

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERI de Tizi-Ouzou**  
**Faculté de Génie Electrique et d'informatique**  
**Département d'informatique**



# Mémoire de fin d'étude

## Mémoire de fin d'étude

*En vue de l'obtention du diplôme de Master en informatique*  
*Option Conduite de projet informatique*

### *Thème*

*Conception et réalisation d'une application  
Android « blocage d'appels indésirables »*

**Réalisé par :**

Melle: Dihia SEHL

Melle: Sabrina KHAZEM

**dirigé par:**

Mr:M. KERBICHE

Promotion 2014/2015

## *Remerciement*

*Nous tenons avant tout à remercier Dieu tout puissant de nous avoir donné la force et la volonté pour achever ce modeste travail.*

*Nous tenons à remercier particulièrement nos parents ; notre succès demeure de loin le fruit de leurs longues années de sacrifices et d'éducation.*

*Nos vifs remerciements, accompagnés de toute notre gratitude, vont tout d'abord à notre promoteur Mr KERBICHE.M, pour son suivi et son engagement lors de l'élaboration de ce projet.*

*Nous remercions vivement les membres du jury qui nous ont fait l'honneur d'accepter d'évaluer notre travail*

*Nous ne pouvons nous empêcher d'avoir une pensée pour ceux et celles qui ont répondu présents et nous ont offert leur soutien moral dans les moments difficiles et qui étaient à nos cotés pour partager avec nous les moments de joie.*

# ❧ Dédicaces ❧

*Je dédie ce modeste travail :*

*À mes très chers parents Nacer et Zahia qui n'ont pas cessé de m'encourager durant tout mon cursus que dieu me les garde ;*

*À mes frères Salah et Radouane et mon unique sœur Souhila que j'aime plus que tout, à mon Futur neveu "Wassim"*

*À mes cousins, cousines..., toute la famille*

*et tous ceux que j'aime,*

*À Tous mes amis Nawuel Hayet Fateh Anis Samir Raffik  
Omar, Sabrina;*

*À ma meilleure, ma deuxième sœur, ma Pimou à moi "Fatima"*

*Ainsi qu'à toutes les personnes qui nous ont aidé à la réalisation de notre travail. Et en particulier*

*Said et Ghiles*

*Sehl Dihia*

# ❧ Dédicaces ❧

*Je dédie ce modeste travail :*

*A ma très chère mère qui a été la première à m'encourager à aller loin dans les études et qui m'a inculqué le goût du travail, de la rigueur et de l'ambition.*

*A Mon défunt Père que le Seigneur a rappelé à lui. Que Dieu lui accorde son paradis.*

*A mon unique frère Khaled à qui je souhaite la réussite au bac cette année Inshallah .*

*A ma binôme et amie Dihia SEHL*

*A tous mes amis à qui je souhaite toute la réussite .*

*A tous mes collègues de travail de l'agence de voyages air plus Tourisme que j'admire beaucoup .*

*A tous les enseignants qui ont contribué à m'inculquer leur savoir depuis mes tous débuts.*

*À toutes personnes ayant contribué de près ou de loin à la réussite de ce modeste travail, en particulier Said et Ghiles*

*Sabrina KHAZEM.*

# Liste des figures

<b>Numéro des figures</b>	<b>Titres des figures</b>	<b>Pages</b>
<b>N°I.1</b>	Réseaux personnelles sans fils (WPAN)	<b>5</b>
<b>N°I.2</b>	les réseaux locaux sans fils (WLAN)	<b>6</b>
<b>N°I.3</b>	les satellites de communications	<b>7</b>
<b>N°I.4</b>	les réseaux sans fils avec infrastructure	<b>8</b>
<b>N°I.5</b>	les réseaux ad-hoc	<b>8</b>
<b>N°II.1</b>	logo d'Android	<b>16</b>
<b>N°II.2</b>	Historique d'Android	<b>16</b>
<b>N°II.3</b>	Ecran d'accueil d'Android 1.5	<b>17</b>
<b>N°II.4</b>	Ecran d'accueil d'Android 1.6	<b>17</b>
<b>N°II.5</b>	Ecran d'accueil d'Android 2.2	<b>18</b>
<b>N°II.6</b>	Ecran d'accueil d'Android 2.3	<b>19</b>
<b>N°II.7</b>	Ecran d'accueil d'Android 3.0	<b>19</b>
<b>N°II.8</b>	Ecran d'accueil d'Android 4.0	<b>20</b>
<b>N°II.9</b>	Ecran d'accueil d'Android 4.1	<b>20</b>

<b>N°II.10</b>	Ecran d'accueil d'Android 4.4	<b>21</b>
<b>N°II.11</b>	Architecture d 'Android	<b>24</b>
<b>N°II.12</b>	les composants d'une activité	<b>26</b>
<b>N°II.13</b>	Cycle de vie d'une activité	<b>27</b>
<b>N°II.14</b>	Android SDK	<b>30</b>
<b>N°II.15</b>	L'émulateur	<b>31</b>
<b>N°II.16</b>	Bureau virtuel	<b>32</b>
<b>N°II.17</b>	Exemple de notification	<b>33</b>
<b>N°III.1</b>	Les diagrammes de modélisation	<b>39</b>
<b>N°III.2</b>	La démarche adoptée pour la modélisation.	<b>40</b>
<b>N°III.3</b>	Diagramme de cas d'utilisation globale	<b>41</b>
<b>N°III.4</b>	Diagramme de séquence de cas d'utilisation <<Ajouter un nouveau numéro>>	<b>42</b>
<b>N°III.5</b>	Diagramme de séquence de cas d'utilisation « Afficher la liste des numéros indésirable »	<b>43</b>
<b>N°III.6</b>	Diagramme de séquence de cas d'utilisation « Supprimer un numéro de la liste »	<b>44</b>
<b>N°III.7</b>	le diagramme de classe générale	<b>45</b>

<b>N°IV.1</b>	Logo Eclipse	<b>50</b>
<b>N°IV.2</b>	plate-forme Eclipse	<b>51</b>
<b>N°IV.3</b>	Portail des développeurs Android	<b>52</b>
<b>N°IV.4</b>	Interface d'installation du SDK Android	<b>53</b>
<b>N°IV.5</b>	Interface d'installation ADT	<b>54</b>
<b>N°IV.6</b>	select a wizard	<b>55</b>
<b>N°IV.7</b>	Création d'un nouveau projet Android	<b>55</b>
<b>N°IV.8</b>	Configuration du projet	<b>56</b>
<b>N°IV.9</b>	Création d'une activité	<b>57</b>
<b>N°IV.10</b>	créé une nouvelle activité en blanc avec une barre d'action	<b>58</b>
<b>N°IV.11</b>	Arborescence du projet	<b>59</b>
<b>N°IV.12</b>	L'icone de l'application	<b>60</b>
<b>N°IV.13</b>	Menu principal de l'application	<b>61</b>
<b>N°IV.14</b>	Interface d'ajout d'un numéro	<b>62</b>
<b>N°IV.15</b>	Interface d'affichage de la liste des numéros indésirables	<b>63</b>
<b>N°IV.16</b>	Interface de suppression d'un numéro	<b>64</b>

# Liste des tableaux

<b>Numéro des tableaux</b>	<b>Titre des tableaux</b>	<b>Page</b>
<b>N°II.1</b>	Fonctionnalités d'Android.	<b>23</b>
<b>N°III.1</b>	Spécification des tâches	<b>42</b>
<b>N°III.2</b>	Spécification des scénarios	<b>43</b>

# Sommaire

---

## Sommaire

### Liste des figures

### Liste des tables

Introduction Générale.....	2
----------------------------	---

## Chapitre I : Les réseaux sans fil

I.1 Introduction : .....	3
I.2 Réseaux sans fil : .....	3
I.2.1 Définition du réseau sans fil .....	3
I.2.2 Caractéristiques des réseaux sans fil : .....	4
I.2.3 Techniques de transmission dans les réseaux sans fil .....	4
I.2.4 Classification des réseaux sans fil : .....	4
I.2.4.1 Classification selon l'étendue: .....	5
I.2.4.1.1 Réseaux personnels sans fil (WPAN): .....	5
I.2.4.1.2 Les réseaux locaux sans fil (WLAN): .....	6
I.2.4.1.3 Réseaux métropolitains sans fils (WMAN) .....	7
I.2.4.1.4 Réseaux étendus sans fils (WWAN).....	7
I.2.4.2 Les satellites de communication: .....	7
I.2.4.2.1 .Classification selon le mode de communication: .....	8
I.2.5 Evolution des systèmes cellulaires : .....	10
I.2.5.1 Les réseaux de la première génération (1G): .....	10
I.2.5.2 les réseaux de la deuxième génération (2G) .....	10
I.2.4.3 Les réseaux de la troisième génération (3G) .....	12
I.2.5.4 les réseaux de la quatrième génération (4G).....	13
I.2.5.4.1 Les applications de la quatrième génération .....	13
I.2.6 les avantages des réseaux sans fil .....	14
I.3 Conclusion.....	14

## Chapitre II : Android

# Sommaire

---

II.1 Introduction:	15
II.2 Description d'Android :	15
II.3 Open Handset Alliance:	16
II.4 Bugdroïde	16
II.5 historique :	17
II.6 Fonctionnalité d'android :	22
II.7 La plateforme android:	24
II.7.1 Architecture d'Android	24
II.7.1.1 Les niveaux du système d'exploitation d'Android	25
II.7.1.1.1 Linux Kernel :	25
II.7.1.1.2 Bibliothèque et environnement	25
II.7.1.1.3 Développement de l'application (Application Framework)	26
II.7.1.1.4 Application	26
II.7.2 Les composants d'une application Android	27
II.7.2.1 L'activité (Activity)	27
II.7.2.2 Content Provider	29
II.7.2.3 Service	30
II.7.2.4 Broadcast Receiver	30
II.8 Développement d'application pour Android	30
II.8.1 Le SDK Android	30
II.8.1.1 les outils fournis par le SDK	31
II.8.1.1.1. AVD (Android Virtual Device):	31
II.8.1.1.2 L'ADB (Android Debug Bridg)	32
II.8.1.1.3. Le DDMS (Dalvik Debug Monitor Server)	32
II.9 Concepts et innovation	32
II.9.1 Le bureau virtuel	33
II.9.2 Les notifications	33
II.9.3 SQLite	34
II.9.4 Android Market	34
II.10 Présentation du langage java	34
II.10.1 Historique	34
II.10.2 Les caractéristiques de Java	35
II.10.3 Outils de développement	36

# Sommaire

---

II.11 Les avantages d'Android.....	36
II.12 Conclusion .....	38

## Chapitre III : Analyse et conception

III.1 Introduction: .....	39
III.2 Les Méthodes de conception .....	39
III.3 Présentation de l'UML:.....	39
III.3.1 Modélisation avec l'UML:.....	40
III.3.2 La démarche de modélisation avec l'UML: .....	41
III.4 Analyse : .....	41
III.4.1 Identification des acteurs .....	41
III.4.2 Identification des besoins.....	41
III.4.3 Spécification des tâches : .....	42
III.4.4 Spécification des scénarios .....	42
III.5 Conception:.....	43
III.5.1 Les diagrammes des cas d'utilisation .....	43
III.5.2 Les diagrammes de séquence.....	44
III.5.3 les diagrammes de classes.....	47
III.6 Conclusion .....	48

## Chapitre IV : Réalisation

IV.1 Introduction .....	49
IV.2 Environnement de travail .....	49
IV.2.1 Environnement matériel .....	49
IV.2.2 Environnement cible.....	49
IV.2.3 Environnement logiciel.....	49
IV.2.3.1.Système d'exploitation.....	49
IV.2.3.2 Langage de programmation .....	50
IV.2.3.3 Eclipse.....	50
IV.2.3.4 IDE Eclipse .....	50

# Sommaire

---

IV.3	Prise en main de l'environnement Android.....	51
IV.3.1	Software Development Kit (SDK) .....	52
IV.3.2	Le plugin ADT pour Eclipse .....	53
IV.3.3	Android.....	54
IV.3.3.1	Les étapes de création d'un nouveau projet.....	55
IV.4	Les différentes interfaces de notre application.....	60
IV.4.1	Lancement de l'application.....	60
IV.4.2	Le menu principal de l'application.....	61
IV.4.3	Interface d'ajout d'un numéro .....	62
IV.4.4	Interface d'affichage de la liste des numéros indésirables: .....	63
IV.4.5	Interface de suppression d'un numéro : .....	64
IV.5	Conclusion.....	64
	Conclusion générale .....	65

# **Introduction générale**

## Introduction Générale

Durant les dernières années, les développements et les progrès technologiques dans le domaine des communications mobiles se sont succédés avec un rythme très rapide , Ces progrès technologiques, en combinaison avec d'autres facteurs constitueront des éléments déterminants qui influenceront le marché mobile dans les années à venir .

Depuis quelques mois, les smartphones sont dotés d'une puissance plus importante et d'espaces de stockage conséquents. Les téléphones tendent à devenir des objets artistiques, presque de reconnaissance sociale, et possèdent des fonctionnalités qu'aucun téléphone ne pouvait espérer auparavant : connexion haut débit, localisation GPS, boussole, accéléromètre, écran tactile souvent multipoint, marché d'applications en ligne... Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité. La plate-forme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes par son ouverture ; elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme. Tous les développeurs sont ainsi sur un même pied d'égalité, qu'ils soient une grande entreprise ou quelques jeunes dans un garage ; tous peuvent ajouter des mobilités à des applications existantes.

Notre étude porte sur la réalisation d'une application android qui permet de filtrer des appels indésirables car on s'est tous retrouvé dans la situation où on ne souhaite plus répondre à une personne.

Pour cela, nous avons organisé ce mémoire comme suit :

- Le premier chapitre présente des généralités sur les réseaux sans fil.
- Le deuxième chapitre va être consacré pour l'android.
- Le troisième chapitre va illustrer l'analyse et la conception de notre application
- Le quatrième chapitre présente les outils utilisés pour la réalisation de notre application. Enfin une conclusion générale qui achèvera notre mémoire.

# **Chapitre I : Les réseaux sans fil**

## I.1 Introduction :

Aujourd'hui, on observe une nouvelle évolution : les utilisateurs veulent pouvoir rester connectés en toutes circonstances et en tous lieux. Cette évolution est rendue possible grâce à la multiplication de nouveaux réseaux sans fil.

La classification des réseaux sans fil a mis en évidence deux catégories possibles suivant deux paramètres celui de l'étendue et du mode de communication. On ce qui concerne la première catégorie elle regroupe le Bluetooth ainsi que le WIFI par contre la deuxième catégorie, elle regroupe les modes de communication avec infrastructure et sans infrastructure.

L'évolution des réseaux sans fil ne s'arrête pas à l'apparition du Wifi ou du Bluetooth. Cette évolution a atteints également les réseaux des systèmes cellulaires.

## I.2 Réseaux sans fil : [1]

La radio, la télévision, plus récemment la téléphonie, ont eu un déploiement dont le succès a tenu essentiellement à la maîtrise de techniques de transport de l'information sans fil. Toutes les autres techniques de transport utilisant un support conducteur électrique ou optique se sont trouvées un jour ou l'autre freinées par des problèmes techniques. Ces problèmes de travaux publics, ou de bâtiment font que les techniques utilisant le câble sont hautement discriminatoires dans la possibilité d'accès à l'information. Un câble n'amènera jamais le réseau à un coût uniforme que ce soit dans une entreprise, dans un pays ou dans le monde. L'explosion de la téléphonie mobile dans des pays où le téléphone filaire était très en retard est une démonstration flagrante de cette réalité.

### I.2.1 Définition du réseau sans fil

Un réseau est un ensemble des noeuds reliés entre eux par des liens ou canaux de communication dans le but d'échanger des informations.

Selon le type des noeuds, on distingue : les réseaux de télécommunication dans lesquels les noeuds sont les stations mobiles, les stations de base (BTS), les contrôleurs des stations de base (BSC), les commutateurs (MSC) ; et les réseaux informatiques dans lesquels les noeuds sont les ordinateurs, les imprimantes, les routeurs, les switchers ou tout autre équipement informatique.

Les réseaux sans fils permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres. De plus l'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes comme c'est le cas avec les réseaux filaires. En contrepartie se pose le problème de la réglementation relatives aux transmissions radio-électriques. De plus les ondes hertziennes sont difficiles à confiner dans une surface géographique restreinte, il est donc facile pour un pirate d'écouter le réseau si les informations circulent en clair. Il est donc nécessaire de mettre en place les dispositions nécessaires de telle manière à assurer une confidentialité des données circulant sur les réseaux sans fils.

## **I.2.2 Caractéristiques des réseaux sans fil :**

La principale caractéristique des réseaux sans fil est qu'ils sont libres de tout câblage ou autre matériel encombrant. Voici un ensemble de caractéristiques propres aux réseaux sans fil :

- Débit : suivant la technologie ainsi que le type du média utilisé, le débit peut constituer un grand inconvénient, par exemple : les réseaux de capteurs fonctionnent avec un débit de quelques Kbps tandis que le WIFI peut atteindre 100 Mbps.
- Fiabilité : les signaux transmis via les ondes-radio subissent des perturbations (erreurs de transmission, coupures...) dues à l'environnement.
- Sécurité : la diffusion des messages dans l'air rend les réseaux sans fil sujets à écoute (piratage des messages).
- Energie : en cas de mobilité ou de fonctionnement autonome, les réseaux sans fil sont confrontés au problème d'énergie, une gestion des ressources énergétiques est nécessaire pour augmenter la durée de vie d'un réseau.
- Mobilité : la mobilité est l'un des atouts majeurs des réseaux sans fil.

## **I.2.3 Techniques de transmission dans les réseaux sans fil [2]**

il existe principalement deux méthodes pour la transmission dans les réseaux sans fil:

### ➤ **Transmission par les ondes infrarouges**

la transmission par les ondes infrarouges nécessite que les appareils soient en face l'un des autres et aucun obstacle ne sépare l'émetteur du récepteur.(car la transmission est directionnelle).cette technique est utilisée pour créer des petits réseaux de quelques dizaines de mètres. (télécommande de : télévision, les jouets, voitures...).

### ➤ **Transmission par les ondes radios.**

La transmission par les ondes radios est utilisée pour la création des réseaux sans fil qui a plusieurs kilos mètres. Les ondes radios ont l'avantages de ne pas êtres arrêtés par les obstacles car sont émises d'une manière omnidirectionnelle. Le problème de cette technique est perturbations extérieurs qui peuvent affecter la communication à cause de l'utilisation de la même fréquence par exemple.

## **I.2.4 Classification des réseaux sans fil :**

Le choix de classement des réseaux sans fil est fait suivant des paramètres tels que l'étendue (WPAN, WIFI, WIMAX, GSM, GPRS, UMTS) ou bien le mode de communication (avec ou sans infrastructure).

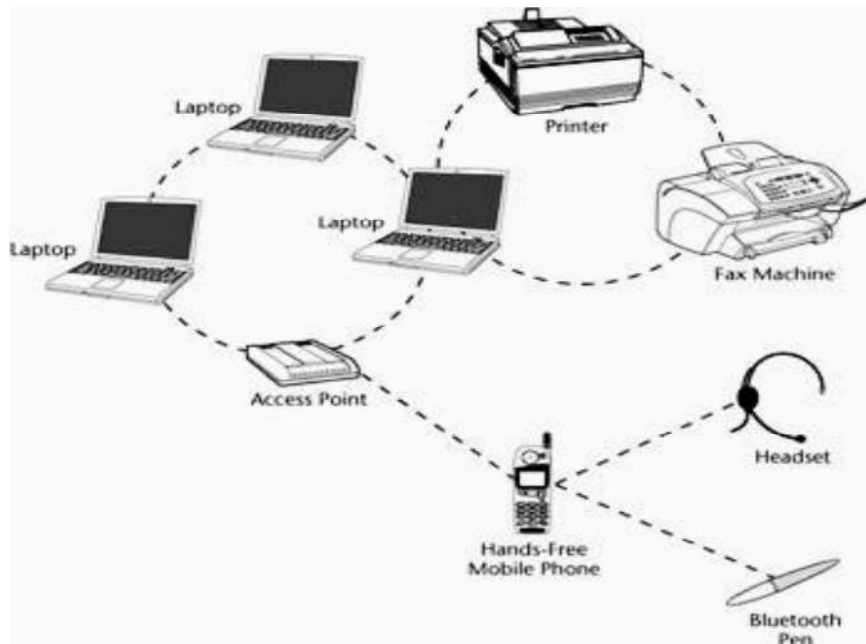
### **I.2.4.1 Classification selon l'étendue:**

L'étendue ou encore la portée est la surface géographique couverte par un réseau sans fil, cette surface varie des réseaux personnels WPAN à des réseaux plus grands tels que le WIMAX ou le GSM.

### I.2.4.1.1 Réseaux personnels sans fil (WPAN):

Le réseau personnel sans fils (appelé également réseau individuel sans fils ou réseau domotique sans fils et noté WPAN pour Wireless Personal Area Network) concerne les réseaux sans fils d'une faible portée : de l'ordre de quelques dizaines mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fils entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPAN :

- La principale technologie WPAN est la technologie **Bluetooth**, lancée par Ericsson en 1994, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Bluetooth, connue aussi sous le nom IEEE 802.15.1, possède l'avantage d'être très peu gourmand en énergie, ce qui le rend particulièrement adapté à une utilisation au sein de petits périphériques. La version 1.2 réduit notamment les interférences avec les réseaux Wi-Fi.
- **HomeRF** (Home Radio Frequency), lancée en 1998 par le HomeRF Working Group (formé notamment par les constructeurs Compaq, HP, Intel, Siemens, Motorola et Microsoft) propose un débit théorique de 10 Mbps avec une portée d'environ 50 à 100 mètres sans amplificateur. La norme HomeRF soutenue notamment par Intel, a été abandonnée en Janvier 2003, notamment car les fondeurs de processeurs misent désormais sur les technologies Wi-Fi embarquée (via la technologie Centrino, embarquant au sein d'un même composant un microprocesseur et un adaptateur Wi Fi).
- La technologie **ZigBee** (aussi connue sous le nom IEEE 802.15.4) permet d'obtenir des liaisons sans fil à très bas prix et avec une très faible consommation d'énergie, ce qui la rend particulièrement adaptée pour être directement intégré dans de petits appareils électroniques (appareils électroménagers, hifi, jouets, ...).
- Enfin les liaisons **infrarouges** permettent de créer des liaisons sans fils de quelques mètres avec des débits pouvant monter à quelques mégabits par seconde. Cette technologie est largement utilisé pour la domotique (télécommandes) mais souffre toutefois des perturbations dues aux interférences lumineuses. L'association irDA (infrared data association) formée en 1995 regroupe plus de 150 membres.

