

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études
pour l'obtention de la licence en Informatique

Thème

Réalisation d'une application client/serveur (gestion d'une banque) avec JDBC et MySQL selon le modèle à trois couches sous Netbeans.

Réalisé par :

GHAFFOUR Inaàm Sihem

BENLEDGHEM Rafika

Sous la direction de Mr. BENMAMMAR Badr

Présenté le 27 Juin 2013 devant la commission d'examination composée de

- *Mr. BELABED* (Examineur)
- *Mr. MERZOUG* (Examineur)

Année universitaire : 2012-2013

Remerciements // // // // // // // //

Ce travail a été dirigé par Monsieur Badr Benmammar, Maître de conférences classe B et professeur d'informatique à l'université de Tlemcen. On lui adresse à cette occasion nos sincères remerciements pour nous avoir encadré. On tient aussi très honnêtement, à lui exprimer nos reconnaissances pour avoir énormément appris sur les applications client/serveur dans le domaine de gestion bancaire durant le semestre préparatoire. Ce sont au fait sa pédagogie, son savoir faire, sa passion pour la recherche, sa vocation pour l'encadrement, sa sagesse et son dévouement pour la formation qui nous ont encouragés à travailler sous sa direction.

Introduction Générale	5
Introduction Générale	6
Chapitre I.....	8
Aperçu général sur les technologies utilisées dans une application Client-serveur.....	8
1.1 Introduction	9
1.2 Système bancaire à distance.....	9
1.3 Technologies utilisées	10
1.3.1 Le langage Java.....	10
1.3.1.1 Introduction	10
1.3.1.2 Définition.....	11
1.3.1.3 L'utilité de Java.....	11
1.3.1.4 Un bref historique de Java	11
1.3.1.5 Les caractéristiques du Java	12
1.3.1.6 Définition d'un package java	12

1.3.1.7	Quelques type de packages	13
1.3.1.8	Environnement de développement de java.....	13
1.3.2	L'environnement de développement Netbeans	14
1.3.2.1	Introduction	14
1.3.2.2	Définition de NetBeans	14
1.3.2.3	L'utilité de la plateforme NetBeans	14
1.3.2.4	Les caractéristiques de NetBeans	15
1.3.2.5	Les versions de NetBeans selon un ordre chronologique.....	15
1.3.3	Le moteurs de publication dynamiques JSP.....	16
1.3.3.1	Définition de JSP (Java Server Page)	16
1.3.3.2	La différence entre Jsp et Servlet.....	16
1.3.3.3	Les points en commun entre Jsp et Servlet	16
1.3.3.4	Les caractéristiques du Jsp.....	17
1.3.3.5	Les avantages de JSP	17
1.3.4	Le serveur de bases de données MySQL.....	18
1.3.4.1	Introduction au MySQL	18
1.3.4.2	Définition de MySQL	18
1.3.4.3	Avantages de MySQL.....	18
1.3.5	Le driver JDBC.....	19
1.3.5.1	Introduction	19
1.3.5.2	Définition de JDBC.....	19
1.3.5.3	Les caractéristiques de JDBC.....	19
1.3.6	Le langage UML.....	20
1.3.6.3	Introduction	20
1.3.6.2	Définition.....	20
1.3.6.3	L'intérêt de l'UML.....	20
1.3.6.4	Caractéristiques d'UML.....	21
1.3.6.5	Les points forts d'UML	21
CHAPITRE II.....		23
Etude détaillée des fonctionnalités du système		23
2.1 Introduction		24

2.2	Spécification du système.....	24
2.2.1	Eléments de base	24
2.2.2	Identification des cas d'utilisation	25
2.2.3	Diagrammes de séquence	26
2.2.4	Diagramme de Classe	31
2.2.4	Affichage de la base de données.....	32
2.3	Réalisation de l'application	34
	Conclusion général.....	38

Introduction Générale

Introduction Générale

L'évolution de la civilisation humaine caractérisée par une urbanisation accélérée a conduit progressivement l'homme à adopter une approche de management de la qualité dans le domaine de la gestion de l'entreprise moderne, qu'elle soit à caractère commercial, industriel, de service ou autre.

La notion de gestion implique en elle même la maîtrise totale et contrôlée du processus que l'on veut mettre en œuvre. Elle suppose la mise à plat de sa complexité et la connaissance parfaite des interactions des éléments qui la composent.

L'établissement d'un plan et d'une stratégie de gestion constitue les deux outils de la prévention de toute aléa susceptible d'altérer son fonctionnement. Idéalement elle suppose la construction d'un algorithme de décision bien contenu, planifié, équilibré, qui évite l'introduction d'erreur inopinée et impose l'obligation de les corriger de manière itérative et de toute façon inefficace. Cela suppose bien évidemment pour nous, son pilotage par l'automatisation dont l'informatique est pour l'instant la seule discipline capable de l'assurer.

La gestion informatique d'un processus peut parfois suffire à elle même et d'autre fois être associée à l'automatique pour donner ce qu'on appelle l'automatisation ; outil qui est utilisé principalement en industrie.

Quant à notre sujet l'informatique demeure amplement suffisante pour peu que l'informaticien sache utiliser et combiner les outils dont il a acquis les connaissances nécessaires.

Pour cela nous avons pour tâche de développer une application client/serveur qui a pour objet de réaliser une base de données de gestion d'une banque en utilisant un système de gestion de base de données (SGBD) qui est MySQL appartenant à la firme Microsoft traité en trois couches grâce à l'environnement de développement spécifique de Java dénommé NetBeans.

L'intitulé de notre travail est ainsi énoncé de la façon suivante « Réalisation d'une application client/serveur (gestion d'une banque) avec JDBC et MySQL selon le modèle à trois couches sous NetBeans ».

Ce mémoire comprend deux parties :

- Une partie théorique se rapportant au système bancaire
- Une description sommaire des outils informatiques fondamentaux permettant de développer une application de gestion de base de données de la banque.

Chapitre I

Aperçu général sur les technologies
utilisées dans une application
Client-serveur

1.1 Introduction

Nous consacrons la première partie de ce chapitre à la présentation du fonctionnement du système bancaire selon son fonctionnement classique. Par la même occasion et en anticipant sur l'objet de notre travail nous proposerons des éléments de traitement à distance des opérations bancaires que nous soumettrons au client pour lui éviter et éviter également aux agents des banques tout contact direct.

Le client pourra alors bénéficier de toute la technologie des opérations offertes non seulement à distance mais également en temps réel et en dehors des horaires classiques du travail. Le bénéfice apparaît donc de toute évidence très important.

Dans ce cadre l'écueil et le risque du « hacking » demeure non négligeable. Pour cela nous proposerons à nos examinateurs de tenir compte des éléments de sécurité pour s'opposer à ce fléau des temps modernes. Remarquons que ces mesures de sécurité ne font pas partie de notre travail.

La deuxième partie par contre, se base sur la présentation des outils, des environnements et des logiciels utilisés dans la réalisation de notre application. Nous avons jugé très utile de présenter aussi les différentes caractéristiques et avantages de chacun d'eux.

1.2 Système bancaire à distance

Les banques en ligne sont des banques réelles consultables à distance. Elles permettent aux clients de réaliser toutes les opérations bancaires qui lui sont propres et utiles, ceci en ligne à partir de l'internet moyennant un simple pc et une connexion.

