

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'enseignement supérieur et de la recherche scientifique**  
**Université Mouloud Mammeri de Tizi-Ouzou Faculté**  
**du génie électrique et d'informatique.**



**Le département d'informatique.**

*En vu de l'obtention du diplôme*

*De MASTER 2 (LMD) en Conduite De Projets Informatiques.*

## **Thème**

**Conception et réalisation d'une  
Application mobile pour les  
clients.**

**Cas : Direction générale OOREDOO  
Ouled Fayet Alger**

**Dirigé par :**

**Mr L.DJEMAA**

**Réalisé par :**

**Mlle Sara DJEBROUNI.**

**Mlle Ferroudja KORSO.**

**Promotion: 2017/2018**



# Remerciements

Nous adressons en premier lieu notre reconnaissance à notre **DIEU** tout puissant, de nous donner la santé et la volonté d'entamer et de terminer ce mémoire.

Nous adressons le grand remerciement à notre promoteur **Mr DJEMAA** qui a accepté de nous encadrer, pour ses conseils, ses dirigés de début à la fin de ce travail.

Ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et l'encadrement de **Mme BOUTHAINA KHEDIRI, Mr HAMZA KHELIF** qui a proposé le thème de ce mémoire, nous les remercions pour la qualité de leur encadrement exceptionnel, pour leur patience, leur rigueur, leur accueil et leur disponibilité durant notre préparation de ce mémoire.

Nous sommes conscientes de l'honneur que nous a fait le membre de jury d'avoir accepté d'examiner notre travail.

Nous remercions aussi le personnel de OOREDOO pour leurs accueil et disponibilité durant notre stage.

# Dédicaces

Je dédie ce modeste travail :

A ma maman, Aucun hommage ne pourrait être à la hauteur de l'amour dont elle ne cesse de me combler, Que dieu lui procure bonne santé et longue vie.

A mes sœurs.

A mes très chers amis, qui sont toujours à mes côtés.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible.

Je vous dis merci.

**Sara**

## ***Introduction générale***

### ***Chapitre I: Présentation de l'organisme d'accueil***

#### ***Partie 1:***

I-1-Présentation de l'organisme d'accueil Ooredoo .....	01
I-2-Historique .....	01
I-4-Objectifs .....	02
I-5-Structure de l'organisation .....	02
I-6-Ooredoo face à la concurrence .....	03

#### ***Partie 2:***

I-7-La problématique .....	04
I-8-Contexte du projet .....	05

### ***Chapitre II: Android & les systèmes d'exploitation mobiles***

Introduction .....	06
II-1- Les systèmes d'exploitation mobile & le développement des applications mobiles	
II-1-1-Les systèmes d'exploitation mobiles .....	06
II-1-1-1-Définition .....	06
II-1-1-2-Les différents systèmes d'exploitation mobile .....	06
II-1-1-3-Comparaison entre les différents systèmes d'exploitation mobile .....	11
II-1-2-les applications mobiles .....	13
II-1-2-1-Définition .....	13
II-1-2-2-Histoire des applications mobiles .....	14
II-1-2-3-Type d'application mobile .....	14
II-2-La plateforme Android .....	15
II-2-1-Définition .....	15

II-2-2-Open Handset Alliance .....	15
II-2-3-Historique .....	15
II-2-4-Pourquoi Android .....	16
II-2-5-Le bugdroid .....	17
II-2-6-Versions d'Android .....	17
II-2-7-Chronologie des versions .....	17
II-2-8-Architecture d'Android .....	19
II-2-9-Environnement de développement Android .....	22
II-2-10-Les composant principaux d'une application Android .....	23
II-2-11-Développement d'une application Android : les étapes .....	28

## ***Chapitre III: Analyse et Conception***

<b>PARTIE I : UML</b> .....	30
Introduction .....	30
III-1-Définition d'UML .....	30
III-2-Historique UML .....	30
III-3-Modélisation avec UML .....	30
III-4-Les objectifs d'UML .....	31
III-5-Diagrammes d'UML .....	31
III-6-Pourquoi UML .....	32
III-7-Les points forts d'UML .....	33
<b>PARTIE II</b>	
III-8-La démarche de modélisation avec l'UML .....	33
III-9- Identification des besoins .....	34
III-10-Identification des acteurs .....	34
III-11- Spécification des tâches .....	34

III-12-Les cas d'utilisation .....	37
III-13-Les diagrammes .....	43
III-13-1-Diagramme de cas d'utilisation .....	43
III-13-2-Diagramme de séquence .....	46
III-13-3-Diagramme de classes .....	49
III-14-La base de données .....	53

## ***Chapitre IV: Réalisation***

IV-1-Environnement matériel et logiciel .....	55
IV-1.1. Environnement matériel .....	55
IV-1.2. Environnement logiciel .....	57
IV-2. Présentation de quelque interface de notre application .....	62
IV-2.1.Interface authentification .....	62
IV-2.2.Interface inscription .....	63
IV-2.3.Interface principale .....	65
IV-2.4.Interface solde .....	66
IV-2.5.Interface changer mot de passe .....	67

Conclusion

Biographie

## ***Liste des figures***

<b>Figure I.1 :</b> Logo depuis 2014 .....	01
<b>Figure I.2 :</b> Organigramme d'Ooredoo .....	02
<b>Figure I.3 :</b> la part du marché de chaque opérateur 2012-2015 .....	04
<b>Figure II.1 :</b> IPHONE X sous IOS12 .....	07
<b>Figure II.2 :</b> TELEPHONE BLACKBERRY .....	08
<b>Figure II.3 :</b> Téléphone Nokia sous symbian OS .....	09
<b>Figure II.4 :</b> Téléphone sous windows phone .....	10
<b>Figure II.5 :</b> Téléphone SAMSUNG S8+ sous ANDROID 7 .....	11
<b>Figure II.6 :</b> le Bugdroid .....	17
<b>Figure II.7 :</b> Architecture d'Android .....	20
<b>Figure II.8 :</b> Environnement de développement Android .....	23
<b>Figure II.9 :</b> Cycle de vie d'une activité .....	25
<b>Figure III.1 :</b> diagramme de UML 2.5 .....	32
<b>Figure III.2 :</b> La démarche adoptée pour la modélisation.....	34
<b>Figure III.3 :</b> diagramme de cas d'utilisation.....	45
<b>Figure III.4:</b> diagramme de séquence de cas d'utilisation « lancer l'application ».....	47
<b>Figure III.5 :</b> diagramme de séquence de cas d'utilisation « connexion » .....	48
<b>Figure III.6 :</b> diagramme de séquence de cas d'utilisation « Inscription » .....	49
<b>Figure III.7 :</b> diagramme de classe.....	52
<b>Figure V.1 :</b> Architecture matériel du système .....	55
<b>Figure. IV.2 :</b> Les différentes technologies utilisées dans l'application.....	56
<b>Figure. IV.3 :</b> Architecture 3-tiers du point de vue technologique.....	56
<b>Figure. IV.4 :</b> Logo framwork yii .....	57
<b>Figure. IV.5 :</b> Interface de l'environnement PhpStorm .....	58



<b>Figure. IV.6 :</b> Exemple d'un fichier XML .....	59
<b>Figure. IV.7 :</b> Interface de l'environnement Android Studio 1.1.0 .....	59
<b>Figure. IV.8 :</b> Interface de l'Android SDK Manager .....	60
<b>Figure IV.8 :</b> Interface authentication.....	63
<b>Figure IV.9:</b> Interface inscription.....	64
<b>Figure IV.10 :</b> Interface accueil.....	65
<b>Figure IV.11 :</b> Interface mon solde .....	66
<b>Figure IV.12 :</b> Interface changer mot de passe.....	67



## *Introduction générale*

Ces dernières années, l'émergence de téléphones intelligents a changé la définition des téléphones portables. Le téléphone n'est plus seulement un outil de communication, mais aussi une partie essentielle de la communication et de la vie quotidienne des gens.

Diverses applications ont ajouté un plaisir illimité pour la vie des gens. Il est certain que l'avenir du réseau sera le terminal mobile. Maintenant, le système Android sur le marché de l'électronique devient de plus en plus populaire, en particulier sur le marché des smartphones.

En raison de l'open source, certains outils de développement sont gratuits, donc il existe de nombreuses applications générées. Cela a fortement inspiré les gens à utiliser le système Android.

En outre, il fournit une plate-forme matérielle très pratique pour les développeurs afin qu'ils puissent dépenser moins d'effort pour réaliser leurs idées. Cela permet à Android de développer davantage.

La multinationale Ooredoo est l'une des leaders dans le secteur de télécommunication en Algérie, disposant d'un marché très important en termes de nombre de clients, donc elle vise toujours à les satisfaire en connaissant leurs attentes et leurs besoins.

De ce fait, les dirigeants de Ooredoo nous ont donné la mission de mise en place d'une application Android pour le SelfCare destinée aux clients de cet opérateur téléphonique.

Le présent rapport synthétise tout le travail que nous avons effectué dans cette perspective, il est organisé en quatre chapitres comme suit :

- **Chapitre I** : il portera sur la présentation de l'organisme d'accueil qu'est donc la multinationale Ooredoo.
- **Chapitre II** : dans ce chapitre, nous allons présenter les systèmes d'exploitation conçus pour fonctionner sur un appareil mobile.
- **Chapitre III** : nous avons présenté la démarche qu'on a adopté pour la modélisation de notre application en utilisant le langage de modélisation UML.
- **Chapitre IV** : nous avons présenté l'environnement de développement de notre application ainsi que quelques captures des interfaces de cette dernière.

Enfin nous donnons une conclusion récapitulant le travail réalisé sans oublier de citer les références bibliographiques sur lesquelles nous nous sommes appuyées pour mener à bien ce travail.

# **Chapitre I**

## ***Présentation de l'organisme d'accueil***

**PARTIE 1 :****I-1-Présentation de l'organisme d'accueil Ooredoo :**

**Ooredoo** précédemment connu sous le nom **Nedjma** (étoile en langue arabe) est une compagnie internationale leader dans le domaine des télécommunications, C'est la marque commerciale mobile de **Wataniya Télécom Algérie**, elle offre des services de téléphonie mobile et d'Internet dans le marché du Moyen-Orient, d'Afrique du Nord et du Sud-Est asiatique. **Ooredoo** a publié dans un communiqué avoir plus de 114 millions de clients en septembre 2015, dans la même année la compagnie a réalisé des revenus de 8,79 milliards de dollars. La maison mère d'**Ooredoo** se situe à Qatar.

**I-2-Historique :**

**Ooredoo Algérie**, la filiale algérienne du groupe Ooredoo a vu le jour le 21 novembre 2013, connu auparavant sous le nom 'Nedjma'. Elle est présente en Algérie depuis le 23 décembre 2003. Puis, elle a été commercialement lancée le 24 août 2004, en offrant aux Algériens une gamme d'offres et de services novateurs, conformément aux standards internationaux. Elle compte aujourd'hui plus de 13 millions d'abonnés et près de 4000 employés.



**Figure I.3 :** Logo depuis 2014

### I-4-Objectifs :

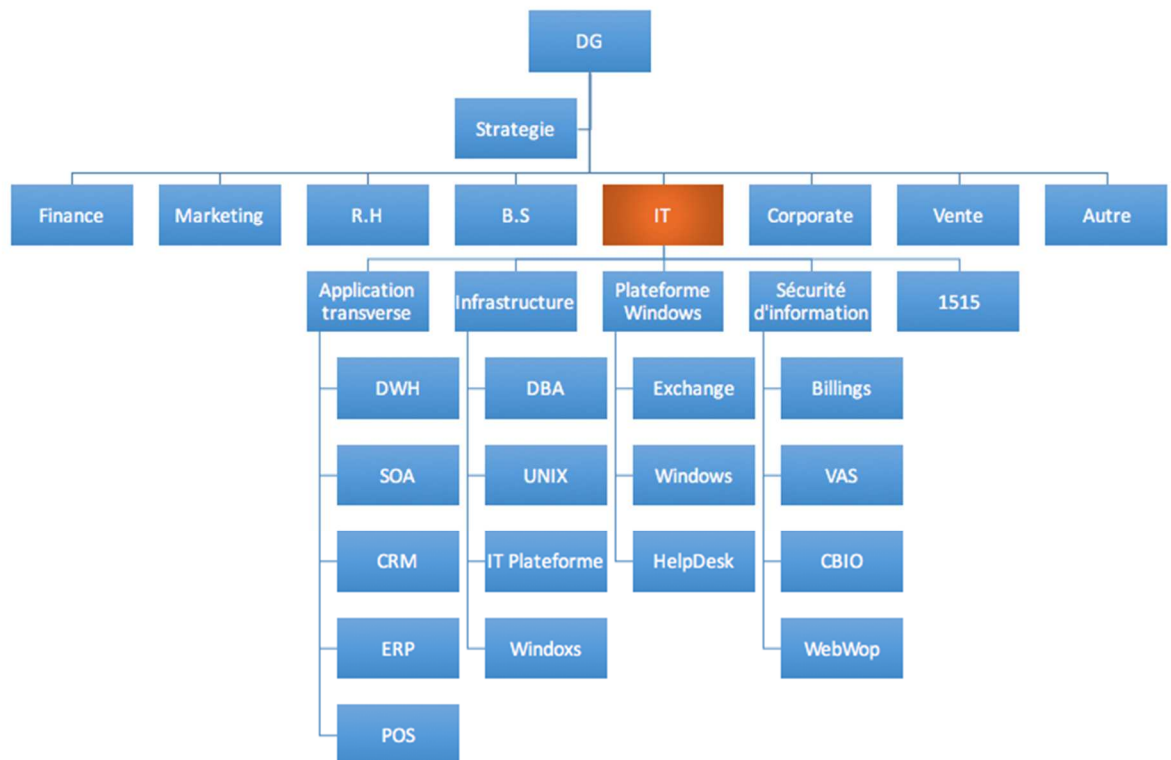
Ooredoo offre aux clients une qualité de services, grâce à ses objectifs ambitieux, son effectif humain compétent et aux matériels mis en œuvre pour les atteindre.

Parmi les objectifs d'Ooredoo :

- Accroître la satisfaction des clients.
- Optimiser l'utilisation des ressources techniques, technologique.
- Développer en permanence les compétences distinctives.
- Anticiper les attentes du marché.
- Construire des relations solides avec leurs partenaires.
- Accroître la satisfaction des actionnaires en augmentant leur rentabilité.

### I-5-Structure de l'organisation :

L'organisation d'Ooredoo est structurée en pyramide comme le montre l'organigramme suivant :



**Figure I.5 :** Organigramme d'Ooredoo

Rôle de chaque composant organisationnel :

**La direction générale :**

Détermine la vision et la stratégie de l'entreprise.

**Le département stratégie :**

Compare la stratégie déterminée avec l'opérationnel.

**Le département de finances :**

Travaille en collaboration avec les équipes de direction pour gérer et rendre compte des résultats de l'entreprise, assurer la budgétisation et identifier et atténuer les risques.

**Le département de marketing :**

Génère des offres/services dans le domaine des télécoms afin de satisfaire les besoins des clients et avoir plus de revenus.

**Le département de ressources humaines (R.H) :**

Gère tout ce qui concerne le recrutement, la paie, la gestion des compétences.

**Le département de business solution (B.S) :**

Étudie la possibilité de la mise en place d'une solution informatique.

**Le département d'I.T :**

Direction technologique des systèmes informatiques, supporte et implémente les actions concrètes déniées par le marketing. Elle est constituée du service application transverse, infrastructure, plateforme Windows, sécurité d'information et le service 1515.

**Le département des ventes :**

Veille au respect des procédures de vente, de procéder à des contrôles périodiques par rapport au budget des ventes, d'accroître le chiffre d'affaire en améliorant les méthodes de vente et d'assurer de la satisfaction des commandes client.

**Le département de service client :**

Gère la relation client, réclamations, service après-vente, etc.

**Autres :** affaire juridique, audit, etc.

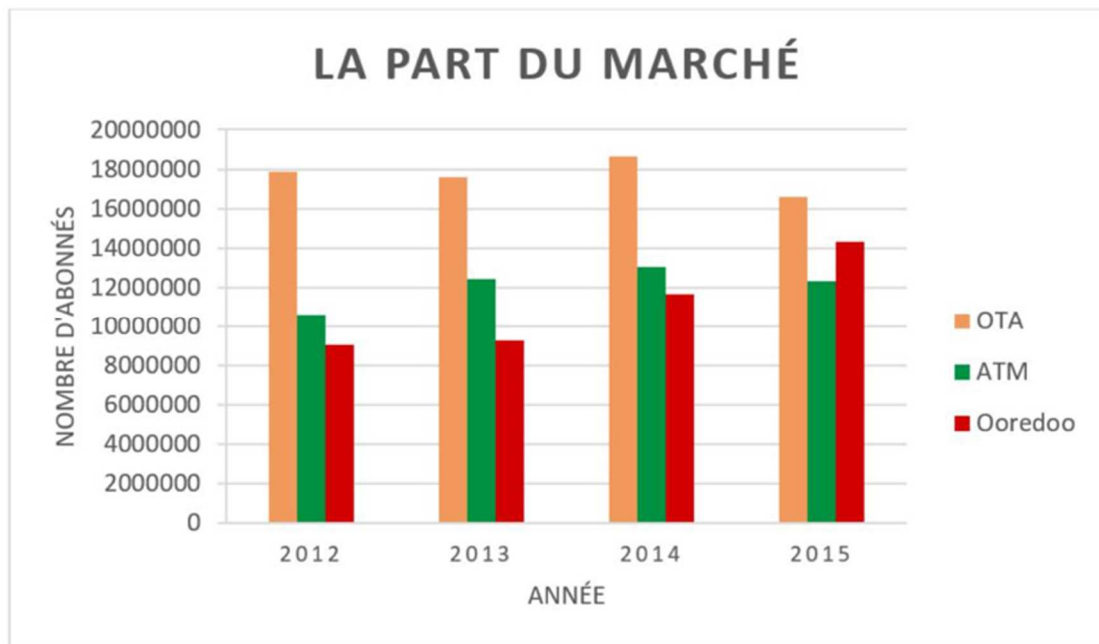
Nous avons effectué notre stage dans le département IT et le département de marketing online au sein du service en vue de la préparation de notre mémoire.

