activités et des produits de l'étude préliminaire

2. Elaboration du cahier de charge

2.1. Présentation du projet :

Grâce à une boutique en ligne, on peut choisir et payer des articles comme dans un magasin réel. Pour acheter un article de cette boutique virtuelle, il suffit le plus souvent de choisir les articles désirés puis de les mettre dans un panier d'achat virtuel. Le client peut ensuite, passer une commande et payer sa facture par carte bancaire ou par un autre moyen de paiement. La commande sera livrée en fonction du choix du client et selon les modalités définies par le responsable de la boutique.

2.2. Grands choix techniques :

Pour réaliser ce projet nous allons utiliser une approche itérative et incrémentale, fondée sur le processus en Y.

Nous avons choisi aussi un certain nombre de techniques-clés. Ces techniques-clés sont principalement :

- Le langage de modélisation UML.
- Le processus de développement en Y (2TUP).
- L'environnement de programmation Eclipse
- L'adoption d'une architecture JEE multicouche (client/serveur 3-tiers).
- Langage Java pour la programmation Web (Jsp, Servlet, Java Beans, ...)
- Les frameworks Spring et hibernate.
- Le serveur Apache Tomcat.
- Le SGBDR MYSQL.

2.3. Recueil des besoins fonctionnels :

Notre projet concentre sur la réalisation d'une application web pour les achats en ligne utilisant l'architecture Java EE, pour garanti les fonctions suivantes.

- Consulter le catalogue interactif des produits disponibles.
- Ce site donne aux visiteurs (internauts) la possibilité de s'inscrire, effectuer leurs commandes en ligne (achat en ligne), et de recevoir une confirmation.
- Gérer les relations avec les clients,
- Gérer les articles (ajout, modification, ou suppression),
- Assurer la bonne sauvegarde de données concernant les informations des clients et leurs commandes.

2.4. Recueil des besoins opérationnels :

- **La confidentialité des informations** : Les informations ne partagent pas à n'importe qui, seuls peut y accéder ceux qui en ont le droit.
- **La disponibilité des services** : Les services et les informations doivent être accessibles aux personnes autorisées quand elles en ont besoin.
- **L'autorisation d'accès** : Le site web contient des informations sensibles ou destinées seulement à un groupe de personnes restreint les personnes qui ont accès à des pages sont bien celles auxquelles qui ont l'autorisation d'accès qui décrit par une authentification (login et le mot de passe).

3. Description de contexte du système

Dans cette partie nous développons un premier modèle UML de niveau contexte, pour pouvoir établir les frontières fonctionnelles du système. La description du contexte du système contient 3 actions :

- Identification des acteurs.
- Identification des messages entre le système et ses acteurs.
- La réalisation du diagramme du contexte.

3.1. Identification des acteurs :

Un acteur représente l'abstraction d'un rôle joué par des entités externes (Utilisateur, dispositifs

matériels ou autre système) qui interagit directement avec le système étudié. [Roques07]

En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs du système sont :

- **Visiteur:** est une personne qui est entrain de fouiller internet, cherchant un produit ou un service pour l'acheter ou pour avoir une idée sur les catalogues et les prix, et même pour créer un compte sur le site.
- **Client :** un visiteur ayant déjà créé un compte sur le site, il a le droit de suivre le processus d'achat en toutes sécurités et la confidentialité concernant ces informations personnelles.
- **Administrateur (le web master) :** une personne qui assure le dynamisme du site et gère toutes les taches concernant les articles, les clients, les commandes, et la gestion de payement et de livraison.

Acteur	Description de rôle
visiteur	<ul style="list-style-type: none"> • Consulter le catalogue • Rechercher un article • Gérer le panier • Créer un compte
Le client	<ul style="list-style-type: none"> • S'authentifier • Consulter le catalogue • Rechercher un article • Gérer le panier • Passer la commande • Régler la facture • Gérer le profil
L'administrateur	<ul style="list-style-type: none"> • S'authentifier • Gérer articles pour chaque catalogue • Gérer les comptes • Voir les statistiques

Tableau II-1:Les acteurs et leurs rôles

3.2. Identification des messages :

Un message représente la spécification d'une communication unidirectionnelle entre les objets qui transportent de l'information avec l'intention de déclencher une activité chez le récepteur. Cette notion de message est également tout à fait applicable pour décrire les interactions de plus haut niveau entre les acteurs et le système.[Roques07]

Pour les acteurs, demandez-vous quels sont les messages qui déclenchent un comportement du système attendu par l'acteur dans le cadre de son activité. Pour le système, demandez-vous quels sont les messages émis à l'intention d'un acteur particulier, et qui portent une information utilisée par ce destinataire.[Roques07]

✓ **Le système émet les messages suivants:**

- La liste des articles disponibles (catalogue).
- Le formulaire d'inscription.
- Le formulaire d'authentification.
- Le compte client.
- Le formulaire d'ajout d'un article.
- Le formulaire de modification d'un article.
- La liste des articles recherchés.
- Les informations sur une commande.
- La liste des commandes.
- La facture.

✓ **Le système reçoit les messages suivants:**

- La saisie des informations et des critères de recherche.
- La saisie des informations d'inscription.
- La saisie des informations d'authentification.
- La saisie des informations concernant une commande.
- La saisie des informations concernant un article.
- La modification des informations concernant un article.

3.3. Modélisation de contexte :

➤ Diagramme de contexte dynamique:

Les messages entre le système et les acteurs identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique en utilisant un diagramme de communication de la manière suivante: [Roques07]

- Le système étudié est représenté par un objet central.
- Cet objet central est entouré par d'autres objets symbolisant les différents acteurs.
- Des liens relient le système à chacun des acteurs.
- Sur chaque lien sont montrés les messages en entrées et en sorties du système.

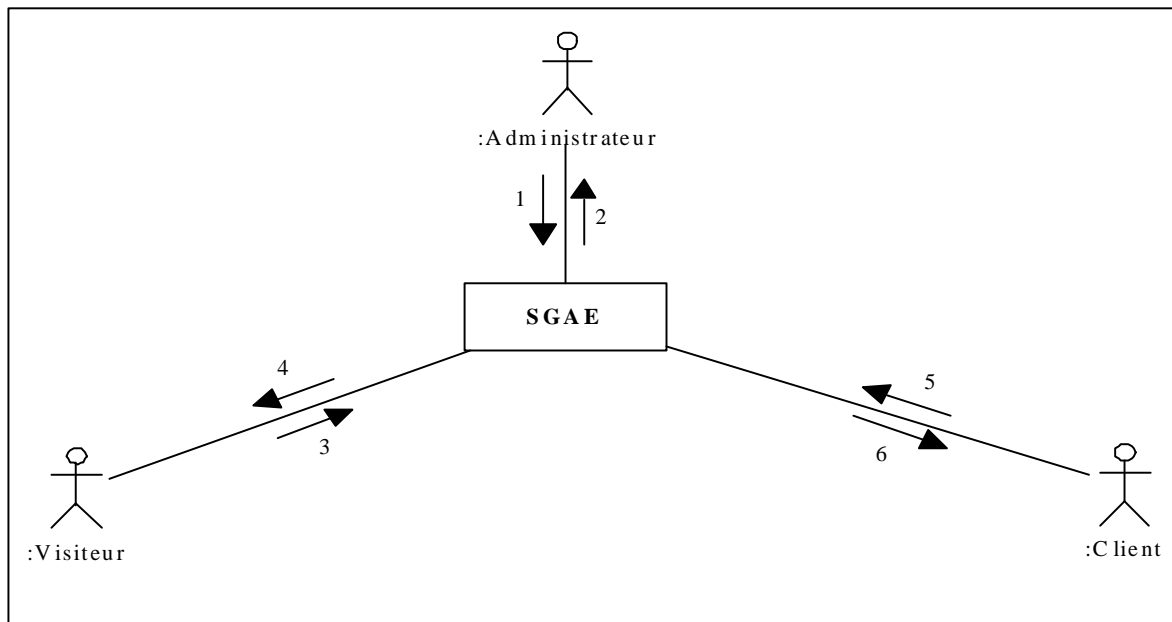


Figure II-3: Le diagramme de contexte dynamique du système SGAE

➤ Description détaillée des messages

1: Administrateur ► SGAE

- Les informations concernant le compte d'authentification (login et mot de passe).
- La mise à jour des informations sur les articles (suppression, modification, ajout).
- La gestion des informations sur les comptes.

2: SGAE ► Administrateur
<ul style="list-style-type: none">• Confirmation des informations d'authentification.• Confirmation des mises à jour concernant les articles.• Confirmation des mises à jour concernant les comptes.• Les informations concernant les commandes.
3: Visiteur ► SGAE
<ul style="list-style-type: none">• Les informations et les critères de la recherche• Les informations d'inscription.
4: SGAE ► Visiteur
<ul style="list-style-type: none">• Le formulaire d'authentification.• Liste des articles (catalogue).• Les informations concernant l'état du panier.
5: Client ► SGAE
<ul style="list-style-type: none">• Les informations et les critères de la recherche.• Les informations concernant une commande.• La liste des articles achetés avec ses informations.
6: SGAE ► Client
<ul style="list-style-type: none">• Les informations du compte client.• Liste des articles (catalogue).• Les informations concernant l'état du panier.• Les informations de la commande.• Les informations concernant la facture.

Tableau II-2:Légende des messages de diagramme de contexte

Nous remarquons que les deux acteurs Client et Visiteur partagent trois cas d'utilisation : ils sont également acteurs de « Consulter le catalogue », « Rechercher un article », et « Gérer le panier ».UML nous fournit la solution en permettant de créer un acteur généralisé « **Internaute** », dont Client et Visiteur seront des spécialisations.

Nous allons également prendre en compte le système externe de « **Paiement sécurisé** », nécessaire au moment du paiement en ligne.

4. Conclusion

Dans ce chapitre, nous avons déterminé les besoins fonctionnels en concédant le système comme une boîte noire et les besoins opérationnels (techniques), afin d'étudier sa place dans le système métier plus global de l'entreprise, Nous avons développé un modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

Tout ça prépare le terrain pour entamer la conception de notre projet par l'étape de capture des besoins fonctionnels et de capture des besoins techniques dans le chapitre suivant.

Chapitre III



Capture des besoins fonctionnels et techniques

1. Introduction

Ce chapitre vient pour compléter l'étude fonctionnelle et technique ébauchée durant l'étude préliminaire dans le chapitre précédent. L'objectif de la capture des besoins consiste à déterminer ce que le système doit faire, c.à.d. le « quoi » à fournir aux développeurs une meilleure compréhension des fonctionnalités du système qu'ils doivent développer, elle comporte deux étapes : la capture des besoins fonctionnels et la capture des besoins techniques.

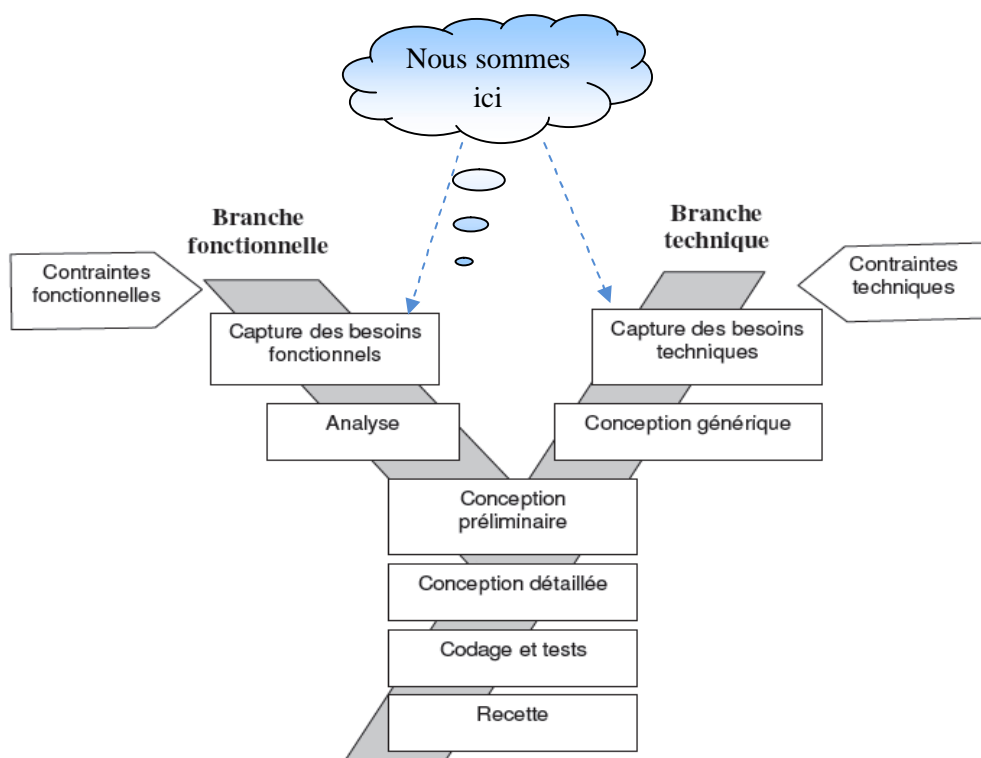


Figure III-1: L'étape de capture des besoins

2. Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y, elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire. La figure suivante illustre le résumé des activités et les produits et la démarche de capture des besoins fonctionnels. [Roques07]

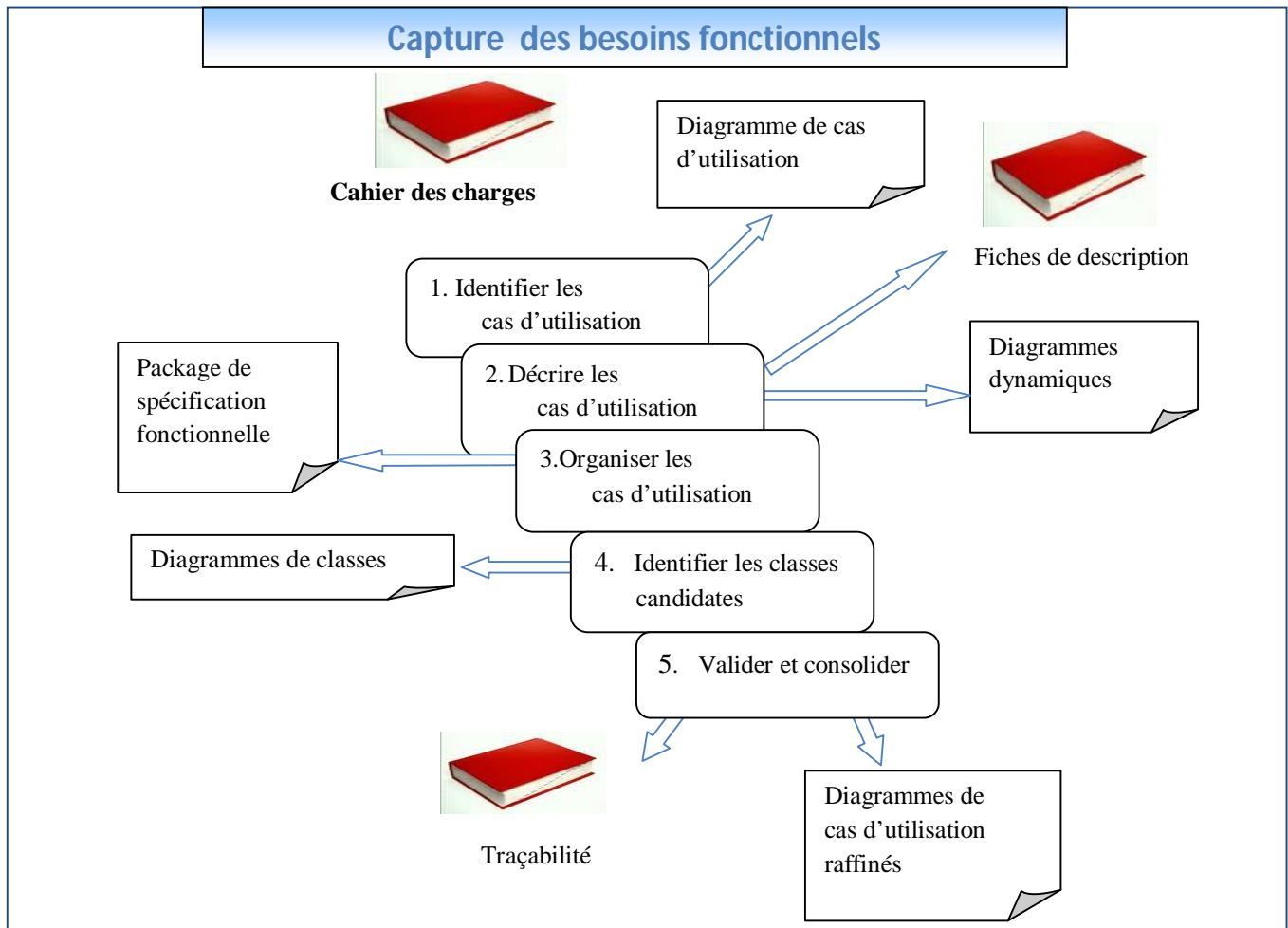


Figure III -2: La démarche de capture des besoins fonctionnels

2.1. Déterminer les cas d'utilisations:

- **Définition :** Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [Roques07]

Liste des cas d'utilisations

- Consulter le catalogue.
- Rechercher un article.
- Gérer le panier.
- Créer un compte.
- Gérer le profil.
- Passer la commande.
- Consulter la facture.
- Régler la facture.
- S'authentifier.
- Gérer les articles pour chaque catégorie.
- Gérer comptes.
- Voir les statistiques.

Tableau III -1: La liste des cas d'utilisation du système SGAE**➤ Identification des cas d'utilisation :**

En premier lieu, nous donne un aperçu des fonctionnalités futures que doit implémenter le système. Cependant, il nous faut plusieurs itérations pour ainsi arriver à constituer des cas d'utilisation complets. D'autres cas d'utilisation vont apparaître au fur et à mesure de la description de ceux-là, et l'avancement dans le recueil des besoins fonctionnels.

Pour constituer les cas d'utilisation, il faut considérer l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message. En regroupant les intentions fonctionnelles en unités cohérentes, on obtient les cas d'utilisation.

Cas d'utilisation	Acteur(s)	Message(s) émis/reçus
Consulter le catalogue	Visiteur/Client	<u>Reçoit</u> : Les informations sur le contenu du site.
Rechercher un article	Visiteur/Client	<u>Émet</u> : Les informations et les critères de la recherche. <u>Reçoit</u> : La liste des articles.
Gérer le panier	Visiteur/Client	<u>Reçoit</u> La liste des articles dans le panier.
Créer un compte	Visiteur	<u>Émet</u> : Les informations personnelles de visiteur. <u>Reçoit</u> : Le compte client.
Gérer le profil	Client	<u>Émet</u> : Les nouvelles informations personnelles. <u>Reçoit</u> : Le compte client modifié.
Passer la commande	Client	<u>Émet</u> : Les informations de la commande.
Consulter la facture	Client	<u>Reçoit</u> : Les informations concernant la facture d'une commande (montantTotal).
Régler la facture	Client	<u>Émet</u> : Les informations sur le compte bancaire et de la commande.
S'authentifier	Administrateur/Client	<u>Reçoit</u> : les informations d'authentification (login, motpasse).
Gérer les articles pour chaque catégorie	Administrateur	<u>Émet</u> : Les informations de MAJ d'un article. <u>Reçoit</u> : La nouvelle liste des articles.
Gérer les comptes	Administrateur	<u>Reçoit</u> : Les informations des comptes.
Voir les statistiques	Administrateur	<u>Reçoit</u> : Les statistiques.

Tableau III-2 : La liste des acteurs et des messages par cas d'utilisation du système SGAE

➤ Diagramme de cas d'utilisation:

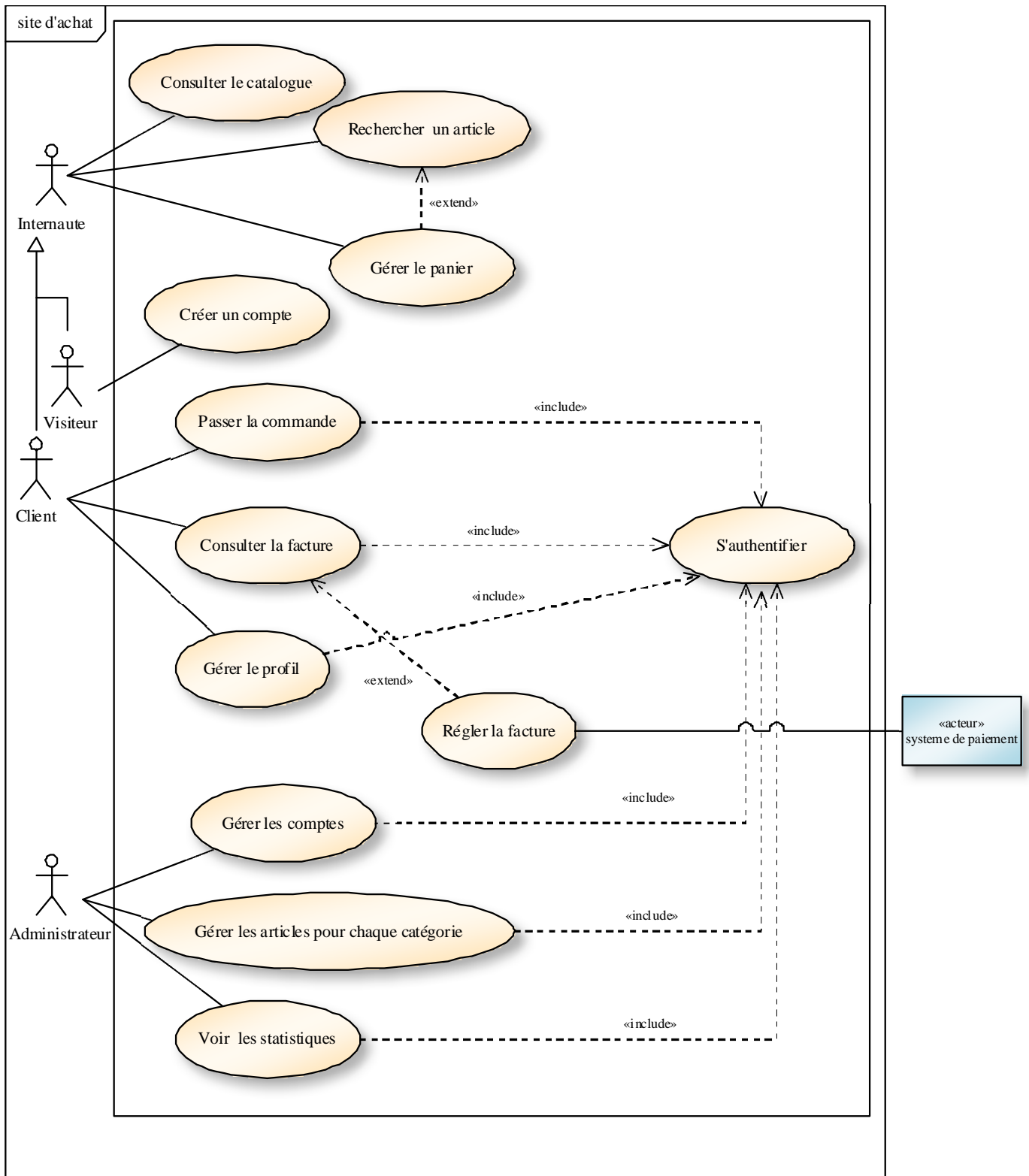


Figure III -3: Le diagramme de cas d'utilisation

2.2. Description détaillée des cas d'utilisation:

Nous allons maintenant détailler chaque cas d'utilisation qui doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système et les séquences d'actions principales qu'il est susceptible d'effectuer.

Les descriptions vont être organisées de la façon suivante :

- Un sommaire d'identification : va résumer les propriétés du cas d'utilisation.
- Une description détaillée : des pré conditions au déclenchement du cas d'utilisation doivent être spécifiées, un scénario nominal décrivant celui-ci additionné à des scénarios alternatifs et d'exceptions.
- Les diagrammes : plusieurs diagrammes vont apparaître pour apporter une compréhension additive au cas d'utilisation.

2.2.1. Description textuelle des cas d'utilisation :

La fiche de description d'un cas d'utilisation n'est pas normalisée par UML. Nous utilisons ici la structure proposée par l'auteur dans [Roques 07].

Cas d'utilisation : Consulter le catalogue	
Sommaire d'identification	
Titre	consulter le catalogue
But	Permet de faire une vision globale sur le contenu du site.
Acteur	Visiteur/Client
Description	
Pré condition	/
Démarrage	Ce cas d'utilisation commence lorsque l'acteur demande au système de consulter le catalogue.
Enchaînements	<p>Enchaînement(a) : <u>Consulter le catalogue</u></p> <ul style="list-style-type: none"> - le visiteur demande de consulter le catalogue. - Le système affiche la page demandée.

Arrêt	Ce cas d'utilisation est terminé lorsque la consultation est terminée ou annulée.
Post condition	/

Cas d'utilisation : Rechercher un article	
Sommaire d'identification	
Titre	rechercher un article
But	Permet de lancer une recherche simple ou multicritères.
Acteur	Visiteur/Client
Description	
Pré condition	Le catalogue est disponible.
Démarrage	Ce cas d'utilisation commence lorsque l'acteur lance une recherche.
Enchaînements	<p>Enchaînement(a) : <u>rechercher un article.</u></p> <ul style="list-style-type: none"> - L'acteur demande une recherche simple ou multicritère. - Le système affiche le formulaire de recherche. - L'acteur saisit les données nécessaires. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - L'acteur valide les informations saisies. - Le système vérifie la validité des informations. <p style="padding-left: 40px;">S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est afficher sur l'écran avise l'acteur que les informations saisies non valide (article non disponible).
Arrêt	Lorsque la liste des articles recherchés est affichée.
Post condition	Liste des articles.

Cas d'utilisation : Gérer le panier	
Sommaire d'identification	
Titre	gérer le panier
But	Permet de faire la mise à jour des articles dans le panier.
Acteur	Visiteur/Client
Description	
Pré condition	/
Démarrage	Ce cas d'utilisation commence lorsque l'internaute demande au système de gérer le panier.
Enchaînements	<p>Enchaînement(a) : <u>gérer panier</u></p> <ul style="list-style-type: none"> - Le client demande l'accès au panier. - Le système lui affiche l'état de son panier. Chaque article sélectionné est présenté sur une ligne, avec ses caractéristiques. - le client mettre à jour son panier (modifier les quantités, retirer, ou ajouter quelques articles, etc.). - Le système confirme cette mise à jour. <p style="padding-left: 40px;">Si le panier est vide : [Exception1]</p>
Exception1	- Un message d'erreur est afficher sur l'écran avise le client (Votre panier est vide) .
Arrêt	Ce cas d'utilisation est terminé lorsque la mise à jour est terminée.
Post condition	L'état du panier est modifié.

Cas d'utilisation : Créer un compte	
Sommaire d'identification	
Titre	Créer un compte
But	Permet de s'inscrire sur le site.
Acteur	Visiteur.
Description	
Pré condition	/
Démarrage	Ce cas d'utilisation commence lorsque le visiteur demande de créer un compte.
Enchaînements	<p>Enchaînement(a) : <u>Saisir les informations du visiteur</u></p> <ul style="list-style-type: none"> - Le visiteur demande de créer un compte. - Le système affiche le formulaire d'inscription. - Le visiteur saisit les informations. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - Le visiteur valide les informations saisies. - Le système vérifie la validité des informations. <p>S'il y a des erreurs, Déclencher : [Exception1].</p>
Exception1	Un message d'erreur est affiché avise le visiteur que les informations saisies non valide (champ vide, informations erronées, etc.).
Arrêt	Lorsque les informations saisies sont enregistrées ou annulées.
Post condition	Un compte client a été créé.

Cas d'utilisation : Gérer le profil	
Sommaire d'identification	
Titre	Gérer profil
But	Permet de gérer le profil.

Acteur	client
Description	
Pré condition	Client s'authentifier
Démarrage	Ce cas d'utilisation commence lorsque le client demande au système de gérer (consulter et modifier) son profil.
Enchaînements	<p>Enchaînement(a) : <u>Mettre à jour les informations</u></p> <ul style="list-style-type: none"> - Le client demande de consulter les informations de profil. - Le système affiche le profil client. - Le client modifie ses informations. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - Le client valide les informations modifiées. - Le système vérifie la validité des informations. <p>S'il y a des erreurs, Déclencher : [Exception1].</p>
Exception1	Un message d'erreur est affiché avise le client que les informations saisies non valides (champ vide, informations erronées, etc.).
Arrêt	Lorsque la modification est terminée ou annulée.
Post condition	Profil client modifié.

Cas d'utilisation : Passer la commande	
Sommaire d'identification	
Titre	Passer commande
But	Permet de passer une commande.
Acteur	client
Description	

Pré condition	Client s'authentifié Client Gérer le panier
Démarrage	Ce cas d'utilisation commence lorsque le client demande au système de passer une commande.
Enchaînements	<p>Enchaînement(a) : <u>Saisir les informations de la commande</u></p> <ul style="list-style-type: none"> - Le client demande de passer une commande. - Le système affiche le formulaire d'une commande. - Le client saisit les informations nécessaires. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - Le client valide les informations saisies. - Le système vérifie la validité des informations. <p>S'il y a des erreurs, Déclencher : [Exception1].</p>
Exception1	Un message d'erreur est affiché avise le client que les informations saisies non valides (champ vide, informations erronées, etc.).
Arrêt	Lorsque la commande est créée ou annulée.
Post condition	Une commande a été ajoutée.

Cas d'utilisation : Consulter la facture	
Sommaire d'identification	
Titre	Consulter la facture
But	Permet de consulter la facture.
Acteur	client
Description	

Pré condition	Le client s'authentifier. La commande est déjà passée.
Démarrage	Ce cas d'utilisation commence lorsque le client demande au système de consulter la facture.
Enchaînements	Enchaînement(a) : <u>consulter la facture</u> <ul style="list-style-type: none"> - Le client demande de consulter la facture. - Le système affiche la liste des factures existe. - Le client sélectionne une facture. - Le système affiche la facture avec les différentes informations concernant les articles, les prix unitaires et le montant total.
Arrêt	Lorsque la consultation de la facture est terminée.
Post condition	/

Cas d'utilisation : Régler la facture	
Sommaire d'identification	
Titre	régler la facture
But	Permet de régler la facture.
Acteur	Client et le système de paiement.
Description	
Pré condition	Le client s'authentifier. La commande est déjà passée.
Démarrage	Ce cas d'utilisation commence lorsque le client demande au système de régler la facture.

Enchaînements	<p>Enchaînement(a) : régler la facture</p> <ul style="list-style-type: none"> - Le client demande de régler la facture. - Le système affiche la liste des factures. - Le client choisit une facture. - Le système affiche un formulaire qui contient les informations de la commande. - Le client sélectionne le paiement par carte bancaire et valide sa réservation. Il doit pour cela fournir : (numéro de la carte de crédit, son type, etc.). <p>Enchaînement(b) : Valider les informations saisies</p> <ul style="list-style-type: none"> - Le client valide les informations saisies. - Le système envoie les informations au système externe de paiement. - Le système de paiement vérifie la validité de la carte. <p>S'il y a des erreurs, Déclencher : [Exception1].</p>
Exception1	Un message d'erreur est affiché avise le client le système de paiement détecte que les informations sur la carte sont incomplètes ou erronées
Arrêt	Lorsque la facture est réglée ou annulée.
Post condition	Une nouvelle facture est réglée.

Cas d'utilisation : S'authentifier.	
Sommaire d'identification	
Titre	S'authentifier.
But	Permet de saisir les informations (login, mot de passe) des compte clients ou administrateurs du site.
Acteur	Client, Administrateur
Description	
Pré condition	L'existence d'un compte client (ou administrateur).

Démarrage	Ce cas d'utilisation commence lorsque le client (ou administrateur) demande au système d'accéder au site.
Enchaînements	<p>Enchaînement(a) : <u>Saisir les informations d'authentification</u></p> <ul style="list-style-type: none"> - Le client/administrateur demande de s'authentifier. - Le système lui affiche la page d'authentification. - Le client/administrateur remplit le formulaire (nom d'utilisateur, mot de passe). <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - Le client/administrateur valide les informations saisies. - Le système vérifie les informations saisies par l'utilisateur et le renvoie vers la page voulue. <p>S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est affiché sur l'écran avise le client (ou l'administrateur) que les informations saisies non valides.
Arrêt	Lorsque les informations sont enregistrées ou annulées.
Post condition	/

Cas d'utilisation : Gérer les articles pour chaque catégorie	
Sommaire d'identification	
Titre	Gérer les articles pour chaque catégorie.
But	Permet de gérer les articles d'une Catégorie.
Acteur	administrateur
Description	
Pré condition	Administrateur s'authentié
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système de gérer les articles d'une catégorie.

Enchaînements	<p>Enchaînement(a) : <u>mise à jour des articles.</u></p> <ul style="list-style-type: none"> - L'administrateur demande la mise à jour des articles. - Le système affiche la page de mise à jour des articles. - L'administrateur met à jour les informations (ajouter, supprimer ou modifier article). <p>Enchaînement(b) : <u>confirmation des MAJ.</u></p> <ul style="list-style-type: none"> - Le système lui demande une confirmation. - L'administrateur confirme la mise à jour. <p style="padding-left: 40px;">S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est affiché sur l'écran avise l'administrateur que les informations saisies non valides.
Arrêt	Lorsque les informations sont enregistrées ou annulées
Post condition	Un nouveau article est ajouté, modifié ou supprimé.

Cas d'utilisation : Ajouter un article.	
Sommaire d'identification	
Titre	Ajouter article.
But	Permet d'ajouter un article à la base de données du site.
Acteur	Administrateur
Description	
Pré condition	L'Administrateur s'authentifier.
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système d'ajouter un article au site.
	Enchaînement(a) : <u>Ajouter un article</u>

Enchaînements	<ul style="list-style-type: none"> - L'administrateur demande d'ajouter un article. - Le système lui affiche la page d'ajout au catalogue. - L'administrateur saisit les informations de ce nouvel article. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p> <ul style="list-style-type: none"> - L'administrateur valide les informations saisies. - Le système vérifie la validité des informations. <p>S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est affiché sur l'écran avise l'administrateur que les informations saisies non valides et article existe déjà.
Arrêt	Lorsqu'un article est ajouté.
Post condition	Un article a été ajouté dans la base de données.

Cas d'utilisation : Modifier un article.

Sommaire d'identification

Titre	Modifier un article.
But	Permet de Modifier des informations d'un article (prix,quantiteStock...)
Acteur	Administrateur

Description

Pré condition	L'Administrateur s'authentifier.
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système de modifier des informations d'un article au site.
Enchaînements	<p>Enchaînement(a) : <u>Modifier un article</u></p> <ul style="list-style-type: none"> - L'administrateur choisit et sélectionne un article pour le modifier. - Le système lui affiche la page demandée. - L'administrateur modifie les informations souhaitées. <p>Enchaînement(b) : <u>Valider les informations saisies</u></p>

	<ul style="list-style-type: none"> - L'administrateur valide et confirme les modifications saisies. - Le système vérifie la validité des informations. <p>S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est affiché sur l'écran avise l'administrateur que les informations saisies non valides (L'information mal remplir).
Arrêt	Lorsque l'opération de modification es terminée.
Post condition	Un article a été modifié.

Cas d'utilisation : Supprimer un article.	
Sommaire d'identification	
Titre	Supprimer un article.
But	Permet de Supprimer un article à la base de donné du site.
Acteur	Administrateur
Description	
Pré condition	L'Administrateur s'authentifier.
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système de supprimer un article au site.
Enchaînements	<p>Enchaînement(a) : <u>Supprimer un article</u></p> <ul style="list-style-type: none"> - L'administrateur choisit et sélectionne un article pour le supprimer. <p>Enchaînement(b) : <u>Valider la suppression</u></p> <ul style="list-style-type: none"> - L'administrateur confirme la suppression. - Le système supprime l'article de la base de données.
Arrêt	Lorsque l'Opération de la suppression réussite.
Post condition	Un article a été supprimé.

Cas d'utilisation : Gérer les comptes	
Sommaire d'identification	
Titre	Gérer comptes
But	Permet de gérer les comptes clients.
Acteur	Administrateur
Description	
Pré condition	Administrateur s'authentié.
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système de gérer un compte.
Enchaînements	<p>Enchaînement(a) : gérer compte.</p> <ul style="list-style-type: none"> - L'administrateur demande au système de gérer un compte. - Le système affiche la liste des comptes. - L'administrateur sélectionne un compte pour le consulter ou bien le bloquer en cas de problème de sécurité. - Le système confirme l'opération. <p>S'il y a des erreurs, déclenche [Exception1].</p>
Exception1	Un message d'erreur est afficher sur l'écran avise l'administrateur que la liste des compte est vide.
Arrêt	Lorsque la mise à jour est terminée ou annulée.
Post condition	/.

Cas d'utilisation : Voir les statistiques	
Sommaire d'identification	
Titre	Voir statistiques
But	Permet de voir les différentes statistiques qui concernent les commandes (les achats).

Acteur	administrateur
Description	
Pré condition	Administrateur s'authentié.
Démarrage	Ce cas d'utilisation commence lorsque l'administrateur demande au système de voir les statistiques.
Enchaînements	<p>Enchaînement(a) : Consulter statistiques</p> <ul style="list-style-type: none"> - L'administrateur demande de voir les statistiques. - Le système affiche les critères des choix. - L'administrateur sélectionne un choix. - Le système affiche les statistiques.
Arrêt	Lorsque la consultation est terminée.
Post condition	/.

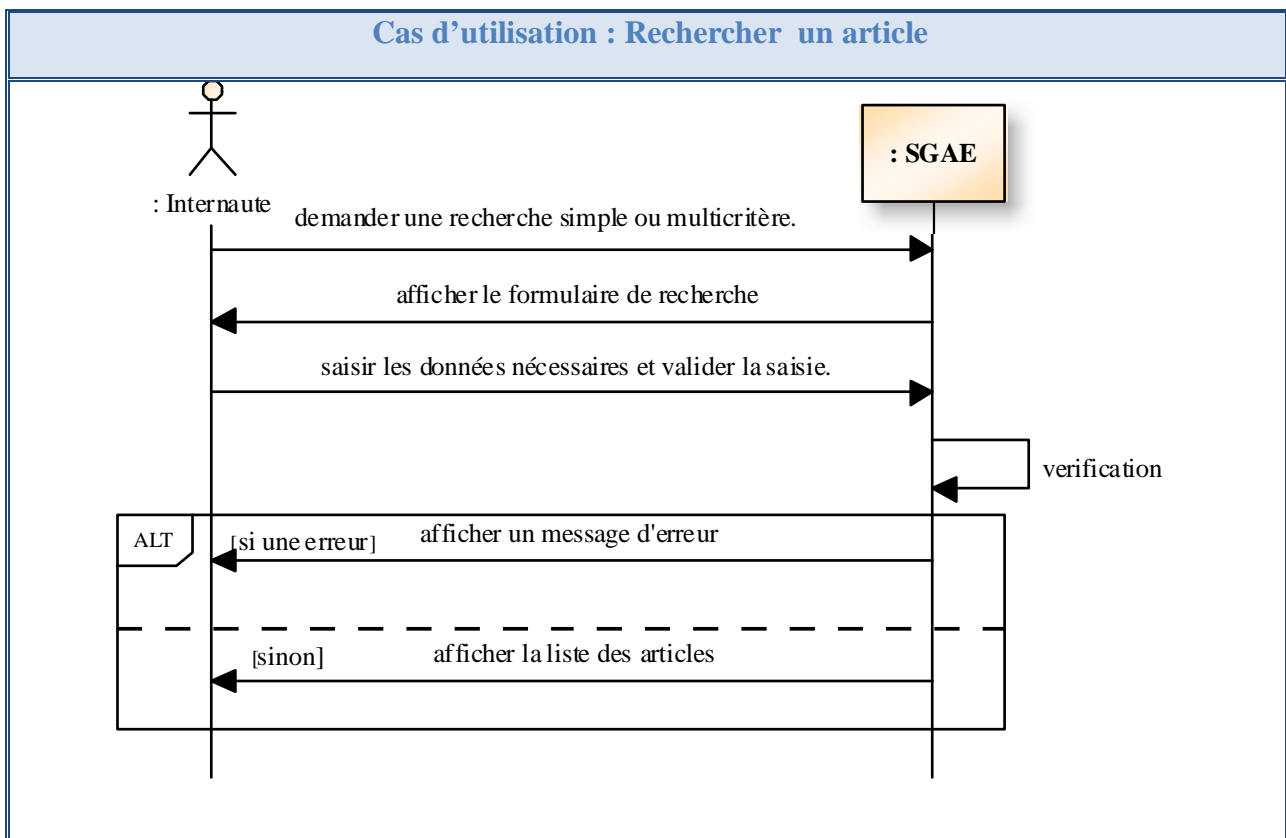
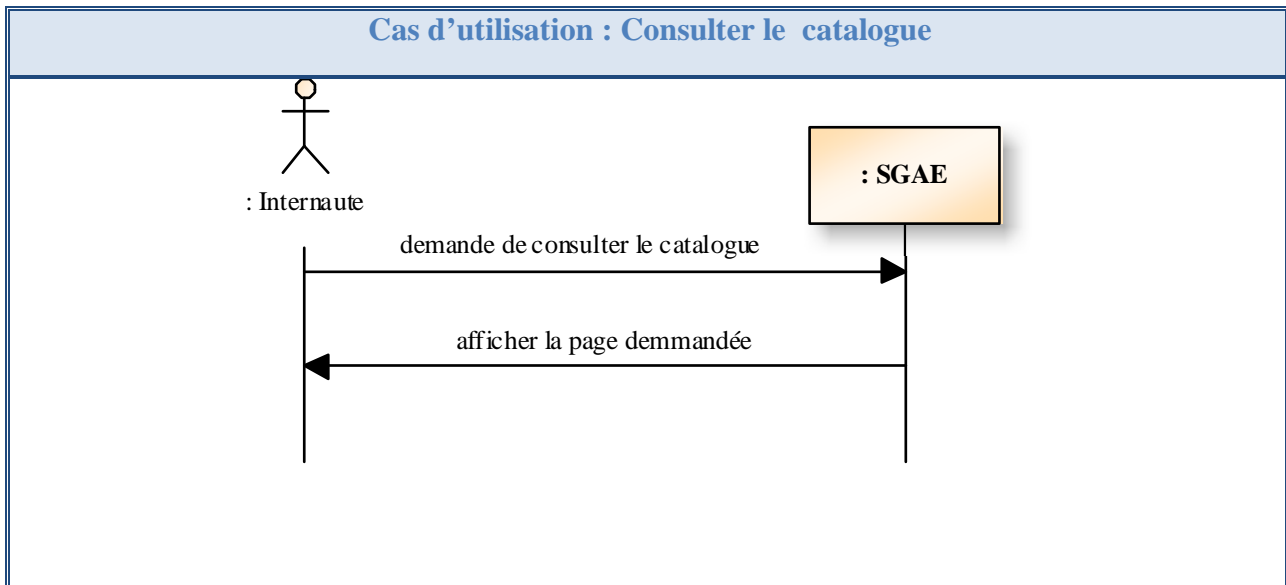
Tableau III -3: Description textuelle des cas d'utilisation fonctionnels du système SGAE

2.2.2. Description graphique des cas d'utilisation :

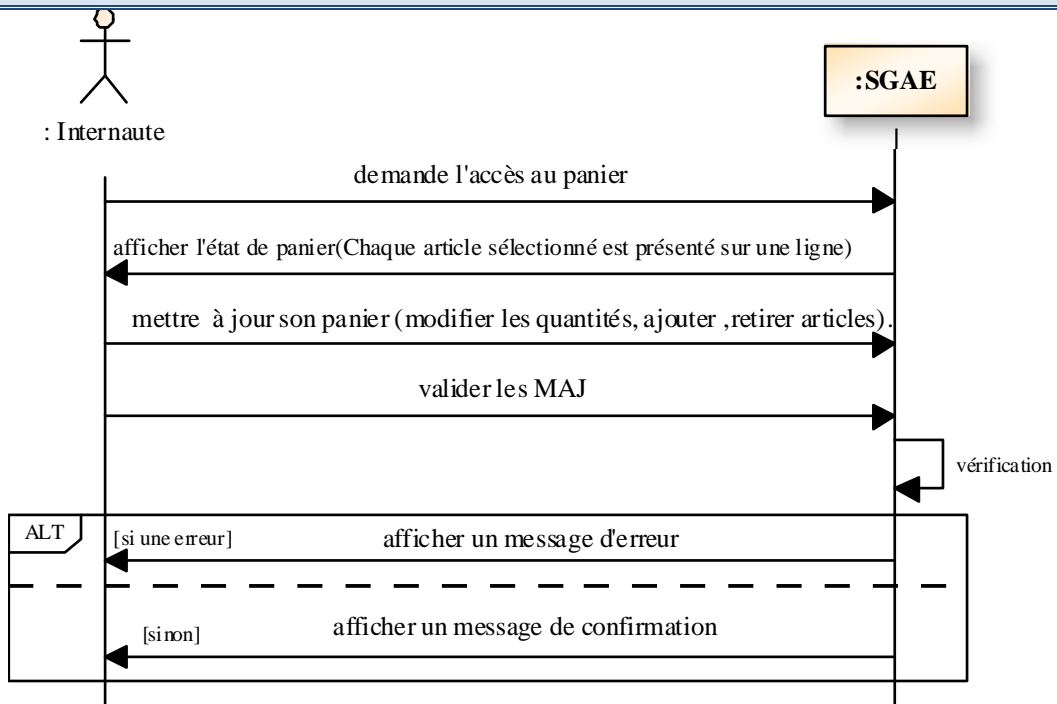
La description textuelle des cas d'utilisation est indispensable, car elle seule permet de communiquer facilement et précisément avec les utilisateurs. En revanche, le texte présente des désavantages puisque il est difficile de montrer comment les enchaînements se succèdent, il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes d'UML (diagramme d'activité, diagramme d'état transition, diagramme de séquence, et diagramme de collaboration).