Elles conservent la possibilité d'accès à ces services directs au guichet lorsque la nécessité se fait sentir, telle que la discussion d'un contra ou bien la contraction d'un prêt qui nécessite le plus souvent d'apposer une signature. Il semble néanmoins que des banques exclusivement en ligne, existent sans structure physiquement accessible au client. Dans ce cas leur prestation demeure limitée aux seules activités uniquement en ligne tel que « master card ».

Ce qu'on appelle banque en ligne sont enfaite des services bancaires à l'intérieur d'un système bancaire.

Il est ainsi aisé de comprendre que ces services comprennent les éléments ci-dessous :

- Consultation de compte : solde, type de compte, numéro de compte,
- Création de nouveaux comptes (pour un client possédant préalablement un compte),
- Opération de virements (par ex de compte à compte),
- Réception de fonds publics,
- Opérations de crédit,
- Paiement en ligne,
- Règlement de facture (achat et vente),
- Dons et legs.

1.3 Technologies utilisées

1.3.1 Le langage Java

1.3.1.1 Introduction

Un langage de programmation est un outil de communication, permettant à un individu de dialoguer avec une machine en lui soumettant des instructions et en analysant les données matérielles fournies par le système, généralement un ordinateur.

Il existe de nombreux langages de programmation dont la plupart sont réservés à des domaines spécialisés, par exemple :

- C : système d'exploitation (Unix/Linux)...
- C++ : applications de grande taille...
- JAVA, C# : applications de grande taille, web...

- LISP : prototypage, systèmes experts...

Parmi tous les langages existants, notre choix a porté sur le langage Java. Celui-ci correspond aux objectifs que nous avons préalablement fixés pour notre projet de fin d'étude. Connu pour sa modernité, sa grande performance dans le domaine du web, il permettra sans doute de répondre à nos attentes dans le développement de notre mémoire. [1]

1.3.1.2 Définition

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes modernes et performants, notamment dans la construction des utilitaires, des jeux et des applications professionnelles. Java est utilisé sur plus de 850 millions d'ordinateurs de bureau et plus d'un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle. [2]

1.3.1.3 L'utilité de Java

Beaucoup d'applications et de sites Web ne fonctionnent pas sans Java. Le nombre de ces applications augmente chaque jour du fait de la rapidité, la sécurité de la portabilité de ce logiciel. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts. [2]

1.3.1.4 Un bref historique de Java

- 1995, mai : premier lancement commercial du JDK 1.0
- 1996, janvier : JDK 1.0.1
- 1996, septembre : lancement du JDC
- 1997, février : JDK 1.1
- 1998, décembre : lancement de J2SE 1.2 et du JCP
- 1999, décembre : lancement J2EE
- 2000, mai : J2SE 1.3

- 2002, février : J2SE 1.4
- 2004, septembre : J2SE 5.0
- 2006, mai : Java EE 5, décembre : Java SE 6.0
- 2008, décembre : Java FX 1.0
- 2009 décembre : Java EE 6
- 2010 : janvier : rachat de Sun par Oracle
- 2011 : juillet : Java SE 7 [3]

1.3.1.5 Les caractéristiques du Java

- Java est interprété.
- Java est portable.
- Java est orienté objet.
- Java est simple.
- Java est fortement typé.
- Java assure la gestion de la mémoire.
- Java est sûre.
- Java est économique.
- Java est multitâche. [4]

1.3.1.6 Définition d'un package java

En Java, il existe un moyen de regrouper des classes voisines ou qui couvrent un même domaine : ce sont les packages. Pour réaliser un package, on écrit un nombre quelconque de classes dans plusieurs fichiers d'un même répertoire et au début de chaque fichier on met la directive.

D'une façon générale, l'instruction package associe toutes les classes qui sont définies dans un fichier source à un même package. Le mot clé package doit être la première instruction dans un fichier source et il ne doit être présent qu'une seule fois dans le fichier source (une classe ne peut pas appartenir à plusieurs packages). [3]

1.3.1.7 Quelques type de packages

Il existe plusieurs types de packages : le package par défaut identifié par le point qui représente le répertoire courant. Il permet de localiser les classes qui ne sont pas associées à un package particulier. Les packages standards qui sont empaquetés dans le fichier classes .zip et les packages personnels. Le compilateur implémente automatiquement une commande *import* lors de la compilation d'un programme Java même si elle ne figure pas explicitement au début du programme : *import*

Voici la liste de quelques packages :

- java. Applet
- java.awt
- java.awt.color
- java.awt.datatransfer
- java.awt.dnd
- java.awt.event
- java.awt.font
- java.awt.geom
- java.awt.im
- java.awt.im.spi
- java.awt.image
- java.beans etc. [6]

1.3.1.8 Environnement de développement de java

Les principales Versions du JDK

- jdk 1.0 : première version, orienté développement Web (applet)
- jdk 1.1 : version plus costaud, destiné aussi aux applications d'entreprises.
- jdk 1.2 : améliore et enrichie la 1.1 : collections, interface utilisateur swing etc.
- jdk 1.3 : de nouvelles performances et de nouvelles classes (3D . . .)

- jdk 1.4 : amélioration de la vitesse d'exécution, de nouvelles fonctionnalités (XML, String...)
- jdk 1.5 : collections typées, encapsulation automatique des types primitifs.
- Jdk1.6 : fortement dégradé pout lancer un ide et effectuer n'importe quelle action uml.
- Jdk1.7 : Il permet l'édition d'HTML 5, et intègre la mise à jour du serveur Glassfish en 3.1.1. [6]

1.3.2 L'environnement de développement Netbeans

1.3.2.1 Introduction

Les environnements de développement intégrés (EDI), sont des logiciels regroupant un ensemble d'outils nécessaires au développement logiciel dans un (ou plusieurs) langage(s) de programmation.

Parmi tous les environnements de développement existant notre choix a porté sur NetBeans à la faveur de sa rapidité à mettre en place une application web qui est l'un des points forts de ce dernier.

1.3.2.2 Définition de NetBeans

C'est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web). [4]

1.3.2.3 L'utilité de la plateforme NetBeans

Il y aura toujours une place dans le monde pour "les clients lourds". La Plateforme NetBeans apporte aux applications bureautiques les mêmes avantages que l'architecture J2EE apporte aux applications coté-serveur:

- **Un contexte de déploiement runtime** pour des fonctionnalités arbitraires qui simplifient le développement,
- **Une boîte à outils** qui permet de gagner beaucoup de temps en développement et en effort,

- **Un ensemble d'abstractions** qui permet aux développeurs de se concentrer sur la business logic, et non de réécrire de la logique de routine et des composants requis par la plupart des applications,
- **Un ensemble de Standards** pour rehausser et renforcer la consistance et l'interopérabilité entre les applications et les systèmes d'exploitation. []

1.3.2.4 Les caractéristiques de NetBeans

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web, auto-complétion, gestion multi-projets, intégration des documents de référence, support de solutions de gestion de versions, intégration de débogueur...).

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement JDK est requis pour les développements en Java. []

NetBeans permet également de supporter différents autres langages, comme :

- UML
- SOA (WSDL, XML),
- Python,
- C, C++,
- JavaScript,
- XML,
- Ruby,
- PHP
- HTML.