### **I-6-Ooredoo face à la concurrence :**

Dans un secteur très concurrentiel et rapidement évolutif, Ooredoo fait son mieux pour garder et évoluer sa part du marché, notant aussi que le marché de télécommunication en Algérie est stable en terme de nombre d'opérateurs et prestataires, mais rapidement évolutif en terme d'abonnés vu la variété des offres et des technologies présentées ces dernières années comme la 3G et la 4G [2], les opérateurs présents dans le marché de télécommunication sont :

- WTA : Ooredoo Algérie.
- ATM : Algérie Telecom Mobilis.
- OTA : Orascom Telecom Algérie.

La Figure I.6 représente la part du marché de chaque opérateur en Algérie en termes de nombre d'abonnés



**Figure I.6 :** la part du marché de chaque opérateur 2012-2015

Le parc d'abonnés de la 3G a augmenté de 92% en passant de 8,509 millions abonnés en décembre 2014 à 16,319 en décembre 2015. Un parc composé, en sa majorité (88,01%), par des abonnés prépayés.

## **PARTIE 2 :**

### **I-7-La problématique :**

Actuellement les applications mobiles sont en pleine croissance grâce aux succès des smartphones entre autres ceux qui fonctionnent sous Android OS. Cette tendance offre aux entreprises de formidables opportunités pour créer de nouveaux services ou élargir ces supports de communication.

La multinationale **Ooredoo** est l'une des leaders dans le secteur de télécommunication en Algérie, disposant d'un marché très important en terme de nombre de clients, donc elle vise toujours à les satisfaire en connaissant leurs attentes et leurs besoins.



Dans un secteur concurrentiel et à évolution très rapide, la société devra toujours prendre les meilleures décisions et au bon moment pour garder son positionnement au marché et atteindre ses objectifs.

Une application mobile permet de faire évoluer la communication de **Ooredoo** avec ses clients et de mieux les fidéliser. Le nombre d'utilisateurs de smartphones qui ne cesse d'augmenter, permettra à **Ooredoo** de mieux se positionner sur le marché et d'être plus compétitif.

Ainsi, l'objectif de notre projet est de réaliser ce besoin en développant une application mobile sous Android OS qui permet à **Ooredoo** de réaliser un pas de plus vers les nouvelles technologies.

### **I-8-Contexte du projet :**

Notre projet consiste à développer une application mobile qui vise comme clientèle toute personne possédant une carte SIM **Ooredoo** et un Smartphone sous Android OS 5.0 et souhaitant recevoir les différents services offerts par l'opérateur téléphonique **Ooredoo**.

Notre application mobile va permettre aux abonnés, d'avoir des informations sur les différents services et produits **Ooredoo**, de pouvoir activer/désactiver les différents forfaits (voix et data) en un clic et de pouvoir consulter les informations sur leur compte.

L'application est composée de rubriques suivantes :

**-Mon compte :** ici, il s'agit d'un espace pour gérer les paramètres de compte.

Cette interface vous affiche nom de l'abonné, numéro de téléphone, email avec la possibilité de le modifier.

**- Mes lignes:** cette interface vous affichera les lignes associés au client OOREDOO avec possibilité de recharger une ligne, consulter sa consommation internet et ses points de fidélité ainsi que consulter les offres et les nouveautés de cette ligne.

**- Points de vente:** cette interface vous donnera la possibilité de trouver les points de vente les plus proches à votre position.

Cette application permet aussi de consulter les différentes offres spéciales de Ooredoo relatives du mois en cours.

### **Conclusion :**

Dans ce chapitre on a commencé par présenter l'organisme d'accueil, ensuite on a déterminé les besoins de l'entreprise. Dans le prochain chapitre on va traiter du système d'exploitation choisi pour notre projet.

# ***Chapitre II***

## ***Android & les systèmes d'exploitation mobiles***

## **Introduction :**

Les nouvelles technologies sont introduites chaque jour. La plupart d'entre elles sont basées sur des téléphones portables. De simples à intelligents, les téléphones mobiles se sont radicalement transformés en centres d'information et de communication fondamentaux pour la vie moderne.

Donc, les gens évoluent pour utiliser les smartphones de plus en plus. Il existe de nombreux systèmes d'exploitation mobiles différents utilisés par les smartphones.

Ce chapitre donne un aperçu des différents systèmes d'exploitation mobiles qui sont les plus couramment disponibles sur le marché, et se focalise sur la plateforme ANDROID que nous allons aborder en détails.

## **II-1- Les systèmes d'exploitation mobile & le développement des applications mobiles**

### **II-1-1-Les systèmes d'exploitation mobiles**

#### **II-1-1-1-Définition**

Un système d'exploitation mobile (OS mobile) est un OS construit exclusivement pour un appareil mobile, tel qu'un smartphone, un assistant numérique personnel (PDA), une tablette ou un autre système d'exploitation mobile intégré. Les systèmes d'exploitation mobiles populaires sont Android, Symbian, iOS, BlackBerry OS et Windows Mobile.

Un système d'exploitation mobile est responsable de l'identification et la définition des fonctions et fonctions des périphériques mobiles, y compris les claviers, la synchronisation des applications, le courrier électronique, la molette et la messagerie texte.

#### **II-1-1-2-Les différents systèmes d'exploitation mobile :**

Dans cette section, nous aborderons quelques plateformes existantes, de iPhone OS à Android en passant par BlackBerry, Symbian OS, Windows Mobile. Cela nous permettra d'avoir une idée assez générale des systèmes d'exploitation tournant sur mobile, et d'essayer de faire un positionnement d'Android dans l'environnement des systèmes d'exploitation pour mobile.

#### **➤ iOS : iPhone Operating System**

Un logiciel d'exploitation mobile développé par Apple et exclusivement pour le matériel Apple. iPhone OS d'Apple, est le premier OS pour téléphone tactiles qui a véritablement lancé la vague des Smartphone. Ce logiciel d'exploitation gère Apple iPhone, iPad, iPod Touch et Apple TV. Il a d'abord été développé en 2007 et des

mis à jour majeures sont diffusées chaque année. Son concurrent le plus important est le smartphone exploité par Android. Il se caractérise par une interface peu chargée, qui permet l'exécution de tâches rapidement pour utilisateurs réguliers, mais qui laisse peu de place à la personnalisation.



Figure II.1 : IPHONE X sous IOS12

### ➤ BBOS : BlackBerry Operating System

Tout comme l'iPhone, BlackBerry est aussi un téléphone très en vue. Au départ clairement positionné sur le marché des entreprises, la fonction majeure qui a fait décoller le BlackBerry était le push mail. L'utilisateur n'a alors plus besoin de consulter périodiquement sa boîte pour vérifier s'il n'a pas de nouveaux messages. Ceux-ci lui parviennent directement comme un banal SMS.



**Figure II.2 : TELEPHONE BLACKBERRY**

➤ **Symbian Operating System :**

Est un système d'exploitation qui a été conçu par Symbian Ltd. Symbian OS fournit les fonctionnalités essentielles du système d'exploitation, notamment le cœur du système, ainsi que les API communes et une interface utilisateur de référence. Il a été adopté par différents fabricants de téléphones portables de 2G et 3G (Nokia, Sonny, Ericsson, Motorola, Samsung, etc).



**Figure II.3 :** Téléphone Nokia sous symbian OS

➤ **Windows Mobile :**

Le challenger du marché apparu en 2010, a su s'inspirer de ses concurrents et de son expérience passée dans le domaine mobile tout en innovant. Proposant une interface simple et épurée ainsi que de multiples possibilités de personnalisation. Il permet d'arriver rapidement à l'exécution d'une tâche ce qui a particulièrement séduit les utilisateurs.



Figure II.4 : Téléphone sous windows phone

### ➤ Android Operating System :

Est un système d'exploitation open source utilisant le noyau Linux, pour Smartphone, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007.

Android est un système d'exploitation fondé sur un noyau Linux, il comporte une interface spécifique, développée en java, les programmes sont exécutés via un interpréteur JIT.

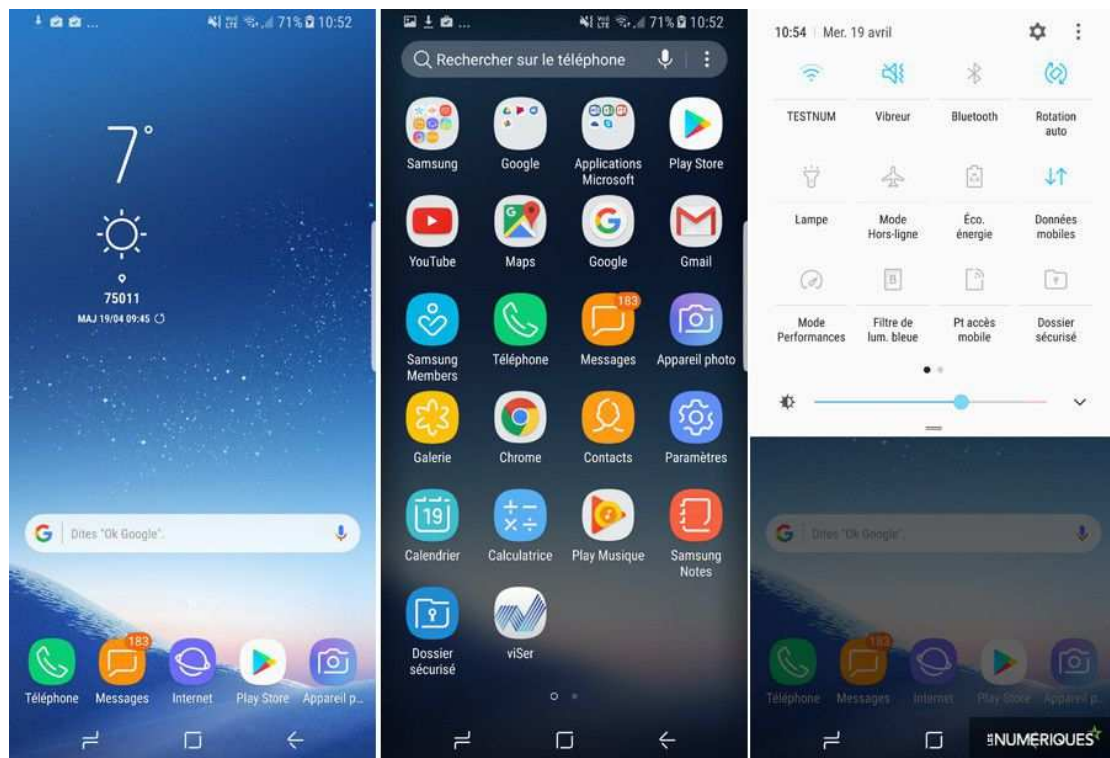


Figure II.5 : Téléphone SAMSUNG S8+ sous ANDROID 7 nougat

### II-1-1-3-Comparaison entre les différents systèmes d'exploitation mobile :

Le tableau suivant offre un comparatif détaillé des fonctionnalités de sécurité des périphérique iPhone, Android et Windows Phone, tels que le paramétrage des applications, la gestion des politiques et la sécurité pendant le transit.



➤ **Fonctionnalités de sécurité :**

<b>Fonction</b>	<b>Apple iOS</b>	<b>Android</b>	<b>Windows Phone</b>
<b>Réinitialisation de l'appareil</b>	Oui	oui	Oui
<b>Effacement des données de l'entreprise</b>	Oui	Non, uniquement avec les solutions tierces telles que Touchdown	Oui (seulement pour la messagerie)
<b>Verrouillage de l'appareil</b>	Oui	Oui	Oui
<b>Chiffrement de l'appareil</b>	iPhone 3GS ou supérieur	Android 3.0 ou supérieur	Oui
<b>Chiffrement des médias</b>	Sans objet	Spécifique au fabricant	Oui
<b>Règle de mot de passe</b>	Oui Règle de complexité propre à l'entreprise	Oui Règle de complexité propre à l'entreprise	Oui Règle de complexité propre à l'entreprise
<b>Restriction du périphérique</b>	Oui	Oui	Oui

**Tableau II.1 :** Comparatif de fonctionnalités de sécurité des OS mobile.

➤ **Contrôle des applications :**

Fonction	Apple iOS	Android	Windows Phone
<b>Installation des applications</b>	Oui Installation de l'application par l'utilisateur	Oui Installation de l'application par l'utilisateur	Oui Uniquement via l'App store d'entreprises
<b>Suppression des applications</b>	Oui Possibilité de suppression silencieuse des applications administrées	Oui Suppression de l'application par l'utilisateur	Oui Le système d'exploitation permet la suppression de toutes les applications publiées par même éditeur
<b>Contrôle du lancement des applications</b>	Non	Oui Via des solutions tierces	Non

**Tableau II.2 :** Comparatif du contrôle des applications des OS mobile.**II-1-2-les applications mobiles****II-1-2-1-Définition :**

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un Smartphone, un baladeur numérique, une tablette tactile, ou encore certains ordinateurs fonctionnant avec le système d'exploitation Windows Phone. Elles sont pour la plupart distribuées depuis des plateformes de téléchargement (parfois elles-mêmes contrôlées par les fabricants de Smartphone) telles que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google /Android), ou encore le Windows Phone Store (plateforme de Microsoft).

Sur certaines plateformes, les applications peuvent aussi être installées à partir des sources tierces, via un site non affilié au distributeur d'origine.

### II-1-2-2-Histoire des applications mobiles :

Les applications mobiles sont apparues dans les années 1990. Elles sont liées aux développements d'Internet, des télécommunications, des réseaux sans fils et des technologies agents et à l'apparition et la démocratisation des terminaux mobiles : Smartphone, tablettes tactiles...

### II-1-2-3-Type d'application mobile :

#### ➤ Application native :

L'application native est une application mobile que l'on télécharge (par exemple sur l'App Store ou le Play Store) et l'on installe ensuite sur son Smartphone ou sur sa tablette, comme un logiciel sur son ordinateur.

#### ➤ Application web mobile :

Une application web mobile désigne un site web adapté pour les mobiles. L'application n'a pas besoin d'être téléchargée et installée puisqu'il s'agit d'un simple site web accessible via un navigateur mobile. Les langages de programmation sont le HTML5, le CSS3 et le JS(JavaScript)

#### ➤ Comparaison entre une application native et une application web mobile :

Application native	Application web mobile
Accessibilité de l'application hors connexion ;	Non accessible en mode hors connexion ;
Coût de développement important ;	Coût de développement moins important ;
Problème de compatibilité après les mises à jour ;	Compatible avec tous les navigateurs ;
Pas la même application sur les différentes plateformes (Apple, Android, Windows Phone, BlackBerry) ;	Un seul et unique code pour les différentes plateformes.
Meilleure expérience utilisateur ;	Manque de fluidité sur les anciens modèles de Smartphone ;

**Tableau II.3** : comparaison entre une application native et une application web mobile

### ➤ **Application hybride :**

Une application hybride est une application utilisant le navigateur web intégré du support (Smartphone ou tablette) et les technologies web (HTML, CSS et le JavaScript) pour fonctionner sur différent OS (iOS, Android, Windows Phone, etc..). Une telle application utilise les fonctionnalités natives des Smartphone et peut être distribuée sur les plateformes d'applications telles que l'AppStore, le Google Play, etc.

## **II-2-La plateforme Android :**

### **II-2-1-Définition :**

Android est un progiciel et un système d'exploitation basé sur linux pour les appareils mobiles tels que les tablettes et les Smartphones.

Il est développé par Google et plus tard par l'OHA (Open Handset Alliance). Le langage Java est principalement utilisé pour écrire le code Android même si d'autres langues peuvent être utilisées.

L'objectif du projet Android est de créer un produit réel réussi qui améliore l'expérience mobile pour les utilisateurs finaux.

### **II-2-2-Open Handset Alliance :**

Le 5 novembre 2007 fut annoncé la création de l'OHA (Open Handset Alliance) : un consortium crée à l'initiative de Google réunissant à ses débuts une trentaine d'entreprises. La plus grande partie des entreprises étaient des operateurs mobiles, des constructeurs, des industriels et des éditeurs de logiciels. Le rôle de l'OHA était de favoriser l'innovation sur les appareils mobiles en fournissant une plateforme véritablement ouverte et complète.

Début 2014, l'OHA regroupe plus de quatre-vingts sociétés membres

### **II-2-3-Historique :**

Au début, Android Incorporation a été fondé par Andy Rubin en octobre 2003 à Palo Alto en Californie aux États-Unis. Ensuite le 17 août 2005, Google a acquis cette entreprise et depuis, elle est dans la filiale de Google Incorporation. Les employés clés d'Android Inc sont Andy Rubin, Rich Miner, Chris White et Nick Sears.