Nous avons choisi le diagramme de séquence par ce qu'il est facilement compris, il représente les interactions entre objets en précisant la chronologie des échanges des messages.

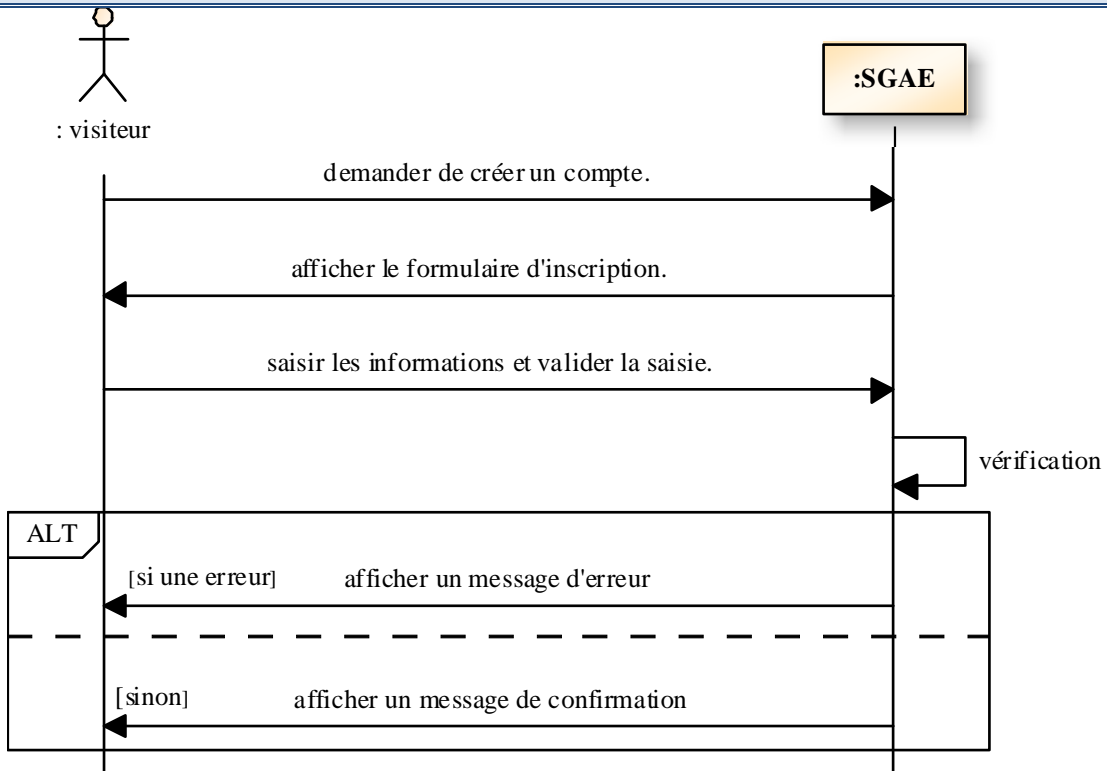
➤ Diagramme de séquence système :

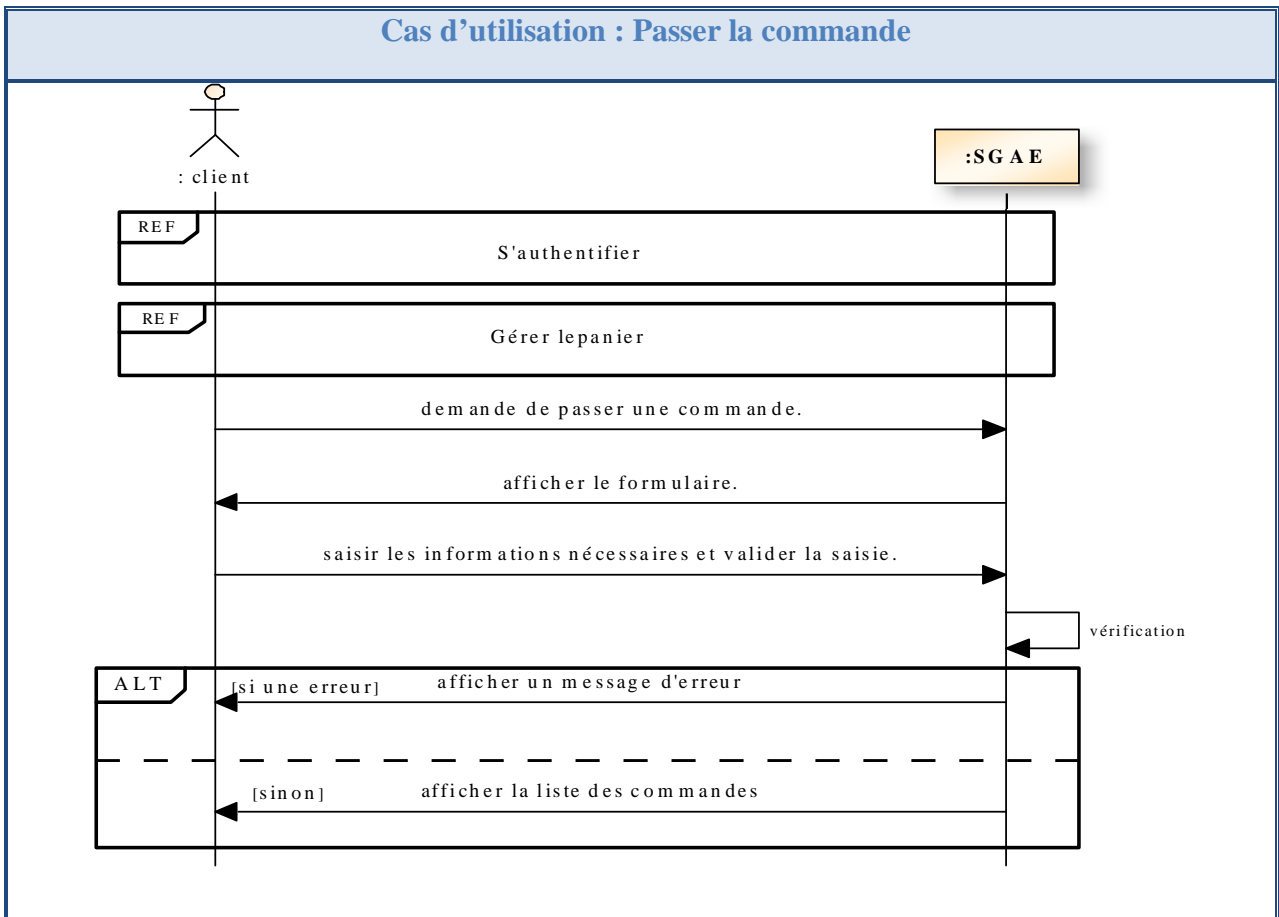
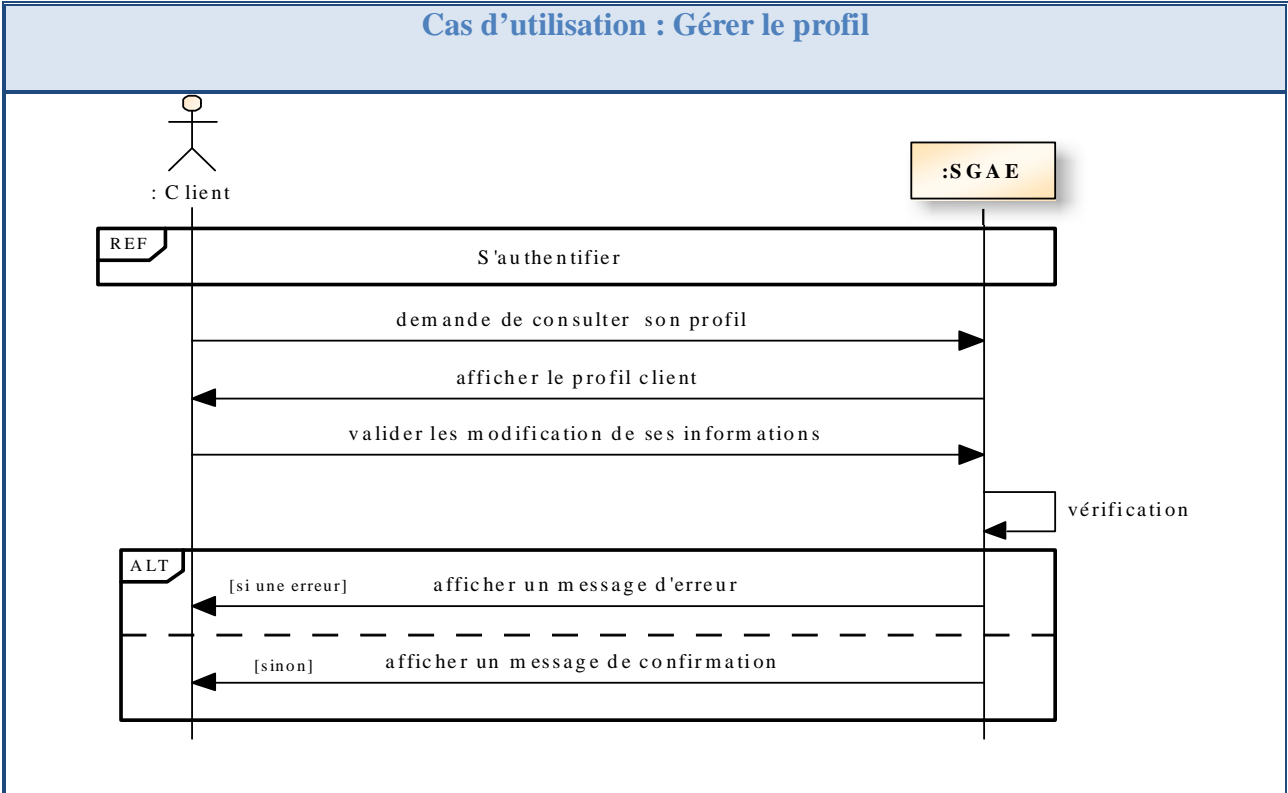


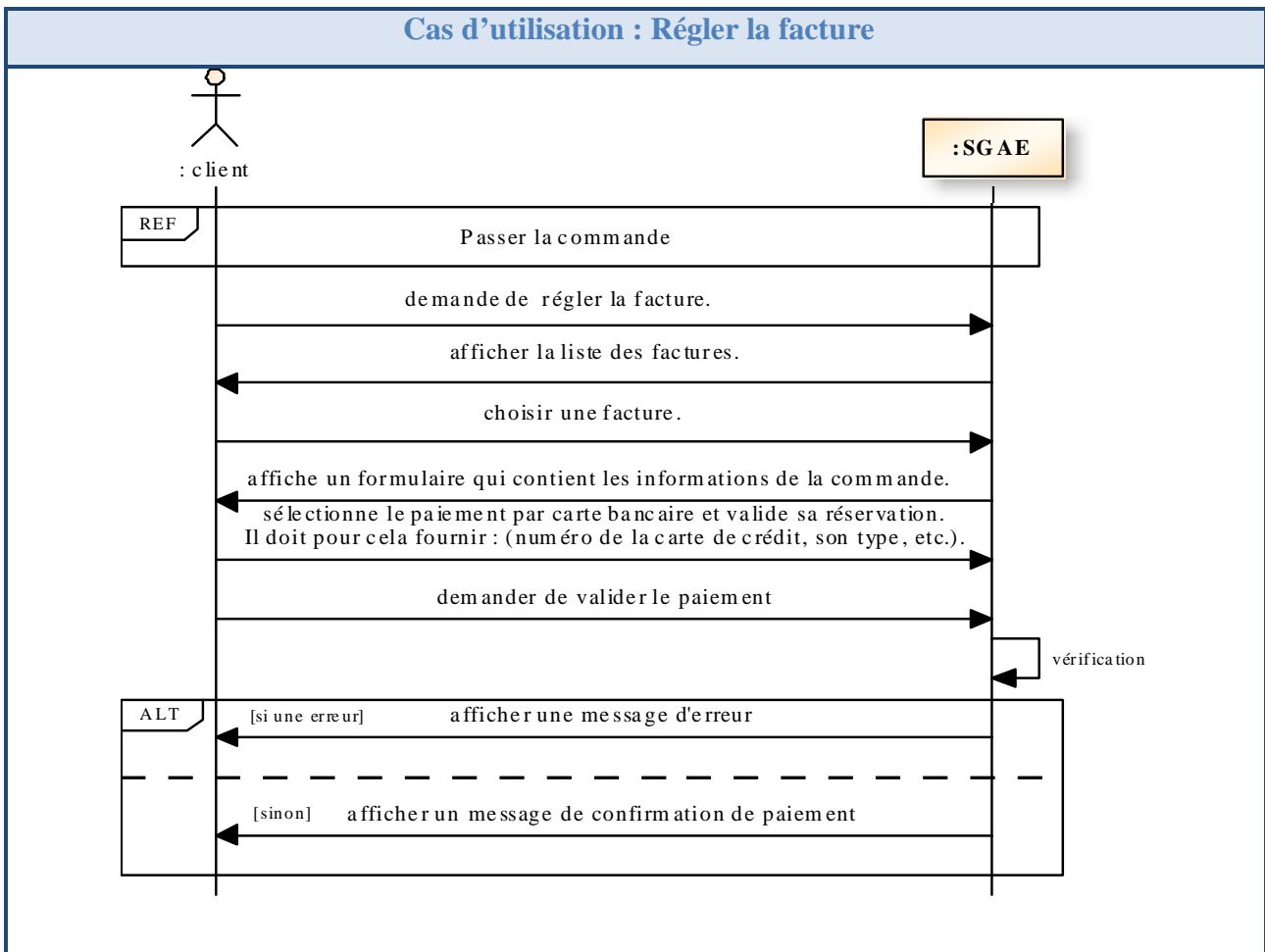
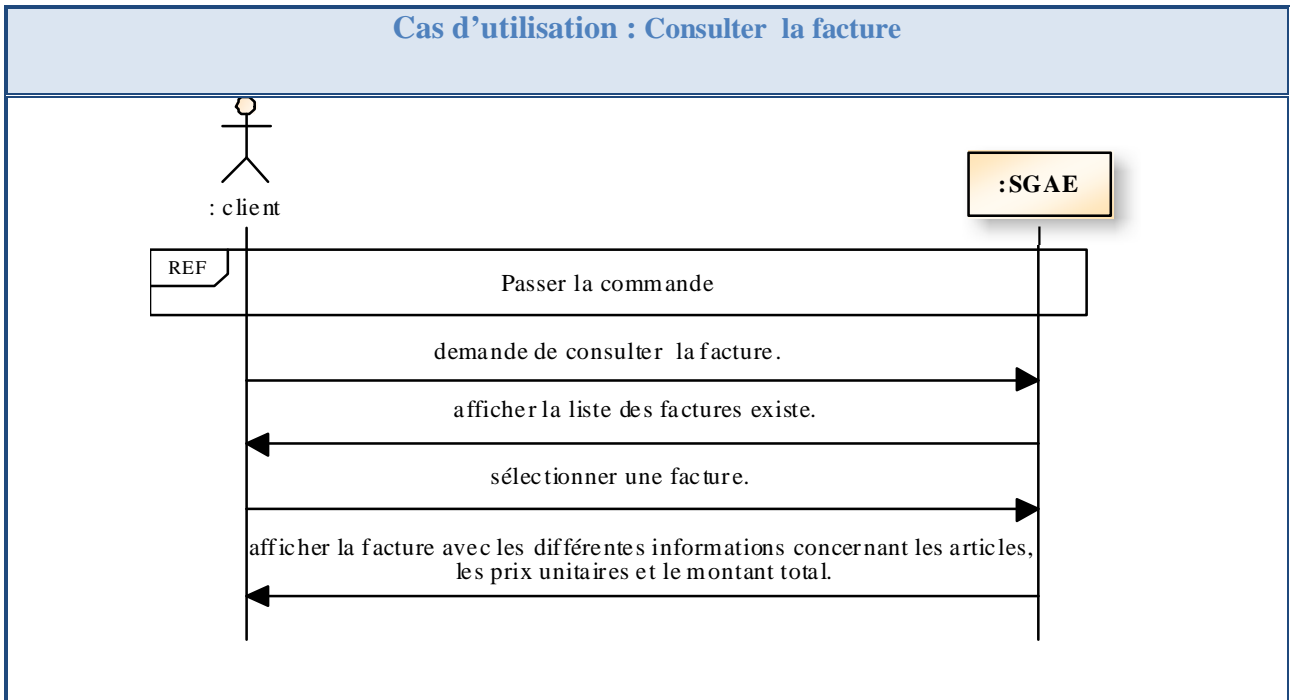
Cas d'utilisation : Gérer le panier

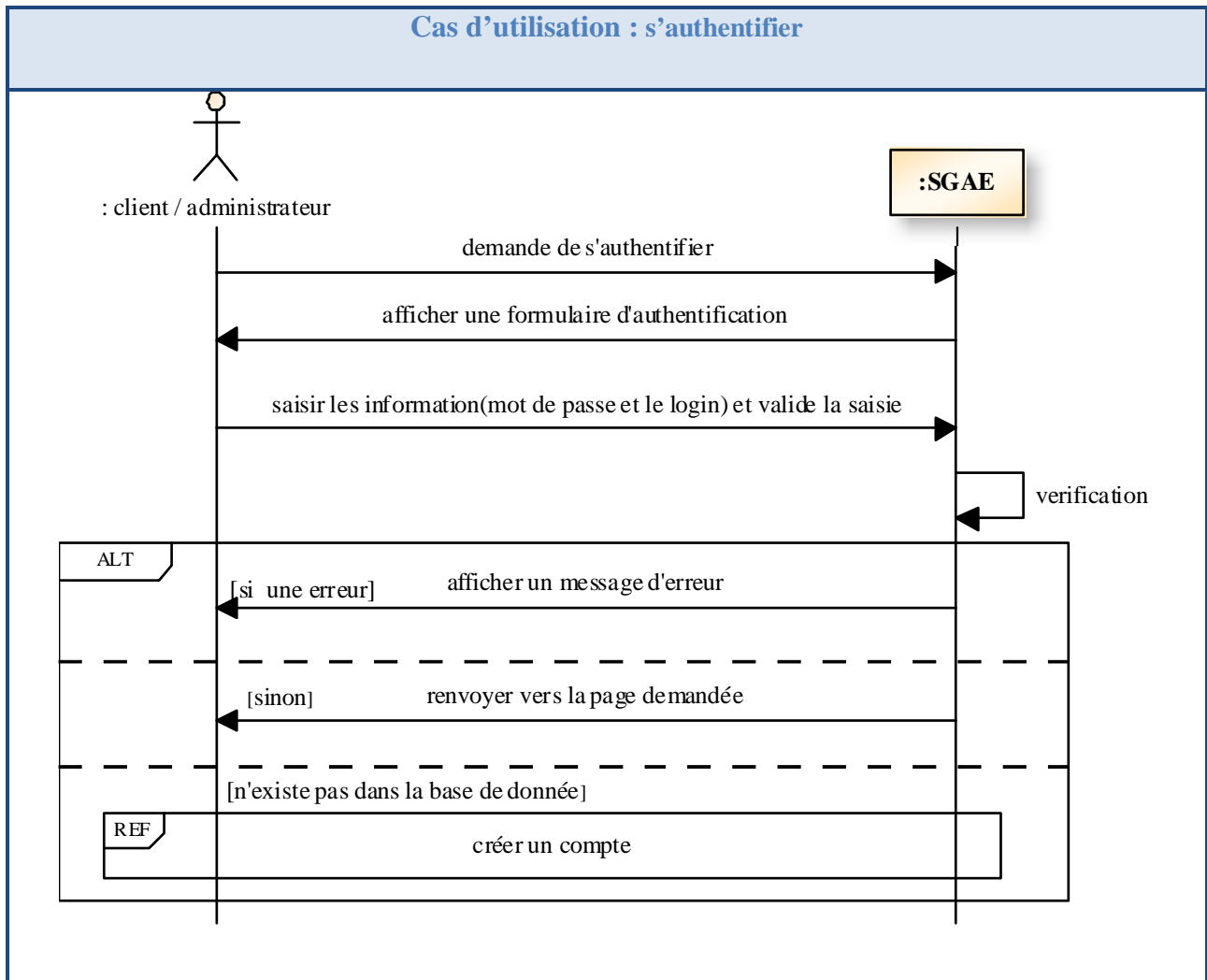


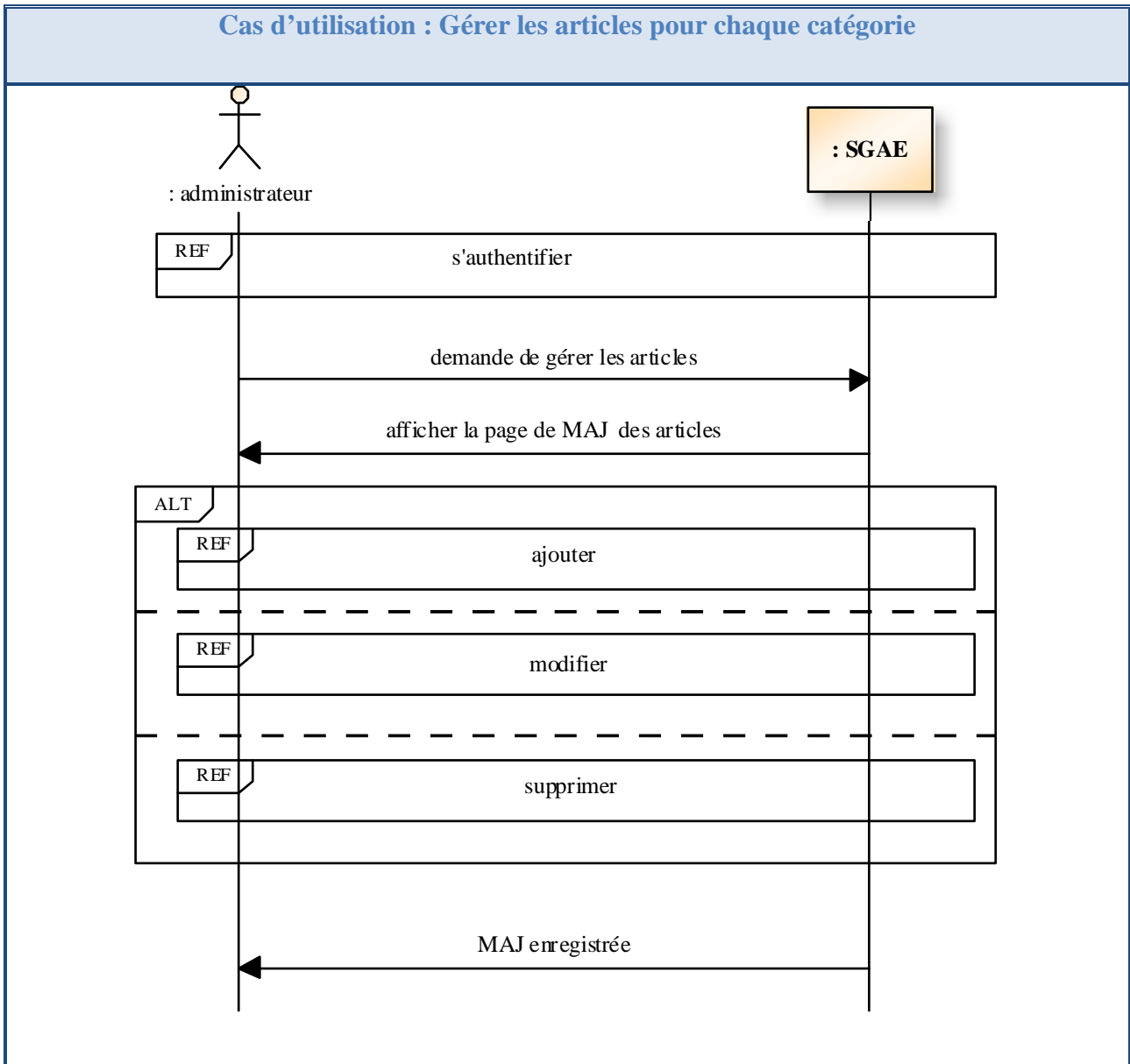
Cas d'utilisation : Créer un compte

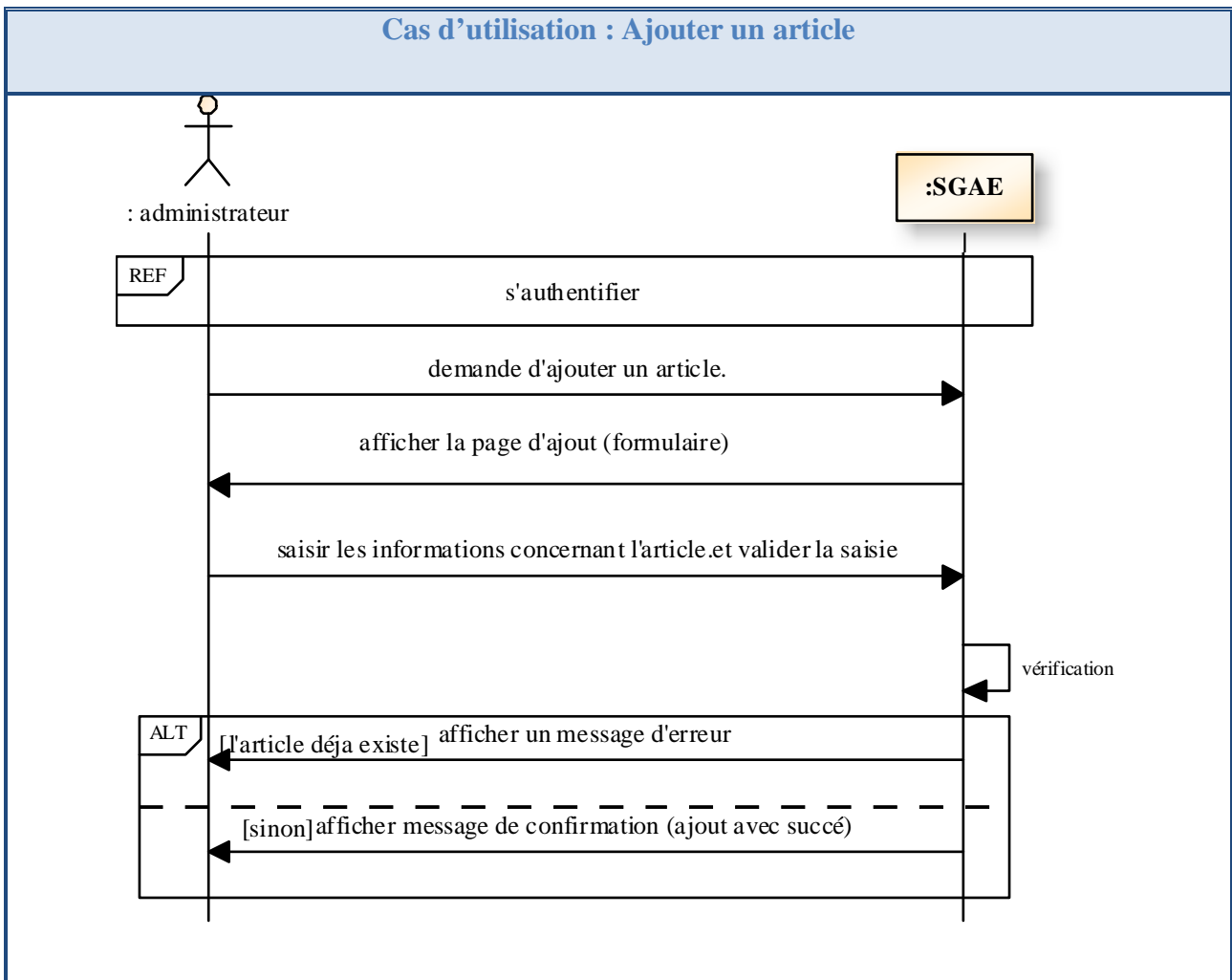


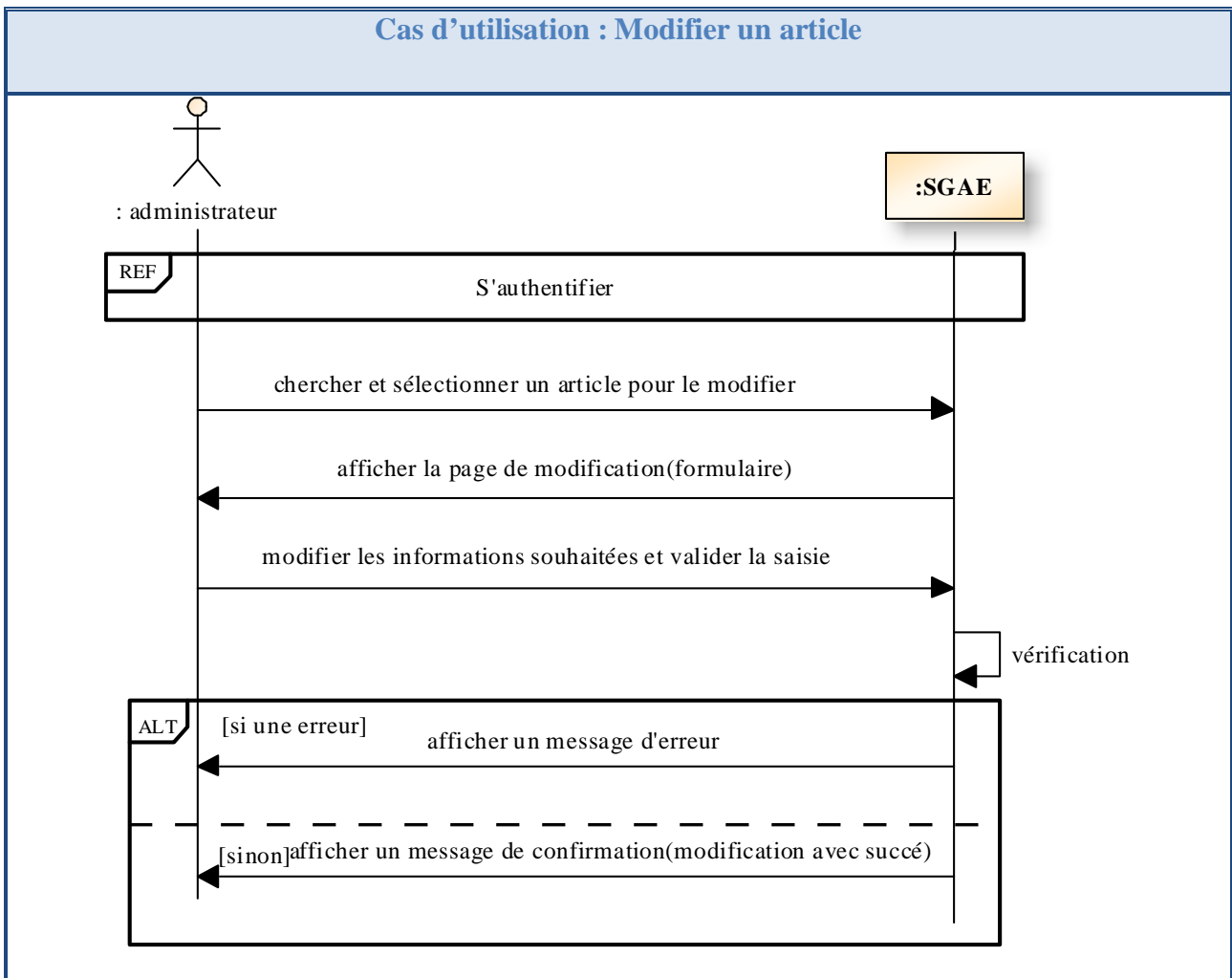


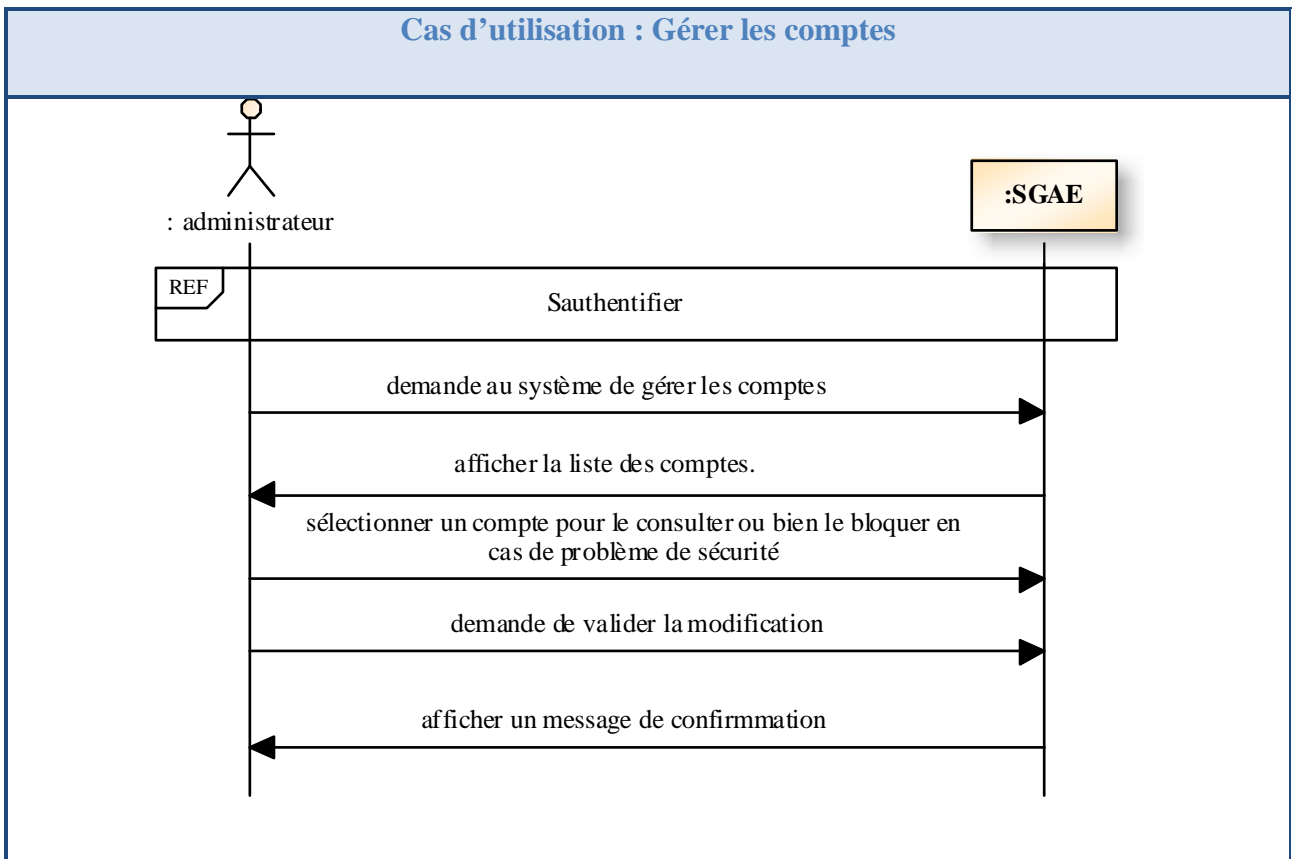
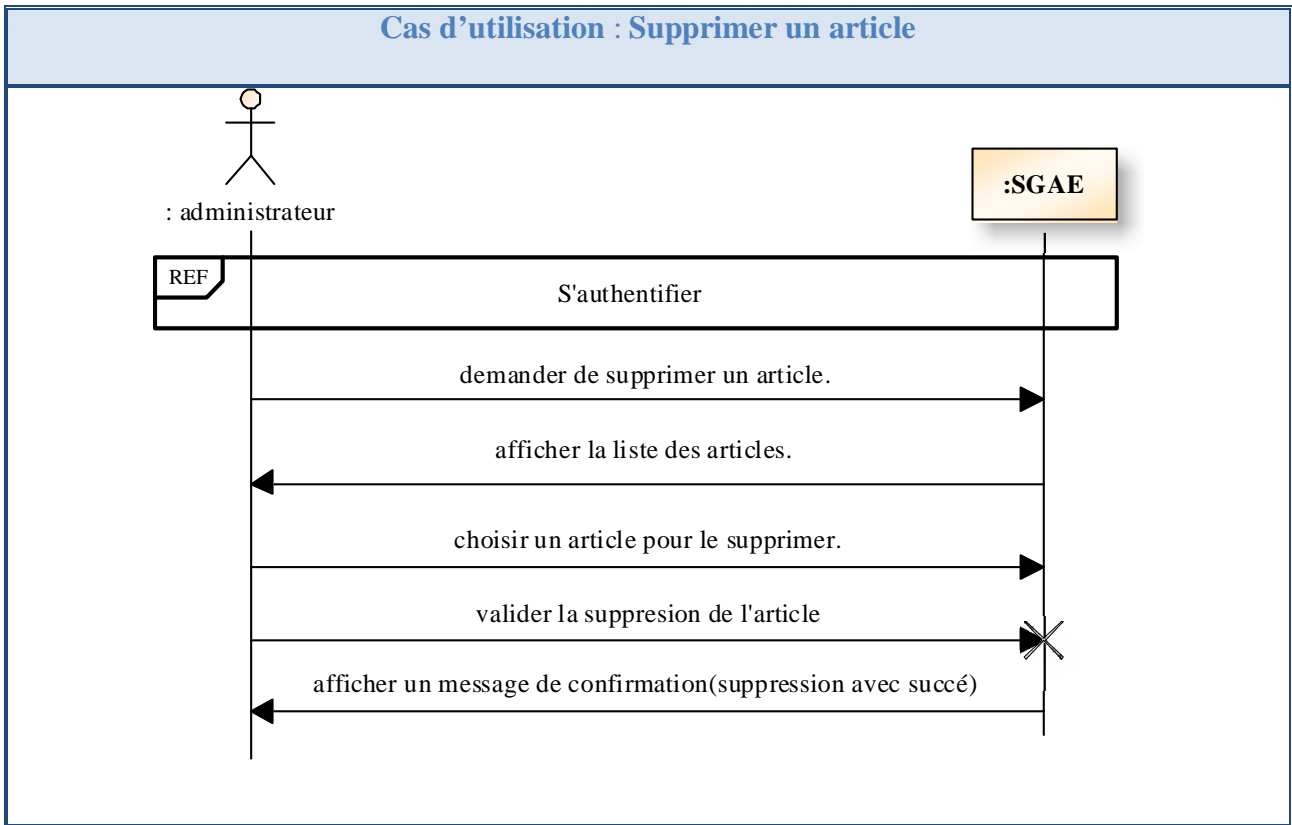












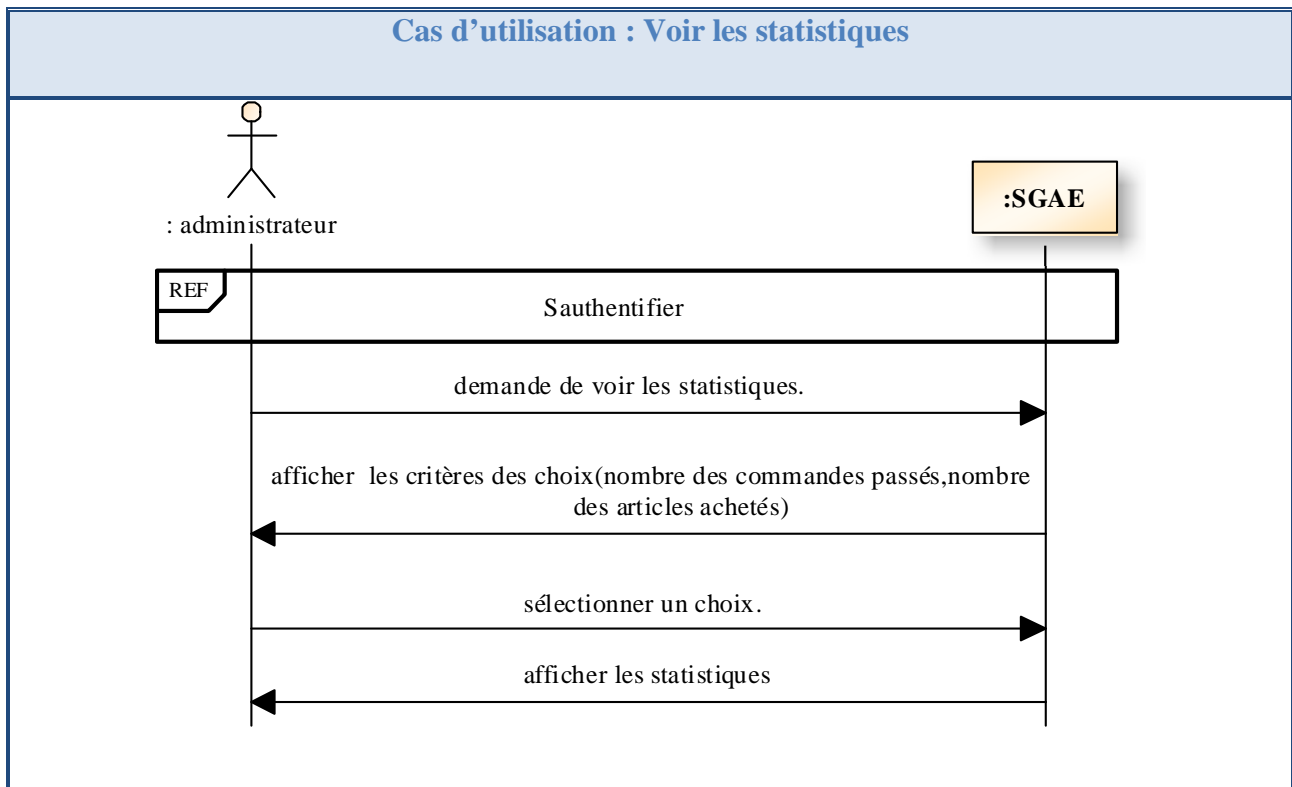
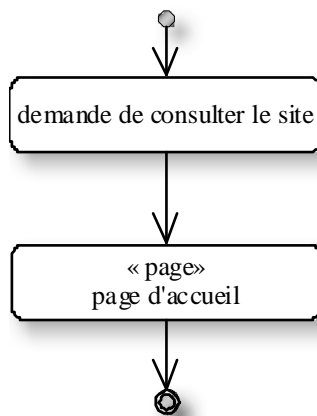


Tableau III -4: Diagramme de séquence système des cas d'utilisation fonctionnels du système **SGAE**

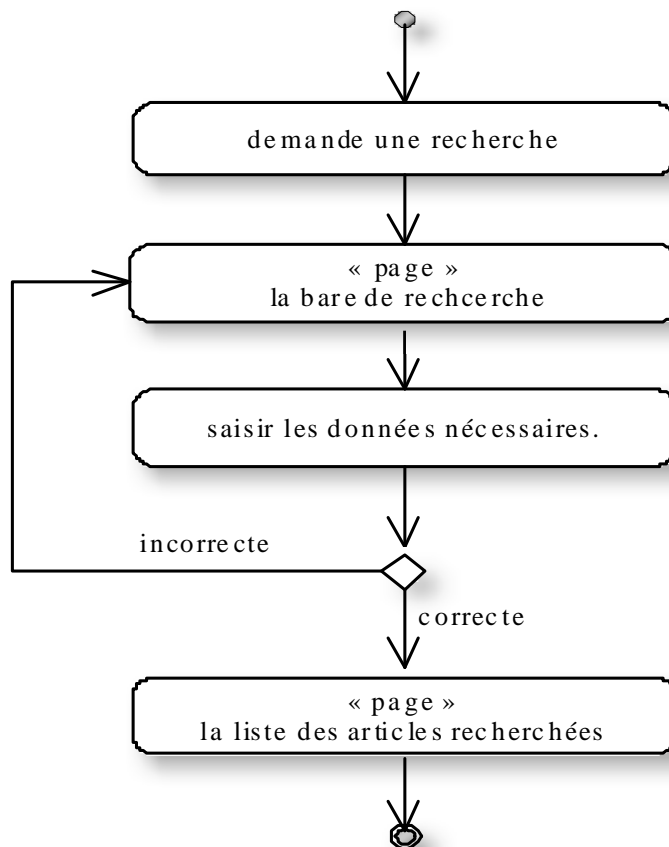
➤ **Diagramme d'activité :**

Pour documenter les cas d'utilisations, nous avons choisi le diagramme de d'activités, car il permet de consolider les enchaînements de la fiche textuelle, comme nous l'avions représenté informellement. Ce diagramme est également très utile en cas d'actions parallèles. De plus, les utilisateurs le comprennent aisément, car il ressemble à un organigramme traditionnel. Il permet enfin d'identifier d'un seul coup d'œil la famille des scénarios d'un cas d'utilisation qui décrivent toutes les réactions du système. Il suffit en effet de dessiner les différents chemins du diagramme d'activité qui passent par toutes les transitions entre actions.

