



N° Réf :.....

Centre Universitaire
Abd elhafid Boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatiques

**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 mobile pour la gestion des
services bancaires**

Préparé par : - Zemmouri Amel
- Boulaiche Nassima

Soutenue devant le jury :

Encadré par : BOUFAGHES Hamida..... M.A.B
Président : DOUAS Bilal M.A.A
Examineur : HEDJAZ Sabine M.A.B

Année universitaire : 2015/2016

Remerciements

Tout d'abord, je tiens à remercier Allah, le clément et le miséricordieux de m'avoir donné le courage d'accomplir ce modeste travail.

*Mes sincères reconnaissances vont à mon encadreur **Boufaghes Hamida** qui m'a soutenue avec sa patience, ses encouragements et ses conseils.*

Un grand merci sera également adressé aux membres du jury qui m'ont fait l'honneur de présider et d'examiner ce modeste travail.

*Notre reconnaissance s'adresse à **nos familles** qui ont su nous apporter, sans relâcher leur soutien durant toutes ces longues années d'études.*

Enfin, nous remercions également, tous les enseignants, qui ont assurés notre formation durant cycle universitaire.

A tous, un grand Merci.

Dédicaces

Je remercie dieu qui a toujours été à mes côtés.

Je dédie ce modeste travail

Aux fleurs de ma vie, Ma mère, ma raison d'être, ma raison de vivre,

La lanterne qui éclaire mon chemin

Et m'illumine de douceur et d'amour

Mon père, en signe d'amour, de reconnaissance et de gratitude pour tous

Les soutiens et les sacrifices dont il a fait preuve à mon égard

*Mes dédicaces vont également à mon frère « **Mohamed Ali** »*

*Et mes sœurs « **Safa, Aya et Oumaima** »*

*Sans oublier de dédier ce mémoire à mon encadreur Mme **Boufaghes Hamida**,*

*À mon binôme **Nassima**,*

À tous mes amis «

En témoignage de l'amitié sincère qui nous a liés Et des bons moments passés ensemble À tous les gens qui ont cru en moi et qui me donnent l'envie D'aller en avant

Je veux remercier tous, votre soutien et vos encouragements me donnent

La force de continuer

Amel

Dédicaces

Je remercie dieu qui a toujours été à mes côtés.

Je dédie ce modeste travail

Aux fleurs de ma vie, Ma mère, ma raison d'être, ma raison de vivre,

La lanterne qui éclaire mon chemin

Et m'illumine de douceur et d'amour

Mon père, en signe d'amour, de reconnaissance et de gratitude pour tous

Les soutiens et les sacrifices dont il a fait preuve à mon égard

Mes dédicaces vont également à mes frères « Adil, Nadir, Mohamed, Bedreddine,

Samir et Anis »

Et ma sœur « Fatima zohra »

A ma tante « Fadiya »

Sans oublier de dédier ce mémoire à mon encadreur Mme Boufaghes Hamida,

À ma binôme Amel,

A tous mes amis «»

En témoignage de l'amitié sincère qui nous a liés Et des bons moments passés ensemble A tous les gens qui ont cru en moi et qui me donnent l'envie D'aller en avant

Je veux remercier tous, votre soutien et vos encouragements me donnent

La force de continuer

Nassima

Résumé

La banque est parmi les plus importantes entreprise de service, qui offre des services en introduisant les nouvelles technologies de l'information et de la communication. Vu l'importance des services bancaires au sein de cet organisme, il a été choisi pour résoudre les problèmes des clients.

L'objectif de ce mémoire concerne la Conception et la réalisation d'une application mobile pour la gestion des services bancaires.

Pour atteindre ces objectifs, il faut utiliser un ensemble d'outils de conception tel que le processus de développement logiciel 2TUP, le langage de modélisation UML, ainsi que l'utilisation d'outil de développement et de programmation tels que le langage JAVA sous la plateforme ANDROID et le SGBDR MySQL.

Mots clés : Services bancaires, UML, 2TUP, MySQL, JAVA, ANDROID, Mobile.

الملخص

البنك هو من بين أكبر شركة للخدمات التي تقدم خدمات عن طريق إدخال تكنولوجيا الاعلام والاتصالات الجديدة. ونظرا لأهمية الخدمات المصرفية داخل هذه المؤسسة ، تم اختياره لحل مشاكل العملاء.

والهدف من هذه المذكرة التي تتعلق بتصميم وإنتاج تطبيقات الهاتف الذكي لإدارة الخدمات المصرفية

ولتحقيق هذه الأهداف، يجب استخدام مجموعة من أدوات التصميم مثل : لغة النمذجة UML

وإستخدام أدوات التطوير والبرمجة مثل لغة JAVA تحت منصة أندرويد ، و وسيلة التصميم 2TUP

و لتنفيذ قاعدة البيانات نستخدم MySQL .

الكلمات المفتاحية : الخدمات المصرفية ، ANDROID , MySQL, 2TUP, UML, JAVA ,الهاتف

Summary

The bank is among the largest service company, which provides services by introducing new information and communications technology. Given the importance of banking services within the organization, he was chosen to solve customer problems.

The objective of this memory relates to the design and production of a mobile application for the management of banking services.

To achieve these goals, use a set of design tools such as 2TUP software development process, the UML modeling language, and the use of development and programming tools such as the Java language under the ANDROID platform and the MySQL RDBMS.

Keywords: Banking, UML, 2TUP, MySQL, JAVA, ANDROID, Mobile.

Liste des tableaux

Chapitre1 : Présentation de l'organisme d'accueil

Tableau	Description	page
Tableau 1.1	Les avantages et les inconvénients de ses applications	13

Chapitre2 : Etude préliminaire

Tableau	Description	page
Tableau 2.1	Légende des messages du diagramme de contexte dynamique	22

Chapitre3 : Capture des besoins

Tableau	Description	page
Tableau 3.1	Liste préliminaire des cas d'utilisation du système	26
Tableau 3.2	description textuelle des cas d'utilisation	37
Tableau 3.3	Description graphique des cas d'utilisation	47
Tableau 3.4	Liste des classes candidates	48
Tableau 3.5	Responsabilités des classes candidates	51
Tableau 3.6	Diagramme de classe des cas d'utilisation	56
Tableau 3.7	Description textuelle des cas d'utilisation	61

Chapitre4 : Analyse

Tableau	Description	page
Tableau 4.1	Diagramme de classe préliminaire par catégories	64
Tableau 4.2	Diagramme de classe détaillé par catégories	67
Tableau 4.3	Diagramme de séquence pour les cas d'utilisation	77

Chapitre5 : Conception

Tableau	Description	page
Tableau 5.1	Typage des attributs des classes du système	82
Tableau 5.2	Traduction des associations	84
Tableau 5.3	Equivalence entre les concepts objets et relationnels	84

Liste des figures

Chapitre1 : Présentation de l'organisme d'accueil

Figure	Description	page
Figure 1.1	Organigramme de la Banque	9
Figure 1.2	les interfaces de l'Application « Accès Compte »	12
Figure 1.3	les interfaces de l'Application « Banque Zitouna »	12
Figure 1.4	les interfaces de l'Application « Attijari Mobile »	13

Chapitre2 : Etude préliminaire

Figure	Description	page
Figure 2.1	Situation de l'étude préliminaire dans 2TUP	15
Figure 2.2	Le diagramme de contexte dynamique de système	21

Chapitre3 : Capture des besoins

Figure	Description	page
Figure 3.1	la situation de la capture des besoins dans 2TUP	24
Figure 3.2	Diagramme de cas d'utilisation « Client »	27
Figure 3.3	Diagramme de cas d'utilisation « Administrateur »	28
Figure 3.4	Architecture client /serveur du système	57
Figure 3.5	Configuration matérielle du système	58
Figure 3.6	Architecture à deux niveaux (Client / Serveur)	59
Figure 3.7	Modèle préliminaire des composants du système	59
Figure 3.8	Modèle de spécification logicielle du système	60

Chapitre4 : Analyse

Figure	Description	page
Figure 4.1	Situation du découpage en catégorie dans 2TUP	62
Figure 4.2	Découpage en catégorie du système	63
Figure 4.3	Modèle structurel d'analyse	65
Figure 4.4	Diagramme de classe	68

Chapitre5 : Conception

Figure	Description	page
Figure 5.1	Situation de la conception préliminaire dans 2TUP	79
Figure 5.2	Modèle de déploiement du système	80
Figure 5.3	Le modèle relationnel du système	85

Chapitre6 : Réalisation

Figure	Description	page
Figure 6.1	Application MyPuplicWifi	89
Figure 6.2	Emulateur Android	90
Figure 6.3	Interface Super Administrateur/ Administrateur	92
Figure 6.4	Page d'authentification de super administrateur	92
Figure 6.5	Page d'accueil de super administrateur	93
Figure 6.6	Page ajouter banque	93
Figure 6.7	Page supprimer banque	94
Figure 6.8	Page modifier banque	94
Figure 6.9	Interface d'accueil	95
Figure 6.10	Menu principal (page pour l'utilisateur de l'application)	96

Figure 6.11	Annuaire agences	96
Figure 6.12	Convertir devises	97
Figure 6.13	cours devise	98
Figure 6.14	Localiser banque	98
Figure 6.15	Interface d'authentification	99
Figure 6.16	Authentification non validé	99
Figure 6.17	Page réservé aux abonnés	100
Figure 6.18	Page consulter compte	100
Figure 6.19	Page commande chèque	101
Figure 6.20	Page afficher mouvements	101
Figure 6.21	Virements bancaires	102
Figure 6.22	Page désactive compte	102
Figure 6.23	Page à propos	103

Liste des acronymes

Acronyme	Description
PME	Petites et Moyennes Entreprises
PMI	Project Management Institute
SARL	Société à Responsabilité Limitée
EURL	Entreprise Unipersonnelle à Responsabilité Limitée
GPS	Global Positioning System
2TUP	2 Track Unified Process
UML	Unified Modeling Language
BDD	Base De Données
SGBD	Système de Gestion de Base de Données
TCP/IP	Transmission Control Protocol/ Internet Protocol
JVM	Java virtual machine
IOS	Internetworking Operating System
SMT	Simultaneous Multi Threading
SWT	Standard Widget Toolkit
IBM	International Business Machines
IDE	Integrated Development Environment
API	Interface de Programmation Applicative
URL	Uniform Resource Locator
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
SQL	Structured Query Language
WIFI	Wireless Fidelity

SOMMAIRE

Introduction générale	1
------------------------------------	---

Chapitre 1 : Présentation de l'organisme d'accueil

1. Introduction	3
2. Historique	3
2.1 La période de l'indépendance à 1978	3
2.2 La période de 1978 à 1986	4
2.3 La période de 1986 à 1990	5
2.4 La période de 1990 à ce jour	5
3. Présentation de la banque	5
3.1 Définition d'une banque	5
3.2 Les caractéristiques d'une banque	6
3.2.1 La gestion de l'argent des clients	6
3.2.2 Individu / société / entreprise	6
3.2.3 Acceptation des dépôts	6
3.2.4 L'accord de crédits	6
3.2.5 Facilité des opérations de paiement	6
3.2.6 Une tendance continue vers la perfection des services	6
3.2.7 Fonctions croissantes	7
3.2.8 Un lien entre les différents acteurs économiques	7
3.2.9 Finance des entreprises	7
3.2.10 Identité du nom	7
4. Les fonctions de base des banques	7
5. Présentation structurelle et fonctionnelle de la banque	8
5.1 La direction de l'agence	10
5.2 Back office	10
5.3 Front office	10
5.4 Pôle clientèle	10
5.5 Pôle transactions	11

6. Présentation de la structure concernée	11
6.1 Back office.....	11
6.2 Les taches assurées par le back office	11
7. Les Applications mobiles bancaires	11
7.1 Accès Compte	11
7.2 Banque Zitouna.....	12
7.3 Attijari Mobile	12
8. Conclusion	14

Chapitre 2 : Etude préliminaire

1. Introduction	15
2. Situation de l'étude préliminaire dans 2TUP	15
3. Elaboration du cahier des charges	16
3.1 Présentation du sujet	16
3.1.1 Problématique.....	16
3.1.2 Objectifs	17
3.1.3 Cadre de projet	17
3.2 Grands choix techniques	17
3.3 Spécification des besoins fonctionnels	18
3.4 Spécification des besoins non fonctionnels	18
4. Description du contexte du système à réaliser	19
4.1 Identification des acteurs du système	19
4.2 Identification des messages du système.....	20
4.2.1 Les messages émis par le système.....	20
4.2.2 Les messages reçus par le système.....	20
4.3 Le diagramme de contexte du système	21
5. Conclusion	23

Chapitre 3 : Capture des besoins

1. Introduction	24
2. Situation de la capture des besoins dans 2TUP	24

3. Capture des besoins fonctionnels	25
3.1 Identification des cas d'utilisation	25
3.1.1 Qu'est ce qu'un cas d'utilisation.....	25
3.1.2 Liste préliminaire des cas d'utilisation du système.....	25
3.1.3 Diagramme de cas d'utilisation.....	27
3.2 Description des cas d'utilisations	29
3.2.1 Description textuelle des cas d'utilisation.....	29
3.2.2 Description graphique des cas d'utilisation	37
3.3 Identification des classes candidates.....	48
3.3.1 La liste des classes candidates.....	48
3.3.2 Responsabilités des classes	48
3.4 Diagramme de classe participantes	52
4. Capture des besoins techniques	57
4.1 Spécification technique du point de vue matériel	57
4.1.1 Style d'architecture en niveaux	57
4.1.2 Configuration matérielle du système.....	58
4.2 Spécification d'architecture	58
4.2.1 Le style d'architecture en tiers	58
4.2.2 Modèle préliminaire des composants du système	59
4.3 Elaboration du modèle de spécification logicielle	59
4.3.1 Identification des cas d'utilisation techniques	60
4.3.2 Modèle de spécification logicielle.....	60
4.3.3 Description textuelle des cas d'utilisation techniques.....	61
5. Conclusion	61

Chapitre 4 : Analyse

1. Introduction	62
2. Situation de l'analyse dans 2TUP	62
3. Découpage en catégorie	63
3.1 Définition de catégorie.....	63
3.2 Elaboration des diagrammes de classes préliminaire par catégories	64
3.3 Modèle structurel	65
4. Développement du modèle statique	65

4.1 Affiner les classes	65
4.2 Affiner les associations	65
4.3 Ajouter les attributs	66
4.4 Ajouter les opérations	66
4.5 Diagramme de classes détaillées par catégorie	66
4.6 Diagramme de classes	68
5. Développement du modèle dynamique	69
6. Conclusion	78

Chapitre 5 : Conception

1. Introduction	79
2. Conception préliminaire	79
2.1 Situation de la conception préliminaire dans le processus 2TUP	79
2.2 Développement du modèle de déploiement	80
3. Conception détaillée	81
3.1 Typage des attributs	81
4. Règles de transformation	82
4.1 Traduction des associations du système	83
4.2 Passage du modèle objet au modèle relationnel	84
5. Le modèle relationnel	85
6. Conclusion	86

Chapitre 6 : Réalisation

1. Introduction	87
2. Présentation des technologies utilisée	87
2.1 Développement d'application pour Android	87
2.1.1 Langage de programmation java	87
2.1.2 Le SDK Android	87
2.1.3 JDK	87
2.1.4 L'environnement de développement Eclipse	88
2.1.5 ADT pour Eclipse (Android Development Tools)	88

2.1.6 API Google.....	88
2.1.7 MyPuplicWifi.....	88
2.1.8 Emulateur	89
2.2 Développement d'application web	90
2.2.1 WampServer.....	90
2.2.2 MySQL.....	90
2.2.3 PHP.....	90
2.2.4 Feuilles de style (CSS)	91
2.2.5 JavaScript	91
3. Interfaces de l'application	91
3.1 Interface serveur	92
3.1.1 Interface Super Administrateur/Administrateur.....	92
3.1.2 Espace super administrateur.....	92
3.1.3 Ajouter banque	93
3.1.4 Supprimer banque	94
3.1.5 Modifier banque	94
3.2 Interface client	95
4. Conclusion	103
Conclusion générale	104
Références bibliographiques.....	105

Introduction générale

L'omniprésence et la commodité des téléphones mobiles apportent de nouveaux changements dans la société qui ne sont pas pensé plus tôt. Grace à leurs rôles primitifs comme d'outils de communication, ils ont progressé pour devenir une extension de la personnalité de l'utilisateur. La prospérité de cette technologie s'explique par la variété des applications.

Actuellement, les nouvelles technologies de l'information et de la communication évoluent de manière exponentielle. Les systèmes d'information performants existent et leur usage est de plus en plus aisé dans la mesure où ils permettent l'intégration des données provenant de source différentes. Cependant ces technologies ne sont pas toujours mises en œuvre partout où besoin en est, du fait soit de l'inaccessibilité pour certains et pour d'autres du fait de la méconnaissance de leur existence.

De nos jours, les banques comme toutes les institutions économiques cherchent à améliorer ses services. Grâce à l'émergence de technologie de communication (Smartphone, tablettes...) les banques peuvent offrir des services à distance via les technologies de communications. Un client d'une banque se connecte à leurs comptes pour le consulter, effectuer des virements ou afficher les mouvements par l'intermédiaire de leur téléphone mobile. Ces banques permettent de couvrir un grand nombre de ces services et au même temps entrent en relation avec leur clientèle et informer ses clients de nouveaux services.

A cause de l'augmentation des demandes sur les services bancaires, les clients souffre de plusieurs problèmes pour satisfaire leurs demandes, en même temps les employés de la banque aussi rencontrent des obstacles pour effectuer leur travail bien comme il faut.

Pour remédier à ces problèmes nous allons essayer de développer une application mobile destinée pour les banques afin de constituer une interface entre le client et sa banque et offrir un ensemble de services pour la clientèle dont le but est d'atteindre les objectifs suivants :

- Faciliter les opérations de consultation (compte, annuaire, cours devise) ;
- Faciliter les virements bancaires;
- Faciliter l'opération de conversion devise.

Pour atteindre ces objectifs, et bâtir un produit cohérent et évolutif à ses besoins, nous avons choisi UML (Unified Modeling language) pour la modélisation, avec la démarche 2TUP comme processus interactif pas-à-pas pour la construction des modèles.

UML est un langage graphique utilisé pour représenter, spécifier, construire et documenter les systèmes d'information à dominance logicielle. Il permet d'écrire avec un langage standardisé les plans d'élaboration et de construction de logiciels.

Nous avons également choisi le Java sous la plateforme ANDROID pour développement de l'application, et le MySQL comme SGBDR pour la gestion de la base de données.

Le présent mémoire est organisé en six chapitres dont nous donnons une brève description dans les lignes suivantes :

Chapitre 1 : il est consacré à la présentation des notions générales que nous allons utiliser pour le développement de notre système. Puis une présentation de l'organisme d'accueil qui englobe une description générale de l'entreprise ainsi que le champ d'étude.

Chapitre 2 : présente le sujet de notre étude, et introduit la modélisation de son contexte.

Chapitre 3 : il est consacré à la capture des besoins qui comportent deux étapes : la capture des besoins fonctionnels et celles des besoins techniques.

Chapitre 4 : définit l'analyse de notre système en le découpant en catégories jusqu'à l'arrivée au modèle dynamique.

Chapitre 5 : il définit la conception qui décrit le passage du modèle à objet au modèle relationnel, ce passage est justifié par le SGBD à utiliser.

Chapitre 6 : c'est l'étape de réalisation du système qui présente l'environnement de développement, avec quelques interfaces graphiques de l'application.

Enfin, ce document se termine par une conclusion générale et une bibliographie qui comprend les références des ouvrages et les ressources Web relative à notre travail.

Chapitre 01

Présentation de l'organisme d'accueil

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Historique**
- 3. Présentation de la banque**
- 4. Les fonctions de base des banques**
- 5. Présentation structurelle et fonctionnelle de la banque**
- 6. Présentation de la structure concernée**
- 7. Les Applications mobiles bancaires**
- 8. Conclusion**

1. Introduction

Le développement économique d'un pays dépend principalement de son système bancaire où le secteur, dans le domaine financier, est considéré comme une pièce importante de l'entreprise moderne.

2. Historique

Dans toute économie, la banque joue un rôle important. Elle est l'une des premières ressources de financement de l'activité économique, l'Algérie a mis en place depuis l'indépendance jusqu'à nos jours, diverses réglementations en matière bancaire.

2.1 La période de l'indépendance à 1978

L'Algérie a mis en place dès le 29 août 1962, un Trésor public qui prend en charge les activités traditionnelles de la fonction, malgré la nationalisation des banques, nous tenons à signaler, que la fonction exceptionnelle du « crédit à l'économie » s'est développée dans le temps. [1]

La Banque Centrale d'Algérie ayant été instituée le 12 décembre 1962, fut dotée de tous les statuts d'un institut d'émission afin de créer les conditions favorables à un développement ordonné de l'économie nationale. La Banque Centrale exerce les fonctions d'émission de la monnaie fiduciaire, de direction et de surveillance du crédit, ainsi que la gestion des réserves de change. [1]

Le 07 mai 1963, on assiste à la création de la Caisse Algérienne de Développement « qui apparaît comme une direction du plan par son rôle dans l'établissement des programmes d'investissements publics ou d'importation, à la direction du Trésor par son rôle de gestion du budget et de la contre valeur des aides étrangères, à une banque d'affaires par la participation qu'elle est habilitée à prendre à un établissement de crédit à court, moyen et long terme à une banque de commerce extérieur et une caisse des marchés de l'État ». [1]

La Caisse Nationale d'Épargne et de Prévoyance (C.N.E.P) créée le 10 août 1964, avait pour rôle la collecte de l'épargne destinée à la construction du logement. Son activité sera orientée par la suite vers le financement des programmes planifiés d'habitat collectif. [1]

En raison du peu d'empressement des banques étrangères à s'impliquer dans le financement du développement, préférant les opérations de commerce extérieur qui procurent une rentabilité immédiate, la Banque Centrale d'Algérie été contrainte d'agir directement dans le financement de l'économie, en attendant la nationalisation de ces banques, ce qui a donné naissance à trois banques commerciales dénommées primaires. [1]

En 1966, ce fut la création de la **Banque Nationale d'Algérie (B.N.A)** qui disposait d'un monopole légal en matière de traitement des opérations bancaires du secteur public, agricole, industriel et commercial. Le 29 décembre de la même année, fut créé le **Crédit Populaire d'Algérie (C.P.A)**, pour le financement de l'artisanat, de l'hôtellerie et les professions libérales. [1]

2.2 La période de 1978 à 1986

En 1978, le système bancaire algérien affecte le financement des investissements dits « stratégiques » au Trésor public, sous forme de concours remboursables à long terme. Le crédit bancaire à moyen terme est supprimé du système de financement des investissements planifiés, à l'exception de certaines activités (transport et services). [1]

La loi de finance de 1982 fait introduire une nouvelle doctrine. L'intervention des banques primaires dans le financement des investissements publics devrait obéir aux critères de rentabilité financière. [1]

En ce qui concerne le secteur privé, le secteur bancaire intervient rarement comme pourvoyeur de crédits d'investissement. Ce secteur à tendance à s'autofinance. [1]

Cette période s'est caractérisée par la création de deux banques primaires : La première, étant la **Banque de l'Agriculture et du Développement Rural (B.A.D.R)**, elle fut créée en 1982. La seconde banque créée en 1985, est issue du **Crédit Populaire d'Algérie (C.P.A)**, il s'agit de la **Banque de Développement Local (B.D.L)**. Elle a pour rôle le financement des unités économiques locales. [1]

2.3 La période de 1986 à 1990

Il est important de relever, que la loi bancaire de 1986 relative au régime des banques et du crédit, n'a pas été mise en œuvre et ce, à cause de l'accélération du processus de réforme

fondé sur les critères de rentabilité financière, car « le système de financement qui a prévalu avant cette loi a connu des phénomènes corrélatifs tels que la faiblesse, voire l'inexistence de marchés monétaire et financier, un faible degré de bancarisation de l'économie, la domiciliation obligatoire unique des entreprises publiques auprès des banques primaires, et enfin l'absence d'une véritable politique d'encadrement du crédit ».[1]

2.4 La période de 1990 à ce jour

Aujourd'hui, le secteur bancaire est engagé dans une mutation qui devrait se traduire une bancarisation plus importante et par des opérations plus rapides. La mondialisation peut s'accélérer par la mise en œuvre de partenariats avec les institutions bancaires et financières internationales. Le plan de relance de l'économie et de privatisation offre en outre d'autres opportunités, notamment dans l'ingénierie financière, le montage financier des grands projets et le développement de financement de type leasing ou capital-risque, je précise que la banque d'Algérie a récemment publié des textes qui garantissent les droits des investisseurs étrangers en Algérie, comme la liberté de transfert de produits en cas de désinvestissement. [2]

Le secteur public dispose aujourd'hui d'un réseau important de la connaissance des métiers classiques de banque, d'un personnel formé. Les banques étrangères souhaiteraient investir en Algérie et apporter une technologie moderne pourrait sans doute trouver des opportunités de partenariat avec les banques publiques algériennes. [2]

3. Présentation de la banque

3.1 Définition d'une banque

Une banque est une entreprise spécialisée dans le commerce de l'argent, elle joue, d'une part, le rôle de l'intermédiaire financier entre les déposants de l'argent et les demandeurs de prêts, elle reçoit de l'argent de gens qui veulent le réserver sous forme de dépôts, et les prêtent à ceux qui en ont besoin. Elle propose, d'une autre part, une multitude de services financiers comme la gestion quotidienne des comptes bancaires et des moyens de paiement de ses clients. Elle intervient sur les marchés financiers en investissant dans les bourses. [3]

3.2 Les caractéristiques d'une banque

3.2.1 La gestion de l'argent des clients

Une banque est une institution financière qui se charge de la gestion de l'argent des clients et le traitement de leurs activités commerciales en les facilitant. Et pour attirer le plus grand nombre des clients, les gens proposent des services annexes concurrentiels en incitant les gens à déposer l'argent chez eux. [3]

3.2.2 Individu / société / entreprise

Le rôle de la banque ne se limite pas à des institutions bien particulières, une banque peut être une personne, une entreprise ou une société. Une société bancaire désigne une société qui est dans les affaires liés aux banques. [3]

3.2.3 Acceptation des dépôts

Une banque collecte et réserve l'argent du gens sous forme de dépôts qui sont généralement remboursables sur demande ou après l'expiration d'un délai fixe. Le dépôt de l'argent dans ces institutions financières donne plus de sécurité aux clients. [3]

3.2.4 L'accord de crédits

Une banque prête de l'argent à ceux qui en besoin sous forme de crédits bancaires contre des intérêts et des garanties. C'est l'activité bancaire la plus rentable car elle représente plus des deux tiers des recettes des banques. [3]

3.2.5 Facilité des opérations de paiement

Les services sur lequel se basent les banques pour attirer le plus grand nombre de clients, sont les divers moyen de paiement qu'elles proposent à leurs client : les chèques bancaires, les virements bancaires, les guichets automatiques...etc. [3]

3.2.6 Une tendance continue vers la perfection des services

Une banque est une institution qui recherche jour après l'autre à perfectionner les services qu'elle présente. [3]

3.2.7 Fonctions croissantes

Le secteur bancaire est un concept évolutif. Il y'a une expansion continue et une diversification en ce qui concerne les fonctions, les services et les activités d'une banque. [3]

3.2.8 Un lien entre les différents acteurs économiques

Une banque agit comme un lien de liaison entre les emprunteurs et les prêteurs d'argent. C'est un point de rencontre de l'offre de l'argent et sa demande. [3]

3.2.9 Finance des entreprises

Les banques jouent aussi à travers les crédits qu'ils accordent le rôle de financer les investissements. [3]

3.2.10 Identité du nom

L'une des caractéristiques les plus spécifiques aux banques, c'est qu'elles doivent toujours ajouter le mot «**banque**» à leur nom afin de permettre aux gens de savoir qu'il s'agit d'une banque qui se spécialise dans le commerce de l'argent. [3]

4. Les fonctions de base des banques

Les banques sont des entreprises qui ont pour profession habituelle la collecte des dépôts et la redistribution sous forme de crédits. Elles jouent le rôle le plus important dans le circuit économique dans l'absence du marché financier :

- la banque a pour fonction principale de collecter des dépôts et de mettre à disposition de crédit. C'est un intermédiaire financier.
- Elle joue une fonction monétaire qui se traduit par la sécurisation des fonds de posés et par la relocation des ressources des agents.
- Les banques font jouer également une fonction de création monétaire par la mise à disposition de crédit auprès des ménages des entreprises. Cette création monétaire de la banque se traduit à son tour, par la mise à disposition de sa clientèle de moyen de paiement qui met en jeu la monnaie scripturale.
- Actuellement, elle joue d'autre rôles tel que : un opérateur de change, un gestionnaire au sien de la bourse.

5. Présentation structurelle et fonctionnelle de la banque

L'organigramme de la banque est approuvé par le ministre des Finances, il se compose des structures suivantes :

- La direction de l'agence ;
- Secrétariat ;
- Front office ;
- Back office ;
- Pôle de transactions.

La figure suivante représente l'organigramme de la banque :

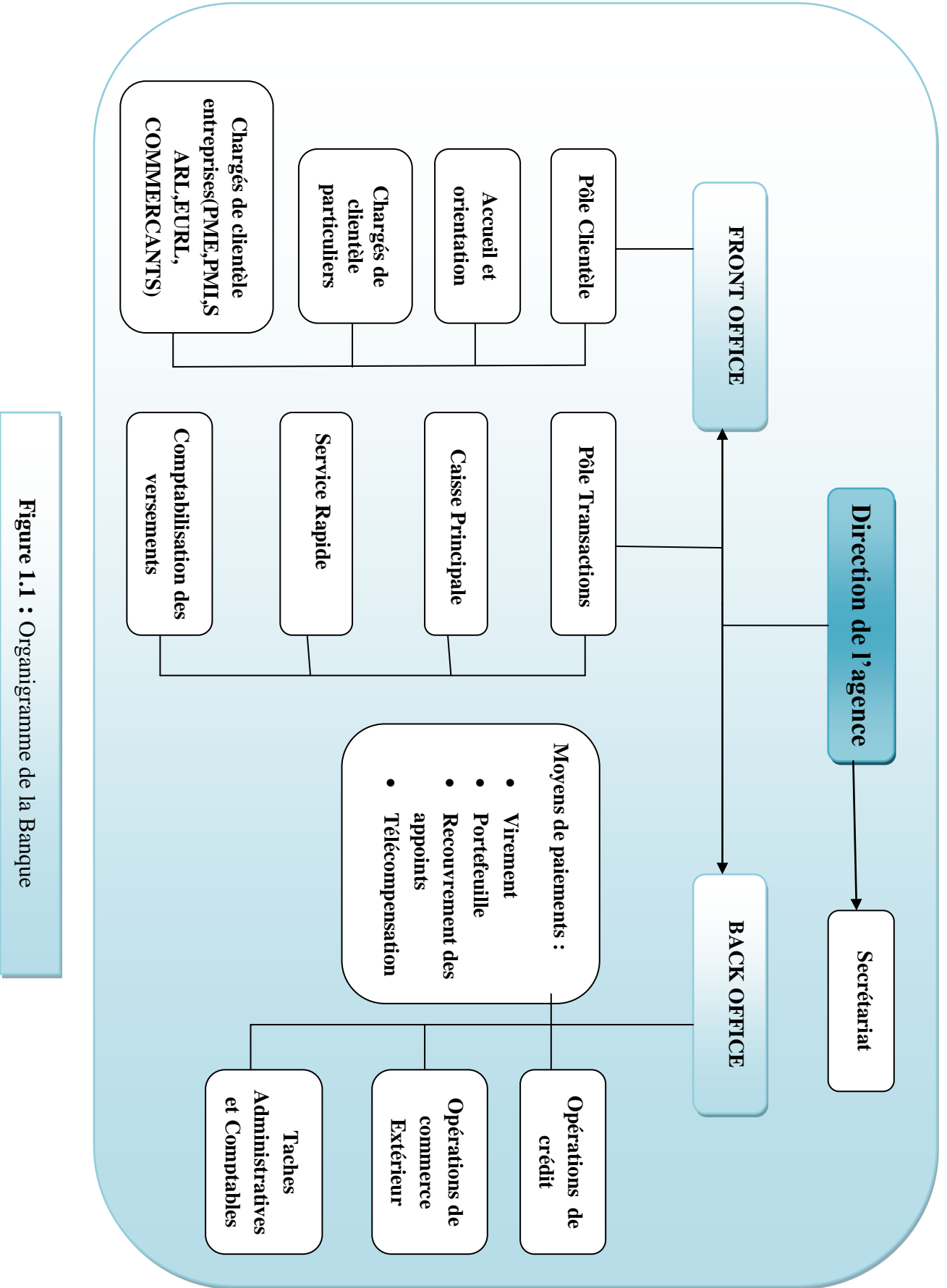


Figure 1.1 : Organigramme de la Banque

5.1 La direction de l'agence

Composée de :

- Directeur général ;
- le secrétaire.

5.2 Back office

Composé de :

- Opérations de crédit ;
- Opérations de commerce extérieur ;
- Taches administratives et comptables ;
- Moyens de paiements (virements, portefeuille, recouvrement des appoints, télécompensation).

5.3 Front office

Composé de :

- Pôle clientèle ;
- Pôle transactions.

5.4 Pôle clientèle

Composé de :

- Accueil et orientation ;
- Chargés de clientèle particuliers ;
- Chargés de clientèle entreprise (PME, PMI, SARL, EURL, COMMERCANTS).

5.5 Pôle transactions

Composé de :

- Caisse principale ;
- Service rapide ;
- Comptabilisation des versements.

6. Présentation de la structure concernée

6.1 Back office

Le service de Back office est relié à la direction de l'Agence.

6.2 Les tâches assurées par le Back office

- Effectuer des opérations de crédit ;
- Effectuer des opérations de commerce extérieur ;
- Effectuer des virements, consulter la liste des virements ;
- Effectuer des tâches administratives et de comptabilité.

7. Les Applications mobiles bancaires

Dans cette partie, nous présentons des cas de figures des applications mobiles bancaires sur l'échelle internationale et nationale. Dans cette dernière, ces applications mobiles bancaires ne sont pas gratuites et ne sont pas appliquées.

7.1 Accès Compte

Une application mobile de la Banque Postale (en France). L'application « Accès Compte » est gratuite à l'utilisation et au téléchargement. Il s'agit d'une application accessible aux clients de la banque postale, leurs fonctions détaillées :

- Consulter les comptes ;
- Effectuer des virements, consulter la liste des virements, ... ;
- Donne la possibilité d'accéder directement au compte de l'utilisateur ;
- Simple, pratique, consultation et suivi les virements depuis un téléphone mobile.

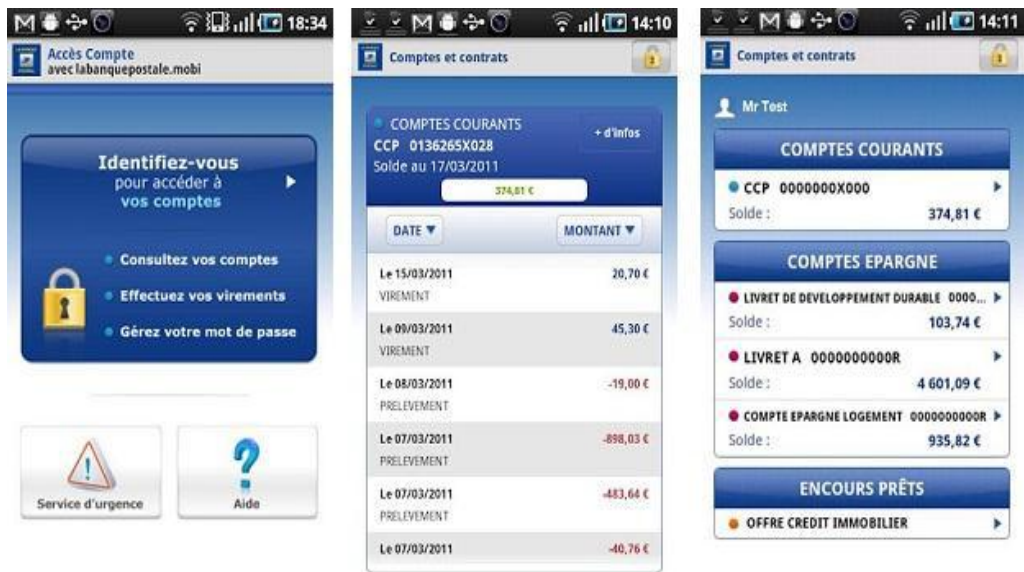


Figure 1.2 : les interfaces de l'Application « Accès Compte »

7.2 Banque Zitouna

Est une banque commerciale tunisienne pour gère les comptes des clients, simulation des crédits, etc.



Figure 1.3 : les interfaces de l'Application « Banque Zitouna »

7.3 Attijari Mobile

Pour faciliter la gestion des comptes, Attijariwafa Bank a créé l'application « Attijari Mobile » qui permet de consulter et de gérer les comptes en toute confidentialité et en toute sécurité, quel que soit le lieu ou l'endroit du client.



Figure 1.4 : les interfaces de l'Application «Attijari Mobile»

Dans le tableau suivant nous présentons quelques avantages et inconvénients de ses applications.

Avantages	Inconvénients
<ul style="list-style-type: none"> • L'utilisation de l'application plus compréhensible et simple pour l'utilisateur novice • Gain de temps : consultation, virement à distance et en ligne • Consultation des comptes des utilisateurs, cours devise... • la majorité des applications sont destinée à une seule banque 	<ul style="list-style-type: none"> • Absence de la position des agences bancaires • la majorité des applications mobiles dans le secteur bancaire n'utilise pas la géo localisation pour localiser ses agences

Tableau 1.1 : Les avantages et les inconvénients de ses applications.

8. Conclusion

Nous avons représenté l'organisme d'accueil et quelques notions introductives pour mieux comprendre et cerner notre thème, qui seront utilisées dans la réalisation de ce projet. Ces notions nous permettent de définir les objectifs globaux, pour pouvoir remédier aux problèmes que nous avons constatés dans les services bancaires.



Chapitre 02

Etude préliminaire

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Situation de l'étude préliminaire dans 2TUP**
- 3. Elaboration du cahier des charges**
- 4. Description du contexte du système**
- 5. Conclusion**

1. Introduction

L'étude préliminaire est la première phase de notre processus de développement. Elle consiste à effectuer un repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples.