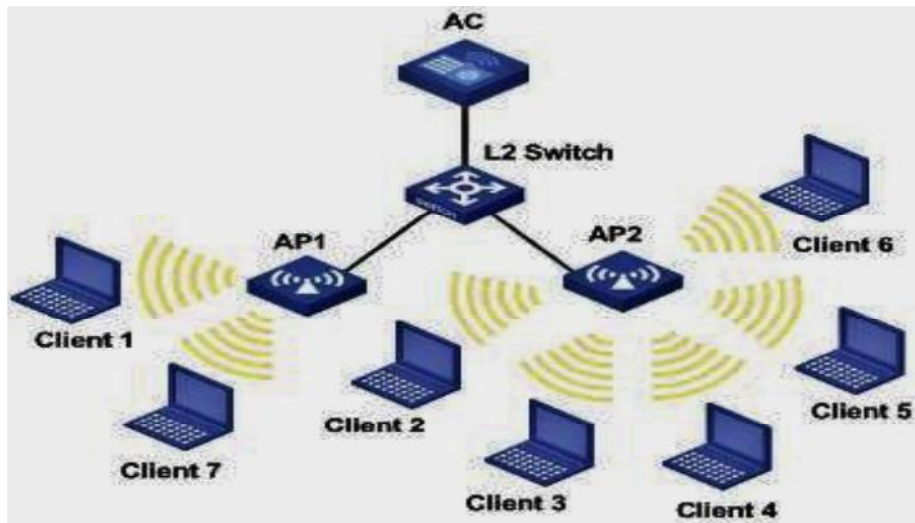


**Figure I.1** : Réseaux personnelles sans fils (WPAN)

### I.2.4.1.2 Les réseaux locaux sans fil (WLAN):

Le réseau local sans fil (WLAN pour Wireless Local Area Network) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre-eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes :

- Le **WiFi** (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.
- **hiperLAN2** (High Performance Radio LAN 2.0), norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute), permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz.
- **DECT** (Digital Enhanced Cordless Telecommunication), norme des téléphones sans fils domestiques. Alcatel et Ascom développent pour les environnements industriels, telles les centrales nucléaires, une solution basée sur cette norme qui limite les interférences. Les points d'accès résistent à la poussière et à l'eau. Ils peuvent surveiller les systèmes de sécurité 24/24h et se connecter directement au réseau téléphonique pour avertir le responsable en cas de problème.



**Figure I.2** : les réseaux locaux sans fils (WLAN)

### I.2.4.1.3 Réseaux métropolitains sans fils (WMAN)

Le réseau métropolitain sans fils (WMAN pour Wireless Metropolitan Area Network) est connu sous le nom de Boucle Locale Radio (BLR). Les WMAN sont basés sur la norme IEEE 802.16. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication.

### I.2.4.1.4 Réseaux étendus sans fils (WWAN)

Le réseau étendu sans fils (WWAN pour Wireless Wide Area Network) est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fils les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fils. Les principales technologies sont les suivantes :

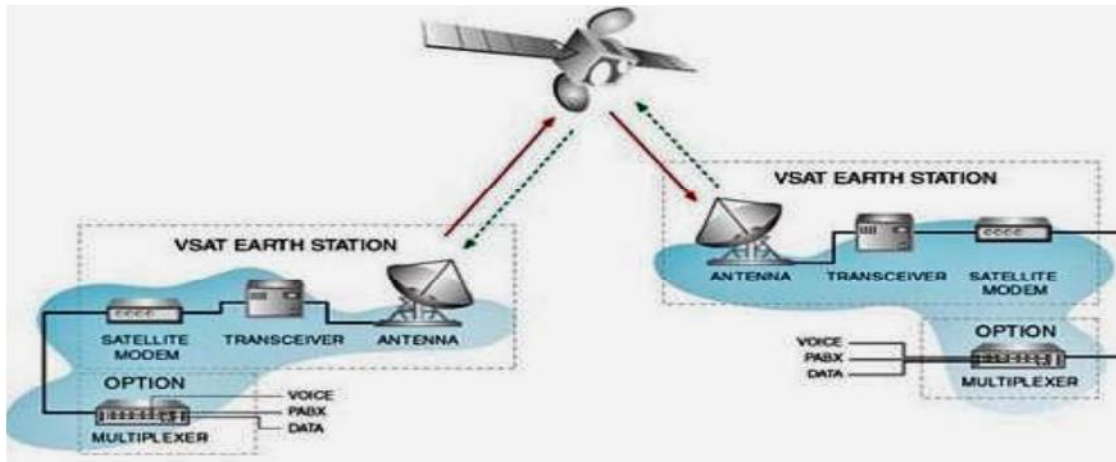
- **GSM** (Global System for Mobile Communication ou Groupe Spécial Mobile)
- **GPRS** (General Packet Radio Service)
- **UMTS** (Universal Mobile Telecommunication System)
- **Wimax** (standard de réseau sans fils poussé par Intel avec Nokia, Fujitsu et Prowim). Basé sur une bande de fréquence de 2 à 11 GHz, offrant un débit maximum de 70 Mbits/s sur 50km de portée, certains le placent en concurrent de l'UMTS, même si ce dernier est davantage destiné aux utilisateurs itinérants.

### I.2.4.2 Les satellites de communication:

Les réseaux satellitaires sont constitués d'une constellation de petits satellites (une cinquantaine) positionnés dans l'espace pour des besoins de télécommunication et permettent de couvrir n'importe quelle zone géographique grâce à leurs positions lointaines par rapport à la terre. Ils sont en soi des dispositifs d'émission, et ont été utilisés pour la téléphonie mobile qui se caractérise par la transmission des données à bas débit.

L'inconvénient majeur des satellites est qu'ils ont tout à fait une grande propagation retardée en raison des distances parcourues par les ondes radio.

On envisage prochainement l'augmentation du débit afin d'effectuer une transmission intégrantes des applications multimédias.



**Figure I.3 : les satellites de communications**

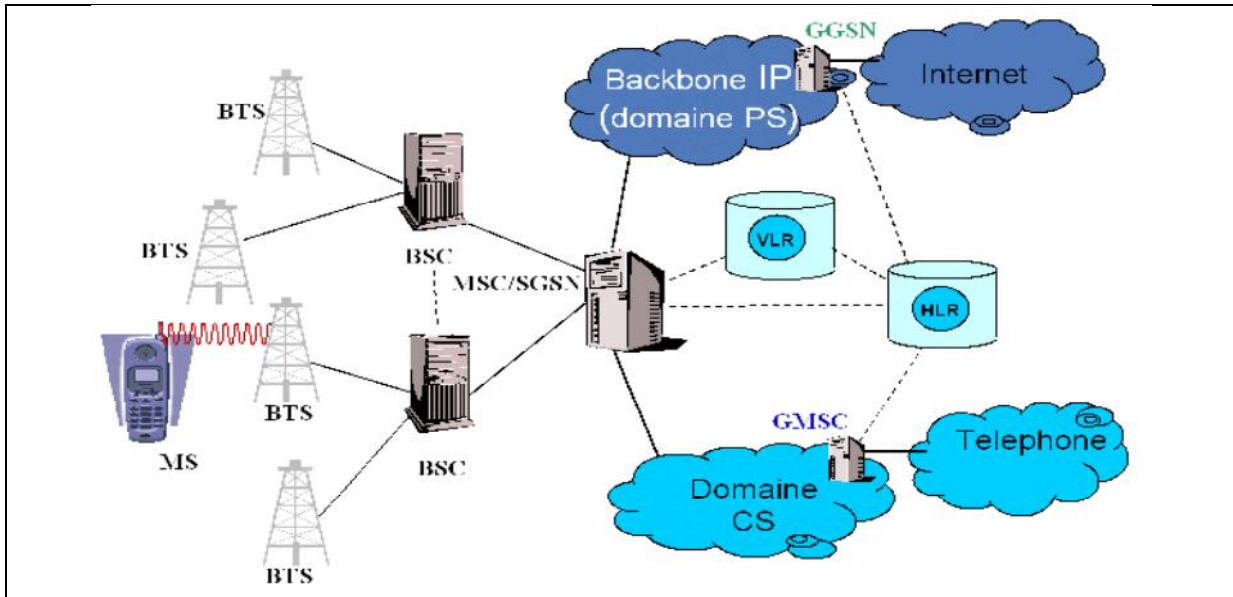
### **I.2.4.2.1 .Classification selon le mode de communication:[3]**

#### **I.2.4.2.1.1 Mode de communication avec infrastructure:**

Dans ce genre de réseaux, la communication entre les différents éléments est faite à travers une station de base ayant des points d'accès, ces derniers servent à connecter les différentes unités mobiles pour constituer le réseau.

Tout message doit impérativement passer par une station de base avant d'atteindre sa destination finale.

Ce type de réseaux est le plus utilisé, principalement dans les réseaux cellulaires de téléphonie mobiles.



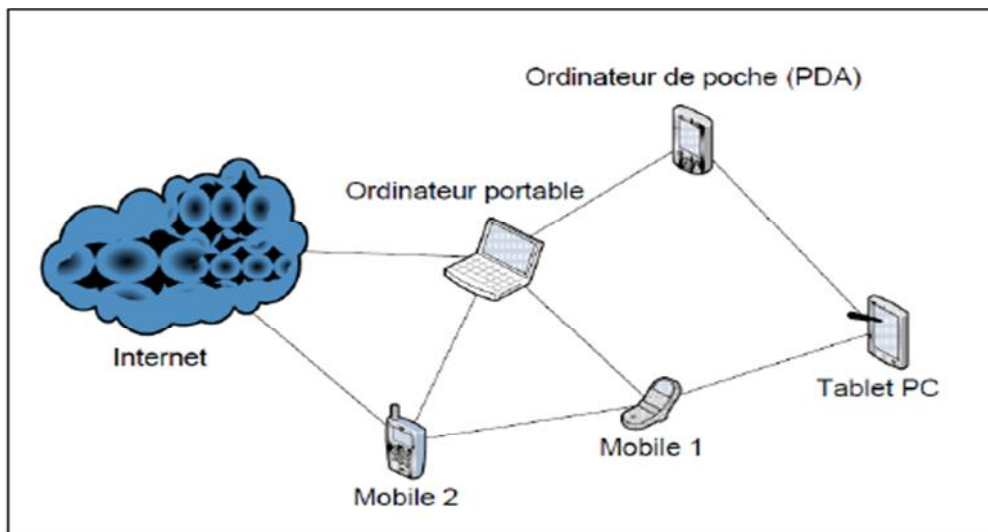
**Figure I.4 :** les réseaux sans fils avec infrastructure

### I.2.4.2.1.2 Mode de communication sans infrastructure ( ad-Hoc)

Dans les réseaux sans infrastructure, les membres du réseau font deux fonctions différentes, ils servent en même temps de client et de point d'accès.

Une communication entre deux éléments du réseau peut passer par un ou plusieurs autres éléments suivant la topologie du réseau.

Ce type de réseaux est adapté le plus aux réseaux ad hoc et autres communications instantanées (par exemple : réseau d'ordinateurs portables...)



**Figure I.5 :** les réseaux ad-hoc

## **I.2.5 Evolution des systèmes cellulaires [4]**

Le terme génération sert à désigner les améliorations incrémentales survenues au cours de l'évolution des systèmes cellulaires. Dans cette section, nous allons présenter brièvement les différentes générations de systèmes cellulaires et leur architecture générale.

### **I.2.5.1 Les réseaux de la première génération (1G)**

Dits aussi analogique. Ce terme caractérise la méthode utilisée pour transporter de l'information sur ces systèmes. On retrouve dans ce cas deux grandes catégories : le sans fil et le cellulaire, mais on se focalisera beaucoup plus sur le second (cellulaire).

Ces réseaux sont introduits dans les années 80 en Amérique du Nord et aux pays Nordiques pour la transmission de la voix (téléphonie mobile), et malgré leur analogie, ils utilisent une signalisation numérique dans plusieurs aspects de leurs réseaux y compris l'interface air. Cette analogie là ne permet pas de sécuriser les communications qui ne se sont pas cryptées, et ce qui caractérise ces réseaux c'est bien la transmission de donnée par modulation de fréquence, et celle de la radiophonie limitée en terme de zone géographique et nombre d'abonnées.

Les terminaux de la 1G étaient lourds et économisaient beaucoup d'énergie à cause de leurs grands besoins d'émettre pour de longues distances. Le canal était parasité, la restitution du signal était faible, et de nombreuses interférences s'imposaient engendrant des perturbations dans la qualité des émissions. Et vu la mobilité des terminaux qui était restreinte, les réseaux cellulaires ont été conçus dans le but de l'augmenter. Prenons le cas du réseau le plus répandu aux USA : l'AMPS (Advanced Mobile Phone Service) qui représente le premier système cellulaire opérationnel introduit en 1983.

Selon les caractéristiques, le partage de ressources se fait par un multiplexage FDMA et l'allocation des fréquences et comprise entre 800 et 900 Mhz. Son identification s'effectue par le numéro d'abonné grâce au (préfixe régional + numéro d'abonné).

On retrouve également le TACS (Total Acces Communication System), la version Européenne de l'AMPS qui appartient toujours à la même catégorie (cellulaire), et l'ETACS (Extended Total Acces Communication System) qui est la version améliorée et contient un nombre plus important de canaux de communication.

D'après ce passage, on peut bien constater que les réseaux de la 1G présentent beaucoup de points faibles y compris la faible capacité en termes d'utilisateurs et les contraintes de la ressource énergétique des terminaux. Néanmoins, on a surmonté certaines par des ajouts technologiques en introduisant les réseaux cellulaires de la 2G.

### **I.2.5.2 les réseaux de la deuxième génération (2G)**

La deuxième génération (2G) de systèmes cellulaires repose sur une technologie numérique a été développée à la fin des années 1980. Ces systèmes cellulaires utilisent une technologie numérique pour la liaison ainsi que pour le signal vocal. Ce système apporte une meilleure qualité ainsi qu'une plus grande capacité à moindre coût pour l'utilisateur.

En raison de la faible qualité du CT1, plusieurs constructeurs européens ont développé en 1984, le système CT2, téléphone sans fil de deuxième génération. L'objectif est de fournir une

norme qui permet aux utilisateurs de changer de réseau et d'opérateur tout en gardant le même terminal.

Ce système ne fournit pas de couverture nationale mais est installé à des endroits où le trafic d'appel est important, c'est-à-dire les gares, les aéroports, les centres commerciaux et les restaurants.

Le PHS (Personal Handyphone System) a été développé au Japon sur de bases légèrement différentes que le système européen, c'est un système sans fil offrant un accès public, aussi bien en usage domestique que professionnel à la fois à l'intérieur et à l'extérieur avec un terminal de poche à très faible coût. Il a été introduit commercialement en 1995 au Japon comme un système d'accès public. Des fonctionnalités lui ont été rajoutées comme la mise en place de boucles locales sans fil, la transmission de données à des débits de l'ordre de 32 ou 64 Kbit/s à l'itinérance.

Au début des années 1990 plusieurs technologies numériques cellulaires ont fait leur apparition :

Le premier grand standard est le GSM (Global System for Mobile Communication) c'est le standard le plus utilisé en Europe et aux Etats-Unis. Les bandes de fréquences utilisées par le GSM varient en fonction des pays. En Europe le GSM utilise les bandes de fréquence 900 MHz et 1800 MHz alors qu'aux USA il utilise la bande de fréquence 1900 MHz. Les GSM ne sont plus identifiés via l'appareil comme pour la première génération mais via la carte SIM (Subscriber Identity Module). Cette carte permet d'identifier l'utilisateur et est indépendante du terminal. Cette norme permet un débit maximal de 9,6 kbps/s. son réseau spécifique s'appelle PLMN (Public Land Mobile Network), chaque opérateur ayant le sien propre. Il est relié au Réseau Téléphonique Commuté Public (RTPC).

Des extensions à la norme GSM ont été mises au point, c'est le cas du GPRS (General Packet Radio System) baptisée 2,5G cette norme offre des débits supérieurs au GSM, qui sont maintenant de l'ordre de 114 kbit/s. Une autre évolution du GSM a été observée, c'est le standard EDGE (Enhanced Data Rates for GSM Evolution), il modifie le type de modulation, cette norme de transition n'est pas encore de la 3G mais porte le nom de 2,75G. Elle permet d'offrir des débits allant jusqu'à 473 kbits/s mais offrant des couvertures plus réduites. Ce débit a été limité pour répondre aux spécifications IMT2000.

La nouveauté par rapport à la première génération est la séparation entre le terminal et l'utilisateur, en effet l'utilisateur pourra utiliser différents terminaux tout en conservant les mêmes services (numéros de téléphones et carnet d'adresses). L'apparition de la carte SIM permet la mise en oeuvre de systèmes de sécurité (authentification au niveau du réseau, protection des communications, cryptage).

Il est maintenant possible de transmettre également des données numériques (SMS : Short Message Service) Ou des messages multimédias (MMS). Les systèmes cellulaires numériques de deuxième génération ont favorisé l'arrivée d'un terminal portable avec une autonomie acceptable. Cependant ces terminaux n'offrent pas une mobilité globale et il existe un manque de compatibilité entre ces différents systèmes. Ce qui a conduit au développement de la troisième génération des réseaux de mobiles (**3G**).

### I.2.4.3 Les réseaux de la troisième génération (3G)

L'UIT travaille sur la troisième génération des réseaux mobiles depuis 1985, d'abord sous le nom de FPLMTS (Future Public Land Mobile Télécommunication System) puis sous le nom d'IMT 2000 (International Mobile Telecommunications System for the year 2000). La standardisation a commencé en 1990 en Europe à l'ETSI. Pour l'UMTS aux Etats-Unis, la principale proposition s'appelle IS-9.

La troisième génération (3G) de systèmes cellulaires est une génération de systèmes mobiles labellisé IMT 2000 par l'UIT. Ce système permet des services de communications plus rapides notamment pour la voix, la télécopie, l'Internet de n'importe quel endroit et à tout moment. L'UIT IMT-2000 est la norme internationale de la 3G a ouvert la voie à de nouvelles applications et services comme par exemple le divertissement multimédia, la localisation des services, Visiophonie, MMS vidéo, Vidéo à la demande, Télévision.

Les propriétés générales de la 3G sont les suivantes :

- Couverture totale et mobilité complète jusqu'à 144 Kbps voire 384 Kbps/s.
- Couverture plus limitée et mobilité jusqu'à 2 Mbps/s.
- Grande flexibilité pour introduire des nouveaux services.

La recommandation IMT 2000 n'a malheureusement pas fait l'unanimité et de nombreux organismes de standardisation ont préféré développer leur projet standard tout en gardant une certaine compatibilité avec la norme de base. Elle est ainsi devenue le nom générique pour toute une famille de standards. Ce sont les Japonais qui ont démarré le processus en normalisant une version CDMA. Le W-CDMA (Wideband-Code Multiple Access) suivis des Européens et des Américains. L'UMTS proposé par les Européens porte sur un découpage en fréquences puis sur chaque fréquence par un TDMA et enfin, sur chaque slot par un CDMA. Cette méthode porte le nom de TD-CDMA (Time Division-CDMA). Un rapprochement s'est effectué entre Européens et Japonais pour aboutir à la norme finale de l'UMTS qui a été normalisée par l'ETSI. L'interface air normalisé comprend le W-CDMA pour les bandes de fréquences TDD (Time Domain Duplex), les bandes UMTS non appariées correspondant au passage des signaux sur une fréquence par sens de transmission. Le choix s'est porté sur la méthode TD-CDMA. Aux Etats Unis, le comité TR45.5 du TIA a proposé le standard CDMA2000 qui est une extension de l'IS-95. La même année 1998, le comité TR45.3 responsable de la normalisation IS-136 proposait une autre norme fondée sur l'accès en division dans le temps sous le nom d'UWC-136.

Plusieurs ensembles de fonctionnalités ont été définis, Les fonctions retenues pour la première génération de 2000 à 2005 sont les suivantes :

- Mobilité du terminal,
- Mobilité personnelle,
- Environnement personnel virtuel (VHE),
- Fonctions permettant de recevoir et d'émettre des applications multimédias : 144 Kbps/s à mobilité forte et 384 Kbps/s à faible mobilité,
- Handover avec les systèmes de deuxième génération,
- Réseau fixe devant suivre les technologies paquet et circuit,
- Interconnexion des réseaux RNIS, X25 et Internet.

La deuxième génération qui à été nommé 3G++ à partir de 2005 jusqu'à 2010 devrait fournir les fonctionnalités suivantes :

- Débits pouvant atteindre en pointe 2 Mbps/s par utilisateur pour un réseau très peu chargé,
- Handovers entre les membres de la famille IMT 2000,
- Passage à un réseau coeur de type IP contrôlé par un système de gestion par politique.

A partir de 2010 la troisième génération (3G) c'est faite remplacer par la quatrième génération (4G).

### **I.2.5.4 les réseaux de la quatrième génération (4G)**

La quatrième génération de réseaux mobiles fait appel aux techniques dites ITS (Intelligent Transportation Systems), qui permettent d'associer différentes technologies en un même ensemble. A l'intérieur des bâtiments, ce sont principalement les systèmes sans fil de type WPAN (Wireless Personal Area Network) ou WLAN (Wireless Local Area Network) qui est utilisés par le terminal pour se connecter vers l'extérieur. Dans les environnements moins fortement urbanisés, les techniques GPRS ou UMTS peuvent prendre le relais, tandis que dans les milieux peu peuplés, le satellite se substitue aux autres techniques pour maintenir la connexion avec les réseaux d'opérateurs. Les systèmes 4G apportent une nouvelle augmentation des débits mais sans changer fondamentalement les vitesses à partir de chaque terminal. L'apport principal de cette génération vient de la qualité de service disponible à partir du terminal utilisateur. Cette qualité de service, qui a pour origine l'adoption de la norme IEEE 802.11e, autorise l'apparition de la téléphonie et de la visioconférence sur ces réseaux. Pratiquement tous les types de portables de cette génération étant censés être dotés d'une carte IEEE 802.11e, c'est la généralisation de l'environnement IP qui devient la caractéristique fondamentale des réseaux. Le terminal de quatrième génération communique avec les serveurs de politique des opérateurs, ou NSP (Network Services Provider), et détermine à tout instant le meilleur réseau accessible. Pour cela, les terminaux mobiles doivent être dotés de l'ensemble des coupleurs et cartes de communication nécessaires pour se connecter aux différents réseaux. Des handovers hétérogènes permettent de passer d'un NSP à un autre sans couper la communication.

#### **I.2.5.4.1 Les applications de la quatrième génération**

La vidéo MPEG-12 Virtual Home Environnement ou application d'environnement de domicile virtuel. Le VHE permet de se connecter n'importe où dans le monde et sur n'importe quel type de terminal, tout en retrouvant son environnement de travail personnel, y compris ses paramètres de sécurité.

L'internet ambiant, un utilisateur quelconque doit pouvoir s'arrêter sur un banc public et consulter à l'aide de son portable, de son terminal ou de son PDA les bases de données Internet. Pour cela, une adresse IP lui est affectée automatiquement par une technique DHCP.

### I.2.6 les avantages des réseaux sans fil [5]

- **Abordabilité.** Les frais de câblage étant réduits, voire éliminés, un réseau sans fil coûte moins cher à installer, exploiter et étendre qu'un réseau câblé.
- **Commodité d'accès.** Vous pouvez accéder aux ressources réseau à partir de n'importe quel emplacement situé dans la zone de couverture du réseau, tel qu'une salle de conférence, ou à partir d'une zone d'accès sans fil publique.
- **Simplicité d'installation et d'expansion.** Avec un réseau sans fil, nul besoin de faire courir des câbles pour raccorder les ordinateurs, imprimantes et autres périphériques entre eux ou à Internet. L'ajout de nouveaux utilisateurs au réseau est également très simple.
- **Sécurité.** Les progrès constants en matière de normes et de protocoles rendent dans la plupart des cas les réseaux sans fil aussi sûrs que les réseaux câblés. Un réseau sans fil offre des fonctionnalités de sécurité robustes telles que le chiffrement des données (afin de protéger les informations qui transitent sur le réseau), l'authentification des utilisateurs (qui identifie les ordinateurs qui tentent de se connecter au réseau) et l'accès sécurisé pour les visiteurs et invités.

### Conclusion

Dans ce chapitre nous avons vu l'évolution et la multiplication des réseaux sans fils et des réseaux des systèmes cellulaires. Cette évolution a engendré et a permis l'apparition d'une nouvelle technologie qu'on appelle « les applications intelligentes » qui s'appuie sur différents systèmes d'exploitation tel que « Android ».

Dans le chapitre suivant nous parlerons d'Android et ses caractéristiques.

# **Chapitre II : Android**

### II.1 Introduction:

Android est une excellente opportunité pour appréhender le développement d'applications mobiles ambitieuses.

