



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOU D MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT INFORMATIQUE

Mémoire

En vue de l'obtention du diplôme Master II en Informatique
Specialité : Reseaux Mobilité et Systemes Embarqués(RMSE)

Thème

Réalisation d'une passerelle SMS/Email

Proposé et dirigé par :

Mr DAOUI Mehammed.

Réalisé par :

Mlle MOUSSA YAHIA Lilia

Mr ABIR Lyes

Promotion: 2014/2015

Remerciements

*Nous tenons à remercier
chaleureusement :*

*En premier lieu le bon dieu de nous avoir
donné force, foi, courage et patience pour
réaliser ce travail.*

*Notre promoteur M. Daoui pour sa
précieuse aide, ses conseils, ses
orientations qui nous ont été d'une grande
importance.*

*A tous ceux qui ont contribué de près ou
de loin à la réalisation de ce travail.*

*Enfin, nos remerciements s'adressent aux
membres du jury qui vont nous faire
l'honneur de juger notre travail.*

Dédicaces

*Je dédie ce modeste travail à mes parents
auxquels je dois tout mon respect et que je
ne remercierais jamais assez pour leurs
soutien et encouragements tout au long de
mes études.*

A mes deux frères.

*A mes meilleures amies qui furent
toujours présentes*

Célia et Melissa.

A toi Lyes.

*ALynda et Sedik pour les échanges et
partages d'expérience durant
l'élaboration de ce mémoire.*

A tous mes amis.

Lilia

Dédicaces

*Je dédie ce modeste travail à mes parents
et mes grands-parents qui m'ont toujours
conseillé et orienté auxquels je dois tout
mon respect et que je ne remercierais
jamais assez.*

A mes deux frères et à ma sœur.

*A mes meilleurs amis qui furent toujours
présents.*

Lyes

Sommaire

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I

Introduction :	3
I.1 Réseaux mobiles :	3
I.1.1 Les générations des réseaux mobiles :	3
I.1.1.1 La première génération (1G) :	4
I.1.1.2 La deuxième génération (2G):	4
I.1.1.3 La troisième génération (3G):	4
I.1.1.4 La quatrième génération :	5
I.2 SMS (short message service):	5
I.2.1 Définition :	6
I.2.2 Bref historique :	6
I.2.3 Rédaction :	6
I.2.4 Classes des SMS :	6
I.2.5 Evolution des domaines d'utilisation :	7
I.3 Le réseau GSM :	7
I.3.1 Définition :	7
I.3.2 Architecture organisationnelle d'un réseau GSM :	8
I.3.3 Architecture fonctionnelle d'un réseau GSM :	9
I.3.3.1 Le sous-système radio : BSS (Base Station Sub-system) :	9
I.3.3.2 Le sous-système réseau NSS (Network Station Sub-system) : ..	9
I.3.3.3 Le sous-système opérationnel :	10
I.3.4 Services offerts par GSM :	10
I.4 le réseau GPRS (General Packet Radio Service) :	11
I.4.1 Définition :	11
I.4.2 Commutation par paquet :	11
I.4.3 Services offerts par le GPRS :	11
I.5 Technologie d'internet :	12
I.5.1 Définition :	12
I.5.2 L'évolution de l'internet :	12
I.5.3 Les différentes technologies d'accès à internet :	13

Sommaire

I.5.3.1 Connexions filaires :.....	13
I.5.3.2 Connexions mobiles à internet :.....	14
I.5.4 Protocoles logiciels :	15
I.6 La passerelle SMS Mail :.....	16
I.6.1 Définition :	16
I.6.2 GSM appareil de passerelle :	17
I.6.3 Les différents protocoles utilisés dans les passerelles SMS/mail :	17
I.6.4 Services offerts par une passerelle SMS/Mail :	19
Conclusion :	21

Chapitre II

Introduction :	22
II.1 Système embarqué :	22
II.1.1 Définition :.....	22
II.1.2 architecture générale d'un système embarqué :.....	22
II.1.3 Caractéristiques :.....	24
II.1.4 Complexité :	25
II.1.5 Classification des systèmes embarqués :	25
II.1.6 Types de systèmes embarqués temps réel :	26
II.1.7 Les langages de développement utilisés :.....	27
II.1.8 Domaines d'utilisations des systèmes embarqués :	27
II.1.9 différents systèmes d'exploitation pour l'embarqué :.....	29
II.2 Android :	29
II.2.1 Définition :.....	29
II.2.2 Historique :.....	29
II.2.3 Architecture générale d'Android :.....	30
II.2.4 Les différentes versions d'Android :.....	32
II.2.5 Répartition actuelle des versions d'Android :	32
II.2.6 Le moteur d'exécution Android :.....	33
II.2.7 Concurrence et parts du marché :	34
II.2.8 application Android :	34

Sommaire

II.2.8.1 vue générale d'une application Android :	34
II.2.8.2 vue détaillée :	36
II.2.9 Concept et innovation d'Android:	37
II.2.9.1 Bureau virtuel (home):.....	37
II.2.9.2 Les Wigdets :.....	38
II.2.9.3 Centralisation des notifications :.....	38
Conclusion :	40

Chapitre III

Introduction	41
III.1 Objectifs de notre travail :	41
III.2 Analyse et conception :	41
III.3 Analyse :	41
III.3.1 Architecture générale :	43
III.3.2 Fonctionnement du système :	43
III.3.3 spécification des besoins :	44
III.4 Conception :	45
III.4.1 Diagrammes de séquences :	45
III.4.1.1 Diagramme de séquence pour envoyer un email :	46
III.4.1.2 Diagramme de séquence pour recevoir un email :	47
III.4.1.3 Diagramme de séquence pour activer ou désactiver le service : .	49
III.4.1.4 Diagramme de séquence pour afficher la liste des emails :	50
III.4.2 Diagrammes de classes :	51
III.4.3 Quelques algorithmes:	54
III.4.3.1 Algorithme d'extraction des données:	54
III.4.3.2 Algorithme de tri des messages:	55
Conclusion :	57

Chapitre IV

Introduction :	58
IV.1 Environnement du travail :	58
IV.1.1 Environnement matériel :	58
IV.1.2 Environnement cible :	58

Sommaire

IV.1.3 Environnement logiciel :	58
IV.1.3.1 IDE Eclipse :	59
IV.1.3.2 Les bibliothèques java :	59
IV.1.3.3 LE SDK :	60
IV.1.3.4 Le SDK Manager :	60
IV.1.3.5 ADT pour Eclipse:	61
IV.1.3.6 Android Virtual Device Manager :	61
IV.1.3.7 Emulateur Android:	62
IV.2 Fonctionnement de nos Applications:	63
IV.2.1 Application Android pour le client :	63
IV.2.2 Application pour le serveur :	65
IV.3 Les fonctions de notre application :	65
1. Envoi d'un Email :	65
2. Activer ou désactiver le service passerelle :	67
3. Réception d'un email :	69
4. Afficher la liste des messages reçus :	70
5. Activer ou désactiver un service sur la partie serveur :	71
Conclusion :	73
Conclusion générale :	74
Annexes I :	75
Annexes II :	77

Liste des Figures

Chapitre I

Figure I.1: Architecture organisationnelle d'un réseau GSM.....	8
Figure I.2 : Architecture fonctionnelle d'un réseau GSM.	9
Figure I.3 : Représentation d'une passerelle.....	17
Figure I.4 : Organisation du protocole POP3.....	18
Figure I.5 : Organisation du protocole IMAP.....	19

Chapitre II

Figure II.1 : Vue simplifié d'un système embarqué.	23
Figure II.2: Architecture typique d'un système embarqué	24
FigureII.3 : Architecture générale d'un système embarqué temps réel.	26
Figure II.4 : Le Bugdroid.....	30
Figure II.5: Architecture générale d'Android	31
Figure II.6 : Table illustrant la part de chaque version d'Android.....	33
Figure II.7 : La machine Dalvik	33
Figure II.8 : Les différentes parts du marché en pourcentage	34
Figure II.9 : Vue générale d'une application	36
Figure II.10 : Le home	38
Figure II.11 : Exemple de notification.....	39

Chapitre III

Figure III.1 : Architecture préconisé de l'application	42
Figure III.2 : Architecture détaillée de notre application	43
Figure III.3 : Diagramme de séquence pour le cas « envoyer un email »	46
Figure III.4: Diagramme de séquence pour le cas « recevoir un email ».....	47
Figure III.5 : Diagramme de séquence pour le cas « activer ou désactiver un service ».....	49
Figure III.6 : Diagramme de séquence pour le cas « afficher la liste des emails »	50
FIGURE III.7: Diagramme de classe de la partie Serveur.....	51

Liste des Figures

Figure III.8: Diagramme de classe de la partie client.....	53
---	----

Chapitre IV

Figure IV.1: Interface principale d'éclipse	59
Figure IV.2 : Choix de la version de SDK Manager.....	60
Figure IV.3 : Installation des packages.....	61
Figure IV.4 : Android Virtual Device Manager.....	62
Figure IV.5 : Exemple d'émulateur	63
Figure IV.6: Le fichier MANIFEST	64
Figure IV.7 : Activités principale de l'application.....	65
Figure IV.8: Envoyer un email	66
Figure IV.9 : Affichage du toast	66
Figure IV.10: Activer le service passerelle.....	67
Figure IV.11 : Désactiver le service	68
Figure IV.12 : Notification	69
Figure IV.13 : Affichage du message	70
Figure IV.14 : Affichage de la liste des messages	71
Figure IV.15 : Activer le service.....	71
Figure IV.16 : Désactiver le service	72

Liste des acronymes

- SMS : Short Message Service.
- AMPS: Advanced Mobile Phone System.
- TACS: Total Access Communications System.
- NMP: Nordic Mobile Téléphone.
- GSM: Global System for Mobile communications.
- CDMA : Code Division Multiple Access.
- TDMA: Time Division Multiple Access.
- GPRS: General Packet Radio System.
- UMTS: Universal Mobile Telecommunications System.
- GPP: 3rd Generation Partnership Project.
- SIM: Subscriber Identity Module.
- USIM: Universal Subscriber Identity Module.
- PDA: Personal Digital Assistant.
- LED : Light Emitting Diode.
- GPRS: General Packet Radio Services.
- EDGE: Enhanced Data rates for GSM Evolution.
- BSS: Base Station Sub-system.
- BTS : Base Transceiver Station.
- NSS: Network Station Sub system.
- RTCP: Real time Transport Control Protocol.
- RNIS: Réseau Numérique à Intégration de Services
- Le MSC : Mobile Services Switching Center.
- Le HLR : Home Location Register.
- Le VLR : Visitor Location Register.
- L'AuC : Authentication Center.
- ADSL: Asymmetric Digital Subscriber Line.
- RTC: Réseau Téléphonique Commuté.
- OSI: Open Systems Interconnection.

Liste des acronymes

- IP : Internet Protocol.
- TCP : Transmission Control Protocol.
- UDP: User Datagram Protocol.
- HTTP : Hyper Text Transfer Protocol.
- FTP : File Transfer Protocol.
- FTPS: File Transfer Protocol Secure.
- SMTP : Simple Mail Transfer Protocol.
- IMAP : Internet Message Access Protocol.
- IRC : Internet Relay Chat.
- NNTP: Network News Transfer Protocol.
- DNS : Domain Name System.
- ICMP : Internet Control Message Protocol.
- SSL: Secure Sockets Layer.
- TLS: Transport Layer Security.
- FPGA : Field Programmable GateArray.
- ASIC: Application Specific Integrated Circuits.
- IHM : Interface homme machine.
- CPU: Central Processing Unit.
- ASIC: Application Specific Integrated Circuit.
- SoC: System on Chip.
- eCos: Embedded Configurable Operating System.
- GPS : Global Positioning System.

Introduction

générale

Introduction Générale

présentation de l'environnement dans lequel notre application a été développée, les outils utilisés ainsi que quelques interfaces de cette dernière.

combinaison de technologies (Etc) et de plateformes (depuis de fonctionnalités de l'application) et de la prise en compte des MSN/Mail à introduire pour l'adaptation de la technologie mobile plus vital que jamais

Chapitre I: Communications Mobiles.

Introduction :

Après l'utilisation de moyens primitifs, commençant avec les premiers services postaux organisés, le développement du télégraphe, du téléphone, des communications sans fil puis numériques. C'est sans doute un des domaines où la technologie a évolué le plus rapidement. Les développements récents dans le domaine des communications sans fil et l'introduction de l'internet dans les réseaux cellulaires ont permis aux utilisateurs de gagner du temps grâce à la mobilité de leurs téléphones. L'utilisateur peut à présent consulter son courrier électronique ou encore faire des téléchargements de fichiers, de sonneries, de musique ...etc.

Dans ce chapitre nous allons aborder les différents moyens technologiques qui assurent cette nouvelle mobilité. Nous allons d'abord présenter la technologie SMS, la technologie internet, les différentes générations des réseaux mobiles. Pour parler ensuite des réseaux GSM, GPRS et les services offerts par ces derniers. Pour finir nous parlerons de la passerelle Mail SMS.

I.1 Réseaux mobiles :

Un réseau de téléphonie mobile est un réseau téléphonique qui permet l'utilisation simultanée de millions de téléphones sans fil, immobiles ou en mouvement, y compris lors de déplacements à grande vitesse et sur une grande distance.

Dans ce qui suit nous allons citer les différentes générations de la téléphonie mobile.

I.1.1 Les générations des réseaux mobiles :

Communiquer convenablement est certainement l'une des activités humaines qui paraît le plus simple, le plus naturel qui est en fait très compliqué.

Les réseaux de communication sont apparus depuis déjà très longtemps, véhiculant de l'information d'une source vers une destination. Du télégraphe des années 40 aux systèmes mobiles actuels, l'évolution a été considérable. Plusieurs technologies ont coexistées, notamment dans la téléphonie pour le transport de la voix. Peu après, les réseaux ont évolués pour transporter d'autres types d'informations telles que les données informatiques ou la vidéo.

Dans ce qui suit nous présenterons les différentes générations, plus particulièrement la deuxième génération.

I.1.1.1 La première génération (1G) :

La première génération de téléphonie mobile repose sur un système de communications mobiles analogiques et est constituée d'appareils relativement volumineux.

Il n'existe pas de technologie dominante dans la première génération mais plusieurs technologies concurrentes. La plus ancienne est le système américain AMPS (Advanced Mobile Phone System). D'autres systèmes furent développés après comme le TACS (Total Access Communications System) mis en place dans le royaume uni et le NMP (Nordic Mobile Téléphone) scandinave. Ces systèmes sont caractérisés par l'utilisation des transmissions analogiques ayant une faible bande passante, une couverture restreinte à quelques régions et une absence totale d'interopérabilité. La communication est entièrement dédiée à la parole.

I.1.1.2 La deuxième génération (2G):

La seconde génération des réseaux a marqué une rupture avec la première génération des téléphones cellulaires grâce au passage de l'analogique vers le numérique, elle permet d'offrir des services de transmission et de données en plus de la parole.

Plusieurs standards ont été proposés essentiellement le GSM (Global System for Mobile communications), le CDMA (Code Division Multiple Access) et le TDMA (Time Division Multiple Access).

Grâce aux réseaux 2G, il est possible de transmettre la voix ainsi que des données numériques de faible volume, notamment des messages textes (SMS, pour Short Message Service) ou encore des messages multimédias (MMS, pour Multimedia Message Service).

Des extensions de la norme GSM ont été mises au point afin d'en améliorer le débit. C'est le cas notamment du standard GPRS (General Packet Radio System), qui permet d'obtenir des débits théoriques de l'ordre de 114 kbit/s, plus proche de 40 kbit/s dans la réalité. Cette technologie ne rentre pas dans le cadre de l'appellation « 3G » a été baptisée 2.5G.

I.1.1.3 La troisième génération (3G):

Elle permet des services de communications plus rapides notamment la voix, la télécopie et 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, ...etc.

La principale norme utilisée dans la troisième génération est l'UMTS qui a pour objectif non seulement la réalisation d'une couverture à l'échelle planétaire mais également

d'unifier les normes de transport d'information sur les réseaux mobiles et apporter une technologie permettant des débits élevés autorisant l'exploitation des services multimédia sur des périphériques mobiles.

Actuellement, le développement de ce standard est supervisé par 3GPP. Le réseau cœur est gardé compatible avec le GSM. Pour le réseau d'accès, il utilise une nouvelle plage de fréquences n'interférant pas avec ce dernier. Ils peuvent donc fonctionner ensemble dans la même surface géographique. Il en résulte que les utilisateurs peuvent continuer à utiliser leurs cartes SIM GSM dans les réseaux UMTS mais ils seront obligés d'utiliser des terminaux mobiles 3G.

I.1.1.4 La quatrième génération :

La technologie 4G est la nouvelle génération des standards téléphoniques, en voie de développement à travers le monde, y compris en France où elle n'est pas accessible pour l'instant que dans quelques grandes agglomérations. Alors qu'au sein du réseau 3G, les données internet et conversations téléphoniques étaient séparées, le débit devant alors se partager entre les utilisateurs connectés, la 4G réunit l'ensemble de ces données. Cela garantit un transfert de données de meilleure qualité.

Comme pour le passage de la 2G à la 3G, les terminaux mobiles doivent être adaptés à la nouvelle génération 4G, ce qui est déjà le cas pour bon nombre de produits qui ont été mis sur le marché récemment. En effet, qu'il s'agisse d'une clé mobile ou d'un smartphone, leur adaptation aux nouveaux protocoles IPv6, fournis par la connexion 4G, est nécessaire. De leur côté, les opérateurs commencent peu à peu à proposer des forfaits adaptés.

I.2 SMS (short message service)

Le minimessage est le premier service génération de GSM qui est rapidement devenu un moyen de communication très populaire.

Les SMS ont pris le pas sur d'autres moyens de communication, et offrent aux clients plus de liberté et d'instantanéité. Les SMS sont aujourd'hui majoritairement utilisés dans les circonstances où l'écrit est le mieux adapté en particulier lorsque l'on a besoin de transmettre un message à une personne sans vouloir la déranger (réunion, heure tardive...) ou bien lorsque son environnement immédiat ne permet pas une conversation téléphonique dans de bonnes conditions (bus, train, lieux bruyants). Mais de plus en plus les SMS sont aussi utilisés

pour partager des émotions et permettre l'attention sympathique, le témoignage d'affection : souhaiter bon anniversaire, adresser ses félicitations.

I.2.1 Définition :

Plus connu sous le sigle de SMS ou texto, il permet de transmettre des messages de plusieurs millions de caractères découpés en sous messages se composant de 160 caractères alphanumériques si on utilise des données latines, de 140 caractères si on utilise des données binaires et de 70 caractères pour les langues non latine comme l'arabe ou chinois.

I.2.2 Bref historique :

A l'origine, les SMS ont été inventés par une équipe finlandaise dans le but d'aider les personnes malentendantes à communiquer.

Le SMS a été lancé en 1992 (le premier SMS commercial qui a été envoyé) et devenu jusqu'à ce jour le service d'envoi de donnée sans fil qui a connu le plus grand succès.

I.2.3 Rédaction :

Sur les téléphones classiques deux mode de saisie peuvent être disponible

- **Le mode ABC** : on utilise le clavier de manière classique (chaque touche correspond à plusieurs caractères).
- **Le mode intuitif** : dit T9, dans ce mode les pressions sur plusieurs touches (5 pressions pour un mot de 5 lettres) correspondent à quelques centaines de combinaisons possibles, seules celles correspondant à un mot connu du terminal sont proposées au choix final.

I.2.4 Classes des SMS [17]:

Un SMS reçu sur le mobile est traité de manière différente suivant sa classe. La classe est définie dans le SMS Data CodingScheme :

- **classe 0** : (flash SMS) le message est directement affiché à l'utilisateur sur l'écran du mobile à la réception. Un rapport est envoyé ensuite au centre de service. Le message n'est enregistré ni dans la mémoire du téléphone ni dans la carte SIM. Il est effacé dès que l'utilisateur a validé la visualisation.

- **classe 1** : le message est enregistré dans la mémoire du téléphone et si cette mémoire est pleine, dans la carte SIM par défaut.
- **classe 2** : le message est enregistré sur la carte USIM. Un accusé de réception est envoyé au centre de service une fois que le message a bien été transféré sur l'USIM.
- **classe 3** : le message est transféré sur un équipement externe connecté au mobile (PDA, PC portable...).

Le SMS est défini pour l'envoi d'un message court d'un mobile à un autre. Depuis une dizaine d'années, il existe également la possibilité d'envoi de SMS à partir d'un PC connecté. Ce type de message est souvent pour les usagers de marketing et de promotion. Il permet d'envoyer des SMS vers un grand nombre de destinataires.

I.2.5 Evolution des domaines d'utilisation :

De nos jours les SMS sont également utilisés dans les communications machine a machine, par exemple il existe des afficheurs à LED contrôlés par des SMS. Par ailleurs, de nombreuses entreprises ou collectivités locales utilisent des services d'envoi de SMS par internet (envoi d'un SMS par courrier ou via une requête http), par exemple de nombreuses villes préviennent automatiquement leurs habitants par SMS lorsque leurs passeports ou cartes d'identités sont disponibles.

