

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

## Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de Projets Informatiques**

*Présenté par*  
**Lydia MEZIANI**

Thème

## **Conception et réalisation d'une application Mobile Banking**

*Mémoire soutenu publiquement le ...../...../20..... Devant le jury composé de :*

**Président : M<sup>r</sup> S.SADOU**

**Encadreur : M<sup>r</sup> A.HAMMACHE**

**Examineur : M<sup>r</sup> Y. YACINE**

**Examineur : M<sup>r</sup> H. RADJA**

## Remerciements

Je tiens à exprimer en tout premier lieu ma gratitude et mes sincères remerciements,  
à mon promoteur, Monsieur Arezki HAMMACHE. Merci de m'avoir encadré, soutenue et si bien orienté et d'avoir été patient tout au long de ce modeste projet.

Je remercie les membres du jury pour leurs attentions et intérêt envers notre travail.

Merci de nous avoir honorés de votre présence.

Je tiens à remercier, l'ensemble des enseignants de département d'informatique, et toute personne qui a contribué de près ou de loin à l'élaboration de ce projet.

## **Dédicaces**

Je dédie ce modeste travail,

A la mémoire de mon père et mon grand-père,

A ma très chère maman qui m'ont toujours soutenue,

A ma très chère sœur,

A tous les membres de ma famille, paternelle et maternelle,

A tous mes amis,

A toute les personnes que je connais et que je n'ai pas citées.

Lydia.M

## SOMMAIRE

Introduction générale.....	1
I.1 Introduction.....	2
I.2 Présentation Android.....	2
I.2 .1 Définition Android.....	2
I.2.2 Historique et version Android.....	2
I.2.2.1 Historique.....	2
I.2.2.1 Les versions d'Android.....	3
I.2.3 Architecture Android.....	7
I.2.3.1 Applications.....	8
I.2.3.2 Framework de développement.....	9
I.2.3.3 Bibliothèques.....	9
I.2.3.4 Android Runtime.....	9
I.2.3.5 Linux Noyau.....	10
I.2.4 Les concurrents du système Android.....	10
I.3 Développement d'applications Android.....	12
I.3.1 Définition d'une Application Mobile.....	12
I.3.2 Définition d'une application Android.....	12
I.3.2 Les Composants Principaux d'Android.....	12
I.3.2.1 Intent.....	12
I.3.2.2 View.....	12
I.3.2.3 Activity.....	13
I.3.2.4 Service.....	13
I.3.2.5 ContentProvider.....	13
I.3.2.6 BroadcastReceiver.....	13
I.4 OUTILS DE DEVELOPMENT D'APPLACATION ANDROID.....	14
L'écosysteme d'android s'appuie sur deux piliers.....	14
Android Studio.....	14
I.5 Présentation de quelques applications M-Banking.....	15
I.5.1 Applications M-Banking internationales.....	15

I.5.1.1 Accès compte .....	15
I.5.1.1 .1.....	16
I.5.1.2 Application Société Générale.....	16
I.5.1.....	16
I.5.2 Applications M-Banking nationales.....	17
I.5.2.1 CCP ALGÉRIE POSTE DZ.....	17
I.6 Conclusion .....	18
II.1 Introduction.....	19
II.2 Cadre du projet .....	19
II.3 Présentation de quelques concepts de M-Banking et de E-banking .....	19
II.3.1 Présentation d'E-Banking .....	19
II.3.2 Présentation M-Banking .....	20
A)Le WAP Banking .....	21
B)L'SMS Banking.....	21
II.3.3 Les techniques de M-Banking .....	21
II.3.4 Le secteur de E-Banking en Algérie .....	23
II.4 La critique de la situation .....	23
II.4.1 La solution proposée :.....	23
II.5 Présentation de l'UML .....	24
II.5.1 Définition et Historique .....	24
II.5.1.1 Définition.....	24
II.5.1.2 Historique .....	24
II.5.2 Les diagrammes d'UML.....	25
II.5.3 Démarche de modélisation.....	27
II.6 Analyse .....	27
II.6.1 Définition des besoins.....	27
II.6.1 .1 Spécification des besoins fonctionnels .....	28
II.6.1.2 Spécification des besoins non fonctionnels .....	28
II .6.2 Identification des acteurs .....	29
II.6.3 Diagramme de contexte .....	29

II.6.4 Les diagrammes de cas d'utilisation.....	30
II.6.4.1 Diagramme de cas d'utilisation globale.....	30
II.6.4.2 Diagramme de cas d'utilisation commandé chèquiers.....	31
II.6.4.3 Diagramme de cas d'utilisation Consulter annuaire réseau agences .....	32
II.6.4.4 Diagramme de cas d'utilisation contacter banque .....	33
II.6.4.5 Diagramme de cas d'utilisation effectuer un virement vers un autre .....	34
II.6.4.6 Diagramme de cas d'utilisation modifier mot de passe .....	35
II.6.4.7 Diagramme de cas d'utilisation Affichage des informations client.....	36
II.6.4.8 Diagramme de cas d'utilisation Opposer une carte bancaire.....	37
II.7 Conception.....	38
II.7.1 Diagramme de séquence .....	38
II.7.1.1 Diagramme de séquence Authentification.....	39
II.7.1.2 Diagramme de séquence demander un chéquier.....	40
II.7.1.3 Diagramme de séquence effectuer un virement.....	41
II.7.2 Diagramme de class.....	42
II.7.3 Le modèle relationnel .....	43
II.8 Conclusion .....	44
III.1 Introduction .....	45
III.2 Outil de développement.....	45
III.2.1 Architecture de l'application .....	45
III.3 Environnement technique du travail.....	46
III.3.1 Outil de conception .....	46
III.3.2 Outils de développement .....	46
III.3.2.1 Environnement de développement : .....	47
III.3.2.1.1 Android Studio .....	47
III.3.2.1.2 Comparaison d'Android Studio et Eclipse.....	47
Présentation des outils .....	49
III.3.2.2 SGBDR MySQL.....	49

III.4 Langages de programmation .....	50
III.5 Présentation de quelques interfaces de l'application.....	51
III.5.1 Formulaire d'authentification.....	52
III.6 Conclusion.....	59
Conclusion générale .....	60

## Liste des figures

Figure I-1 Part de marché mondiale OS Mobiles (%) [1].	3
Figure I.2 Distribution des versions d'Android [3].	7
Figure I.3 l'architecture de la plateforme Android [4].	8
Figure I.4 Application « Access Compte ».	16
Figure I.6 Application «CCP ALGÉRIE POSTE DZ ».	18
Figure II.1 Vue d'UML [12].	26
Figure II.2 « Démarche de modélisation ».	27
Figure II.3 « Diagramme de Contexte ».	30
Figure II.4 « Diagramme de cas d'utilisation global ».	30
Figure II.6 Diagramme de cas d'utilisation «Consulter annuaire des agences».	32
Figure II.7 Diagramme de cas d'utilisation «contacter banque».	33
Figure II.8 Diagramme de cas d'utilisation «effectuer un virement vers un autre compte».	34
Figure II.8 Diagramme de cas d'utilisation «Modification du mot de passe».	35
Figure II.9 Diagramme de cas d'utilisation «Affichage des informations du client de la banque».	36
Figure II.10 Diagramme de cas d'utilisation «opposer une carte ».	37
Figure II.9 Diagramme de séquence «S'authentifier».	39
Figure II.11 Diagramme de séquence «Effectuer un virement».	41
Figure II.12 « Diagramme de classe».	42
Figure II.10 Diagramme de séquence «Demander un chéquier».	40
Figure III.1 : Architecture générale de l'application mobile.	46
Figure III.2 Menu l'outil d'Android Studio permettant de créer une interface graphique.	49
Figure III.3 : Page d'authentification.	52
Figure III.4 : Menu principal de l'application.	53
Figure III.5 : Affichage des mouvements sur opposer une carte.	53
Figure III.6 : Affichage des mouvements sur la carte Information.	54
Figure III.7 : Commander un chéquier.	55
Figure III.8 : Mouvement sur commander un chéquier.	55
Figure III.9 : Modification du code secret.	56
Figure III.10 : Contacter la banque par téléphone ou par mail.	57



## Liste des tableaux

Tableau I.1 Distribution des versions d'Android [3]. .....	7
Tableau I.2 : Comparaison entre les systèmes d'exploitation mobile [5]. .....	11
Tableau II.1 : Description détaillée des acteurs.....	29
Tableau II.1 « Fiche de description du cas d'utilisation Commander chèquiers ».....	31
Tableau II.2 « Fiche de description du cas d'utilisation : Consulter annuaire réseau agences ».....	32
Tableau II.4 « Fiche de description du cas d'utilisation : Contacter banque».....	33
Tableau II.5 « Fiche de description du cas d'utilisation : effectuer un virement».....	34
Tableau II.6 « Fiche de description du cas d'utilisation : Modifier mot de passe».....	35
Tableau II.7 « Fiche de description du cas d'utilisation : Affichage des informations».....	36
Tableau II.8 « Fiche de description du cas d'utilisation : Opposer une carte».....	37

*Chapitre I*

*Le Développement  
d'application*

*Mobile*

## **Introduction générale**

Le système bancaire a connu une véritable révolution en ce début du 21<sup>e</sup> siècle et ce, avec l'introduction des nouvelles technologies. Le monde de la banque est passé en un laps de temps de la banque en tant que structure et où chacun de nous avait à se déplacer pour un quelconque service à une banque en ligne grâce à des outils de dernière génération. Elle a érigé un système informatique capable d'offrir des services à distance.

Actuellement, les banques exploitent les potentialités de ces outils de communication en particulier la téléphonie mobile comme canal de distribution et comme service afin de concrétiser ses objectifs.

De nos jours, les banques offrent des services à distance via l'équipement du Smartphone qui grâce à sa richesse, le rend idéal pour gérer plusieurs processus relatifs à la banque. Les clients d'une banque se connectent à leurs comptes bancaires par l'intermédiaire de leur téléphone portable aux services de paiement par téléphonie mobile, qui couvrent un plus grand nombre de services de paiement et permet ainsi aux banques d'entrer en relation avec leur clientèle et les informer de leurs nouveaux services. Pour les clients, les services bancaires par mobile représentent un concept au potentiel considérable (pouvoir faire des virements n'importe où, n'importe quand).

Les Smartphones permettent à la clientèle de la banque de simplifier le support client et d'augmenter la satisfaction client. Dans ce cadre, nous allons essayer de réaliser une application mobile destinée aux banques Algériennes afin de constituer une interface entre le client et sa banque et offrir un ensemble de services pour la clientèle.

Notre travail est donc organisé comme suit:

Dans le premier chapitre intitulé : « Le développement d'application mobile » l'objectif est de présenter les notions de base sur le développement d'application Android, un bref historique ainsi que ses versions, son architecture, les outils de développement d'application d'Android, ses composants, présentation de quelques applications M-Banking.

Dans le deuxième chapitre intitulé : « Analyse et Conception » présente l'analyse et la conception de notre application mobile en utilisant le langage de modélisation UML.

Puis, dans le troisième chapitre intitulé « Réalisation » un aperçu sur l'environnement de développement est illustré ainsi qu'une description de certaines composantes de notre application telles que la base de données et quelques interfaces clients.

## **I.1 Introduction**

Les Smartphones sont des appareils extrêmement sophistiqués, qui fournissent des fonctionnalités en plus de celles des téléphones mobiles classiques comme la télévision, la navigation sur le web, la consultation et l'envoi de courriers électroniques, la messagerie vocale et visuelle, etc.

Le présent chapitre a pour objectif de présenter les notions de base sur les systèmes d'Android, notamment sa définition, Historique, son Architecture ainsi que quelques concepts sur le développement d'application Android.

## **I.2 Présentation Android**

### **I.2 .1 Définition Android**

Android est un système d'exploitation édité par Google pour les appareils embarqués et/ou mobiles, comme les Smartphones ou les tablettes. On le retrouve aussi dans certains ordinateurs de bord de voitures, télévisions, autoradios.

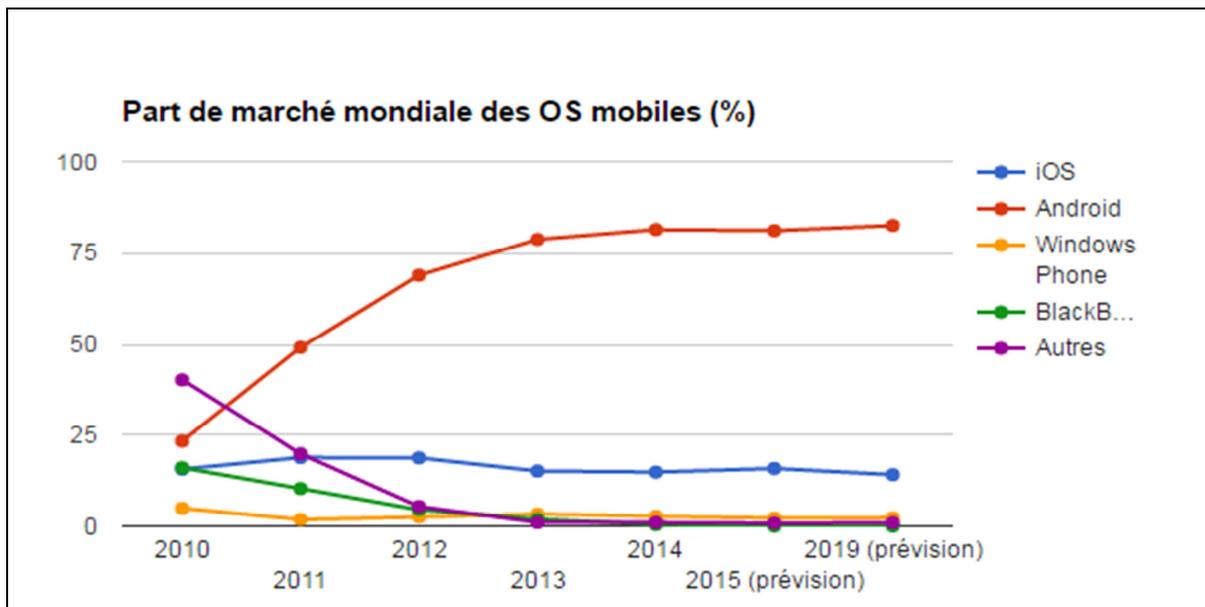
De plus, de nombreux prototypes d'appareils électroménagers, comme des réfrigérateurs ou des machines à laver, fonctionnant sous Android ont été présentés ces derniers temps, permettant ainsi de lancer une machine à l'aide de son téléphone, ou encore d'être prévenu par son frigo lorsqu'il manque certains aliments.

### **I.2.2 Historique et version Android**

#### **I.2.2.1 Historique**

Android était développé par la startup Android Inc en 2003, puis racheté par Google en 2005. Pour pouvoir réaliser un système complet, ouvert et gratuit dans le monde du mobile, une coalition de 35 entreprises évoluant dans l'univers du mobile, dont Google, a été créée. Ce rassemblement se nomme l'Open Handset Alliance (OHA) et composé aujourd'hui de 80 membres. Cette alliance a pour but de développer un système open source « c'est-à-dire dont les sources sont disponible librement sur internet » pour l'exploitation sur mobile, Android.

Android est à l'heure actuelle le système d'exploitation pour smartphones et tablettes le plus utilisé Source IDC Via [1]. L'image ci-dessous l'illustre :



**Figure I-1** Part de marché mondiale OS Mobiles (%) [1].

Les terminaux visés par Android incluent les téléphones portables, Netbook/Smartbook, tablettes multimédia, automobile, GPS, Réfrigérateur, etc.

### I.2.2.1 Les versions d'Android

Android est actuellement disponible en version 6.0. Marshmallow. Les versions se succèdent rapidement et les changements qui les accompagnent sont souvent conséquents en termes de nouvelles fonctionnalités et d'améliorations.

