

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

En vue de l'obtention du diplôme de Master en informatique

Option Conduite de projet informatique

Thème

Conception et réalisation d'une application

Android « filtrage de SMS indésirable »

Réalisé par :

M^r: MATOUK AGHILES

Dirigé par :

M^{me}: HADDAOUI

Promotion 2014/2015

Remerciements

*- Nous remercions DIEU de nous avoir donné santé, courage, force et patience
pour réaliser ce travail.*

*- Nos vifs remerciements accompagnés de toute
notre gratitude vont tout d'abord à notre promotrice Mme HADDAOUI
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.*

*- Enfin, nos remerciements vont à toute personne
ayant contribué de près ou de loin à l'aboutissement
de ce modeste travail*

Dédicaces

Je dédie ce modeste travail :

*A celle qui a bercé mes rêves: **ma mère,***

*A celui qui a nourri mes ambitions: **mon père***

A ma chère sœur

A la mémoire de mes grands parents,

A celle qui m'a aidé tout le long de se travail.

A mes amis (es) qui m'ont aidé de près ou de loin.

.

Aghiles

Abréviation

Liste des tableaux

Liste des figures

Introduction Générale

Chapitre I : les Réseaux sans fils

Introduction	2
I.1 Réseaux sans fil	2
I.1.1 Définition du réseau sans fil	2
I.1.2 Caractéristiques des réseaux sans fil	2
I.1.3 Classification des réseaux sans fil	3
I.1.3.1 Classification selon l'étendue	3
I.1.3.1.1 Réseaux personnels sans fil (WPAN)	3
I.1.3.1.1.1 Les réseaux locaux sans fil (WLAN)	4
I.1.3.1.2 Les satellites de communication	5
I.1.3.2 Classification selon le mode de communication	6
I.1.3.2.1 Mode de communication avec infrastructure	6
I.1.3.2.2 Mode de communication sans infrastructure (ad-Hoc)	7
I.1.4 Evolution des systèmes cellulaires	7
I.1.4.1 Les réseaux de la première génération (1G)	7
I.1.4.2 les réseaux de la deuxième génération (2G)	8

Sommaire

I.1.4.3 Les réseaux de la troisième génération (3G)	9
I.1.4.4 les réseaux de la quatrième génération (4G)	11
I.1.4.4.1 Les applications de la quatrième génération	11
Conclusion	12

Chapitre II : Android

Introduction	13
II.1 Le système d'exploitation d'Android	13
II.1.1 Création d'Android	13
II.1.3 Bugdroid	13
II.1.4 Historique et évolution des versions d'Android	14
II.1.5 Les avantages d'Android	16
II.1.6 Présentation de la plateforme Android	17
II.1.6.1 Architecture d'Android	17
II.1.6.1.1 Les niveaux du système d'exploitation d'Android	18
II.1.6.1.1.1 Premier niveau : Linux Kernel	18
II.1.6.1.1.2 Deuxième niveau : Bibliothèque et environnement	18
II.1.6.1.1.3 Troisième niveau : Développement de l'application (Application Framework)	19
II.1.6.1.1.4 Quatrième niveau : Application	19
II.1.6.2 Les composants d'une application Android	20
II.1.6.2.1 L'activité (Activity)	20

Sommaire

II.1.6.2.1.1 Cycle de vie d'une activité.....	21
II.1.6.2.1.2 L'Etats d'une activité.....	22
II.1.6.2.1.3 Notification de l'état d'une activité	23
II.1.6.2.2 Content Provider	23
II.1.6.2.3 Service	23
II.1.6.2.4 Broadcast Receiver	23
 II.2 Développement d'application pour Android	24
II.2.1 SDK Android (Softwar Development Kit)	24
II.2.1.1 Les outils fournit par le SDK	25
II.2.1.1.1 AVD (Android Virtual Device)	25
II.2.1.1.2 L'ADB (Android Debu Bridg)	25
II.2.1.1.3 Le DDMS (Dalvik Debug Monitor Server)	26
 II.3 Concepts et innovation	26
II.3.1 Le bureau virtuel	26
II.3.2 Les notifications	27
II.3.3 SQLite	27
II.3.4 Android Market	27
 II.4 Présentation du langage java	28
II.4.1 historique	28
II.4.2 Les caractéristiques de JAVA.....	28

Sommaire

II.4.3 Outils de développement	29
Conclusion	29

Chapitre III : Conception et réalisation

Introduction	30
III.1.Présentation de l'UML	30
III.1.1.Modélisation avec l'UML	30
III.1.2.La démarche de modélisation avec l'UML	31
III.2.Conception.....	31
III.2.1.Les diagrammes des cas d'utilisation	32
III.2.2.Les diagramme de séquences	33
III.3 Réalisation	35
III.3.1 Présentation de notre application	35
III.3.2.Environnement de travail	35

Sommaire

III.3.2.1. Environnement matériel	35
III.3.2.2. Environnement cible	35
III.3.2.3. Environnement logiciel	35
III.3.2.3.1. Système d'exploitation	35
III.3.2.3.2. Langage de programmation	36
III.3.2.3.3. Eclipse	36
III.3.2.3.4. IDE Eclipse	36
III.3.3. Prise en main de l'environnement Android	37
III.3.3.1. Software Development Kit (SDK)	38
III.3.3.2. Le plugin ADT pour Eclipse	39
III.3.3.3. Android	40
III.3.3.3.1 Les étapes de création d'un nouveau projet	41
III.4.3. Mode de fonctionnement de l'application.....	45
III.4.3.1. Le menu principal de l'application	45
III.4.3.2. Interface d'ajout d'un numéro.....	45
III. 4.3.3. Interface d'affichage de la liste des numéros indésirable	46
III.4.3.4. Interface de suppression d'un numéro	47
Conclusion	48

Conclusion générale

Références bibliographiques

Liste des figures

Liste des figures

Numéro des figures	Titre des figures	pages
N°1	Réseaux personnelles sans fils (WPAN)	4
N°2	Les réseaux locaux sans fils (WLAN)	5
N°3	Les satellites de communications	6
N°4	Les réseaux sans fils avec infrastructure	6
N°5	Les réseaux ad-hoc	7
N°6	Logo d'Android	13
N°7	L'architecture du système Android	17
N°08	Différence entre la JVM et Dalvik	19
N°09	Les composants d'une activité	20
N°10	Cycle de vie d'une activité	21
N°11	Android SDK	24
N°12	L'émulateur	25
N°13	Bureau virtuel	26

N°14	Exemple de notification	27
N°15	La démarche adoptée pour la modélisation.	31
N°16	Diagramme de cas d'utilisation générale.	32
N°17	Diagramme de séquence de cas d'utilisation « Ajouter un nouveau numéro »	33
N°18	Diagramme de séquence de cas d'utilisation « Afficher la liste des numéro indésirable »	34
N°19	Diagramme de séquence de cas d'utilisation « Supprimer un numéro de la liste »	34
N°20	Diagramme de classe « Ajouter un nouveau numéro »	35
N°21	Diagramme de classe « Afficher la liste des numéro indésirable »	36
N°22	Diagramme de classe « Supprimer un numéro de la liste »	36
N°23	Logo Eclipse et site web.	38
N°24	plate-forme d'éclipse.	39
N°25	Portail des développeurs Android.	40
N°26	Interface d'installation du SDK Android.	41

N° 27	Interface d'installation ADT.	42
N° 28	Select a wizard	43
N° 29	Création d'un nouveau projet Android	43
N° 30	Configuration du projet	44
N° 31	Création d'une activité	45
N° 32	Crée une nouvelle activité en blanc avec une barre d'action	45
N° 33	Arborescence du projet	46
N° 34	Menu principal.	47
N° 35	Interface d'ajout d'un numéro.	48
N° 36	Interface d'affichage de la liste des numéros indésirable.	48
N°37. A	Interface de saisie d'un numéro à supprimer	49
N° 37. B	Interface de confirmation de la suppression.	49

Liste des tableaux

Liste des tableaux

Numéro des tableaux	Titre des tableaux	pages
N°1	L'évolution des versions Android	14-15
N°2	les différents états d'une activité et leurs caractéristiques	22

Abbreviations

2D : Deux dimensions

3D : Trois dimensions

1G: Première génération

2G: Deuxième génération

3G: Troisième génération

4G: Quatrième génération

A

ADB: Android Debug Bridge

ADT: Android Development Tools

API: Application Programming Interface

AVD: Android Virtual Device

B

BDD: Behavior-Driven Development

D

DDMS: Dalvik Debug Monitor Server

G

GSM: Global System for Mobile

GPRS: General Packet Radio Service

GPS: Global Positioning System

GHZ : Giga hertz

GO : Giga Octect

H

HTC: High Tech Computer

HTTP: HyperText Transfer Protocol

I

IDE : Environnement de Développement Intégré

J

JDK: Java Development Kit

JVM: Java Virtual Machine

JCP: Java Community Process

JSR: Java Specification Request

K

Kbps : kilobits per second

N

NDK: Native Development Kit

O

OHA: Open Handset Alliance

OMG: Object management groupe

P

PDA: Personal digital assistant

S

SD: Secure Digital

SDK: Software Development Kit

SQL: Structured Query Language

U

UMTS: Universal Mobile Telecommunications System

UML: Unified Modelling Language

W

WIFI: Wireless fidelity

WLAN: Wireless Local Area Network

WPAN: Wireless Personal Area Network

WIMAX: Worldwide Interoperability for Microwave Access

WORA: Write Once, Run Anywhere

X

XML: Extensible Markup Language

Introduction générale

Introduction générale

Depuis le commencement de l'informatique moderne, le besoin et l'envie récurrents de mettre en réseau les différents éléments informatisés que l'on peut avoir en sa possession se fait sentir. Cette mise en réseau peut se faire simplement pour échanger des messages (communication), pour échanger des données (stockage), ou encore pour partager la puissance de calcul. Les applications sont multiples et quasi illimitées. C'est dans cette optique qu'ont été déployés les premiers réseaux militaires, et universitaires, puis le plus grand d'entre eux : l'internet.

Les Smartphones sont des appareils extrêmement sophistiqués, qui fournissent des fonctionnalités en plus de celles des téléphones mobiles classiques comme la télévision, la navigation sur le web, la consultation et l'envoi de courrier électroniques, la messagerie vocale et visuelle, etc. Dernièrement, on a aussi vu les Smartphones sophistiqués bénéficiant rapidement de la reconnaissance et synthèse vocale. Les Smartphones exécutent tous divers logiciels et applications grâce à des systèmes d'exploitation spécialement conçus pour les mobiles. Les Smartphones peuvent être personnalisés en y installant des applications additionnelles telles que des jeux ou des utilitaires grâce aux magasins d'applications en ligne (stores).

Le marché de la téléphonie portable connaît actuellement une véritable révolution. Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en rachetant une startup travaillant sur un système d'exploitation ouvert pour terminal mobile : Android.

Dans le cadre de notre mémoire, il nous a été demandé de faire la conception, le développement et l'intégration d'une application embarquée Android qui permet a son utilisateur de filtrer les SMS indésirables.

Ce mémoire est composé de :

- Le premier Chapitre qui traite les réseaux sans fils.
 - Le deuxième Chapitre va être consacré à « Android ».
 - Le troisième Chapitre va illustrer la conception et la réalisation de notre travail.
- Enfin une conclusion générale qui achèvera notre mémoire.

Les Réseaux sans fils

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.1 Réseaux sans fil :

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. [5]

I.1.1 Définition du réseau sans fil

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

Selon le type des nœuds, on distingue : les réseaux de télécommunication dans lesquels les nœuds 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 nœuds sont les ordinateurs, les imprimantes, les routeurs, les switchers ou tout autre équipement informatique. [7]

I.1.2 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.1.3 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.1.3.1 Classification selon l'étendue [4]

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.1.3.1.1 Réseaux personnels sans fil (WPAN)

Le réseau personnel sans fil appelé WPAN concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimantes, téléphones portables, appareils domestiques...) ou un assistant PDA à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes.

Il existe principalement plusieurs technologies utilisées pour les WPAN tels que Bluetooth.