Android Inc a été destiné à la caméra mais plus tard elle a été déplacé vers les téléphones intelligents en raison du faible marché de la caméra. Android est le

surnom de Andy Rubin donné par ses collègues à cause de son amour pour les robots.

En 2007, Google annonce le développement du système d'exploitation Android.

Le 21 octobre 2008, Google et l'OHA annoncèrent la mise à disposition du code source de la plateforme Android en open source, sous la licence Apache 2.0.

En novembre 2008 fut lancé l'Android Market, le magasin d'applications de Google permettant aux développeurs et éditeurs de logiciels Android de proposer leurs applications aux utilisateurs Android. En mars 2012, dans l'optique d'unifier ses services, Google renomma son magasin en « Play Store ». Ce magasin est disponible directement sur les appareils Android via l'application Play Store, mais aussi via le site internet : <http://play.google.com/store>

Le premier smartphone Android est sorti en octobre 2008 aux Etats-Unis et en mars 2009 en France. Ce fut le HTC Dream G1.

Depuis, de nombreux constructeurs ont mis sur le marché un nombre impressionnant d'appareils fonctionnant sous Android. Début 2014, la société Samsung, propose plus de vingt modèles différents de smartphones, et onze modèles de tablettes

Les derniers appareils sont maintenant équipés du NFC (Near Field Communication) qui leur permet de communiquer en champ proche, comme par exemple pour le paiement sans contact.

Beaucoup de chemin a été parcouru depuis l'apparition de la plateforme Android. Chaque semaine apporte son lot de nouveaux smartphones et tablettes.

#### **II-2-4-Pourquoi Android ?**

Aujourd'hui, Android est le système d'exploitation mobile le plus utilisé dans le monde avec plus de 80% de parts de marché dans les Smartphones

Android gère des centaines de millions d'appareils mobiles dans plus de 190 pays à travers le monde. C'est la plus grande base installée de n'importe quelle plateforme mobile et de plus en plus rapide : chaque jour, un autre million d'utilisateurs mettent en marche leurs appareils Android pour la première fois et commencent à chercher des applications, des jeux et d'autres contenus numériques. Android offre une plate-forme de classe mondiale pour créer des applications et des jeux pour les utilisateurs Android partout, ainsi qu'un marché ouvert pour leur distribution instantanée.

Android est une pile logicielle open source pour une large gamme d'appareils mobiles et un projet open source correspondant dirigé par Google.

### II-2-5-Le bugdroid

Le personnage nommé Bugdroid est la mascotte verte utilisé par Google et toutes les communautés Android pour représenter le système d'exploitation (OS).



Figure II.6 : le Bugdroid.

### II-2-6-Versions d'Android :

Le système d'exploitation Android évolue très rapidement : la progression du nombre de fonctionnalités est, à l'image de celle de ses parts du marché, stupéfiante.

Ces fonctionnalités, améliorations et correction de bogues, ont fait leur apparition au fil du temps dans plusieurs versions.

A noter que les versions majeures sont intitulées avec des noms de desserts dans l'ordre alphabétique : **Apple pie, Banana bread, Cupcake, Donut, Eclair, Foryo, Gingerbread, Honeycomb, Ice cream sandwich, Jelly bean, Kitkat, Lollipop, Marshmallow, Nougat, Oreo.**

### II-2-7-Chronologie des versions :

Version	Nom de Version	Date de sortie	Part de marché Android	API Level
Android 8.1	Oreo	5 déc. 2018	1%	27
Android 8.0	Oreo	21 août 2018	1%	26
Android 7.1.2	Nougat	4 avr. 2018	6,2%	25
Android 7.1.1	Nougat	5 déc. 2016	0,4 %	25
Android 7.1	Nougat	4 oct. 2016	0,4 %	25
Android 7.0	Nougat	22 août 2016	2,4 %	24

Version	Nom de Version	Date de sortie	Part de marché Android	API Level
<b>Android 6.0.1</b>	Marshmallow	7 déc. 2015	31,3 %	23
<b>Android 6</b>	Marshmallow	5 oct. 2015	31,3 %	23
<b>Android 5.1.1</b>	Lollipop	21 avr. 2015	23,1 %	22
<b>Android 5.1</b>	Lollipop	9 mars 2015	23,1 %	22
<b>Android 5.0.2</b>	Lollipop	19 déc. 2014	9,4 %	21
<b>Android 5.0.1</b>	Lollipop	2 déc. 2014	9,4 %	21
<b>Android 5.0</b>	Lollipop	17 oct. 2014	9,4 %	21
<b>Android 4.4.4</b>	KitKat	23 juin 2014	20,8 %	19
<b>Android 4.4.3</b>	KitKat	14 avr. 2014	20,8 %	19
<b>Android 4.4.2</b>	KitKat	9 déc. 2013	20,8 %	19
<b>Android 4.4.1</b>	KitKat	5 déc. 2013	20,8 %	19
<b>Android 4.4</b>	KitKat	31 oct. 2013	20,8 %	19
<b>Android 4.3</b>	Jelly Bean	24 juil. 2013	1,5 %	18
<b>Android 4.2.2</b>	Jelly Bean	11 févr. 2013	5,4 %	17
<b>Android 4.2.1</b>	Jelly Bean	27 nov. 2012	5,4 %	17
<b>Android 4.2</b>	Jelly Bean	13 nov. 2012	5,4 %	17
<b>Android 4.1.2</b>	Jelly Bean	9 oct. 2012	3,7 %	16
<b>Android 4.1.1</b>	Jelly Bean	23 juil. 2012	3,7 %	16
<b>Android 4.1</b>	Jelly Bean	9 juil. 2012	3,7 %	16
<b>Android 4.0.4</b>	Ice Cream Sandwich	28 mars 2012	1 %	15
<b>Android 3.2.6</b>	Honeycomb	15 févr. 2012	0 %	13
<b>Android 4.0.3</b>	Ice Cream Sandwich	16 déc. 2011	1 %	15
<b>Android 3.2.4</b>	Honeycomb	15 déc. 2011	0 %	13
<b>Android 4.0.2</b>	Ice Cream Sandwich	28 nov. 2011	0 %	14
<b>Android 4.0.1</b>	Ice Cream Sandwich	19 oct. 2011	0 %	14
<b>Android 4.0</b>	Ice Cream Sandwich	18 oct. 2011	0 %	14
<b>Android 3.2.2</b>	Honeycomb	30 sept. 2011	0 %	13
<b>Android 2.3.7</b>	Gingerbread	21 sept. 2011	1 %	10

Version	Nom de Version	Date de sortie	Part de marché Android	API Level
Android 3.2.1	Honeycomb	20 sept. 2011	0 %	13
Android 2.3.6	Gingerbread	2 sept. 2011	1 %	10
Android 2.3.5	Gingerbread	25 juil. 2011	1 %	10
Android 3.2	Honeycomb	15 juil. 2011	0 %	13
Android 2.3.4	Gingerbread	10 mai 2011	1 %	10
Android 3.1	Honeycomb	10 mai 2011	0 %	12
Android 3.0	Honeycomb	22 févr. 2011	0 %	11
Android 2.3.3	Gingerbread	9 févr. 2011	1 %	10
Android 2.3	Gingerbread	6 déc. 2010	0 %	9
Android 2.2	Froyo	20 mai 2010	0 %	8
Android 2.1	Eclair	12 janv. 2010	0 %	7
Android 2.0.1	Eclair	3 déc. 2009	0 %	6
Android 2.0	Eclair	26 oct. 2009	0 %	5
Android 1.6	Donut	15 sept. 2009	0 %	4
Android 1.5	Cupcake	30 avr. 2009	0 %	3
Android 1.1	Banana bread	9 févr. 2009	0 %	2
Android 1.0	Apple pie	23 sept. 2008	0 %	1
Android 0.9		22 août 2008	0 %	

Tableau II.4 : Tableau des version d'Android

**La version 5.0 :**

(Aussi appelée Lollipop), Son numéro de version et son nom de code ont été annoncés le 15 octobre 2014, pour une disponibilité publique le 3 novembre 2014. Les changements les plus importants d'Android 5.0 sont sa disponibilité sur les nouvelles plateformes Android TV et Android Auto, ainsi que l'amélioration de l'autonomie de la batterie via le projet Volta.

Cette version se caractérise par :

- Nouvelle interface / design ("Material design")
- Amélioration de la rapidité
- Amélioration de la gestion de la batterie

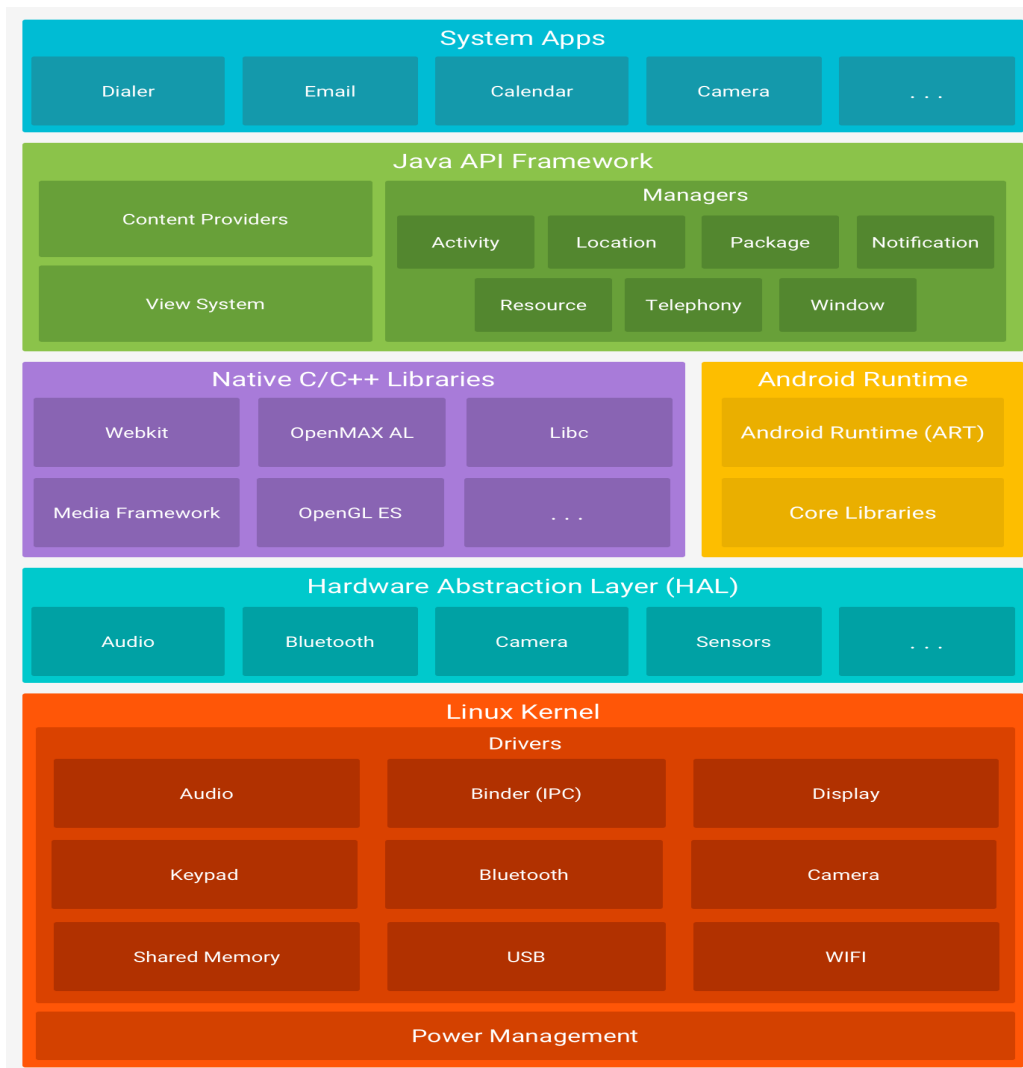
**II-2-8-Architecture d'Android :**

Android est une pile applicative pour les appareils mobiles qui comprend un système d'exploitation, middleware et des applications clés. Le SDK Android fournit



les outils et les API nécessaires pour commencer à développer des applications sur la plateforme Android en utilisant le langage de programmation Java.

Le schéma suivant illustre les principaux composants du système d'exploitation Android.



**Figure II.7 : Architecture d'Android.**

### II-2-8-1-Applications système

Android vient avec un ensemble d'applications principales pour le courrier électronique, la messagerie SMS, les calendriers, la navigation sur Internet, les contacts et plus encore. Les applications incluses avec la plate-forme n'ont aucun statut spécial parmi les applications que l'utilisateur choisit d'installer. Ainsi, une application tierce peut devenir le navigateur Web par défaut de l'utilisateur, SMS

messenger, ou même le clavier par défaut (certaines exceptions s'appliquent, telles que l'application Paramètres du système).

### II-2-8-2-Le Framework (Java API Framework)

L'intégralité du jeu de fonctionnalités du système d'exploitation Android est disponible via les API écrites dans le langage Java. Ces API forment les blocs de construction dont on a besoin pour créer des applications Android en simplifiant la réutilisation des composants et des services essentiels du système, qui incluent les éléments suivants:

- Un système de vue riche et extensible qu'on peut utiliser pour créer l'interface utilisateur d'une application, y compris des listes, des grilles, des zones de texte, des boutons et même un navigateur Web incorporé
- Un gestionnaire de ressources, fournissant l'accès à des ressources autres que le code, telles que les chaînes localisées, les graphiques et les fichiers de mise en page.
- Un Notification Manager qui permet à toutes les applications d'afficher des alertes personnalisées dans la barre d'état.
- Un gestionnaire d'activités qui gère le cycle de vie des applications et fournit une pile de retour de navigation commune
- Les fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications, telles que l'application Contacts, ou de partager leurs propres données.

### II-2-8-3-Les bibliothèques (Libraries)

De nombreux composants et services principaux du système Android, tels que ART et HAL, sont construits à partir du code natif qui nécessitent des bibliothèques natives écrites en C et C ++.

Si on développe une application qui nécessite un code C ou C ++, on peut utiliser le NDK Android pour accéder à certaines de ces bibliothèques de plates-formes natives directement à partir de votre code natif.

La liste ci-dessous énumère quelques bibliothèques disponibles dans Android :

- **Bibliothèque système C.** Implémentation (dérivée de BSD) de la bibliothèque standard C (libc), optimisée pour les systèmes Linux embarqués.
- **Bibliothèques multimédias.** Basées sur StageFright, elles permettent le support de nombreux formats audio et vidéo, tels que MPEG4, H.264, MP3, AAC, AMR, JPG et PNG
- **SurfaceFlinger.** Permet l'accès au sous-système d'affichage.
- **LibWebCore.** Moteur de rendu de pages Internet basé sur Webkit. Cette bibliothèque est donc principalement utilisée dans le navigateur et dans les vues web embarquées (WebView).
- **Skia.** Moteur graphique 2D.

- **Bibliothèques 3D.** Implémentation basée sur OpenGL ES 1.0 API et plus récemment OpenGL ES 2.0.
- **FreeType.** Rendu des polices de caractères.
- **SQLite.** Base de données légère et puissante.

#### II-2-8-4-Moteur d'exécution Android (Android Runtime)

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de machine virtuelle Dalvik.

Dalvik VM est une implémentation de machine virtuelle ayant été conçue pour optimiser l'exécution multiple de machines virtuelles.

ART est écrit pour exécuter plusieurs machines virtuelles sur des périphériques à mémoire faible en exécutant des fichiers DEX, un format bytecode conçu spécialement pour Android optimisé pour une empreinte mémoire minimale. Les chaînes d'outils, telles que Jack, compile des sources Java dans le bytecode DEX, qui peut fonctionner sur la plate-forme Android.

Android comprend également un ensemble de bibliothèques d'exécution de base qui fournissent la plupart des fonctionnalités du langage de programmation Java, y compris certaines fonctionnalités de langage Java 8, que le framework API Java utilise.

#### II-2-8-5-Couche d'abstraction matérielle (HAL)

La couche d'abstraction matérielle (HAL) fournit des interfaces standard qui exposent les fonctionnalités matérielles du périphérique au framework API Java de niveau supérieur. Le HAL se compose de plusieurs modules de bibliothèque, chacun impliquant une interface pour un type spécifique de composant matériel, tel que l'appareil photo ou le module bluetooth. Lorsqu'une API Framework lance un appel pour accéder au matériel du périphérique, le système Android charge le module de bibliothèque pour ce composant matériel.

#### II-2-8-6-Noyau Linux (Linux Kernel)

La base de la plate-forme Android est le noyau Linux. Par exemple, l'Android Runtime (ART) repose sur le noyau Linux pour les fonctionnalités sous-jacentes telles que le threading et la gestion de la mémoire de bas niveau.