1.3.2.5 Les versions de NetBeans selon un ordre chronologique

- 7.2 *sortie* Juillet 24, 2012
- 7.1.2 *sortie* Avril 26, 2012
- 7.1.1 *sortie* Février 29, 2012
- 7.1 *sortie* Janvier 5, 2012
- 7.0.1 *sortie* Aout 1, 2011
- 7.0 *sortie* Avril 19, 2011
- 6.9.1 *sortie* Aout 4, 2010
- 6.9 *sortie* Juin 15, 2010
- 6.8 *sortie* Décembre 10, 2009
- 6.7.1 *sortie* Juillet 27, 2009

- 6.7 *sortie* Juin 29, 2009
- 6.5.1 *sortie* Mars 16, 2009
- 6.5 *sortie* Novembre 20, 2008
- 6.1 *sortie* Avril 28, 2008
- 6.0 *sortie* Décembre 3, 2007
- 5.5.1 *sortie* Mai 24, 2007
- 5.5 *sortie* Octobre 30, 2006
- 5.0 *sortie* Janvier2006
- 4.1 *sortie* Mai 2005
- 4.0 *sortie* Décembre 2004
- 3.6 *sortie* Avril2004
- 3.5 *sortie* Juin2003

1.3.3 Le moteurs de publication dynamiques JSP

1.3.3.1 Définition de JSP (Java Server Page)

C'est une technologie Java qui permet la génération de pages web dynamiques. La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements sous formes de classes Java définissant un Bean ou une Servlet. Ceci est d'autant plus facile que les JSP (Java Server Pages) définissent une syntaxe particulière permettant d'appeler un Bean et d'insérer le résultat de son traitement dans la page HTML dynamiquement. Ces dernières sont les raisons pour lesquelles on a choisit le moteur Jsp. [4]

1.3.3.2 La différence entre Jsp et Servlet

- La servlet générée est compilée et sauvegardée puis elle est exécutée. Les appels de la JSP sont beaucoup plus rapides car la servlet, conservée par le serveur, est directement exécutée.
- Il y a plusieurs manières de combiner les technologies JSP, les beans/EJB et les servlets en fonction des besoins pour développer des applications web.
- Comme le code de la servlet est généré dynamiquement, les JSP sont relativement difficiles à déboguer.

1.3.3.3 Les points en commun entre Jsp et Servlet

Les servlets et les JSP ont de nombreux points communs puisque qu'une JSP est finalement convertie en une servlet. (Les JSP sont basées sur les servlets : tout ce qui est fait par une servlet pour la génération de pages dynamiques peut être fait avec une JSP).

- Le choix d'utiliser l'une ou l'autre de ces technologies ou les deux doit être fait pour tirer le meilleur parti de leurs avantages.
- Il est préférable d'utiliser les JSP pour générer des pages web dynamiques. Mais l'usage des servlets est obligatoire si celles-ci doivent communiquer directement avec une applet ou une application et non plus avec un serveur web. [7]

1.3.3.4 Les caractéristiques du Jsp

Langage

- Java

Mode d'exécution

- Compilé en pseudo code (bytecode)

Principaux avantages:

- Repose sur la plate-forme Java dont elle hérite des avantages

Principaux inconvénients

- Débogage assez fastidieux
- Beaucoup de code à écrire

1.3.3.5 Les avantages de JSP

- L'utilisation de Java par les JSP permet une indépendance de la plate-forme d'exécution mais aussi du serveur web utilisé.
- La séparation des traitements et de la présentation : la page web peut être écrite par un *designer* et les *tags* Java peuvent être ajoutés ensuite par le développeur. Les traitements peuvent être réalisés par des composants réutilisables (des Java beans).

1.3.4 Le serveur de bases de données MySQL

1.3.4.1 Introduction au MySQL

Une base de données permet de manipuler les informations de manière efficace, de les enregistrer, de les trier, de les lire et d'y effectuer des recherches. Les principaux concurrents de MySQL sont PostgreSQL, Microsoft SQL Server et Oracle. Mais notre choix a porté sur le serveur MySQL qui contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données. MySQL est donc un serveur multiutilisateur et multithread. Il utilise SQL (Structured Query Language), le langage standard des requêtes des bases de données. MySQL est disponible depuis 1996, mais son développement remonte à 1979. Il s'agit de la base de données open-source la plus employée au monde et elle a reçu le Linux Journal Readers' Choice Award à plusieurs reprises. [9]

1.3.4.2 Définition de MySQL

MySQL est un système de gestion de bases de données relationnelles robuste et rapide. Il est très utilisé dans les projets libres et dans le milieu industriel. MySQL est un SGBDR facile à utiliser qui convient très bien pour la plupart des sites web. La rapidité de développement a été, depuis le début l'objectif principal de ceux qui l'ont écrit. Pour cela ils ont décidé de proposer moins de fonctionnalités, mais son installation et son utilisation sont plus aisées.

1.3.4.3 Avantages de MySQL

MySQL possède sur plusieurs avantages :

- des performances élevées ;
- un coût réduit ;
- une simplicité de configuration, et d'apprentissage ;
- sa portabilité ;
- l'accessibilité de son code source ;
- la disponibilité du support. [9]

1.3.5 Le driver JDBC

1.3.5.1 Introduction

Les développeurs Java ont constaté qu'ils nécessitent un cadre de travail qui permettra de construire une interface uniforme sur les systèmes de connectivité de base de données d'un tel cadre permettrait au programmeur d'écrire une interface de base de données unique sur de nombreuses plates-formes ce cadre de travail est connu comme l'interface Java Database Connectivity.

1.3.5.2 Définition de JDBC

JDBC est une API de niveau SQL qui permet de passer les instructions SQL dans les arguments des interfaces JDBC. Pour que cela puisse se faire de façon indépendante de la base de données, JDBC impose que les fournisseurs de base de données offrent une implémentation de ses interfaces. Ces implémentations routent les appels SQL à la base de données en des appels propriétaires quelle reconnaît. [8]

1.3.5.3 Les caractéristiques de JDBC

Les caractéristiques d'internationalisation du pilote JDBC de Microsoft SQL Server incluent ce qui suit :

- Prise en charge d'une version complètement localisée dans les mêmes langues que SQL Server.
- Prise en charge des conversions de langage Java 1.4 pour les données SQL Server sensibles aux paramètres régionaux.
- Prise en charge des langues internationales, indépendamment du système d'exploitation.

[10]

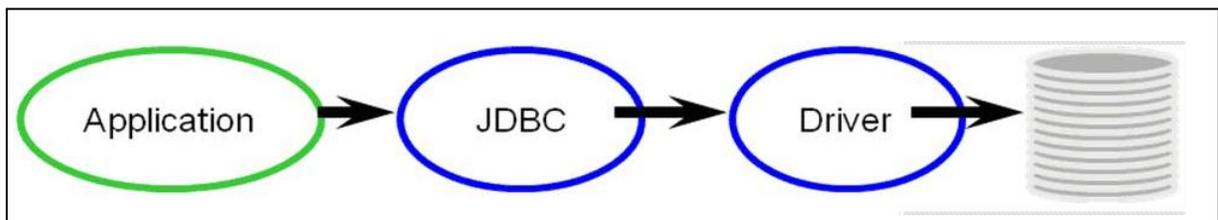


Figure 5-4 : Architecture d'un JDBC. [11]

1.3.6 Le langage UML

1.3.6.3 Introduction

Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes. UML, qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts du logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques.