Nous commencerons à déterminer les besoins fonctionnels en considérant le système comme une boîte noire, afin d'étudier sa place dans le système. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML au niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

2. Situation de l'étude préliminaire dans 2TUP

L'étude préliminaire est la première étape dans le processus de développement 2TUP. Elle prépare les étapes les plus formelles de la capture des besoins fonctionnels et de la capture des besoins techniques. [4]

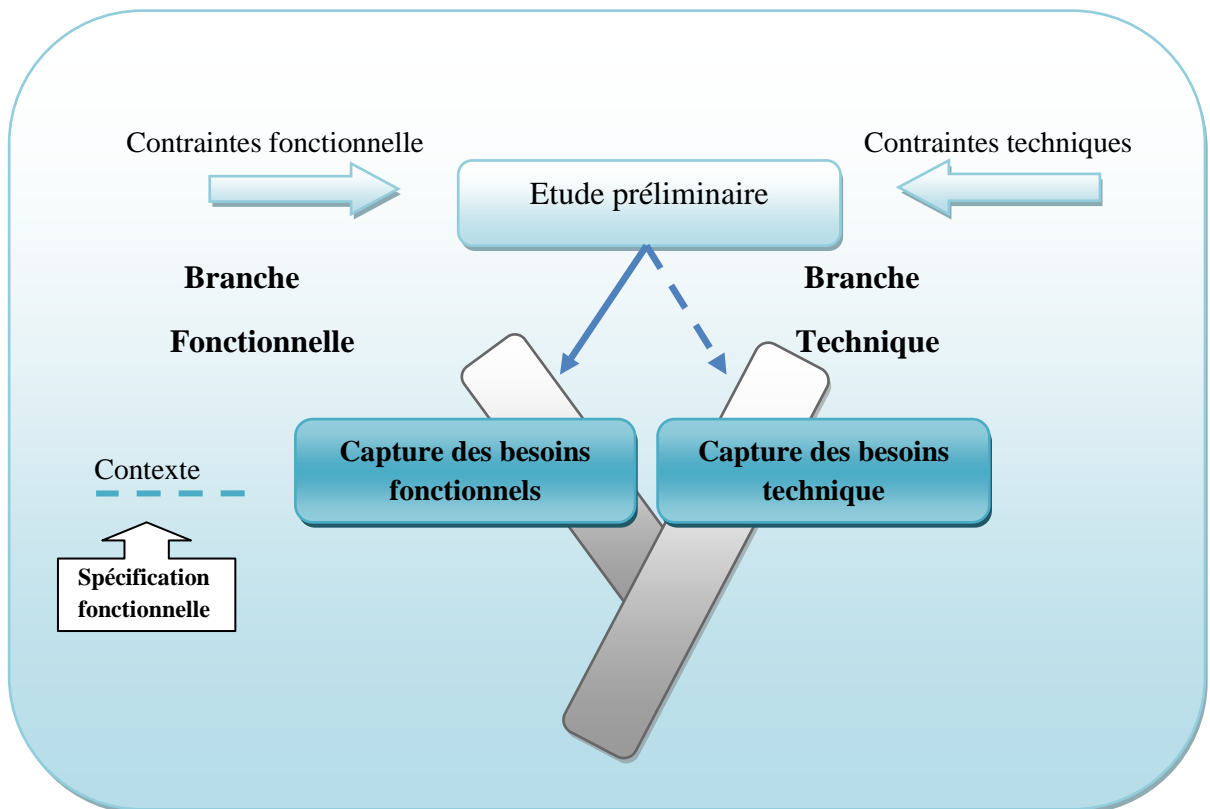


Figure 2.1 : Situation de l'étude préliminaire dans 2TUP

3. Elaboration du cahier des charges**3.1 Présentation du sujet**

Durant notre visite à la banque « BADR » nous avons constaté:

Le Back office est un service attaché à la direction de l'agence. Il est constitué de plusieurs bureaux :

- Bureau opérations de crédit ;
- Bureau opérations de commerce extérieur ;
- Bureau tâches administratives et comptable ;
- Bureau moyens de paiements
 - Virements ;
 - Portefeuille ;
 - Recouvrement des appoints ;
 - Télécompensation.

3.1.1 Problématique

Le secteur bancaire est le plus sensible dans l'économie, vu les difficultés et les obstacles rencontrés du côté de la banque et du côté client. Ainsi la relation entre la banque et sa clientèle n'est pas en bon état parce que :

- Les clients souffrent de la longueur de la période d'attente au niveau des guichets afin d'effectuer des simples opérations comme: consulter son compte, afficher les mouvements de son compte ou effectuer des virements bancaires...
- Les clients sont obligés de respecter les horaires de travail de la banque, d'attendre son tour et à savoir quand et quelle durée doit attendre, sans oublier les problèmes de déplacement des régions rurales.

D'une autre part, l'employé de la banque lui-même a des difficultés d'acquiescer ce grand nombre et il n'aura pas la capacité de satisfaire les besoins de ces clients dans les meilleurs délais.

En effet, avec toutes ces difficultés et ces obstacles ; un canal de services bancaires à distance fonctionnant par téléphonie mobile (une application mobile gratuite et rapide) pourrait être préférable.

3.1.2 Objectifs

L'objectif de notre projet est de réaliser une application mobile pour faciliter l'utilisation des services bancaires qui permettent aux utilisateurs d'accéder à ces services n'importe où et n'importe quand.

Cette application mobile doit répondre aux besoins suivants :

- Faciliter les opérations de consultation (compte, annuaire, cours devise) ;
- Faciliter les virements bancaires;
- Faciliter l'opération de conversion devise ;
- Localisation de la position de l'utilisateur ;
- L'affichage de la position de la banque choisit.

3.1.3 Cadre de projet

Les services financiers mobiles représentent le nouveau produit de la banque à distance. De plus en plus nombreuses sont les applications mobiles qui reposent sur une plate-forme de mobilité en équipant les périphériques mobiles consommateur les plus récents avec la connectivité en temps réel et l'intégration aux systèmes service client des banques. C'est pourquoi la clientèle peut bénéficier des services bancaires en ligne. Ces applicatifs réduisent les coûts et augmentent la satisfaction du client. En effet, le client n'a pas besoin de se déplacer à sa banque et attendre aux guichets pour bénéficier de service, c'est sa banque qui lui rend visite. Et au niveau de la banque abaissent les coûts de transaction et chargent de service en particulier le coût de mise en place et de maintien d'un réseau de distribution pour les banques.

3.2 Grands choix techniques

Afin de maîtriser les risques, nous souhaitons utiliser une approche itérative et incrémentale, fondée sur le processus en Y. Les technologies que nous avons choisies pour ce projet sont :

- le processus de développement en y (2TUP) ;
- le langage de modélisation UML ;
- Système de gestion de bases de données relationnelles(MySQL) ;
- langage de programmation JAVA sous la plateforme ANDROID ;
- Architecture 2 tiers (client/serveur).

3.3 Spécification des besoins fonctionnels

Après une étude de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation.

Les besoins utilisateur :

- Consulter compte ;
- Demande l'affichage de mouvement ;
- Commander chèques ;
- Effectuer virement bancaire ;
- Consulter annuaire agence ;
- Consulter cours devise.

3.4 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

- La disponibilité : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.
- La sécurité de l'accès aux informations critiques : nous devons prendre en considération la confidentialité des données de clients surtout au niveau de l'authentification.
- La fiabilité : les données fournies par l'application doivent être fiables.
- La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.
- Une solution ouverte et évoluée : l'application peut être améliorée par l'ajout d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.

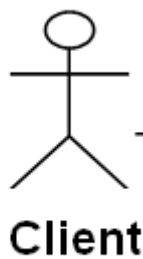
4. Description du contexte du système à réaliser

Cette description se compose de trois activités successives :

- énumérer les acteurs susceptible d'interagir avec le système ;
- Identification des messages entre les acteurs et le système ;
- Modéliser le contexte et faire une description détaillée des messages (diagramme de contexte). [5]

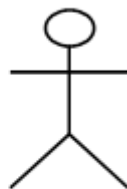
4.1 Identification des acteurs du système

Un acteur est un utilisateur type de système. Il représente une responsabilité par rapport au système ou un rôle plutôt qu'une personne physique. Il est donc une entité externe qui interagit directement avec le système en émettant et en recevant des messages. [5]



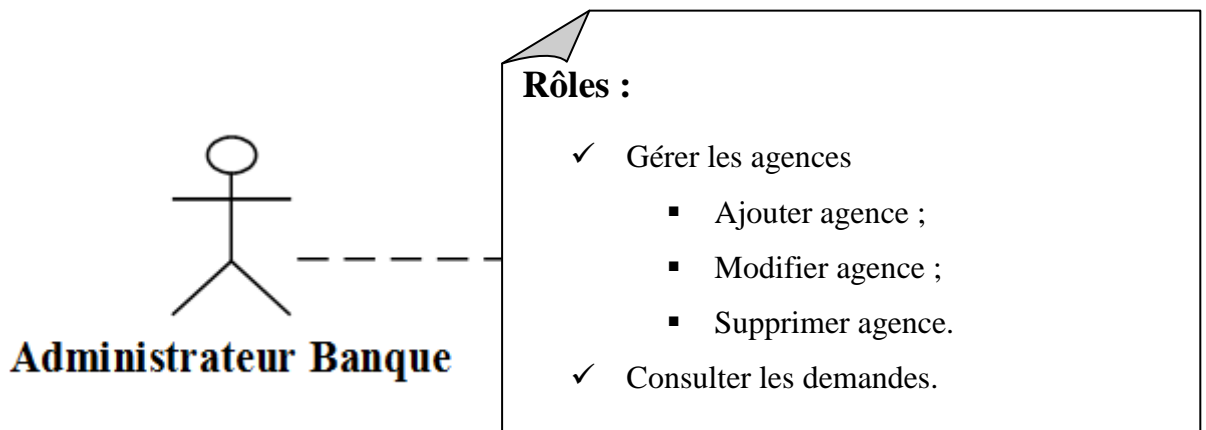
Rôles :

- ✓ Consulter compte ;
- ✓ Effectuer virement bancaire ;
- ✓ Désactiver compte ;
- ✓ Commander chèque ;
- ✓ Afficher les mouvements.



Rôles :

- ✓ Gérer les banques
 - Ajouter banque ;
 - Modifier banque ;
 - Supprimer banque.
- ✓ Gérer les administrateurs des banques
 - Ajouter administrateur banque ;
 - Modifier administrateur banque ;
 - Supprimer administrateur banque.



4.2 Identification des messages du système

Le message est le seul moyen de communication entre les objets pour déclencher une activité chez le récepteur. Cette notion est utilisée également pour décrire les interfaces de plus haut niveau entre les acteurs et le système. [5]

4.2.1 Les messages émis par le système

- Information des comptes ;
- Information des chèques ;
- Information sur devise;
- Information des agences ;
- Information des banques.

4.2.2 Les messages reçus par le système

- Demande consulter comptes ;
- Demande commander chèques ;
- Demande effectuer virement bancaire;
- Information sur les mouvements du compte ;
- Information sur devises ;
- Information sur les agences ;
- Information sur les banques.

4.3 Le diagramme de contexte du système

La modélisation du contexte consiste à représenter de façon synthétique tous les messages échangés entre système et acteurs sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique.

Ce diagramme dynamique est présenté grâce à un diagramme de collaboration UML comme suit :

- Le système étudié est représenté par un objet central ;
- Cet objet central entouré par un autre objet symbolisant les différents acteurs ;
- Des liens relient le système à chaque acteur ;
- Sur chaque lien sont montrés les messages en entrée et en sortie du système. [5]

Le schéma suivant représente le diagramme de contexte de système :

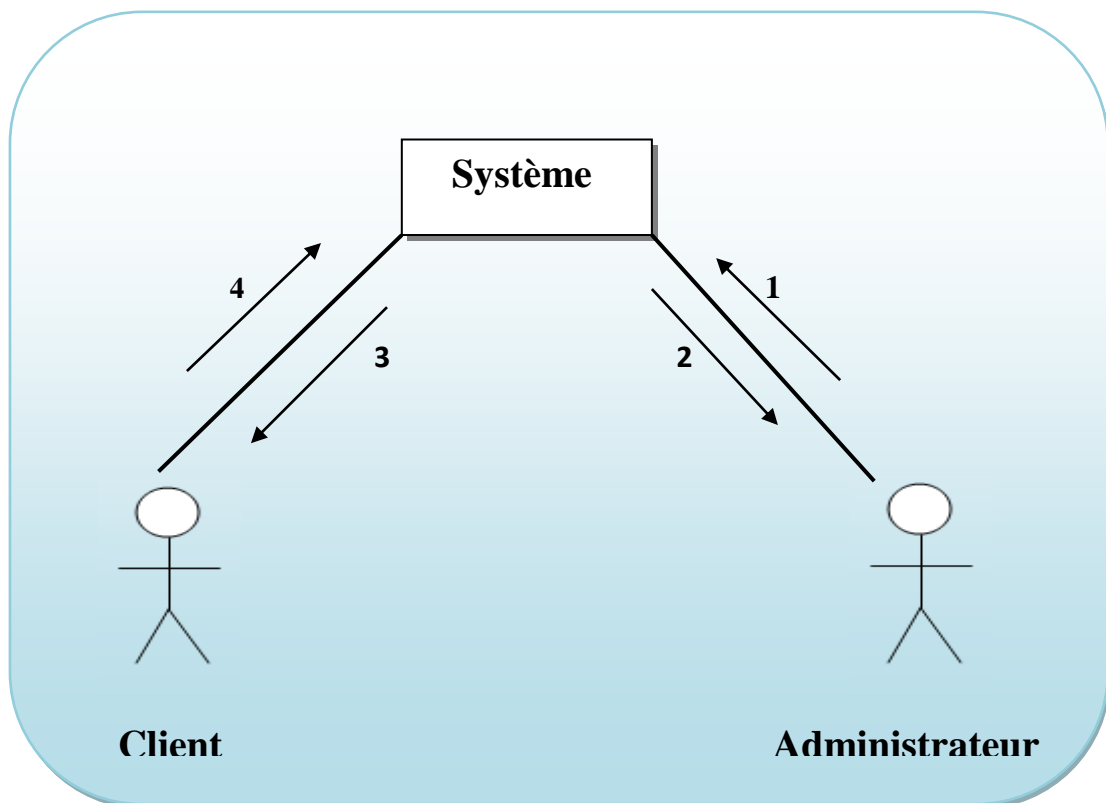


Figure 2.2 : Le diagramme de contexte dynamique de système

Légende des messages du diagramme de contexte dynamique	
1 : Administrateur	<p style="text-align: center;">—————→ système</p> <ul style="list-style-type: none"> • Information sur devises ; • Information sur les agences ; • Information sur les banques.
2 : système	<p style="text-align: center;">—————→ Administrateur</p> <ul style="list-style-type: none"> • Information des agences ; • Information des banques ; • Information sur devise.
3 : système	<p style="text-align: center;">—————→ Client</p> <ul style="list-style-type: none"> • Information des comptes ; • Information sur devise ; • Information des chèques.
4 : Client	<p style="text-align: center;">—————→ système</p> <ul style="list-style-type: none"> • Demande consulter comptes ; • Demande commander chèques ; • Demande effectuer virement bancaire ; • Information sur les mouvements du compte ; • Information sur devises.

Tableau 2.1 : Légende des messages du diagramme de contexte dynamique

5. Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les critères qu'on doit prendre en considération, l'étude préliminaire prépare les étapes plus formelles de capture des besoins fonctionnels et techniques que nous allons décrire au chapitre suivant.



Chapitre 03

Capture des besoins

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Situation de la capture des besoins dans 2TUP**
- 3. Capture des besoins fonctionnels**
- 4. Capture des besoins techniques**
- 5. Conclusion**

1. Introduction

Nous allons traiter le rôle que tient UML pour compléter la capture des besoins ébauchée durant l'étude préliminaire. Les étapes de capture des besoins fonctionnels et de capture des besoins techniques sont faites en parallèle, mais dans ce chapitre on va montrer premièrement la capture des besoins fonctionnels et en suite la capture des besoins technique. 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. [4]

Elle est complétée au niveau de la branche droite du Y par la capture des besoins technique et prépare l'étape suivante de la branche gauche « l'analyse ». [4]

2. Situation de la capture des besoins dans 2TUP

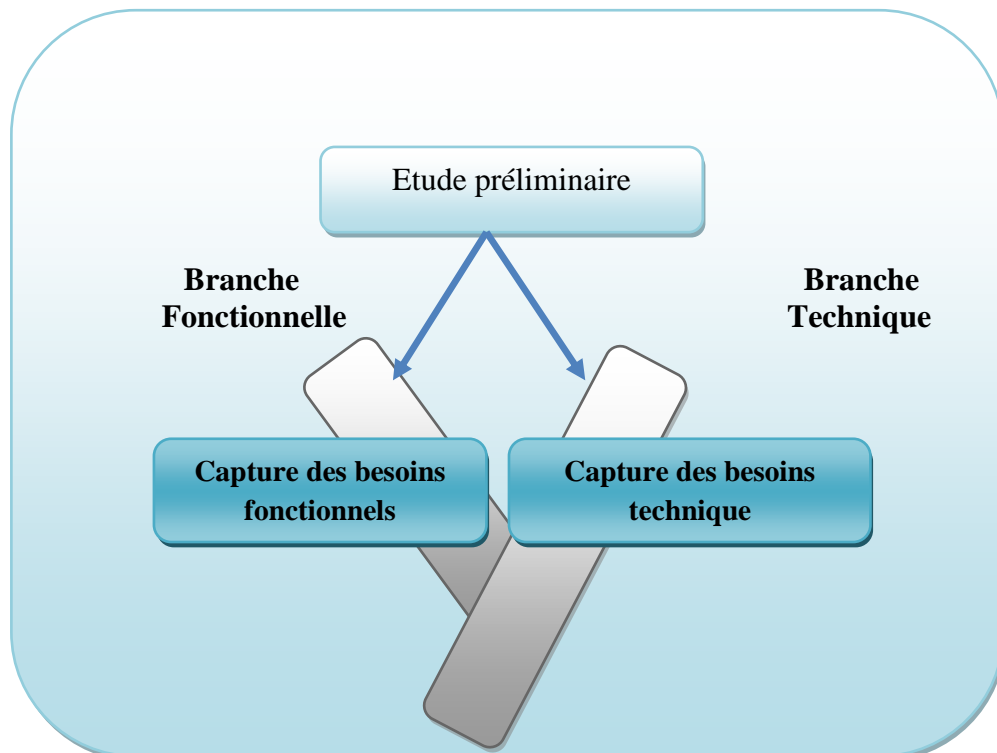


Figure 3.1 : la situation de la capture des besoins dans 2TUP

3. Capture des besoins fonctionnels

Par besoins fonctionnels on désigne l'ensemble des besoins liés au métier et domaine traité par le système.

La technique des cas d'utilisation est la pierre angulaire de cette étape. Elle va nous permettre de préciser l'étude du contexte fonctionnel du système. Elle décrit les différentes façons qu'auront les acteurs d'utiliser le futur système.

Nous allons suivre les étapes suivantes :

- Identifier les cas d'utilisation du système par ces acteurs ;
- Décrire les cas d'utilisation ;
- Elaborer les diagrammes des classes participantes des cas d'utilisation. [6]

3.1 Identification des cas d'utilisation

Il s'agit de la solution UML pour représenter le modèle conceptuel.

3.1.1 Qu'est ce qu'un cas d'utilisation

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. [7]

3.1.2 Liste préliminaire des cas d'utilisation du système

La liste de cas d'utilisation du système est représentée dans le tableau récapitulatif suivant :

Cas d'utilisation		Acteur
Consulter compte		Client
Commander chèques		Client
Désactiver compte		Client
Consulter cours devise		Client
Consulter annuaire agence		Client
Afficher les mouvements		Client
Convertir devise		Client
Effectuer virement bancaires		Client
Localiser banque		Client
Gérer banques	-Ajouter banque - Modifier banque -Supprimer banque	Super Administrateur
Gérer administrateurs	-Ajouter administrateur - Modifier administrateur -Supprimer administrateur	Super Administrateur
Gérer agences	-Ajouter agence - Modifier agence -Supprimer agence	Administrateur banque

Tableau 3.1 : Liste préliminaire des cas d'utilisation du système

3.1.3 Diagramme de cas d'utilisation

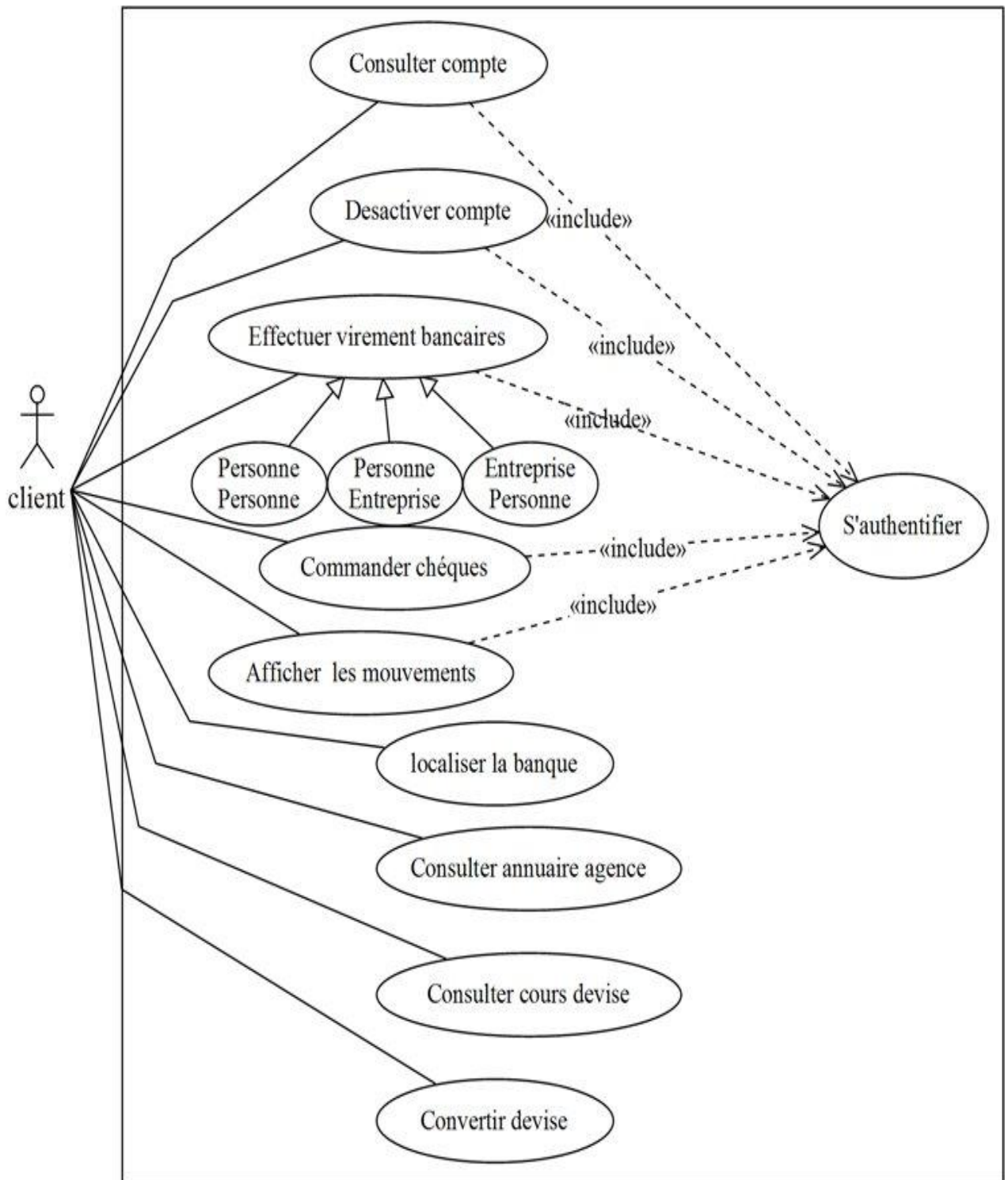


Figure 3.2 : Diagramme de cas d'utilisation « Client »

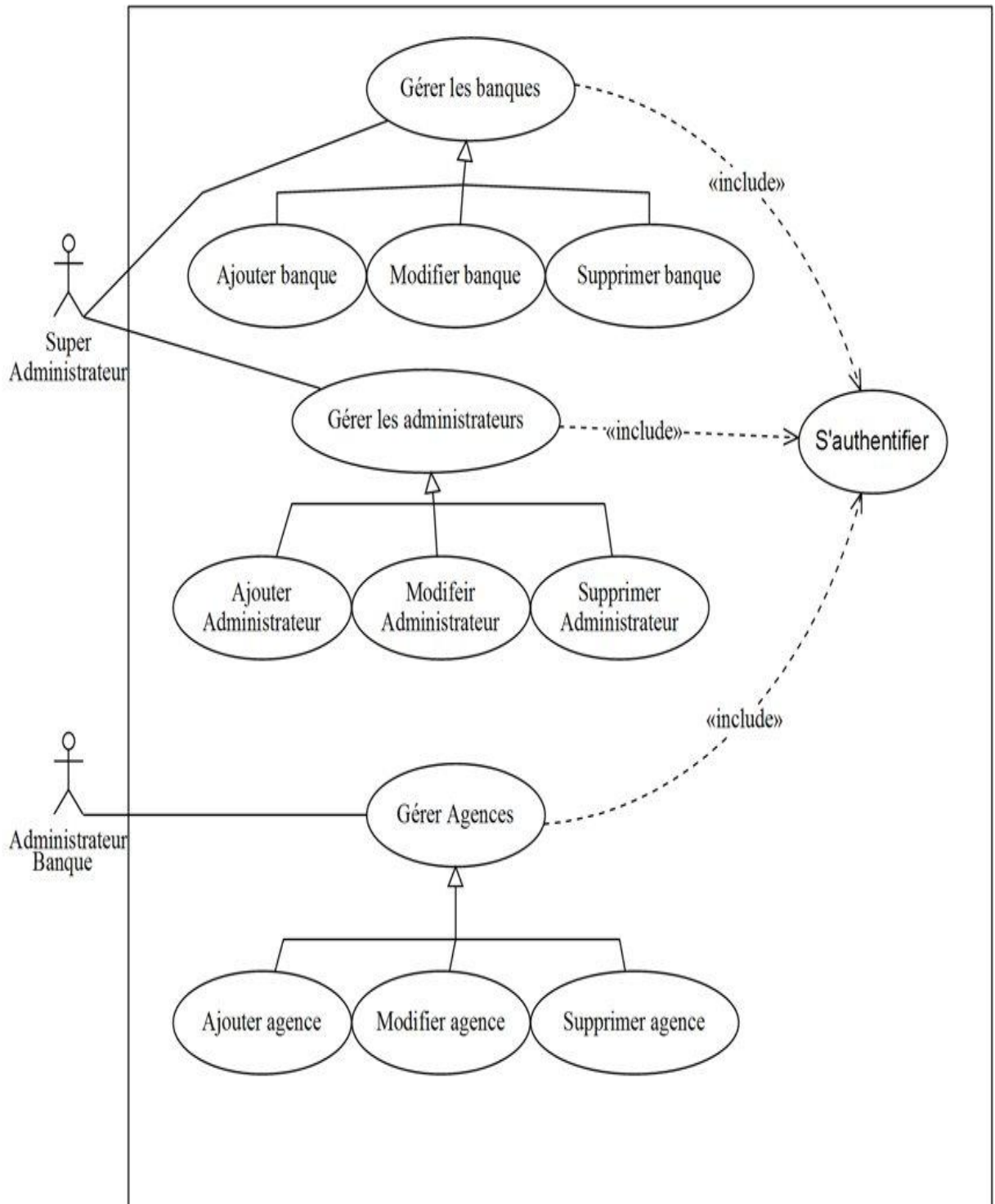


Figure 3.3 : Diagramme de cas d'utilisation « Administrateur »

3.2 Description des cas d'utilisations

3.2.1 Description textuelle des cas d'utilisation

La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée par UML.

Nous utilisons ici la structure proposée par « Pascal Roques ». [6]

Cas d'utilisation	S'authentifier
Acteur	Client.
But	Vérifier l'autorisation d'accès au système.
Pré condition	/
Post condition	Le client est s'authentifier par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. le client demande l'accès au système. 2. le système affiche le formulaire d'authentification. 3. le client saisit l'identifiant et le mot de passe. 4. le système vérifie la validation des informations saisie puis lance le menu principal de l'application.
Scénario alternatif	<ol style="list-style-type: none"> 3.1 L'identifiant et/ou mot de passe incorrect. <ol style="list-style-type: none"> 3.1.1 Le système informe le client que l'identifiant et/ou le mot de passe sont incorrects. 3.1.2 Le système propose au client de renseigner une nouvelle fois l'identifiant et le mot de passe. 3.1.3 Reprise de l'enchaînement du scénario nominal au point 2.

Cas d'utilisation	Consulter compte
Acteur	Client.
But	Permettre au client de consulter les informations d'un compte.
Pré condition	Le client est authentifié.
Post condition	Les informations de compte sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. le client demande de consulter compte. 2. le système demande le numéro de compte. 3. le client saisit le numéro de compte. 4. le système affiche les informations du compte.

Cas d'utilisation	Afficher les mouvements
Acteur	Client.
But	Permettre au client de demander l'affichage de tous les mouvements.
Pré condition	Le client est authentifié.
Post condition	Les informations du compte sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. le client demande d'afficher les mouvements du compte. 2. le système demande le numéro du compte. 3. le client saisit le numéro de compte. 4. le système affiche les informations du compte.

Cas d'utilisation	Effectuer virement bancaires
Acteur	Client.
But	Permettre au client d'effectuer des virements bancaires.
Pré condition	Le client est authentifié.
Post condition	Le virement est effectué.
Scénario nominal	<ol style="list-style-type: none"> 1. le client demande d'effectuer des virements bancaires. 2. le système affiche le formulaire. 3. le client remplit les champs et valide l'opération. 4. le système affiche un message de confirmation.

Cas d'utilisation	Commander chèque
Acteur	Client.
But	Permettre au client de commander le chèque.
Pré condition	Le client est authentifié.
Post condition	La demande est envoyée.
Scénario nominal	<ol style="list-style-type: none"> 1. le client demande de commander le chèque. 2. le système affiche un formulaire permettant de commander le chèque. 3. le client lance la commande de chèque. 4. le système envoyer la demande.

Cas d'utilisation	Désactiver compte
Acteur	Client.
But	Permettre au client de désactiver le compte.
Pré condition	Le client est authentifié.
Post condition	Le compte est désactivé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client demande de désactiver le compte. 2. Le système demande le numéro du compte. 3. Le client saisit le numéro de compte. 4. Le système demande au client s'il veut vraiment désactiver le compte. 5. Le client valide l'opération. 6. Le système affiche un message de succès.

Cas d'utilisation	Consulter cours devise
Acteur	Client.
But	Permettre au client de consulter le cours devise.
Pré condition	/
Post condition	Les informations sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client demande de consulter le cours devise. 2. Le système affiche les informations.

Cas d'utilisation	Consulter Annuaire Agence
Acteur	Client.
But	Permettre au client de consulter annuaire d'agence.
Pré condition	Le client choisit l'agence.
Post condition	Les informations de l'agence sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client demande de consulter l'annuaire d'agence. 2. Le système affiche la liste des agences. 3. Le client choisit l'agence. 4. Le système afficher les informations de l'agence.

Cas d'utilisation	Conversion devise
Acteur	Client.
But	Permettre au client de conversion devise.
Pré condition	Le client saisit le montant.
Post condition	La conversion est effectuée.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client demande la conversion devise. 2. Le système demande le montant et l'unité de conversion. 3. Le client saisit le montant et choisit l'unité. 4. Le système affiche le résultat.

Cas d'utilisation	Localiser la banque
Acteur	Client.
But	Permettre au client de localiser la banque.
Pré condition	Le client choisit la banque.
Post condition	La position est affiché.
Scénario nominal	<ol style="list-style-type: none"> 1. Le client demande de localiser la banque. 2. Le système affiche la liste des banques. 3. Le client choisit la banque. 4. Le système affiche la position de la banque.

Cas d'utilisation	Gérer Banques
Acteur	Super Administrateur.
But	Permettre de modifier, supprimer ou ajouter des banques.
Pré condition	Super Administrateur est authentifié.
Scénario nominal	<p>Le super administrateur choisit une des actions suivantes :</p> <ul style="list-style-type: none"> • Ajouter banque • Modifier banque • Supprimer banque

Cas d'utilisation	Ajouter banque
Acteur	Super Administrateur.
But	Permettre au super administrateur d'ajouter une banque.
Pré condition	Super Administrateur est authentifié.
Post condition	Nouvelle banque ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche un formulaire d'ajout. 2. Super administrateur remplit le formulaire. 3. Le système enregistre les informations de la banque. 4. Le système affiche un message « enregistrée avec succès ».
Scénario d'erreur	<ol style="list-style-type: none"> 2.1 La banque existe déjà. 2.1.2 Le système informe le super administrateur que la banque existe.

Cas d'utilisation	Modifier banque
Acteur	Super Administrateur.
But	Permettre au super administrateur de modifier les informations de la banque.
Pré condition	Super Administrateur est authentifié.
Post condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le super administrateur demande la modification des informations de la banque. 2. Super administrateur remplit le formulaire. 3. Le système contrôle la saisie des informations modifiée puis valider.
Scénario alternatif	<ol style="list-style-type: none"> 3.1 les informations sont incorrectes ou incomplètes. 3.1.1 Le système affiche un message d'erreur. 3.1.2 Reprise de l'enchaînement à partir du point 2.

Cas d'utilisation	Supprimer banque
Acteur	Super Administrateur.
But	Permettre au super administrateur de supprimer une banque.
Pré condition	Super Administrateur est authentifié.
Post condition	La banque est supprimée.

Scénario nominal	<ol style="list-style-type: none"> 1. Super administrateur demande la suppression de la banque. 2. Le système demande au super administrateur s'il veut vraiment supprimer la banque. 3. Super administrateur valide l'opération. 4. Le système affiche un message de confirmation.
-------------------------	---

Cas d'utilisation	Gérer Administrateurs
Acteur	Super Administrateur.
But	Permettre de modifier, supprimer ou ajouter des administrateurs.
Pré condition	Super Administrateur est authentifié.
Scénario nominal	<p>Le super administrateur choisit une des actions suivantes :</p> <ul style="list-style-type: none"> • Ajouter administrateurs • Modifier administrateurs • Supprimer administrateurs

Cas d'utilisation	Ajouter Administrateurs
Acteur	Super Administrateur.
But	Permettre au super administrateur d'ajouter un administrateur.
Pré condition	Super Administrateur est authentifié.
Post condition	Nouveau administrateur ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche un formulaire d'ajout. 2. Super administrateur remplit le formulaire. 3. Le système enregistre les informations de l'administrateur. 4. Le système affiche un message « enregistrée avec succès ».
Scénario d'erreur	<ol style="list-style-type: none"> 2.1 L'administrateur existe déjà. 2.1.2 Le système informe le super administrateur que l'administrateur existe.

Cas d'utilisation	Modifier Administrateurs
Acteur	Super Administrateur.
But	Permettre au super administrateur de modifier les informations de l'administrateur.

Pré condition	Super Administrateur est authentifié.
Post condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Super administrateur demande la modification des informations de l'administrateur. 2. Super administrateur remplit le formulaire. 3. Le système contrôle la saisie des informations modifiée puis valider.
Scénario alternatif	<ol style="list-style-type: none"> 3.1 Les informations sont incorrectes ou incomplètes. <ol style="list-style-type: none"> 3.1.1 Le système affiche un message d'erreur. 3.1.2 Reprise de l'enchaînement à partir du point 2.

Cas d'utilisation	Supprimer Administrateurs
Acteur	Super Administrateur.
But	Permettre au super administrateur de supprimer un administrateur.
Pré condition	Super Administrateur est authentifié.
Post condition	L'administrateur est supprimé.
Scénario nominal	<ol style="list-style-type: none"> 1. Super administrateur demande la suppression de l'administrateur. 2. Le système demande au super administrateur s'il veut vraiment supprimer cet administrateur. 3. Super administrateur valide l'opération. 4. Le système affiche un message de confirmation.

Cas d'utilisation	Gérer Agences
Acteur	Administrateur.
But	Permettre de modifier, supprimer ou ajouter des agences.
Pré condition	L'Administrateur est authentifié.
Scénario nominal	<p>L'administrateur choisit une des actions suivantes :</p> <ul style="list-style-type: none"> • Ajouter agence • Modifier agence • Supprimer agence

Cas d'utilisation	Ajouter Agences
Acteur	Administrateur.
But	Permettre à l'administrateur d'ajouter une agence.
Pré condition	L'Administrateur est authentifié.
Post condition	Nouvelle agence ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche un formulaire d'ajout. 2. L'administrateur remplit le formulaire. 3. Le système enregistre les informations de l'agence. 4. Le système affiche un message « enregistrée avec succès ».
Scénario d'erreur	<ol style="list-style-type: none"> 2.1 L'agence existe déjà. 2.1.2 Le système informe l'administrateur que l'agence existe.

Cas d'utilisation	Modifier Agences
Acteur	Administrateur.
But	Permettre à l'administrateur de modifier les informations de l'agence.
Pré condition	L'Administrateur est authentifié.
Post condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'Administrateur demande la modification des informations de l'agence. 2. L'administrateur remplit le formulaire. 3. Le système contrôle la saisie des informations modifiée puis valider.
Scénario alternatif	<ol style="list-style-type: none"> 3.1 Les informations sont incorrectes ou incomplètes. <ol style="list-style-type: none"> 3.1.1 Le système affiche un message d'erreur. 3.1.2 Reprise de l'enchaînement à partir du point 2.

Cas d'utilisation	Supprimer Agences
Acteur	Administrateur.
But	Permettre à l'administrateur de supprimer une agence.
Pré condition	L'Administrateur est authentifié.
Post condition	L'agence est supprimée.