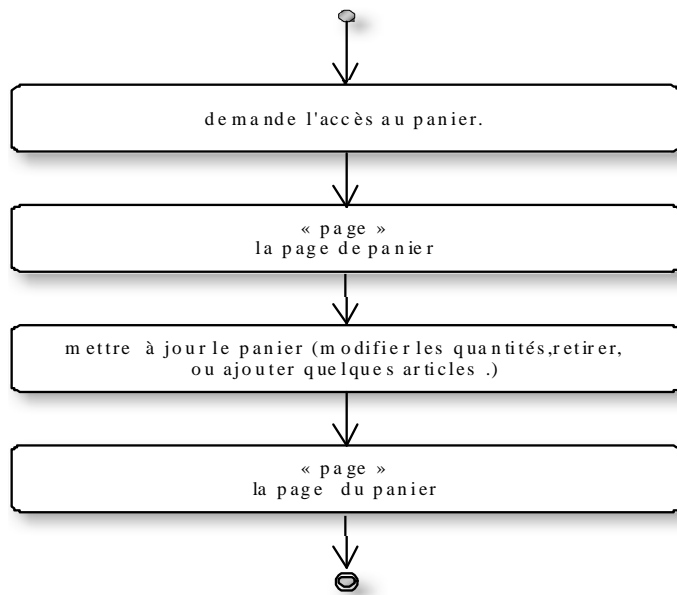
Cas d'utilisation : Consulter le catalogue



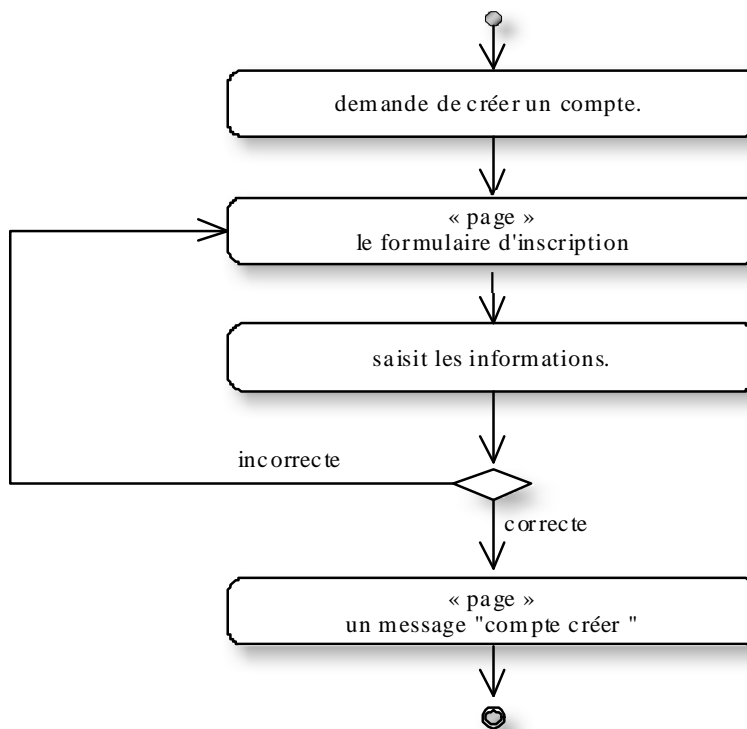
Cas d'utilisation : Rechercher un article



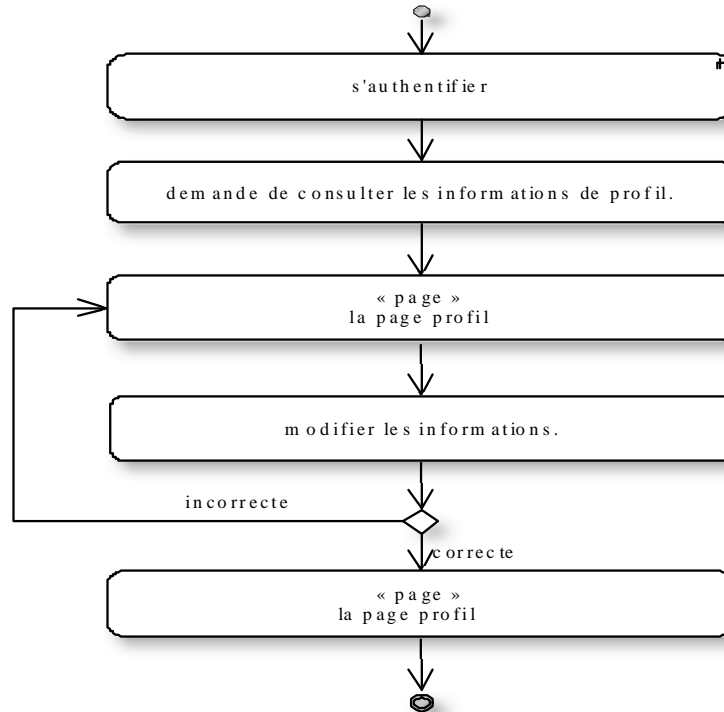
Cas d'utilisation : Gérer le panier



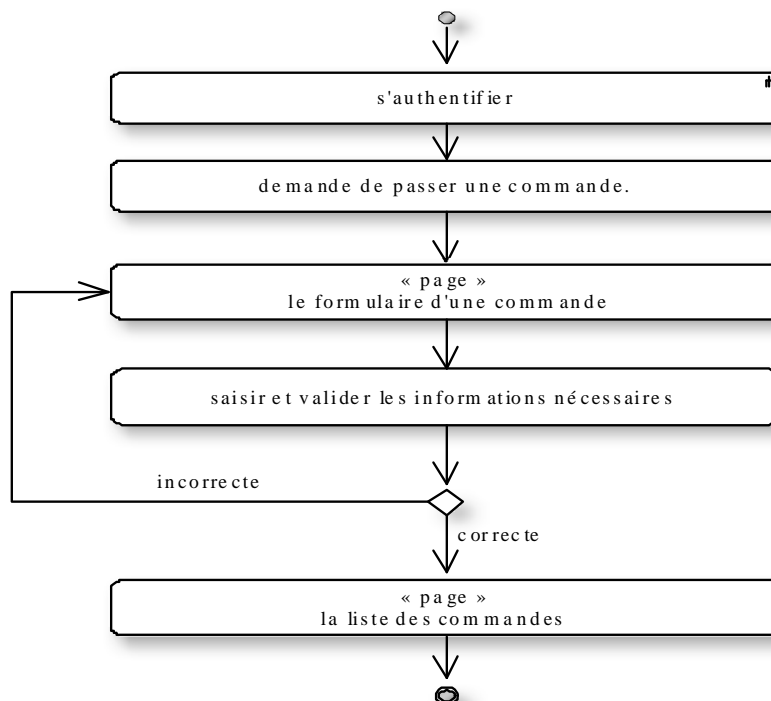
Cas d'utilisation : Créer un compte



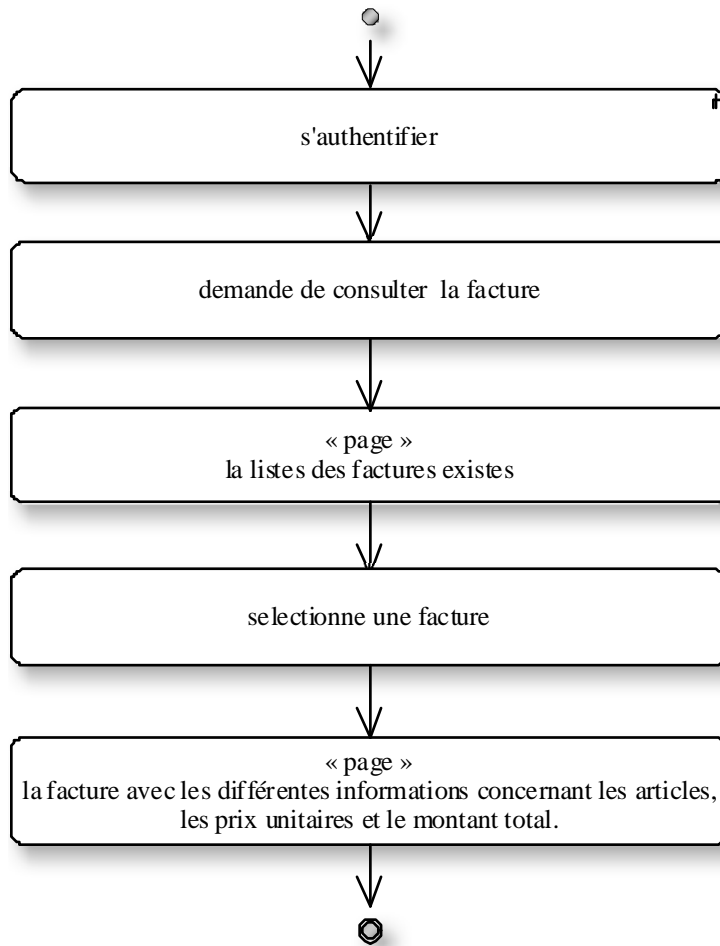
Cas d'utilisation : Gérer le profil



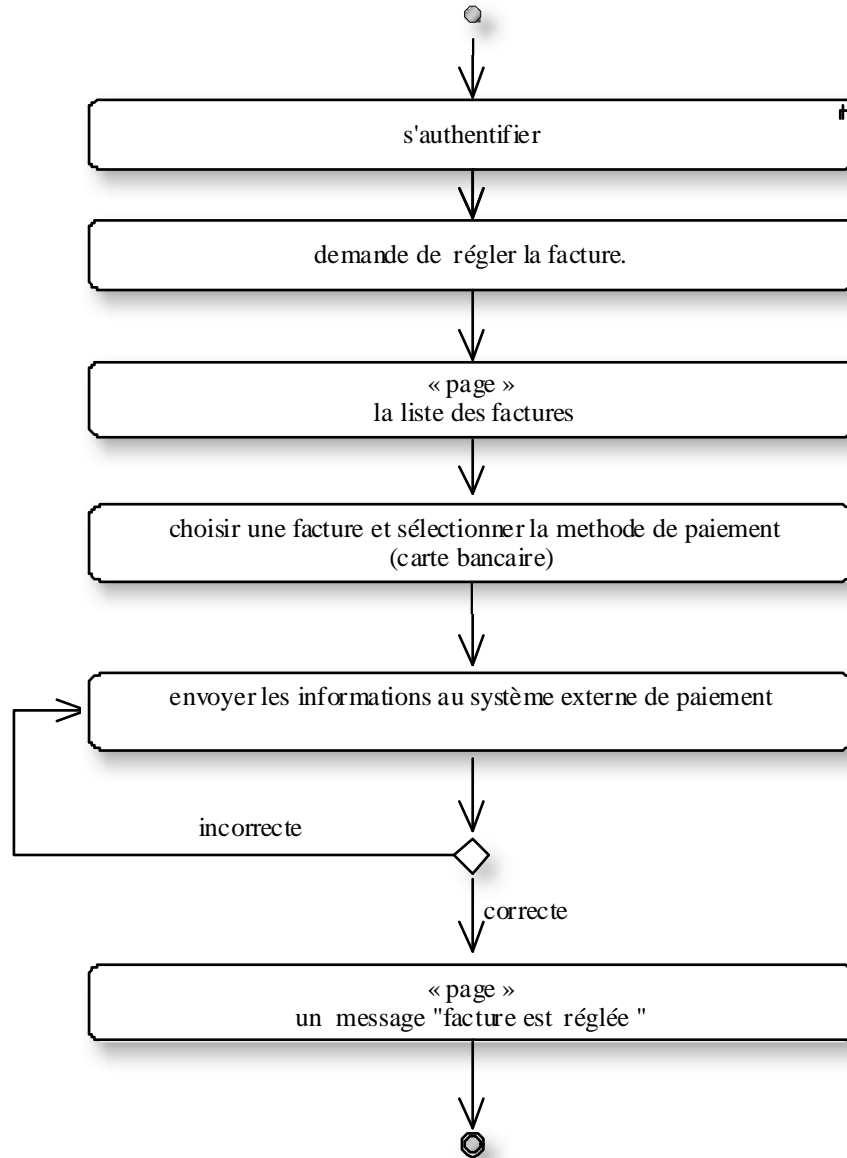
Cas d'utilisation : Passer la commande



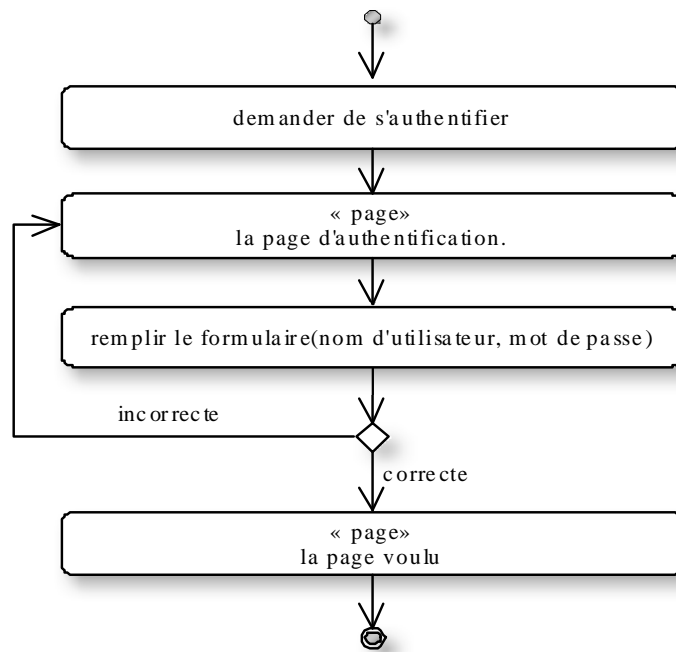
Cas d'utilisation : Consulter la facture



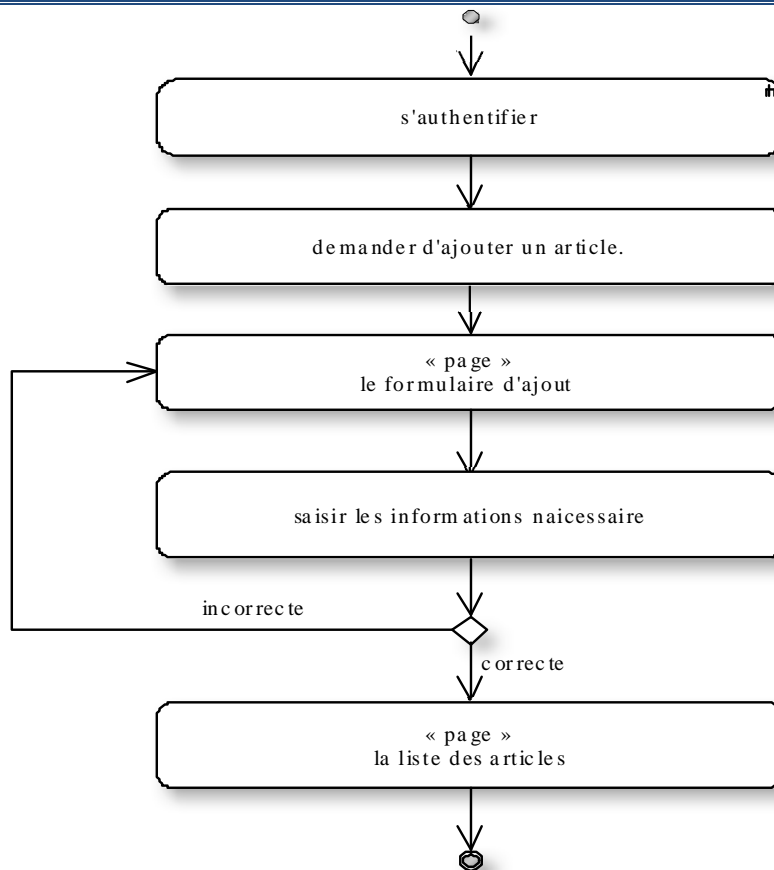
Cas d'utilisation : Régler la facture



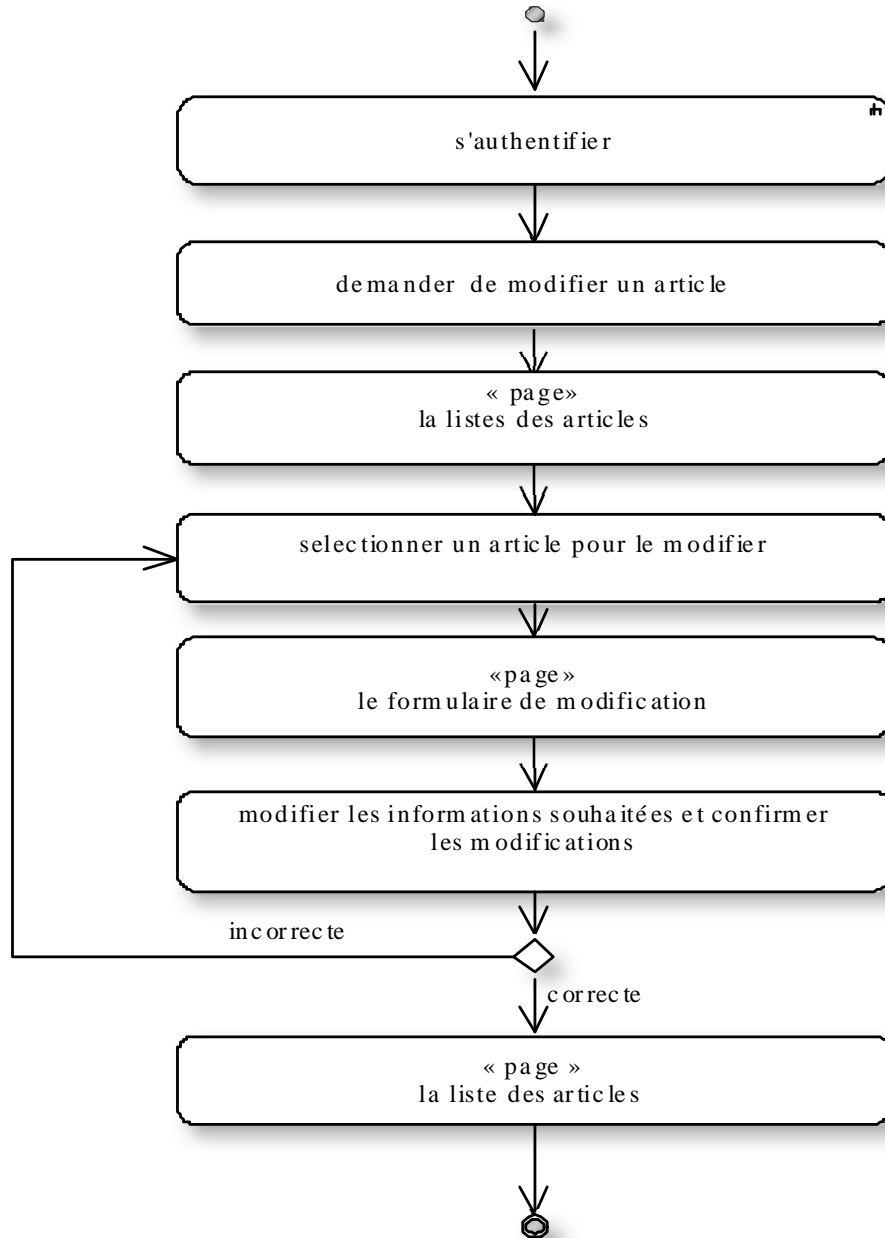
Cas d'utilisation : s'authentifier



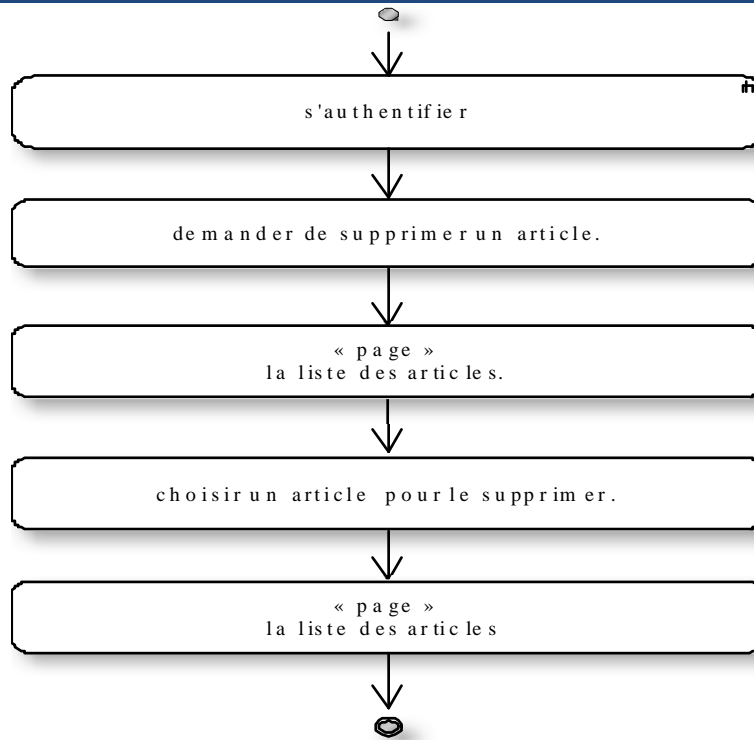
Cas d'utilisation : Ajouter un article



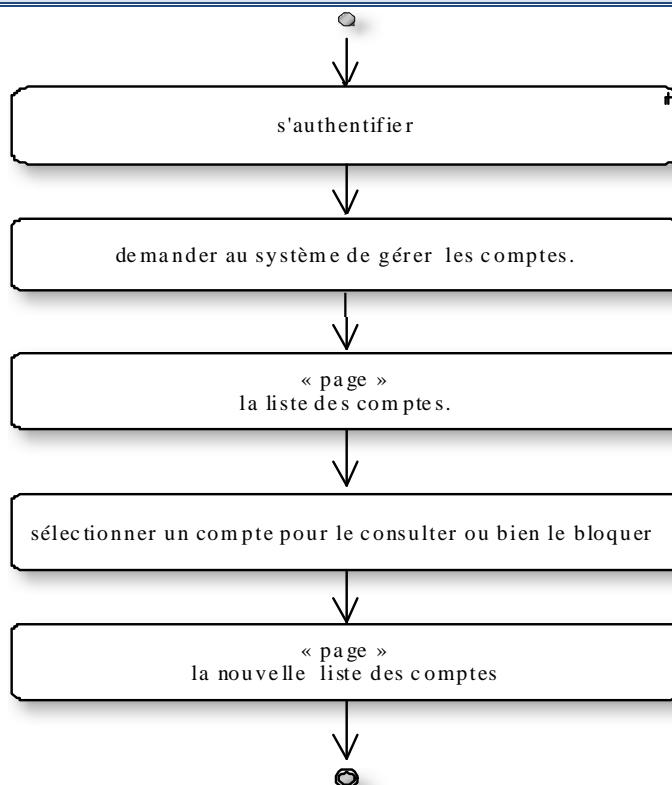
Cas d'utilisation : Modifier un article



Cas d'utilisation : Supprimer un article



Cas d'utilisation : Gérer les comptes



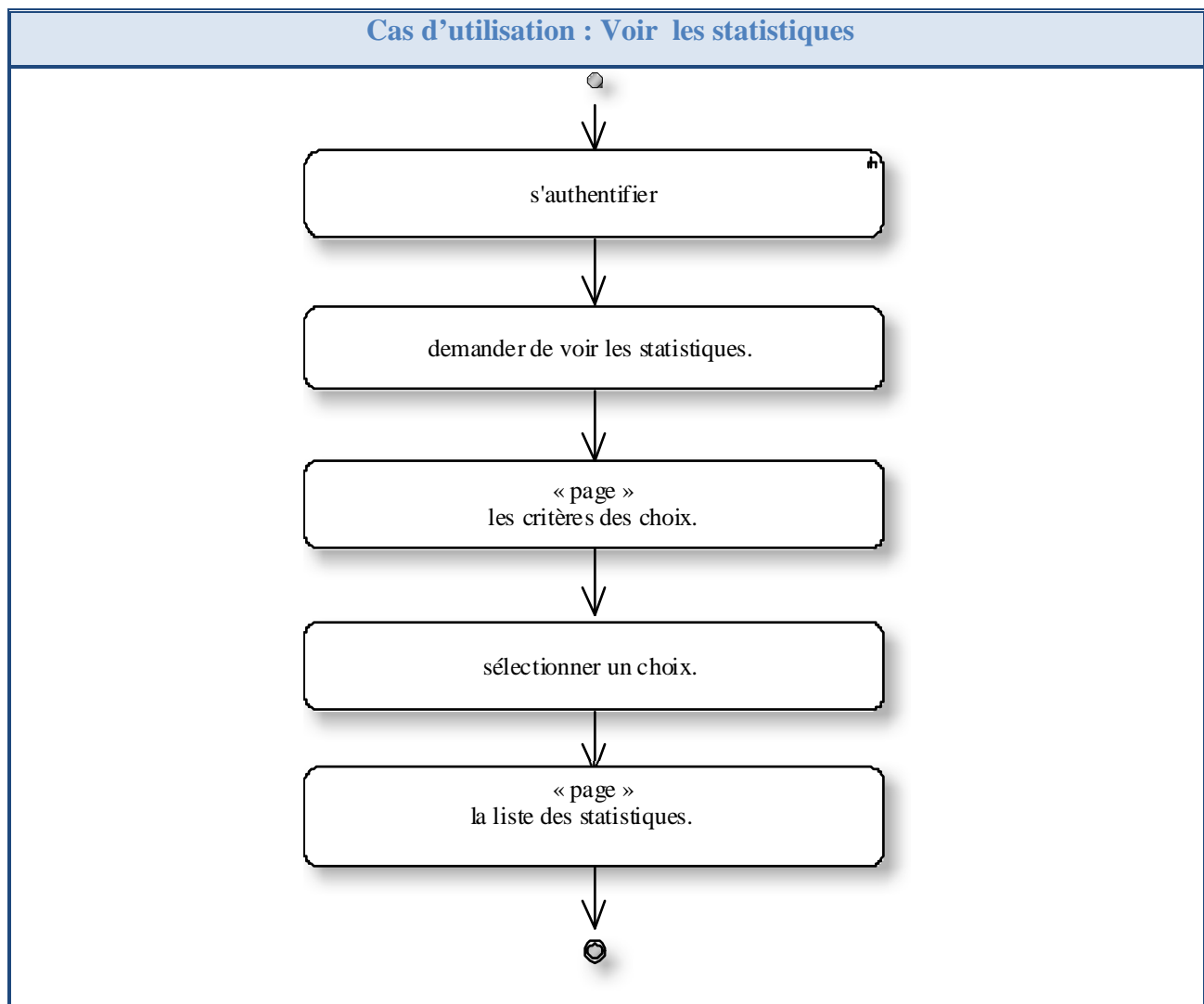


Tableau III -5: Diagramme d'activité des cas d'utilisation fonctionnels du système SGAE

2.3. Identification des classes candidates:

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse. La technique utilisée pour identifier les classes candidates est la suivante :

- Chercher les noms communs importants dans les descriptions textuelles des cas d'utilisation.
- Vérifier les propriétés « objet » de chaque concept (identité, propriétés, comportement), puis définir ses responsabilités.

2.3.1. Définition :

Une responsabilité est une sorte de contrat, ou obligation, pour une classe. Elle se place à un

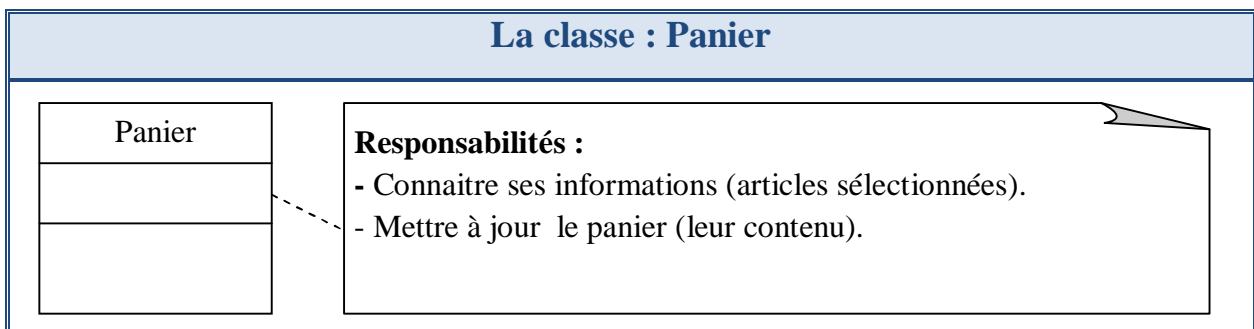
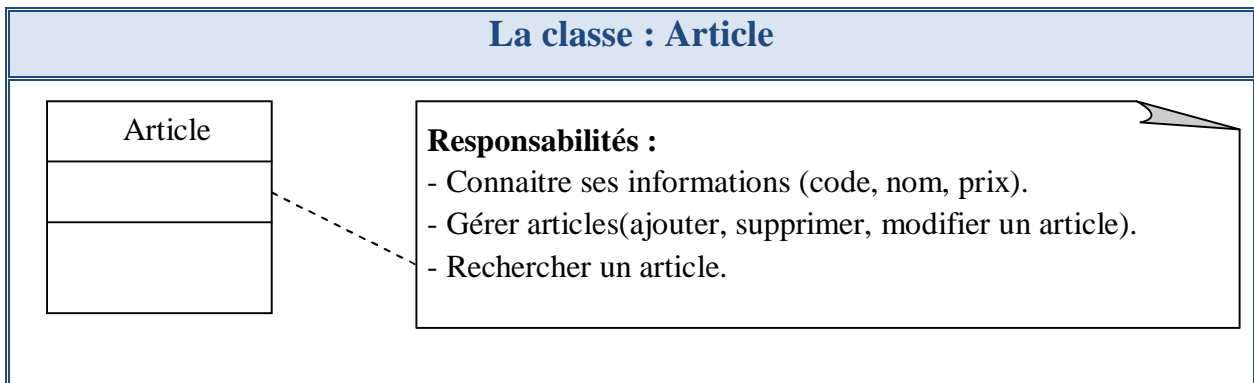
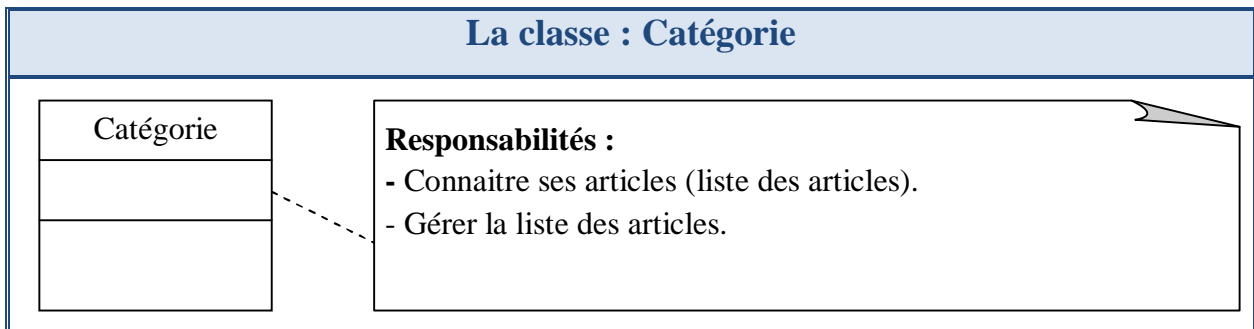
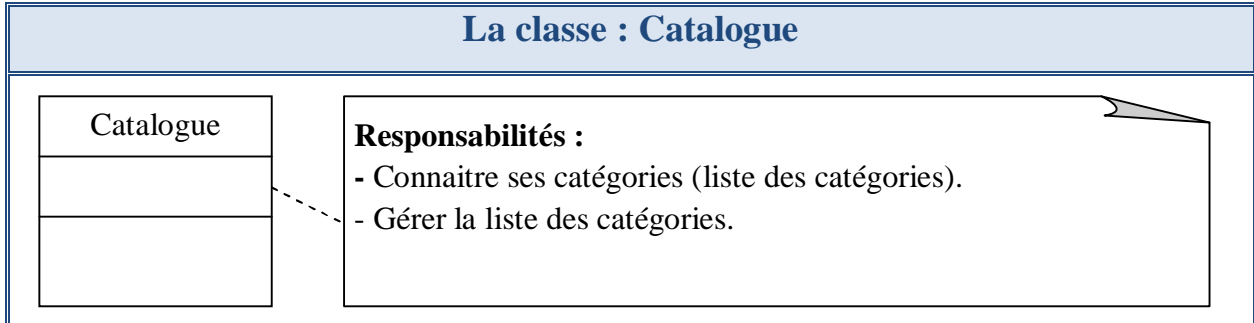
niveau d'abstraction plus élevé que les attributs ou les opérations. On formalise ensuite ces concepts métier sous forme de classes et d'associations rassemblées dans un diagramme statique pour chaque cas d'utilisation.

Ces diagrammes préliminaires sont appelés « diagrammes des classes participantes », n'ont pas pour objectif d'être complet. Ils servent uniquement à démarrer la découverte des classes du modèle d'analyse. [Roques07]

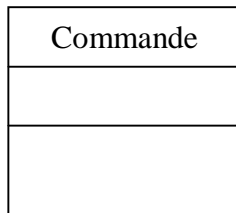
2.3.2. La liste des classes candidates :

La liste des classes candidates
✓ Catalogue
✓ Catégorie
✓ Article
✓ Panier
✓ Commande
✓ Facture
✓ Client
✓ Administrateur
✓ Role
✓ LigneCommande

2.3.3. Responsabilités des classes :



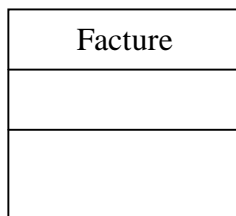
La classe : Commande



Responsabilités :

- Passer une commande.
- Connaitre ses informations (numéro commande, date...).
- Connaitre l'état d'une commande.

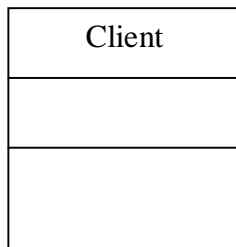
La classe : Facture



Responsabilités :

- Connaitre ses informations (montant total).
- Régler la facture.

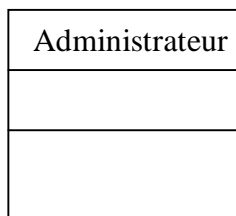
La classe : Client



Responsabilités :

- Connaitre ses informations (login, motpass, nom, prénom ...).
- Gérer le profil

La classe : Administrateur



Responsabilités :

- Connaitre ses informations (login, motpass, nom, prénom ...).

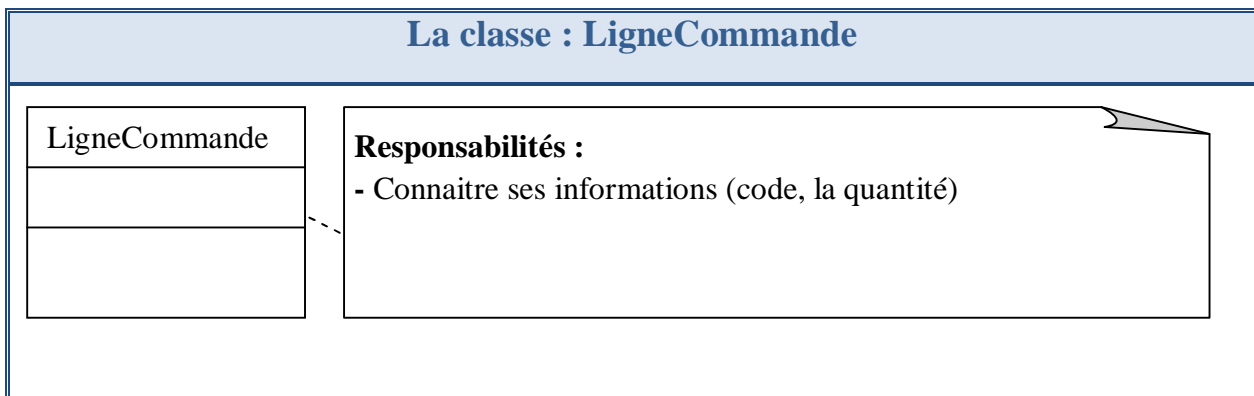
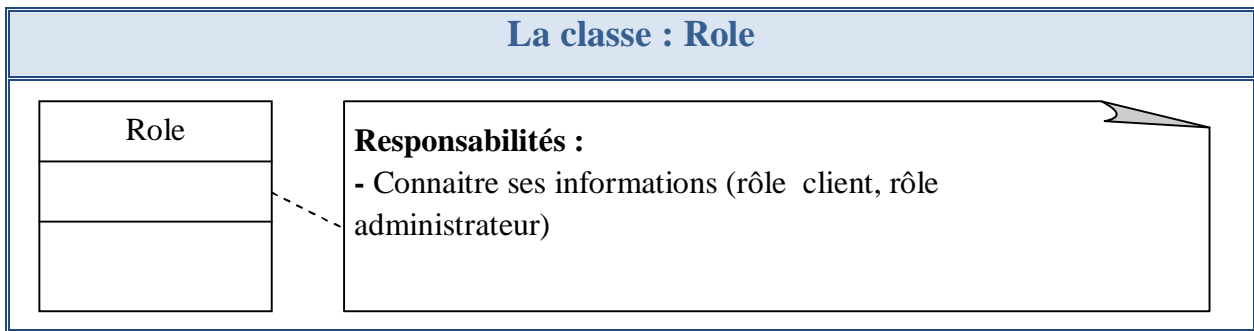


Tableau III -6: Les classes candidates et ses responsabilités

2.4. Diagramme de classes participantes :

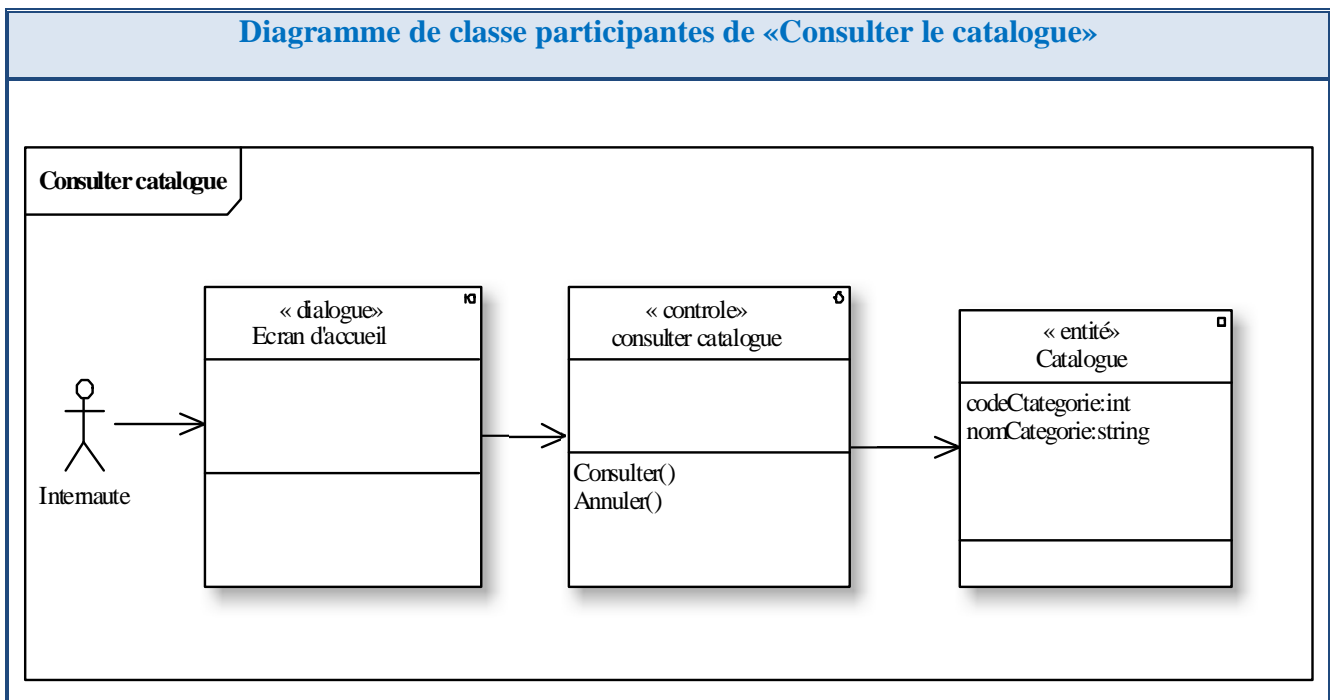


Diagramme de classe participantes de «Rechercher un article »

