

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Systèmes informatiques**

Présenté par

Prénom OUCHEBBOUK Massinissa

Prénom IDRENMOUCHE Karim

Thème

Réalisation d'une application Android pour l'agrégation de contenus

Mémoire soutenu publiquement le ...28.../...06.../ 2016..... devant le jury composé de :

Président : M Rachida AOUDJIT

Encadreur : M Mohammed DAOUI

Co-Encadreur : M Massinissa MANSOURI

Examineur : M Malika BELKADI

Examineur : M Rebiha HADAoui

Remerciements

C'est avec un grand plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Nous tenons tout d'abord à remercier notre promoteur Monsieur DAOUI MEHAMMED pour son soutien, son sérieux, sa gentillesse et surtout pour son aide précieuse tout au long de l'élaboration de ce travail.

Nos vifs remerciements s'adressent à tous les membres de l'équipe Symphotech Solution pour leur encadrement, assistance, soutien, leur disponibilité et leurs précieux conseils durant la période du stage(particulièrement Massinissa Mansouri).

Nous tenons aussi à exprimer l'honneur qui nous est fait par les membres du jury, en acceptant de juger notre travail.

Nous nous acquittons, enfin, volontiers d'un devoir de gratitude et de remerciements à tous les enseignants du département informatique pour la qualité de l'enseignement qu'ils ont bien voulu nous prodiguer durant nos études afin de nous fournir une formation efficiente.

Dédicaces

*Je dédie notre travail à toute ma famille et
mes amis*

*A toutes les personnes qui ont participé de
près ou de loin à l'aboutissement de ce
travail.*

*Et surtout à mon binôme Massinissa sans
qui ce mémoire n'aurait pas pu être
effectué.*

*Un grand hommage a Mr Taleb Hakim qui
a toujours été un modèle, un honneur de
travailler avec lui (On T'oubliera Jamais
tu resteras gravé dans notre mémoire)*

Karim

Dédicaces

*Je dédie notre travail à toute ma famille,
mes amis et Katia*

*A toutes les personnes qui ont participé de
près ou de loin à l'aboutissement de ce
travail.*

*Et surtout à mon binôme Karim sans qui
ce mémoire n'aurait pas pu être effectué.*

Massinissa

SOMMAIRE

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

Chapitre 1 : Les Systèmes d'exploitation mobile

Introduction.....	2
I.Système d'exploitation des Smartphones	2
I.1. iPhone OS	2
I.2. BlackBerry.....	3
I.3. Symbian OS.....	3
I.4. Windows Mobile	4
I.5. Android.....	5
II. Comparaison entre les différents systèmes d'exploitation mobiles.....	5
III. Présentation de la plateforme Android	6
III.1 .Architecture d'Android.....	6
III.1.1- Application	7
III.1.2- Framwork Applicatif.....	7
III.1.3- Les Librairies.....	7
III.1.4- Environnement d'exécution Android.....	8
III.1.5-Le noyau linux	8
III.2. Les concepts d'une application Android	9
III.2.1-L'activité (Activity)	9
III.2.1-1 .Etat d'une activité	9
III.2.1-2 .Cycle de vie d'une activité.....	10
III.2.2 .Les Services.....	11
III.2.3 .Content Provider	12
III.2.4 .Broadcast Receiver	12
III.2.5 . Intent	12
IV.Historique et versions d'Android	12
IV.1.Open Handset Alliance.....	12
IV.2. Bugdroïde.....	13
IV.3. Versions d'Android.....	13
V.Différents équipements utilisant Android	15
V.1.Les Smartphones.....	15
V.2.Les Tablettes	16
V.3.Les télévisions	16
V.4.Montre	16
V.5.Consoles de jeux vidéo.....	16
V.Conclusion.....	17