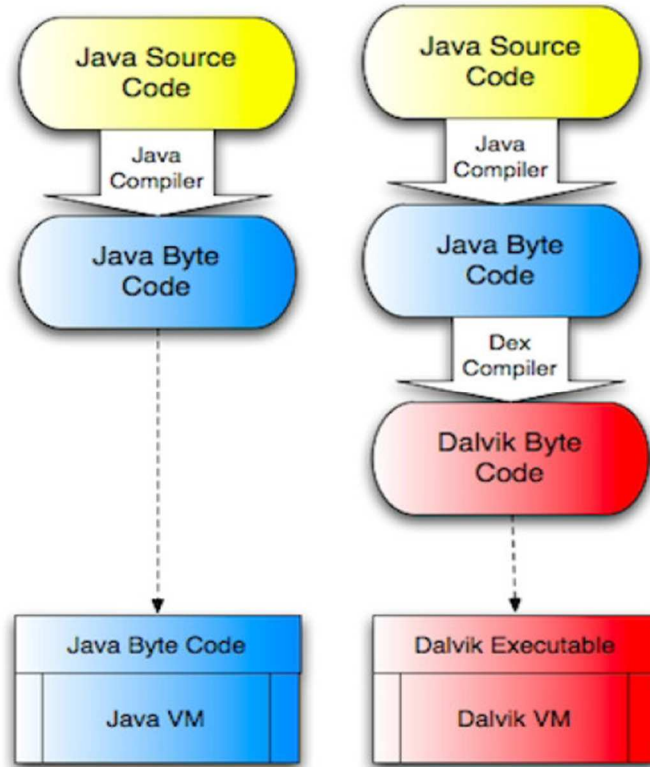
L'utilisation d'un noyau Linux permet à Android de profiter des principales fonctionnalités de sécurité et permet aux fabricants de périphériques de développer des pilotes matériels pour un noyau bien connu.

#### II-2-9-Environnement de développement Android :

L'environnement d'exécution d'Android est la machine virtuelle Dalvik, qui est

incorporé dans le système d'exploitation Android et son rôle est de permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité.

Les programmes sont écrits en JAVA puis compilés avec des outils Java afin d'obtenir un bytecode qui sera lui-même recompilé avec l'outil (**dex**) pour obtenir un code adapté à la machine Dalvik, comme le montre le schéma suivant :



**Figure II.8 :** Environnement de développement Android.

#### **II-2-10-Les composant principaux d'une application Android :**

Les composants de l'application sont les éléments essentiels d'une application Android. Chaque composant est un point d'entrée par lequel le système ou un utilisateur peut entrer dans votre application. Certains composants dépendent des autres.

Il existe quatre types différents de composants d'application:

- Activités.
- Prestations de service.
- Récepteurs de diffusion.
- Fournisseurs de contenu.

Chaque type a un but distinct et un cycle de vie distinct qui définit comment le composant est créé et détruit. Les sections suivantes décrivent les quatre types de composants de l'application.

### II-2-10-1-Activités

Une activité est le point d'entrée pour interagir avec l'utilisateur. Il représente un seul écran avec une interface utilisateur.

Une activité facilite les interactions clés suivantes entre le système et l'application :

- Suivre ce que l'utilisateur prend actuellement en charge (ce qui est sur l'écran) pour s'assurer que le système continue d'exécuter le processus qui héberge l'activité.
- Sachant que les processus précédemment utilisés contiennent des éléments auxquels l'utilisateur peut revenir (activités arrêtées), et donc plus prioritairement de garder ces processus en place.
- Aider l'application à gérer son processus afin de permettre à l'utilisateur de revenir à des activités avec son état antérieur restauré.
- Fournir un moyen pour les applications de mettre en œuvre les flux d'utilisateurs entre eux et pour que le système puisse coordonner ces flux. (L'exemple le plus classique ici étant partagé.)

#### II-2-10-1-a-Les états d'une activité :

##### Active (Resumed) :

- L'activité est visible en totalité.
- Elle est sur le dessus de la pile.
- Elle a le focus, ce qui veut dire que l'utilisateur agit directement sur l'activité et peut utiliser toute son intégralité.

##### Suspendue (Paused) :

L'activité est partiellement visible à l'écran. C'est le cas lors de la réception d'un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message. Ce n'est pas sur l'activité suspendue qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sous-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser (stopper) de l'application qui l'obstrue, puis il pourra à nouveau interagir avec elle.

##### Arrêtée (Stopped) :

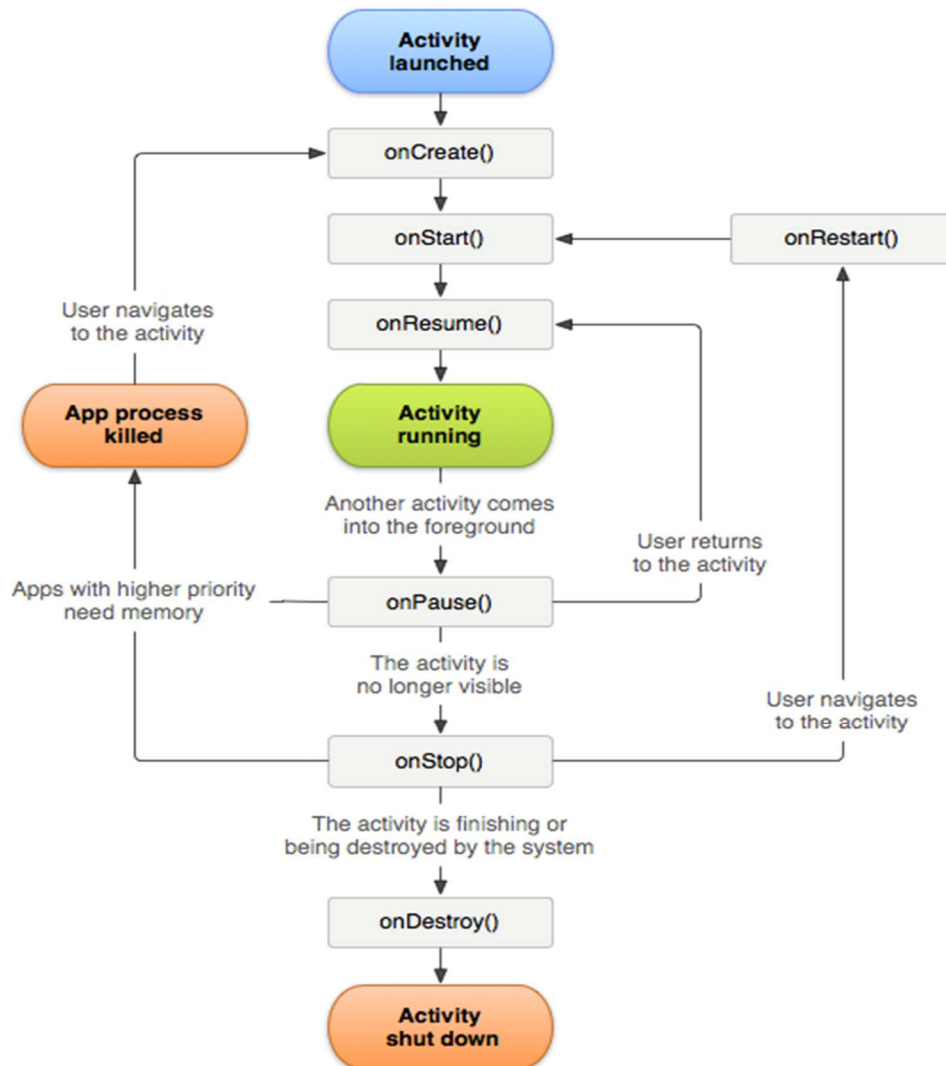
L'activité est tout simplement masquée par une autre activité, on ne peut plus la voir.

L'application n'a évidemment plus le focus, et on ne peut pas agir dessus. Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue l'application pour libérer de la mémoire système.

#### II-2-10-1-b-Cycle de vie d'une activité

Le cycle de vie d'une activité décrit les états dans lesquels l'activité peut se trouver entre sa création, l'instanciation, et la destruction de cette instance. L'image ci-dessous, représente ce cycle.

Chaque changement d'état produit un appel à une méthode spécifique pouvant être surchargée dans la classe de l'activité.



**Figure II.9 :** Cycle de vie d'une activité

Ces méthodes sont listées ci-dessous :

- `onCreate()` :

La première méthode du cycle de vie appelée lors de la création d'une activité. Elle n'est appelée qu'une seule et unique fois durant tout le cycle de vie de l'activité ;

la fonction de cette méthode est de permettre d'initialiser l'activité, créer des vues qui doivent être affichées à la création de l'activité, récupérer des instances de vue...

- `onStart()` :

Elle est le pendant de la méthode `onStop`. Elle est appelée après la méthode `onCreate` ou après la méthode `onRestart` si l'activité était dans l'état stoppé et qu'elle revient du nouveau au premier plan. Elle précède l'affichage de l'activité, plus précisément de sa vue.

- `onResume()` :

La méthode `onResume` est le pendant de la méthode `onPause`. Elle est appelée après la méthode `onStart` ou après la méthode `onPause` si l'activité était en pause et qu'elle revient de nouveau au premier plan. A ce stade, l'activité est affichée à l'utilisateur. Une fois cette méthode quittée, l'activité va fonctionner normalement au premier plan et pouvoir interagir avec l'utilisateur.

- `onPause()` :

Cette méthode est le pendant de la méthode `onResume`. Elle est appelée juste avant qu'une autre activité prenne la main et passe au premier plan. Il faut donc qu'elle soit à la fois rapide pour ne pas bloquer l'activité suivante et efficace, ayant pour rôle de sauvegarder les données persistantes de l'activité et de stopper les tâches consommatrices de CPU, de mémoire...

- `onStop()` :

Cette méthode est le pendant de la méthode `onStart`. Elle permet de libérer certaines ressources.

- `onRestart` :

Cette méthode peut être appelée après la méthode `onStop` si l'activité revient au premier plan. Puis elle est suivie à son tour d'un appel à la méthode `onStart`. La méthode `onRestart` permet de redemander certaines ressources libérées dans la méthode `onStop`.

- `onDestroy()`

Elle est le pendant de la méthode `onCreate`. Elle est appelée après un appel à la méthode `finish` ou directement par le système s'il a besoin de ressources. C'est la dernière méthode mise à la disposition du développeur à être appelée avant la destruction effective et irréversible de l'activité, elle permet de libérer des ressources liées à l'activité.

Seule la méthode `onCreate()` est obligatoire. Cependant, dès que la complexité des activités augmente (ouverture de fichiers, écouteurs sur des capteurs, etc.), il devient nécessaire d'implémenter les autres méthodes pour gérer finement les différents états de l'activité.

Un objet `Bundle savedInstanceState` est passé en paramètre aux méthodes `onCreate`, `onStart` et `onResume`. Cet objet représente l'instance sauvegardée de l'activité. Ainsi, lorsqu'on revient sur cette dernière (après une pause ou un arrêt) et si elle est toujours en mémoire, l'activité est dans le même état que lorsqu'on l'a quittée.

### **II-2-10-2-Les service**

Un service est un point d'entrée à usage général pour garder une application en cours d'exécution pour toutes sortes de raisons. C'est un composant qui s'exécute en arrière-plan pour effectuer des opérations de longue durée ou effectuer des travaux pour des processus distants. Un service ne fournit pas d'interface utilisateur. Par exemple, un service peut jouer de la musique en arrière-plan lorsque l'utilisateur se trouve dans une application différente, ou il peut récupérer des données sur le réseau sans bloquer l'interaction de l'utilisateur avec une activité. Un autre composant, comme une activité, peut démarrer le service et le laisser fonctionner ou s'y lier afin d'interagir avec lui.

### **II-2-10-3-Récepteurs de radiodiffusion**

Un récepteur de diffusion est un composant qui permet au système de fournir des événements à l'application en dehors d'un flux d'utilisateur régulier, ce qui permet à l'application de répondre aux annonces diffusées sur l'ensemble du système. De nombreuses émissions proviennent du système, par exemple, une émission annonçant que l'écran a été éteint, que la batterie est faible ou qu'une image a été capturée.

### **II-2-10-4-Fournisseurs de contenu**

Un fournisseur de contenu gère un ensemble partagé de données d'application qu'on peut stocker dans le système de fichiers, dans une base de données SQLite, sur le Web ou sur tout autre emplacement de stockage persistant auquel votre application peut accéder. Grâce au fournisseur de contenu, d'autres applications peuvent consulter ou modifier les données si le fournisseur de contenu l'autorise. Par exemple, le système Android fournit un fournisseur de contenu qui gère les informations de contact de l'utilisateur.

Les fournisseurs de contenu sont également utiles pour lire et écrire des données privées à l'application et non partagées. Par exemple, l'application Notepad utilise un fournisseur de contenu pour enregistrer des notes.



**Conclusion :**

Ce chapitre nous a permis d'aborder les différents systèmes d'exploitation conçus pour fonctionner sur les appareils mobiles. Ainsi on a étudié le positionnement d'Android par rapport aux autres systèmes en spécifiant ses notions de bases, ses caractéristiques et ses possibilités d'utilisation.

La popularité d'Android et sa généralisation sur la plupart des équipements du public, smart, table, téléphone, nous ont poussé à choisir ce système comme cible pour notre application.

# *Chapitre III*

## *Analyse & Conception*

## PARTIE I : UML

### Introduction :

Lorsqu'on souhaite réaliser un système logiciel performant l'étape de conception est très importante pour la réussite de ce projet informatique, car elle vise à définir une feuille de route du projet, le concevoir et le valider avant de passer à la réalisation du système. Elle permet aussi d'avoir une bonne réflexion avant de passer à l'action, une bonne organisation du travail et une bonne communication entre les différents intervenants dans le projet.

Pour se faire, nous allons suivre dans ce chapitre une démarche qui s'appuie sur le langage UML, conçu pour la visualisation, la spécification et construction des systèmes logiciels. UML commence par une phase d'analyse pour spécifier les différents acteurs du système ainsi que leurs besoins, leurs tâches et leurs scénarios. Ensuite, une phase de conception qui va s'appuyer sur les résultats de la phase d'analyse et sa modélisation s'appuie sur l'élaboration des diagrammes de séquences et de classes.

### III-1-Définition d'UML :

Le **langage de modélisation unifié**, de l'anglais *Unified Modeling Language (UML)*, est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter, spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

### III-2-Historique UML :

Au début des années 90, une cinquantaine de méthodes d'analyse et de conception objet existaient. Parmi elles, trois étaient considérées comme les plus importantes : BOOCH de Grady Booch, OMT de James Rumbaugh et enfin OOSE d'Ivar Jacobson. Ces trois auteurs ont ensuite décidé d'unir leurs efforts au sein de la société Rational Software et en 1996 la version 0.9 d'Unified Modeling Language (UML) est proposée. Deux éléments importants sont à noter :

- 1) le terme unified signifie que les auteurs ont essayé de regrouper les éléments importants des concepts objets
- 2) le terme langage montre qu'il s'agit d'un langage de modélisation et non d'une méthode

Les principaux acteurs du secteur informatique ont ensuite participé à cet effort, et UML 1.0 a été proposé à l'Object Management Group (OMG). Cet organisme international chargé de définir des standards dans le domaine de l'objet, normalise UML 1.1 en 1997. Cette norme a depuis continué d'évoluer et nous en sommes aujourd'hui à la norme 2.5 sortie en le 09/2013. UML est un langage qui permet de modéliser non seulement des applications informatiques ou des structures de données, mais également les activités d'un domaine : mécanique, biologie, processus métier ...

### III-3-Modélisation avec UML :

Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer, il permet :

- De visualiser le système comme il est ou comme il devrait l'être.
- De valider le modèle vis à vis des clients
- De spécifier les structures de données et le comportement du système.
- De fournir un guide pour la construction du système.
- De documenter le système et les décisions prises.

#### III-4-Les objectifs d'UML :

- Représenter des systèmes entiers.
- Etablir un couplage explicite entre les concepts et les artefacts exécutable.
- Prendre en compte les facteurs d'échelle.
- Créer un langage de modélisation utilisable à la fois par les humains et les machines.
- Recherche d'un langage commun :
  - Utilisable par toutes les méthodes.
  - Adapté à toutes les phases du développement.
  - Compatible avec toutes les techniques de réalisation.

#### III-5-Diagrammes d'UML :

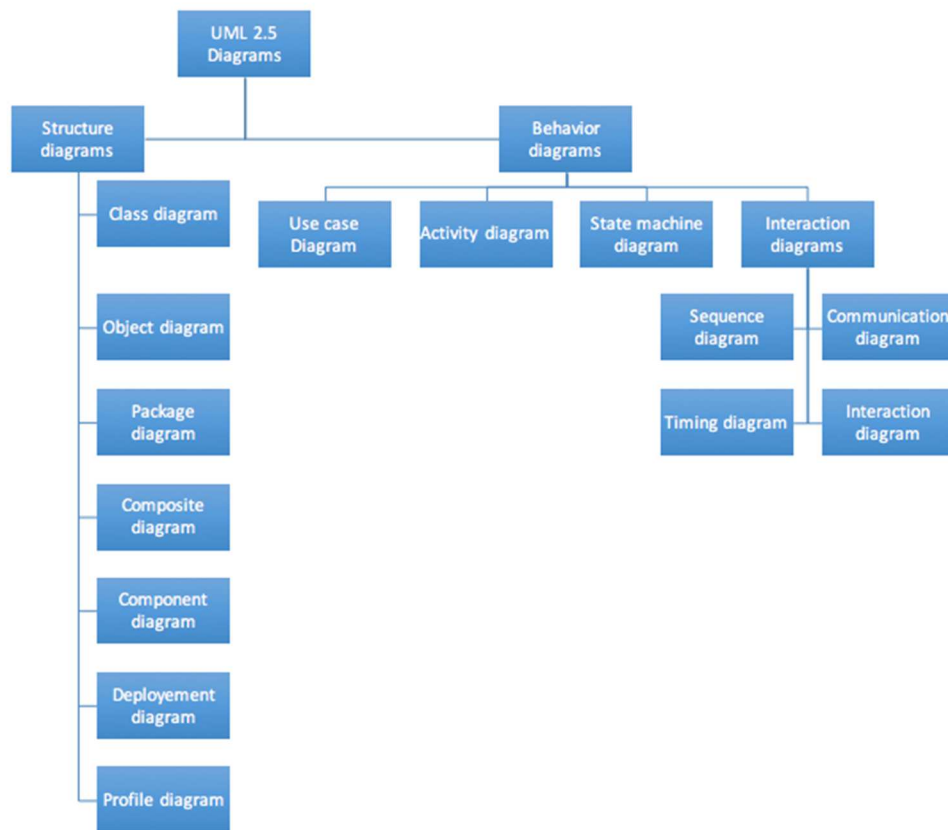
Ils sont au nombre de quatorze (14), de trois (3) types :

**- les diagrammes de structure :** diagramme de classe, diagramme composite, diagramme de composants, diagramme de déploiement, diagramme d'objets, diagramme de package, diagramme de profile

**- les diagrammes de comportement :** diagramme d'activité, diagramme de cas d'utilisation, diagramme d'état-transition

**- les diagrammes dynamiques :** diagramme de séquence, diagramme de communication, diagramme global d'interaction, diagramme de temps

Les diagrammes les plus utilisés sont les suivants : cas d'utilisation, diagramme de séquence, diagramme de classe, diagramme de package, diagramme composite, diagramme d'état-transition.