1.3.6.2 Définition

UML signifie Unified Modeling Language. La justification de chacun de ces mots nous servira de fil conducteur pour cette présentation. [12]

- Langage unifié pour la modélisation objet
- Langage de modélisation des applications construites à l'aide d'objets, indépendant de la méthode utilisée

1.3.6.3 L'intérêt de l'UML

- UML comble une lacune importante des technologies objet. Il permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation.
- UML est un langage formel, défini par un méta modèle.
- UML normalise les concepts objet.
- UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet.
- UML est donc bien plus qu'un simple outil qui permet de "dessiner" des représentations mentales...
- Il permet de parler un langage commun, normalisé mais accessible, car visuel. Il représente un juste milieu entre langage mathématique et naturel, pas trop complexe.
- Une autre caractéristique importante d'UML, est qu'il cadre l'analyse.
- UML permet de représenter un système selon différentes vues complémentaires : les diagrammes.
- UML opte en effet pour l'élaboration des modèles, plutôt que pour une approche qui impose une barrière stricte entre analyse et Conception. [4]

1.3.6.4 Caractéristiques d'UML

- UML est basé sur un méta-modèle : Le méta-modèle d'UML en fait un langage formel possédant les caractéristiques suivantes :
- un langage sans ambiguïtés
- un langage universel pouvant servir de support pour tout langage orienté objet
- un moyen de définir la structure d'un programme
- une représentation visuelle permettant la communication entre les acteurs d'un même projet
- une notation graphique simple, compréhensible même par des non informaticiens
- UML comme visualisation complète d'un système
- UML offre une manière élégante de représenter le système selon différentes vues complémentaires grâce aux diagrammes.
- UML permet d'économiser de l'argent : si le modèle ne convient pas au client, il sera "simple" à modifier, contrairement à une application directement implémentée. [13]

1.3.6.5 Les points forts d'UML

- **UML est un langage formel et normalisé**
 - gain de précision
 - gage de stabilité
 - encourage l'utilisation d'outils
- **UML est un support de communication performant**
 - Il cadre l'analyse
 - Il facilite la compréhension de représentations abstraites complexes
 - Son caractère polyvalent et sa souplesse en font un langage universel
- **5 vues statiques du système :**
 - diagrammes de cas d'utilisation (Fonctionnel)
 - diagrammes de classes

- diagrammes d'objets
- diagrammes de composants
- diagrammes de déploiement
- **4 vues dynamiques du système :**
 - diagrammes de séquence
 - diagrammes de collaboration
 - diagrammes d'états-transitions
 - diagrammes d'activités. [14]

CHAPITRE II

Etude détaillée des fonctionnalités du système

2.1 Introduction

Dans ce chapitre, nous allons consacrer une partie pour présenter quelques diagrammes UML, Ils sont élaborés tout au long du cycle de vie du développement de notre projet c.à.d. depuis le recueil des besoins jusqu'à la phase de conception.

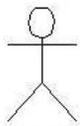
Et en ce qui concerne la deuxième partie, elle se base sur l'affichage de notre application détaillé, avec les différentes opérations bancaires permises via internet que l'on a effectués et d'autant plus l'affichage de notre base de données.

2.2 Spécification du système

Parmi les diagrammes UML largement connus par les informaticiens, on cite :

- *le diagramme de cas d'utilisation* : il permet de recueillir, d'analyser et d'organiser les besoins. Avec lui débute l'étape d'analyse de notre système.
- *Diagramme de séquence* : il permet de représenter des interactions entre objets et acteurs, selon un point de vue temporel avec une chronologie des envois de messages, c'est un type de diagramme d'interaction.
- *Diagramme de classe* : il exprime la structure statique du système en terme de classes, ainsi qu'aux relations entre ces classes.

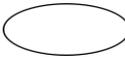
2.2.1 Eléments de base



Acteur : entité (personne ou système) externe qui échange de l'information (entrée/sortie)

- L'acteur peut consulter ou modifier l'état du système.
- En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin.
- Les acteurs peuvent être classés (hiérarchisés) en faisant une sorte d'héritage.

- Les acteurs utilisés dans nos diagrammes sont :
 - ✓ **Administrateur** : agent banquier devant assurer la bonne inscription de tout client se présentant à la banque.
 - ✓ **Client** : toute personne possédant un compte bancaire spécifique. Il est considéré comme le principal utilisateur de l'application.

 **Cas d'utilisation**: cet ensemble d'actions réalisées par le système, en réponse à une action d'un acteur, est présenté dans le diagramme de la figure 2-2-2 dont les caractéristiques sont les suivantes :

- Les uses cases peuvent être structurés.
- Les uses cases peuvent être organisés en paquetages (packages).
- L'ensemble des use cases décrit les objectifs (le but) du système.

 **Objet** : instance de classes

 **Bande d'activation** : on l'utilise pour l'envoi des messages synchrones.

 **Ligne de vie** : cette ligne existe tant qu'il ya une interaction entre les objets.

2.2.2 Identification des cas d'utilisation

Ce diagramme permet de décrire les services les plus importants rendus par le système de gestion bancaire à distance.

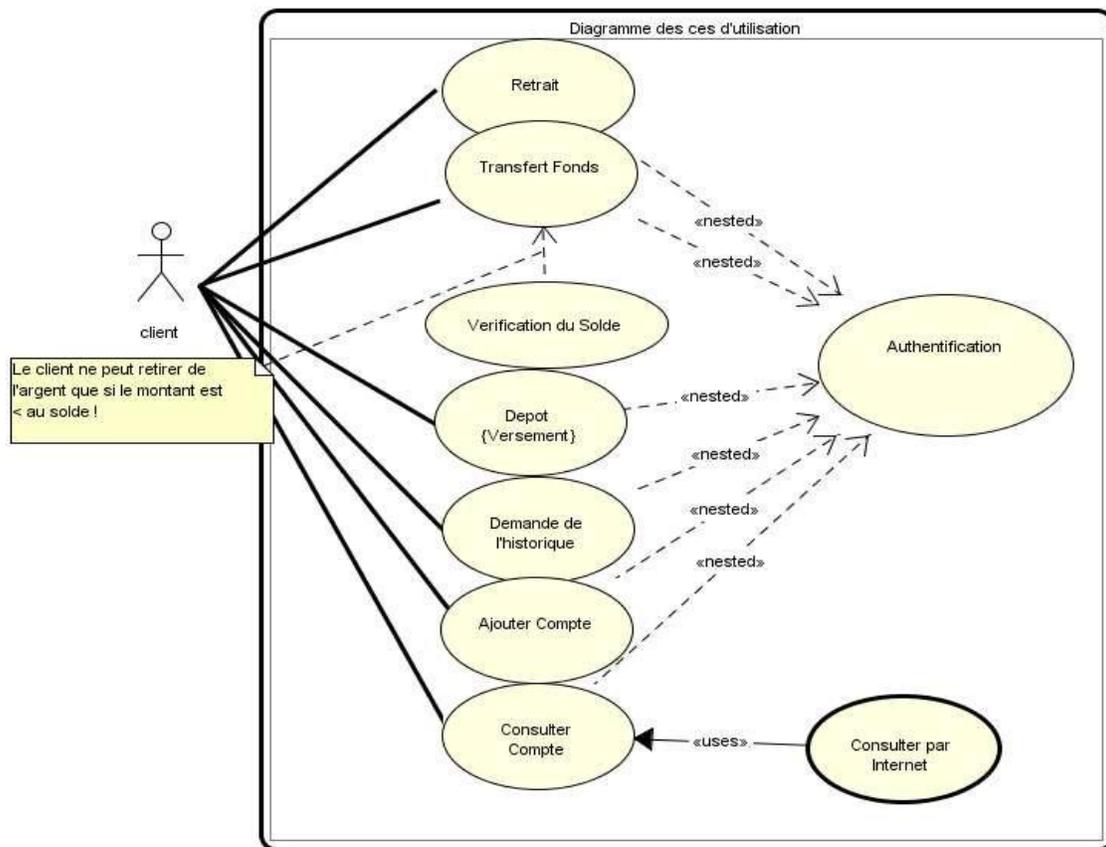


Figure 2-2-2 : Diagramme des cas d'utilisation

2.2.3 Diagrammes de séquence

Une séquence est une scénarisation théorique d'un cas d'utilisation précis en impliquant toutes les possibilités auxquelles ce dernier peut être confronté. Dans cette partie, on va présenter différents diagrammes de séquence comme suit :

- *Cas d'utilisation : L'ajout d'un nouveau client*
 - ✓ Un nouveau client qui se présente au niveau de la banque sera pris en charge par l'agent bancaire (administrateur),
 - ✓ L'administrateur aura accès au formulaire d'inscription du client pour vérifier la validité et l'intégrité des informations,
 - ✓ Dans le cas où toutes les informations nécessaires seront bien saisies et valides, l'administrateur pourra valider l'ajout du client.