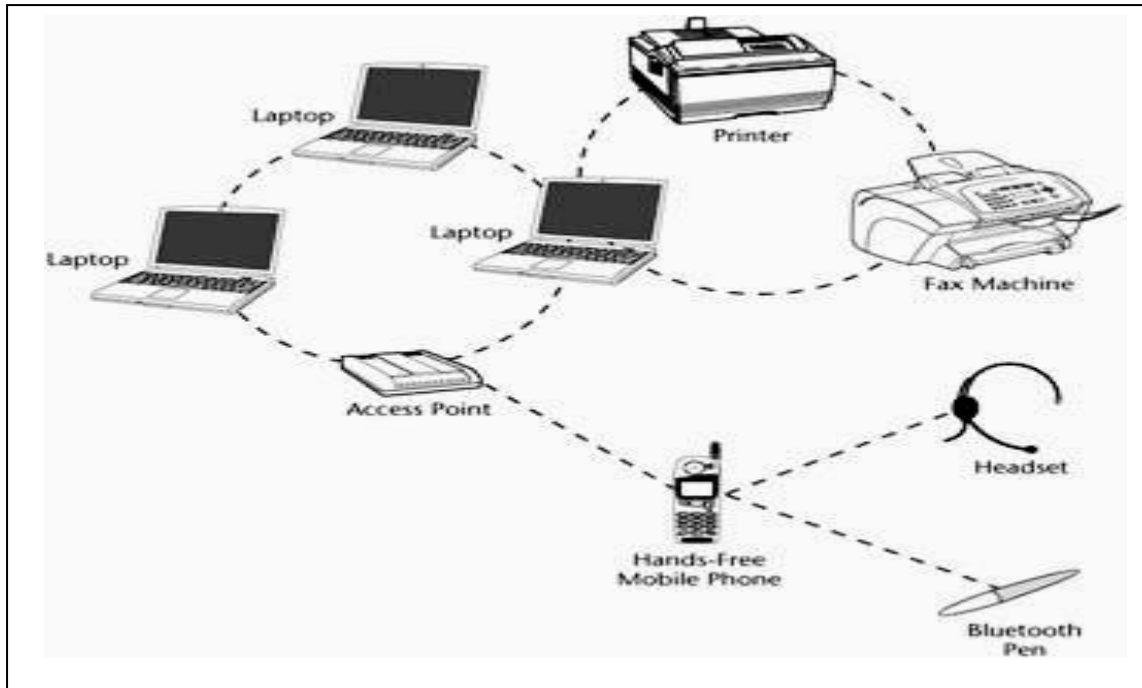


Figure 01 : Réseaux personnelles sans fils (WPAN) [28]

I.1.3.1.1 Les réseaux locaux sans fil (WLAN)

Les WLAN qui sont des solutions sans fil pour les réseaux LAN, permettent d'assurer un périmètre plus étendu d'une centaine de mètre environ et sont conçus pour bâtir des réseaux d'entreprises autour d'une infrastructure fixe. La norme la plus représentative de ces réseaux est bien la norme IEEE802.11 connu sous le nom de WI-FI (Wireless-Fidelity) qui existe depuis 1997.

Cette technologie se caractérise par une transmission de données par ondes radio avec un débit allant jusqu'à 2Mbps sur une bande de 2.4Ghz pour une capacité de 10Mw.

La puissance à l'intérieur d'un bâtiment est de 100Mw pour des fréquences comprises entre 2446.5 et 2483.5 Mhz et la transmission de données se fait à base de signaux infrarouges.

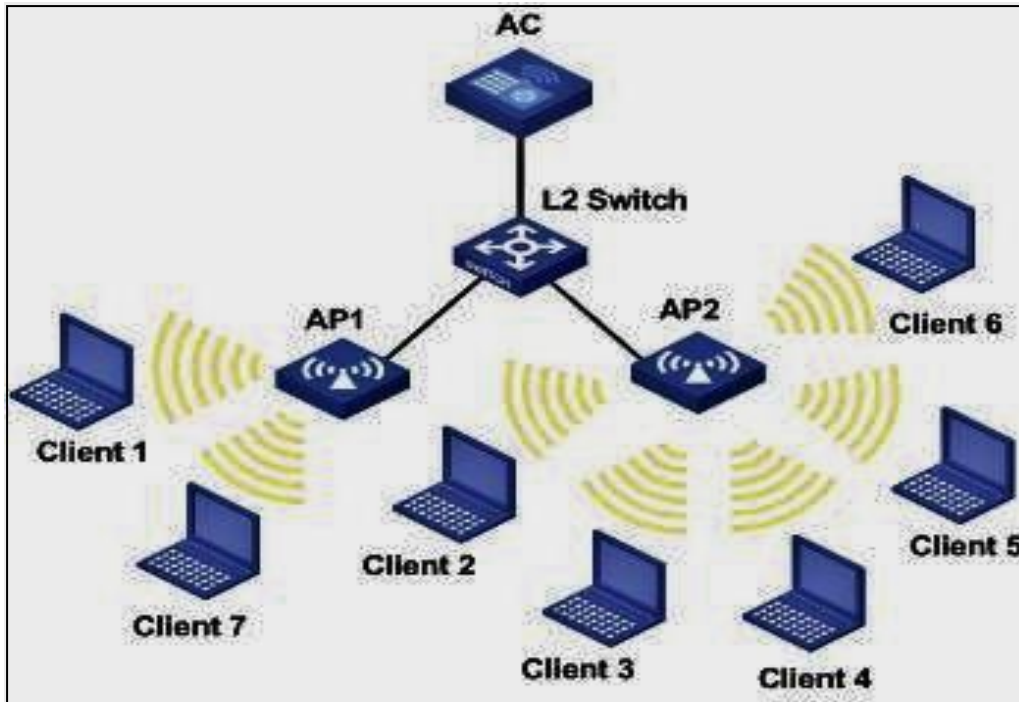


Figure 02 : les réseaux locaux sans fils (WLAN) [29]

I.1.3.1.2 Les satellites de communication [4]

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.

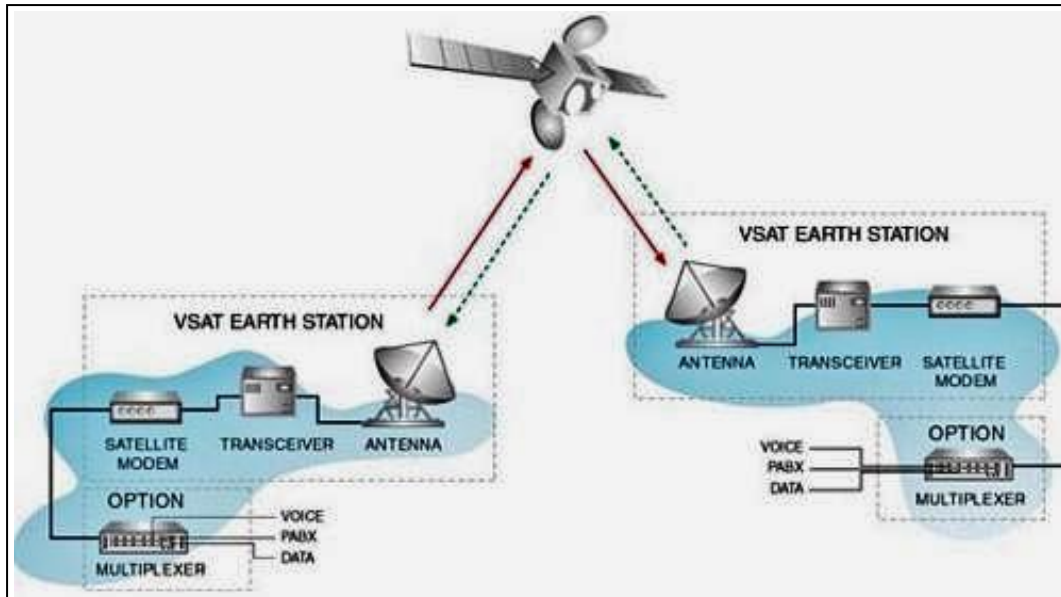


Figure 03 : les satellites de communications [30]

I.1.3.2 Classification selon le mode de communication

I.1.3.2.1 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.

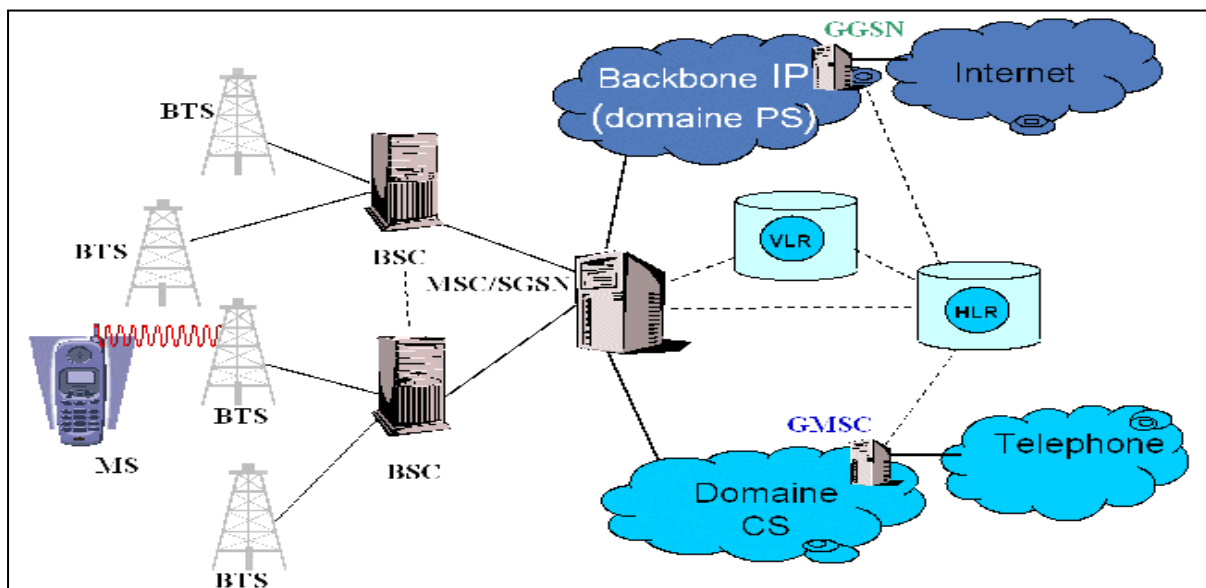


Figure 04 : les réseaux sans fils avec infrastructure [31]

I.1.3.2.2 Mode de communication sans infrastructure (ad-Hoc) [4]

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...)

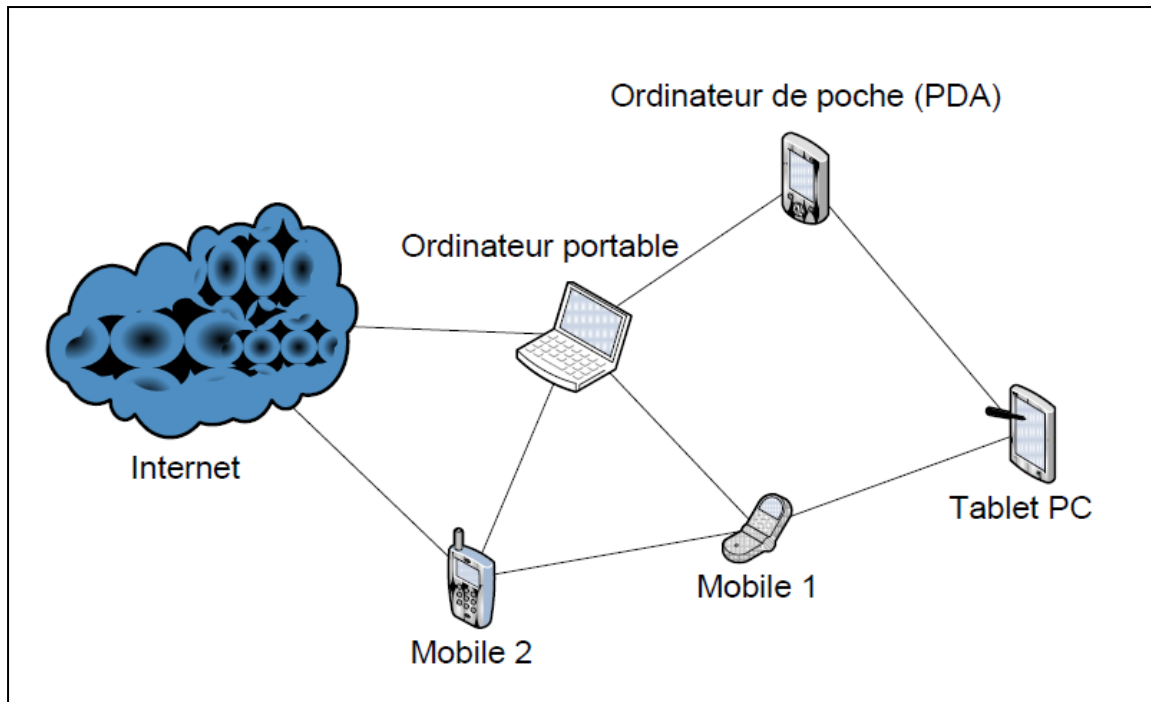


Figure 05 : les réseaux ad-hoc [6]

I.1.4 Evolution des systèmes cellulaires

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.1.4.1 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 point 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. [12]

1.1.4.2 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. [16]

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 œuvre de systèmes de sécurité (authentification au niveau du réseau, protection des communications, cryptage). [12]

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.1.4.3 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 Télécommunications 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. [12]

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. [17]

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 cœur de type IP contrôlé par un système de gestion par politique.
- [12]

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

I.1.4.4 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 802.11^e, 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 802.11^e, 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. [12]

I.1.4.4.1 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

(Dynamic Host Configuration Protocol) dans la version IPv4 ou par son équivalent dans la version IPv6. Grâce à cette adresse, l'utilisateur peut accéder au réseau Internet ou aux réseaux des ISP qui lui sont accessibles. [12]

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.