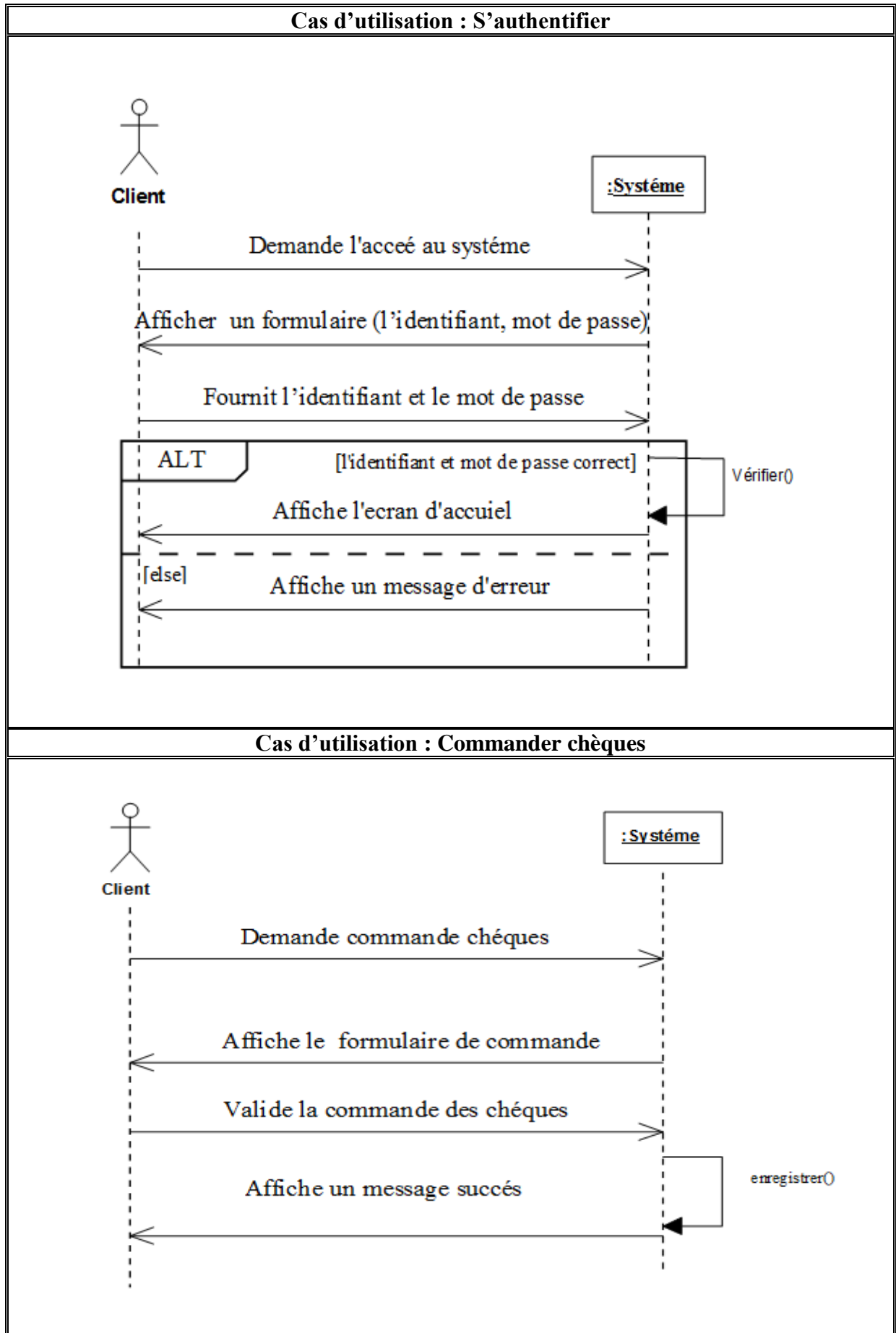
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur demande la suppression de l'agence.2. Le système demande à l'administrateur s'il veut vraiment supprimer l'agence.3. L'administrateur valide l'opération.4. Le système affiche un message de confirmation.
-------------------------	---

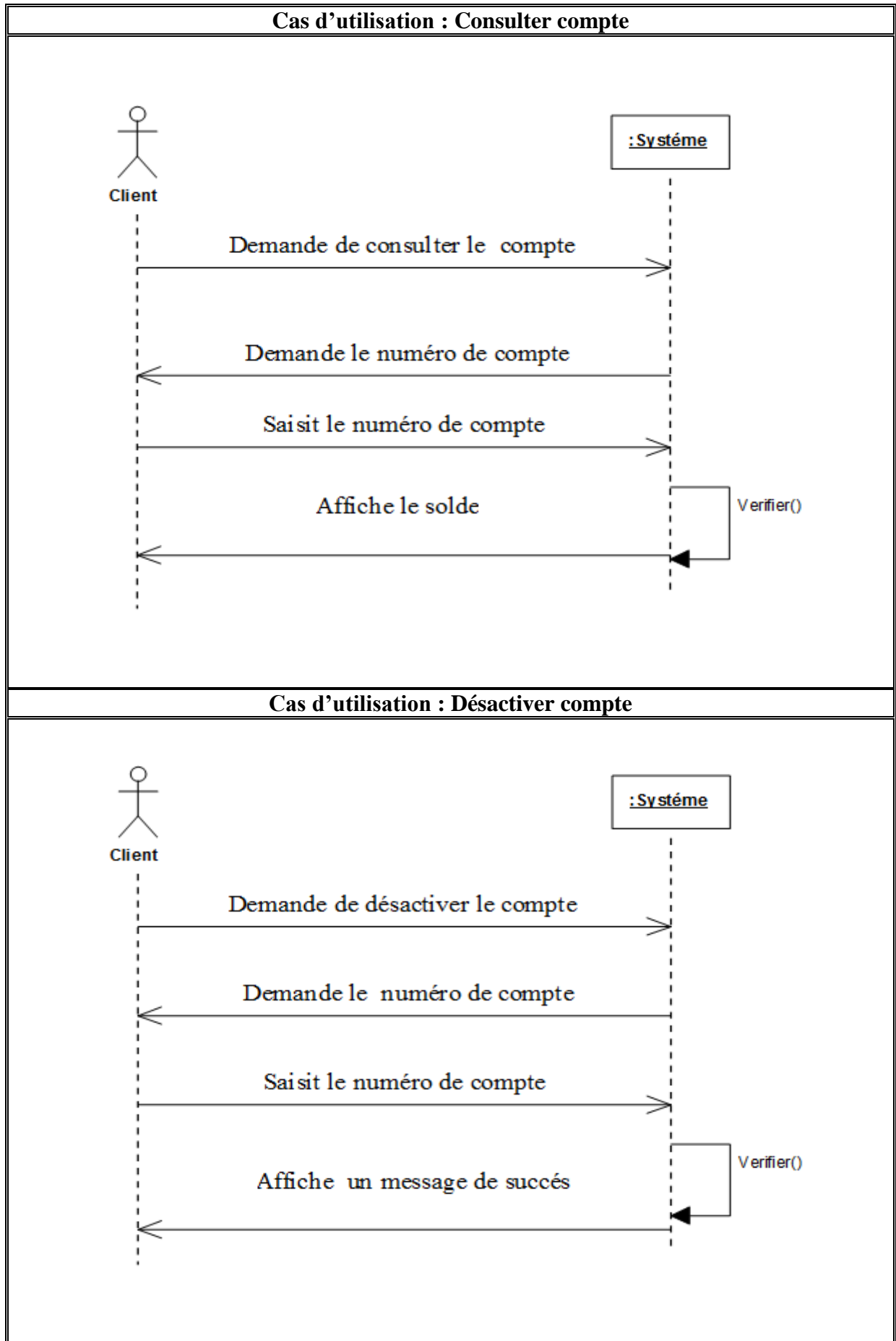
Tableau 3.2 : description textuelle des cas d'utilisation

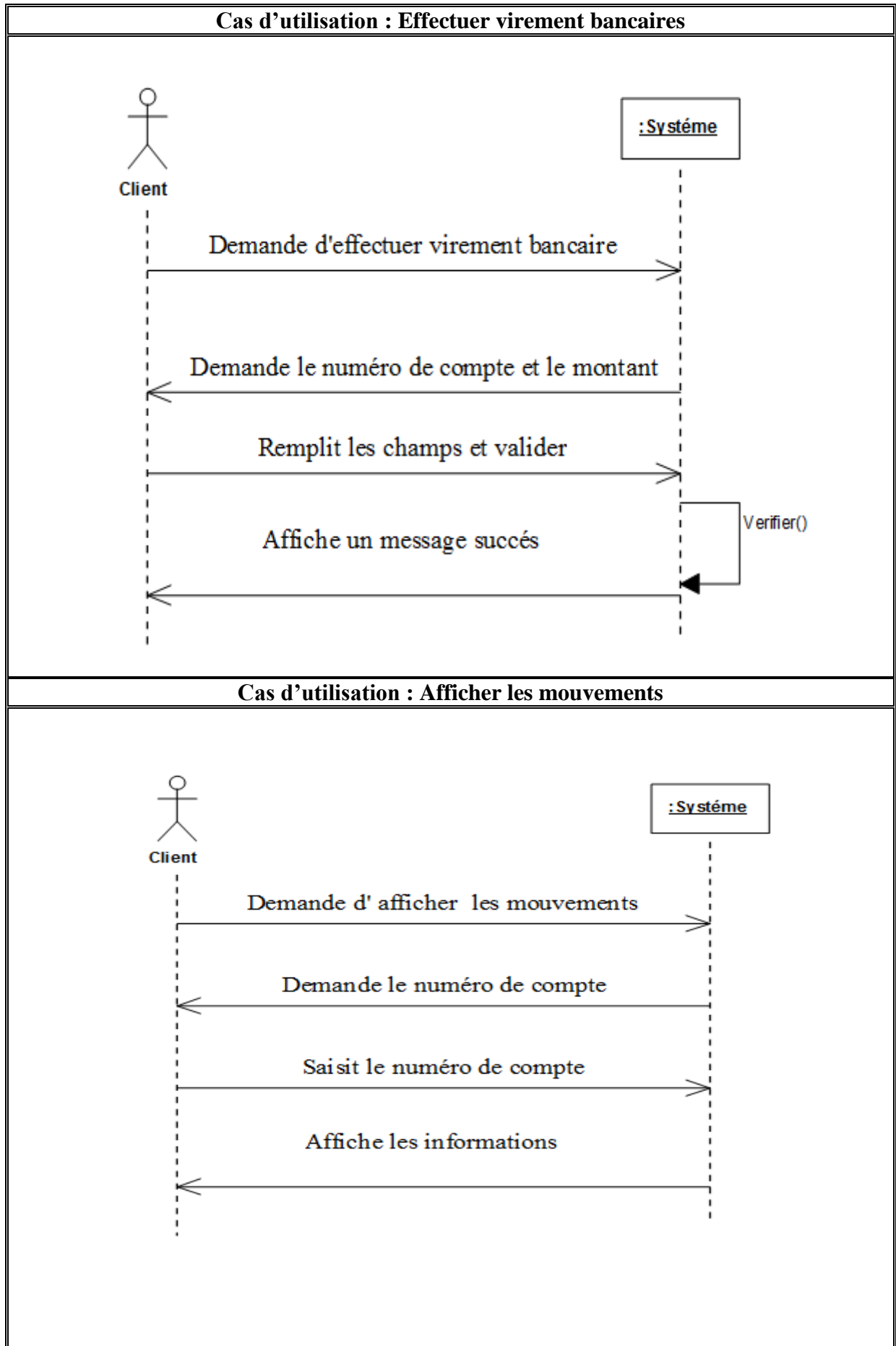
3.2.2 Description graphique des cas d'utilisation

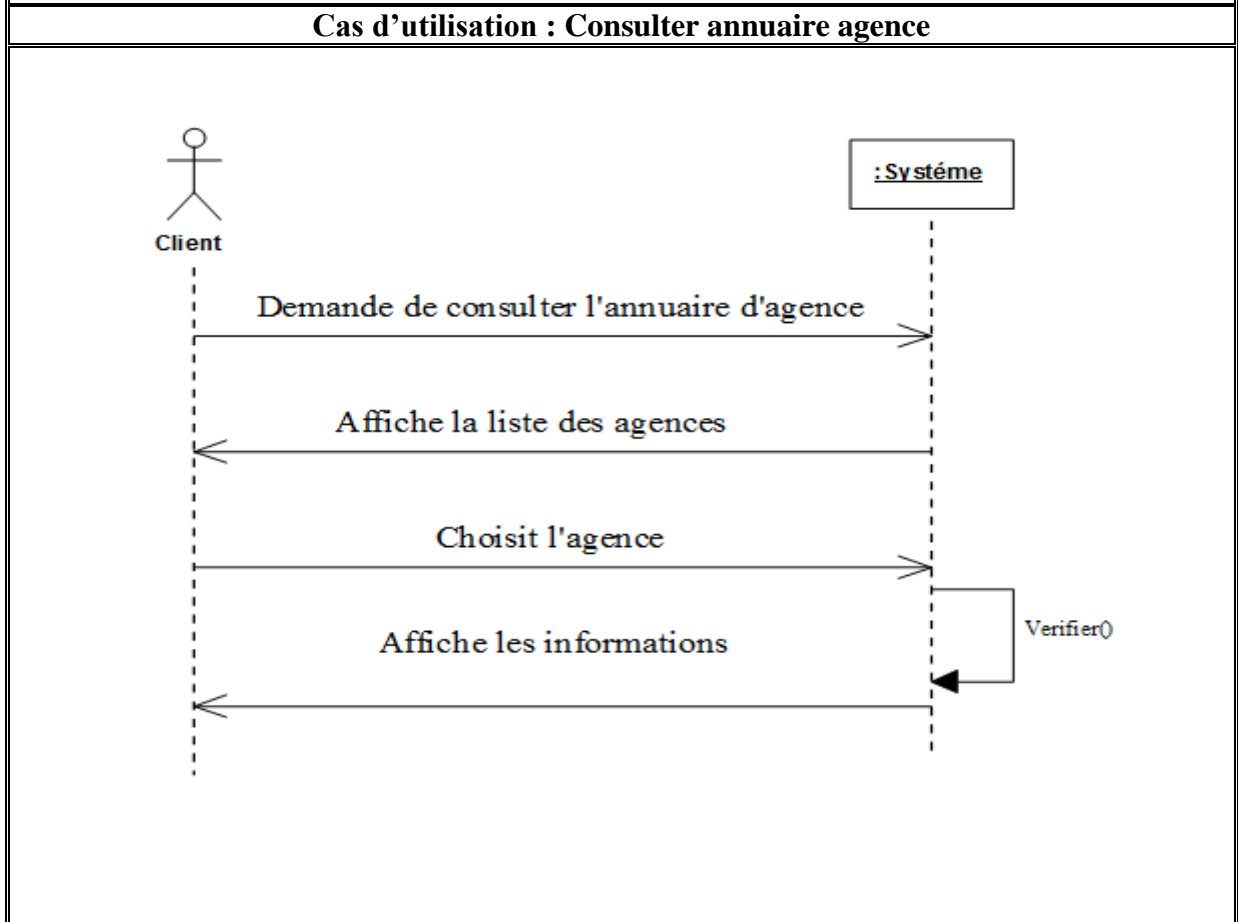
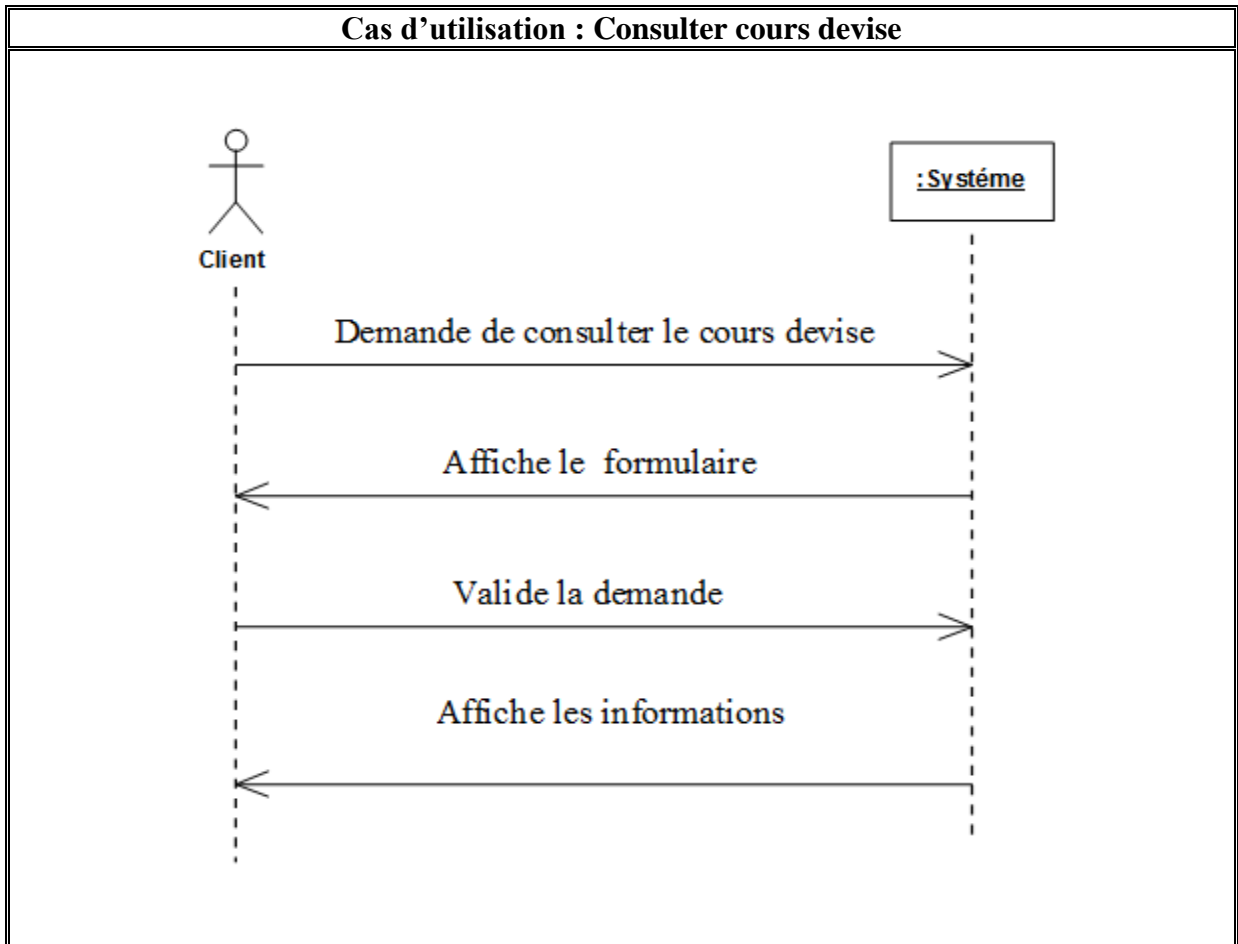
Pour documenter les cas d'utilisation, la description textuelle est insuffisante, car elle seul ne permet pas de communiquer facilement et précisément avec les utilisateurs. Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques qui appartenant au niveau supérieur de formalisation. [6]

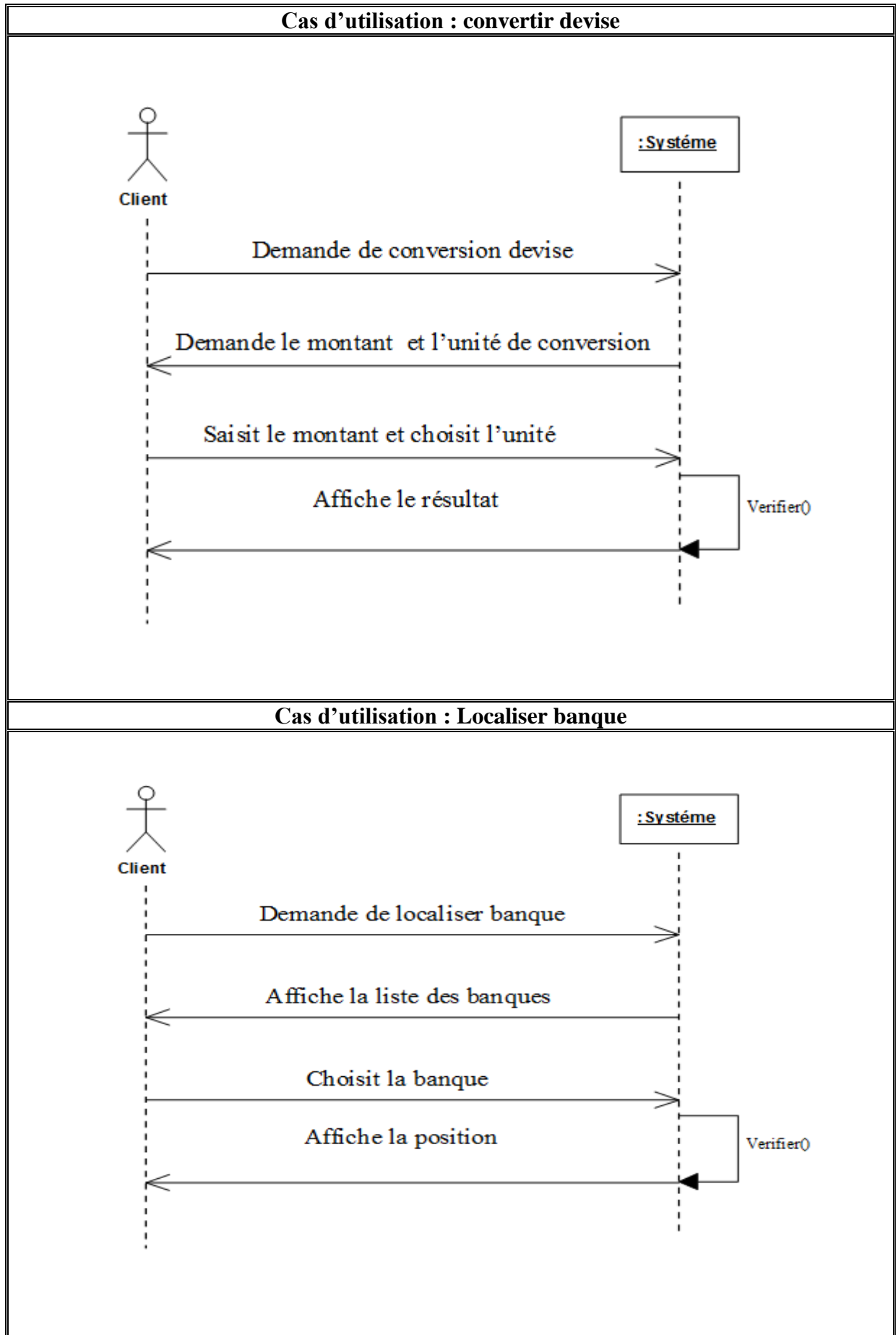
Nous choisissons pour cela parmi les diagrammes dynamique (diagramme de collaboration, diagramme d'état transition, diagramme d'activité) : le diagramme de séquence, c'est une bonne illustration des scénarios particuliers, car il est facilement compris par l'utilisateur. [6]

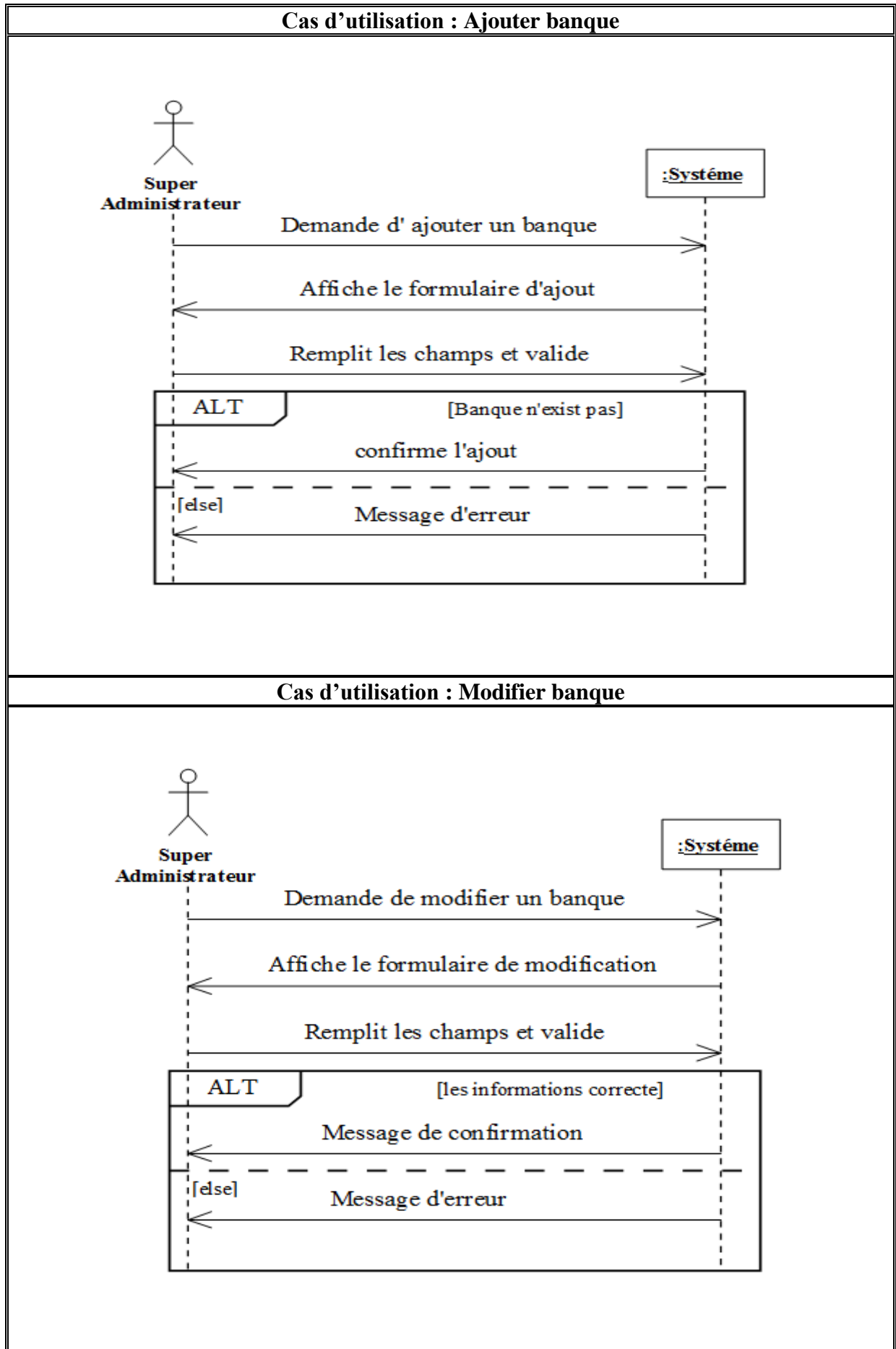


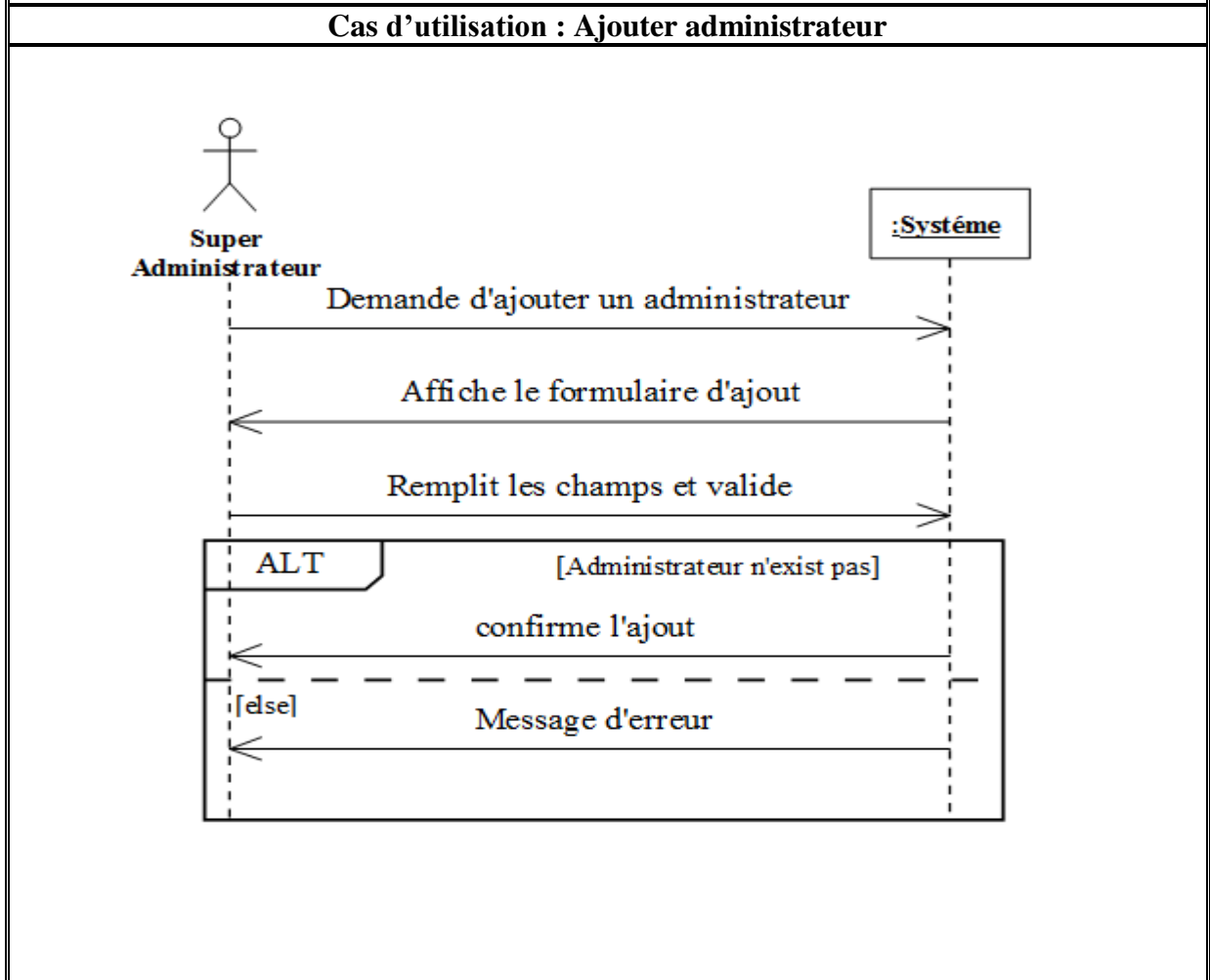
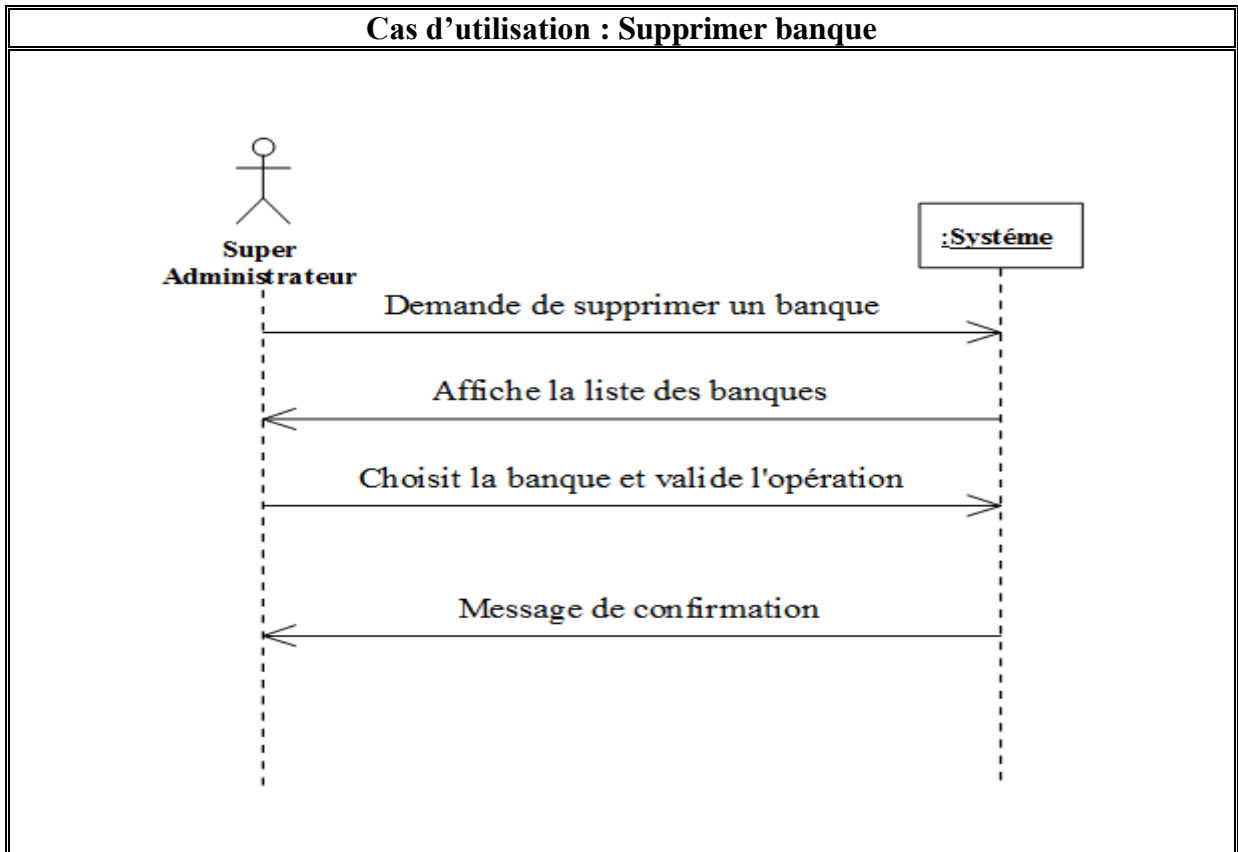