Rappelons les points clés d'Android en tant que plate-forme :

- ❖ elle est innovante car toutes les dernières technologies de téléphonie y sont intégrées : écran tactile, accéléromètre, GPS, appareil photo numérique etc. ;
- ❖ elle est accessible car en tant que développeur vous n'avez pas à acheter de matériel spécifique (si vous voulez aller plus loin que l'utilisation d'un émulateur, un téléphone Android pour effectuer vos tests vous sera toutefois nécessaire), ni à connaître un langage peu utilisé ou spécifique : le développement sur la plate-forme Android est en effet réalisé en langage Java, un des langages de programmation les plus répandus ;
- ❖ elle est ouverte parce que la plate-forme Android est fournie sous licence open source, permettant à tous les développeurs et constructeurs de consulter les sources et d'effectuer les modifications qu'ils souhaitent.

Ce chapitre est une introduction à la plate-forme, aux outils et à la configuration de notre environnement de travail.

### II.2 Description d'Android [6]

Android est un système d'exploitation pour Smartphones et tablettes tactiles conçu par Android. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radio-réveils ou des autoradios et même des voitures. Il a été développé par une petite startup qui fut achetée en 2007 par Google qui poursuit activement son développement avec l'Open Handset Alliance, Android est distribué sous licence open source depuis 2008. Ce système est assez nouveau auprès des programmeurs.

Selon Google qui est le majeur distributeur, Android est une plateforme puissante, moderne, sécurisée et ouverte. Elle est basée sur le kernel Linux 2.6 (noyau Linux 2.6) et utilisant la plateforme Java pour ses applications. Elle est entièrement gratuite et sa plateforme très flexible ce qui permet aux développeurs d'intégrer, d'agrandir et de remplacer les composants existants et d'adapter les applications aux besoins du client ou les remplacer entièrement, l'utilisateur peut donc personnaliser facilement son appareil.

De plus, il n'y a pas de distinction entre les applications natives et les applications qui sont développées par les développeurs, toutes sont disponibles sur l'**Android Market** (maintenant appelé **Google Play Store**).

En termes d'applications, Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps.

### II.3 Open Handset Alliance:

L'Open Handset Alliance (abrégé OHA) est un consortium regroupant de grands constructeurs et développeurs de logiciels dont le but est de développer des normes ouvertes pour les appareils de téléphonie mobile.

Le consortium a été créé le 5 novembre 2007 à l'initiative de Google qui a su fédérer autour de lui 34 compagnies. On note ainsi la présence d'opérateurs comme NTT DoCoMo, Sprint Nextel, Telecom Italia ou Bouygues Telecom. Des équipementiers tels que Samsung Electronics, Motorola Mobility ou LG Electronics. Des semi-conducteurs comme Intel ou Nvidia. Et enfin des incontournables de l'Internet dont eBay. On remarquera quelques grands absents dont Nokia.

### II.4 Bugdroïde

Le personnage nommé Bugdroid est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous licence « creative commons by (3.0) » et peut donc être utilisé librement.

Le site Engadget annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari : Gauntlet: The Third Encounter.



**Figure II.1** : logo d'Android

### II.5 historique : [7]



**Figure II.2** : Historique d'Android

Android a débuté avec la sortie de la version 1.0 « Apple pie » en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités.

Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces noms de codes suivent une logique alphabétique, en voici quelques-unes :

**Android 1.5 « Cupcake »** a apporté de nombreuses améliorations. La première grande étape fut une mise à jour du noyau vers la version 2.6.27 qui a rendu le système plus stable et corrigea le manque d'API et rend le système plus utilisable.



**Figure II.3 :** Ecran d'accueil d'Android 1.5

**Android 1.6 « Donut »**, **Android 2.0** et **2.1 « Eclair »** ont apporté d'importantes améliorations respectivement sur les fonctionnalités et sur l'interface graphique du système.



**Figure II.4 :** Ecran d'accueil d'Android 1.6

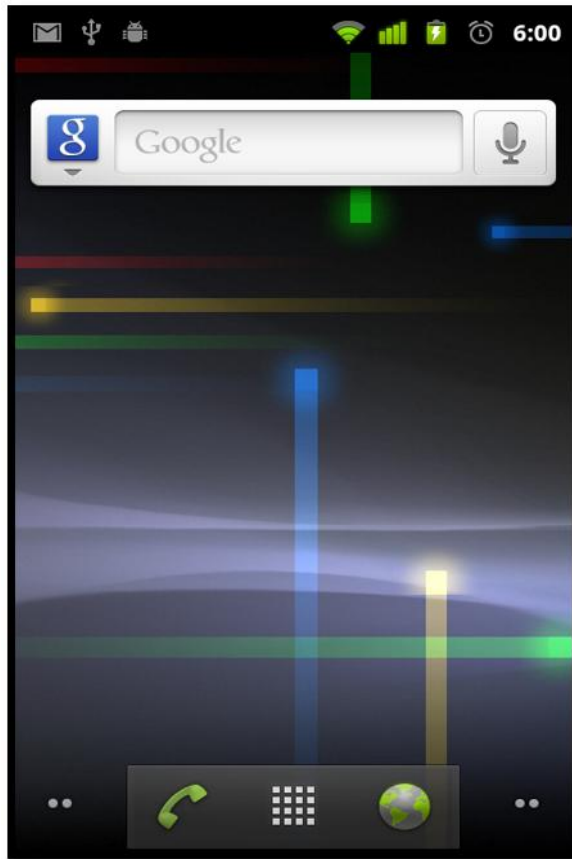
**Android 2.2 « Froyo »** a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. Aussi, Google annonce-t-elle que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8.



**Figure II.5 :** Ecran d'accueil d'Android 2.2

**Android 2.3 « Ginerbread »** Dernière version dédiée uniquement aux smartphones. Cette version est parfois utilisée sur de petites tablettes.

Gingerbread était un système très réussi, qui devint à l'époque le système mobile le plus populaire.



**Figure II.6 :** Ecran d'accueil d'Andoird 2.3

**Android 3.0 « Honeycomb »** est spécialement étudié pour les tablettes tactiles. Les premiers modèles devraient être annoncés au CES 2011.

On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont sûrement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.



**Figure II.7 :** Ecran d'accueil d'Andoird 3.0

**Android 4.0 « Ice Cream Sandwich »** devrait arriver très vite (mi 2011) pour rajouter encore plus de fonctionnalités aux terminaux. Pour le développement, ces nouvelles versions d'Android devraient proposer de nouveaux composants permettant de réaliser des applications avec une ergonomie plus adaptée aux tablettes tactiles.



**Figure II.8:** Ecran d'accueil d'Android 4.0

**Android 3.0** et **Android 4.0** devraient apporter plus d'outils aux constructeurs leur permettant de proposer des tablettes tactiles, qui seront capables de rivaliser (surtout au niveau de l'ergonomie) avec Ipad.

**Android 4.1 « Jelly Bean »** Il ajoute un système de notification améliorée, la reconnaissance vocale sans connexion internet, et le « Project Butter » qui augmente la fluidité d'Android.



**Figure II.9 :** Ecran d'accueil d'Android 4.1

**Android 4.4 « KitKat »** Consommation en ressource moins élevée nécessitant moins de RAM, nouvelles icônes plus soignées, la barre du bas et celle de statut deviennent transparentes sur certains menus et changent de couleur en fonction du contenu affiché.



**Figure II.10 :** Ecran d'accueil d'Android 4.4

### II.6 Fonctionnalité d'android : [8]

Android a été conçu pour intégrer au mieux les applications existantes de Google, comme le service de courrier Gmail, l'agenda Google Calendar ou encore la cartographie Google Maps.

Voici quelques fonctionnalités proposées par Android classées par version :

Version	Nom	API Level	Fonctionnalités
Android 1.0	Apple Pie (tarte aux pommes)	1	Fonctionnalités de base (téléphonie, SMS, réseaux de données, accès au répertoire, ...etc)
Android 1.1	Banana Split	2	<ul style="list-style-type: none"> <li>• corrections de bogues (alarme, Gmail...)</li> <li>• Amélioration des fonctionnalités MMS</li> <li>• Introduction des applications payantes sur l'Android Market</li> </ul>
Android 1.5	Cupcake (petit gâteau)	3	<ul style="list-style-type: none"> <li>• Nouveau clavier avec auto complétion</li> <li>• Support Bluetooth</li> <li>• Enregistrement de vidéos</li> <li>• Widgets</li> <li>• Animations améliorées</li> </ul>
Android 1.6	Donut (Beignet)	4	<ul style="list-style-type: none"> <li>• Recherche vocale améliorée</li> <li>• Indicateur d'utilisation de la batterie</li> <li>• Apparition de fonctionnalités pour les réseaux privés virtuels (VPN)</li> </ul>
Android 2.0 ; 2.0.1 ; 2.1	Eclair (sorte de gâteau à la crème)	5 ; 6 ; 7	<ul style="list-style-type: none"> <li>• Interface graphique améliorée</li> <li>• Gestion des contacts changée</li> <li>• Support d'HTML 5</li> <li>• Support Bluetooth 2.1</li> <li>• Clavier virtuel amélioré</li> <li>• Refonte de l'installation et des mises à jour du kit de développement</li> <li>• Réduction de la consommation de la batterie</li> <li>• Quelques améliorations graphiques comme l'écran de verrouillage</li> <li>• Nouveau bureau à 5 écrans virtuels</li> </ul>

			<ul style="list-style-type: none"> <li>• Fonds d'écran animés</li> <li>• Nouveaux effets 3D (notamment sur la gestion des photos)</li> <li>• Amélioration du client Gmail</li> </ul>
Android 2.2	Froyo (yaourt glacé)	8	<ul style="list-style-type: none"> <li>• Nouveaux Widgets</li> <li>• Gestion des points d'accès au réseau</li> <li>• Support multilingue</li> </ul>
Android 2.3	Gingerbread (pain d'épice)	9	<ul style="list-style-type: none"> <li>• Amélioration d'interface graphique</li> <li>• Téléphonie IP</li> <li>• Nouveau clavier virtuel</li> </ul>
Android 3.0	Honeycomb (rayon de miel)	11	<ul style="list-style-type: none"> <li>• Optimisé pour les tablettes et équipement à écran large</li> <li>• Multitâche et système de notification amélioré</li> </ul>
Android 4.0	Ice cream sandwich (Sandwich à la crème glacée)	12	<ul style="list-style-type: none"> <li>• Nouvelle interface graphique</li> <li>• Amélioration de la sécurité</li> <li>• Beaucoup de raccourcis (Appareil photo, accès sdcard, ... etc)</li> </ul>
Android 4.1	Jelly Bean (dragée)	16	<ul style="list-style-type: none"> <li>• Interface utilisateur plus fluide</li> <li>• Amélioration de l'accessibilité</li> <li>• Notifications extensibles</li> <li>• Recherche vocale améliorée</li> <li>• Application appareil photo améliorée</li> </ul>

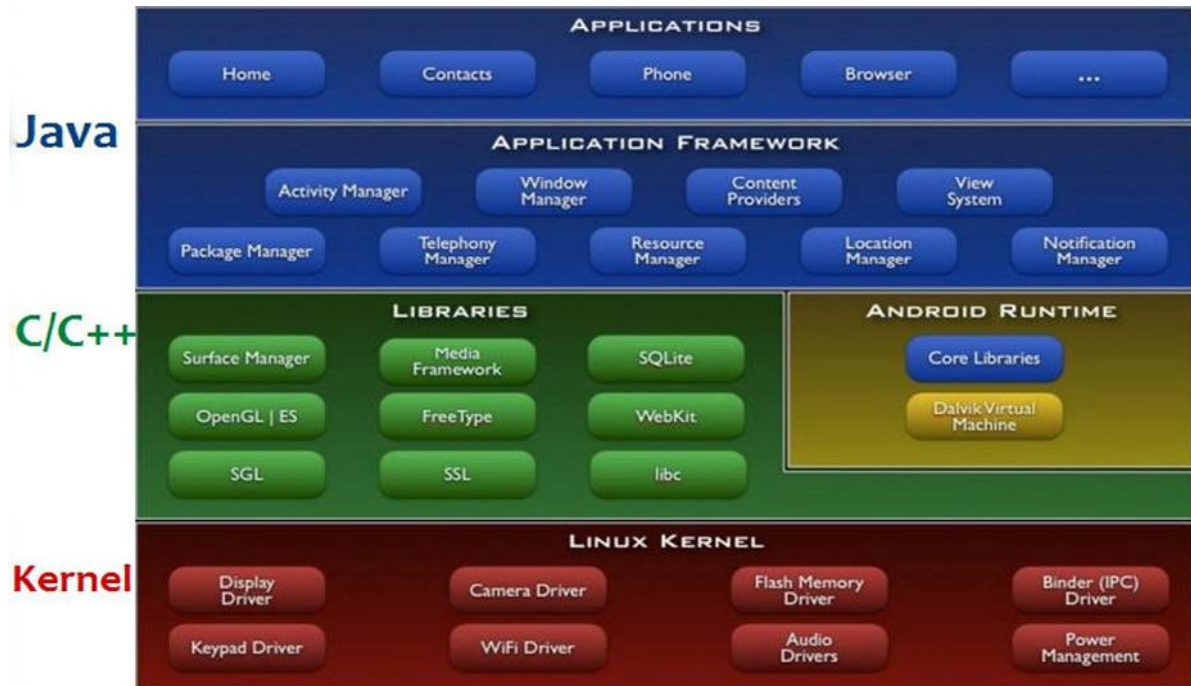
**Tableau I.1 : Fonctionnalités d'Android.**

## II.7 La plateforme android: [9]

### II.7.1 Architecture d'Android

Architecture en "pile logicielle":

La figure suivante schématise l'architecture d'Android.



**Figure II.11 : Architecture de l'Android**

Android est basé sur un kernel linux 2.6.xx.

Au-dessus de cette couche, on retrouve les librairies C/C++ utilisées par un certain nombre de composants du système Android.

Au-dessus des librairies, on retrouve l'Android Runtime. Cette couche contient les librairies cœurs du Framework ainsi que la machine virtuelle exécutant les applications.

Au-dessus de la couche "Android Runtime" et des librairies cœurs, on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework, il y a les applications.

#### II.7.1.1 Les niveaux du système d'exploitation d'Android

##### II.7.1.1.1 Linux Kernel :

Android est basé sur un kernel linux 2.6 pour ces avantages multiple comme : sa gestion mémoire et de processus reconnu pour sa stabilité et ses performances, soutien de

bibliothèque partagé, il fournit aussi les drivers pour communiquer entre les hardwares et leurs logiciels.

### II.7.1.1.2 Bibliothèque et environnement

#### ➤ Les librairies

Elle renferme les bibliothèques natives du système. Elles sont généralement utilisées par les applications natives du système. Parmi les bibliothèques nous pouvons par exemple citer bionic qui renferme entre autres une bibliothèque C et une bibliothèque pour le support du C++. Selon la page Wikipédia sur bionic], il s'agit d'une version modifiée de la bibliothèque standard C de BSD et est propre à Android. Si les applications utilisateurs sont des applications écrites en Java, il existe tout de même quelques applications natives dans le système. Il s'agit de services propres au système et de quelques outils en ligne de commande qui peuvent s'avérer utiles pour un développeur. Ces applications n'ont pas pour but d'être utilisées par l'utilisateur lambda.

#### ➤ L'environnement (Android Runtime)

C'est l'environnement d'exécution des applications Android. Il contient la bibliothèque Java utilisable par les applications ainsi qu'une machine virtuelle appelée Dalvik. La bibliothèque reprend une partie de la bibliothèque Java standard plus quelques bibliothèques supplémentaires propres à Android. La machine virtuelle Dalvik elle interprète le bytecode dans lequel les applications Android ont été compilées. Le bytecode Dalvik est différent du bytecode Java d'où l'usage de Dalvik à la place des machines virtuelles Java standard. Depuis Android 4.4, une deuxième machine virtuelle, ART, qui est livrée sous forme expérimentale avec la plateforme.

### II.7.1.1.3 Développement de l'application (Application Framework)

C'est l'ensemble des services et ressources fournies par le système Android. Il s'agit principalement de services tournant dans quelque processus clés du système Android tels que system server et mediaserver. La commande service liste retourne sous Android la liste des services présents dans le système. Sous Android 2.3, la commande retourne une liste de 50 services et sous Android 4.2 68 services. Parmi ces services nous pouvons citer ServiceManager, LocationManager. ServiceManager recense tous les autres services tournant sur le téléphone et joue un rôle d'annuaire pour les applications souhaitant accéder à un service en particulier sur le téléphone. Lorsqu'une application de navigation souhaite par exemple connaître la localisation de l'utilisateur, elle va dans un premier temps demander au Service Manager la référence de Location Service. Une fois cette référence récupérée, elle pourra demander à Location Service les données de géolocalisation. Surface linger est un autre service avec un rôle crucial sous Android. C'est lui qui est chargée de composer et dessiner ce que les différentes applications souhaitent afficher à l'écran de l'appareil.

### II.7.1.1.4 Application

Elle renferme les applications fournies par défaut sur le téléphone ainsi que celles qui seront installées plus tard par l'utilisateur. Il s'agit des applications avec lesquelles l'utilisateur pourra interagir en général (ex : application de messagerie et gestion des contacts). Les applications Android sont principalement écrites en Java. Android donne cependant la possibilité d'écrire du code natif et de l'appeler au sein de l'application grâce à une interface appelée Java Native Interface dont le rôle est d'interfacer du code Java avec du code natif écrit en C.

### II.7.2 Les composants d'une application Android

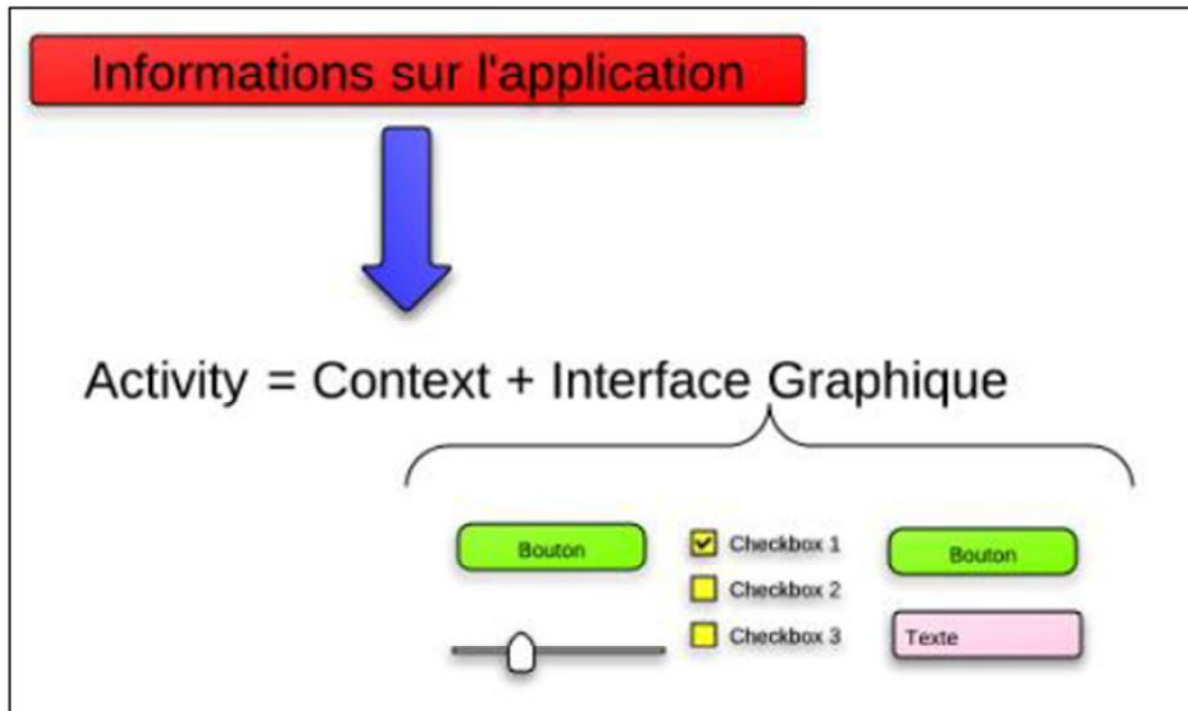
Les applications Android sont composées de 4 types de composants :

#### II.7.2.1 L'activité (Activity)

Un composant Activity est une interface utilisateur. Une application de messagerie électronique peut par exemple avoir trois composants Activity : un pour naviguer entre les répertoires de la messagerie, le deuxième pour l'affichage d'un message et le dernier pour l'Édition et l'envoi.

On peut dire aussi que la plupart des applications se compose plusieurs écrans. Chaque écran peut être réalisé par une activité. Si un nouvel écran s'ouvre, le système utilise une pile d'histoire pour stocker les écrans précédents et pouvoir reprendre l'état précédent ou enlever cet état.

De plus, une activité contient des informations sur l'état actuel de l'application : ces informations s'appellent le contexte. Ce contexte constitue un lien avec le système Android ainsi que les autres activités de l'application, comme le montre la figure suivante.



**Figure II.12:** les composants d'une activité

#### II.7.2.1.1 Cycle de vie d'une activité [10]

Une activité n'a pas de contrôle direct sur son propre état, il s'agit plutôt d'un cycle rythmé par les interactions avec le système et d'autres applications. Voici un schéma qui présente ce que l'on appelle **le cycle de vie d'une activité**, c'est-à-dire qu'il indique les étapes que va traverser notre activité pendant sa vie, de sa naissance à sa mort. Vous verrez que chaque étape du cycle est représentée par une méthode.

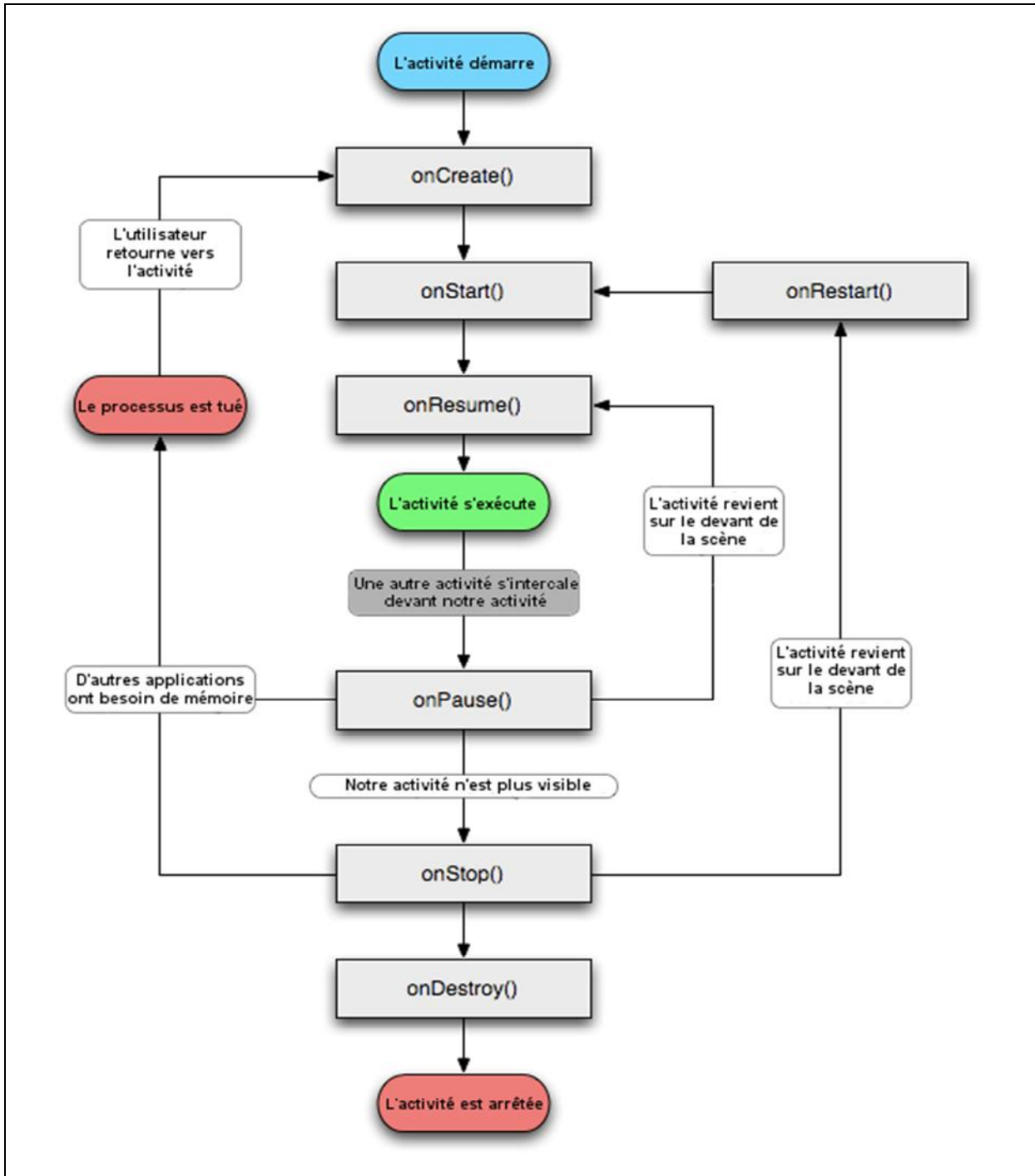


Figure II.13: Cycle de vie d'une activité

Une activité peut se trouver dans quatre états qui se différencient surtout par leur visibilité :

- **Active (Running) :**

L'activité est visible en totalité.