Android

Introduction

Ces dernières années l'utilisation des applications « intelligentes » sur les Smartphones et tablettes deviennent de plus en plus fréquentes et cela grâce aux différents systèmes d'exploitation mobile tel que : « Android », « IOS1 » ou bien « Windows Phone » qui ne cessent de se développer. [2]

Ce chapitre sera scindé en trois parties distinctes : dans la première partie nous nous intéresserons au développement mobile sous Android, en détaillant son historique, ces caractéristiques ainsi que son architecture. La deuxième partie sera dédiée à la mise en évidence des éléments nécessaires au bon développement d'une application Android. La troisième et dernière partie s'intéressera à la conception et l'innovation.

II.1 Le système d'exploitation d'Android

II.1.1 Création d'Android

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

Android est à l'heure actuelle le système d'exploitation pour Smartphones et tablettes le plus utilisé. Les terminaux visés par Android incluent les téléphones portables, Net book/Smart book, tablettes multimédia, automobile, GPS, Réfrigérateur, etc.

II.1.3 Bugdroid

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 être utilisé librement.

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



Figure 06 : Logo d'Android

II.1.4 Historique et évolution des versions d'Android [18]

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en Octobre 2008. Par la suite Android a connu plusieurs mises à jour. 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.

Version	Nom	API Level	Fonctionnalités
Android 1.0		1	-Fonctionnalités de base (téléphonie, SMS, réseaux de données, accès au répertoire, ...etc.)
Android 1.1	small oven (petit four)	2	-correction de bugs (alarme, Gmail) -Amélioration des fonctionnalités MMS. -Introduction des applications payantes sur l'Android Market.
Android 1.5	Cupcake (coupe gâteau)	3	-Nouveau clavier avec auto complétion. -Support Bluetooth. -Enregistrement de vidéo. -Widgets. -Animations améliorées
Android 1.6	Donut (Beignet)	4	-Recherche vocale améliorée -Indicateur d'utilisation de la batterie -Apparition de fonctionnalités pour les réseaux privés virtuels (VPN)
Android 2.0	Eclair (Sorte de gâteau à la crème)	5	-Interface graphique améliorée -Gestion des contacts changée -Support d'HTML 5 -Support Bluetooth 2.1 -Clavier virtuel amélioré -Refonte de l'installation et des mises à jour du Kit de développement
Android 2.1	Eclair	7	-Nouveau bureau à 5 écrans virtuels -Fonds d'écran animé -Nouveaux effets 3D (notamment sur la gestion des photos) -Amélioration du client Gmail.
Android 2.2	Foyo (Yaourt glacé)	8	-Nouveaux widgets -Gestion des points d'accès au réseau -Support multilingue
Android 2.3	Gingerbread (Pain d'épice)	9	-Amélioration d'interface graphique téléphonie IP -Nouveau clavier virtuel
Android 3.0	Honeycomb (Rayon de miel)	11	-Optimisé pour les tablettes et équipement à écran large -Multitâche et système de notification amélioré
Android 3.1	Honeycomb (Rayon de miel)	12	-Retouches de l'interface -USB Host & Accessories Developer Kit -La possibilité de récupérer directement à travers les applications les informations du compte SIP de l'utilisateur -Support des proxys HTTP pour les points d'accès Wi-

			Fi
Android 3.2	Honeycomb (Rayon de miel)	13	-Amélioration du support hardware, incluant des optimisations pour certaines tablettes -Augmentation pour les applications de la possibilité d'accéder aux fichiers de la carte SD -Compatibilité pour les applications non-optimisées pour les résolutions d'écran des tablettes tactiles
Android 4.0	Ice cream sandwich (Sandwich à la crème glacée)	14	-Nouvelle interface graphique -Amélioration de la sécurité -Beaucoup de raccourcis (Appareil photo, accès SDcard)
Android 4.1	Jelly Bean (Dragée)	16	-Amélioration de l'accessibilité -Recherche vocale hors-ligne -Notifications extensibles
Android 4.2	Jelly Bean (Dragée)	17	-Nouvelle application et widget horloge intégrant une horloge mondiale, un chronomètre et un minuteur -Intégration de nouvelles fonctionnalités dans Google Now -Amélioration de l'accessibilité
Android 4.3	Jelly Bean (Dragée)	18	-Simplification de l'assistant d'installation -Changement d'utilisateur plus rapide -Amélioration de l'Appareil Photo -Ajout de nouvelles langues
Android 4.4	KitKat	19	-Compatibilité avec le Bluetooth MAP. -Mise à jour de l'application Téléchargements permettant une meilleure gestion avec de nouvelles options de triage et une nouvelle interface. -Mise à jour de l'application E-mail avec des dossiers imbriqués, des photos des contacts et une navigation améliorée. -Mise à jour du clavier Google, en plus d'un changement de design.
Android 5.0	Lollipop (sucette)	20	-Projet Volta qui permet d'optimiser la consommation d'énergie et de gagner en autonomie -Amélioration du système de notifications -Activation par défaut du chiffrement des données utilisateur -Disponibilité d'Android TV et Android Auto

Tableau 1 : L'évolution des versions Android

II.1.5 Les avantages d'Android [9]

- **Open Source et Gratuit (ou presque)**

Le contrat de licence pour Android respecte les principes de l'open source, c'est-à-dire que vous pouvez à tout moment télécharger les sources et les modifier selon vos goûts. Ajouté à cela Android utilise des bibliothèques open source puissantes, comme par exemple SQLite pour les bases de données et OpenGL pour la gestion d'images 2D et 3D.

Android est gratuit, autant pour vous que pour les constructeurs. S'il vous prenait l'envie de produire votre propre téléphone sous Android, alors vous n'auriez même pas à ouvrir votre porte-monnaie. En revanche, pour poster vos applications sur le Play Store, il vous en coûtera la modique somme de 25\$. Ces 25\$ permettent de publier autant d'applications que vous le souhaitez, à vie.

- **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 lignes 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).

II.1.6 Présentation de la plateforme Android

II.1.6.1 Architecture d'Android

La figure suivante illustre les composants principaux du système d'exploitation Android.

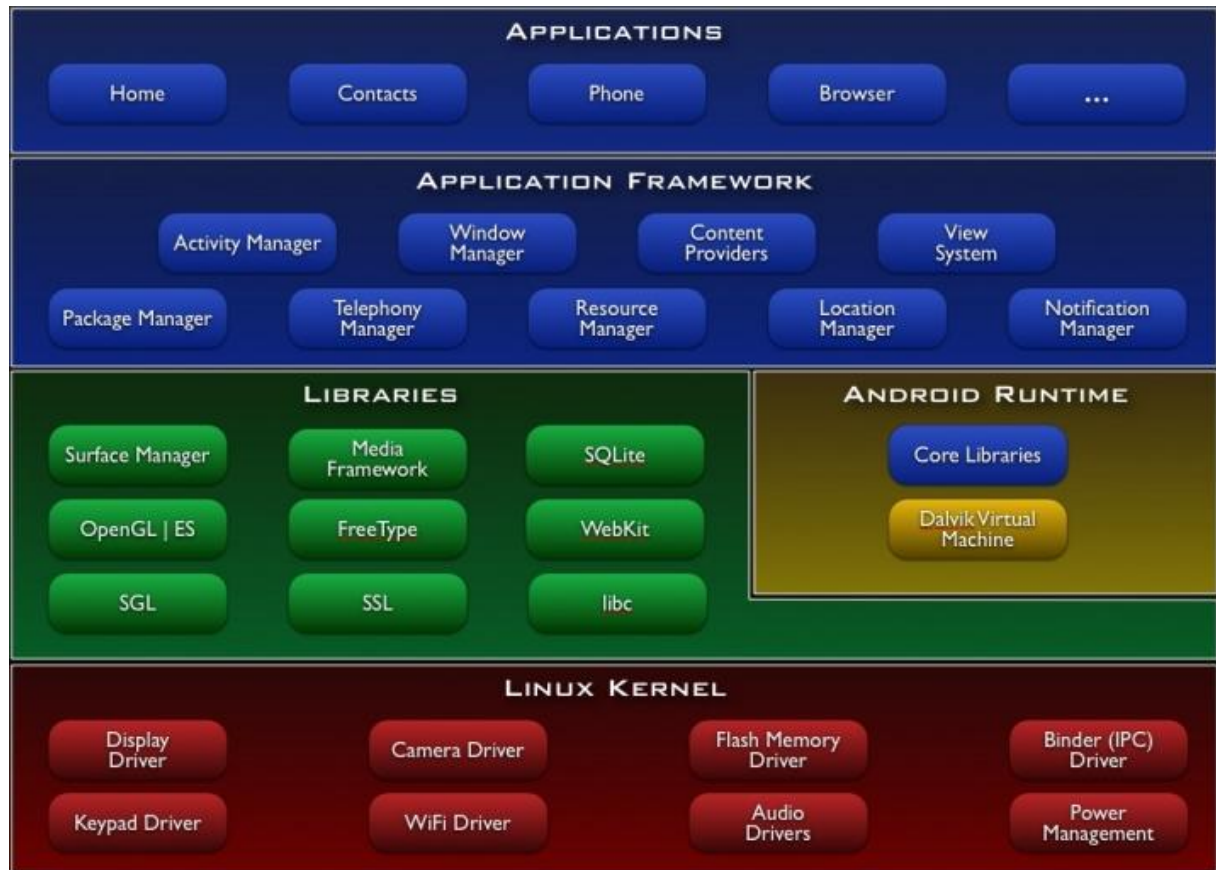


Figure 07: L'architecture du système Android [19]

Android est basé sur un kernel linux 2.6.xx, au dessus du kernel il y a "l'hardware abstraction layer" qui permet de séparer la plateforme logique du matériel.

Au dessus de cette couche d'abstraction 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 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. [20]

II.1.6.1.1 Les niveaux du système d'exploitation d'Android

II.1.6.1.1.1 Premier niveau : 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.1.6.1.1 .2 Deuxième niveau : Bibliothèque et environnement

a)- 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 [22], 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.

b)- 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 [13] [22], qui est livrée sous forme expérimentale avec la plateforme.