Des solutions d'envoi de SMS par internet sont également utilisées dans les entreprises, les associations ou les collectivités locales par des envois ponctuels (confirmer un RDV)

I.3 Le réseau GSM :

Actuellement le réseau GSM français compte pas moins de 58 millions d'utilisateurs. Le portable est devenu en quelques années un produit de consommation courante. Ce petit trésor de technologie ouvre la porte à de nombreuses applications électroniques sans fil à celui qui sait l'interfacer avec un PC.

I.3.1 Définition :

Le GSM (global system for mobile communications) est un système cellulaire et numérique de seconde génération pour la téléphonie mobile.

Les services avancés et l'architecture du GSM ont fait de lui un modèle pour la troisième génération des systèmes cellulaires, le réseau UMTS.

La norme GSM a été étendue par l'organisation 3GPP pour prendre en charge de plus hauts débits et le transport de données en mode « paquet » par les extensions GPRS (General Packet Radio Services) puis EDGE (Enhanced Data rates for GSM Evolution). Ces 2 modes peuvent cohabiter avec le mode « voix commutée » du GSM et utilisent les mêmes antennes et les mêmes bandes de fréquence.

I.3.2 Architecture organisationnelle d'un réseau GSM :

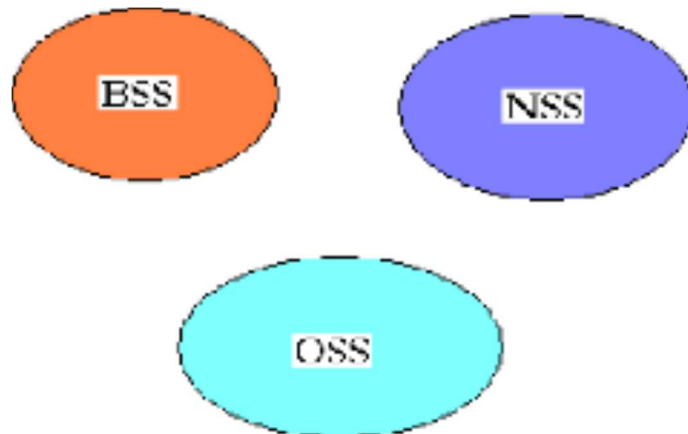


Figure I.1: Architecture organisationnelle d'un réseau GSM.

I.3.3 Architecture fonctionnelle d'un réseau GSM :

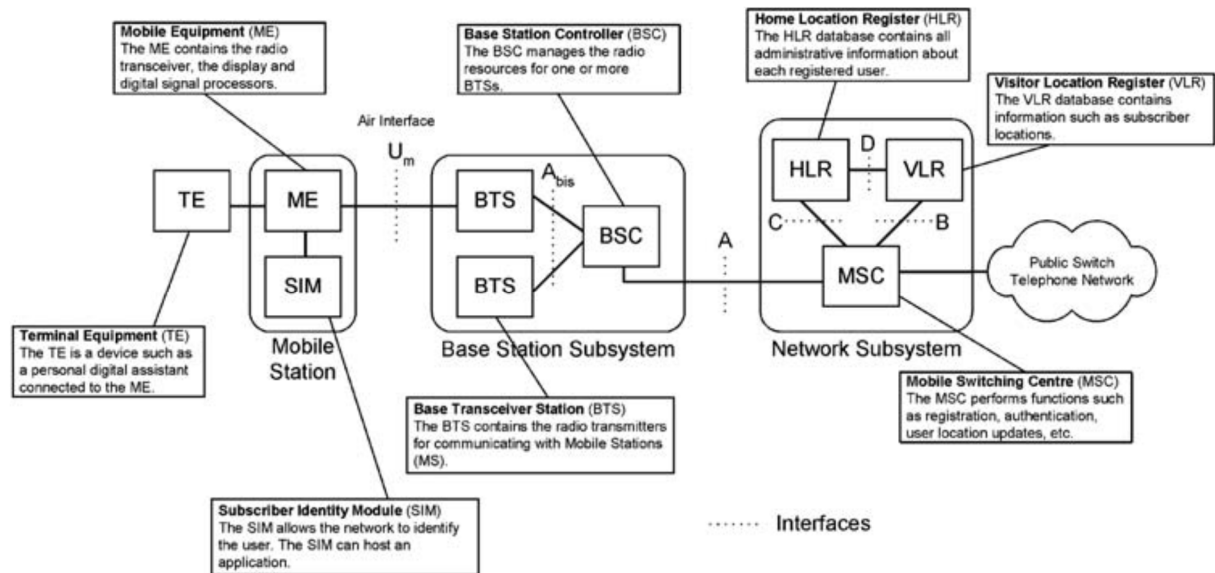


Figure I.2 : Architecture fonctionnelle d'un réseau GSM.

I.3.3.1 Le sous-système radio : BSS (Base Station Sub system) :

Sa fonction principale est la gestion de l'attribution des ressources radio indépendamment des données de leur identité ou de leur communication.

On distingue dans BSS :

➤ **La station de base BTS (Base Transceiver Station) :**

Elle permet le dialogue avec le mobile sur l'interface Air (aussi appelée interface Radio ou interface U_m).

I.3.3.2 Le sous-système réseau NSS (Network Station Sub system) :

Il assure principalement les fonctions de commutation et de routage. C'est donc lui qui permet l'accès au réseau public RTCP ou RNIS. En plus des fonctions indispensables de commutation, on y retrouve les fonctions de gestion de la mobilité, de la sécurité et de la confidentialité qui sont implantées dans la norme GSM.

On distingue dans NSS :

➤ **Le MSC** (Mobile Services Switching Center) :

C'est la partie centrale du NSS. Il prend en charge l'établissement des communications vers les abonnés GSM. Du fait de la mobilité, l'implantation de la seule fonction de commutation n'est pas suffisante. Le MSC gère la mobilité et les fréquences et enregistre la localisation des abonnés visiteurs (base de données VLR).

➤ **Le HLR** (Home Location Register) :

C'est la base de données qui gère les abonnés d'un PLMN donné. Elle contient toutes les informations relatives à l'abonnement et aux droits d'accès. D'autre part, le HLR est une base de données de localisation. Il mémorise pour chaque abonné le VLR où il est enregistré.

➤ **Le VLR** (Visitor Location Register) :

C'est la base de données qui gère les abonnés présents dans une certaine zone géographique. Ces informations sont une copie de l'original conservé dans le HLR.

➤ **L'AuC** (Authentication Center) :

Il mémorise pour chaque abonné une clé secrète utilisée pour authentifier les demandes de services et pour le chiffrement des communications. Un autre est en général associé à chaque HLR.

I.3.3 Le sous-système opérationnel :

Il a pour rôle la gestion et la supervision du réseau. C'est la fonction dont l'implémentation est laissée avec le plus de liberté dans la norme GSM.

Dans notre cas le système opérationnel n'est pas représenté.

I.3.4 Services offerts par GSM :

Un réseau GSM offre une gamme des services disponibles sur un réseau moderne (voix, données, fax et messagerie). La messagerie est l'un des services offert par le réseau GSM. La nature du message est variable, elle dépend du réseau et de la capacité déterminant de l'abonnement.

On regroupe les services offerts par le réseau GSM en 3 groupes. Le groupe le plus important et de loin les services vocaux (services d'appels) avec la téléphonie avec différents débits de parole possible et même des débits variables. Dans le groupe des services vocaux il y a aussi les appels d'urgences, procédure standardisée en Europe avec le 112 contrairement à

la messagerie vocale qui ne fut identifié comme besoin que plus tard dans le réseau GSM. Le deuxième groupe peu utilisé à ses débuts et qui explose maintenant est le groupe des services de messages courts (services SMS). Il y a bien sur les services de messages courts point à point c'est à dire la transmission de message de 160 caractères de ou vers un mobile ainsi que la diffusion de messages courts vers des mobiles entrant dans une zone particulière de couverture. Ce dernier service est utilisé notamment pour des raisons publicitaires. Enfin plus récemment est apparu le message dit multimédia qui ajoute à la transmission de messages courts un caractère multimédia comme des sonneries ou des images voir de la vidéo. Enfin le dernier groupe est celui des services supplémentaires qui sont essentiellement des services de maintenance et surtout des services de transmission de donnée dans notre cas en GSM uniquement en mode connecté, le mode paquet n'apparaissant en effet que dans le GPRS.

I.4 le réseau GPRS (General Packet Radio Service)

I.4.1 Définition :

GPRS est né d'une évolution de la deuxième génération des réseaux mobiles. C'est une technologie orientée paquet destinée à fonctionner sur des réseaux GSM permettant un débit de donnée plus élevé.

I.4.2 Commutation par paquet :

Le principal avantage de la commutation de paquet réside dans le fait que les ressources radio ne sont utilisées que lorsque les utilisateurs émettent ou reçoivent des données.

Dans le cas de GPRS, le canal de données est partagé entre les utilisateurs au rythme des émissions/réceptions de données de chacun. Cette technologie permet d'une part d'optimiser l'utilisation des ressources radio et d'autre part de minimiser le temps de transmission des données.

I.4.3 Services offerts par le GPRS :

En terme de service, le GPRS offre la possibilité de faire du point à point en mode <avec connexion> comme en mode <sans connexion> et du point à multipoint. Ce dernier peut être de type broadcast, service est distribué aux utilisateurs d'une même zone de couverture ou multicast semblable au multicast des réseaux IP, pour lequel le service est distribué aux utilisateurs d'un groupe, quelle que soit leur position géographique.

La technologie GPRS est une technologie de transport et d'interfaçage de réseau prenant en compte les contraintes du réseau GSM .GPRS permet donc de mettre en place un

tunnel IP entre le périphérique et un serveur d'information .le mode de fonctionnement autorise par ailleurs une connexion permanente au réseau, donc éventuellement une connexion permanente à internet. Cette connexion permanente à internet pourrait également permettre d'implanter des applications de type serveur dans les périphériques portables comme par exemple un mini serveur web capable de fournir quelques pages web de type home page, ou bien des dispositifs d'écoute mobile par le net ,on pourrait également envisager des dispositifs de localisation géographique en exploitant des informations au niveau de la couche GSM.

I.5 Technologie d'internet :

Les technologies internet évoluent rapidement, sont largement répandues et font l'objet de multiples usage dans la population et les organisations.

L'adaptation des technologies d'internet entraînent des changements à grande échelle sur la communication humaine au sens le plus large.

I.5.1 Définition :

Internet est le réseau informatique mondialement accessible au public qui a révolutionné le monde des ordinateurs et des communications comme rien d'autre auparavant.

C'est un réseau constitué de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux.

I.5.2 L'évolution de l'internet :

Internet évolue dans trois domaines :

- **Nouveaux supports transmission :** qui sont en particulier les canaux satellite et le câble.

Le travail vise à permettre aux paquets IP d'être acheminés de façon performante pour obtenir de " l'Internet sur satellite " (et donc un accès Internet via une parabole) ou de " l'Internet sur câble " (et donc un accès Internet via le même support que la télévision par câble). Un problème important vient du fait que ces supports sont asymétriques. Par exemple, il est possible pour un utilisateur de recevoir des paquets IP d'un satellite avec une simple parabole, mais il est impossible de renvoyer des paquets directement vers le satellite. Or la très grande majorité des applications (courrier électronique, web, audio et vidéo conférence, jeux)

supposent un échange de données de façon bidirectionnelle entre les participants, d'où la nécessité d'un travail spécifique pour intégrer ces nouveaux supports.

- **Support de la mobilité :**

L'explosion du nombre des ordinateurs portables nécessite un traitement adéquat de la mobilité. En fait, la notion de mobilité est vaste et couvre au moins trois situations : les ordinateurs portables qui peuvent être transportés d'un point à un autre et reconnectés à différents points du réseau, les ordinateurs mobiles qui restent connectés (via un réseau radio à paquets ou un réseau de téléphonie cellulaire), et les réseaux mobiles (reliant par exemple les capteurs d'une automobile en mouvement à une station de diagnostic). Le travail essentiel dans le domaine de la mobilité vise donc à développer des mécanismes efficaces pour acheminer des paquets entre des utilisateurs en mouvement, chose non triviale car il faut d'abord repérer la position des utilisateurs, les identifier, et enfin trouver un chemin les reliant.

- **Support d'un nombre croissant d'utilisateurs de l'Internet :**

Chaque utilisateur (plus précisément, chaque interface réseau) est caractérisée par une adresse IP qui est théoriquement unique dans tout le réseau (donc dans le monde entier). Donc le nombre d'adresses à allouer augmente avec le nombre d'utilisateurs, qui lui-même augmente de façon exponentielle. Or, le protocole IP actuel ne permet d'allouer qu'un nombre d'adresses relativement faible. Ce problème (et d'autres liés à la taille des tables de routage) ont donné lieu au développement d'un nouveau protocole IP, appelé IP6 (IP version 6). IP6 contient autant d'adresses qu'il existe d'atomes dans l'univers, ce qui devrait suffire pour au moins quelques années. De plus, IP6 a été conçu pour faciliter la mise en place de services autres que le " best effort " et mieux adaptés aux applications qui ont des contraintes temporelles fortes (comme les applications multimédia, l'audioconférence, ou les jeux). IP6 a également été conçu pour être compatible avec IP4 afin que la transition d'IP4 vers IP6 soit fluide.

I.5.3 Les différentes technologies d'accès à internet :

I.5.3.1 Connexions filaires [5]:

- **ADSL** :(Asymetric Digital Subscriber Line) désigne la liaison numérique sur la ligne de l'abonné téléphonique. Technologie de connexion de loin la plus répandue.

La possibilité d'avoir en même temps une conversation téléphonique et une connexion internet haut débit sur une même ligne téléphonique est rendue possible par l'utilisation de bandes de fréquence différentes, séparées chez l'abonné ainsi qu'au central téléphonique par des filtres spéciaux.

Cette technologie (ADSL / Câble) apparaît toutefois optimale au niveau rapport qualité/prix pour la plupart des besoins de PME (petites et moyennes entreprises).

- **RTC :** le Réseau Téléphonique Commuté qui est utilisé pour nos conversations téléphoniques. Ces dernières y sont transmises par des signaux analogiques.

Dans les débuts de l'accès à internet pour le grand public un modulateur-démodulateur, ou modem, installé généralement dans l'ordinateur de l'internaute, opérait la conversion entre les modes analogique et numérique de réception ou d'envoi des données. Cette technologie offre toutefois des débits très faibles

- **Le câble :** c'est une technologie dédiée à la distribution des flux télévisuels.

De multiples adaptations ont permis d'augmenter la capacité montante du réseau, et donc notamment vers internet, ce qui a amené au développement d'une réelle offre haut débit.

- **La fibre optique :** La fibre optique est constituée d'un fil en verre ou en plastique ayant la propriété de conduire la lumière. Elle offre un débit de transfert de données nettement supérieur à celui des câbles coaxiaux et peut donc être utilisée tout à la fois pour les liaisons téléphoniques, télévisuelles ou informatiques.

I.5.3.2 Connexions mobiles à internet :

- **GPRS :** Le General Packet Radio Service ou GPRS est une norme pour la téléphonie mobile dérivée du GSM, elle permet un débit de données allant de 40 kbit/s à un maximum théorique de 171 kbit/s. Elle transmet les données par paquet en empruntant simultanément plusieurs canaux GSM.

Le GPRS est accessible partout, et offre une grande couverture géographique ainsi il convient aux personnes souhaitant envoyer des emails ou consulter internet à distance.

- **WIFI :** un réseau WIFI 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 sur un rayon de quelques dizaines de mètres autour d'une antenne .

I.5.4 Protocoles logiciels :

Les protocoles logiciels utilisés sur internet sont les conventions structurant les échanges d'informations nécessaires au transfert des contenus applicatifs pour l'utilisateur final. Ils permettent notamment d'identifier les interfaces (donc les machines), de s'assurer de la réception des données envoyées, et de l'interopérabilité.

Internet fonctionne suivant un modèle en couches, similaire au modèle OSI. Les éléments appartenant aux mêmes couches utilisent un protocole de communication pour s'échanger des informations.

Chaque protocole a des fonctions propres et, ensemble, ils fournissent un éventail de moyens permettant de répondre à la multiplicité et à la diversité des besoins sur Internet.

Les principaux protocoles sont classés selon leur couche (IP, TCP et UDP) ; couches applicatives :

- IP (Internet Protocol) aussi appelé IPv4 : protocole réseau qui définit le mode d'échange élémentaire entre les ordinateurs participant au réseau en leur donnant une adresse unique sur celui-ci. Cependant, en raison du nombre d'internautes croissant, une nouvelle norme voit le jour ; nommée IPv6, elle permet d'accueillir un plus grand nombre d'utilisateurs.
- TCP : responsable de l'établissement de la connexion et du contrôle de la transmission. Il s'assure que le destinataire a bien reçu les données, au contraire d'UDP.
- HTTP (HyperText Transfer Protocol) : protocole mis en œuvre pour le chargement des pages web.
- HTTPS : pendant du HTTP pour la navigation en mode sécurisé.
- FTP (File Transfer Protocol) : protocole utilisé pour le transfert de fichiers sur Internet.
- FTPS (File Transfer Protocol Secure) : pendant du FTP pour le transfert de fichiers sécurisés
- SMTP (Simple Mail Transfer Protocol) : mode d'échange du courrier électronique en envoi.
- POP3 (Post Office Protocol version 3) : mode d'échange du courrier électronique en réception.

- IMAP (Internet Message Access Protocol) : un autre mode d'échange de courrier électronique.
- IRC (Internet Relay Chat) : protocole de discussion instantanée.
- NNTP (Network News Transfer Protocol) : protocole de transfert de message utilisé par les forums de discussion Usenet
- SSL ou TLS : protocoles de transaction sécurisée, utilisés notamment pour le paiement sécurisé.
- UDP : permet de communiquer, de façon non fiable mais légère, par petits datagrammes.
- DNS (Domain Name System) : système de résolution de noms Internet.
- ICMP (Internet Control Message Protocol) : protocole de contrôle du protocole IP.

À la suite de l'épuisement des adresses IPv4, le protocole IPv6 a été développé. Celui-ci dispose d'un espace d'adressage considérable.

Les téléphones mobiles sont le fruit d'une évolution technologique, ils sont allés au-delà de leur rôle primitif d'outils de communication et ont progressé pour devenir plus fonctionnels (détection d'alerte, réception d'emails...etc). Au-delà de la voix ils offrent d'autres services qui doivent être explorés et exploités tels que le service SMS et le service e-mail.

I.6 La passerelle SMS Mail :

Le message SMS est très populaire, de plus en plus utilisé dans le monde professionnel des affaires.

L'interfonctionnement entre SMS et e-mail est activé en permettant la conversion d'un SMS message vers un e-mail message et vice versa. Cette conversion est effectuée grâce à une passerelle email.

I.6.1 Définition :

Une passerelle SMS/MAIL permet à un ordinateur d'envoyer ou de recevoir des SMS vers ou à partir d'un réseau de télécommunications. La plupart des messages sont finalement acheminés dans les réseaux de téléphonie mobile. De nombreuses passerelles SMS se charge de la conversion des médias de l'email et d'autres formats.

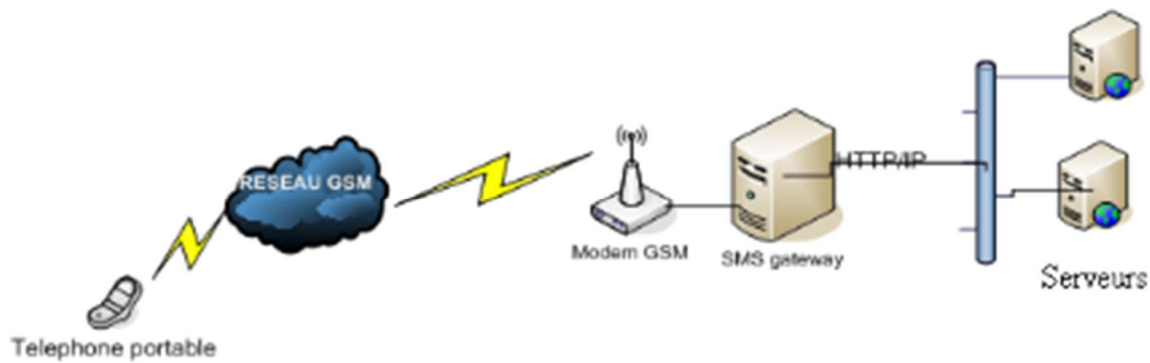


Figure I.3 : Représentation d'une passerelle.

I.6.2 GSM appareil de passerelle :

Une passerelle est un dispositif qui a intégré la connectivité sans fil GSM. Il permet à des messages texte SMS d'être envoyés ou reçus par courrier, à partir de pages Web ou d'autres applications de logiciels par l'acquisition d'un identifiant unique de Subscriber Identity Module du téléphone mobile, ou "carte SIM".