Elle est sur le dessus de la pile, c'est elle qui a le *focus*. C'est ce que l'utilisateur consulte en ce moment même et il peut l'utiliser dans son intégralité et agir directement dessus.

- **Paused (en pause) :**

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 et permettre d'y répondre par exemple.

Ce n'est pas sur cette activité 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 de l'application qui l'obstrue, puis il pourra à nouveau interagir avec.

- **Stopped (Arrêtée) :**

L'activité n'est pas visible à l'écran.

Si une activité est complètement masquée par une autre activité, il est arrêté. Il conserve tous les états membres et de l'information, cependant il n'est plus visible pour l'utilisateur que sa fenêtre est cachée et il sera souvent tués par le système lorsque la mémoire est nécessaire ailleurs.

- **Dead (Mort) :**

Cette activité a terminé ou il n'a jamais été démarré.

Si une activité est en pause ou arrêtée, le système peut chuter l'activité de la mémoire, soit par lui demandant de se terminer, ou tout simplement tuer le processus. Quand il est affiché de nouveau à l'utilisateur, il doit être redémarré et restauré à son état antérieur.

Les transitions d'états sont captées par les méthodes suivantes :

- **onCreate()** : l'activité est en création
- **onStart()** : l'activité va devenir visible
- **onResume()** : l'activité est maintenant visible
- **onPause()** : l'activité va être mise en pause
- **onStop()** : l'activité ne sera plus visible
- **onDestroy()** : l'activité va être détruite

### II.7.2.2 Content Provider

Un ContentProvider est un composant servant au partage des données d'une application. Son rôle est de servir d'interface entre l'application souhaitant accéder aux données et les données. Ces données sont généralement stockées dans une base de données locale SQLite mais aucune restriction n'est imposée sur la manière de stocker les données. La liste des Contacts est par exemple stockée dans une base de données locale dont l'accès se fait à travers un composant de type ContentProvider.

### II.7.2.3 Service

Un service est un composant effectuant des tâches en arrière plan. Il est utilisé pour effectuer de longues tâches internes à l'application ou à exécuter une tâche à la demande d'une application. Dans un client File Transfer Protocol, un service est utilisé pour envoyer les

données sur le réseau à les recevoir. Cela évite de bloquer l'application jusqu'à la fin de l'envoi.

### II.7.2.3 Broadcast Receiver

Un Broadcast Receiver est un composant utilisée pour écouter les messages en large diffusion sur le système. Un exemple de ce type de message est la réception d'un nouveau SMS. Lorsqu'un nouveau SMS est reçu par le téléphone, le système envoie un message en broadcast pour notifier les différentes applications d'envoi et réception de SMS. Ce composant ne possède aucune interface graphique et n'est pas censée exécuter de longues tâches.

## II.8 Développement d'application pour Android [11]

Pour développer des applications pour Android, il est nécessaire d'installer les éléments suivant :

- l'environnement de développement intégrer Eclipse (<http://www.eclipse.org>).
- Le SDK Android (<http://developer.android.com/sdk>).

### II.8.1 Le SDK Android

Un SDK, un kit de développement dans notre langue, est un ensemble d'outils permettant de développer pour une cible particulière. Par exemple pour développer pour une console de jeu vidéo, on utilise un SDK spécifique pour développer des applications pour cette console.

Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android. Il est disponible sous forme d'un fichier zip ou d'un exécutable, il représente l'outillage indispensable au développeur Android. Ce kit contient en effet les outils nécessaires pour programmer avec Android, exécuter ses programmes, tester, et livrer ses programmes Android à ses clients.

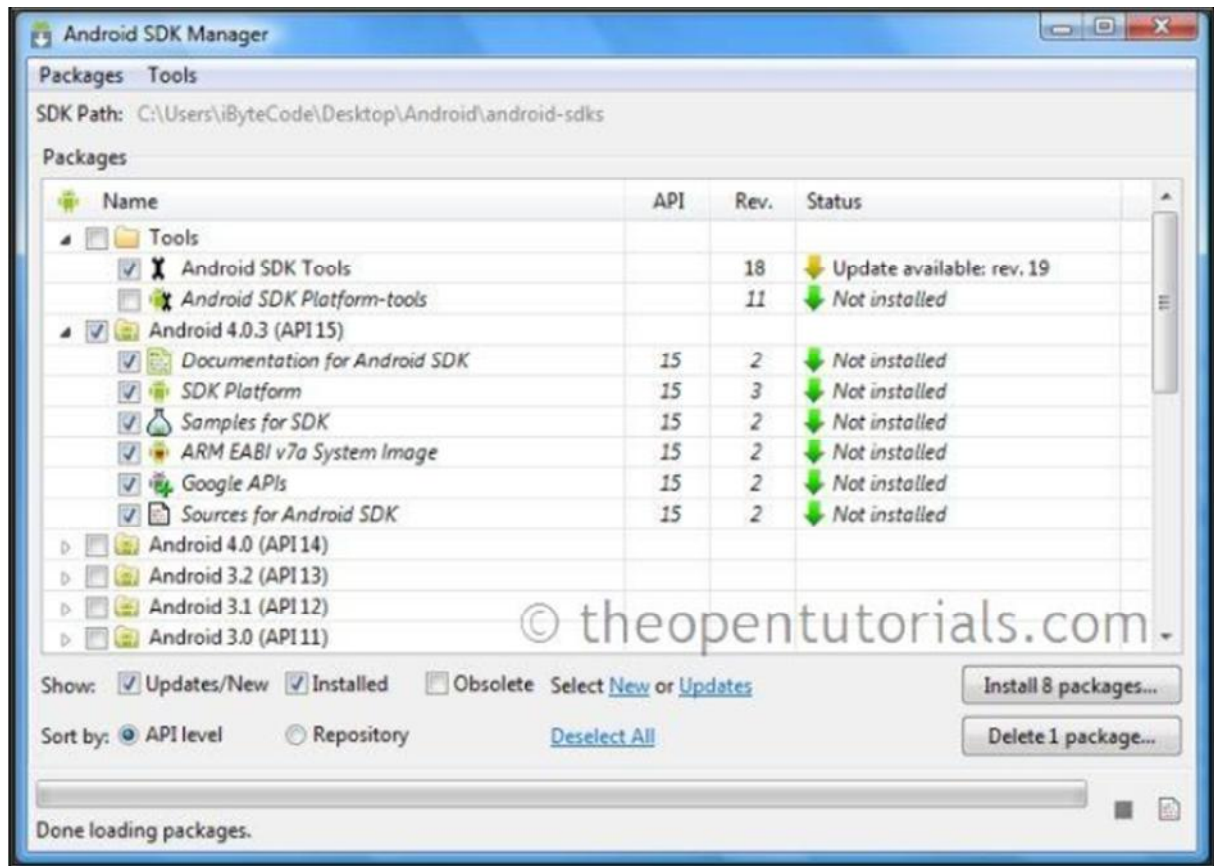


Figure II.14: Android SDK

### II.8.1.1 les outils fournis par le SDK

#### II.8.1.1.1. AVD (Android Virtual Device):

L'**Android Virtual Device**, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui se fait passer pour un appareil sous Android à votre ordinateur. C'est la raison pour laquelle vous n'avez pas besoin d'un périphérique sous Android pour développer et tester la plupart de vos applications ! En effet, une application qui affiche un calendrier par exemple peut très bien se tester dans un émulateur, mais une application qui exploite le GPS doit être éprouvée sur le terrain pour que l'on soit certain de son comportement.



**Figure II.15:** L'émulateur

#### II.8.1.1.2 L'ADB (Android Debug Bridg)

ADB est un outil disponible en ligne de commande polyvalent. Il permet de communiquer avec une instance de l'émulateur directement ou avec un téléphone physique à l'aide d'un câble USB standard. Il donne la possibilité d'installer ou de désinstaller des applications, d'envoyer ou de récupérer des fichiers et de se connecter en ligne de commande sur l'appareil. Il est aussi utilisé par le SDK pour publier et déboguer des applications.

#### II.8.1.1.3. Le DDMS (Dalvik Debug Monitor Server)

Android DDMS est l'Android Dalvik Debug Monitor Server, il permet d'exécuter quelques fonctions pour vous aider à déboguer votre application (simuler un appel ou une position géographique par exemple) et d'avoir accès à d'autres informations utiles.

## II.9 Concepts et innovation

Android possède aussi un environnement très riche, ainsi que des fonctionnalités toute aussi intéressantes, voici quelques unes auxquelles nous feront appel dans la réalisation de notre projet :

### II.9.1 Le bureau virtuel

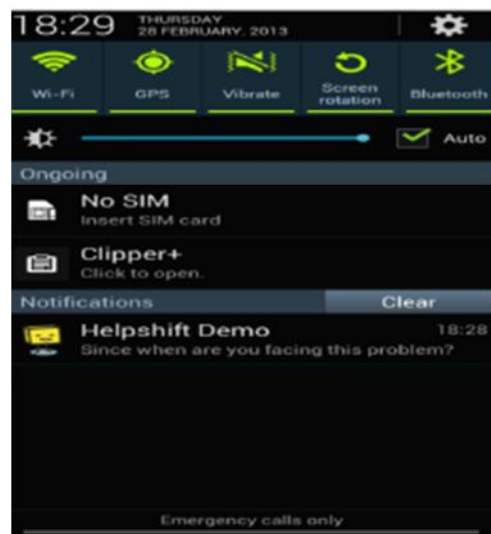
Le bureau virtuel est l'écran d'accueil du Smartphone, il est composé de 5 parties (ou plus). Chacune est personnalisable, il est possible d'y mettre des raccourcis (vers des applications, des fichiers, des dossiers, des contacts) ou des widgets (calendrier, horloge, notes,... etc). L'image de fond s'étend aussi sur tout le bureau, et bouge dès qu'on passe sur une partie à une autre ce qui donne l'impression que le contenu fait partie du décor. Cependant et dans le but de faire la différence, les constructeurs ajoute leurs touches personnelles par exemple : le constructeur HTC possédant l'interface sense qui est dotée de 7 bureaux. Le problème avec cette personnalisation est que lorsqu'une version nouvelle d'Android sort, chaque constructeur doit procéder aux adaptations dans sa propre version. Donc soit la mise à jour n'est jamais effectuée, soit elle sort plusieurs semaines après la version de base.



**Figure II.16 :** Bureau virtuel

### II.9.2 Les notifications

Une notification est une indication qui s'affiche sur la barre qui se situe en haut d'un téléphone Android. Cette notification sert à prévenir un utilisateur de certains événements, comme la réception d'un message ou d'un appel manqué, le niveau de batterie, ainsi que les connectivités :réseau, Bluetooth, GPS...et



**Figure II.17:** Exemple de notification

### II.9.3 SQLite

Android ne fournit aucune base de données de son propre chef. C'est donc au développeur que revient la tâche d'en créer un type SQLite.

SQLite est une bibliothèque écrite en C, et comme son nom l'indique, utilise un dialecte de SQL pour effectuer des requêtes, des manipulations de données, et des définitions de données. Il est si efficace en termes de mémoire que le moteur d'exécution d'Android peut l'inclure dans son intégralité. Contrairement aux serveurs de base de données classiques, comme MySQL, il ne reproduit pas le schéma habituel client-serveur mais est directement intégré aux programmes. L'intégralité de la base de données (déclaration, table, index et données) est stockée dans un fichier indépendant de la plateforme.

### II.9.4 Android Market

Android Market est l'ancien nom du market Android (application store), Android Market est devenu Google Play le 6 mars 2012. Android Market alias Google Play est un site internet et une application qui permettent d'acheter et d'installer des applications Android (jeux vidéo et utilitaires) mais aussi de louer ou acheter des films, des séries TV, de la music, des livres. Le Play Store (anciennement Android Market) est une plateforme immense et très visitée, c'est donc une mine d'opportunités pour quiconque.

### II.10 Présentation du langage java [12]

#### II.10.1 Historique

Développé par Sun Microsystems depuis la fin des années 1980, JAVA est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Il existe deux types de programmes en JAVA : les applets et les applications. Une application autonome (stand alone program) est une application qui s'exécute sous le contrôle direct du système d'exploitation. Une applet est une application qui est chargée par un navigateur et qui est exécutée sous le contrôle d'un plug-in de ce dernier.

Les principaux événements de la vie de JAVA sont les suivants :

- 1995 Mai: premier lancement commercial
- 1996 Janvier : JDK 1.0
- 1996 Septembre : lancement du JDC
- 1997 Février : JDK 1.1
- 1998 Décembre : lancement de J2SE et du JCP
- 1999 Décembre : lancement J2EE
- 2000 Mai : J2SE 1.3
- 2002 J2SE 1.4
- 2004 J2SE 5.0

#### II.10.2 Les caractéristiques de Java

JAVA possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

➤ **Java est interprété**

le source est compilé en pseudo code ou byte code puis exécuté par un interpréteur Java : la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le byte code, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir quasiment les mêmes résultats sur toutes les machines disposant d'une JVM.

➤ **Java est portable**

il est indépendant de toute plate-forme, il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du byte code.

➤ **Java est orienté objet:**

comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les

autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).

➤ **Java est simple**

le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...

➤ **Java est fortement typé**

toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.

➤ **Java assure la gestion de la mémoire**

l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.

➤ **Java est sûr**

la sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme Java planté ne menace pas le système d'exploitation. Il ne peut pas y avoir d'accès direct à la mémoire. L'accès au disque dur est réglementé dans une applet.

➤ **Java est économe**

le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.

➤ **Java est multitâche**

il permet l'utilisation de threads qui sont des unités d'exécution isolées. La JVM, elle-même, utilise plusieurs threads.

### II.10.3 Outils de développement

La communauté Java est très productive car elle regroupe :

- Sun, le fondateur de Java à travers le JCP (Java Community Process) qui est le processus de traitement des évolutions de Java dirigé par Sun. Chaque évolution est traitée dans une JSR (Java Specification Request) par un groupe de travail constitué de différents acteurs du monde Java ;
- des acteurs commerciaux dont tous les plus grands acteurs du monde informatique excepté Microsoft ;
- la communauté libre qui produit un très grand nombre d'api et d'outils pour JAVA.

### II.11 Les avantages d'Android [13]

#### ❖ **Open Source et Gratuit:**

Le gros avantage d'Android, contrairement à Windows Mobile et Mac OS mobile (IOS), vient du fait qu'il soit **Open Source**. Cela signifie qu'un constructeur, ou un opérateur, a la possibilité de modifier à la base du système, en ajoutant, en natif, toutes les applications qu'il souhaite.

L'autre avantage d'Android est sa **Gratuité**. En effet, un constructeur de téléphone mobile, ou un opérateur, qui souhaite intégrer l'OS de Google à un smartphone ne devra pas acheter de licence au géant de l'Internet.

Le fait que l'OS ne soit pas payant, permettent, aux constructeurs ou aux opérateurs, de proposer des mobiles moins chers que s'ils étaient équipés de Windows Mobile.

#### ❖ **Facile à développer:**

Toutes les APIs mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire que vous pouvez envoyer un SMS en seulement deux ligne de code.

#### ❖ **Flexible**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les Smartphones, les tablettes, la présence ou l'absence de clavier, différents processeurs... on trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android. Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal (si votre application nécessite d'utiliser le Bluetooth, seuls les terminaux équipés de Bluetooth pourront la voir sur le Play Store).

#### ❖ **Complémentaire**

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est que vous pouvez combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.

#### ❖ **Facile à développer**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire que vous pouvez envoyer un SMS en seulement deux lignes de code (concrètement, il y a un peu d'enrobage autour de ce code, mais pas tellement).

### **II.12 Conclusion**

Dans ce chapitre, nous avons fait une étude de l'état de l'art d'Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture d'Android, à savoir les principaux composants du système.

Le chapitre suivant présente les détails de l'analyse et conception de notre application.

# **Chapitre III : Analyse et conception**

### III.1 Introduction:

La conception de toute solution informatique est d'une grande importance et elle doit être traitée avec rigueur et précision, car elle constitue la base du système à développer. Avant de s'engager dans la conception, il est impératif de passer par la phase d'analyse qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs besoins. Puis on passe à la conception qui, en s'appuyant sur les résultats de la phase d'analyse, donnera la description détaillée du système cible et des objectifs à atteindre.

Pour ce faire, notre démarche va s'appuyer sur le langage UML, qui permet la conception et développement de l'application.

### III.2 Les Méthodes de conception [14]

Les méthodes utilisées dans les années 80 pour organiser la programmation fonctionnelle (notamment MERISE) étaient fondées sur une modélisation séparée des données et des traitements.

Lorsque la programmation orientée objet prend de l'importance au début des années 90, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995, mais uniquement trois d'entre elles se sont détachées nettement après quelque années : OMT, BOOCH et OOSE.

Chaque méthode avait ses avantages et ses partisans. Le nombre de méthode en compétition s'était réduit mais le risque d'un éclatement subsistait : la profession pouvait se diviser entre ses trois méthodes, créant autant de continents intellectuel qui auraient eu du mal à communiquer. C'est pour cela que l'unification de ses trois méthodes c'est avérée indispensable.

Pour la modélisation de notre application, nous allons utiliser le langage de modélisation unifié UML.

### III.3 Présentation de l'UML:

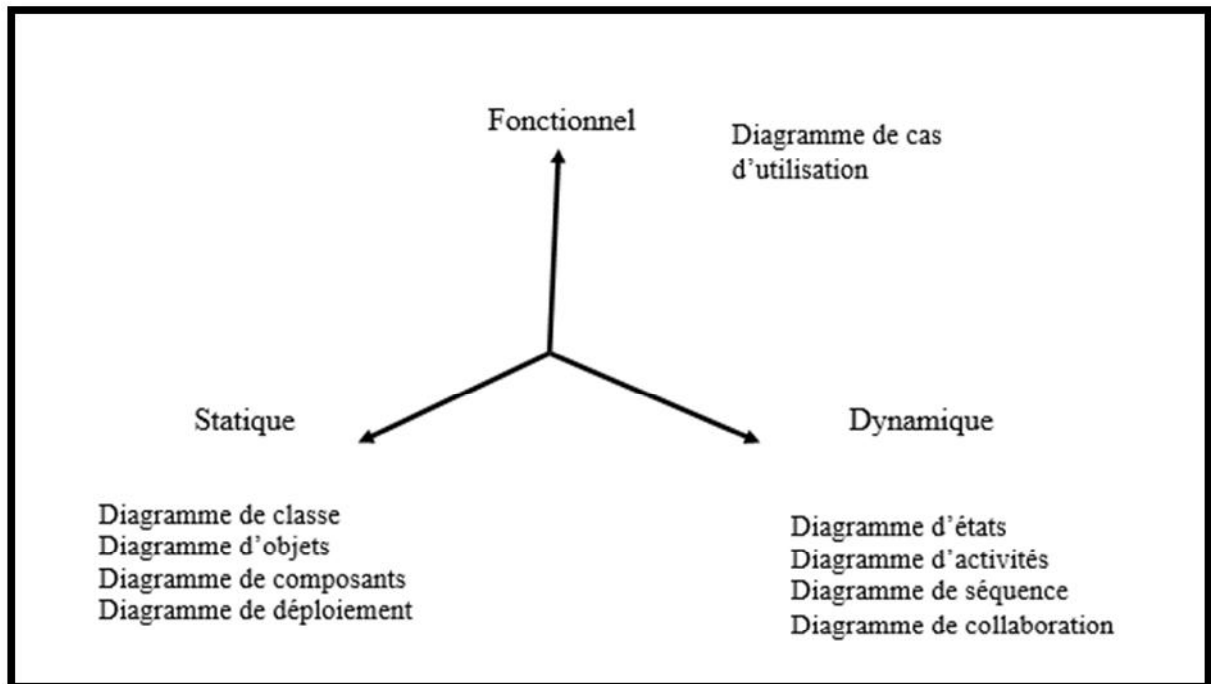
UML est un langage de modélisation objet qui décrit la Mise en oeuvre de la notation développée en réponse à l'appel lancé par OMG (Object management groupe) dans le but de définir une notation standard par modélisation des Applications construites à l'aide des objets.

UML possède 9 diagrammes de modélisation, répartis sur trois axes du niveau conceptuel :

- **Fonctionnel**
- **Structurel**
- **Temporel**

### III.3.1 Modélisation avec l'UML:

UML propose 9 diagrammes de modélisation, réparties sur trois axes du niveau conceptuel



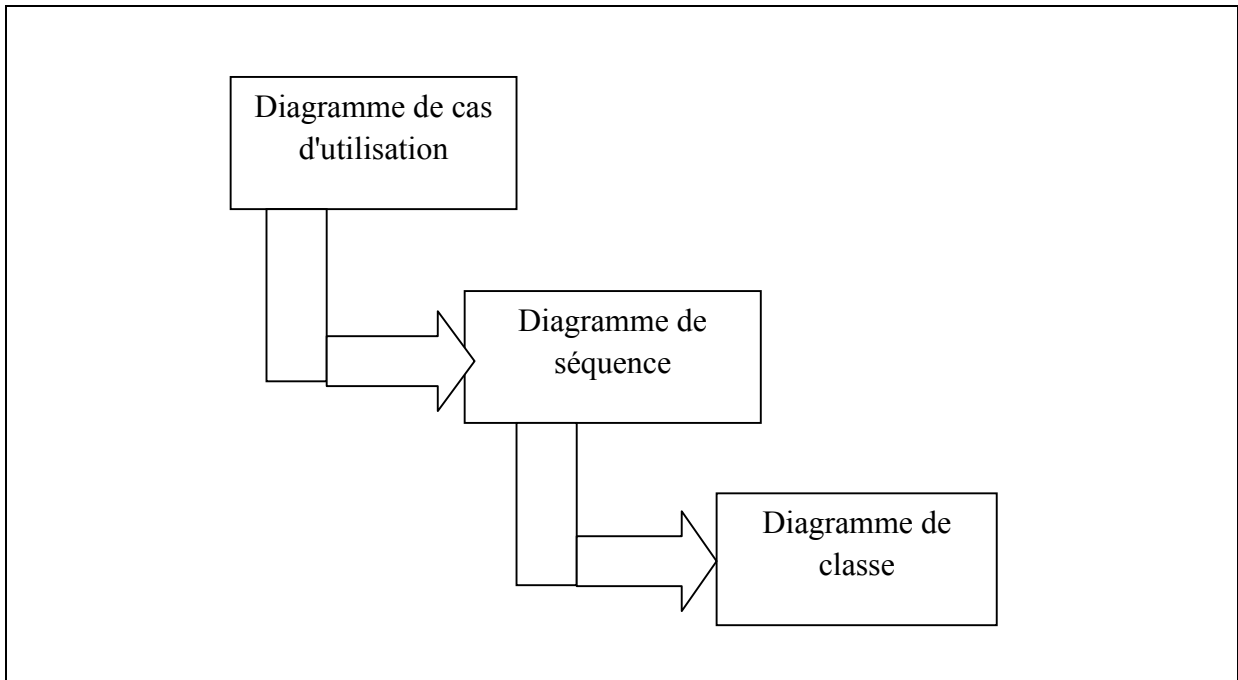
**Figure III.1** : Les diagrammes de modélisation

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 itératif 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.3.2 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.4 Analyse :

#### III.4.1 Identification des acteurs [15]

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 toute personne qui utilise une machine (Téléphone, Tablette, etc..) qui marche sous le système Android 2.1 ou une version ultérieure. On les regroupe sous le nom utilisateur « **User** ».