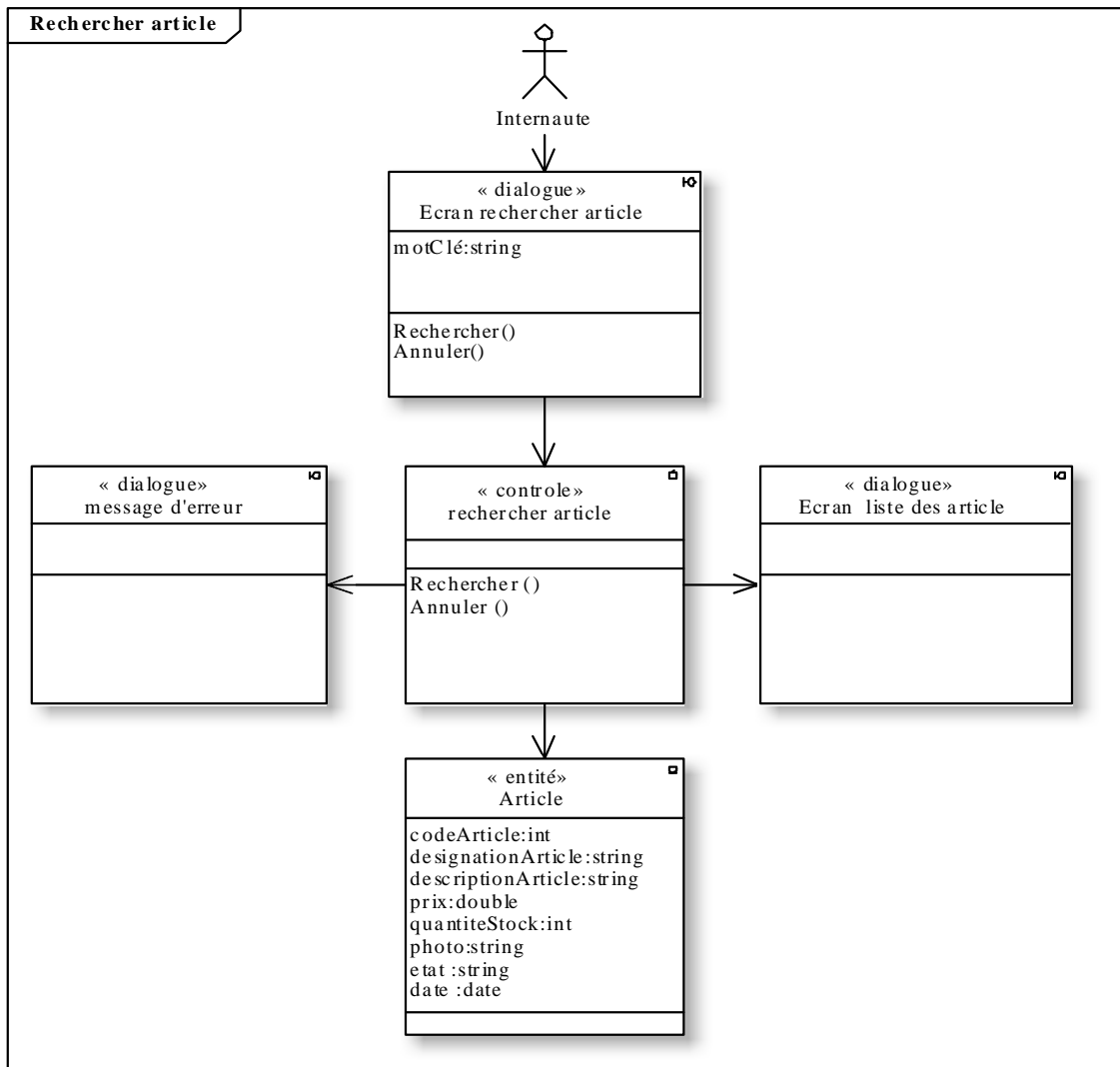


Diagramme de classe participantes de «Gérer le panier »

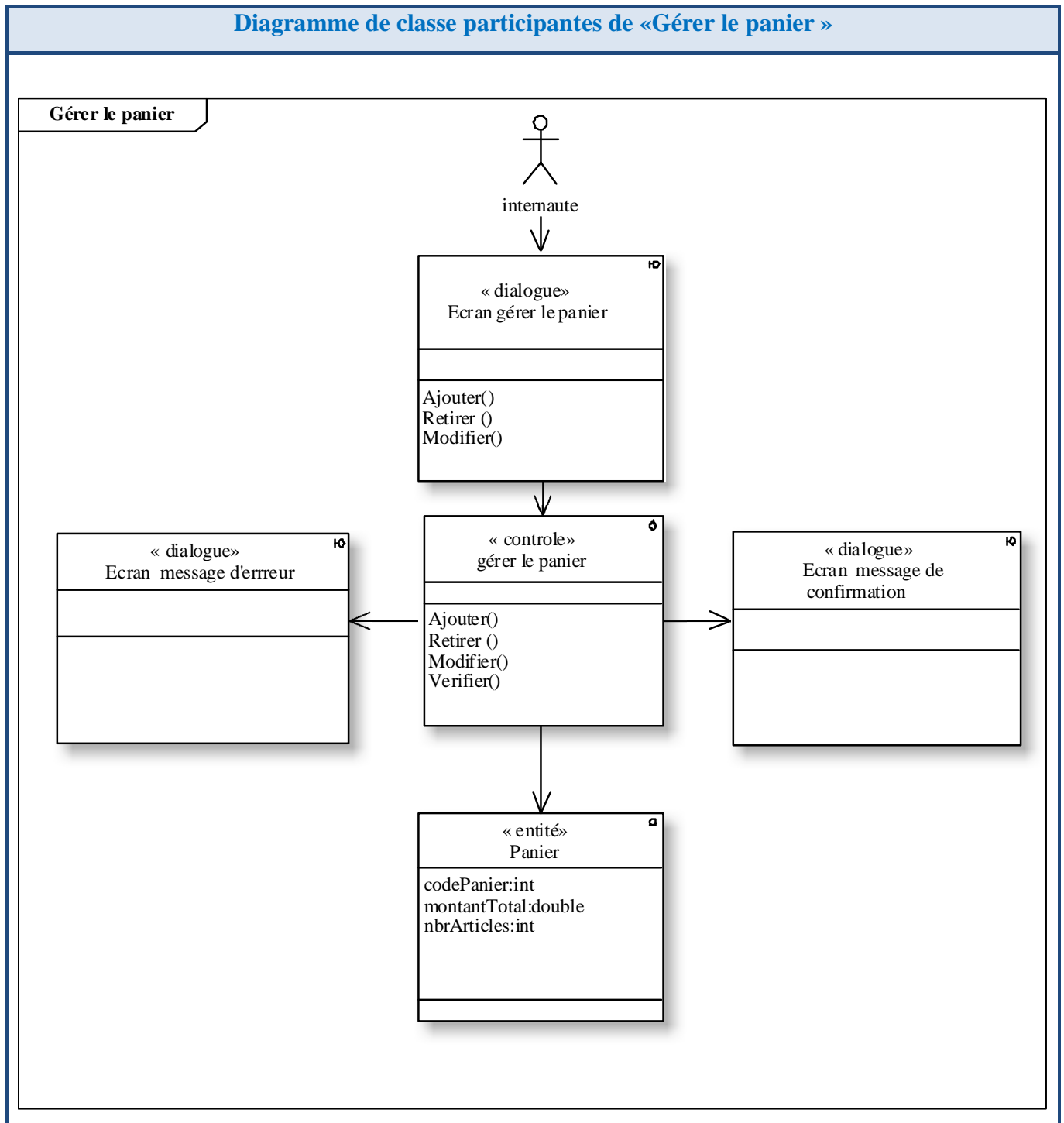


Diagramme de classe participantes de « Créer un compte »

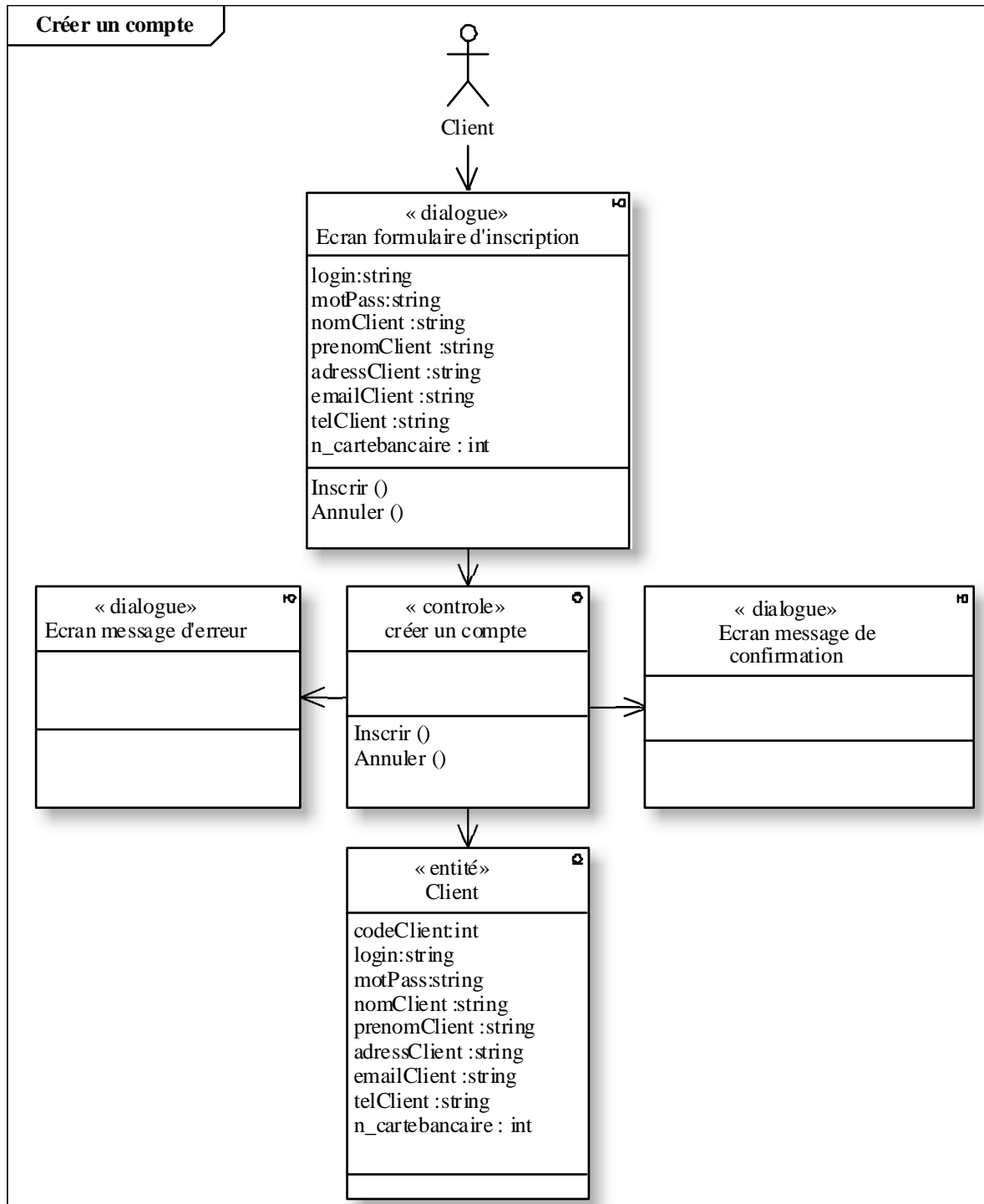


Diagramme de classe participantes de « S'authentifier »

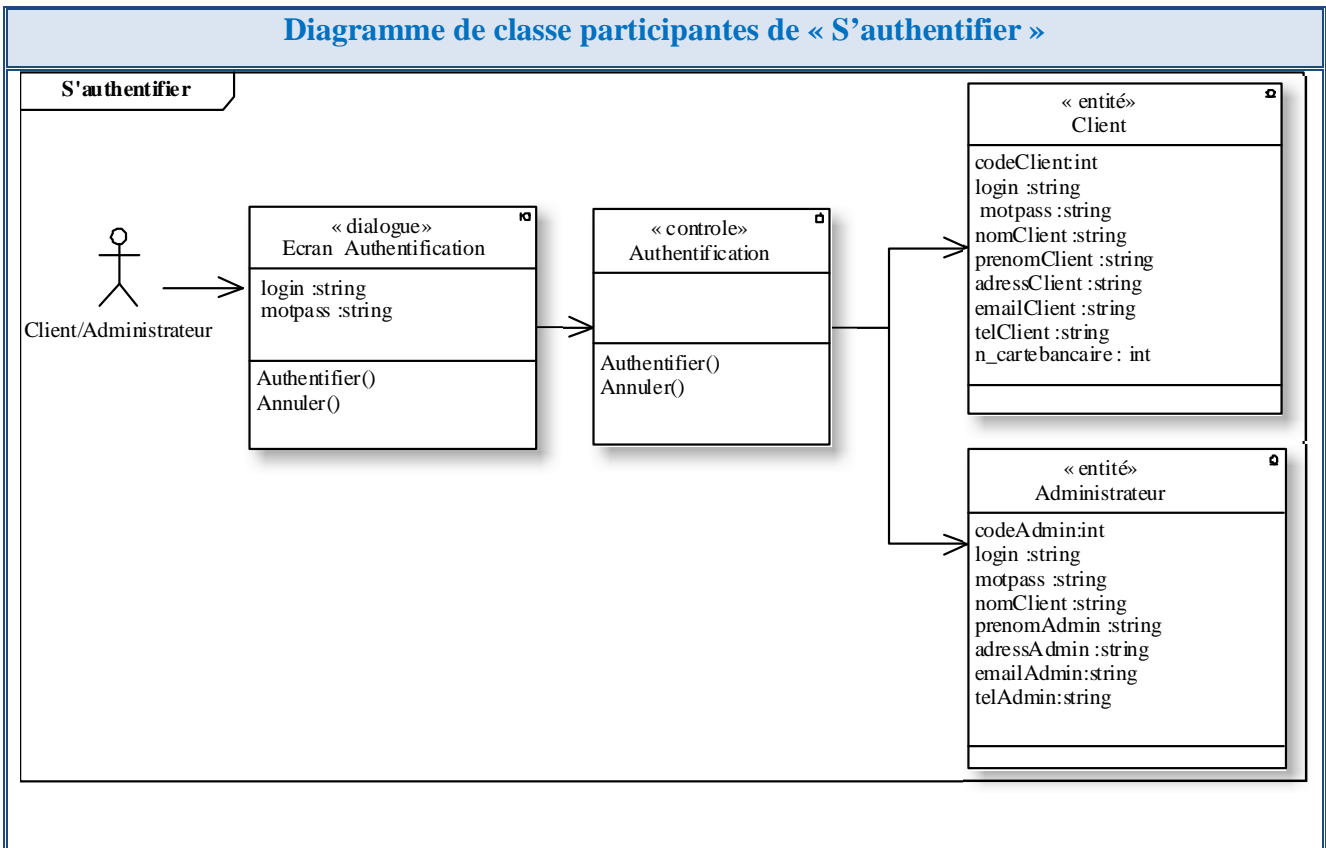


Diagramme de classe participantes de « Passer la commande »

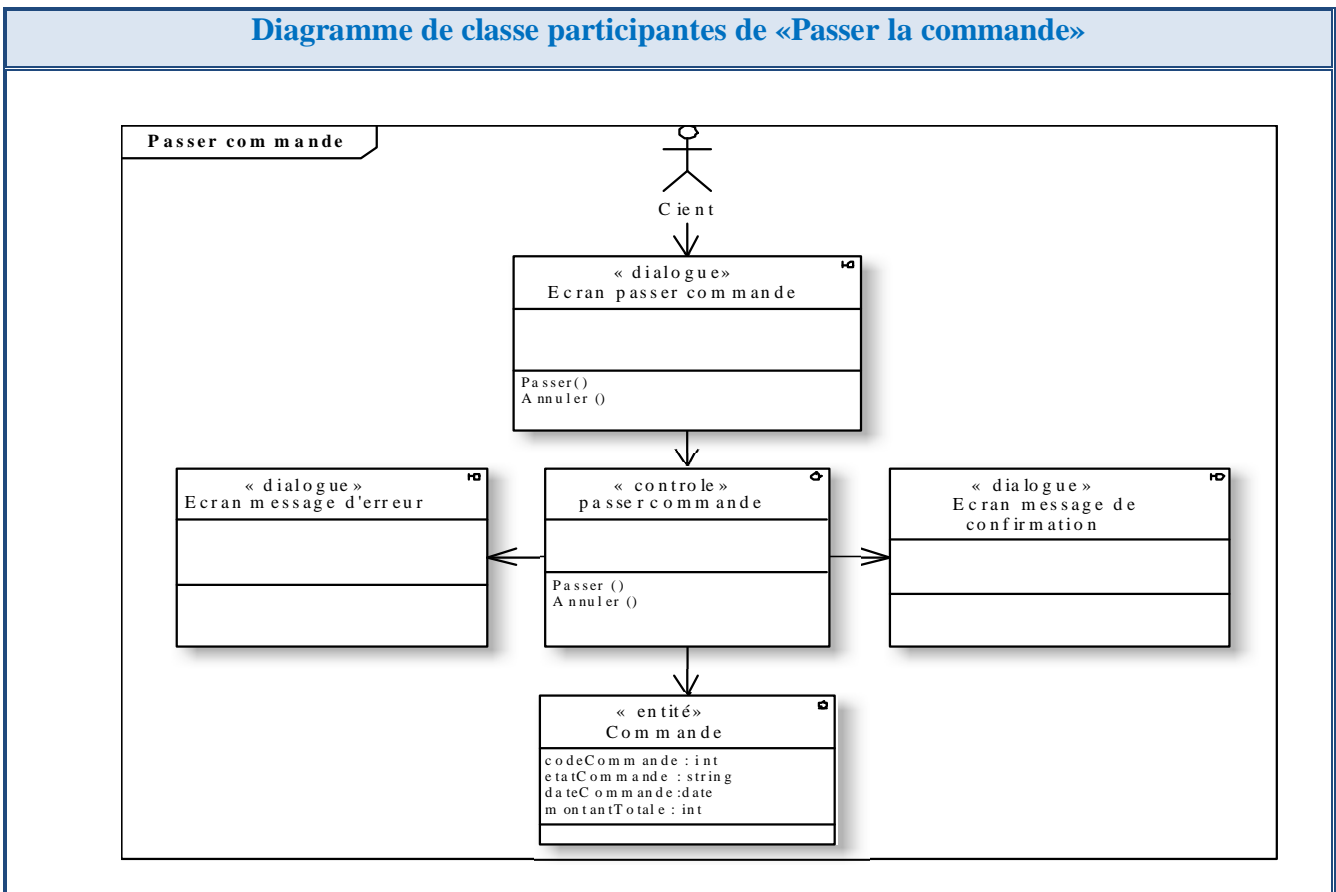


Diagramme de classe participantes de «Consulter la facture »

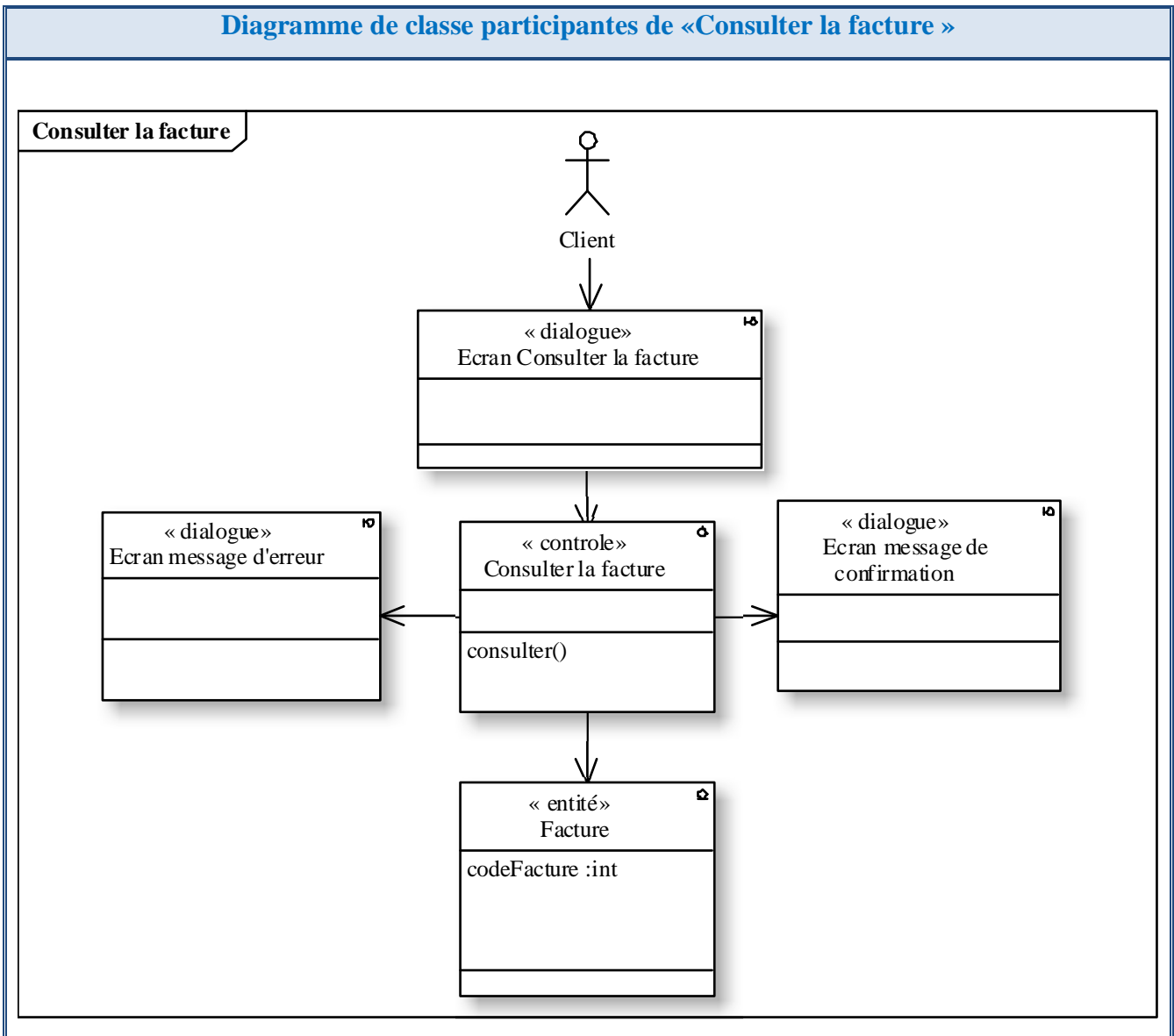


Diagramme de classe participantes de «Régler la facture »

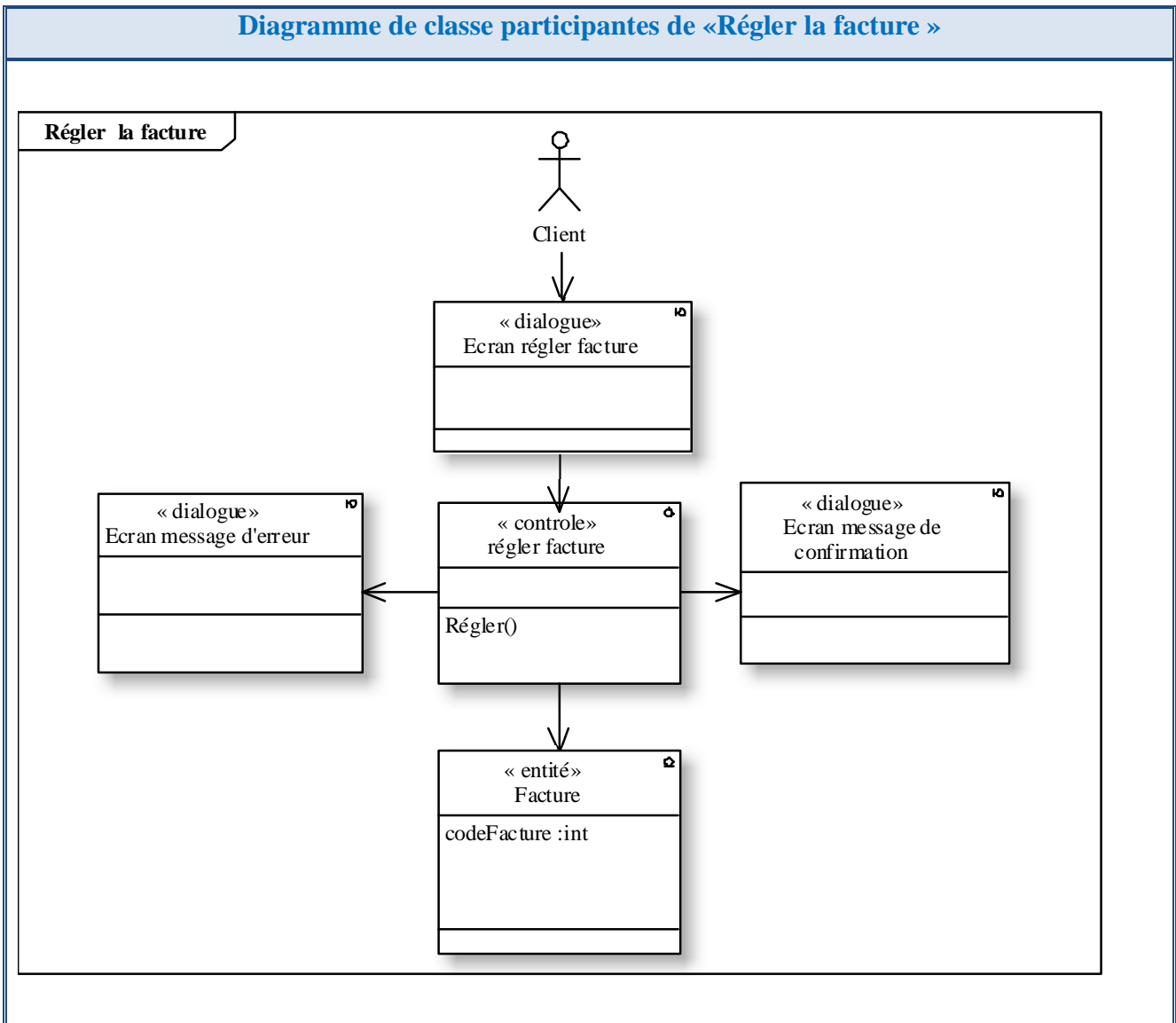


Diagramme de classe participantes de « Ajouter un article »

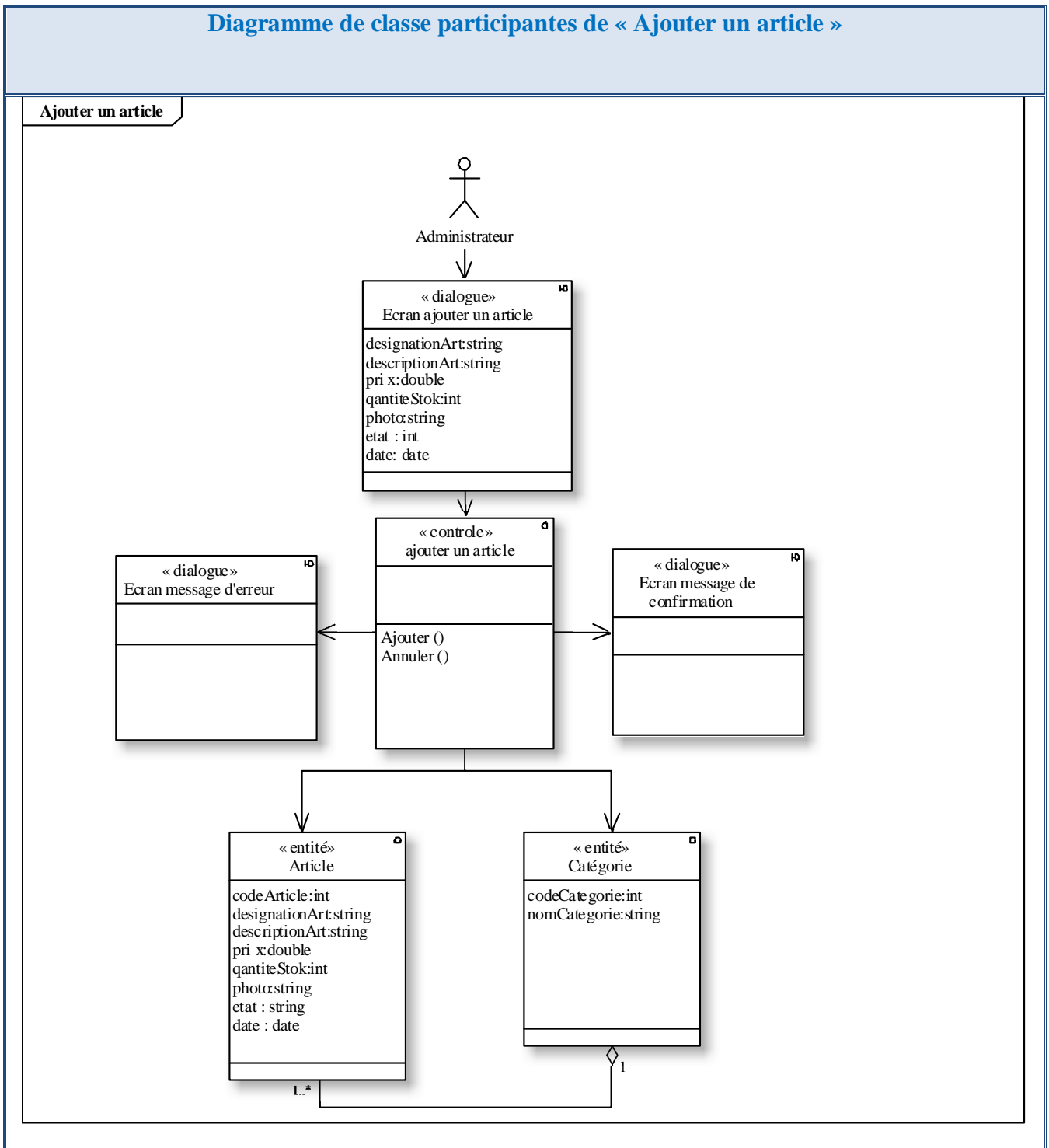


Diagramme de classe participantes de «Modifier un article»

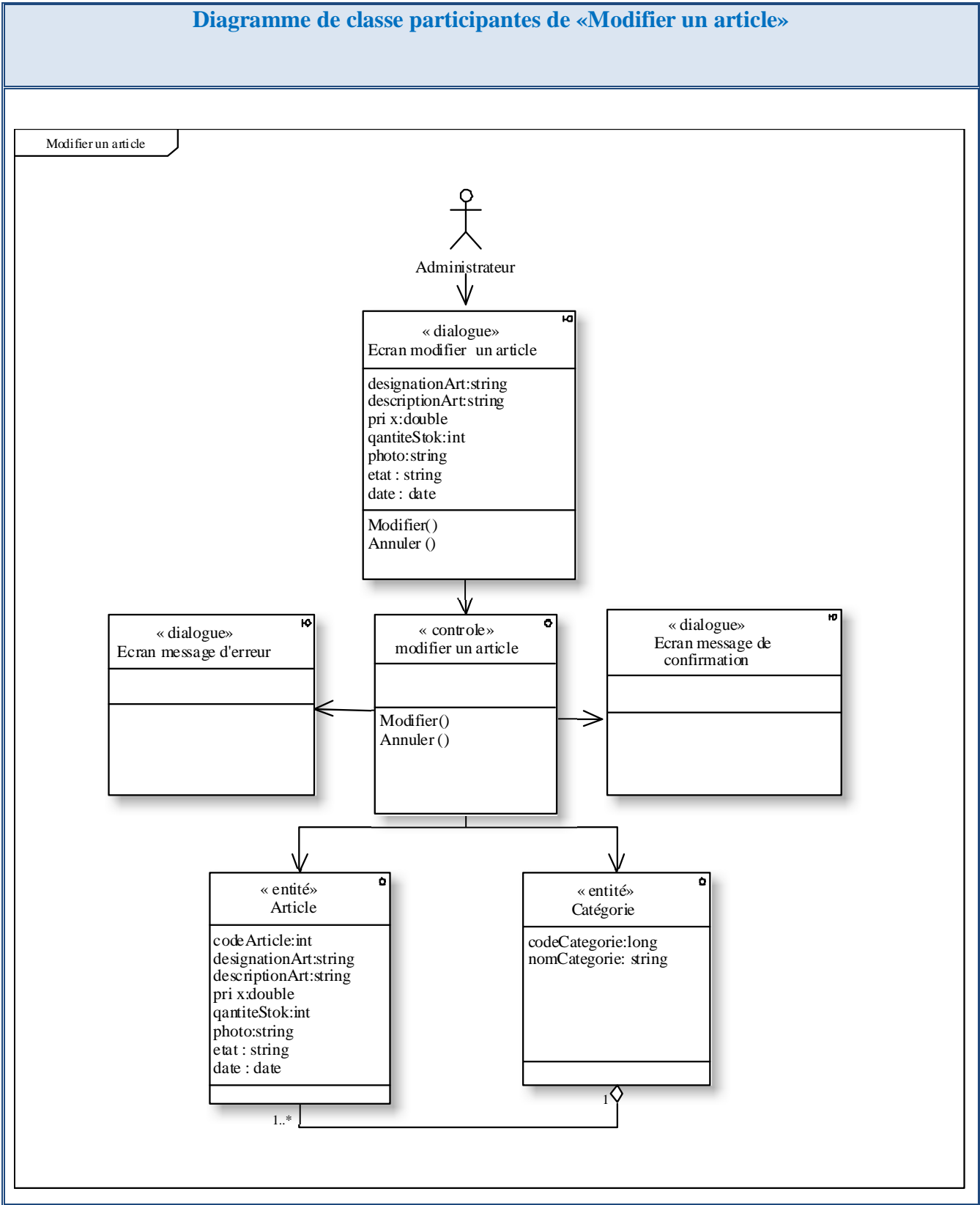


Diagramme de classe participantes de « Supprimer un article »

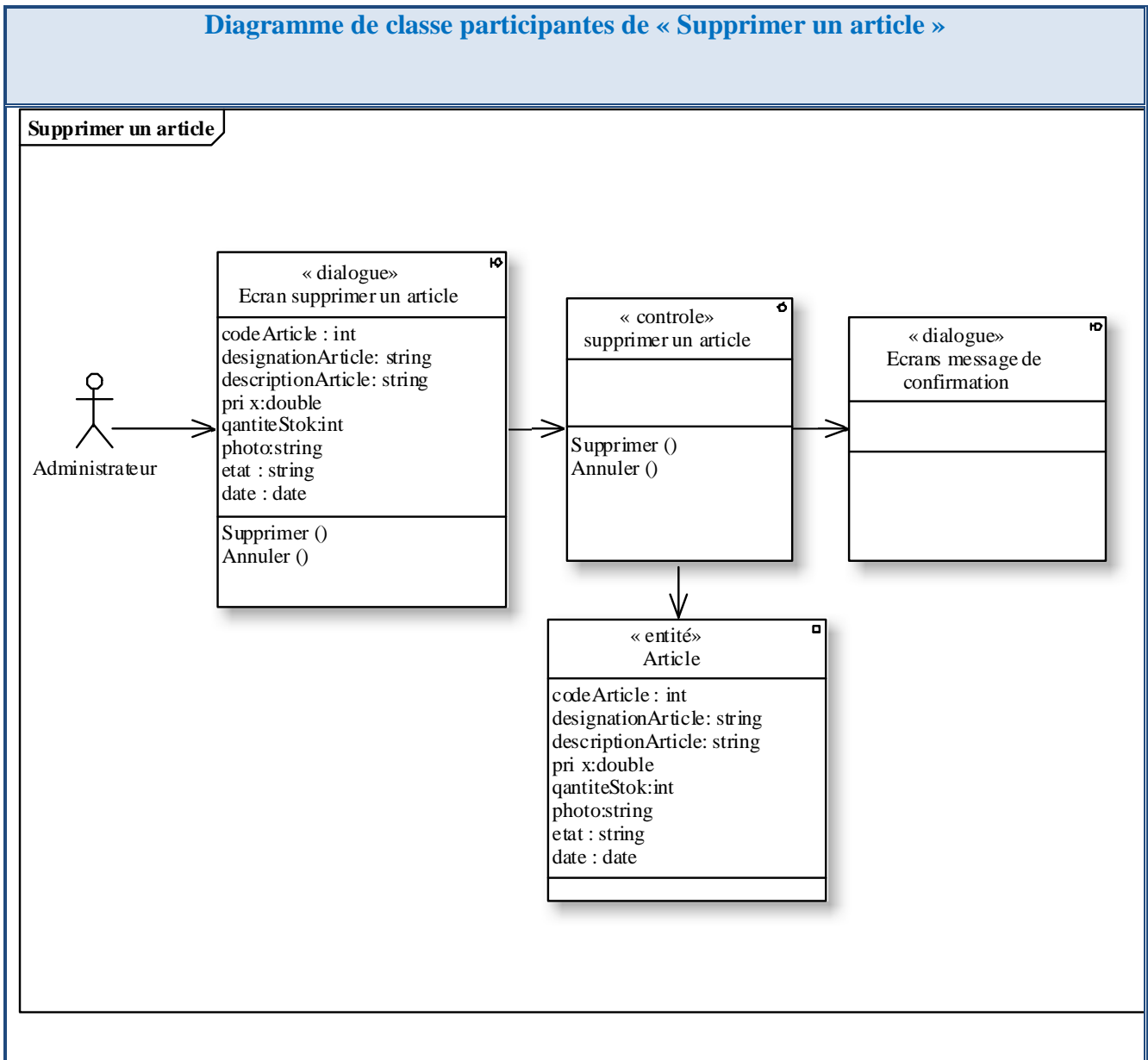
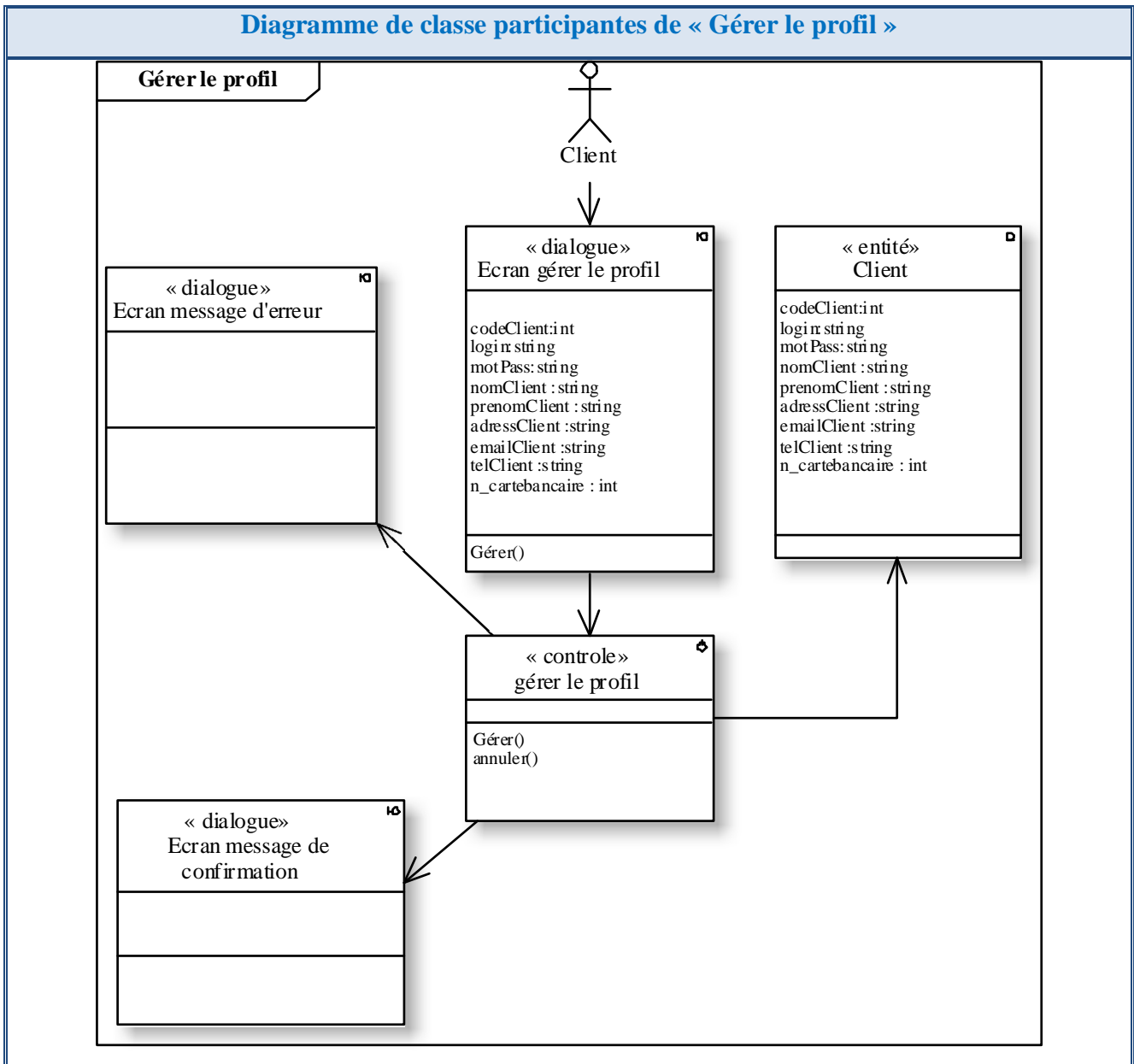


Diagramme de classe participantes de « Gérer le profil »



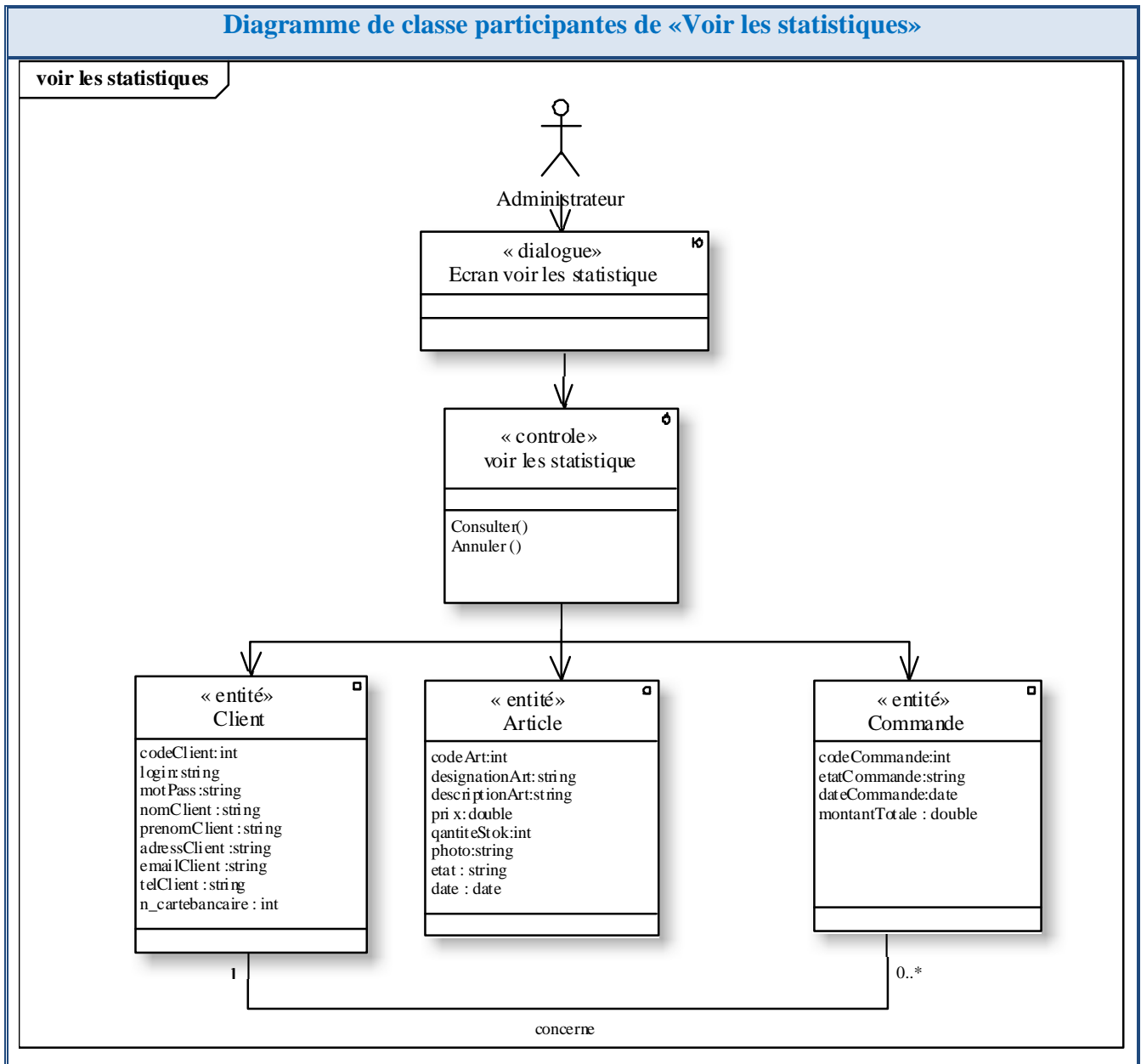


Tableau III -7: Les diagrammes de classes participantes pour tous les cas d'utilisations

3. Capture des besoins techniques

La capture des besoins techniques est la première étape de la branche droite du cycle en Y, c'est une étape de prise en compte des contraintes techniques et logicielles.