La connexion au réseau de téléphonie mobile est effectuée par l'acquisition d'un numéro de carte SIM de l'opérateur mobile et l'installer dans la passerelle. En règle générale, les appareils directs à mobile passerelle sont utilisés pour des centaines de milliers de messages texte par mois. Les appareils modernes offrent maintenant la capacité de transmettre jusqu'à 100 000 messages chaque jour. Plusieurs vendeurs qui, historiquement, ont fourni un matériel pour la passerelle GSM pour la voix ont aussi la capacité de SMS. Certains sont plus primitifs que d'autres. Les dispositifs les plus capables sont conçus avec la gestion des SIM pour réguler le nombre de SMS par SIM, ODBC pour se connecter à une base de données, interfaces et HTTP pour interagir avec des applications tierces.

I.6.3 Les différents protocoles utilisés dans les passerelles SMS/mail :

- **SMTP (Simple Mail Transfer Protocol) :** est un protocole de communication utilisé pour transférer le courrier électronique (courriel) vers les serveurs de messagerie électronique. SMTP est un protocole assez simple (comme son nom l'indique). On commence par spécifier l'expéditeur du message, puis le ou les destinataires d'un message, puis, en général après avoir vérifié leur existence, le corps du message est transféré.

SMTP ne permet pas de récupérer à distance des courriels arrivés dans une boîte aux lettres sur un serveur. Les standards Post Office Protocol (POP) et IMAP ont été créés dans ce but.

- **POP3** : permet de récupérer les courriers électroniques situés sur un serveur de messagerie électronique. En dehors d'un paramétrage spécifique, POP se connecte au serveur de messagerie, s'authentifie, récupère le courrier, peut effacer le courrier sur le serveur et se déconnecte.

La figure qui suit montre l'organisation du protocole POP3

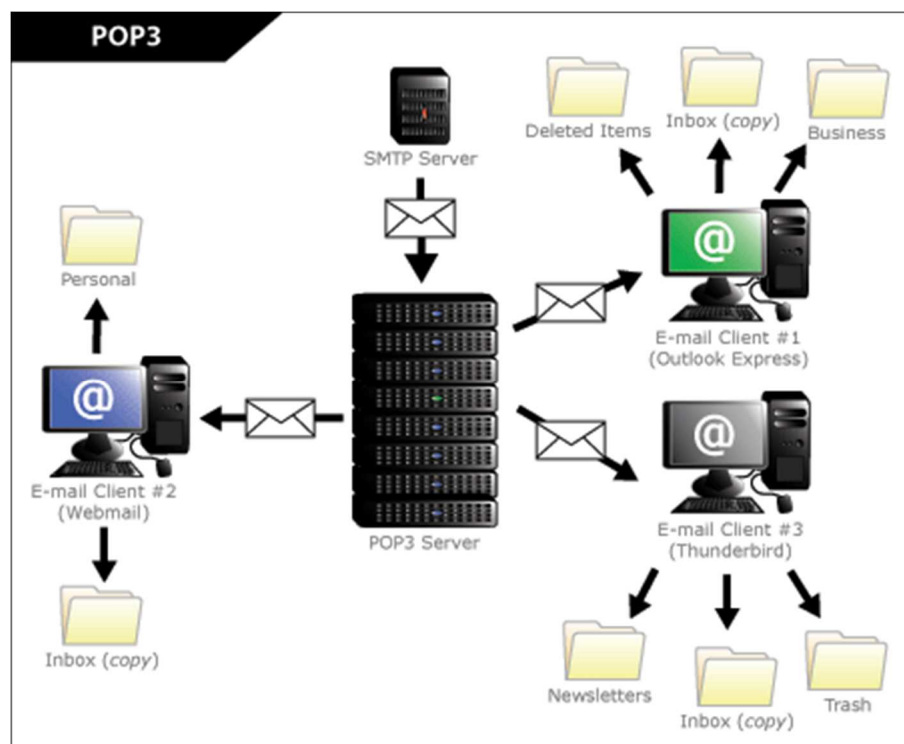


Figure I.4 : Organisation du protocole POP3.

POP3 stocke tout le courrier entrant dans une boîte de réception sur le serveur central qui est ensuite généralement téléchargé et organisé avec un client e-mail individuelle. Parce que toute l'organisation et les modifications ne sont pas stockés sur le serveur, les modifications apportées au compte e-mail ne seront reflétées que sur le client e-mail qui a fait les changements

- **IMAP (Internet Message Access Protocol)** : est un protocole qui permet de récupérer les courriels électroniques déposés sur des serveurs de messagerie. Ce protocole permet de laisser les courriels sur le serveur dans le but de pouvoir les consulter de différents clients de messagerie.

Le fait que les messages soient archivés sur le serveur fait que l'utilisateur peut y accéder depuis n'importe où sur le réseau et que l'administrateur peut facilement faire des copies de sauvegarde.

Dans ce qui suit nous allons présenter une figure illustrant le fonctionnement du protocole IMAP.

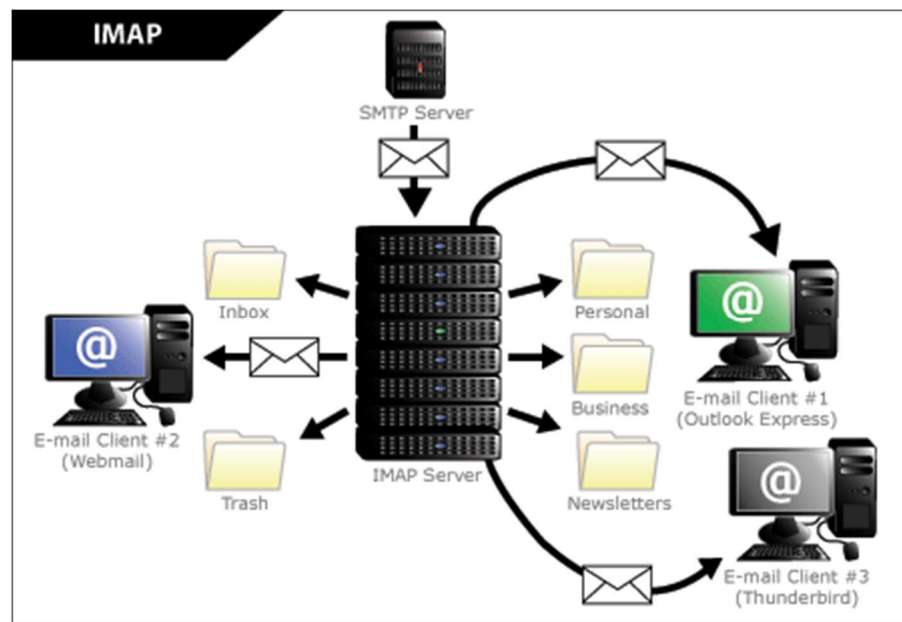


Figure I.5 : Organisation du protocole IMAP.

IMAP organise et stocke les e-mails dans des dossiers sur le serveur. L'e-mail est alors accessible et affichée à la même place quel que soit le client e-mail que vous utilisez (par exemple Outlook Express, Thunderbird, Webmail... etc). Parce que toutes les informations sont stockées sur le serveur, les modifications sur le compte e-mail (par exemple la création / suppression / changement de nom de dossiers ou de messages) sera automatiquement mis à jour pour tous les utilisateurs qui ont un accès à ce compte e-mail, indépendamment du client e-mail ou l'emplacement.

I.6.4 Services offerts par une passerelle SMS/mail :

Une passerelle mail SMS offre des solutions efficaces de traitement de messages, elle peut transformer des mails en SMS, ou bien générer des SMS automatiquement en partant de fichiers de type Excel. Elle s'intègre dans le système d'information de l'entreprise. Le journal des messages est mis à votre disposition sur la passerelle SMS.

La passerelle SMS, se charge de la réussite de nos communications SMS professionnel et marketing.

GPRS fourni l'établissement instantané de la connexion à internet, comme il permet de rester connecter en permanence jusqu'à ce qu'il se déconnecte en ne payant que lorsque les données sont transmises, par conséquent, les services à base de paquets GPRS devait coûter moins d'où l'appel à l'utilisation gratuit des SMS et l'apparition des passerelle SMS/Mail.

Conclusion :

Dans ce chapitre nous avons abordés les différentes technologies dont la technologie des SMS, d'internet et de la téléphonie comme nous avons vu les réseaux mobiles et leur évolution actuellement des outils ont été développés afin de faciliter la tâche aux programmeurs et leurs permettre d'effectuer tous les tests qui puisse y avoir. Dans le chapitre qui suit nous allons introduire les outils de développement mobile.

Chapitre II : Les Systèmes Embarqués.

Introduction :

Depuis quelques années, le marché de la téléphonie mobile connaît une véritable révolution, menée par Samsung, Apple et Nokia, d'où l'apparition de nouvelles technologies telle que « Android » afin de pouvoir développer des applications pour tous types d'appareils mobiles.

Dans ce chapitre nous allons introduire les systèmes embarqués, leurs architectures, leurs caractéristiques ainsi que leurs classifications d'une part et d'autre part nous allons présenter L'Android, son architecture générale, les différentes versions, puis nous allons parler de la machine virtuelle Dalvik qui et l'environnement d'exécution des applications Android. Nous finirons avec les différents éléments d'une application Android.

II.1 Système embarqué**II.1.1 Définition :**

Un système embarqué est un système complexe autonome et ne possède pas d'entrées sorties standards qui intègre des logiciels et le matériel dédiés à une tâche bien précise. Il contient un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés en mémoire.

Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables FPGA (Field Programmable GateArray), des circuits dédiés à des applications spécifiques ASIC (Application Specific Integrated Circuits) ou des modules analogiques sont utilisés en plus.

II.1.2 architecture générale d'un système embarqué :

Un système embarqué dispose d'une architecture semblable à celle des ordinateurs standards, en effet, on y retrouve les mêmes composants (processeur, mémoire, dispositifs d'entrée et sortie) interconnectés à l'aide des bus.

La figure qui suit représente une vue simplifiée d'un système embarqué.

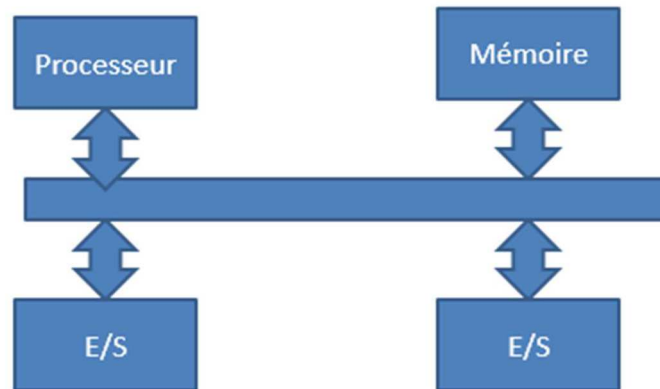


Figure II.1 : Vue simplifié d'un système embarqué.

L'architecture d'un système embarqué peut varier selon le système. Il est constitué de deux composants de base, une unité centrale de traitement (CPU) et un système d'exploitation, comme on peut trouver d'autres composants supplémentaires tels que les systèmes auxiliaires et l'interface IHM pour la reconfiguration du système.

Le fonctionnement du système se résume ainsi:

- Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique.
- L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC et éventuellement de système externes traite l'information
- Le traitement génère éventuellement une sortie qui est envoyée vers la sortie, les systèmes auxiliaire, les ports de monitoring ou l'IHM

La figure II.2 représente cette architecture.

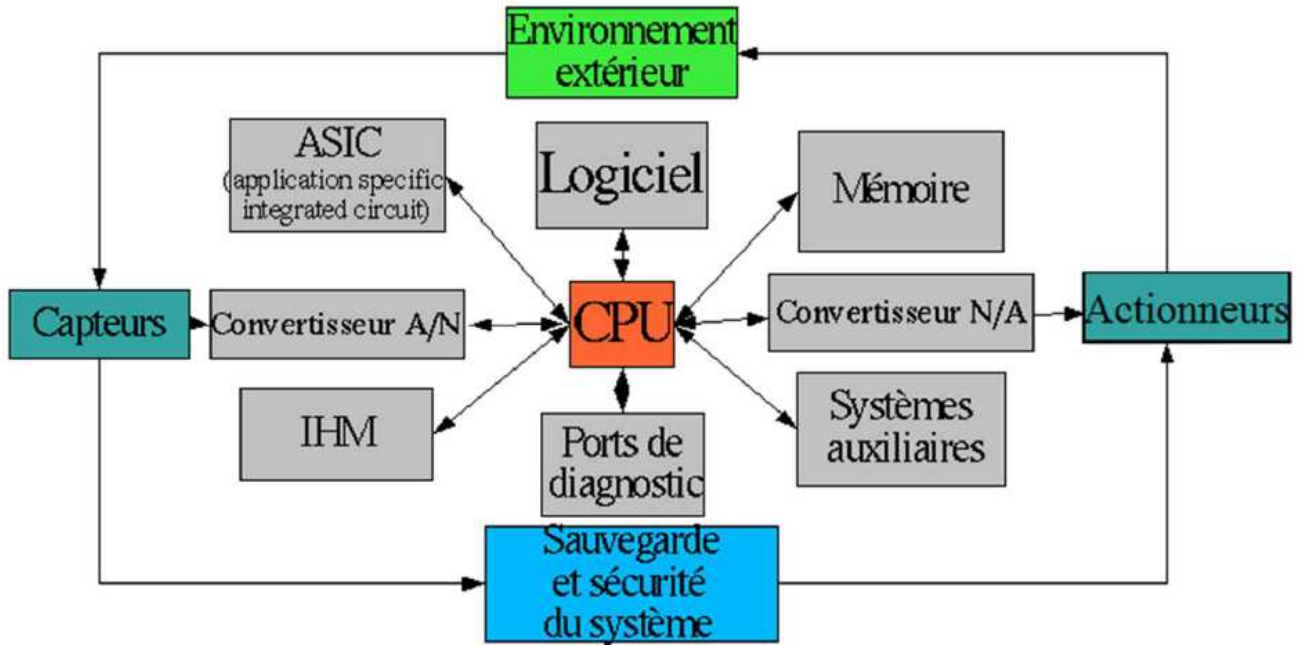


Figure II.2: Architecture typique d'un système embarqué.

II.1.3 Caractéristiques :

Contrairement aux systèmes universels, les systèmes embarqués sont dédiés pour effectuer des tâches précises. Certaines de ces tâches doivent répondre à des contraintes de temps réels (le temps de réponse de ces systèmes est aussi important que l'exactitude des résultats) pour des raisons de fiabilité et de rentabilité alors que d'autres permettent de simplifier le système en ayant peu de contraintes au niveau de la performance.

Les systèmes embarqués sont la plupart du temps enfouis dans des machines qui doivent fonctionner en continu pendant de nombreuses années sans erreurs. C'est pourquoi ce genre de système est autonome et le développement des logiciels se fait de façon minutieuse avec beaucoup d'attention.

Pour ce qui est de la communication, généralement les objets pour lesquels les systèmes embarqués sont conçus, sont des objets communicants, ainsi ces systèmes sont dotés de modules (Radio, GSM, GPS...) qui leur permettent de communiquer et de recevoir des informations sur de longues distances.

II.1.4 Complexité :

Les systèmes embarqués requièrent souvent un faible encombrement (faible poids) comme les PDA(Personal Digital Assistant). Leur technologie fait alors appel à l'électronique et à des applications portables où l'on doit minimiser aussi bien l'encombrement que la consommation électrique [1]. Par conséquent, la réalisation du packaging afin de faire cohabiter sur une faible surface de l'électronique analogique, de l'électronique numérique, des composantes RF (Radiofréquence) sans interférences est une tâche difficile. En effet, les performances des systèmes sur carte deviennent obsolètes dans le contexte des besoins actuels.

Dans les stratégies de conceptions actuelles, un système embarqué est généralement intégré sur un support silicium unique constituant ainsi un système complet intégré sur une puce SoC (System on Chip).

II.1.5 Classification des systèmes embarqués [1]:

On peut classer les systèmes embarqué comme suit :

- **Système transformationnel :**

Assure des activités de calcul, Il lit ses données et ses entrées lors de son démarrage, puis fournit ses sorties, et meurt ensuite.

- **Système interactif :**

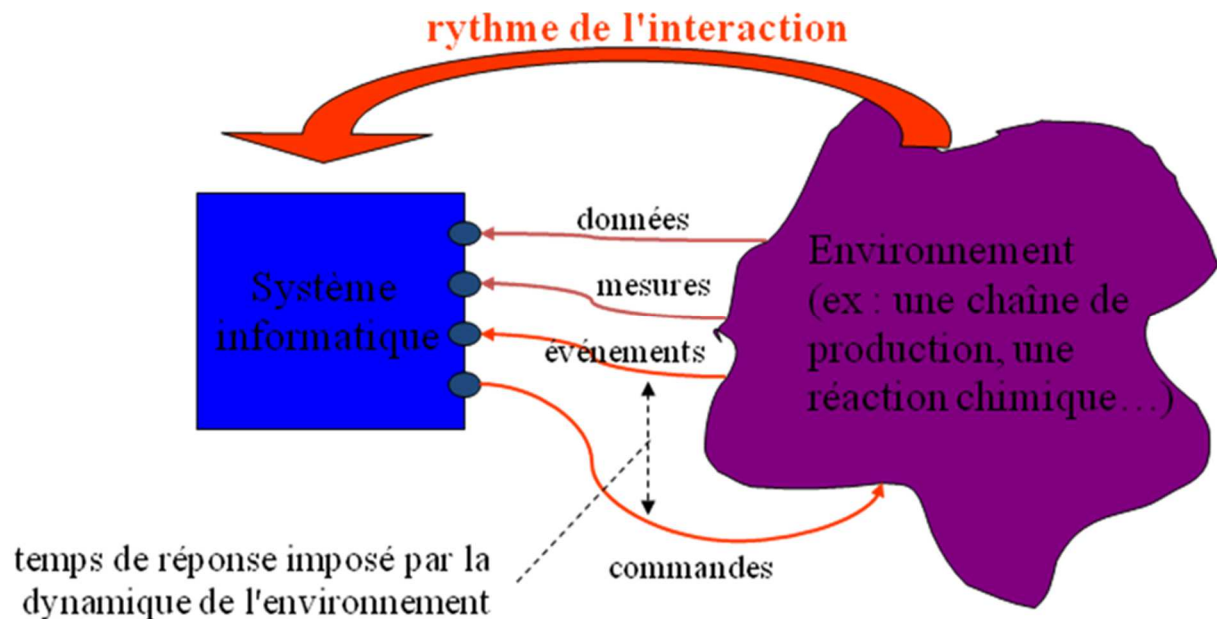
Système en interaction quasi permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés), le rythme de l'interaction est déterminé par le système et non par l'environnement.

- **Système réactif :**

Système en interaction permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passées), mais le rythme de l'interaction est déterminé par l'environnement et non par le système.

- **Systèmes embarqués temps réels :**

Un système est dit temps réel lorsque l'information après acquisition et traitement reste encore pertinente cela veut dire que dans le cas d'une information arrivant de façon périodique (interruption), les temps d'acquisition et de traitement doivent rester inférieurs à la période de rafraichissement de cette information.



FigureII.3 : Architecture générale d'un système embarqué temps réel.

II.1.6 Types de systèmes embarqués temps réel :

On distingue deux types de systèmes embarqués temps-réel d'abord il y a les systèmes temps réels stricts/durs (hard real time) qui respectent des échéances précises, le non-respect de ces échéances peut avoir des conséquences graves sur le fonctionnement du système ou sur son environnement (auto pilotage, freinage, assistance médicalisée...). Les échéances ne doivent jamais être dépassées. En deuxième lieu il y a les systèmes temps réels mou (soft real time) qui sont moins stricts dans le sens où le non-respect des échéances ralentit le système sans conséquences graves (billetterie automatique...). Le système doit répondre au mieux, le plus rapidement possible.

La plupart des systèmes embarqués sont dit mutlirate ou multi-période

- Les données sont capturées à un certain rythme
- Les traitements sur ces données ne sont pas forcément à la même granularité
- Différents traitements peuvent intervenir de manière indépendante
- Les actionneurs fonctionnent à une fréquence différente

II.1.7 Les langages de développement utilisés :

Pour des raisons évidentes de contraintes matérielles, le langage assembleur a longtemps été le choix de prédilection des technologies de l'embarqué car il permettait à la fois d'optimiser la taille du code généré mais aussi ses performances et il permettait aux développeurs d'avoir un contrôle total sur le processeur et le matériel.

L'évolution des performances du matériel et des compilateurs permet aujourd'hui de se tourner vers des solutions plus confortables. Aujourd'hui, le C est le langage de programmation le plus utilisé, en effet des compilateurs sont disponibles pour presque chaque processeur en usage, et il y a un très grand corps de programmeurs expérimentés en C. En outre, le C a l'avantage d'être indépendant du processeur ce qui permet aux programmeurs de se concentrer sur les algorithmes et les applications, plutôt que sur les détails de l'architecture d'un processeur particulier. Le langage C est bien adapté au logiciel embarqué car il permet une programmation relativement proche du matériel (à une nature « de bas niveau »), il donne aux programmeurs en systèmes embarqués un haut degré de contrôle direct sur le matériel sans sacrifier les avantages des langages de haut niveau.