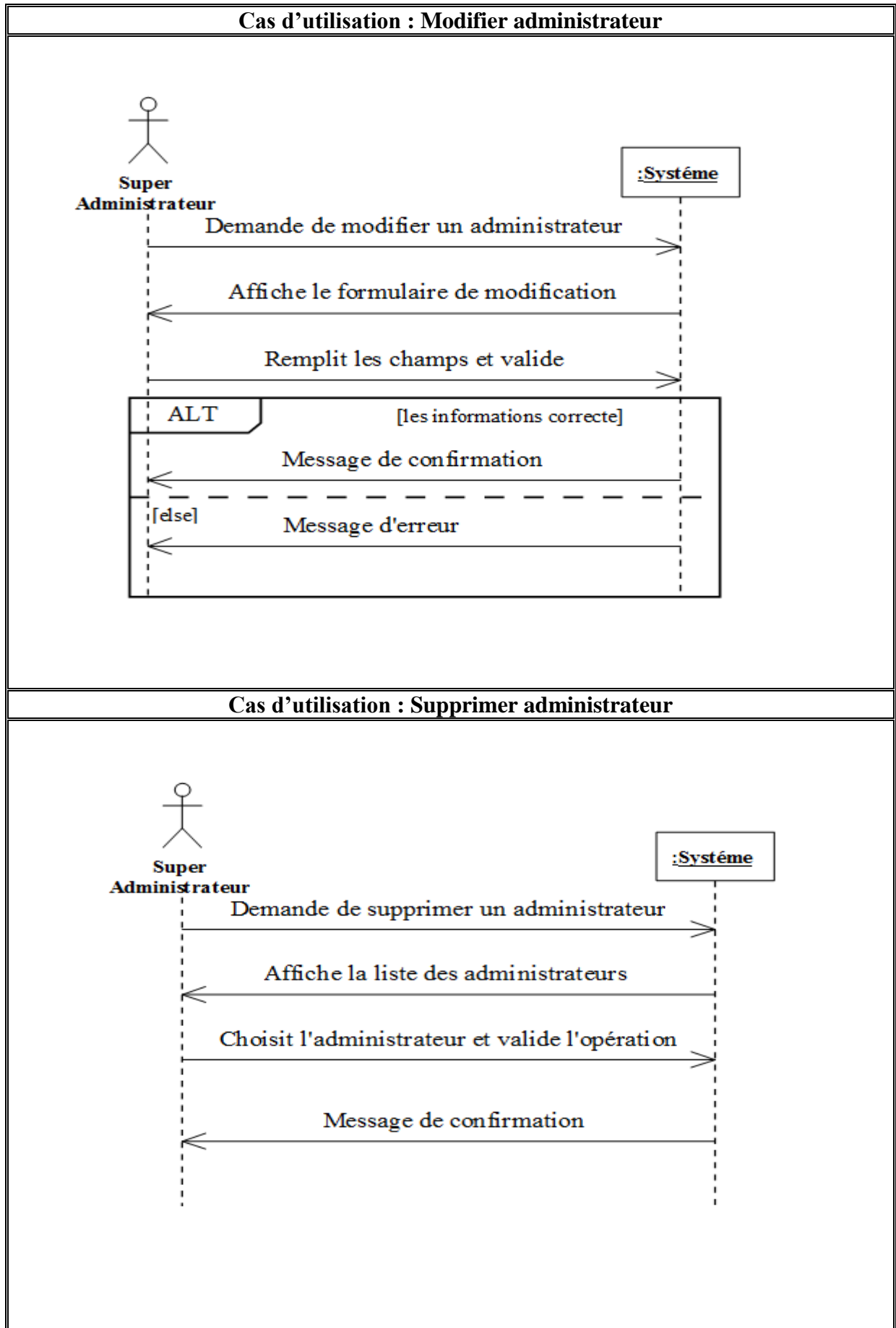


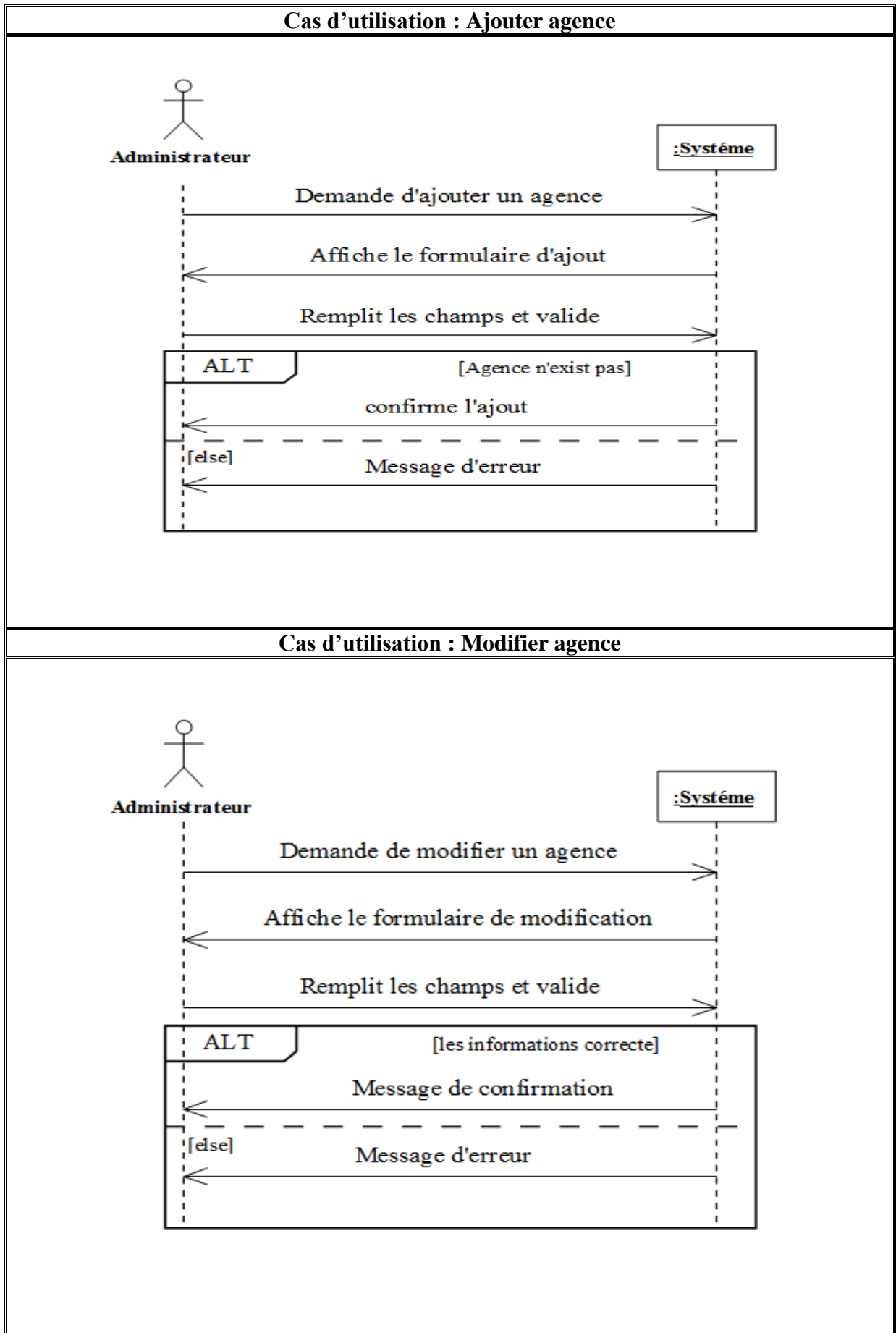












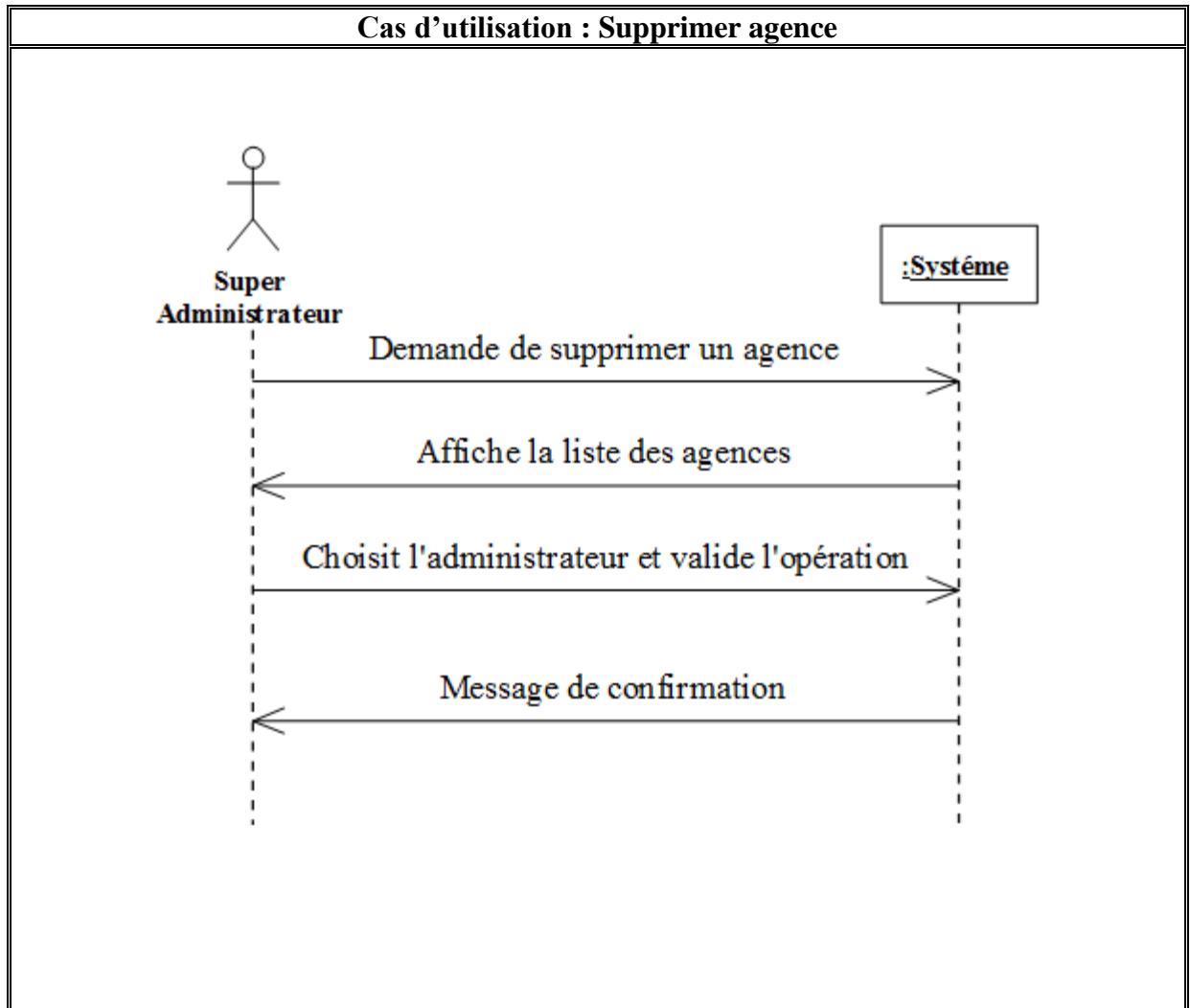


Tableau 3.3 : Description graphique des cas d'utilisation

3.3 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. [6]

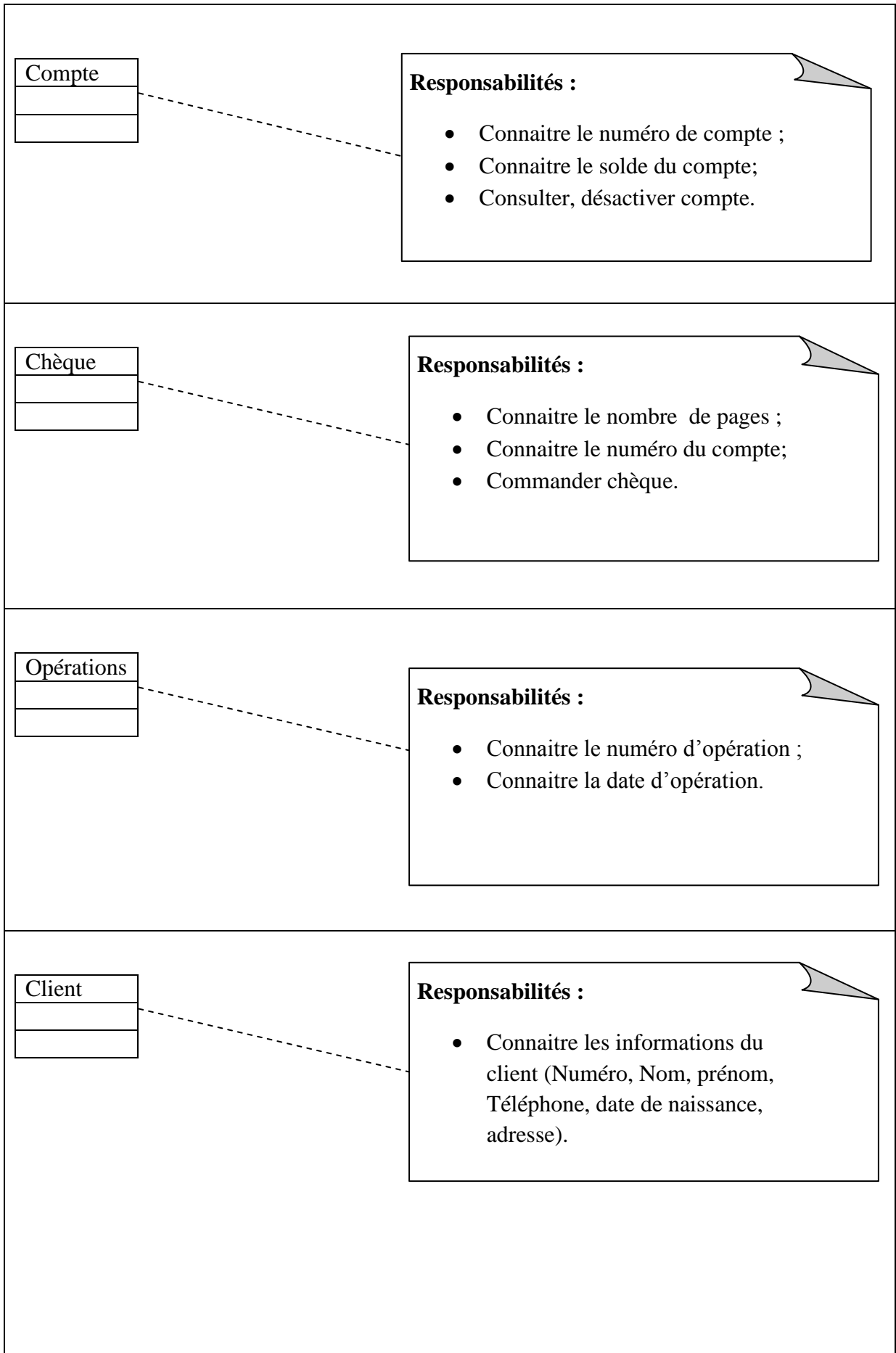
3.3.1 La liste des classes candidates

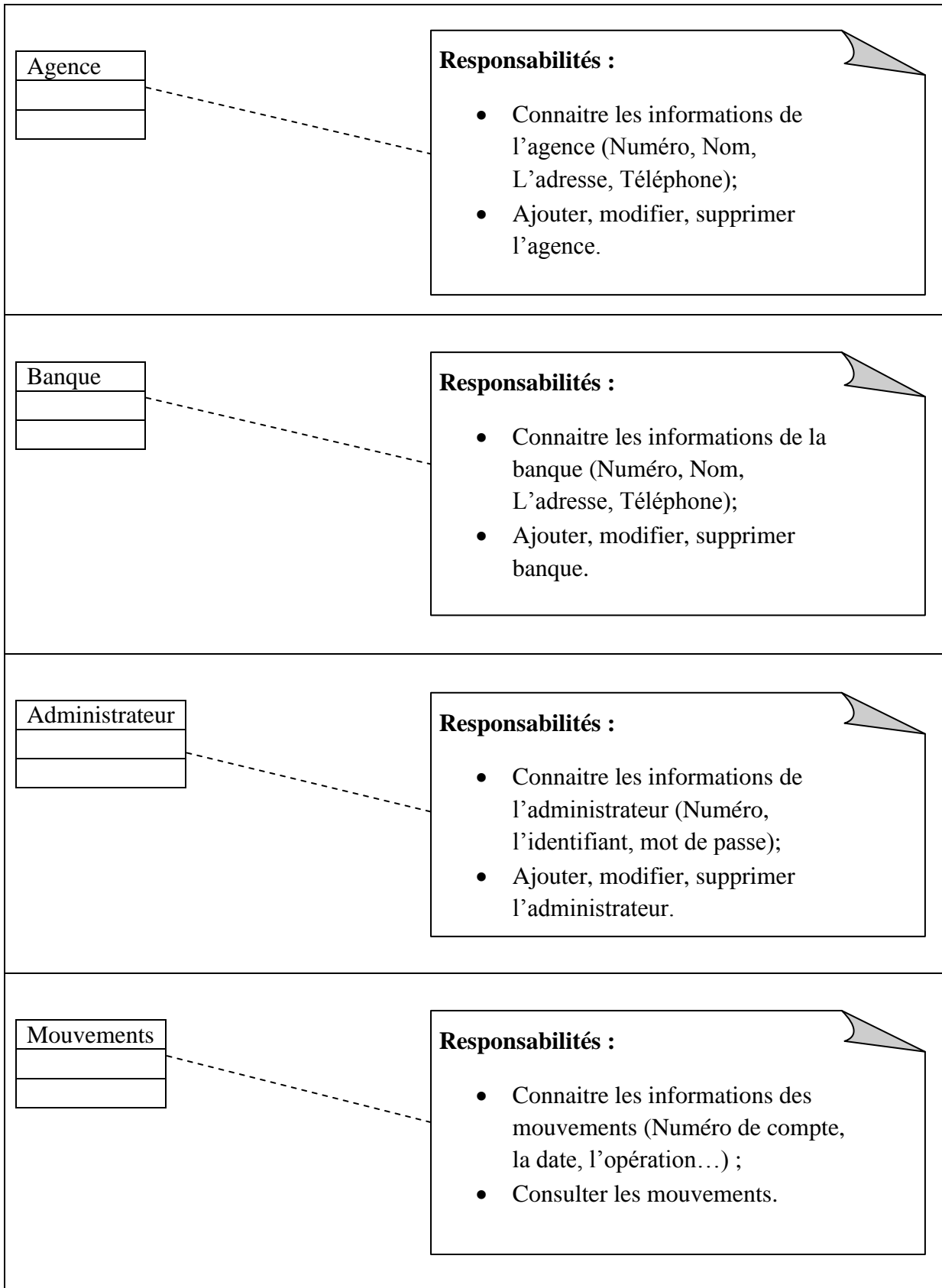
Liste des classes candidates
- Compte ; - Chèque ; - Client ; - Opérations ; - Mouvements ; - Devise ; - Annuaire ; - Virement ; - Agence ; - Banque ; - Administrateur.

Tableau 3.4 : Liste des classes candidates

3.3.2 Responsabilités des classes

Une responsabilité est une sorte de contrat, ou d'obligation, pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. En fait, on peut dire que les attributs, les opérations, les associations représentent les propriétés élémentaires qui contribueront à remplir les responsabilités de la classe. [6]

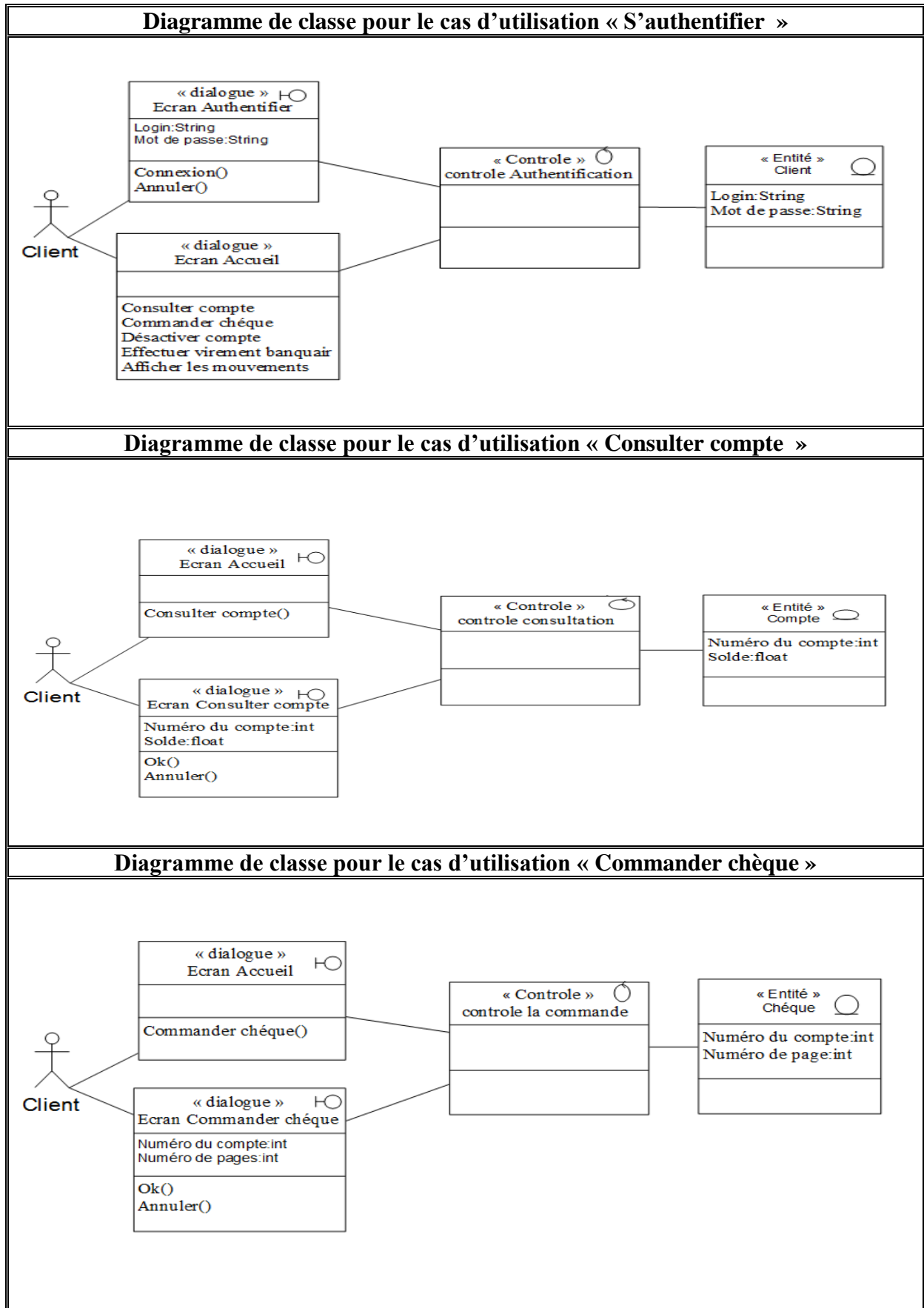


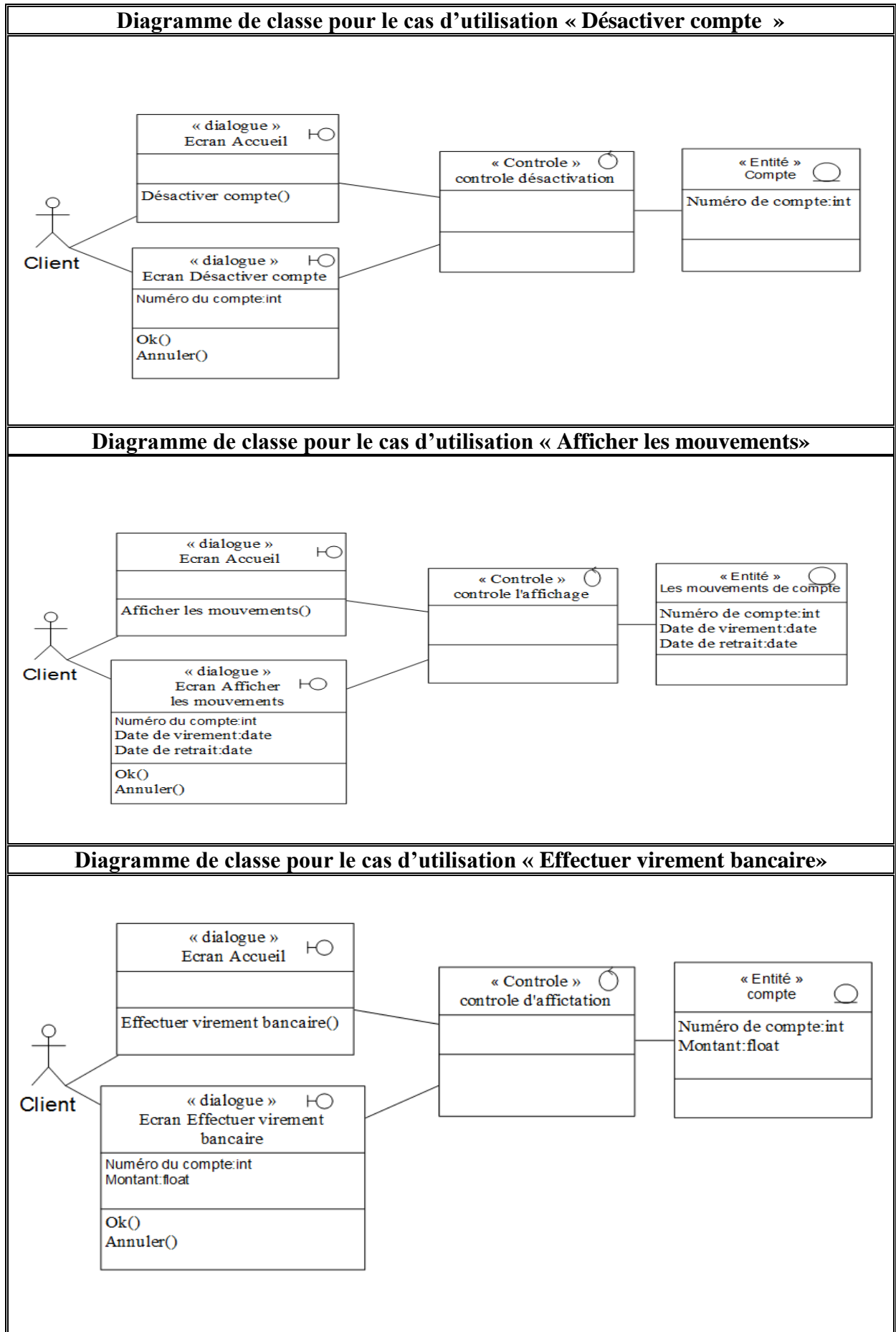


<p>Annuaire</p>	<p>Responsabilités :</p> <ul style="list-style-type: none"> • Connaitre ses informations (wilaya, Nom_Ag, Adresse, Tel ...); • Consulter l'annuaire.
<p>Devise</p>	<p>Responsabilités :</p> <ul style="list-style-type: none"> • Connaitre ses informations (Code_dev, prix_vent, prix_achat); • Consulter cours devise ; • Convertir devise.
<p>Virement</p>	<p>Responsabilités :</p> <ul style="list-style-type: none"> • Connaitre ses informations (Montant, compte_S, compte_D) ; • Effectuer virement.

Tableau 3.5 : Responsabilités des classes candidates

3.4 Diagramme de classe participantes





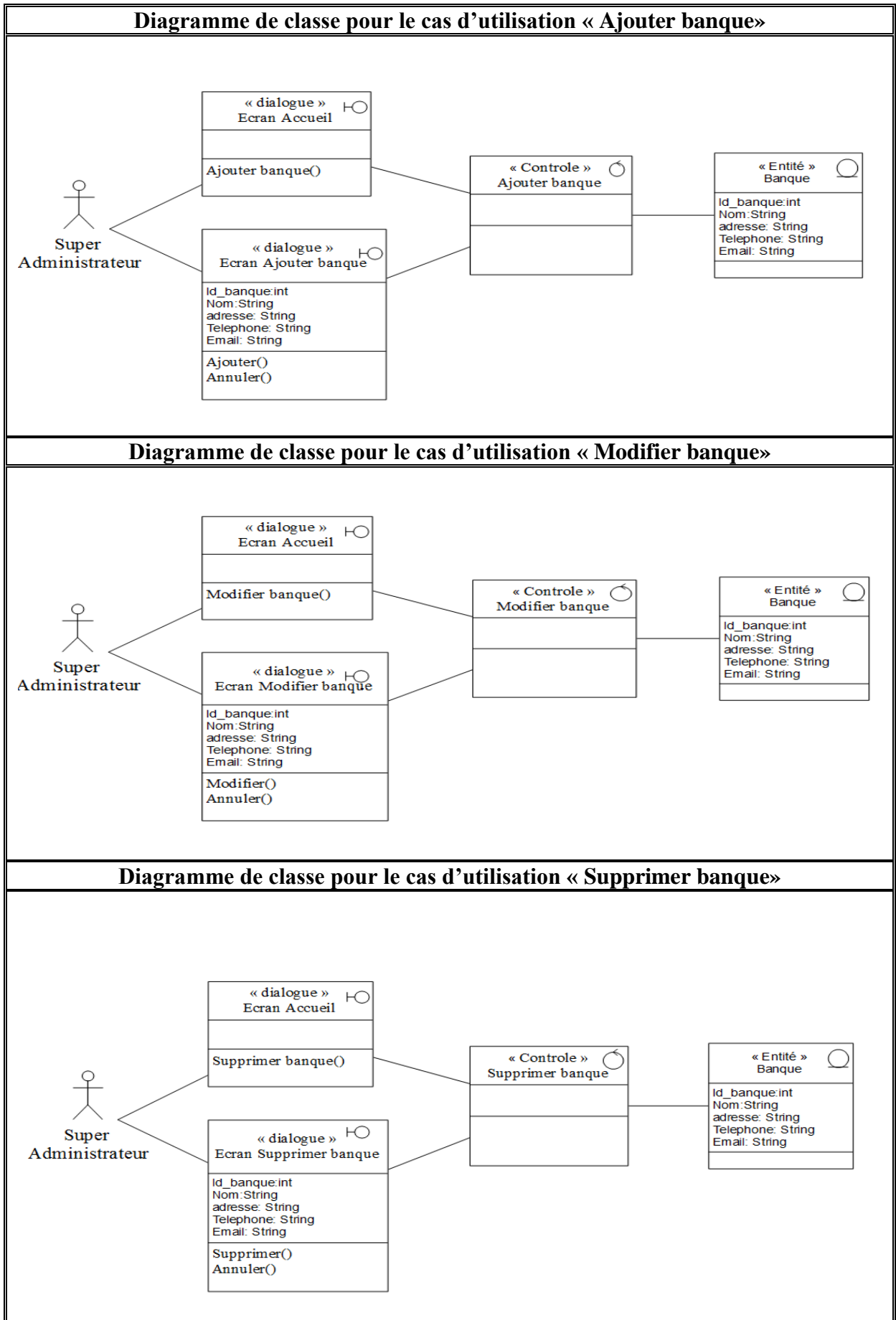


Diagramme de classe pour le cas d'utilisation « consulter cours devise »

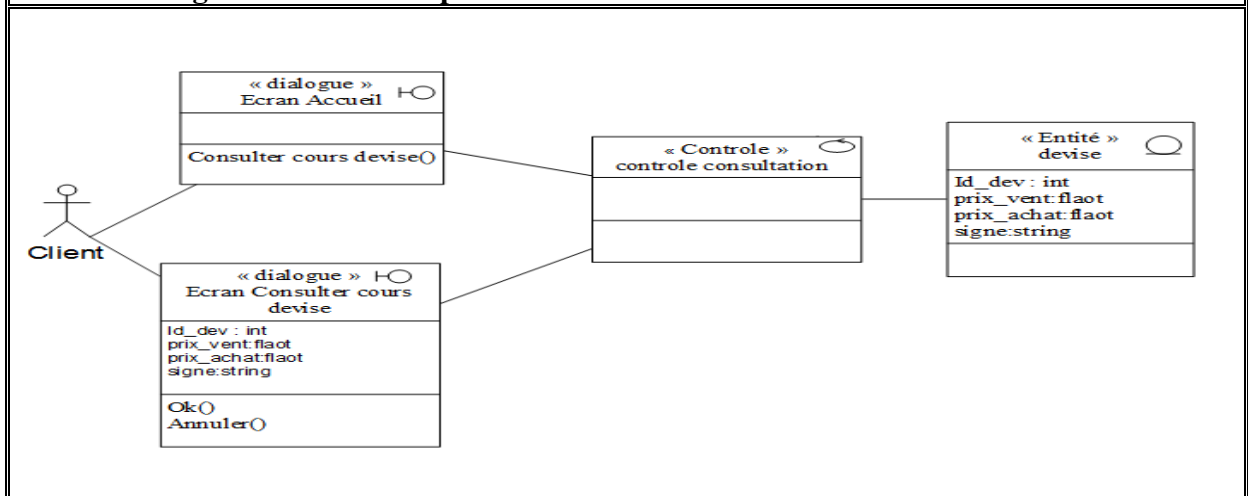


Diagramme de classe pour le cas d'utilisation « consulter annuaire agences »

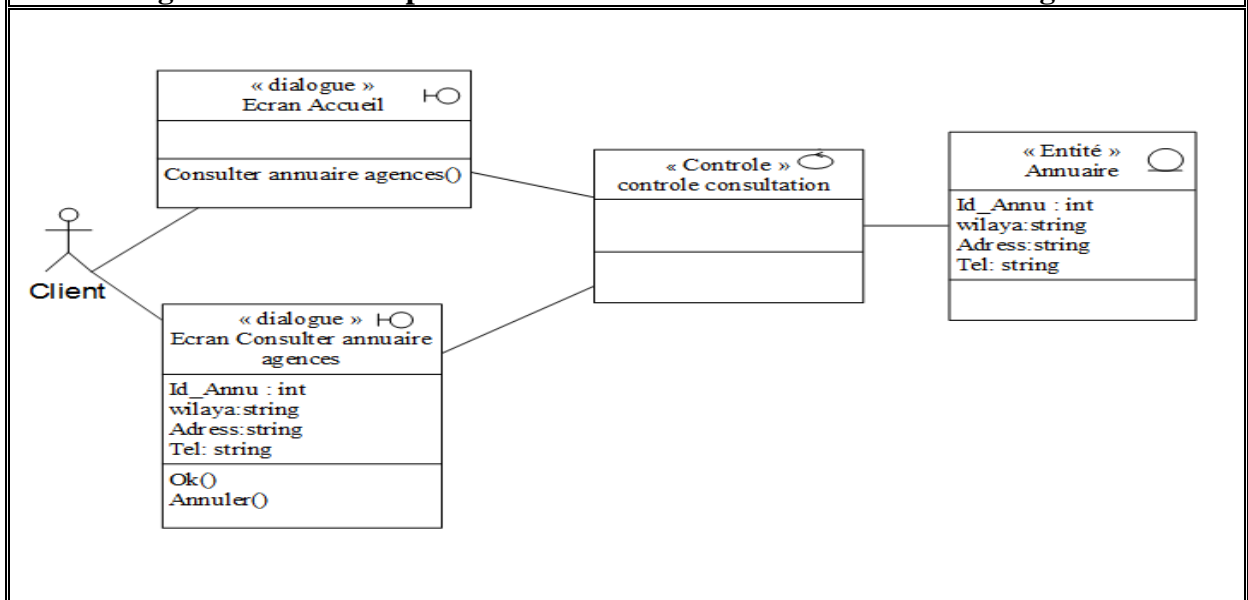
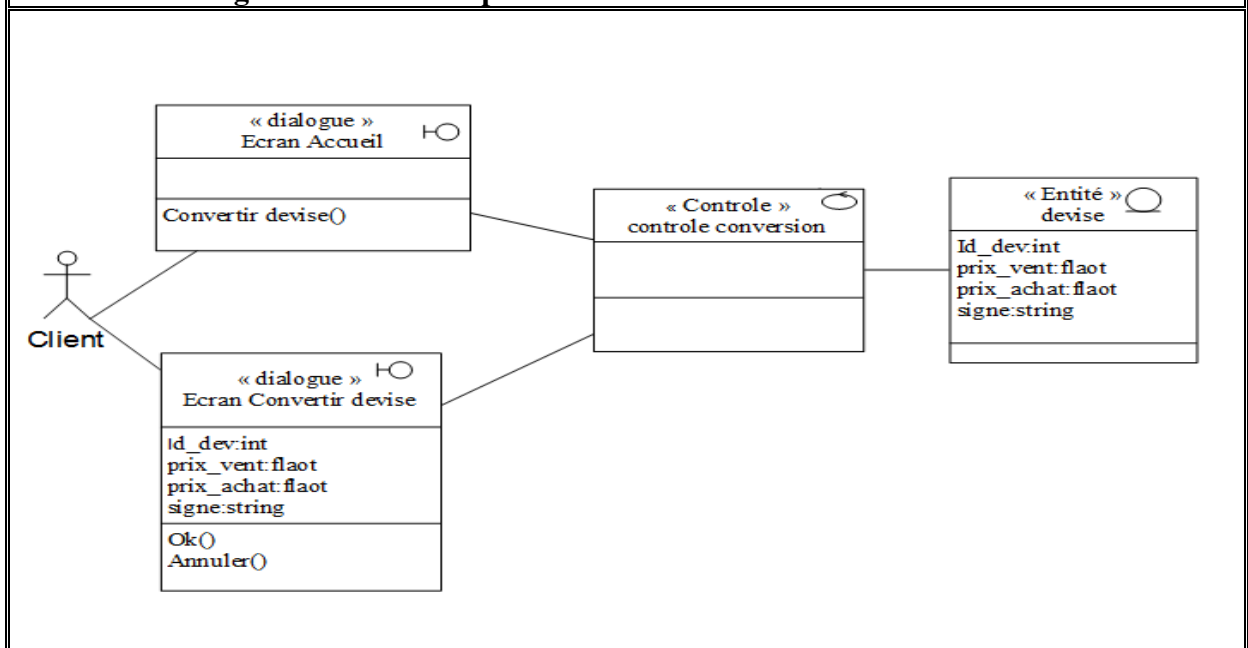


Diagramme de classe pour le cas d'utilisation « convertir devise »



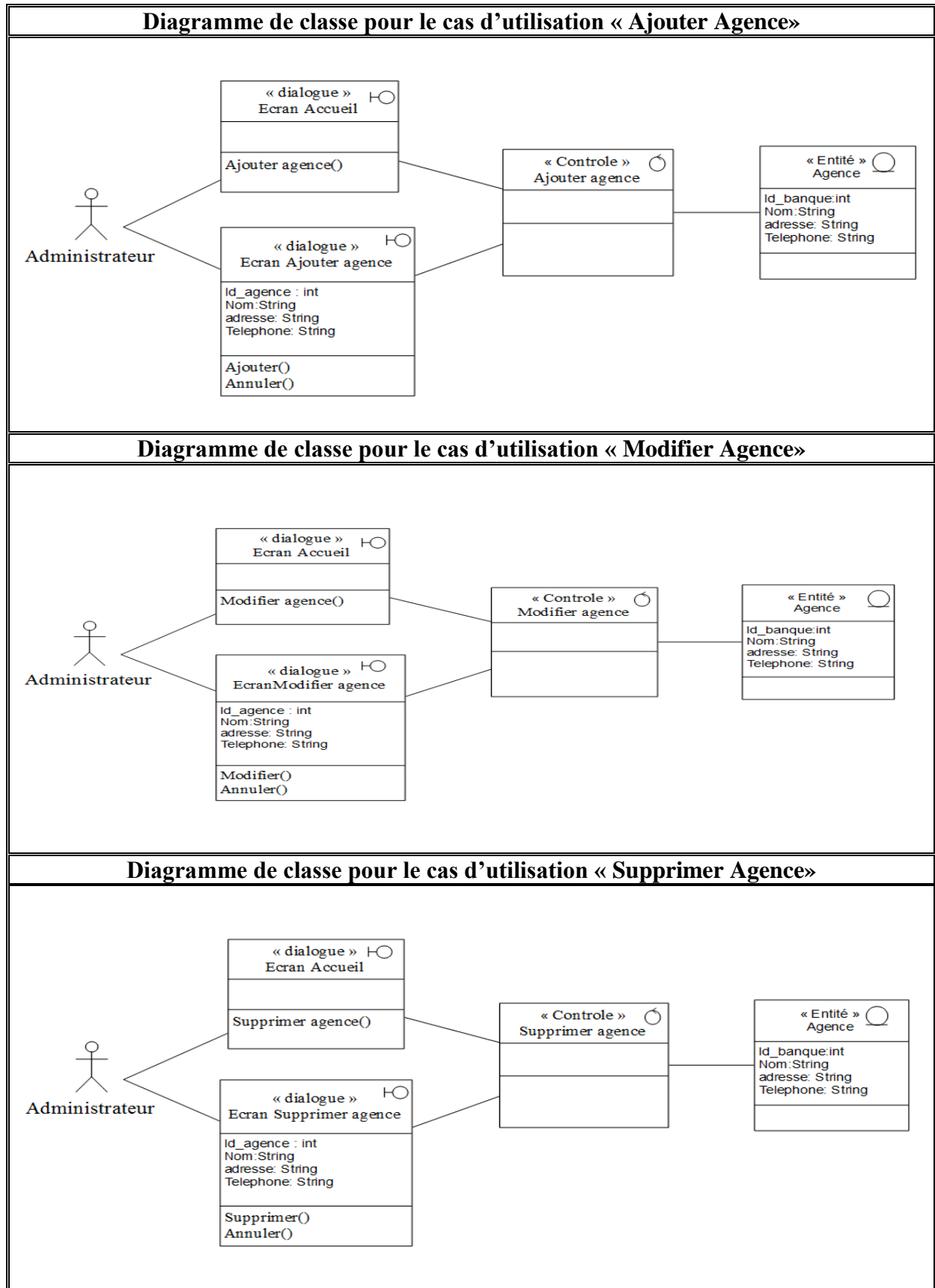


Tableau 3.6 : Diagramme de classe des cas d'utilisation

4. Capture des besoins techniques

La capture des besoins technique est la première étape de la branche droite du cycle en Y, elle est primordiale pour la conception de l'architecture. C'est une étape de prise en compte des contraintes techniques et logicielles. [8]

Nous allons suivre les étapes suivantes :

4.1 Spécification technique du point de vue matériel

Les choix stratégiques de développement impliquent les contraintes relatives à la configuration du matériel et la façon dont seront organisés et déployés les composants d'exploitation du système. [8]

4.1.1 Style d'architecture en niveaux

L'architecture du système est une architecture client serveur à deux niveaux.