Elle se fait selon deux points de vue :

- **Matériel:** sert à choisir une configuration matérielle adéquate selon une architecture matérielle.

- **Logiciel:** sert à exprimer les différentes parties du système par des composants d'exploitation et leurs organisations par des modèles de spécification logicielle.

Pour la réalisation de cette étape nous allons suivre le processus suivant :

- Spécification techniques du point de vue matériel.
- Spécification d'architecture.
- Élaboration du modèle de spécification logicielle.
- Organisation en couches du modèle de spécification.

La démarche de capture des besoins techniques est illustrée dans la figure qui suit :

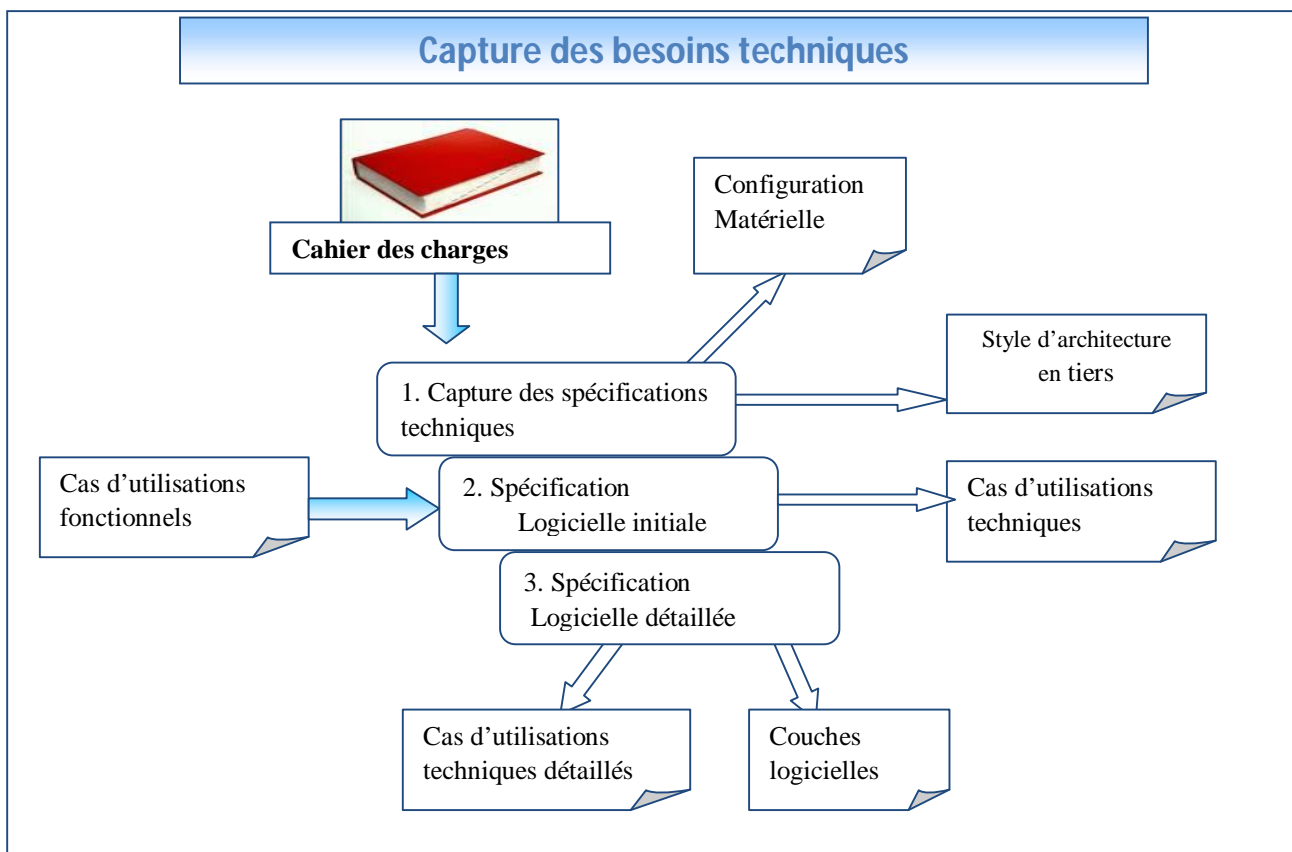


Figure III -4: La démarche de capture des besoins techniques

3.1. Spécification techniques du point de vue matériel :

Les prés requis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels et de celle des choix stratégiques de développement. Ces choix impliquent des contraintes relatives à la configuration du réseau matériel. Elles sont de natures géographiques, organisationnelles, et

techniques. [Roques07]

Le matériel à mettre dans un système dépend de :

- **Style d'architecture en niveaux:** qui spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système. L'architecture peut être à deux niveaux, à trois niveaux, ou multi niveaux dans une organisation plus complexe.

[Roques07]

- **Contraintes techniques:** peuvent se manifester de différentes manières :

- La sécurité.
- La disponibilité
- La performance. [Roques07]

- **La configuration matérielle du SGAE :**

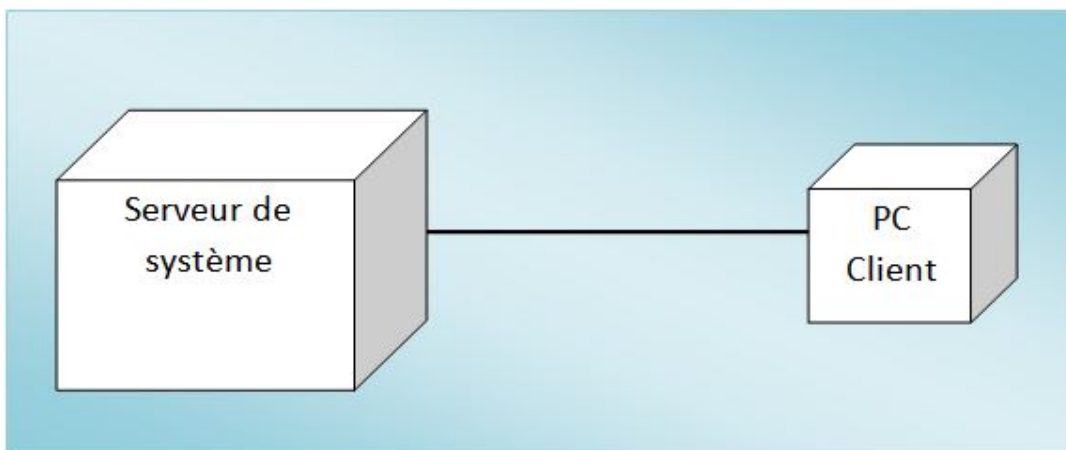


Figure III -5: Configuration matérielle de système SGAE

3.2. Spécification d'architecture :

L'expression des prés requis techniques implique également le choix d'un style d'architecture client/serveur ce choix conditionne la façon dont seront organisés et déployés les composants d'exploitation du système. [Roques07]

- **Le style d'architecture en tiers:** tiers signifie « partie », spécifier l'organisation des composants d'exploitation mis en œuvre pour réaliser le système. Chaque partie indique une responsabilité technique à laquelle souscrivent les différents composants d'exploitation d'un système. [Roques07]
- **Le composant d'exploitation:** un composant d'exploitation est une partie du système logiciel qui doit être connue, installée, déclarée, manipulée par les exploitants du système. Il assure des fonctions bien identifiées dans le système. [Roques07]

L'application réalisée repose sur une architecture client/serveur de 3 tiers. L'architecture 3 tiers est composée de trois niveaux :

1. Le client (navigateur web): le demandeur de ressources.
2. Le serveur d'applications (appelé aussi middleware "niveau intermédiaire"): il est chargé de fournir la ressource mais faisant appel à un autre serveur.
3. Le serveur de données : fournissant les données au serveur d'applications.

Dans notre travail, notre choix porte sur « Mozilla Firefox » comme navigateur Web, un serveur d'applications « Tomcat Apache » et le serveur de base de données « MySQL ».

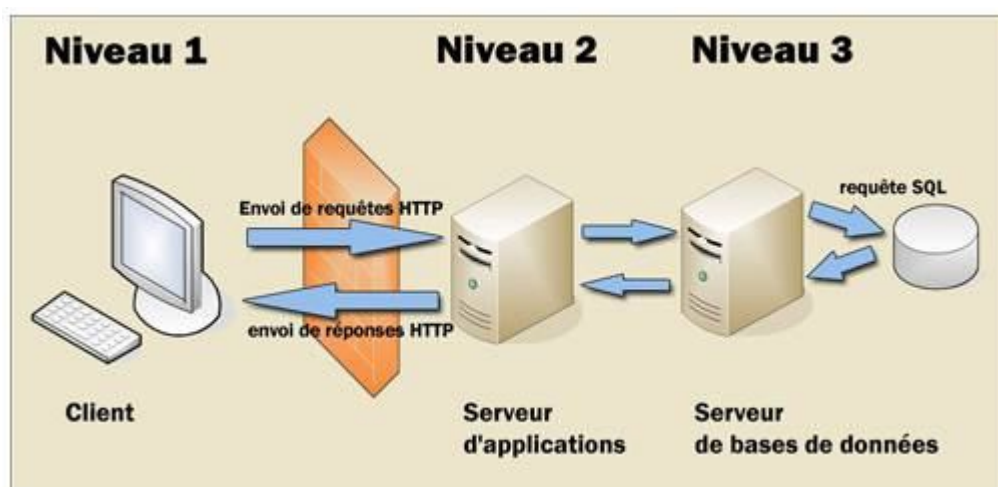


Figure III -6: Architecture 3 tiers

3.2.1. Spécification d'organisation du modèle de déploiement SGAE :

La spécification d'architecture à composants métier 3tiers mène à répartir les composants d'exploitation selon les responsabilités suivantes :

- **Le stockage des données:** correspond à un système de gestion de base de données relationnelle (SGBDR) qui est MySQL.
- **Gestion d'application:** correspond aux différents composants d'application et de service du serveur web, les applications sont développées en JAVA EE.

3.3. Elaboration du modèle de spécification logicielle :

C'est à cette étape que l'on va se référer aux cas d'utilisation techniques. Un cas d'utilisation technique est un cas qui ne produit aucune valeur fonctionnelle (métier) mais fournit des services techniques à l'exploitant, c.à.d. la personne qui utilise le cas. Des exemples de services techniques sont la

connexion au système et l'utilisation de l'aide.

- **L'exploitant:** il s'agit d'un acteur au sens d'UML, qui bénéficie des services techniques du système. L'utilisateur classique d'une application est dans ce sens un exploitant carin bénéficie au moins du service de connexion à l'application.
- **Le cas d'utilisation technique :** un cas d'utilisation technique est destiné à l'exploitant, c'est une séquence d'action produisant une valeur ajoutée opérationnelle ou purement techniques. [Roques07]

Les exploitants du système SGAE sont :

- L'utilisateur: qui utilise une des applications du système SGAE. La majorité des acteurs de la branche fonctionnelle sont donc des utilisateurs dans le domaine technique.
- L'ingénieur d'exploitation: qui est chargé de déployer et de dépanner le système.

Les exploitants du système SGAE sont :

- Gérer l'intégrité.
- Gérer les erreurs.
- Gérer la sécurité.

3.3.1. Modèle de spécification logicielle :

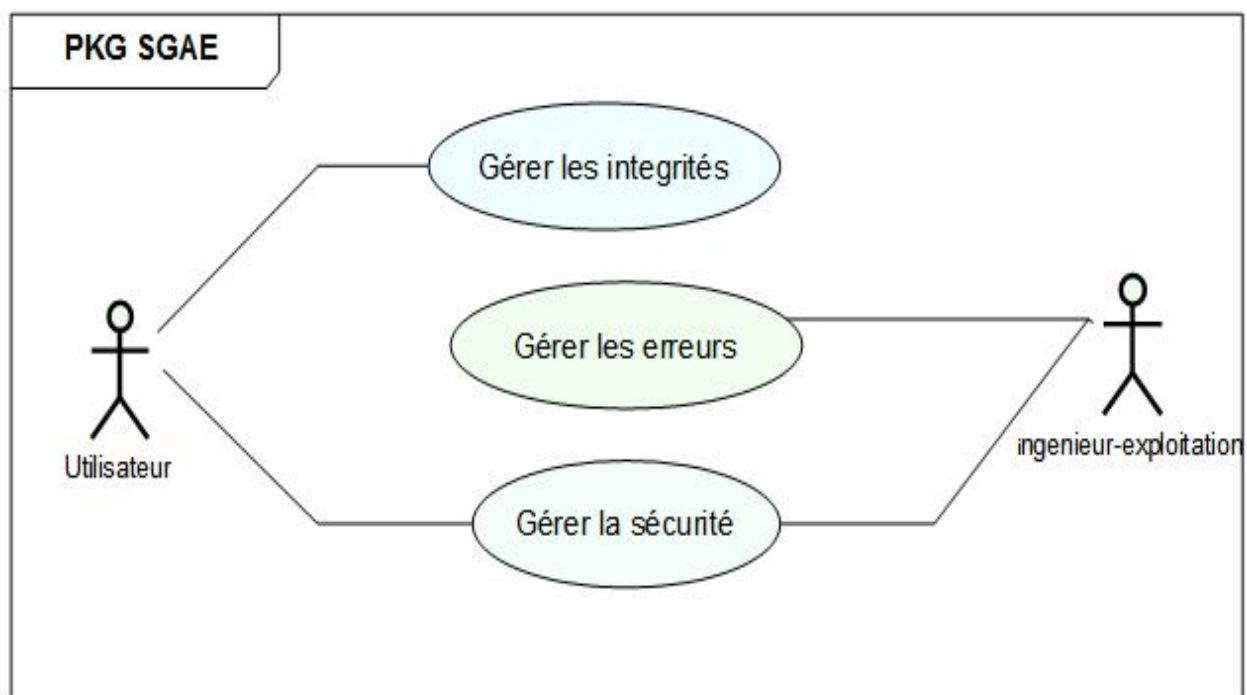


Figure III -7: Modèle de spécification logicielle initial

3.3.2. Description textuelle des cas d'utilisation technique :

Cas d'utilisation 1 : Gérer l'intégrité	
Intention	C'est le mécanisme qui empêche la mise à jour simultanée d'une même entité par plusieurs utilisateurs.
Action	Empêche la mise à jour simultanée.

Cas d'utilisation 2 : Gérer les erreurs	
Intention	Gérer les erreurs de frappe et de manipulation.
Action	Spécifie l'erreur, la traiter, et le déboguer.

Cas d'utilisation 3 : Gérer la sécurité	
Intention	L'utilisateur doit se connecter, s'authentifier et être reconnu par le système à travers le mot de passe.
Action	Permission de faire des opérations.

Tableau III -8: Description textuelle des cas d'utilisation techniques

3.3.3. Organisation en couche du modèle de spécification :

Le modèle de spécification logicielle est cependant trop sommaire pour permettre une spécification technique détaillé, Dans ce contexte, il est difficile de pouvoir préciser de manière détaillé les comportements techniques attendus, si l'on n'organise pas la spécification suivants les différentes responsabilités de traitement. [Roques07]

- **Couche logicielle :**

Une couche logicielle représente un ensemble de spécification ou de réalisation qui respectivement expriment ou mettent en œuvre un ensemble de responsabilité techniques et homogènes pour un système logiciel. [Roques07]

Les couches s'empilent en niveaux pour couvrir des transformations logicielles successives, de

sorte que la couche d'un niveau ne puisse utiliser que les services des couches des niveaux inférieurs.
[Roques07]

- ✓ **Présentation** : Restitue les données à l'utilisateur, et transforme ses actions en événement de l'application.
- ✓ **Application** : représente les objets de contrôle et pilote les règles de l'application.
- ✓ **Métier** : Représente les objets du métier et implémente leurs règles de gestion.
- ✓ **Accès aux données** : Restitue les représentations métier à partir du moyen de stockage.
- ✓ **Stockage des données** : Assure la persistance des données. [Roques07]

Dans le modèle UML, les couches logicielles correspondent à des packages.

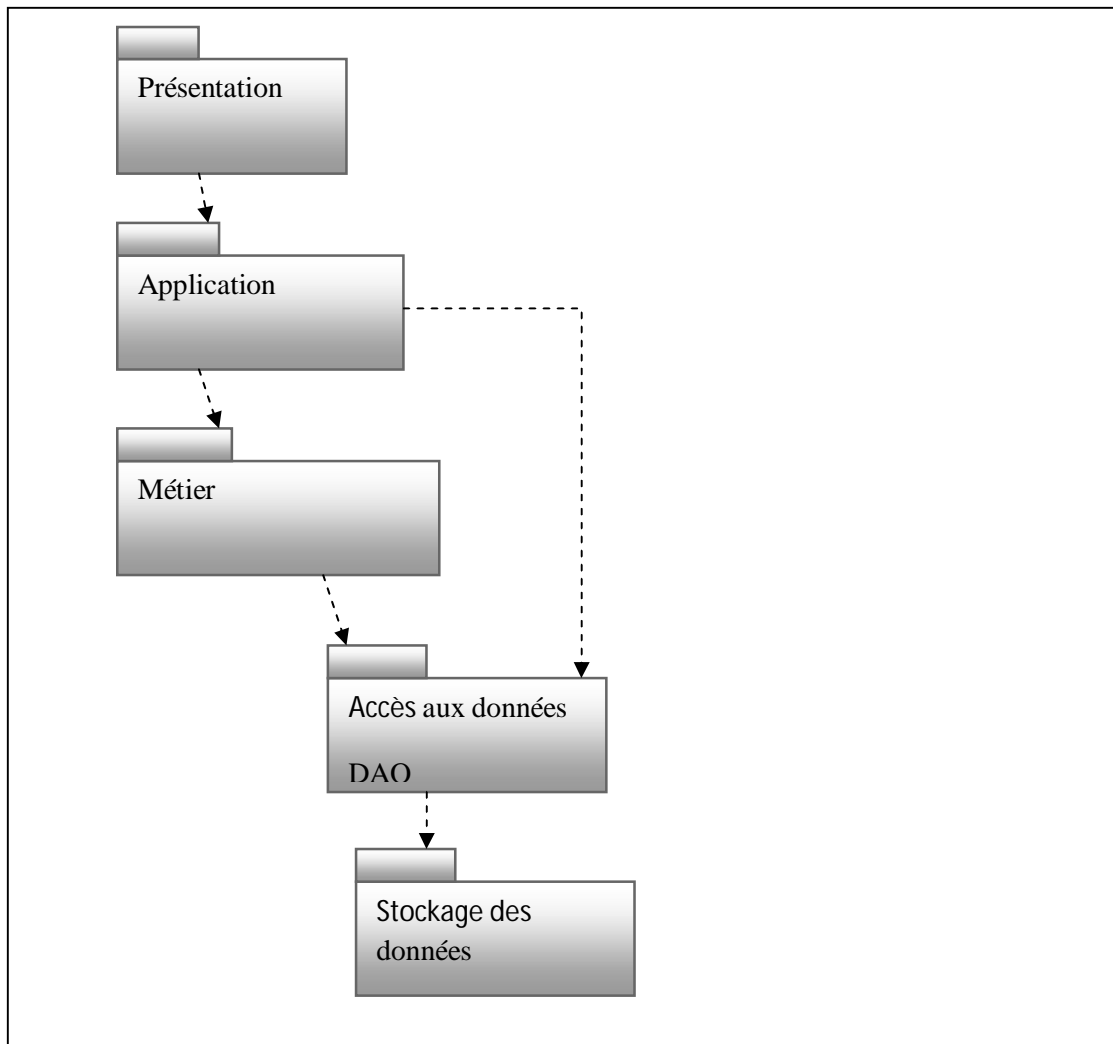


Figure III -8: Organisation du modèle de spécification logicielle

4. Conclusion

La capture des besoins fonctionnels joue un rôle essentiel dans le cadre d'une part, de compléter les recueils initiales des besoins opérés pendant l'étude préliminaire, et de l'autre part, elle donne une première vue pour le prochain chapitre concernant l'analyse afin d'identifier les classes du modèle statique qui présentent une des approches orientées objets.

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description de métiers des utilisateurs, ni de la description applicative.

Chapitre IV



Analyse

1. Introduction

L'étape d'analyse est entamée à la suite de l'étape de capture des besoins fonctionnels, Elle est située sur la branche gauche du processus 2TUP.

Dans ce chapitre là on va traiter la phase de l'analyse objet du système SGAE, en suivant les étapes suivantes :

- Le découpage en catégorie
- Le développement du modèle statique,
- Le développement du modèle dynamique.

La figure qui suit illustre la situation de l'étape d'analyse dans le processus 2TUP :

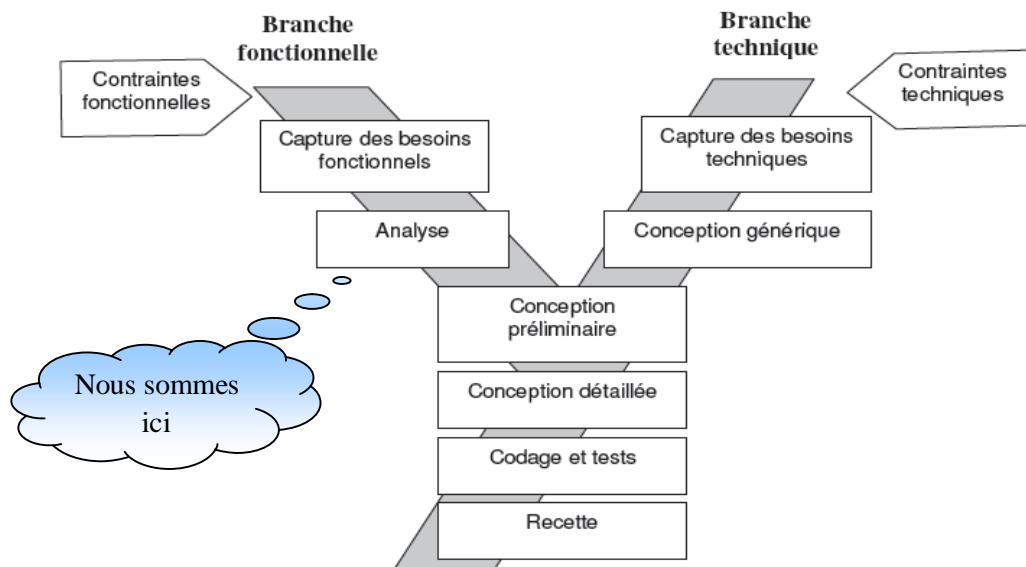


Figure IV-1: L'étape d'analyse

2. Découpage en catégories

Le découpage en catégories constitue la première activité de l'étape d'analyse, Cette phase utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

Une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe. [Roques07]

Démarche d'élaboration du modèle structurel d'analyse :

La démarche mise en œuvre dans ce chapitre suit les étapes suivantes :

- Répartir les classes candidates en catégories.
- Elaborer les diagrammes de classe préliminaires par catégorie.
- Décider des dépendances entre catégories.

2.1. Répartition des classes candidates en catégories :

Le découpage en catégories de notre projet a donné le résultat suivant :

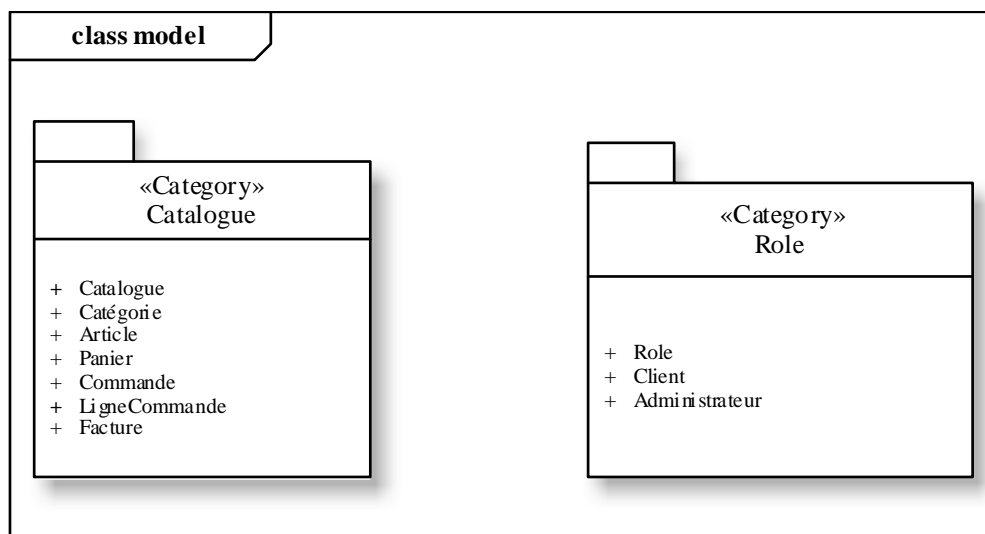


Figure IV-2: Découpage en catégories

2.2. Elaboration de diagramme de classe préliminaires pour chaque catégorie:

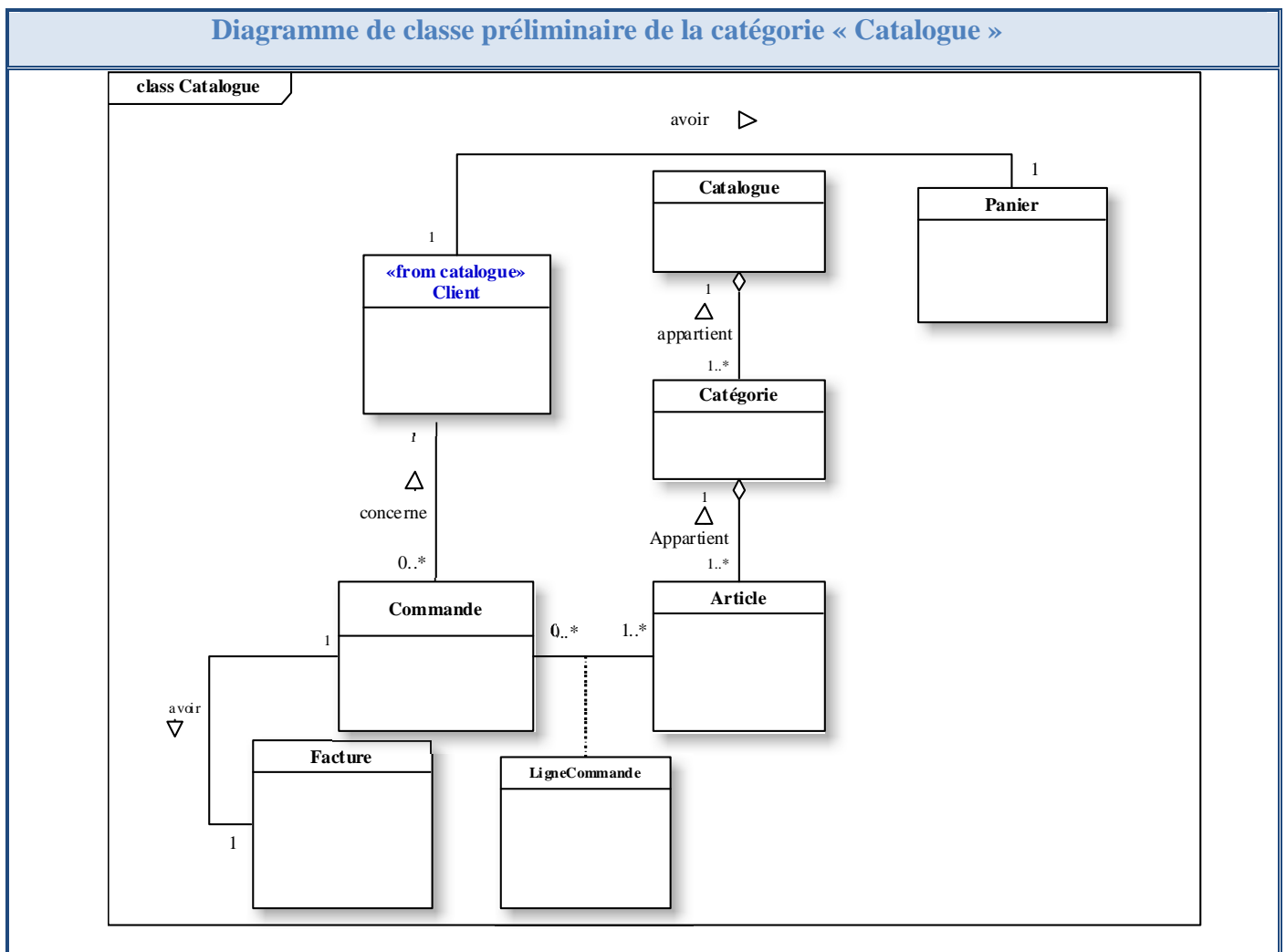
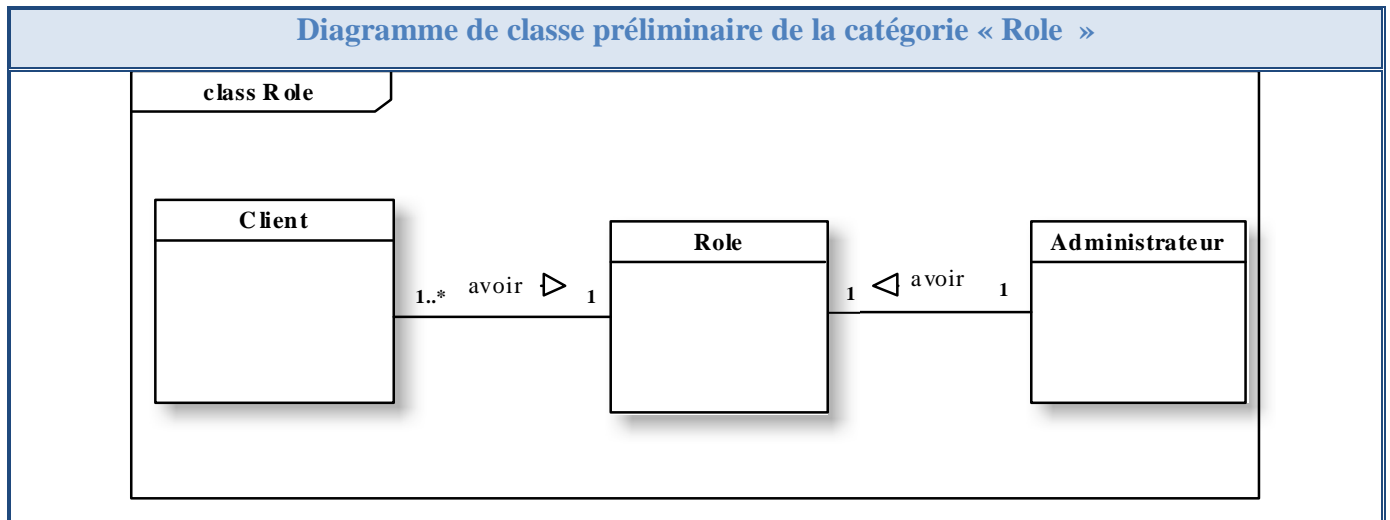


Tableau IV -1: Les diagrammes de classe préliminaire des catégories

2.3. Dépendance entre les catégories :

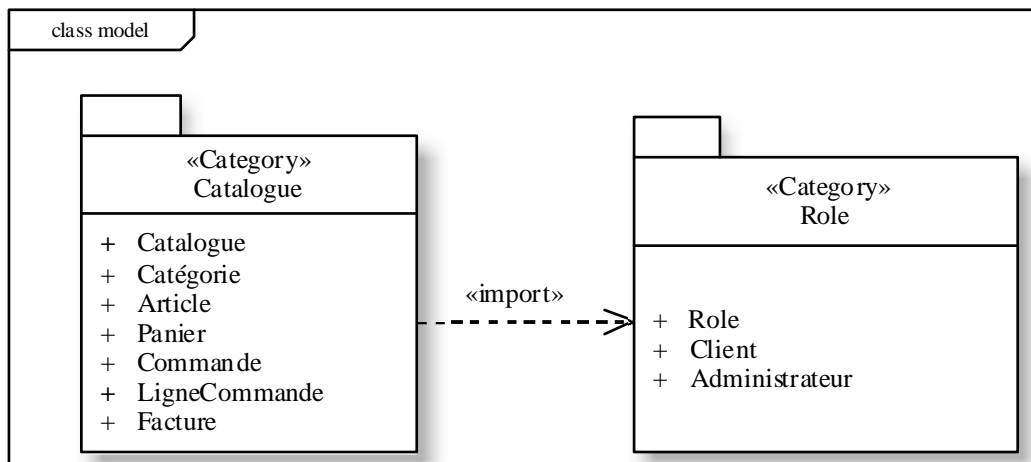


Figure IV -3: Modèle structurel d'analyse

3. Développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Les diagrammes de classes établis sommairement dans les diagrammes de classes participantes, puis réorganisés lors du découpage en catégories, vont être détaillés, complétés, et optimisés. [Roques07]

La démarche de développement du modèle statique est synthétisée par la figure suivante.

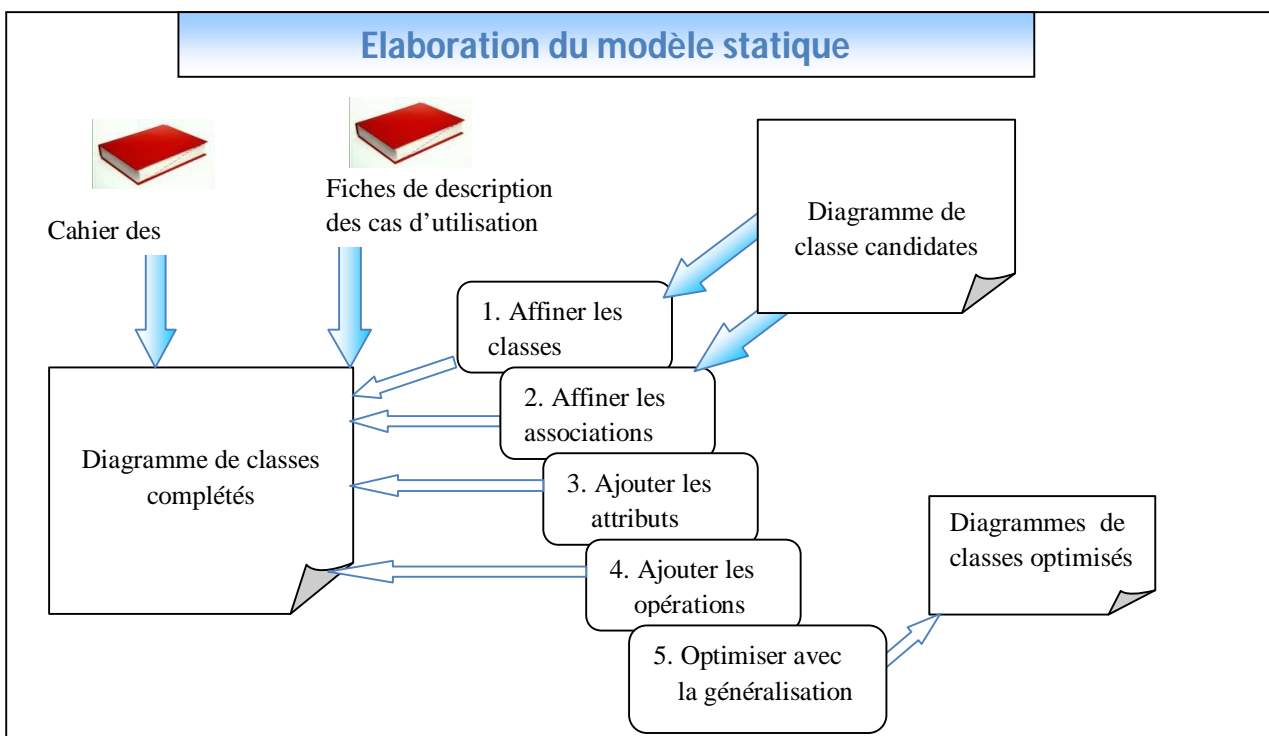


Figure IV -4: La démarche d'élaboration du modèle statique

3.1. Affiner les classes :

Les classes identifiées lors de l'étude des cas d'utilisations sont simplement des classes candidates pour l'analyse objet, il convient désormais de les examiner de manière détaillée, d'en éliminer certaines, ou au contraire d'en ajouter d'autre. [Roques07]

3.2. Affiner les associations :

Les associations représentent des relations conceptuelles entre les classes. On peut également dire qu'elles impliquent des responsabilités en termes de navigation, la navigation dans un modèle statique représente la capacité à obtenir des informations en parcourant les associations entre les classes. On peut donc considérer les associations comme porteuses d'une partie fondamentale de la structure statiques des classes.

Affiner les associations convient de les valider, les préciser, en éliminer, et en ajouter. Il s'agit d'une activité, consiste également à utiliser deux notions complémentaires fournies par UML, l'agrégation et la composition, qui sera complétée grâce à l'identification des attributs. [Roques07]

3.3. Ajouter les attributs :

Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeurs possible partagé par tous les objets de la classe. Dans un modèle d'analyse, nous conservons uniquement comme attributs les propriétés simples des classes que le système doit mémoriser et utiliser. [Roques07]

3.4. Ajouter les opérations :

Une opération représente un service, un traitement qui peut être demandé à n'importe quel objet de la classe. Une opération est l'abstraction de ce qu'on peut réaliser sur un objet, et elle est partagée par tous les objets de la classe. [Roques07]

3.5. Diagramme de classe détaillé pour chaque catégorie:

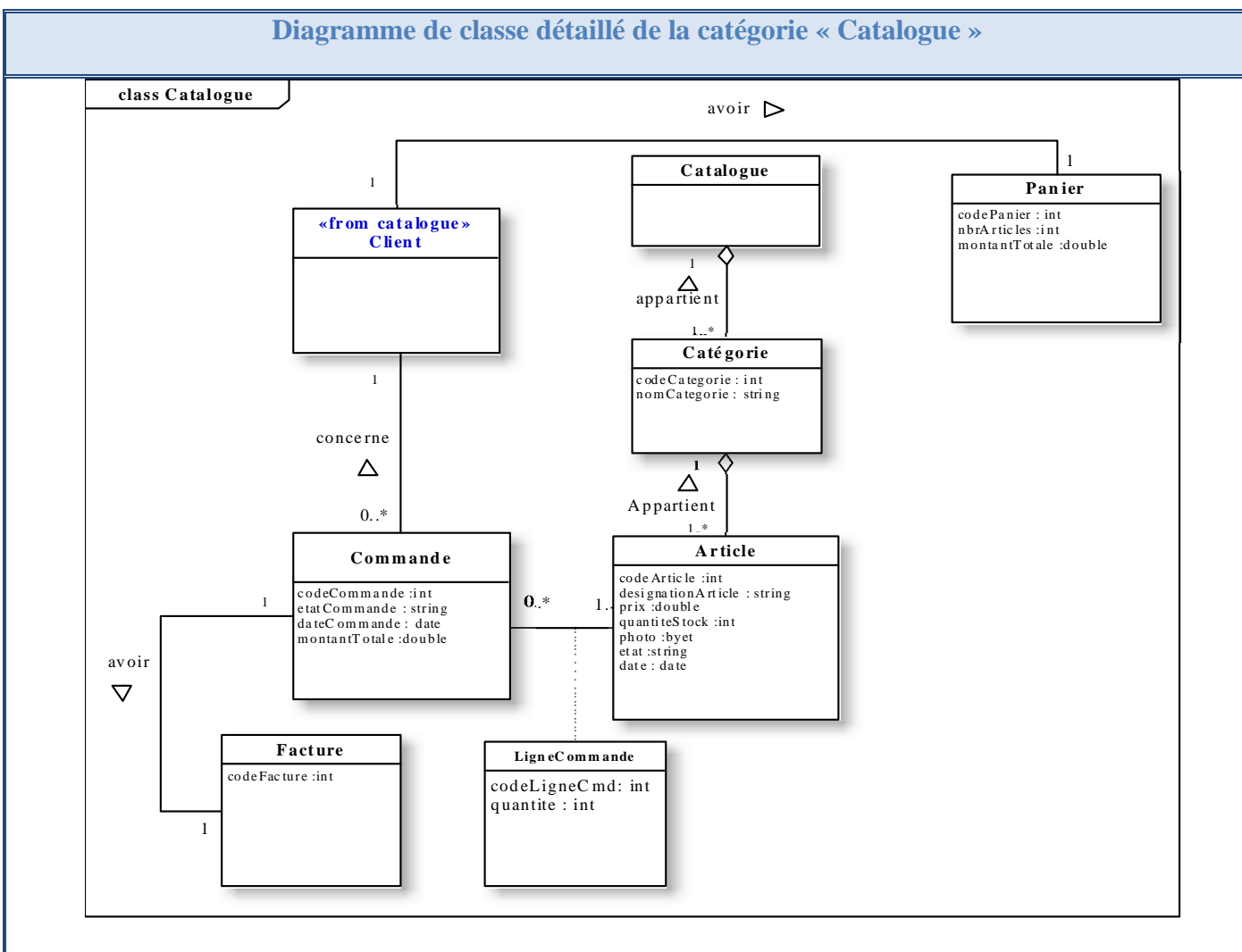
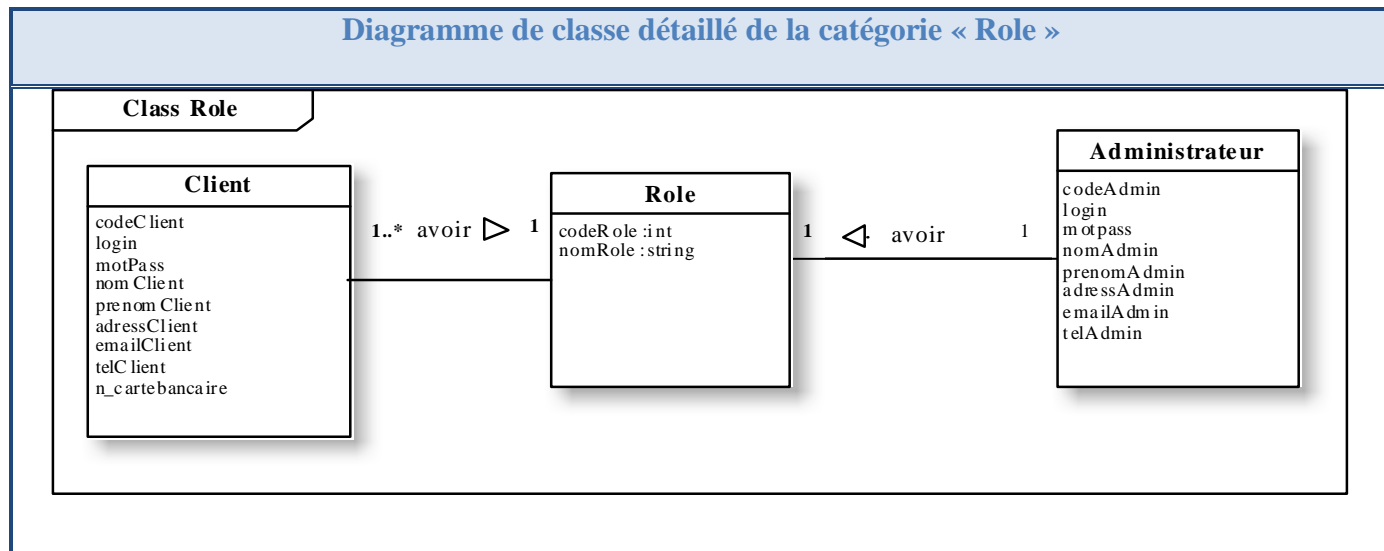


Tableau IV -2: Les diagrammes de classe détaillé des catégories

4. Développement de modèle dynamique

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative, fortement couplée avec l'activité de la modélisation statique. [Roques07]

La figure suivante illustre la démarche d'élaboration du modèle dynamique.