**Figure III.1 :** diagramme de UML 2.5

UML permet de représenter des modèles, mais il ne définit pas de processus d'élaboration de modèles. Les auteurs d'UML conseillent tout de même une démarche pour favoriser la réussite d'un projet, cette démarche doit être :

- Une démarche itérative et incrémentale : Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maîtriser l'inconnu.
- Une démarche guidée par les besoins des utilisateurs : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin.

Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.

- Une démarche centrée sur l'architecture logicielle : c'est la clé de voute de succès d'un développement, les choix stratégiques définiront la qualité du logiciel

### III-6-Pourquoi UML ?

UML a été créé dans le souci de favoriser à la fois la description, l'analyse et la conception d'un système logiciel afin de le réaliser aussi bien que possible.

Il devient un triple acteur en permettant :

- de décrire les plans de construction du système logiciel, en y intégrant tous les ingrédients conceptuels, toutes les fonctions et toutes les bases de données, toutes les classes et tous les composants logiciels ;
- de valider ces plans avant de commencer la programmation ;
- de réaliser effectivement le système.

Etant donné qu'il est le fruit de la fusion de plusieurs méthodes objets, il utilise l'approche objet en présentant un langage de description universel.

Il permet grâce à un ensemble de diagrammes très explicites, de représenter l'architecture et le fonctionnement des systèmes informatiques complexes en tenant compte des relations entre les concepts utilisés et l'implémentation qui en découle.

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en fait un langage universel.

### III-7-Les points forts d'UML :

UML est un langage formel et normalisé :

- Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.
- UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.

## PARTIE II :

### III-8-La démarche de modélisation avec l'UML :

La figure suivante donne la représentation graphique de la démarche à suivre pour la modélisation de notre application.

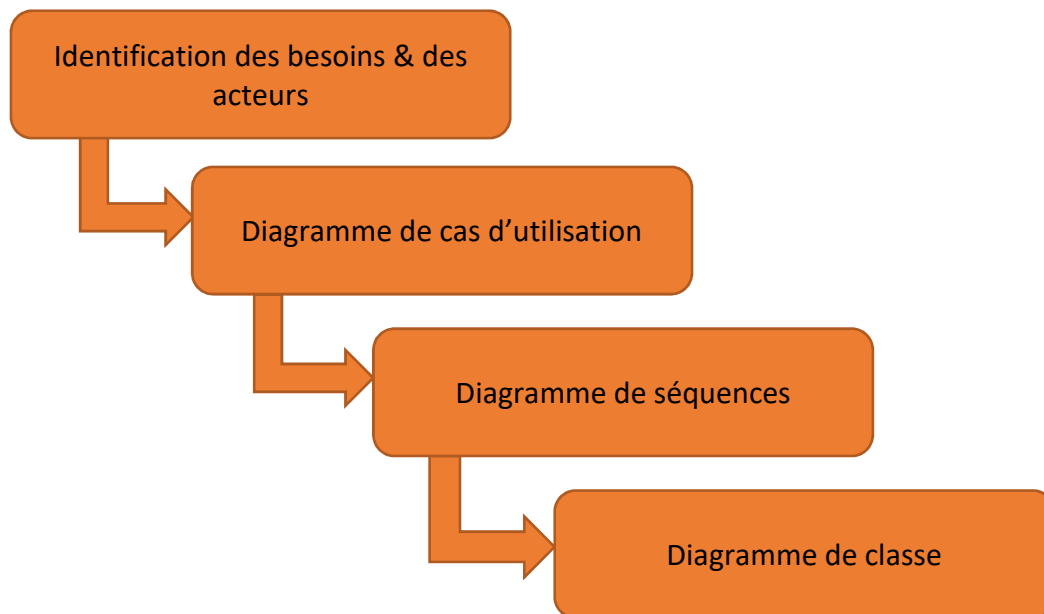


Figure III.2 : La démarche adoptée pour la modélisation

### III-9- Identification des besoins :

L'objectif principal de notre application est de permettre aux clients Ooerdoo de gérer en toute simplicité leur compte depuis un smartphone et tablette, cette application est en quelque sorte un espace client en une application.

Les fonctionnalités de notre application sont :

- Inscription et création d'un compte utilisateur ;
- Connexion au compte utilisateur avec authentification et qui permet :
  - Consultation et suivi de la consommation ds la lignes : forfait, internet, promos,
  - Accès aux nouveautés et promotions ;
  - Recharge d'une ligne en toute sécurité ;
  - Accès au programme de fidélité : suivi des points et conversion ;
  - Recherche des points de vente Ooredoo les plus proches.

### III-10-Identification des acteurs

Notre système communique avec des acteurs afin de réaliser un ensemble d'opérations. Un acteur représente un rôle que peut jouer l'utilisateur dans le système. L'acteur est associé à un cas d'utilisation, c'est-à-dire qu'il peut interagir avec lui et participer à son scénario.

Les acteurs de notre système sont :

**Client :** toute personne qui utilise une machine (Téléphone, Tablette, etc..) qui marche sous le système Android 6.0 ou une version ultérieure. Cet acteur doit lancer l'application, pour accéder à son compte.

### III-11- Spécification des tâches :

**III-11-1-Définition :**

Une tâche est l'ensemble des différentes fonctions qui peuvent être accédées par un acteur bien spécifié.

Les acteurs définis précédemment effectuent un certain nombre de tâches, ces tâches sont résumées dans le tableau ci-dessous :

Acteur	Taches
Client	<p><b>T1</b> : lancer l'application.  <b>T2</b> : inscription.  <b>T3</b> : connexion.  <b>T4</b> : initialisation du mot de passe.  <b>T5</b> : consulter les lignes associées au client.  <b>T6</b> : consultation de la consommation du crédit de chaque ligne.  <b>T7</b> : recharger une ligne.  <b>T8</b> : consulter la consommation d'internet d'une ligne.  <b>T9</b> : recharger internet d'une ligne.  <b>T10</b> : consulter les nouveautés et les promotions.  <b>T11</b> : consulter les points fidélité d'une ligne.  <b>T12</b> : convertir les points de fidélité.  <b>T13</b> : afficher les points de vente.  <b>T14</b> : gérer les paramètres du compte.</p>

**Tableau III.4 : Spécification des tâches**

**III-11-2- Spécification des scénarios :**

Un scénario représente une succession particulière d'enchaînement s'exécutant du début à la fin du cas d'utilisation. Un ensemble de scénarios pour un cas d'utilisation représente tout ce qui peut arriver lorsqu'un cas d'utilisation est mis en œuvre.

Acteur	Tache	Scenario
Client	<b>T1</b> : lancer l'application.	<b>S1</b> : cliquer sur l'icône de l'application.
	<b>T2</b> : inscription.	<b>S2</b> : après le lancement de l'application, cliquer sur le bouton « Inscription », puis remplir le formulaire puis cliquer sur « valider ».
	<b>T3</b> : connexion.	<b>S3</b> : après le lancement de l'application, remplir les



		champs de connexion, puis cliquer sur le bouton « Connexion ».
	<b>T4</b> : initialisation du mot de passe.	<b>S4</b> : après le lancement de l'application, cliquer sur le lien « mot de passe oublié ? », remplir le formulaire et valider.
	<b>T5</b> : Choisir une ligne.	<b>S5</b> : après la connexion une interface s'affiche, cliquer sur l'item « mes ligne », et choisir une ligne.
	<b>T6</b> : consultation de la consommation du crédit d'une ligne.	<b>S6</b> : après avoir choisi une ligne, cliquer sur « mon solde », puis sur l'item consulter solde.
	<b>T7</b> : recharger la ligne.	<b>S7</b> : après avoir choisi une ligne, cliquer sur « mon solde », puis sur l'item recharger solde.
	<b>T8</b> : consulter la consommation d'internet.	<b>S8</b> : après avoir choisi une ligne, cliquer sur mon solde, puis sur l'item consulter internet.
	<b>T9</b> : recharger internet.	<b>S9</b> : après avoir choisi une ligne, cliquer sur mon solde, puis sur l'item recharger internet.
	<b>T10</b> : consulter aux nouveautés et aux promotions.	<b>S10</b> : après avoir choisi une ligne, une interface des offres s'affiche,
	<b>T11</b> : consulter les points fidélité.	<b>S11</b> : après avoir choisi une ligne, cliquer sur mon solde, puis sur l'item consulter points de fidélité.
	<b>T12</b> : convertir les points de fidélité.	<b>S12</b> : après avoir choisi une ligne, cliquer sur mon solde, puis sur l'item convertir points de fidélité.
	<b>T13</b> : afficher les points de vente.	<b>S13</b> : après la connexion une interface s'affiche, cliquer sur l'item « points de vente ».

	<b>T14</b> : gérer les paramètres du profile.	<b>S15</b> : cliquer sur le l'item « Mon Compte » puis cliquer sur le bouton « modifier mes informations ».
--	---	---

Tableau III.5 : scenarios

**III-12-Les cas d'utilisation :**

Un cas d'utilisation décrit une séquence d'actions qui fournit une valeur mesurable à un acteur et est dessinée comme une ellipse horizontale.

**III-12-1-Description textuelle des cas d'utilisation :****Cas d'utilisation : Inscrire pour créer un compte client.****Scenarios : S2****Acteur :** Client.**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône de l'application ;
2. Le système affiche l'interface d'accueil ;
3. L'utilisateur clique sur l'item « inscription » de l'interface ;
4. Le système affiche l'interface d'inscription avec formulaire d'inscription ;
5. L'utilisateur remplit le formulaire et clique sur le bouton valider.

Tableau III.6: cas d'utilisation « s'inscrire pour créer un compte clients ».

**Cas d'utilisation : connecter pour accéder au compte client.****Scenarios : S3****Acteur :** Client.**Description :**

1. Le système affiche l'interface de connexion avec formulaire pour s'authentifier ;
2. L'utilisateur remplit le formulaire et clique sur le bouton valider ;
3. Le système vous permet l'accès au compte.

Tableau III.7 : cas d'utilisation « connecter pour accéder au compte »

**Cas d'utilisation : afficher la consommation du crédit.****Scenarios : S6****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne, une interface s'affiche.
5. L'utilisateur clique sur l'item « mon solde ».
6. Le système affiche l'interface des soldes de la ligne ;
7. L'utilisateur clique sur le bouton « consulter » de la ligne solde ;
8. Le système affiche le résultat ;

**Tableau III.8 : cas d'utilisation « afficher mon solde »****Cas d'utilisation : Réinitialiser le mot de passe.****Scenarios : S4****Acteur : Client.****Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône de l'application ;
2. Le système affiche l'interface d'accueil ;
3. L'utilisateur clique sur le lien « mot de passe oublié » ;
4. Le système affiche l'interface de réinitialisation du mot de passe avec un formulaire ;
5. L'utilisateur remplit le formulaire puis clique sur le bouton valider.

**Tableau III.9 : cas d'utilisation : « réinitialiser le mot de passe ».**

**Cas d'utilisation : recharger le crédit.****Scenarios : S7****Acteur : Client.****Description :**

9. L'utilisateur doit se connecter ;
10. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
11. Le système affiche une interface des lignes associées à l'utilisateur.
12. L'utilisateur choisie une ligne, une interface s'affiche.
13. L'utilisateur clique sur l'item « mon solde ».
14. Le système affiche l'interface des soldes de la ligne ;
15. L'utilisateur clique sur le bouton « recharger » de la ligne solde ;
16. Le système affiche un formulaire ;
17. L'utilisateur remplit le formulaire puis clique sur « envoyer »

**Tableau III.10** : cas d'utilisation : « recharger le crédit ».**Cas d'utilisation : afficher la consommation d'internet.****Scenarios : S8****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne, une interface s'affiche.
5. L'utilisateur clique sur l'item « mon solde ».
6. Le système affiche l'interface des soldes de la ligne ;
7. L'utilisateur clique sur le bouton « consulter » de la ligne internet ;
8. Le système affiche une interface du résultat;

**Tableau III.11** : cas d'utilisation : « afficher la consommation d'internet »

**Cas d'utilisation : recharger Internet.****Scenarios : S9****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne, une interface s'affiche.
5. L'utilisateur clique sur l'item « mon solde ».
6. Le système affiche l'interface des soldes de la ligne ;
7. L'utilisateur clique sur le bouton « recharger » de la ligne internet ;
8. Le système affiche un formulaire ;
9. L'utilisateur remplit le formulaire puis clique sur « envoyer »

**Tableau III.12 : cas d'utilisation « recharger Internet ».****Cas d'utilisation : consulter les nouveautés.****Scenarios : S10****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne.
5. Le système affiche une interface des offres.

**Tableau III.13 : cas d'utilisation « consulter les nouveautés ».**

**Cas d'utilisation : consulter les points de fidélité.****Scenarios : S11****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne, une interface s'affiche.
5. L'utilisateur clique sur l'item « mon solde ».
6. Le système affiche l'interface des soldes de la ligne ;
7. L'utilisateur clique sur le bouton « consulter » de la ligne points de fidélité ;
8. Le système affiche une interface du résultat ;

**Tableau III.14:** cas d'utilisation « consulter les points de fidélité »**Cas d'utilisation : convertir les points de fidélité.****Scenarios : S12****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mes lignes ».
3. Le système affiche une interface des lignes associées à l'utilisateur.
4. L'utilisateur choisie une ligne, une interface s'affiche.
5. L'utilisateur clique sur l'item « mon solde ».
6. Le système affiche l'interface des soldes de la ligne ;
7. L'utilisateur clique sur le bouton « convertir » de la ligne points de fidélité ;
8. Le système affiche une interface du résultat ;

**Tableau III.15:** cas d'utilisation « convertir les points de fidélité ».

**Cas d'utilisation : chercher les points de vente****Scenarios : S13****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « points de vente ».
3. Le système affiche le résultat.

**Tableau III.16 :** cas d'utilisation « chercher les points de vente ».

**Cas d'utilisation : gérer les paramètres du compte.****Scenarios : S14****Acteur : Client.****Description :**

1. L'utilisateur doit se connecter ;
2. Le système affiche une interface, l'utilisateur clique sur l'item « mon compte ».
3. Le système affiche l'interface mon compte.
4. L'utilisateur clique sur le bouton « modifier mes informations »
5. Le système affiche le formulaire de modification ;
6. L'utilisateur remplit les champs puis clique sur valider en haut.
7. Le système affiche une interface du résultat ;

**Tableau III.17:** cas d'utilisation « gérer les paramètres du compte ».

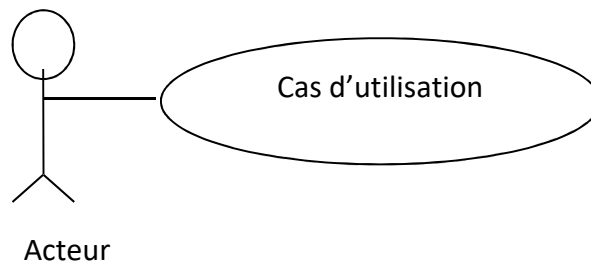
**III-13-Les diagrammes :****III-13-1-Diagramme de cas d'utilisation : use case Diagram**

Montre les interactions fonctionnelles entre les acteurs et le système à l'étude  
Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils représentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.



- **Cas d'utilisation :**

Un cas d'utilisation (use case) modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

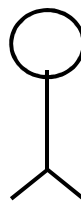


- **Les acteurs**

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation.

En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système.