Chapitre 2 : les agrégateurs de contenus

Introduction.....	18
Description d'un agrégateur	18
Interface d'accès aux services	19
RSS	19
1. Domaines d'utilisation.....	19
2. Type d'un fichier RSS	20
4. Métadonnées	21
5. Description de chaque article.....	21
6. Modules.....	22
Applications mobiles pour lire les flux RSS	22
Conclusion	23

Chapitre 3 : Analyse et Conception

Partie1 : Analyse

Introduction.....	24
Les Méthodes de conception	24
Présentation de l'UML.....	24
1. Modélisation avec l'UML.....	25
2. La démarche de modélisation avec l'UML	26
1. Identification des acteurs.....	26
2. Identification des besoins	27
4. Spécification des tâches	27
4.1-Définition.....	27
2. Spécification des scénarios	27
Les cas d'utilisation.....	28
1. Cas d'utilisation s'identifier à l'espace Administrateur	28
2 .Cas d'utilisation User	29
3. Diagramme de cas d'utilisation	29
3.1-Diagramme de cas d'utilisation « Administrateur» (Coté serveur)	30
3.2-Diagramme de cas d'utilisation «User» (Coté mobile)	31

Partie2 : Conception

Introduction.....	32
1. Conception générale	32
1.1-Les composants d'une application Android	32
Conception détaillée	33
1. Diagramme de séquence.....	33
1.1-Authentification (Coté serveur)	34

1.2-Recherche simple (Coté mobile)	35
1.3-Recherche avancée (Coté mobile)	36
1.4-Recherche par localisation (Coté mobile)	37
Diagramme de classe général	38
Conclusion	39

Chapitre 4 : Réalisation

Introduction.....	40
Environnement matériel et logiciel.....	40
1. Environnement matériel	40
1.1-Architecture matérielle	40
1.2-Matériels utilisés	40
1.3-Technologie	41
2. Environnement logiciel.....	42
2.1-PHP	42
2.2-Yii Framwork	42
2.3-PhpStorm	43
2.3-JAVA	43
2.2-XML	43
2.3-Android Studio	44
2.4-SDK	45
2.5-Git.....	45
Protocole et format de données	46
1. Protocole de communication	46
2. Format de données communiquées	46
Interface Homme/Machine de l'application	47
1. Interface « serveur Web »gestion des fournisseurs	47
2. Interface « Client Android» Accueil.....	48
2. Interface recherche par mots clés.....	49
3. Interface recherche avancée.....	50
4. Recherche par localisation (GPS)	51
5. Interface liste des fournisseurs (journaux)	52
Conclusion	53

Conclusion générale

Conclusion générale	54
---------------------------	----

Liste des figures

Chapitre 1

Fig. I.1. Icône représentant le système iOS.....	2
Fig. I.2. Exemple d'un téléphone utilisant BlackBerry OS.....	3
Fig. I.3. Exemple d'un téléphone sous OS Symbian.....	4
Fig. I.4. Icône représentant le système Windows Mobile	4
Fig. I.5. Performances des Systèmes d'Exploitation mobiles (Edouard Ricque)	5
Fig. I.6. Parts de marché des Systèmes d'Exploitation mobile	6
Fig. I.7. Architecture d'un système Android	7
Fig. I.8. Différence entre la JVM et Dalvik.....	8
II.9. Cycle de vie d'une activité.....	11
I.10. Personnage Bugdroïde	13
Fig. I.11. Répartition des versions d'Android	15

Chapitre 2

Fig. II.1 : Logo des flux RSS	19
Fig. II.2 : fichier XML.....	20

Chapitre 3

Figure III.1 : Les diagrammes de modélisation	25
Figure III.2 : La démarche adoptée pour la modélisation.....	26
Fig. III.3. Cas d'utilisation « s'identifier à l'espace Administrateur »	28
Fig. III.4. Cas d'utilisation « Interface principal des utilisateurs »	29
Fig. III .5 « Diagramme de cas d'utilisation pour l'administrateur »	30
Fig. III .6 « Diagramme de cas d'utilisation pour un utilisateur».....	31
Fig. III .7 Composition d'une application Android.....	33
Fig. III.8: Diagramme de séquence du cas d'utilisation «Authentification»	34
Fig. III.9: Diagramme de séquence du cas d'utilisation «Recherche simple»	35
Fig. III.10: Diagramme de séquence du cas d'utilisation «Recherche avancées»	36
Fig. III.11: Diagramme de séquence du cas d'utilisation «Recherche par localisation»	37
Fig. III.12. Diagramme de classe général	38