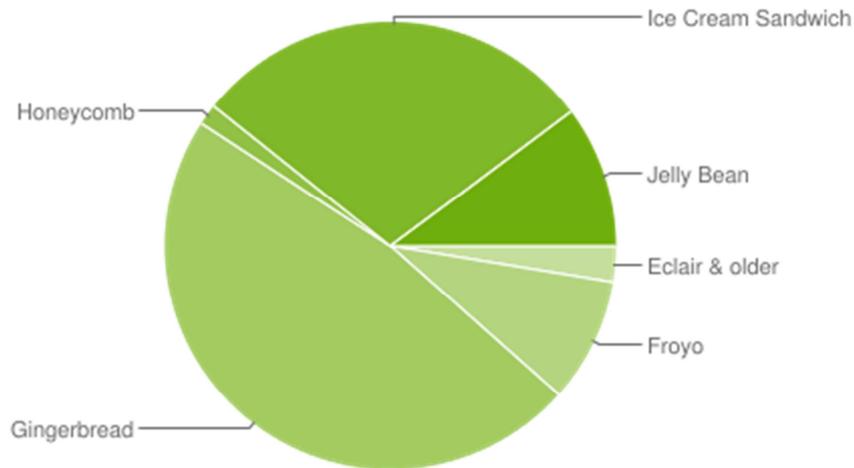
Dans ce qui suit nous aborderons les différentes versions d'Android et leur fonctionnalité :

Version	Dernière révision	Nom de code	Date de sortie	Caractéristiques (uniquement celles "Open Source" liées à l'AOSP)	Version du noyau
1.0	1.0	Aucun <sup>16</sup>	11 novembre 2007	Version connue uniquement ou presque des développeurs car c'est la version du SDK distribuée avant la sortie du premier téléphone Android.	?
1.1	1.1	Petit Four <sup>16</sup>	22 octobre 2008	Version incluse dans le premier téléphone, le HTC G1/Dream.	
1.5	mai 2010	Cupcake	30 avril 2009	Nouvelles fonctionnalités et mises à jour de l'interface graphique.	Linux 2.6.27
1.6		Donut <sup>17</sup>	15 septembre 2009		Linux 2.6.29
2.0		Eclair <sup>17</sup>	26 octobre 2009		
2.2. x	2.2.3, 2011	Froyo <sup>17</sup>	20 mai 2010	Vitesse améliorée, nouvelles fonctionnalités et mises à jour de l'interface graphique.	Linux 2.6.32
2.3. x	2.3.7, 2012	Gingerbread <sup>17,18</sup>	6 décembre 2010	Dernière version dédiée uniquement aux smartphones. Cette version est parfois utilisée sur de petites tablettes.	Linux 2.6.35
3. x.x	3.2.6, 2012	Honeycomb <sup>19</sup>	22 février 2011	Réservé aux tablettes tactiles et aux téléviseurs connectés <sup>20</sup> , cette mise à jour comprend de nombreux changements dans l'interface.	?
4.0. x	4.0.4, 2012	Ice Cream Sandwich <sup>17,21,22</sup>	19 octobre 2011	Cette nouvelle version, fortement inspirée d'Honeycomb,	Linux 3.0.1

Version	Dernière révision	Nom de code	Date de sortie	Caractéristiques (uniquement celles "Open Source" liées à l'AOSP)	Version du noyau
				unifiée <sup>22</sup> pour smartphones, tablettes et Google TV apporte de nombreux changements.	
4.1. x	4.1.2, 2012	Jelly Bean <sup>23</sup>	9 juillet 2012	Ajout d'un système de notification améliorée, de la reconnaissance vocale sans connexion internet et du « Project Butter » qui augmente la fluidité d'Android.	Linux 3.0.31
4.2. x	4.2.2, 15 février 2013		13 novembre 2012	Nouvelle interface de l'appareil photo, d'un système multi-compte uniquement sur tablette, et de Type Gesture permettant d'écrire avec le clavier rien qu'en glissant le doigt.	Linux 3.0.4
4.3. x	4.3.1, 24 juillet 2013		24 juillet 2013	Support du Bluetooth SMART (en) basse consommation et ajout de la norme AVRCP (en) 1.3, gestion multi-utilisateur plus poussée, support d'OpenGL ES 3.0, nouvelle interface de l'appareil photo, mises à jour de sécurité et SlimPort (en).	
4.4. x	4.4.4, 19 juin 2014	KitKat <sup>24,25</sup>	31 octobre 2013	Consommation en ressource moins élevée nécessitant moins de RAM, nouvelles icônes plus soignées, la barre du bas et celle de statut deviennent transparentes sur certains menus et changent de couleur en	

Version	Dernière révision	Nom de code	Date de sortie	Caractéristiques (uniquement celles "Open Source" liées à l'AOSP)	Version du noyau
				fonction du contenu affiché.	
5.0.x	5.0.2, 19 décembre 2014	Lollipop	3 novembre 2014	Material Design, moteur d'exécution ART, support du 64 bits, Android TV, Android Auto, projet Volta, économiseur de batterie, OpenGL ES 3.1, Android Extension Pack, heads-up notifications.	Linux 3.10
5.1.x	5.1.1, 21 avril 2015		9 mars 2015 <sup>26</sup>	Support de plusieurs cartes SIM, raccourci pour joindre un réseau Wi-Fi ou contrôler un appareil Bluetooth, protection par blocage en cas de perte ou vol, appel voix en haute définition, amélioration de la stabilité et des performances	
6.0	6.0.1	Marshmallow	5 octobre 2015	Autonomie en veille augmentée grâce au système "Doze", nouvelle animation de démarrage, Google Now on Tap, personnalisation par System UI Tuner, gestion des autorisations, Android Pay, support natif du capteur d'empreinte digitale.	

**Tableau I.1** Les différentes versions d'Android [29].



**Figure I.2** Distribution des versions d'Android [3].

Version	Nom	Distribution
1.6	Donut	0.2%
2.1	Eclair	2.4%
2.2	Froyo	9.0%
2.3 - 2.3.2	Gingerbread	0.2%
2.3.3 - 2.3.7	Gingerbread	47.4%
3.1	Honeycomb	0.4%
3.2	Honeycomb	1.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	29.1%
4.1	Jelly Bean	9.0%
4.2	Jelly Bean	1.2%

**Tableau I.1** Distribution des versions d'Android [3].

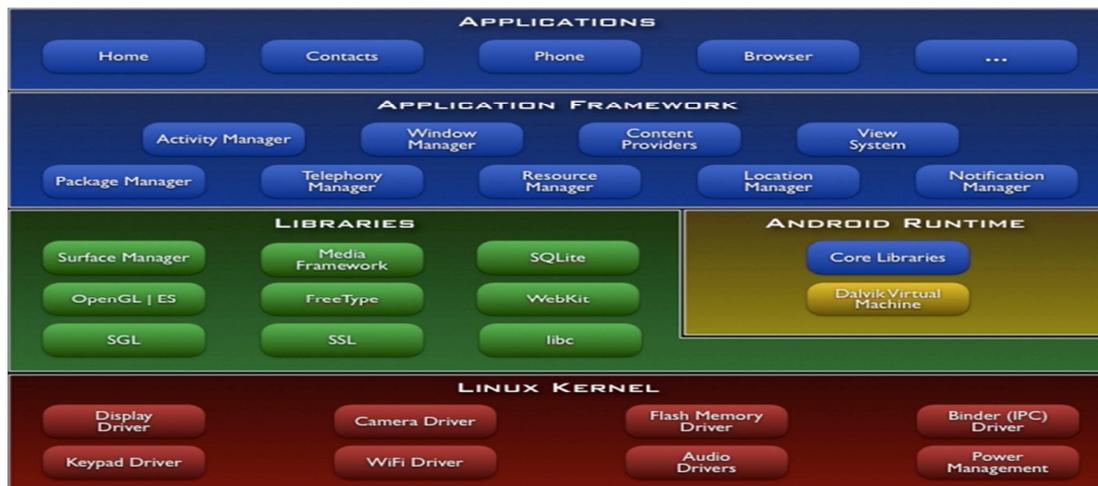
### I.2.3 Architecture Android

Android bénéficie d'une architecture en couche complète faisant de lui une plateforme riche, dédiée aux appareils mobiles. Android est en fait un système de la famille des Linux, pour une fois sans les outils GNU. L'OS s'appuie sur:

- Un noyau Linux (et ses drivers)
- Une couche d'abstraction pour l'accès aux capteurs (HAL)
- Une machine virtuelle: *Dalvik Virtual Machine*
- Des applications (navigateur, gestion des contacts, application de téléphonie...)

- Des bibliothèques (SSL, SQLite, OpenGL ES, etc...).
- Dalvik est le nom de la machine virtuelle open-source utilisée sur le système Android. Cette machine virtuelle exécute des fichiers «.dex, », « .class » classiques. Ce format évite par exemple la duplication des String constantes. La machine virtuelle utilise elle-même moins d'espace mémoire et l'adressage des constantes se fait par un pointeur de 32 bits.
- Dalvik n'est pas compatible avec une JVM du type Java SE ou même Java ME. La librairie d'accès est donc redéfinie entièrement par Google.

La figure suivante illustre les principaux composants du système d'exploitation Android. Chaque section sera décrite dans ce qui suit :



**Figure I.3** l'architecture de la plateforme Android [4]

Android est basé sur un kernel linux 2.6 au-dessus de cette couche, on retrouve les librairies C/C++ utilisées par un certain nombre de composants du système Android. Au-dessus des librairies, on retrouve l'Android Runtime. Cette couche contient les librairies coeurs du Framework ainsi que la machine virtuelle exécutant les applications. Au-dessus de la couche "Android Runtime" et des librairies coeurs, on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework, il y a les applications.

### I.2.3.1 Applications

Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur, etc. toutes écrites en JAVA.

### **I.2.3.2 Framework de développement**

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes.

Les développeurs sont libres de profiter du matériel périphérique et des informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Les développeurs ont un accès complet au même framework API utilisé par les applications de base.

L'architecture d'application est conçue pour simplifier la réutilisation des composants; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités (soumis à des contraintes de sécurité appliquées par le Framework).

Ce même mécanisme permet aux composants d'être remplacés par l'utilisateur.

Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données.
- Un gestionnaire de ressources.

### **I.2.3.3 Bibliothèques**

Couche logicielle basse pour utiliser :

- Les formats multimédia : images, audio et vidéo enregistrement comme rendu
- Les dessins 2D et 3D, bitmap et vectoriel
- Une base de données SQL (SQLite)

### **I.2.3.4 Android Runtime**

Android inclut un ensemble de bibliothèques de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java. Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik.

Dalvik a été écrite pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

### **I.2.3.5 Linux Noyau**

Android est basé sur un noyau linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X window system).

Enfin, Androïde utilise un noyau avec différents patches pour la gestion de l'alimentation, le partage mémoire, etc. Permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles. Androïde n'est pas linux mais il est basé sur un noyau linux. Pourquoi sur un noyau linux ?

Le noyau linux a un système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances.

Le modèle de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70.

De plus :

Le noyau linux fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de bibliothèques entre différents processus, le chargement et le déchargement de modules à chaud.

Le noyau linux est entièrement open source et il y a une communauté de développeurs qui l'améliorent et ajoutent des drivers.

C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un noyau linux.

## **I.2.4 Les concurrents du système Android**

Il existe plusieurs systèmes d'exploitation spécifiques aux Smartphones. Selon l'article publié en 2013 par l'International Data Corporation (IDC) [5].

Le tableau suivant illustre une comparaison entre le système Android et ses concurrents

<b>plateforme Mobile</b>	Android	iOS	Windows mobile	BlackBerry
<b>Société</b>	Google	Apple	Microsoft	Rim
<b>Caractéristiques</b>	-Code source ouvert -APIs ouvert Minimum de 128 Mo RAM -Stockage amovible - Radio et fonctionnalité de la téléphonie	- Radio et fonctionnalité de la téléphonie  -Stockage amovible	-Compatibilité avec tous les logiciels de « Windows » - Multi tâche : robuste	-Standard d'or pour courriel -Multitâches -Bonne intégration avec d'autres plateformes -Déploiement facile  -Haute Sécurité
<b>Langage de programmation</b>	Java	C	C++, C#/VB	Java
<b>IDE +SDK</b>	(Android SDK pour Eclipse Windows et Mac OS) -Netbeans (Android plugin)	XCode IDE, iOS SDK	Visual tools (gratuit)	BlackBerry JDE plugin sous visual studio et Eclipse. -BlackBerry MDS studio avec BlackBerry MDS Runtime
<b>Environnement</b>	Windows, Linux	Mac	Windows, Mac, Linux	Windows, Mac

**Tableau I.2 :** Comparaison entre les systèmes d'exploitation mobile [5].

## I.3 Développement d'applications Android

### I.3.1 Définition d'une Application Mobile

Une application mobile est un programme autonome conçu pour s'exécuter sur un terminal mobile, comme un Smartphone ou une tablette tactile.

### I.3.2 Définition d'une application Android

Une application Android est une application mobile spécifiquement développée pour les smartphones utilisant le système d'application Android. Comme les applications iPhone dont elles sont souvent des répliques, les applications Android sont de nature très variables :

- Jeux
- Mobile commerce
- Utilitaire
- Service d'information

### I.3.2 Les Composants Principaux d'Android

Le développement sur Android s'appuie sur des classes importantes du framework. Ces classes sont, en quelque sorte, les “briques” élémentaires sur lesquelles ces applications reposeront. Elles sont tellement indispensables, qu'il est tout simplement impossible de construire une application sans passer par au moins l'une d'elles. Cette partie décrit la façon globale et succincte ces différentes “briques”.

#### I.3.2.1 Intent

Les Intents sont des objets permettant de faire passer des messages contenant de l'information entre composants principaux. La notion d'Intent peut être vue comme une demande de démarrage d'un autre composant, d'une action à effectuer. La raison d'être des Intents provient du modèle de sécurité d'Android. Chaque application est en effet sandboxée. Cela veut dire qu'une application A ne peut accéder aux données d'une application B. Grâce aux Intents, les applications ont la possibilité de fournir leurs services ou données si elles le souhaitent.

#### I.3.2.2 View

Les Views sont les composants de base de l'interface graphique. Elles permettent de construire l'interface utilisateur. Les *widgets* (nom donné à des composants graphiques

“avancés” : une barre de progression, par exemple, est un widget), composants graphiques ou autres *layout* (composant permettant de placer les différents composants graphiques à l'écran) héritent en fait de cette classe élémentaire. Le rendu/dessin d'une View s'effectue par l'intermédiaire d'un Canvas (qu'on peut assimiler à une feuille transparente sur laquelle on dessine avec un crayon (Paint). Pour finir, la vue est le principal composant qui s'occupe de gérer les actions utilisateurs (appui sur l'écran, sur le clavier, etc.).

### **I.3.2.3 Activity**

Le concept d'“Activity” repose sur la notion d'interaction utilisateur. Une “Activity” représente la fenêtre ou tout simplement l'écran qui sera affiché à l'utilisateur. Elle permet également de gérer des fonctionnalités telles que l'appui sur la touche [MENU] ou l'affichage de messages d'alerte (Toast).

### **I.3.2.4 Service**

La notion de service est un avantage certain sur l'iPhone. En effet, un Service est en fait un programme tournant en tâche de fond et n'ayant pas d'interface graphique (ce qui n'est pas réalisable, à l'heure où j'écris ces lignes, sur iPhone OS). L'exemple commun illustrant au mieux cette notion est celui du lecteur mp3. Un lecteur mp3 ne nécessite pas, pour la plupart du temps, d'interface graphique et doit tourner en tâche de fond laissant la possibilité aux autres applications de travailler/s'exécuter librement.

### **I.3.2.5 ContentProvider**

Les ContentProvider sont, comme l'exprime leurs noms, des gestionnaires de données. Ils permettent de partager l'information entre applications. Imaginons une application qui permette de conserver les cartes de visite virtuelles d'un ensemble de personnes. Ces cartes de visite contiennent généralement le nom, le prénom et un moyen de contact de la personne. Un tel programme peut être créé sous forme de ContentProvider ce qui lui permettra de fournir à d'autres applications présentes sur le système les informations sur une personne. Une application tierce d'envoi de courriel peut par exemple interroger ce ContentProvider afin d'obtenir l'adresse courriel d'un contact.

### **I.3.2.6 BroadcastReceiver**

Un BroadcastReceiver est une application qui est à “l'écoute” des autres applications. Ce type d'application tente de répondre à l'Intent qui lui est adressés. Il ne fait donc rien d'autres que d'être à l'écoute des Intent envoyés par d'autres composants applicatifs.

## I.4 OUTILS DE DEVELOPPEMENT D'APPLACATION ANDROID

L'écosystème d'android s'appuie sur deux piliers

- Le langage Java
- Le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur

Le kit de développement donne accès à des exemples, de la documentation mais surtout à l'API de programmation du système et à un émulateur pour tester ses applications. Stratégiquement, Google utilise la licence Apache pour Android ce qui permet la redistribution du code sous forme libre ou non et d'en faire un usage commercial.

Le plugin *Android Development Tool* permet d'intégrer les fonctionnalités du SDK à Eclipse. Il faut l'installer comme un plugin classique en précisant l'URL du plugin. Ensuite, il faut renseigner l'emplacement du SDK (préalablement téléchargé et décompressé) dans les préférences du plugin ADT [6].

### Android Studio

Est un système d'exploitation mobile basé sur le noyau Linux et développé actuellement par Google. Le système a d'abord été conçu pour les smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions (Android TV), les voitures (Android Auto), les ordinateurs (Android-x86) et les smartwatch (Android Wear) [7].

Le système a été lancé en juin 2007 à la suite du rachat par Google en 2005 de la startup du même nom. En 2015, Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80 % de parts de marché dans les smartphones [7].

**Java Development Kit (JDK)** : désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.

Il existe plusieurs éditions de JDK, selon la plate-forme Java<sup>1</sup> considérée (et bien évidemment la version de Java ciblée) :

- JSE pour la Java 2 Standard Edition également désignée J2SE,
- JEE, sigle de Java Enterprise Edition également désignée J2EE,
- JME 'Micro Edition', destinée au marché mobiles ,
- etc.