L'acteur est représenté dans l'UML comme suit :



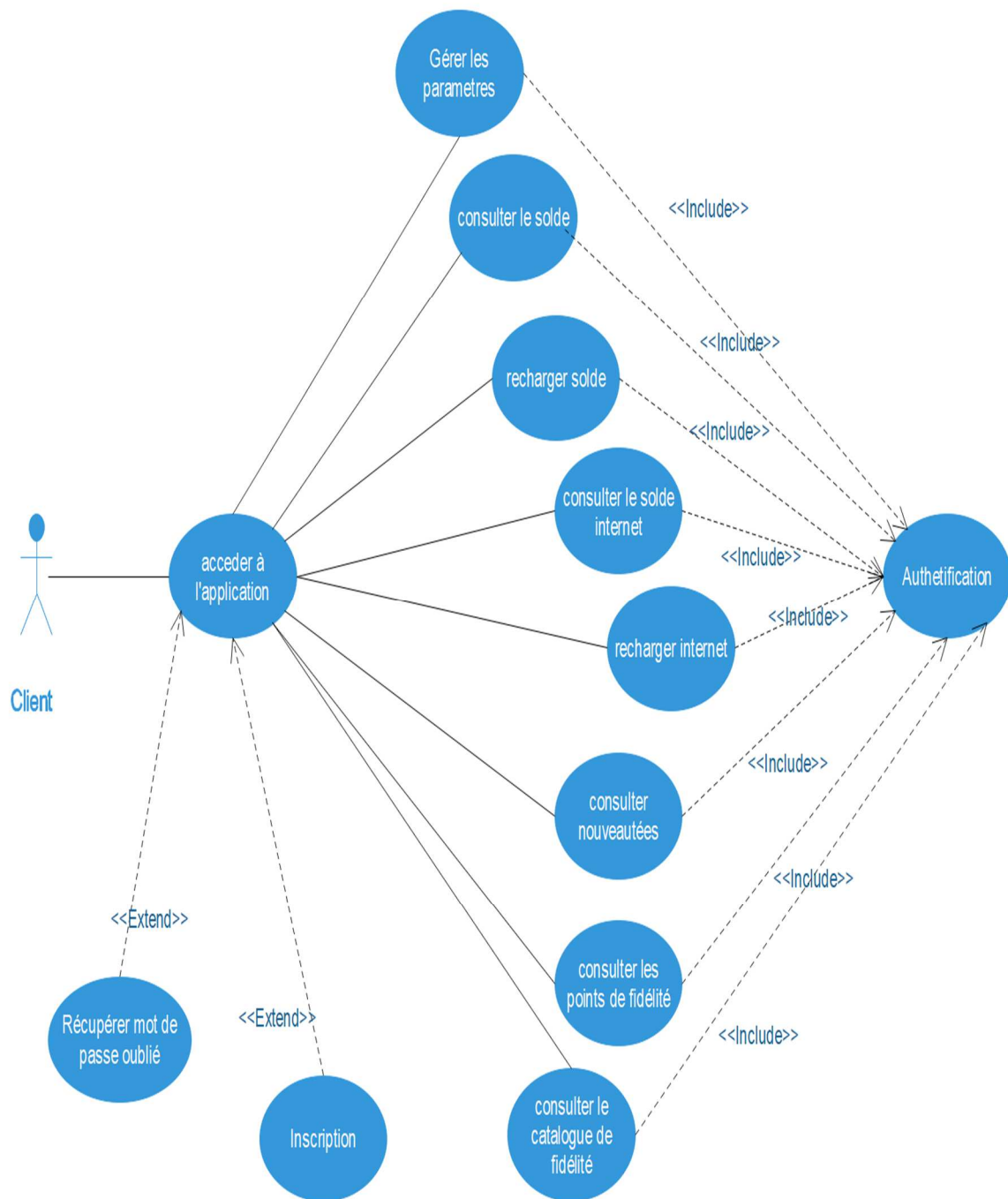
- **Liens entre cas d'utilisation : include et extend**

Il est parfois intéressant d'utiliser des liens entre cas (sans passer par un acteur), UML en fournit de deux types : la relation utilise (include) et la relation étend (extend).

**L'utilisation de cas :** La relation **utilise** (include) est employée quand deux cas d'utilisation ont en commun une même fonctionnalité et que l'on souhaite factoriser celle-ci en créant un sous cas, ou cas intermédiaire, afin de marquer les différences d'utilisation.

**Extension de cas (extend) :** nous dirons qu'il y a extension d'un cas d'utilisation quand un cas est globalement similaire à un autre, tout en effectuant un peu plus de travail (voire un travail plus spécifique). Cette notion à utiliser avec discernement permet d'identifier des cas particuliers (comme des procédures à suivre en cas d'incident) dès le début ou lorsque l'attitude face à un utilisateur spécifique du système doit être spécialisée ou adaptée. Il s'agit grosso modo d'une variation du cas d'utilisation normale.

**La relation de généralisation ou spécialisation :** Les cas d'utilisation descendant hérite de la description de leur parent commun. Chacun entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires

**Figure III.3 :** diagramme de cas d'utilisation.

**III-13-2-Diagramme de séquence : Séquence diagram :**

Un diagramme de séquence met en évidence le classement des messages par ordre chronologique. On forme un diagramme de séquence en plaçant d'abord les objets qui participent à l'interaction en haut du diagramme. Le long de l'axe des abscisses.

En générale. On place l'objet qui débute l'interaction à gauche, puis on continue en progressant vers la droite, les objets les plus subordonnés étant tout à fait à droite. On place ensuite les messages envoyés et reçus par ces objets le long de l'axe des ordonnées, par ordre chronologique, du haut vers le bas. Cela donne au lecteur une indication visuelle claire du flot de contrôle dans le temps.

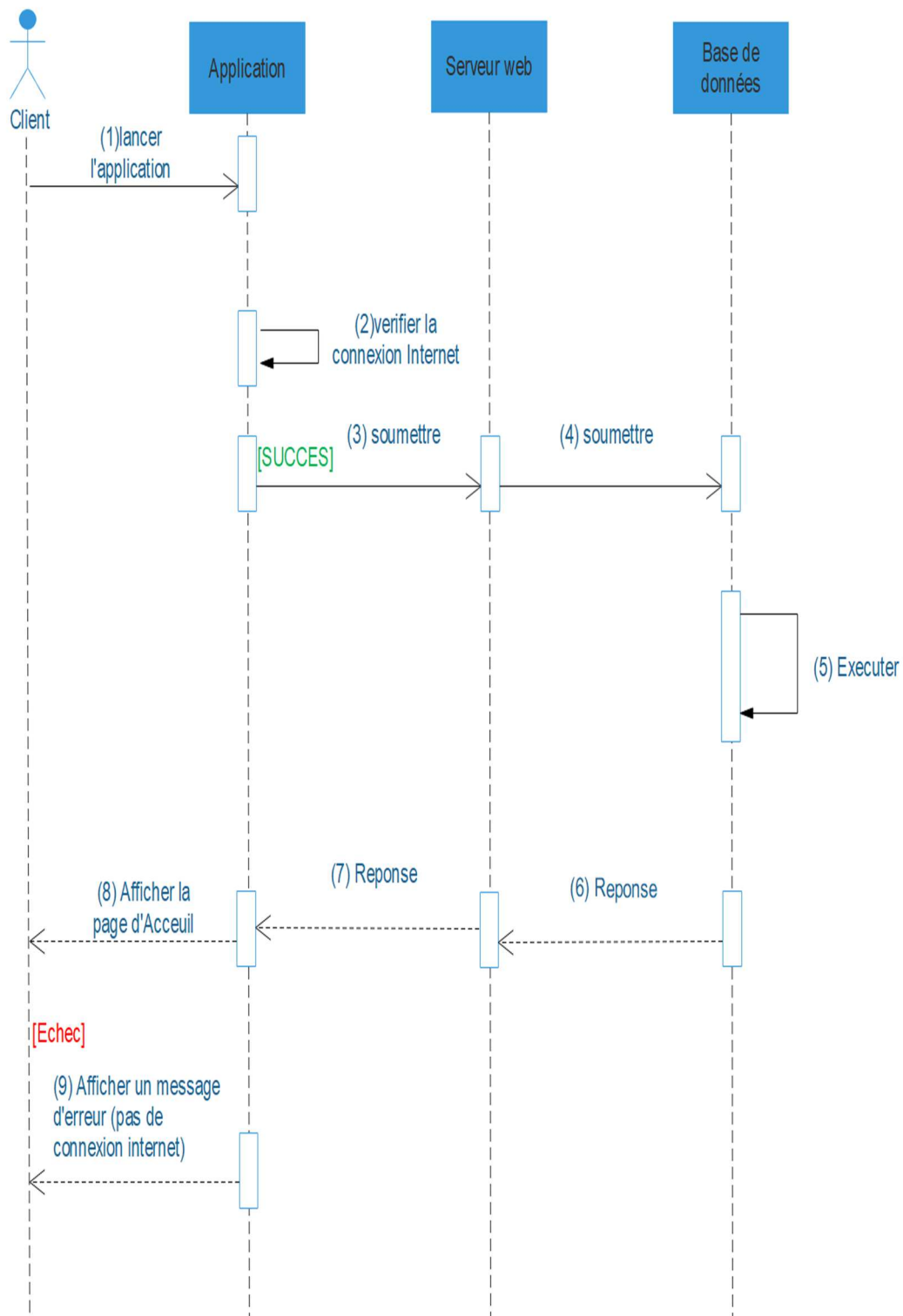
En générale, les diagrammes de séquence contiennent :

- **L'objet :**

Est une manifestation concrète d'une abstraction à laquelle on peut appliquer un ensemble d'opérations et qui possède un état capable de mémoriser les effets de ces opérations. On représente un objet en soulignant son nom. Le lien : est une liaison sémantique entre objets, en générale, il s'agit d'une instance d'une association. Chaque fois qu'une classe est reliée à une autre par une association, il peut y avoir un lien entre les instances des deux classes, et chaque fois qu'un lien existe entre deux objets, le premier objet peut envoyer un message au deuxième.

- **Le message :**

Est la spécification d'une communication entre objets, qui transporte des informations et qui s'affiche dans le but de déclencher une activité. La réception d'une instance de message peut être considérée comme une instance d'un événement.



**Figure III.4:** diagramme de séquence de cas d'utilisation « lancer l'application ».

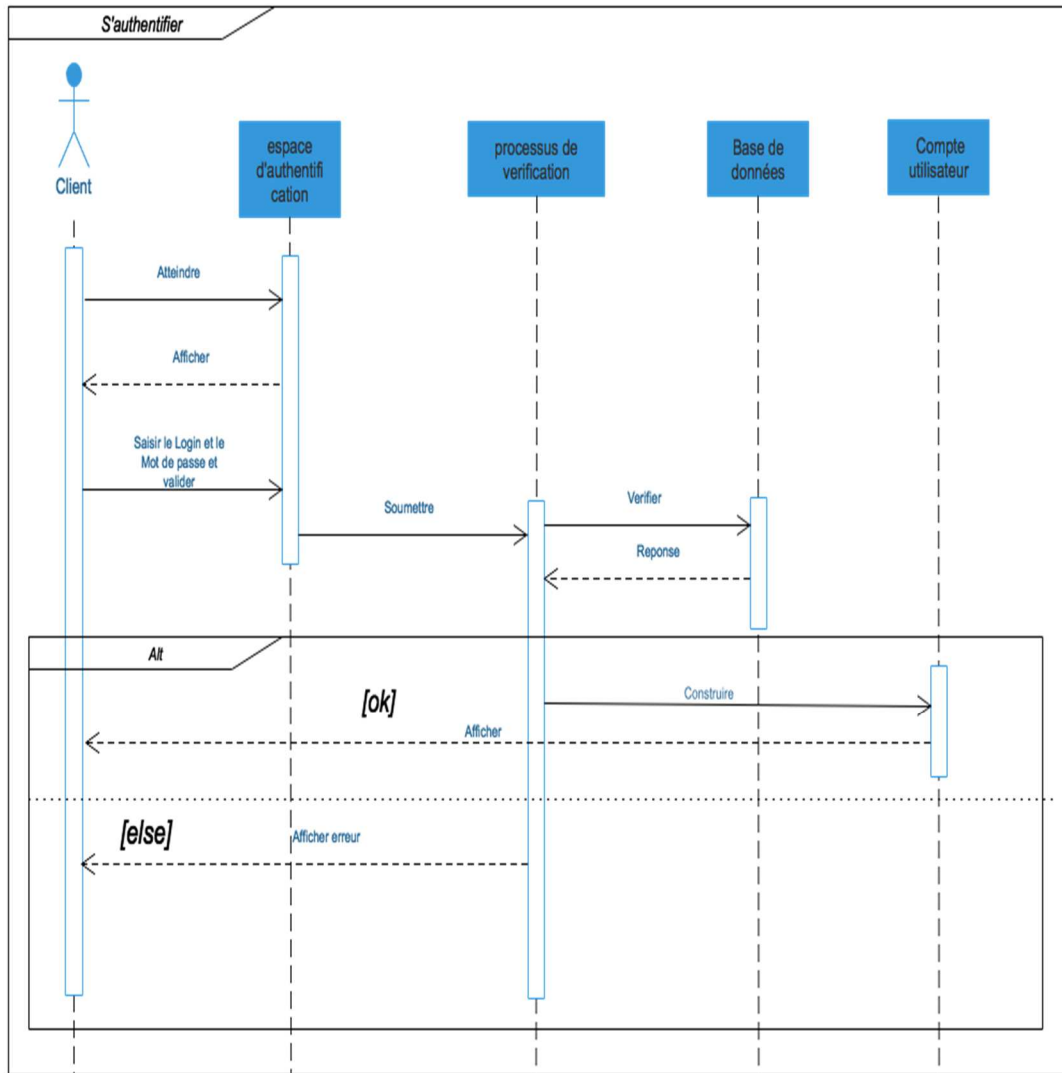


Figure III.5 : diagramme de séquence de cas d'utilisation « connexion ».

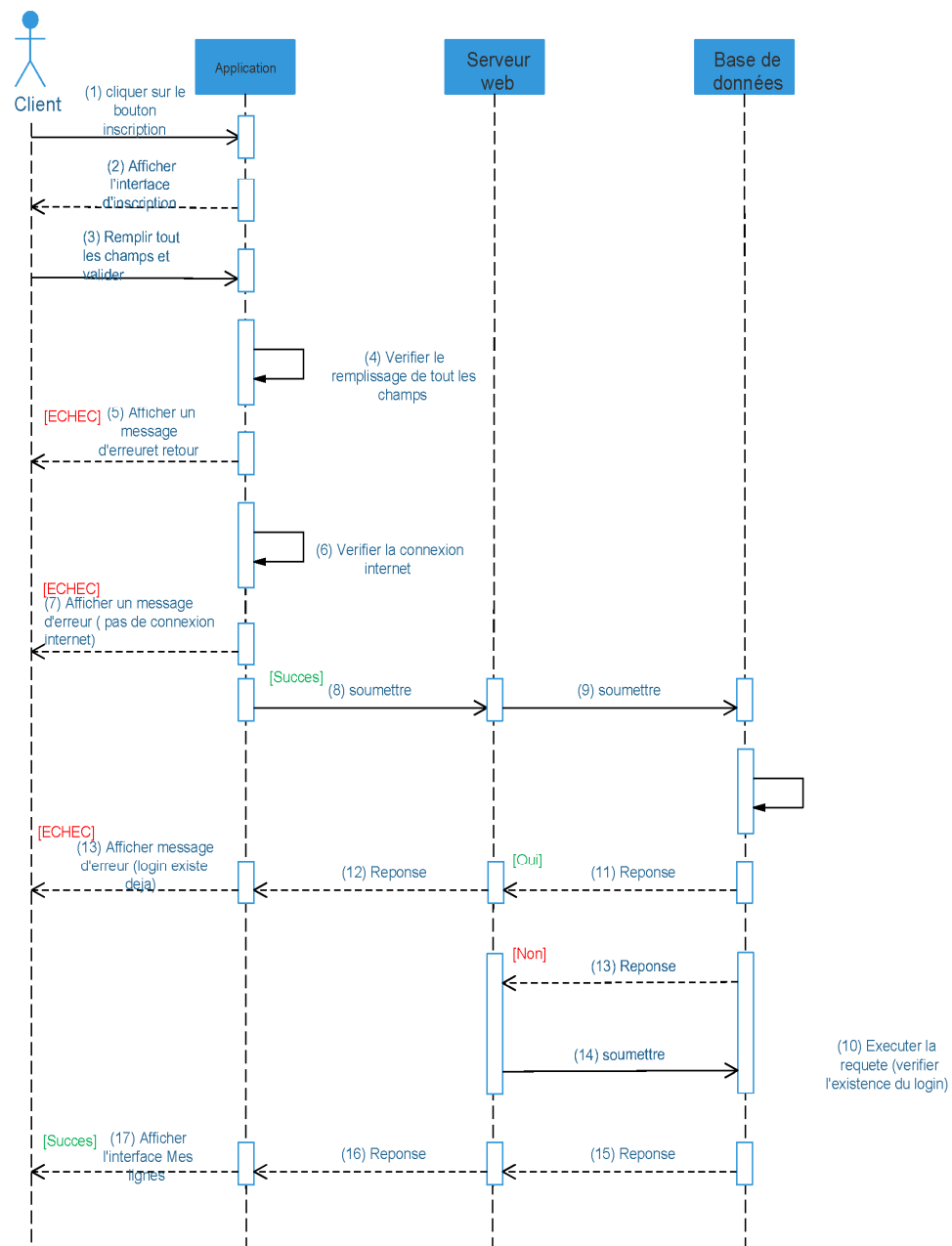


Figure III.6 : diagramme de séquence de cas d'utilisation « Inscription ».

### III-13-3-Diagramme de classes : Class diagram :

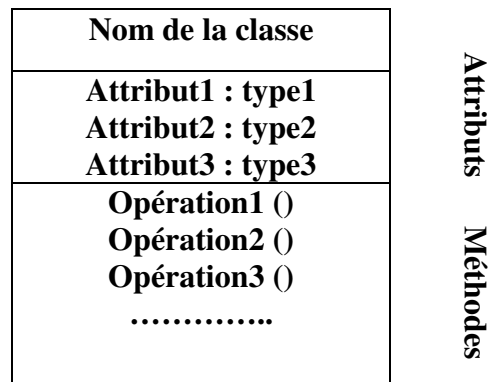
Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ses classes. Outre les classes, ils représentent un ensemble d'interfaces et de paquetages.