Le langage C++ est aussi l'un des choix favori des développeurs, il représente la version orientée objet du C qui offre une meilleure abstraction des données.

D'autres langages orientés objet sont utilisés, et celui qui s'impose est le Java notamment JavaAndroid qui est utilisé pour développer des applications destinées au système qui porte le même nom (Android), et qui est l'un des systèmes les plus connus pour la téléphonie mobile est les tablettes.

II.1.8 Domaines d'utilisations des systèmes embarqués :

L'ingénierie des systèmes embarqués est un domaine très vaste, il est constamment tourné vers l'amélioration du confort et de la simplicité d'utilisation. D'une manière générale ce type de système est destiné à améliorer le confort d'utilisation de certains équipements, aussi ils sont utilisés dans le domaine de l'automobile, pour le contrôle du processus industriels, dans l'aéronautique, et dans les systèmes de communications tel que la téléphonie mobile, comme ils sont utilisés dans les systèmes Android qui seront présentés plus loin.

II.1.9 différents systèmes d'exploitation pour l'embarqué :

Le système d'exploitation (Operating System), est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeu vidéo, ...).

Dans ce qui suit nous allons présenter quelques systèmes d'exploitation pour l'embarqué :

➤ **TinyOS**

TinyOS est un système d'exploitation open-source, flexible conçu pour des réseaux de capteurs sans fil. Il respecte une architecture basée sur une association de composants, réduisant la taille du code nécessaire à sa mise en place. TinyOS a été conçu pour minimiser la consommation en énergie du capteur. Ainsi, lorsqu'aucune tâche n'est active, il se met automatiquement en veille.

➤ **MULTOS**

MULTOS est un système d'exploitation multi application, destiné aux cartes à puce. C'est un système hautement sécurisé, extrêmement stable et disposant de normes de sécurité élevées.

➤ **Linux embarqué**

Linux embarqué est un système d'exploitation basé sur Linux destiné aux systèmes avec ressources limitées. C'est une adaptation des fonctionnalités du noyau linux aux systèmes embarqués. Les systèmes embarqués sous Linux disposent généralement de peu de RAM et utilisent fréquemment de la mémoire flash plutôt qu'un disque dur. Comme ils sont souvent dédiés à un nombre de tâches réduites sur une cible matérielle bien définie.

➤ **eCos :**

eCos (embedded Configurable operating system) est un système d'exploitation temps réel. Il permet de réaliser des applications temps réel. Il est implémenté en C/C++ et contient des couches logicielles et API.

➤ **Android**

L'utilisation de l'OS Android s'étend progressivement à tous les secteurs d'activité directement concernés par les systèmes multimédia, comme l'automobile ou l'aéronautique.

II.2 Android :

II.2.1 Définition :

Des milliers (voir beaucoup plus) de personnes (plus ou moins professionnelles) développent des applications sous Android. Android est un système d'exploitation mobile open source conçu pour des tablettes tactiles, terminaux mobiles et smartphones.

Les services offerts par Android facilitent notamment l'exploitation des réseaux de télécommunications GSM, Bluetooth, WIFI et UMTS, la manipulation des médias, notamment de la vidéo, de l'audio et des images JPEG ainsi que d'autres formats, l'exploitation des senseurs tels que les capteurs de mouvements, la caméra, la boussole et le récepteur GPS, l'utilisation de l'écran tactile, le stockage en base de données, l'affichage de pages web, l'exécution multitâche des applications et l'envoi de messages SMS.

II.2.2 Historique [14]:

Android commence en octobre 2003, où la société Android Inc. est créée. Officiellement, elle développe des logiciels pour mobiles. Mais en réalité, elle se préparait à sortir un tout nouveau système d'exploitation pour Smartphones. En 2005, Google rachète cette entreprise, et sort une première version bêta en novembre 2007, avant de lancer la version 1.0 en septembre 2008 avec le HTC Dream. À partir de ce moment-là, le rythme des nouvelles sorties est très élevé : pas moins de 32 versions différentes sont apparues à notre jour.

La version actuelle est la 5.0, nommée " Lollipop", et est disponible depuis 2014. Toutefois, peu de Smartphones ont été mis à jour, cette tâche étant déléguée aux constructeurs.

Plus de 1 300 000 applications sont disponibles sur le marché en juillet 2014 et huit téléphones sur dix achetés au niveau mondial fonctionnent sous Android. Ce système d'exploitation est, en effet, gratuit pour les constructeurs, ce qui les incite à sortir de nouveaux Smartphones : de ce fait, plus de 30 marques proposent des Smartphones ou des tablettes : HTC, Samsung, Sony Ericsson, condor etc.

Android, même s'il est majoritaire, n'est pas le seul sur le marché des Smartphones. Il doit, en effet, faire face à la concurrence d'Apple son principal adversaire avec son iPhone. De plus, Google doit aussi compter sur la concurrence de Microsoft avec Windows Mobile 8, et RIM (Blackberry), qui espère gagner quelques parts du marché. Quant à Nokia et son système Symbian, il est en nette perte de vitesse car il n'a pas beaucoup évolué ces dernières années.

Le personnage nommé Bugdroid illustré dans la figure II.4 est la mascotte verte utilisée par Google et toutes les communautés Android pour représenter le système d'exploitation (OS).



Figure II.4 : Le Bugdroid.

II.2.3 Architecture générale d'Android :

L'architecture de l'Android se compose de 5 couches. Ce sont le noyau Linux, les bibliothèques et le moteur d'exécution Android, le cadre de l'application et la couche d'application. Plus précisément, Android est constitué de plusieurs parties dépendantes et nécessaires.

La figure II.5 présente les différentes couches de la pile logicielle d'Android

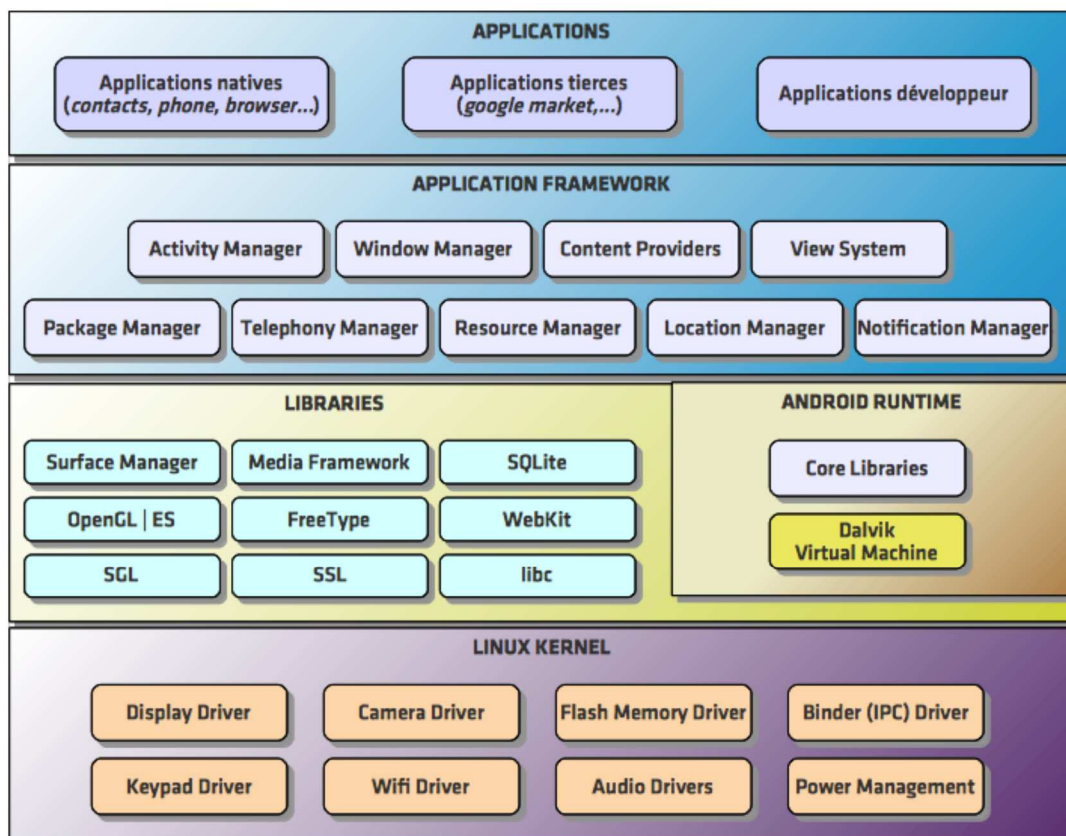


Figure II.5: Architecture générale d'Android.

Android bénéficie d'une architecture en couche complète faisant de lui une plateforme riche, dédiée aux appareils mobiles.

Il est à base du noyau linux qui permet de faire le pont entre la partie matérielle et la partie logicielle et profitant des services système de base tels que la sécurité, la gestion mémoire, gestion de processus, etc. A un niveau supérieur se trouvent un ensemble de bibliothèques écrites en C/C++ jouant le rôle d'un middleware (on en cite le système de bibliothèque C, les médiathèques, le SGL, etc.). C'est sur cette couche que se greffe l'Android Runtime, comprenant la machine virtuelle java et ses bibliothèques. Vient ensuite la plateforme logicielle, nommée aussi Framework de développement, écrite en java et permettant de mutualiser les ressources entre les applications Java. Elle offre aux développeurs la possibilité de produire des applications diverses et innovantes à travers un ensemble d'API.

Enfin, et à un niveau plus supérieur se situe un ensemble d'applications sous forme de paquets apk. Les applications fournies par Android sont telles qu'un navigateur web, un client mail, un calendrier, un gestionnaire de contacts, etc.

II.2.4 Les différentes versions d'Android :

Les différentes versions d'Android ont toutes des noms de desserts « en anglais » depuis la sortie de la version 1.5 et suivent une logique alphabétique (de A vers Z) : Apple Pie (Tarte aux pommes), version connue uniquement ou presque des développeurs car c'est la Sdk distribuée avant la sortie du premier téléphone Android.

1.0 : Apple Pie (Tarte aux pommes), sortie en 23 septembre 2008.

1.1 : Banana bread (Pain de la banane), sortie en 9 février 2009.

1.5 : Cupcake (Petit Gâteau), sortie en avril 2009.

1.6 : Donut (Beignet), sortie en 15 septembre 2009.

2.0 (2.0.1) : sortie le 26 octobre 2009, version appelée Éclair au départ mais, à cause de nombreux bugs, vite remplacée par la 2.0.1 puis par la 2.1. Cette version 2.0 est très peu connue.

2.1 : Éclair, sortie en janvier 2010.

2.3 (2.3.7) : Gingerbread (Pain d'épice), sortie le 6 décembre 2010, support la VOIP et SIP.

3.0 (3.2.6) : Honeycomb (Rayon de miel), sortie le 2 février 2011, version pour grandes tablettes et télévisions connectées.

4.0.1 (4.0.4) : IceCream Sandwich (Sandwich à la crème glacée), version unifiée pour Smartphone, tablette et Google TV, fortement inspirée d'Honeycomb, sortie le 19 octobre 2011.

II.2.5 Répartition actuelle des versions d'Android [18]:

Le tableau suivant résume la répartition des différentes versions Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	6.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.9%
4.1.x	Jelly Bean	16	17.3%
4.2.x		17	19.4%
4.3		18	5.9%
4.4	KitKat	19	40.9%
5.0	Lollipop	21	3.3%

Figure II.6 : Table illustrant la part de chaque version d'Android.

II.2.6 Le moteur d'exécution Android :

Dalvik est une machine virtuelle conçue dès le départ pour les appareils mobiles et leurs ressources réduites (peu de puissance de calcul et peu de mémoire). La majorité, voire la totalité des applications sont exécutées par la machine virtuelle Dalvik.

Les applications java, développées pour Android, doivent être compilées au format Dalvik exécutable (.dex) avec l'outil (dx). Cet outil compile les (.java) en (.class) et ensuite il convertit ces (.class) en (.dex).

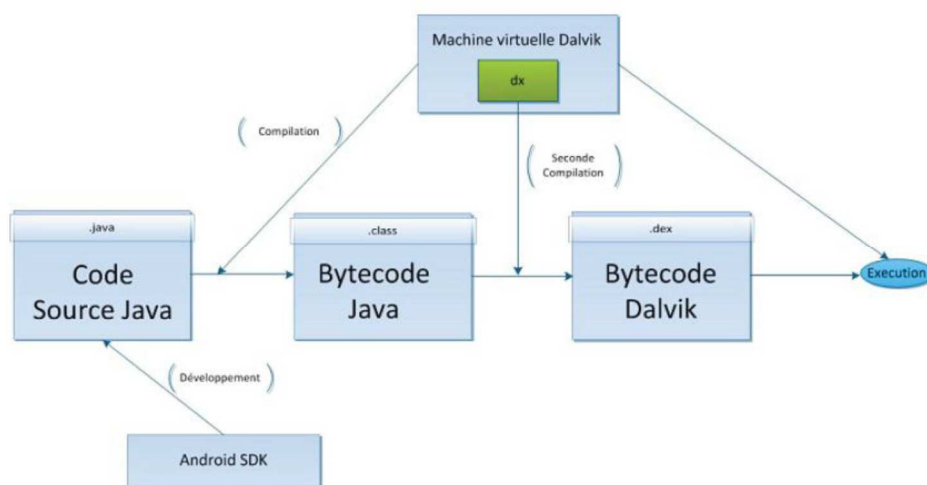


Figure II.7 : La machine Dalvik.

II.2.7 Concurrence et parts du marché :

Ses concurrents principaux sont Apple, le Windows Phone et le BlackBerry. Mais actuellement, Android dispose d'une très grande part du marché de la téléphonie avec environ 80%.

Dans l'ensemble, Android est une réussite dans son domaine, mais il n'est pas non plus à l'abri d'une remontée d'Apple qui pourrait toujours sortir un nouveau système ou appareil innovant.

Ci-dessous, un diagramme montrant les différentes parts du marché en pourcentage :

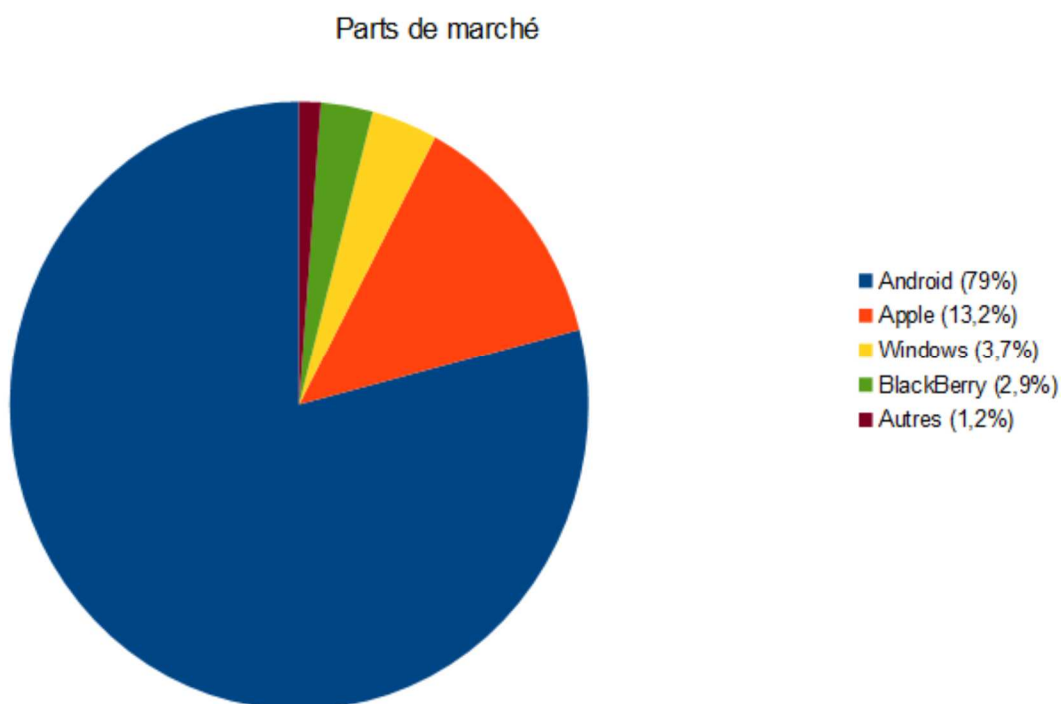


Figure II.8 : Les différentes parts du marché en pourcentage.

II.2.8 application Android

II.2.8.1 vue générale d'une application Android :

Une application Android est un assemblage de composants liés via un fichier de configuration à savoir :

- Les activités.
- Les vues et contrôles.
- Les ressources.
- Le fichier de configuration appelé également manifest.

- **Les activités (android.app.Activity):**

Il s'agit d'une partie de l'application présentant une vue à l'utilisateur. Une application peut avoir une ou plusieurs activités (par exemple pour une application de messagerie on pourrait avoir une Activity pour la liste des contacts et une autre pour l'éditeur de texte).

- **Les vues et contrôles :**

Les vues sont les éléments de l'interface graphique que l'utilisateur voit et sur lesquels il pourra agir. Les vues contiennent des composants organisés selon diverses mises en pages. Quant aux contrôles (boutons, champs de saisie, case à cocher...) sont eux-mêmes un sous-ensemble des vues. Ils ont besoin d'accéder aux textes et aux images qu'ils affichent, ces textes et ces fichiers seront puisés dans les fichiers ressources de l'application.

- **Le fichier de configuration :**

C'est un indispensable à chaque application qui décrit entre autres :

- Le point d'entrée de notre application (quel code doit être exécuté au démarrage de l'application).
- Quels composants constituent ce programme.
- Les permissions nécessaires à l'exécution du programme (accès à internet, accès à l'appareil photo, accès au WIFI pour notre cas).

Une illustration explicative de ces concepts est représentée par le schéma de la figure II.9.

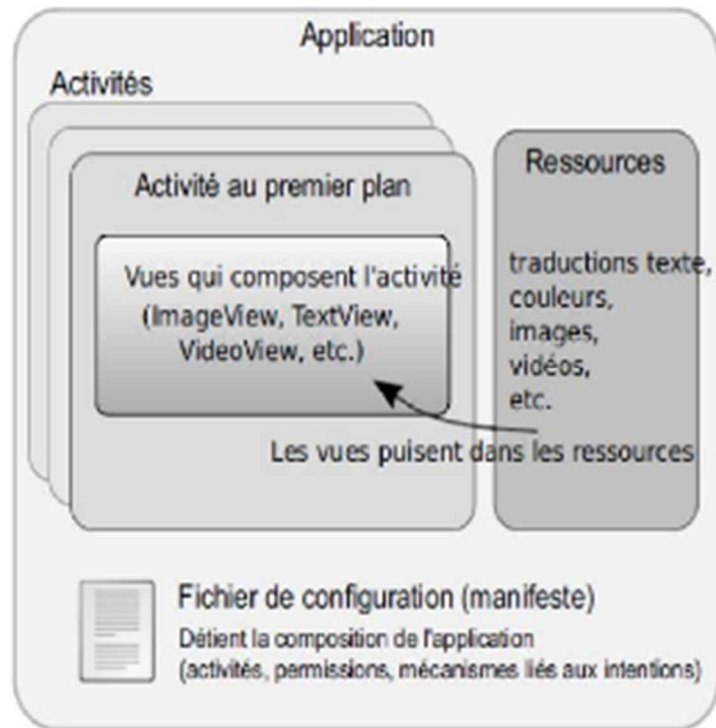


Figure II.9 : Vue générale d'une application.

II.2.8.2 vue détaillée :

Plus une application est complexe, plus de nombre de composants utilisés sera grand, dont ce qui suit nous allons présenter d'autres composant principaux d'une application Android :

- Services.
 - Fournisseurs de contenu.
 - Widget.
 - Objet Intent.
 - Récepteurs d'Intent.
 - Notification.
- des services : Les services n'ont pas d'interface graphique et tournent en tâche de fond. Il est possible de s'inscrire à un service et de communiquer avec celui-ci en utilisant l'API Android.
 - des fournisseurs de contenus: permettent le partage d'informations au sein d'une application et entre applications
 - des widgets: une vue accrochée au Bureau d'Android

- des Intents: permettent d'envoyer un message pour un composant externe avec ou sans le nommer explicitement
- des récepteurs d'Intents: permettent à une application d'être à l'écoute des autres afin de répondre aux objets Intent qui lui sont destinés et qui sont envoyés par d'autres composants applicatifs (par exemple changement de fuseau horaire, appel entrant...).
- des notifications: permettent de notifier l'utilisateur de la survenue d'événements.

Les tableaux 1 et 2 représentent les différents composants applicatifs et les mécanismes d'interaction et leurs classes respectives :

Nom	Classe ou paquetage concerné
Activité	android.app.Activity
service	android.app.Service
Fournisseurs de contenu (content provider)	android.content.ContentProvider
widget	android.appwidget.*

Tableau 1 : Les composants applicatifs et les classes associé.

Nom	Classe concernée
Intent	android.content.Intent
Récepteur d'Intents (BroadcastReceiver)	android.content.BroadcastReceiver
Notification	android.app.Notifications

Tableau 2 : Les mécanismes applicatifs et les classes associé.