À chacune de ces plateformes correspond une base commune de Development Kits, plus des bibliothèques additionnelles spécifiques selon la plate-forme Java que le JDK cible, mais le terme de JDK est appliqué indistinctement à n'importe laquelle de ces plates-formes.

## **Eclipse**

Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement 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.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Plusieurs logiciels commerciaux sont basés sur ce logiciel libre, comme par exemple IBM Lotus Notes 8, IBM Symphony ou Websphere Studio Application Developer [8].

## **I.5 Présentation de quelques applications M-Banking**

Au cours de ces vingt dernières années, les TIC ont connu une croissance spectaculaire. D'une société essentiellement industrielle, l'économie mondiale évolue progressivement vers une société dite de « l'information ». Cette évolution, généralement reprise sous l'appellation « nouvelle économie », se caractérise par un phénoménal potentiel de croissance, d'intégration et d'emploi. Les technologies numériques ont rendu l'accès, le traitement, le stockage et la transmission des informations, la consultation des comptes bancaires en ligne de plus en plus aisés et de moins en moins onéreux. De larges perspectives d'exploitation et de développement pour de nouveaux produits et services ont été ouvertes.

### **I.5.1 Applications M-Banking internationales**

Il existe des milliers d'applications de ce genre à l'échelle internationale et parmi celles-ci :

#### **I.5.1.1 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.

### I.5.1.1 .1 Fonctions détaillées de l'application

- Consulter les comptes (historique des 16 dernières opérations).
- Effectuer des virements, consulter la liste des virements etc.
- Donne la possibilité d'accéder directement au compte de l'utilisateur.

La figure suivante illustre des prises d'écran de l'application « Access Comptes »

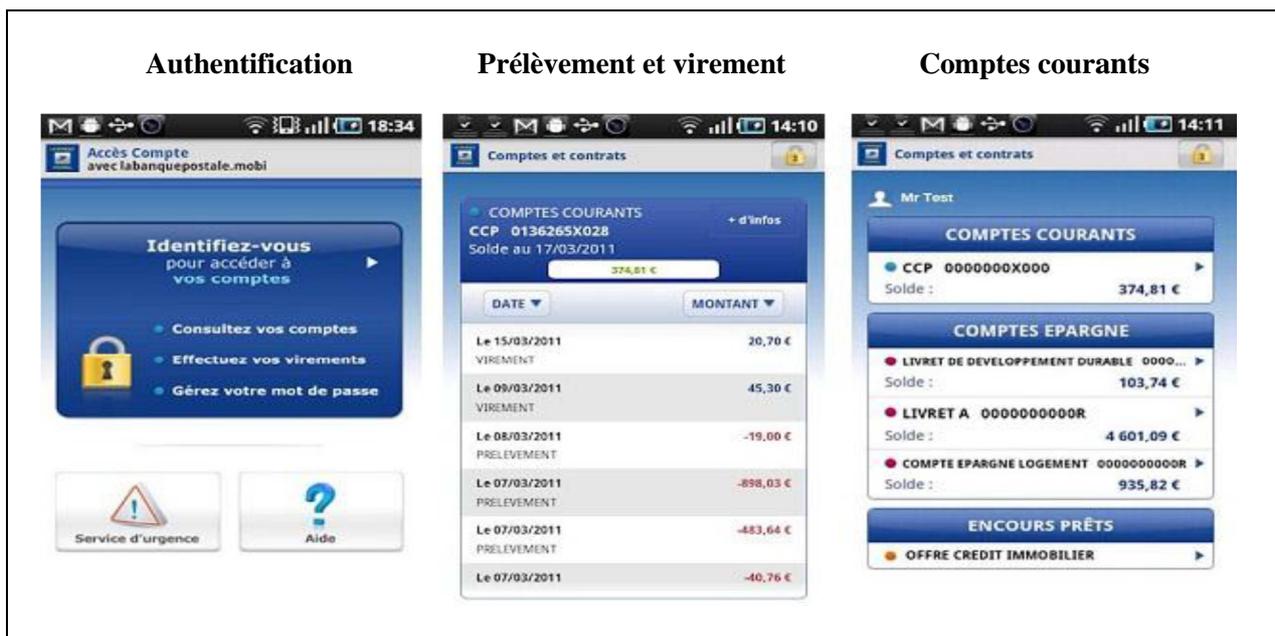


Figure I.4 Application « Access Compte ».

### I.5.1.2 Application Société Générale

La Société générale est une des principales banques françaises et une des plus anciennes. Elle fait partie des trois piliers de l'industrie bancaire française non mutualiste (aussi appelés « les Trois Vieilles ») avec LCL (Crédit lyonnais) et BNP Paribas.

#### I.5.1.2.1 Fonctionnalités

Localiser les agences et les distributeurs autour de l'utilisateur en France, gérer les dépenses, convertir l'euro et le dollar dans de nombreuses devises étrangères, suivre les informations boursières, simuler le financement des projets, retrouver les numéros des agences bancaires, envoyer des réclamations et suggestions.

La figure suivante illustre des prises d'écran de l'application « Société générale » »



Figure I.5 Application « Société Générale ».

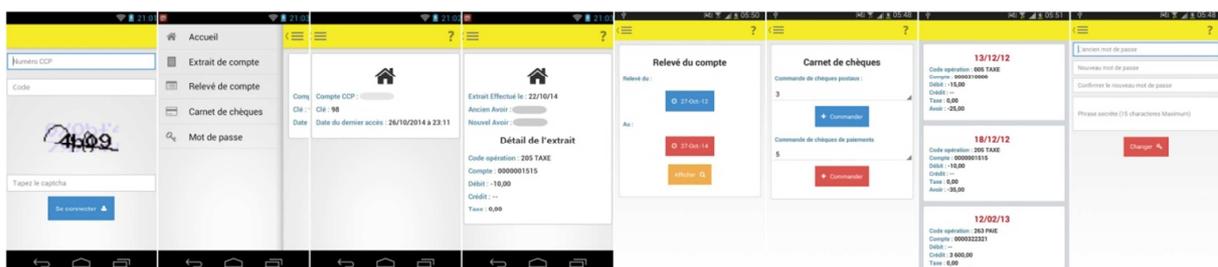
## I.5.2 Applications M-Banking nationales

Sur le plan national, il existe une application M-Banking qui est

### I.5.2.1 CCP ALGÉRIE POSTE DZ

Fournit les services utiles de la Banque Nationale Algérienne. Elle permet de : consulter le compte CCP, commander un carnet de chèque.

La figure suivante illustre la prise d'écran de l'application CCP ALGÉRIE POSTE DZ



**Figure I.6** Application «CCP ALGÉRIE POSTE DZ ».

## **I.6 Conclusion**

Ce chapitre nous a permis de donner une vue général sur les notions de base d'Android son architecture, ses différentes versions, son historique, les concurrents du système d'android ainsi que ses composants principaux, les outils de développements et enfin la présentation de quelques applications M-Banking internationales et nationales.

Au cours du chapitre qui suit, nous présenterons une conception détaillée pour l'analyse et conception.

*Chapitre II*  
*Analyse & conception*

## **II.1 Introduction**

Une étape essentielle de tout cycle de développement logiciel ou conceptuel consiste à effectuer une étude préalable. Le but de cette phase est de comprendre le contexte du système. Il s'agit d'éclaircir au mieux les besoins fonctionnels et non fonctionnels, apparaître les acteurs et identifier les cas d'utilisation. Dans ce chapitre, nous allons exprimer les besoins sous forme de diagrammes de cas d'utilisation, dans la partie conception on a présenté le cadre du projet, les concepts de m-banking et e-banking et solution proposée.

## **II.2 Cadre du projet**

De plus en plus, nombreuses sont les applications mobiles qui reposent sur une plateforme 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 ses services, c'est sa banque qui lui rend visite. Et au niveau de la banque, abaissement des coûts de transaction et de charge de service, en particulier le coût de mise en place et de maintien d'un réseau de distribution.

C'est dans ce cadre que s'inscrit notre travail qui consiste à développer une application mobile pour une banque algérienne.

## **II.3 Présentation de quelques concepts de M-Banking et de E-banking**

Dans ce qui suit nous présenterons quelques-uns concepts des au liées domaine d'E-Banking et M-Banking.

### **II.3.1 Présentation d'E-Banking**

L'E-banking ou encore le (Inter) net Banking, Web Banking ou Online Banking signifie « La banque sur Internet » [9]. Tous ces termes désignent l'utilisation de l'Internet par une institution financière en vue d'offrir à ses clients une gamme de services bancaires plus ou moins larges, allant de la simple vitrine commerciale à la gestion à distance de transactions financières. E-Banking ou banque électronique désigne donc le fait de se servir d'un outil électronique, comme l'ordinateur, pour effectuer les différentes transactions bancaires. L'E-Banking permet l'accès aux comptes, le transfert de fonds d'un compte vers un autre, l'information sur le solde, le transfert de fonds vers le compte d'un tiers, le paiement de factures, etc. Les possibilités sont nombreuses et permettent d'économiser beaucoup de temps aux gens qui utilisent ces services [10].

**Parmi les services d'E-Banking**

L'Internet Banking.

Le guichet Automatique Bancaire (GAB).

Le WAP Banking

Le SMS Banking

**II.3.2 Présentation M-Banking**

Le Mobile Banking désigne une spécificité du Mobile Business adaptée au métier de la Banque. C'est en quelque sorte une adaptation du canal de banque à distance classique de « ebanking » sur terminal mobile. D'autre part le e-banking est l'ensemble des services bancaires assurés par voie électronique « electronic banking » et donc par Internet : demande d'informations, commande de chéquier, service bancaire, consultation de comptes, virements et achats de produits financiers. De nos jours, les réseaux de connexion mobile permettent d'obtenir une connectivité plus sûre et moins coûteuse. En fonction des capacités techniques (téléphone voix, données, internet, SMS, WAP) et de la diversité des terminaux mobiles (téléphone mobile, Pocket PC, PDA, Smartphone), on peut classer le Mobile Banking comme suit [11] :

- Le PDA banking
- Le SMS banking, le SMS Payment
- Le WAP banking
- Le Mobile Internet banking

Le Mobile Embedded Software banking Le paiement par SMS est classé dans le Mobile Banking et non dans le Mobile Payment, car il ne se substitue pas à une carte de paiement bancaire de type carte bleue ou carte VISA. Les solutions de Mobile Banking peuvent servir à consulter les informations personnelles d'un client au travers d'un mobile. Elles ont ensuite permis d'effectuer des transactions bancaires de type virement et achat/revente d'actions : c'est le Mobile Banking transactionnel. Les applications mobiles « toujours disponibles et toujours connectées » vous fournissent une communication très personnalisée et ciblée par l'envoi automatique des notifications à votre clientèle. Vos applications de gestion de comptes peuvent ainsi transmettre toute la gamme d'informations, du moment d'une panne locale jusqu'aux estimations de temps de résolution. Avec l'accès automatique au support critique, vous gagnez du temps en évitant des appels inutiles.

Le Mobile Banking est un moyen de communication utilisant le téléphone portable qui s'est très fortement répandu ces dernières années, pour :

- Faciliter l'accès aux services bancaires ;
- Diversifier et améliorer l'offre de services bancaires auprès de la clientèle;
- Réduire les coûts de transaction pour les clients dans les zones éloignées ;
- Le Mobile Banking réunit les deux applications 'SMS Banking' et 'WAP Banking' ;

### **A) Le WAP Banking**

Le WAP Banking permet d'accéder au compte bancaire par l'intermédiaire d'Internet mobile. E-Banking et réseaux mobiles La sécurité des transactions effectuées par ce procédé est traitée tout comme la sécurité d'application web, avec l'envoi d'informations chiffrées depuis le mobile. Cette forme d'opérations mobiles convient aux combinés de modèles récents qui soutiennent les technologies WAP, GPRS, 3G ou EDGE. Cette technique peut être déployée de deux façons :

Soit par site web réparti entre la carte SIM du client et le serveur de la banque (certaines pages web logent dans la carte SIM du client) ; à cet effet les informations fournies sont envoyées via une connexion Internet vers le serveur de la banque qui les traite et renvoie le résultat.

Soit par site web centralisé au niveau du serveur de la banque ; à cet effet le client établit une connexion Internet entre le serveur web du client et sa carte SIM avant toute sollicitation de service.

### **B) L'SMS Banking**

Le SMS-Banking est une branche de l'E-Banking qui combine le SMS et le téléphone mobile. A ce titre, Les clients de la banque peuvent gérer leur compte, visualiser leurs soldes, demander des chèquiers, faire des virements, payer des factures et d'autres transactions bancaires en utilisant leur téléphone mobile.

## **II.3.3 Les techniques de M-Banking**

Les technique de M-banking sont des réseaux mobiles et de transmission de données, ils utilisent la technologie GSM et depuis UMTS). Ces normes internationales permettent aux équipements mobiles de bien sûr téléphoner, mais aussi d'effectuer des transmissions de données. Les techniques disponibles pour la transmission de données sont:

- L'échange de données par SMS : bien adapté à la transmission de données textuelles de petite taille.

- L'établissement d'une liaison TCP/IP qui selon la technologie (GSM, UMTS) et les options d'abonnement peut être à différentes vitesses:
- CSD (circuit switched data), proche d'un protocole modem traditionnel: permet l'échange de données mais interdit les appels téléphoniques durant la connexion en mode data.
- GPRS (General Packet Radio Service): permet l'échange de données à 64 kbps (dans la pratique plutôt 40 kbps) tout en restant disponible pour recevoir un appel: néanmoins, l'échange simultané de données et la téléphonie n'est pas possible.
- UMTS: échange de données à haute vitesse (de l'ordre de 300 kbps).
  
- Sur la liaison TCP/IP ainsi établie, les données peuvent être échangées selon tous types de protocoles:
- Le WAP, version simplifiée et plus légère du Web, dont le navigateur Wap est embarqué dans la plupart des portables commercialisés actuellement. Le Web, embarqué dans les portables haut de gamme.
- Email (SMTP), également disponible sur les portables haut de gamme.
  
- Échange de messages multimédia MMS permettant la transmission de messages composites intégrant texte, photo, sons, vidéo, etc.

Toute application spécifique, qui peut être soit embarquée sur le portable (par exemple sous forme d'application Java J2ME téléchargée

Si le portable le supporte ou sur un autre équipement (PC, Palm, etc.) utilisant le portable pour accéder au réseau.

Ces différentes techniques permettent de réaliser de nombreuses applications mobiles professionnelles dans le domaine bancaire offrir les fonctionnalités suivantes en temps réel: authentification de l'utilisateur, mise à disposition des informations utiles aux clients concernant la banque, services à la clientèle des outils liés à la banque, synthèse des comptes bancaires de type chèque, carte bancaire, placements liste des opérations d'un compte, virements internes et externes, services professionnels à forte valeur ajoutée, services destinés aux opérateurs télécoms mobiles et intégration au système d'information bancaire : messagerie, bases de données. Les Smartphones puissants actuels vous permettent d'étendre vos outils de service client en ligne aux périphériques mobiles. Pour cela, plusieurs processus sont rationalisés grâce à la mobilité, car les applications peuvent identifier et localiser automatiquement le client, en lui fournissant toutes autres données pertinentes.

### **II.3.4 Le secteur de E-Banking en Algérie**

Le secteur bancaire algérien a connu ces dernières années de nombreuses mutations (privatisation des banques publiques, arrivée de nouveaux acteurs issus du Moyen-Orient, de l'Europe etc). A ce titre, les banques algériennes doivent aujourd'hui refondre leur système d'information afin d'accélérer leurs développements dans le cadre de la modernisation de ses infrastructures et de l'amélioration constante de la qualité des prestations financières dispensées à sa clientèle.

### **II.4 La critique de la situation**

Le client cherche avant tout une réponse à ses interrogations. Il recherche une information pertinente et rapide lui permettant de satisfaire ses besoins, sans avoir à se déplacer, n'importe quel jour (y compris le week-end), à n'importe quelle heure et être en parfait accord avec la révolution des nouvelles technologies. L'élément "temps, espace" est primordial dans la délivrance de l'information. L'attente et le déplacement du client est pénible et quelquefois pour une simple opération de consultation, donc cette méthode apparaît inefficace.

#### **II.4.1 La solution proposée :**

Nous proposons une solution qui permet d'améliorer le quotidien de la clientèle algérienne, cette solution consiste en une application qui leur permettra de consulter leur compte bancaire à n'importe quel moment et en peu de temps via leurs Smartphones ou encore leurs tablettes. L'application envisagée, dans le secteur bancaire est une application mobile, sur une plateforme Android. Cette application permettra à un client d'accéder à partir d'un terminal mobile de bénéficier des services informationnels et services transactionnels sur sa banque en toute sécurité après vérification des paramètres d'accès au niveau d'un serveur distant. Le réseau internet via la suite des protocoles TCP/IP permet la connexion entre le terminal mobile, le serveur d'application et la transmission de message se fait par le langage JSON qui constitue un moyen de communication.

