

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master en Informatique

OPTION : Réseaux, Mobilité et Systèmes Embarqués

Thème

Application Android pour Ventes et Achats en ligne

Réaliser par le binôme:

**HAMMAD Saïd.
HAMZA Younes.**

Encadré par :

Mme. Malika Belkadi.

**Soutenu le: 28/09/2016
Devant le jury composé de:**

**M^{me}. Rachida AOUDJIT
M. Mehammed DAOUI
M^{me}. Rebiha HADAoui**



Promotion 2015/2016

Remerciements

En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apportées leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à remercier sincèrement M^{me} M. Belkadi, qui, en tant que Directeur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour. J'exprime ma gratitude à tous les consultants et internautes rencontrés lors des recherches effectuées et qui ont accepté de répondre à mes questions avec gentillesse. Je n'oublie pas mes parents pour leur contribution, leur soutien et leur patience Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenus et encourager au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicace:

A mes chers parents et mes grand-mères

A mon frère : Arezki

A mes sœurs : Siham et Amina

A la mémoire de mes grands-pères

Aux personnes très chères pour moi : Mouhamed, Hamid,

Smail, Boualem, Ghania, Zahia et Zahra

A mes Enseignants: M M·Bouzina, M A·takharboucht,

M M·Raab et M^{me} F·Djaout·

A tous mes ami(e)s Samira Oslm, Salim, Younes,

Mohamed

A mon binôme Younes qui était très patient avec moi

tout au long de cette période du travail·

Je dédie ce modeste travail

Merci...

H·Saïd

Dédicace:

A mes très chers parents

Pour tout l'amour dont vous m'avez entouré,

Pour tout ce que vous avez fait pour moi. Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices...

Je ferai mon mieux pour rester un sujet de fierté à vos yeux avec l'esprit de ne jamais vous décevoir...

A mes trois frères Sadi, Amine, Mohamed.

Vous occupez une place particulière dans mon cœur, je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès. Je vous dirai tout simplement merci, je vous aime tant.

Que le bon Dieu veille sur vous...

A mes oncles et tantes et cousins

A mes très chers ami(e)s

En témoignage de l'amitié sincère qui nous a liées et des beaux moments passés ensemble je vous dédie ce travail en vous souhaitant tout le bonheur du monde...

A mon binôme Said

En souvenir de nos éclats de rire, des bons moments et des nuits blanches.

J'espère de tout mon cœur que notre amitié durera éternellement et que le succès, le bonheur et la bonne étoile vous accompagnent durant toute la vie...

Merci...

H.Younes

Sommaire :

LISTE DES ABREVIATIONS	XI
LISTE DES FIGURES	XII
LISTE DES TABLEAUX	XIV
INTRODUCTION GENERALE	1

Chapitre I : ANDROID

I.1 : INTRODUCTION	3
I.2 : DEFINITION D'ANDROID	3
I.3 : OHA (l'Open Handset Alliance)	4
I.4 : HISTOIRE D'ANDROID	4
I.5 : BUGDROID	5
I.6 : LES DIFFERENTES VERSIONS D'ANDROID	6
I.7 : REPARTITION DES VERSIONS	7
I.8 : L'ARCHITECTURE GENERALE D'ANDROID	8
I.8.1:LES APPLICATIONS.....	8
I.8.2:UN FRAMEWORK APPLICATIF	8
I.8.3:BIBLIOTHEQUE	9
I.8.4:MOTEUR D'EXECUTION D'ANDROID (Android Runtime)	9
I.8.5:LINUX KERNEL	9
I.9 : LES COMPOSANTS D'UNE APPLICATION	10
I.10 : PRISE EN MAN DE L'ENVIRONNEMENT	11
I.10.1:LE SDK ANDROID	11
I.10.2:ADT POUR ECLIPSE	11
I.10.3:EMULATEUR	12
I.11 : CONCEPTS ET INNOVATION.....	12
I.11.1:BUREAUX VIRTUELS	12
I.11.2:WIDGETS	13
I.11.3:CENTRALISATION DES NOTIFICATIONS	13
I.11.4:BOUSSOLE ELECTRONIQUE	14
I.12 : CONCLUSION	14

Chapitre II : GENERALITES SUR LES RESEAUX & LE MODELE CLIENT / SERVEUR

II.1 : INTRODUCTION	15
II.2 : DEFINITION.....	15
II.3 : AVANTAGES DES RESEAUX.....	15
II.4 : CLASSIFICATION DES RESEAUX	16
II.4.1 : SELON LA TAILLE	16
II.4.1.1 : UN RESEAU PERSONNEL (PAN)	16
II.4.1.2 : UN RESEAU LOCAL (LAN)	16
II.4.1.3 : UN RESEAU METROPOLITAIN (MAN).....	16
II.4.1.4 : UN RESEAU ETENDU(WAN)	16
II.4.2 : SELON LA TOPOLOGIE.....	17
II.4.2.1 : EN ANNEAU	17
II.4.2.2 : EN BUS	17
II.4.2.3 : EN ETOILE	18
II.4.2.4 : EN MAILLE	18
II.4.3 : SELON LE MODE DE DIFFUSION.....	19
II.4.3.1 : LE MODE DE DIFFUSION.....	19
II.4.3.2 : LE MODE POINT A POINT	19
II.4.4 : SELON LE MODE DE COMMUNTAION	20
II.4.4.1 : LA COMMUTATION DE CIRCUITS	20
II.4.4.2 : LA COMMUTATION DE MESSAGES	20
II.4.4.3 : LA COMMUTAION DE PAQUETS.....	20
II.5 : LE MODELE OSI	21
II.5.1 : DEFINITION	21
II.5.2 : LES COUCHES DU MODELE OSI.....	21
II.6 : LE MODELE TCP/IP	22
II.6.1 : DEFINITION	22
II.6.2 : LES COUCHES DU MODELE TCP/IP.....	22
II.7 : LA PILE DE PROTOCOLES	23
II.7.1 : PROTOCOLES D'APPLICATION	23
II.7.2 : PROTOCOLES DE TRANSPORT.....	23
II.7.3 : PROTOCOLES INTERNET	23
II.7.4 : PROTOCOLES D'ACCES AU RESEAU	23
II.8 : L'INTERNET, INTRANET ET EXTRANET :	24
II.8.1 : INTERNET	24
II.8.2 : EXTRANET	24
II.8.3 : INTRANET	24

II.9 : LE MODELE CLIENT / SERVEUR	25
II.9.1 : DEFINITION	25
II.9.2 : LES TYPES DE SERVEURS.....	26
II.9.2.1 : SERVEUR D'APPLICATION WEB	26
II.9.2.2 : SERVEUR DE FICHIERS	26
II.9.2.3 : SERVEUR DE BASE DE DONNEES	26
II.9.2.4 : SERVEUR DE TRA NSACTIONS	26
II.9.2.5 : SERVEUR DE GROUPWARE.....	26
II.10 : ARCHITACTURE CLIENT / SERVEUR.....	27
II.10.1 : DEFINITION.....	27
II.10.2 : LES DIFFERENTS TYPES D'ARCHITECTURE CLIENT / SERVEUR.....	27
II.10.2.1 : ARCHITECTURE 2-TIERS	27
II.10.2.1.1 : AVANTAGES ET INCONVENIENTS D'UNE ARCHITECTURE A 2-TIERS	28
II.10.2.2 : ARCHITECTURE 3-TIERS	28
II.10.2.2.1 : AVANTAGES ET INCONVENIENTS D'UNE ARCHITECTURE A 3-TIERS	28
II.10.2.3 : ARCHITECTURE N-TIERS.....	29
II.11 : WAMPSERVER	29
II.11.1 : DEFINITION.....	29
II.11.2 : LE SERVEUR APACHE.....	29
II.11.3 : LE SERVEUR MYSQL.....	30
II.11.3.1 : LE LANGAGE SQL.....	30
II.11.3 : L'OUTIL PHPMYADMIN	31
II.12 : LE LANGAGE PHP	32
II.13 : LA CONNECTIVITE D'UNE APPLICATION ANDROID AU SERVEUR	32
II.13.1:FORMAT JSON	32
II.13.2: LA SYNTAXE	33
II.13.3:AVANTAGES ET INCONVENIENTS	33
II.13.4: UTILISATION DE FORMAT JSON.....	35
II.13.4.1 : COTE CLIENT	35
II.13.4.2 : COTE SERVEUR.....	35
II.13.4.3 : L'ECHANGE DE DONNEES	35
II.14 : CONCLUSION	36

Chapitre III : ANALYSE ET CONCEPTION

III.1 : INTRODUCTION	37
III.2 : PRESENTATION GENERALE D'UML	37
III.2.1 : DEFINITION.....	37
III.2.2 : DIFFERENTES VUES ET DIAGRAMMES D'UML	37
III.2.2.1 : LES VUES STATIQUES OU STRUCTURELS (Structural Diagrams)	38
III.2.2.2 : LES VUES DYNAMIQUES OU COMPORTEMENTAUX.....	39
III.3 : LA MODELISATION AVEC UML	40
III.3.1 : ANALYSE	40
III.3.2 : IDENTIFICATION DES BESOINS	41
III.3.3 : IDENTIFICATION DES ACTEURS	41
III.4 : SPECIFICATION DES TACHES	42
III.5 : SPECIFICATION DES SCENARIOS	43
III.6 : LES CAS D'UTILISATION	45
III.7 : LA CONCEPTION	51
III.8 : LE NIVEAU APPLICATIF	51
III.8.1 : LES CAS D'UTILISATION.....	51
III.8.2 : LES ELEMENTS D'UN DIAGRAMME DES CAS D'UTILISATION	51
III.8.2.1 : LES ACTEURS	51
III.8.2.2 : LES CAS D'UTILISATION	51
III.8.3 : RELATION ENTRE ACTEURS ET CAS D'UTILISATION	52
III.8.3.1 : LA RELATION D'ASSOCIATION.....	52
III.8.4 : LES RELATIONS ENTRE CAS D'UTILISATION	52
III.8.4.1 : RELATION D'INCLUSION	52
III.8.4.2 : RELATION D'EXTENSION	52
III.9 : DIAGRAMME DES SEQUENCES	53
III.10 : DIAGRAMME D'ACTIVITÉS	58
III.10.1 : REPRESENTATION	58
III.11 : LES DIAGRAMMES DES CLASSES	67
III.11.1 : DEFINITION	67
III.12 : LE NIVEAU DONNEES	69
III.13 : CONCLUSION	70

Chapitre IV : REALISATION

IV.1 : INTRODUCTION	71
IV.2 : CHOIX DE L'ARCHITECTURE DE L'APPLICATION	71
IV.3 : L'ENVIRONNEMENT	72
IV.3.1 : ENVIRONNEMENT MATERIEL.....	72
IV.3.2 : ENVIRONNEMENT LOGICIEL	72
IV.3.2.1: LE PLUGIN DE DEVELOPPEMENT D'ECLIPSE : ADT	72
IV.3.2.2: SDK ANDROID	73
IV.3.2.3: ECLIPSE PHP	73
IV.3.2.4: WAMPSEVER	74
IV.3.2.5: LE SERVEUR WEB 000WEBHOST	74
IV.4 : LANGAGE DE PROGRAMATION	75
IV.4.1 : JAVA	75
IV.4.2 : XML.....	75
IV.4.1 : PHP	75
IV.4.1 : MYSQL	75
IV.5 : PROTOCOLE ET FORMAT DE DONNEES	75
IV.5.1 : PROTOCOLE DE COMMUNICATION.....	75
IV.5.2 : FORMAT DE DONNEES UTILISEES.....	76
IV.6 : TESTER L'APPLICATION SUR L'EMULATEUR	77
IV.7 : TESTER L'APPLICATION SUR LE TELEPHONE.....	77
IV.8 : FONCTIONNEMENT DE NOTRE APPLICATION	78
IV.8.1 : La PAGE INDEX	78
IV.8.2 : LA PAGE D'ACCUEIL	79
IV.8.3 : LA PAGE DE CONNEXION	80
IV.8.4 : LA PAGE D'INSCRIPTION	81
IV.8.5 : LA PAGE MON COMPTE	82
IV.8.6 : LA PAGE DEPOSER UNE ANNONCE	83
IV.8.7 : LA PAGE RECHERCHE D'UNE ANNONCE	84
IV.8.8 : LA PAGE AIDE	85
IV.9 : CONCLUSION.....	86
CONCLUSION GENERALE	87
BIBLIOGRAPHIE & WEBOGRAPHIE.....	88

Liste des Abréviations :

ADT	Android Devloppent Tools
AIM	Association for automatic Identification and Mobility
API	Application Programming Interface
ARM	Architecture Reference Manual
CPU	Central Processing Unit
DDMS	Dalvik Debug Monitor Service
GPS	Global Positioning System
HTC	High-Throughput Computing
IDS	Insurance Data Services
IDE	Integrated Development Environment
IOS	iPhone OS
JSON	Java Script Object Notation
LG	Lucky-Goldstar
OS	Operating System
PC	Personal Computer
PDA	Personal Digital Assistant
PDF	Portable Document Format
RAM	Random Access Memory
SD	Secure Digital
SDK	Software Development Kit
SMS	Short Message Service
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language
WAMP	Windows Apache MySQL Php

Liste des Figures :

Chapitre I :

Figure I.1 : Le logo de l'Android	5
Figure I.2 : Les différents logos et noms des versions d'Android	6
Figure I.3 : Graphique illustrant la part de chaque version d'Android sur Google Play	7
Figure I.4 : Architecture générale d'Android	8
Figure I.5 : Transformation d'un fichier .Java à un fichier .dex	9
Figure I.6 : Cycle de vie d'une Activité sous Android	10
Figure I.7 : Le SDK Android	11
Figure I.8 : L'émulateur	12
Figure I.9 : Bureaux Virtuels	13
Figure I.10 : Les notifications	14
Figure I.11 : Boussole électronique	14

Chapitre II :

Figure II.1 : Classification des réseaux selon la taille	17
Figure II.2: Classification des réseaux selon la topologie	18
Figure II.3 : Les couches du modèle OSI	21
Figure II.4 : Correspondance entre les couches du modèle OSI et TCP/IP	22
Figure II.5 : Suite de protocoles TCP/IP.....	24
Figure II.6 : L'Internet, Intranet et Extranet.....	25
Figure II.7 : Architecture Client/Serveur	27
Figure II.8 : Architecture Client/Serveur à deux niveaux	27
Figure II.9: Architecture Client/Serveur à trois niveaux	28
Figure II.10 : Architecture Client/Serveur à N niveaux	29
Figure II.11 : L'interface phpMyAdmin	31
Figure II.12 : Exemple simple, définition d'un menu JSON vs XML	34
Figure II.13 : Schéma d'échange de données entre un client Android et un serveur MySQL...	35

Chapitre III :

Figure III.1 : Les différentes étapes de la modélisation UML suivies pour notre conception ...	40
Figure III.2 : Diagramme de cas d'utilisation	53
Figure III.3 : Diagramme de séquence des cas d'utilisation « Lancer l'application »	54
Figure III.4 : Diagramme de séquence des cas d'utilisation « Inscription »	55
Figure III.5 : Diagramme de séquence des cas d'utilisation « Connexion»	56
Figure III.6 : Diagramme de séquence des cas d'utilisation « Déposer une annonce »	57
Figure III.7 : Diagramme d'activités du cas d'utilisation «Avoir une aide à propos l'application»	59
Figure III.8 : Diagramme d'activités du cas d'utilisation « Inscription »	60
Figure III.9 : Diagramme d'activités du cas d'utilisation « Connexion pour accéder au compte utilisateur »	61
Figure III.10 : Diagramme d'activités du cas d'utilisation « Déposer une annonce »	62
Figure III.11 : Diagramme d'activités du cas d'utilisation « Modifier une annonce »	63
Figure III.12 : Diagramme d'activités du cas d'utilisation « Supprimer une annonce »	64
Figure III.13 : Diagramme d'activités du cas d'utilisation « Envoyer un message »	65
Figure III.14 : Diagramme d'activités du cas d'utilisation « Consulter la liste des annonces favorites »	66
Figure III.15 : Diagramme général de classes	68

Chapitre IV :

Figure IV.1 : Architecture à trois niveaux	71
Figure IV.2 : Éclipse ADT	72
Figure IV.3 : PHP-Éclipse	73
Figure IV.4 : Le WampServer	74
Figure IV.5 : Le serveur en ligne 000webhost	74
Figure IV.6 : Le protocole de communication utilisé	76
Figure IV.7 : JavaScript Object Notation	76
Figure IV.8 : Émulateur Android de java SDK	77
Figure IV.9 : Page index d'application	78
Figure IV.10 : Page d'accueil	79
Figure IV.11 : Page de connexion	80
Figure IV.12 : Page d'inscription	81

Figure IV.13 : Page mon compte	82
Figure IV.14 : Page déposer une annonce	83
Figure IV.15 : Page détails d'une annonce	84
Figure IV.16 : Page recherche d'une annonce	85
Figure IV.17 : Page Aide	86

Liste des Tableaux :

Chapitre I :

Tableau I.1 : La répartition des versions.....	7
---	----------

Chapitre II :

Chapitre III :

Tableau III.1 : Spécification des tâches.....	42
Tableau III.2: Spécification des scenarios	43
Tableau III.3 : Cas d'utilisation « Inscrire pour créer un compte client ».....	45
Tableau III.4 : Cas d'utilisation Effectuer une recherche selon le prix, catégorie et l'endroit ..	45
Tableau III.5 : Cas d'utilisation « Connecter pour accéder au compte client ».....	46
Tableau III.6 : Cas d'utilisation «Afficher les détails d'une annonce »	46
Tableau III.7 : Cas d'utilisation « Déposer une nouvelle annonce ».....	47
Tableau III.8 : Cas d'utilisation « Afficher mes annonces »	47
Tableau III.9 : Cas d'utilisation « Modifier une annonce »	48
Tableau III.10 : Cas d'utilisation « Supprimer une annonce »	49
Tableau III.11 : Cas d'utilisation « Afficher mes favoris ».....	49
Tableau III.12 : Cas d'utilisation « Afficher mes messages »	49
Tableau III.13 : Cas d'utilisation « Ajouter une annonce à mes favoris »	50
Tableau III.14 : Cas d'utilisation « Envoyer un message à propos d'une annonce ».....	50
Tableau III.15 : Table inscription.....	69
Tableau III.16 : Table annonce.....	69
Tableau III.17 : Table voiture	70
Tableau III.18 : Table location.....	70

Chapitre IV :

INTRODUCTION GÉNÉRALE

Un Smartphone (téléphone intelligent), est un téléphone mobile évolué disposant de fonctions d'un assistant numérique personnel, d'un appareil photo numérique et d'un ordinateur portable. La saisie des données se fait le plus souvent par le biais d'un écran tactile ou d'un clavier. Selon le principe d'un ordinateur, il peut exécuter divers logiciels/applications grâce à un système d'exploitation spécialement conçu pour mobiles, et donc en particulier fournir des fonctionnalités en plus de celles des téléphones mobiles classiques comme : l'agenda, la télévision, le calendrier, la navigation sur le Web, la consultation et l'envoi de courrier électronique, la géolocalisation, le dictaphone/magnétophone, la calculatrice, la boussole, l'accéléromètre, le gyroscope, la messagerie vocale et visuelle, la cartographie numérique, etc. Les Smartphones les plus sophistiqués bénéficièrent rapidement de la reconnaissance vocale et de la synthèse vocale. Il est possible de personnaliser son Smartphone en y installant des applications additionnelles tel que des jeux ou des utilitaires via un magasin d'applications en ligne différent pour chaque système d'exploitation comme Google Play sur Android ou encore l'App Store sur IOS. Les Smartphones ont besoin d'une connexion internet haut débit vers un réseau de téléphonie mobile pour pouvoir utiliser le maximum de leur potentiel. À partir de la fin 2007, le marché des Smartphones s'étend considérablement jusqu'à dépasser en quelques années celui des téléphones mobiles « classiques ».

Depuis quelques années, les équipements mobiles sont de plus en plus complexes et robustes. Les appareils mobiles deviennent une combinaison de la technologie informatique et de la technologie du téléphone mobile. Dans l'avenir, les mobiles peuvent remplacer généralement le rôle de l'ordinateur portable. C'est pour quoi, les systèmes d'exploitation pour les mobiles, de plus en plus, deviennent importants, particulièrement pour les fournisseurs, les développeurs et les éditeurs d'applications pour les mobiles.

Grâce à la capacité et le développement des équipements mobiles, on peut déployer des services pour les sociétés basés sur la technologie du téléphone mobile. Actuellement, il existe plusieurs systèmes d'exploitation pour les mobiles comme Windows Phone, Symbian, BlackBerry... De plus, il y a les plateformes libres ou code source ouvert comme Android, Moblin, etc.

INTRODUCTION GÉNÉRALE

Dans le cadre de préparation de notre diplôme de Master en Informatique, nous avons été amenés à effectuer notre projet sous android.

Il s'agit d'étudier et de développer une application mobile sous cette plateforme. Avec l'avancée et l'émergence des technologies mobiles, les développements embarqués sont de plus en plus demandés sur le marché. Avoir un Smartphone est devenu incontournable pour les jeunes de nos jours. Nous, nous intéressons au développement d'une application « Algérie Ventes et Achats », à installer sur les Smartphones permettant de créer des annonces qui peuvent être des offres ou des demandes, la possibilité de les modifier et même d'ajouter des photos, l'intéresser à une annonce il peut communiquer avec celui qui l'a déposée avec l'échange des messages ou un appel téléphonique afin de négocier le prix par exemple et de réaliser l'achat à la fin.

Enfin le but de notre application est de soutenir le marché en ligne et satisfaire les clients à propos du choix et de la qualité.

Ce rapport peut ainsi être subdivisé en quatre chapitres. Le premier consiste à la présentation des fonctionnalités d'android. Le second chapitre sera réservé à la présentation des généralités sur les réseaux ainsi le modèle Client/Serveur. Le troisième chapitre consiste en l'analyse et la conception avec la méthode UML qui servira d'une avant-première de l'application. Enfin, le quatrième et dernier chapitre c'est la partie réalisation qui consiste à présenter l'application avec les fonctionnalités de base et une notice permettant de comprendre comment utiliser le logiciel.



I.1 Introduction :

De par son ergonomie et les capacités du matériel, l'arrivée de l'iPhone d'Apple a bouleversé le paysage des systèmes d'exploitation mobiles, tant par les usages proposés que par les possibilités offertes aux utilisateurs avec le marché d'applications Apple Store.

La firme de Cupertino a su en quelques mois instaurer sa suprématie dans le monde de la téléphonie et surtout initier de nouveaux usages, qui ont conduit au raz-de-marée que l'on connaît. Aujourd'hui, les utilisateurs ne cessent de comparer leurs téléphones à l'iPhone : ergonomie, écran tactile, simplicité, multitude d'applications, etc. De créateur de nouveaux usages, l'iPhone est devenu un véritable standard pour les utilisateurs.

Devant les lacunes des offres des constructeurs historiques, une coalition s'est créée pour mettre au point et promouvoir un système d'exploitation mobile concurrent. Ce rassemblement, qui a vu le jour fin 2007, se nomme l'*Open Handset Alliance* et se compose aujourd'hui de 47 acteurs qui ont pour objectif de créer et de promouvoir un système complet, ouvert et gratuit dans le monde du mobile Android.

Tordons cependant le cou aux simplifications et notamment à celle trop souvent faite : « Android = Google phone ». Google est certes l'acteur majeur de ce rassemblement au tour d'Android, mais les téléphones Android disponibles sur le marché n'ont rien à voir avec cette firme, mis à part le « Nexus One », lancé tout début 2010 et qui est fabriqué spécialement pour Google. L'offre des constructeurs s'élargit d'ailleurs chaque jour davantage.

I.2 Définition d'Android :

Android [1], est un système d'exploitation pour Smartphones, Tablettes tactiles, PDA et terminaux mobiles. C'est un système open source utilisant le noyau Linux. Il a été lancé par une startup rachetée par Google en 2005.

CHAPITRE I : ANDROID

I.3 OHA (l'Open Handset Alliance):

Android est développé par l'OHA (Open Handset Alliance), une alliance internationale de compagnie. Cette alliance se compose de compagnie ne faisant pas partie du même secteur, Ainsi elle se compose :

- d'opérateurs mobiles (Vodafone, Telefonica, Telecom Italia, China Mobile, etc.)
- de fabricants de téléphones mobiles (Asus, HTC, LG, Motorola, etc.)
- de fabricants de semi-conducteur (Intel, Nvidia, ARM, etc.)
- d'éditeurs de logiciels (bay, Google, PacketVideo, etc.)
- de distributeurs (Aplix corporation, Borqs, TAT)

Aujourd'hui il y a 1,5 milliards de télévisions dans le monde. 3.2 milliard de personnes ont accès à internet. Mais en fin mars 2016, il y avait 6,8 milliards de connexions mobiles dans le monde, soit plus de 9,3 mobiles connectés pour 10 personnes, ce qui fait que le téléphone portable est le produit connaissant le plus grand succès dans le monde. C'est pour cela que l'OHA s'est lancée sur le secteur du mobile. Ils espèrent fournir une plateforme mobile innovante et performante, fournissant aux utilisateurs une nouvelle expérience d'utilisation de leurs mobiles.

I. 4 Histoire d'Android :

En avril 2004 [2] lors de la conception d'android par une équipe muni par Andy Rubin, un ancien d'android Inc, ce dernier déclare qu'Android a été prévu à l'origine pour être intégré dans des appareils photo plus précisément les smart camera, mais face à la baisse des ventes d'appareils photo et l'augmentation des ventes de smart phone, celui-ci est réorienté vers ce marché porteur.

En juillet 2005, Google a acquit Android Inc, une petite startup qui développait des applications pour téléphones mobiles. C'est à ce moment là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'était sûr, dans quels marchés il allait se positionner.

CHAPITRE I : ANDROID

Après ce rachat, à Google, une équipe dirigée par Andy Rubin a commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA soit créée officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que

Google développait des applications mobiles de son moteur de recherche, qu'ils développaient un nouveau téléphone mobile, etc.

En 2007, le 5 novembre, l'OHA a été officiellement annoncée, ainsi que son but. Développer des standards open source pour appareil mobile. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6.

En 2015, Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80 % de parts de marché dans les Smartphones.

I.5 Bugdroid :

Le personnage nommé Bugdroid est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous licence « creative commons by (3.0) » et peut donc être utilisé librement. Le site Engadget annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari : *Gauntlet: The Third Encounter*



Figure I.1 : Le logo de l'Android.

CHAPITRE I : ANDROID

I.6 Les différentes versions d'Android :

Les différentes versions d'android [2] ont toutes des noms de dessert(en anglais) et suivent un ordre Alphabétique depuis la version 1.5.