II.2.9 Concept et innovation d'Android :

II.2.9.1 Bureau virtuel (home):

Le bureau virtuel est l'application principale du système, il est étendu sur plusieurs parties, Chacune est personnalisable, il est possible d'y mettre des raccourcis (vers des applications, des fichiers, des dossiers, des contacts, etc.) ou des widgets. L'image de fond s'étend aussi sur les différentes parties et bouge légèrement quand on change de partie, ce qui donne l'impression que le contenu fait partie du décor.

La figure suivante présente un exemple du home



Figure II.10 : Le home.

II.2.9.2 Les Widgets :

Il est possible de placer des widgets (calendrier, horloge, notes, ...) sur le bureau, ils permettent de jeter un œil rapide sur une application installée et lui accordé un accès direct à certaines fonctionnalités de l'application sans avoir à l'ouvrir, ou même ouvrir l'application.

Parmi les plus utilisés, nous avons donc : la galerie photo, le lecteur musical, l'horloge, la météo, l'agenda, la gestion de batterie etc. Et ce qui est génial, c'est de pouvoir les choisir, les modifier, les placer sur la page de notre choix.

II.2.9.3 Centralisation des notifications :

La barre du haut reçoit les statuts (réseaux utilisés, niveau de batterie, modes vibreur/sonnerie/silencieux, alarme, etc.) et les notifications. Celles-ci proviennent des applications, et peuvent avertir de l'arrivée d'emails, de SMS, d'appels en absence, ou bien afficher des informations relatives à Google Talk, à l'avancement des téléchargements en cours, etc.

Dans ce qui suit nous allons présenter une figure qui illustre un exemple de barre de notification.

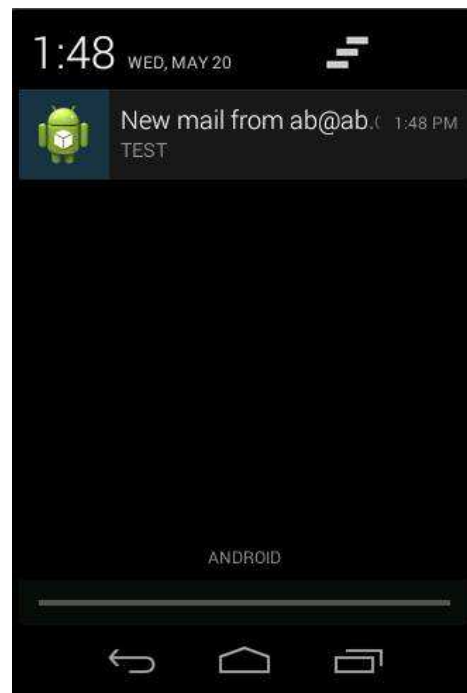


Figure II.11 : Exemple de notification.

Conclusion :

Dans ce chapitre nous avons introduit les systèmes embarqués, nous avons aussi présenté quelques systèmes d'exploitation utilisés dans le domaine en particulier Android, pour cela, nous avons donné son architecture et ses différents éléments.

Dans le chapitre suivant, nous décrirons notre approche qui aboutira à réaliser notre passerelle SMS/Mail dont une partie doit fonctionner sur le système Android.

Chapitre III : Analyse et Conception.

Introduction

Les SMS ont pris racine dans l'ensemble des tranches d'âge. Très utilisé dans les services (alertes bancaires), dans les médias (alerte infos), ils sont un moyen de communication, d'information et de promotion pour tout type d'entreprise.

Dans ce chapitre nous présentons notre contribution à la technologie passerelle SMS/Mail. Pour ce faire, nous allons étudier le problème pour entamer une analyse de nos besoins. Nous passerons ensuite à la partie conception logicielle en s'appuyant sur les résultats de la phase d'analyse afin de pouvoir résoudre le problème.

III.1 Objectifs de notre travail :

L'objectif principal de notre travail est d'offrir aux utilisateurs la possibilité d'envoyer et de récupérer leurs courriels électronique sur un mobile Android sans avoir accès à internet, en utilisant le SMS comme passerelle. Notre application est constituée de deux parties la partie client développée sous Android et la partie serveur implémentée sur un PC équipé d'une liaison internet et d'une clé GPRS.

III.2 Analyse et conception :

L'analyse et la conception de toute solution informatique est d'une grande importance. 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 analyse qui permet d'identifier les différents besoins. Vient ensuite la conception qui, en s'appuyant sur les résultats de la phase analyse, donnera une description plus ou moins détaillé.

Pour ce faire nous allons nous inspirer du langage UML afin de mieux expliquer les caractéristiques structurelles et comportementales de notre application.

III.3 Analyse :

Il est primordiale à la conception de tout système informatique de choisir le modèle d'architecture qui lui sera adéquat. Ce modèle doit assurer un bon fonctionnement, de meilleures performances ainsi que la réutilisation et l'interconnexion fiable de ce système avec d'autres.

Dans notre application on distingue deux parties bien différentes mais liées entre elles. La partie s'exécutant sur un équipement mobile (un smartphone Android) et la partie exécutée

sur un ordinateur relié à l'internet. De ce fait nous avons opté pour l'architecture client/ serveur. La figure III.1 donne l'architecture préconisée pour notre application.

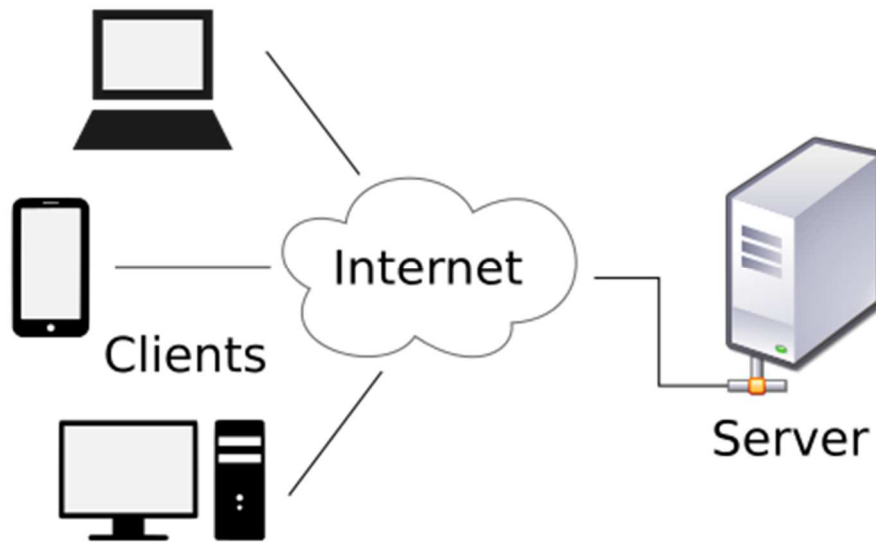


Figure III.1 : Architecture préconisé de l'application.

L'architecture client/serveur désigne un mode de communication entre plusieurs composants d'un réseau. Chaque entité est considérée comme un client ou un serveur. Chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

La figure III.2 donne l'architecture détaillée de notre application

III.3.1 Architecture générale :

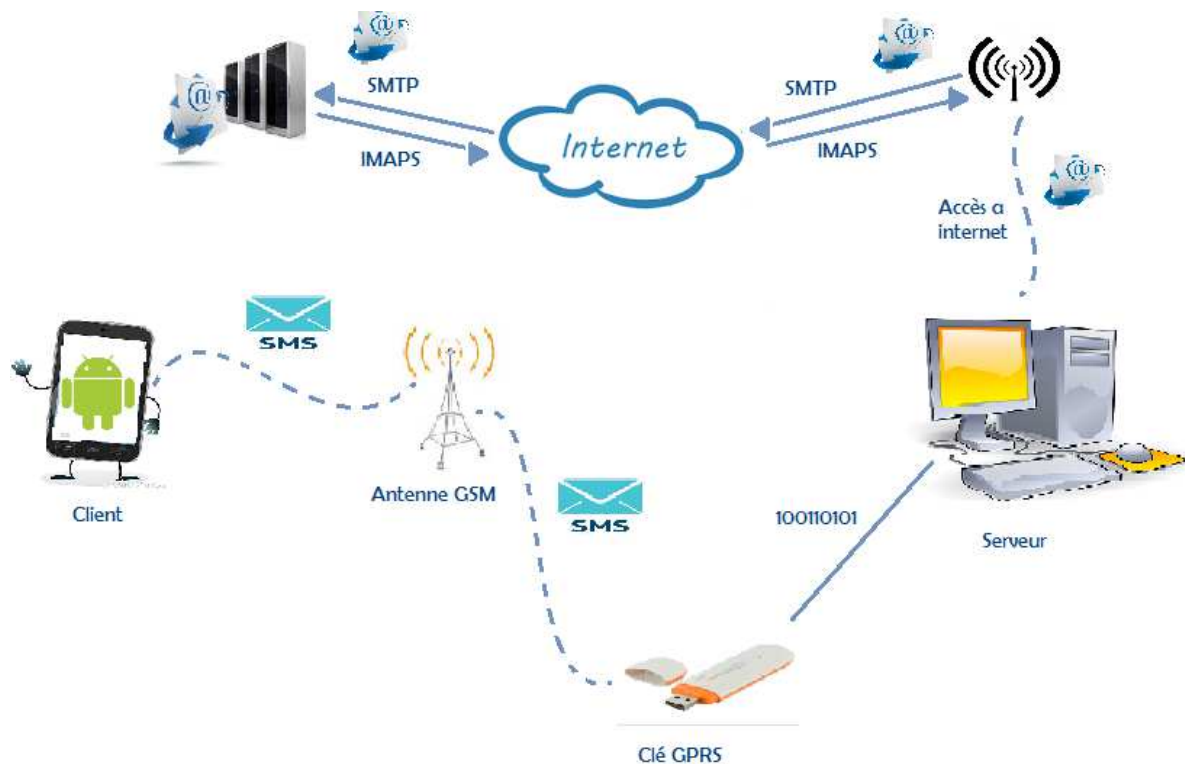


Figure III.2 : Architecture détaillée de notre application.

III.3.2 Fonctionnement du système :

Comme cité précédemment, notre système est divisé en deux parties distinctes, la partie client et la partie serveur :

❖ La partie client :

Cette partie est implémentée sur notre appareil Android. Elle s'occupe d'écrire des emails et de les envoyer par SMS vers le serveur de façon transparente pour l'utilisateur, l'envoi ou la réception du SMS se fait en arrière-plan (l'utilisateur a l'impression d'envoyer ou de recevoir un email). A la réception d'un nouvel SMS celui-ci est examiné pour savoir si ce dernier correspond à un Email auquel cas, il sera traité et sauvegardé dans notre base de données. Par la suite une notification est affichée sur l'écran du mobile de l'utilisateur lui permettant de l'afficher. Par ailleurs, si ce dernier désire répondre à ce message, en l'absence d'une connexion internet, une interface est mise à sa disposition afin de spécifier l'adresse de destination, l'objet

et le contenu de son message. Ces informations seront utilisées pour être envoyées au serveur en format SMS via le réseau GSM.

❖ **La partie serveur :**

Cette partie de l'application est implémentée sur un ordinateur. Dans un premier temps, à la réception d'un SMS sur la clé GPRS, ce dernier (le SMS) est récupéré par l'application serveur via le port série. Le serveur s'occupe de la reconstitution de l'email, et cela en extrayant les informations contenues dans le SMS. Par la suite, l'email est transféré vers l'adresse de destination via internet.

D'autre part, lors de la réception d'un message dans la boîte mail, le serveur s'occupe de le convertir en SMS afin de l'envoyer à l'utilisateur via la clé GPRS.

III.3.3 spécification des besoins :

L'utilisateur de notre application cliente pourra accéder aux fonctionnalités suivantes :

- lancer l'application.
- Activer le service passerelle.
- Envoie d'un email.
- Réception d'un email.
- Affichage d'un email.
- affichage de la liste des emails.
- Suppression d'un email.
- Suppression de tous les emails.
- Désactiver le service passerelle.

Lors de l'activation du service, un processus en arrière-plan est à l'écoute de l'arrivée de nouveaux messages. Lors de la désactivation du service l'écoute est arrêtée et les SMS sont pris en charge par l'application Android dédiée.

Sur la partie serveur, notre application offre les services suivants

- Récupère les emails (de l'internet) et les SMS (de la clé GPRS)
- traite (conversion de l'email en SMS ou du SMS vers l'email).
- Envoie du SMS ou de l'email résultant.

III.4 Conception :

Dans cette partie nous allons proposer notre solution en définissant le fonctionnement du système et les différentes étapes que nous avons suivi afin de mener à bien notre travail.

III.4.1 Diagrammes de séquences :

Un diagramme de séquences est un diagramme qui représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquences met toujours l'accent sur l'ordre chronologique des messages.

Les composants d'un diagramme de séquences 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 de ces flèches figure un texte nous informant du message envoyé entre les objets.

Ci-dessous nous présentons quelques diagrammes de séquences :

III.4.1.1 Diagramme de séquence pour envoyer un email

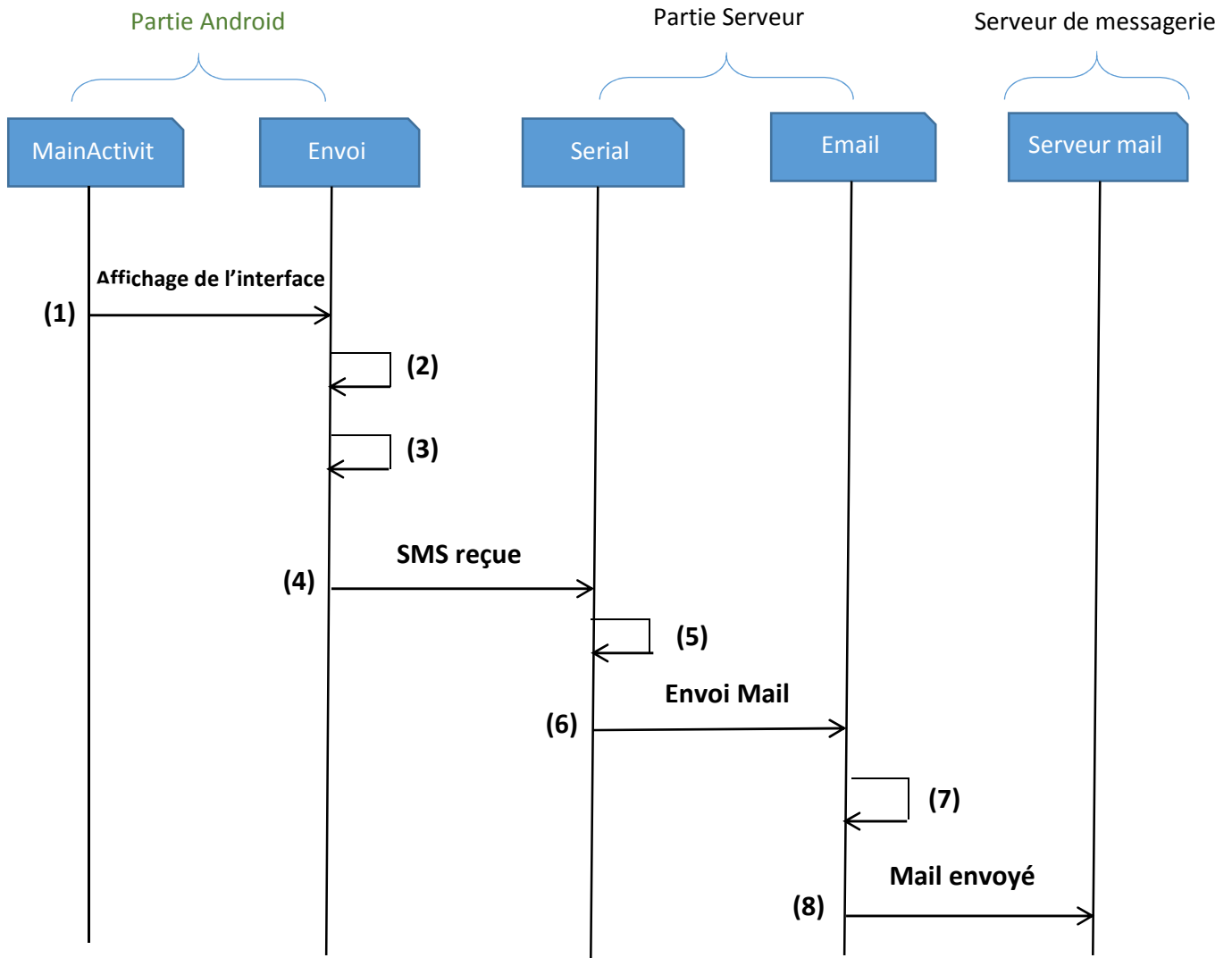


Figure III.3 : Diagramme de séquence pour le cas « envoyer un email ».

Description :

- (1) : lorsque on clique sur le bouton Envoi l'activité MainActivity fait appel à l'activité Envoi.
- (2) : récupération des valeurs de chaque champ adresse, objet et message de l'interface.
- (3) : concaténer les valeurs et leur ajoute le caractère '#' ou début et les séparer par le caractère '?' (#adresse@adresse.dz?objet?message).
- (4) : envoi le résultat de la concaténation par SMS vers la puce de la clé GPRS.
- (5) : lecture du SMS sur le port série.

- (6) : la méthode « appelenvoie() » de la classe serial fais appelle à la méthode envoi de la classe mail, en lui donnant le contenu du SMS comme paramètre.
- (7) : extraction de l'adresse, de l'objet et du message du contenu du SMS.
- (8) : envoi de l'email à l'adresse spécifiée.

III.4.1.2 Diagramme de séquence pour recevoir un email :

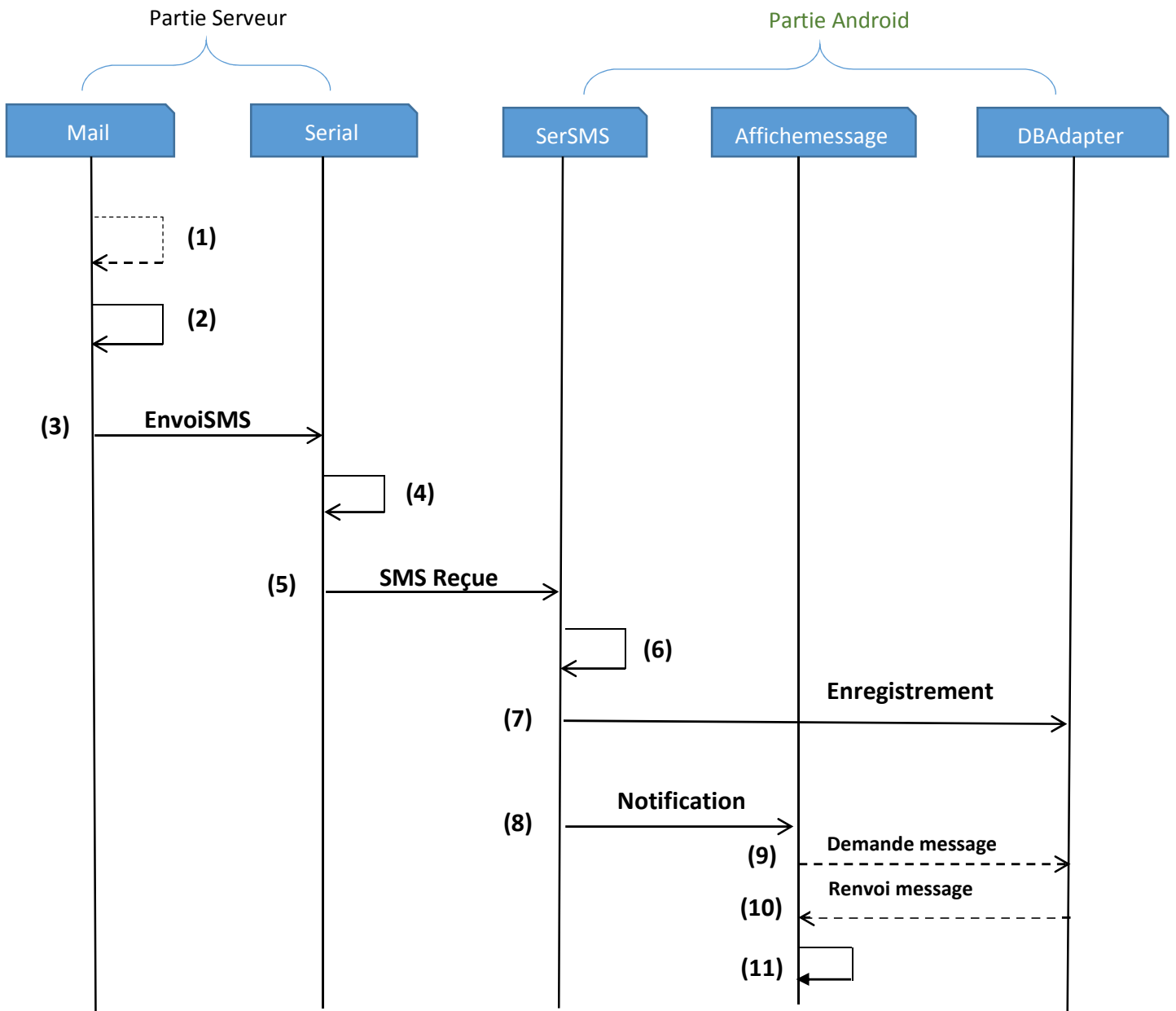


Figure III.4: Diagramme de séquence pour le cas « recevoir un email ».