Le terminal mobile envoie les informations afin de s'authentifier et une demande au serveur qui decode ces données et les analyse pour décision.

## II.5 Présentation de l'UML

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Dans le cadre de notre projet on a utilisé la méthodologie UML pour la modélisation des différents diagrammes.

### II.5.1 Définition et Historique

#### II.5.1.1 Définition

UML (*Unified Modeling Language*) que l'on peut traduire par "langage de modélisation unifié", est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée afin de visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, l'UML est à présent un standard adopté par l'Object Management Group (OMG) [12] [13].

#### II.5.1.2 Historique

Les méthodes utilisées dans les années 1980 pour organiser la programmation impérative (notamment Merise) étaient fondées sur la modélisation séparée des données et des traitements. Lorsque la programmation par objets prend de l'importance au début des années 1990, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.) mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour des trois méthodes suivantes:

L'OMT de James Rumbaugh (*General Electric*) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système ;

L'OOD de Grady Booch, définie pour le *Department of Defense*, introduit le concept de paquetage (*package*) ;

L'OOSE d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou *use cases*).

Chaque méthode avait ses avantages et ses inconvénients. Le nombre de méthodes en compétition s'était réduit, mais le risque d'un éclatement subsistait. En effet, la profession

pouvait se diviser entre ces trois méthodes, créant autant de continents intellectuels qui auraient du mal à communiquer.

A partir de ces trois méthodes, une méthode commune a été développée pour fédérer leurs apports respectifs. Cette méthode l'UML (*Unified Modeling Language*) est née de cet effort de convergence. L'adjectif *unified* est là pour marquer qu'UML unifie, donc remplace. En fait, et comme son nom l'indique, UML n'a pas l'ambition d'être exactement une méthode mais un langage.

L'unification a progressé par étapes. En 1995, Booch et Rumbaugh se sont mis d'accord pour construire une méthode unifiée, *Unified Method 0.8*. En 1996, Jacobson les a rejoints pour produire

UML 0.9. Notons le remplacement du mot *méthode* par le mot *langage*, plus modeste. Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG. L'OMG adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information à objets. La version d'UML en cours en 2008 est UML 2.1.1 et les travaux d'amélioration se poursuivent. UML est donc non seulement un outil intéressant mais une norme qui s'impose en technologie à objets et à laquelle se sont rangés tous les grands acteurs du domaine, acteurs qui ont d'ailleurs contribué à son élaboration.

## **II.5.2 Les diagrammes d'UML**

La modélisation proposée par le langage UML se réalise principalement sous forme graphique, en usant de divers types de diagrammes spécifiques, répartis en deux groupes ;

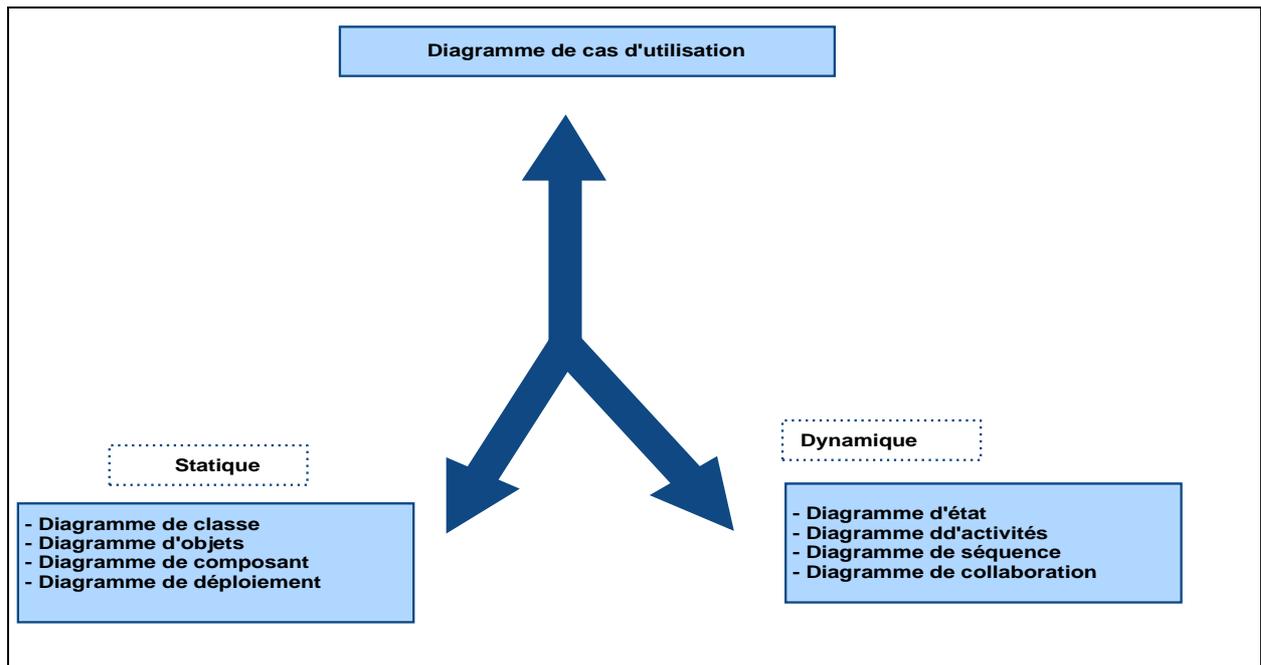


Figure II.1 Vue d'UML [12].

- Diagrammes structurels ou diagrammes statiques (UML Structure)
  - diagramme de classes (Class diagram)
  - diagramme d'objets (Object diagram)
  - diagramme de composants (Component diagram)
  - diagramme de déploiement (Deployment diagram)
  - diagramme de paquetages (Package diagram) (UMLv2)
  - diagramme de structures composites (Composite structure diagram) (UMLv2)
- Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)
  - diagramme de cas d'utilisation (Use case diagram)
  - diagramme d'activités (Activity diagram)
  - diagramme d'états-transitions (State machine diagram)
- Diagrammes d'interaction (Interaction diagram)
  - diagramme de séquence (Sequence diagram)
  - diagramme de communication (Communication diagram)
  - diagramme global d'interaction (Interaction overview diagram)
  - diagramme de temps (Timing diagram) (UMLv2)

Certains de ces diagrammes sont indépendants, alors que d'autres servent de base de travail ou bien sont la continuité d'autres diagrammes. Pour notre modélisation, nous utilisons les diagrammes suivants :

### II.5.3 Démarche de modélisation

Comme nous l'avons précisé dans la définition, l'UML n'est pas une méthode de conception, mais un langage de modélisation. De ce fait, l'UML ne dispose pas d'une démarche structurée pour la conception des SI. La démarche que nous proposons de suivre est basée sur l'utilisation de quatre diagrammes, que nous illustrons dans la figure ci-dessous

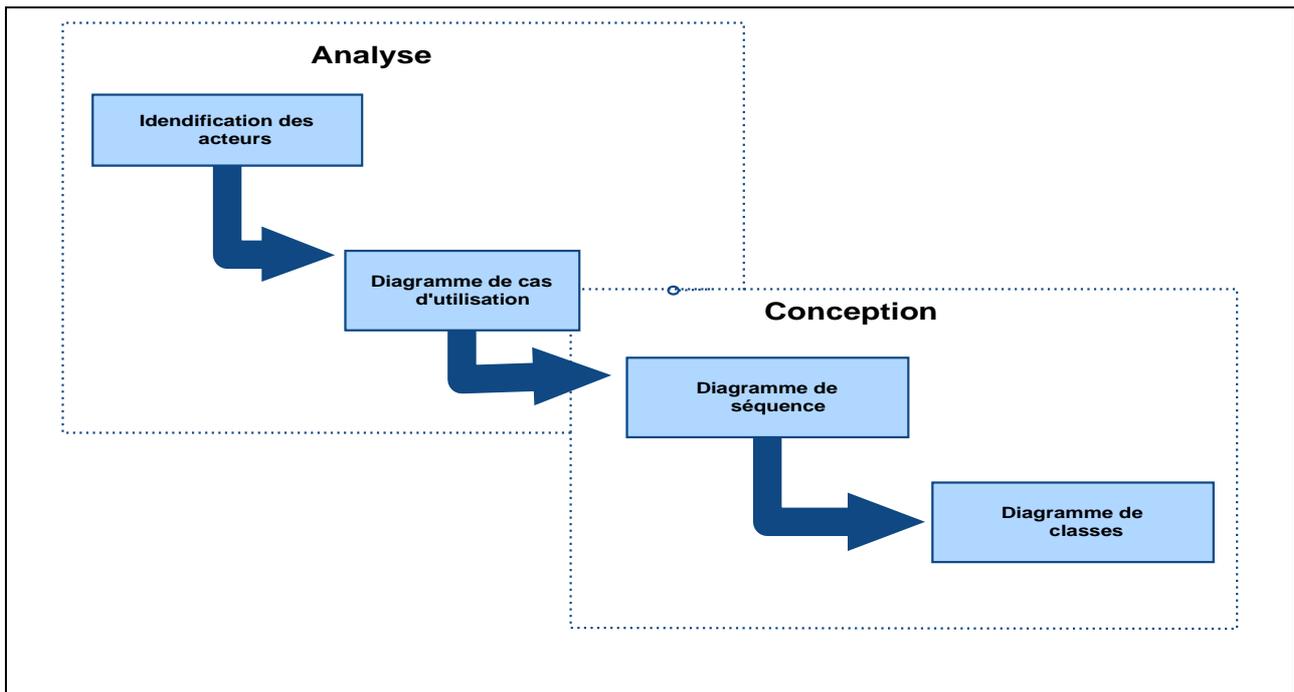


Figure II.2 « Démarche de modélisation »

## II.6 Analyse

### II.6.1 Définition des besoins

La spécification des besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante afin de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

### **II.6.1 .1 Spécification des besoins fonctionnels**

Après une étude détaillée, 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 utilisateurs :

- L'authentification de client.
- La commande de carnet de chèques.
- Virement vers un autre compte.
- Modification du mot de passe.
- Annuaire des agences.
- Contact banque par mail ou par téléphone.
- Consulter les informations personnelles.
- Consulter le solde.
- Cartes.

### **II.6.1.2 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. En ce qui concerne notre application, nous avons dégagé les besoins suivants :

- La disponibilité : l'application doit être disponible pour être utilisée par n'importe quel utilisateur.
- La sécurité de l'accès aux informations critiques : nous devons prendre en considération la confidentialité des données des 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.

- La possibilité de retourner au menu principal de l'application à partir de n'importe quelle fenêtre de celle-ci.

## II .6.2 Identification des acteurs

Un acteur est une entité qui interagit avec le système dans notre cas nous avons les acteurs décrit dans le tableau suivant.

Le tableau ci-dessus définit les acteurs de l'application Mobile Banking ainsi que leurs rôles.

L'acteur principal de l'application Mobile Banking est "Client de la Banque".

Acteur	Rôle	Type
Client banque (utilisateur mobile)	Authentification du client.  Commande de carnet de chèque.  Virement vers un autre compte.  Opposer une carte bancaire.  Annuaire des agences.  Contact banque par mail ou par téléphone.  Modification du mot de passe.	Principal

**Tableau II.1** : Description détaillée des acteurs.

## II.6.3 Diagramme de contexte

Le diagramme de contexte permet de spécifier les principaux acteurs et de définir les interactions entre eux et le système modélisé [12].

La figure suivante illustre le diagramme de contexte de notre application Mobile

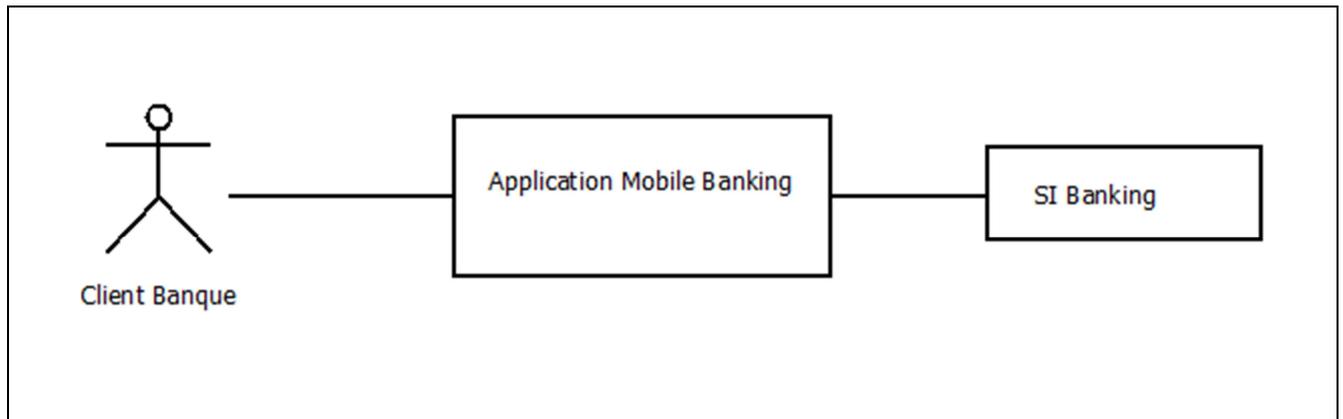


Figure II.3 « Diagramme de Contexte »

### II.6.4 Les diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux Utilisateurs du système. C'est le premier diagramme du modèle UML au niveau duquel s'assure la relation entre l'utilisateur et les objets que le système met en œuvre [12].

#### II.6.4.1 Diagramme de cas d'utilisation globale

Dans cette partie, nous illustrer ce cas d'utilisation par un diagramme de cas d'utilisation global puis une description textuelle détaillée pour chaque cas d'utilisation.

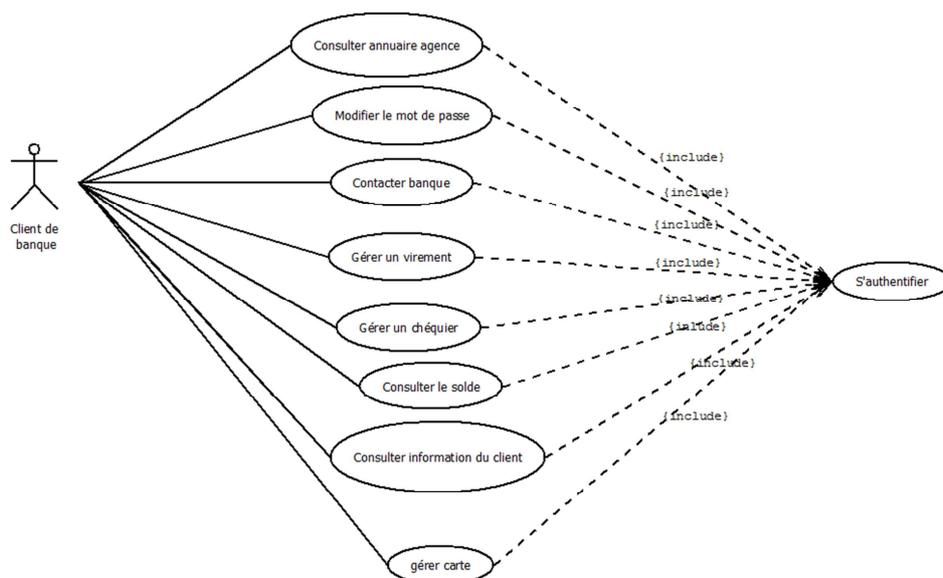


Figure II.4 « Diagramme de cas d'utilisation global »

II.6.4.2 Diagramme de cas d'utilisation commandé chéquiers

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «Commander un chéquier ». Si le client de la banque choisit de commander un chéquier, il doit être authentifié d'abord puis faire sa demande en choisissant les champs appropriés et enfin valider ses choix.