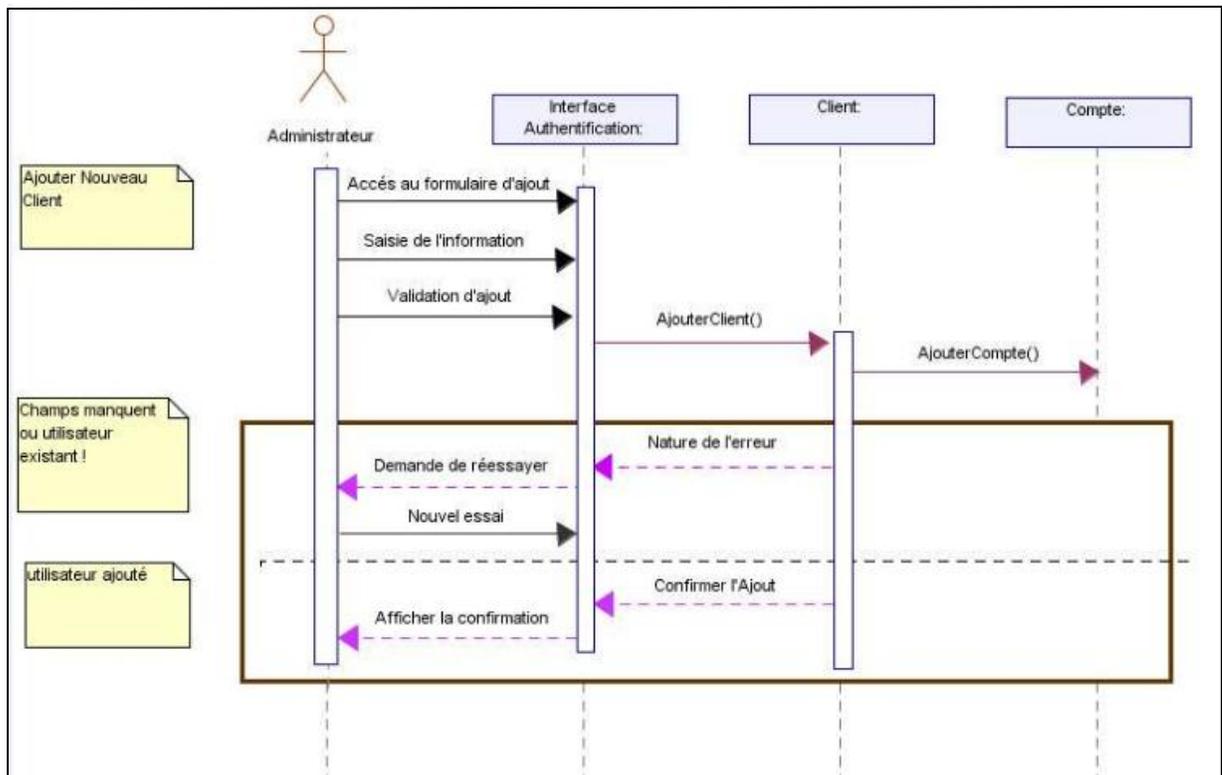


Figure 2-2-3-1 : Diagramme de séquence : Ajouter nouveau client

- *Cas d'utilisation : Inscription à distance et authentification*
 - ✓ L'inscription à distance se fait dans quelques banques,
 - ✓ La phase d'authentification doit précéder toute opération bancaire voulue, à cette étape le client doit saisir le code d'identification et le password d'une manière correcte,
 - ✓ Ensuite, le système envoie une requête à la table client pour vérifier que ce client existe dans la base de données en qualité de client adhérent,
 - ✓ le système pourra donc s'assurer de la validité de ces informations ; il lui donnera par la suite accès vers sa page personnelle.

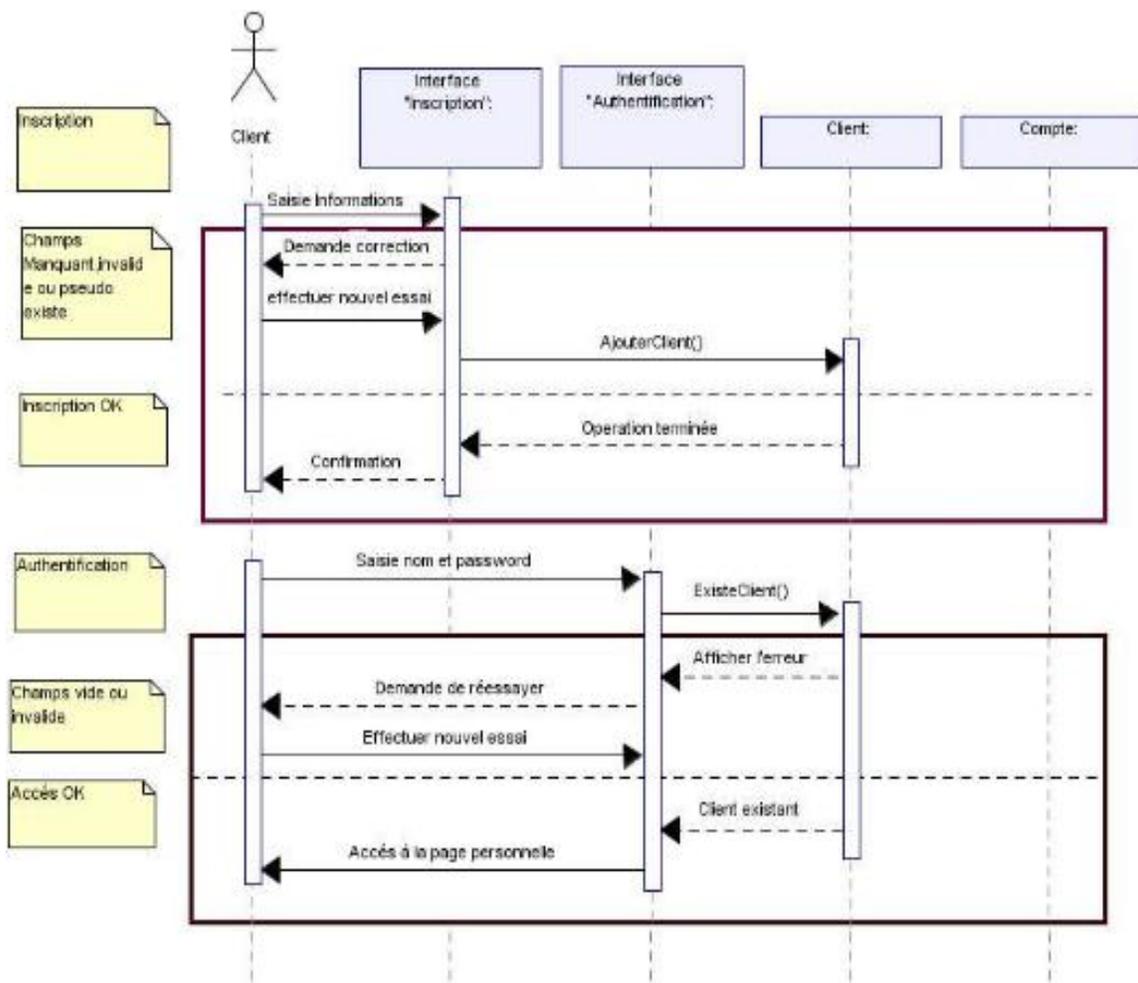


Figure 2-2-3-2 : Diagramme de séquence : Inscription et authentification.