- Banana Bread (cake a la banane) 9 ou beta, version 1.1 .
- Cupcake (petit gâteau-avril 2009), API Level 3, version 1.5 .
- Donut (beignet-septembre2009), API Level 4, version 1.6 .
- Eclair (Eclair-janvier 2010), API Level 7, version 2.1 .
- FroYo (Frozen Yogourt/Yaourt glacé-mai2010), API Level 8, version 2.2 .
- Gingerbread (Pain d'épice-décembre2010), API Level 9, version 2.3 .
- Honeycomb (Gâteau de Miel-février 2011),API Level 11, version 3.0 .
- Ice Cream Sandwich API Level 14, version 4.0.
- Décembre 2011, API Level 15, version 4.0.3.
- Jelly Bean (Bonbon à la gelée/ Dragée-juillet 2012), API Level 16. version 4.1.
- Octobre 2012, API Level 17 version inclus dans la Samsung GALAXY camera, version 4.2 0.
- juillet 2013, API Level 18, version 4.3.
- KitKat, octobre 2013, API Level 19, version 4.4 .
- Lollipop, 3 novembre 2014, API21, version 5.0 .
- Lollipop, 9 mars 2015, API22, version 5.1 .
- Marshmallow , 5 octobre 2015, API23, version 6.0 .
- Nougat , 22 aout 2016, API24, version 7.0.



Figure I.1 : Les différents logos et noms des versions d'Android.

CHAPITRE I : ANDROID

I.7 Répartition des versions :

Répartition des différentes versions sur le Google Play Store, au 2 mai 2016, Basée sur le nombre d'appareils Android qui se sont connectés au Google Play Store dans une période de 14 jours se terminant à la date de collecte des données.d'après le site Wikipédia :

Version	Nom de code	Date de sortie	Version API	%
2.2-2.2.3	Froyo	20 mai 2010	8	0,1 %
2.3.3 - 2.3.7	Gingerbread	6 décembre 2010	10	1,7 %
4.0.3 - 4.0.4	Ice Cream Sandwich	19 octobre 2011	15	1,6 %
4.1.x	Jelly Bean	9 juillet 2012	16	6 %
4.2.x		13 novembre 2012	17	8,3 %
4.3.x		24 juillet 2013	18	2,4 %
4.4-4.4.4	KitKat	31 octobre 2013	19	29,2 %
5.0-5.0.2	Lollipop	3 novembre 2014	21	14,1 %
5.1.x		9 mars 2015	22	21,4 %
6.0	Marshmallow	8 octobre 2015	23	15,2 %

Tableau I.1: La répartition des versions

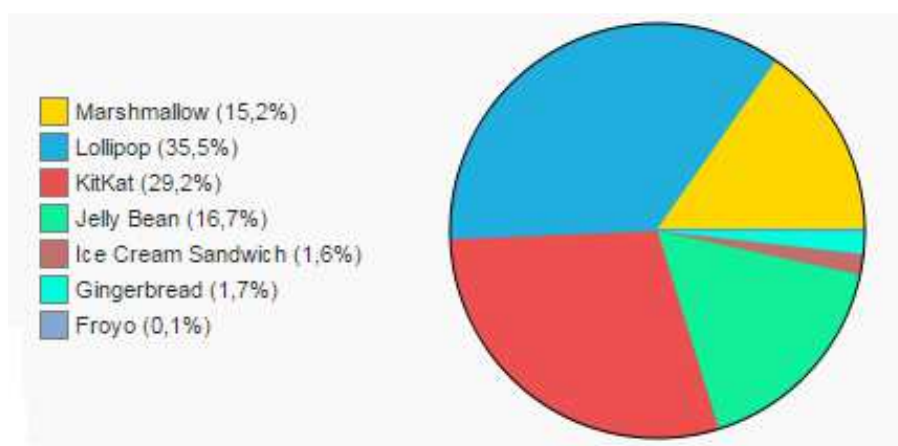


Figure I.2 : Graphique illustrant la part de chaque version d'Android sur le Google Play Store en 1 août 2016.

CHAPITRE I : ANDROID

I.8 L'Architecture générale d'Android :

L'architecture de l'Android [3] se compose de cinq couches. Ce sont le noyau Linux, les bibliothèques 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.

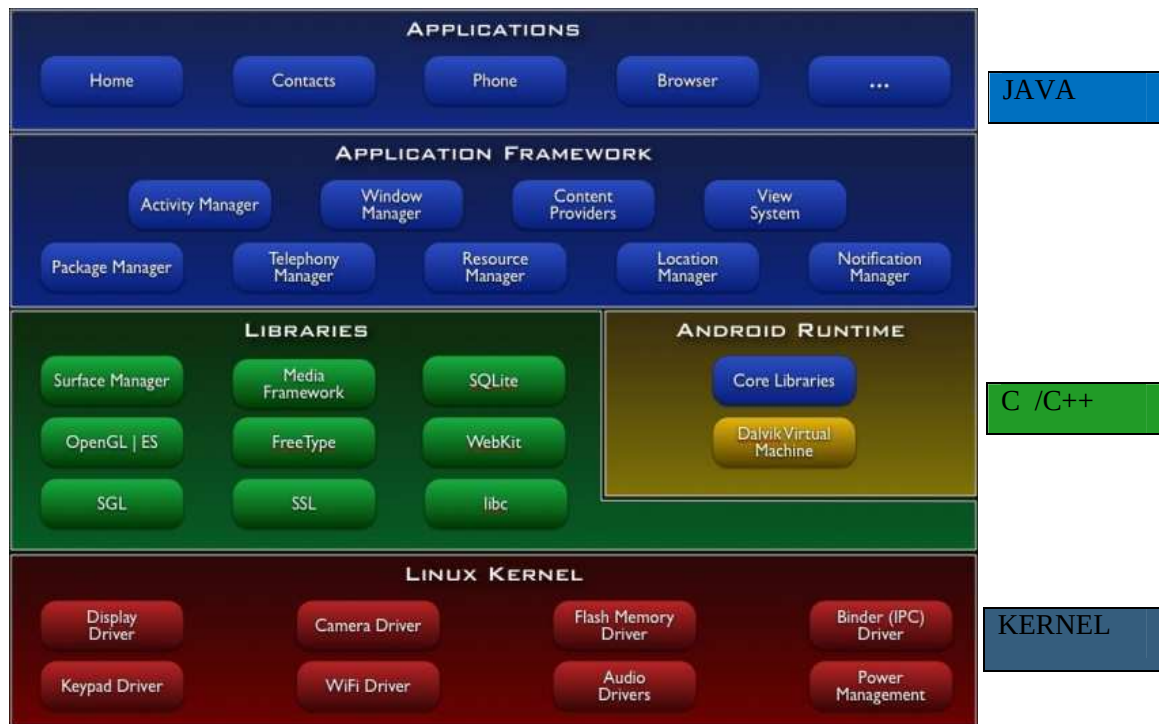


Figure I.3 : Architecture générale d'Android.

I.8.1 Les Applications : Android est fourni avec un ensemble de programmes de base (nommés applications natives) permettant d'accéder à des fonctionnalités comme les courriels, les SMS, le calendrier, les photos, le web, etc. Ces applications sont développées à l'aide du langage JAVA. Pour l'utilisateur final, c'est la seule couche accessible et visible. Il est bien sûr possible d'installer des applications tierces de provenance variées : Android Market, site Internet et bien sûr, directement par le biais de votre ordinateur.

I.8.2 Un framework applicatif (Application framework) : qui expose de façon agnostique les services du système à la couche applicative, comme le Window Manager (gestionnaire de fenêtrage) et le Location Manager (gestionnaire de géolocalisation), les Content Providers (gestionnaires de contenu), la téléphonie et les capteurs.

CHAPITRE I : ANDROID

I.8.3 Bibliothèque : C'est une collection de plusieurs bibliothèques qui sont basées sur les langages C/C++ utilisées par les applications Android comme:

- **Bionic :** c'est une bibliothèque qui implémente la bibliothèque Libc (bibliothèque de la fonctionnalité), Webkit navigateur (pour web navigateur), MediaFramework (pour soutenir vidéo, audio normal), SQLite (pour stoker des données, etc...).
- **HAL** (Hardware Abstraction Libraries) : pour séparer entre le hardware et la plateforme logique.

I.8.4 Moteur d'exécution d'Android (Android Runtime) : se compose des bibliothèques du cœur (CoreLibraries pour JAVA), la machine virtuelle Dalvik (fournit un environnement qui peut fonctionner sur un CPU limité et dans les situations de faible énergie. Il utilise le format du fichier .DEX pour optimiser les ressources comme CPU, RAM. Pour les fichiers .jar, le système utilise un outil intégré pour convertir le "byte code" .jar en .dex .



Figure I.5 : Transformation d'un fichier. Java à un fichier .dex

I.8.5 Linux Kernel : Android est basé sur le noyau linux2.6 alors, il à plusieurs avantages comme une grande mémoire, gestion de processus, modèle de sécurité, soutien de bibliothèque partagée, etc. il fournit les drivers pour communiquer entre les hardwares et leurs logiciels. De plus, il y a les parties d'augmentation comme la gestion d'énergie.

CHAPITRE I : ANDROID

I.9 Les composants d'une application :

Les applications Android sont composées de 4 principaux :

- **Activity** : Gestion de la vie des applications, car une application se compose de plusieurs écrans. Et chaque écran peut être réalisé par une activité, implémentée sous la forme d'une classe qui hérite de la classe Activity.
- **Services** : Les services n'ont pas d'interface graphique et tournent en tâche de fond (en arrière-plan). C'est-à-dire qu'une application s'exécute quand une autre application est en train de s'exécuter comme les services de lecture de musique.
- **Broadcast receivers** : Il écoute et réagit aux annonces broadcast (réactions sur les événements extérieurs). Par exemple appel entrant, changement de fuseau horaire.
- **Content providers** : Gestion du partage de données entre applications.

En résumé Android permet le partage de composant entre applications ainsi que de gérer leur cycle de vie. Cycle de vie des activités plus précisément. Ce cycle de vie est exprimé par la figure ci-dessous.

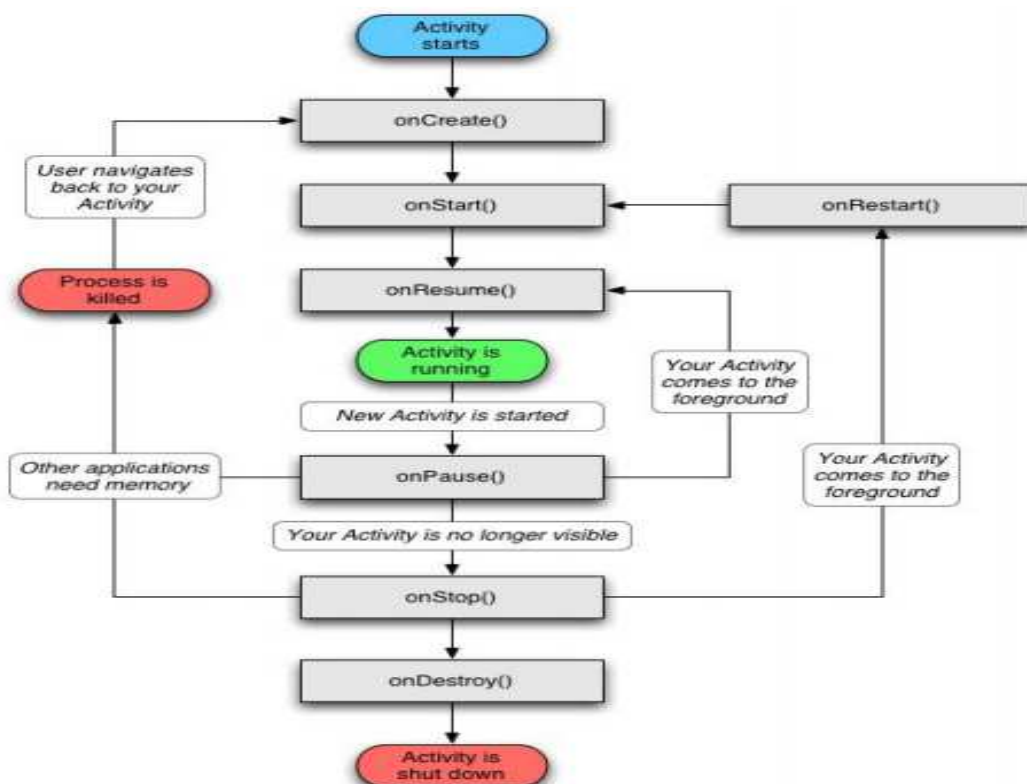


Figure I.6 : Cycle de vie d'une Activité sous Android.

CHAPITRE I : ANDROID

I.10 Prise en main de l'environnement :

I.10.1 Le SDK Android :

L'outil le plus important est le SDK Android [4], facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.). Il permet également de télécharger les différentes versions de Google APIs (APIs pour intégrer des fonctionnalités liées à Google tels que Maps etc.) ou de la documentation JavaDoc. Son fonctionnement est similaire aux gestionnaires de paquets de Linux.

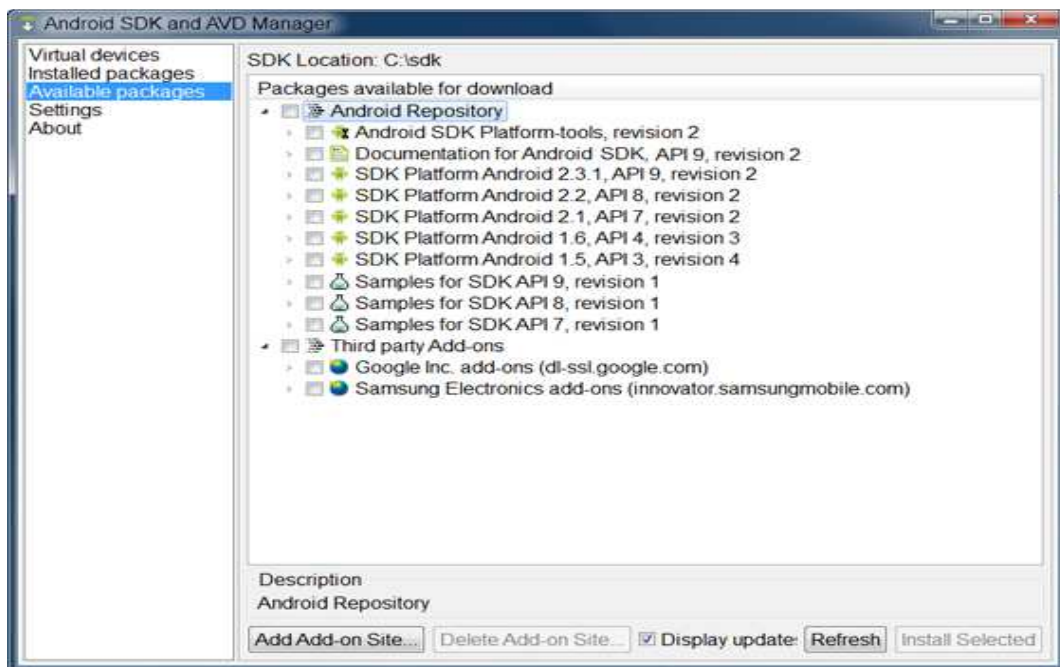


Figure I.7 : le SDK Android.

I.10.2 ADT pour Eclipse :

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

Android Développent Tools (ADT) [4] est très complet et surtout très pratique : conception graphique d'interfaces utilisateur, debug sur un téléphone, gestion de l'architecture de fichiers d'une application etc.

CHAPITRE I : ANDROID

I.10.3 Emulateur :

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. La figure suivante illustre un exemple d'un émulateur.

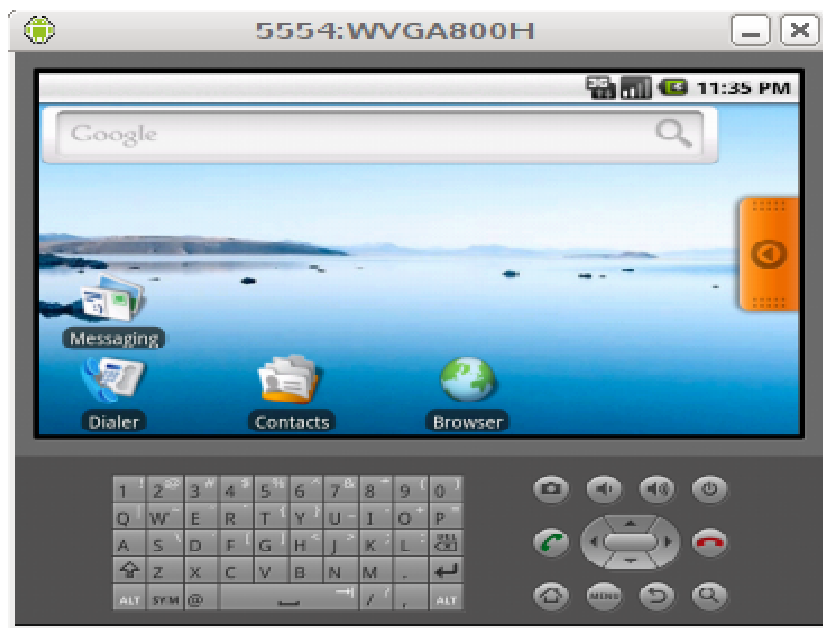


Figure I.8: L'émulateur.

I.11 Concept et innovation :

I.11.1 Bureaux Virtuels :

Le bureau est étendu sur 5 parties (ou plus, ou moins). 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.

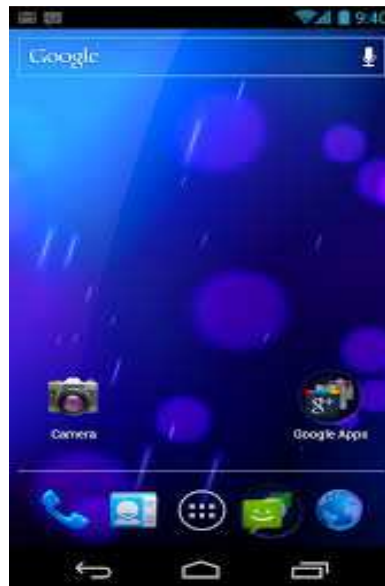


Figure I.9: Bureaux Virtuels

I.11.2 Widgets :

Comme pour les bureaux des systèmes d'exploitation récents, il est possible de placer des widgets (calendrier, horloge, notes, ...) sur le bureau. Ils permettent de donner des informations variables et de fournir des interactions.

I.11.3 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.

CHAPITRE I : ANDROID

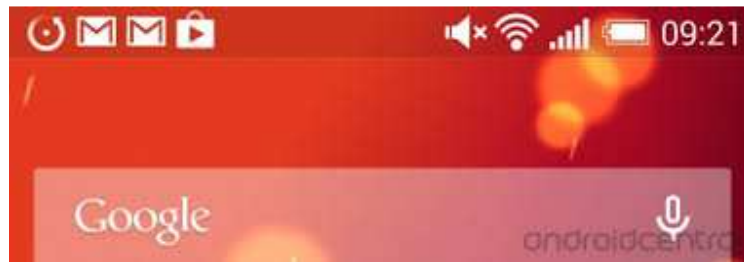


Figure I.10 : Les notifications.

I.11.4 Boussole électronique :

De nombreux programmes tirent parti d'une boussole électronique intégrée à certains appareils, le premier étant *Google Maps* qui propose le mode *Street View* en utilisant la boussole et l'accéléromètre, ce qui permet de mieux se repérer. Dans les autres cas, *Google Maps* déduit la direction (comme tous les logiciels utilisant le GPS) des coordonnées précédentes. Certaines applications s'en servent également pour détecter des champs magnétiques et ainsi permettre la détection de certains métaux.

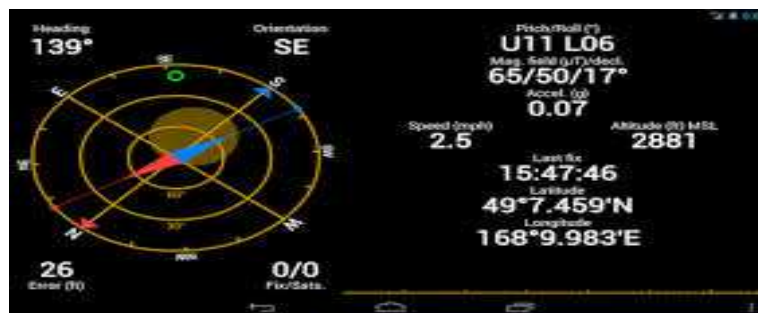


Figure I.11 : La boussole électronique.

I.12 Conclusion :

Ce chapitre nous a permis d'aborder les différentes versions et leur répartition de systèmes d'exploitation android conçus pour fonctionner sur les appareils mobiles. Ainsi on a étudié les différentes couches de l'architecture d'Android et les composants d'une application android, qui nous permet de développer facilement notre application.

Le chapitre suivant sera consacré aux généralités sur les réseaux et le modèle client/serveur de notre application.

CHAPITRE II
GÉNÉRALITÉS SUR
LES RESEAUX ET
LE MODELE
CLIENT/SERVEUR

II.1 Introduction :

Aujourd'hui, l'utilisation de la technologie pour développer et renforcer notre capacité de communication arrive à un tournant. La généralisation de l'utilisation des réseaux à l'échelle mondiale s'est opérée plus vite que quiconque aurait pu l'imaginer. L'évolution rapide des réseaux induit un bouleversement des interactions sociales, commerciales, politiques et même personnelles. Ce chapitre aura pour objectif de présenter quelques notions sur les réseaux informatiques en premier lieu puis donnera un aperçu sur le modèle Client/Serveur.

II.2 Définitions :

- **Réseau:** C'est un ensemble d'ordinateurs (ou de périphériques) autonomes connectés entre eux et qui sont situés dans un certain domaine géographique.
- **Téléinformatique :** informatique à distance. C'est une science qui associe les problèmes liés au traitement des données et aux problèmes liés au transport de l'information.

II.3 Avantages des réseaux :

Les Réseaux permettent :

- De partager les fichiers.
- Le transfert de fichier.
- Le partage d'application : compilateur, système de gestion de base de données (SGBD).
- Partage d'imprimante.
- L'interaction avec les utilisateurs connectés : messagerie électronique, conférence électronique, Talk, ...
- Le transfert de données (réseaux informatiques).
- Le transfert de la parole (réseaux téléphoniques).
- Le transfert de la parole, de la vidéo et des données (réseaux à intégration de services ou multimédia).

II.4 Classification des réseaux :

Les réseaux informatiques [5] qui permettaient à leur origine de relier des terminaux passifs à des gros ordinateurs centraux autorisent à l'heure actuelle l'interconnexion de tous types, d'ordinateurs que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Les services qu'ils offrent font partie de la vie courante des entreprises des administrations (banques, gestion, commerce, bases de données, recherche ...) et des particuliers (messagerie, loisirs, services d'informations par minitel et Internet ...).

Ces réseaux peuvent être classés selon différents critères :

II.4.1 Selon la taille :

On peut faire une première classification des réseaux à l'aide de leur taille [5] comme on peut le voir dans la figure 1.1.

II.4.1.1 Un réseau personnel (Personal Area Network ou PAN) : désigne un type de réseau informatique restreint en termes d'équipements, généralement mis en œuvre dans un espace d'une dizaine de mètres. Les bus utilisés les plus courants pour l'établissement d'un réseau individuel sont l'USB, les technologies sans fil telles que Bluetooth, l'infrarouge.

II.4.1.2 Un réseau local (Local Area Network ou LAN) : peut s'étendre de quelques mètres à quelques kilomètres et correspond au réseau d'une entreprise. Il peut se développer sur plusieurs bâtiments et permet de satisfaire tous les besoins internes de cette entreprise.

II.4.1.3 Un réseau métropolitain (Metropolitan Area Network ou MAN) : interconnecte plusieurs lieux situés dans une même ville, par exemple les différents sites d'une université ou d'une administration, chacun possédant son propre réseau local.

II.4.1.4 Un réseau étendu (Wide Area Network ou WAN) : permet de communiquer à l'échelle d'un pays, ou de la planète entière, les infrastructures physiques pouvant être terrestres ou spatiales à l'aide de satellites de télécommunications.

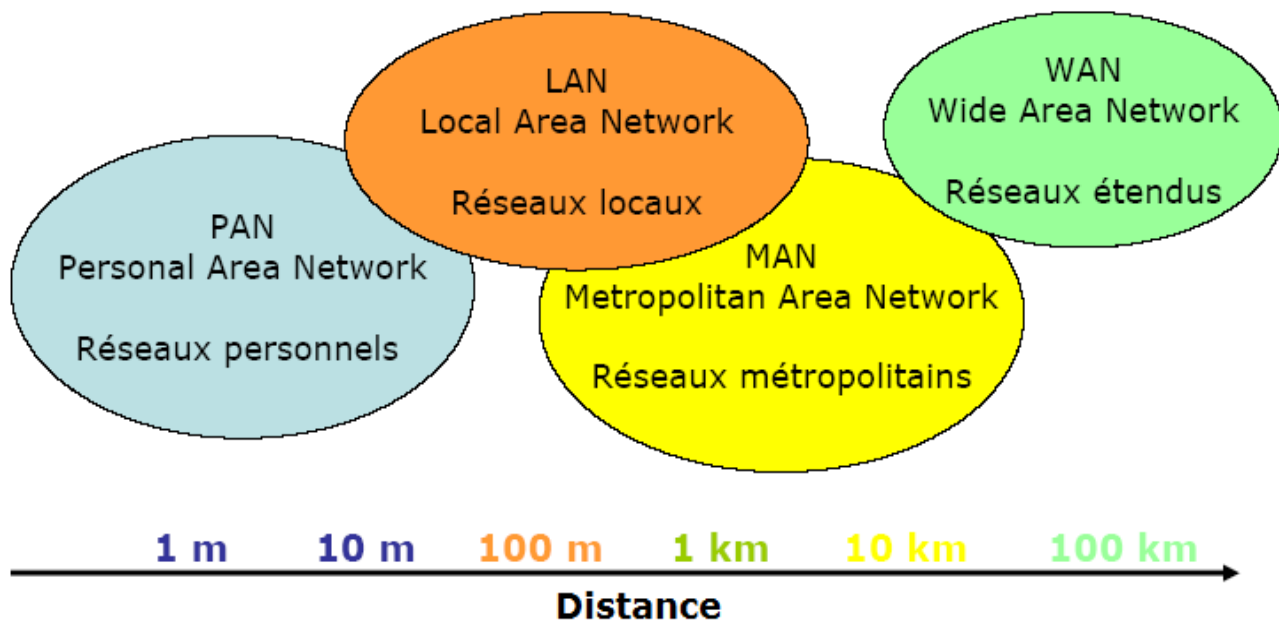


Figure II.1 : Classification des réseaux selon la taille.