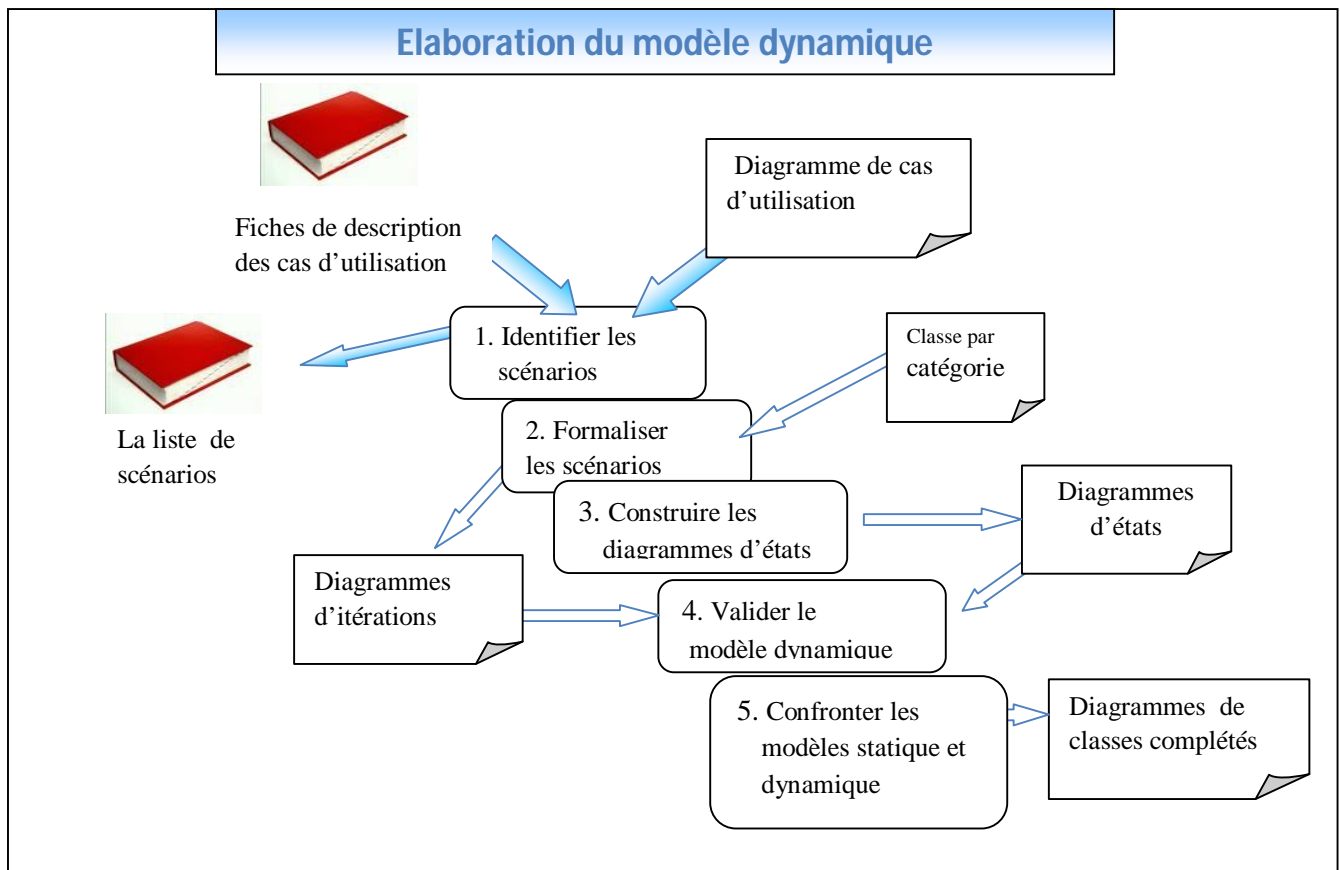
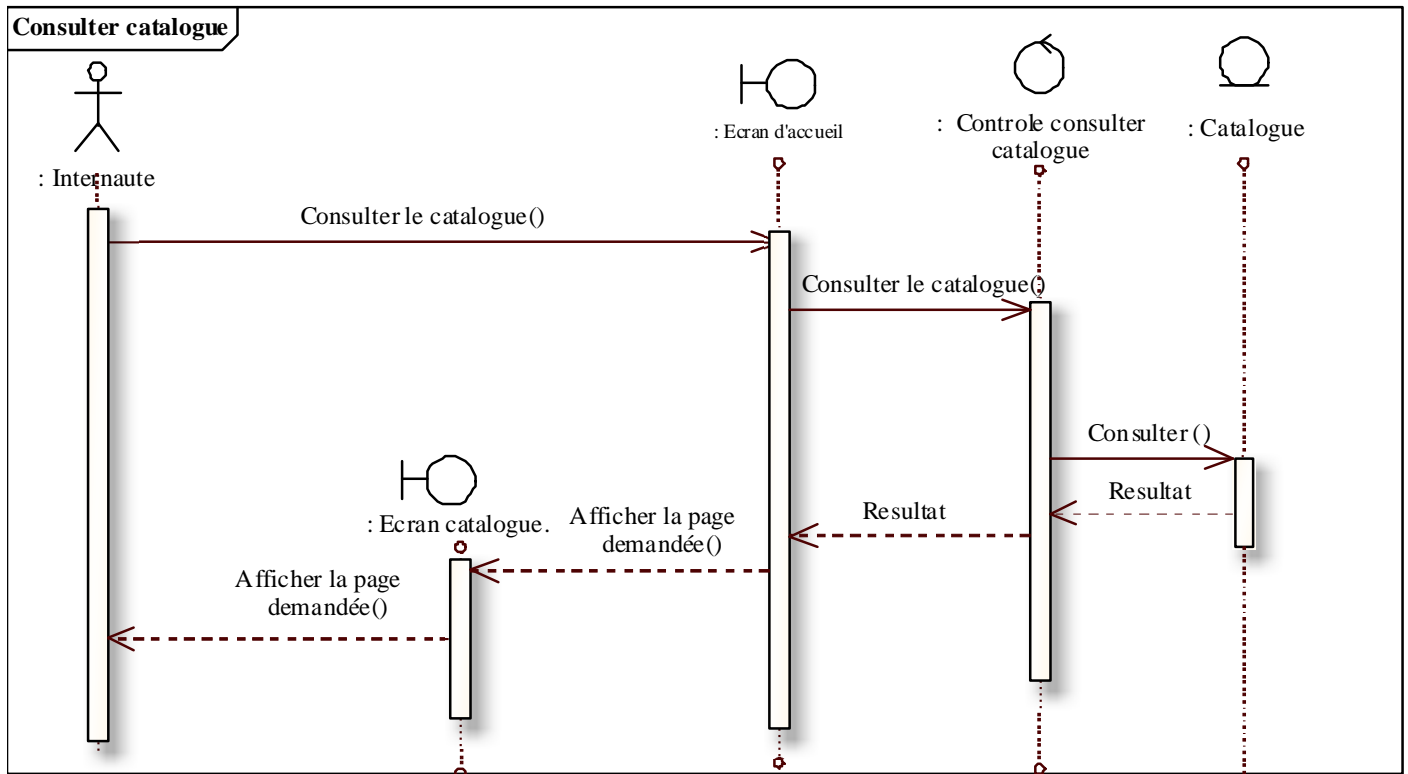


Figure IV -5: La démarche d'élaboration du modèle dynamique

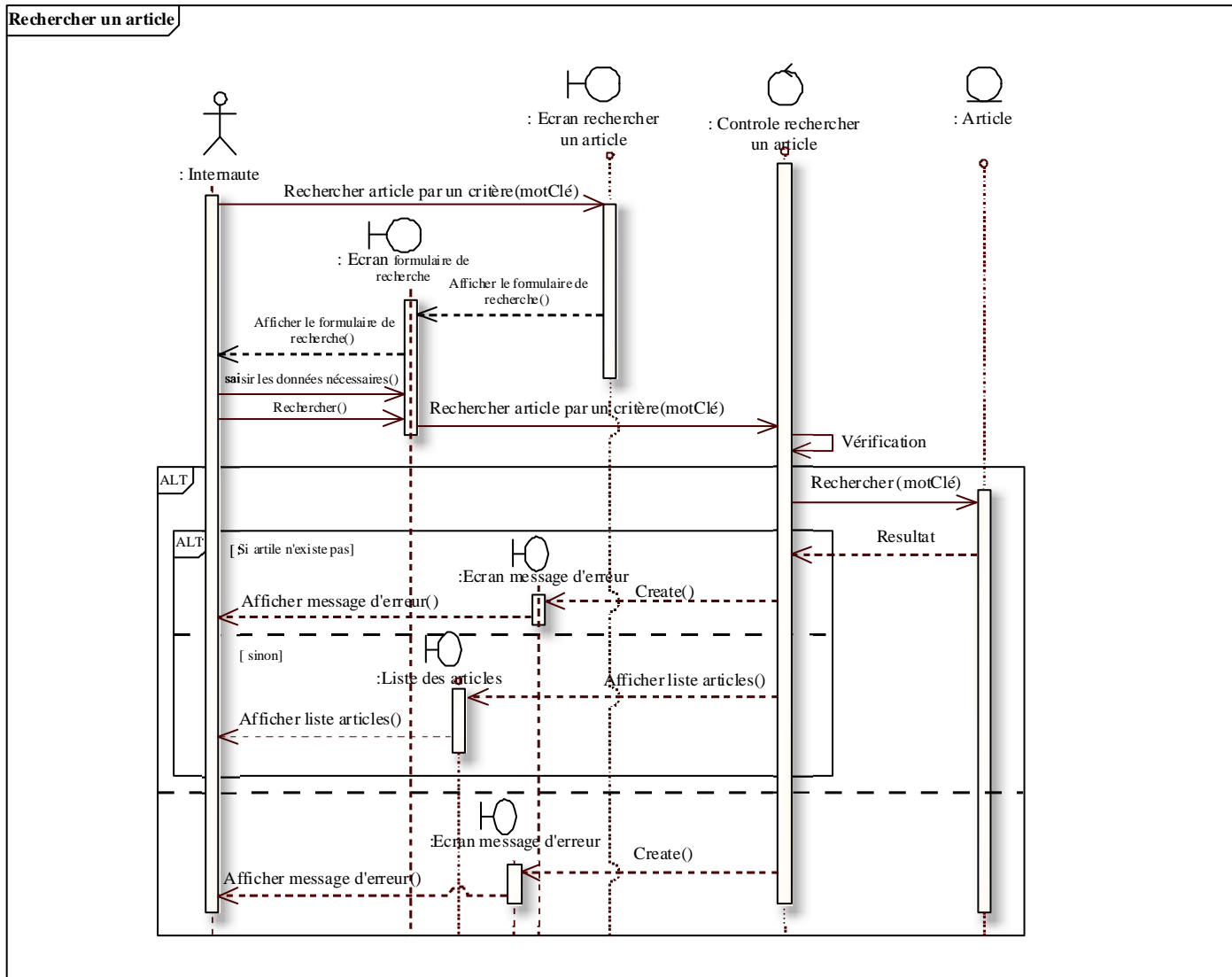
4.1. Diagramme de séquence :

Les diagrammes de séquences permettent de représenter des collaborations entre objet selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. un diagramme de séquence qui est plus apte à représenter un scénario dans le contexte d'un cas d'utilisation. Pour élaborer le modèle dynamique de notre système on va utiliser le diagramme séquence.

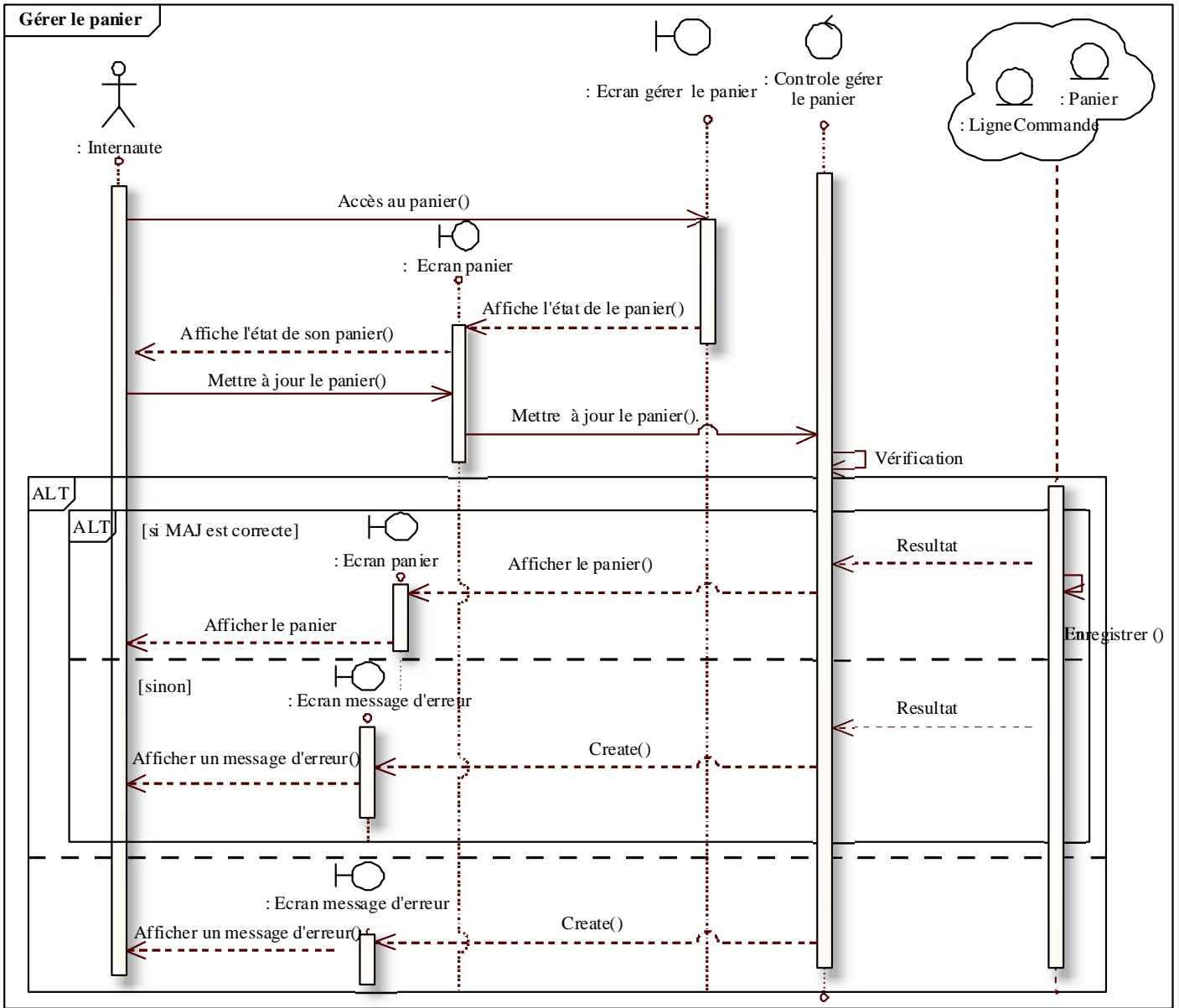
➤ Cas d'utilisation : Consulter le catalogue



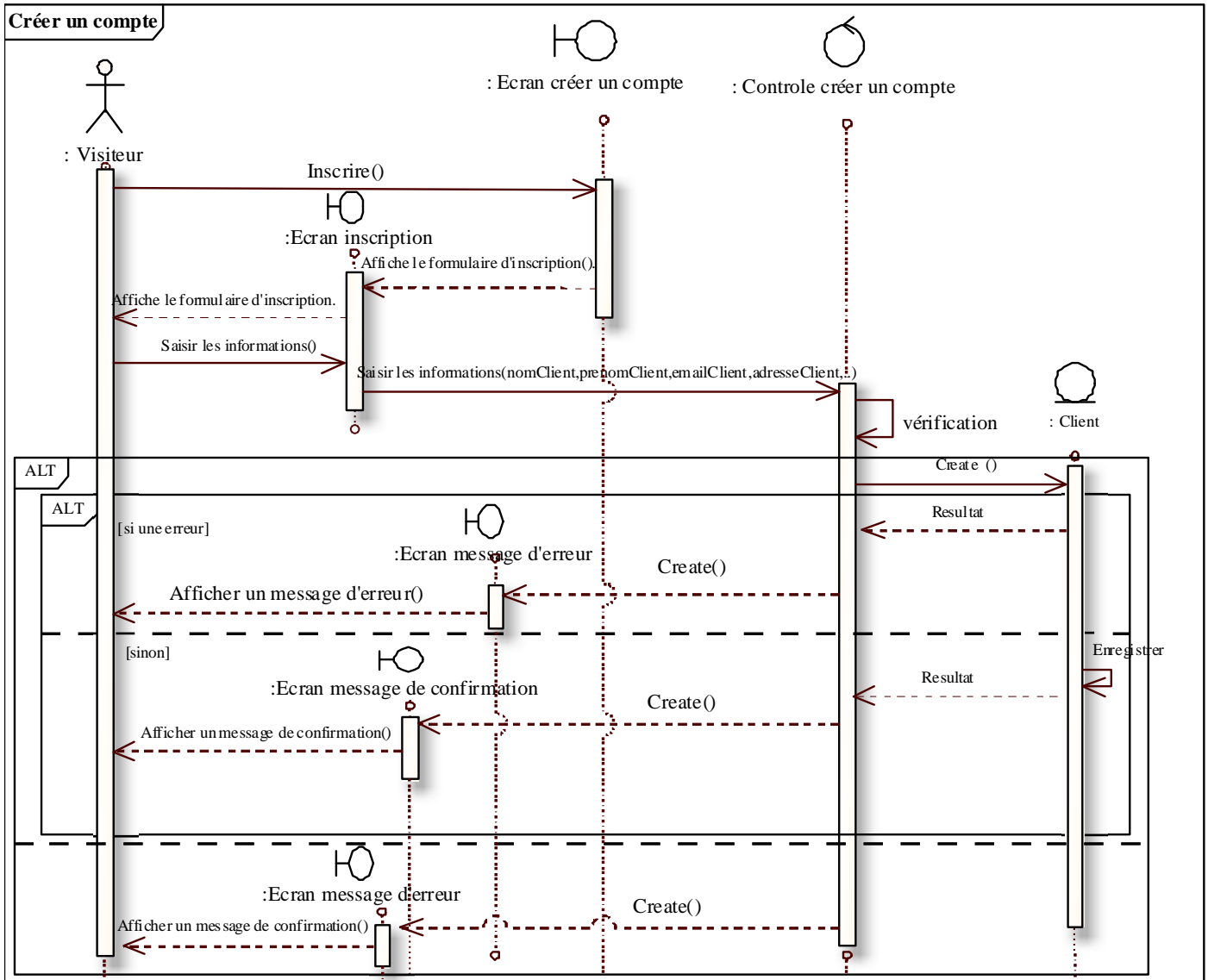
➤ Cas d'utilisation : Rechercher un article



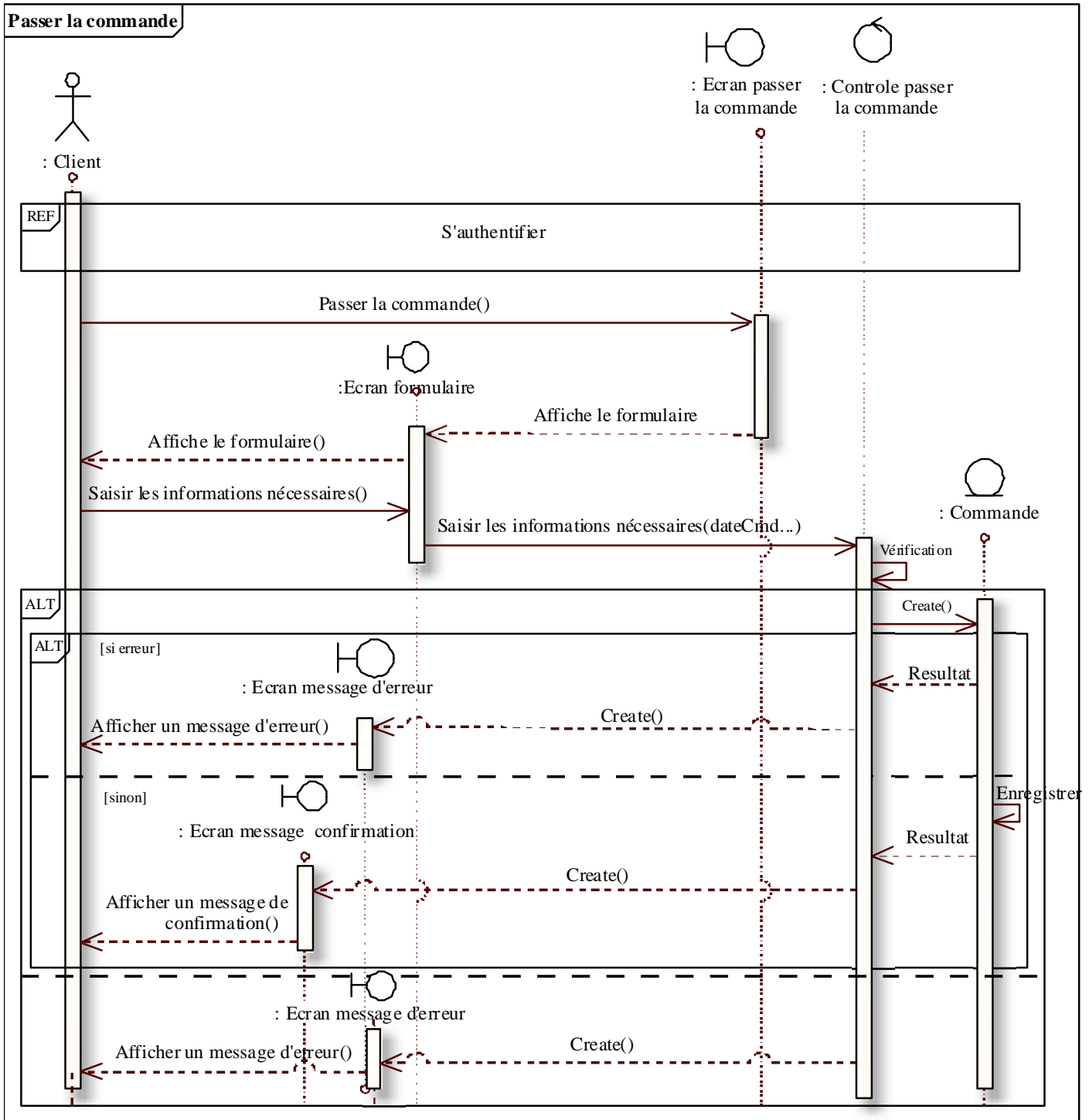
➤ Cas d'utilisation : Gérer le panier



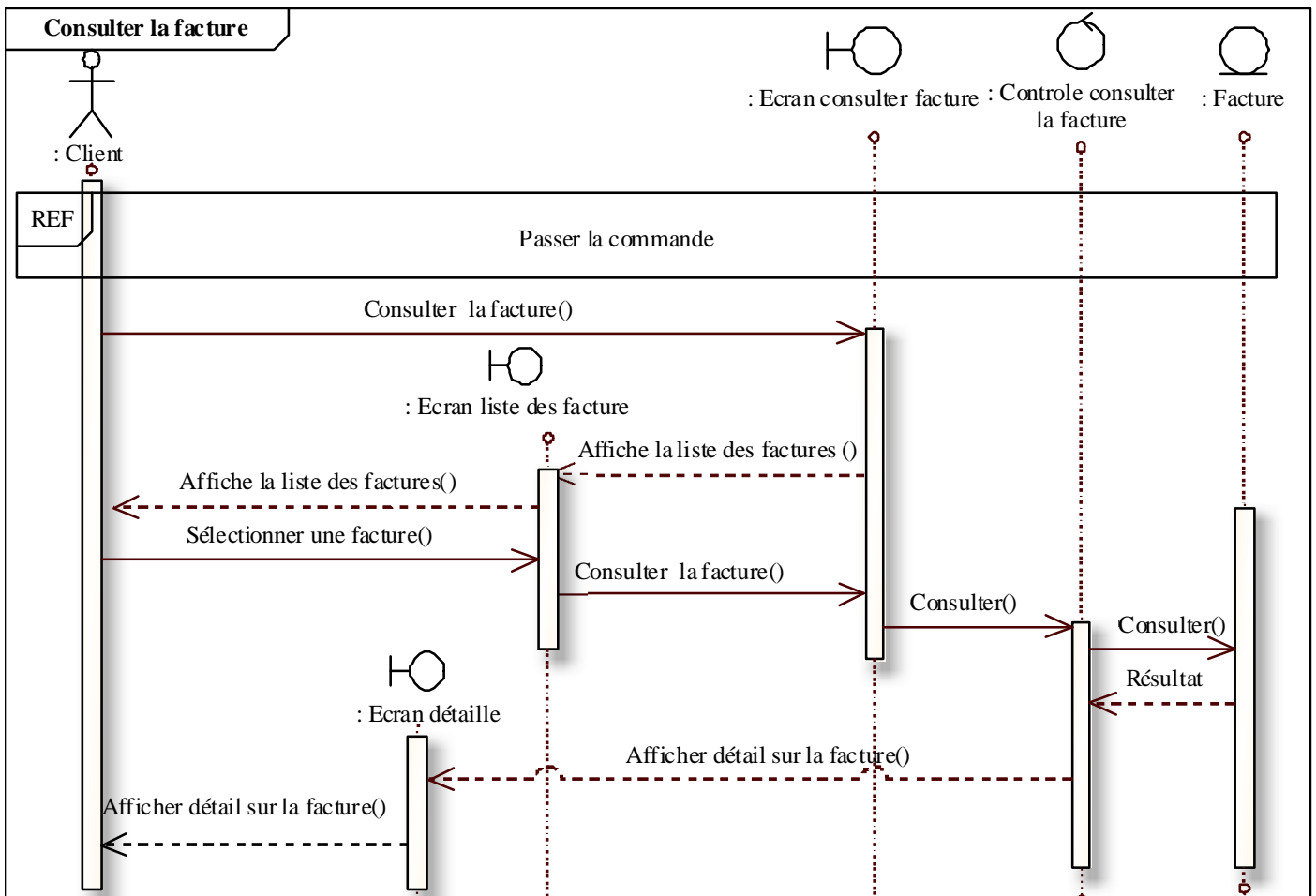
➤ Cas d'utilisation : Créer un compte



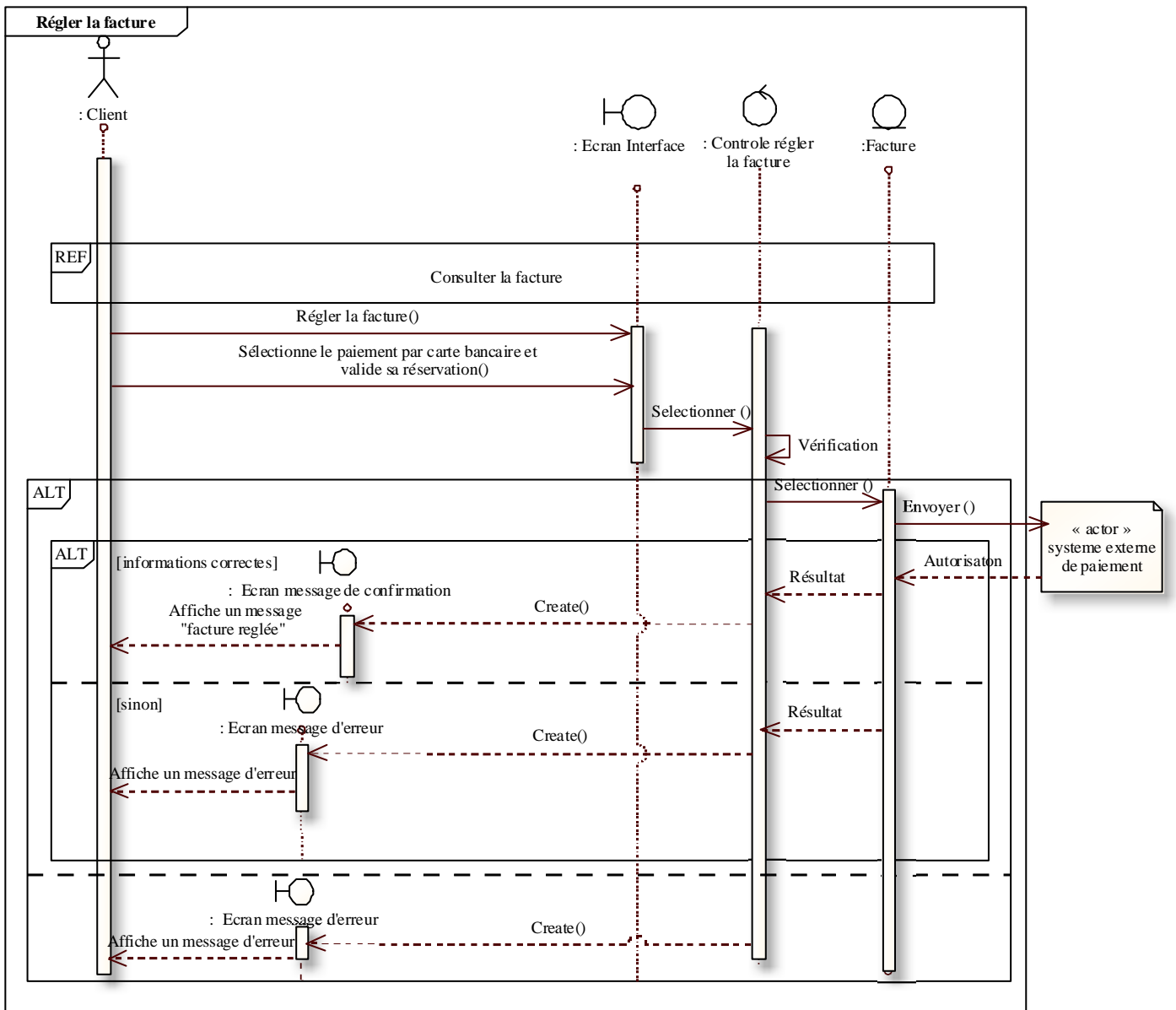
➤ Cas d'utilisation : Passer la commande



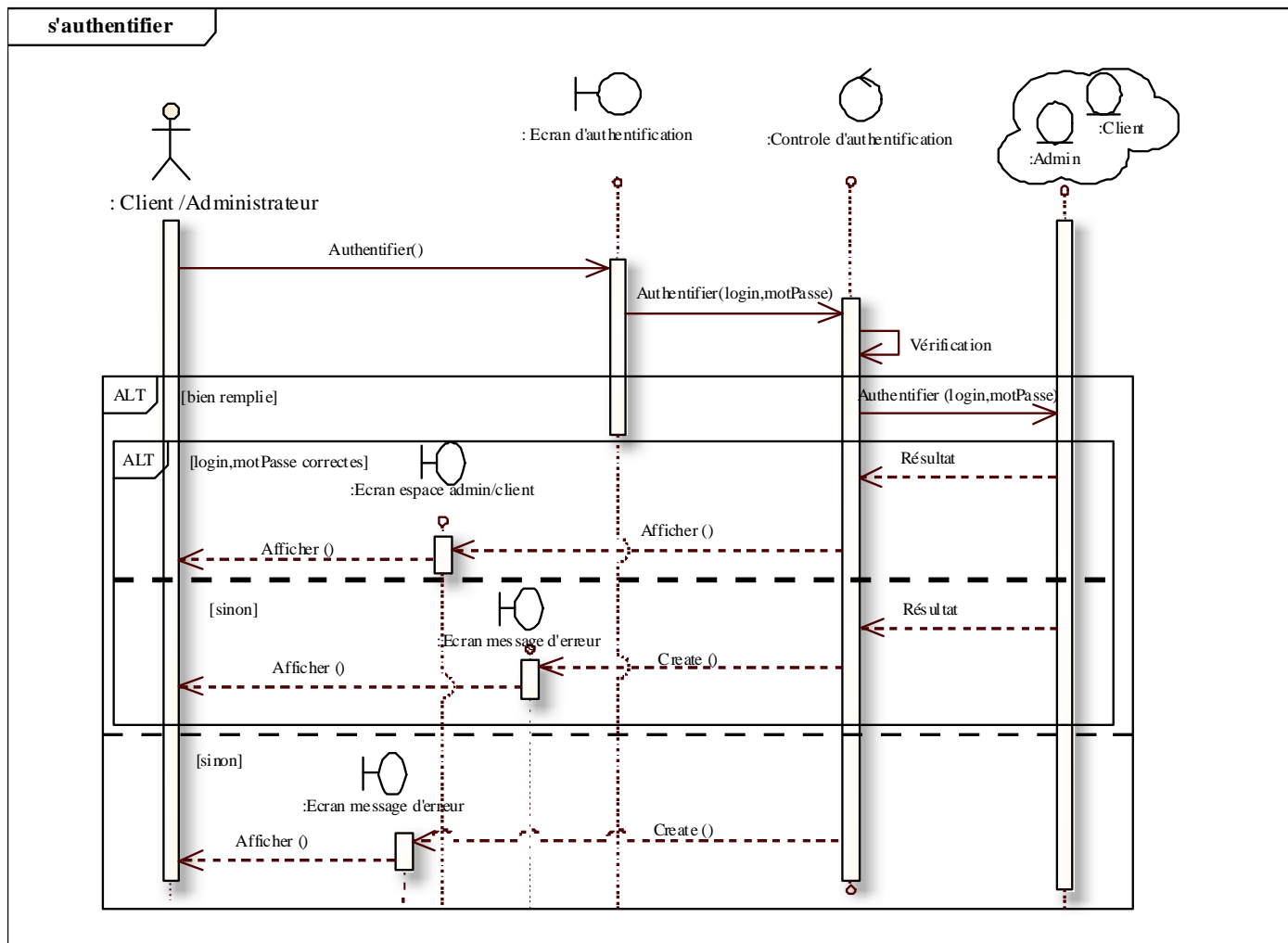
➤ Cas d'utilisation: Consulter la facture.



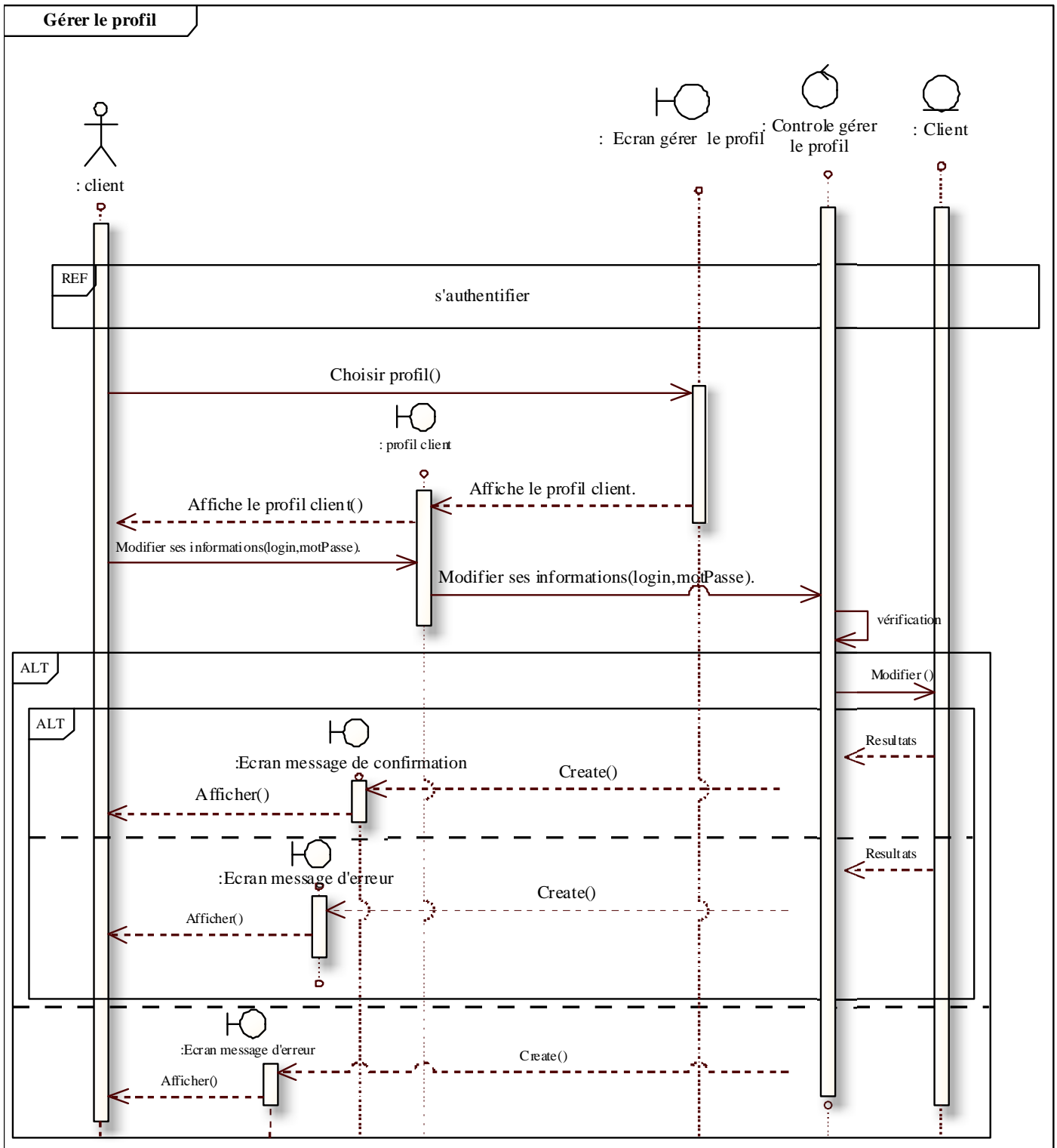
➤ Cas d'utilisation: régler la facture.