- *Cas d'utilisation : Consultation du compte par le client*
 - ✓ Cette phase est précédée par l'étape d'authentification déjà expliquée,
 - ✓ Après avoir accéder à sa page personnelle, le client doit choisir l'opération consulter compte,
 - ✓ Le système va envoyer une requête vers la table compte pour lui afficher son solde.

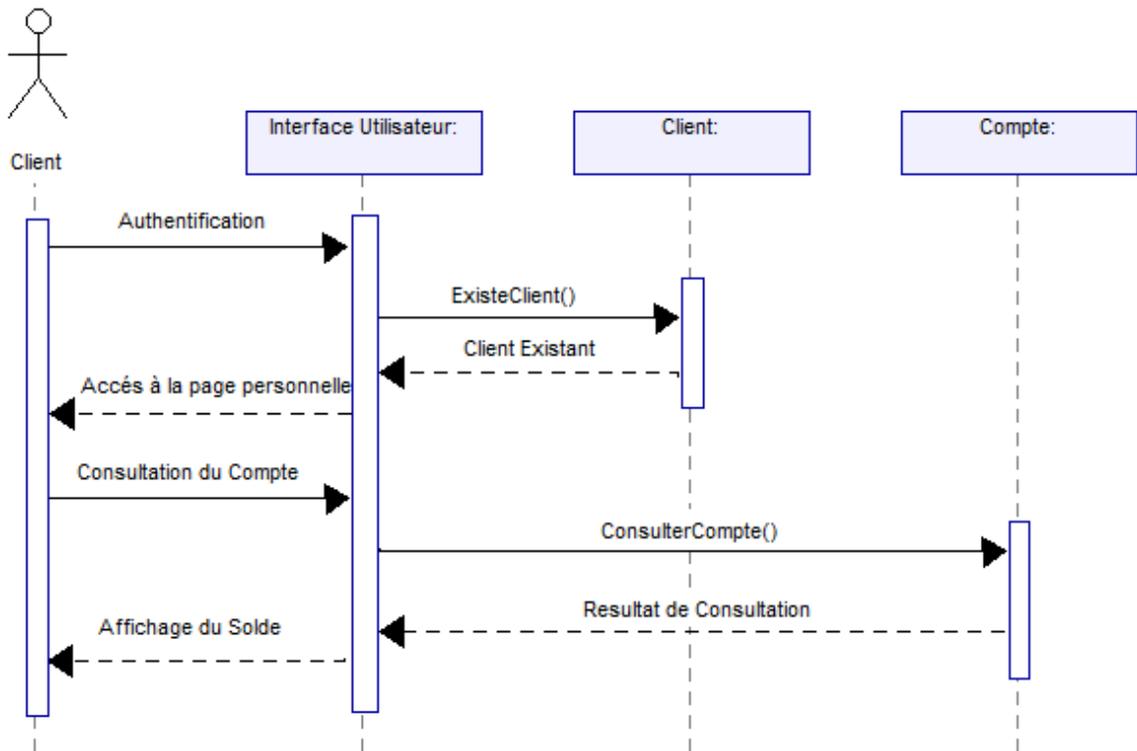


Figure 2-2-3-3 : Diagramme de séquence : Consultation du compte.

- *Cas d'utilisation : Suppression et ajout d'un compte*
 - ✓ Ces deux opérations sont permises pour tout client possédant déjà un compte bancaire,
 - ✓ Concernant **l'ajout d'un nouveau compte**, le client va choisir l'opération ajouter compte,
 - ✓ Il doit remplir par la suite le formulaire affiché sur sa page personnelle,
 - ✓ Si tout est bien rempli avec la spécification du solde de départ, alors l'apparition de l'affichage montre un succès de l'opération. Ainsi, le nouveau compte sera mis à la disposition du client,
 - ✓ Si le client désire **Supprimer le compte** déjà sélectionné, le système est conçu pour vérifier si le solde est nul. Sinon, l'opération va échouer.