Description du diagramme de cas d'utilisation commander un chéquier

La figure illustre le diagramme de cas d'utilisation «Commander un chéquier»

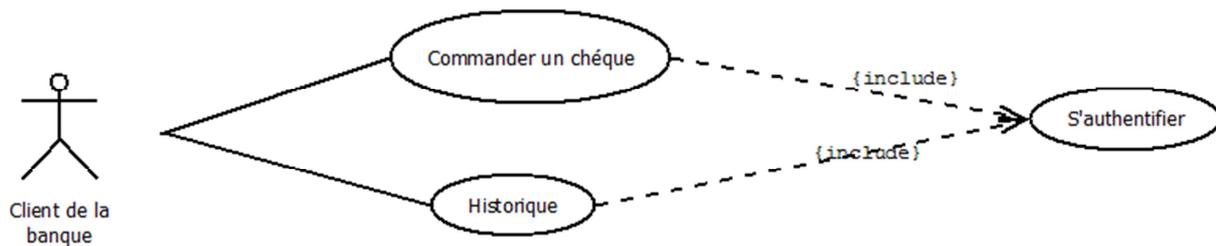


Figure II.5 « Diagramme commander chéquiers »

<b>SOMMAIRE</b>	
Titre :	<b>Commander chéquiers</b>
But :	Le client peut commander un chéquier.
Résumé :	Commander chéquiers
Acteur :	Client banque
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
Pré conditions	Post conditions
- Le client est authentifié.	- Message de confirmation de prise en charge de la demande client du chéquier
<b>Scénario nominal</b>	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système présente une page permettant de se connecter au menu de l'application. 3. L'utilisateur lance la commande de chéquiers via l'icône « commander chéquiers » 4. Le système effectue la mise à jour à la base de données du compte utilisateur et affiche message de succès.	
<b>Enchaînement alternatif</b>	
1.a Le client n'a pas rempli les champs ou les données sont incorrectes. <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur.</li> <li>2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.</li> </ol>	

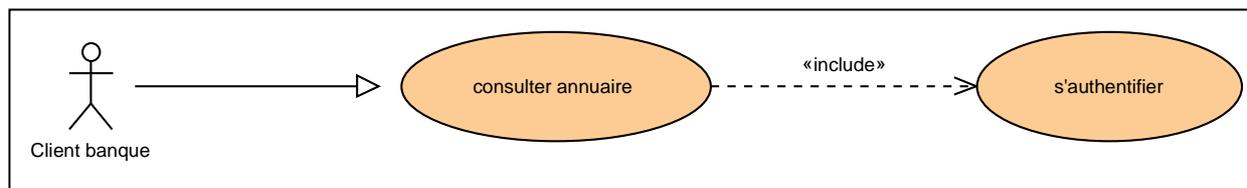
Tableau II.1 « Fiche de description du cas d'utilisation Commander chéquiers »

**II.6.4.3 Diagramme de cas d'utilisation Consulter annuaire réseau agences**

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «Consulter L'annuaire des agences». Si le client de la banque veut voir l'annuaire des agences disponible, celui-ci doit être authentifié d'abord puis la faire choisir cette option et enfin la liste des agences est lister.

**Description du diagramme de vas d'utilisation consulter annuaire des agences**

La figure illustre le diagramme de cas d'utilisation «Consulter annuaire des agences »



**Figure II.6** Diagramme de cas d'utilisation «Consulter annuaire des agences».

SOMMAIRE	
Titre :	Consulter annuaire réseau agences
But :	Le client peut consulter la liste des agences et leurs coordonnées.
Résumé :	Exposer la liste des agences commerciales de la banque.
Acteur :	Client banque
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Le client est authentifié.	- Affichage liste des agences avec leurs coordonnées.
Scénario nominal	
1. Le client se connecte à l'application par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le client clique sur l'icône « Réseau agences » 4. Le système présente une page permettant de consulter la liste des agences.	
Enchaînement alternatif	
1. Le système affiche un message d'erreur. 2. Retour à l'étape 1 du scénario nominal pour relancer à nouveau la connexion.	

**Tableau II.2** « Fiche de description du cas d'utilisation : Consulter annuaire réseau agences ».

II.6.4.4 Diagramme de cas d'utilisation contacter banque

Ce diagramme, nous permet de décrire le scénario du cas d'utilisation «Contacter banque». Si le client de la banque veut avoir des informations, il n'a qu'à adresser un Mail à l'agence ou leur passer un coup de fil.

Description du diagramme de cas d'utilisation contacter banque

La figure illustre le diagramme de cas d'utilisation «Contacter banque »

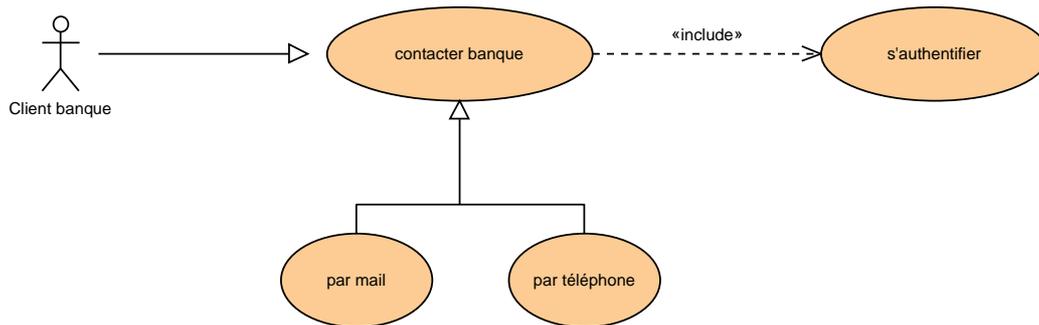


Figure II.7 Diagramme de cas d'utilisation «contacter banque»

SOMMAIRE	
Titre :	<b>Contacter banque</b>
But :	Il permet de consulter les coordonnées de la banque et de contacter la banque par mail ou par téléphone.
Résumé :	Consulter coordonnées de la banque. Contacter banque par téléphone ou appeler banque.
Acteur :	Client banque
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Le client est authentifié.	Présenter coordonnées de la banque et possibilité de le joindre par mail ou par téléphone.
Scénario nominal	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le système présente une page permettant de contacter sa banque. 4. Le client peut envoyer un mail à la banque ou appeler le service clientèle de sa banque.	
Enchaînement alternatif	

Tableau II.4 « Fiche de description du cas d'utilisation : Contacter banque».

II.6.4.5 Diagramme de cas d'utilisation effectuer un virement vers un autre

Compte

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Effectuer un virement ». Si le client de la banque choisit de faire un virement, il doit choisir le compte auquel il veut faire ce virement.

Description du diagramme de cas d'utilisation effectuer un virement

La figure illustre le diagramme de cas d'utilisation « Effectuer un virement»

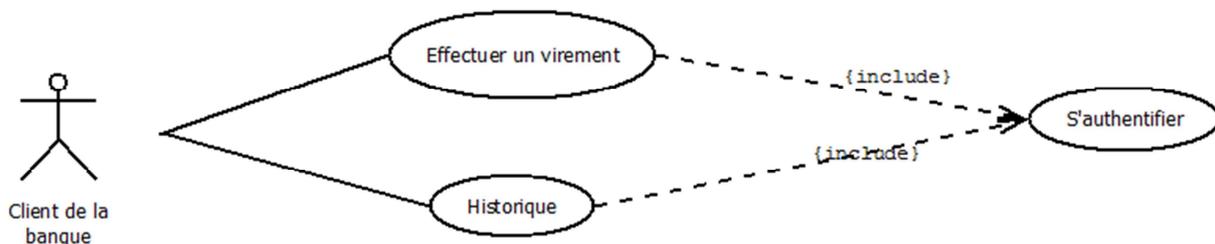


Figure II.8 Diagramme de cas d'utilisation «effectuer un virement vers un autre compte».

SOMMAIRE	
Titre :	<b>Effectuer un virement</b>
But :	Il permet de faire un virement vers un autre compte ainsi que de voir l'historique de toute ses opérations.
Résumé :	Effectuer un virement vers un autre compte. Consulter l'historique.
Acteur :	Client banque
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Le client est authentifié.	Possibilité de faire un virement vers un autre compte et consulter historique.
Scénario nominal	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le système présente une page permettant pour faire un virement de son compte vers un autre compte. 4. Le client peut faire un virement vers un autre compte. 5. Le client peut consulter l'historique de ses virements.	
Enchaînement alternatif	

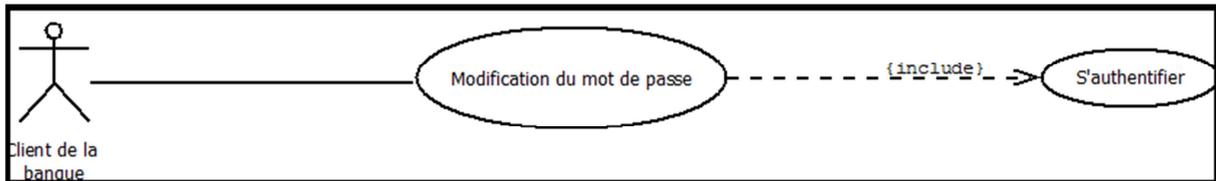
Tableau II.5 « Fiche de description du cas d'utilisation : effectuer un virement».

**II.6.4.6 Diagramme de cas d'utilisation modifier mot de passe**

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «Modifier un mot de passe » le client doit être authentifié pour pouvoir le changer.

**Description du diagramme de cas d'utilisation modifier un mot de passe**

La figure illustre le diagramme de cas d'utilisation « Modification du mot de passe»



**Figure II.8** Diagramme de cas d'utilisation «Modification du mot de passe».

<b>SOMMAIRE</b>	
Titre :	<b>Modification du mot de passe</b>
But :	Il permet de faire la modification du mot de passe du client de la banque.
Résumé :	Modifier le mot de passe du client
Acteur :	Client banque
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
Pré conditions	Post conditions
- Le client est authentifié.	Possibilité de modifier le mot de passe
Scénario nominal	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le système présente une page permettant de modifier le mot de passe.	
Enchaînement alternatif	

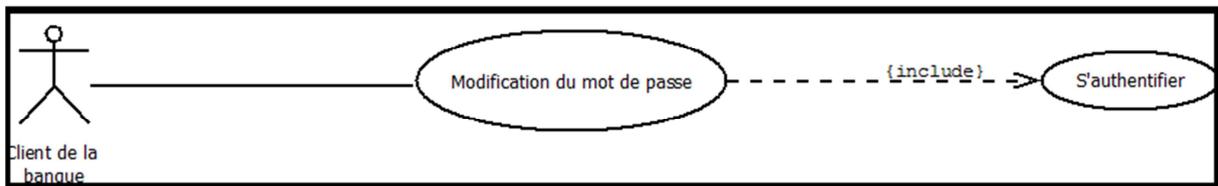
**Tableau II.6** « Fiche de description du cas d'utilisation : Modifier mot de passe».

**II.6.4.7 Diagramme de cas d'utilisation Affichage des informations client**

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «Affichage des informations client » le client doit être authentifié pour pouvoir changer son mot de passe.

**Description du diagramme de cas d'utilisation affichage des informations du client**

La figure illustre le diagramme de cas d'utilisation « Affichage des informations du client de la banque»



**Figure II.9** Diagramme de cas d'utilisation «Affichage des informations du client de la banque».

<b>SOMMAIRE</b>	
Titre :	<b>Affichage des informations client</b>
But :	Il permet de consulter les informations personnelles.
Résumé :	Consulter information
Acteur :	Client banque
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
Pré conditions	Post conditions
- Le client est authentifié.	Présenter les informations du client et possibilité de les consulter.
Scénario nominal	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le système présente une page permettant de consulter ses informations.	
Enchaînement alternatif	

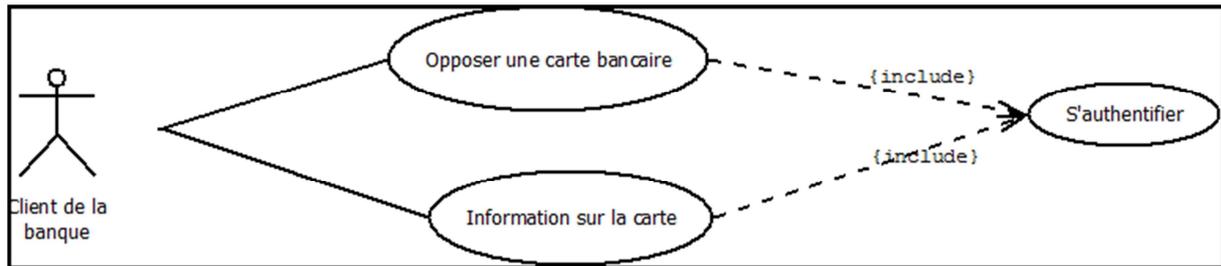
**Tableau II.7** « Fiche de description du cas d'utilisation : Affichage des informations».

**II.6.4.8 Diagramme de cas d'utilisation Opposer une carte bancaire**

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «opposer une carte » le client doit être authentifié pour avoir les informations de la carte et pouvoir opposé la carte.

**Description du diagramme de cas d'utilisation opposer une carte**

La figure illustre le diagramme de cas d'utilisation « opposer une carte »



**Figure II.10** Diagramme de cas d'utilisation «opposer une carte ».

<b>SOMMAIRE</b>	
<b>Titre :</b>	<b>Opposer une carte</b>
<b>But :</b>	Il permet de faire opposition d'une carte. Consulter les informations de la carte.
<b>Résumé :</b>	Consulter information
<b>Acteur :</b>	Client banque
<b>DESCRIPTION DES ENCHAINEMENTS</b>	
<b>Pré conditions</b>	<b>Post conditions</b>
- Le client est authentifié.	Faire opposition a la carte. Affichage des informations de la carte.
<b>Scénario nominal</b>	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le système présente une page permettant de consulter ses informations. 4. Le système présente une page qui permettra de faire l'opposition de la carte.	
<b>Enchaînement alternatif</b>	

**Tableau II.8** « Fiche de description du cas d'utilisation : Opposer une carte».

## **II.7 Conception**

La conception est la deuxième étape dans ce chapitre. Elle se traduit par les descriptions des diagrammes de cas d'utilisation, séquences et le diagramme de classe global[12].

### **II.7.1 Diagramme de séquence**

Les diagrammes de séquence servent à illustrer les cas d'utilisations décrits précédemment. Ils permettent de représenter la succession chronologique des opérations réalisées par un acteur et qui font passer d'un objet à un autre pour représenter un scénario. Dans cette partie, nous allons décrire les scénarios les plus importants ainsi que leurs représentations par les diagrammes de séquence.

II.7.1.1 Diagramme de séquence Authentification

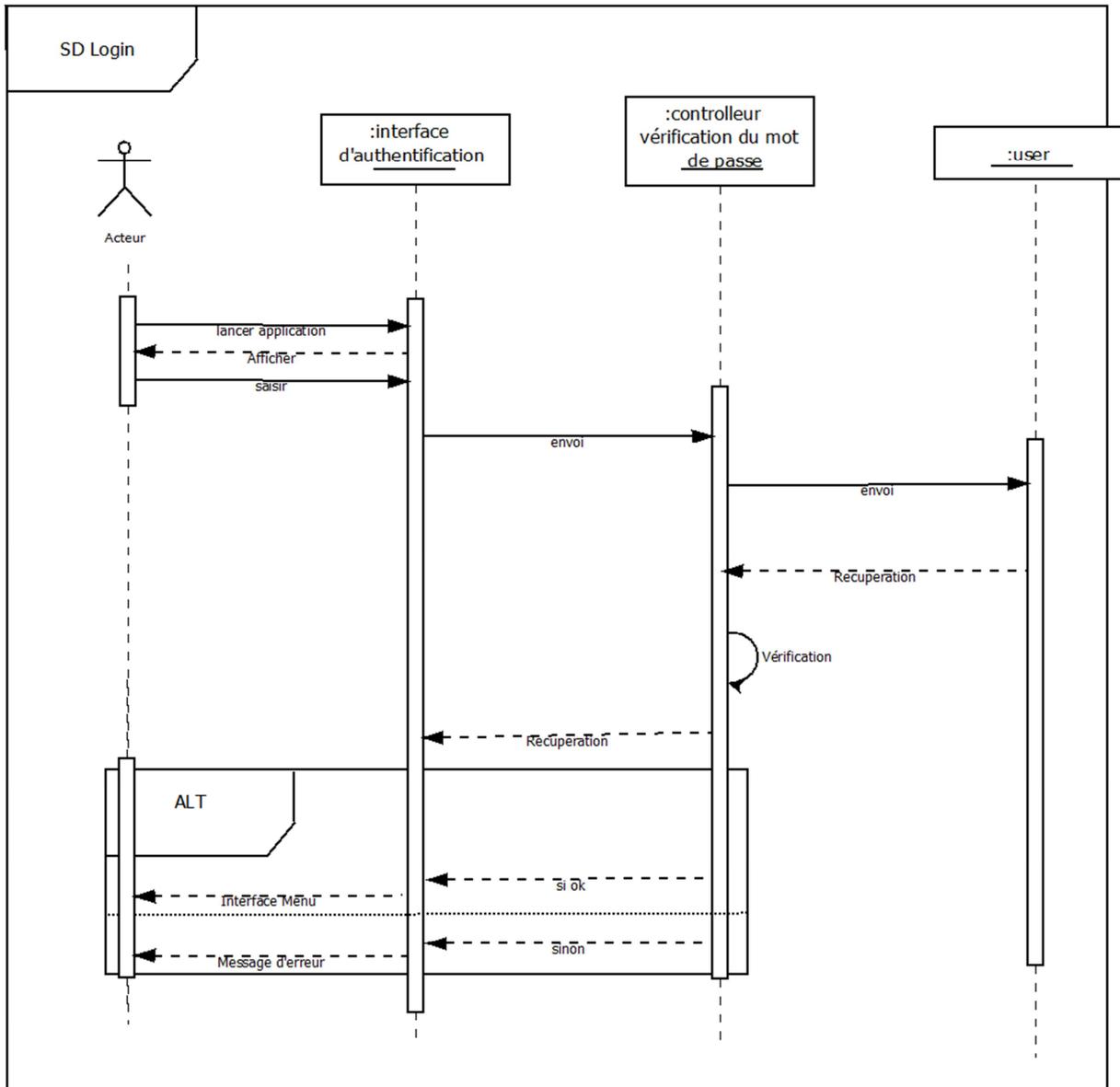


Figure II.9 Diagramme de séquence «S’authentifier».