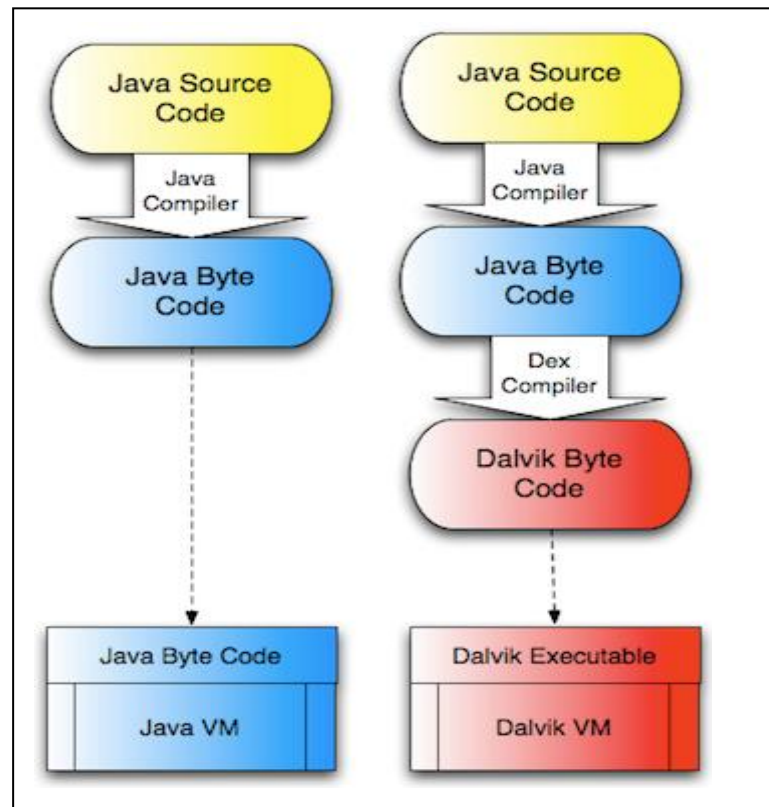


Figure 08 : Différence entre la JVM et Dalvik

II.1.6.1.1.3 Troisième niveau : 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 Service-Manager, 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.1.6.1.1.4 Quatrième niveau : 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. [14]

II.1.6.2 Les composants d'une application Android [14]

Une application Android est écrite en Java. Contrairement à l'application Java standard, une application Android peut posséder plusieurs points d'entrée. Plus précisément, une application Android peut avoir plusieurs composants et chacun d'eux peut-être un point d'entrée dans le programme. Il existe quatre types de composants : Activity, ContentProvider, Service et BroadcastReceiver.

II.1.6.2.1 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'historique 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.

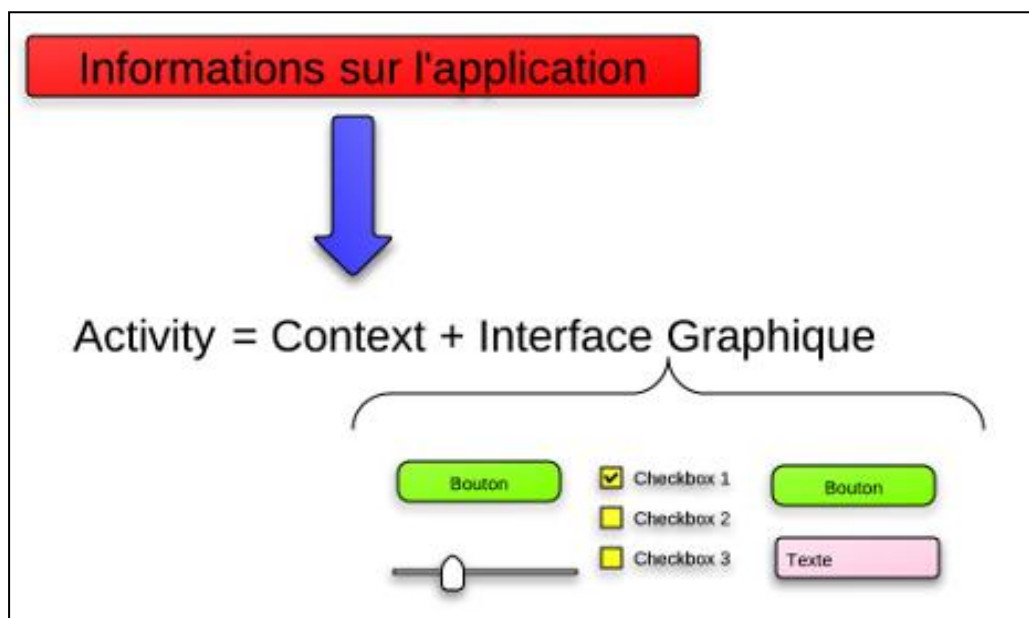


Figure 09 : Les composants d'une activité

II.1.6.2.1.1 Cycle de vie d'une activité [9]

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. [8]

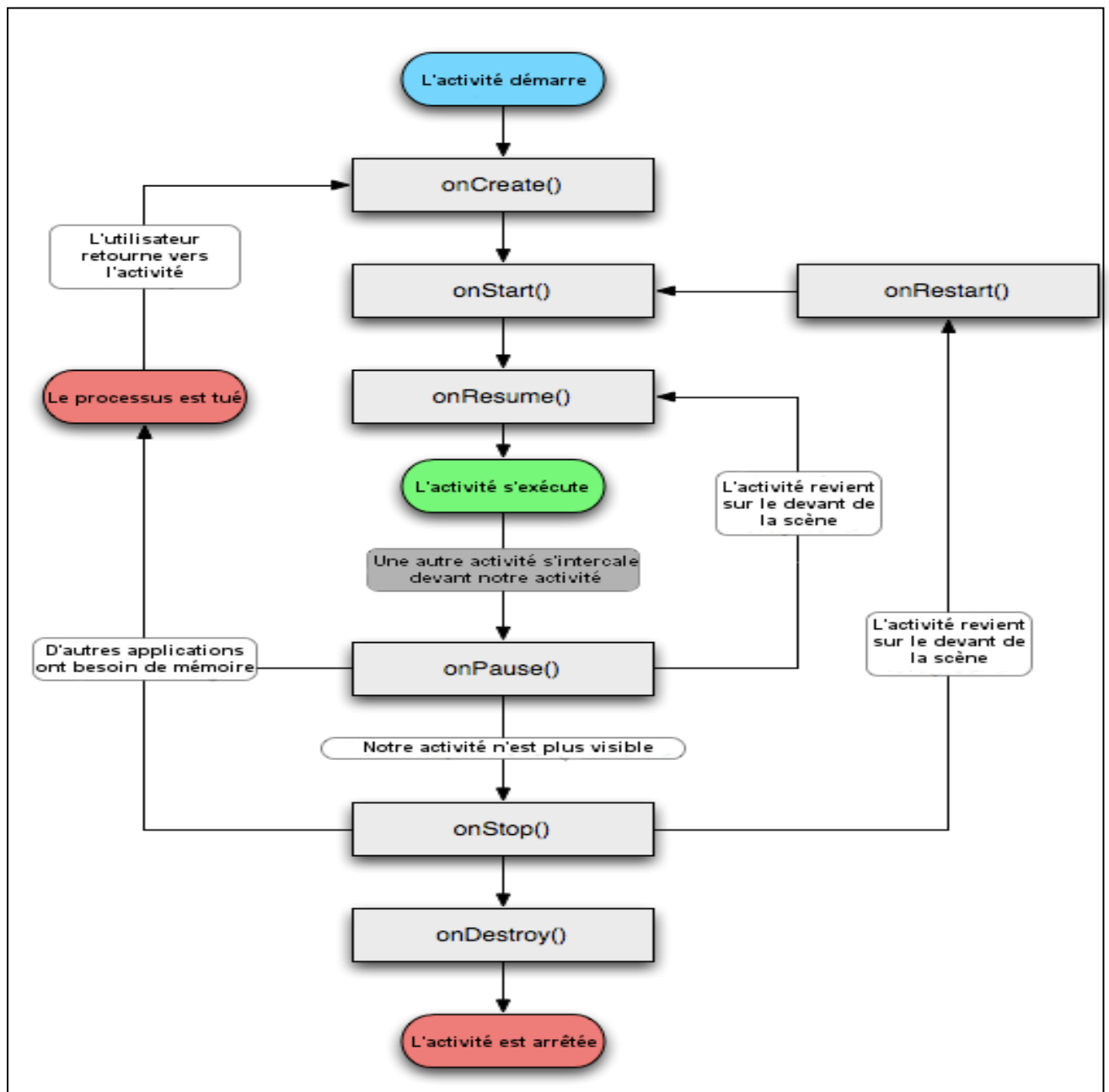


Figure 10 : Cycle de vie d'une activité

II.1.6.2.1.2 L'Etats d'une activité [9]

Etat	Visibilité	Description
Active (« active » ou « Running »)	L'activité est visible en totalité	Elle est sur le dessus de la pile, c'est ce que l'utilisateur consulte en ce moment même et il peut l'utiliser dans son intégralité. C'est cette application qui a le focus, c'est-à-dire que l'utilisateur agit directement sur l'application.
Suspendue (« paused »)	L'activité est partiellement visible à l'écran. C'est le cas quand vous recevez un SMS et qu'une fenêtre semi-transparente se pose devant votre activité pour afficher le contenu du message et vous permettre d'y répondre par exemple.	Ce n'est pas sur cette activité qu'agit l'utilisateur. L'application n'a pas le focus, c'est l'application sus-jacente qui l'a. pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis l'utilisateur pourra à nouveau interagir avec. Si le système a besoin de mémoire, il peut très bien tuer l'application (cette affirmation n'est plus vraie si vous utiliser un SDK avec l'API 11 minimum).
Arrêtée (« stopped »)	L'activité est tout simplement oblitérée par une autre activité, on ne peut plus la voir du tout.	L'application n'as évidemment plus le focus, et puisque l'utilisateur ne peut pas la voir, il ne peut agir dessus. Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue votre application pour libérer de la mémoire système.

Tableau 02: les différents états d'une activité et leurs caractéristiques

II.1.6.2.1.3 Notification de l'état d'une activité

Quand une activité doit être basculée d'un état à un autre, le système Android l'informe du changement d'état à travers des messages appropriés. Ces messages (dits événements) peuvent être exploités par le programmeur pour mieux gérer la mémoire et les autres ressources (par exemple, relâcher un pointeur vers un fichier, ou commuter les données temporaires vers une base de données).

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

- ✓ `onCreate ()` : cette méthode sera exécutée à la création de l'activité par le système (avant son démarrage).
- ✓ `onStart ()` : l'activité va devenir visible cette méthode sera exécutée lors de démarrage de l'activité.
- ✓ `onPause()` : l'activité va être mise en pause.
- ✓ `onResume ()` : l'activité est maintenant visible, cette méthode sera exécutée lorsque l'activité résume son exécution après la suspension (pause).
- ✓ `onStop ()` : l'activité ne sera plus visible.
- ✓ `onDestroy ()` : l'activité va être détruite.

II.1.6.2.2 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.1.6.2.3 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.1.6.2.4 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.2 Développement d'application pour Android

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.2.1 SDK Android (Softwar Developpent Kit)

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.

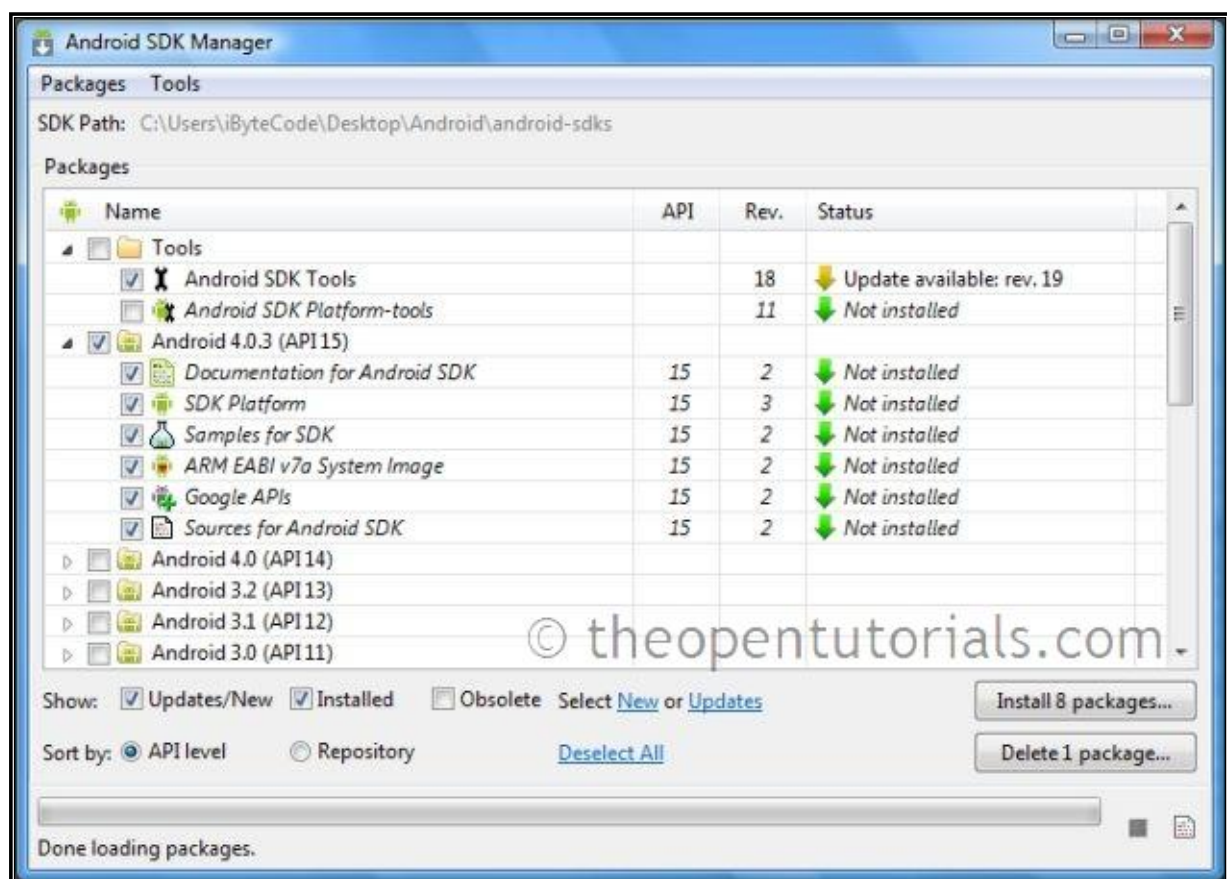


Figure 11 : Android SDK

II.2.1.1 Les outils fournis par le SDK

II.2.1.1.1 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. [9]