Chapitre 4

Fig. IV.1 : <i>Architecture matériel du système</i>	40
Fig. IV.2 : Architecture 3-tiers du point de vue technologique.....	41
Fig. IV.3 : Logo framwork yii.....	42
Fig. IV.4 : Interface de l'environnement PhpStorm	43
Fig. IV.5. Exemple d'un fichier XML	44
Fig. IV.6 : Interface de l'environnement Android Studio 1.1.0.....	44
Fig. IV.7 : Interface de l'Android SDK Manager	45
Fig. IV.8 : Interface de gestion des fournisseurs.....	47
Fig. IV.9 : Interface accueil	48
Fig. IV.10 : Interface recherche par mots clés	49
Fig. IV.11 : Interface recherche avancée.....	50
Fig. IV.12 : Recherche par localisation (GPS)	51
Fig. IV.13 : Interface liste des fournisseurs	52

Liste des tableaux

Chapitre 3

Tableau III.1 : Spécification des tâches.....	27
Tableau III.2 : Spécification des scénarios	28

Chapitre 4

Tableau IV.1 : Les différentes technologies utilisées dans l'application.....	41
---	----

Introduction générale

Depuis son existence, l'internet a énormément influencé le monde par sa révolution, notamment après l'apparition du web qui rend beaucoup de services aux internautes du fait qu'il leur permet d'étudier, de communiquer, de suivre l'actualité ...etc.

L'information électronique fait partie des services offerts par le web. Il peut être sommairement défini comme l'ensemble des échanges numériques liées.

L'objectif du travail est de réaliser une application d'Android qui permet de lire des flux RSS pour suivre l'actualité des journaux sur un Smartphone

Notre travail a été réalisé au sein de l'équipe Web de la société Symphotech Solutions à Draa-Ben-Khedda Tizi- Ouzou, c'est une boîte de développement informatique, société de services et d'ingénierie informatique qui propose des solutions informatiques et web innovantes répondant aux besoins de tout type d'entreprise. Elle est aussi spécialisée dans la création de sites internet et la conception de logiciels spécifiques.

Ce travail s'est déroulé en trois étapes. La première étape consistait à nous familiariser avec l'environnement Android et choisir les outils envisageables pour l'aboutissement du projet. Par la suite, nous avons procédé la modélisation et le développement de l'application.

Ce rapport est subdivisé en quatre chapitres :

Dans le premier chapitre « ***Les Systèmes d'exploitations mobiles*** » nous allons présenter les systèmes d'exploitation conçus pour fonctionner sur un appareil mobile. Dans le deuxième chapitre« ***Les agrégateurs de contenu*** » les agrégateurs ainsi que leur utilisation dans les applications mobiles seront présenté. Dans le troisième chapitre« ***Analyse et Conception*** » nous élaborons une conception détaillée des cas d'utilisation, des diagrammes de séquence, ainsi que le diagramme de classe complet.

Enfin dans le dernier chapitre« ***Réalisation***» nous commencerons d'abord par présenter l'environnement matériel et logiciel, puis, l'état de la réalisation.

*Les Systèmes
d'exploitation
mobiles*

Introduction

Un système d'exploitation mobile est un système d'exploitation conçu pour fonctionner sur un appareil mobile, Dans cette section, nous aborderons quelques plateformes existantes, Cela nous permettra d'avoir une idée assez générale des systèmes d'exploitation tournant sur mobiles, et d'essayer de faire le positionnement d'Android dans l'environnement des systèmes d'exploitation pour mobile.