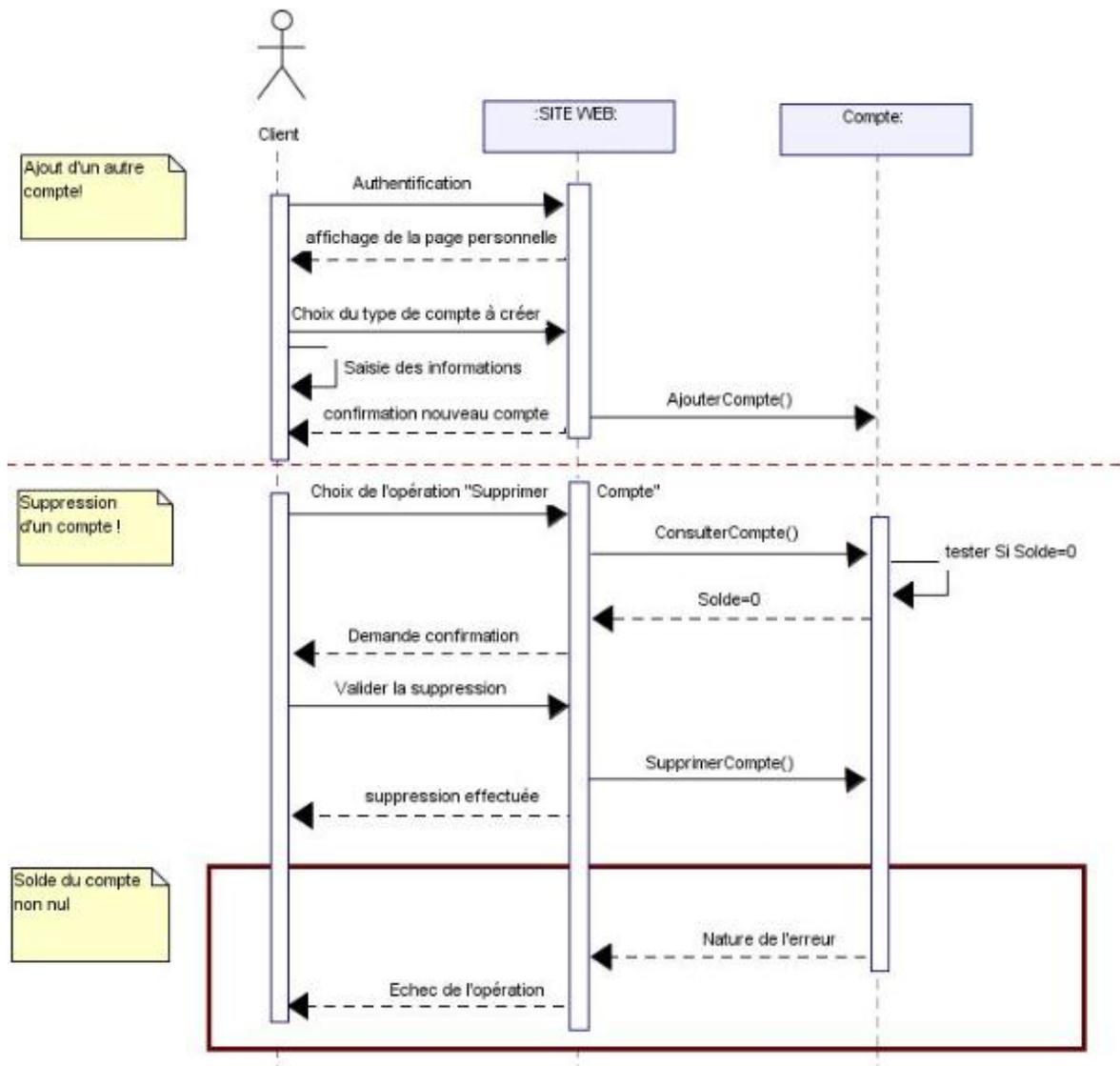


Figure 2-2-3-4 : Diagramme de séquence : Ajout ou suppression d'un compte bancaire.

- *Cas d'utilisation : Virement bancaire*

- ✓ Pour effectuer l'opération de virement le client doit saisir le numéro de compte du destinataire et le montant qu'il veut transférer,
- ✓ Le système doit envoyer une requête à la table compte pour tester si son solde est suffisant pour satisfaire cette opération.

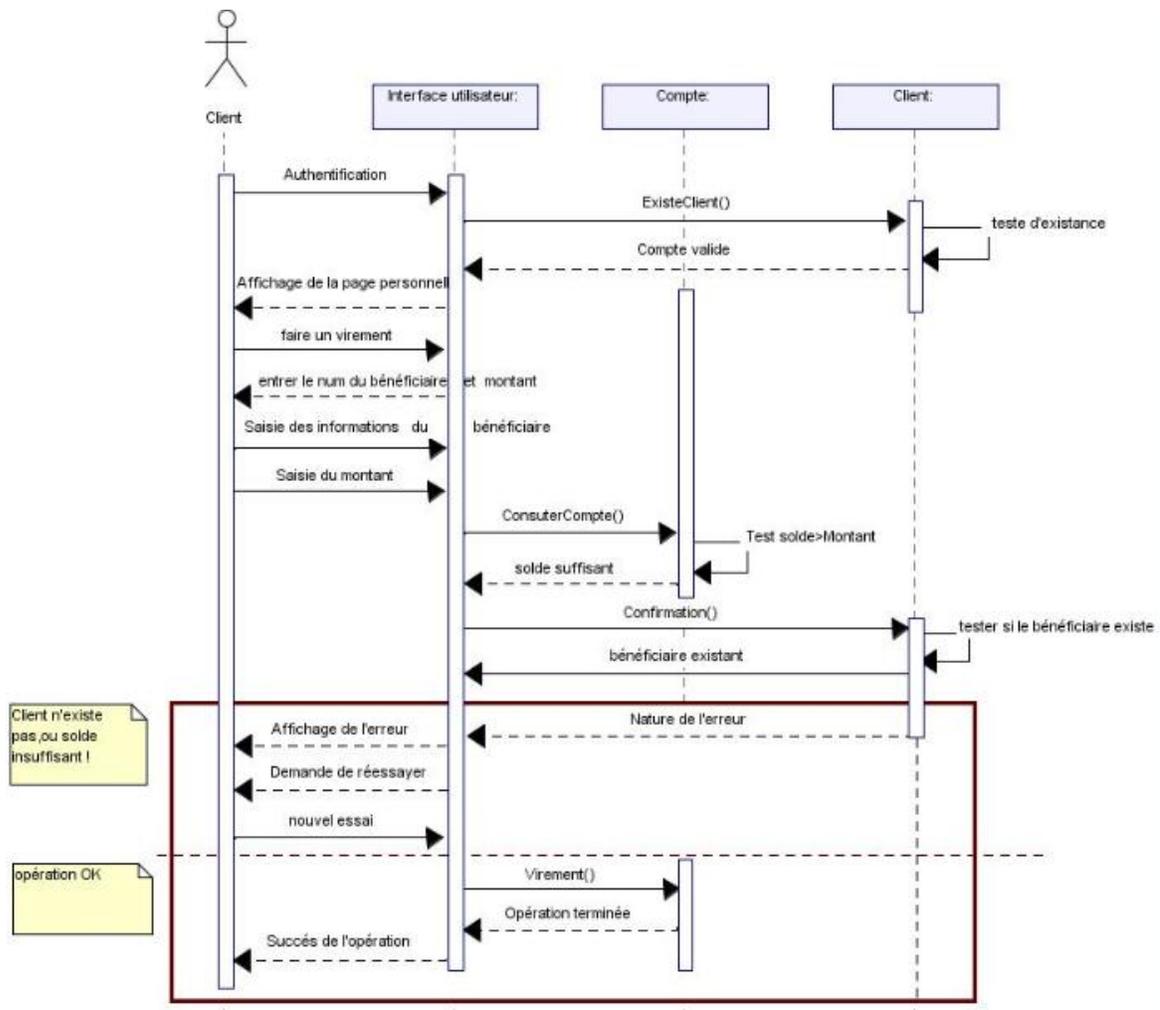


Figure 2-2-3-5 : Diagramme de séquence : Effectuer un virement à distance.

2.2.4 Diagramme de Classe

Les diagrammes de classes servent à comprendre la structure de classe des projets, on les utilise pour personnaliser, partager et présenter avec d'autres classes les informations relatives au projet. [19]

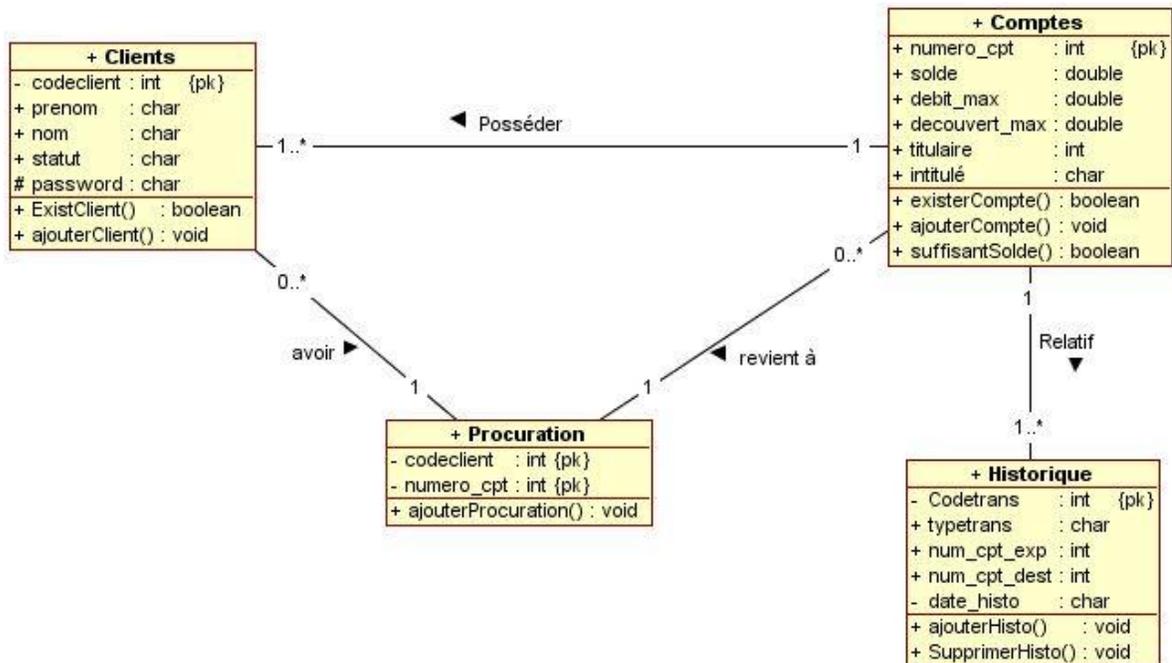


Figure 2-2-3-6 : Diagramme de classes

2.2.4 Affichage de la base de données

On a construit la base de données sur la base du logiciel EasyPHP-12.1, celle-ci se présente comme suit :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
clients	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	16 Kio	-
comptes	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	32 Kio	-
historique	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	16 Kio	-
procuration	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	32 Kio	-
4 tables	Somme	8	InnoDB	latin1_swedish_ci	96 Kio	0 o