Description :

- (1) : connexion au serveur mail et récupération du dossier INBOX (boite de réception)
- (2) : si il y a des mails non lu, l'adresse de l'émetteur de l'email ainsi que l'objet et le message sont récupéré, ces derniers seront concaténés et séparés par des « ? », puis “# “ sera ajouter au début (#adresse@emetteur.dz?objet?message).
- (3) : la méthode « recu » de la classe « mail » fait appel à la méthode « envoiSMS » de la classe serial, avec le résultat de la concaténation comme argument.
- (4) : la méthode « envoiSMS » envoi la commande « AT+CMGS= “+213.....”/n » à la clé GPRS, et attend “>” comme réponse de cette dernière, puis la méthode lui envoi le résultat de la concaténation comme argument suivi de «ctrl+c », enfin le SMS sera envoyé via la clé.
- (5) : le service « ServSMS » reçoit le SMS envoyé par la clé GPRS, puis vérifie si il commence par un “#“, ensuite l'adresse, l'objet et le message seront extrait par le service.
- (6) : les données extraites sont enregistrés dans la base de données en appelant la méthode « insererunsms » de la classe « DBAdapter ».
- (7) : notification à l'utilisateur qu'il y a un nouveau message qui est reçu.
- (8) : lorsque l'utilisateur clique sur la notification, cette dernière fait appel à l'activity « Affichemessage », et à l'aide des extra elle va lui donner la position du message à afficher.
- (9) : l'activity « Affichemessage » demande les données de l'email qui est à une certaine position à la classe « DBAdapter » avec la méthode « recupererEmail(position) ».
- (10): renvoi des données de la classe « DBAdapter ».
- (11) : affichage des données (adresse, objet, contenu).

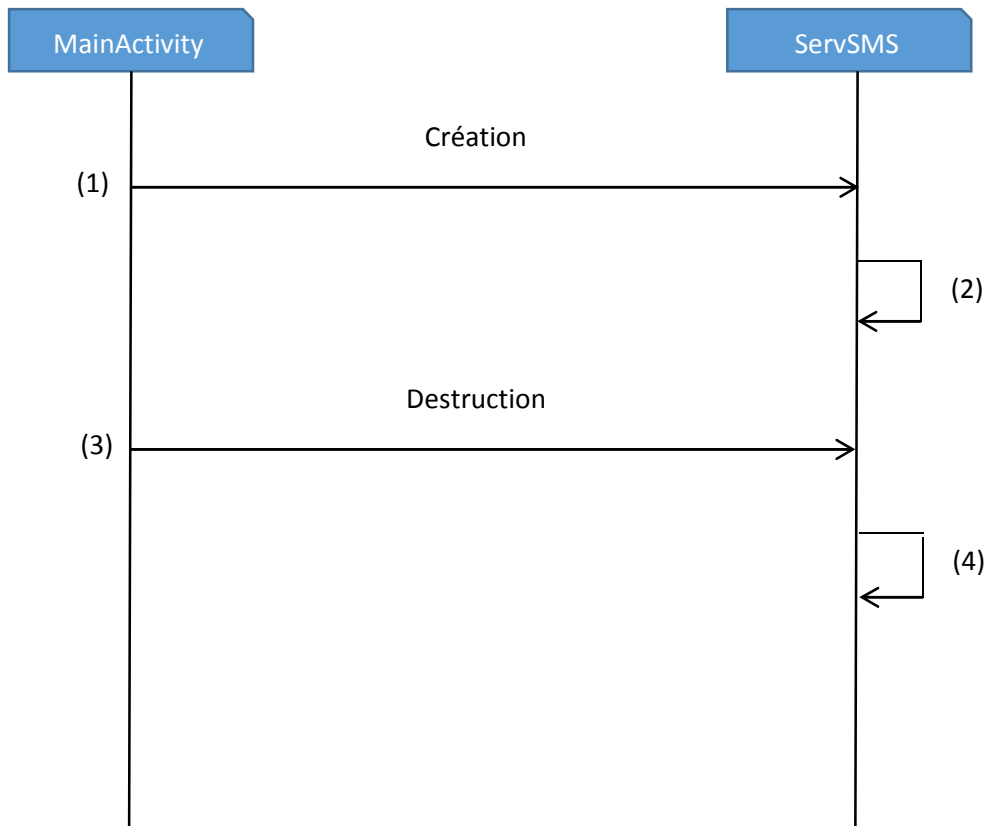
III.4.1.3 Diagramme de séquence pour activer ou désactiver le service :

Figure III.5 : Diagramme de séquence pour le cas « activer ou désactiver le service ».

Description :

- (1) : création du service avec la méthode « StartService (intent) ».
- (2) : le service est enregistré dans liste des receveurs de diffusion d'Android, et il est en attente de l'arrivée de nouveaux messages.
- (3) : la destruction du service se fait avec la méthode « stopService () ».
- (4) : le service arrête ce qu'il fait et il fait appel à la méthode « onDestroy », puis il est détruit.

III.4.1.4 Diagramme de séquence pour afficher la liste des emails :

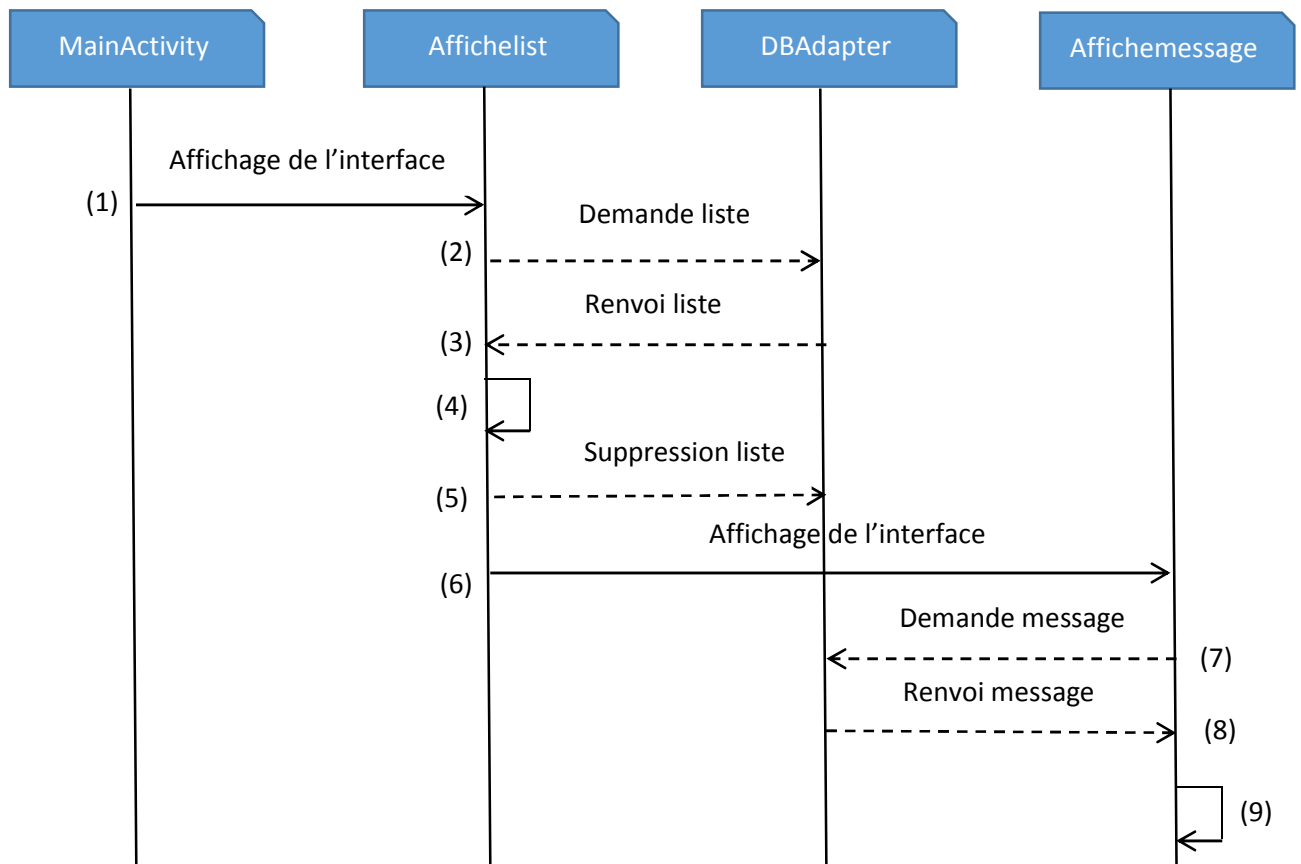


Figure III.6 : Diagramme de séquence pour le cas « afficher la liste des emails ».

Description :

- (1) : lors du clique sur le bouton « Afficher Liste », l'activity « MainActivity » fait appel à l'activity « Affichelist » à l'aide de la méthode « startActivity(intent) ».
- (2) : après la création de l'activity « Affichelist », elle demande à la classe « DbAdapter » qui gère la base de données de lui envoyer la liste des messages qui se trouve dans la base de données avec la méthode « recupererLaListeDesMessage () ».
- (3) : la méthode recupererLaListeDesMessage () renvoie la liste des messages enregistrés
- (4) : affichage de la liste des messages.
- (5) : l'activity « Afficheliste » peut faire appel à la classe « DBAdapter » pour supprimer tous les messages, avec la méthode « truncate () ».

- (6) : pour afficher un message on clique sur le message, puis l'activity « Affichelist » fait appel à l'activity « Affichemessage » en spécifiant la position du message dans l'intent.
- (7) : l'activity demande à la classe DBAdapter de lui renvoyer un seul message en spécifiant sa position, avec la méthode « recupereEmail (position) ».
- (8) : la classe « DBAdapter » renvoie le message.
- (9) : l'activity affiche le contenu du message.

III.4.2 Diagrammes de classes

➤ Partie serveur :

La figure III.3 présente une vue de la partie serveur qui est constituée de plusieurs classes interagissant entre elles afin d'accomplir des tâches bien précises. Dans ce qui suit nous allons définir le rôle de chacune d'entre elles.

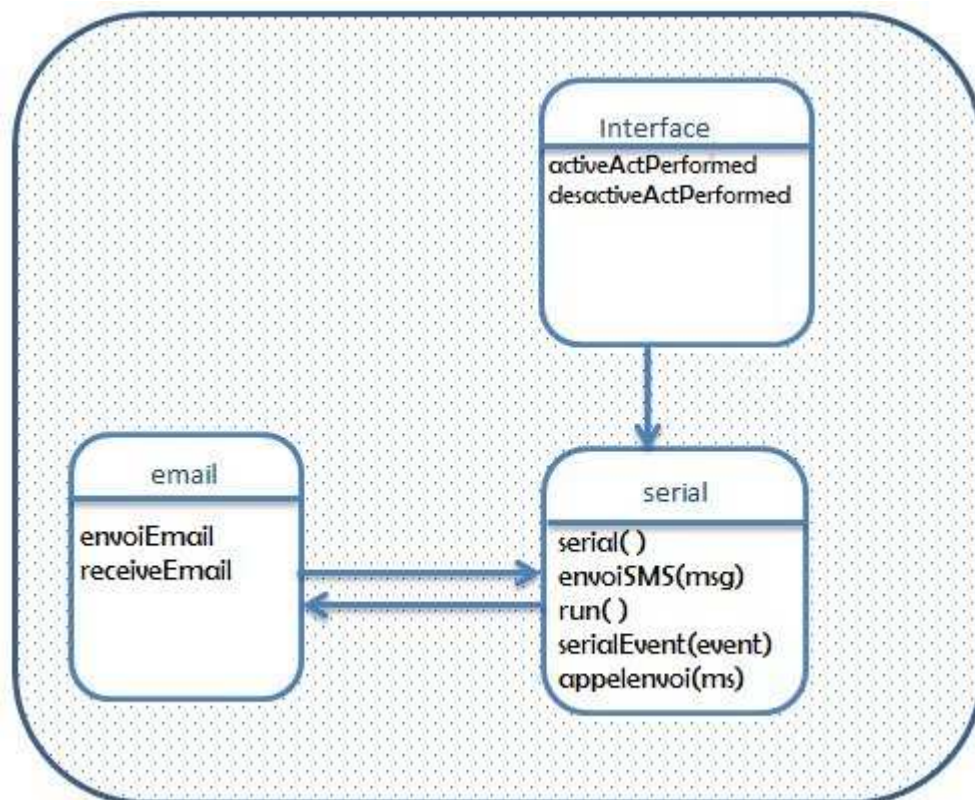


FIGURE III.7: Diagramme de classe de la partie Serveur.

Comme la figure ci-dessus le montre notre serveur comporte 3 classes :

La classe « interface » hérite de « javax.swing.JApplet », on l'utilise pour l'interface de notre serveur, il y a aussi les méthodes « actionPerformed » et

« deactivateActionPerformed », qu'on utilise pour activer ou détruire le thread de la classe « serial »

La classe « serial » hérite de « Runnable » pour utiliser les threads, elle hérite aussi de « SerialPortEventListener » pour écouter le port série. Dans cette classe on a implémenté la méthode « serial () », qui ouvre et initialise le port série. La méthode « envoiSMS(msg) » est implémentée pour l'envoi des commandes AT qui sont nécessaires pour l'envoi de SMS à la clé GPRS via le port série. l'annexe II donne un aperçu des commandes AT nécessaire à la gestion de la clé GPRS. La méthode « run() » appelle la méthode « receiveEmail() » de la class « email » chaque 1 minute. La méthode « serialEvent(event) » récupère l'état du port série et le teste, dès qu'il y a des données reçu sur le port transmises par la clé (DATA_AVAILABLE), cette méthode fait appel à la méthode « appelenvoi(ms) » en lui donnant les donnée reçu. La méthode « appelenvoi(ms) » teste si les données contiennent un "#", si c'est le cas elle fait appel à la méthode « envoimail(ms) » de la classe « email ».

La classe « email » qui implémente deux méthodes essentielles pour la gestion des emails, la première méthode étant « receiveEmail() » permet de récupérer les nouveaux mails reçus (non lus), et cela en se connectant au serveur mail, et en ouvrant le dossier INBOX, par la suite cette dernière récupère seulement les emails non lus, l'adresse de l'émetteur, l'objet et le message sont concaténés et séparés par le caractère « ? », ensuite le caractère spécial « # » est ajouté au début, à la fin la méthode « receiveEmail() » renvoie le résultat à la méthode « envoiSMS(msg) » de la classe « serial » pour l'envoyer par SMS. La deuxième méthode est « envoiEmail(ms) » qui sert à envoyer un email. Son rôle et de récupérer le SMS reçu par la clé, ensuite l'adresse, l'objet et le message sont extrait, elle construit l'email à partir de ces données et l'envoi au serveur mail.

➤ **Partie client :**

La figure III.4 présente les différentes interactions entre les classes qui composent la partie client (l'application Android) avec une description du rôle de chacune d'entre elles.

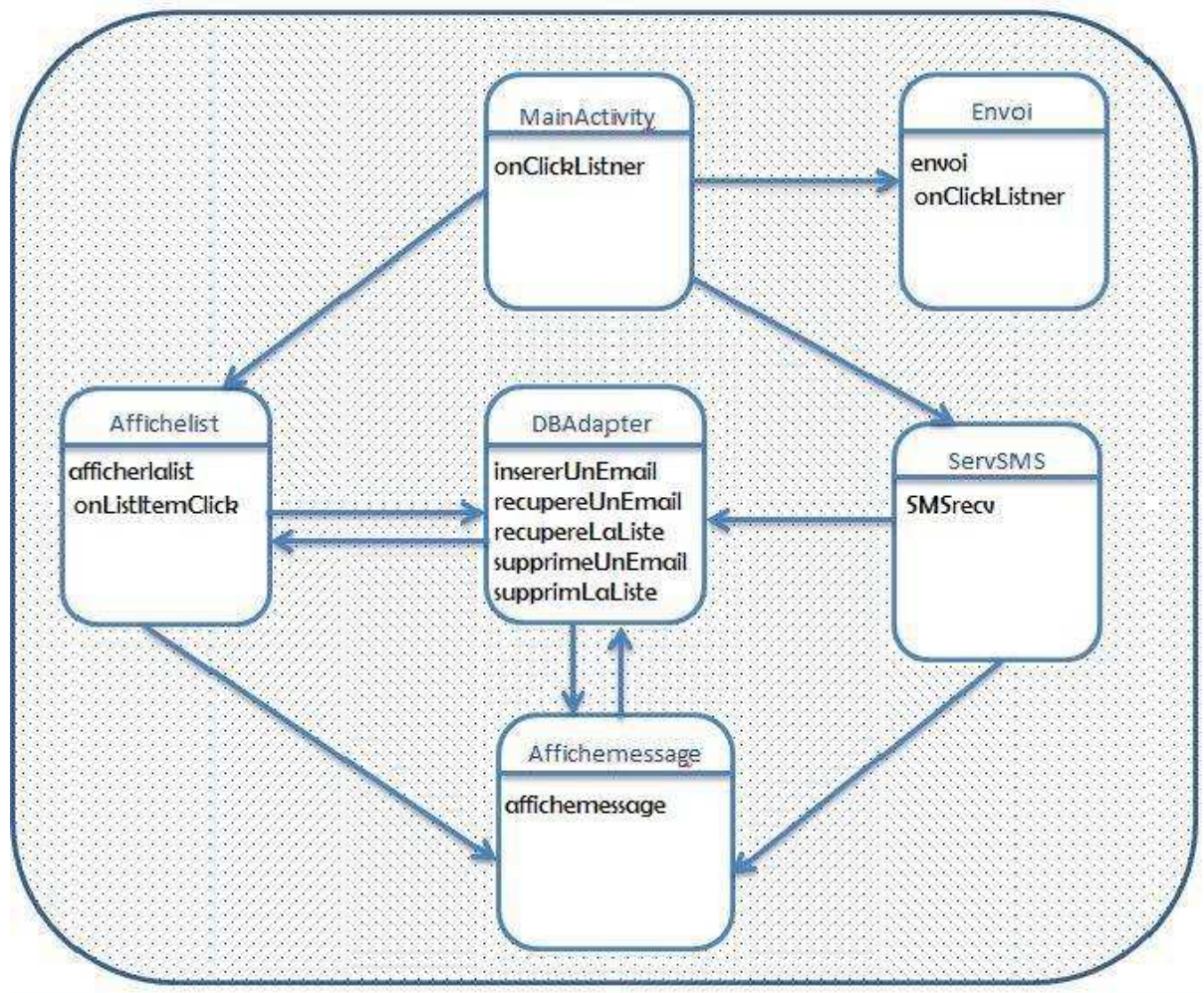


Figure III.8: Diagramme de classe de la partie client.

Comme la figure ci-dessus le montre, notre application Android est composée de six classes :

Les classes « MainActivity », « envoi », « Affichelist », « Affichemessage » sont des Activity, ces dernières implémentent les méthodes nécessaires pour les différents états de cycle de vie d'une activity. La classe « ServSMS » est un service Android.

La classe « DBAdapter » : s'occupe de gérer la base de données, cette dernière implémente plusieurs méthodes, leurs noms illustrent parfaitement leurs fonctionnalités.

La classe « MainActivity » : implémente la méthode « onClickListener (view) », qui lui permet d'écouter l'état des boutons, et de faire appel à l'activity liée au bouton.

La classe « envoi » : implémente la méthode onClickListner (view), qui fait appel à la méthode « envoi () », qui permet de récupérer le contenu des champs, puis les concaténer en

ajoutant un “ # “ au début, ensuite les champs sont séparés avec « ? », enfin le SMS est envoyé au numéro spécifier.

La classe « ServSMS » hérite de la classe « Service » qui lui permet de s’exécuter en arrière-plan. la classe « ServSMS » implémente la méthode « RecvSMS » qui hérite de le classe « BroadCastReceiver ». la méthode « RecvSMS » permet de récupérer le contenu du SMS reçu ,et de vérifier si il commence par le caractère spécial « # » ,si c’est le cas, les données (adresse,obj,message) sont extraites, ensuite la méthode « insererUnEmail(adr,obj,msg) » de la classe DBAdapter est appelée. Par la suite une notification est affichée à l’écran permettant à l’utilisateur de visualiser l’email grâce à l’appel à l’activity « Affichmessage ».

La class « Afficheliste » implémente la méthode « affichelist », qui fait appel à la méthode « recupereLaListe » de la class « DBAdapter ».la méthode « affichelist » permet d’afficher la liste des emails renvoyés par la méthode « recupereLaListe ».la méthode « onListItemClick » fait appel à l’activity « Affichmessage »en lui spécifiant la position du message.