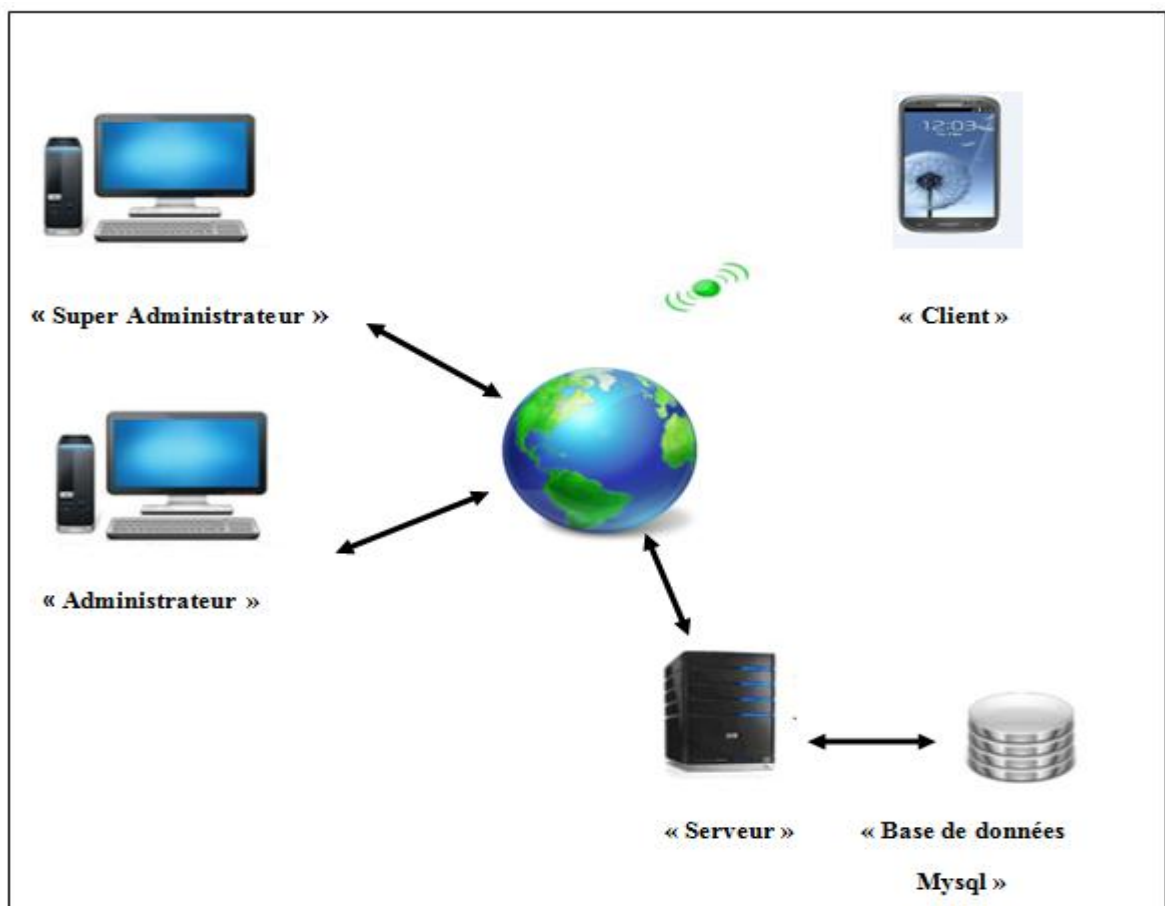


Figure 3.4 : Architecture client /serveur du système

4.1.2 Configuration matérielle du système

La configuration matérielle de notre système schématisée par un diagramme de déploiement UML est représentée par la figure suivante :

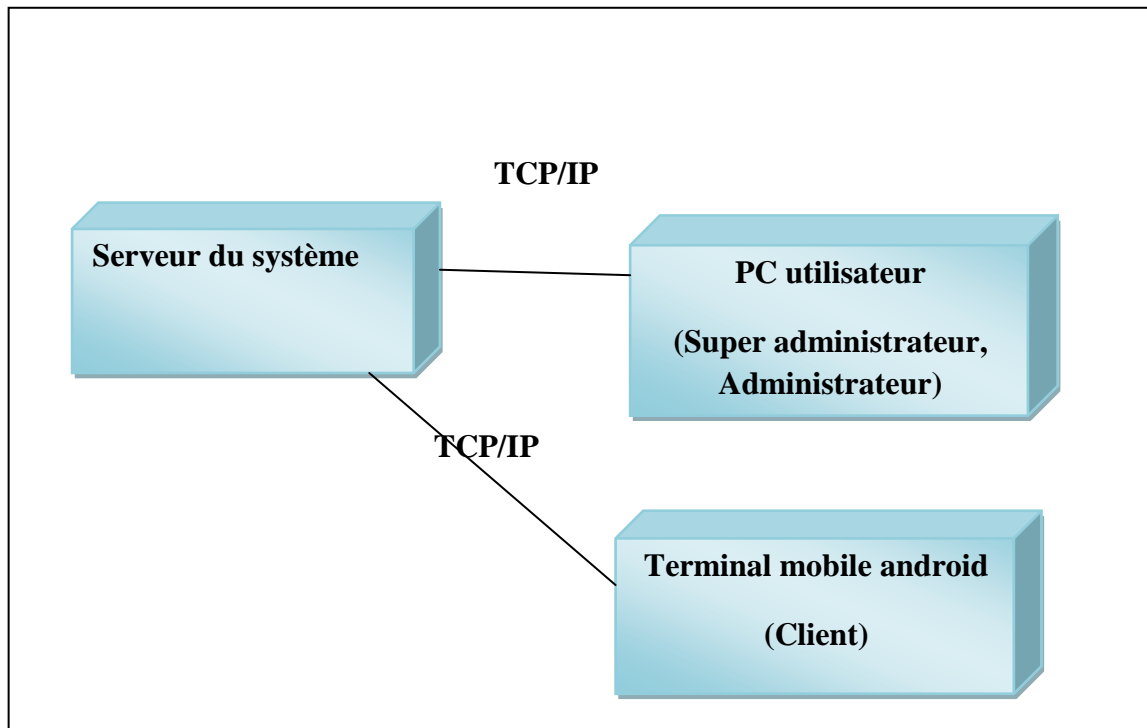


Figure 3.5 : Configuration matérielle du système

4.2 Spécification d'architecture

L'expression des pré-requis technique implique également le choix d'un style d'architecture client/serveur. Ce choix conditionne la façon dont seront organisées et déployés les composants d'exploitation du système. [8]

4.2.1 Le style d'architecture en tiers

Le style d'architecture en tiers (tiers signifie « partie » en anglais) spécifie 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. [8]

- **Le style d'architecture en 2 tiers**

Le style d'architecture 2-tiers correspond à la configuration la plus simple d'un système client/serveur. Dans ce cas, il incombe aux clients de gérer l'interface utilisateur et les processus d'exploitation. Les serveurs ont pour responsabilité de traiter le stockage de données. Ce type d'architecture est parfaitement adapté aux systèmes départementaux. [8]

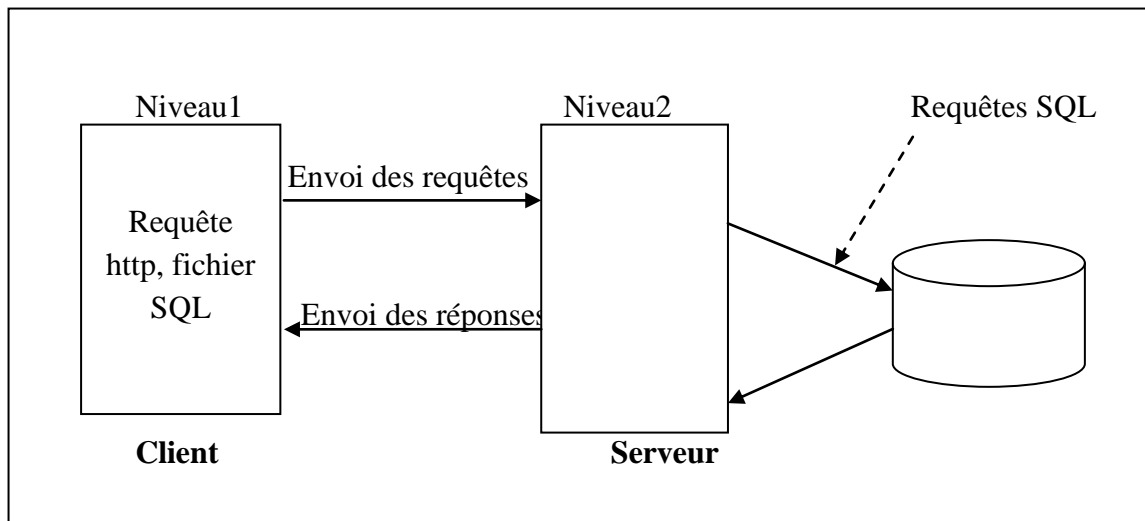


Figure 3.6 : Architecture à deux niveaux (Client / Serveur)

4.2.2 Modèle préliminaire des composants du système

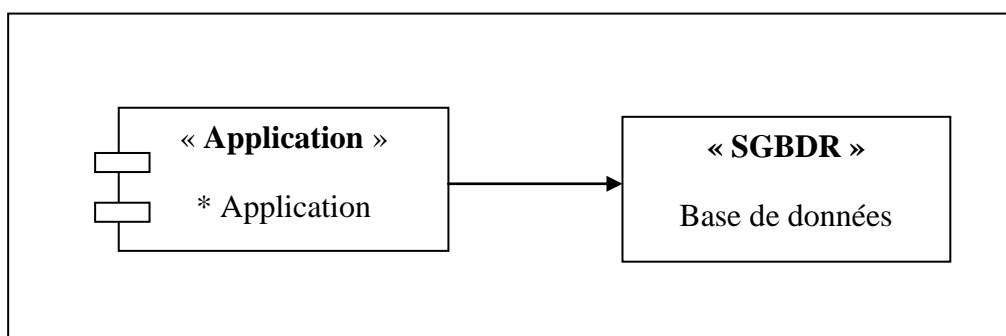


Figure 3.7 : Modèle préliminaire des composants du système

4.3 Elaboration du modèle de spécification logicielle

Une fois que la spécification technique et l'architecture est exprimée, on s'intéresse aux fonctionnalités propres du système technique en procédant à une spécification logicielle. Dans ce cas, on utilise les cas d'utilisation de manière différente que pour la spécification

fonctionnelle. C'est pourquoi nous avons introduit le concept d'exploitant et de cas d'utilisation technique.

- **Exploitant** : un exploitant est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités technique de système. Tout système informatique possède au minimum un exploitant qui est « l'utilisateur de système ». [8]
- **Cas d'utilisation technique** : un cas d'utilisation technique est destiné à l'exploitant, c'est une séquence d'actions produisant une valeur ajoutée opérationnelle ou purement technique. [8]

4.3.1 Identification des cas d'utilisation techniques

Les exploitants du système sont :

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

Les cas d'utilisation technique du système sont :

- S'authentifier ;
- Gérer les droits d'accès ;
- Gérer les erreurs ;
- Utilise l'aide. [7]

4.3.2 Modèle de spécification logicielle

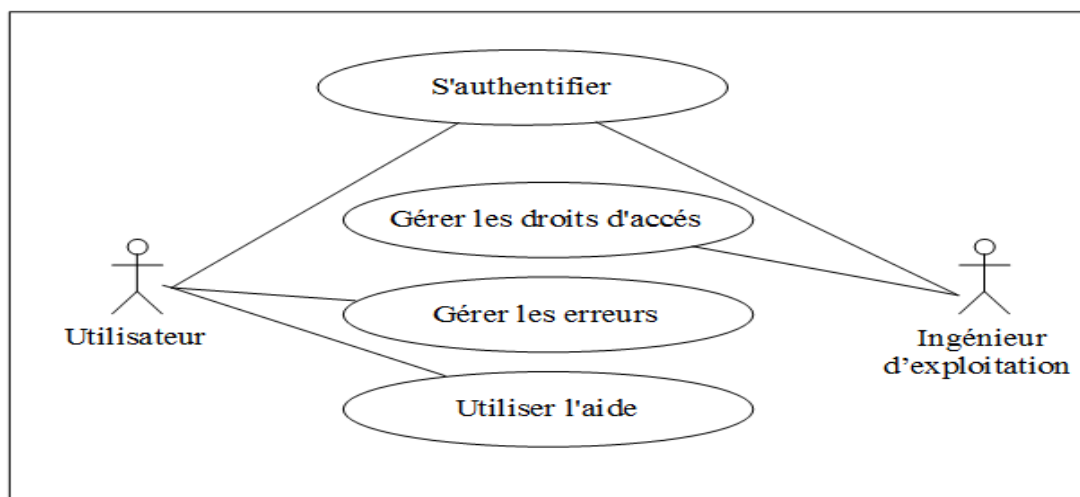


Figure 3.8 : Modèle de spécification logicielle du système

4.3.3 Description textuelle des cas d'utilisation techniques

Cas d'utilisation : S'authentifier	
Intention	Tous les utilisateurs du système doivent être reconnus par le système à travers les mots de passes.
Action	Contrôler et autoriser l'accès au système.
Cas d'utilisation : Gérer les droits d'accès	
Intention	Donner la possibilité a l'administrateur de gérer les droits d'accès au système.
Action	Définir les noms des utilisateurs et leurs mots de passes.
Cas d'utilisation : Gérer les erreurs	
Intention	Gérer les erreurs de frappe et de manipulation.
Action	Spécifier l'erreur, puis la traiter.
Cas d'utilisation : utiliser l'aide	
Intention	Touts les utilisateurs peuvent utiliser l'aide du système.
Action	Offrir un module d'aide pour faire fonctionner le système.

Tableau 3.7 : Description textuelle des cas d'utilisation

5. Conclusion

La phase de capture des besoins est une étape qui œuvre d'un coté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, et d'un autre coté, elle donne une première vue pour la prochaine étape concernant l'analyse, afin d'identifier les classes candidates du modèle statique qui présentent une des approches orienté objet. La phase de capture des besoins techniques couvre par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description des métiers des utilisateurs ni de la description applicative.



Chapitre 04

Analyse

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Situation de l'analyse dans 2TUP**
- 3. Découpage en catégorie**
- 4. Développement du modèle statique**
- 5. Développement du modèle dynamique**
- 6. Conclusion**

1. Introduction

La phase d'analyse représente la deuxième étape de la branche gauche du cycle en Y. Elle est consacrée à l'analyse d'objet de notre système juste après l'étude préliminaire et la capture des besoins fonctionnels et technique. Consiste à étudier précisément les spécifications fonctionnelles afin d'obtenir une idée de ce que va réaliser le système en terme de métiers. Elle est constituée de 3 activités :

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

2. Situation de l'analyse dans 2TUP

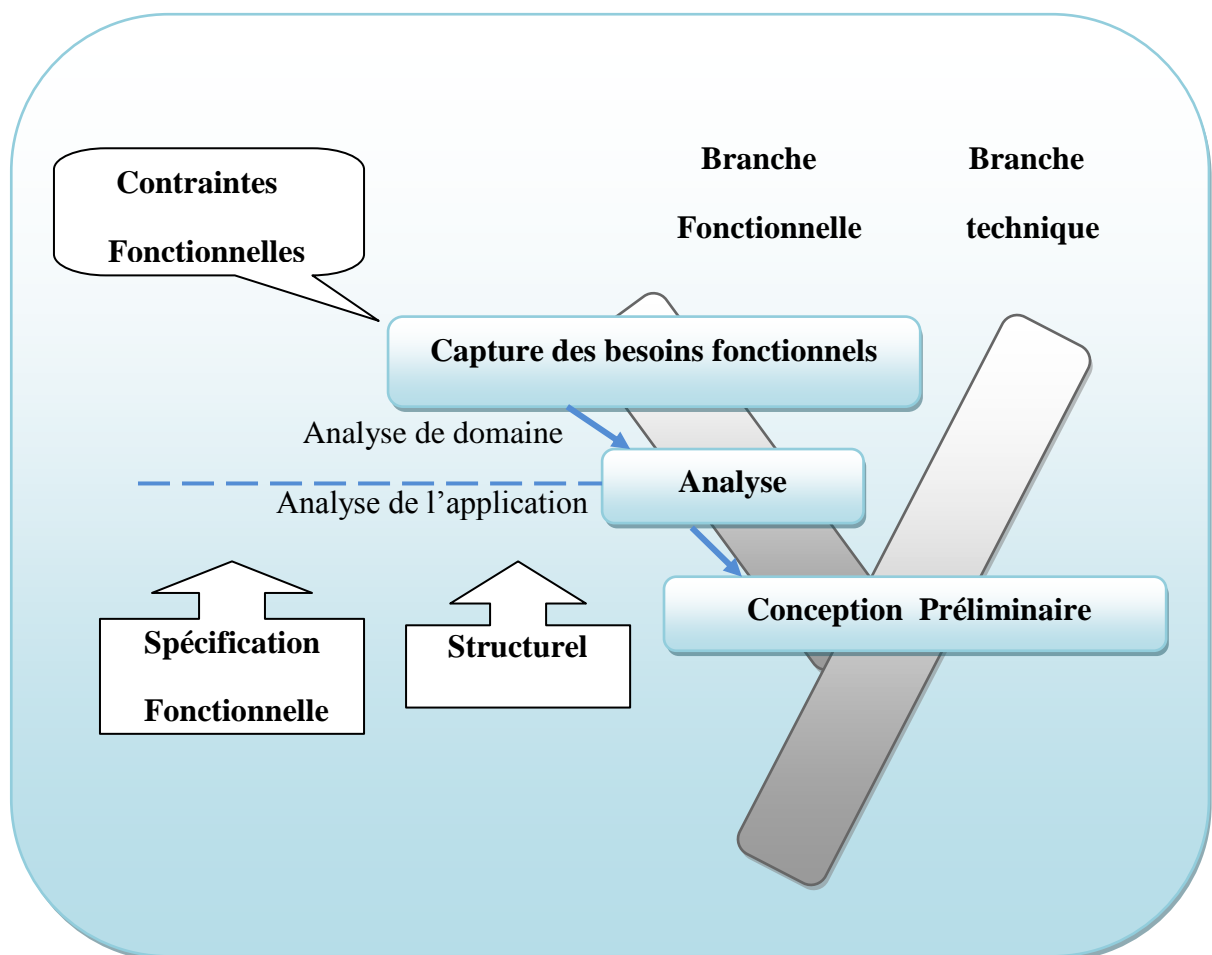


Figure 4.1 : Situation du découpage en catégorie dans 2TUP

3. Découpage en catégorie

Le découpage en catégories constitue la première activité de l'étape d'analyse, permet de déterminer les classes fondamentales du projet en utilisant les diagrammes de classes participantes dégagées dans l'étape de captures des besoins fonctionnels, Il 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.

3.1 Définition de catégorie

Une catégorie est un regroupement logique de classes à fort cohérence interne et faible couplage externe, dans la notion UML elle est représentée graphiquement par des stéréotypes des package. [5]

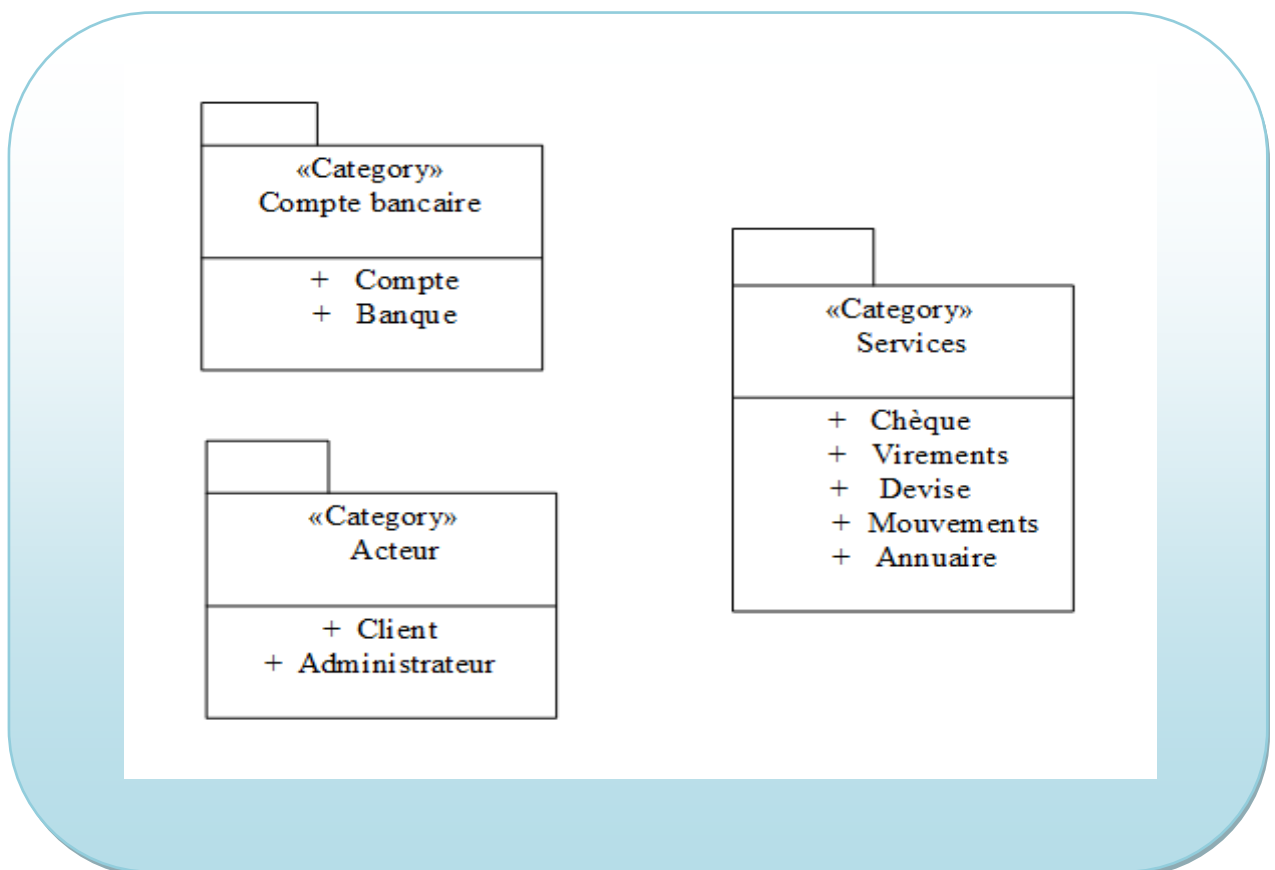


Figure 4.2 : Découpage en catégorie du système

3.2 Elaboration des diagrammes de classes préliminaire par catégories

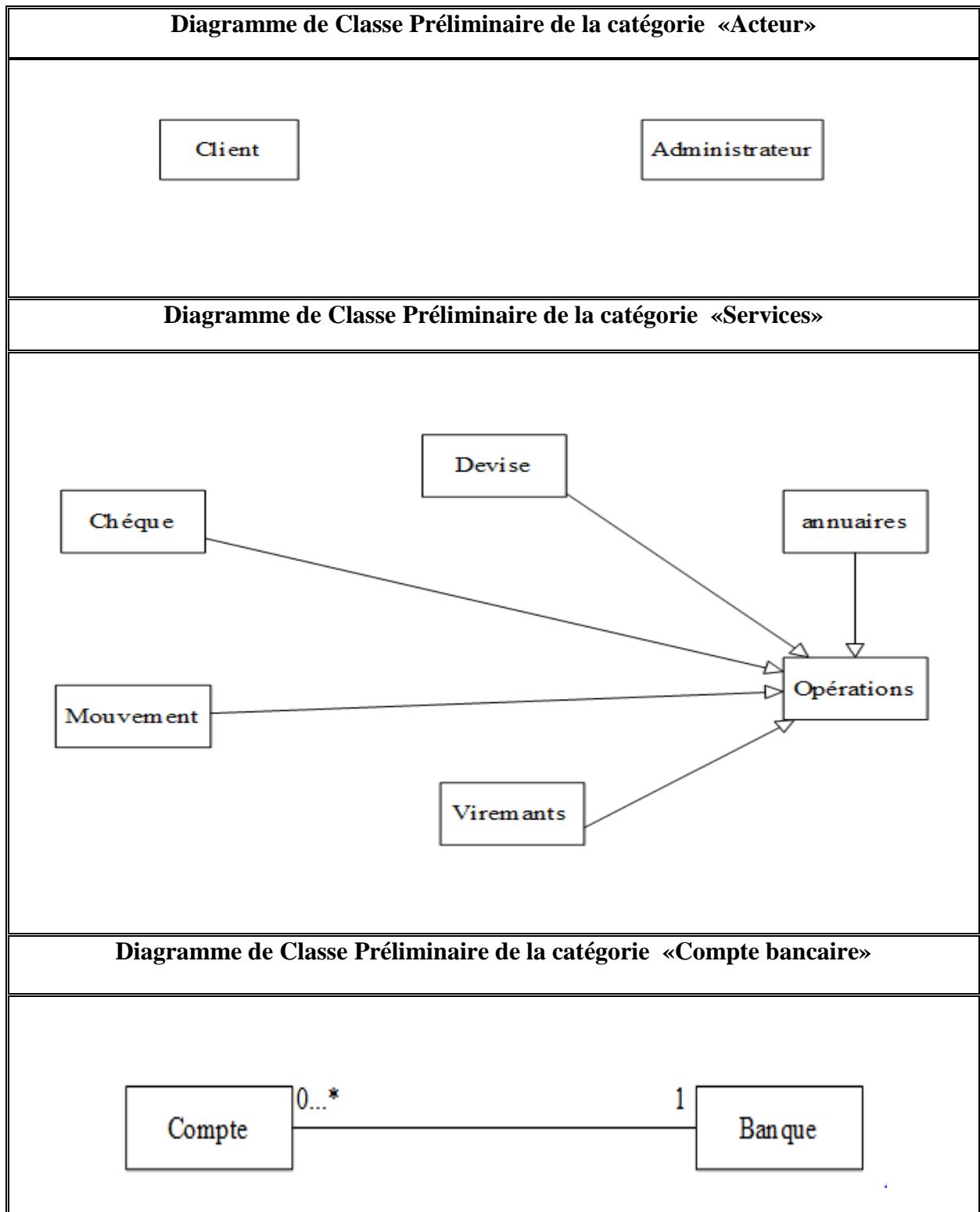


Tableau 4.1 : Diagramme de classe préliminaire par catégories

3.3 Modèle structurel

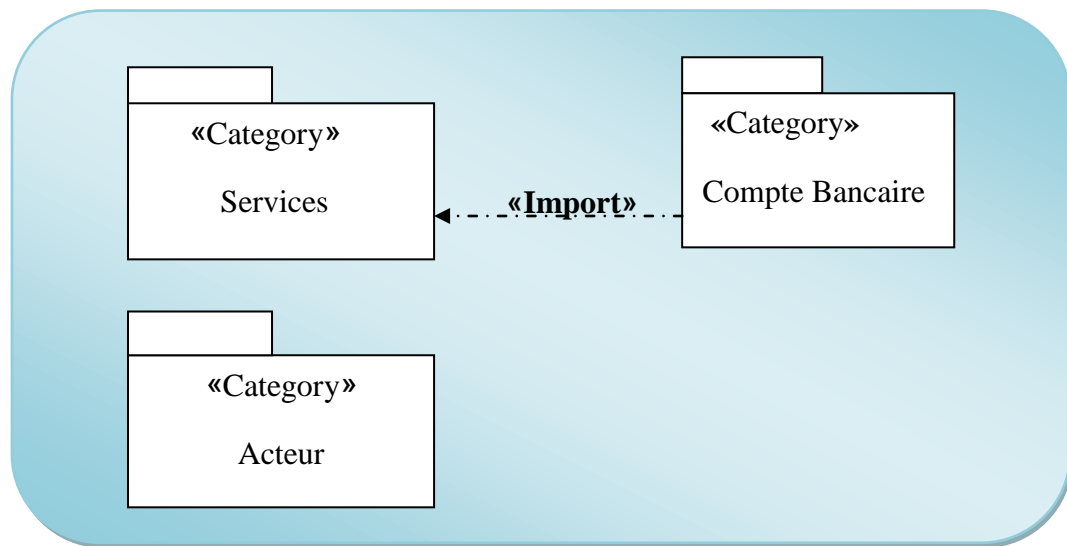


Figure 4.3 : Modèle structurel d'analyse

4. Développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y et succède le découpage en catégories. Les diagrammes de classes établis sommairement dans les diagrammes de classe participante, puis réorganisés lors du découpage en catégories, vont être détaillés, complétés, et optimisés dans ce chapitre. [5]

4.1 Affiner les classes

Les classes identifiées lors de l'étude des cas d'utilisation, puis réparties dans les catégories, 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. [5]

4.2 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 une partie fondamentale de la structure statique des classes.

Affiner les associations convient de les valider, les préciser, en éliminer, et en ajouter. Il s'agit d'une activité itérative, qui sera complétée grâce à l'identification des attributs. [5]

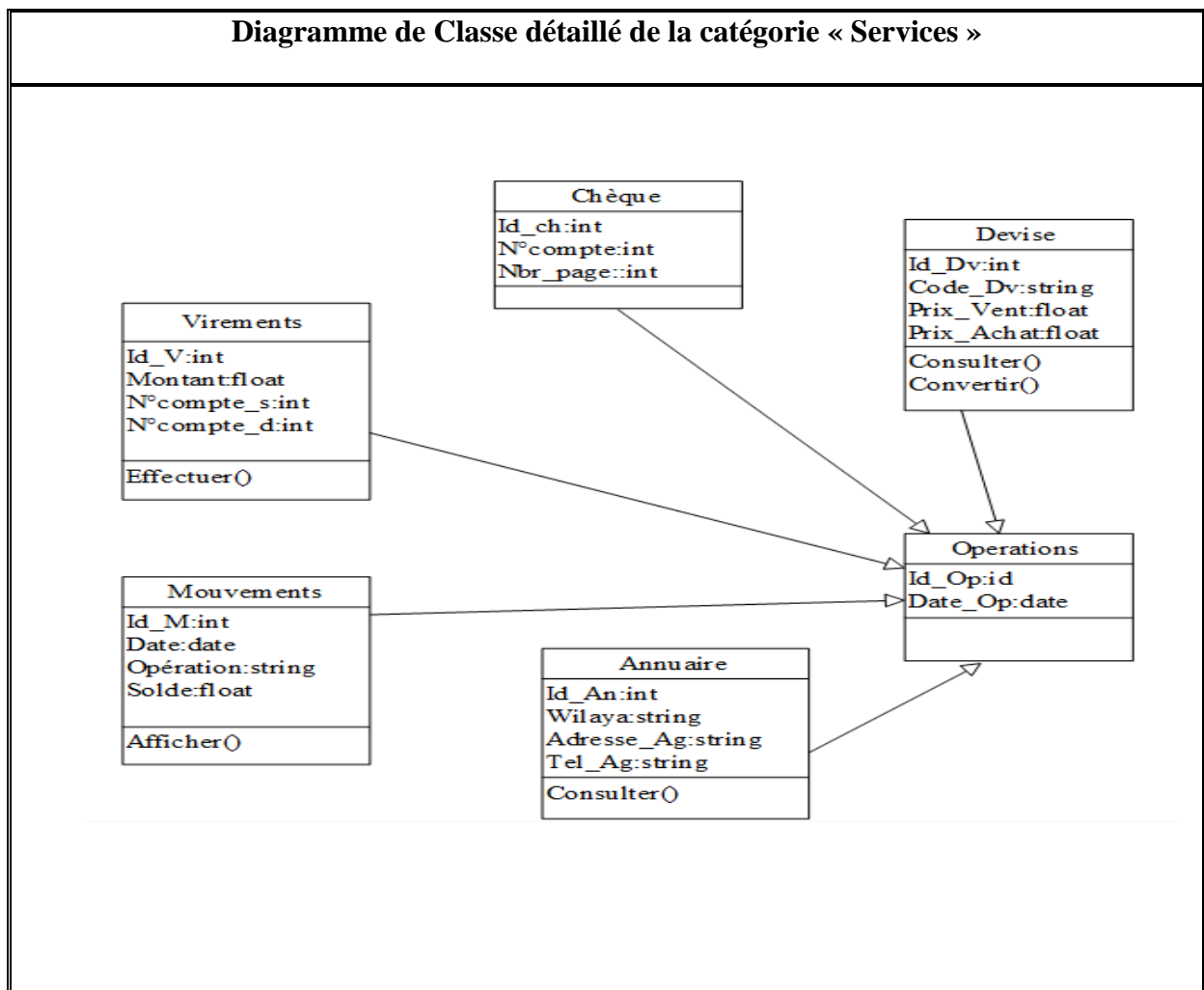
4.3 Ajouter les attributs

Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeurs possible partagé par tous les objets de classe. Dans un modèle d'analyse, nous conservons uniquement comme attributs les propriétés simples des classes que le système doit les mémoriser et utiliser. [5]

4.4 Ajouter les opérations

Une opération représente un service, 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. [5]