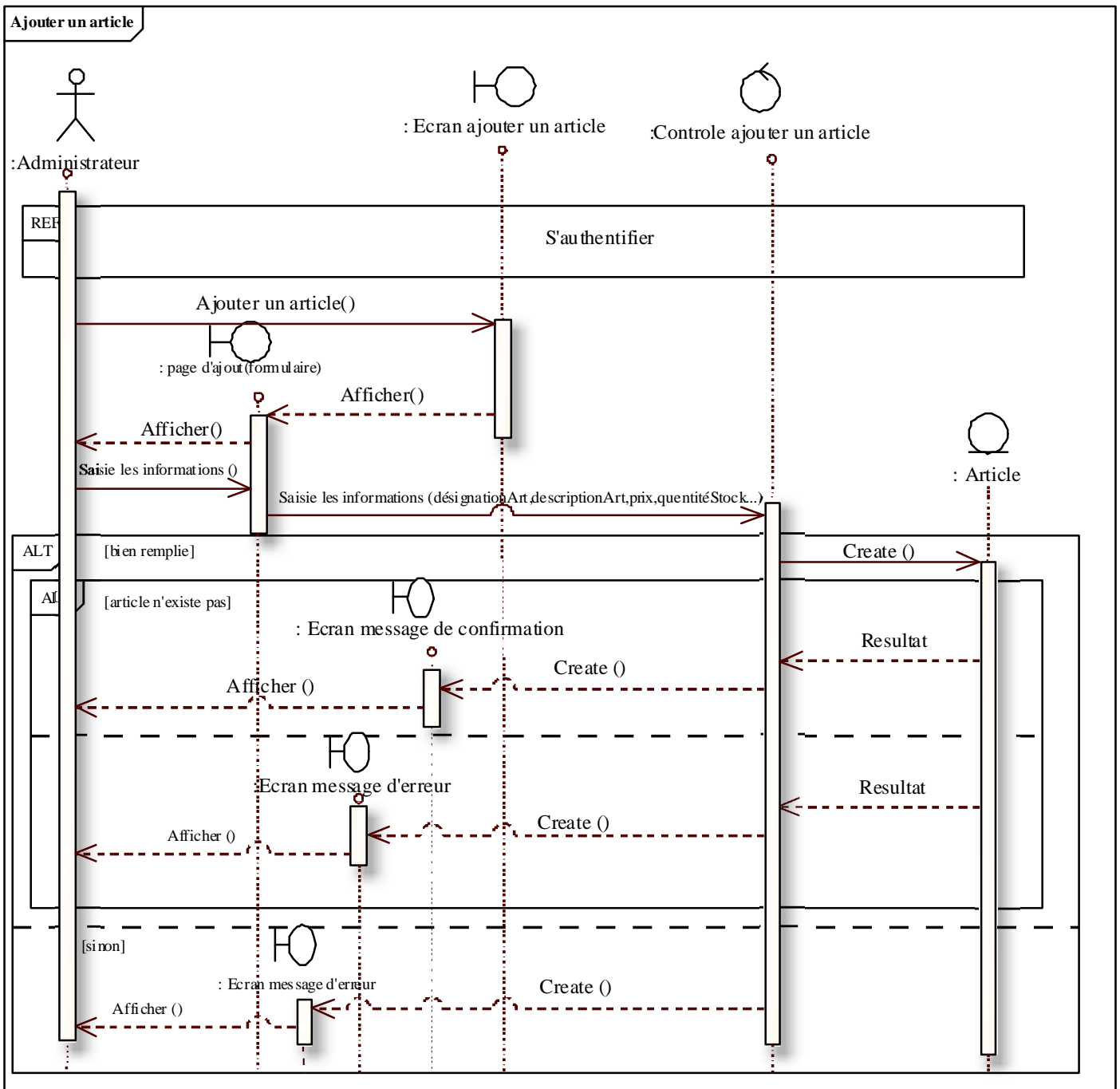
➤ Cas d'utilisation: S'authentifier



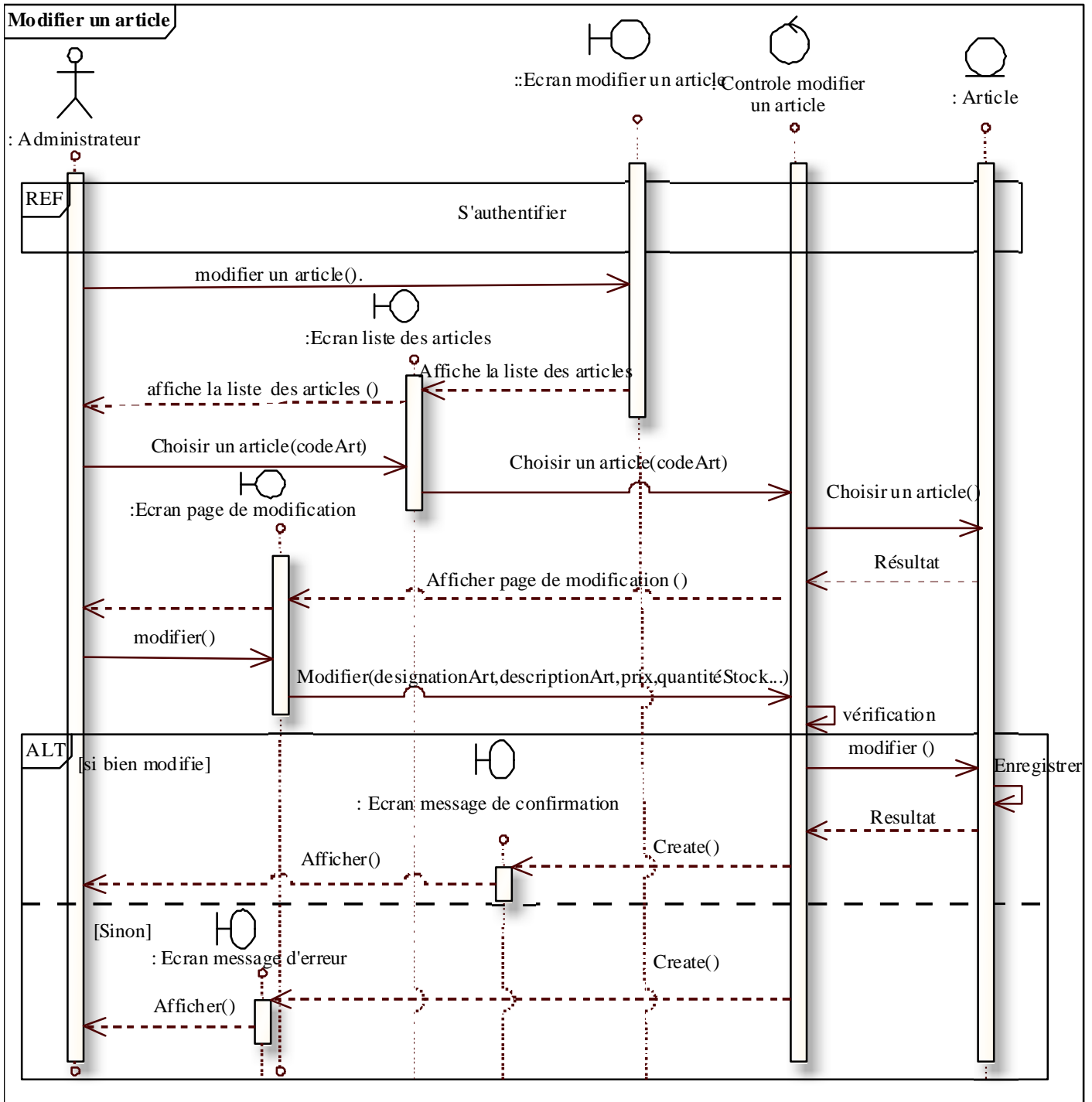
➤ Cas d'utilisation : Gérer le profil



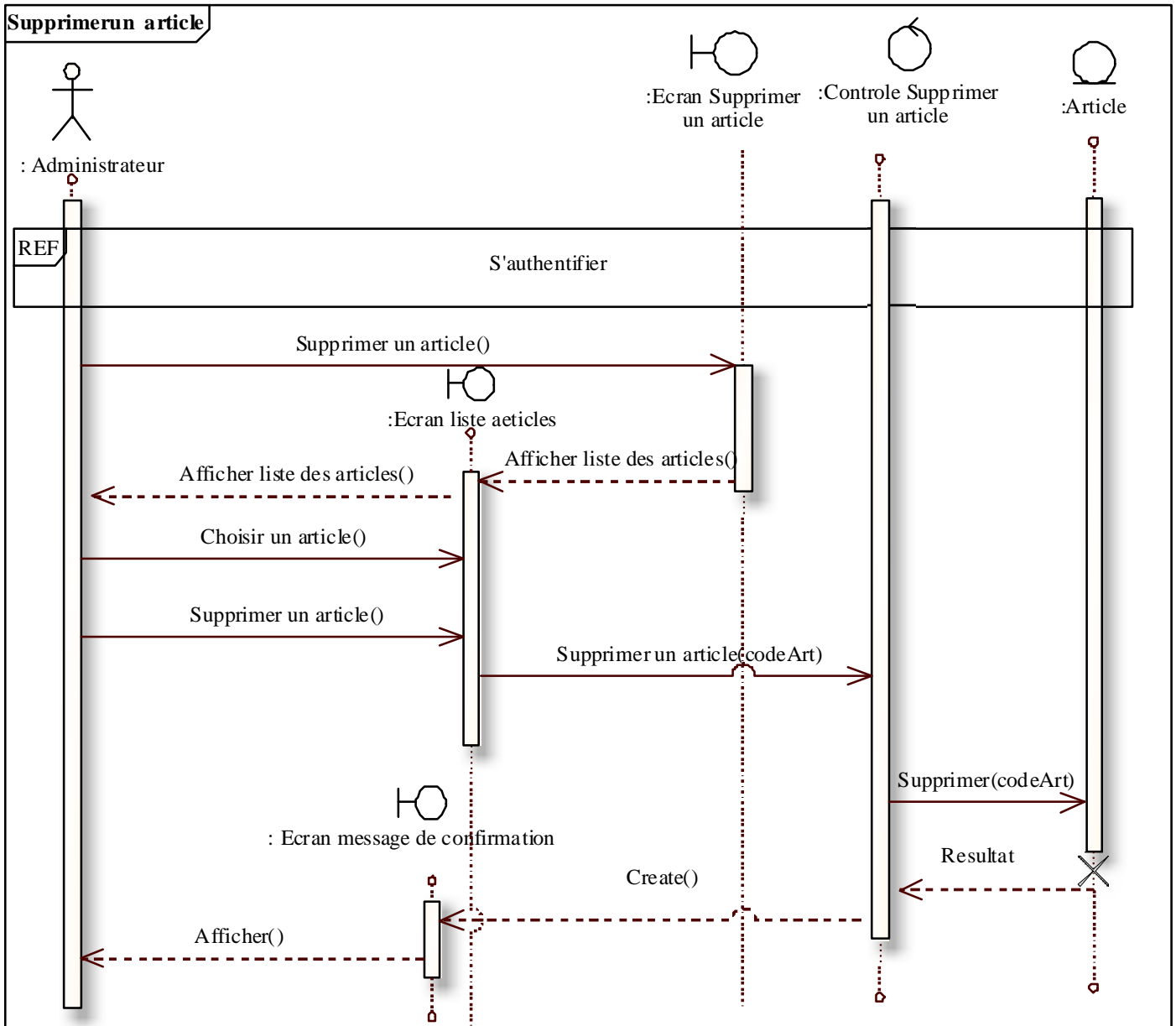
➤ Cas d'utilisation: Ajouter un article



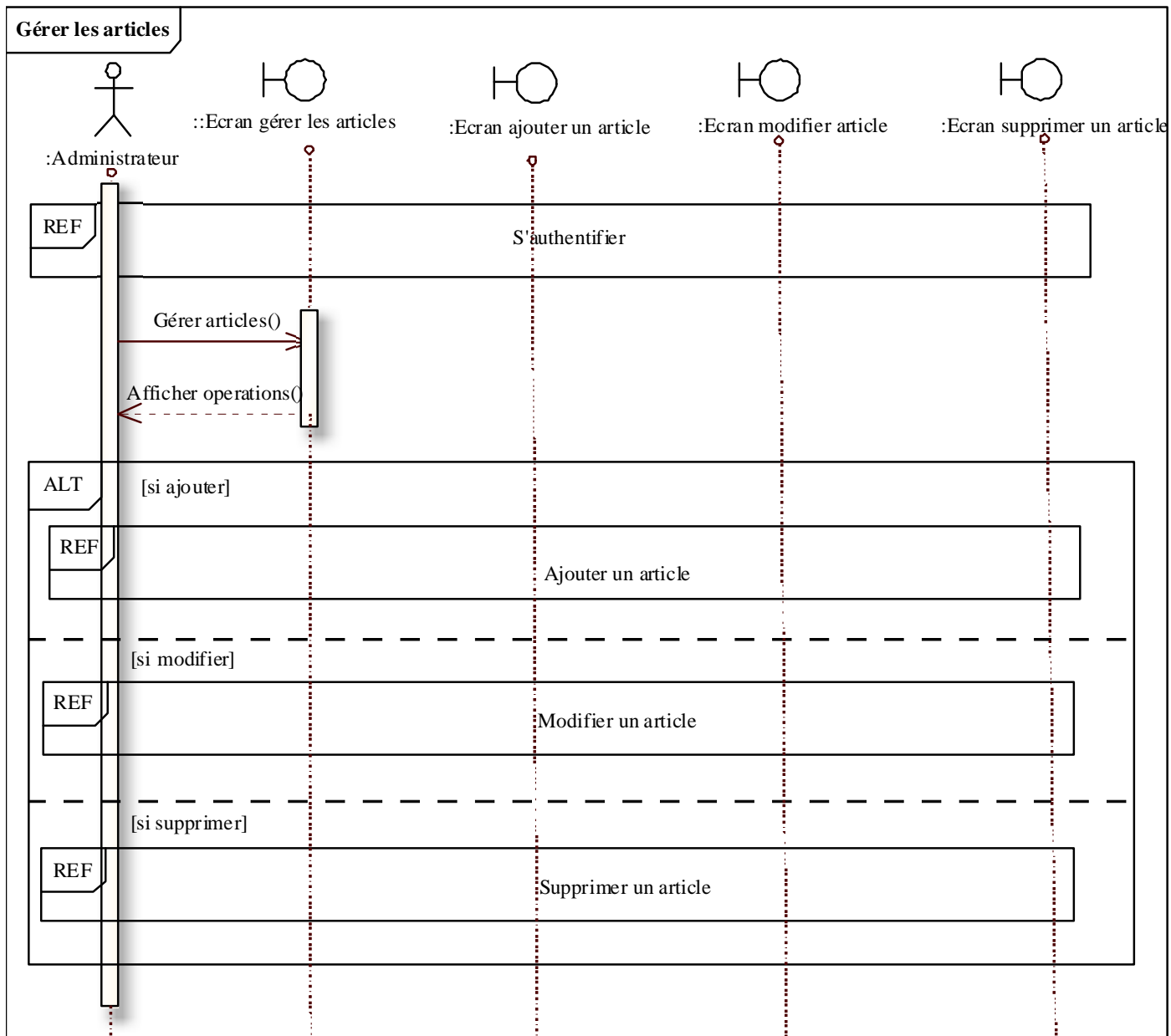
➤ Cas d'utilisation: Modifier un article



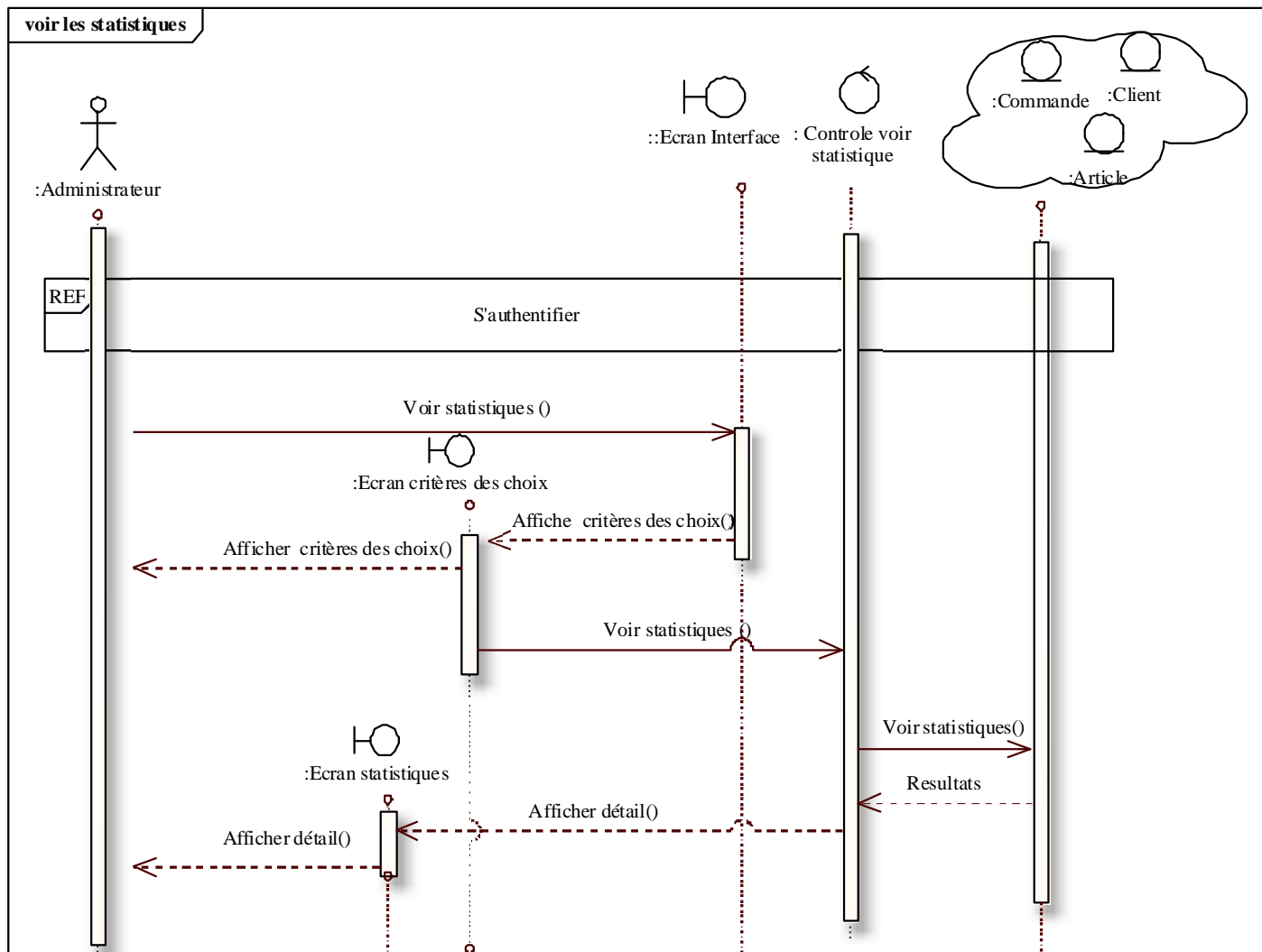
➤ Cas d'utilisation: Supprimer un article



Cas d'utilisation: Gérer les articles pour chaque catégorie



➤ Cas d'utilisation: Voir les statistiques



➤ Cas d'utilisation: Gérer les comptes

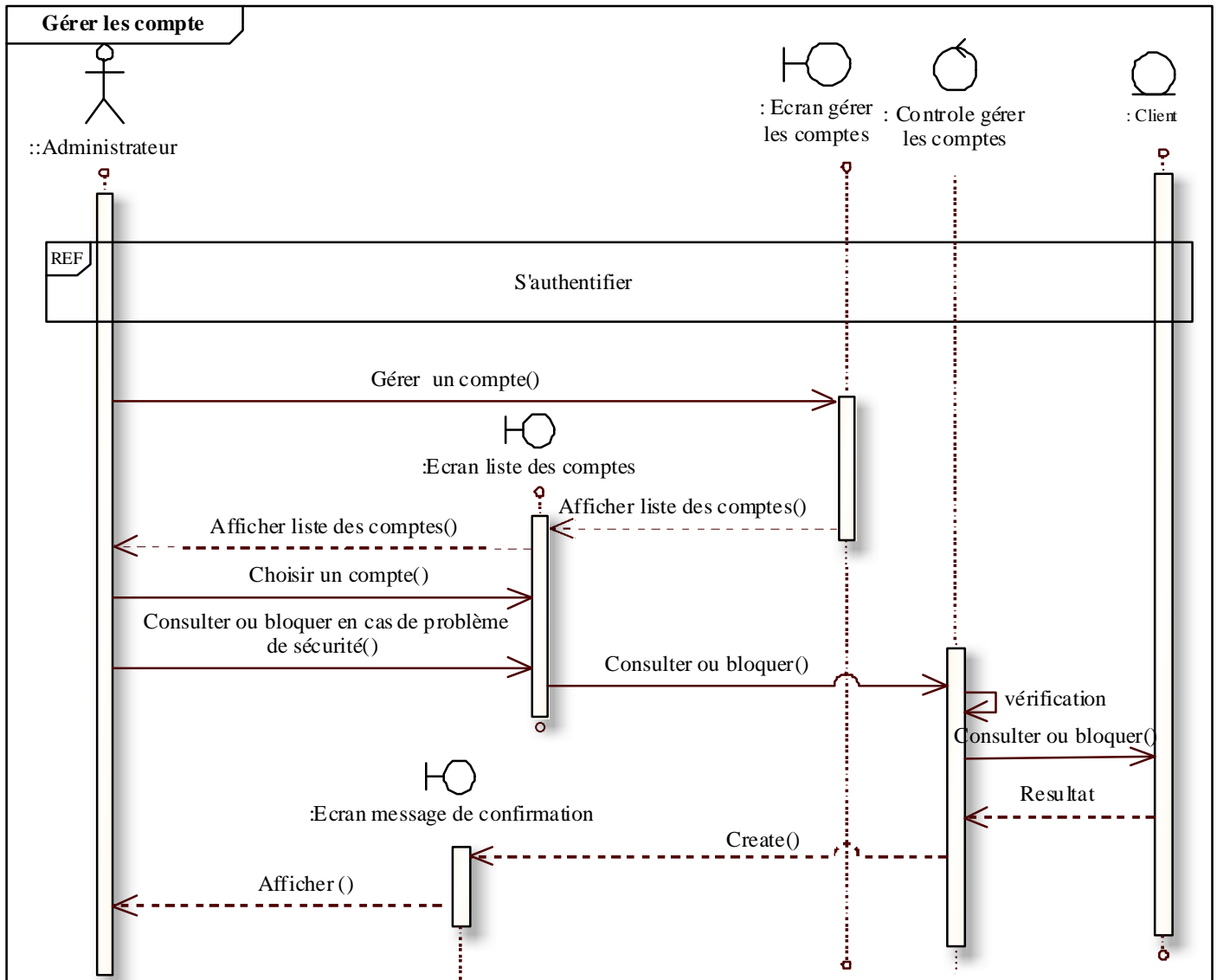


Figure IV -6: Les diagrammes de séquences du système SGAE.

5. Conclusion

Dans ce chapitre nous avons présenté l'analyse objet du système, les classes issues des besoins fonctionnels sont regroupées en catégories pour organiser le modèle structurel d'analyse. Ce modèle nécessite un travail d'analyse détaillée de la structure des classes. Celui-ci est considéré comme une base pour le développement du modèle statique et dynamique. Nous allons décrire dans le chapitre suivant la conception du notre système.

Chapitre V



Conception

1. Introduction

La phase de conception se déroule pendant trois étapes qui sont la conception générique qui est une étape de la branche droite, du processus en Y, et les deux étapes de la branche milieu du processus qui sont la conception préliminaire, c'est l'étape où s'effectue la fusion des études fonctionnelles et techniques, et la conception détaillée qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. [Roques07]

La figure suivante illustre la phase de conception dans le processus 2TUP.

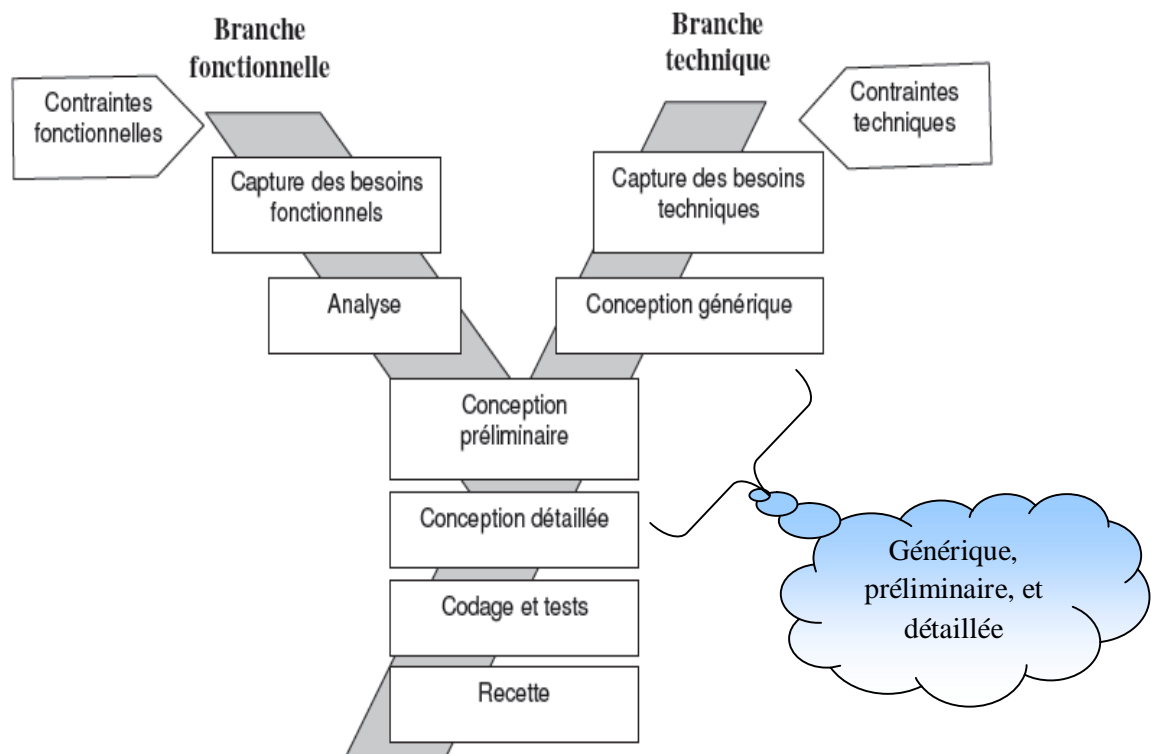


Figure V-1: L'étape de la conception

2. Conception générique

2.1. Définition :

La conception générique est située dans la branche droite du processus 2TUP, c'est la deuxième activité de la branche technique, elle consiste à développer la solution qui répond aux spécifications techniques présentées à l'étape de capture des besoins techniques. La conception générique s'appuie sur le développement de frameworks répondant aux spécifications techniques. L'organisation en architecture technique puis en composants doit ensuite répondre à des objectifs de réutilisation, de fabrication et de déploiement. [Roques07]

2.2. Elaboration du modèle logique :

L'élaboration du modèle logique nécessite l'utilisation des framework, car les classes techniques fonctionnent rarement seule, c'est pourquoi le concept clé de la conception générique est le framework technique. [Roques07]

2.2.1. Définition d'un Framework technique:

En pratique, un framework (traduction littérale: squelette d'application) est un ensemble de classes et de mécanismes associés à une architecture logicielle qui fournissent un ou plusieurs services techniques ou métiers aux applications qui s'appuient dessus. Un Framework technique ne concerne que les responsabilités de la branche droite du processus en Y.

Le modèle logique de conception technique est organisé par package de classe qui représente les frameworks développés pour résoudre les problèmes purement techniques. Nous utilisons pour spécifier les concepts de Framework technique le stéréotype « Technical Framework ». Un certain nombre de Framework ont donc été développés, qui servent à simplifier l'utilisation de telle ou telle technologie, les plus connus sont : (Spring, Jsf, Hibernate, ...).

Pour notre projet, on utilise les framework Spring et Hibernate, afin de simplifier le développement de la présentation et le contrôle de notre application, en respectant le pattern **MVC2**, qui est le modèle recommandé pour les applications J2EE. (Spring, Hibernate et MVC sont déjà détaillé dans le chapitre I).

- **Design pattern :**

Les design patterns ou modèles de conception décrivent des organisations pratiques de classes objets. Ces organisations résultent souvent d'une conception empirique, le concepteur objet tente de faciliter la réutilisation et la maintenance du code. On peut donc concevoir un modèle d'application comme une forme d'organisation transposable à plusieurs applications. [W11]

2.2.2. Organisation des Frameworks techniques:

L'organisation du modèle logique reprend les couches logicielles. A chaque couche correspond un framework technique. [Roques07]

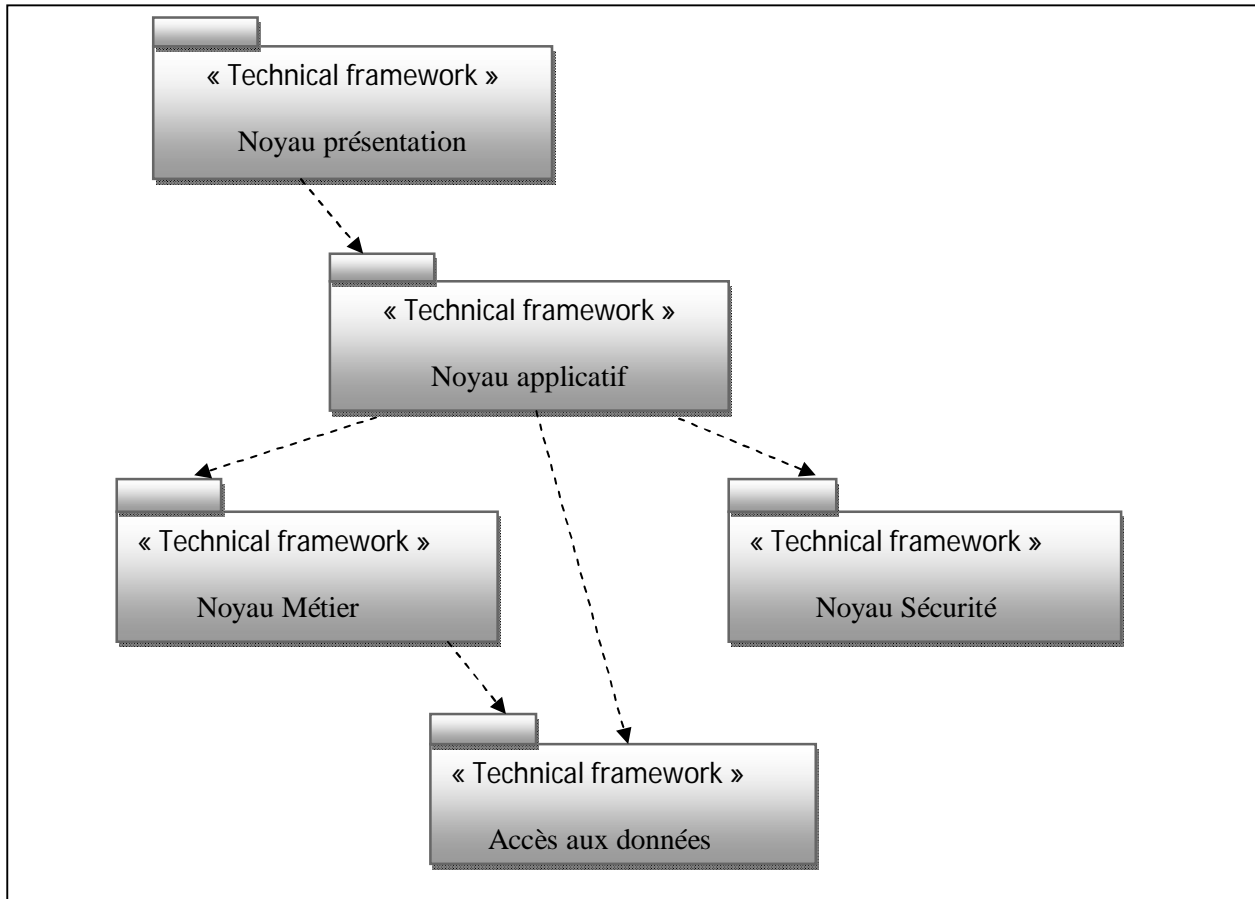


Figure V-2 : Organisation du modèle logique de SGAE.

- ✓ **Noyau Présentation:** Le noyau présentation définit les classes, les interfaces et les mécanismes de base pour réaliser l’affichage d’un objet.
- ✓ **Noyau Applicatif:** Le noyau d’application charge les modèles de fonctionnement et contrôle les commandes d’une application en utilisant les classes métier.
- ✓ **Noyau sécurité :** Le noyau sécurité définit les stratégies de contrôle et d’authentification nécessaire pour protéger le système. Il est composé des niveaux de sécurité suivants:
 - La sécurité côté SGBDR : chaque utilisateur se voit attribuer un ensemble de privilèges se rapportant à l’exploitation du système.
 - La sécurité côté applicatif : se traduit par la définition de comptes user (login/mdp) pour tous les utilisateurs du système avec la possibilité de changer de mot de passe.
- ✓ **Noyau métier :** Définit les éléments permettant d’identifier les objets métier et de définir leurs attributs caractéristiques.
- ✓ **Noyau d’accès aux données :** Définit les mécanismes de chargement, de sauvegarde, de mise à jour et de recherche des objets. [Roques07]

3. Conception préliminaire

3.1. Définition :

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car Elle représente le cœur. A cette étape on va effectuer la fusion des études fonctionnelles et techniques, La conception préliminaire est avant tout affaire d'organisation ; elle appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les Frameworks de la conception technique. [Roques07]

3.2. Développement du modèle de déploiement

- **Définition du poste de travail :** Le poste de travail représente un ou plusieurs acteurs pouvant être localisés sur une machine d'un type particulier, et remplissant une fonction identifiée dans l'entreprise, le poste de travail ne représente pas forcément une machine physique, mais peut consister en plusieurs machines, à condition qu'elles donnent lieu au même type de déploiement. . [Roques07]

- **Le modèle de déploiement :**

Les modèles de déploiement et de configuration matérielle s'expriment tous deux à l'aide d'un digramme de déploiement. 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. Pour notre système on va donner le schéma suivant qui représente son digramme de déploiement.

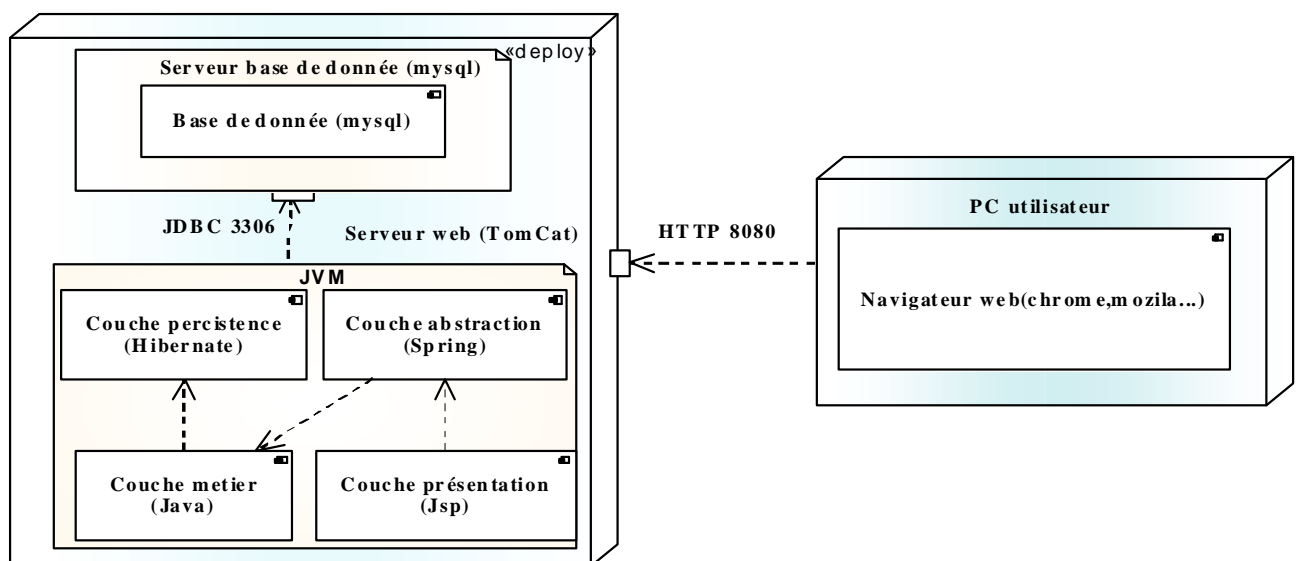


Figure V-3 : Le modèle de déploiement du SGAE.

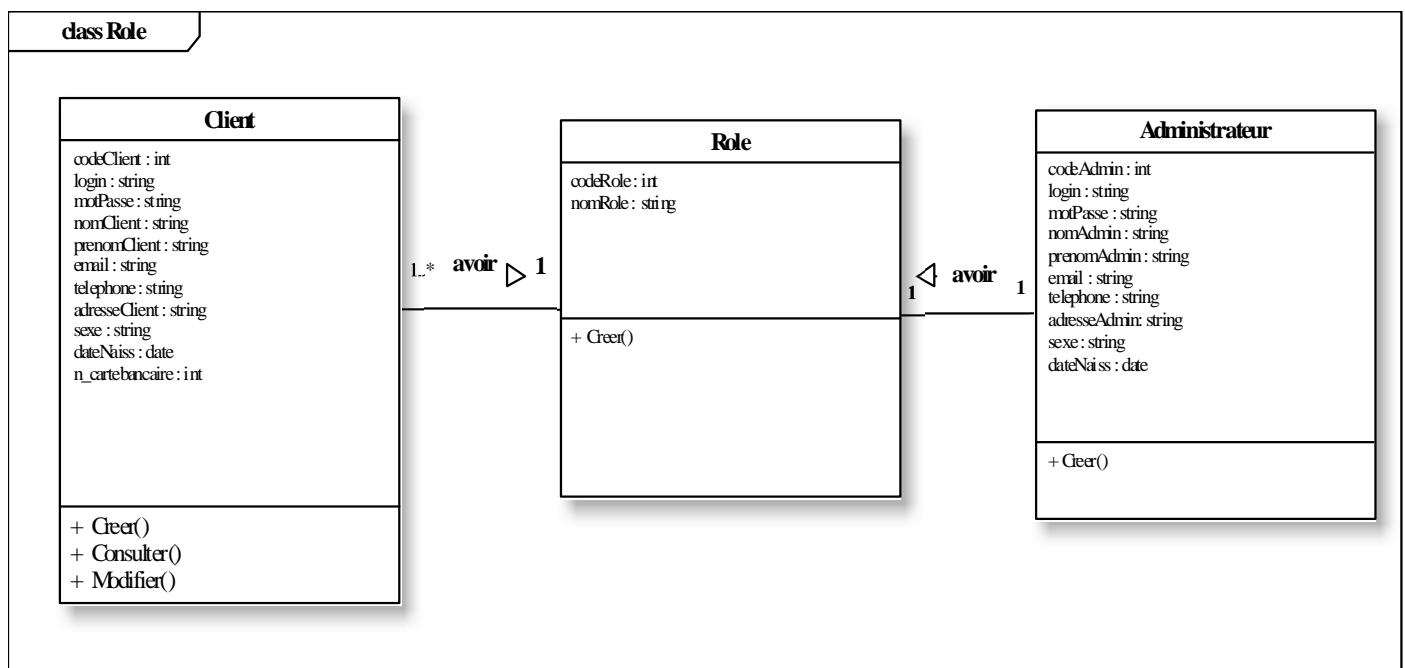
4. Conception détaillé

4.1. Définition:

L'étape de conception détaillé est la dernière étape de la conception, dans cet étape les concepteurs construisent les classes par la conception des attributs, des opérations, et faire aussi la conception des associations, ensuite ils construisent les différentes tables de l'application.

4.1.1. Concevoir les attributs et les opérations:

La conception des attributs consiste principalement à définir le type des attributs identifiés en analyse, Pour chaque attribut on donne leur type, qui est en général un type de base, pour concevoir ou documenter une méthode, UML met à notre disposition toute la palette des diagramme du modèle dynamique, nous avons basé sur le diagramme de séquence réalisé dans l'étape d'analyse.



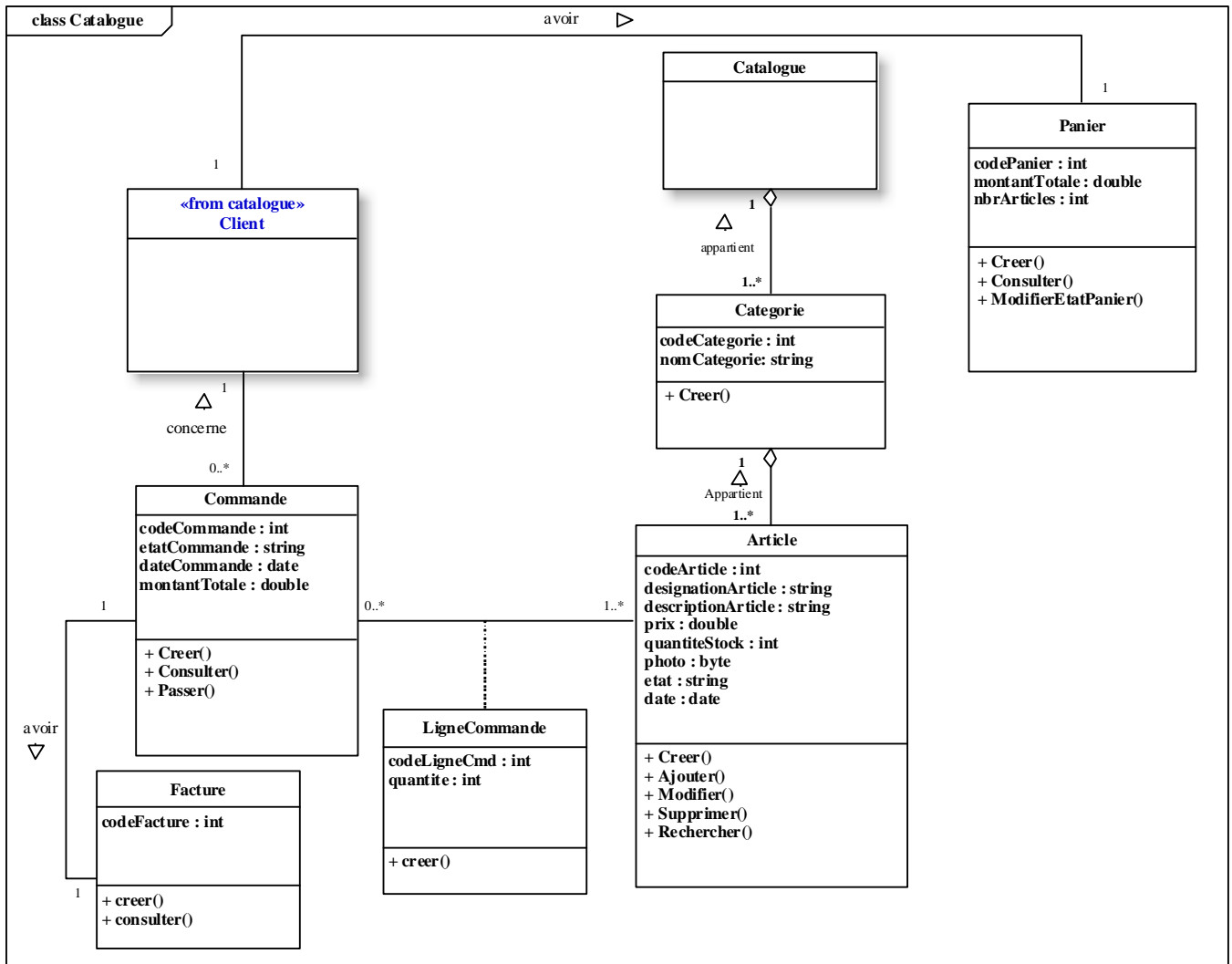


Figure V-4 : La conception des attributs et des opérations

4.1.2. Le diagramme de classe:

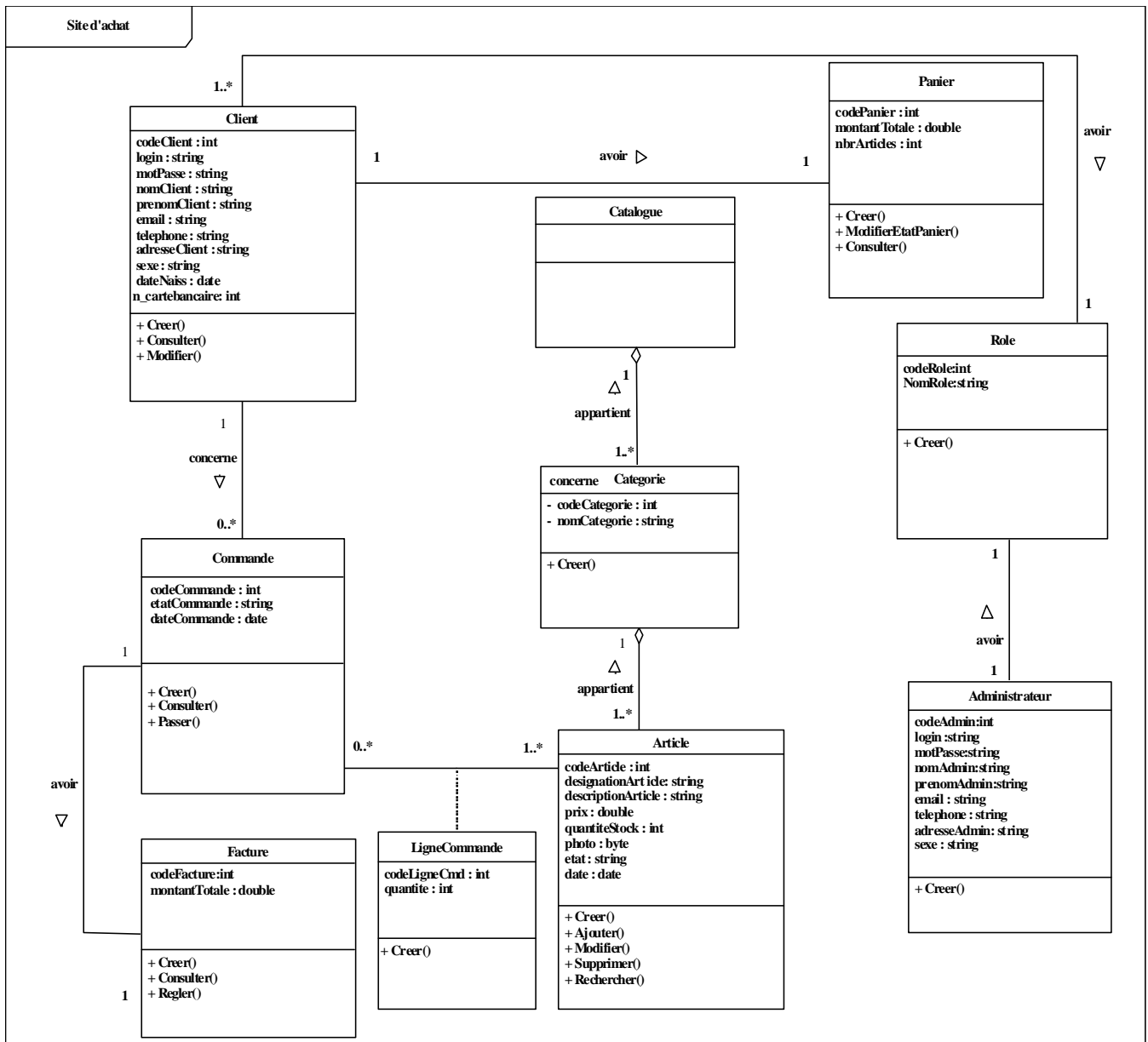


Figure V-5 : Le digramme de classe

4.1.3. Concevoir les Associations :

L’association est un concept inconnu de la plupart des langages orientés objet. Elle se transforme en attribut ou en tableau d’attributs suivant sa multiplicité. [Roques07]

Les règles de transformation :

Une association ayant une multiplicité « 0 - * » ou « 1 - * » : La transformation est faite par l’ajout en qu’en tant qu’attribut simple de la clé de la classe situé du coté de multiplicité 0 ou 1, dans la classe située du coté de la multiplicité *.

Une association ayant une multiplicité « * - * » et les classes associations : cette association dans le diagramme de classe se traduit par une relation supplémentaire ayant pour clé, la concaténation des clés primaires des classes connectés à l'association chaque attributs devient clé étrangère, les attributs de la classe association doivent être ajouté à la nouvelle relations, ces attributs ne sont ni clé primaire, ni clé étrangère.

Une association ayant une multiplicité « 1 - 1 » : dans ce cas d'association il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. Cette règle permet d'éviter les valeurs nulles dans la base de données.

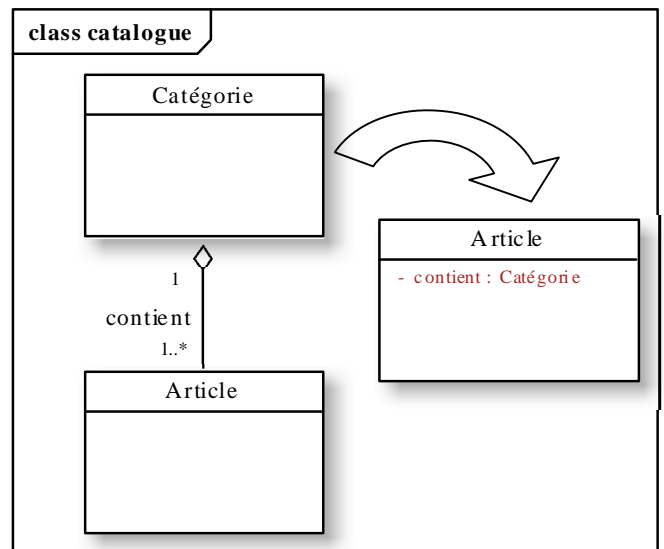
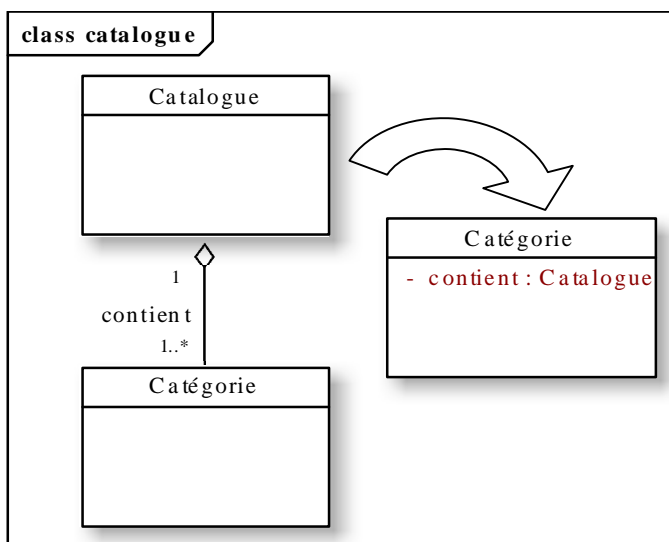
La traduction des associations de notre système va illustrer dans les figures suivantes :

Une association d'héritage:

Méthode1 : Créer une relation avec tous les attributs des classes (la super classe et les sous-classes), et il faut ajouter un attribut pour distingue le type des objets.