II.4.2 Selon la topologie :

L'arrangement physique des éléments constitutifs d'un réseau est appelé topologie physique. On trouve les topologies suivantes :

II.4.2.1 En anneau : Dans un réseau possédant une topologie en anneau, les ordinateurs sont situés sur une boucle et communiquent chacun à leur tour. En réalité, dans une topologie anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé MAU, Multi station Access Unit) qui va gérer la communication entre les ordinateurs qui lui sont reliés en impartissant à chacun d'entre eux un temps de parole.

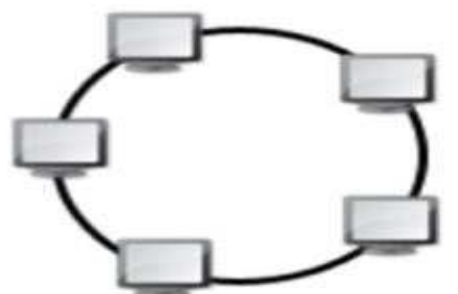
II.4.2.2 En bus : Dans une topologie en bus, tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial. Le mot « bus » désigne la ligne physique qui relie les machines du réseau. Cette topologie a pour avantage d'être facile à mettre en œuvre et de posséder un fonctionnement simple. En revanche, elle est extrêmement vulnérable étant donné que si l'une des connexions est défectueuse, l'ensemble du réseau en est affecté.

II.4.2.3 En étoile:

Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel central appelé concentrateur (en anglais hub). Ce concentrateur a pour rôle d'assurer la communication entre les nœuds de réseau (échange d'informations entre les nœuds du réseau). Une topologie en étoile est beaucoup moins vulnérable car une des connexions peut être débranchée sans paralyser le reste du réseau.

II.4.2.4 En maille :

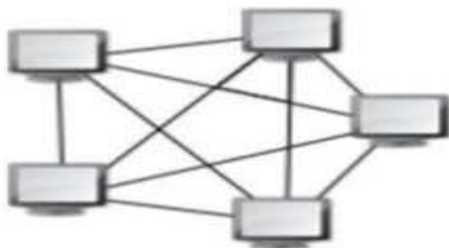
Une topologie maillée, est une évolution de la topologie en étoile, elle correspond à plusieurs liaisons points à points. Une unité réseau peut avoir (1, N) connexions point à point vers plusieurs autres unités. Chaque terminal est relié à tous les autres. L'inconvénient est le nombre de liaisons nécessaires qui devient très élevé. Cette topologie se rencontre dans les grands réseaux de distribution (Exemple: Internet). L'information peut parcourir le réseau suivant des itinéraires divers, sous le contrôle de puissants superviseurs de réseau, ou grâce à des méthodes de routage réparties.



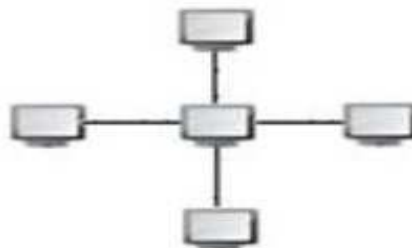
ARCHITECTURE EN ANNEAU



ARCHITECTURE EN BUS



ARCHITECTURE EN MAILLE



ARCHITECTURE EN ETOILE

Figure II.2 : Classification des réseaux selon la topologie.

II.4.3 Selon le mode de diffusion :

On peut également différencier les réseaux selon leur mode de diffusion :

II.4.3.1 Le mode en diffusion :

Ce mode de fonctionnement consiste à partager un seul support de transmission. Chaque message envoyé par un équipement sur le réseau est reçu par tous les autres. C'est l'adresse spécifique placée dans le message qui permettra à chaque équipement de déterminer si le message lui est adressé ou non. À tout moment un seul équipement a le droit d'envoyer un message sur le support, il faut donc qu'il écoute au préalable si la voie est libre; si ce n'est pas le cas il attend selon un protocole spécifique à chaque architecture. Les réseaux locaux adoptent pour la plupart le mode diffusion sur une architecture en bus ou en anneau et les réseaux satellitaires ou radio suivent également ce mode de communication. Dans une telle configuration la rupture du support provoque l'arrêt du réseau, par contre la panne d'un des éléments ne provoque pas (en général) la panne globale du réseau.

II.4.3.2 Le mode point à point :

Dans le mode diffusion point à point le support physique (le câble) relie une paire d'équipements seulement. Quand deux éléments non directement connectés entre eux veulent communiquer ils le font par l'intermédiaire des autres nœuds du réseau. Dans le cas de l'étoile le site central reçoit et envoie tous les messages, le fonctionnement est simple, mais la panne du nœud central paralyse tout le réseau. Dans une boucle simple, chaque nœud recevant un message de son voisin en amont le réexpédie à son voisin en aval. Pour que les messages ne tournent pas indéfiniment le nœud émetteur retire le message lorsqu'il lui revient. Si l'un des éléments du réseau tombe en panne, alors tout s'arrête. Ce problème est partiellement résolu par la double boucle dont chacune des boucles fait tourner les messages dans un sens opposé. En cas de panne d'un équipement, on reconstitue une boucle simple avec les éléments actifs des deux boucles, mais dans ce cas tout message passera deux fois par chaque nœud. Il en résulte alors une gestion plus complexe. Dans le maillage régulier l'interconnexion est totale ce qui assure une fiabilité optimale du réseau, par contre c'est une solution coûteuse en câblage physique. Si l'on allège le plan de câblage, le maillage devient irrégulier et la fiabilité peut rester élevée mais elle nécessite un routage des messages selon des algorithmes parfois complexes. Dans cette architecture il devient presque impossible de prévoir le temps de transfert d'un nœud à un autre.

II.4.4 Selon le mode de commutation :

Le réseau doit permettre l'échange de messages entre les abonnés quelle que soit leur localisation. La commutation rassemble toutes les techniques qui réalisent la mise en relation de deux abonnés quelconques. Il existe trois techniques de commutation :

II.4.4.1 La commutation de circuits :

C'est historiquement la première à avoir été utilisée, par exemple dans le réseau téléphonique à l'aide des autocommutateurs. Elle consiste à créer dans le réseau un circuit particulier entre l'émetteur et le récepteur avant que ceux-ci ne commencent à échanger des informations. Ce circuit sera propre aux deux entités communicantes et il sera libéré lorsque l'un des deux coupera sa communication. Par contre, si pendant un certain temps les deux entités ne s'échangent rien le circuit leur reste quand même attribué. C'est pourquoi, un même circuit (ou portion de circuit) pourra être attribué à plusieurs communications en même temps. Cela améliore le fonctionnement global du réseau mais pose des problèmes de gestion (files d'attente, mémorisation, etc.).

II.4.4.2 La commutation de messages :

Elle consiste à envoyer un message de l'émetteur jusqu'au récepteur en passant de nœud de commutation en nœud de commutation. Chaque nœud attend d'avoir reçu complètement le message avant de le réexpédier au nœud suivant. Cette technique nécessite de prévoir de grandes zones tampon dans chaque nœud du réseau, mais comme ces zones ne sont pas illimitées il faut aussi prévoir un contrôle de flux des messages pour éviter la saturation du réseau. Dans cette approche il devient très difficile de transmettre de longs messages.

II.4.4.3 La commutation de paquets:

Un message émis est découpé en paquets et par la suite chaque paquet est commuté à travers le réseau comme dans le cas des messages. Les paquets sont envoyés indépendamment les uns des autres et sur une même liaison on pourra trouver les uns derrière les autres des paquets appartenant à différents messages. Chaque nœud redirige chaque paquet vers la bonne liaison grâce à une table de routage. La reprise sur erreur est donc ici plus simple que dans la commutation de messages, par contre le récepteur final doit être capable de reconstituer le message émis en réassemblant les paquets. Ceci nécessitera un protocole particulier car les paquets peuvent ne pas arriver dans l'ordre initial, soit parce qu'ils ont emprunté des routes différentes, soit parce que l'un d'eux a dû être réémis suite à une erreur de transmission.

II.5 Le modèle OSI :

II.5.1 Définition :

À l'origine, le modèle OSI [5] (*en anglais Open Systems Interconnection*) a été conçu par l'organisation ISO pour fournir un cadre dans lequel concevoir une suite de protocoles système ouverts. L'idée était que cet ensemble de protocoles serait utilisé pour développer un réseau international qui ne dépendrait pas de systèmes propriétaires.

II.5.2 Les couches du modèle OSI :

Le modèle OSI est un modèle qui comporte sept couches. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.

Les sept couches du modèle OSI sont :

- Couche physique.
- Couche liaison des données.
- Couche réseau.
- Couche transport.
- Couche session.
- Couche présentation.
- Couche application.

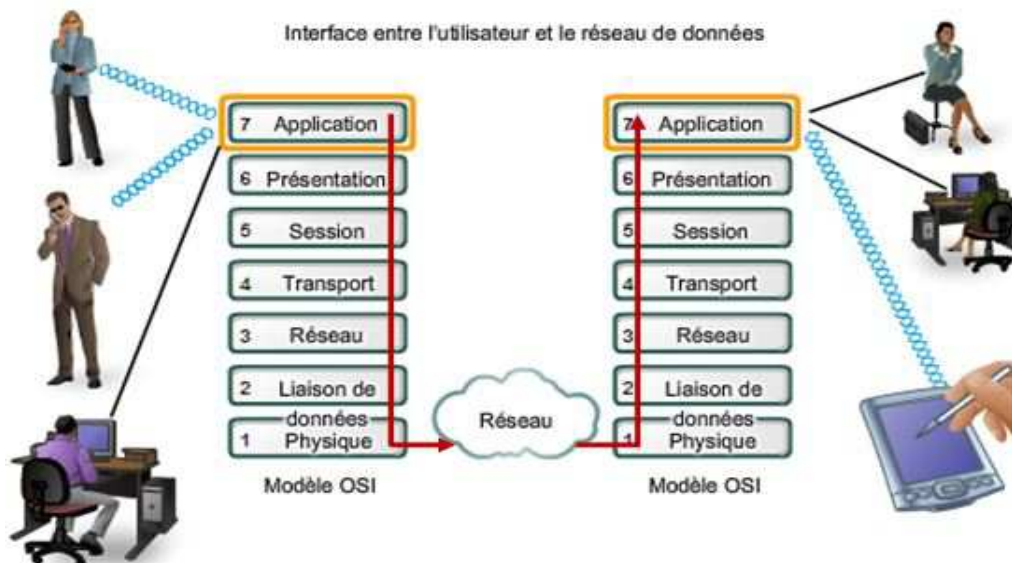


Figure II.3: Les couches du modèle OSI.

II.6 Le modèle TCP/IP :

II.6.1 Définition :

TCP/IP est un protocole ou bien une règle de communication [5]. C'est le principal protocole utilisé sur internet. TCP/IP signifie Transmission Control Protocol / Internet Protocol. Il s'agit des procédés de communication qui forment les bases d'Internet. Littéralement on peut le traduire par le protocole de contrôle de transmission et le protocole Internet.

TCP prend en charge l'ouverture et le contrôle de la liaison entre deux périphériques et le protocole d'adressage IP assure quant à lui le routage des données transmises ou reçues. Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules ou couches).

II.6.2 Les couches du modèle TCP/IP :

Le modèle TCP/IP contient uniquement quatre couche qui ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

Les couches de modèle TCP/IP sont :

- La couche Accès réseau.
- La couche Internet.
- La couche Transport.
- La couche Application.

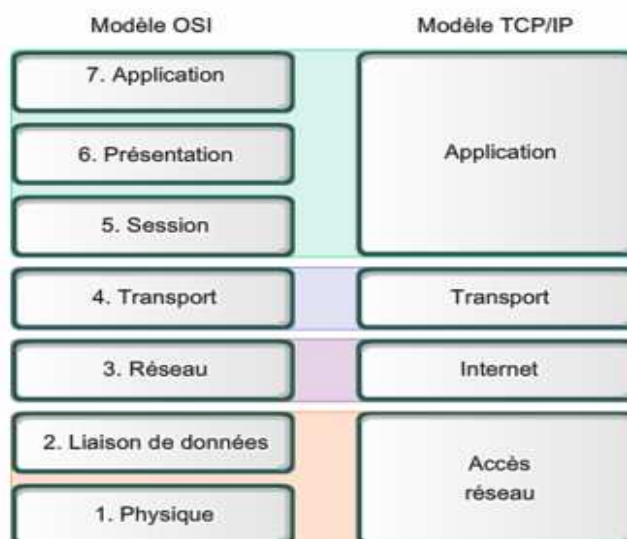


Figure II.4 : Correspondance entre les couches du modèle OSI et TCP/IP.

II.7 La pile de protocoles :

Les protocoles réseau définissent un format et un ensemble communs de règles [6] d'échange des messages entre les périphériques. L'interaction entre un Serveur et un Client constitue un exemple de l'utilisation d'une suite de protocoles dans des communications réseau.

Les différents protocoles fonctionnent entre eux pour garantir que les messages sont reçus et compris par les deux parties.

II.7.1 Protocoles d'application : le protocole de transfert hypertexte (HyperText Transfer Protocol, protocole HTTP) régit la manière dont un serveur et un client interagissent. Ce protocole décrit le contenu et la mise en forme des requêtes et des réponses échangées entre le client et le serveur. Les logiciels du client et du serveur Web implémentent le protocole HTTP dans le cadre de l'application.

II.7.2 Protocoles de transport : le protocole de contrôle de transmission (Transmission Control Protocol, TCP) gère les conversations individuelles entre les serveurs et les clients. Le protocole TCP divise les messages HTTP en petites parties appelées segments. Ces segments sont envoyés entre les processus du serveur Web et du client exécutés sur l'hôte de destination. Ce protocole est également responsable du contrôle de la taille et du débit d'échange de messages entre le serveur et le client.

II.7.3 Protocoles Internet : le protocole IP (IP, *Internet Protocol*) se charge d'encapsuler en paquets les segments mis en forme par le protocole TCP, de les attribuer aux adresses appropriées et de les remettre à l'hôte de destination en utilisant le meilleur chemin.

II.7.4 Protocoles d'accès au réseau : les protocoles d'accès au réseau décrivent deux fonctions principales, d'une part la communication sur une liaison de données et d'autre part la transmission physique des données sur le support réseau. Les protocoles de gestion de liaison de données prennent les paquets depuis le protocole IP et les formatent pour les transmettre à travers les supports. Les normes et les protocoles des supports physiques régissent la manière dont les signaux sont envoyés, ainsi que leur interprétation par les clients destinataires. Ethernet est un exemple de protocole d'accès au réseau.

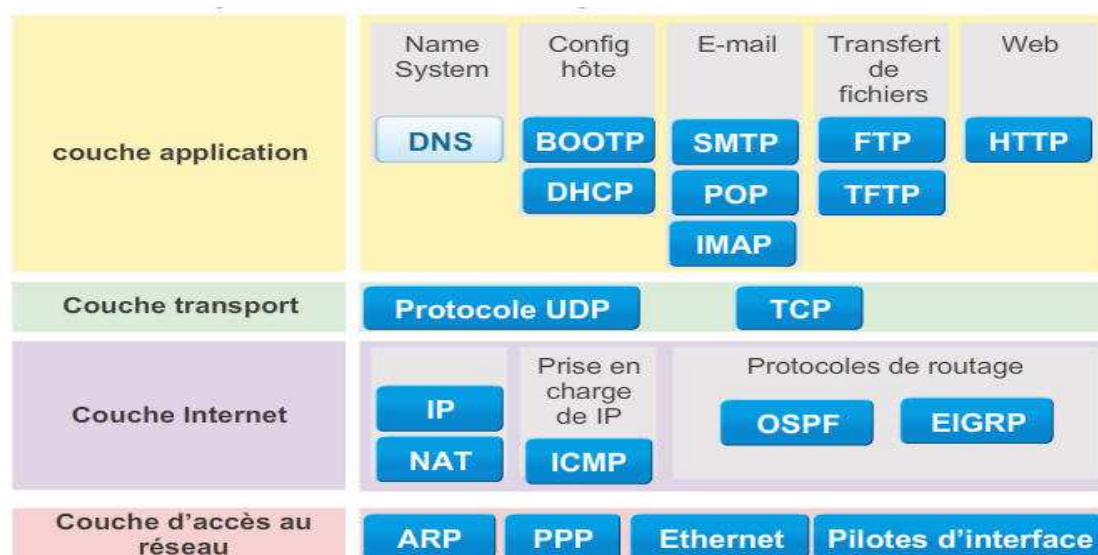


Figure II.5 : Suite de protocoles TCP/IP.

II.8 L'Internet, Intranet et Extranet :

II.8.1 Internet :

L'internet est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données. C'est un réseau de réseaux, composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. Internet transporte une vaste quantité d'informations.

II.8.2 Extranet :

C'est une extension du système d'information d'une entreprise à des partenaires, qui est situé au delà de réseau. L'accès à l'extranet doit être sécurisé dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise.

II.8.3 Intranet :

Un intranet est un ensemble de services internet (par exemple un serveur web) internes à un réseau local, c'est-à-dire accessibles uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisibles (ou inaccessibles) de l'extérieur.

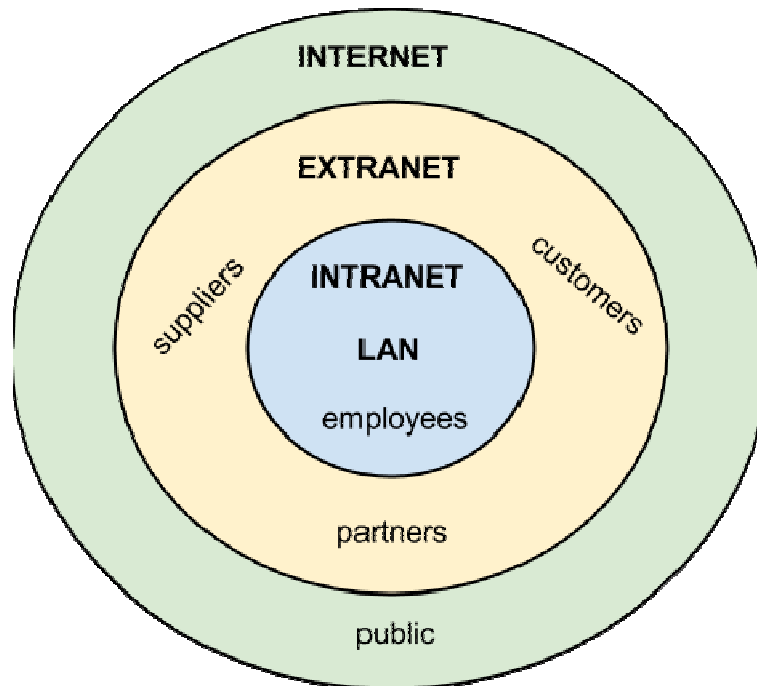


Figure II.6: L'Internet, Intranet et Extranet.

II.9 Le modèle Client/Serveur :

II.9.1 Définition:

L'environnement **client-serveur** désigne un mode de communication à travers un réseau [7] entre plusieurs programmes ou logiciels : l'un, qualifié de client(s), envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également le périphérique sur lequel est exécuté le logiciel client, et le serveur, le périphérique sur lequel est exécuté le logiciel serveur.

En général, les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en ce qui concerne la puissance de calcul, les entrées-sorties et les connexions réseau. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette, Smartphone...), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

II.9.2 Les types de Serveurs :

Il existe une grande variété de logiciels serveurs en fonction des besoins à servir :

II.9.2.1 Serveurs d'application Web :

L'internet est la plus grande application client/serveur dite intergalactique, ce nouveau modèle consiste en des clients légers et portables qui communiquent avec de très gros serveurs. Le serveur Web par exemple, renvoie des documents lorsque le client les demande par leurs noms. Clients et serveurs communiquent via un protocole de type RPC appelé HTTP (Hyper Text Transmission Protocol).

II.9.2.2 Serveurs de fichiers :

Dans ce modèle de serveur de fichier, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers, qui sont utiles pour partager des fichiers sur le réseau. Sa faiblesse réside dans l'obtention de l'information qui nécessite de nombreux échanges de messages sur le réseau.

II.9.2.3 Serveurs de base de données :

Dans ce modèle les clients envoient des requêtes sous forme de messages en direction de serveurs. Le résultat de chaque requête est envoyé au client. Le serveur utilise sa propre capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client et de le laisser en faire la sélection. Ainsi la puissance répartie est utilisée de façon beaucoup plus efficace.

II.9.2.4 Serveurs de transactions :

Dans ce modèle les clients invoquent des procédures distantes résident sur le serveur qui comporte un moteur de base de données. Ces procédures distantes exécutent un ensemble d'instructions, l'échange sur le réseau consiste en un seul message de requête/réponse.

II.9.2.5 Serveurs de groupware :

Le groupware s'intéresse à la gestion d'informations semi structurées telles que le texte, l'image, le courrier, la messagerie et l'ordonnancement des tâches.

II.10 Architecture Client/Serveur

II.10.1 Définition [7]:

Architecture de réseau dans laquelle toutes les informations sont localisées sur des ordinateurs ayant le rôle de serveur, et accessible aux ordinateurs clients.

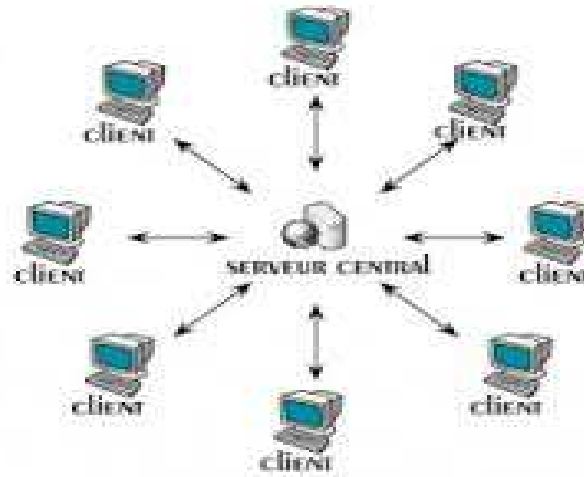


Figure II.7 : Architecture Client/serveur.

II.10.2 Les différents types d'architecture Client/Serveur:

II.10.2.1 Architecture 2-TIERS :

L'architecture à deux niveaux (auss appelée architecture 2-tiers) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

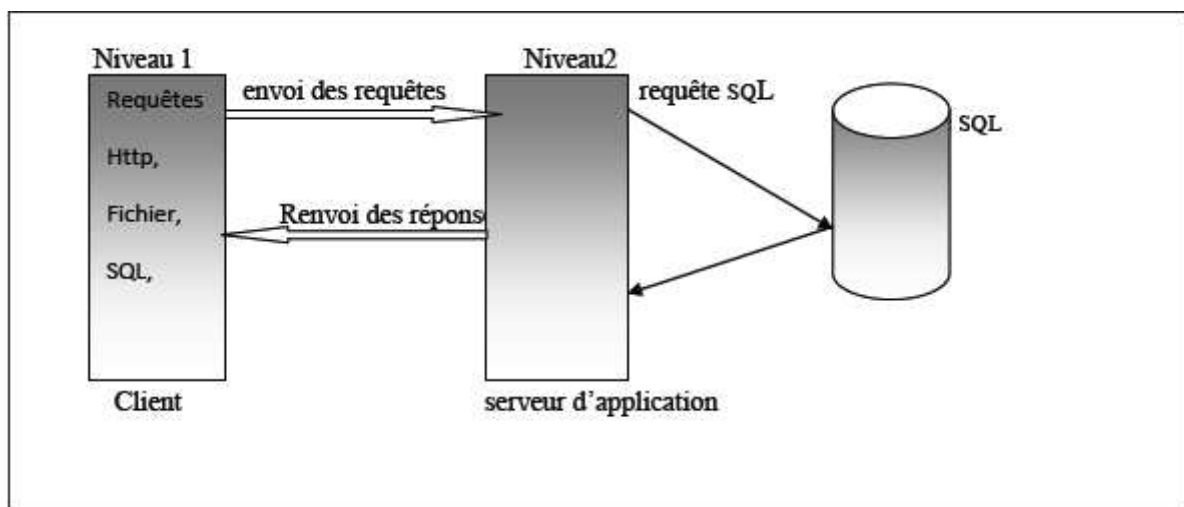


Figure II.8 : Architecture client/serveur à deux niveaux.

II.10.2.1.1 Avantage et Inconvénients d'une architecture à 2- tiers :

L'avantage de cette architecture est l'absence de duplication de données, ainsi sa gestion simple de la cohérence et de l'intégrité des données, et la maîtrise globale des traitements. En revanche, ce modèle est trop rigide qui n'assure pas l'évolutivité et économiquement trop coûteux.

II.10.2.2 Architecture 3-TIERS :

Dans l'architecture 3-tiers (appelée également architecture à 3 niveaux), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre:

- Un client, c'est-à-dire le périphérique (ex: ordinateur) demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur
- Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

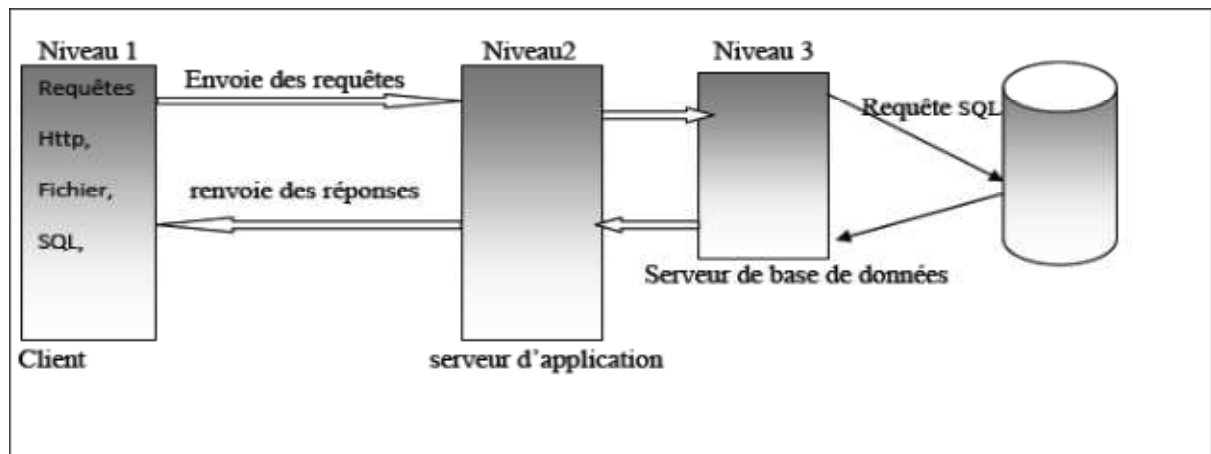


Figure II.9 : Architecture client/serveur à trois niveaux.