4.5 Diagramme de classes détaillées par catégorie



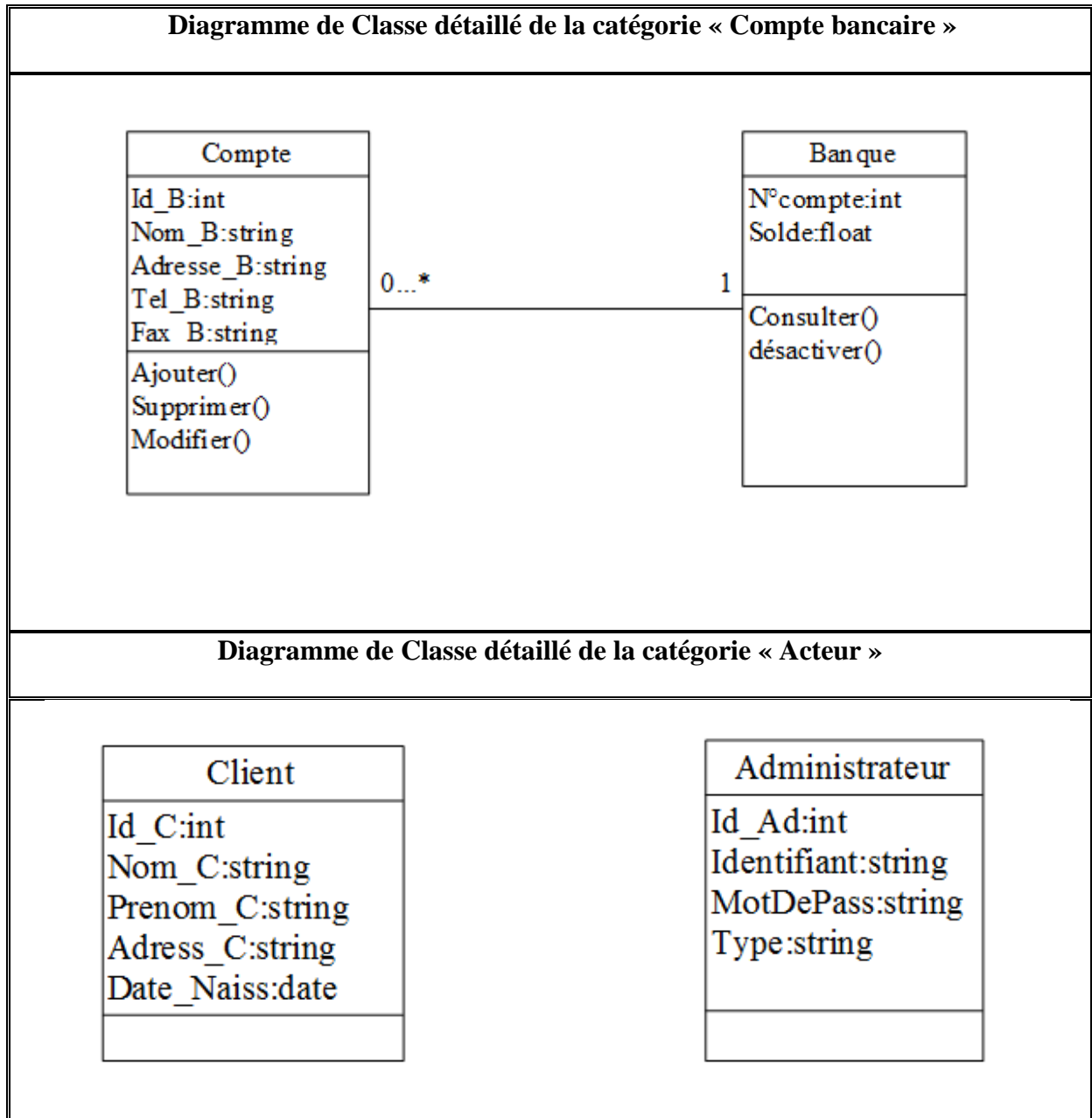


Tableau 4.2 : Diagramme de classe détaillé par catégories

4.6 Diagrammes de classes

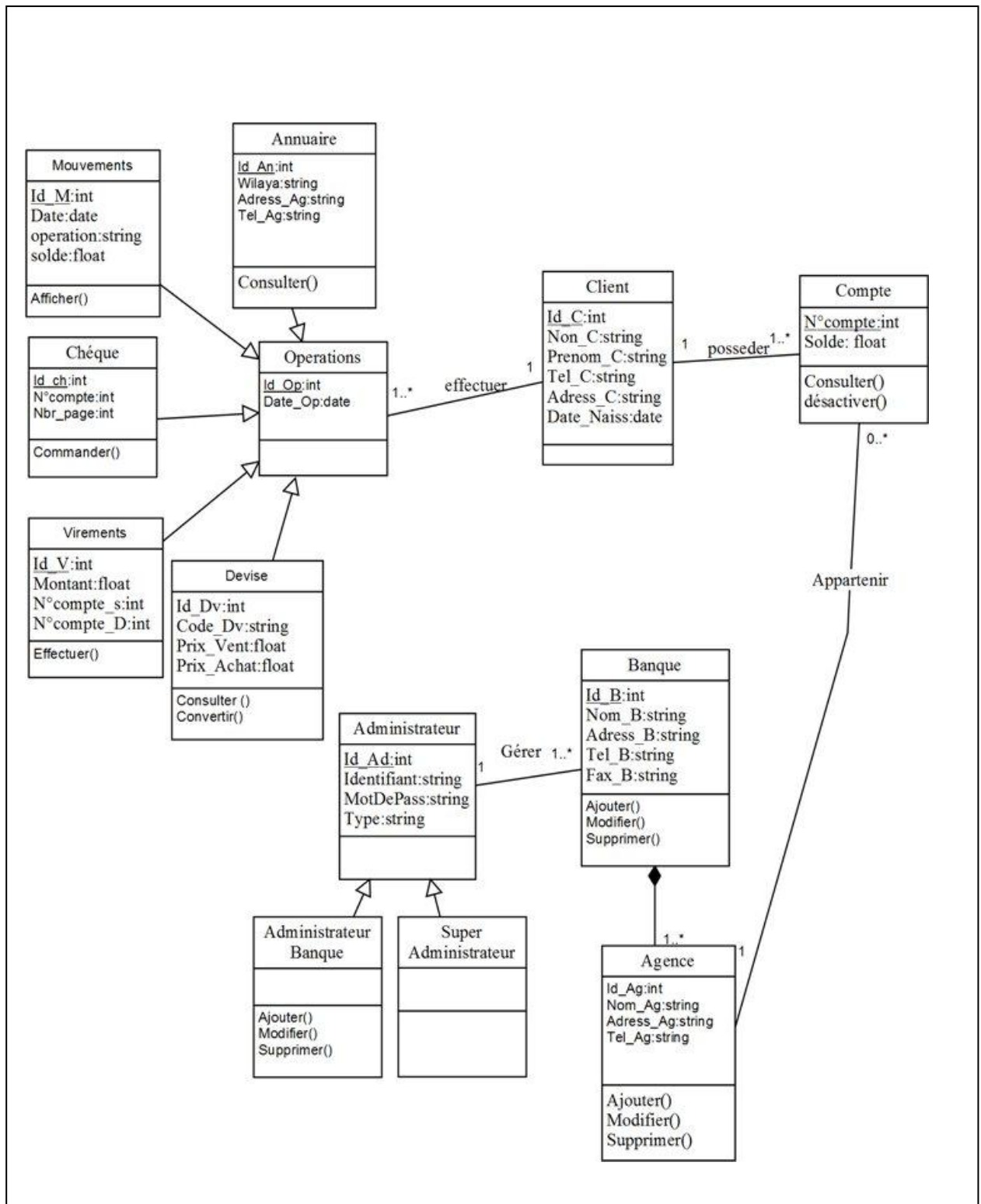


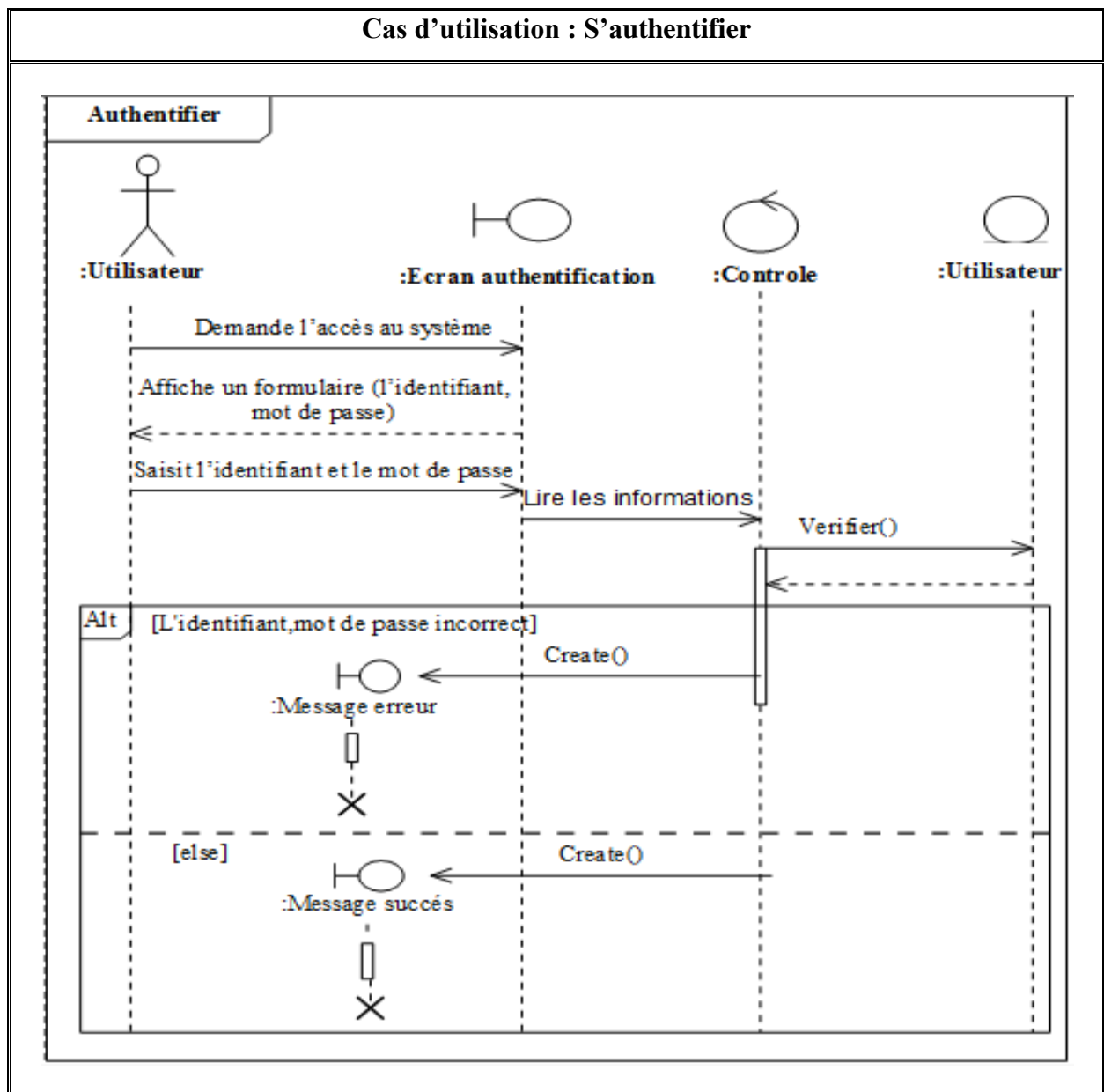
Figure 4.4 : Diagramme de classe

5. Développement du modèle dynamique

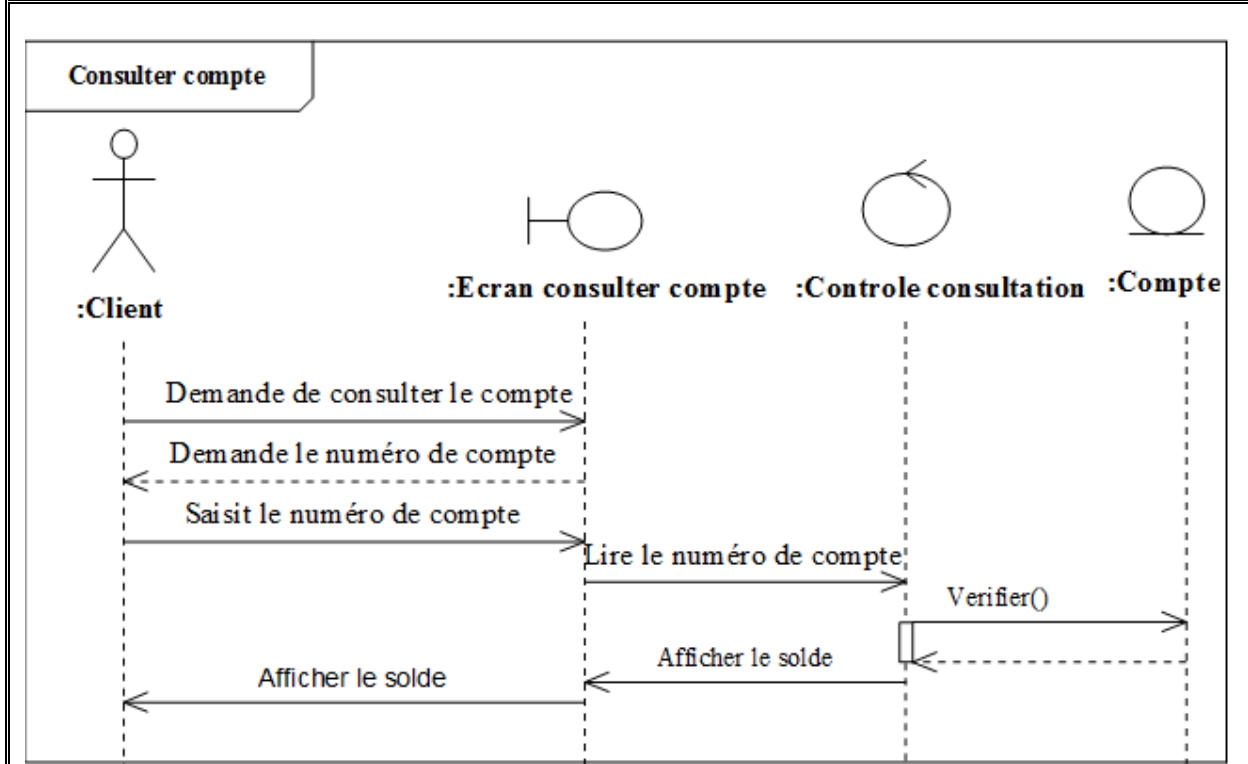
Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative, fortement couplée avec l'activité de modélisation statique. [5]

Le diagramme de séquence est une représentation schématique d'un scénario dans le contexte d'un cas d'utilisation. Il permet de représenter les interactions entre objets en précisant la chronologie des échanges. [4]

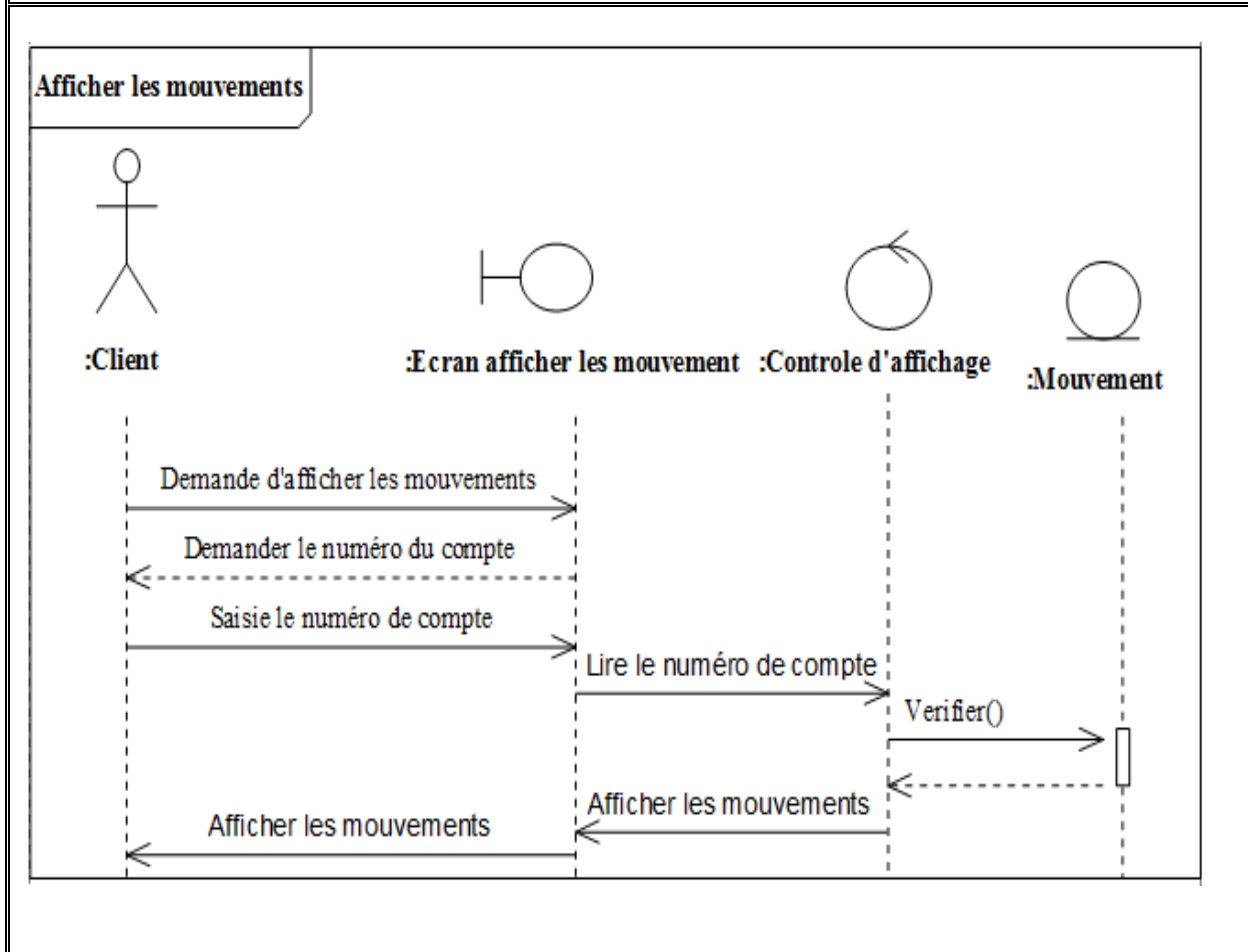
Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il correspond à une sélection d'enchaînement d'un cas d'utilisation. [5]



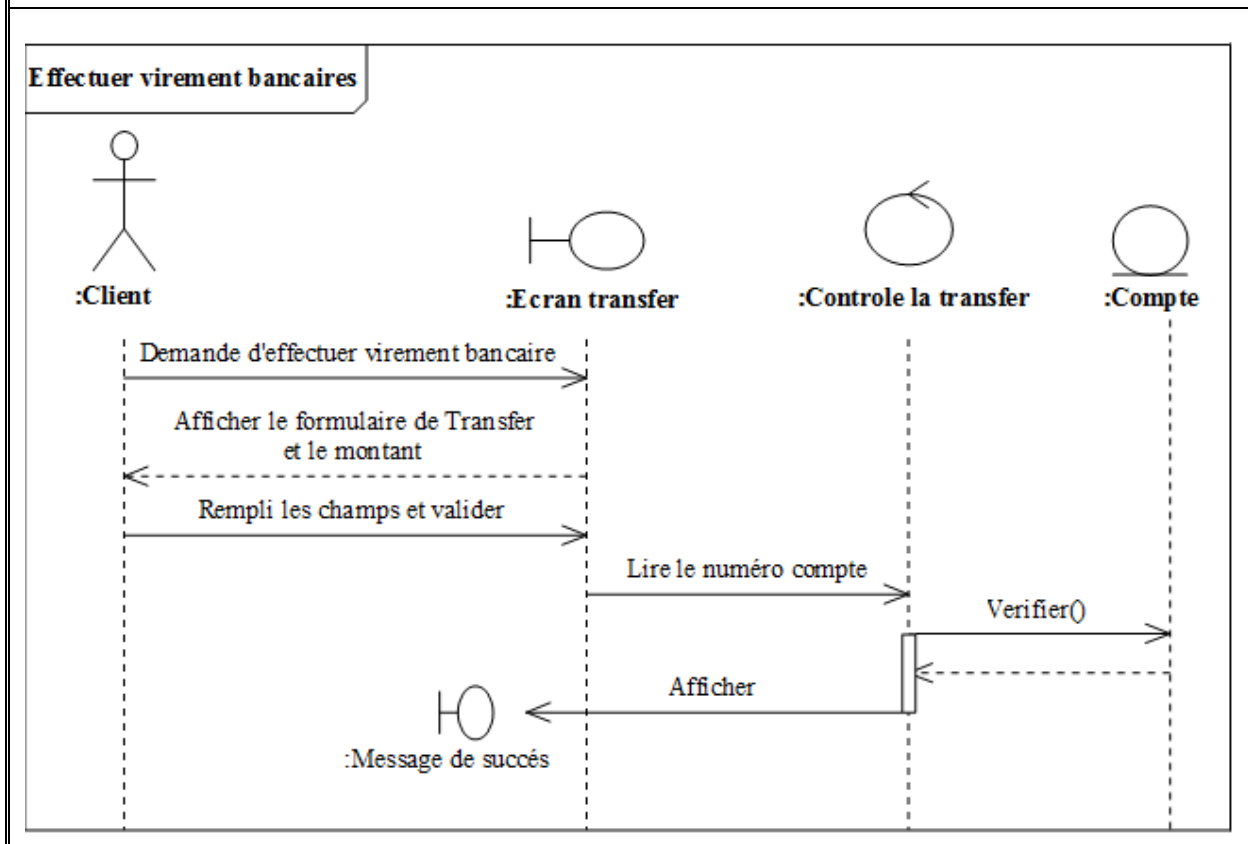
Cas d'utilisation : Consulter compte



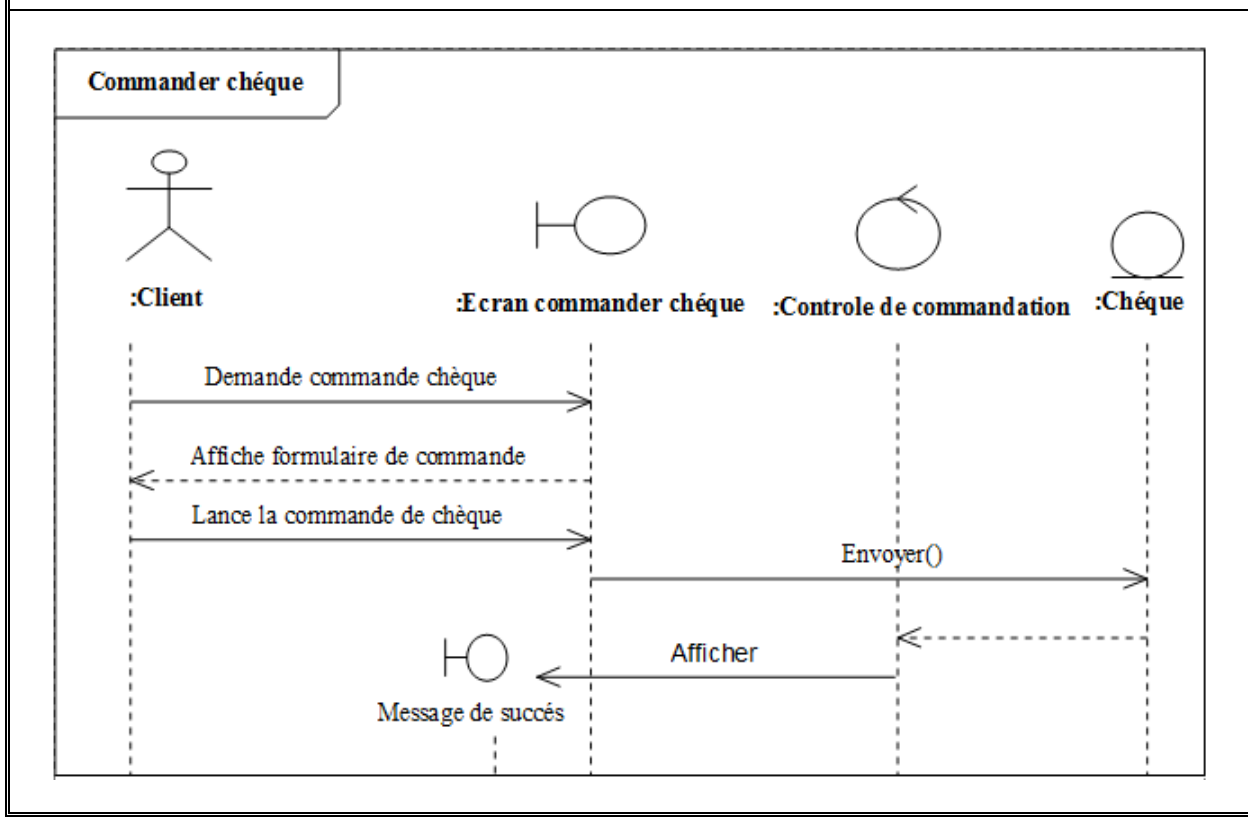
Cas d'utilisation : Afficher les mouvements

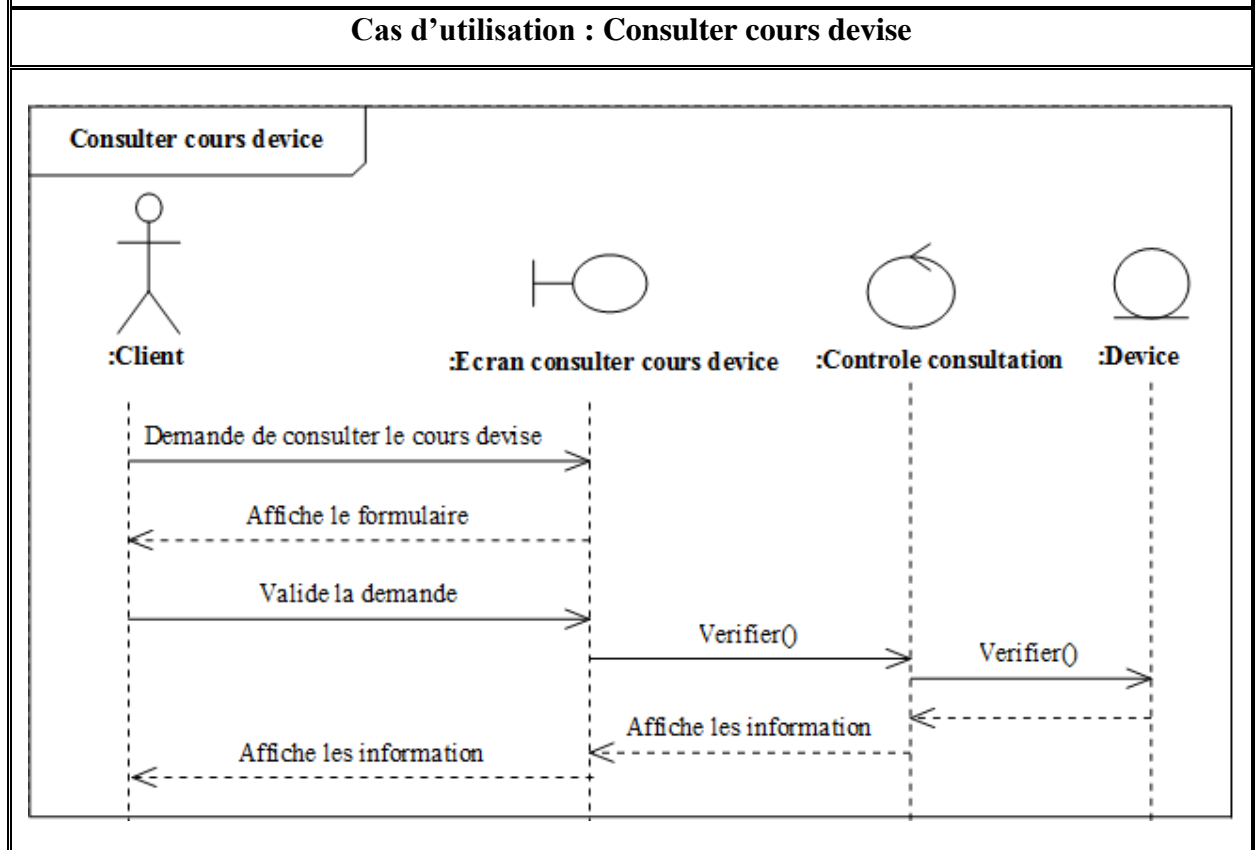
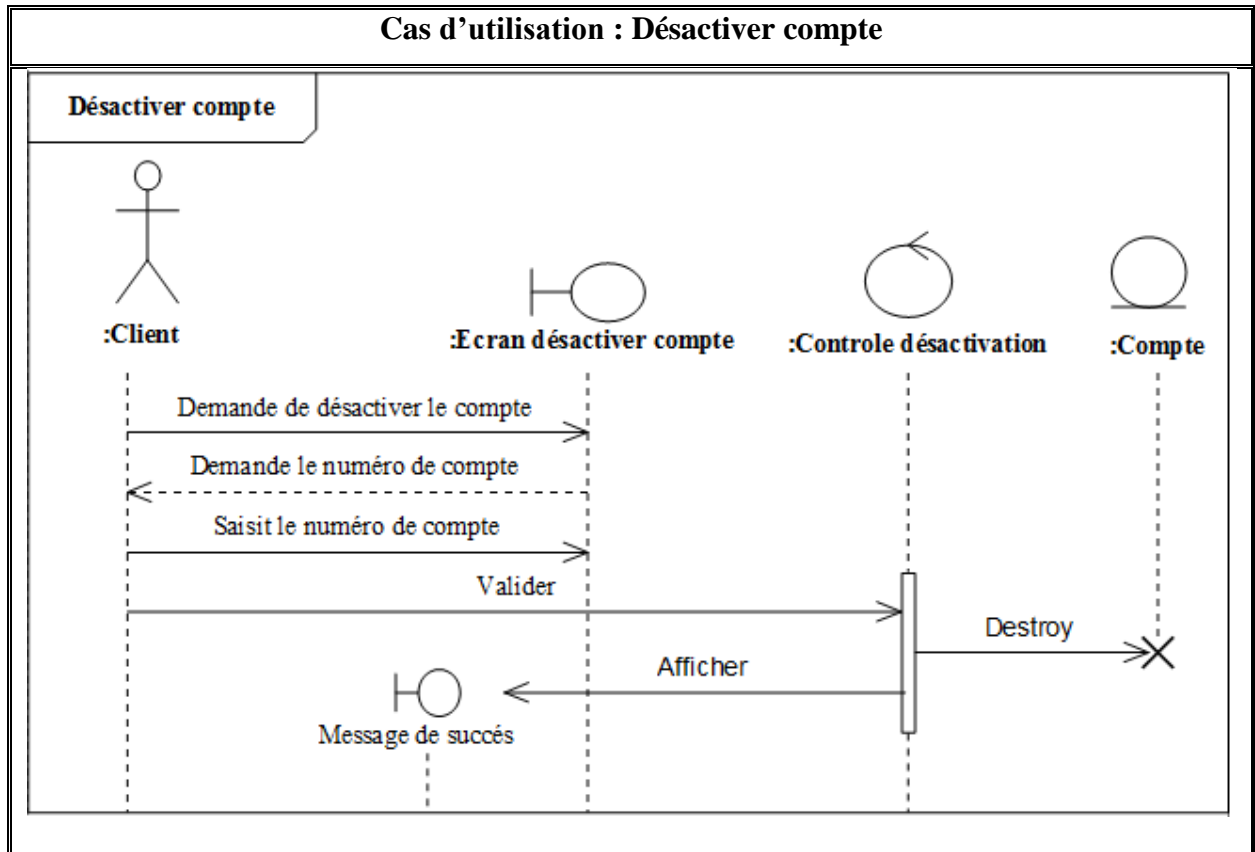


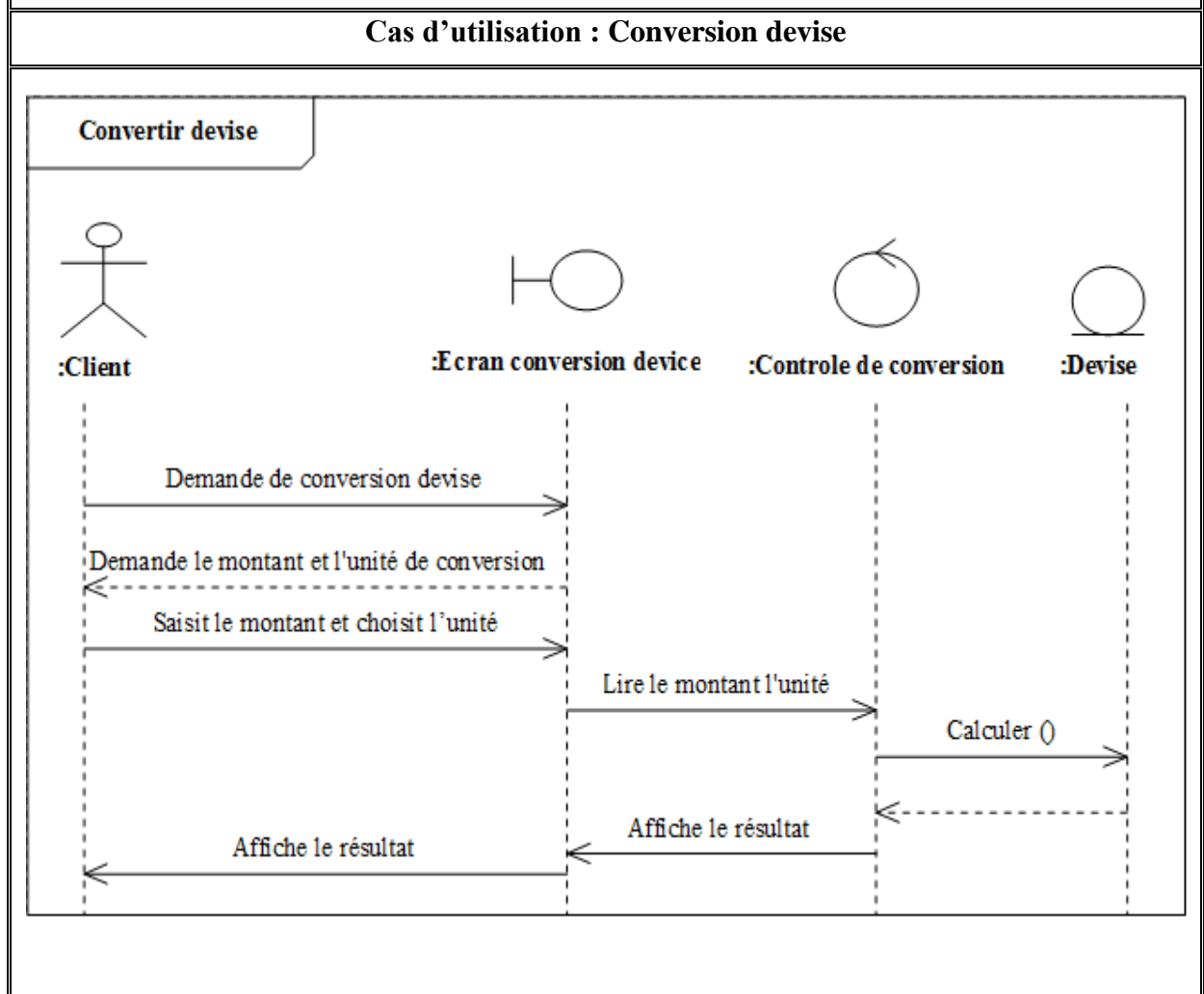
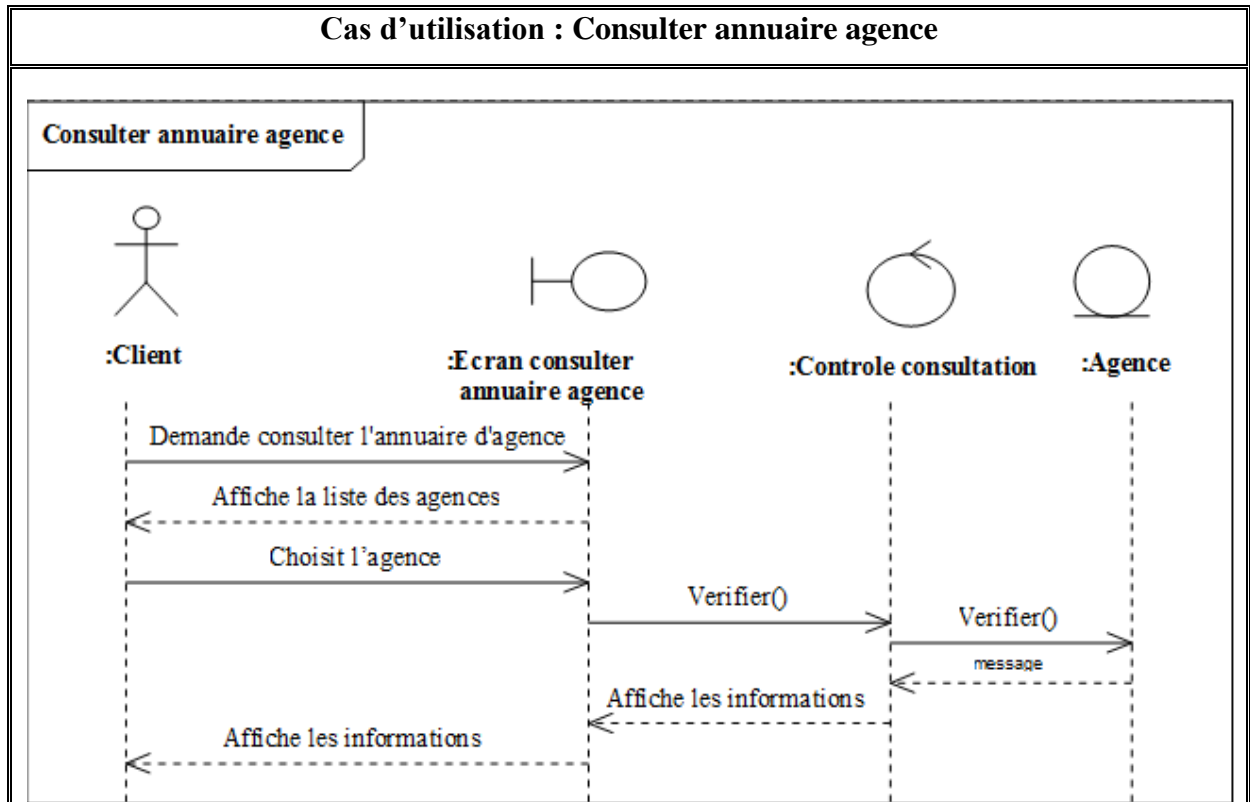
Cas d'utilisation : Effectuer virement bancaires