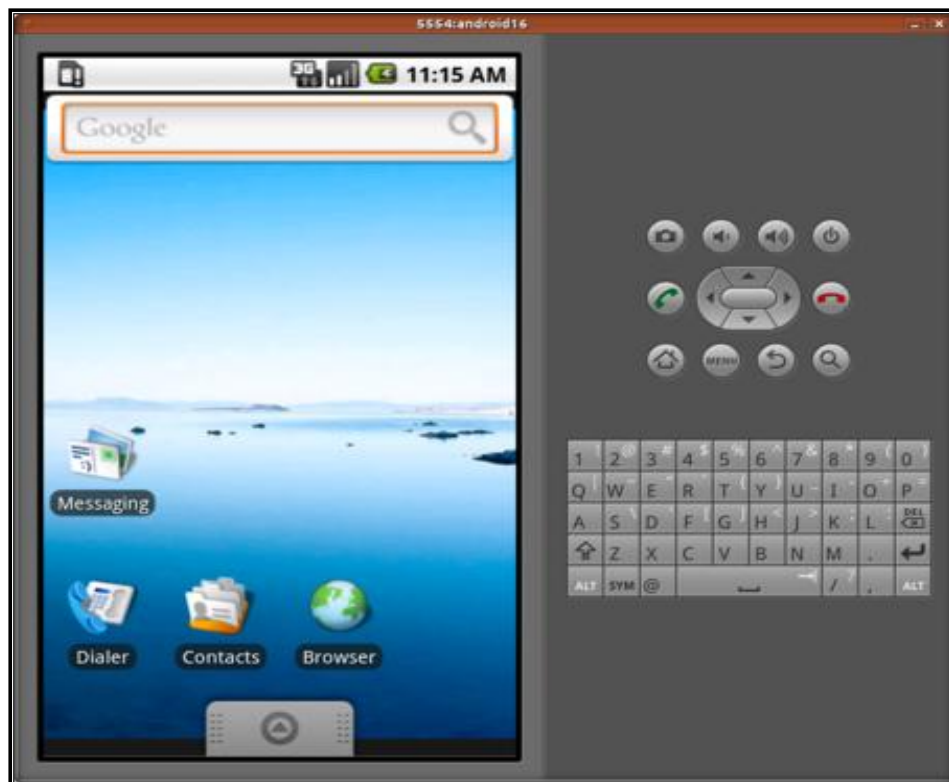


Figure 12 : L'émulateur

II.2.1.1.2 L'ADB (Android Debug Bridge)

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. [23]

II.2.1.1.3 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. [8]

II.3 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.3.1 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 ajoutent 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 13 : Bureau virtuel

II.3.2 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...etc.

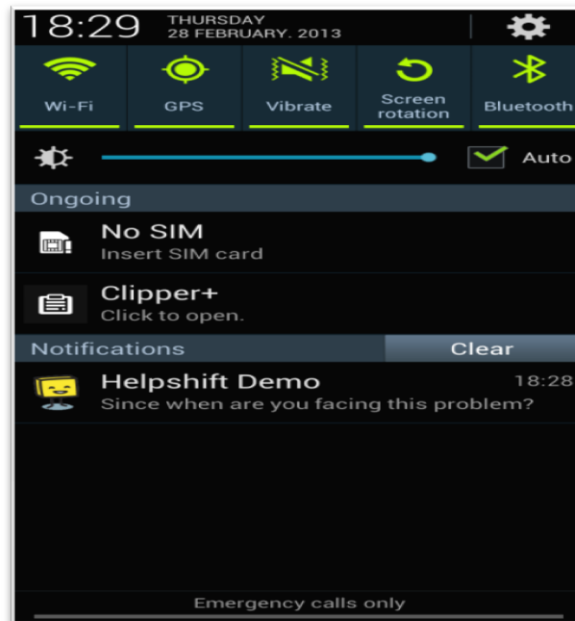


Figure 14 : Exemple de notification

II.3.3 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. [15]

II.3.4 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. [24]

II.4 Présentation du langage java [11]

II.4.1 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 Mais: 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.4.2 Les caractéristiques de JAVA

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

- JAVA est indépendante de toute plate-forme.
- JAVA est orienté objet.
- JAVA est simple.
- JAVA est fortement type.
- JAVA assure la gestion de la mémoire.
- JAVA est sûre.
- JAVA est économe.
- JAVA est multitâche.

II.4.3 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.

Conclusion

Dans ce chapitre on a pu mettre en évidence toutes les caractéristiques d'Android ainsi que les éléments nécessaires pour le bon développement d'une application Android.

Le chapitre suivant est composé de deux parties :

- Conception
- Réalisation

Conception et réalisation

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.1.Présentation de l'UML

UML (Unified Modelling Language) est un langage de modélisation objet qui décrit la Mise en œuvre 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. [1]

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

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

III.1.1.Modélisation avec l'UML

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 : [10]

- ✓ **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.1.2.La démarche de modélisation avec l'UML

La démarche de modélisation choisie pour concevoir notre application peut être représentée graphiquement comme suite :

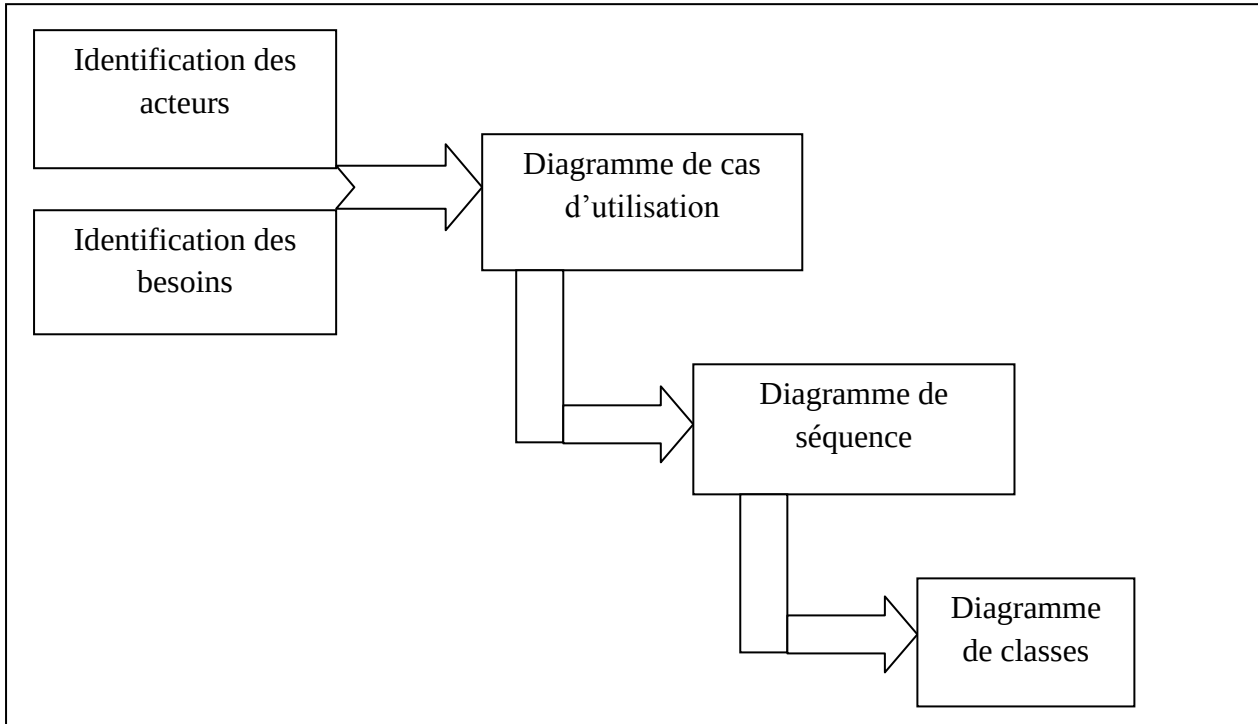


Figure 15 : La démarche adoptée pour la modélisation.

III.2.Conception

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 œuvre 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.2.1. Les diagrammes des cas d'utilisation

Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation. [3]

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.

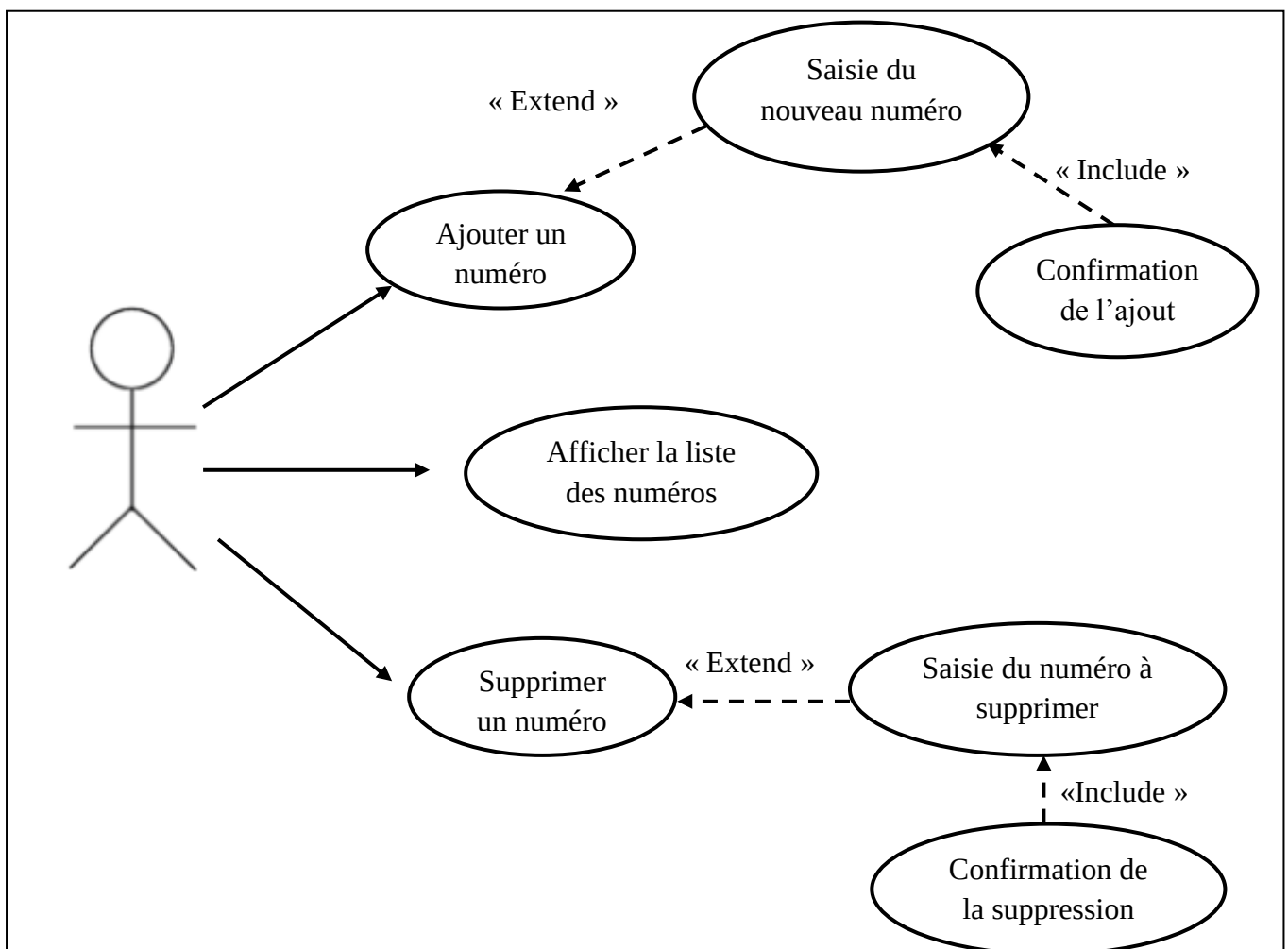


Figure 16: Diagramme de cas d'utilisation générale.