La class « Affichmessage » implémente la méthode « affichemessage », qui fait appel à la méthode « recupereUnEmail » de la class « DBAdapter », et affiche le message renvoyé par cette dernière.

III.4.3 Quelques algorithmes:

III.4.3.1 Algorithme d’extraction des données:

Cet algorithme permet d’extraire les différents champs que contient un email, qui sont de la forme suivante (#adresse@adresse.dz?objet?message)

```
Boolean adb=true,obb=false,msb=false;
ms = messageBody;
for(int i =1; i<ms.length();i++)
{
if((ms.charAt(i) != '?') && (adb == true) )
    {
adrr += ms.charAt(i);
    }
else
if ((adb == true)&&(obb == false))
```

```

    {
adb = false; obb = true;
    }
Else
If ((obb == true) && (ms.charAt(i) != '?'))
    {
obj += ms.charAt(i);
    }
else
if ((obb == true)&&(msb == false))
    {
obb = false; msb = true;
    }
else
if (msb == true)
    {
msg += ms.charAt(i);
    }
}

```

III.4.3.2 Algorithme de tri des messages:

Cet algorithme permet de trier les messages sur la partie serveur et de prendre en compte que ceux qui commencent par le caractère spécial “#”

```

appelenvoi(String sms)
{
int i=0; String smsss =""; Boolean notend = true;
while((sms.charAt(i)!= '#')&&(notend==true) )
{
i++;
if(sms.length()==i)
{
notend=false;
}}
if(sms.charAt(i)=='#')

```

```
{  
//la ou commence le contenu du sms  
while(i != sms.length())  
{  
smsss+=sms.charAt(i); i++;  
}  
email.envoiEmail(smsss);  
}  
}
```

Conclusion :

Nous avons consacré ce chapitre à l'étude, l'analyse et la conception de notre travail. Nous avons proposé une démarche de modélisation pour développer notre application.

Dans le chapitre suivant nous aborderons la réalisation et la mise en œuvre de notre l'application.

Chapitre IV : Réalisation et mise en œuvre.

Introduction :

Dans ce dernier chapitre nous nous intéressons aux outils de réalisation de notre projet, où nous avons utilisé Eclipse comme principal IDE, avec le SDK et l'ADT pour la partie Android, et avec quelques bibliothèques (RXTX, JavaMail) pour la partie serveur. Nous avons pour but de réaliser une application qui propose une passerelle SMS/Mail.

IV.1 Environnement du travail :**IV.1.1 Environnement matériel :**

Pour notre projet, nous avons utilisé le matériel suivant pour le développement de notre application :

- Un ordinateur portable Intel(R) Core (TM) 2 Duo de 4GHZ
- Une clé GPRS ooredoo
- Un smartphone Condor avec les caractéristiques suivantes :
OS : Android 4.4.2 kitkat.
Quad Core 1.2 GHZ.

IV.1.2 Environnement cible :

L'application est destinée pour tous les smartphones équipés d'un système Android 4.2.2, toutefois elle est utilisable sur les versions supérieures d'Android.

IV.1.3 Environnement logiciel :

Pour pouvoir réaliser nos applications dans de bonnes conditions, il est nécessaire de préparer l'environnement de travail. Pour le préparer, nous avons effectué différentes tâches que nous décrivons dans ce qui suit :

- Installation d'Eclipse (comme pour le développement d'application Java classique).
- Le Plugin Android pour Eclipse ADT (Android Developer Tools), qui adapte Eclipse au développement d'application sous Android.
- Installation SDK (Software Development Kit) qui va contenir tous les outils nous permettant la construction de notre application.
- Intégration de la bibliothèque RXTX pour la gestion du port série.
- Intégration de la bibliothèque JavaMail nous permettant la communication entre le serveur mail et l'ordinateur.

IV.1.3.1 IDE Eclipse :

Eclipse est un IDE qui permet de programmer dans différents langages grâce ses nombreux plug-ins et notamment le plug-in d'Android. 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.

Pour le développement de notre application nous avons utilisé Eclipse LUNA, dont l'interface principale est illustrée dans la figure IV.1:

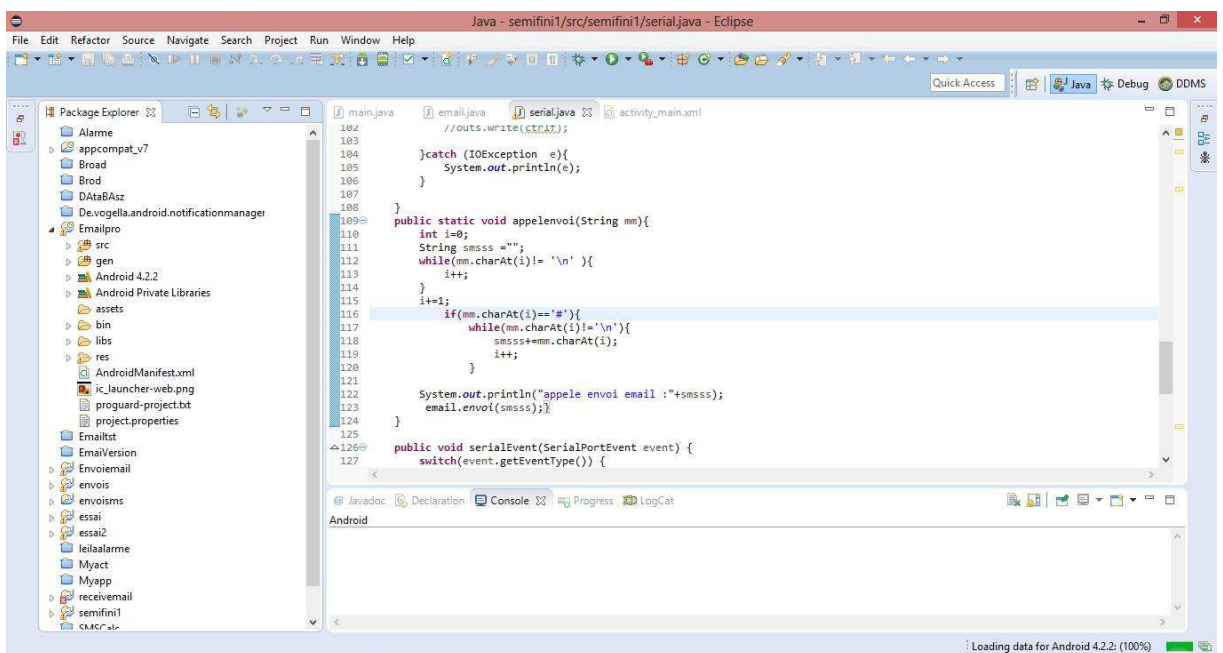


Figure IV.1: Interface principale d'éclipse.

IV.1.3.2 Les bibliothèques java :

Les bibliothèques logicielles est une collection de fonctions, qui peut être déjà compilée et prête à être utilisée par des programmes et mettent en œuvre une interface de programmation (API).

Dans ce qui vient nous allons citer quelques bibliothèques utilisés

- **La bibliothèque RXTX :** RXTX est une bibliothèque Java pour l'accès aux ports série. Cette API est open source et reprend celle définie par Sun javax.comm. RXTX comporte un ensemble de classes *gnu.io.** et des bibliothèques natives (.dll) précompilées pour un ensemble d'OS, de version et d'architecture.

- **La bibliothèque javaMail** : est l'API standard de gestion des courriers électroniques, elle utilise plusieurs protocoles tel que les protocoles POP3, SMTP, IMAP.

IV.1.3.3 LE SDK :

Le Kit de développement d'Android est un ensemble complet d'outils de développement, il inclut des bibliothèques logicielles, un débogueur, un émulateur, des outils et des exemples correspondant à la version concernée.

Le SDK Android compte plusieurs outils et utilitaires pour nous aider à créer, tester et déboguer nos projets. Dans ce qui suit nous allons présenter brièvement quelques uns de ces outils.

IV.1.3.4 Le SDK Manager :

Le SDK est un ensemble d'outils de développement qui permet de créer de nouvelles applications.

Son installation se fait de la manière suivante :

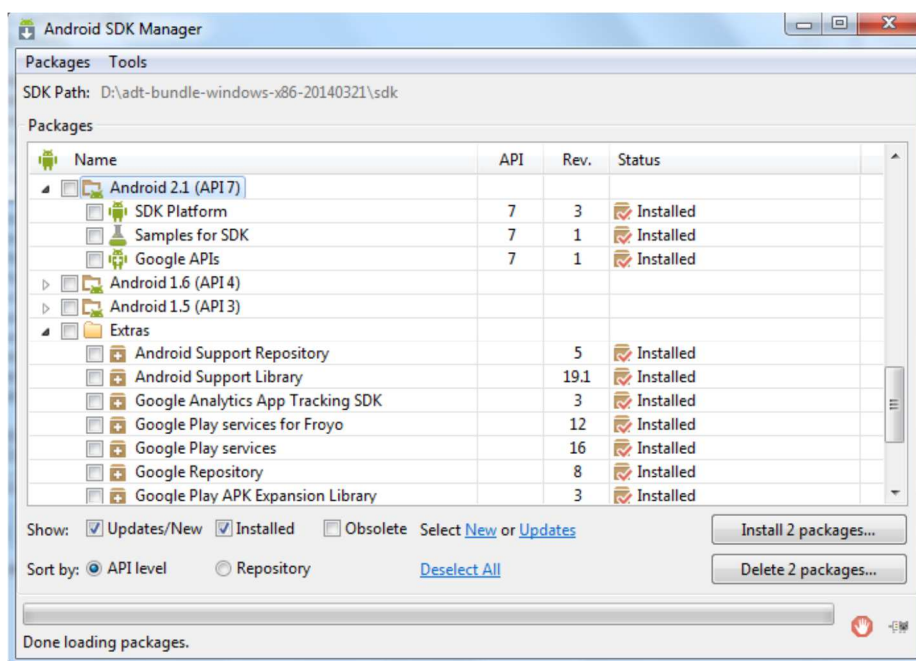


Figure IV.2 : Choix de la version de SDK Manager.

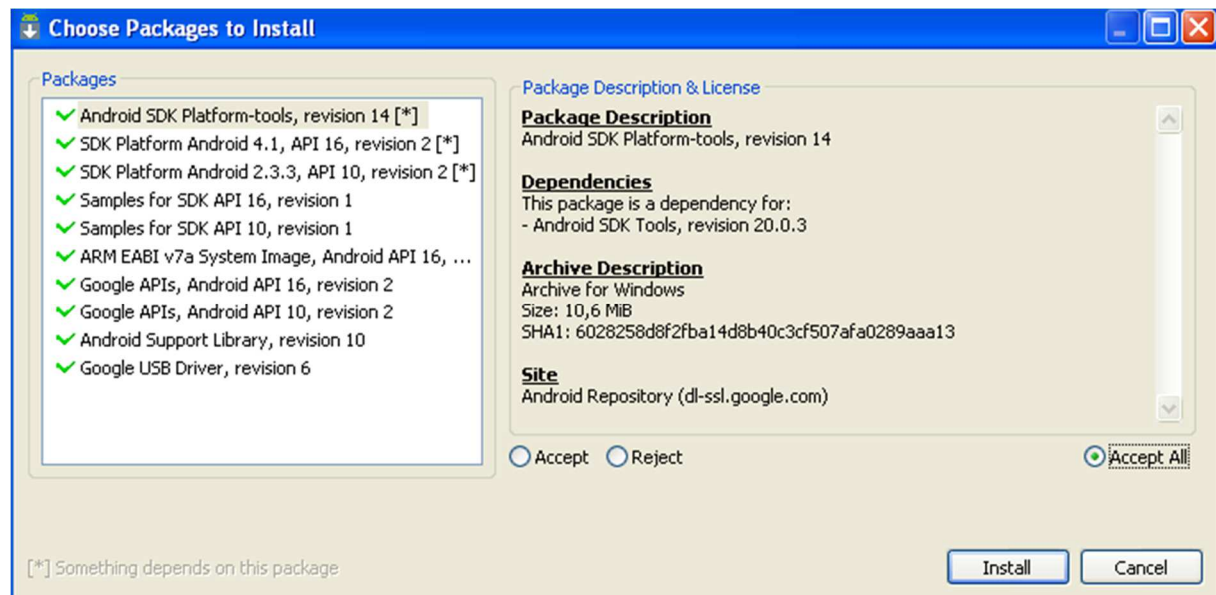


Figure IV.3 : Installation des packages.

IV.1.3.5 ADT pour Eclipse:

Le langage privilégié pour le développement d'applications Android est Java. Google a donc tout naturellement conçu un plugin pour Eclipse.

Android Développent Tools (ADT) est très complet et surtout très pratique il offre l'accès à de nombreuses fonctionnalités qui aident à développer des applications Android.

IV.1.3.6 Android Virtual Device Manager :

L'AVD Manager fournit une interface utilisateur graphique dans laquelle nous pouvons créer et gérer des périphériques virtuels Android (AVDS).

Cet outil permet de tester les applications sur plusieurs plateformes matérielles sans avoir besoin d'acheter les téléphones correspondants.

La figure IV.4 présente l'Android Device Manager

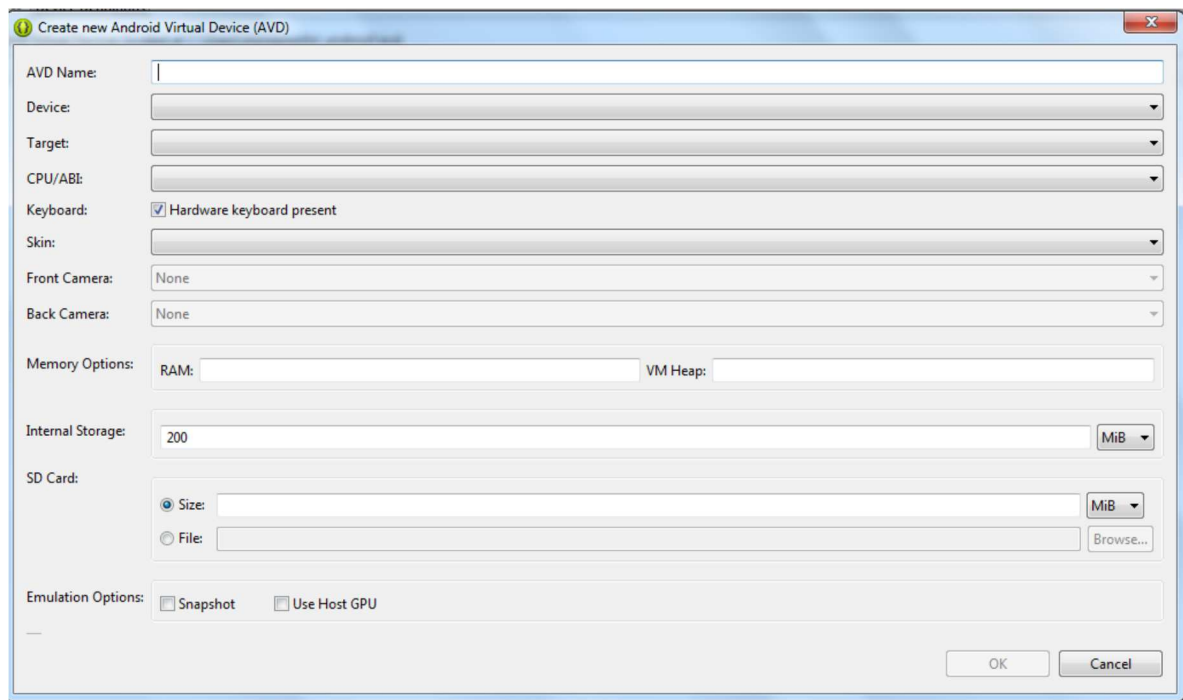


Figure IV.4 :Android Virtual Device Manager.

IV.1.3.7 Emulateur Android:

Le SDK Android comprend un émulateur d'appareil mobile.

L'émulateur Android est un dispositif qui fournit un appareil mobile virtuel s'exécutant sur un ordinateur et nous permettant de tester des applications Android.

L'émulateur Android imite toutes les fonctionnalités matérielles et logicielles d'un dispositif mobile typique tel que la lecture des fichiers audio et vidéo, stockage des données et nous pouvons également simuler l'envoi et la réception d'appels et de SMS.

La figure ci-dessous illustre un exemple d'émulateur.

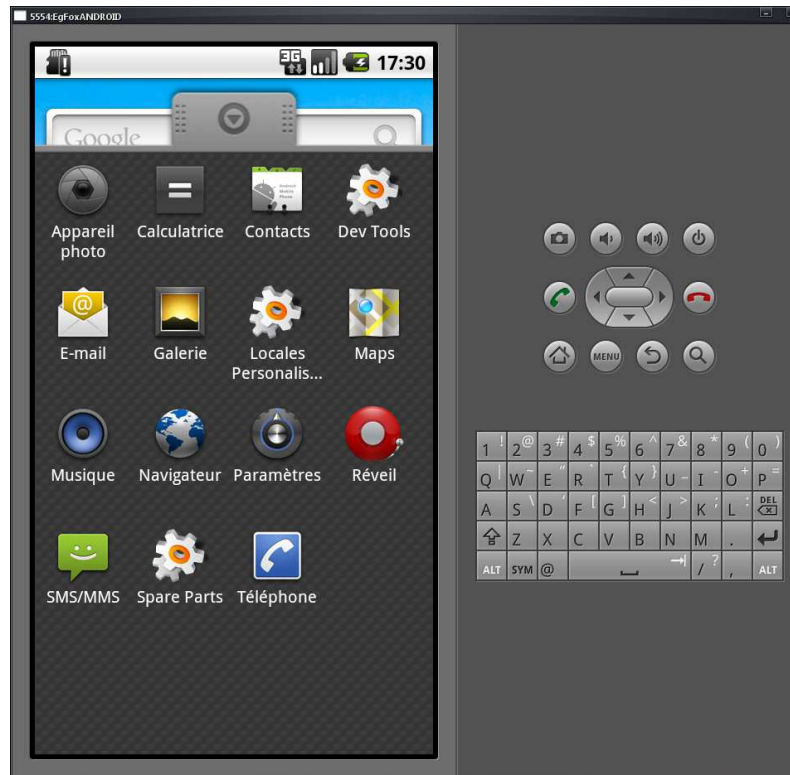
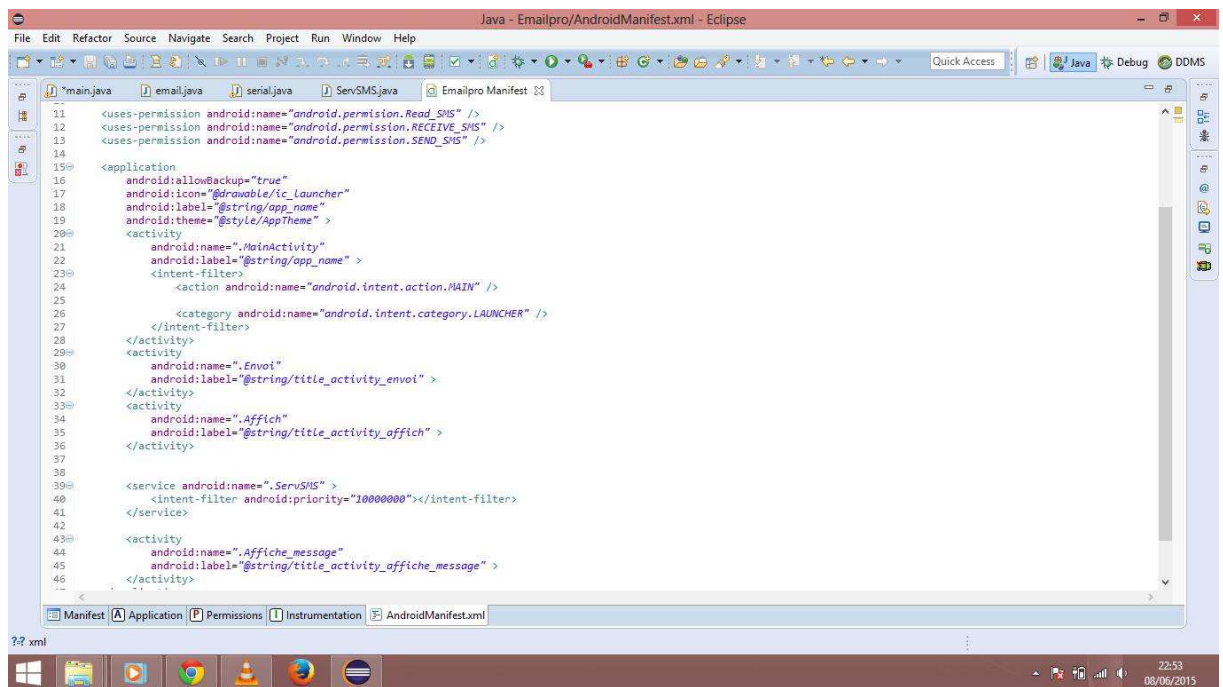


Figure IV.5 : Exemple d'émulateur.