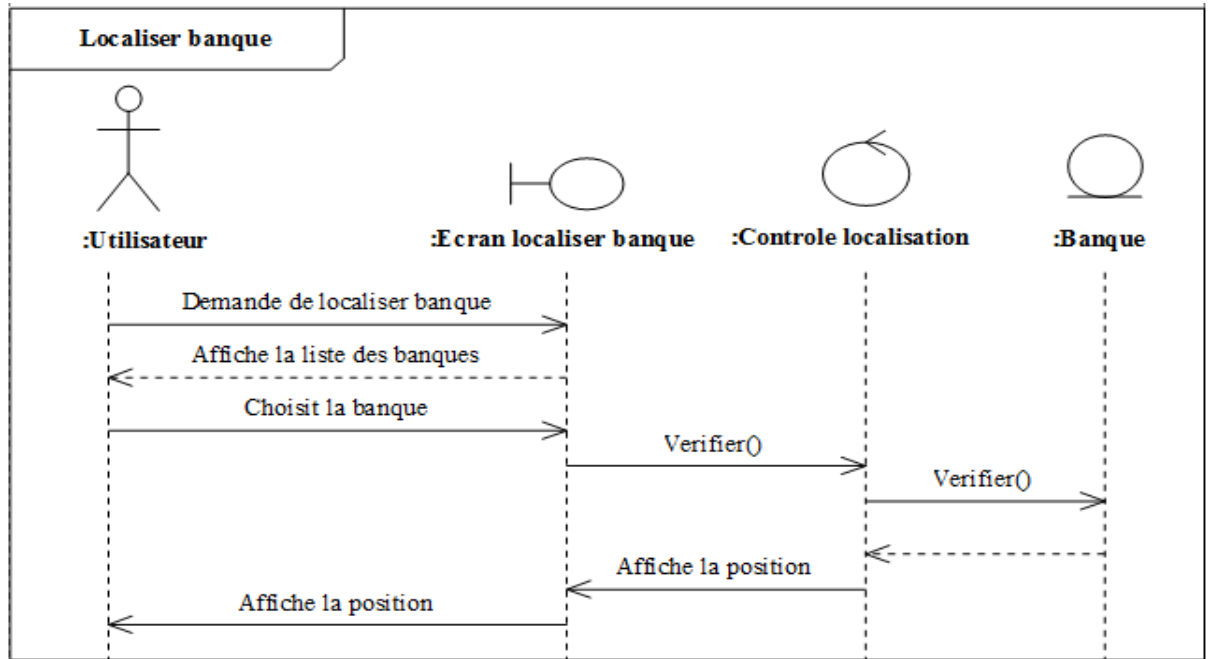
Cas d'utilisation : Commander chèque



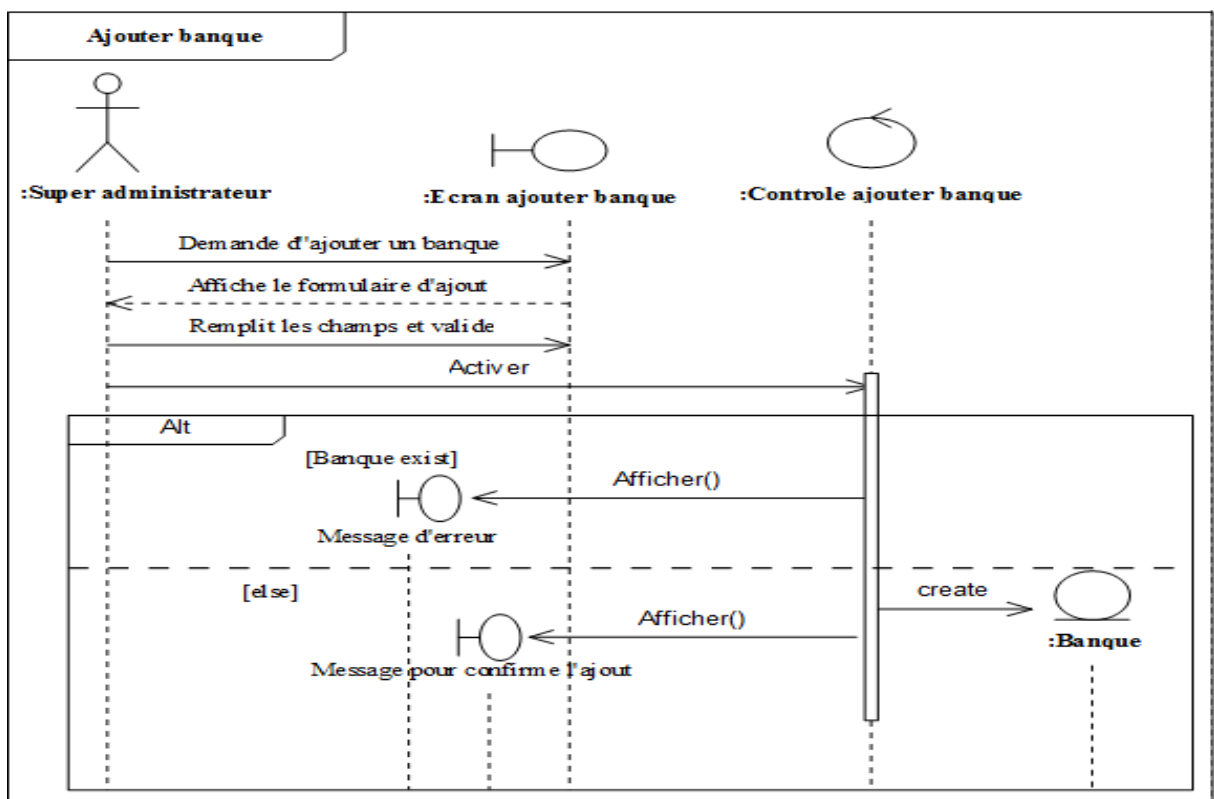




Cas d'utilisation : Localiser banque

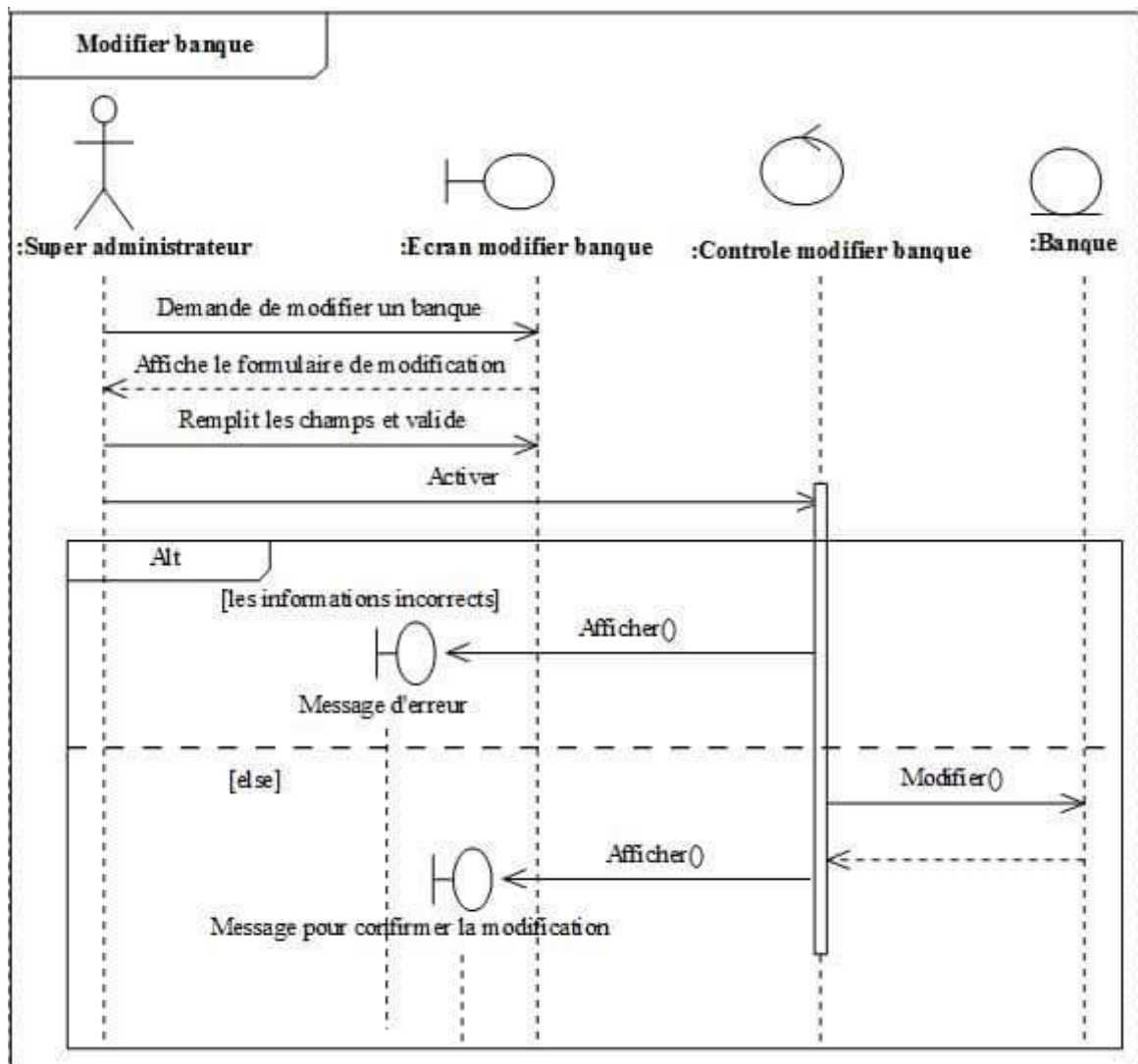


Cas d'utilisation : Ajouter banque



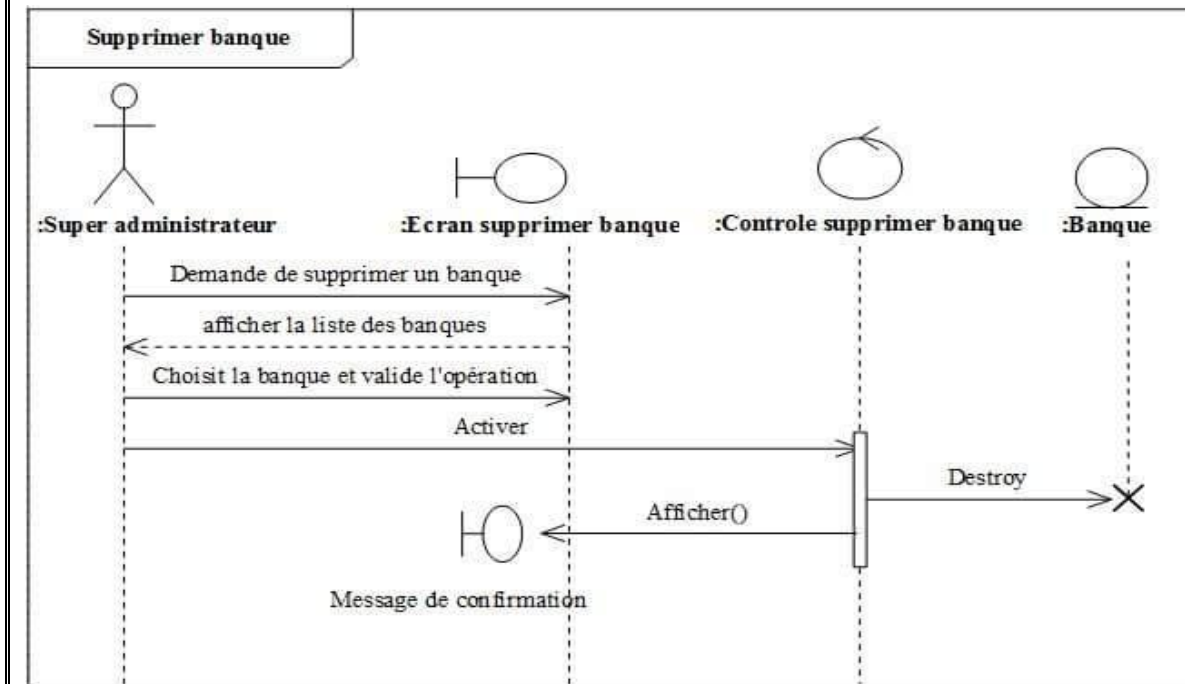
Remarque : « Ajouter administrateur » a le même enchainement que « Ajouter banque »

Cas d'utilisation : Modifier banque



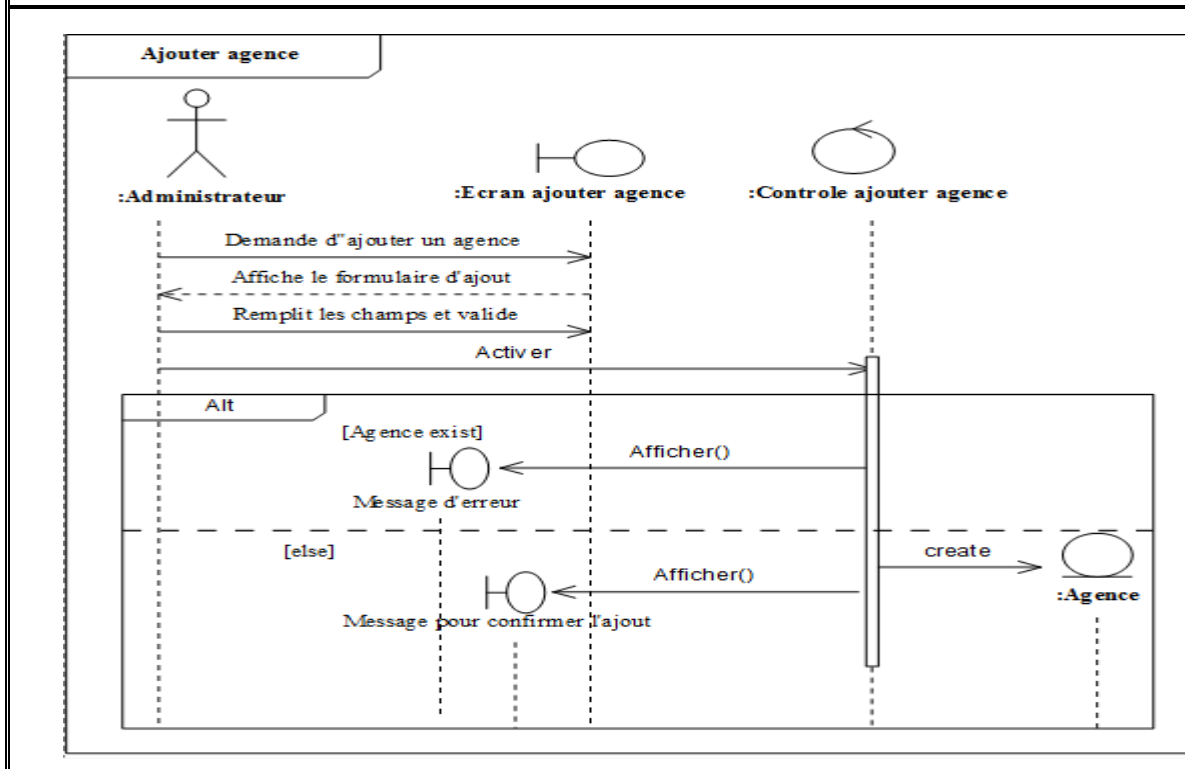
Remarque : « Modifier administrateur » a le même enchainement que « Modifier banque »

Cas d'utilisation : Supprimer banque



Remarque : « Supprimer administrateur » a le même enchainement que « Supprimer banque »

Cas d'utilisation : Ajouter agence



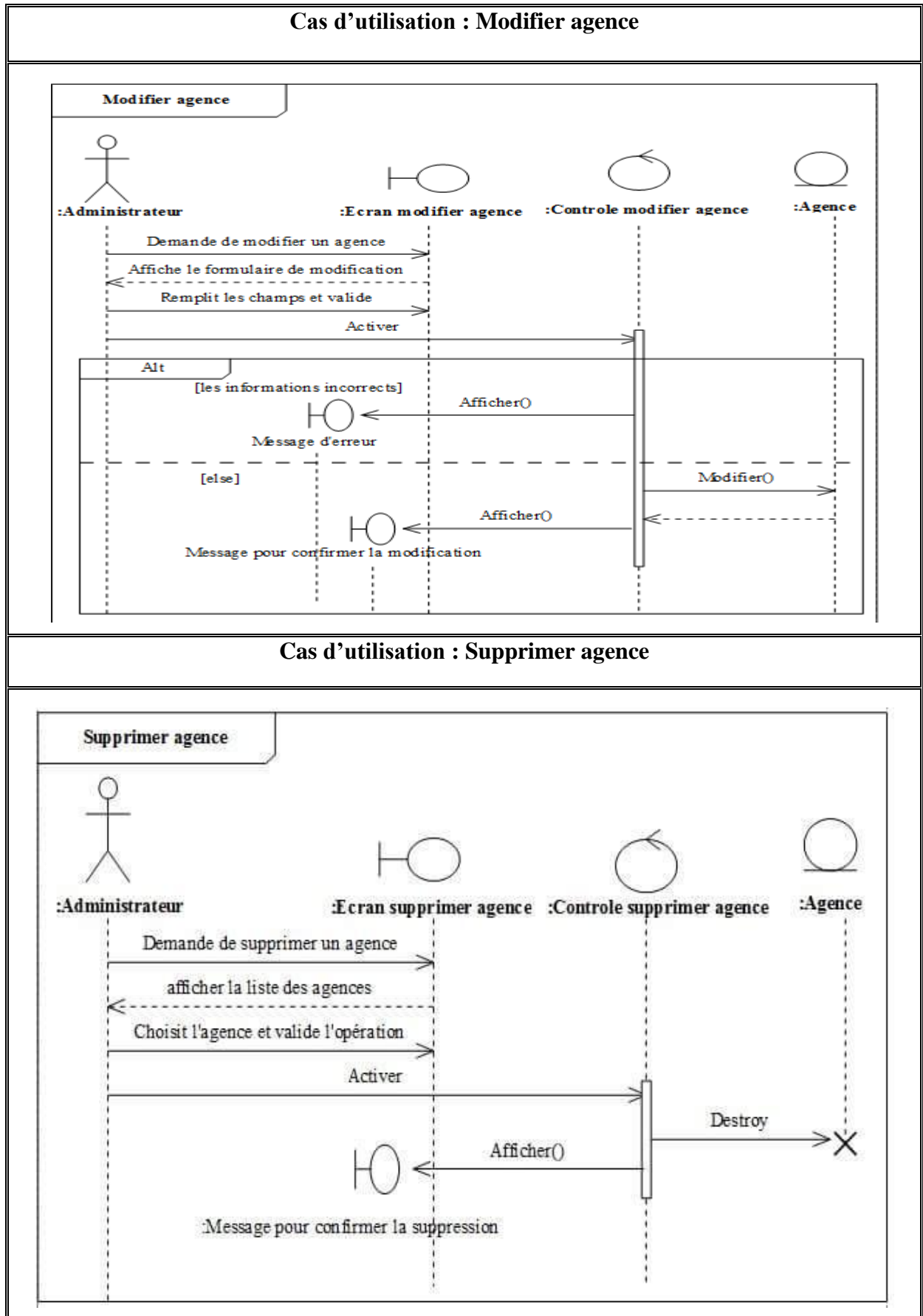


Tableau 4.3 : Diagramme de séquence pour les cas d'utilisation

6. Conclusion

Dans ce chapitre nous avons représenté l'analyse des objets du système, les classes issues. Des besoins fonctionnels sont regroupés en catégories pour organiser le modèle structurel d'analyse. Ce modèle nécessite un travail d'analyse détaillé de la structure des classes. Celui-ci est considéré comme une base pour le développement du modèle dynamique.

Nous allons décrire dans le chapitre suivant la conception du système.



Chapitre 05

Conception

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Conception préliminaire**
- 3. Conception détaillée**
- 4. Règles de transformation**
- 5. Le modèle relationnel**
- 6. Conclusion**

1. Introduction

Nous allons étudier maintenant le rôle d'UML lors de l'étape de conception préliminaire. Les diagrammes UML servent ici plus particulièrement à construire et à documenter la façon dont le développement de la solution doit être organisé.

2. Conception préliminaire

La conception préliminaire consiste à tenir compte des besoins exprimés dans la phase de capture des besoins fonctionnels. [9]

C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. Elle est avant tout affaire d'organisation, elle s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse. [10]

2.1 Situation de la conception préliminaire dans le processus 2TUP

La conception préliminaire est située dans le milieu du processus 2TUP. Elle fusionne les contraintes techniques de la branche technique avec les contraintes fonctionnelles de la branche fonctionnelle.

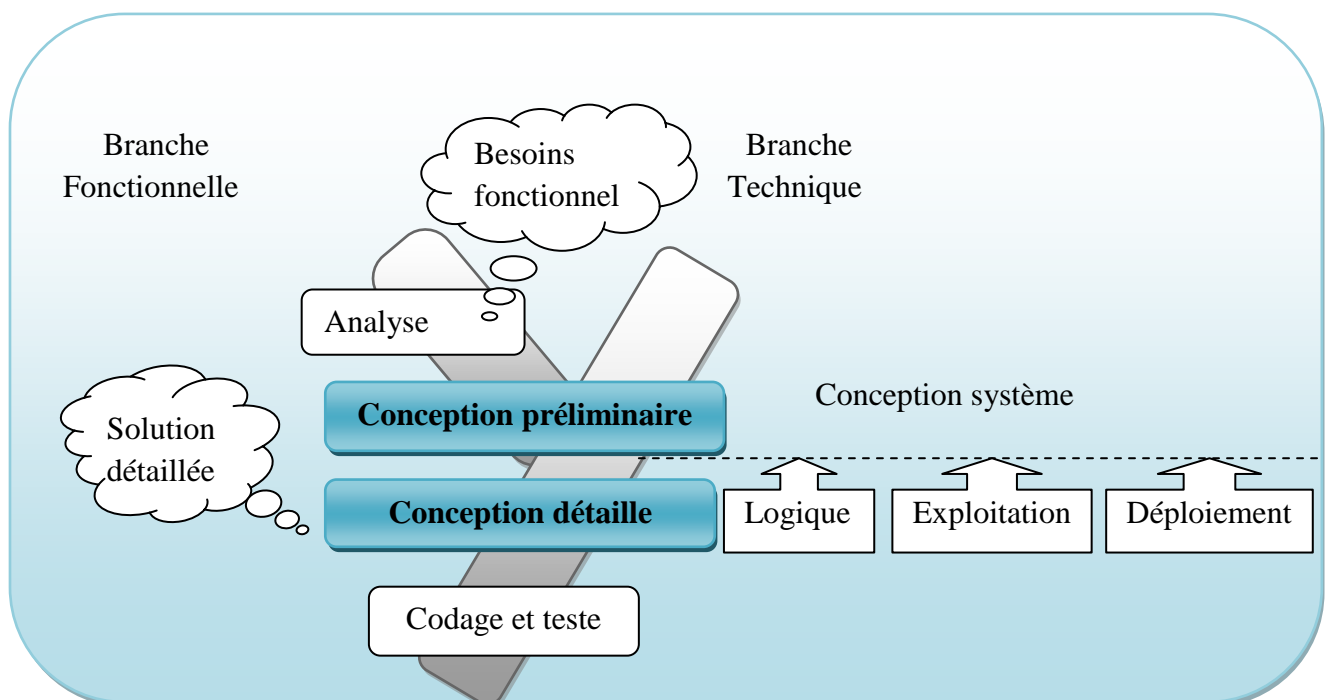


Figure 5.1 : Situation de la conception préliminaire dans 2TUP

2.2 Développement du modèle de déploiement

Le déploiement d'une solution client/serveur se construit sur la définition des postes 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. [11]

- Le modèle de configuration matérielle a été utilisé pour exprimer les contraintes de mise en œuvre au niveau physique.
- Le modèle de déploiement considère plutôt chaque nœud comme un poste de travail, Il exprime la répartition physique des fonctions métiers du système et permet de justifier la localisation des bases de données et des environnements de travail.

Le modèle de déploiement du système est illustré par la figure suivante :

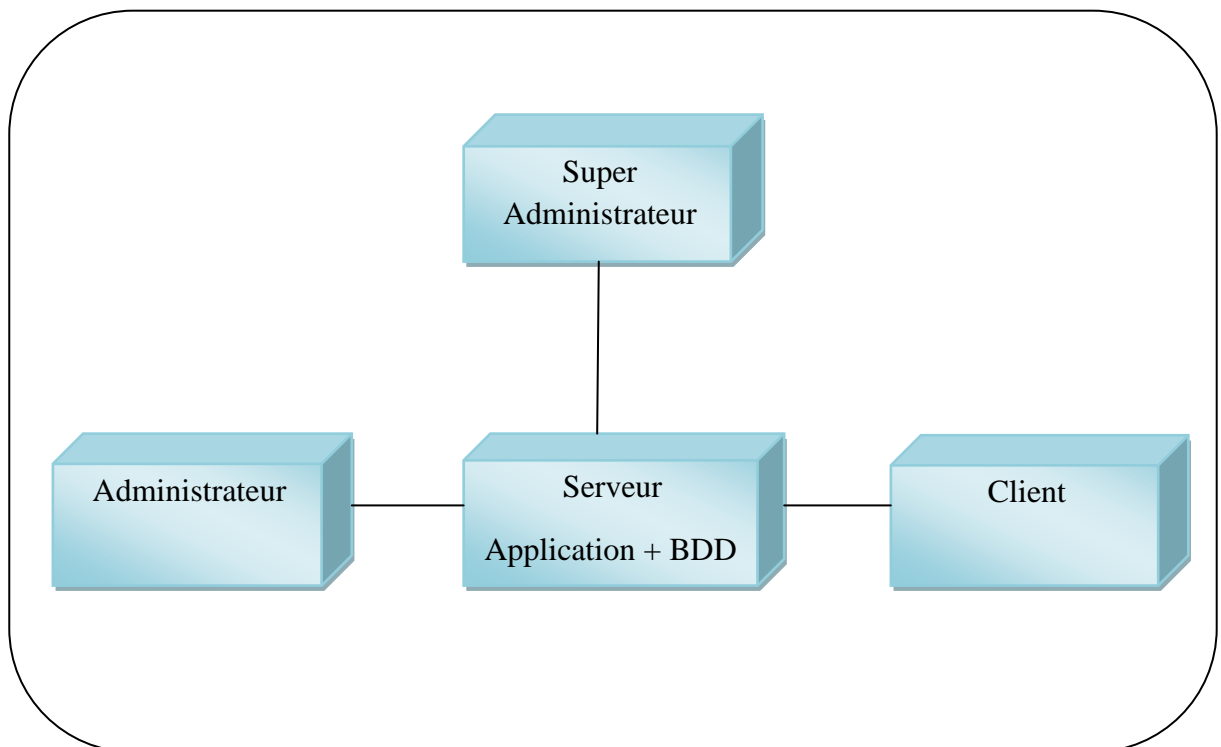


Figure 5.2 : Modèle de déploiement du système

3. Conception détaillée

L'étape de conception détaillée est l'avant dernière étape avant le codage. A cette étape, on commence à raffiner les choix de conception en incluant les spécificités du langage ou environnement de développement. [12]

3.1 Typage des attributs

Cette étape est composée de deux phases la première concerne les attributs qui consiste principalement à définir le type des attributs identifiés dans le chapitre d'analyse et la deuxième concerne les opérations qui consiste à définir les différentes opérations effectuées sur chaque classe. [10]

La conception des attributs des classes du système est présentée dans le tableau suivant :

<table border="1"> <tr><td>Compte</td></tr> <tr><td>N°compte: int Solde : float</td></tr> <tr><td>Consulter () Désactiver ()</td></tr> </table>	Compte	N°compte: int Solde : float	Consulter () Désactiver ()	<table border="1"> <tr><td>Client</td></tr> <tr><td>Id_client :int Nom_C : varchar Prénom_C: varchar Tel_C : varchar Adresse_C :varchar Date_Naiss :date</td></tr> </table>	Client	Id_client :int Nom_C : varchar Prénom_C: varchar Tel_C : varchar Adresse_C :varchar Date_Naiss :date	<table border="1"> <tr><td>Banque</td></tr> <tr><td>Id_Banq: int Nom_B :varchar Tel_B :varchar Adresse_B :varchar</td></tr> <tr><td>Ajouter () Modifier () Supprimer ()</td></tr> </table>	Banque	Id_Banq: int Nom_B :varchar Tel_B :varchar Adresse_B :varchar	Ajouter () Modifier () Supprimer ()	
Compte											
N°compte: int Solde : float											
Consulter () Désactiver ()											
Client											
Id_client :int Nom_C : varchar Prénom_C: varchar Tel_C : varchar Adresse_C :varchar Date_Naiss :date											
Banque											
Id_Banq: int Nom_B :varchar Tel_B :varchar Adresse_B :varchar											
Ajouter () Modifier () Supprimer ()											
<table border="1"> <tr><td>Chèque</td></tr> <tr><td>Id_chéq :int Nbr_page :int</td></tr> <tr><td>Commander ()</td></tr> </table>	Chèque	Id_chéq :int Nbr_page :int	Commander ()	<table border="1"> <tr><td>Devise</td></tr> <tr><td>Id_dev: int Code_dev : varchar Prix_vent : float Prix_achat : float</td></tr> <tr><td>Consulter () Convertir ()</td></tr> </table>	Devise	Id_dev: int Code_dev : varchar Prix_vent : float Prix_achat : float	Consulter () Convertir ()	<table border="1"> <tr><td>Agence</td></tr> <tr><td>Id_Ag :int Non_Ag :varchar Adress :varchar Tel :varchar</td></tr> <tr><td>Ajouter () Modifier () Supprimer ()</td></tr> </table>	Agence	Id_Ag :int Non_Ag :varchar Adress :varchar Tel :varchar	Ajouter () Modifier () Supprimer ()
Chèque											
Id_chéq :int Nbr_page :int											
Commander ()											
Devise											
Id_dev: int Code_dev : varchar Prix_vent : float Prix_achat : float											
Consulter () Convertir ()											
Agence											
Id_Ag :int Non_Ag :varchar Adress :varchar Tel :varchar											
Ajouter () Modifier () Supprimer ()											
<table border="1"> <tr><td>Administrateur</td></tr> <tr><td>Id_Admi: int Identifiant :varchar MotDePasse :varchar Type : varchar</td></tr> <tr><td>Ajouter () Modifier () Supprimer ()</td></tr> </table>	Administrateur	Id_Admi: int Identifiant :varchar MotDePasse :varchar Type : varchar	Ajouter () Modifier () Supprimer ()		<table border="1"> <tr><td>Opérations</td></tr> <tr><td>Id_Op :int Date_Op : date</td></tr> </table>	Opérations	Id_Op :int Date_Op : date				
Administrateur											
Id_Admi: int Identifiant :varchar MotDePasse :varchar Type : varchar											
Ajouter () Modifier () Supprimer ()											
Opérations											
Id_Op :int Date_Op : date											

Mouvement	Annuaire	Virement
Id_Mouv :int	Id_Annu :int	Id_Vir :int
Date : date	Wilaya :varchar	Montant : float
Opération :varchar	Adress : varchar	Comp_S :int
	Tel : varchar	Comp_D: int
Afficher()	Consulter()	Effectuer()

Tableau 5.1 : Typage des attributs des classes du système

4. Règles de transformation

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règle.

- **Transformation des classes**

Chaque classe du diagramme UML devient une relation, les attributs de la classe deviennent attributs de la relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant (Clé primaire).

Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un à la relation (les outils proposent l'ajout de tels attributs).

- **Transformation des associations**

Les règles de transformation des associations dépendent des multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- ✓ **Association 1..***

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. On peut se rappeler cette règle de la manière suivant : la clé de la relation père migre dans la relation fils.

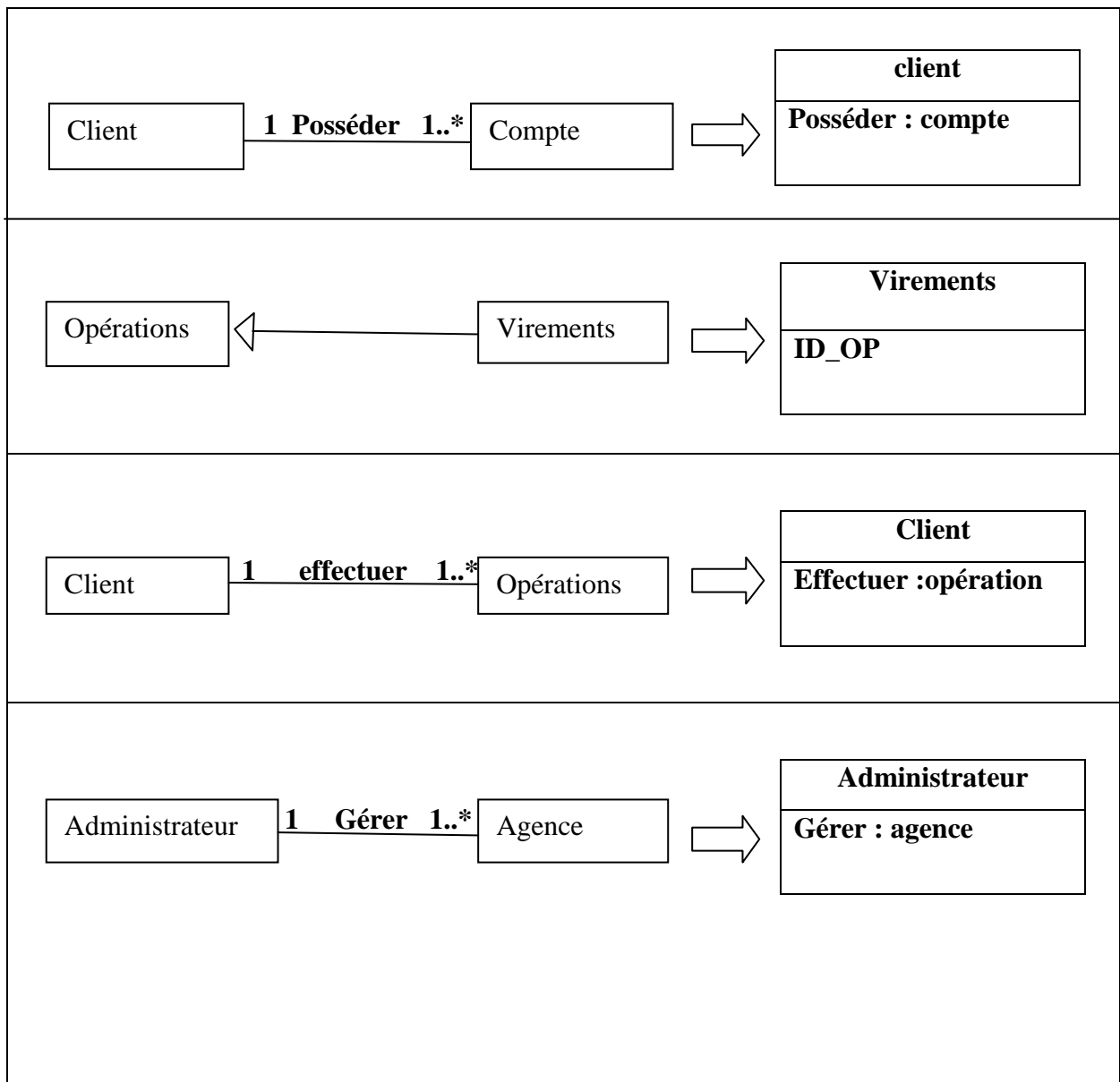
✓ **Association *.* et classes-associations**

L'association/classe-association devient une relation. La clé primaire de cette relation est la concaténation des clés primaires des classes connectées à l'association. Chaque attribut devient clé étrangère. Les attributs de l'association /classe-association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

✓ **Association 1-1**

Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la Classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

4.1 Traduction des associations du système



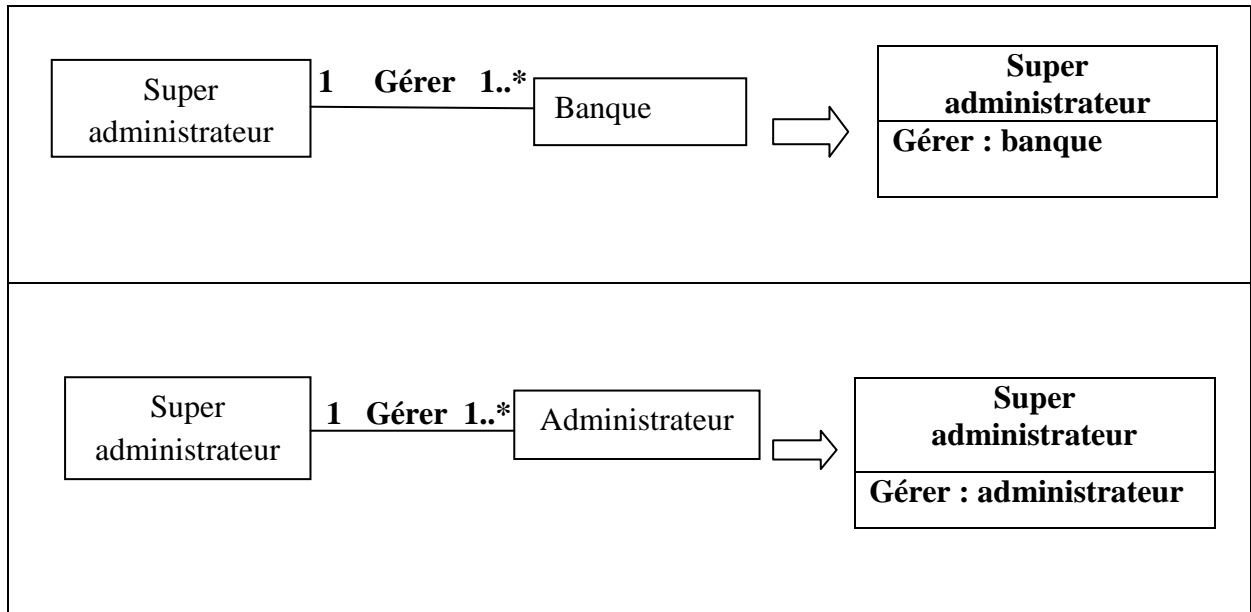


Tableau 5.2 : Traduction des associations

4.2 Passage du modèle objet au modèle relationnel

- L'utilisation d'un SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Les deux structures ayant des analogies, les équivalences exprimées au tableau sont généralement utilisées.
- Une classe définit une structure de données à laquelle souscrivent des instances ; elle correspond donc à une table au modèle relationnel : chaque attribut, donne lieu à une colonne, chaque instance stocke ses données dans une ligne (*T-uplet*) et son OID sert de clé primaire. [13]

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type composé	Colonnes ou clé étrangère
Instance	T-uplet
OID	Clé primaire
Association	Clé étrangère ou table de liens
Héritage	Clé primaire identique sur plusieurs tables

Tableau 5.3 : Equivalence entre les concepts objets et relationnels.

5. Le modèle relationnel

Le diagramme suivant montre le modèle relationnel de notre base de données

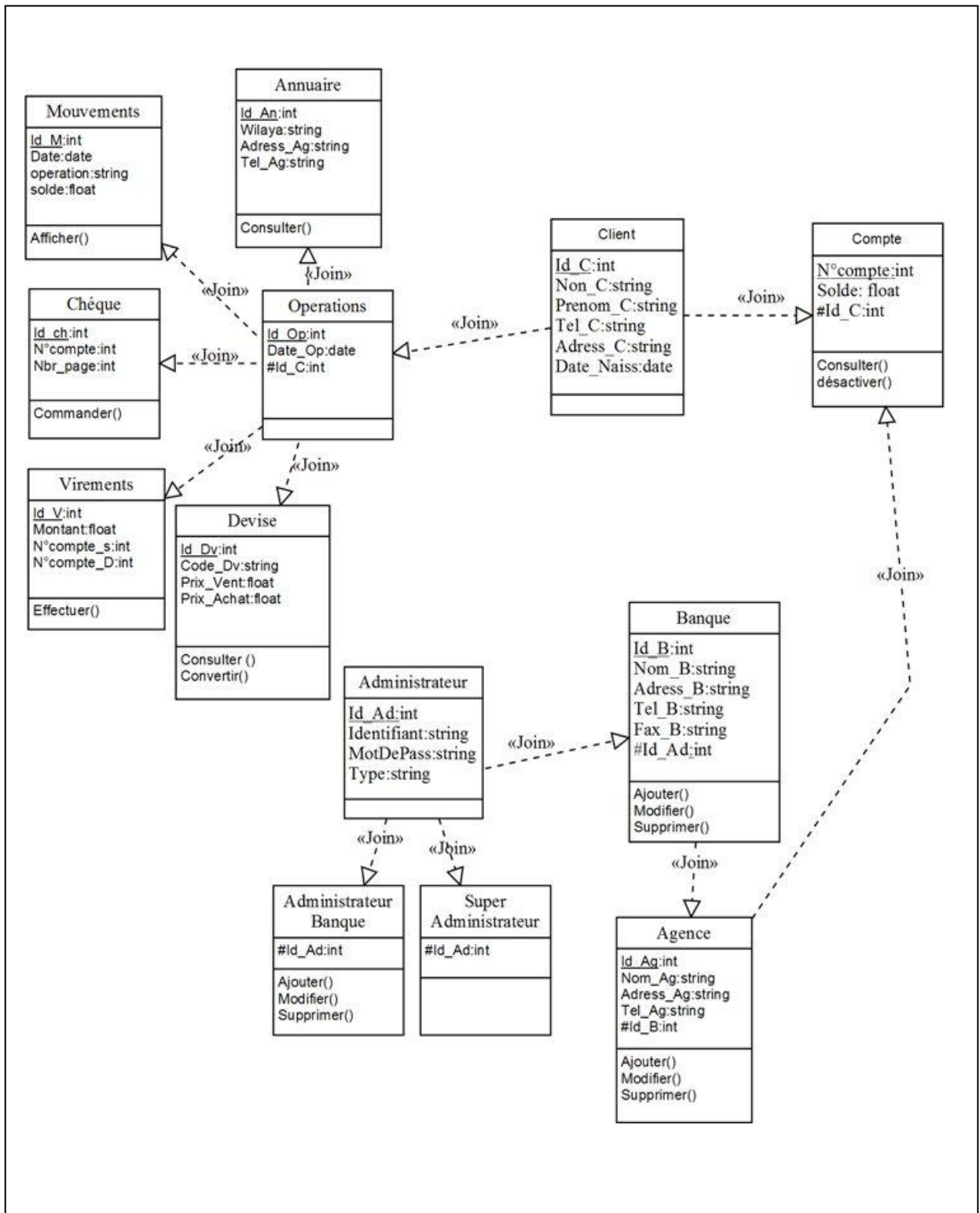


Figure 5.3 : Le modèle relationnel du système

6. Conclusion

Dans ce chapitre nous avons étudié la conception du système. La conception préliminaire nous a permis de spécifier la configuration matérielle de notre futur système. La conception détaillée nous a permis d'élaborer un modèle relationnel représentant les différentes tables de la base de données du système et de montrer les différentes jointures existantes entre ces différentes tables, donc il nous reste de passer à la phase de réalisation de notre futur système.



Chapitre 06

Réalisation

Au sommaire de ce chapitre

- 1. Introduction**
- 2. Présentation des technologies utilisées**
- 3. Interfaces de l'application**
- 4. Conclusion**

1. Introduction

La phase de réalisation consiste à passer du résultat de la conception détaillée à un ensemble de programmes ou de composants programmation.