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objet de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet.

Les diagrammes de classes contiennent généralement les éléments suivant :

- **Les classes :**

Une classe (class) est la description d'un ensemble d'objets ayant une structure (attributs), un comportement (opérations et méthodes) et des relations similaires. Une classe est généralement représentée par un rectangle ayant trois compartiments : le nom de la classe (avec éventuellement un stéréotype et des propriétés), les attributs et les opérations. On peut simplifier la représentation en ne conservant que le premier compartiment. Inversement, on peut rajouter des compartiments (éventuellement nommés) pour les règles de gestion (business rules), les responsabilités, les événements, les exceptions, etc. Une classe est symbolisée par un rectangle.



- **Attribut :**

Un attribut est une propriété nommée d'une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Une classe peut ne pas avoir, comme elle peut avoir un ou plusieurs attributs.

- **Opération :**

Une opération est une abstraction de ce que peut réaliser un objet et qui est réalisable par tous les objets de la classe. Une classe peut ne pas avoir comme elle peut avoir plusieurs opérations.

- **Les relations d'association d'agrégation et de composition :**
  - **Association** : relation sémantique durable entre deux classes, qui décrit un ensemble de liens entre instances. Une association est bidirectionnelle par défaut, sauf si l'on restreint sa navigabilité en ajoutant une flèche.
  - **Rôle** : nom donné à une extrémité d'une association ; par extension, manière dont les instances d'une classe voient les instances d'une autre classe au travers d'une association.
  - **Multiplicité** : le nombre d'objets (min...max) qui peuvent participer à une relation avec un autre objet dans le cadre d'une association. Multiplicités fréquentes :
    - 0..1 = optionnel (mais pas multiple)
    - 1 = exactement 1
    - 0..\* = \* = quelconque
    - 1..\* = au moins 1
  - **Une agrégation** : est un cas particulier d'association non symétrique exprimant une relation de contenance.



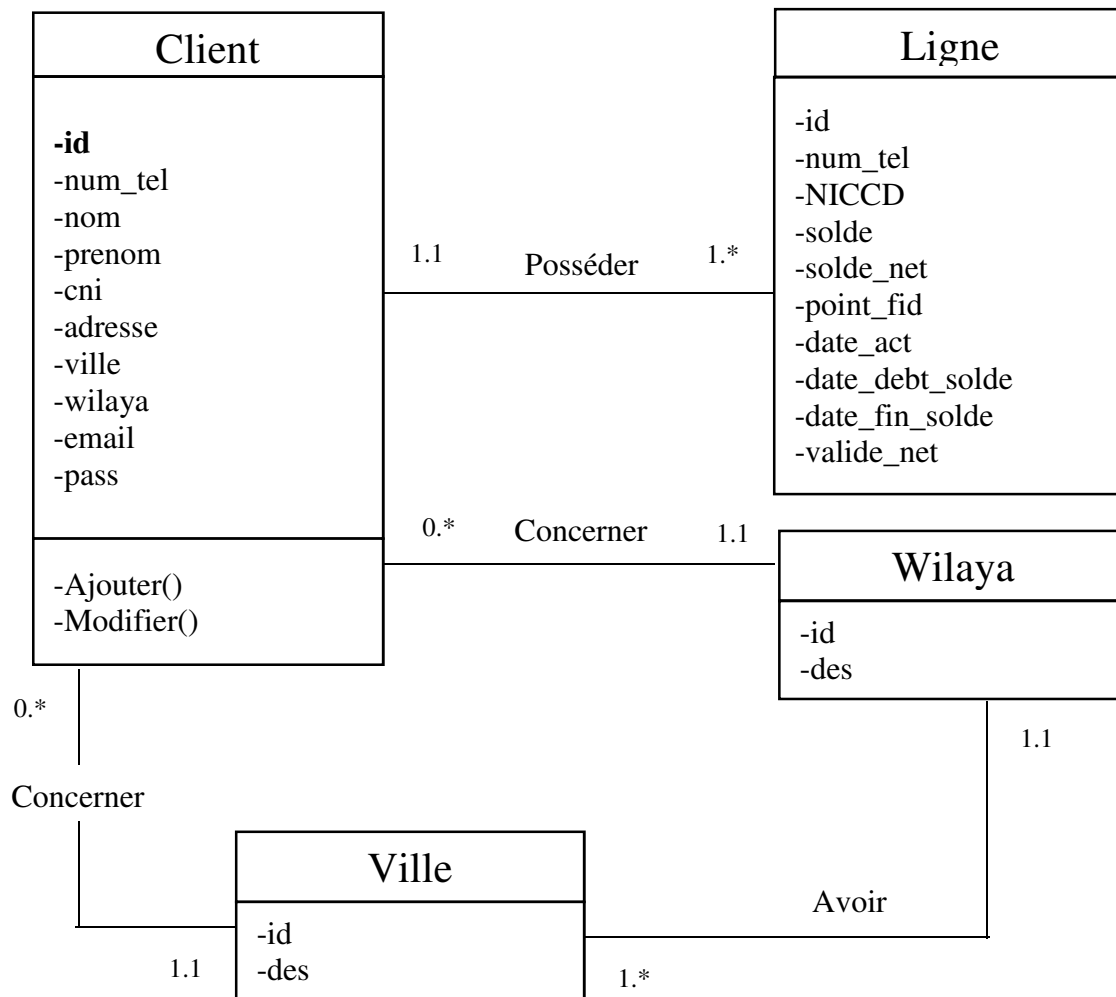


Figure III.7 : diagramme de classe.

**III-14-La base de données :**

Vu la sensibilité des données du secteur de la téléphonie et de la communication ; nous étions dans l'obligation de créer une nouvelle base de données pour notre application. Cette base de données contient les tables suivantes :

**La table Client:**

Champs	Type de donnée	Clé(s)	Signification
id	Int	Primaire	Identificateur client
num_tel	varchar		Numero du telephone
nom	varchar		Nom
prenom	varchar		Prénom
cni	varchar		Carte nationale
adresse	varchar		Adresse
commune	varchar		Commune
wilaya	varchar		Wilaya
email	varchar		Email
pass	varchar		Mot de passe
id_wilaya	int	index	Identificateur wilaya
id_ville	int	index	Identificateur ville

**La table Ligne:**

Champs	Type	Clés	Signification
id	int	Primaire	Identificateur de la ligne
num_tel	varchar		Numéro de téléphone
NICCD	varchar		Numéro de série de la carte SIM
solde	double		Solde
solde_net	double		Solde Internet
Point_fid	int		Solde de points de fidélité
date_act	D		Date d'activation de la ligne

Date_debt_solde	D		Date de début du rechargement de crédit
Date_fin_solde	D		Date fin du rechargement crédit
Validite_net	D		Validité du solde Internet
Id_client	int	index	Identificateur client

**La table ville :**

Champs	Type	Clés	Signification
id	int	Primaire	Identificateur ville
des	varchar		Nom de la ville
Id_wilaya	int	index	Identificateur de la wilaya

**La table wilaya :**

Champs	Type de donnée	Clé(s)	Signification
id	int	Primaire	Identificateur de wilaya
des	varchar		Nom de la wilaya

**Conclusion :**

Dans ce chapitre nous avons présenté la modélisation du système à réaliser en utilisant les diagrammes d'UML en commençant par la spécification des besoins et les divers cas d'utilisation, puis la conception des diagrammes des cas d'utilisation, diagrammes de séquences, diagramme des classes, et enfin nous avons défini les différentes tables de notre application.

Dans le chapitre qui va suivre, nous allons décrire les outils qui vont servir à développer notre application, ainsi que ses interfaces principales.

# ***Chapitre IV***

## ***Réalisation***

## Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, dans ce dernier chapitre nous allons présenter l'environnement de développement, les outils choisis pour l'implémentation de notre application, ainsi que des captures d'écrans de quelques interfaces, présentant les différentes fonctionnalités de notre système vues dans le chapitre précédent.

### IV-1-Environnement matériel et logiciel

#### IV-1.1. Environnement matériel

##### IV-1.1. 1-Architecture matérielle

L'architecture de notre application est à 3 niveaux (architecture 3-tiers), elle est partagée entre :

- Le client Android : Conteneur d'application et demandeur de ressources.
- Le serveur Web : Vue que les données seront communiquées entre deux environnements hétérogènes, le rôle principal du serveur web est de gérer la communication entre le client Android et le serveur de base de données,
- Le serveur de base de données fournis les données au serveur web.

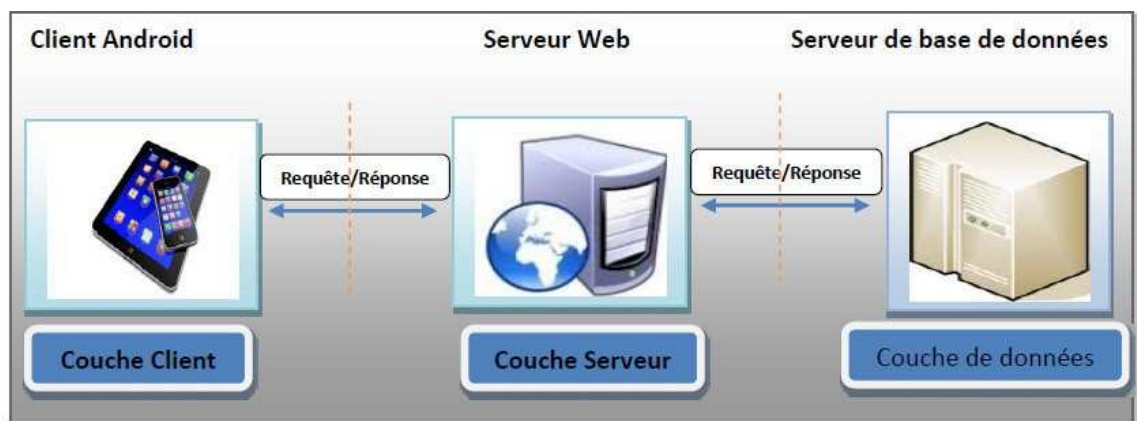


Fig. IV.1 : Architecture matériel du système




##### IV-1.1. 2-Matériels utilisés

Pour la réalisation de l'application nous avons utilisé un pc portable pour le développement ayant les caractéristiques suivantes :

- DELL XPS13 Intel core i7 2.270-2, 90 GHz.
- 16 Go de mémoire vive.
  - Windows 10 professionnel - 64 bits.
- Un Smartphone Samsung pour réaliser les tests.

### IV-1.1.3-Technologie

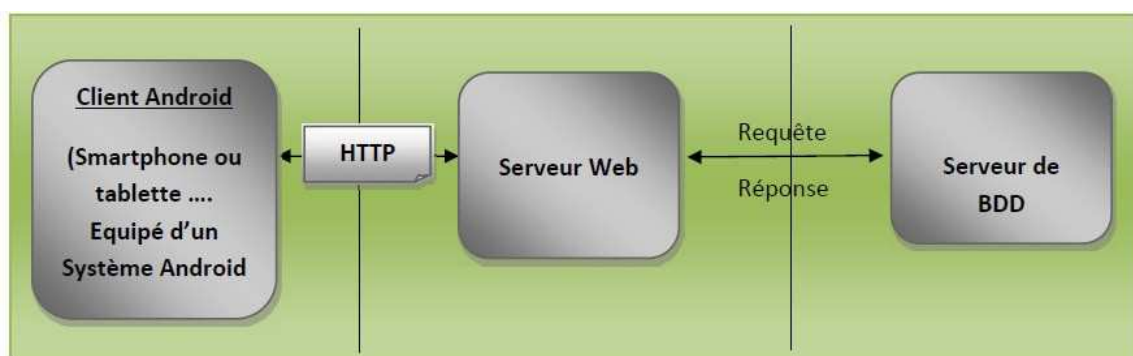
Ci-dessous un tableau représentant les différentes technologies utilisées dans notre application :

	<u>Android</u> Système d'exploitation mobile open source pour smartphone, PDA et terminaux mobiles
	<u>PHP</u> Langage de scripts libre principalement utilisé pour produire des pages web dynamiques
	<u>MySQL</u> Système de gestion de base de données (SGBD).

**Tableau IV.2 :** Les différentes technologies utilisées dans l'application

La méthode la plus répandue faire se connecter à une base de données MySQL à partir d'un appareil Android, est de l'effectuer par l'intermédiaire d'un serveur web. Le SGBD MySQL est habituellement utilisé avec le langage PHP, donc la façon la plus simple et la plus évidente est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP du système Android. Nous avons codé les données dans le format JSON.

L'architecture 3-tiers du point de vue technologique, le client est la plateforme Android, le serveur web est le PHP et le serveur de bases de données est le MySQL.



**Fig. IV.3 :** Architecture 3-tiers du point de vue technologique

### IV-1.2. Environnement logiciel

Pour pouvoir réaliser une application dans des bonnes conditions il faut choisir son environnement de travail selon les besoins. Nous avons opté pour la réalisation d'une application mobile sous le système Android ce qui nous impose de travailler sous Android Studio avec le langage JAVA et le SDK Android pour son développement, afin d'obtenir un fichier à extension.apk qui sera par la suite installé sur des terminaux mobiles de différents types fonctionnant avec le système Android 3.0 ou plus.

Le développement d'applications pour Android se fait entièrement en Java, ce dernier est un langage puissant orienté objet, utilisé très largement dans le monde du développement, en utilisant des IDE tels que : Eclipse, NetBeans et Android Studio.

#### IV-1.2.1-PHP

**PHP: Hypertext Preprocessor**, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage interprété orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme la base de la création des sites Internet dits dynamiques.

#### IV-1.2.2-Yii Framework

Le Yii Framework ("Yes, It Is") est un *framework* pour PHP 5 et utilise le paradigme de programmation orientée objet. Il est destiné au développement d'applications Web. Yii nécessite au moins la version 5.1.0 de PHP pour la version 1 et 5.4.0 de PHP pour la version 2. Une documentation complète est disponible. La communauté autour du projet est très active. Le créateur et développeur principal de Yii est [Qiang Xu], qui a également développé et maintenu le framework PHP Prado pendant 3 ans. Yii est d'ailleurs le successeur officiel de Prado.



**Fig. IV.4** : Logo framwork yii

### IV-1.2.3-PhpStorm

PhpStorm est un éditeur pour PHP, HTML et JavaScript, édité par JetBrains. Il contient une correction automatique des erreurs.

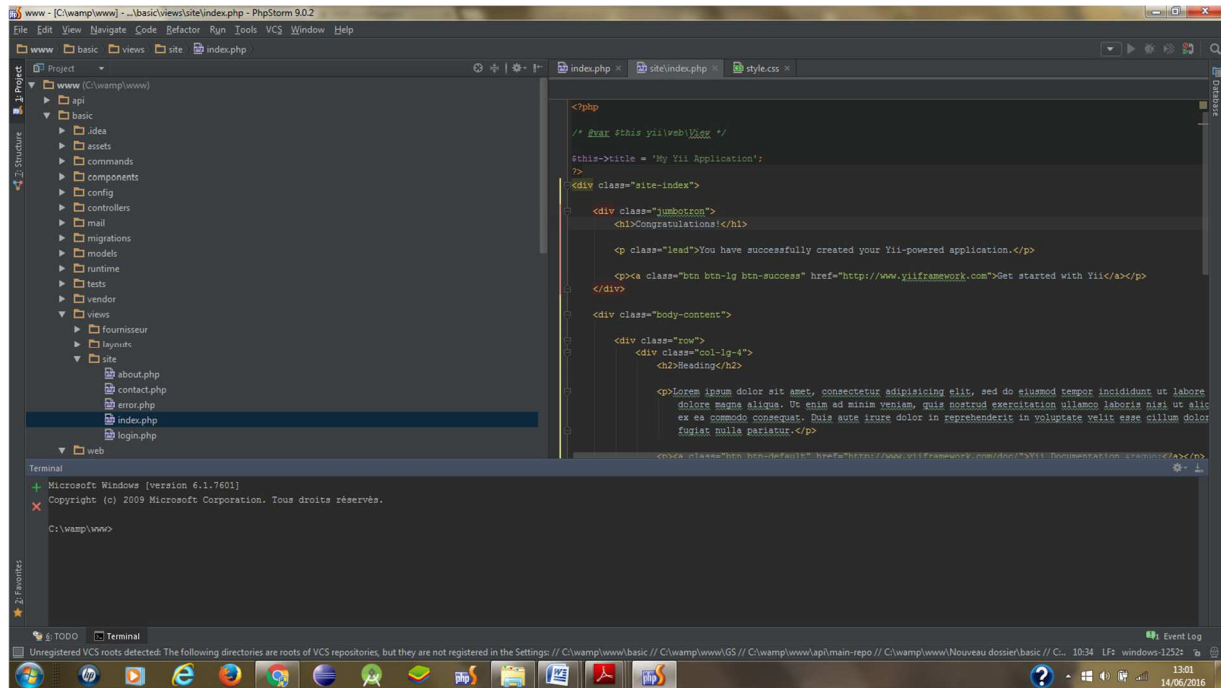


Fig. IV.5 : Interface de l'environnement PhpStorm

### IV-1.2.4-JAVA

Est un langage de programmation informatique orienté objet, La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou Linux, avec peu ou pas du tout de modifications. Pour cela, diverses plateformes associées visent à garantir la portabilité des applications développées en Java.