#### III.4.2 Identification des besoins

L'objectif principal de notre application est de permettre à l'utilisateur de filtrer les appels indésirables.

Les fonctionnalités de notre application sont :

- Bloquer tous les numéros .
- Bloquer les numéros non enregistrés.
- Bloquer les numéro de la liste.
- Annuler tout

### III.4.3 Spécification des tâches :

**Définition :** Une tâche est l'ensemble des différentes fonctions qui peut être accédés pas un acteur bien spécifique.

L'acteur définis précédemment (**user**) effectue un certain nombre de tâches, ces tâches sont résumées dans le tableau ci-dessous :

Acteurs	Taches
<b>User</b>	T01 : Ajouter un nouveau numéro dans la liste T02 : afficher la liste des numéros indésirables T03 : Supprimer un numéro de la liste T04 : Bloquer les numéros non enregistrés T05 : Bloquer un ou plusieurs numéros de la liste T06 : Bloquer tous les numéros T07 : Annuler tout ( débloquer tous les numéros)

**Tableau III.1 : Spécification des tâches**

### III.4.4 Spécification des scénarios

**Définition :** Un scénario est une instance (occurrence) d'un cas d'utilisation, à chaque fois qu'une instance d'un acteur déclenche un cas d'utilisation, un scénario est créé. Ce scénario suivra un chemin particulier dans le cas d'utilisation.

**NB :** Durant notre étude les scénarios seront symboliser par **Si** (**i** : représente le numéro du scénario).

Appareil : définit un Téléphone ou Tablet

Le tableau ci-dessous récapitule les différents scénarios :

Acteurs	Taches	Scénarios
<b>User</b>	T01 : Ajouter un nouveau numéro dans la liste	S01 : Cliquer sur l'icône de l'application. S02 : Cliquer sur le bouton « Ajouter numéro » S03 : Saisir le nouveau numéro S04 : Cliquer sur le bouton « Ajouter »
	T02 : afficher la liste des numéros indésirables	S01, S05 : Cliquer sur le bouton « Voir liste »

	T03 : Supprimer un numéro de la liste	S01, S05 S06 : Sélectionner un numéro S07 : Cliquer sur le bouton « Supprimer »
	T04 : Bloquer les numéros non enregistrés	S01, S07 : Cliquer sur le bouton « numéro non enregistré »
	T05 : Bloquer un ou plusieurs numéro de la liste	S01, S08: Cliquer sur le bouton « Liste » S05, S06
	T06 : Bloquer tous les numéros	S01 S09 : Cliquer sur le bouton «Tous les appels»
	T07 : Annuler tout	S01 S10 : Cliquer sur le bouton « Annuler tout »

**Tableau III.2 : Spécification des scénarios**

### III.5 Conception: [16]

Le processus de conception de notre projet se caractérise par un seul niveau : le niveau applicatif.

Le niveau applicatif s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML, A cet effet nous avons adopté la démarche suivante:

- Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en oeuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré.
- Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un digramme d'activité.
- Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

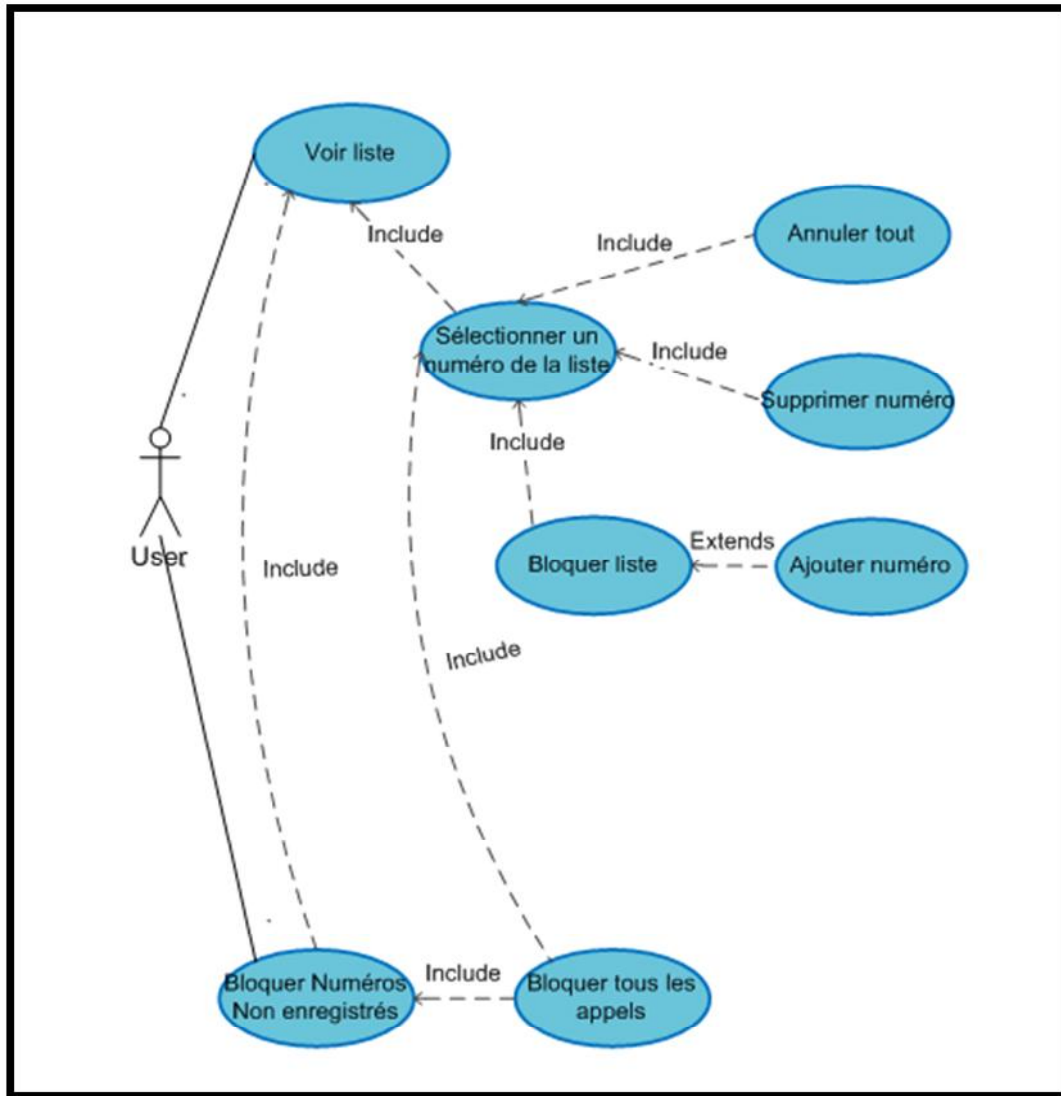
#### III.5.1 Les diagrammes des cas d'utilisation

Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent 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.

- **La relation d'inclusion (include) :** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette

relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.

- **La relation d'extension (Extend)** : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.



**Figure III.3** : Diagramme de cas d'utilisation globale

### III.5.2 Les diagrammes de séquence

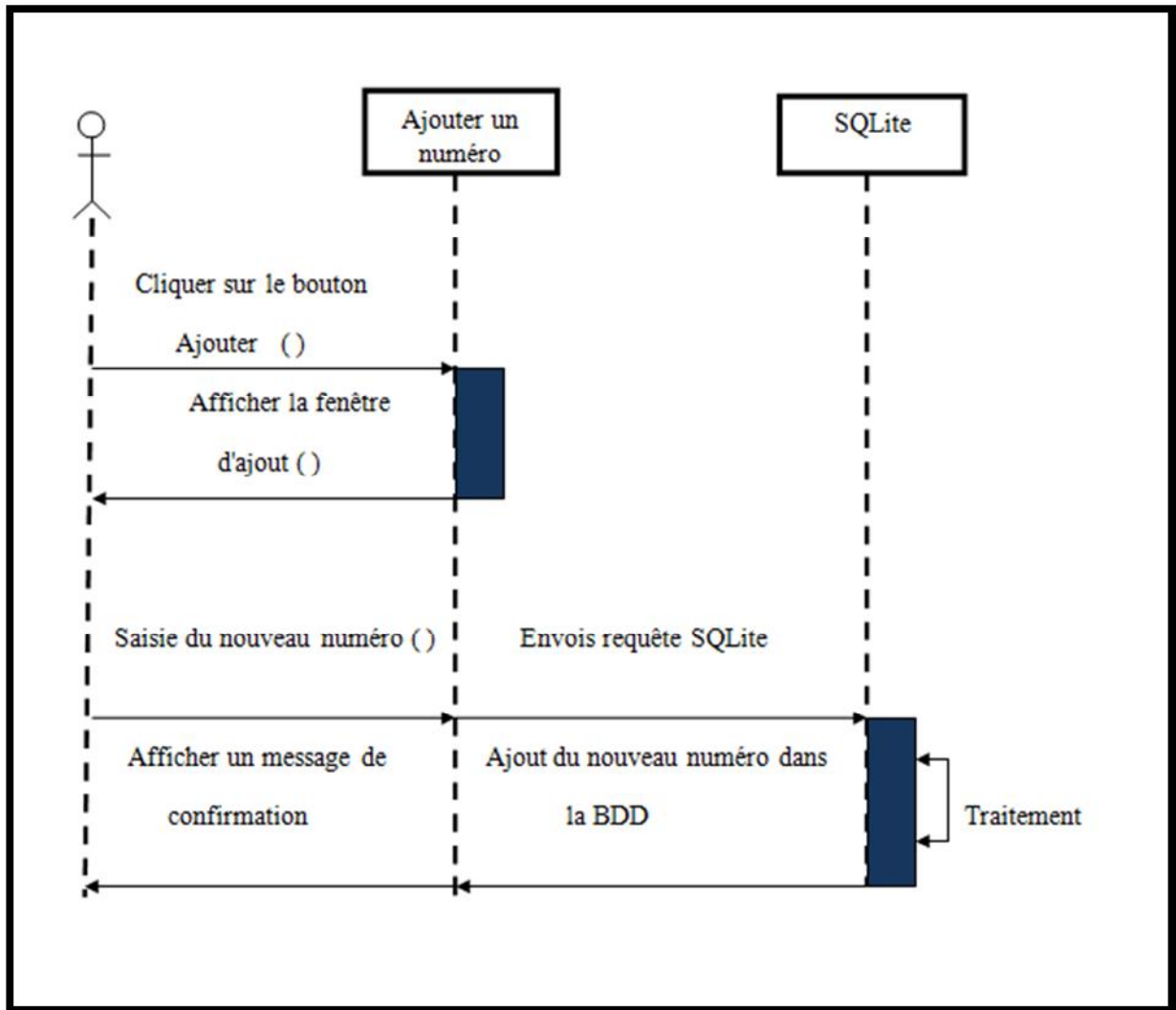
Un diagramme de séquence permet de spécifier les interactions qui existent entre un groupe d'objet selon un point de vue temporel, on y met l'accent sur la chronologie des envois de message.

Le diagramme de séquence est toujours de haut vers le bas, il illustre l'ordre dans lequel les messages sont envoyés entre les objets. Il peut servir à illustrer un cas d'utilisation.

Les composants d'un diagramme de séquence sont les suivants :

- **Les objets** : ils apparaissent dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.
- **Les messages** : ils sont représentés par des flèches directionnelles. Au-dessus des flèches directionnelles figurent un texte nous informant du message envoyé entre les objets.

Ci-dessous nous présentons quelques diagrammes de séquences :



**Figure III.4** : Diagramme de séquence de cas d'utilisation « Ajouter un nouveau numéro »

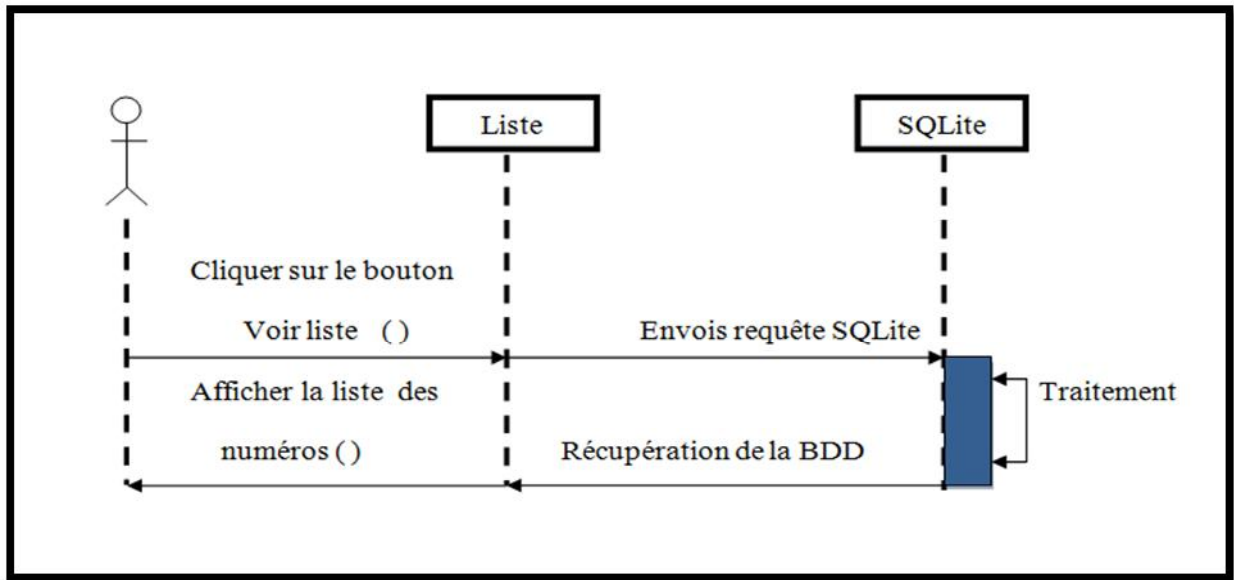


Figure III.5: Diagramme de séquence de cas d'utilisation « Afficher la liste des numéros indésirable »

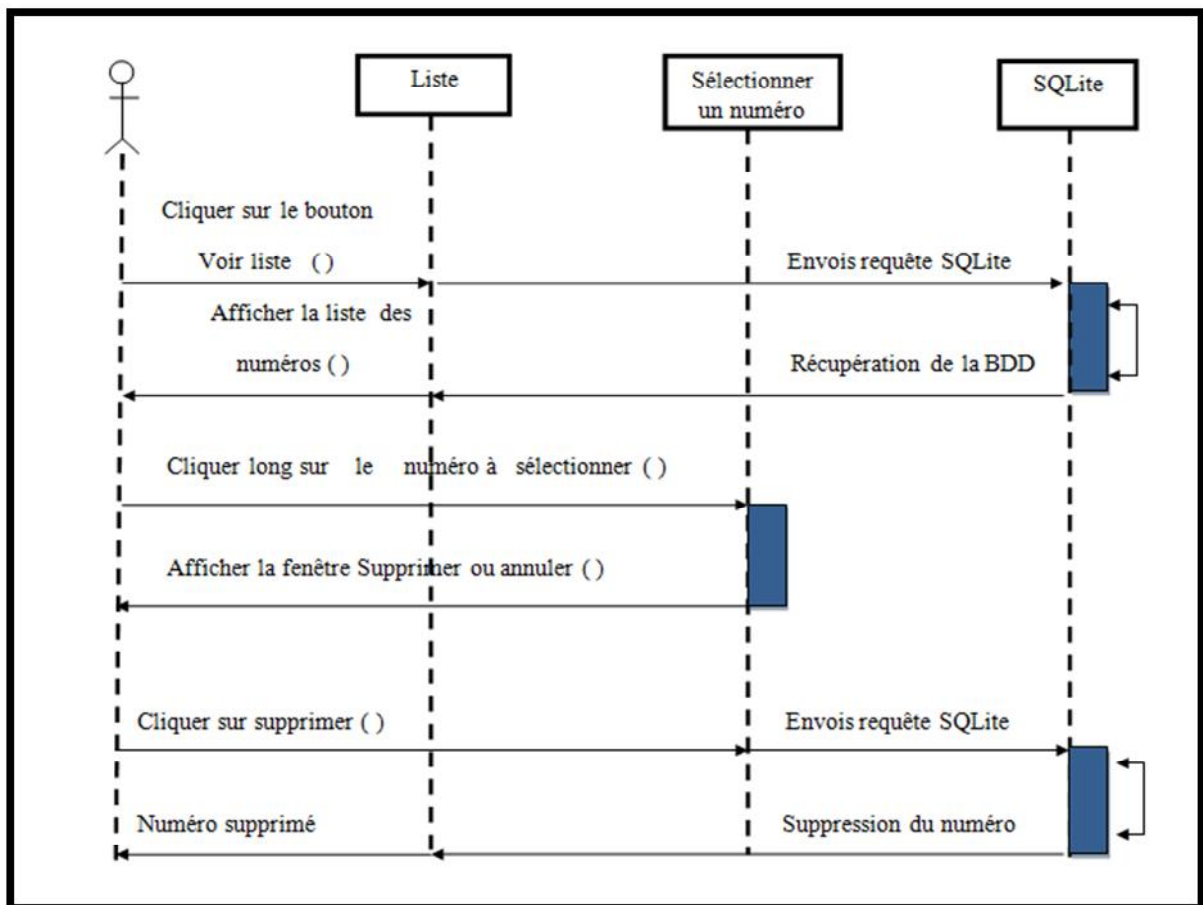


Figure III.6 : Diagramme de séquence de cas d'utilisation « débloquer un numéro de la liste »

## III.5.3 les diagrammes de classes

Le **diagramme de classes** est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

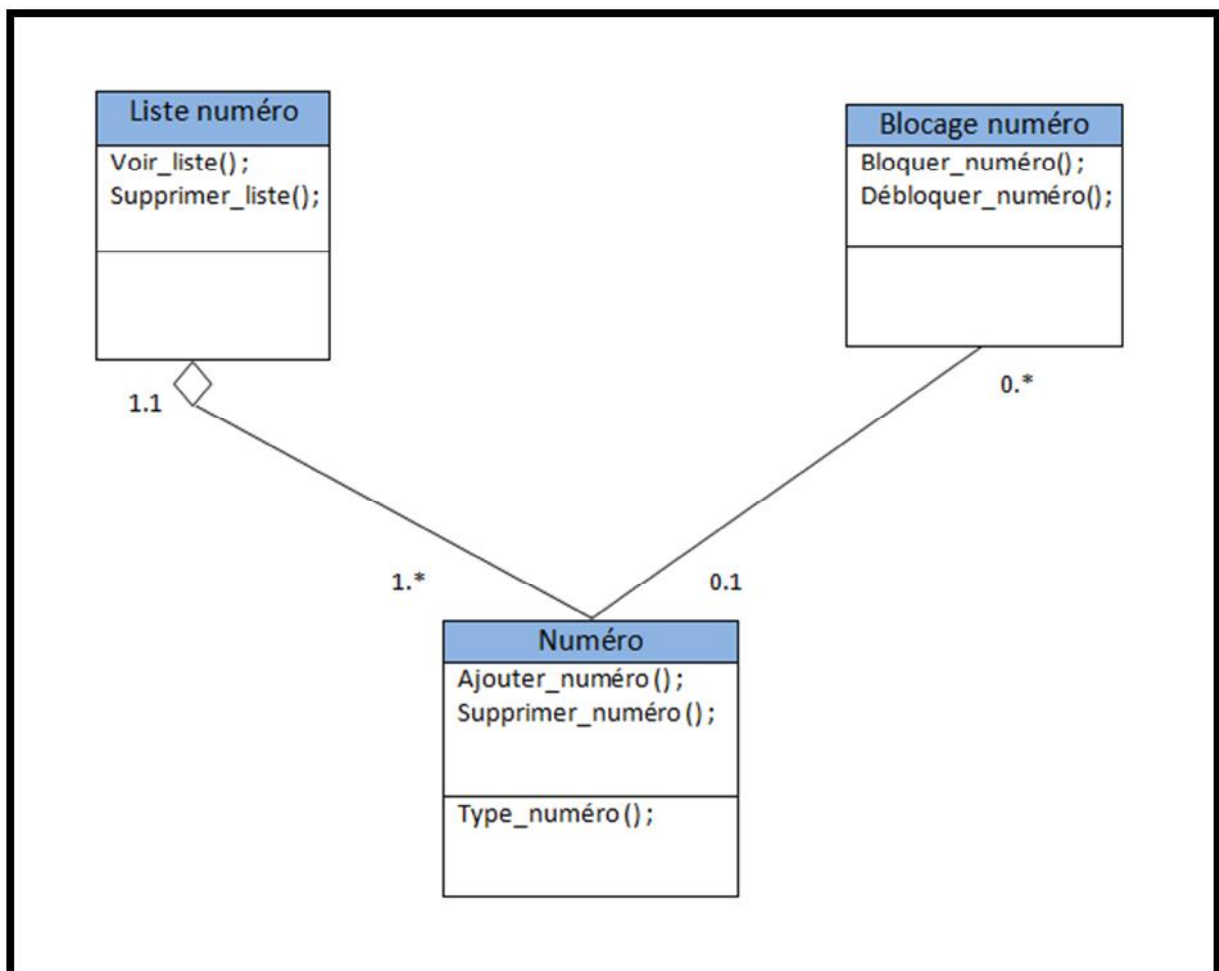


Figure III.7 : le diagramme de classe générale

### III.6 Conclusion

Dans ce chapitre, nous avons introduit les fonctionnalités de notre application. Pour les atteindre, nous avons proposé une solution fondée sur une analyse et une conception modélisées à l'aide du langage de modélisation unifié UML. Pour cela, nous avons défini les acteurs de notre application, les tâches qu'ils assurent et les scénarios associés à chaque tâche. Nous avons élaboré les diagrammes de cas d'utilisation global, élaboré les diagrammes de séquence et enfin le diagramme de classe générale

Le chapitre suivant sera consacré à la réalisation de notre application, en présentant les outils de développement utilisés et les différentes fonctionnalités de notre application à travers ses différentes interfaces.

# **Chapitre IV : Réalisation**

## IV.1 Introduction

Après la présentation de l'analyse et de la conception de notre logiciel nous allons présenter dans ce chapitre l'environnement de développement de ce dernier ainsi que les différents outils utilisés pour sa réalisation puis expliquer son fonctionnement en présentant quelques interfaces illustratives.