II.7.1.2 Diagramme de séquence demander un chéquier

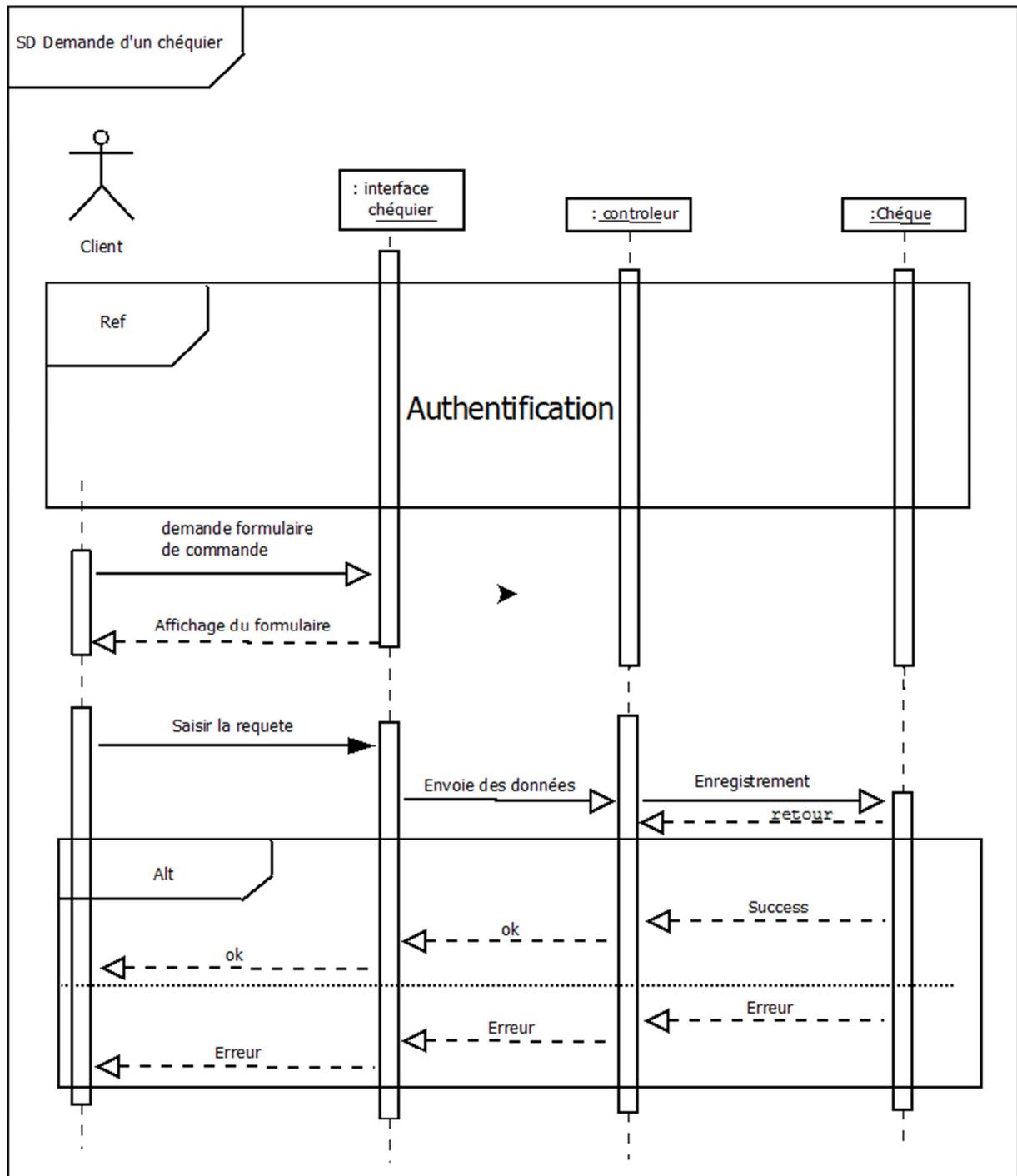


Figure II.10 Diagramme de séquence «Demander un chéquier»

II.7.1.3 Diagramme de séquence effectuer un virement

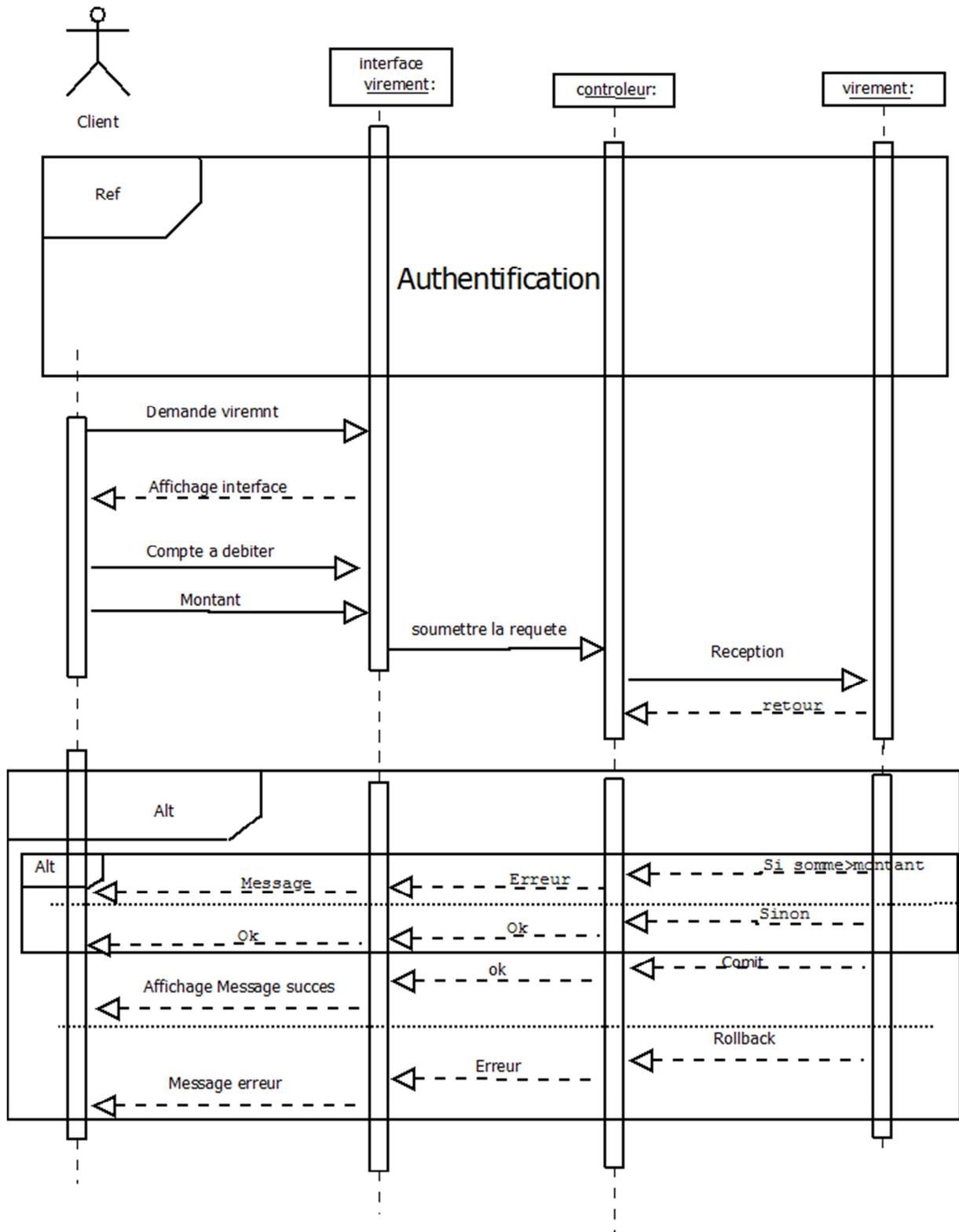


Figure II.11 Diagramme de séquence «Effectuer un virement».

II.7.2 Diagramme de class

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation) [12] [13].

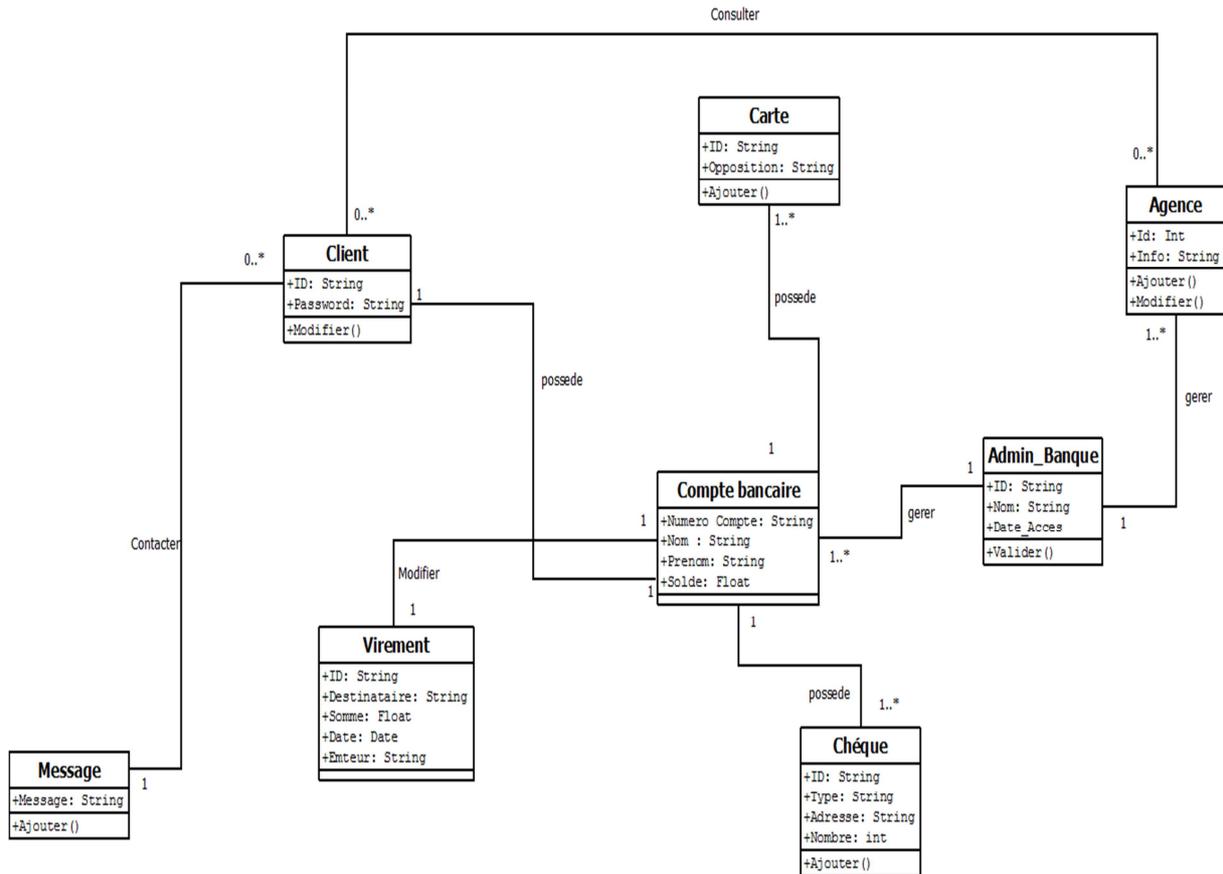


Figure II.12 « Diagramme de classe »

### II.7.3 Le modèle relationnel

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel et ce, en utilisant des règles de passages de l'UML vers le relationnel.

#### A) Concepts relationnels :

Les concepts relationnels utilisés dans le modèle relationnel de données sont [14]

- **Domaine** : Un domaine est un ensemble de valeur que peut prendre une donnée.
- **Relation** : Une relation est le sous ensemble de produit cartésien de domaine, ce sous ensemble sera désigné par un nom qui sera celui de la relation.
- **Attribut** : c'est une colonne d'une relation, caractérisé par un nom.
- **Tuple** : c'est la liste des valeurs d'une ligne d'une relation.
- **Cardinalité** : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.
- **L'arité** : est le nombre d'attributs d'une relation.
- **Clé** : On distingue trois types de clés:
- **Clé primaire** : ensemble d'attributs dont les valeurs permettent de distinguer les n-uplets les uns des autres (notion d'identifiant).
- **Clé étrangère** : Attribut qui est clé primaire d'une autre entité.
- **Clé candidate** : Ensemble minimum d'attributs susceptibles de jouer le rôle de la clé primaire

#### B) Règles de passage

##### Règles relatives aux objets

Règle I : Toute entité est traduite en une table relationnelle dont les caractéristiques sont les suivantes :

- le nom de la table est le nom de l'entité ;
- la clé de la table est l'identifiant de l'entité ;
- les autres attributs de la table forment les autres colonnes de la table

**Règle II** : Toute relation binaire plusieurs à plusieurs est traduite en une table relationnelle dont les caractéristiques sont les suivantes :

- le nom de la table est le nom de la relation ;
- la clé de la table est formée par la concaténation des identifiants des entités participant à la relation ;

- les attributs spécifiques de la relation forment les autres colonnes de la table.

**Règle III** : Toute relation binaire un à plusieurs est traduite :

Soit par un report de clé : l'identifiant de l'entité participant à la relation côté N est ajoutée comme colonne supplémentaire à la table représentant l'autre entité. Cette colonne est parfois appelée *clé étrangère*. Le cas échéant, les attributs spécifiques à la relation sont eux aussi ajoutés à la même table ;

Soit par une table spécifique dont les caractéristiques sont les suivantes :

- Le nom de la table est le nom de la relation ;
- La clé de la table est l'identifiant de l'entité participant à la relation côté 1 ;
- Les attributs spécifiques de la relation forment les autres colonnes de la table.

**Règle 4** : Toute relation binaire un à un est traduite, au choix, par l'une des trois solutions suivantes :

- Fusion des tables des entités qu'elle relie;
- Report de clé d'une table dans l'autre;
- Création d'une table spécifique reliant les clés des deux entités.

## **II.8 Conclusion**

Dans ce chapitre nous avons conçu une application Mobile pour une banque en se basant sur les diagrammes du langage UML, à savoir le diagramme de cas d'utilisation, le diagramme de séquences et le diagramme de classe. Dans le prochain chapitre nous mettrons en pratique ce système.

# *Chapitre III*

## *Réalisation*

### III.1 Introduction

Dans cette partie, nous allons décrire tous les outils matériels et logiciels de développement et de conception, avec lesquels cette application a été réalisée.

Nous commencerons par une brève description des outils de développements, de l'architecture de l'application mobile, et finir par les outils de développement utilisé pour réaliser ce projet tout en expliquant les différentes fonctionnalités des interfaces.

### III.2 Outil de développement

Dans cette partie, nous décrivons tous les outils matériels et logiciels de développement et de conception, avec lesquels cette application a été réalisée. Nous allons commencer par l'architecture de notre application mobile, les outils de développement utilisé pour réaliser ce projet et nous allons finir par présenter quelques-unes des interfaces de cette application.

#### III.2.1 Architecture de l'application

Nous décrivons dans Cette section le fonctionnement de l'application mobile «MyBank Mobile Banking». Elle est connectée à un serveur de bases de données distant, via Internet, afin de récupérer les données en utilisant des requêtes SQL, ce qui nécessite alors l'intégration d'un serveur web entre l'application client et le serveur de bases de données.

D'où l'architecture de cette application mobile est une architecture 3-tiers partagée entre: Le client Android : Il s'agit du demandeur de ressources et consommateurs des web services. Le serveur Web : Il permet de gérer la communication entre le client Android et le serveur de base de données.

Le serveur de base de données : Il permet de fournir les données au serveur web. Afin de gérer la base de données, l'exécuter des scripts était en PHP (en utilisant le protocole HTTP du système Android) et codé les données dans le format JSON.

La figure ci-dessous schématise l'architecture générale cette application mobile.



Figure III.1 : Architecture générale de l'application mobile.

### III.3 Environnement technique du travail

La configuration matérielle des machines de développement et de déploiement utilisées ainsi que les différents logiciels qui y sont installés :

Processeur Intel Core i3, RAM 4GB, Système d'exploitation Windows 7 Entreprise (x64)  
Logiciels installés Android studio, UML Dia, WampServer, Apache Tomcat Server Mysql.

#### III.3.1 Outil de conception

- **Dia**

Dia est un logiciel édité par l'entreprise Rational Machines pour créer et éditer les différents diagrammes du modèle UML (Unified Modeling Language) d'un logiciel.

#### III.3.2 Outils de développement

Dans cette partie nous allons décrire les outils de développement utilisé pour réaliser ce projet.