### IV-1.2.5-XML

XML (eXtensible Markup Language, soit « Langage de balisage extensible ») est un langage de balisage définissant un format universel de représentation des données, permettant de décrire la structure hiérarchique d'un document. Le fichier à extension .xml contient à la fois les données et les indications sur le rôle que jouent ces données, ces indications (ou balises) permettent de déterminer la



structure des documents.

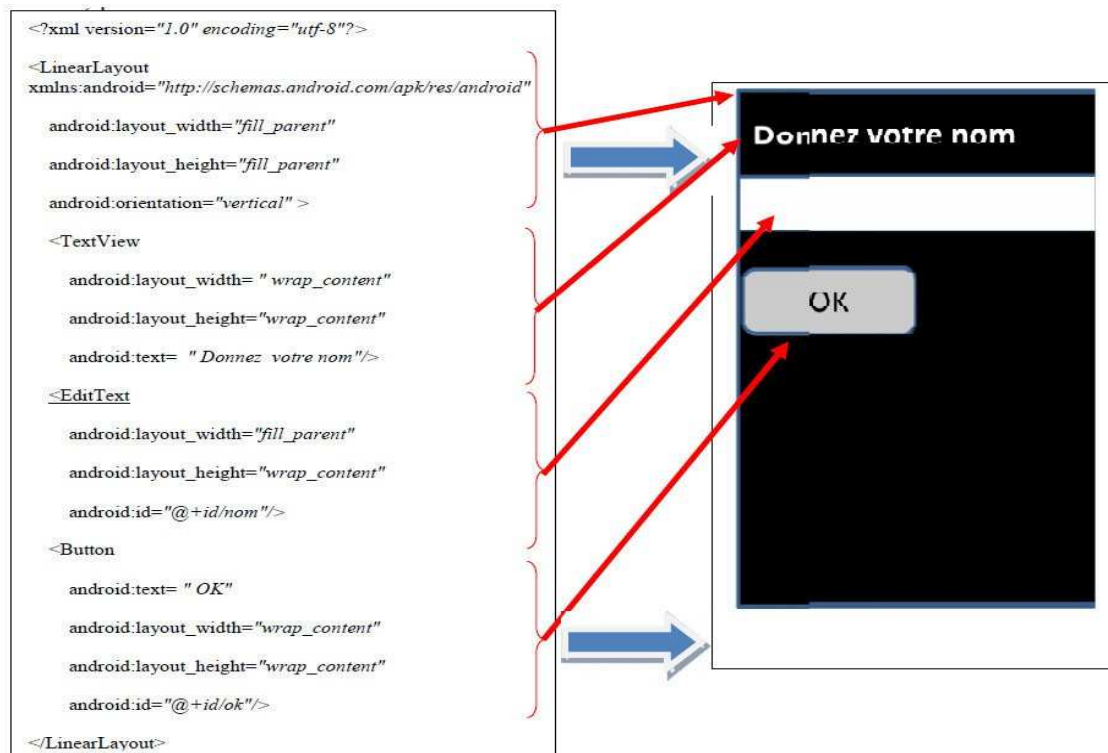


Fig. IV.6 : Exemple d'un fichier XML

#### IV-1.2.6-Android Studio

Android Studio est un environnement de développement des applications Android. Il est basé sur la version IntelliJ.

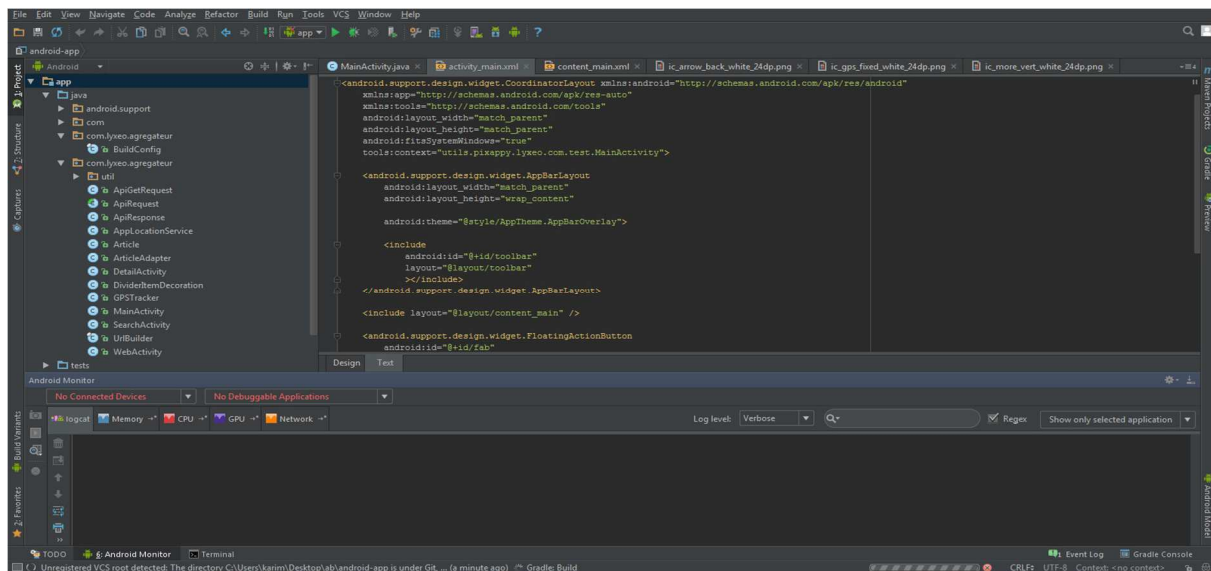


Fig. IV.7 : Interface de l'environnement Android Studio 1.1.0

### IV-1.27-SDK

Signifie Software Development Kit, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique. Un SDK contient du code, permettant de concevoir une interface ou une partie d'une interface numérique (web, mobile, jeux, logiciels de recherches, widget météo...). Ce code est conçu avec le langage de programmation correspondant au terminal (ordinateur, téléphone, tablette...) et au système de navigation ciblés. Par exemple, pour développer une application mobile (logiciel) pour iPhone (terminal mobile avec système d'exploitation iOS), il faut utiliser le SDK iOS mobile. Ce code est organisé sous forme de librairies (ou bibliothèques logicielles) c'est-à-dire des collections de fonctions prédéfinies, de points d'accès à du matériel et à des fonctionnalités système (ou natives) d'un terminal.

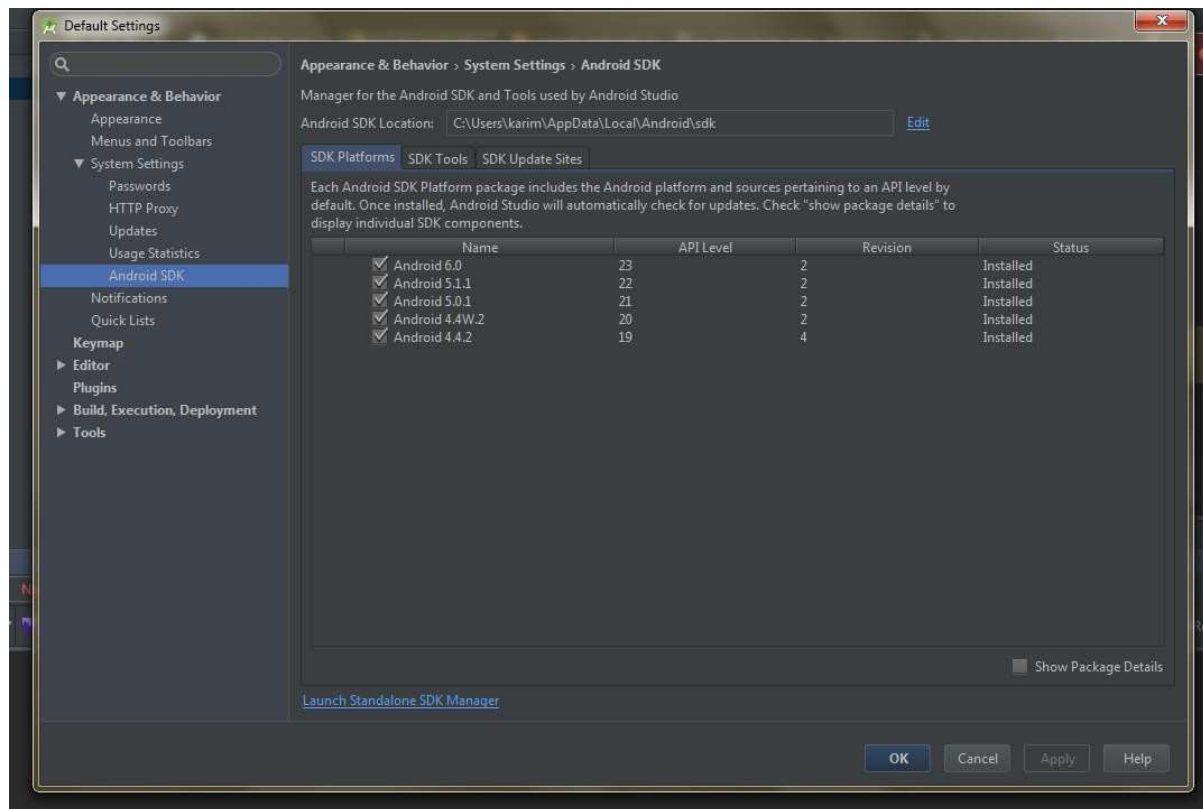


Fig. IV.8 : Interface de l'Android SDK Manager

### IV-1.28-Git

Git est un logiciel de gestion de versions, il est utilisé principalement par les développeurs. En effet, il est quasi exclusivement utilisé pour gérer des codes sources, car il est capable de suivre l'évolution d'un fichier texte ligne de code par ligne de code. Il est fortement conseillé pour gérer un projet informatique.

Ce type de logiciel est donc par conséquent deux utilités principales :

- ✓ Suivre l'évolution d'un code source : pour retenir les modifications effectuées sur chaque fichier et être ainsi capable de revenir en arrière en cas de problème.
- ✓ Travailler à plusieurs : Si deux personnes modifient un même fichier en même temps, leurs modifications doivent pouvoir fusionner sans perte d'information.

Quelques commandes de Git :

Git Commit : permet d'enregistrer la nouvelle version du code source.

Git Push: permet d'envoyer le code source sur le serveur pour l'enregistrer.

Git Pull: permet de télécharger les nouveautés du projet modifié depuis le serveur.

Git Status: permet de savoir l'état des fichiers modifiés (s'il y en a) par apport au dernier commit effectué.

### **IV-1.3. Protocole et format de données**

#### **IV-1.3.1. Protocole de communication**

Dans notre projet, nous avons utilisé le protocole HTTP, afin de communiquer les données entre le client Mobile et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client (facilite le dispatch des fonctions).

#### **IV-1.3.2. Format de données communiquées**

JSON (JavaScript Object Notation) est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter des informations structurées.

Un document JSON ne comprend que deux éléments structurels :

- ✓ Des ensembles de paires nom / valeur.
- ✓ Des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- ✓ Des objets.
- ✓ Des tableaux.
- ✓ Des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou nulle.

Le principal objectif du choix d'utilisation du JSON, dans notre application, est qu'il est simple à mettre en œuvre. Comme il présente les avantages suivants:

- ✓ Facile à apprendre, car sa syntaxe est réduite et non-extensible ;
- ✓ Ses types de données sont connus et simples à décrire ;
- ✓ Peu verbeux et léger, ce qui le rend bien adapté aux terminaux mobiles contrairement au langage XML qui est très verbeux.

## **IV-2. Présentation de quelque interface de notre application**

### **Interface authentication**

Dès que le client lance l'application une interface d'accueil va être affichée, pour se connecter le client doit remplir les champs login (numéro de téléphone) et mot de passe puis cliquer sur le bouton connexion, afin de pouvoir se connecter à son propre compte. En cas d'erreur un message va être affiché.



**Figure IV.9 :** Interface authentification.

### **Interface inscription**

Une fois l'application lancée une interface d'accueil va être affichée, pour s'inscrire il suffit de cliquer sur le bouton Inscription, une autre interface pour l'inscription va s'afficher le client doit remplir les champs puis cliquer sur le bouton valider, pour créer son propre compte. En cas d'erreur un message va être affiché.



The image shows a mobile application interface for registration. At the top, there is a red header bar with the word "Inscription" in white and a right-pointing arrow. Below the header, the title "Inscription" is centered. The form consists of several input fields and dropdown menus:

- Numéro de téléphone :** A text input field with an orange border.
- Nom :** A text input field.
- Prénom :** A text input field.
- Adresse :** A text input field.
- Numéro de carte d'identité :** A text input field.
- Wilaya :** A dropdown menu with "Tizi ousou" selected.
- Ville :** A dropdown menu with "Tizi ousou" selected.
- E-mail :** A text input field.

The status bar at the top of the screen shows 4G, signal strength, 100% battery, and the time 00:43.

**Figure IV.10:** Interface inscription.

### Interface principale :

Une fois l'authentification est faite une autre interface d'accueil va apparaître



Figure IV.11 : Interface accueil .

### Interface solde

Après l'authentification, si l'utilisateur clique sur le bouton mes lignes une interface va apparaître, le client doit cliquer sur l'item mon solde et une interface affichant les soldes de la ligne apparaît .



**Figure IV.12 :** Interface mon solde.



### Interface changer mot de passe

En cliquant sur le bouton mon compte une interface va s'afficher cette interface contient les coordonnées du client, s'il clique sur le bouton modifier mot de passe une fenêtre va s'afficher le client doit remplir les champs et cliquer sur le bouton confirmer en cas d'erreur un message va s'afficher



**Figure IV.13 :** Interface changer mot de passe.

### **Conclusion**

Dans ce chapitre nous avons présenté les plates-formes matérielles et logicielles sur lesquelles nous avons développé notre projet, ainsi que les technologies employées. Nous avons, par la suite, présenté les interfaces les plus significatives de notre application.

# *Conclusion*

## Conclusion générale

L'élaboration de notre travail est dans le but de concevoir une application dédiée aux terminaux mobiles disposant de la plateforme Android. Cette application permet aux abonnés, d'avoir des informations sur les différents services et produits **Ooredoo** directement à partir de leur Smartphone.

Pour ce faire, nous avons recouru à différentes technologies, outils et orientation jugées nécessaires pour aboutir à l'objectif de notre application.

La réalisation de ce travail nous a permis d'élargir aussi bien nos connaissances théoriques que pratiques en rapport avec la technologie pour terminaux mobiles, et d'acquérir de nouvelles connaissances sur le langage de modélisation « UML » qui est une technique bien bénéfique dans la conception des applications. On a pu aussi améliorer nos connaissances sur le domaine de l'embarqué en abordant plusieurs aspects techniques d'Android.

Ce projet nous a également donné l'occasion de mieux connaître le milieu des sociétés de service informatique et de consolider nos expériences professionnelles.

Nous avons pu réussir la tâche qui nous a été confiée, sauf. Néanmoins notre application peut être améliorée et étendue en intégrant les fonctionnalités liées au grand compte (comptes sociétés).

## Bibliographie

<http://www.universalis.fr/encyclopedie/systemes-d-exploitation-informatique/>

<https://www.javatpoint.com/android-history-and-versions>

[http://www.frandroid.com/android/410307\\_dou-vient-la-mascotte-dandroid-bugdroid](http://www.frandroid.com/android/410307_dou-vient-la-mascotte-dandroid-bugdroid)

<http://belandroid.e-monsite.com/pages/lexique/autre/bugdroid.html>

<https://www.quora.com/Who-invented-the-Android-mascot-robot>

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

<https://blog.erlem.fr/android/31-architecture-generale-d-android>

<http://igm.univ-mlv.fr/~dr/XPOSE2008/android/index.html#histo>

[http://www.fr.w3eacademy.com/android/android\\_environment\\_setup.htm](http://www.fr.w3eacademy.com/android/android_environment_setup.htm)

Benoît CHARROUX, Aomar OSMANI, Yann THIERRY-MIEG, "UML2 pratique de la modélisation"  
2ème édition.

Pascal Roques, "UML2 par la pratique, Etude de cas et exercices corrigés", 5ème Edition,  
Editions EYROLLES.

<http://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-composants-deploiement#L8-3>

<http://www.maxicours.com/se/fiche/5/4/230354.html>

[www.clubic.com/telecharger-fiche49380-apache-http-server-pour-windows.html](http://www.clubic.com/telecharger-fiche49380-apache-http-server-pour-windows.html)

<http://android-studio.fr.uptodown.com/windows>

<http://www.clubic.com/telecharger-fiche430837-phpstorm.html>

<https://firebird.developpez.com/telecharger/detail/id/1702/IBExpert>

Eric Despet, Cyril Pierre de Gever, "Php5 avancé" 4e édition.

<https://openclassrooms.com/courses/structurez-vos-donnees-avec-xml/qu-est-ce-que-le-xml>

## Résumé

L'objectif était de réaliser une application mobile qui vise comme clientèle toute personne possédant une carte SIM **Ooredoo** et un Smartphone sous Android OS 5.0 et souhaitant recevoir les différents services offerts par l'opérateur téléphonique **Ooredoo**.

L'application permet en effet, d'avoir des informations sur les différents services et produits **Ooredoo**, de pouvoir activer/désactiver les différents forfaits (voix et data) en un clic et de pouvoir consulter les informations sur leur compte.

Nous espérons enfin que le travail que nous avons effectué a été à la hauteur.

Android

Android studio

Mobile