## IV.2 Environnement de travail

### IV.2.1 Environnement matériel

Dans notre projet, l'environnement de travail disponible pour le développement de l'application est le suivant :

- PC Portable : Intel Core i5, 2.40 GHZ, 4,00 GO de mémoire vive, Windows 7.

### IV.2.2 Environnement cible

Notre application est destinée pour tous les Smartphones équipé d'un système Android 2.2 dite Froyo et plus.

### IV.2.3 Environnement logiciel

Avant de pouvoir faire du développement, il faut nécessairement préparer l'environnement de travail. Dans notre cas, on a choisi de faire notre développement mobile sur la plateforme Eclipse. Pour la préparer, on a effectué différentes tâches qui sont :

- Installation Eclipse.
- Installation SDK Android.
- Intégration SDK Android sous Eclipse pour pouvoir créer des projets Android.

Après la préparation de logiciel de travail éclipse et pour pouvoir développer une application dans de bonnes conditions, il faut bien savoir choisir son environnement de développement selon les besoins, de ce fait, on a développé une application pour mobile en JAVA, qui va être compilé en fichier DEX par le Dalvik VM ensuite déployé sous une extension APK pour être installé sur un mobile fonctionnant avec le système Android, pour cela on a choisi Eclipse IDE qui nous permettra de réaliser ce travail.

#### IV.2.3.1.Système d'exploitation

Notre application a été développée sous le système d'exploitation Windows 7 service pack1, en langage JAVA. Elle peut être ainsi intégrée dans n'importe quel autre système d'exploitation supportant la machine virtuelle java.

### IV.2.3.2 Langage de programmation

Le langage java est un langage de programmation orienté objet mis au point par Sun Microsystems. Sa caractéristique principale est qu'il est indépendant de toute plate-forme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une «Java Virtual Machine » (JVM).

Ce concept est à la base du slogan de Sun pour java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java Development Kit).

Java est caractérisée aussi par la réutilisabilité de son code ainsi que la simplicité de sa mise en œuvre.

### IV.2.3.3 Eclipse

Eclipse est un IDE qui permet de programmer dans différents langages grâce à ses nombreux plug-ins et notamment le plug-in d'Android. Une interface spécifique permet de gérer des fichiers java et de compiler ses programmes. Les fichiers sont organisés selon une arborescence qui correspond aux paquetages java définis. L'analyse syntaxique permet de mettre en valeur les mots clés dans les fichiers java.

Eclipse dispose aussi d'un système d'auto complétion des fonctions, de détection des erreurs syntaxiques en temps réel sans oublier un système de débogage permettant d'exécuter ses programmes pas à pas.



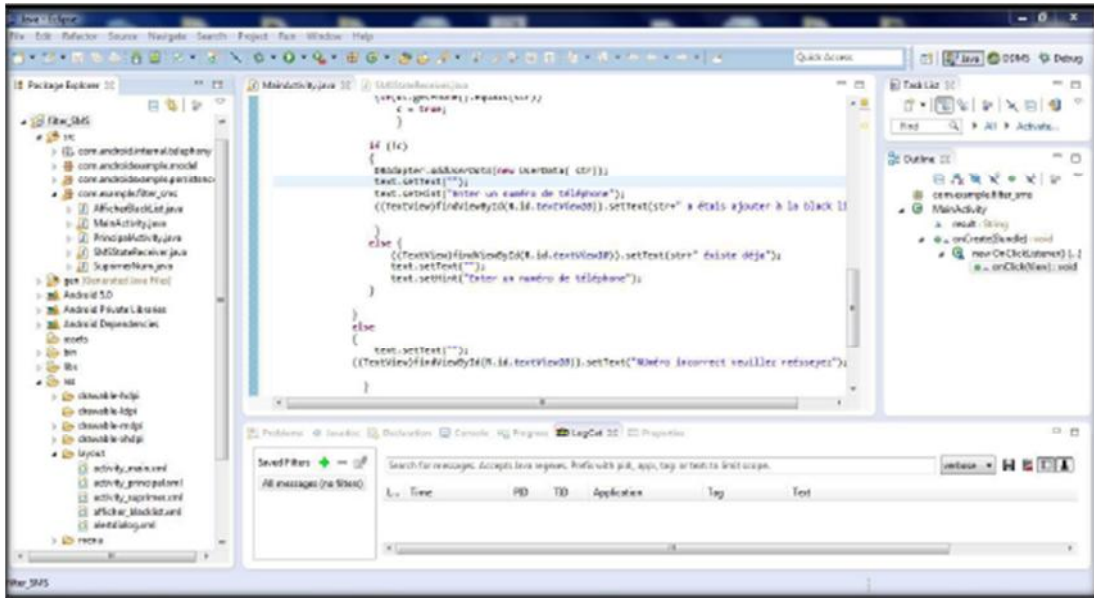
**Figure IV.1 : Logo Eclipse**

### IV.2.3.4 IDE Eclipse

L'IDE Eclipse est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en java. Par ailleurs il existe un grand nombre de modules pour étendre l'IDE Eclipse.

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

Pour le développement de notre application, on a choisi Eclipse KEPLER (Son interface principale est donnée dans la figure ci dessous, avec le plugin ADT de Google).



**Figure IV.2:** plate-forme Eclipse

### IV.3 Prise en main de l'environnement Android

La première étape de notre travail avec l'environnement Android a été d'appréhender le SDK, l'architecture et le développement d'une application ainsi que son déploiement sur un terminal embarquant Android.

Google a mis en place un grand nombre d'outils pour aider les développeurs Android.

- **Le portail des développeurs :**

La première chose à visiter est le portail des développeurs Android, mis en place par Google.



**Figure IV.3** : Portail des développeurs Android

Très complet, ce site présente Android, explique comment installer et utiliser les différents outils (SDK, NDK etc.), propose un ensemble de tutoriels et articles concernant le développement d'applications Android, expose la référence de l'API Android ainsi que les actualités liées à Android.

Le tout est très bien fait et permet de rapidement être confortable vis-à-vis du développement sur Android.

### IV.3.1 Software Development Kit (SDK)

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.).

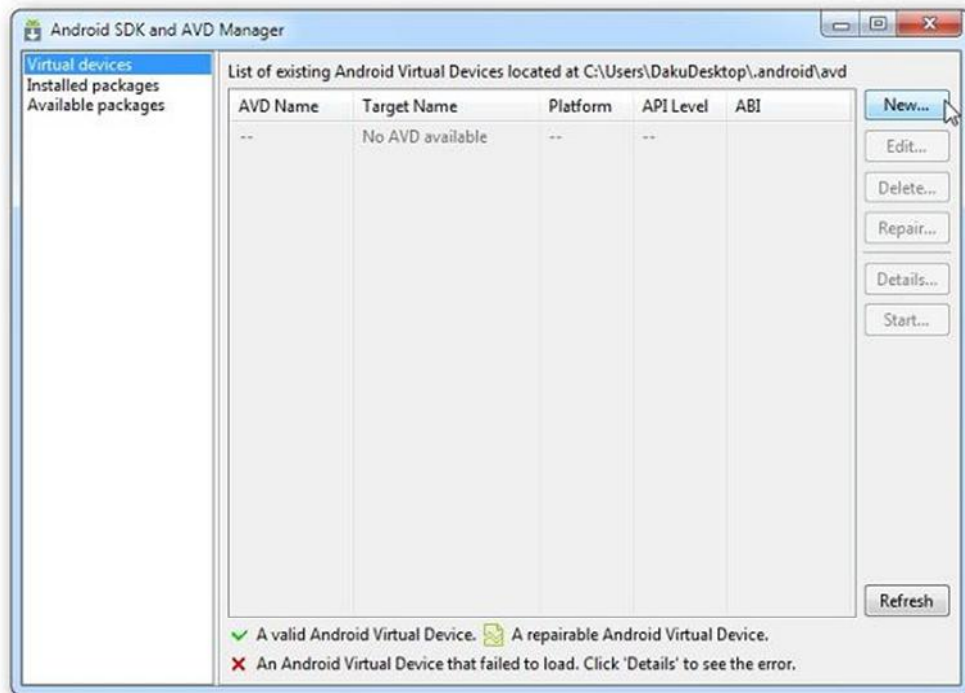
Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google tels que Maps etc.) ou de la documentation JavaDoc. Son fonctionnement est similaire aux gestionnaires de paquets de Linux.

Le SDK Android est composé de plusieurs éléments pour aider les développeurs à créer et à maintenir des applications :

- Des API (interfaces de programmation) ;

- Des exemples de code ;
- De la documentation ;
- Des outils – parmi lesquels un émulateur – permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application .

C'est un kit de développement basé sur le langage Java. Il s'agit des outils que Google fournit pour interagir avec Eclipse



**Figure IV.4:** Interface d'installation du SDK Android

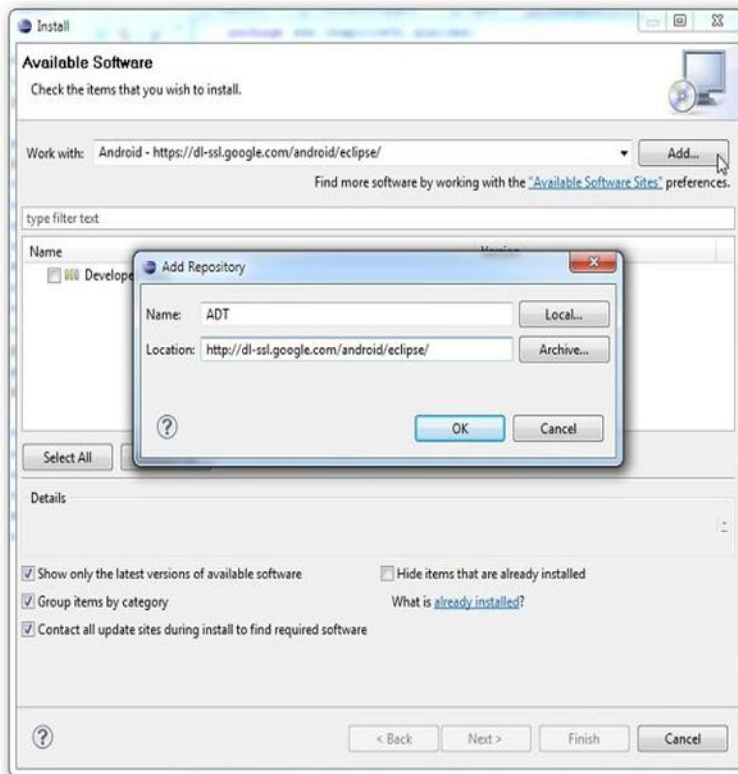
### IV.3.2 Le plugin ADT pour Eclipse

Eclipse est l'Environnement de Développement Intégré (ou IDE) le plus largement utilisé pour la programmation Java; très performant, il est de plus gratuit et open source.

Le langage privilégié pour le développement d'applications Android est justement Java. Google a donc tout naturellement conçu un plugin (Un plugin est un module qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités) pour Eclipse.

Android Development Tools, ou ADT, est très complet et surtout très pratique : conception graphique d'interfaces utilisateur, debug distant sur un téléphone, gestion de l'architecture de fichiers d'une application etc.

Pour développer en Android, on a installé le plugin Android qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android.



**Figure IV.5 :** Interface d'installation ADT.

### IV.3.3 Android

Android est un système d'exploitation open source pour Smartphones, PDA et terminaux mobiles. Il s'agit également d'un langage de programmation basé sur le JAVA et sur le XML. Le JAVA permet d'interagir avec l'utilisateur en faisant la liaison entre l'interface graphique et la base de données. Quant au XML, il permet notamment de décrire les interfaces graphiques.

## IV.3.3.1 Les étapes de création d'un nouveau projet

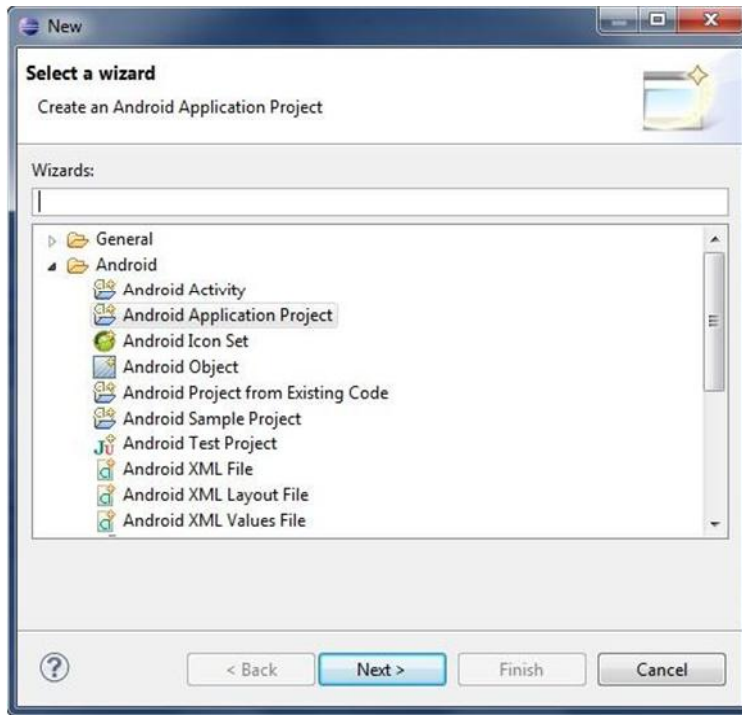


Figure IV.6: select a wizard

Nous lançons Eclipse et nous allons sous file -> New -> Other -> Android Application Project, vous Obtenez la figure suivante :

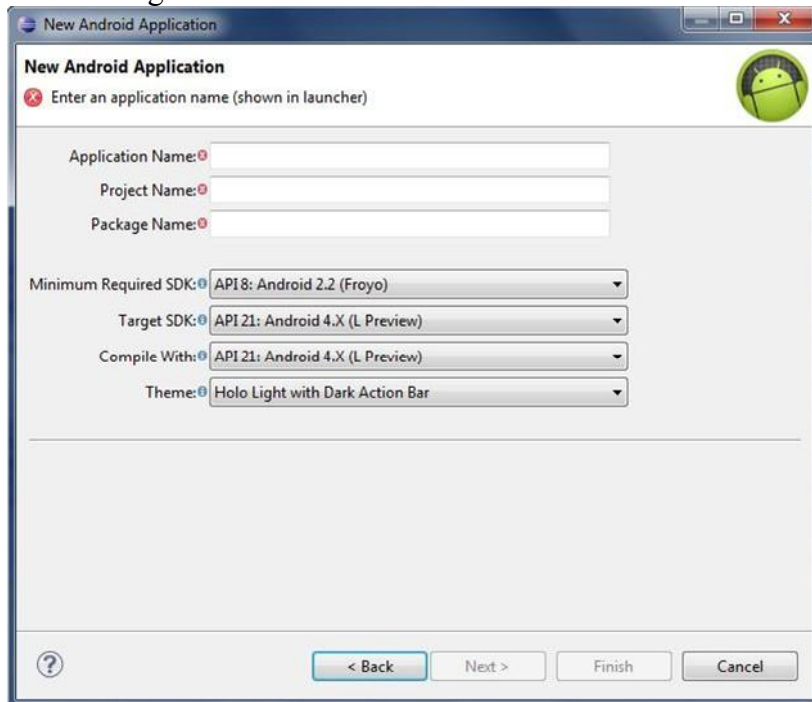
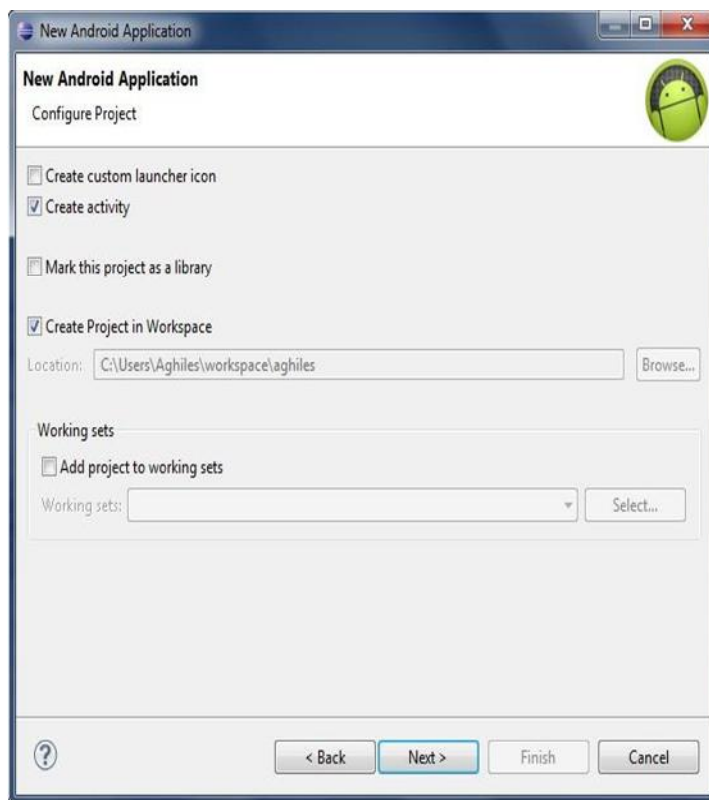


Figure IV.7 : Création d'un nouveau projet Android

Vous remplissez les champs suivants :

- « Project Name » : est le nom de votre projet pour Eclipse.
- « Application Name » : le vrai nom de votre application.
- « Package Name » : package ou l'application sera sauvegarder.
- « Minimum Required SDK » : est la version minimale pour laquelle votre application est destinée.
- « Target SDK » : est la version maximale sous laquelle votre application fonctionne.
- « Compile With » : la version du SDK avec laquelle sera compilée l'application, c'est la même version que celle de l'AVD créée.
- « Thème » : c'est le thème de l'application, puisque aucun des thèmes proposés n'est disponible, le seul choix possible est « None » qui ne signifie aucun thème.

Ensuite cliquez sur « Next » pour passer à l'écran suivant:



**Figure IV.8** : Configuration du projet

Cet écran contient beaucoup d'options, mais on va citer que deux :

- La première, intitulée « Create custom launcher icon », un outil pour vous aider à construire une icône pour votre application à partir d'une image préexistante, cette case ne nous intéresse pas donc vous allez la décocher.
- La seconde, celle qui s'appelle « Create activity », permet de vous faciliter le développement de l'application en faisant faire une partie par Eclipse.

Pour passer à la page suivante, cliquez sur « Next » :



**Figure IV.9:** Création d'une activité

Il s'agit ici d'un outil qui vous demande si vous voulez qu'Eclipse crée une activité pour vous, et si oui à partir de quelle mise en page. Vous allez déclarer que vous voulez qu'il crée une activité, cliquez sur la case à gauche de « Create Activity », et vous allez sélectionner «BlankActivity ».

Cliquez à nouveau sur « Next »:



**Figure IV.10:** crée une nouvelle activité en blanc avec une barre d'action

- « Activity Name » : permet d'indiquer le nom de la classe Java qui contiendra votre activité, ce champ doit donc respecter la syntaxe Java standard.
- « Layout Name » : renseignera sur le nom du fichier qui contiendra l'interface graphique qui correspondra à cette activité.
- « Navigation Type » : il permet de définir facilement comment s'effectueront les transitions entre plusieurs activités, mais il est préférable de le laisser tel qu'il est par défaut.

Vous cliquez sur finish et le projet sera créé selon une certaine structure :

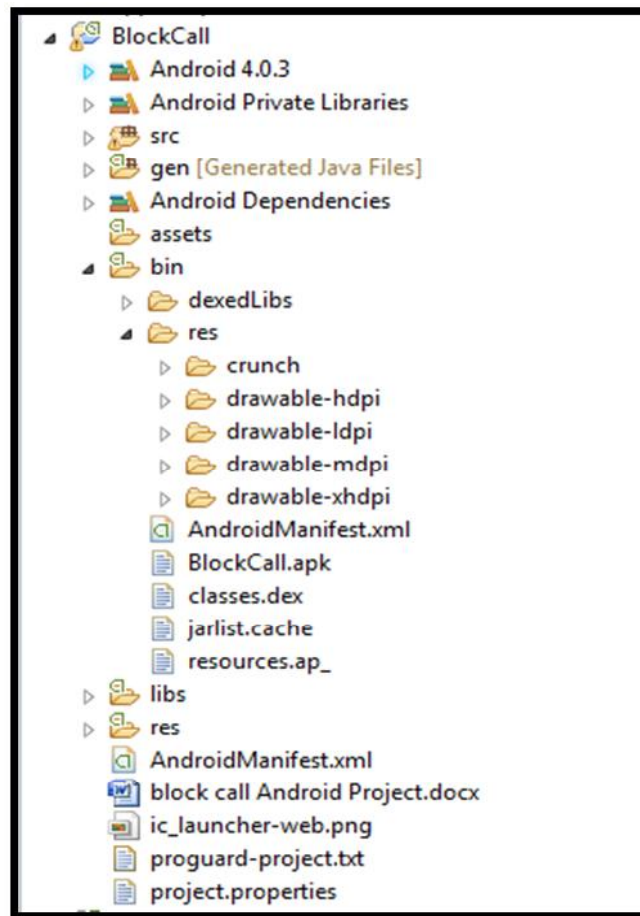
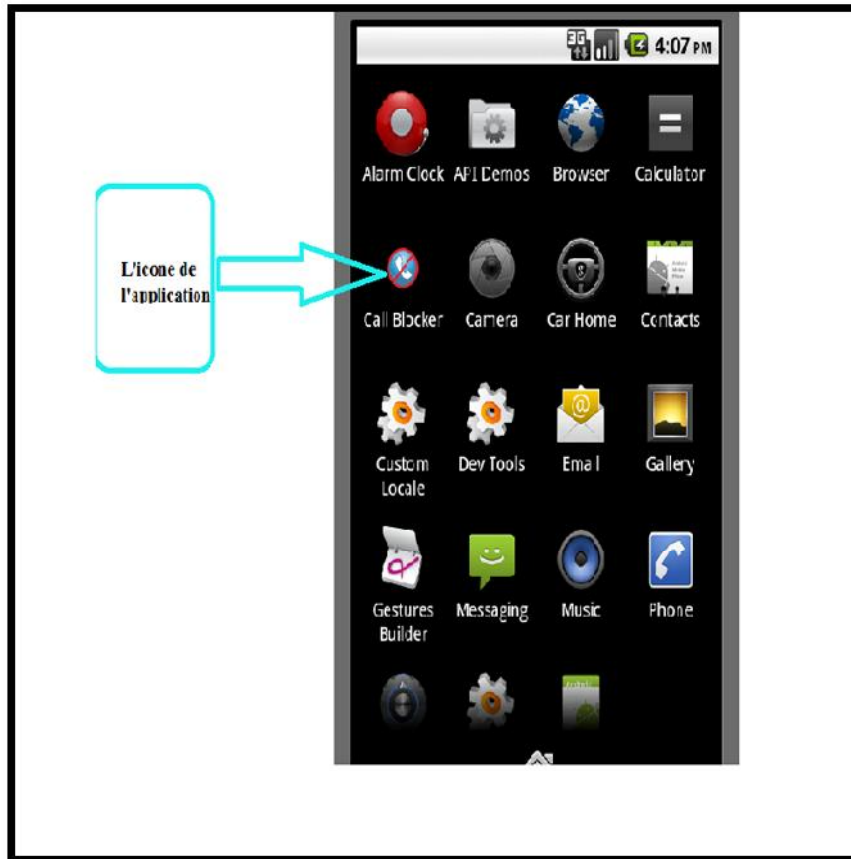


Figure IV.11: Arborescence du projet

## IV.4 Les différentes interfaces de notre application

### IV.4.1 Lancement de l'application

En premier lieu l'utilisateur lance l'application en appuyant sur l'icône lui correspondant dans le menu du Smartphone



**Figure IV.12 :** L'icone de l'application

#### IV.4.2 Le menu principal de l'application

Après un appui sur l'icône, on aura ceci :



**Figure IV.13:** Menu principal de l'application

L'utilisateur dispose de 4 boutons radio :

1. **Tout les appels**
2. **Numéro non enregistré**
3. **liste**
4. **Annuler tout**

et de 2 boutons :

1. **Ajouter un numéro**
2. **Voir liste**

### IV.4.3 Interface d'ajout d'un numéro

En appuyant sur le bouton « Ajouter un numéro » on aura ceci :



**Figure IV.14** : Interface d'ajout d'un numéro

Pour ajouter un numéro il suffit de le saisir et de cliquer sur le bouton « Ajouter»

#### IV.4.4 Interface d'affichage de la liste des numéros indésirables:

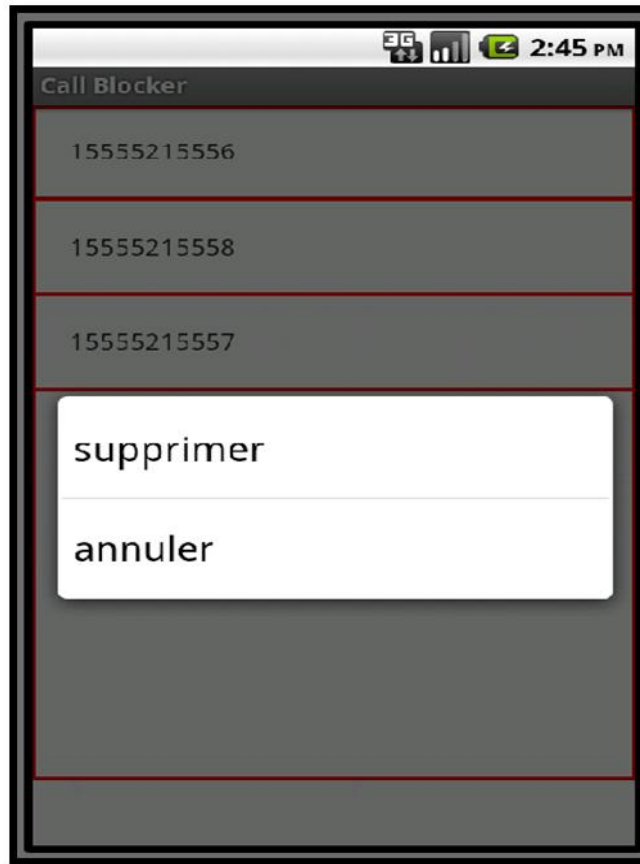
Après avoir cliquer sur le bouton « Voir liste » on aura ceci :



**Figure IV.15:** Interface d'affichage de la liste des numéros indésirables.

### IV.4.5 Interface de suppression d'un numéro :

Après un clique long su un numéro on aura ceci :



**Figure IV.16:** Interface de suppression d'un numéro

En cliquant sur supprimer, le numéro sélectionné va être supprimé de la liste.