### III.3.2.1 Environnement de développement :

Dans cette partie nous allons définir l'environnement de développement utilisé pour réaliser ce projet, faire une comparaison entre Android Studio et Eclipse.

#### III.3.2.1.1 Android Studio

Android studio est l'environnement de développement officiel de Google qui remplace l'IDE d'Eclipse (avec donc exactement les mêmes fonctionnalités) depuis le 8 décembre 2014.

#### III.3.2.1.2 Comparaison d'Android Studio et Eclipse

Même si ces deux possibilités permettent de développer des applications complètes on peut noter **des différences**

SDK	ADT (Eclipse)	Android Studio
Facilité d'installation	Moyen	Simple
Langue	Nombreuses	Anglais
Performance	Peut être lourd	Rapide
Système de build	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion et refactoring	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	A venir

**Gradle** a pour fonctionnalités de:

Permettre la construction de projet

Permettre d'écrire des tâches de construction utilisant le langage Groovy (tests unitaires, ...)

Faciliter l'intégration des plugins (comme Google l'a fait pour Android Studio)

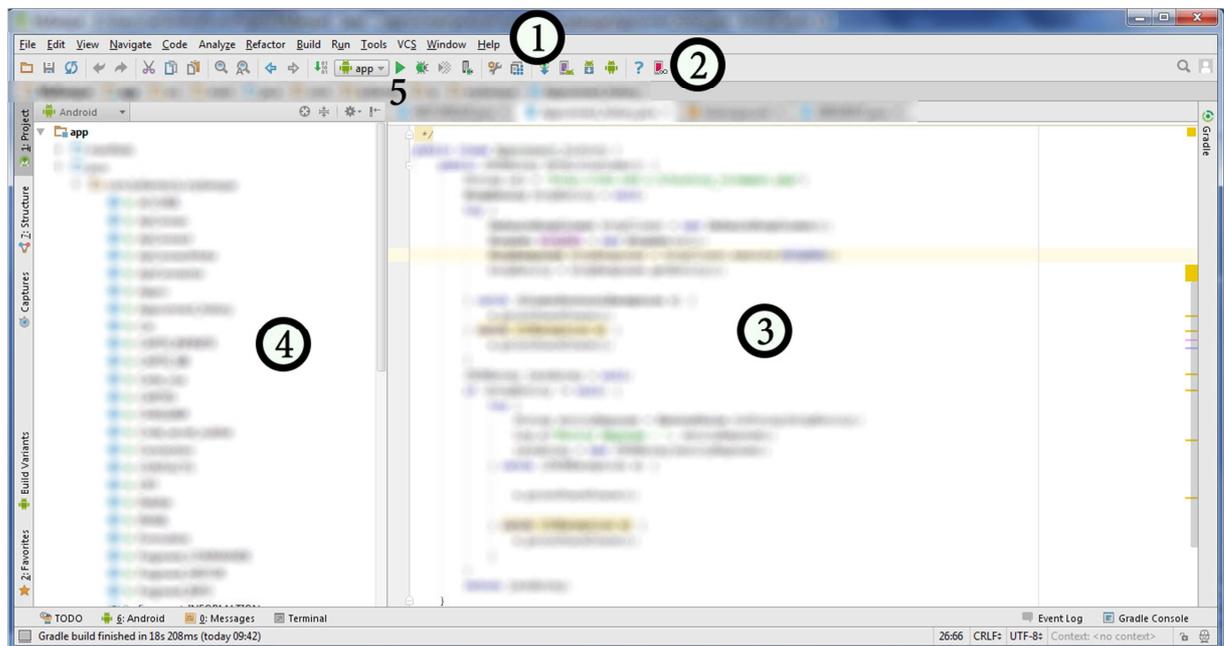
Gérer automatiquement des dépendances

**Maven** est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier.

**Ant** (Another Neat Tool) est l'outil de base permettant d'automatiser des tâches lourdes et répétitives. Ces deux outils représentent l'équivalent en Java de l'outil make d'UNIX.

Comme indiqué précédemment, le développement de l'application sur Android s'est fait dans l'environnement de développement : Android Studio. Le langage informatique fut essentiellement en Java mais aussi en XML (eXtensible Markup Language) pour la mise en page.

## Présentation des outils



**Figure III.2** Menu l’outil d’Android Studio permettant de créer une interface graphique.

1. Les Menu
2. Les boutons raccourcis
3. L’éditeur de code
4. Explorateurs de fichiers
5. Le bouton Lancer

### III.3.2.2 SGBDR MySQL

- **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

- **Caractéristique :**

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multiutilisateurs.

C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Un logiciel qui intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tierces (par exemple en C# ou php), alors il n'y a pas besoin d'acquérir une licence payante MySQL. Ce type de licence double est utilisé par d'autres produits comme le framework de développement de logiciels Qt (pour les versions antérieures à la 4.5)

- **WampServer 64 bits**

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 les bases de données.

### III.4 Langages de programmation

- **Java**

Java est un langage de programmation orientée objet, développé par Sun Microsystems permettant de créer des logiciels compatibles avec des nombreux systèmes d'exploitation et donne aussi la possibilité de développer des programmes pour téléphones portables.

- **PHP**

PHP (Hypertext Preprocessor) est un langage de programmation informatique essentiellement utilisé pour produire des pages web dynamiques via un serveur HTTP.

- **REST**

REST (Representational State Transfer) est un style d'architecture réseau pour Web Services qui met l'accent sur la définition de ressources identifiées par des URI, et utilise les messages du protocole HTTP pour définir la sémantique de la communication client/serveur:

- GET pour le rapatriement d'une ressource ;
- POST pour une création.

REST n'est pas vraiment un standard. Il n'existe donc pas de spécifications de REST. Il faut comprendre le style REST et ensuite concevoir des applications ou des services Web selon ce style. Bien que REST ne soit pas un standard, il utilise des standards. En particulier :

- URI comme syntaxe universelle pour adresser les ressources
- HTTP un protocole sans état (stateless) avec un nombre très limité d'opérations, Des liens hypermedia dans des documents (X)HTML et XML pour représenter à la fois le contenu des informations et la transition entre états de l'application, Les types MIME comme text/xml, text/html, image/jpeg, application/pdf, video/mpeg pour la représentation des ressources.

- **JSON**

JSON (JavaScript Object Notation) un format d'encodage de données structurées en JavaScript qui date de 2002. Il permet de représenter de l'information structurée.

Il n'est pas rare d'associer le couple REST/JSON pour des raisons de simplicité. Son avantage est de fournir un support pour une écriture simple et légère au format texte, relativement compréhensible par les développeurs JavaScript, mais aussi - et surtout - d'être nativement interprété.

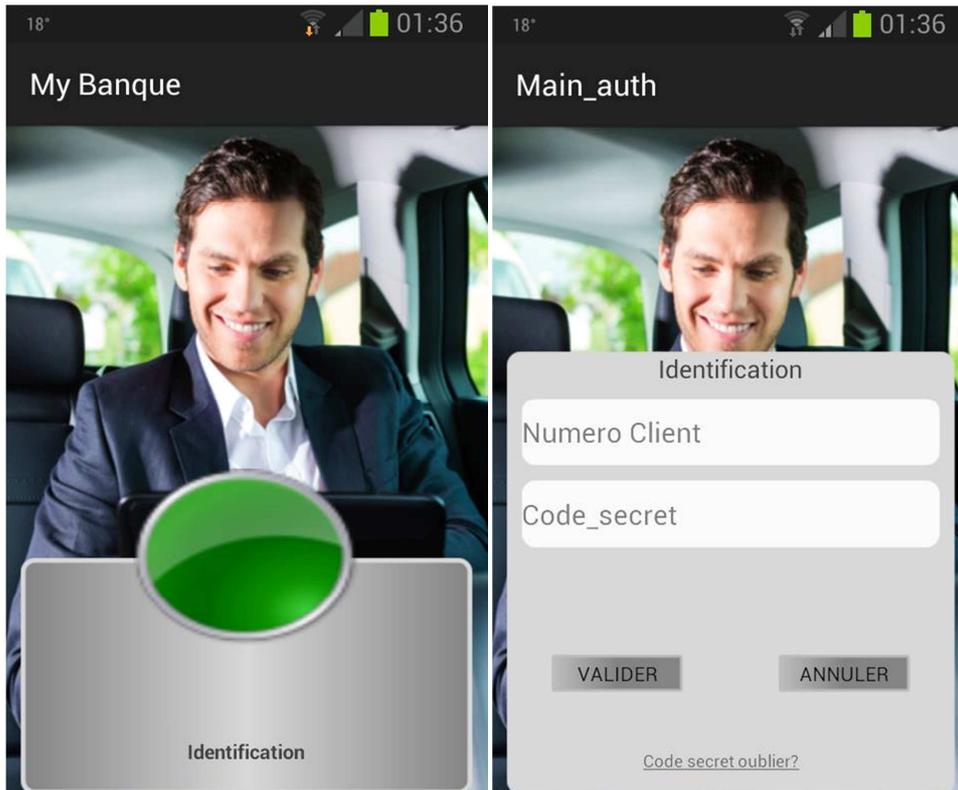
### **III.5 Présentation de quelques interfaces de l'application**

Nous Passons maintenant à la présentation de quelques unes des interfaces de notre application.

### III.5.1 Formulaire d'authentification

Une des premières étapes de cette application consiste en l'authentification, Afin d'accéder aux fonctionnalités de l'application, un écran d'authentification s'offre a l'utilisateur dès l'exécution de cette dernière, toute ressource sollicitée par un appelant est rendue accessible :

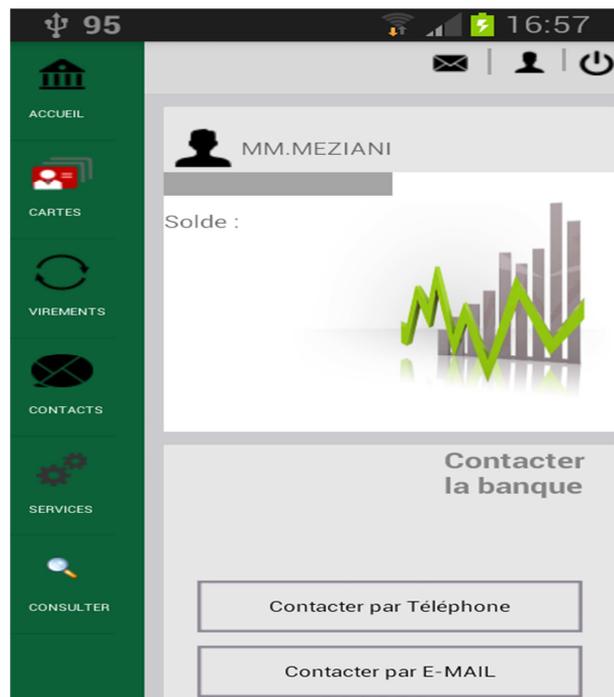
S'il s'est identifié



**Figure III.3 :** Page d'authentification.

Une fois l'abonné authentifié, il peut accéder aux fonctionnalités de l'application. La figure suivante illustre l'interface de Menu principal de l'application

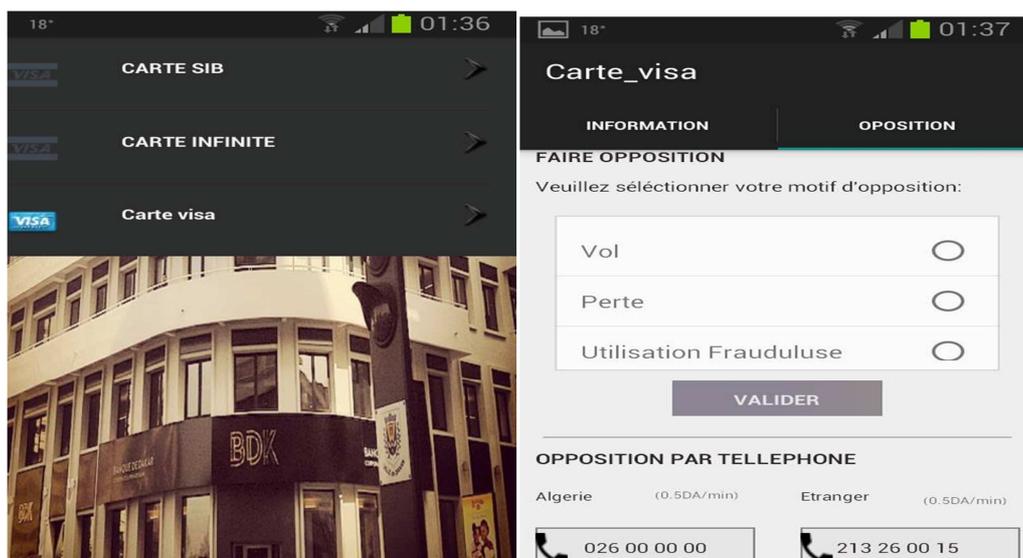
La figure suivante est l'interface du Menu principal de l'application



**Figure III.4 :** Menu principal de l'application.

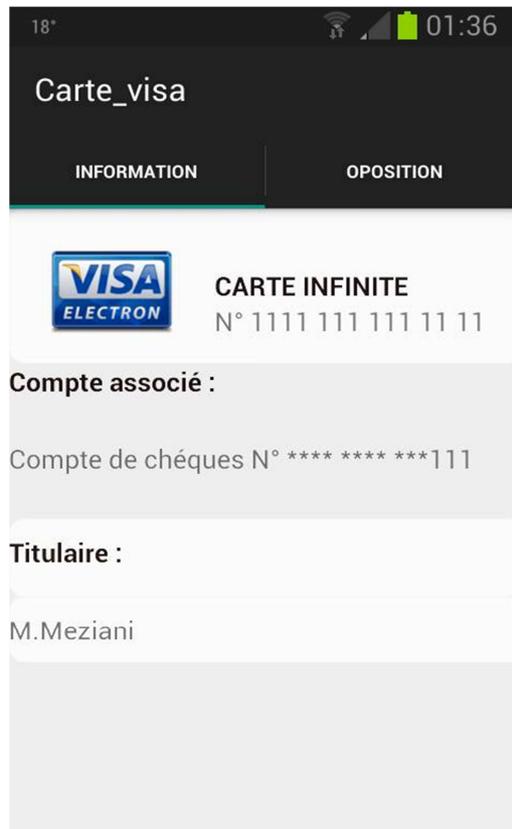
En choisissant l'une des fonctionnalités du menu principal par exemple "CARTES", le client peut consulter les informations de sa carte et faire aussi opposition en cas de perte, vol ou d'utilisation frauduleuse.

L'interface suivante nous permet de faire opposition à une carte bancaire et nous permet aussi de voir les informations de cette dernière.



**Figure III.5 :** Affichage des mouvements sur opposer une carte.

Cette interface nous permet de voir les informations concernant les différentes cartes que le client peut avoir, ainsi que le nom de la carte, compte associé ainsi que le titulaire pour pouvoir faire une opposition sur celle-ci en cas de perte, de vol et de fraude.



**Figure III.6 :** Affichage des mouvements sur la carte Information.

Cette fonctionnalité permet à l'utilisateur de commander un chéquier et faire un choix sur le type de chèque à commander (portefeuille, ordinaire, talon), choisir aussi ou celui-ci doit être livré, dans un deuxième mouvement avoir l'historique des commandes.

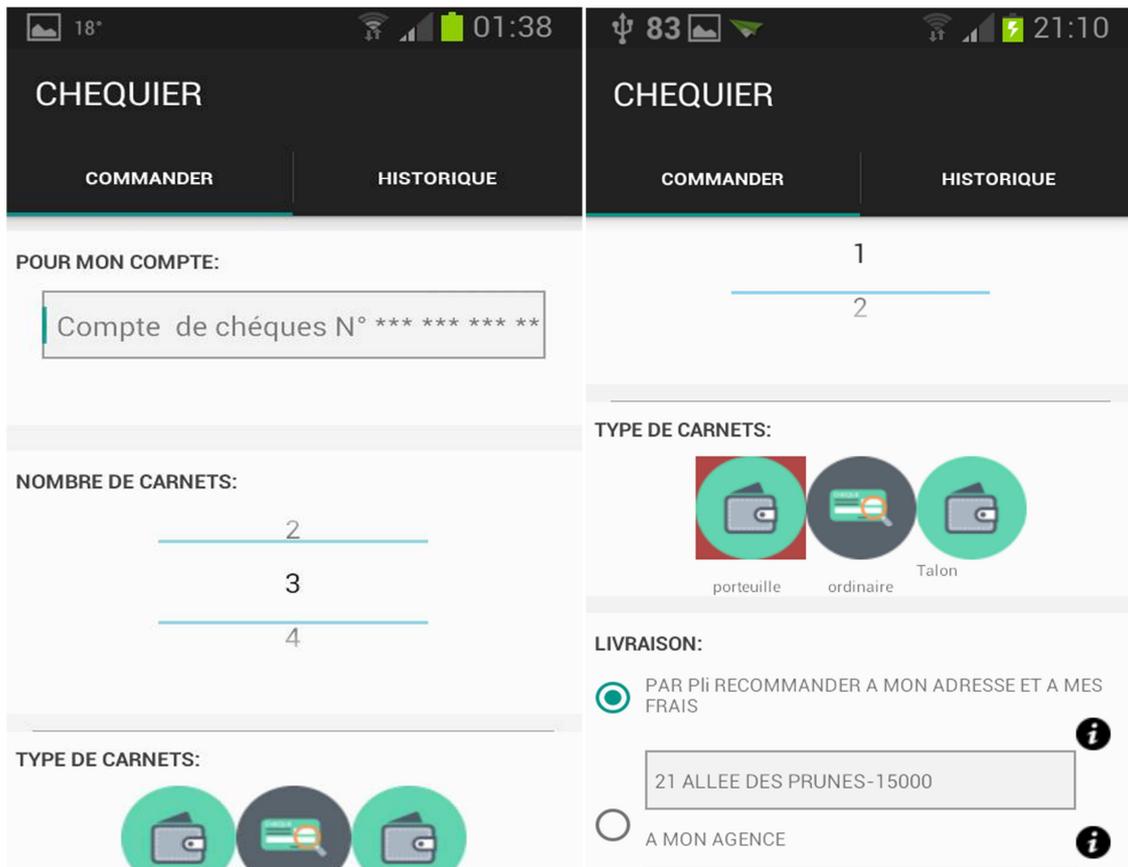


Figure III.7 : Commander un chéquier.