II.10.2.2.1 Avantages et Inconvénients d'une architecture à 3-tiers :

L'avantage de cette architecture s'est qu'elle a une meilleure répartition de charge, économiquement moins cher et plus évolutif. En revanche, elle a une Administration plus compliquée et une mise en œuvre plus compliquée.

II.10.2.3 Architecture N-TIERS :

Elle supprime tous les inconvénients des architectures précédentes parce qu'elle permet de distribuer plus librement la logique applicative et l'utilisation d'interface utilisateurs. Elle est riche et offre une grande capacité d'extension et facilite la gestion des sessions.

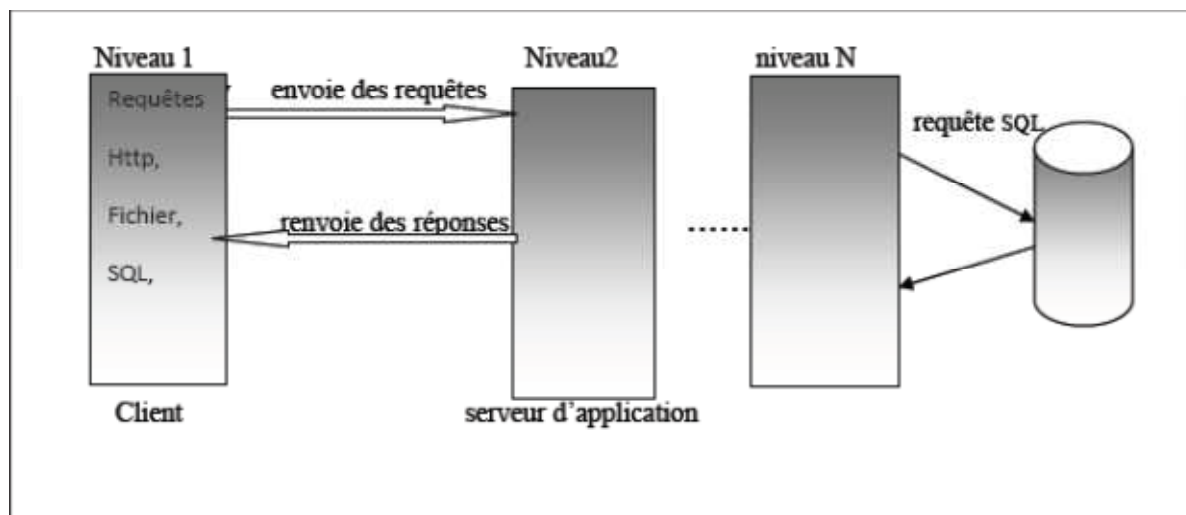


Figure II.10: Architecture client/serveur à N niveaux.

II.11 WampServer :

II.11.1 Définition:

Le WampServer est une plateforme de développement Web de type WAMP (**W**indows **A**pache **M**ySQL **P**HP) [8], permettant de faire fonctionner localement sans se connecter à un serveur externe des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration des bases MySQL.

II.11.2 Le serveur Apache :

Le plus populaire des serveurs web sur Internet depuis 1996. Il a été créé dans la volonté de développer et de maintenir un serveur HTTP sécurisé, efficace et évolutif pour les systèmes d'exploitation modernes. Apache est multi plateformes et gratuit, son installation est facile, rapide et son utilisation pas très compliqué. Grâce à une association avec PHP apache devient un serveur dynamique. Il est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc.

II.11.3 Le serveur MySQL :

Est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.

II.11.3.1 Le langage SQL :

SQL est l'acronyme de Structured Query Language [9]. C'est un langage informatique normalisé servant à exploiter des bases de données relationnelles. Créé en 1974, normalisé depuis 1986, le langage est reconnu par la grande majorité des systèmes de gestion de bases de données relationnelles du marché.

Le langage SQL comporte :

- Une partie langage de manipulation des données (LDD) de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. Les commandes utilisées sont : *CREATE*, *ALTER*, *DROP* et *RENAME* qui peuvent porter sur les objets de type *TABLE*, *INDEX*, *VIEW*, *SEQUENCE*, *SYNONYM* et *USER*.
- Une partie langage de définition des données (LMD) permet de créer et de modifier l'organisation des données dans la base de données. Les commandes utilisées sont : *SELECT*, *INSERT*, *DELETE*, *UPDATE* et *MERGE*.
- Une partie langage de contrôle des données (LCD) permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.
- Une partie langage de contrôle des transactions (LCT) est utilisée pour le contrôle transactionnel dans une base de données. Les commandes utilisées sont : *COMMIT*, *SAVEPOINT*, *ROLLBACK*, *SET TRANSACTION*.

II.11.4 L'outil phpMyAdmin :

PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, qu'ils soient gratuits ou payants, le proposent ce qui permet à l'utilisateur de ne pas avoir à l'installer [10].

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier (xxx.sql) et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

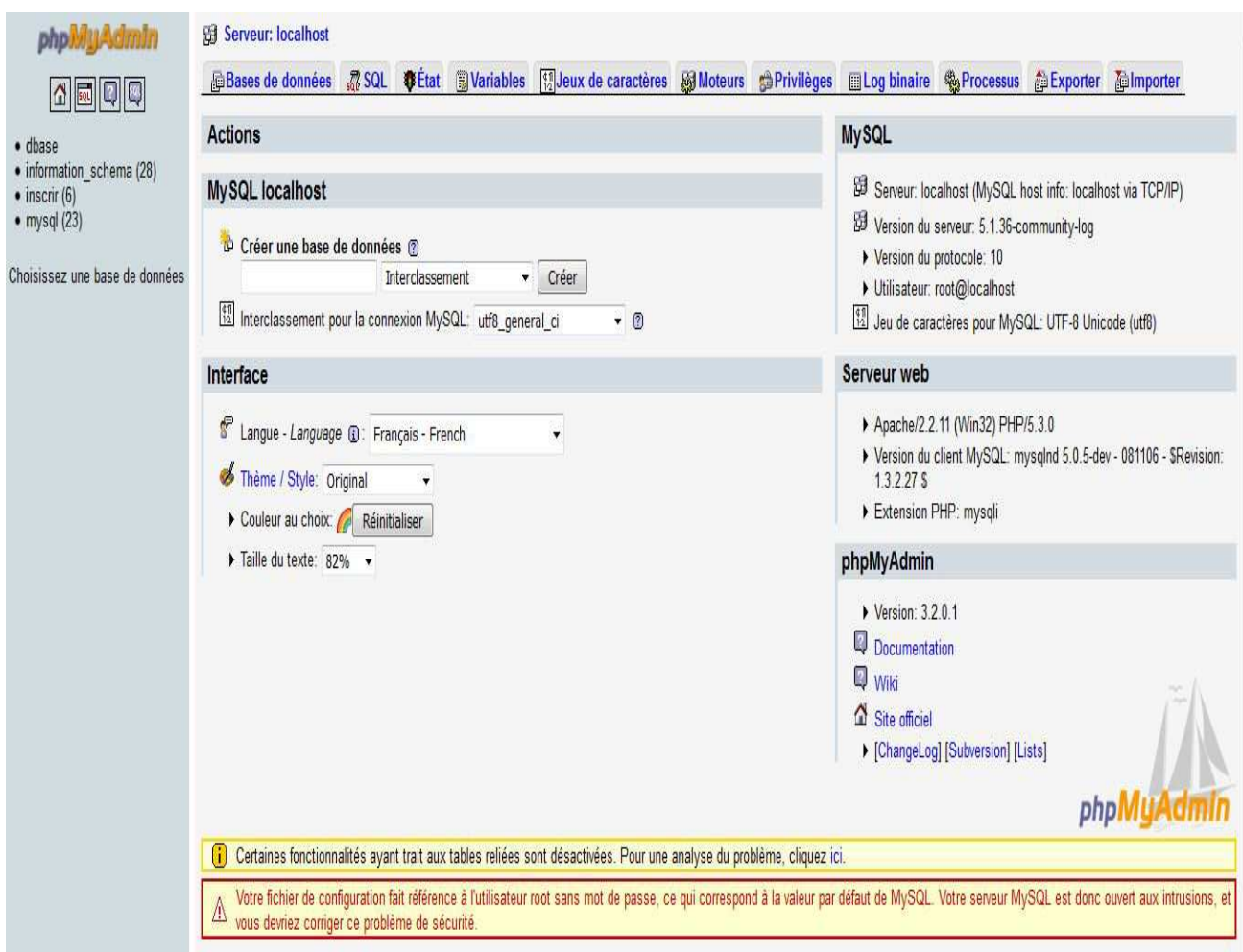


Figure II.11: L'interface phpMyAdmin.

II.12 Le langage PHP :

PHP (*HypertextPreprocessor*) est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

Il est utilisé sur tous les systèmes d'exploitation tels Windows, Gnu Linux ou autre Unix commercial, ce qui en fait un langage très portatif. PHP a récemment acquis une place incontournable dans le développement Web *Open Source*. Sa popularité vient de sa syntaxe, proche du C, de sa vitesse et de sa simplicité. On estime qu'il y a 4,8 millions de serveurs qui utilisent le langage.

Le PHP est un langage de script, comme le HTML et le JavaScript. Il s'insère directement entre des balises. Vous pouvez mélanger du PHP et du HTML sans problème.

```
<body>
< ?
echo " Bonjour tout le monde! " ;
?>
</body>
```

II.13 La connectivité d'une application android au serveur:

II.13.1 Format JSON [11] :

JSON (*JavaScript Object Notation*) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générable par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple : C lui-même, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal.

II.13.2 La syntaxe :

JSON se base sur deux structures :

- Une collection de couples nom/valeur. Divers langages la réifient par un *objet*, un enregistrement, une structure, un dictionnaire, une table de hachage, une liste typée ou un tableau associatif.
- Une liste de valeurs ordonnées. La plupart des langages la réifient par un *tableau*, un vecteur, une liste ou une suite.

Ces structures de données sont universelles. Pratiquement tous les langages de programmation modernes les proposent sous une forme ou une autre. Il est raisonnable qu'un format de données interchangeable avec des langages de programmation se base aussi sur ces structures.

En JSON, elles prennent les formes suivantes :

- Un *objet*, qui est un ensemble de couples nom/valeur non ordonnés. Un objet commence par { (accolade gauche) et se termine par } (accolade droite). Chaque nom est suivi de : (deux-points) et les couples nom/valeur sont séparés par, (virgule).
- Un *tableau* est une collection de valeurs ordonnées. Un tableau commence par [(crochet gauche) et se termine par] (crochet droit). Les valeurs sont séparées par, (virgule).
- Une *valeur* peut être soit une *chaîne de caractères* entre guillemets, soit un *nombre*, soit *true* ou *false* ou *null*, soit un *objet* soit un *tableau*. Ces structures peuvent être imbriquées.
- Une *chaîne de caractères* est une suite de zéro ou plus de caractères Unicode, entre guillemets, et utilisant les échappements avec antislash. Un caractère est représenté par une chaîne d'un seul caractère.

II.13.3 Avantages et inconvénients :

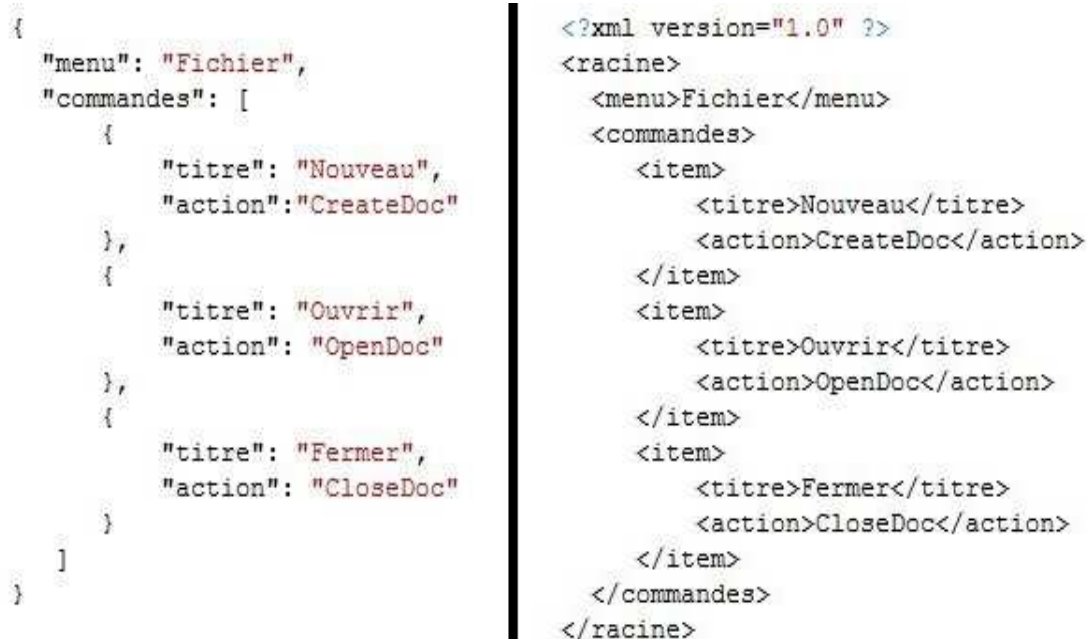
Le format JSON ne présente pas beaucoup de défauts, principalement parce qu'il reste vraiment très simple à lire, comprendre et utiliser.

Ses avantages :

- Format compréhensible par tous (humain et machine). Aucun apprentissage n'est requis puisque la syntaxe n'utilise que quelques marques de ponctuations.
- Ne dépend d'aucun langage (format d'échange de données ouvert). Comme ce format est très ouvert, il est pris en charge par de nombreux langages : JavaScript, PHP, Perl, Python, Ruby, Java,...
- Permet de stocker des données de différents types : chaînes de caractères (y compris des images en base64), nombres, tableaux (array), objets, booléens (true, false), la valeur null.
- Sa structure en arborescence et sa syntaxe simple lui permet de rester très "léger" et efficace.

Ses inconvénients:

- Comme pour tout moyen de stockage de données, il faudra prendre des mesures de sécurité pour garantir la sécurité des données sensibles.
- Le fait que la syntaxe soit rudimentaire peut être un inconvénient dans certains cas. Par exemple, contrairement à XML, il n'y a pas d'identification précise des données (sous forme de balise par exemple), la structure doit donc être connue avant utilisation.



```
{
  "menu": "Fichier",
  "commandes": [
    {
      "titre": "Nouveau",
      "action": "CreateDoc"
    },
    {
      "titre": "Ouvrir",
      "action": "OpenDoc"
    },
    {
      "titre": "Fermer",
      "action": "CloseDoc"
    }
  ]
}
```

```
<?xml version="1.0" ?>
<racine>
  <menu>Fichier</menu>
  <commandes>
    <item>
      <titre>Nouveau</titre>
      <action>CreateDoc</action>
    </item>
    <item>
      <titre>Ouvrir</titre>
      <action>OpenDoc</action>
    </item>
    <item>
      <titre>Fermer</titre>
      <action>CloseDoc</action>
    </item>
  </commandes>
</racine>
```

Figure II.12 : Exemple simple, définition d'un menu JSON vs XML.

II.13.4 Utilisation de format JSON:

Le fichier permet de charger de l'information stockée dans ce format à partir du serveur ou de transmettre au serveur de l'information dans un fichier de ce format, par exemple, le contenu d'un formulaire qui vient d'être rempli. Il y a donc trois aspects: le traitement par le navigateur, par le serveur, et la transmission des données entre les deux.

II.13.4.1 Coté client :

C'est particulièrement simple JSON faisant partie de la norme JavaScript. Le contenu d'un fichier JSON, ou la définition de données dans ce format sont assignés à une variable, laquelle devient un objet du programme.

II.13.4.2 Coté serveur :

Les fichiers au format JSON s'utilisent dans différents langages de programmation, notamment PHP et Java grâce à des parseurs qui permettent d'accéder au contenu, et éventuellement de le convertir en classes et attributs, dans ce langage.

II.13.4.3 L'échange de données :

La méthode la plus répandue de se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec PHP. Donc la façon la plus simple et le plus évident d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP du système Android.

Les données sont codées dans le format JSON, afin de Communiquer les données entre PHP et Android, en exploitant les options faciles à utiliser.

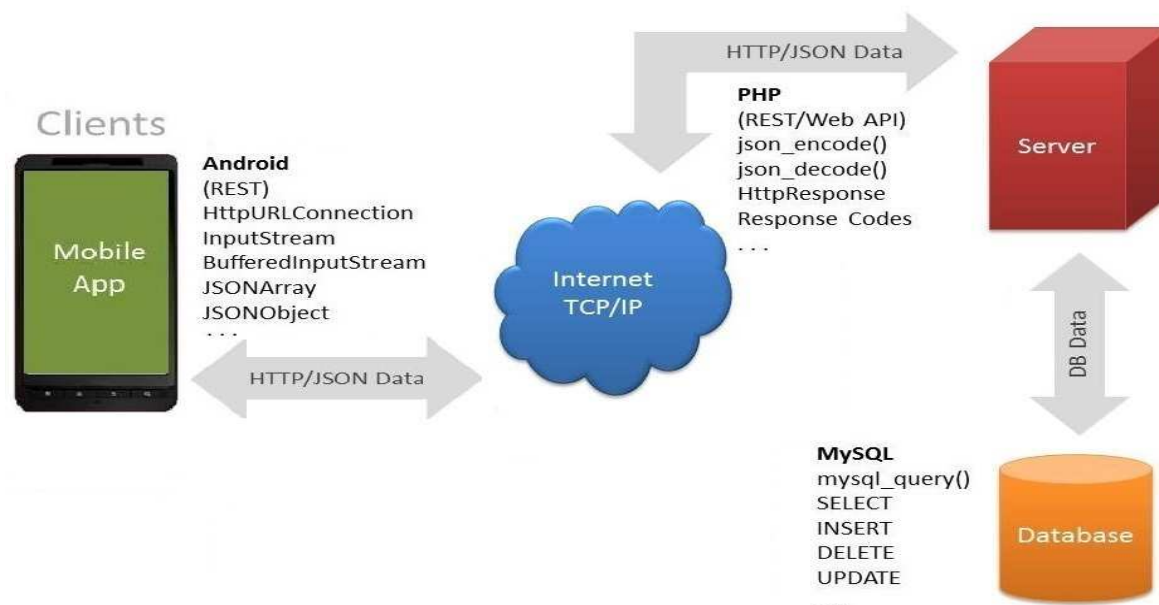


Figure II.13: Schéma d'échange de données entre un client Android et un serveur MySQL.

II.14 Conclusion :

Nous arrivons à présent à la fin de ce deuxième chapitre dans lequel nous avons abordé quelques généralités sur les réseaux et leur fonctionnement, puis nous avons parlé de modèle Client/Serveur afin de bien comprendre le choix de son utilisation.

Le chapitre suivant sera consacré à l'analyse et la conception de notre application.

CHAPITRE III

ANALYSE

ET

CONCEPTION



III.1 Introduction :

Une bonne méthodologie de réalisation de logiciel suppose une bonne maîtrise de la distinction entre l'analyse et la conception. Le lecteur verra qu'en pratique, le respect d'une distinction entre ces deux phases rigoureusement indépendantes n'est pas tenable, mais il est important d'avoir en tête la différence lorsqu'on s'apprête à réaliser un logiciel. Du point de vue des notations employées en UML (le langage sur lequel on va s'appuyer pour réaliser les différents diagrammes), les différences entre l'analyse et la conception se traduisent avant tout par des différences de niveau de détail dans les diagrammes utilisés.

III.2 Présentation générale d'UML :

III.2.1 Définition :

UML (Unified Modeling Language) [IF811], veut dire « langage de modélisation unifié ». Ce langage est né de la fusion de plusieurs méthodes existantes auparavant. Il est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet. Il permet de modéliser toutes les parties de développement d'une application informatique, de sa conception à la mise en route, tout ceci grâce à des diagrammes. Il est vrai que certains de ces diagrammes sont plus adaptés à des informaticiens, mais il en existe ceux qui permettent de voir comment interagit l'application dans son contexte de fonctionnement.

III.2.2 Différentes vues et diagrammes d'UML :

Le métamodèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relient.

Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux **vues**.

Une vue est constituée d'un ou plusieurs **diagrammes**. On distingue deux types de vues:

III.2.2.1 Les vues statiques ou structurelles (Structural Diagrams) : c'est-à-dire représentant le système physiquement.

- **diagrammes d'objets :** modélisent des exemples de classes. Ce type de diagramme est employé pour décrire le système à un instant particulier. En utilisant cette technique, nous pouvons valider le diagramme des classes et ses règles de multiplicité avec des données et des scénarios réels. D'un point de vue de la notation, les diagrammes des objets empruntent des éléments du diagramme des classes.
- **diagrammes de classes :** représentent l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage, marqué par une flèche terminée par un triangle) ou une relation organique (agrégation, marquée par une flèche terminée par un diamant).
- **diagrammes de cas d'utilisation :** identifient les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. Les cas d'utilisation sont utilisés dans la phase d'analyse pour définir les besoins de "haut niveau" du système.
- **diagrammes de composants :** décrivent l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), les données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ces diagrammes permettent de mettre en évidence les dépendances entre les composants (*qui utilisent quoi*).
- **diagrammes de déploiement :** est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés sont principalement les nœuds (ensemble d'éléments matériels du système), les composants (entité logicielle du système), les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

III.2.2.2 Les vues dynamiques ou comportementales (Behavior Diagram) : montrant le fonctionnement du système.

- **diagrammes de séquences :** documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, tel qu'un cas d'utilisation. UML étant conçu pour la programmation orientée objet, les communications entre les classes sont reconnues comme des messages. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.
- **diagrammes de collaboration :** comme les autres diagrammes comportementaux, les diagrammes de collaboration modélisent les interactions entre les objets. Ce type de diagramme est un croisement entre un diagramme des objets et un diagramme des séquences. À la différence du diagramme des séquences qui modélise l'interaction dans un format de type ligne-colonne, le diagramme des collaborations emploie une disposition libre des objets tels qu'on les trouvent dans un diagramme des objets. Ceci facilite la vision de toutes les interactions impliquant un objet particulier.
- **diagrammes d'états-transitions :** les diagrammes d'états transitions sont utilisés pour documenter les divers modes ("états") qu'une classe peut prendre, et les événements qui causent une transition d'état. En comparaison avec les autres diagrammes comportementaux qui modélisent les interactions entre des classes multiples, les diagrammes d'états eux modélisent typiquement les transitions d'une seule classe.
- **diagrammes d'activités :** sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). En regardant un diagramme d'activité, vous trouverez des éléments des diagrammes d'état. En fait, le diagramme d'activités est une variante du diagramme d'états transitions où les "états" représentent des opérations, et les transitions représentent les activités qui se produisent quand l'opération est terminée. L'usage général des diagrammes d'activités permet de faire apparaître les flots de traitements induits par les processus internes par rapport aux événements externes.

III.3 La modélisation avec UML :

La modélisation consiste à créer une représentation simplifiée d'un problème: **le modèle**. Grâce au modèle il est possible de représenter simplement un problème, un concept et le simuler. La modélisation comporte deux composantes :

- L'analyse, c'est-à-dire l'étude du problème.
- La conception, soit la mise au point d'une solution au problème.

Le modèle constitue ainsi une représentation possible du système pour un point de vue donné.

Les étapes suivies pour la modélisation de notre application sont :

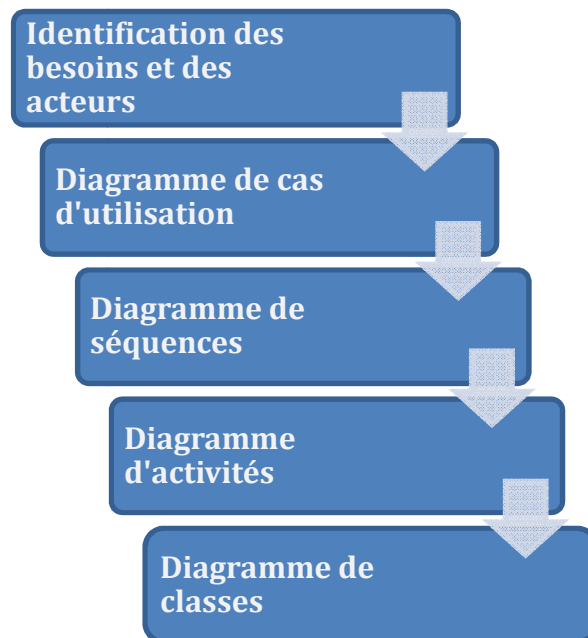


Figure III.1 : Les différentes étapes de la modélisation UML suivies pour notre conception.

III.3.1 Analyse :

Le but de l'analyse est de traduire dans un langage proche de celui des informaticiens les modèles exprimés dans l'expression des besoins. Cependant pour rester compréhensible par les clients ou utilisateurs, il n'intervient que des entités du domaine (métier) considéré. Elle sert d'interface, avec l'expression des besoins, aux dialogues et aux contrats avec les clients et les utilisateurs. L'analyse doit servir de support pour la conception, l'implémentation et la maintenance.

III.3.2 Identification des besoins :

Notre application « Algérie Ventes et Achats » permet à un utilisateur de Smartphone utilisant Android comme système d'exploitation, de déposer une annonce et de consulter les détails de différentes annonces qui sont déposés par d'autres utilisateurs après avoir créé un compte, de contacter ces derniers à propos d'un produit avec un appel téléphonique ou un message électronique afin de réaliser l'achat et satisfaire le besoin.

La liste des fonctionnalités de l'application est la suivante :

- Accueil pour avoir un aperçu sur toutes les annonces ;
- Inscription et création d'un compte utilisateur ;
- Connexion au compte utilisateur avec authentification et qui permet :
 - ✓ De déposer une annonce ;
 - ✓ De consulter les annonces déjà déposées pour les modifier ;
 - ✓ De consulter la liste favoris des annonces ;
 - ✓ De consulter la messagerie ;
 - ✓ De voir les détails de chaque annonce dans l'Accueil.
- Faire une recherche d'une annonce selon la catégorie, le prix et l'endroit ;
- Choisir l'annonce qui vous intéresse ;
- Voir les détails de l'annonce et contacter la personne qui l'a déposée avec un appel téléphonique ou bien un message électronique ;
- Ajouter l'annonce à vos Favoris.