### IV.5 Conclusion

Ce quatrième chapitre a été consacré à la phase de réalisation de notre application. On y a défini tout d'abord l'environnement de développement, puis illustré les différentes interfaces qui la composent.

# **Conclusion générale**

### Conclusion générale

Nous avons, au cours de ce mémoire, conçu et réalisé une application Android pour bloquer les appels indésirables

Pour cela nous avons, dans un premier temps commencé par présenter des généralités sur les réseaux sans fil ainsi que leurs évolution , ensuite nous avons fait une étude de l'état de l'art d'Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture d'Android, à savoir les principaux composants du système.

Après avoir entamé l'analyse et la conception de notre système en utilisant le langage de modélisation UML ainsi que la conception, nous sommes passés à la réalisation.

Au cours de la réalisation de ce projet nous avons enrichis notre savoir et développé nos connaissances informatiques, notamment dans le domaine de la programmation et de la mobilité. En effet, l'application a exigé des connaissances du langage JAVA et des outils de développement indispensables à sa réalisation. La mise en œuvre de notre travail a exigé des connaissances très approfondies en la matière ainsi qu'une bonne maîtrise de la configuration d'éclipse et de l'environnement Android.

Et enfin, notre travail ainsi présenté reste un prototype sur lequel nous espérons apporter plus de connaissances pour le rendre meilleur et plus fonctionnel, ainsi nous aimerions améliorer ce qui suit :

- Ajouter le filtrage des SMS.
- Exploiter d'autres Smartphones: Appel, BlackBerry....etc.

## Bibliographie

- [1] : [hp ://www.commentcamarche.net/contents/1309-reseaux-sans-fil-wireless-networks](http://www.commentcamarche.net/contents/1309-reseaux-sans-fil-wireless-networks)
- [2] : Tout sur les réseaux sans fil Broché – 22 avril 2009 de Fabrice Lemainque
- [3] : [hp ://www.commentcamarche.net/contents/1309-reseaux-sans-fil-wireless-networks](http://www.commentcamarche.net/contents/1309-reseaux-sans-fil-wireless-networks)
- [4] : [hps ://fr.wikipedia.org/wiki/R%C3%A9seau\\_sans\\_fil](https://fr.wikipedia.org/wiki/R%C3%A9seau_sans_fil)
- [5] : [hp ://www.bestcours.com/45-cours-reseaux-transmission-sans-fil.html](http://www.bestcours.com/45-cours-reseaux-transmission-sans-fil.html)
- [6] : <https://fr.wikipedia.org/wiki/Android>
- [7] : Mémoire promotion 2014/2015 Conception et réalisation d'une application Android « filtrage de SMS indésirable » réalisé par Aghiles MATOUK. Dirigé par Mme Haddaoui .
- [8] : Développer une application android [Sylvain Hébuterne](#) (Auteur) - Paru le 11 septembre 2015 - Manuel(broché)
- [9] : <https://openclassrooms.com/courses/creez-des-applications-pour-android/l-architecture-d-android>
- [10] : <http://javamind-fr.blogspot.com/2012/05/android-cycle-de-vie-des-activites-5x.html>
- [11] : <http://www.objjs.com/formation-java/tutoriel-android-installation-sdk-android.html>
- [12] : [https://fr.wikipedia.org/wiki/Java\\_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))
- [13] : <http://gadget-tendance.fr/2012/03/systeme-dexploitation-android-avantages-et-inconvenients/>
- [14] : UML Outil du génie logiciel /Mlle L.Mahdaoui , Mme N.GHENAIT/Pages Bleus /Septembre2007
- [15] : Conception et réalisation d'une application Mobile sous ANDROID, réalisé par Mr BELGACEM Kehir eddine et Mr BCHIRI Nouredine. Proposé par: Mr Y.CHAIEB
- [16] : UML2 pour une modélisation orientée objet /S.Graine/L'abeille 2009

# **Annexe**

## **Le langage de modélisation Unifié ( UML):**

### ➤ **Définition d'un diagramme**

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. C'est une perspective du modèle, pas "le modèle".

Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis). Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d'un système). Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.

Par extension et abus de langage, un diagramme UML est aussi un modèle (un diagramme modélise un aspect du modèle global) .

### ➤ **Caractéristiques des diagrammes UML**

Les diagrammes UML supportent l'abstraction. Leur niveau de détail caractérise le niveau d'abstraction du modèle. La structure des diagrammes UML et la notation graphique des éléments de modélisation est normalisée (document "UML notation guide"). Rappel : la sémantique des éléments de modélisation et de leur utilisation est définie par le méta modèle UML (document "UML semantics").

### ➤ **Les différents types de diagrammes UML**

Il existe 2 types de vues du système qui comportent chacune leurs propres diagrammes :

Les vues statiques :

- Diagrammes de cas d'utilisation .
- Diagrammes d'objets .
- Diagrammes de classes .
- Diagrammes de composants .
- Diagrammes de déploiement .

Les vues dynamiques :

- Diagrammes de collaboration
- Diagrammes de séquence
- Diagrammes d'états-transitions
- Diagrammes d'activités

## ➤ Les cas d'utilisation :

Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système

sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.

La détermination et la compréhension des besoins sont souvent difficiles car les intervenants sont noyés sous de trop grandes quantités d'informations : il faut clarifier et organiser les besoins des clients (les modéliser). Pour cela, les cas d'utilisation identifient les utilisateurs du système (acteurs) et leurs interactions avec le système. Ils permettent de classer les acteurs et structurer les objectifs du système.

Une fois identifiés et structurés, ces besoins :

- Définissent le contour du système à modéliser (ils précisent le but à atteindre).
- Permettent d'identifier les fonctionnalités principales (critiques) du système.

Les use cases ne doivent donc en aucun cas décrire des solutions d'implémentation.

Leur but est justement d'éviter de tomber dans la dérive d'une approche fonctionnelle, où l'on liste une litanie de fonctions que le système doit réaliser.

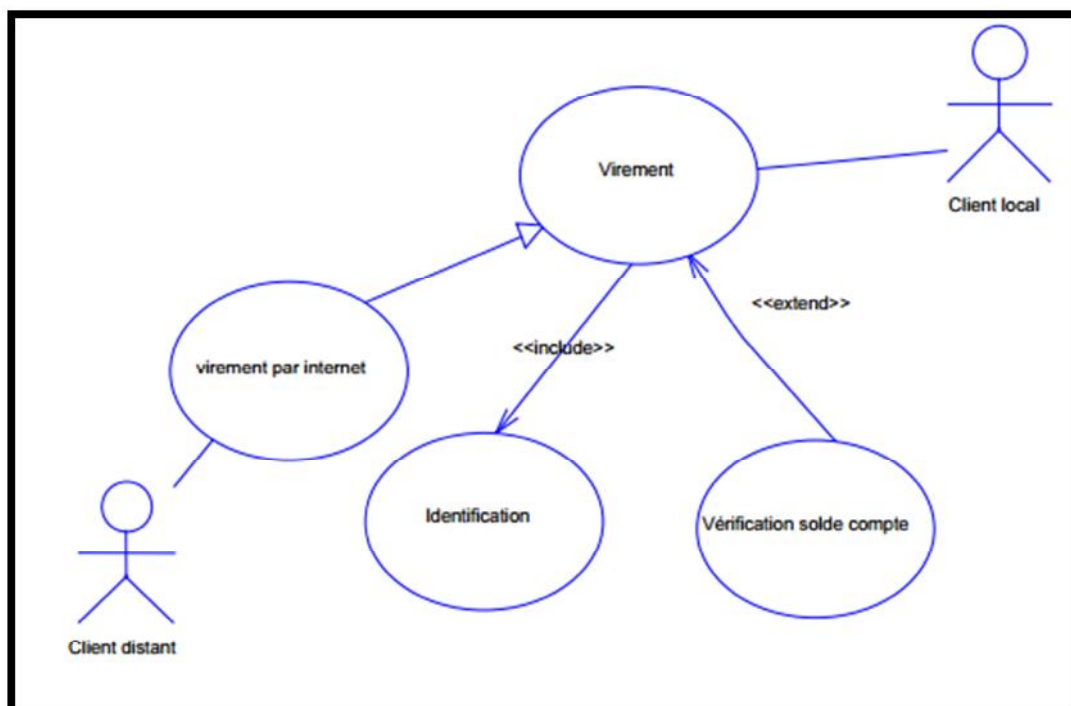


Figure 1 : Exemple de cas d'utilisation

## L'acteur :

La première étape de modélisation consiste à définir le périmètre du système, à définir le contour de l'organisation et à le modéliser. Toute entité qui est en dehors de cette organisation et qui interagit avec elle est appelé acteur selon UML.



**Figure 2 :** Nom de l'acteur

Un acteur est un type stéréotypé représentant une abstraction qui réside juste en dehors du système à modéliser.

Un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système. (la même personne physique peut donc être représentée par plusieurs acteurs en fonction des rôles qu'elle joue).

Pour identifier les acteurs, il faut donc se concentrer sur les rôles joués par les entités extérieures au périmètre.

Dans UML, il n'y a pas de notion d'acteur interne et externe. Par définition, un acteur est externe au périmètre de l'étude, qu'il appartienne ou pas à la société.

Enfin, un acteur n'est pas nécessairement une personne physique : il peut être un service, une société, un système informatique ...

Il existe 4 catégories d'acteurs :

- Les acteurs principaux : les personnes qui utilisent les fonctions principales du système .
- Les acteurs secondaires : les personnes qui effectuent des tâches administratives ou de maintenance.
- Le matériel externe : les dispositifs matériels incontournables qui font partie du domaine de l'application et qui doivent être utilisés.

- Les autres systèmes : les systèmes avec lesquels le système doit interagir.

## **La relation**

Elle exprime l'interaction existant entre un acteur et un cas d'utilisation.

Il existe 3 types de relations entre cas d'utilisation :

- La relation de généralisation .
- La relation d'extension .
- La relation d'inclusion .

## **La relation de généralisation**

Dans une relation de généralisation entre 2 cas d'utilisation, le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.

## **La relation d'inclusion**

Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. L'inclusion a un caractère obligatoire, la source spécifiant à quel endroit le cas d'utilisation cible doit être inclus. Cette relation permet ainsi de décomposer des comportements et de définir des comportements partageables entre plusieurs cas d'utilisation.

## **La relation d'extension**

Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à condition. Le comportement ajouté est inséré au niveau d'un point d'extension défini dans le cas d'utilisation destination. Cette relation permet de modéliser les variantes de comportement d'un cas d'utilisation (selon les interactions des acteurs et l'environnement du système).

## **Les scénarios**

Un cas d'utilisation est une abstraction de plusieurs chemins d'exécution. Une instance de cas d'utilisation est appelée : « scénario ».

Chaque fois qu'une instance d'un acteur déclenche un cas d'utilisation, un scénario est créé (le cas d'utilisation est instancié). Ce scénario suivra un chemin particulier dans le cas d'utilisation.

Un scénario ne contient pas de branche du type « Si condition ... alors » car pendant l'exécution, la condition est soit vraie, soit fausse, mais elle aura une valeur.

Après la description des cas d'utilisation, il est nécessaire de sélectionner un ensemble de scénarios qui vont servir à piloter l'itération en cours de développement.

Le choix et le nombre de scénarios à retenir est une étape difficile à réaliser : l'exhaustivité est difficile, voire impossible à atteindre. Le nombre d'instances pour un cas d'utilisation peut être très important, voire infini.

Les scénarios peuvent être classés en :

- scénarios principaux : il correspond à l'instance principal du cas d'utilisation. C'est souvent le chemin « normal » d'exécution du cas d'utilisation qui n'implique pas d'erreurs.
- Scénarios secondaires : il peut être un cas alternatif (un choix), un cas exceptionnel ou une erreur.

Les scénarios sont utiles pour :

- Analyser et concevoir le système .
- Justifier les choix effectués (ils serviront à la documentation des cas d'utilisation) .
- Tester : les scénarios constituent le meilleur moyen de spécifier les tests.

## ➤ Diagrammes de séquence

Le diagramme de séquence est une variante du diagramme de collaboration.

Par opposition aux diagrammes de collaboration, les diagrammes de séquence possèdent intrinsèquement une dimension temporelle mais ne représente pas explicitement les liens entre les objets. Ils privilégient ainsi la représentation temporelle à la représentation spatiale et sont plus aptes à modéliser les aspects dynamiques du système. En revanche, ils ne rendent pas compte du contexte des objets de manière explicite, comme les diagrammes de collaboration. Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

Les messages sont représentés comme dans le diagramme de collaboration. (NB : un message de retour sera représenté avec des traits en pointillés) .

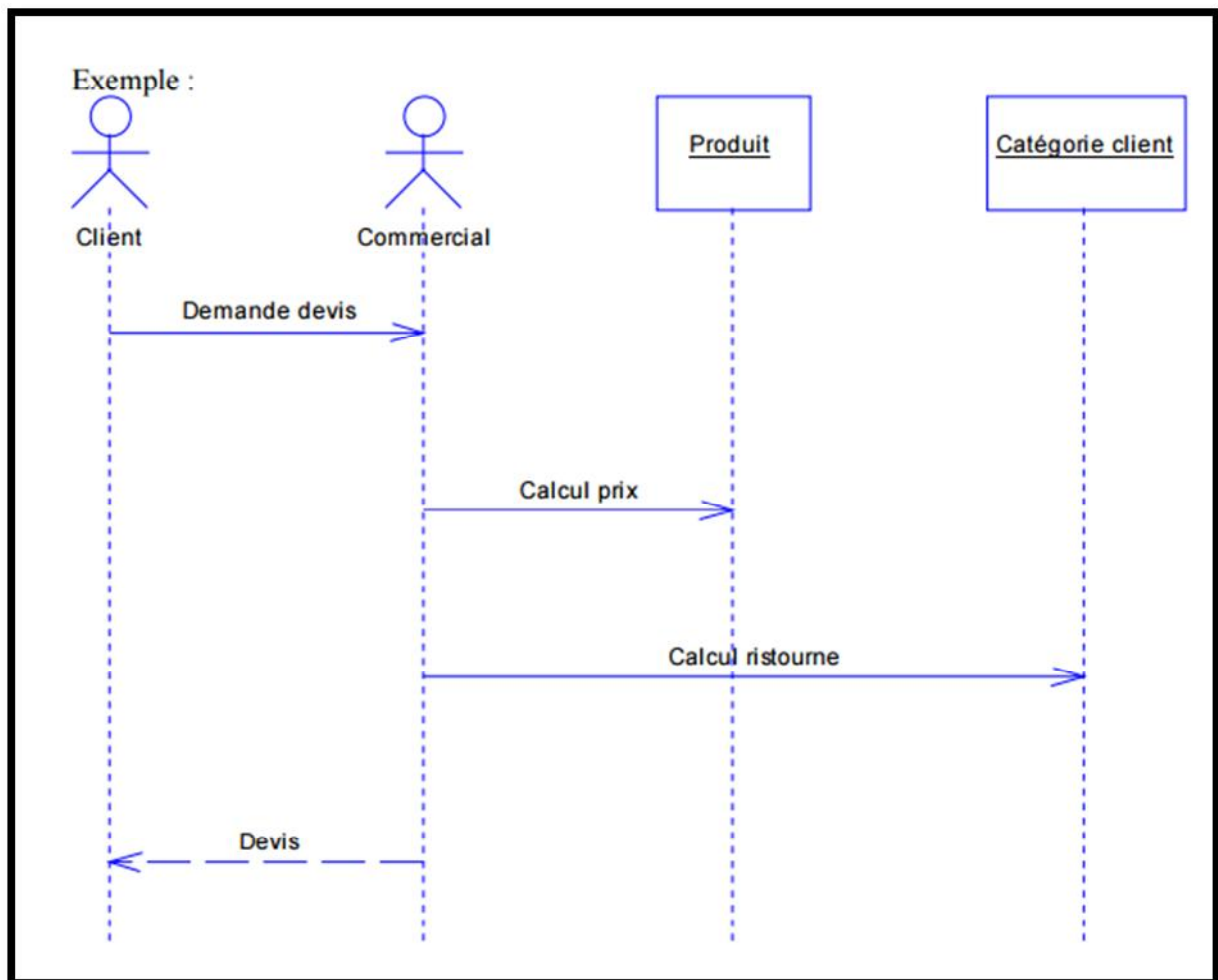


Figure 3 : exemple de diagramme de séquence

### Les interactions

L'interaction se traduit par l'envoi d'un message entre objets. Le diagramme de séquence insiste sur la chronologie des objets en utilisant la ligne de vie des objets.

### Les activations

Les diagrammes de séquence permettent de représenter les périodes d'activité des objets. Une période d'activité correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant .

### ➤ Diagrammes de classes

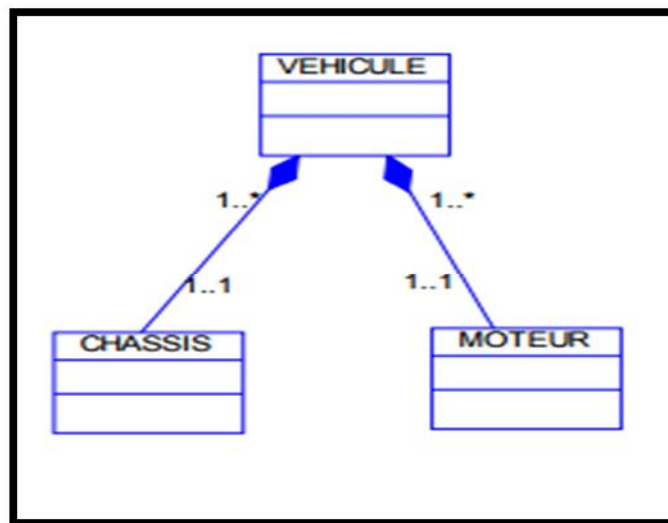
Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes.

L'intérêt du diagramme de classe est de modéliser les entités du système d'information.

Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles ont regroupées dans des classes. Le diagramme met en évidence d'éventuelles relations entre ces classes.

Le diagramme de classes comporte 6 concepts :

- Classe .
- Attribut .
- Identifiant .
- Relation .
- Opération .
- Généralisation / spécialisation .



**Figure 4:** Exemple de diagramme de classe

## La notion de classe

Définition : une classe est une description abstraite (condensée) d'un ensemble d'objets du domaine de l'application : elle définit leur structure, leur comportement et leurs relations.

## La notion d'attribut

Définition : Une classe correspond à un concept global d'information et se compose d'un ensemble d'informations élémentaires, appelées attributs de classe. Un attribut représente la modélisation d'une information élémentaire représentée par son nom et son format. Par commodité de gestion, on choisit parfois de conserver dans un attribut le résultat d'un calcul

effectué à partir d'autres classes : on parle alors d'attribut dérivé. Pour repérer un attribut dérivé : on place un / devant son nom.

## **La notion d'identifiant**

L'identifiant est un attribut particulier, qui permet de repérer de façon unique chaque objet, instance de la classe.

## **La notion d'opération**

Définition : l'opération représente un élément de comportement des objets, défini de manière globale dans la classe. Une opération est une fonctionnalité assurée par une classe. La description des opérations peut préciser les paramètres d'entrée et de sortie ainsi que les actions élémentaires à exécuter.

## **La notion de relation**

S'il existe des liens entre objets, cela se traduit nécessairement par des relations qui existent entre leurs classes respectives. Les liens entre les objets doivent être considérés comme des instances de relations entre classes. Il existe plusieurs types de relations entre classes :

- L'association .
- La généralisation/spécialisation .
- La dépendance .

## **Station de base (BTS):**

La station de base (BTS) est un ensemble d'émetteur récepteur qui assure le contrôle du dialogue radioélectrique entre le mobile et le réseau, elle gère les problèmes liés à la transmission radioélectrique (modulation, démodulation, égalisation, codage et correction d'erreurs, ...) d'un réseau. Elle fournit un point d'entrée dans le réseau aux abonnés présents dans sa cellule pour recevoir ou transmettre des appels.

## **Les contrôleurs des stations de base (BSC):**

Le contrôleur de stations de base est l'organe intelligent du sous système radio. Il gère une ou plusieurs stations de base en communications et il remplit des différentes missions pour les fonctions de communications et d'exploitations. Pour le trafic abonné venant des stations de base, il commande l'allocation des canaux, utilise les mesures effectuées par la BTS pour contrôler les puissances d'émissions du mobile. Cette dernière est fonction de la distance mobile-émetteur de telle manière que la liaison soit toujours de bonne qualité.

Le contrôleur est aussi le relais pour les alarmes et les statistiques de base ainsi qu'une banque de données pour des versions logicielles et les données de configurations téléphoniques. Il gère et prend la décision de l'exécution du hand over. De plus c'est un commutateur qui réalise une concentration vers le BSC.

## **les commutateurs (MSC):**

Le MSC autrement appelé commutateur est un organe électromécanique capable d'assurer une liaison provisoire entre une ligne d'entrées et une ligne de sortie.

## **Débit:**

Un débit permet de mesurer le flux d'une quantité relative à une unité de temps au travers d'une surface quelconque. Son unité dérivée du SI est le mètre cube par seconde ( $m^3/s$ ).