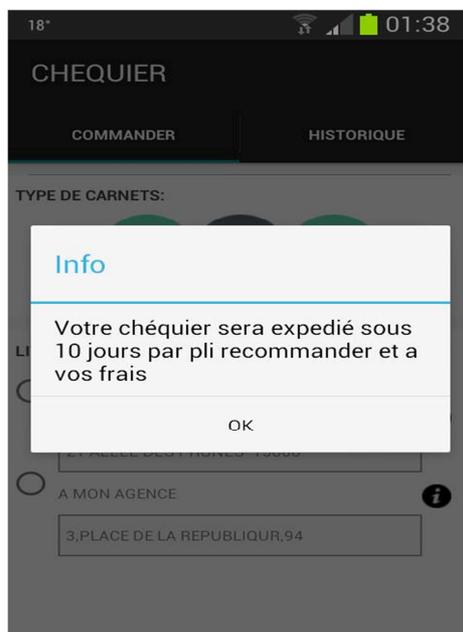
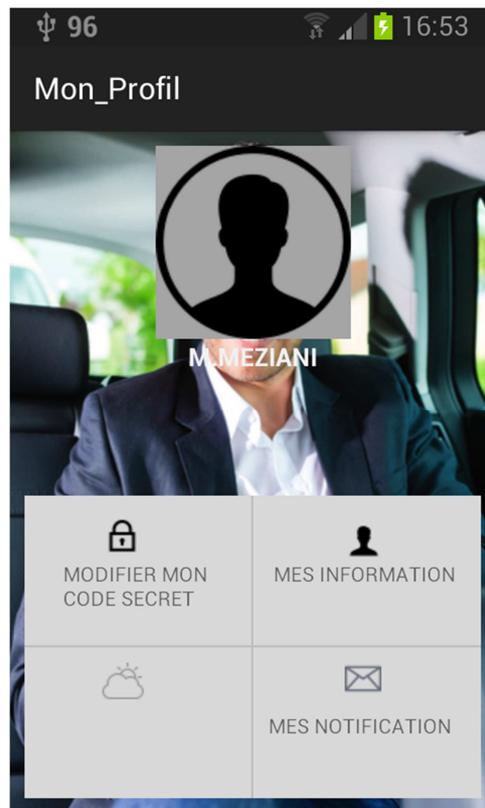


Figure III.8 : Mouvement sur commander un chéquier.

L'abonné a éventuellement la possibilité de modifier son code secret, voir ses informations et ses notifications.



**Figure III.9** : Modification du code secret.

L'utilisateur a la possibilité de contacter sa banque, soit par téléphone soit par Mail.

La figure ci-dessous illustre l'interface qui permet au client de la banque de contacter sa banque par Téléphone ou par Mail.

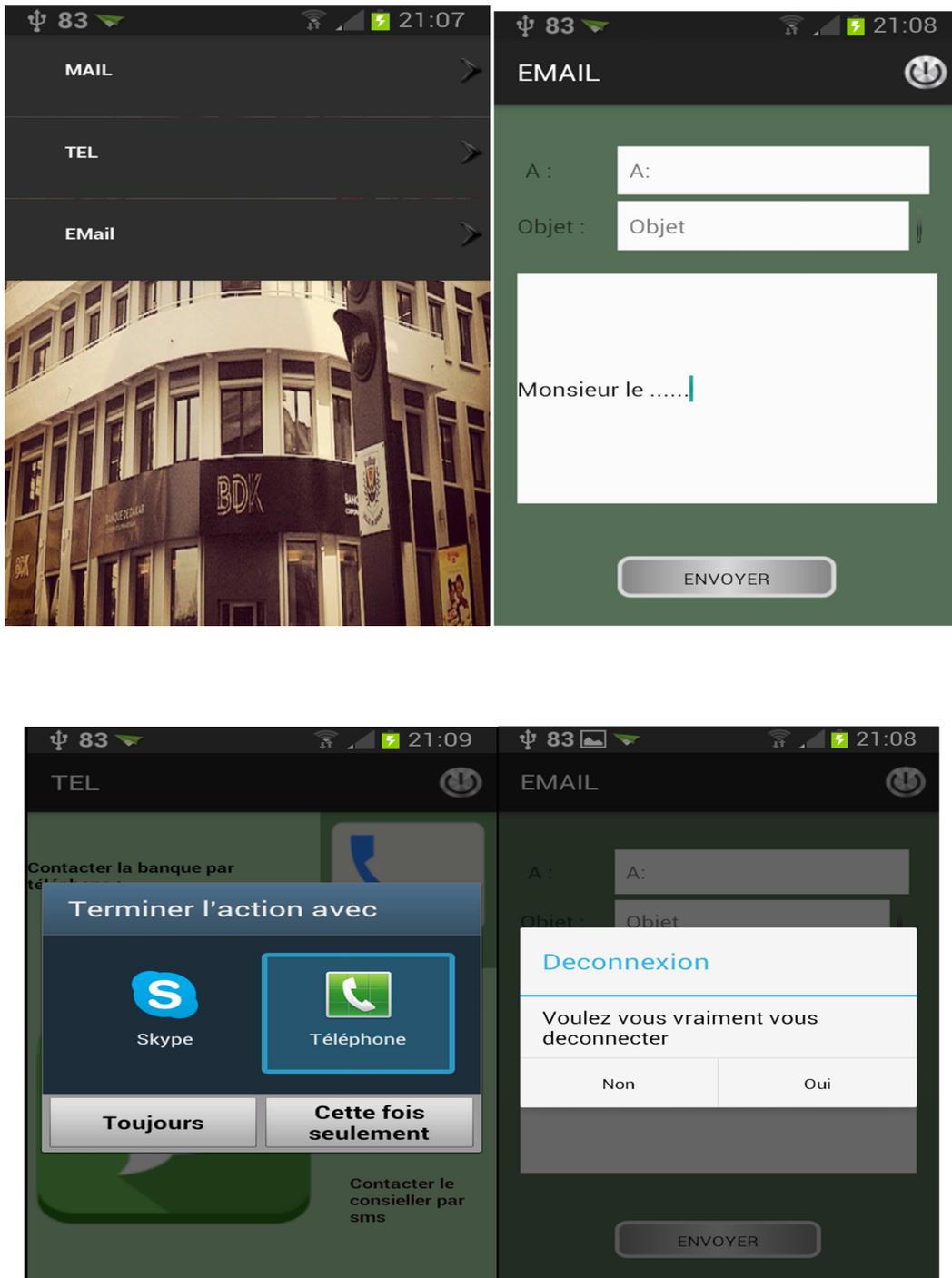


Figure III.10 : Contacter la banque par téléphone ou par mail.

Cette interface permet à l'utilisateur de remplir les champs suivants afin de lui permettre d'obtenir son code secret et celui-ci sera envoyé par voie postale du client ou par Mail.

L'utilisateur saisit les champs suivants : N° Client, Nom, Prénom, Adresse, Téléphone etc.

The image displays two side-by-side screenshots of a mobile application interface. The left screenshot, titled "Code\_secret\_oublier", shows a form with a "N° CLIENT" field and a "CIVILITE" dropdown menu. The dropdown menu is open, showing options: "MADAME", "MONSIEUR", "MADAMOISELLE", and "MADAME". A "CONTINUER" button is at the bottom. The right screenshot, titled "Formulaire", shows a form with fields for "DATE DE NAISSANCE", "VOTRE ADRESS MAIL", "PAYS DE RESIDENCE" (with "ALGÉRIE" selected), "TELEPHONE PRIVILIGIE" (with "FIX PRIVÉ" selected), and "TELEPHONE". A "VALIDER" button is at the bottom. Both screens show a status bar at the top with signal strength, battery, and time (21:07).

**Figure III.11** : Demander son code secret.

Après avoir rempli et validé le formulaire, le client reçoit une notification indiquant que son formulaire est enregistré et envoyé.

### **III.6 Conclusion**

Dans cette dernière partie du projet, nous avons présenté les différents outils de développement, son architecture, les outils de conception de l'application, langage de programmation, ainsi que quelques interfaces.

# Conclusion générale

Les Smartphones sont devenus les accompagnateurs indispensables de l'homme. Ceci est dû principalement à la mise au point de multitudes d'applications mobiles qui couvrent beaucoup de secteurs et parviennent même à satisfaire quelques-uns de nos besoins quotidiens. C'est en considération de ce cadre que nous avons opté pour la réalisation d'une application Mobile Banking sous Android.

Les études conceptuelles et techniques, ainsi que les différentes étapes qui ont abouti à la réalisation de la solution, ont fait l'objet des chapitres de ce rapport. Celui-ci représente pour moi une opportunité irremplaçable, qui m'a permis d'acquérir des connaissances dans le développement mobile, de faire face aux contraintes réelles, et de vivre les différentes phases clés du développement et la production d'une Application.

Pour concevoir ce travail, nous avons adopté les méthodes UML le langage de modélisation en utilisant le logiciel Dia.

L'application mobile est développée en langage JAVA sous Android studio. D'autre part, et aussi le développement des web services PHP permettant de gérer la connexion avec le serveur de base de données. Afin de créer cette base de données, un système de gestion de base de données Mysql a été exploité.

Ces choix nous ont permis de réaliser une application complète. À l'issue de ce projet, nous pouvons envisager les perspectives suivantes :

- Gestion des budgets pour les clients de la banque.
- Augmentation du niveau de sécurité de l'application.
- Amélioration du design de l'application mobile.

# ***BIBLIOGRAPHIE***

# Bibliographie

[1] : <http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>

[2]: [fr.wikipedia.org/wiki/Android](http://fr.wikipedia.org/wiki/Android)

[3]: <https://developer.android.com/about/dashboards/index.html?hl=ko>

[4]:[https://wapiti.telecomlille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2008/Garot-Ombaka/A2\\_archi.html](https://wapiti.telecomlille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2008/Garot-Ombaka/A2_archi.html)

[5]: <http://www.idc.com/getdoc.jsp?containerId=prUS25450615>

[6]:<http://www.univorleans.fr/lifo/Members/JeanFrancois.Lalande/enseignement/android/cours-android.pdf>

[7]: [fr.wikipedia.org/wiki/Android](http://fr.wikipedia.org/wiki/Android)

[8]: <http://www.techno-science.net/?onglet=glossaire&definition=517>

[9]: [https://fr.wikipedia.org/wiki/Banque\\_%C3%A9lectronique](https://fr.wikipedia.org/wiki/Banque_%C3%A9lectronique)

[10]: [ebanking.cpa-bank.dz/part/fr/ide1hom.html](http://ebanking.cpa-bank.dz/part/fr/ide1hom.html)

[11]: <http://fr.slideshare.net/Dharmikpatel7992/mobile-banking-24275438>

<http://www.redcad.org/members/tarak.chaari/cours/coursUML.pdf>

[12] : Pascla Roques, Franck Vallée «UML2 De l'action à l'analyse des besoins à la conception»

[13] : Tarak Chaari «Atelier UML»

[14] : [http://tecfaetu.unige.ch/staf/staf-h/tassini/staf2x/Heidi/last\\_bd.htm](http://tecfaetu.unige.ch/staf/staf-h/tassini/staf2x/Heidi/last_bd.htm)

[15] : <http://socialcompare.com/fr/comparison/android-versions-comparison>

[16]: [http://www.frandroid.com/android/309159\\_repartition-versions-dandroid-lollipop-dessus-20-kitkat-se-stabilise](http://www.frandroid.com/android/309159_repartition-versions-dandroid-lollipop-dessus-20-kitkat-se-stabilise)

[17]: ABI Research, <https://www.abiresearch.com/press/2q-2014-smartphone-resultsforked-android-aosp-gro/>

[18] : UNIVERSITE VIRTUELLE DE TUNIS (uvt), rapport de PFE «Conception et réalisation d'une application mobile M-Banking»,  
<http://pfmh.uvt.rnu.tn/803/1/Conceptionrealisation-application-mobile-M-BANKING.pdf>

[19]: Oracle's Developer Guide,  
[http://otndnld.oracle.com/document/products/as10g/101300/B25221\\_03/web.1013/b13593/undtldev010.html](http://otndnld.oracle.com/document/products/as10g/101300/B25221_03/web.1013/b13593/undtldev010.html)

# Bibliographie

- [20] : Gradeux Vincent, de Gradeux Vincent,  
[http://gardeuxvincent.eu/Documents/ProjetJEE/BACSWW\\_Hibernate\\_Jaxb\\_Spring/content/spring.html](http://gardeuxvincent.eu/Documents/ProjetJEE/BACSWW_Hibernate_Jaxb_Spring/content/spring.html)
- [21] : <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/presentation-android.html#slide2>
- [22] : <http://fr.slideshare.net/cagataycivici/primetime-jsf-withprimefaces-dec-2014>
- [23] : Site Journal du net,  
<http://www.journaldunet.com/encyclopedie/definition/972/34/20/tomcat.shtml>
- [24]: <http://www.Wikipedia.org/wiki/Dia>
- [25] : <http://www.Openclassrooms.com/courses/creez-des-applications-pour-android/installation-et-configuration-des-outils>
- [26] : Gradeux Vincent, de Gradeux Vincent,  
[http://gardeuxvincent.eu/Documents/ProjetJEE/BACSWW\\_Hibernate\\_Jaxb\\_Spring/content/hibernate/hibernate\\_3.html](http://gardeuxvincent.eu/Documents/ProjetJEE/BACSWW_Hibernate_Jaxb_Spring/content/hibernate/hibernate_3.html)
- [27]: [http://fr.wikipedia.org/wiki/Java\\_Development\\_Kit\\_115](http://fr.wikipedia.org/wiki/Java_Development_Kit_115)
- [28] : <https://zestedesavoir.com/tutoriels/335/creez-des-applications-pour-android/420/les-bases-indispensables-a-toute-application/2135/installation-et-configuration-des-outils/#3-eclipse-ladt-et-le-sdk>
- [29]: <http://fr.wikipedia.org/wiki/MySQL>
- [30] : Site officiel WampServer, <http://www.wampserver.com/>
- [31] : <http://www.futurasciences.com/magazines/hightech/infos/dico/d/internet-java-485/>
- [32] : Site officiel du groupe «AddiTeam», <http://www.additeam.com/SSII/php/>
- [33] : Pierre Gambarotto «Technologies pour Web Services faciles : REST, JSON»,  
[https://2009.jres.org/planning\\_files/article/pdf/92.pdf](https://2009.jres.org/planning_files/article/pdf/92.pdf)
- [34] : Site: Figer.com, REST, un style d'architecture universel, , Par: Jean-Paul Figer,  
[http://www.figer.com/publications/REST.htm#.VWCsPU9\\_Okq](http://www.figer.com/publications/REST.htm#.VWCsPU9_Okq)
- [35] : Site officiel du groupe «AddiTeam», <http://www.additeam.com/SSII/uml/>
- [36] : Site MemoireOnLine, Informatique et télécommunication, Par Alain CHIKURU MUGISHO, [http://www.memoireonline.com/10/12/6148/m\\_Conception-duneapplication-web-de-suivi-des-passagers-sur-tous-les-vols-nationaux-et-international14.html](http://www.memoireonline.com/10/12/6148/m_Conception-duneapplication-web-de-suivi-des-passagers-sur-tous-les-vols-nationaux-et-international14.html)

# Bibliographie

[37] : <http://developer.android.com>

[38] : [www.notre-planete.info/PHP/cours\\_1.php](http://www.notre-planete.info/PHP/cours_1.php)

[39] : <http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>

[40] : <http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>

[41] : <http://fr.wikipedia.org/wiki/Android>