III.3.3 Identification des acteurs :

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Les acteurs fournissent de l'information en entrée et reçoivent de l'information en sortie. Les acteurs peuvent être classés selon une hiérarchie (Acteurs principaux, secondaires, ...).

- **Visiteur:** cet acteur peut lancer l'application, pour voir la liste des annonces sans détails et d'effectuer une recherche selon des critères bien précis, d'inscrire pour créer un compte client qui lui permet de se connecter. Enfin consulter l'aide à propos de l'application.
- **Client:** cet acteur doit lancer l'application, pour accéder à son compte qui lui permet de déposer des annonces (offre ou demande), de voir ses anciennes annonces pour les modifier s'il veut, de consulter la liste des favoris et la liste de messagerie électronique. Comme il peut lancer une recherche selon la catégorie, le prix et l'endroit, ainsi de

CHAPITRE III : ANALYSE ET CONCEPTION

sélectionner l'annonce qui l'intéresse et réaliser l'achat en contactant la personne qui a déposé l'annonce soit avec un appel téléphonique ou bien avec un message électronique, et comme il peut ajouter des annonces à la liste des favoris.

III.4 Spécification des tâches :

Il s'agit d'une suite cohérente d'opérations afin d'exécuter un programme. Une unité de tâches est un ensemble structuré d'actions (ou d'étapes de tâche) ayant un seul objectif et dont l'exécution et le contrôle relèvent d'un seul acteur; ces actions doivent être exécutées complètement ou annulées. L'ensemble des tâches effectuées par l'acteur défini préalablement sont représentées dans le tableau suivant :

Acteurs	Taches
Visiteur	T1 : Lancer l'application. T2 : Afficher la liste des annonces sans détails. T3 : Inscrire pour créer un compte client. T4 : Effectuer une recherche selon la catégorie, prix et l'endroit. T5 : Consulter l'aide à propos de l'application.
Client	T1 : Lancer l'application. T2 : Afficher la liste des annonces sans détails. T4 : Effectuer une recherche selon la catégorie, prix et l'endroit. T5 : Consulter l'aide à propos de l'application T6 : Connecter pour accéder au compte client. T7 : Afficher les détails d'une annonce. T8 : Déposer une nouvelle annonce. T9 : Afficher mes annonces. T10 : Modifier une annonce. T11 : Supprimer une annonce. T12 : Afficher mes favoris. T13: Afficher mes messages. T14 : Ajouter l'annonce à mes favoris. T15 : Passer un appel téléphonique à propos d'une annonce. T16 : Envoyer un message à propos d'une annonce.

Tableau III.1 : Spécification des tâches.

III.5 Spécification des scénarios :

Un scénario est une suite d'échanges d'événements entre les acteurs et le système. Séquence d'événements survenant durant une exécution particulière d'un ensemble d'objets constituant une application.

Acteur	Taches	Scénarios
Visiteur	T1 : Lancer l'application.	S1 : Cliquer sur l'icône de l'application.
	T2 : Afficher les annonces sans détails.	S2 : une fois le visiteur a lancé l'application une liste d'annonces affichée automatiquement.
	T3 : Inscrire pour créer un compte client.	S3 : Cliquer sur l'item « Connexion » de menu puis cliquer sur le bouton « Inscription », remplir les champs et cliquer sur le bouton « valider ».
	T4 : Effectuer une recherche selon la catégorie, prix et l'endroit.	S4 : Cliquer sur l'item « Recherche » de menu et choisir les critères de recherche.
	T5 : Consulter l'aide à propos de l'application.	S5 : Cliquer sur l'item « Aide » de menu.
Client	T6 : Connecter pour accéder au compte client.	S6 : Cliquer sur l'item «Connexion» de menu, une nouvelle interface qui s'affiche pour s'authentifier, à la fin cliquer sur le bouton « OK »
	T7 : Afficher les détails d'une annonce.	S7 : une fois que le client a accédé à son compte il doit sélectionner un item dans la liste des annonces de l'Accueil.
	T8 : Déposer une nouvelle annonce.	S8 : une fois que le client a accédé à son compte il doit cliquer sur l'item «Déposer Annonce » de menu et remplir tout les champs puis valider.
	T9 : Afficher mes annonces.	S9 : une fois que le client a accédé à son compte il doit cliquer sur l'item « Mes annonces » de menu.

CHAPITRE III : ANALYSE ET CONCEPTION

	T10 : Modifier une annonce.	S10 : une fois que le client a accédé à ses annonces il doit cliquer sur le bouton « modifier » dans l'interface enfin modifier les champs et valider.
	T11 : Supprimer une annonce.	S11 : une fois que le client a accédé à ses annonces il doit cliquer sur « supprimer » dans l'interface.
	T12 : Afficher mes favoris.	S12 : une fois que le client a accédé à son compte il doit cliquer sur l'item «Favoris » de menu.
	T13 : Afficher mes messages.	S13 : une fois que le client a accédé à son compte il doit cliquer sur l'item «Mes messages » de menu.
	T14 : Ajouter une annonce à mes favoris.	S14 : une fois que le client a accédé à son compte il doit sélectionner un item dans la liste des annonces de l'Accueil pour voir les détails enfin cliquer sur le bouton «j'aime».
	T15 : Passer un appel téléphonique à propos d'une annonce.	S15 : une fois que le client a accédé à son compte il doit sélectionner un item dans la liste des annonces de l'Accueil pour voir les détails puis cliquer sur le bouton «Appel».
	T16 : Envoyer un message à propos d'une annonce.	S16 : une fois que le client a accédé à son compte il doit sélectionner un item dans la liste des annonces de l'Accueil pour voir les détails puis cliquer sur le bouton «Envoyer message», saisir le texte et valider.

Tableau III.2 : Spécification des scenarios.

III.6 Les cas d'utilisation :

Un cas d'utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier [12]. Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Cas d'utilisation : Inscrire pour créer un compte client.

Scénarios: S1, S3.

Rôles : visiteur.

Description :

- 1-le visiteur lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le visiteur clique sur l'item « Connexion » de menu;
- 4-le système affiche l'interface de Connexion ;
- 5- le visiteur clique sur le bouton « Inscription» ;
- 6-le système affiche l'interface d'Inscription avec un formulaire d'inscription ;
- 7- le visiteur remplit le formulaire et clique sur le bouton « valider ».

Tableau III.3 : Cas d'utilisation « Inscrire pour créer un compte client ».

Cas d'utilisation : Effectue une recherche selon le prix, catégorie et l'endroit.

Scénarios : S1, S4.

Rôles : visiteur.

Description :

- 1-le visiteur lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le visiteur clique sur l'item « Recherche » de menu ;
- 4-le système affiche une interface Recherche ;
- 5- le visiteur va choisir la catégorie, le prix et l'endroit;
- 6-le système affiche les annonces selon les critères choisis.

Tableau III.4 : Cas d'utilisation «Effectuer une recherche selon le prix, catégorie et l'endroit».

Cas d'utilisation : Connecter pour accéder au compte client.

Scénarios : S1, S6.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche l'interface de connexion avec un formulaire pour s'authentifier ;
- 5- le client remplit le formulaire et clique sur le bouton «Ok » ;
- 6-le système vous permet l'accès à votre compte.

Tableau III.5:Cas d'utilisation « Connecter pour accéder au compte client ».

Cas d'utilisation : Afficher les détails d'une annonce.

Scenarios: S1, S6, S7.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte ;
- 7- le client clique sur l'item « Accueil» de menu;
- 8- le système affiche les annonces sous forme d'une listview ;
- 9- le client clique sur un item pour sélectionner une annonce;
- 10- le système affiche les détails.

Tableau III.6 : Cas d'utilisation « Afficher les détails d'une annonce ».

Cas d'utilisation : Déposer une nouvelle annonce.

Scénarios : S1, S6, S8.

Rôles : client

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6- le système affiche l'interface « Mon Compte ».
- 7- le client clique sur l'item « Déposer Annonce » de menu;
- 8-le système affiche l'interface « Déposer Annonce ».
- 9- le client remplit le formulaire d'annonce selon la catégorie puis clique sur le bouton « valider »
- 10-le système ajoute l'annonce à la base de données.

Tableau III.7 : Cas d'utilisation « Déposer une nouvelle annonce ».

Cas d'utilisation : Afficher mes annonces.

Scénarios : S1, S6, S9.

Rôles : client.

Description :

- 1- le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le system affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte.
- 7- le client clique sur l'item « Mes annonces » de menu;
- 6-le système affiche l'interface Mes Annonce sous forme d une listview.

Tableau III.8 : Cas d'utilisation « Afficher mes annonces ».

Cas d'utilisation : Modifier une annonce.

Scénarios: S1, S6, S7, S10.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte.
- 7- le client clique sur l'item « Accueil» de menu;
- 8- le système affiche les annonces sous forme d'une listview ;
- 9- le client clique sur un item pour sélectionner une annonce;
- 10- le système affiche les détails.
- 11- le client clique sur le bouton « modifier» de l'interface;
- 12-le système affiche l'interface « Modifier l'annonce » ;
- 13- le client va modifier les champs et valider;
- 14-le système affiche un message de succès.

Tableau III.9 : Cas d'utilisation « Modifier une annonce ».

Cas d'utilisation : Supprimer une annonce.

Scénarios: S1, S6, S7, S11.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte.
- 7- le client clique sur l'item « Accueil» de menu;
- 8- le système affiche les annonces sous forme d'une listview ;
- 9- le client clique sur un item pour sélectionner une annonce;
- 10- le système affiche les détails.

- 11- le client clique sur le bouton « supprimer» de l'interface;
- 12-le système affiche une boîte de dialogue pour confirmer la suppression ;
- 13- le client clique sur le bouton « oui »;
- 14-le système supprime l'annonce.

Tableau III.10 : Cas d'utilisation « Supprimer une annonce ».

Cas d'utilisation : Afficher mes favoris.

Scénarios : S1, S6, S12.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3-le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte.
- 7-le client clique sur l'indicateur «Favoris » de menu;
- 6-le système affiche l'interface Favoris sous forme d une listview.

Tableau III.11 : Cas d'utilisation « Afficher mes favoris ».

Cas d'utilisation : Afficher mes messages.

Scénarios : S1, S6, S13.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Connexion » de menu ;
- 4-le système affiche une interface pour s'authentifier ;
- 5- le client va s'authentifier pour accéder à son compte;
- 6-le système affiche l'interface mon compte.
- 7- le client clique sur l'item «Mes messages» de menu;
- 6-le système affiche les messages sous forme d'une listview.

Tableau III.12 : Cas d'utilisation « Afficher mes messages ».

Cas d'utilisation : Ajouter une annonce à mes favoris.

Scénarios: S1, S6, S7, S14.

Rôles : client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Accueil » de menu ;
- 4-le système affiche une listview qui contient des annonces ;
- 5- le client clique sur l'item d'une listview ;
- 6-le système affiche les détails sur l'annonce sélectionnée.
- 7- le client clique sur le bouton « sauvegarder » ;
- 8-le système ajoute l'annonce dans la liste favoris.

Tableau III.13 : Cas d'utilisation « Ajouter une annonce à mes favoris ».

Cas d'utilisation : Envoyer un message à propos d'une annonce.

Scénarios: S1, S6, S7, S15.

Rôles: client.

Description :

- 1-le client lance l'application en cliquant sur l'icône de l'application ;
- 2-le système affiche l'interface d'accueil ;
- 3- le client clique sur l'item « Accueil » de menu ;
- 4-le système affiche une listview qui contient des annonces ;
- 5- le client clique sur l'item d'une listview ;
- 6-le système affiche les détails sur l'annonce sélectionnée.
- 7- le client clique sur le bouton « Envoyer message » ;
- 8-le système affiche un EditText.
- 9- le client va saisir le texte et valider ;
- 10-le système envoie le message au destinataire.

Tableau III.14 : Cas d'utilisation « Envoyer un message à propos d'une annonce ».

III.7 La conception :

La conception est une étape préliminaire et primordiale qui doit précéder l'étape de développement de toute application informatique. Pour décrire la conception de l'application, on commencera avec les diagrammes de cas d'utilisation et les diagrammes de séquences. Par la suite on passera au diagramme de classes.

III.8 Le niveau applicatif :

III.8.1 Les cas d'utilisation :

Avant de se lancer dans la réalisation d'une application, il faut comprendre, clarifier et structurer les attentes et les besoins des clients. Pour cela, le diagramme des cas d'utilisation (en anglais *use cases*) constitue la première étape de l'analyse UML [13]. Ces cas d'utilisations permettent de représenter le fonctionnement du système vis-à-vis de l'utilisateur, c'est donc une vue du système dans son environnement extérieur. Chaque cas d'utilisation peut être complété par un ensemble d'interactions successives d'une entité en dehors du système (l'utilisateur) avec le système lui-même.

III.8.2 Les éléments d'un diagramme des cas d'utilisation:

III.8.2.1 Les acteurs :

Les acteurs se représentent sous la forme d'un petit personnage (**stick man**) ou sous la forme d'une case rectangulaire (appeler **classeur**) avec le mot clé « actor », chaque acteur porte un nom.

Un acteur est un utilisateur **externe** au système. Cela peut être :

- Une personne.
- Du matériel (capteurs, moteurs, relais...).
- Un autre système.



« Stick man »

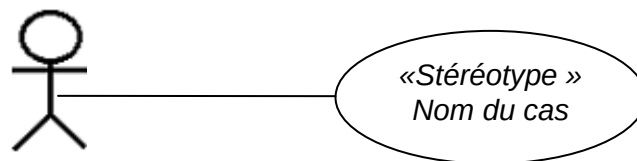
III.8.2.2 Les cas d'utilisation :

Un cas d'utilisation représente une fonctionnalité du système (**visible de l'extérieur du système**). Il se représente par une ellipse contenant le nom du cas d'utilisation (**phrase commençant par un verbe à l'infinitif**) et optionnellement un stéréotype au dessus du nom.

III.8.3 Relation entre acteurs et cas d'utilisation :

III.8.3.1 La relation d'association :

- A chaque acteur est associé un ou plusieurs cas d'utilisation, la relation d'association peut aussi être appelée relation de communication.
- Elle est représentée par un trait reliant l'acteur et le cas d'utilisation. Nous pouvons rajouter sur ce trait un stéréotype qui va préciser la relation de communication (« **communicate** »).



III.8.4 Les relations entre cas d'utilisation :

III.8.4.1 Relation d'inclusion :

La relation d'inclusion sert à enrichir un cas d'utilisation par un autre cas, cette relation est représentée par une flèche pointillée reliant les 2 cas d'utilisation et munie du stéréotype « **include** ». Elle précise qu'un cas d'utilisation contient le comportement défini dans un autre cas d'utilisation et met en commun des comportements communs à plusieurs cas d'utilisation.

III.8.4.2 Relation d'extension :

La relation d'extension enrichit un cas d'utilisation par un autre cas. Cette relation est représentée par une flèche en pointillée reliant le stéréotype « **extend** ». Elle précise qu'un cas d'utilisation peut, dans certains cas, augmenter le comportement d'un autre cas d'utilisation.

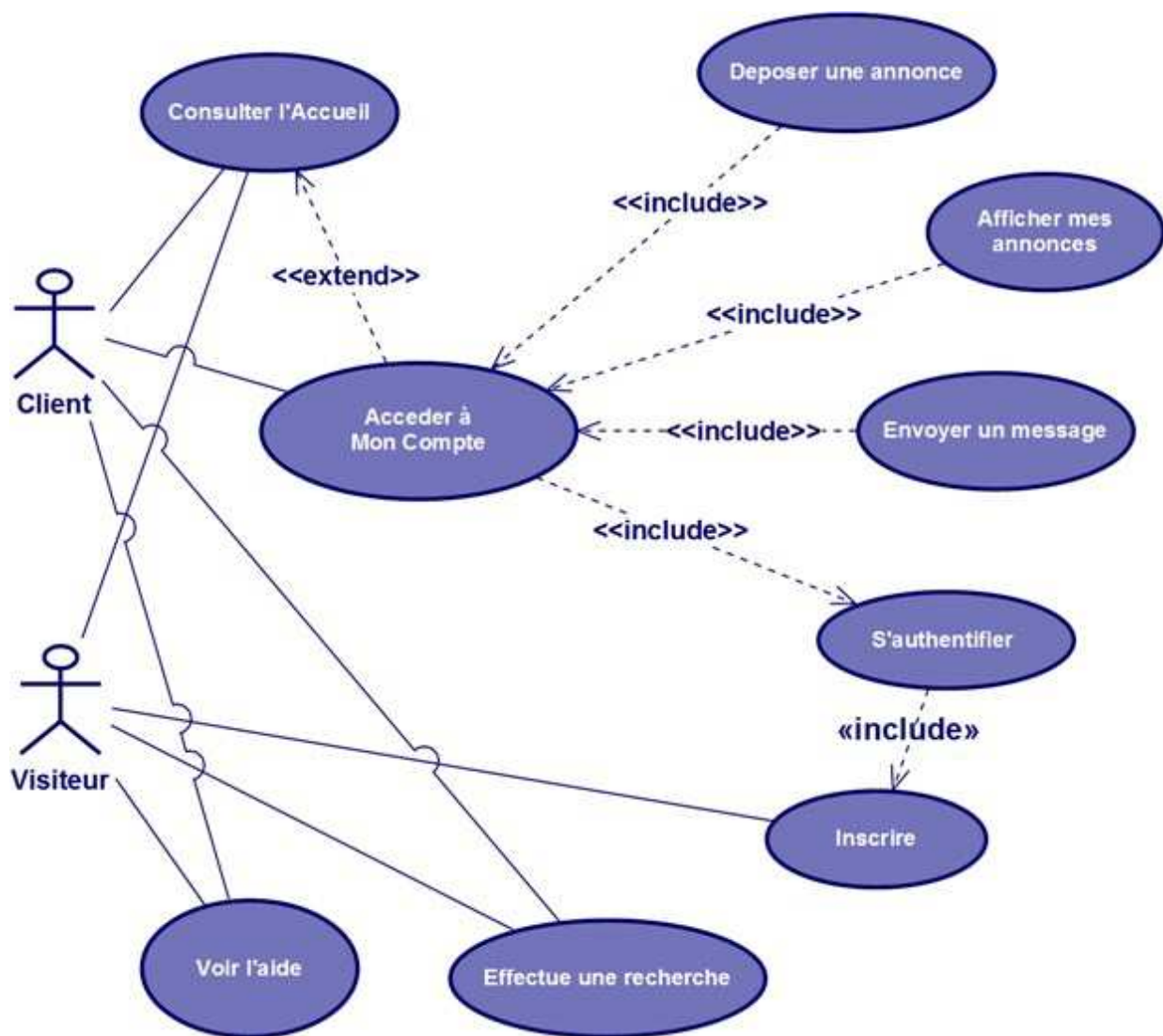


Figure III.2 : Diagramme de cas d'utilisation.

III.9 Diagramme de séquences :

L'objectif du diagramme de séquences est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

A moins que le système à modéliser soit extrêmement simple, nous ne pouvons pas modéliser la dynamique globale du système dans un seul diagramme. Nous ferons donc appel à un ensemble de diagrammes de séquences chacun correspondant à une sous fonction du système.

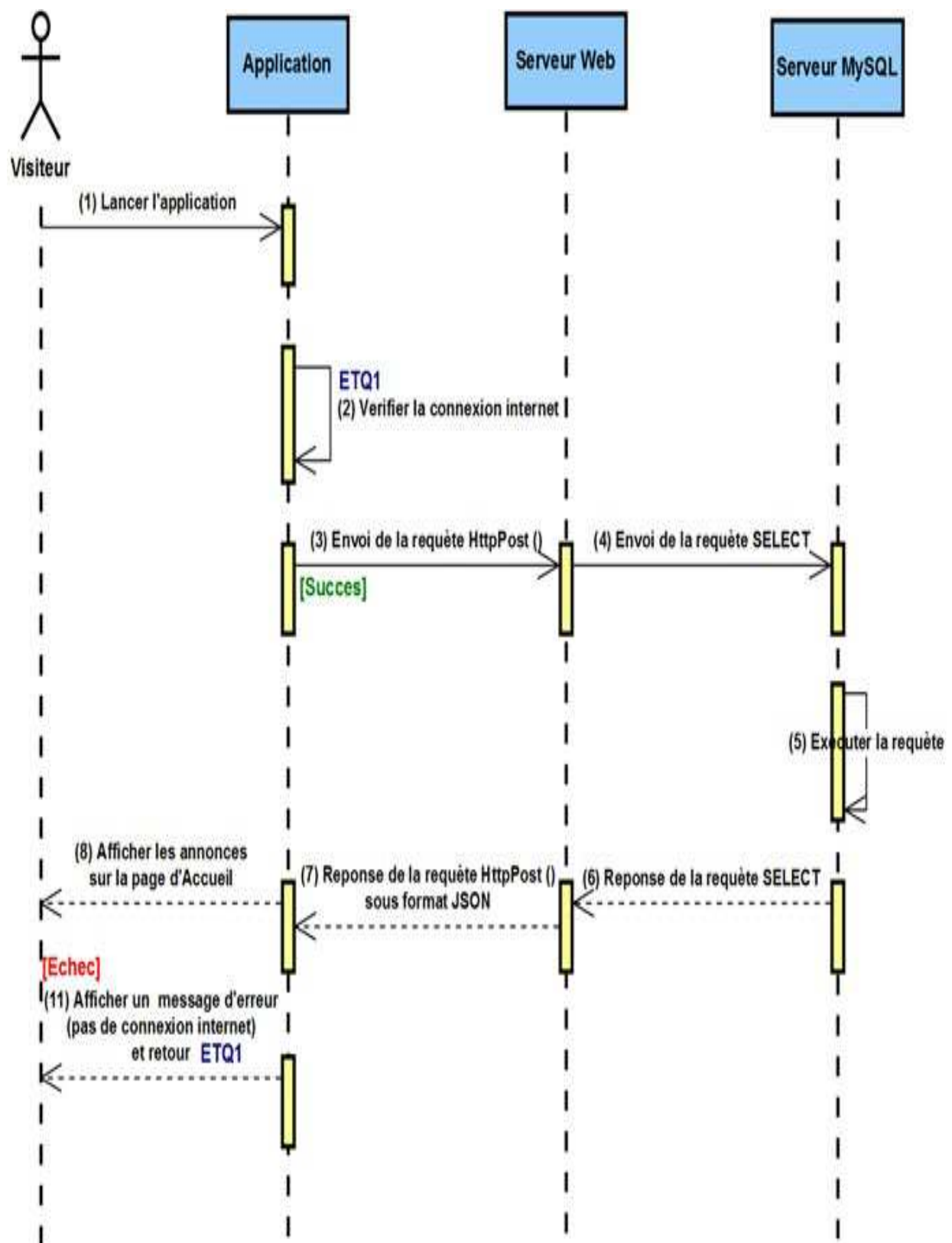


Figure III.3: Diagramme de séquence de cas d'utilisation « Lancer l'application ».

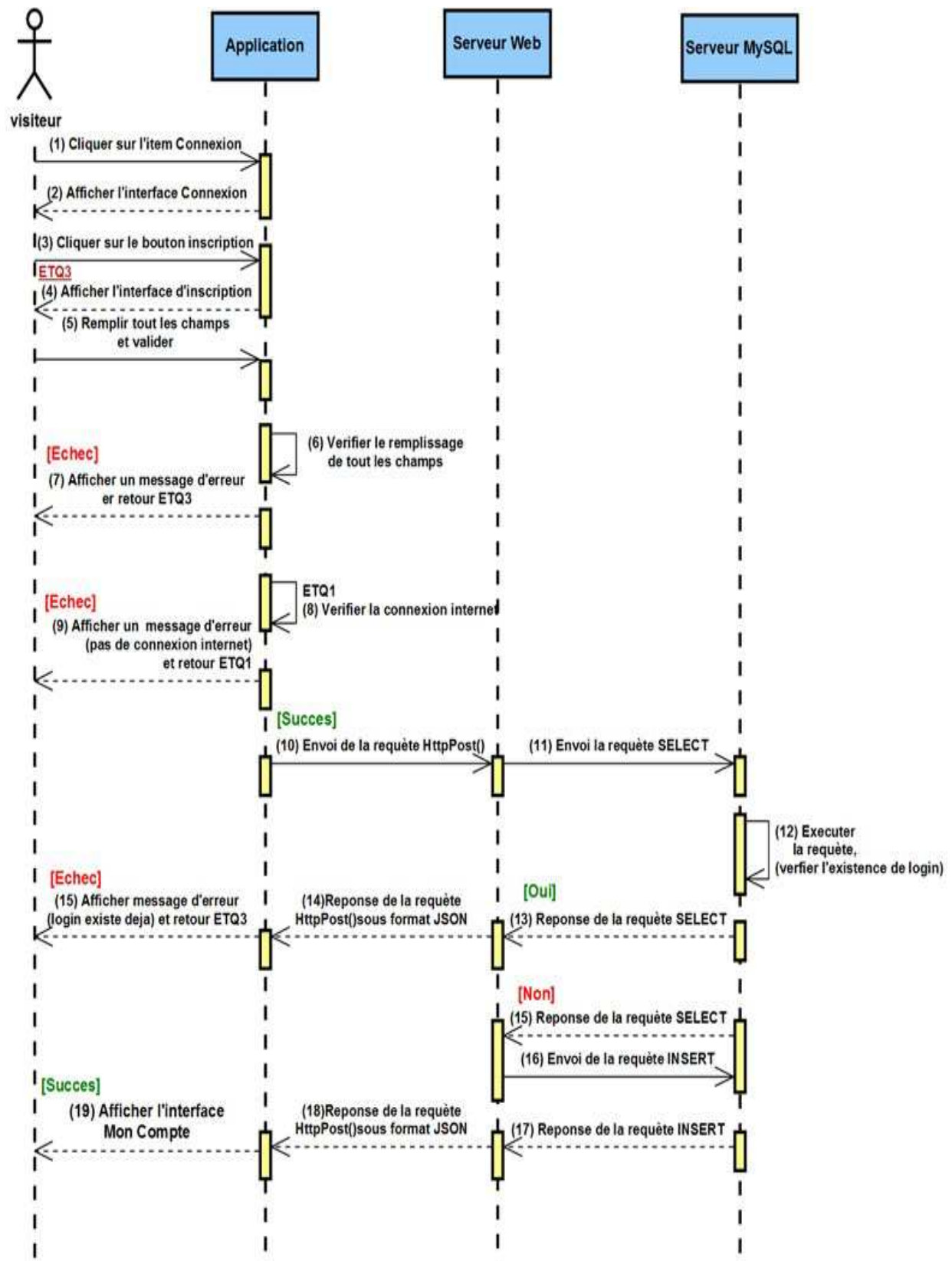


Figure III.4: Diagramme de séquence de cas d'utilisation « Inscription ».