I. Système d'exploitation des Smartphones

Les 4 principaux systèmes sont Android, iOS, Windows phone et Blackberry OS, représentant la quasi-totalité des parts de marché.

I. 1. iPhone OS

iPhone OS d'Apple (icône dans la figure 1.1), est le premier OS pour téléphones tactiles qui a véritablement lancé la vague des Smartphones. Apparue sur le marché en 2007, il a innové dans un domaine qui n'était pas encore développé. Cependant, iOS n'a pas beaucoup évolué depuis sa création. iOS se caractérise par une interface peu chargée, qui permet l'exécution de tâches rapidement pour les utilisateurs réguliers, mais qui laisse peu de place à la personnalisation. Dans de nombreux cas on observe le positionnement de mêmes éléments qui diffère d'une application à l'autre. Une étude a montré que les utilisateurs sont fréquemment ralentis par ce manque d'homogénéité, ce qui entraîne un grand nombre d'erreurs.



Fig. I .1.Icône représentant le système iOS

I. 2. BlackBerry

Tout comme l'iPhone, le BlackBerry (voir figure I. 2) est aussi un téléphone très Utilisé. La fonction majeure qui a fait décoller le BlackBerry était le push mail. L'utilisateur n'a alors plus besoin de consulter périodiquement sa boîte pour vérifier s'il n'a pas de nouveaux messages. Ceux-ci lui parviennent directement comme un simple SMS. Cette fonctionnalité est assurée par les serveurs d'infrastructure du fabricant RIM (Research In Motion) avec un protocole propriétaire. Le mail est donc le point fort des BlackBerry qui a fait son succès auprès des cadres et dirigeants.



Fig. I.2.Exemplaire d'un téléphone utilisant BlackBerry OS

I. 3. Symbian

Symbian est très populaire (particulièrement en Europe) (voir figure II.3), c'est le leader mondial du système d'exploitation pour Smartphones Il offre une plateforme flexible, ce qui veut dire que les constructeurs de téléphones mobiles peuvent facilement y ajouter leurs technologies et infrastructures. De plus, il est soutenu par les grandes manufactures de l'industrie mobile comme Sony Ericsson, Motorola, et Nokia. Ce dernier étant constructeur numéro un des mobiles dans le monde et principal actionnaire de Symbian, cela garantit la position du Symbian dans le marché des systèmes d'exploitation pour les Smartphones. Néanmoins depuis quelques années, la part de marché de Symbian diminue à cause de concurrence avec d'autres plateformes.



Fig. I.3.Exemplaire d'un téléphone sous OS Symbian

I. 4.Windows Mobile

Windows Phone (voir figure 1.4), apparu en 2010, a su s'inspirer de ses concurrents et de son expérience passée dans le domaine du mobile. Proposant une interface simple et épurée ainsi que de multiples possibilités de personnalisation, il permet d'arriver rapidement à l'exécution d'une tâche ce qui a particulièrement séduit les utilisateurs. C'est d'ailleurs Windows Phone qui se trouve être l'OS mobile le plus adapté aux utilisateurs.

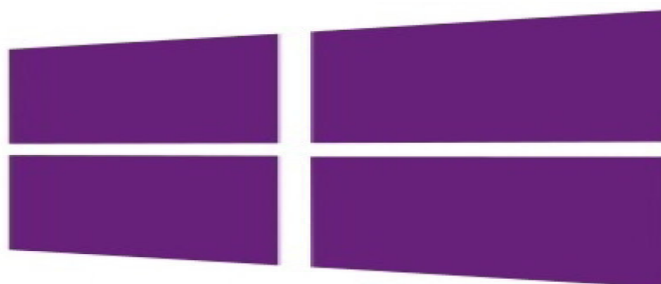


Fig. I.4. Icône représentant le système Windows Mobile

I. 5. Android OS

Android est un OS basé sur linux conçu par Google pour les appareils mobiles. Il est gratuit et complètement ouvert C'est-à-dire que le code source et les APIs sont ouvertes, il a été développé par l'Open Handset Alliance en 2007 et il est devenu une plateforme ouverte en 2008. Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps.