Dans ce chapitre, nous présentons la partie réalisation et la mise en œuvre de notre travail. Pour cela, nous présentons, en premier lieu, l'environnement de travail et les outils de développement utilisés. En second lieu, nous élaborons un manuel explicatif d'utilisation de notre logiciel.

2. Présentation des technologies utilisée

2.1 Développement d'application pour Android

2.1.1 Langage de programmation java

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation. Lors de la création du langage java, il avait été décidé que ce langage devait répondre aux 5 objectifs suivants : Utiliser une méthode orientée objet, permettre à un même programme d'être exécuté sur plusieurs systèmes d'exploitation différents. Pouvoir utiliser de manière les réseaux informatiques, pouvoir exécuter du code distant de manière sûre, être facile à utiliser et posséder les points forts des langages de programmation orientés objet comme C++. [14]

2.1.2 Le SDK Android

C'est un ensemble d'outils permettant aux développeurs de créer des applications pour Android. Le SDK est disponible en téléchargement libre. Android SDK est un outil indispensable pour développer des applications Android. Un SDK (Software Development Kit) en général est un kit de développement ou trousse de développement logiciel. C'est un ensemble d'outils permettant aux développeurs de créer des applications de type défini (par exemple pour iOS, Android ou Symbian). [15]

2.1.3 JDK

Le Java Development Kit (JDK) désigne un ensemble des bibliothèques logicielles de base du langage de programmation Java, ainsi que l'environnement dans lequel le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle Java (JVM) puisse l'interpréter. [16]

2.1.4 L'environnement de développement Eclipse

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM) et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. [17]

2.1.5 ADT pour Eclipse

Android Development Tools (ADT) est un plugin pour l'IDE Eclipse qui est conçu pour vous donner un environnement puissant, intégré dans lequel pour construire des applications Android. [18]

2.1.6 API Google

L'API Google permet d'afficher le Map, sur le téléphone mobile. Le Map (Google Map) est un service gratuit de cartographie en ligne. Le service a été créé par Google. Ce service permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de vue sont disponibles : une vue en plan classique, avec nom des rues, quartier, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier. [19]

2.1.7 MyPublicWifi

Logiciel gratuit pour transformer votre PC en point d'accès WiFi. MyPublicWiFi permet de transformer n'importe quel ordinateur doté d'une carte Wi-Fi et de Windows 7/8/10 en point d'accès Wi-Fi. Il devient ainsi possible de connecter plusieurs dispositifs à un ordinateur en même temps sans passer par un véritable point. MyPublicWiFi peut aussi restreindre l'accès à des serveurs spécifiques ou empêcher l'utilisation de certains services Internet (par exemple des programmes de partage de fichiers). MyHotspot enregistre toutes les pages URL consulté. Vous pouvez vérifier quel client a consulté quelle adresse URL. [20]

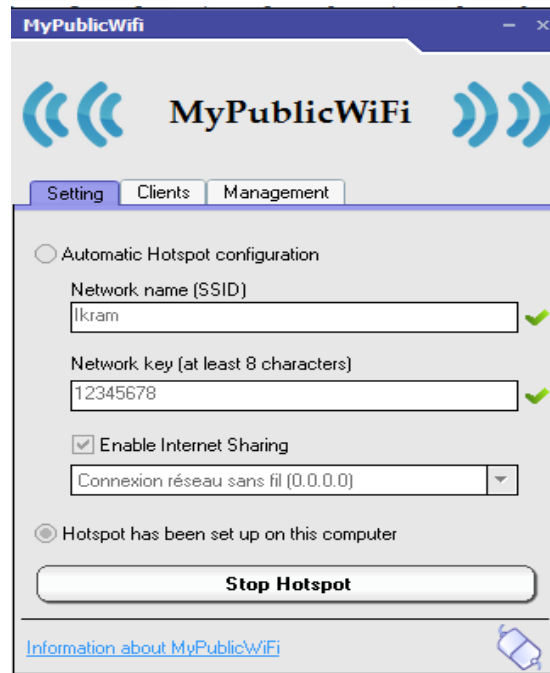


Figure 6.1 : Application MyPuplicWifi

2.1.8 Emulateur

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth. [21]

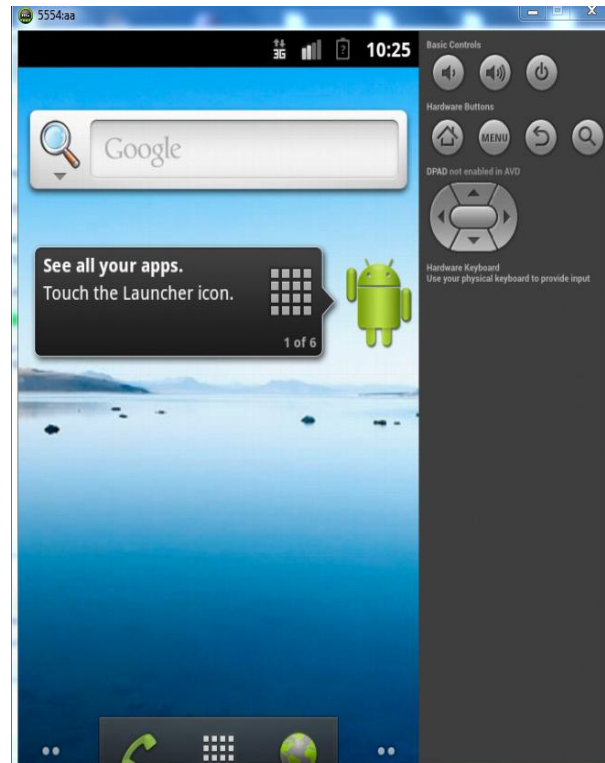


Figure 6.2 : Emulateur Android

2.2 Développement d'application web

2.2.1 WampServer

Est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données. [22]

2.2.2 MySQL

Est un système de gestion de base de données. Son rôle est de stocker et de gérer une grande quantité de données en les organisant sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL. On ne s'occupe plus alors de la manière dont les données sont stockées sur le disque dur, de simples instructions permettent d'ajouter, de supprimer, de mettre à jour et surtout de rechercher des données dans une base de données. [23]

2.2.3 PHP

Est un langage de script. Il permet, de décrire et de créer des pages web, au travers desquelles l'utilisateur peut échanger des informations avec le serveur; c'est ce qu'on appelle

des pages web dynamiques, il permet aussi un affichage dynamique d'information, c'est-à-dire que le texte affiché peut dépendre de variables. Les instructions PHP sont généralement contenues dans des fichiers d'extension PHP. Ces fichiers peuvent contenir du HTML, entremêlé avec le code PHP. Quand un navigateur demande un tel fichier, le serveur Apache exécute les instructions PHP, qui produisent une page HTML. Une fois la page HTML générée, le serveur la renvoie au navigateur, qui ne voit qu'une page HTML. Ses principaux atouts sont :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML ;
- La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, etc) ;
- L'intégration au sein de nombreux serveurs web (Apache...etc.). [24]

2.2.4 Feuilles de style (CSS)

En anglais (Cascading Style Sheets, d'où l'acronyme CSS), elles permettent de spécifier, indépendamment du document HTML lui-même, les attributs de présentation. Au lieu de qualifier chaque balise par un ensemble d'attributs définissant sa présentation graphique, et d'appliquer répétitivement les mêmes attributs à toutes les balises identiques, les feuilles de style définissent ces attributs dans un document séparé, et permet au navigateur d'utiliser le style approprié. [23]

2.2.5 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. [25]

3. Interfaces de l'application

Nous présentons dans ce chapitre quelques interfaces principales de notre application qui illustrent les différents cas d'utilisation déjà vus dans le chapitre étude préliminaire.

3.1 Interface serveur

3.1.1 Interface Super Administrateur/Administrateur

La première interface consultée après l'exécution de notre application, le super administrateur ou l'administrateur s'authentifier (pseudo et mot de passe).



Figure 6.3: Interface Super Administrateur/ Administrateur

3.1.2 Espace super administrateur

Cette espace réservé au super administrateur ; pour offrir le privilège de la mise à jour des données de la base.

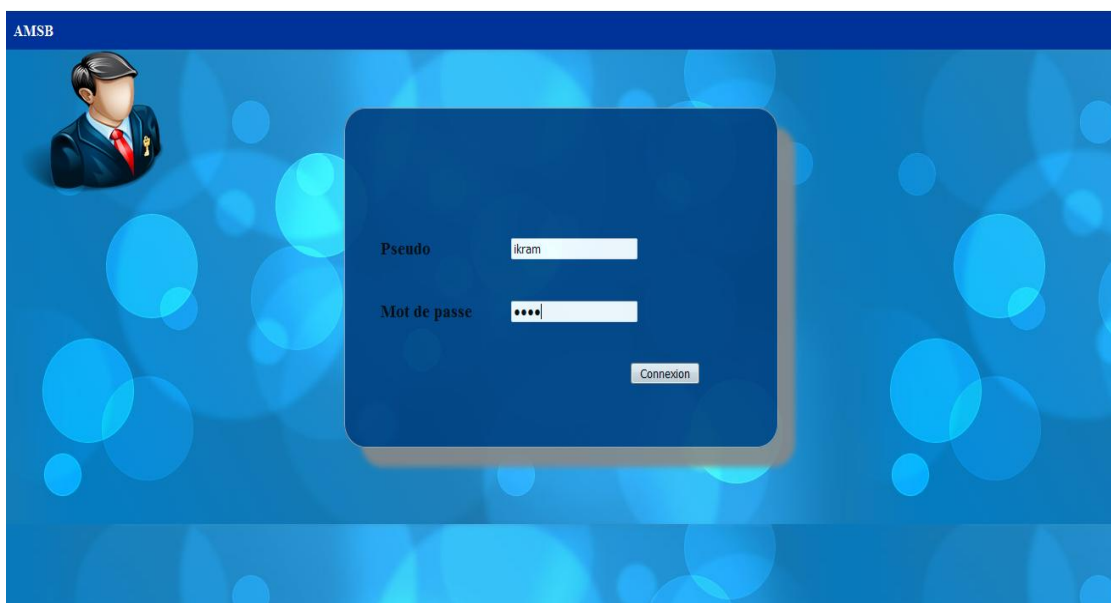


Figure 6.4: Page d'authentification de super administrateur

Après que le super administrateur saisit le pseudo et le mot de passe, la page d'accueil de l'administrateur s'affiche.

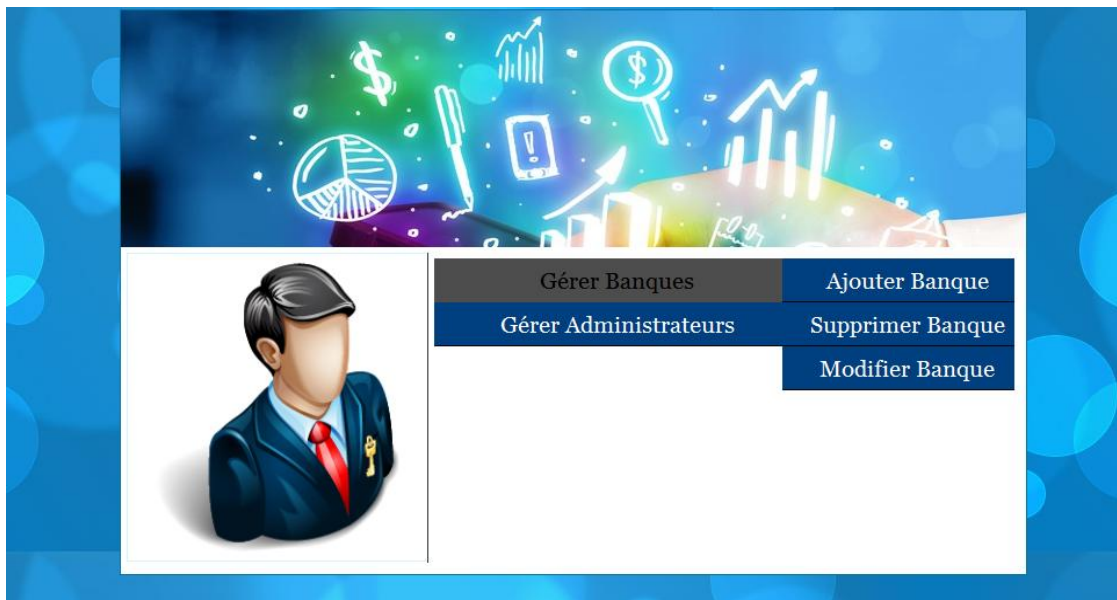


Figure 6.5 : Page d'accueil de super administrateur

Pour la gestion des banques : le super administrateur peut ajouter, modifier et supprimer les banques.

3.1.3 Ajouter banque

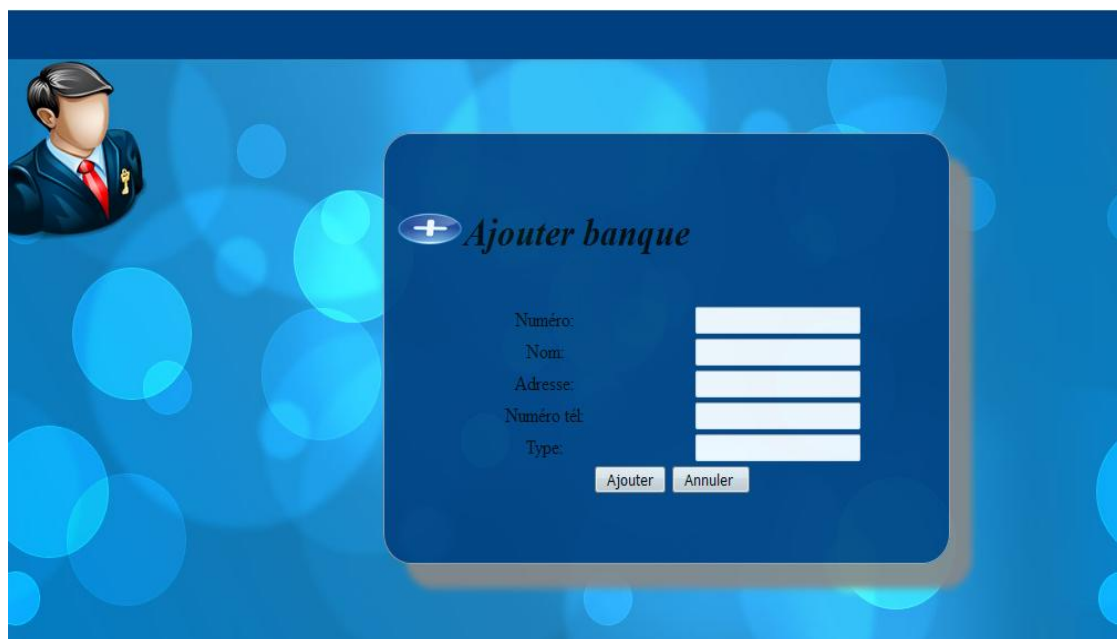


Figure 6.6 : Page ajouter banque

3.1.4 Supprimer banque

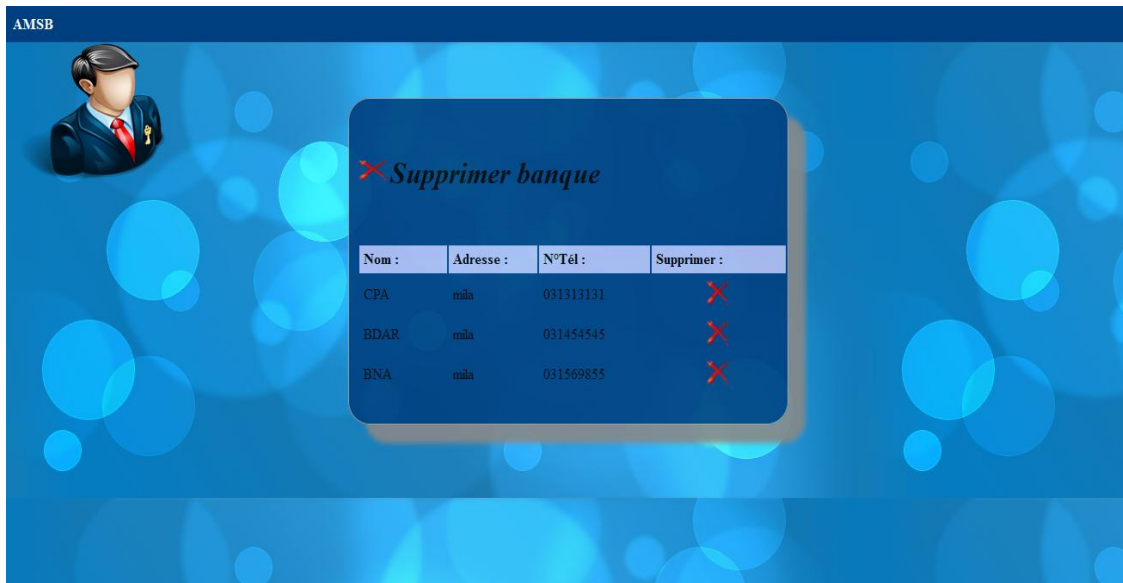


Figure 6.7 : Page supprimer banque

3.1.5 Modifier banque

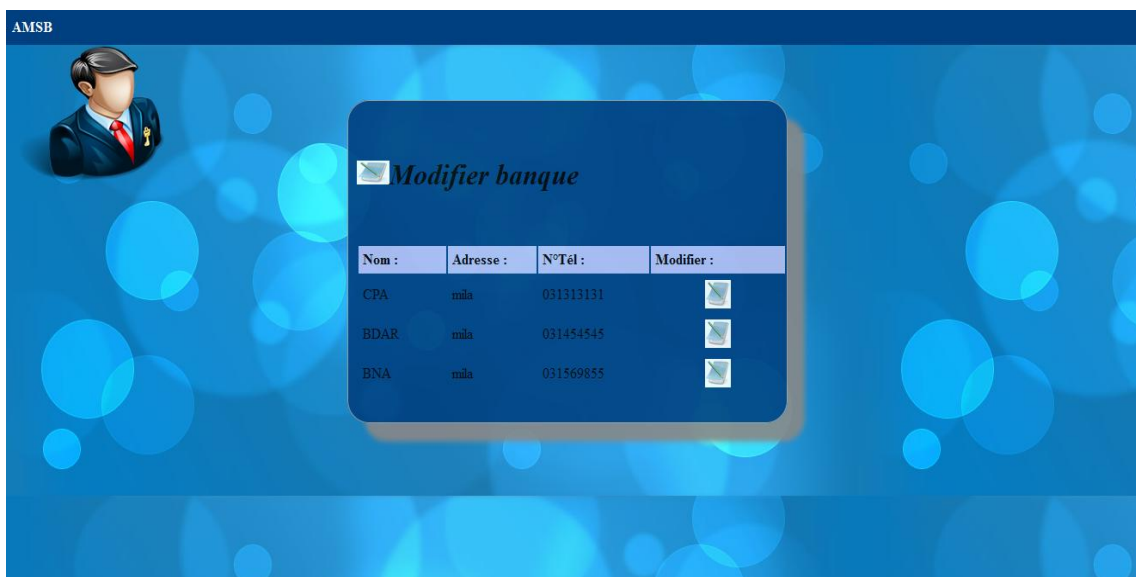


Figure 6.8 : Page modifier banque

Aussi pour la gestion des administrateurs : le super administrateur peut ajouter, modifier et supprimer les administrateurs.

3.2 Interface client

Dans cette partie, nous allons présenter quelques cas d'utilisations, sous forme d'un guide utilisateur.

Cette page est la page d'accueil qui sera affichée pour l'utilisateur. À partir de cette page, l'utilisateur peut accéder au menu principal.



Figure 6.9 : Interface d'accueil

Nous avons décidé de faire une page assez simple en termes de contenu et d'informations. L'utilisateur peut accéder directement à quelques services tels que le service de consultation de cours de devises, consulter l'annuaire des agences, convertir la devise et ainsi il peut accéder au service de localisation de la banque.

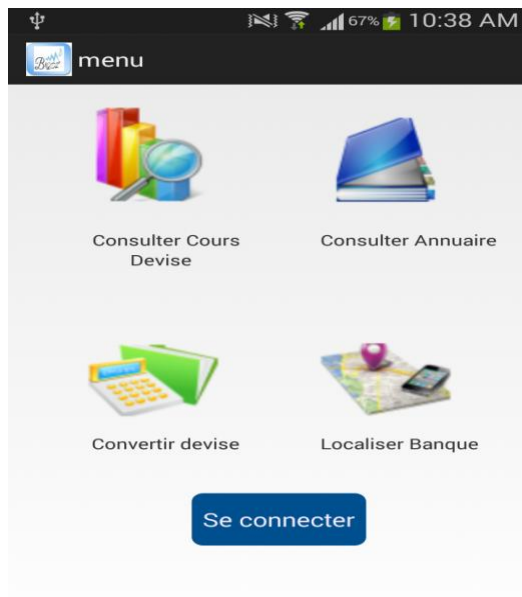


Figure 6.10 : Menu principal (page pour l'utilisateur de l'application)

Dans l'écran « menu » si l'utilisateur choisit « Consulter annuaire », l'interface de cette page est organisée d'une manière lisible qui affiche les informations des agences à l'utilisateur.

Ces informations affichées sont :

- Nom de l'agence,
- Adresse ;
- Numéro de téléphone (fax).

Comme suit :

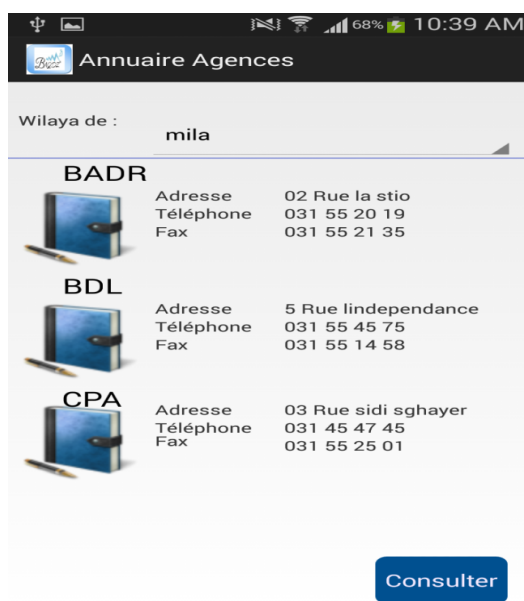
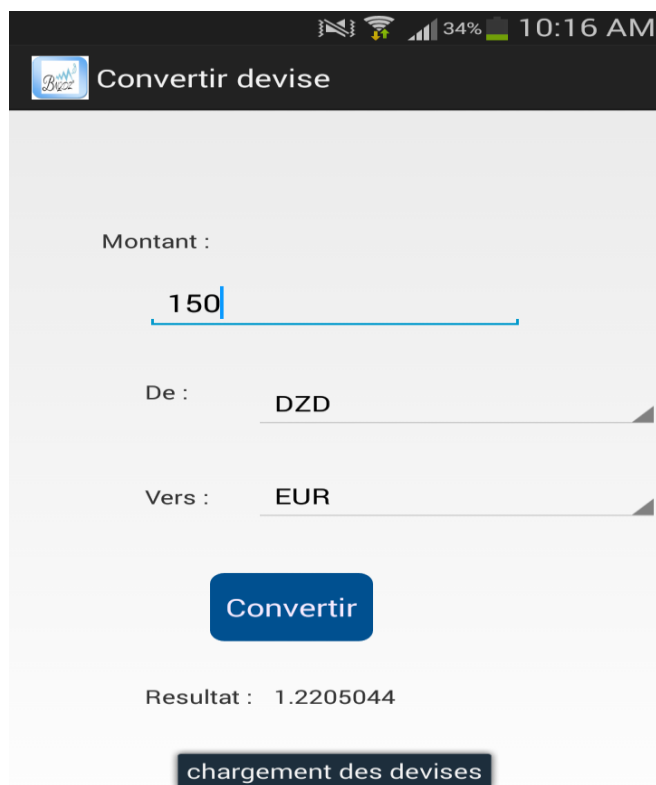


Figure 6.11 : Annuaire agences

Dans l'écran « menu » si l'utilisateur choisit « Convertir devises », d'abord il saisit le montant et ensuite choisit les devises source et cible du formulaire et valide la conversion de billet de banque.



Montant :
150

De : DZD

Vers : EUR

Convertir

Resultat : 1.2205044

chargement des devises

Figure 6.12 : Convertir devises

Dans l'écran « menu » si l'utilisateur choisit « Consulter cours devises », l'interface de cette page est organisée d'une manière lisible qui affiche les cours des devises.

	Code	Prix Achat	Prix Vente
	EUR	130.42	122.9
	USD	107.63	114.2
	SAR	30.46055	28.69
	TND	54.33038	54.8798

[Consulter](#)

Figure 6.13 : cours devise

Dans l'écran « menu » si l'utilisateur choisit « Localiser banque », l'interface de cette page est organisée d'une manière lisible qui affiche la position de la banque choisie.

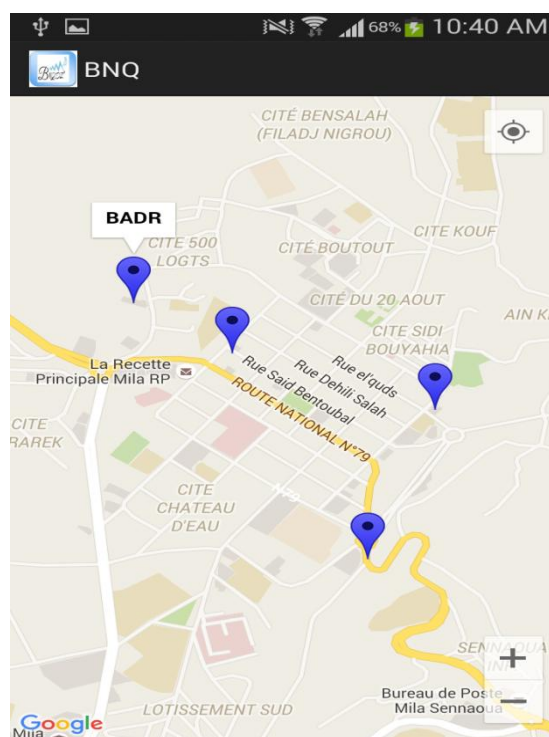


Figure 6.14 : Localiser banque

Dans l'écran « menu » si l'utilisateur choisit « Se connecter », l'utilisateur remplit deux champs (identifiant et mot de passe) et ensuite il clique sur le bouton connexion, puis le serveur vérifie ces informations.



Figure 6.15 : Interface d'authentification

Si l'authentification n'est pas validée par le système, un message d'erreur sera affiché dans le cas où les informations écrites ne sont pas présentes dans notre base de données.

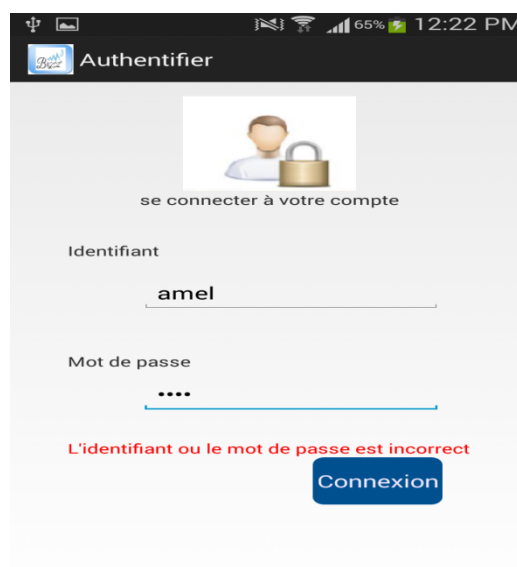


Figure 6.16 : Authentification non validé

Si l'authentification est validée, l'utilisateur accède au menu abonné. Dans cette interface l'abonné peut gérer toutes les fonctionnalités de l'application, l'abonné peut consulter son compte, commander un chéquier... etc



Figure 6.17 : Page réservé aux abonnés

Dans l'écran « abonné » si l'abonné choisit « Consulter compte », dans cette page il doit saisir son numéro de compte et ensuite cliquer sur consulter.

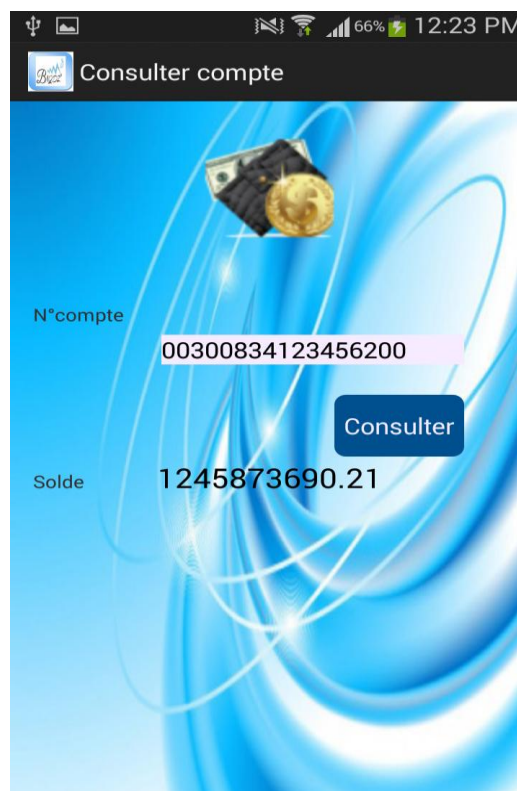


Figure 6.18 : Page consulter compte

Dans l'écran « abonné » si l'abonnée choisit « Commander chéquier », à Partir de cette interface, l'abonné remplit les champs du formulaire, ensuite il valide l'opération de commande de chéquier.

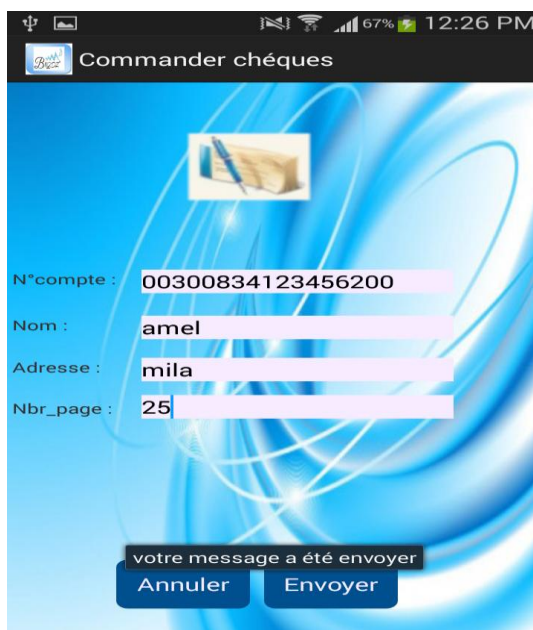


Figure 6.19 : Page commande chèque

Dans l'écran « abonné » si l'abonnée choisit « Afficher les mouvements », à Partir de cette interface l'abonné saisit le numéro de compte et valide pour afficher les mouvements de compte.



Figure 6.20 : Page afficher mouvements

Dans l'écran « abonné » si l'abonnée choisit « Virements bancaires », à Partir de cette interface l'abonné remplit les champs, il aura l'interface suivante ,il doit remplir le formulaire correspondant.



Virements bancaires

Order de virement

DONNEUR D'ORDRE

NOM : amel

Adresse : mila

Date : 2016-05-19

Compte : 00300834123456200

BENEFICIAIRE

NOM : ikram

Compte : 00300843541236200

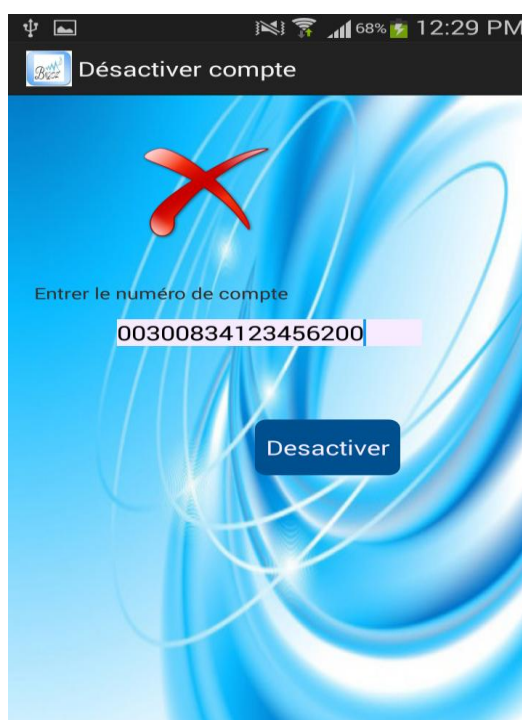
Montant : 258617906.085

votre message a été envoyer

Annuler Envoyer

Figure 6.21 : Virements bancaires

Dans l'écran « abonné » si l'abonnée choisit « Désactiver compte ». à Partir de cette interface, l'abonné désactive son compte.



Désactiver compte

Entrer le numéro de compte

00300834123456200

Desactiver

Figure 6.22 : Page désactive compte

Finalement, comme toute application professionnelle, nous avons ajouté une page nommé « A propos ». Cette page contient les noms de développeurs qui ont créé cette application.



Figure 6.23 : Page à propos

4. Conclusion

Nous avons entamé la phase de réalisation de logiciel qui doit respecter toutes les contraintes citées au paravent. Cette dernière n'est pas une simple étape à réaliser : elle dépend de la complexité du problème à coder à la fin et de ses caractéristiques, comme elle dépend des outils utilisés et des méthodes suivies

Conclusion générale

Dans le tout de cette étude nous avons abordé le développement d'une application mobile pour la gestion des services bancaires ; de deux point de vue modélisation et réalisation, avec le processus 2TUP.

Sur le plan modélisation, l'utilisation du langage UML nous a permis de spécifier et documenter le système à travers ses différents diagrammes, notamment le diagramme des cas d'utilisation permis de capturer et représenter les besoins, le diagramme de classe a permis de représenter l'aspect statique du système. Pour la définition des composants et des postes de travaux, et enfin, les diagrammes de séquence sont utilisés dans les phases de capture de besoins fonctionnel et d'analyse.

Sur le plan réalisation, nous avons choisi le langage Java sous la plateforme Android avec MYSQL pour la gestion de la base de données. En outre, ce projet nous a permis d'enrichir nos connaissances dans des domaines variés comme : L'orienté objet, UML, 2TUP, ...

A travers cette étude nous avons pu mettre en pratique et développer les connaissances que nous avons acquises durant notre cursus universitaire, et en particulier dans les systèmes d'information, les bases de données et les génies logiciels.

Cependant, nous avons senti une grosse pression vue le temps qui est relativement court, donc nous n'avons pas pu réaliser tout ce qu'on a espérer faire.

Cependant, nous envisageons comme perspective du travail réalisé dans ce mémoire de :

- ✓ Publier notre application sur le « Android Market » ;
- ✓ Compléter les autres services et d'ajouter d'autres fonctionnalités :
 - La consultation du compte affiché sous forme d'un graphe ;
 - Localisation de la plus proche agence ;
 - L'envoi d'un SMS lorsqu'il y a un changement dans le compte.
- ✓ Créé une application multiplateforme pour répondre aux demandes des abonnés ;
- ✓ Les services web : établir l'architecture pour profiter des services web ;
- ✓ L'aspect sécurité : l'un des points les plus critiques de cette application concerne. la sécurité, l'application doit être à la pointe d'éviter tout piratage ou l'interception des données.



BIBLIOGRAPHIE

Les ouvrages

- [1] : « Pratiques bancaires des banques étrangères envers les PME algériennes », Mohammed TAHRAOUI Ahmed BOUYACOUB, 2008.
- [4] : « Merise et uml pour la modélisation », Joseph Gabant, Dunod, mars 2004
- [5] : « UML en action de l'analyse des besoins à la conception en java », Pascal ROQUES et Frank VALLEE, édition Eyrolles, 2000.
- [6] : « UML en action de l'analyse des besoins à la conception J2EE », Pascal ROQUES et Franck Vallée.
- [7] : « UML en action de l'analyse des besoins à la conception en java », Pascal Roque & Franck Vallée ; édition Eyrolles 2003
- [9] : « cours méthode de conception »
- [10] : « UML en action » pascal Roques. Eyrolles 2004
- [11] : « UML 2 en action De l'analyse des besoins à la conception » Pascal Roques • Franck Vallée 4e édition février 2007
- [12] : « UML 2 en action De l'analyse des besoins à la conception J2EE » Pascal Roques • Franck Vallée 3e édition
- [13] : « de UML à SQL » jaques guyot. Edition système et information.
- [21] : Mohamed Anouar DAHDEH. « conception, developpement et integration d'une application embarquee de telechargement des applications android » . 2011
- [23] : Philippe Rigaux. Pratique de MySQL et PHP. Edition originale. Version 2001. EYROLLES.

Les sites web

- [2] : [http://www.memoireonline.com/07/09/2323/m_Etude-analytique-dun-financement-bancaire- Credit-dinvestissment-cas-CNEPBANQUE-1.html](http://www.memoireonline.com/07/09/2323/m_Etude-analytique-dun-financement-bancaire-Credit-dinvestissment-cas-CNEPBANQUE-1.html)
- [3] : <http://meilleurbanque.blogspot.com/2012/12/definition-de-la-banque.html>
- [8] : http://boukraa.freehoustia.com/file.php/5/cours_UML_processus.pdf
- [14] : <http://ipeti.forumpro.fr/t21-definition-de-langage-java-javascript>.

[15] : http://fr.wikipedia.org/wiki/Kit_de_d%C3%A9veloppement

[16] : http://fr.wikipedia.org/wiki/Java_Development_Kit

[17] : [http://fr.wikipedia.org/wiki/SEclipse_\(logiciel\)](http://fr.wikipedia.org/wiki/SEclipse_(logiciel))

[18] : <https://stuff.mit.edu/afs/sipb/project/android/docs/tools/sdk/eclipse-adt.html>

[19] : Google API developers, <https://developers.google.com> visité en 02/2012.

[20] : <http://www.mypublicwifi.com/publicwifi/fr/index.html>

[22] : <http://www.wampserver.com/>

[24] : http://www.memoireonline.com/08/09/2507/m_Etude-et-realisation-du-site-web-de-lhotel-la-detente-Gestion-des-reservations-en-ligne1.html

[25] : <http://www.commentcamarche.net/contents/577-javascript-introduction-au-langage-javascript>