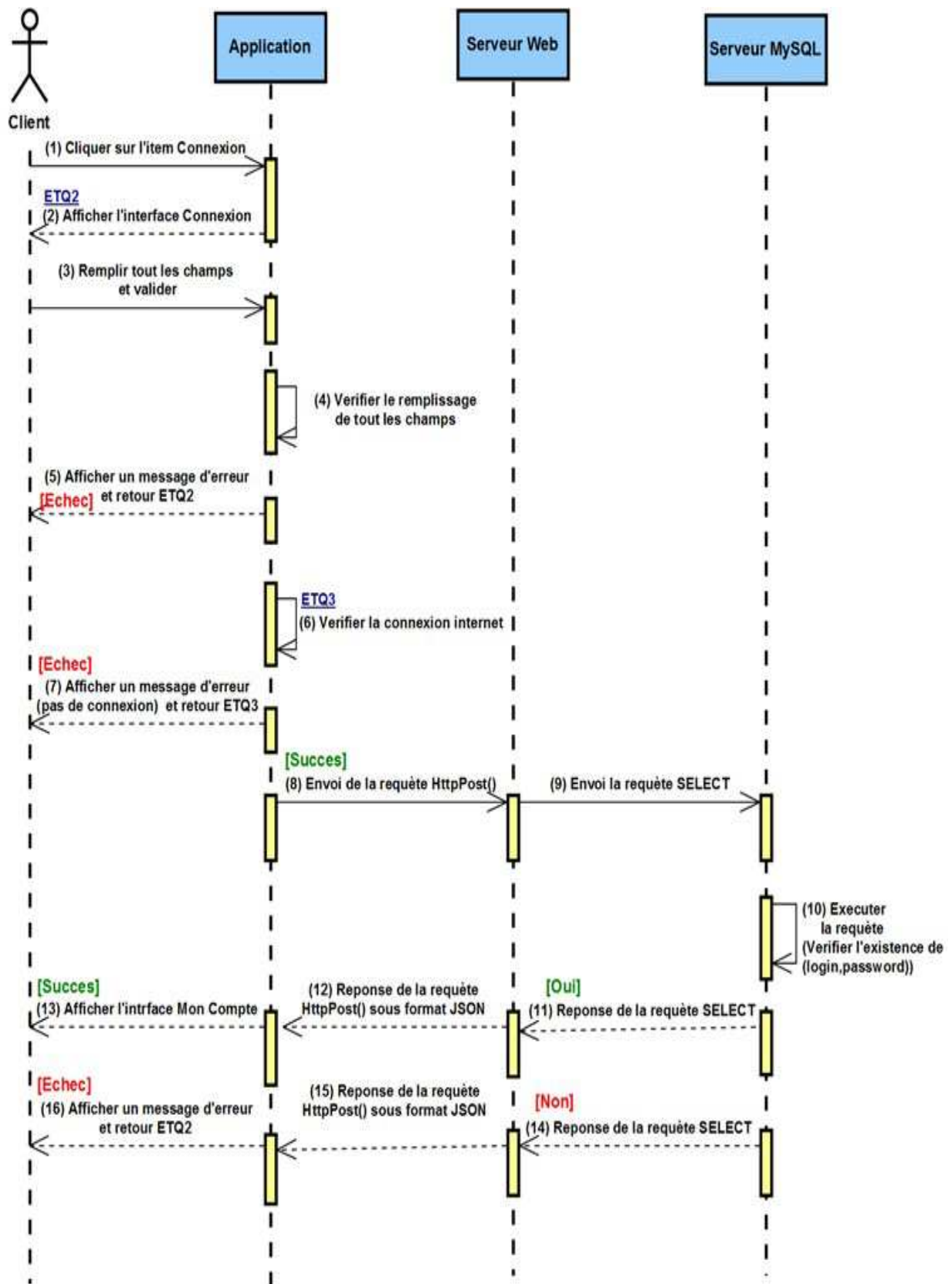


Figure III.5: Diagramme de séquence de cas d'utilisation « Connexion ».

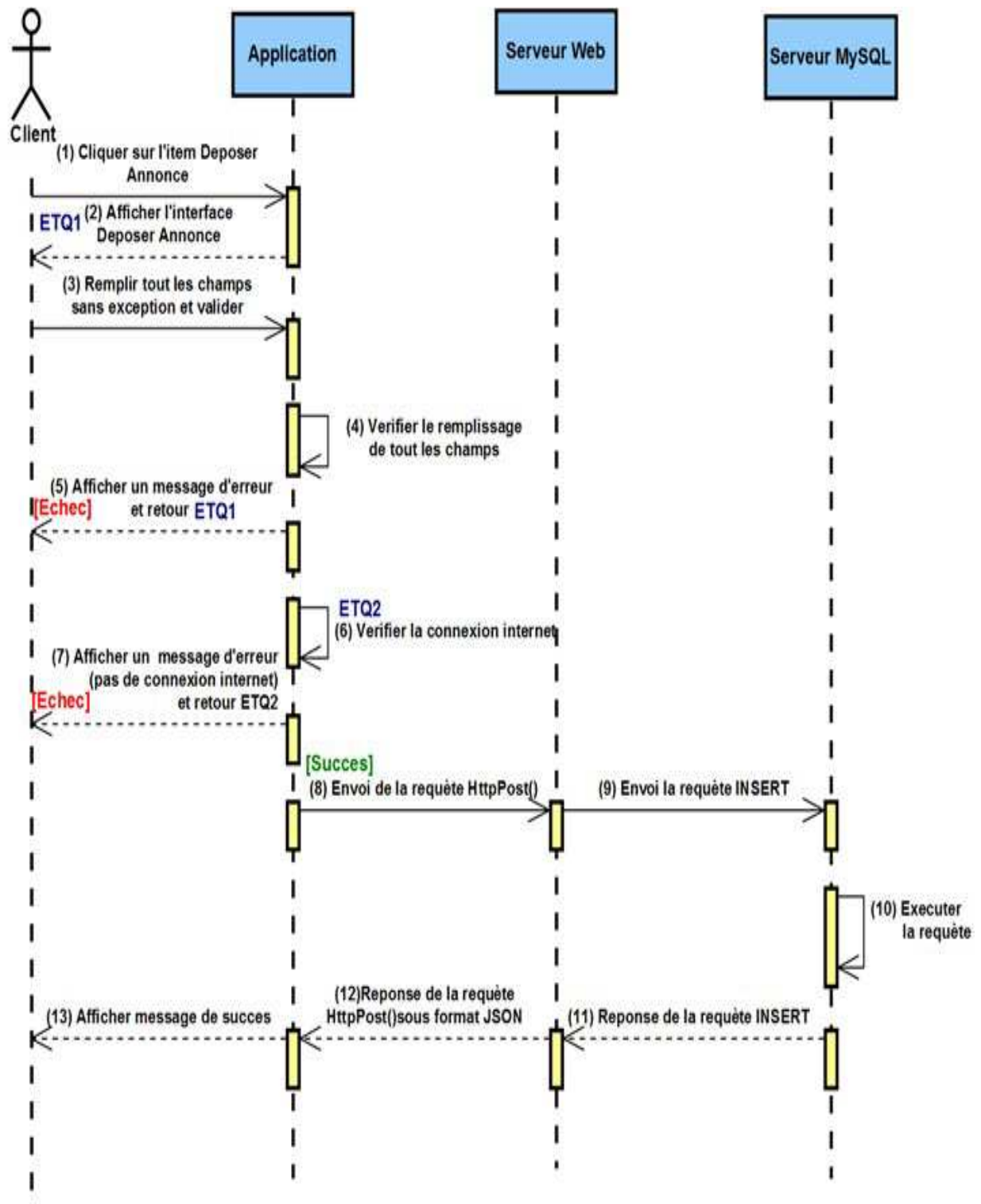


Figure III.6: Diagramme de séquence de cas d'utilisation « Déposer une annonce ».

III.10) Diagramme d'activités :

Les diagrammes d'activités sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). En consultant un diagramme d'activités, on trouvera des éléments des diagrammes d'états. En fait, le diagramme d'activités est une variante du diagramme d'états où les "états" représentent des opérations, et les transitions représentent les activités qui se produisent quand l'opération est terminée. L'usage général des diagrammes d'activités permet de faire apparaître les flots de traitement induits par les processus internes par rapport aux événements externes.

III.10.1) Représentation :

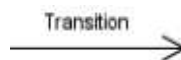
Etat d'activité :

L'état d'activité marque une action faite par un objet. Il est représenté par un rectangle arrondi.



Transition :

Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage. Les transitions sont modélisées par des flèches.



État initial :

L'état initial marque le point d'entrée "la première activité". Il est représenté, comme dans le diagramme d'état, par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.



Etat final :

L'état final marque la fin du déroulement des opérations modélisées. Il peut y avoir des états finaux multiples sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle.



Les diagrammes d'activités dans notre application seront modélisés comme suit :

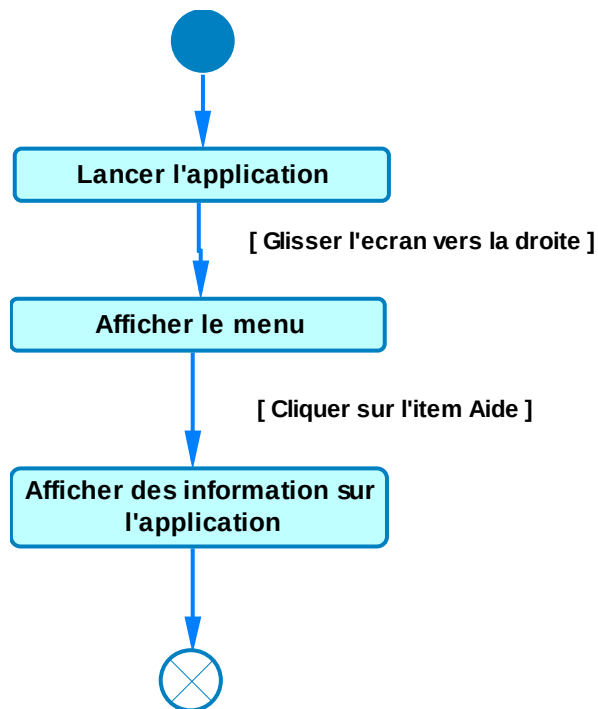


Figure III.7 : Diagramme d'activité du cas d'utilisation « Avoir une aide à propos l'application »

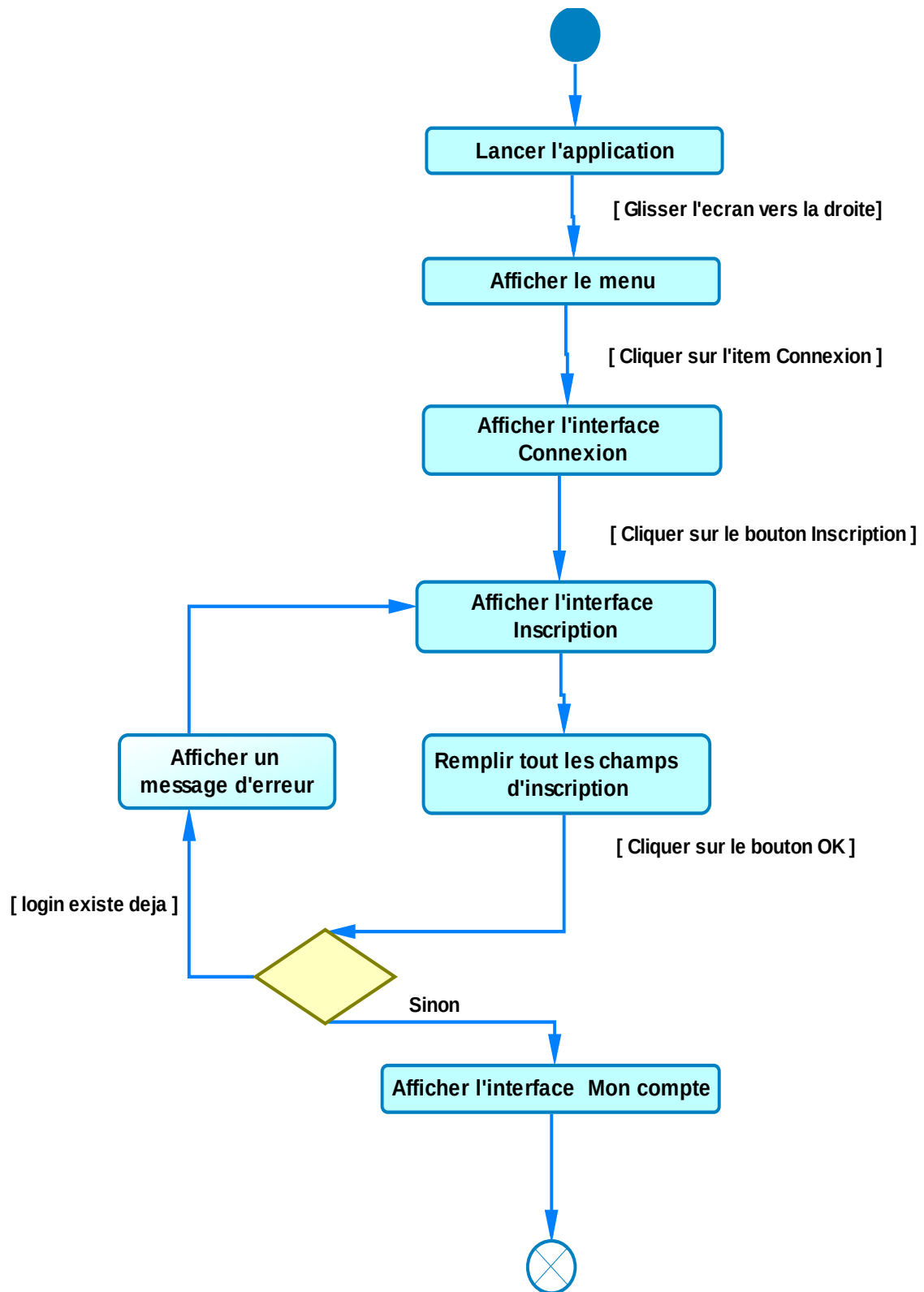


Figure III.8 : Diagramme d'activité du cas d'utilisation« Inscription »

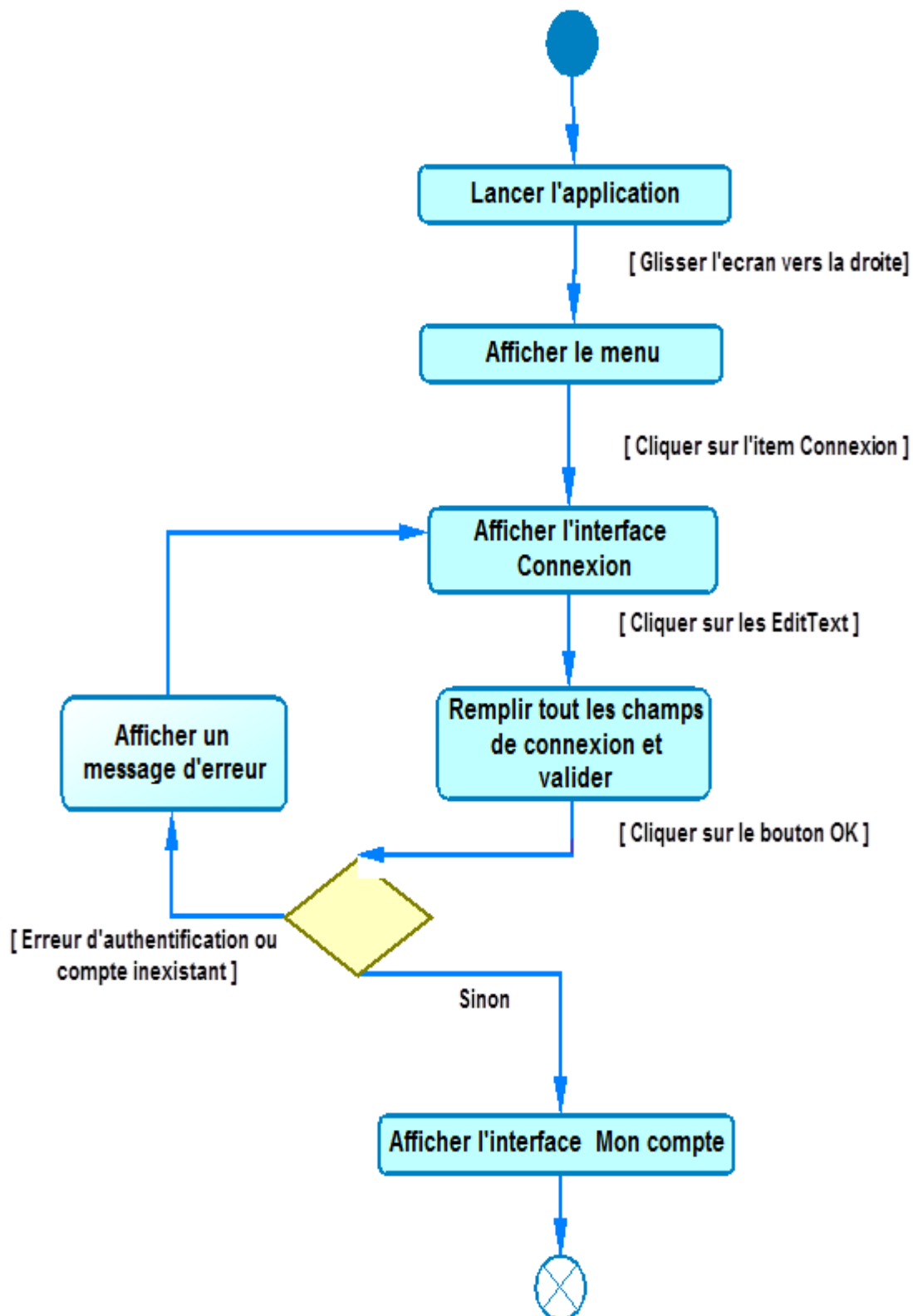


Figure III.9 : Diagramme d'activité du cas d'utilisation «Connexion pour accéder au compte utilisateur »

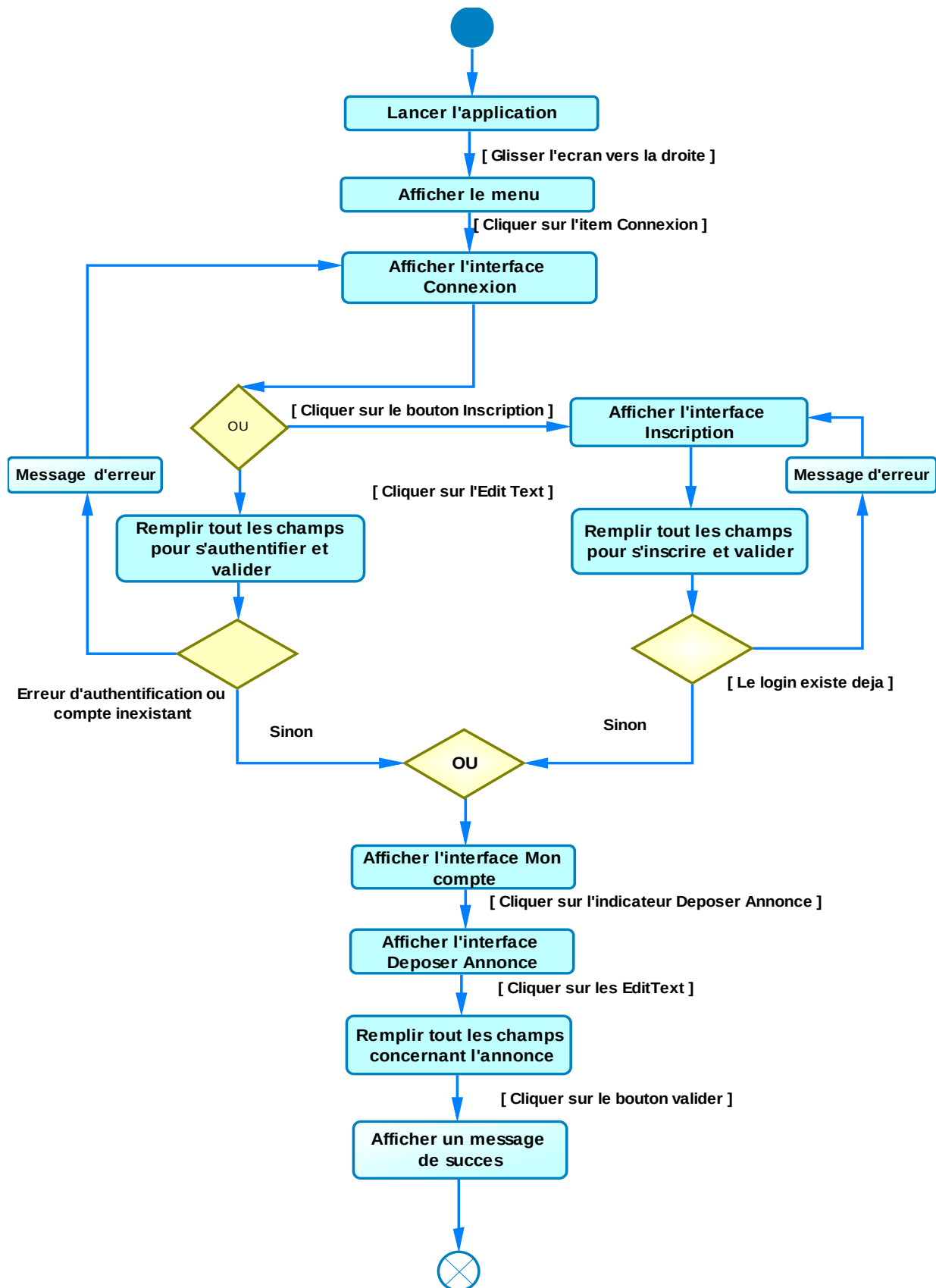


Figure III.10 : Diagramme d'activité du cas d'utilisation « Déposer une annonce ».

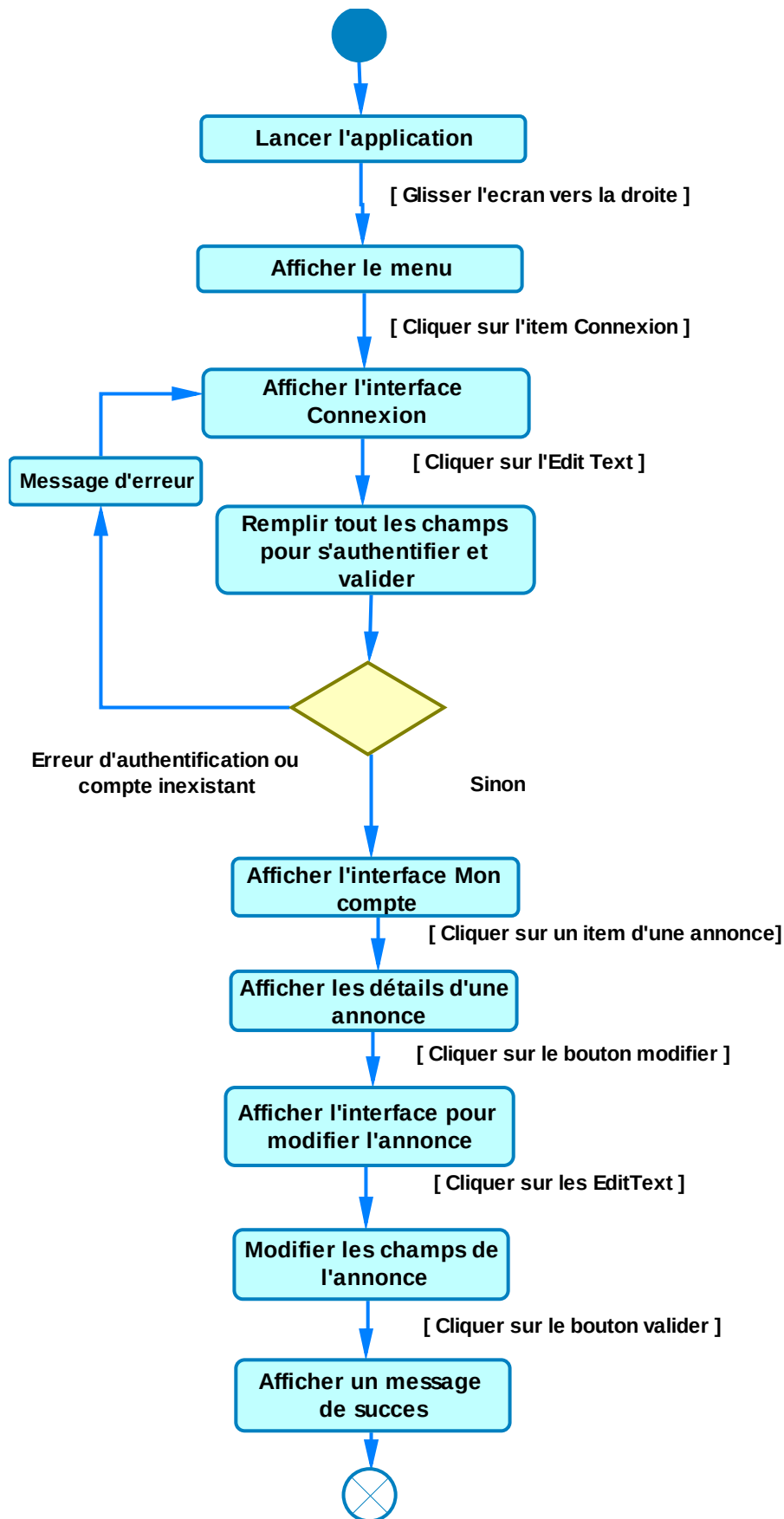


Figure III.11: Diagramme d'activité du cas d'utilisation « Modifier une annonce ».