III.2.2. Les diagramme de séquences

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 :

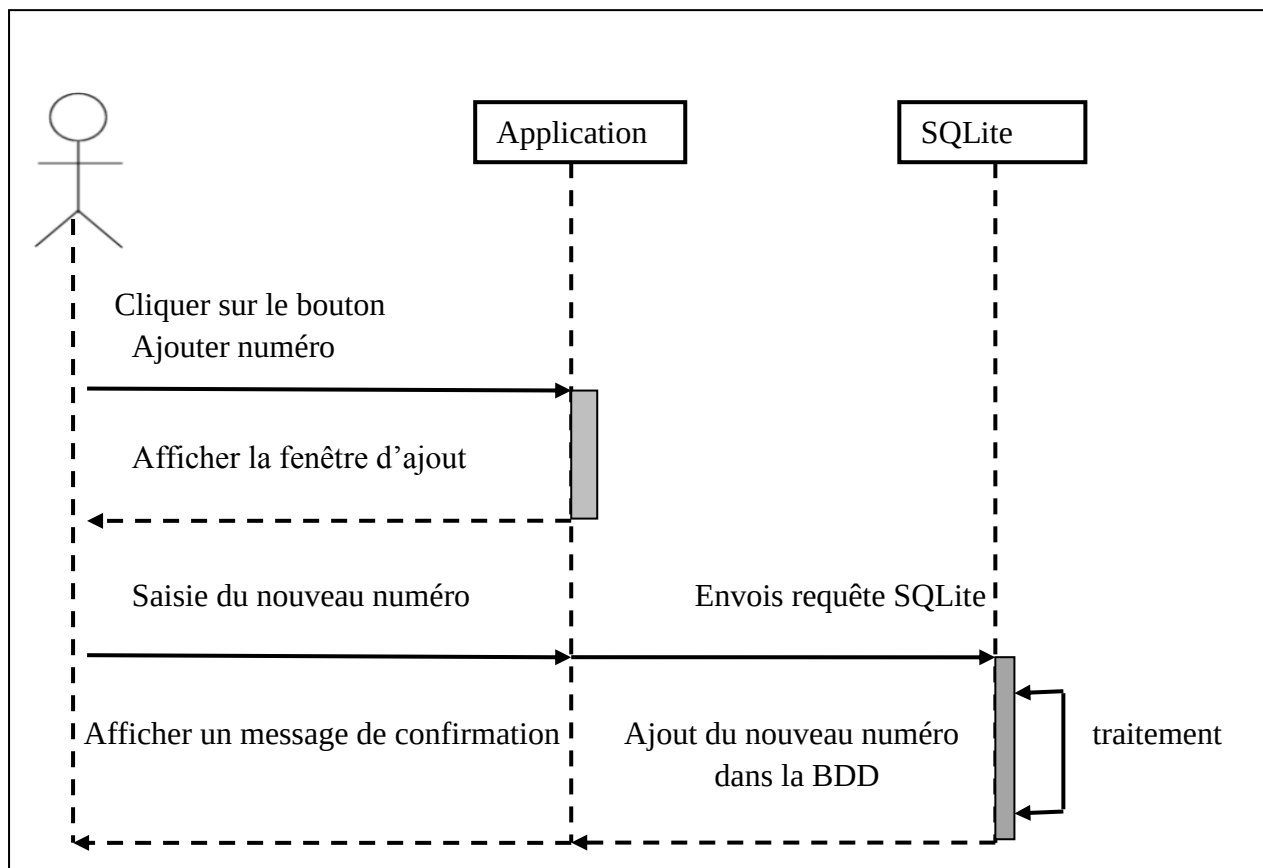


Figure 17: Diagramme de séquence de cas d'utilisation « Ajouter un nouveau numéro »

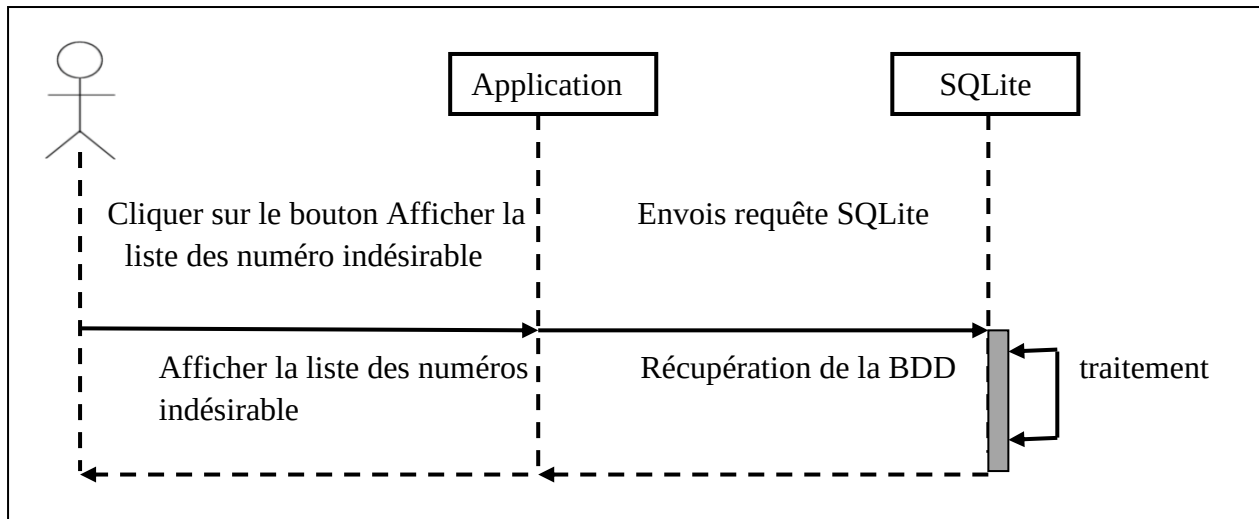


Figure 18: Diagramme de séquence de cas d'utilisation « Afficher la liste des numéros indésirables »

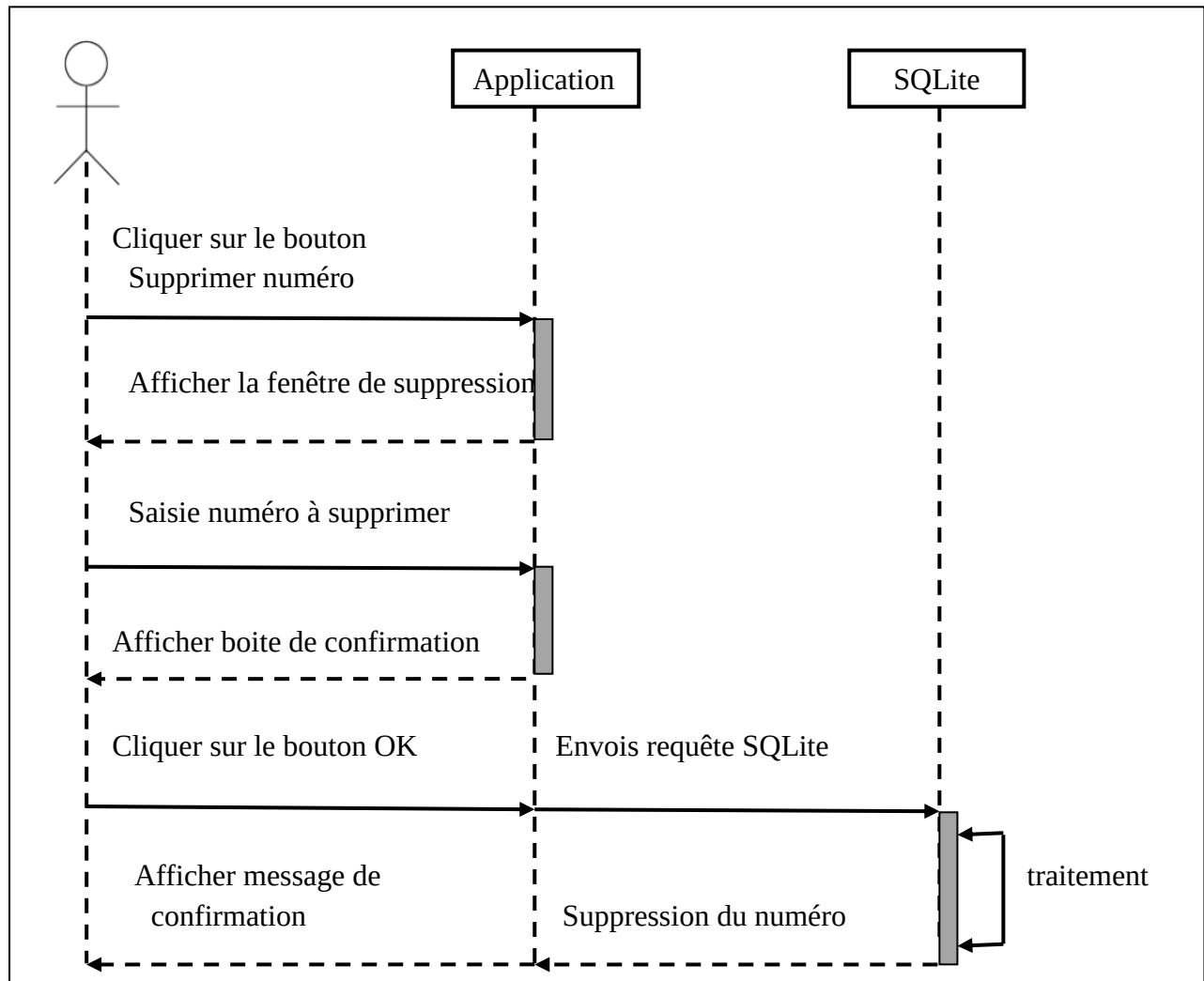


Figure 19: Diagramme de séquence de cas d'utilisation « Supprimer un numéro de la liste »

III.3 Réalisation

III.3.1 Présentation de notre application

La première étape de notre travail est l'installation de l'application en utilisant un fichier « .apk » sur l'appareil de l'utilisateur. Grâce aux permissions qu'elle contient, l'application peut prendre le contrôle de la réception des messages.

La deuxième étape est la mise en pratique de notre application. On doit d'abord ajouter un numéro dans la base SQLite, à la réception d'un message notre application va vérifier que le numéro entrant ne se trouve pas dans la base de données ainsi le message sera filtrer.

III.3.2. Environnement de travail

III.3.2.1. 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, 6,00 GO de mémoire vive, Windows 7.

III.3.2.2. Environnement cible

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

III.3.2.3. 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 eclipse 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.

III.3.2.3.1. 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.

III.3.2.3.2. 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.

III.3.2.3.3. 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.

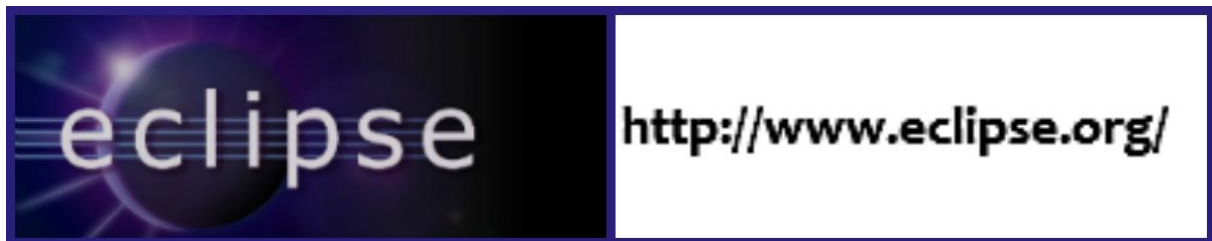


Figure 20: Logo Eclipse et site web.

III.3.2.3.4. 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 21), avec le plugin ADT de Google.