Figure 2-2-4-1 : Structure des tables de la base de données

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 codeclient	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	2 prenom	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	3 nom	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	4 statut	varchar(30)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	5 password	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Figure 2-2-4-2 : Structure de la table Clients

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 numerocpt	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	2 solde	double			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	3 debitmax	double			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	4 decouvertmax	double			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	5 titulaire	int(11)			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	6 intitule	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	7 codeclient	int(11)			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	8 codetrans	int(11)			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Figure 2-2-4-3 : Structure de la table Comptes

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 codetrans	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	2 typetrans	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	3 numcptexpd	int(11)			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	4 numcptdist	int(11)			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	5 solde	double			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus
<input type="checkbox"/>	6 datehisto	date			Non	Aucune		Modifier Supprimer Affiche les valeurs distinctes Primaire Unique plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Figure 2-2-4-4 : Structure de la table Historique

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	numerocpt	int(11)	Non	Aucune				Modifier Supprimer Affiche les valeurs distinctes Primaire Unique Index Spatial plus
2	codeclient	int(11)	Non	Aucune				Modifier Supprimer Affiche les valeurs distinctes Primaire Unique Index Spatial plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Figure 2-2-4-5 : Structure de la table Transaction

2.3 Réalisation de l'application

La présentation de notre application est la dernière phase du déroulement de notre projet, l'affichage de notre application se fait à travers les JSP.



Banque Algérie © est un site web de gestion bancaire à distance
 Banque Algérie © vous permettra de suivre et de gérer vos comptes en ligne
 Banque Algérie © Votre banque, partout dans le monde
 Ma banque privée en ligne
 La banque d'un monde qui change - La banque d'un monde qui bouge - Parlons d'avenir

Figure 2-3-1 : L'interface de l'application

Avant d'effectuer n'importe quelle opération bancaire, il faut tout d'abord s'authentifier ;

L'interface suivante représente la phase d'authentification du client pour accéder à sa page personnelle.



The image shows the authentication interface for Banque d'Algérie. At the top, there is a blue gradient header. Below it, the bank's logo is displayed in Arabic (بنك الجزائر) and French (BANQUE D'ALGERIE). A navigation bar contains two buttons: 'Accueil' and 'Acces client'. Below the navigation bar, a message reads 'Veuillez-vous identifier pour accéder à ce service'. The login form consists of two input fields: 'Code d'identification' and 'Mot de passe'. Below these fields are two buttons: 'Se connecter' and 'Effacer'.

Figure 2-3-2 : L'interface d'authentification

Lorsque le client sera bien identifié, il pourra par la suite choisir l'une des opérations suivantes et répéter l'opération jusqu'à se qu'il termine ses besoins.



Figure 2-3-3 : L'interface des opérations bancaires

Le client peut avoir un ou plusieurs comptes dans la même banque, donc il peut ajouter de nouveaux comptes à condition de spécifier le solde de départ et en précisant le type du compte dont il veut ouvrir, comme il était cité précédemment dans la phase de spécification du système.

The image shows the top part of a web browser window. At the top, there is a blue header with the Banque d'Algérie logo in Arabic and French. Below the logo is a navigation bar with two buttons: "Accueil" and "Acces client". The main content area is titled "Création d'un nouveau compte". It contains a form with the following fields and options:

- A link: [revenir a la page d'accueil](#)
- Text: "Veillez insérer votre code client :"
- Text input field for the client code.
- Text: "Type de compte :"
- Radio button options:
 - Compte dépôt
 - Compte épargne
 - Compte commercial
 - Compte familial
- Text: "Solde départ :"
- Text input field for the starting balance.
- Buttons: "Ajouter" and "Effacer".

Figure 2-3-4 : L'interface d'ajout d'un nouveau compte

Conclusion général

Notre travail a consisté à mettre au point une application client/serveur qui a pour but d'effectuer la gestion bancaire à distance selon le modèle à trois couches et le connecteur de base de données(JDBC) et en se basant sur le système de gestion de base de données(MySQL).

Dans sa conception moderne la gestion personnalisée d'un compte bancaire implique un accès personnalisé et sécurisé du seul client à son compte. Ainsi la requête portera uniquement sur son code d'accès au compte qui ne doit pas être confondu avec le numéro de compte.

Il n'est nullement question de requête basée sur le nom, prénom par exemple ou une quelle conque donnée personnelle du client. L'accès est notamment et surtout interdit (au sens d'impossible) aux agents bancaires.

Malgré toutes les précautions prises dans la construction d'une telle application deux risques majeurs demeurent cependant :

- Le premier est le hacking
- Le deuxième : les catastrophes naturelles ou non naturelles (perte de matériels par inondation, incendie, séisme, guerre, etc...)
- Pour le premier des solutions de plus en plus sophistiqués sont régulièrement développées (emprunte digitale, reconnaissance vocale, emprunte iridienne, etc...)
- Pour le second, des serveurs de sauvetage situés à très grande distance du serveur principale demeure pour l'instant la solution idoine.

[1] <http://www.techno-science.net/?onglet=glossaire&definition=11378>

[2] http://www.java.com/fr/download/faq/whatis_java.xml

- [3] <http://jmdoudoux.developpez.com/cours/developpons/java/chap-presentation.php>

- [4] <http://jmdoudoux.developpez.com/cours/developpons/java/chap-presentation.php>

-

- [6] <http://vincent-lecomte.blogspot.com/2011/10/java-axis2-et-web-service-partie-1.html>

<http://archives.reseaucerta.org/cotelabo/dev/coursJava2/>

- [5] <http://www.jmdoudoux.fr/java/dej/chap-poo.htm>

[8] George.R, 2000 : « JDBC et JAVA ». Ed.O'REILLY,Paris.

[9] Welling.L & Thomson.L,2009 : «PHP & MySQL ». Ed. Pearson, Education. France.

[10] [http://msdn.microsoft.com/fr-fr/library/ms378857\(v=sql.90\).aspx](http://msdn.microsoft.com/fr-fr/library/ms378857(v=sql.90).aspx)

[11] <http://www.geekmantra.com/subsection.php?section=JDBC&subSection=JDBC+Basic>

[12] Olivier.S, 2005-2006 : «Introduction à la modélisation orientée objets avec UML». Ed France

[19] [http://msdn.microsoft.com/fr-fr/library/33864ckt\(v=vs.90\).aspx](http://msdn.microsoft.com/fr-fr/library/33864ckt(v=vs.90).aspx)

e224823.pdf module6_fe.pdf