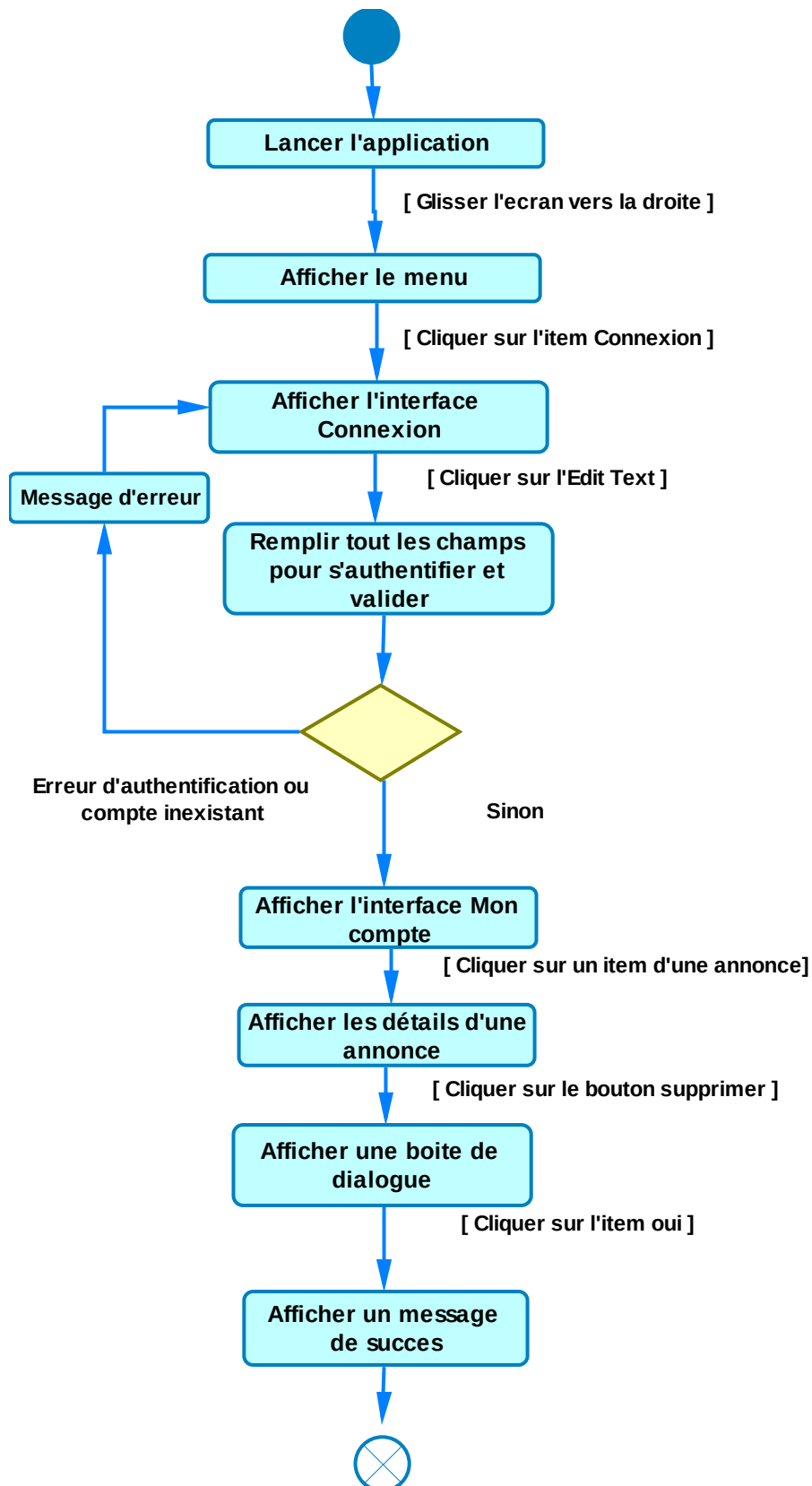


Figure III.12 : Diagramme d'activité du cas d'utilisation « Supprimer une annonce ».

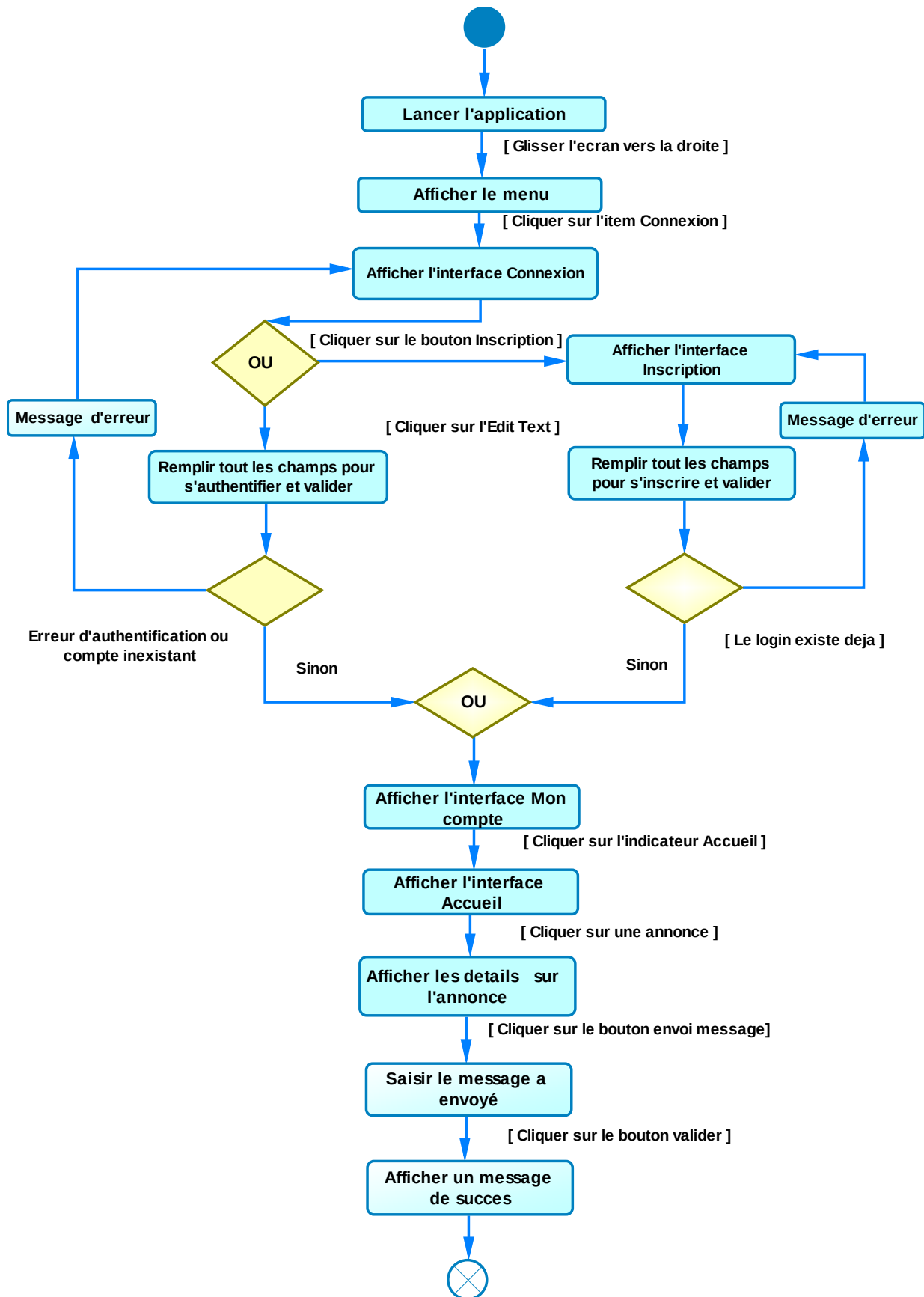


Figure III.13: Diagramme d'activité du cas d'utilisation« Envoyer un message».

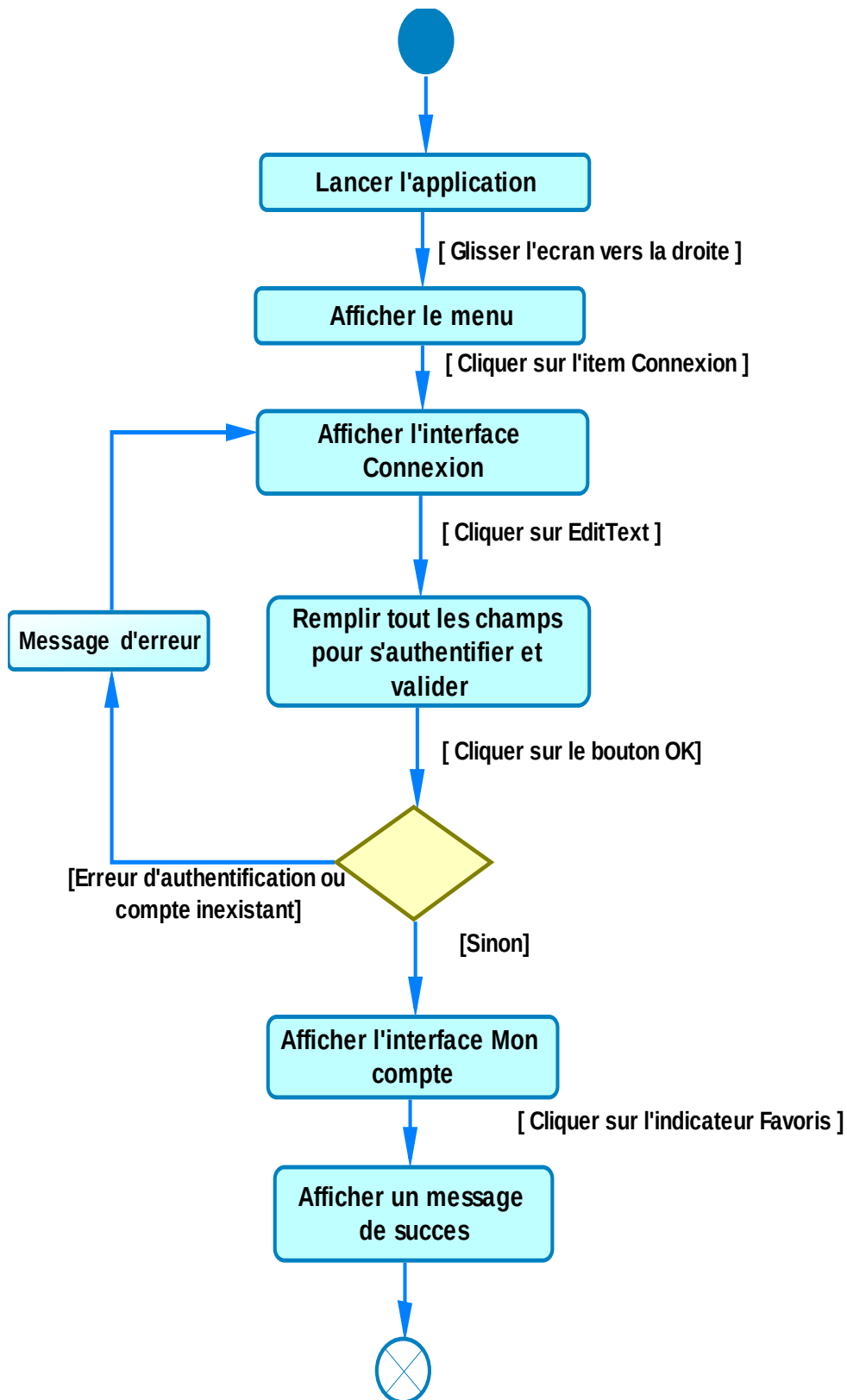


Figure III.14: Diagramme d'activité du cas d'utilisation « Consulter la liste des annonces favorites ».

III.11 Les diagrammes des classes :

III.11.1 Définition :

Le diagramme de classes a toujours été le diagramme le plus important dans toutes les méthodes orientées objets. C'est celui que les outils de génération automatique de code utilisent en priorité. C'est également celui qui contient la plus grande gamme de notations et de variantes, d'où la difficulté d'utiliser correctement tous ces concepts.

Le diagramme de classes constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classes ; seul l'algorithme général et le pseudo-code correspondant peuvent être associés à la modélisation.

La description du diagramme de classes est fondée sur :

- le concept d'objet,
- le concept de classe comprenant les attributs et les opérations,
- les différents types d'associations entre classes.

La figure III.15 illustre le diagramme de classe générale de notre application :

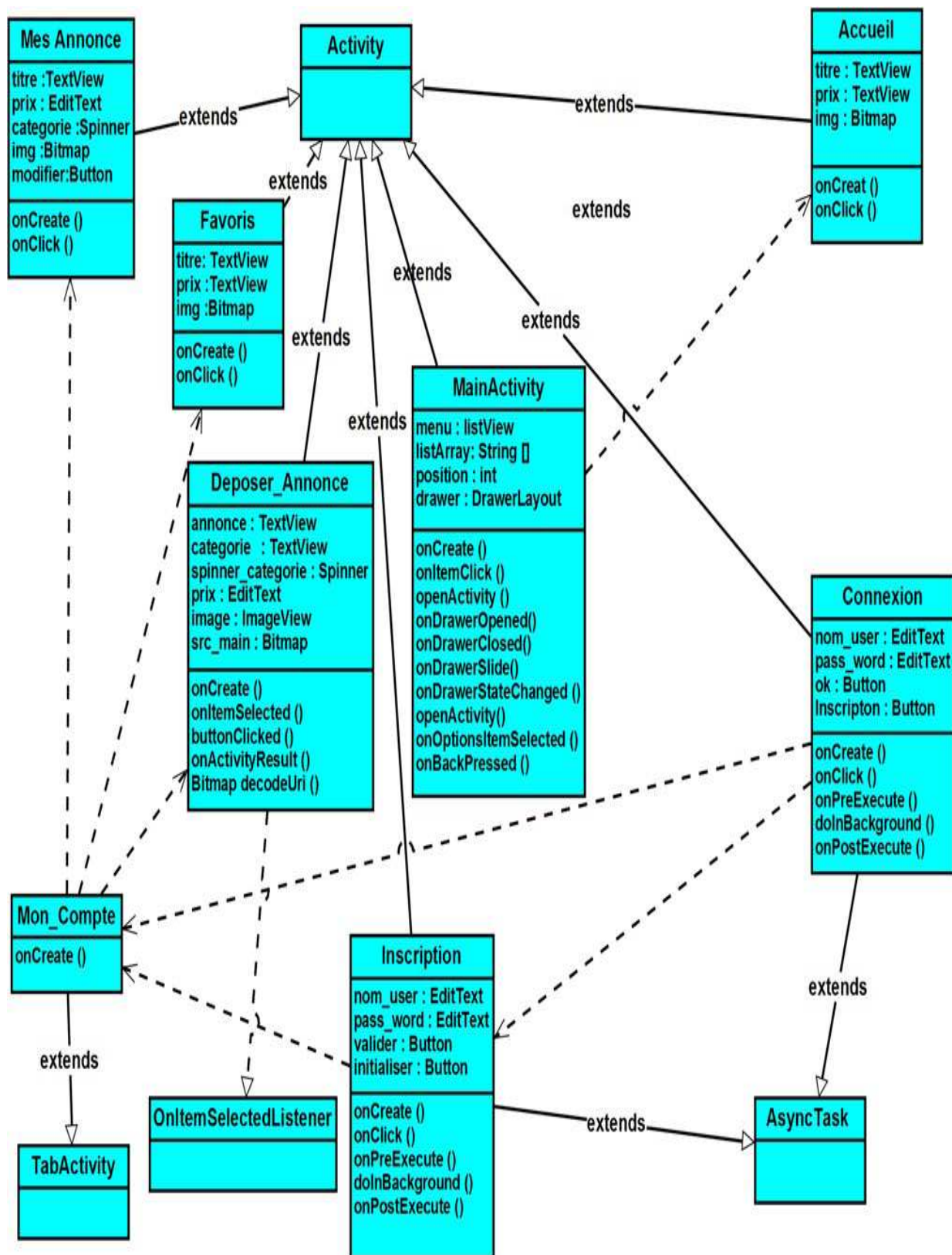


Figure III.15 : Diagramme général de classes.

CHAPITRE III : ANALYSE ET CONCEPTION

III.12 Le Niveau données :

Les tables :

L'application dans tous ces cas d'utilisation manipule les données stockées dans les bases de données, et elle contient les tables suivantes :

Table	Champs	Type des données	Clés	Signification
Client				
	name	varchar	-	nom d'utilisateur
	user_name	varchar	primary	login
	pass_word	varchar	-	mot de passe

Tableau III.15 : Table inscription.

Table	Champs	Type des données	Clés	Signification
Annonce				
	id_annonce	int	primary	Identifiant
	catégorie	varchar	-	la différente catégorie
	type_annonce	varchar	-	type de l'annonce
	titre_annonce	varchar	-	titre de l'annonce
	text_annonce	varchar	-	description de l'annonce
	prix	int	-	prix de l'annonce
	photo_1	mediumblob	-	image principale
	photo_2	mediumblob	-	image secondaire
	photo_3	mediumblob	-	image secondaire
	wilaya	varchar	-	nom de wilaya
	adresse	varchar	-	adresse d'utilisateur
	email	varchar	-	adresse email
	téléphone	int	-	numéro de téléphone

Tableau III.17 : Table annonce.

Table	Champs	Type des donnees	Clés	Signification
Voiture	id_voiture	int	primary	identifiant
	id_annonce	int	foreignKey	identifiant
	marque	varchar	-	marque de voiture
	annee_model	varchar	-	année mis en marche
	kilométrage	int	-	nombre de kilométrage marché
	carburant	varchar	-	le carburant (diesel, essence)
	boite_vitesse	varchar	-	la boite de vitesse

Tableau III.18 : Table voiture.

Table	Champs	Type des donnees	Clés	Signification
Location	id_location	int	primary	identifiant
	id_annonce	int	foreignKey	identifiant
	type_bien	varchar	-	le type de loyer
	meuble	varchar	-	meublée/non meublée
	surface	int	-	la surface d'espace
	pièces	int	-	nombre de pièces

Tableau III.19 : Table location.

III.13 Conclusion :

En clair, comme nous venons de le voir, ce chapitre était consacré, à la conception des diagrammes, des cas d'utilisation qui nous ont permis de dégager l'architecture générale de notre application, représentée par ces diagrammes de cas d'utilisation qui nous renseignent sur la relation homme machine de notre application.

Le chapitre suivant sera consacré à la réalisation de notre application.



CHAPITRE IV

RÉALISATION

IV.1 Introduction :

Après avoir achevé l'étape de conception de l'application, nous allons entamer dans ce chapitre la partie réalisation, le cœur et l'âme du processus de développement de l'application, elle est consacrée à la mise en œuvre de chacun des modules décrits dans le chapitre de la conception. Nous commençons par la description des environnements matériels et logiciels qui nous ont permis de réaliser notre projet, puis nous passons à la présentation des différentes techniques et fonctionnalités de notre application.

IV.2 Choix de l'architecture de l'application :

Notre application se connecte à un serveur de bases de données distant, via Internet, afin de récupérer les données. Ce qui nécessite aussi l'intégration d'un serveur web entre l'application client et le serveur de bases de données. D'où l'architecture de notre application est à 3 niveaux (*architecture 3-tiers*), elle est partagée entre:

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

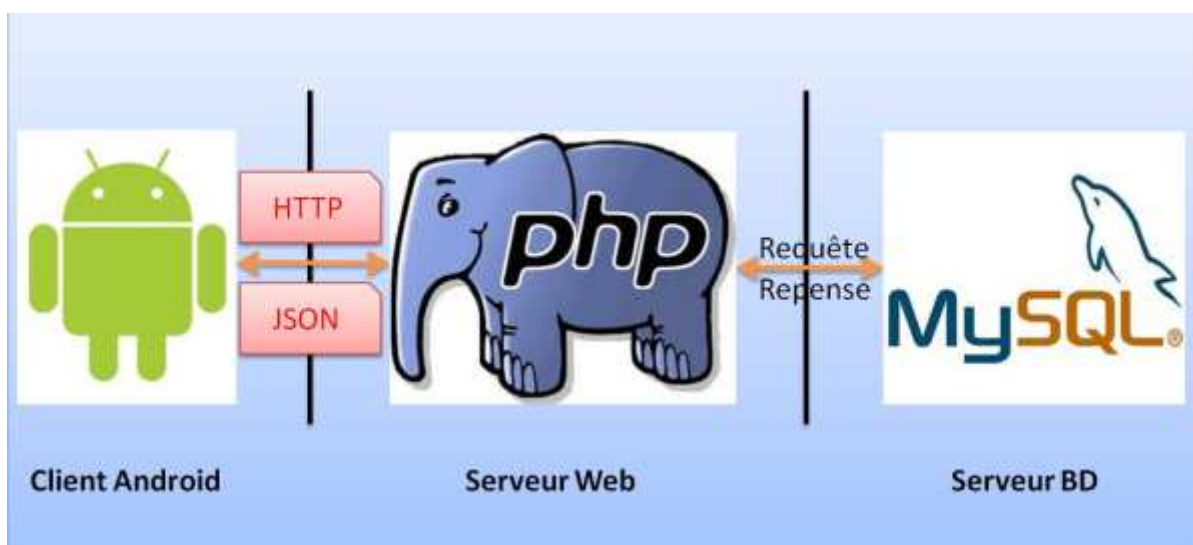


Figure IV.1: Architecture à trois niveaux.

IV.3 L'environnement :

IV.3.1 Environnement matériel :

Nous avons utilisé :

- Un PC Portable ayant les caractéristiques suivantes :
 - ✓ Intel(R) core™ i5 -2450M CPU @ 2.50 GHz.
 - ✓ 4.00 Go de mémoire vive.
 - ✓ Windows 7 - 32 bits.
- Un Smartphone sony xperia Z3 compact , pour réaliser les tests.

IV.3 .2 Environnement logiciel :

IV.3 .2.1 Le plugin de développement d'Eclipse : ADT

ADT (Android Developer Tools) est un plugin pour Eclipse qui fournit une suite d'outils qui sont intégrés à l'IDE Eclipse. Il vous offre l'accès à de nombreuses fonctionnalités qui vous aident à développer des applications Android. ADT offre un accès graphique à la plupart des outils SDK ainsi que d'un outil de conception de l'interface utilisateur pour le prototypage rapide, la conception, et la construction de l'interface utilisateur de notre application.



Figure IV.2: Éclipse ADT.

IV.3 .2.2 SDK Android :

Le SDK est livré avec un certain nombre d'outils couvrant différents aspects du cycle de développement d'une application Android. Le kit de développement propose une boîte à outils complète pour les tâches de compilation, débogage, génération de code AIDL, signature de votre application, etc. Le répertoire Tools du kit de développement contient ainsi un ensemble de programmes dont voici quelques exemples :

- **DDMS**: est un outil de débogage puissant ;
- **mksdcard**: sert à créer des cartes mémoire logicielles utilisables avec l'émulateur ;
- **sqlite3**: permet d'accéder aux fichiers de données au format SQLite.

IV.3 .2.3 Eclipse PHP:

Pour écrire les scripts PHP nous avons choisi le logiciel Eclipse PHP qui est un environnement de développement intégré spécialisé pour PHP, initié par IBM puis ouvert à la communauté Open Source, l'environnement de développement intégré (IDE) Eclipse pour développeurs PHP présente l'énorme avantage de disposer d'un nombre impressionnant de plug-in l'enrichissant dans tous les domaines de la programmation. Fonctionnant en PHP et avec la bibliothèque graphique SWT d'IBM, cet IDE "tourne" indépendamment du système d'exploitation, pourvu qu'une JVM soit installée évidemment.



Figure IV.3: PHP-Eclipse.

IV.3 .2.4 WampServer :

WampServer est une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.



WampServer

Figure IV.4: Le WampServer.

IV.3 .2.5) Le serveur web 000webhost :

000webhost est un serveur en ligne, utilisé par des millions d'utilisateurs, comme le Wampserver il contient un serveur MySQL pour la gestion de base de données après avoir les créer par l'outil phpMyAdmin. 000webhost est capable d'interpréter les requêtes http arrivant des utilisateurs et de fournir une réponse avec ce même protocole.

Ce serveur nous permet de charger et d'hébergement gratuit nos scripts PHP et de les interpréter.



Figure IV.5: Le serveur en ligne 000webhost.

IV.4 Langage de programmation :

IV.4 .1 Java :

Java est un langage de programmation moderne développé par Sun Microsystems, aujourd'hui racheté par Oracle. Il est le langage privilégié pour le développement d'applications Android. Google a donc tout naturellement conçu un plugin pour Eclipse. Une de ses plus grandes forces est son excellente portabilité : une fois votre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc.

IV.4.2 XML :

Le XML est un langage de balisage un peu comme le HTML — le HTML est d'ailleurs indirectement un dérivé du XML. Le principe d'un langage de programmation (Java, C++, etc.) est de donner des ordres à l'ordinateur pour qu'il puisse effectuer des calculs, puis éventuellement de mettre en forme le résultat de ces calculs dans une interface graphique.

IV.4 .3 PHP :

PHP est un open source langage de script rapide et riche en fonctionnalités pour développer des applications Web ou Internet / Intranet Applications.

IV.4 .4 MYSQL :

MySQL est un serveur puissant de base de données open source intégré basé sur un système de gestion de base de données relationnelle (SGBDR) et est capable de gérer une grande base de données de connexion simultanée.

IV.5 Protocole et Format de données :

IV.5 .1 Protocole de communication :

Dans notre projet, nous avons utilisé le protocole HTTP, afin de communiquer les données entre notre application et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client (facilite le dispatch des fonctions). En général, nous utilisons la méthode **Post** pour envoyer des données au programme situé à une URL spécifiée. Dans notre cas la requête Post envoyée à partir de l'application client vers le serveur est de la forme suivante :

`http://10.0.2.2:8080/Cote_serveur/nomFichier?parametre=valeur.`

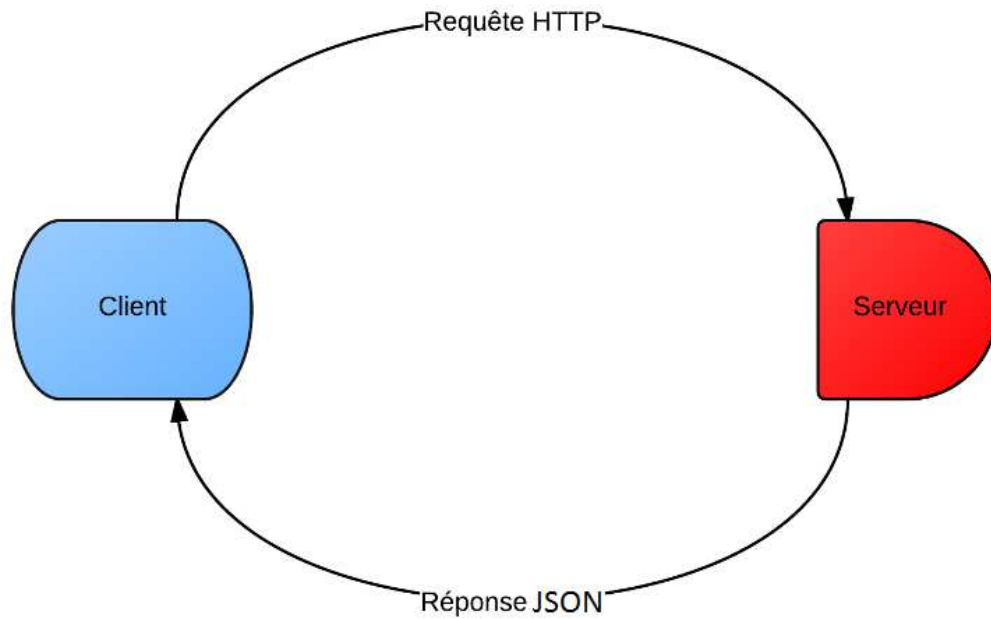


Figure IV.6: Le protocole de communication utilisé.