Méthode2 : Créer une relation pour chaque sous classe, chaque relation se compose des attributs génériques et d'attributs spécifiques.

Méthode3 : Créer une relation par classe et des associations.



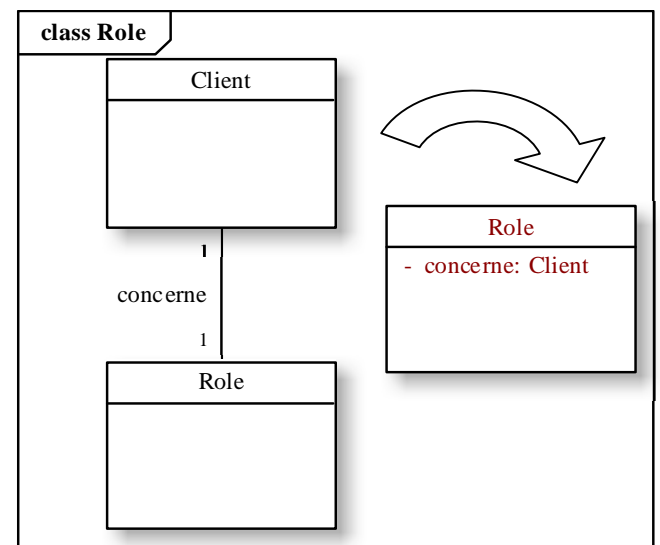
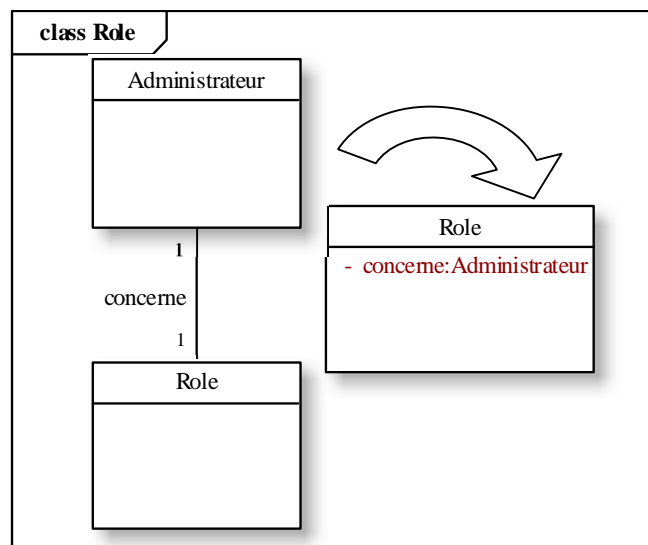
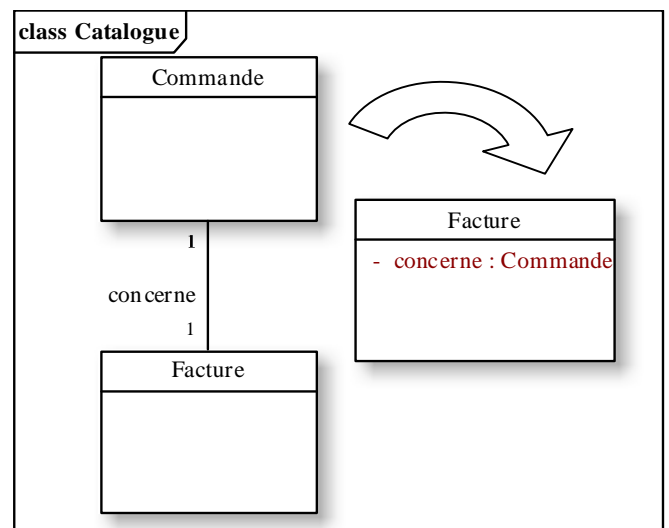
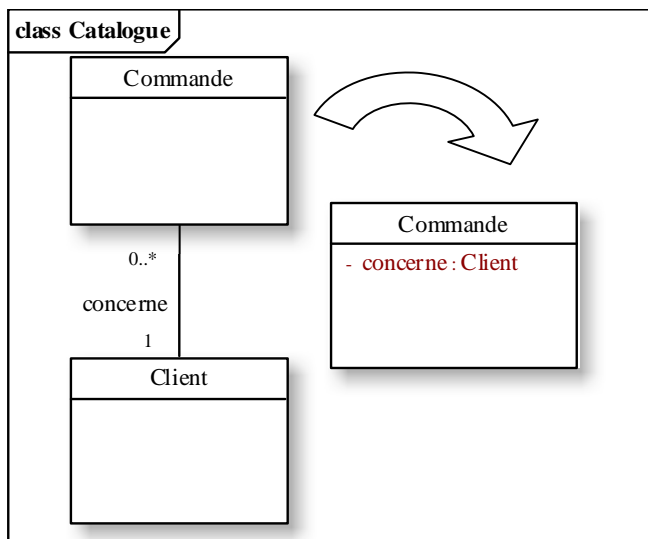
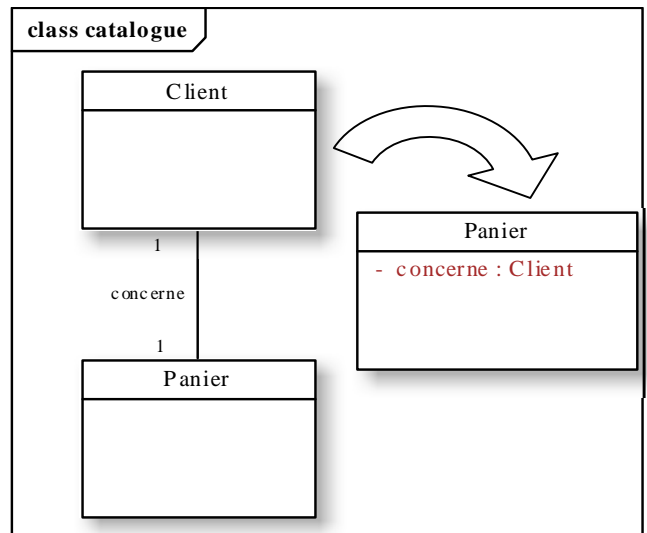
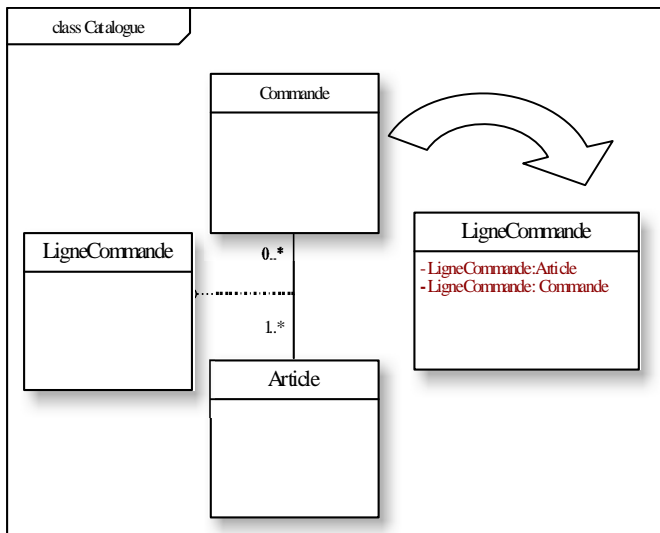


Figure V-6 : La transformation des associations

4.2. Les classes modifiées:

Après la transformation des associations du diagramme de classe de notre système, on va donnée les nouvelles classes illustrés dans la figure qui suit :

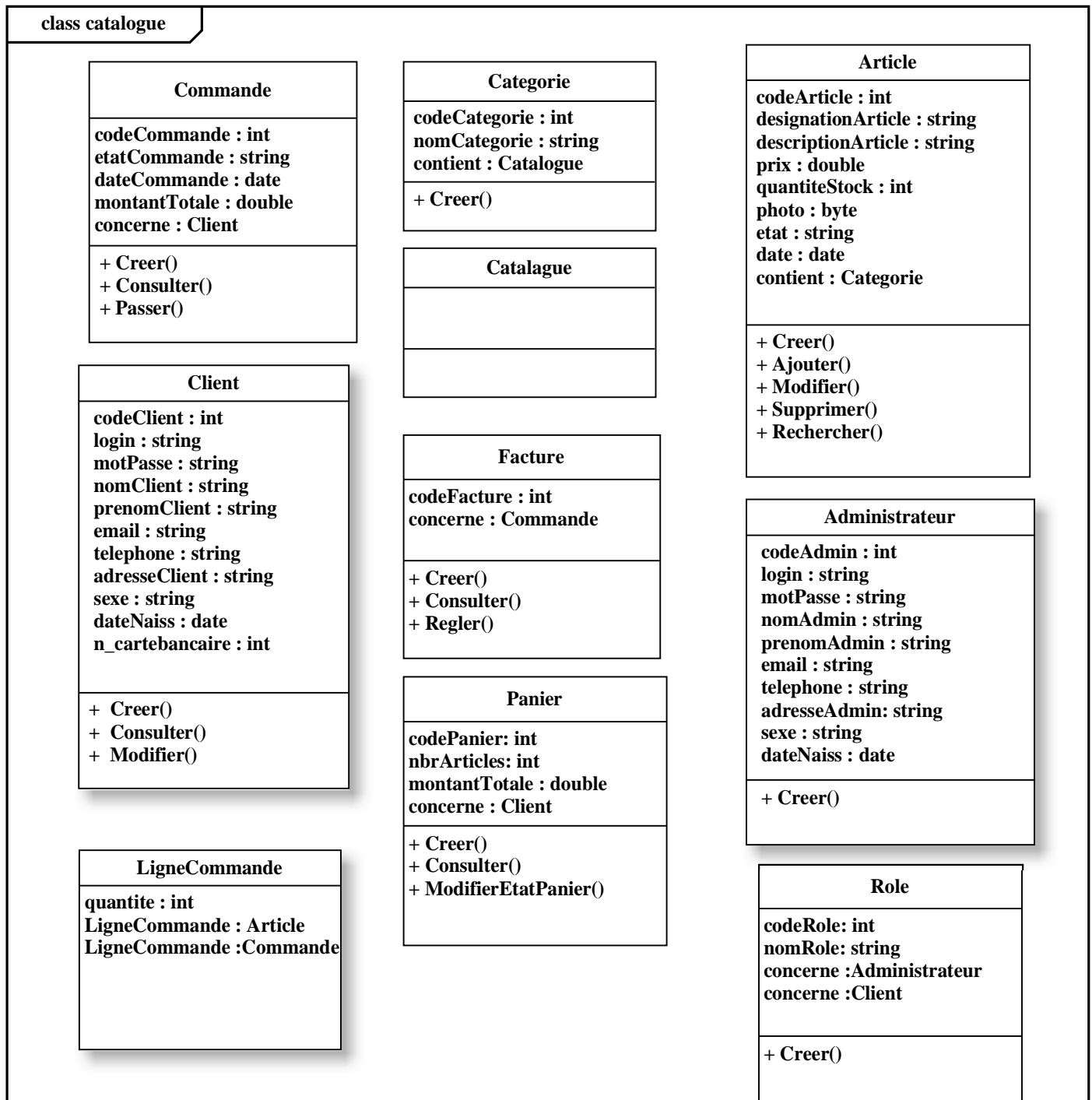


Figure V-7 : Les classes modifiées

4.3. Le Modèle de base de données:

Nous avons utilisé le framework Hibernate qui est une solution open source de type ORM (Object Relational Mapping) pour le monde java, le terme « mapping » Objet/Relationnel décrit la technique consiste à faire le lien entre la représentation objet des données et sa représentation relationnelle.

Hibernate permet d'assurer et de faciliter la persistance des objets d'une application, en réalisant lui-même la création de ces objets et les traitements de remplissage de ceux-ci en accédant à la base de données. Il propose aux développeurs des méthodes d'accès aux données plus efficaces par l'utilisation des requêtes HQL (Hibernate Query Langage) qui peut donc réduire le temps de développement.

Hibernate est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données (MySQL, DB2, Oracle...). Qui assure la portabilité des applications.

5. Conclusion

La phase de conception prend en compte les choix d'architecture technique retenue pour le développement et l'exploitation du système. Elle vise à trouver des solutions informatiques et techniques pour mettre en œuvre et construire le système analysé au cours des phases précédentes.

Il nous reste qu'une dernière étape qui est la phase de réalisation qu'on va l'aborder dans le chapitre suivant.

Chapitre VI



Implémentation

1. Introduction

Dans le cadre de ce dernier chapitre, nous allons présenter brièvement la structure, les différents outils et technologies de programmation utilisée pour l'implémentation et la réalisation de notre application web. Et par la suite nous allons présenter quelques interfaces de notre système.

2. Langage de programmation et choix techniques

Pour la réalisation de notre système nous avons choisi java comme un langage de programmation et une panoplie de technologies autour de lui à savoir : Servlet, JSP, Spring, Hibernate, HTML, JavaScript et les CSS.

2.1. Java :



Java est un langage de programmation et une plate-forme informatique très utilisé, notamment par un grand nombre de développeurs professionnels, ce qui en fait un langage incontournable actuellement, qui ont été développé par Sun Microsystems aujourd'hui racheté par Oracle. [W13]

Le langage Java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation tels que Windows, Mac OS ou Linux. De nombreux types de programmes peuvent être réalisés avec Java telle que les sites web dynamiques, avec J2EE (Java 2 Enterprise Edition, maintenant JEE), etc. [W12]

2.2. Les API java EE :

➤ Servlet :

Une servlet est en réalité une simple classe Java, qui reçoit une requête HTTP envoyée depuis navigateur, effectue des traitements et renvoie la réponse. [W14]

➤ JSP :

Java Server Page est une technique Java qui permet de générer dynamiquement du code HTML. Elle mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client. [W15]

➤ JPA :

Java Persistence API permet fait la correspondance entre le modèle de données relationnel et le modèle objet de la façon la plus facile possible. [W16]

➤ **JSTL :**

JSP Standard Tag Library c'est un ensemble de balises personnalisées (Custom Tag), elle permet d'éviter l'utilisation de code Java dans les pages JSP et de réduire la quantité de code à écrire ainsi que rend le code des pages JSP plus lisible. [W17]

➤ **spring :**



Spring est un framework open source JEE pour les applications n tiers, dont il facilite le développement et les tests. Il est considéré comme un conteneur dit « Léger », c'est-à-dire une infrastructure similaire à un serveur d'applications J2EE. [W18]

➤ **Hibernet :**



Hibernate est une solution open source du type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Il permet donc de représenter une base de données en objets Java et vice-versa. [W9]

2.3. Les standards web :

➤ **CSS:**

Cascading Style Sheets est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. CSS est un le langage de décoration d'une page web. [W19]

➤ **Javascript :**

Le Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. [W20]

➤ **HTML (HyperText Markup Language):**

C'est le langage universel utilisé sur les pages Web, lisibles par tous les navigateurs Web. Ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage. [W21]

3. Outils et environnement de développement

➤ L'IDE ECLIPSE



Eclipse est un IDE, (*Integrated Development Environment*), dédié spécialement au développement de logiciels basés sur Java. C'est un outil puissant, gratuit, libre et multiplateforme, massivement utilisé en entreprise. Il permet l'intégration des outils nécessaires au développement et au déploiement d'une application.

➤ Le serveur Tomcat



Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion.[W22]

Il comporte également un serveur HTTP. Il peut être utilisé en autonomie avec son propre serveur web ou en collaboration avec d'autres. [W22]

➤ MYSQL :



Le SGBD MySQL est supporté par un large éventail d'outils. MySQL est surtout installé pour les applications Web. Ce SGBD est solide, libre, gratuit, est utilisé par de grands groupes spécialisés dans l'internet. [W23]

➤ Pacestar UML Diagrammer:

Pacestar UML Diagrammer est une alternative à d'autres programmes qui aident les programmeurs, ingénieurs système, ou d'autres professionnels pour créer des diagrammes UML. Il est livré avec un grand nombre de fonctionnalités, ainsi que des options de personnalisation et incluent tous les outils nécessaires pour aider les utilisateurs des schémas de conception à des fins diverses.

4. Présentation de quelque interface de notre application

Dans cette partie nous allons présenter les interfaces principales de notre application.

- **L'interface d'accueil**

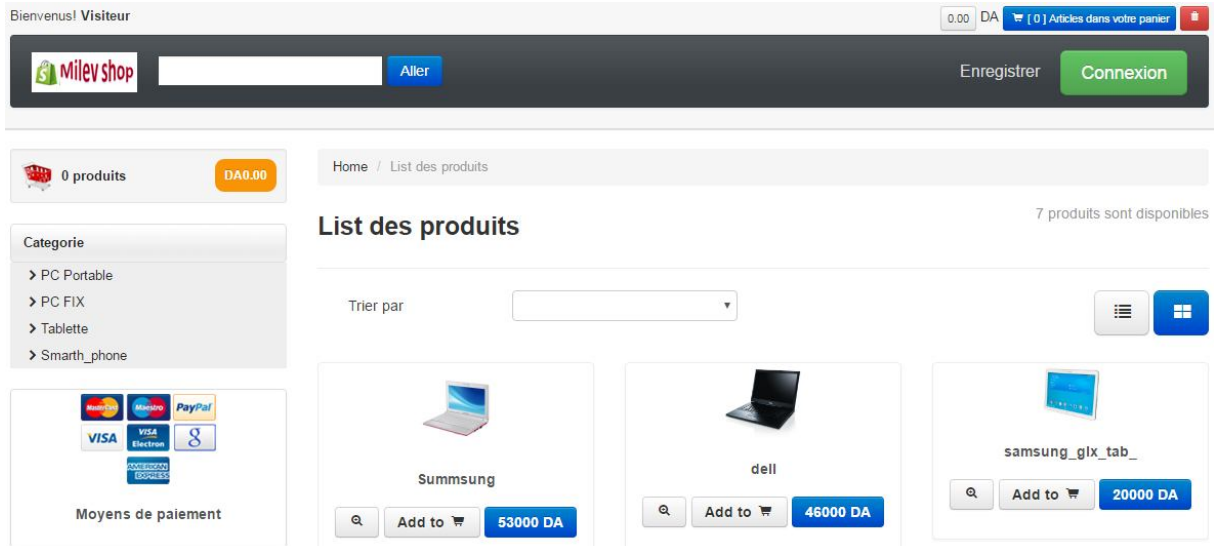


Figure VI-1 : Fenêtre d'accueil

- **L'interface d'inscription**

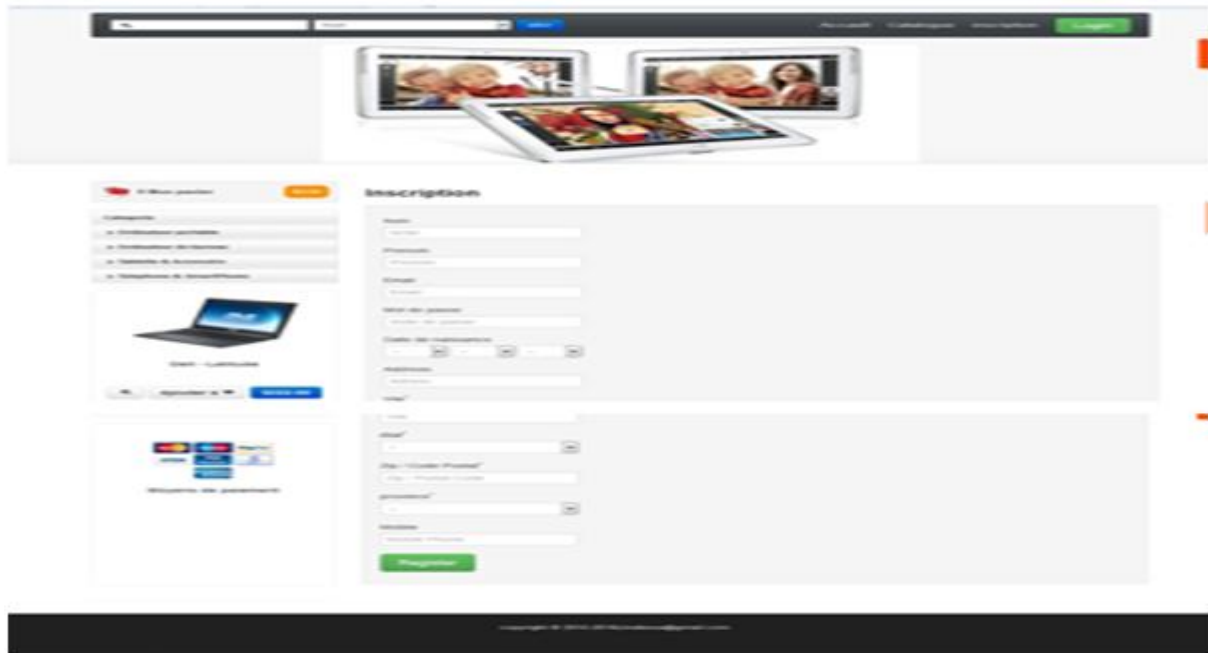


Figure VI -2 : Fenêtre d'inscription

- L'interface panier

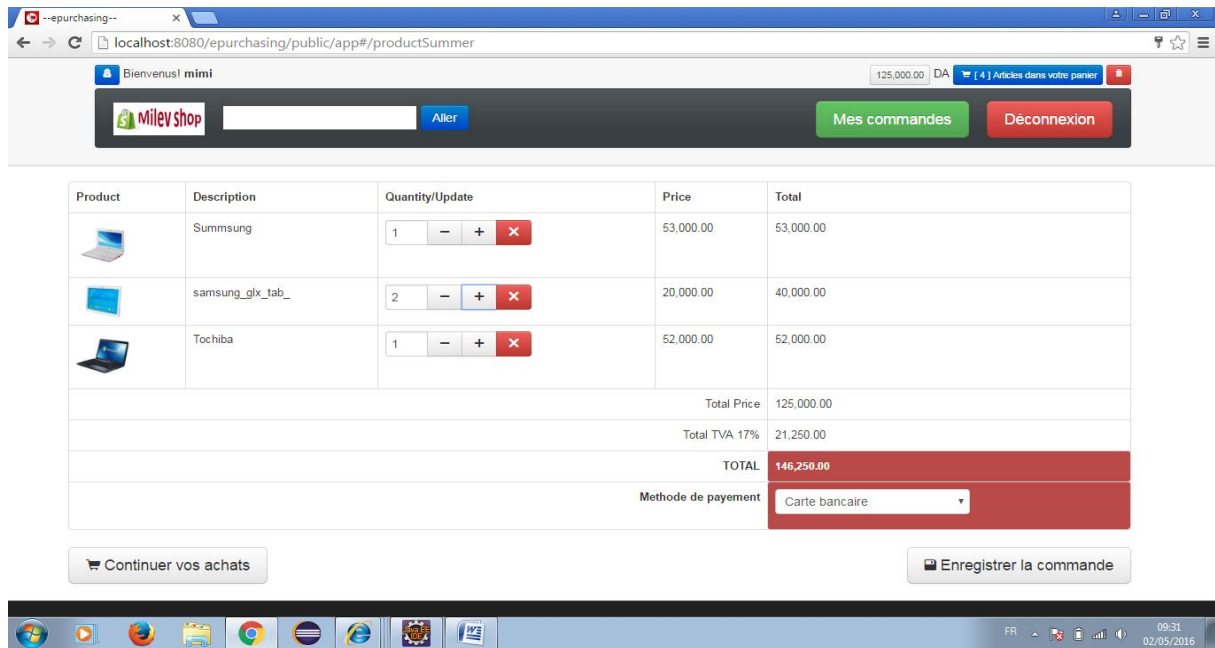


Figure VI -3 : Fenêtre de panier

- L'interface des commandes enregistrées

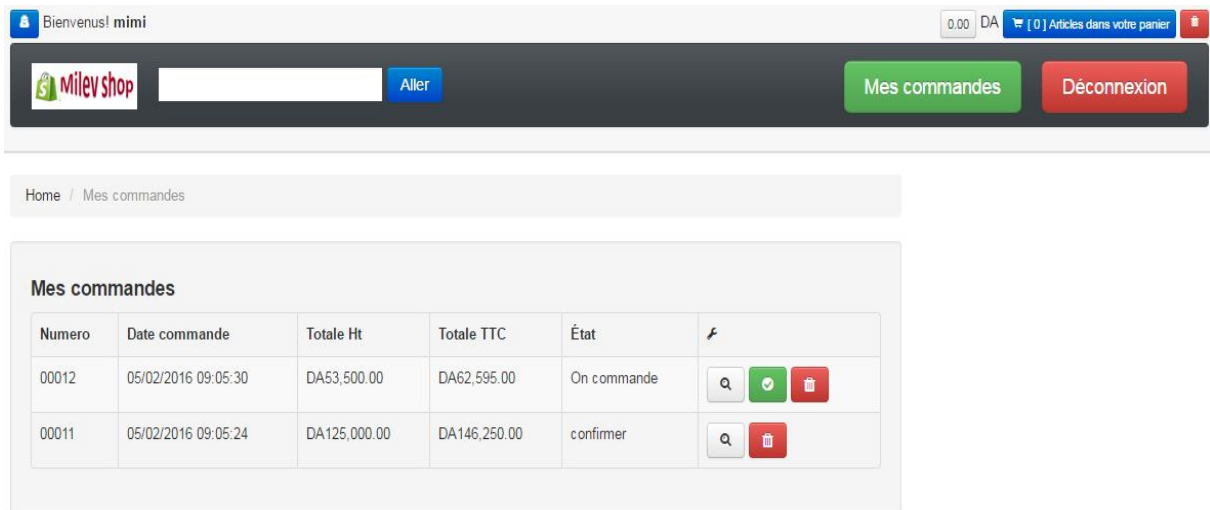


Figure VI-4 : Fenêtre liste des commandes

- L'interface d'authentification

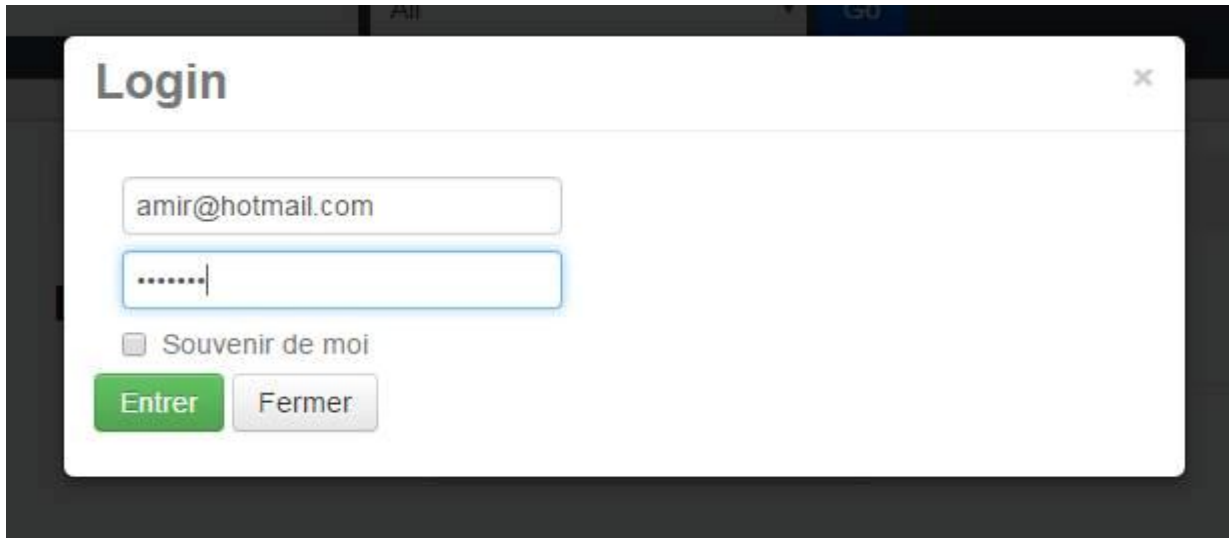


Figure VI -5 : Fenêtre d'authentification de l'administrateur

- L'interface Gérer la liste des articles

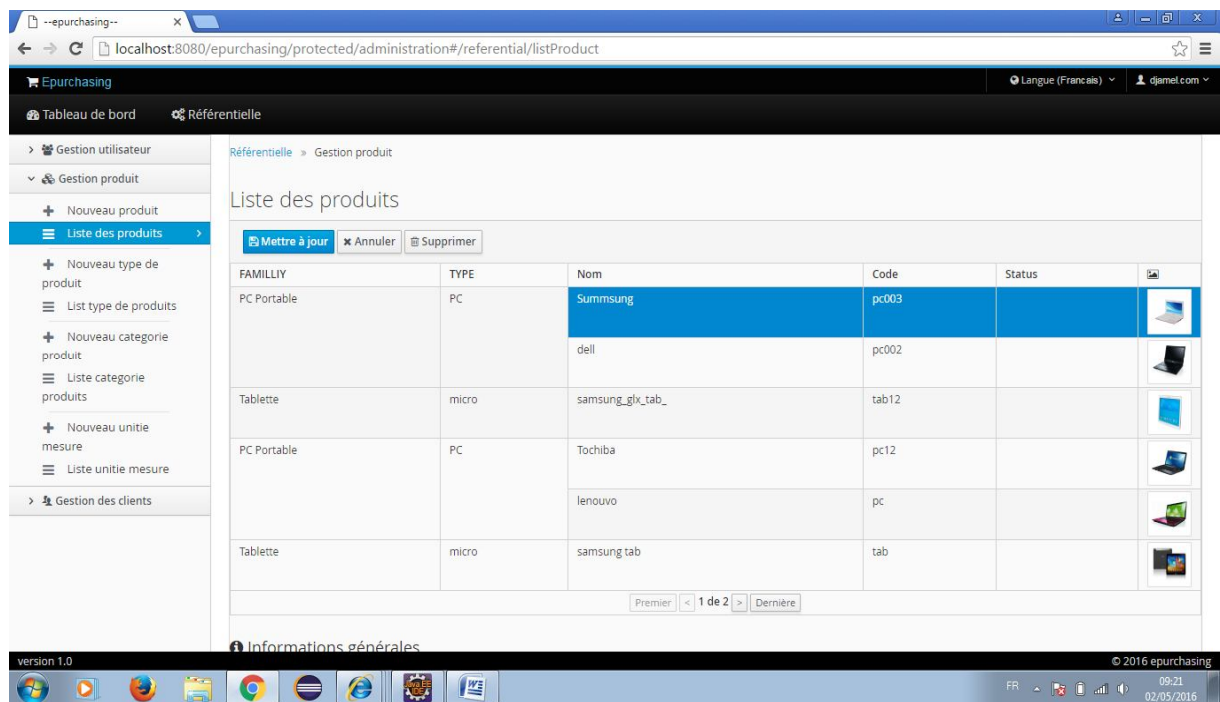


Figure VI -6 : Fenêtre de gestion des articles

• L'interface d'ajout un article

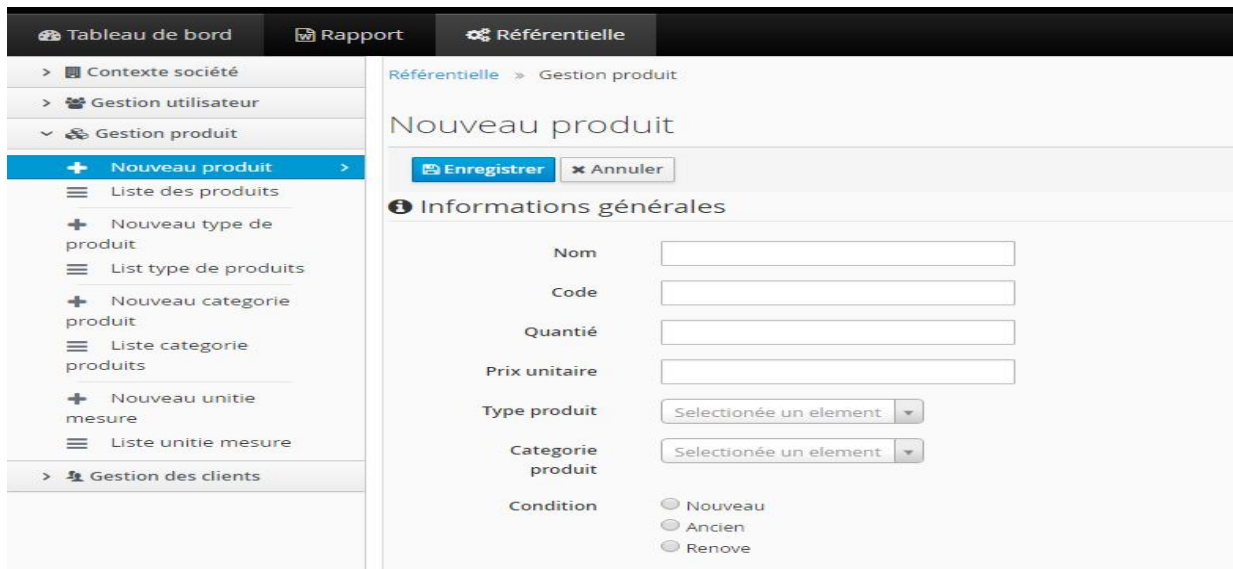


Figure VI -7 : Fenêtre d'ajout d'un article

• L'interface voir les statistiques

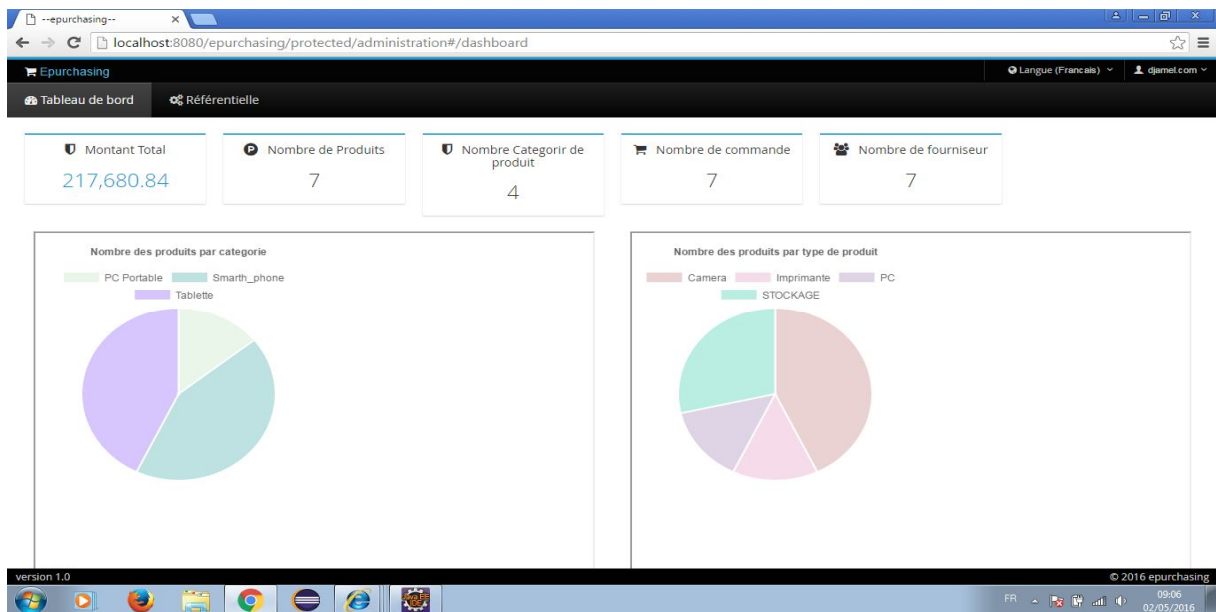


Figure VI -8 : Fenêtre voir les statistiques

5. Conclusion

Dans ce dernier chapitre nous avons présenté tout ce qui est lié au développement de notre application. Nous avons parlé des différents outils et choix technologiques utilisés dans l'implémentation des différents composants. Le chapitre est clôturé par une description détaillée des interfaces graphiques de notre application. Il ne reste qu'à conclure ce projet dans la section suivante.

Conclusion générale

À travers ce projet, nous nous sommes intéressés de conçu et réalisé une application web de gestion des achats en ligne. Une étude a été réalisée, en récoltant les informations nécessaires. Ce projet a contribué à améliorer nos compétences dans plusieurs domaines. Il nous a permis de nous perfectionner en améliorant nos connaissances en conception et en programmation. Nous avons appliqué au maximum possible les règles de base permettant d'avoir une application performante.

Nous avons utilisé le langage UML pour la modélisation du système en suivant les étapes du processus (2 TUP). Au niveau du développement de l'application, nous avons utilisé l'architecture JEE (Jsp, servlet, modèle MVC, etc.) basée sur la souplasse de Java, qui est une plate-forme très puissante, elle intègre différents outils qui facilitent le développement des applications d'entreprise par l'utilisation des frameworks Spring et Hibernate. Pour le déploiement nous avons utilisé le serveur web Tomcat, et le SGBD MYSQL pour la base de données.

Malgré ce qui a été fait, ce travail n'est pas encore terminé il peut être enrichi par d'autres fonctionnalités telles que le paiement en ligne, une boîte de messagerie, etc. Enfin, nous espérons que ce modeste mémoire soit un modèle pour les autres étudiants et que sa lecture a été agréable et claire.

Bibliographies

Les Ouvrages

[Roques07] : Pascal Roques & Frank Vallée : « UML en action de l'analyse des besoins à la conception en Java » édition Groupe Eyrolles, 2007.

[Pascal_web 07] : Pascal Roques : « Les cahiers du programmeur UML2 modélisé une application Web » édition Groupe Eyrolles, 2007.

Les Mémoires

[M1] « Conception et réalisation d'une application web pour l'achat, la vente et la location des immobiliers en ligne»
[Mémoire préparé En vue de l'obtention du diplôme de Master, Année Universitaire
2014/2015, Centre Universitaire Abdelhafidboussouf Mila].

[M2] « Développement d'une application web intégrant une base de données répartie en utilisant la plateforme Java EE»
[Mémoire préparé En vue de l'obtention du diplôme de Master, Année Universitaire
2014/2015, Centre Universitaire Abdelhafidboussouf Mila].

[M3] « Conception et réalisation d'une application web pour une agence immobilière»
[Mémoire préparé En vue de l'obtention du diplôme de Master, Année Universitaire
2014/2015, Centre Universitaire Abdelhafidboussouf Mila].



Web graphics

Les sites web

[W0]:<http://www.aquasys.fr/marketing/panel-2tup.html>.

[W01]:http://elearning.univjijel.dz/elearning/pluginfile.php/3915/mod_resource/content/1/Polycopi%C3%A9-Boukraa_SIMA.pdf.

[W02]:<http://slideplayer.fr/slide/181812/>.

[W03]:<http://www.commentcamarche.net/contents/j2ee/j2ee-intro.php3>.

[W04]:<http://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>.

[W1]:https://fr.wikipedia.org/wiki/Application_web.

[W2]: <https://www.sofracs.com/creation-application-web-mobile>

[W3] :https://fr.wikipedia.org/wiki/Architecture_trois_tiers

[W4]:[http://www.commentcamarche.net/contents/j2ee/j2ee-intro.php3\(j2ee\)](http://www.commentcamarche.net/contents/j2ee/j2ee-intro.php3(j2ee)).

[W5]:<http://javaj2ee-firststeps.blogspot.com/>.

[W6]:<http://www.jmdoudoux.fr/java/dej/chap-frameworks.htm#frameworks-4>.

[W7]:http://wiki.ouieuhoutca.org/doku.php?id=design_patterns.

[W8]:http://wpetrus.developpez.com/java/struts/data/tutorial_struts.pdf.

[W9]:<http://www.jmdoudoux.fr/java/dej/indexavecframes.htm>.

[W10]:

<http://diuf.unifr.ch/drupal/sites/diuf.unifr.ch.drupal.softeng/files/teaching/studentprojects/giller/download/report.pdf>.

[W11]: <http://abrillant.developpez.com/tutoriel/java/design/pattern/introduction/>.

[W12] :[https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))

[w 13] :https://openclassrooms.com/courses/apprenez-a-programmer-en-java?utm_source=google&utm_medium=cpc&utm_term=search&utm_campaign=397812015

[W14] : <https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee/la-servlet>

[W15]: <http://litis.univ-lehavre.fr/~duvallet/enseignements/Cours/JEE/COURS-JSP-4p.pdf>

[W16] : http://spoonless.github.io/epsi-b3-javaee/12_jpa_part1.html

[W17] :

https://archive.org/stream/PHPMySQLAvecFlash8/Les%20cahiers%20du%20programmeur%20Java%20EE%205_djvu.txt

[W18] :[https://fr.wikipedia.org/wiki/Spring_\(framework\)](https://fr.wikipedia.org/wiki/Spring_(framework))

[W19] :<http://glossaire.infowebmaster.fr/css/>

[W20]:<http://www.commentcamarche.net/contents/577-javascript-introduction-au-langage-javascript#qu-est-ce-que-le-javascript>

[W21] :<http://glossaire.infowebmaster.fr/html/>

[W22] :https://quentinv.gitbooks.io/installation-de-tomcat-sur-debian/content/presentation_tomcat.html

[W23] :http://toubkalit.com/chapitre/Rapport_de_conception.pdf

Résumé

Dans le but d'améliorer sa performance, aujourd'hui l'entreprise vise à automatiser la gestion interne de ses activités en faisons appel à des techniques informatiques.

Notre projet se concentre sur l'étude, la conception et la réalisation d'une application web JEE pour les achats en ligne et la possibilité de présenter les articles dans une boutique virtuelle à la disposition de tout le monde.

Pour la réalisation de ce projet, nous avons utilisé UML comme langage de modélisation en s'appuyant sur le processus 2TUP. Pour l'implémentation de notre application Web, nous avons utilisé le langage de programmation JAVA pour le Web (Jsp, Servlet, etc.), avec l'utilisation du framework spring qui met en œuvre le modèle MVC2, et le framework hibernate pour implémenter la couche d'accès aux données

Mots clés: UML, 2TUP, java EE, Tomcat, JSP, Servlet, Spring, Hibernate, etc.

Abstract

In order to improve performance, today the company aims to automate internal management of its activities by the use of IT techniques.

Our project focuses on the study, design and implementation of a web application for online shopping and the opportunity to present items in an online store available for everyone.

For this project we used the UML modeling language based on the 2TUP process. For the implementation of our web application, we used the Web java programming language (Jsp, Servlet, ...), with the use of the framework spring, that implements the MVC2 model, and the framework hibernate to implement the data access layer.

Keywords: UML, 2TUP, JAVA, java EE, Tomcat, JSP, Servlet, Spring, Hibernate, ...

التلخيص

من أجل تحسين الأداء، تهدف الشركة اليوم لتسيير أنشطتها بطريقة أوتوماتكية، وذلك بالاعتماد على تكنولوجيا الاعلام الالي.

يركز مشروعنا على دراسة، تصميم وتنفيذ تطبيق ويب JEE للتسوق عبر الانترنت و إمكانية عرض السلع في متجر افتراضي يكون في متناول الجميع .

من أجل انجاز هذا المشروع استعملنا UML كلغة نمذجة بالاعتماد على عملية 2TUP . لتنفيذ تطبيق الويب، استخدمنا لغة البرمجة جافا الخاصة بالويب (JSP، Servlet، الخ)، مع استخدام إطار Spring من اجل تطبيق النموذج MVC2، والإطار Hibernate لتنفيذ طبقة الوصول إلى البيانات.

الكلمات المفتاحية : UML, 2TUP, java EE, Tomcat, JSP, Servlet, Spring, Hibernate, ...