IV.2 Fonctionnement de nos Applications:

IV.2.1 ApplicationAndroid pour le client :

L'application que nous avons développée pour le mobile Android comporte une Activity principale nommée **MainActivity** qui fait appel à des sous activity (Envoi, Affich, Affich_message) et d'un service nommé **ServSMS** que nous avons relié par les intents, pour ce faire une modification au niveau du Manifest est effectuée comme on peut le voir dans la figure ci-dessous :



```
11 <uses-permission android:name="android.permission.Read_SMS" />
12 <uses-permission android:name="android.permission.RECEIVE_SMS" />
13 <uses-permission android:name="android.permission.SEND_SMS" />
14
15
16 <application
17     android:allowBackup="true"
18     android:icon="@drawable/ic_launcher"
19     android:label="@string/app_name"
20     android:theme="@style/AppTheme" >
21     <activity
22         android:name=".MainActivity"
23         android:label="@string/app_name" >
24         <intent-filter>
25             <action android:name="android.intent.action.MAIN" />
26
27             <category android:name="android.intent.category.LAUNCHER" />
28         </intent-filter>
29     </activity>
30     <activity
31         android:name=".Envoi"
32         android:label="@string/title_activity_envoi" >
33     </activity>
34     <activity
35         android:name=".Affich"
36         android:label="@string/title_activity_affich" >
37     </activity>
38
39     <service android:name=".ServSMS" >
40         <intent-filter android:priority="10000000"></intent-filter>
41     </service>
42
43     <activity
44         android:name=".Affiche_message"
45         android:label="@string/title_activity_affiche_message" >
46     </activity>
47 </application>
```

Figure IV.6: Le fichier MANIFEST.

D'après la **figure IV.6**, on voit bien la présence de nos activités qui sont entre les deux balises `<Activity>` et `</Activity>`, comme on voit la présence du service qui est entre les deux balises `<Service>` et `</Service>`.

On voit aussi la présence de ces instructions. Ce sont les permissions qu'on a ajoutées.

- **android.permission.RECEIVE_SMS** : utilisé pour les broadcastreceiver.
- **android.permission.Read_SMS** : sert à lire des SMS.
- **android.permission.SEND_SMS** : sert à envoyer des SMS.

La figure IV.7 Représente une vue sur notre activité principale :

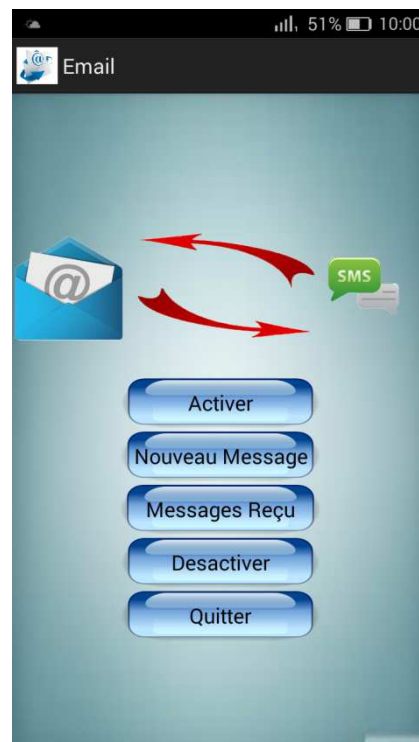


Figure IV.7 : Activités principale de l'application.

IV.2.2 Application pour le serveur :

Cette partie est dotée d'une interface permettant d'Activer ou de désactiver le serveur, si ce dernier est actif, il devient fonctionnel (il écoute l'arrivée de nouveaux messages). S'il est désactivé, le serveur est en mode pause.

Les traitements qui se font dans cette partie sont transparents pour l'utilisateur

IV.3 Les fonctions de notre application :

1. Envoi d'un Email

Le clic sur le bouton « Nouveau Message », nous renvoie vers une nouvelle fenêtre en appelant l'Activity « Envoi », nous permettant de remplir les champs adresse, objet et message et d'envoyer l'email.

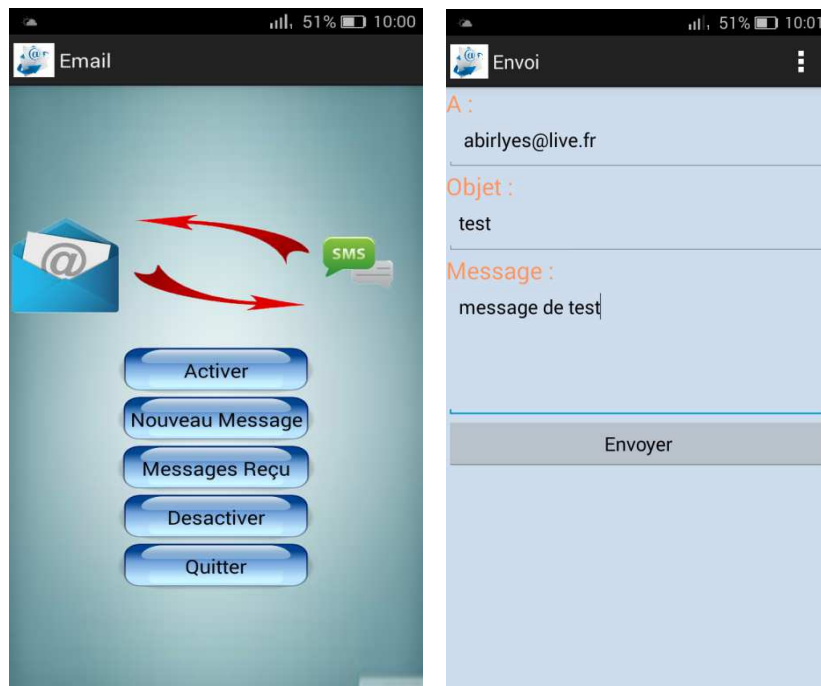


Figure IV.8: Envoyer un email.

Le clic sur le bouton « Envoyer » permet d’envoyer l’email, si le format du champ «adresse » n’est pas correcte, un message sera affiché à l’utilisateur comme le montre la figure suivante :



Figure IV.9 : Affichage du toast.

2. Activer ou désactiver le service passerelle :

- **activer le service** : se fait avec un simple clic sur le bouton « Activer » présent sur l'écran d'accueil, de ce fait le service sera d'abord créer puis démarrer.



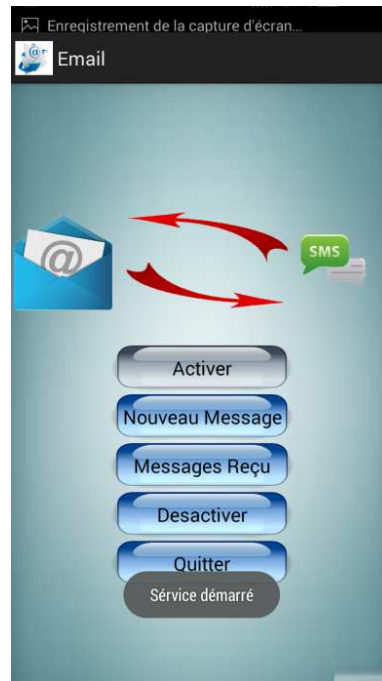


Figure IV.10: Activer le service.

- **désactiver le service** : le service doit être déjà activé dans ce cas, il suffit de cliquer sur le bouton « Desactiver ».



Figure IV.11 : Désactiver le service.

3. Réception d'un email :

A la réception une notification apparaît à l'utilisateur



Figure IV.12 : Notification.

Le clic sur la notification fait appel à l'activité « Affich_message », qui nous permettra de visualiser le contenu du message.

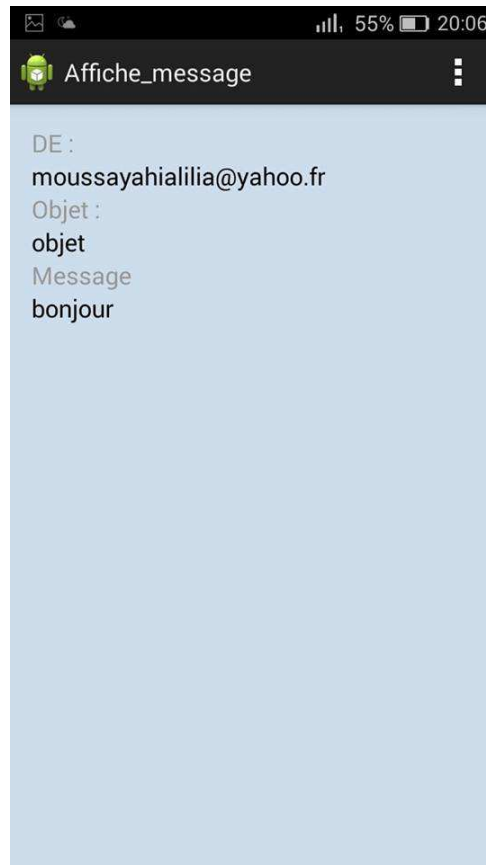


Figure IV.13 : Affichage du message.

4. Afficher la liste des messages reçus

L’affichage de la liste des messages reçus se fait avec un simple clic sur le bouton « Messages Reçus ». La figure IV.14 nous montre la liste des messages reçus.



Figure IV.14 : Affichage de la liste des messages.

5. Activer ou désactiver un service sur la partie serveur

Un simple clic sur le bouton « Activer » ou sur le bouton « Désactiver » permettra d'activer ou de désactiver le service.



Figure IV.15 : Activer le service.



Figure IV.16 : Désactiver le service.

Conclusion :

Dans ce dernier chapitre nous avons présenté une vue générale sur l'application que nous avons eu pour but de réaliser. Nous avons montré les différents rendus visuels que nous offrait l'application ainsi que ses nombreux composants et le rôle et les besoins de chacun.

CONCLUSION ET PERSPECTIVES

L'élaboration de notre projet de fin d'étude était dans le but de concevoir une passerelle SMS/Mail, pour ce faire nous avons choisi le modèle client/serveur. Notre application est composée de deux parties distinctes, la première partie dédiée aux terminaux mobiles Android, qui est la partie client, dispose d'une interface graphique permettant de faciliter l'interaction avec l'utilisateur. La deuxième partie est développée sur un ordinateur, elle consiste en un serveur doté d'une interface permettant de l'activer ou de le désactiver. Il s'occupe de la réception des SMS/Mail, de leur conversion en Mail/SMS respectivement ainsi que de leur envoi vers la destination (clé GPRS/ boîte mail).

Ainsi réalisée, notre application offre les fonctionnalités suivantes :

- Envoie d'un Email.
- La réception d'un Email.
- La consultation du message reçu.
- La visualisation de tous les messages reçus.

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. Ce projet nous a permis de nous intégrer dans le domaine de l'Android tout en nous offrant l'occasion d'acquérir de nouvelles connaissances et de maîtriser le langage de programmation java.

En guise de perspectives nous envisageons d'enrichir notre travail avec quelques fonctionnalités supplémentaires :

- Rendre l'application multiutilisateur dans le sens où plusieurs utilisateurs peuvent accéder à notre passerelle simultanément.
- Au-delà du simple texte, l'envoi du contenu multimédia (image, son,...etc) à travers la passerelle est l'un de nos buts.

Les activités sous Android :

Les applications Android sont constituées d'un ensemble d'activités qui permettent l'interaction avec l'utilisateur, et d'un ensemble de services qui fonctionnent en arrière-plan (masqués) et s'occupent d'effectuer certains traitements précis, comme par exemple la récupération des données GPS en temps réel.

Une activité est la composante principale pour une application Android, elle représente l'implémentation et les interactions des interfaces.

Une activité possède un cycle de vie bien précis, qui permet de contrôler aisément l'application de son lancement jusqu'à sa fermeture.

Le cycle de vie d'une activité :

Voici un diagramme qui explique le cycle de vie d'une activité sous Android

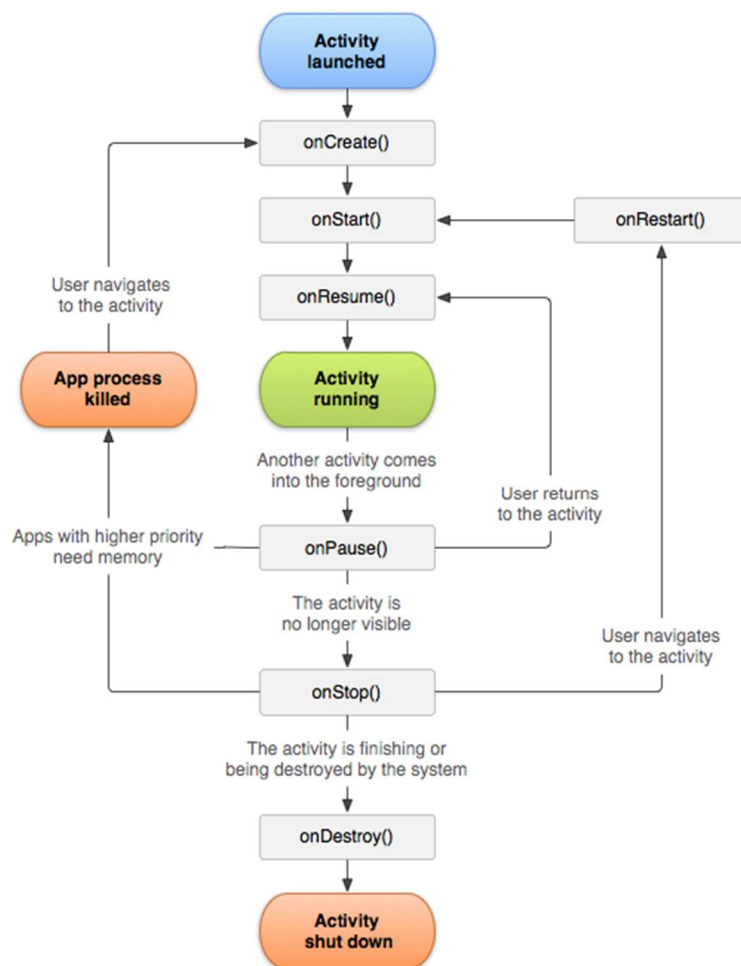


Figure II.6 : cycle de vie d'une activité

Annexe I

Les états principaux d'une activité sont les suivant :

- **onCreate()** : exécutée quand l'utilisateur clique sur l'icône de l'application pour une première fois. cette méthode permet d'initialiser l'activité.
- **onStart()** : lancée tout de suite après le onCreate() lorsque l'activité va devenir visible à l'utilisateur.
- **onResume()** : exécutée lorsque votre application est passée en avant plan.
- **onPause()** : est lancée lorsqu'une autre activité va s'afficher à l'avant plan ou avant chaque arrêt de l'activité.
- **onStop()** : est exécuté lorsque une activité n'est plus visible. Le processus va passer en sommeil. Le système pourra décider de mettre fin définitivement à l'activité si nécessaire.
- **onDestroy()** : met fin à au cycle de vie de l'activité. Exécutée lors de l'arrêt de l'activité et les ressources seront libérées.

Projet ADT :

Un projet basé sur le plugin Android est décomposé de la manière suivante :

- **src/** : les sources Java du projet.
- **libs/** : Bibliothèques tierces.
- **res/** :
 - ✓ **res/drawable**: ressources images
 - ✓ **res/layout** : description des IHMs en XML.
 - ✓ **res/values** : chaînes de caractères et dimensions.
- **gen/** : les ressources auto générées par ADT.
- **assets/** : ressources brutes (raw bytes).
- **bin/** :
 - ✓ **bin/classes**: les classes compilées en **.class**.
 - ✓ **bin/classes.dex** : exécutable pour la **JVM DALVIK**.
 - ✓ **bin/myapp.zip** : les ressources de l'application.
 - ✓ **bin/myapp.apk** : application empaquetée avec ses ressources et prête pour le déploiement.

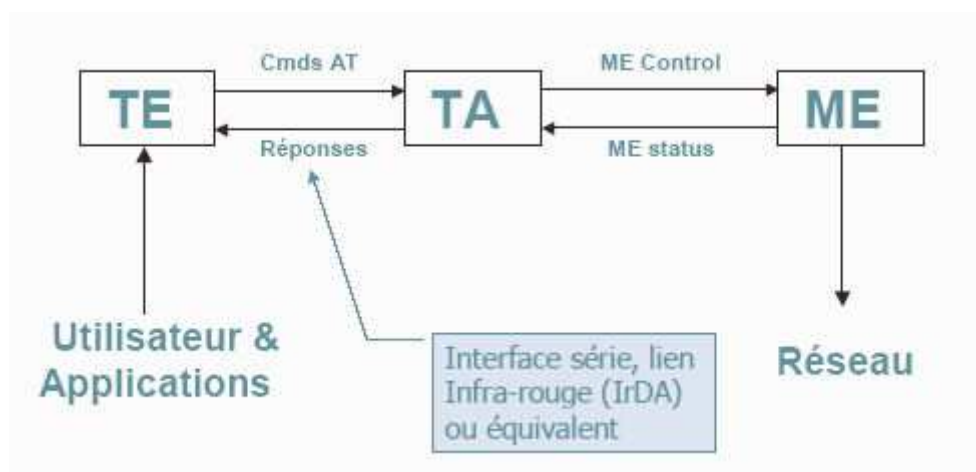
Les commandes AT :

Les commandes AT sont définies dans la norme GSM 07.05 pour les SMS (07.07 pour les appels téléphoniques). AT est l'abréviation de ATtention. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du mobile.

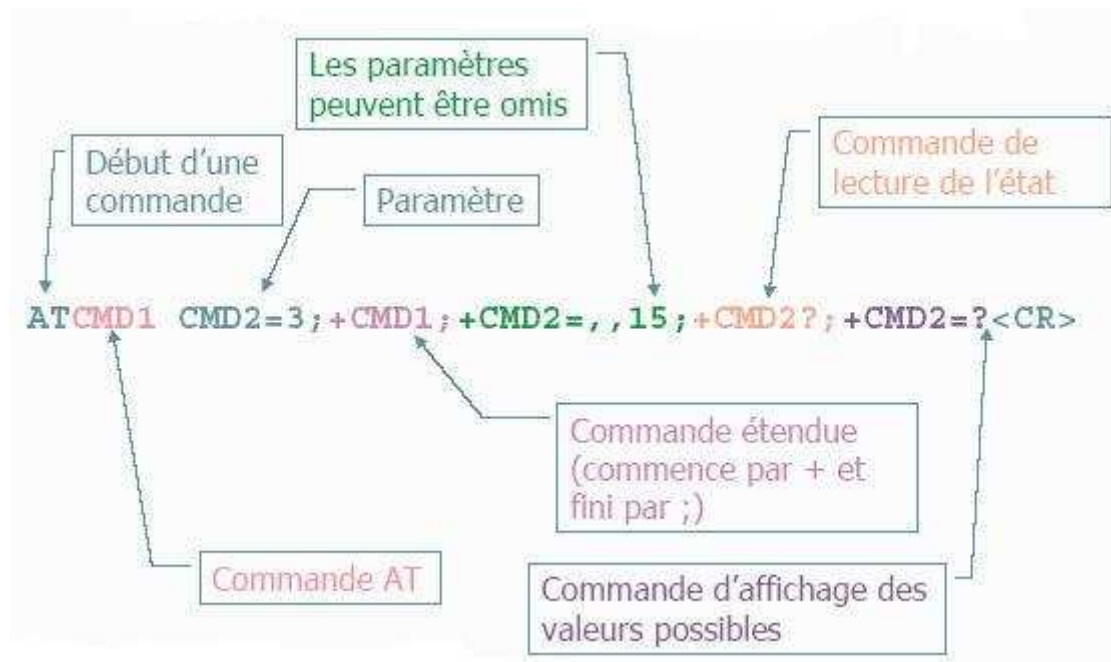
Trois entités sont définies :

- TE : Terminal Equipment (envoi et affiche les commandes).
- TA : Terminal Adaptator (interface entre l'utilisateur et le mobile).
- ME : Mobile Equipment.

Schéma de fonctionnement :



Structure d'une commande AT :



Les commandes AT permettent :

- de composer un numéro de téléphone.
- de commander le raccordement du modem à la ligne (l'équivalent de décrocher le téléphone).
- de connaître l'état de la ligne : tonalité d'invitation à transmettre, ligne occupée...etc.
- de spécifier le type de transmission et le protocole de liaison à utiliser.
- de régler le volume sonore du haut-parleur interne du modem.
- d'envoyer les caractères transmis simultanément vers l'écran.
- d'afficher certains renseignements concernant le modem.
- de manipuler les registres internes du modem.

Quelques exemples de commandes AT :

Commande AT	fonction
AT+CSCS="GSM"	Sélectionne l'ensemble des caractères de l'équipement mobile
AT+CMGF=1	commande définit le modem GSM en mode SMS du texte ou en mode de PDU SMS
AT+CMGS="+21354697564"	envoie un message SMS à un téléphone GSM

Exemples de codes résultat :

Forme étendue	Signification
OK	Confirmation d'exécution d'une commande
CONNECT	La connexion est établie à 300 bit/s
RING	Détection de la tonalité de sonnerie
NO CARRIER	Problème de connexion
ERROR	Erreur de syntaxe dans la ligne de commandes
CONNECT1200	La connexion est établie à 1 200 bit/s
BUSY	Ligne occupée