IV.5 .2 Format des données utilisé :

Après avoir défini le protocole utilisé, un format de données doit être présenté pour le transport de données :

Format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript, nous avons utilisé ce langage car il permet de représenter de l'information structurée.

Le principe est donc que le client Android appelle le script PHP et ce script va récupérer les données à partir de la base de données MySQL. Les données seront encodées au format JSON et le serveur les envoie au client. Ensuite l'application analyse et affiche ces données codées.



Figure IV.7: JavaScript Object Notation.

IV.6 Tester l'application sur L'émulateur :

Il est également très important de pouvoir effectuer des tests facilement et rapidement à chaque étape du développement, sans avoir à installer à chaque fois son application sur un terminal. Heureusement, dans les deux cas, les SDK intègrent un simulateur d'environnement d'exécution (émulateur) reproduisant sur la machine du développeur le comportement du téléphone et permettant de tester le fonctionnement de l'application sur un terminal virtuel. Cependant, ces simulateurs ont des fonctionnalités moindres qu'un vrai terminal.

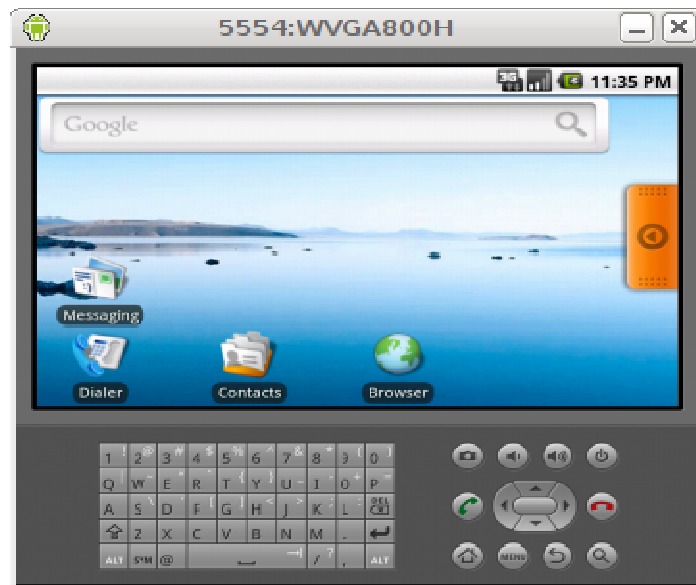


Figure VI.8: Émulateur Android de java SDK.

IV.7 Tester l'application sur Le téléphone :

Pour pouvoir utiliser un vrai téléphone android depuis l'éditeur de code eclipse, nous avons besoin d'installer un driver USB, puisqu'en branchant l'USB, Windows, par défaut, ne connaît pas le type de matériel. En général, le pilote à installer se trouve dans le dossier du sdk installé et plus exactement dans un dossier qui s'appelle google-usb-driver. Selon le processeur que nous avons sur le téléphone, nous choisissons le bon pilote.

Après avoir installé ce driver, nous pouvons alors commencer à utiliser le téléphone depuis eclipse.

Dans eclipse, nous ouvrons la perspective DDMS et dans l'onglet device à gauche, nous pourrions voir tous les émulateurs existants ainsi que le téléphone branché.

IV.8 Fonctionnement de notre application :

Nous allons présenter les écrans de notre application Android ainsi que leurs fonctionnalités sous forme de zone de texte présentant les tâches réalisées via cet élément.

IV.8 .1 Page Index :

Lors de lancement de notre application, une interface de chargement s'affiche contenant le logo de l'application AVA (Algérie Ventes et Achats) :



Figure IV.9: index d'application.

IV.8 .2 La page d'accueil :

Après quelques secondes le menu principal s'affiche, disposant d'une page d'accueil de cinq items (Accueil, Connexion, Recherche, Aide, Quitter)

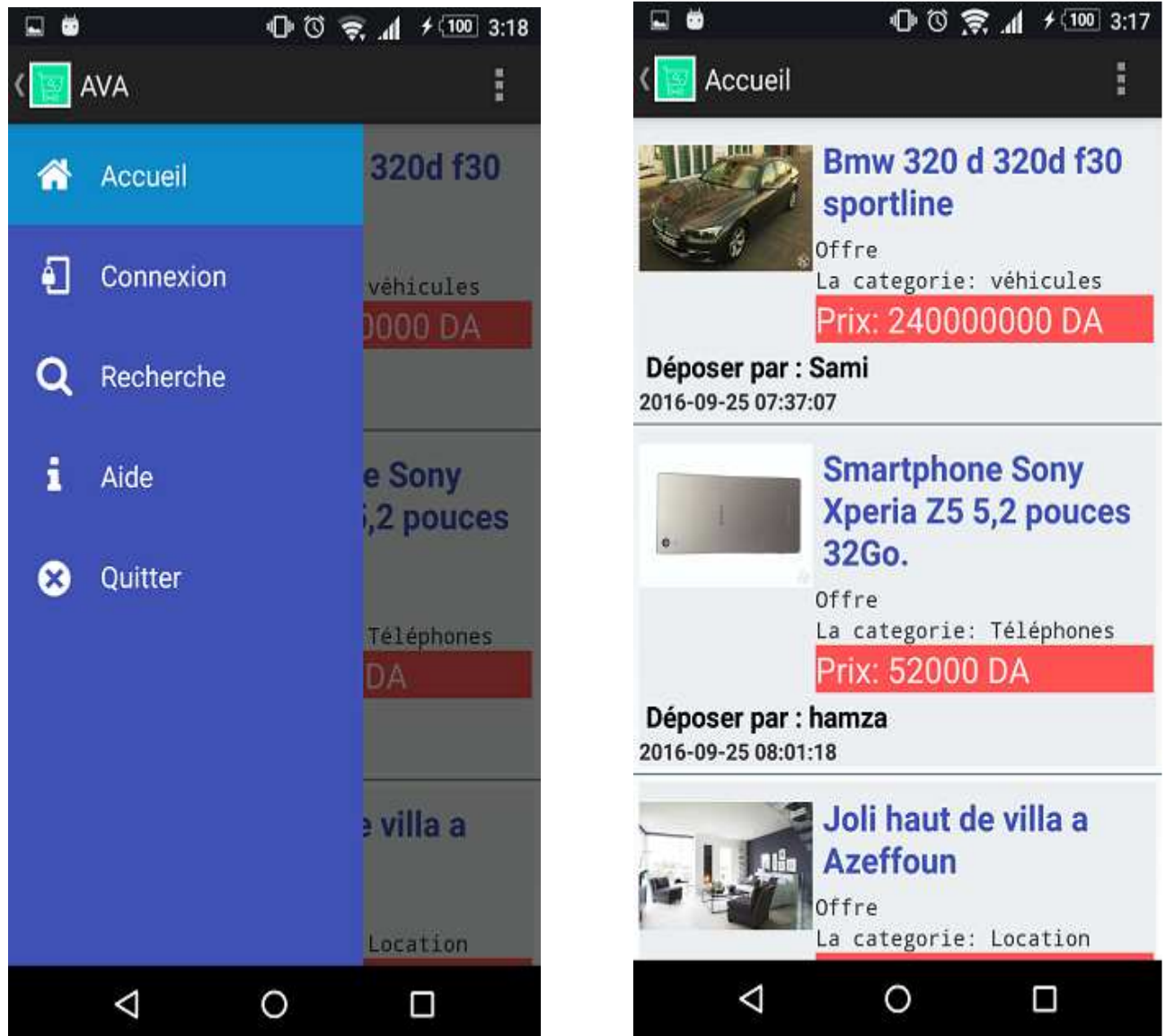


Figure IV.10: Page d'accueil.

IV.8 .3 La page de connexion :

Lorsque l'utilisateur clique sur l'item « Connexion », une interface « Connexion » s'affiche. Elle contient deux `EditText` pour que l'utilisateur puisse s'authentifier afin d'accéder à son compte après avoir cliqué sur le bouton « Connexion ».

Un bouton « Inscription » pour afficher l'interface « Inscription ».

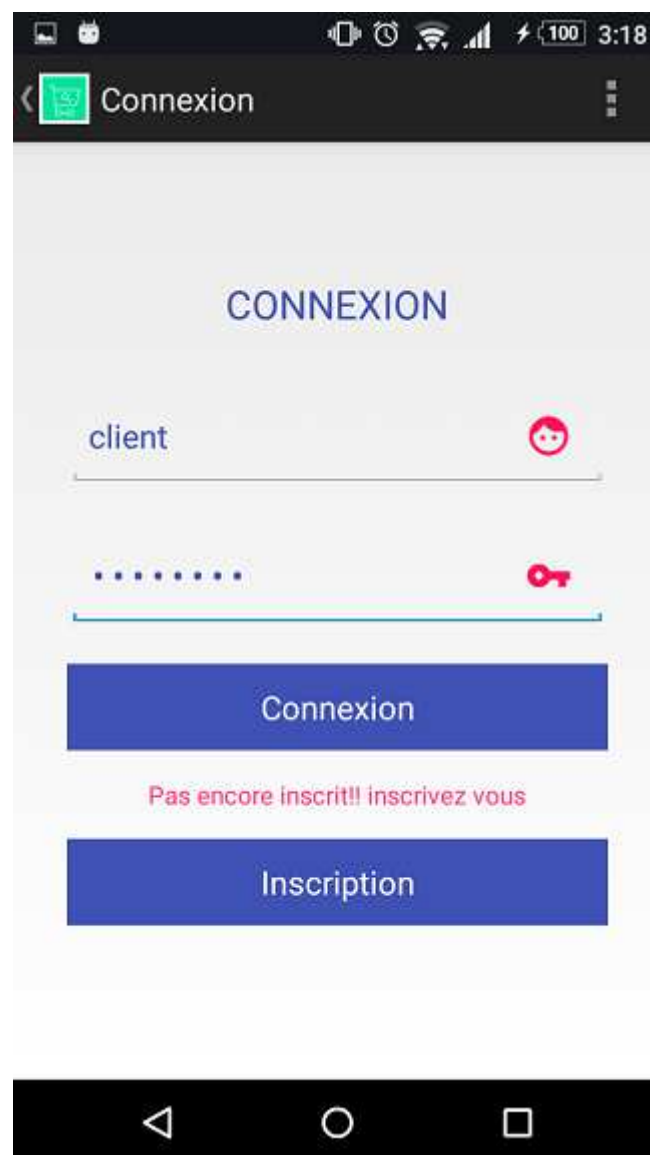


Figure IV.11: Page de connexion.

IV.8 .3 La page d'inscription :

Lorsque l'utilisateur clique sur l'item « Inscription » de l'interface « Connexion », une interface « Inscription » s'affiche. Elle contient un formulaire d'inscription et deux boutons :

- Inscrire : pour valider l'inscription.
- Initialiser : pour initialiser tous les champs de formulaire en cas d'erreur.

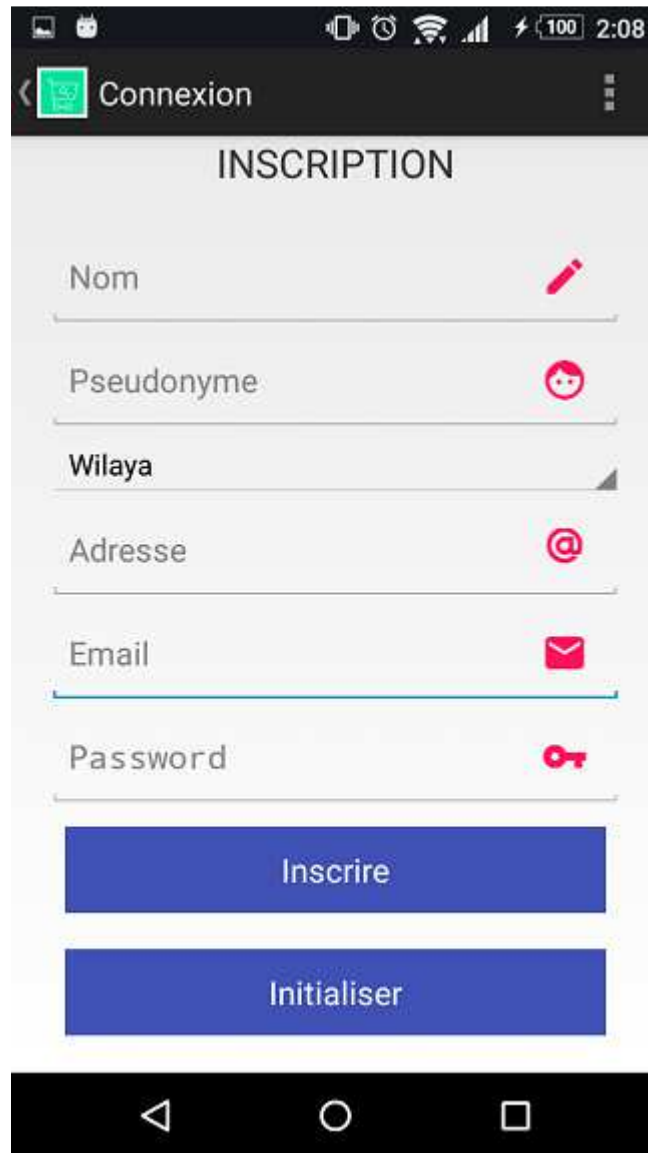


Figure IV.12: Page d'inscription.

IV.8 .4 La page mon compte :

Après l'authentification de client, ce dernier accède à cette interface où il l'aura le pouvoir de déposer une annonce, de consulter l'accueil pour voir les annonces en détails ainsi de consulter ses annonces enfin voir la liste des favoris.

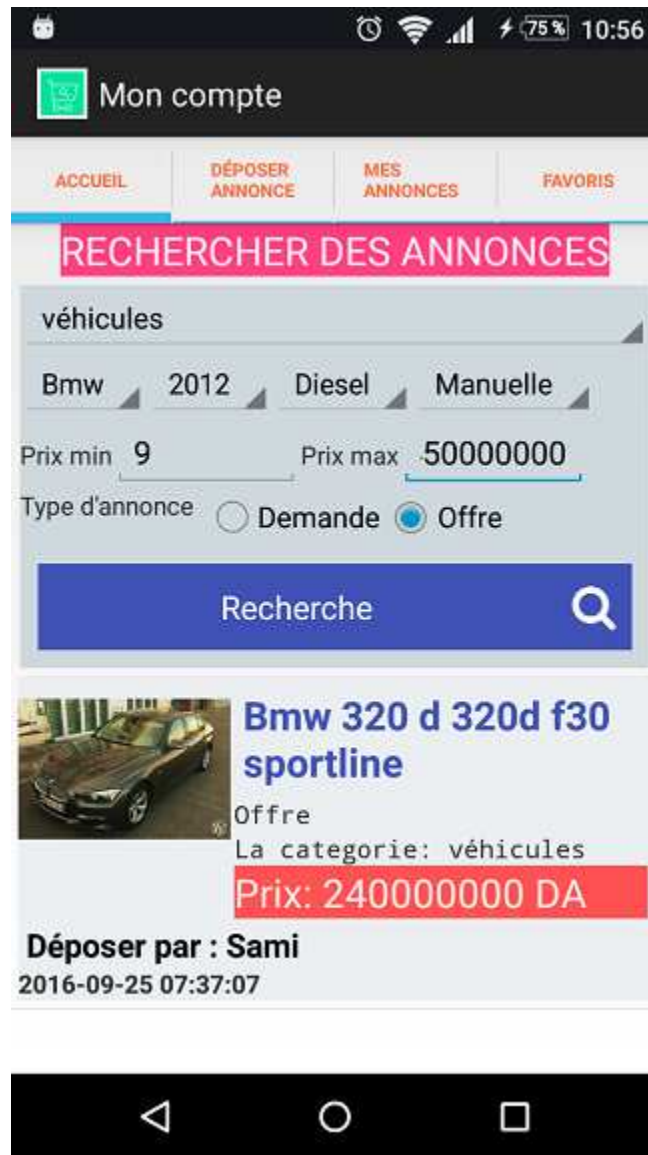


Figure IV.13: Page mon compte.

IV.8 .5 La page déposer une annonce :

Cette interface permet au client de déposer des annonces. Elle contient un formulaire qui doit le remplir et deux boutons:

- Valider : pour valider l'annonce.
- Annuler : en cas le client a hésité de déposer son annonce.

The screenshot shows a mobile application interface for 'Mon compte'. The top navigation bar has four tabs: ACCUEIL, DÉPOSER ANNONCE (selected), MES ANNONCES, and FAVORIS. Below the tabs, the title 'Mettre une annonce' is displayed in pink. The form contains the following fields:

- véhicules**: A dropdown menu showing 'Ferrari', '1992', 'Essence', and 'Manuelle'.
- Type d'annonce**: Radio buttons for 'Demande' and 'Offre' (selected).
- Titre d'annonce**: A text input field containing 'voiture haut gamme'.
- La description**: A text input field containing 'très belle voiture 🚗, en bonne état.'
- Le prix**: A text input field containing '1100000'.
- Photo**: A gallery of three images. The first image shows a green sports car. The second and third images are blue squares with white camera icons and the numbers '2' and '3' respectively.
- Téléphone**: A text input field containing '0674408964'.

At the bottom of the form are two large blue buttons: 'Valider' with a checkmark icon and 'Annuler' with a close icon. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Figure IV.14: Page déposer une annonce.

IV.8 .6 La page détails d'une annonce :

Lorsque le client clique sur une annonce dans l'accueil, une interface « Détail Annonce » s'affiche. Elle contient tous les détails de l'annonce, ainsi le bouton «Appel » pour contacter la personne qui a déposé l'annonce.



Figure IV.15: Page détails d'une annonce.

IV.8 .7 La page recherche d'une annonce:

Cette interface permet à l'utilisateur d'effectuer une recherche selon les critères : prix, catégorie et endroit.

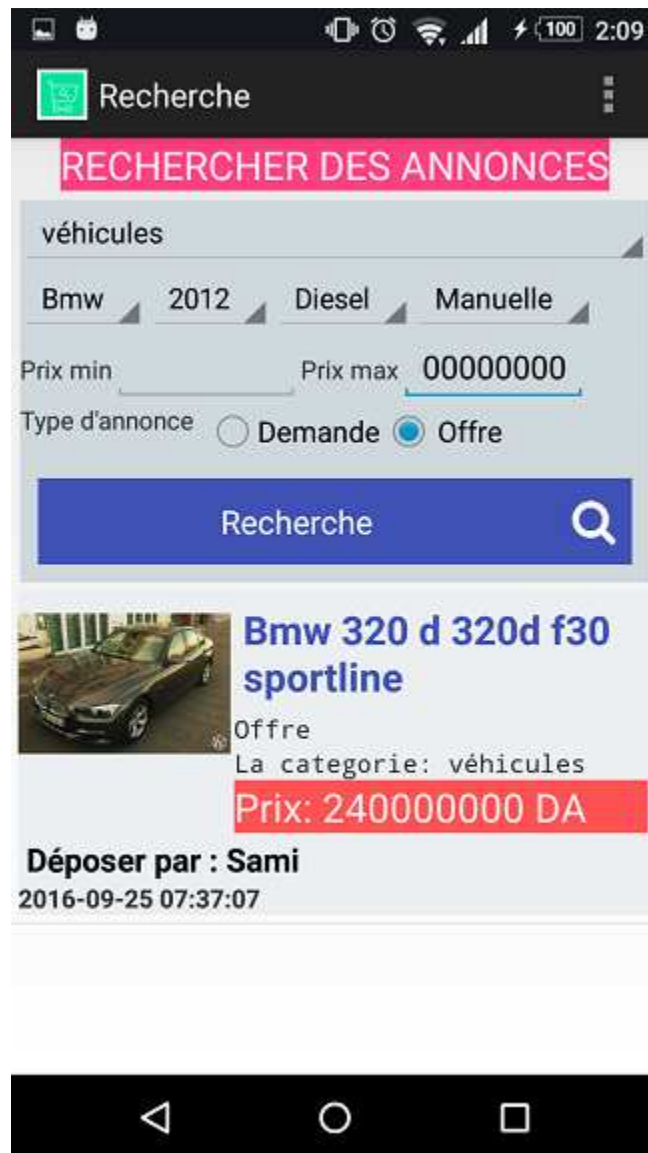


Figure IV.16: Page recherche d'une annonce.

IV.8 .8 La page Aide :

Une interface qui contient des informations sur notre application et le mode d'emplois afin de faciliter leur utilisation.

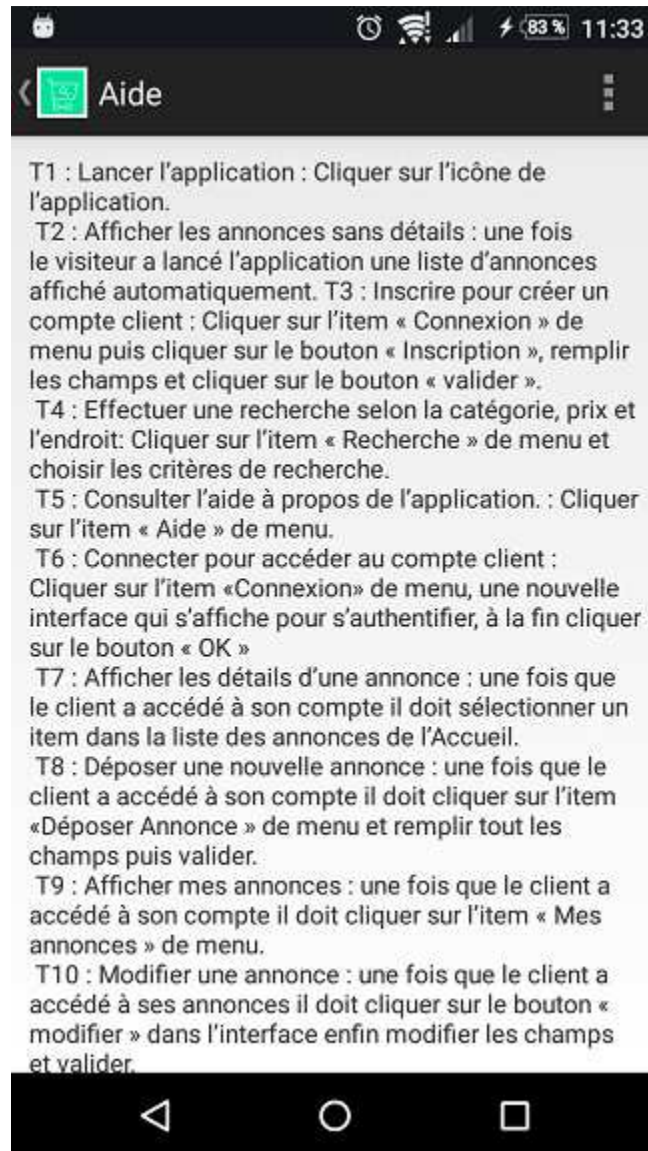


Figure IV.17: Page Aide.

IV.9 Conclusion :

Au cours de ce chapitre, nous avons présenté les différentes phases de la réalisation. Nous avons commencez par recenser les différentes technologies utilisées ainsi que l'environnement matériel et logiciel de notre travail. Ensuite, nous avons exposé certains imprimes écrans témoignant des différentes facettes de notre application.

CONCLUSION GENERALE

Au bout de notre cursus en master informatique, nous avons été chargés de réaliser un projet de fin d'étude. Notre travail s'est basé sur le développement d'une application sur les technologies mobiles (Smartphone). A l'issue de la réalisation de ce travail, nous pouvons affirmer que notre projet nous a été d'une grande utilité dans la mesure où il nous a permis de nous familiariser avec le travail sur une nouvelle plate-forme à savoir la plate-forme Android.

Durant ce travail, nous avons élaboré une étude sur le système d'exploitation « Android » et son importance sur le plan social afin de préciser le but principal pour la future application. Le point de début de notre travail est l'étape d'analyse et de spécification des besoins. Une fois nos objectifs sont fixés nous avons entamé la phase de conception afin de mener à bien notre projet. Nous avons, ensuite, procédé à la phase de réalisation au cours de laquelle nous nous sommes familiarisés avec le langage de programmation JAVA(SDK).

Pour conclure, notre travail peut être sujet à des extensions. En effet, nous envisageons d'ajouter une option de paiement en ligne, et une autre qui permet de localiser une personne sur une map enfin la possibilité de développer sous d'autres plateformes comme IphoneOS ou BlackBerry.

Bibliographie & Webographie

Références Bibliographiques :

- [IF811] Joseph Gabay, David Gabay, UML 2 Analyse et Conception, Consulté le 19-05-2016
- [3] James Douglass Lefruit
Ingénieur de Recherche, INRIA Grenoble (pdf)
- [4] Jean-François Lalande-2016 (pdf : page 5)
- [5] Christian Bulfone /Licence MIASS (pdf)
- [12] Benoît Charroux EFREI (École d'ingénieur), Aomar Osmani université Paris XIII et
Yann Thierry-Mieg université Paris VI, UML2 pratique de la modélisation 2e édition.

Webographie:

- [1]
<http://android-france.fr/android-cest-quoi/>
Android c'est quoi ? Consulté le : 2016-06-13 19:20:12
- [2]
<https://fr.wikipedia.org/w/index.php?title=Android&oldid=129546152>
Android Consulté le : 2016-06-23 17:19:16
- [6]
https://fr.wikipedia.org/w/index.php?title=Pile_de_protocoles&oldid=118011645
Pile de protocoles Consulté le : 2016-07-21 15:10:12
- [7]
<http://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a-3-niveaux>
- [8]
<https://fr.wikipedia.org/w/index.php?title=WampServer&oldid=129674192>
WampServer Consulté le : 2016-09-21 08:08:17
- [9]
<http://laurent-audibert.developpez.com/Cours-BD/?page=langage-sql>
Bases de Données et langage SQL Consulté le : 2016-09-21 20:39:14
- [10]
<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/phpmyadmin-5>
Le cours "phpMyAdmin" sur @OpenClassrooms Consulté le : 2016-09-04 20:30:14
- [11]
<http://www.xul.fr/ajax-format-json.html>
JSON, un format pour échange de données avec JavaScript
- [13]
<http://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html>
UML Cours 1 : Diagrammes de cas d'utilisation Consulté le : 2016-09-13 19:20:12