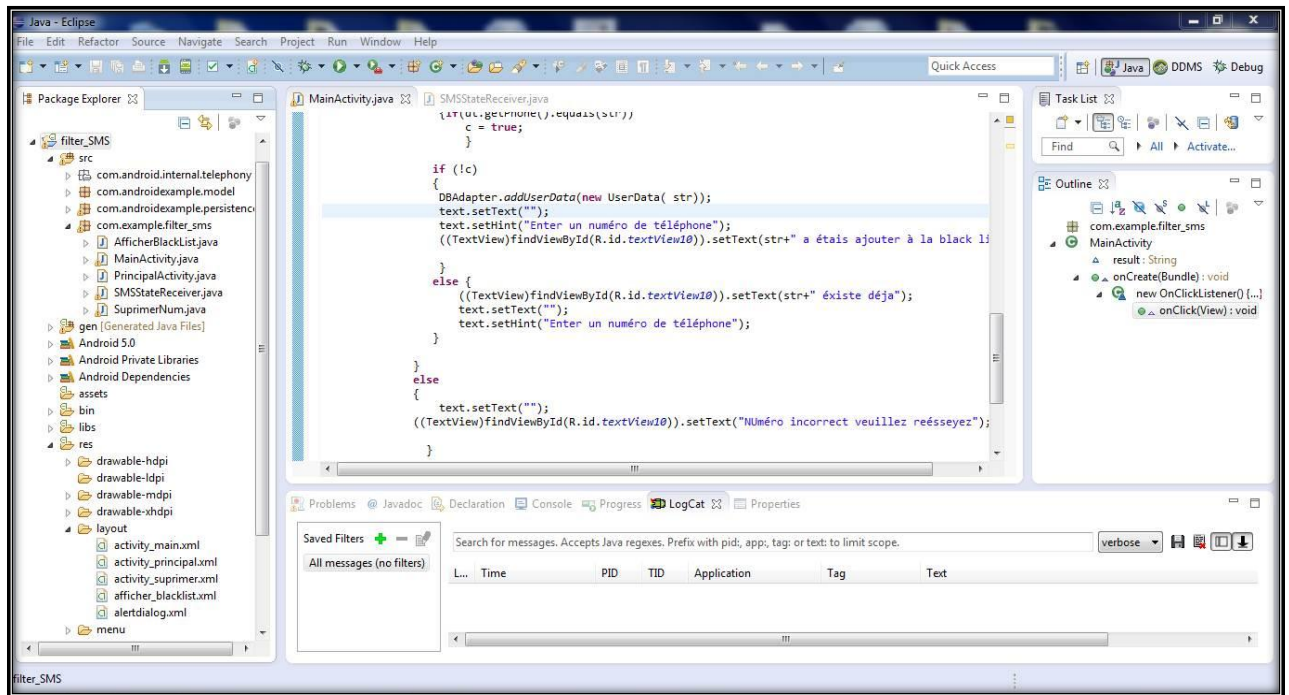


Figure 21: plate-forme d'éclipse.

III.3.3.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. [25].



Figure 22: 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.

III.3.3.1. 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. [26].

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

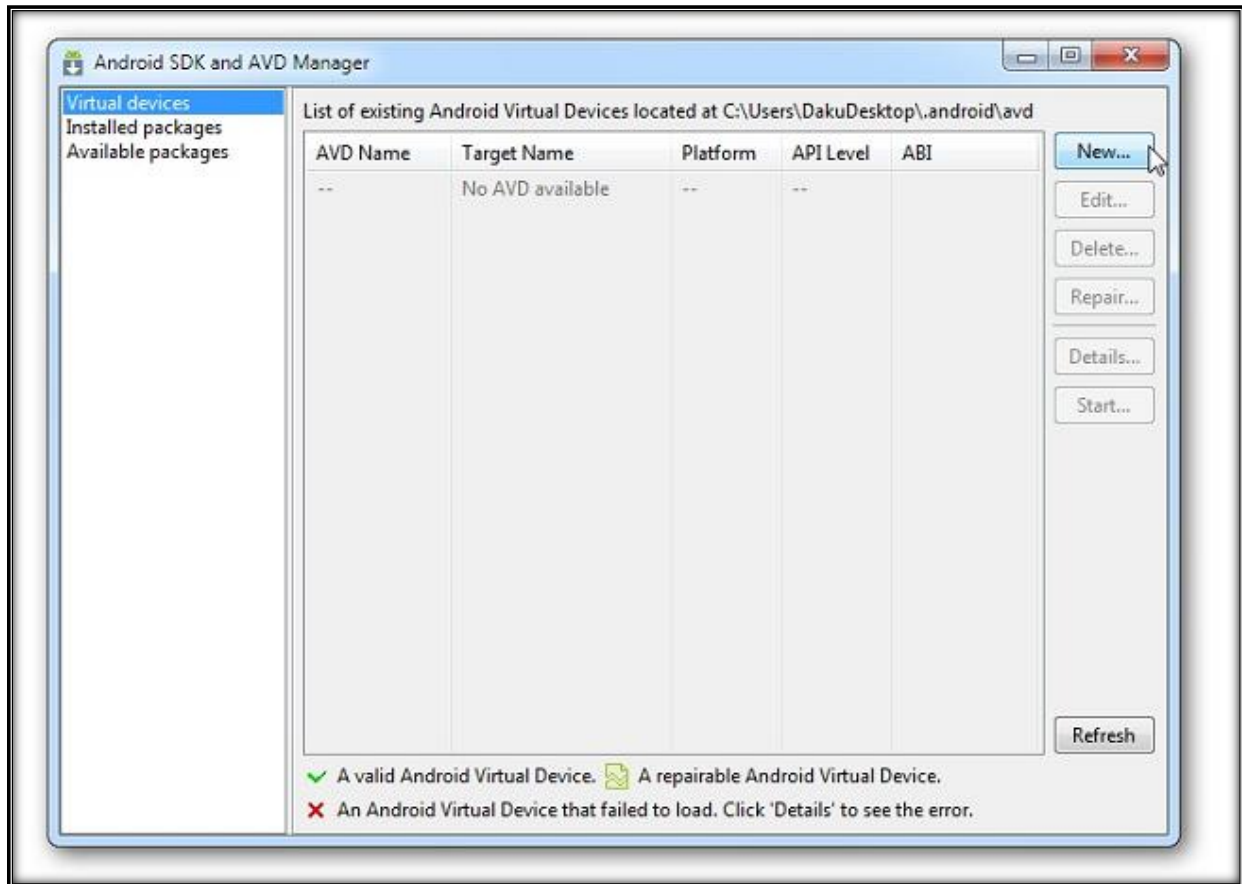


Figure 23: Interface d'installation du SDK Android.

III.3.3.2. 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. [27]

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.

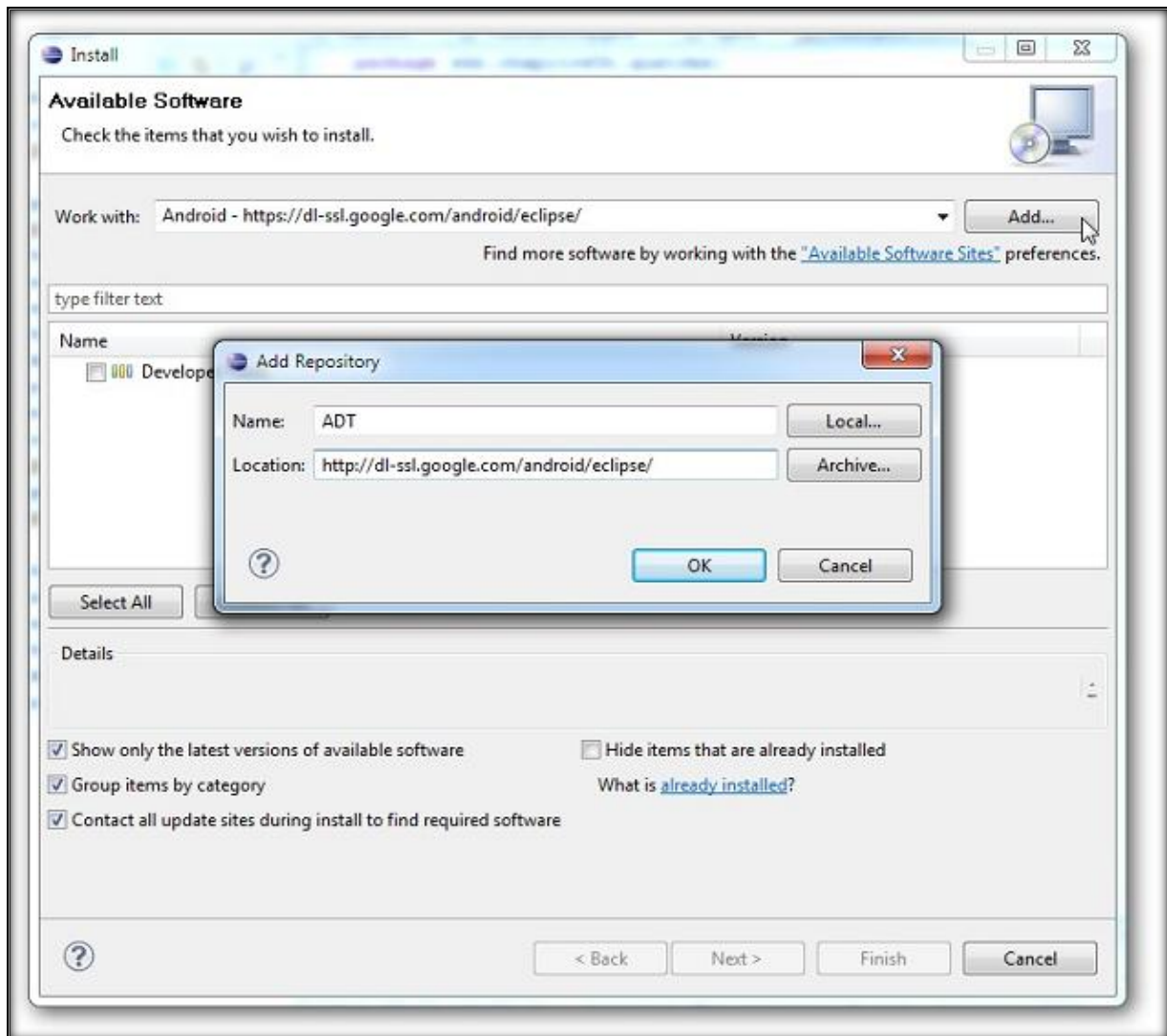


Figure 24: Interface d'installation ADT.

III.3.3.3.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.

III.3.3.3.1 Les étapes de création d'un nouveau projet

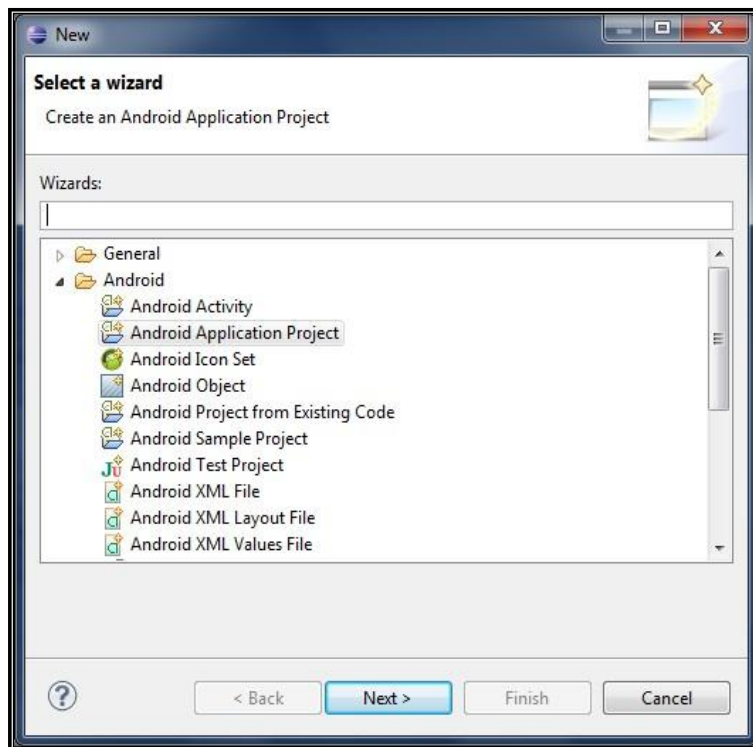


Figure 25: select a wizard

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

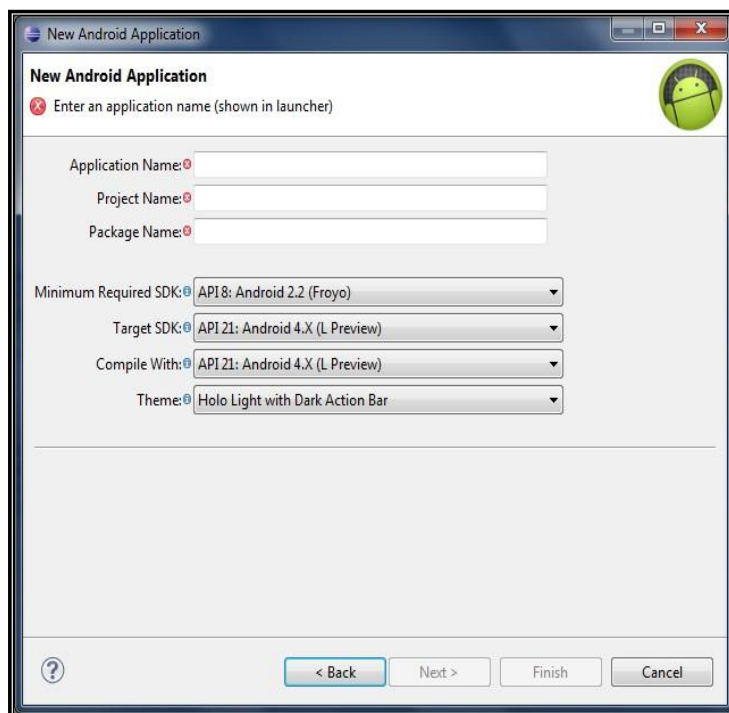


Figure 26 : création d'un nouveau projet Android

- 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:

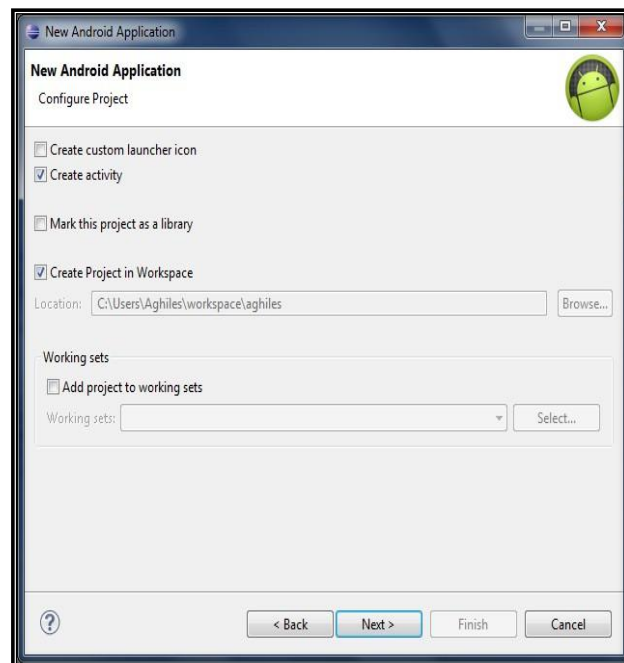


Figure 27: Configuration du projet

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 » :

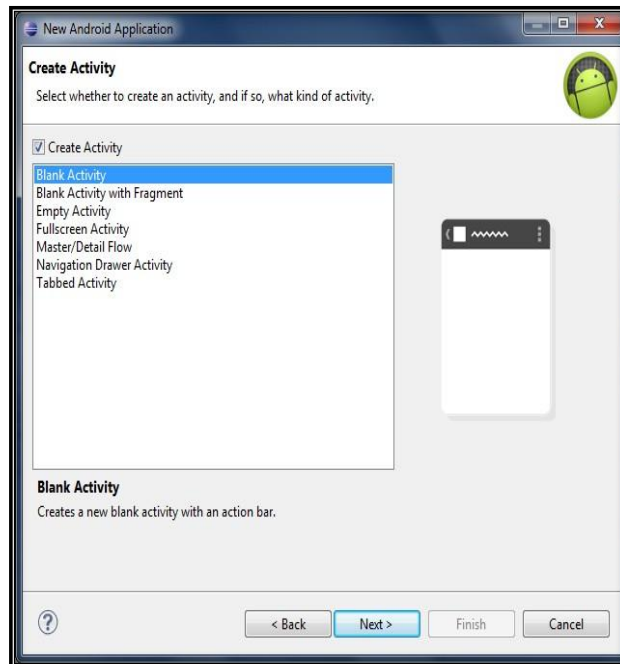


Figure 28: Création d'une activité

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 » :

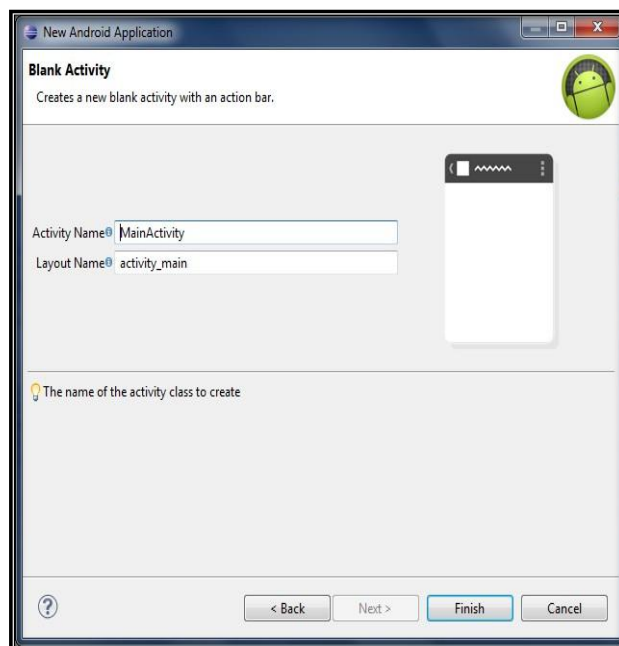


Figure 29: crée une nouvelle activité en blanc avec une barre d'action

- « 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 :

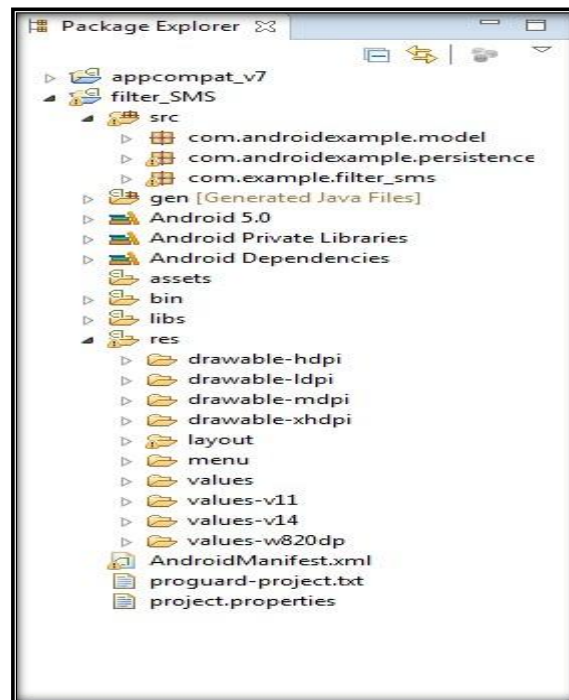


Figure 30: Arborescence du projet

III.4.3.Mode de fonctionnement de l'application

III.4.3.1.Le menu principal de l'application

En premier lieu l'utilisateur lance l'application en appuyant sur l'icône lui correspondant dans le menu du Smartphone, ce qui donne ceci :

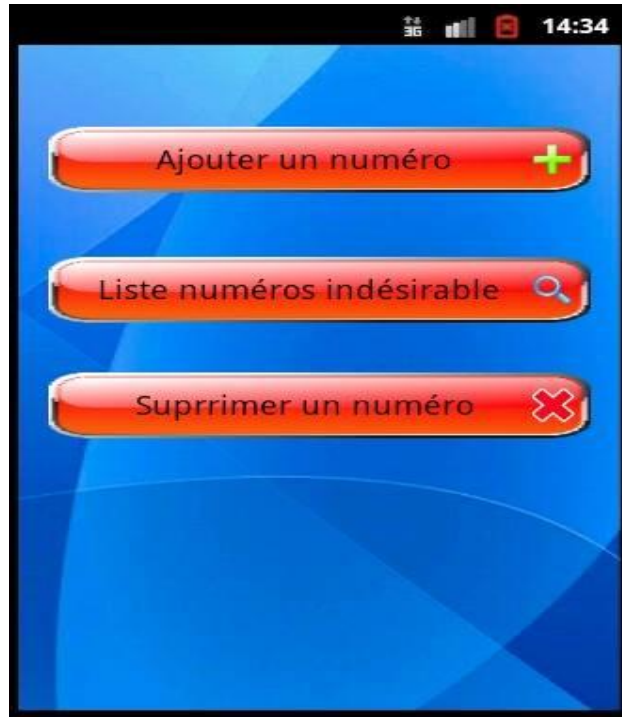


Figure 31: Menu principal.

L'utilisateur dispose de trois boutons :

- 1 : Ajouter un numéro.
- 2 : Afficher la liste des numéros indésirable
- 3 : Supprimer un numéro

III.4.3.2.Interface d'ajout d'un numéro

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



Figure 32: Interface d'ajout d'un numéro.

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

III. 4.3.3.Interface d'affichage de la liste des numéros indésirable :

Après avoir cliqué sur le bouton « afficher la liste des numéros indésirable » on aura ceci :



Figure 33: interface d'affichage de la liste des numéros indésirable.

III.4.3.4.Interface de suppression d'un numéro :

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



Figure 34.A: interface de saisie d'un numéro à supprimer

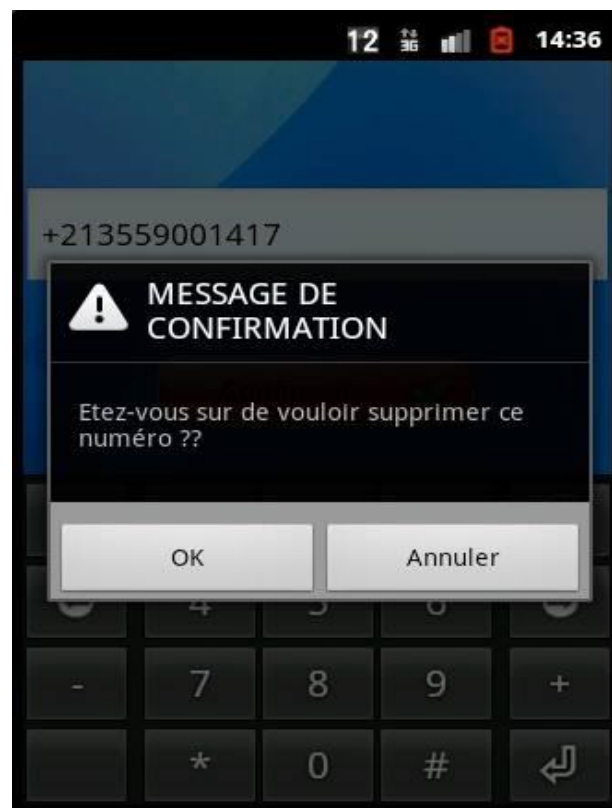


Figure 34.B : Interface de confirmation de la suppression.

Conclusion

Nous avons consacré ce chapitre à la conception et à la réalisation de notre application. En premier lieu nous avons mis en évidence les différents diagrammes des cas d'utilisation. En deuxième lieu on a illustré notre application par les différentes interfaces qui la compose, ainsi que les logiciels qui nous ont permis de réaliser notre travail.

Conclusion générale

Conclusion générale et perspectives

Conclusion générale et perspectives

L'objectif de notre travail était de réaliser une application sous un système d'exploitation Android (version 2.2 et plus) qui consiste à filtrer les messages indésirables.

Notre application ainsi réaliser permet :

- A l'utilisateur d'ajouter, de supprimer des numéros indésirables
- De consulter les numéros indésirables grâce à une base de données
- De gérer la réception des messages (SMS)

Cette réalisation nous a aussi permit d'enrichir notre savoir et de développer 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.

Quoi qu'il en soit cette application est loin de convenir parfaitement aux besoins réels de l'utilisateur, dans un futur proche et des connaissances plus enrichies, nous espérons bien l'étendre afin qu'il puisse y avoir plus de choix, par ailleurs ce projet de fin d'étude est une expérience humaine très enrichissante.

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

- Ajouter le filtrage d'appel
- Exploiter d'autres Smartphones: Appel, Windows, BlackBerry.