Il s'agit d'une notion centrale dans une situation d'écoulement de fluide.

## **Wifi: (Wireless Fidelity )**

Le Wi-Fi est un ensemble de protocoles de communication sans fil régis par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11). Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, décodeur Internet, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux.

## **WPAN:**

Le réseau personnel sans fil (appelé également réseau individuel sans fil ou réseau domestique sans fil et noté WPAN pour Wireless Personal Area Network) concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines mètres. Ce type de réseau sert

généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPAN.

## **WIMAX:**

Le réseau personnel sans fil (appelé également réseau individuel sans fil ou réseau domestique sans fil et noté WPAN pour Wireless Personal Area Network) concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPAN.

## **GSM: (Global System for Mobile Communications )**

Est une norme numérique de seconde génération pour la téléphonie mobile. Le groupe de travail chargé de la définir a été établi en 1982 par la Conférence européenne des administrations des postes et télécommunications (CEPT).

## **GPRS: (General Packet Radio Service )**

Le General Packet Radio Service ou GPRS est une norme pour la téléphonie mobile dérivée du GSM et complémentaire de celui-ci, permettant un débit de données plus élevé. On le qualifie souvent de 2,5G ou 2G+. Le G est l'abréviation de génération et le 2,5 indique que c'est une technologie à mi-chemin entre le GSM (deuxième génération) et l'UMTS (troisième génération).

## **UMTS: (Universal Mobile Telecommunications System )**

L'Universal Mobile Telecommunications System (UMTS) est l'une des technologies de téléphonie mobile de troisième génération (3G). Elle est basée sur la technologie W-CDMA, standardisée par le 3GPP et constitue l'implémentation dominante, d'origine européenne, des spécifications IMT-2000 de l'UIT pour les systèmes radio cellulaires 3G.

L'UMTS est parfois aussi appelé 3GSM, soulignant la filiation qui a été assurée entre l'UMTS et le standard GSM auquel il succède. Elle est également appelée 3G, pour troisième génération.

## **HomeRF :**

est une spécification de réseau sans fil (Shared Wireless Access Protocol-SWAP) permettant à des périphériques domestiques d'échanger des données entre eux.

**ZigBee** : est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme IEEE 802.15.4 pour les réseaux à dimension personnelle (Wireless Personal Area Networks : WPAN).

**HiperLAN** :(ou High Performance radio LAN) est un standard européen de télécommunications créé par l'ETSI (European Telecommunications Standards Institute) et développé par le groupe technique BRAN (Broadband Radio Access Networks). Ce standard est une alternative au groupe de normes IEEE 802.11 plus connu sous la dénomination Wi-Fi.

## **DECT :**

Norme de téléphonie sans-fil numérique définie par l'Institut Européen des Normes de Télécommunication. Un poste téléphonique de base est relié par la technologie DECT à d'autres postes, leur permettant de dialoguer en interne comme en externe avec une transmission fiable et de bonne qualité.

Le rayon des derniers téléphones de norme DECT est en moyenne de 300 mètres. Cette technologie est aujourd'hui concurrencée par le Wifi.

## **FDMA :**

L'accès multiple par répartition en fréquence (ou AMRF, en anglais Frequency Division Multiple Access ou FDMA) est un mode de multiplexage destiné à la téléphonie mobile. Il s'agit d'un découpage en bandes de fréquences de manière à attribuer une partie du spectre à chaque utilisateur. De cette manière, chaque utilisateur se voit attribuer une ou plusieurs bandes de fréquences distinctes (au minimum, une pour l'émission et une pour la réception si le mode duplex FDD est utilisé).

## **ETACS :**

Réseau analogique de téléphonie mobile développé au R-U.

## **GPS :**

Le **Global Positioning System (GPS)** est un système de géolocalisation fonctionnant au niveau mondial et reposant sur l'exploitation de signaux radio émis par des satellites dédiés. En 2011, il est avec GLONASS, un système de positionnement par satellites entièrement opérationnel et accessible au grand public.

Ce système est mis en place par le département de la Défense des États-Unis à des fins militaires. Il est très rapidement apparu que des signaux transmis par les satellites pouvaient être librement reçus et exploités, et qu'ainsi un récepteur pouvait connaître sa position sur la

surface de la Terre, avec une précision sans précédent, dès l'instant qu'il était équipé des circuits électroniques et du logiciel nécessaires au traitement des informations reçues. Une personne munie de ce récepteur peut ainsi se localiser et s'orienter sur terre, sur mer, dans l'air ou dans l'espace au voisinage proche de la Terre.

## **C++ :**

Est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plates-formes matérielles et de systèmes d'exploitation .

## **HTC Corporation :**

Anciennement High Tech Computer Corporation, est un fabricant taïwanais qui fabrique des smartphones destinés au départ à la plate-forme Windows Mobile de Microsoft et depuis 2008, à Android, le système d'exploitation open-source de Google mais aussi à Windows Phone.

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

Java 2 Standard Edition : est l'édition maitresse (ou distribution) du framework 'Plate-forme Java' d'Oracle, destinée typiquement aux applications pour poste de travail .

## **JVM :**

La machine virtuelle Java JVM est un appareil informatique fictif qui exécute des programmes compilés sous forme de bytecode Java.

## **JCP :**

Le Java Community Process (JCP) est une organisation créée par Sun en 1998. Son but est de coordonner l'évolution du langage Java et des technologies qui lui sont associées (voir la plateforme Java). Le JCP s'est doté d'un processus formalisé permettant aux différentes parties intéressées d'être impliquées dans la définition des futures versions et fonctionnalités de la plateforme Java.

## **APK :**

Un fichier Android Package (ou APK) est un format de fichiers pour Android. Un APK (ex. : "nomfich.apk") est une collection de fichiers ("package") compressée pour le système d'exploitation Android. L'ensemble constitue un « paquet ».

## **XML :**

L'Extensible Markup Language (XMLnote 1, « langage à balise extensible » en français) est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés, une application XML respecte généralement certains principes :

La structure d'un document XML est définie et validable par un schéma ;

Un document XML est entièrement transformable dans un autre document XML.

## **L'Object Management Group OMG :**

Est une association américaine à but non lucratif créée en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes.

L'OMG est notamment à la base des standards UML (Unified Modeling Language), MOF (Meta-Object Facility), CORBA (Common Object Request Broker Architecture) et IDL (Interface Definition Language).

L'OMG est aussi à l'origine de la recommandation MDA (Model Driven Architecture) ou ingénierie dirigée par les modèles, avec en particulier le langage standardisé de transformation de modèles QVT (Query/View/Transformation).

## **Diagramme :**

Un diagramme est une représentation visuelle simplifiée et structurée des concepts, des idées, des constructions, des relations, des données statistiques, de l'anatomie etc. employé dans tous les aspects des activités humaines pour visualiser et éclaircir la matière. Un diagramme permet aussi de décrire des phénomènes, de mettre en évidence des corrélations en certains facteurs ou de représenter des parties d'un ensemble.

## **NDK (Native Development Kit) :**

Le NDK (Native Development Kit) est un outil complémentaire du SDK qui permet de construire des portions critiques du code afin de mieux gérer les performances de votre application.

**Les objets (diagrammes) :** ils apparaissent dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.

**Les messages (diagrammes) :** ils sont représentés par des flèches directionnelles. Au-dessus des flèches directionnelles figurent un texte nous informant du message envoyé entre les objets.

### **Un acteur (UML)**

Un acteur est une entité externe qui interagit avec le système (opérateur, centre distant, autre système...).

### **Merise :**

Merise est une méthode d'analyse, de conception et de gestion de projet informatique.

Merise a été très utilisée dans les années 1970 et 1980 pour l'informatisation massive des organisations. Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. Elle est en revanche moins adaptée aux projets transverses aux organisations, qui gèrent le plus souvent des informations à caractère sociétal (environnemental et social) avec des parties prenantes.

### **Interface de programmation API :**

En informatique, une interface de programmation (souvent désignée par le terme API pour Application Programming Interface) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

Dans l'industrie contemporaine du logiciel, les applications informatiques se servent de nombreuses interfaces de programmation, la programmation se fait en réutilisant des briques de fonctionnalités fournies par des logiciels tiers. Cette construction par assemblage nécessite pour le programmeur de connaître la manière d'interagir avec les autres logiciels, qui dépend de leur interface de programmation. Le programmeur n'a pas besoin de connaître les détails de la logique interne du logiciel tiers, et celle-ci n'est généralement pas documentée par le fournisseur.

Des logiciels tels que les systèmes d'exploitation, les systèmes de gestion de base de données, les langages de programmation, ou les serveurs d'applications comportent une interface de programmation.

## **Smartphone :**

Un smartphone, ordiphone ou téléphone intelligent, est un téléphone mobile évolué disposant des fonctions d'un assistant numérique personnel, d'un appareil photo numérique et d'un ordinateur portable. La saisie des données se fait le plus souvent par le biais d'un écran tactile ou, plus rarement d'un clavier ou d'un stylet. Selon le principe d'un ordinateur, il peut exécuter divers logiciels/applications grâce à un système d'exploitation spécialement conçu pour mobiles, et donc en particulier fournir des fonctionnalités en plus de celles des téléphones mobiles classiques comme : l'agenda, la télévision, le calendrier, la navigation sur le Web, la consultation et l'envoi de courrier électronique, la géolocalisation, le dictaphone/magnétophone, la calculatrice, la boussole, l'accéléromètre, le gyroscope, la messagerie vocale visuelle, la cartographie numérique etc. Les appareils les plus sophistiqués bénéficièrent rapidement de la reconnaissance vocale et de la synthèse vocale.

## **Widget :**

Widget (Widget, the World Watcher) est une série télévisée d'animation américaine en 65 épisodes de 25 minutes, créée par Peter Keefe et diffusée entre le 29 septembre 1990 et le 12 avril 1993 en syndication.

En France, la série a été diffusée à partir du 25 janvier 1992 sur Canal+ dans l'émission Canaille Peluche.

Rediffusion à partir de 1993 dans l'émission Big Bang sur France 3.

Au Québec sur Canal Famille.

## **Tablette :**

Une tablette tactile<sup>2</sup>, tablette électronique<sup>3</sup>, ardoise électronique<sup>4</sup>, tablette numérique<sup>5</sup>, ou tout simplement tablette, est un ordinateur portable ultraplats qui se présente sous la forme d'un écran tactile sans clavier et qui offre à peu près les mêmes fonctionnalités qu'un ordinateur personnel<sup>3,2</sup>. Elle permet d'accéder à des contenus multimédias tels que la télévision, la navigation sur le web, la consultation et l'envoi de courrier électronique, l'agenda, le calendrier et la bureautique simple. Il est possible d'installer des applications supplémentaires depuis une boutique d'applications en ligne. En quelque sorte, la tablette tactile est un intermédiaire entre l'ordinateur portable et le smartphone.

## **Clavier virtuel :**

Un clavier virtuel (ou clavier visuel dans l'environnement Microsoft Windows) est un logiciel permettant à l'utilisateur de saisir des caractères. Il peut être utilisé avec plusieurs dispositifs de pointage, comme une souris, ou un écran tactile. Il se substitue au clavier physique.

Un clavier virtuel peut prendre deux formes : Desktop ou Web. Dans le premier cas, il est intégré par défaut au système d'exploitation, tandis que dans le second, il est exécuté au sein du navigateur web, sans téléchargement ni installation de logiciel.

## **Open Source :**

L'Open Source détermine les conditions nécessaires pour qu'une licence soit considérée comme libre par l'Open Source Initiative. Elle est publiée sur le site de l'OSI avec une liste de licences approuvées .

## **Microsoft Windows :**

Windows (littéralement « Fenêtres » en anglais) est au départ une interface graphique unifiée produite par Microsoft, qui est devenue ensuite une gamme de systèmes d'exploitation à part entière, principalement destinés aux ordinateurs compatibles PC .

## **Linux :**

Linux est le nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau Linux. C'est une implémentation libre du système UNIX respectant les spécifications POSIX. Ce système est né de la rencontre entre le mouvement du logiciel libre et le modèle de développement collaboratif et décentralisé via Internet. Son nom vient du créateur du noyau Linux, Linus Torvalds. La Free Software Foundation préfère utiliser le nom GNU/Linux, ce qui a conduit à une certaine controverse.

Les systèmes basés sur Linux sont majoritaires pour les super-ordinateurs et les smartphones. Sur les serveurs informatiques, le marché est partagé avec les autres Unix et Windows. Il est largement utilisé comme système embarqué dans les appareils électroniques : télévision, modem, GPS, etc. Il reste en revanche utilisé par une minorité des ordinateurs personnels de bureau, sa plate-forme d'origine, bien que cela tende à augmenter.

Le système avec toutes ses applications est distribué sous la forme de distributions Linux comme Slackware, Debian ou Red Hat Enterprise Linux.

## **SQLite :**

SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID.

Contrairement aux serveurs de bases de données traditionnels, comme MySQL ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

D. Richard Hipp, le créateur de SQLite, a choisi de mettre cette bibliothèque ainsi que son code source dans le domaine public, ce qui permet son utilisation sans restriction aussi bien dans les projets open source que dans les projets propriétaires. Le créateur ainsi qu'une partie des développeurs principaux de SQLite sont employés par la société américaine Hwaci1.

SQLite est le moteur de base de données le plus distribué au monde, grâce à son utilisation dans de nombreux logiciels grand public comme Firefox, Skype, Google Gears, dans certains produits d'Apple, d'Adobe et de McAfee et dans les bibliothèques standards de nombreux langages comme PHP ou Python. De par son extrême légèreté (moins de 300 Kio<sup>2</sup>), il est également très populaire sur les systèmes embarqués, notamment sur la plupart des smartphones modernes : l'iPhone ainsi que les systèmes d'exploitation mobiles Symbian et Android l'utilisent comme base de données embarquée<sup>3</sup>. Au total, on peut dénombrer plus d'un milliard de copies connues et déclarées de la bibliothèque .

## **Bureau virtuel :**

Un bureau virtuel est un environnement graphique qui peut être démultiplié, afin de travailler à un seul type de tâche dans chaque bureau, au lieu d'avoir toutes les fenêtres réunies dans un seul. Par exemple, l'utilisateur choisira de réserver un bureau aux tâches de retouches photographiques, un autre à internet, un autre aux jeux, etc. Les fenêtres ne sont plus mélangées.

Une autre implémentation des bureaux virtuels, moins fréquemment utilisée, consiste à offrir un bureau plus grand que la taille de l'écran, la souris ou le clavier permettant de se déplacer dans les zones inaccessibles.

Les bureaux virtuels sont disponibles nativement sous de nombreux environnements de bureau de plusieurs systèmes d'exploitation, en particulier des systèmes de type UNIX

implémentant le système de fenêtrage X, tels que GNU/Linux. Les environnements de bureau les plus répandus proposent notamment cette fonctionnalité, activée par défaut : KDE, GNOME et XFCE. Parmi les plus spectaculaires, on trouve ceux de Compiz Fusion, où chaque bureau virtuel peut être la face d'un cube que l'on fera pivoter à l'écran pour accéder à chaque bureau.

Longtemps l'apanage des systèmes UNIX, il est possible depuis Windows XP d'utiliser des logiciels tiers, tels Yod'm 3D, pour reproduire le principe des bureaux virtuels sous Windows. Bien que système UNIX, Mac OS n'intègre cette fonctionnalité nativement que depuis Mac OS X 10.5 ; il fallait auparavant utiliser un logiciel tiers. Depuis OS X 10.7, la gestion des bureaux virtuels, appelés Espaces sur le Mac, est confié à un gestionnaire nommé Mission Control, qui remplace le gestionnaire précédent appelé Exposé<sup>1</sup>.

## **Le Universal Serial Bus USB :**

Le Universal Serial Bus (USB, en français Bus universel en série, dont le sigle, inusité, est BUS) est une norme relative à un bus informatique en transmission série qui sert à connecter des périphériques informatiques à un ordinateur. Le bus USB permet de connecter des périphériques à chaud (quand l'ordinateur est en marche) et en bénéficiant du Plug and Play (le système reconnaît automatiquement le périphérique). Il peut alimenter les périphériques peu gourmands en énergie (disques SSD en particulier). Apparue en 1996, ce connecteur s'est généralisé dans les années 2000 pour connecter souris, clavier d'ordinateur, imprimantes, clés USB et autres périphériques bon marché sur les ordinateurs personnels.

Les performances de l'USB, notamment concernant les débits, se sont grandement améliorées au fil des versions (USB 2.0, 3.0, 3.1...).

Pour plus de clarté, les débits dans cet article seront indiqués en octets et non en bits (pour rappel, 1 octet = 8 bits).

## **Bit :**

Le bit est l'unité la plus simple dans un système de numération, ne pouvant prendre que deux valeurs, désignées le plus souvent par les chiffres 0 et 1. Un bit ou élément binaire peut représenter aussi bien une alternative logique, exprimée par faux et vrai, qu'un chiffre du système binaire.

Dans la théorie de l'information, un bit est la quantité minimale d'information transmise par un message, et constitue à ce titre l'unité de mesure de base de l'information en informatique. La quantité d'information effectivement transmise s'exprime en shannons, et ne peut dépasser la taille du message en bits.

Les systèmes numériques traitent exclusivement des informations réduites en bits, en général associés dans des groupes de taille fixe appelés bytes .

## **Moniteur :**

Un moniteur est une approche pour synchroniser deux ou plusieurs tâches qui utilisent des ressources partagées, habituellement des appareils séparés (imprimantes) ou des ensembles de variables.

Autrefois, les systèmes d'exploitation se réduisaient à un simple programme moniteur enchainant l'exécution des autres (par exemple le DOS de l'IBM 360 à la fin des années 1960). Ce sens est aujourd'hui inusité et n'a rien à voir avec le sens actuel (synchronisation).

## **Serveur :**

Un serveur informatique est un dispositif informatique matériel ou logiciel qui offre des services, à différents clients. Les services les plus courants sont :

- l'accès aux informations du World Wide Web ;
- le courrier électronique ;
- le partage d'imprimantes ;
- le commerce électronique ;
- le stockage en base de données ;
- la gestion de l'authentification et du contrôle d'accès;
- le jeu et la mise à disposition de logiciels applicatifs (optique software as a service).

Un serveur fonctionne en permanence, répondant automatiquement à des requêtes provenant d'autres dispositifs informatiques (les clients), selon le principe dit client-serveur. Le format des requêtes et des résultats est normalisé, se conforme à des protocoles réseaux et chaque service peut être exploité par tout client qui met en œuvre le protocole propre à ce service.

Les serveurs sont utilisés par les entreprises, les institutions et les opérateurs de télécommunication. Ils sont courants dans les centres de traitement de données et le réseau Internet.

## **Android SDK:**

Le kit de développement (SDK) d'Android est un ensemble complet d'outils de développement. Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux. Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure. L'IDE officiellement supporté

était Eclipse combiné au plugin d'outils de développement d'Android (ADT), mais depuis 2015, Google officialise Android Studio qui devient alors l'IDE officiel pour le SDK Android. Les développeurs peuvent utiliser n'importe quel éditeur de texte pour modifier les fichiers Java et XML, puis utiliser les outils en ligne de commande (Java Development Kit et Apache Ant sont obligatoires) pour créer, construire et déboguer des applications Android ainsi que contrôler des périphériques Android (pour déclencher un redémarrage, installer un logiciel à distance ou autre)

## **Google play :**

Google Play est une boutique en ligne créée par Google (le 6 mars 2012) par fusion des services Android Market, Google Movies, Google ebookstore et Google Music. Elle regroupe une boutique d'application pour le système d'exploitation Android, une boutique de location de films et de série télévisées, une boutique d'achat de musique, de livres, de magazines et de la gamme de smartphones et tablettes « Nexus ».

## **Machine virtuelle Dalvik :**

**Dalvik** est une machine virtuelle destinée aux téléphones mobiles et tablettes tactiles, qui est incorporée dans le système d'exploitation Android. Dalvik est destiné à permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité (peu d'espace mémoire et peu de puissance de calcul).

Créé par Dan Bornstein, Dalvik fait partie du runtime, le moteur qui permet l'exécution des applications pour Android. Le nom est celui d'un village de pêcheurs en Islande, pays d'origine de son auteur<sup>1</sup>.

Dalvik offre une alternative à la machine virtuelle de la technologie Java. Les fichiers de bytecode de la technologie Java sont préalablement transformés et consolidés dans un fichier .dex (Dalvik Executable) en vue de leur utilisation par Dalvik.

- **Présentation :**

Le but d'une machine virtuelle comme Dalvik est de permettre d'exécuter le même programme sur une grande variété d'appareils, quelles que soient leurs caractéristiques techniques. Le code exécutable du programme (dit bytecode) est transformé à la volée en instructions spécifique à l'appareil sur lequel le programme est exécuté - c'est la fonction de compilation just-in-time de la machine virtuelle. Dalvik est un des composants clé d'Android. Il est destiné à permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité (peu d'espace mémoire et peu de puissance de calcul). La capacité ordinaire des appareils mobiles contemporains de 2011 étant équivalente à celle d'un ordinateur personnel vieux de dix ans

- **La technologie Java et Dalvik :**

Java est un langage de programmation destiné à être utilisé sur une grande variété d'appareils, selon la devise « write once, run anywhere » (en français écrire une fois, utiliser partout). Les programmes écrits dans ce langage sont traduits en bytecode qui sera ensuite exécuté par une machine virtuelle telle que celle de l'environnement d'exécution Java de Oracle. L'objectif de Dalvik est de fournir une machine virtuelle alternative, adaptée aux limitations des appareils mobiles.

Dalvik exécute un bytecode différent, dont les instructions sont basées sur des registres, alors que la machine virtuelle classique de la technique Java est basée sur la pile. Une machine à registres nécessite moins d'instructions pour effectuer les mêmes opérations qu'une machine à pile et est par conséquent mieux adaptée à un appareil ayant peu de puissance de calcul. En raison de cette différence, les fichiers de bytecode Java ordinaires ne peuvent pas être exécutés tel quel par Dalvik, et un programme inclus dans le SDK Android transforme au préalable ces fichiers en DEX (Dalvik Executable) - les fichiers de bytecode pour Dalvik<sup>3</sup>. Les fichiers exécutables pour Dalvik (.dex) sont créés par consolidation des fichiers de bytecode Java. La taille du fichier .dex est typiquement la moitié de la taille des fichiers de bytecode Java desquels il est issu.

La machine virtuelle Dalvik se sert des services offerts par le noyau Linux. Dans Android il est également possible de créer des applications qui n'utilisent pas Dalvik, et qui se servent directement du noyau Linux, cependant cette technique offre peu d'avantages et est rarement utilisée.

Tous les services fournis par Android ainsi que les moyens matériels des appareils sont mis à disposition à travers Dalvik. Cette machine virtuelle joue le rôle d'écran qui cache les caractéristiques techniques de l'appareil sur laquelle elle est exécutée.

Dalvik est remplacé par Android\_Runtime à partir de la version 5 d'Android sortie en novembre 2014 (ART est disponible en version expérimentale dès Android 4.4). Cependant, à sa sortie, ART n'est pas compatible avec toutes les applications Android.

**CDMA :** (Code Division Multiple Access), utilisant une technique d'étalement de spectre permettant de diffuser un signal radio sur une grande gamme de fréquences.

**TDMA:** (Time Division Multiple Access), utilisant une technique de découpage temporel des canaux de communication, afin d'augmenter le volume de données transmis simultanément. La technologie TDMA est principalement utilisée sur le continent américain, en Nouvelle Zélande et en Asie Pacifique.

**ETACS :** (Extended Total Access Communication System) est une version améliorée du standard TACS développé au Royaume-Uni utilisant un nombre plus important de canaux de communication.