II. Comparaison entre les différents Systèmes d'exploitation mobiles

Dans ce qui suit, nous donnons une comparaison entre les différents systèmes d'exploitation mobiles au niveau de leurs performances en premier lieu, puis de leur popularité et nous examinons les parts de marché de chacun.

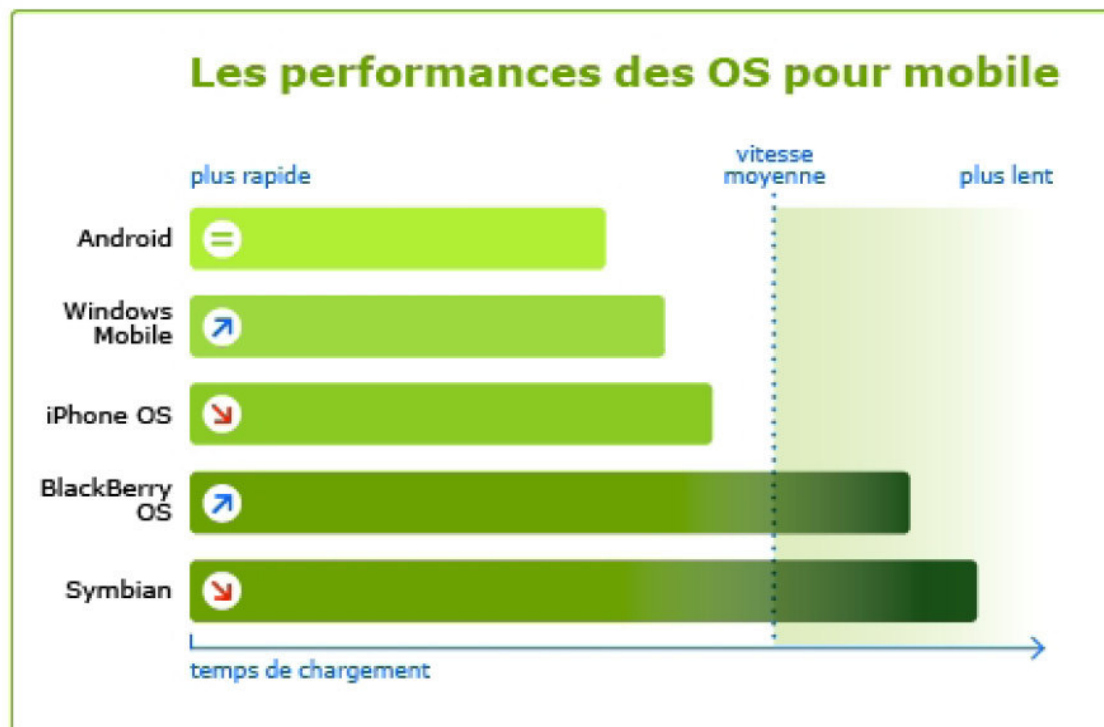


Fig. I.5. Performances des Systèmes d'Exploitation mobiles (Edouard Ricque) ¹

L'histogramme montre les différents systèmes d'exploitation mobiles comparés à base de leurs performances par temps de chargement, on remarque que le système Android présente le meilleur taux des performances d'un OS mobile par rapport aux autres systèmes, suivi du Android est un OS basé sur linux conçu par Google pour les appareils mobiles. Il est gratuit et complètement ouvert C'est-à-dire que le code source et les APIs sont ouvertes, il a été développé par l'Open Handset Alliance en 2007 et il est devenu une plateforme ouverte en 2008. Android a intégré plusieurs services de Google pour accéder rapidement aux

¹Edouard Ricque

services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps. Système Windows Mobile. Ces performances expliquent les différentes parts acquises dans le marché de l'OS mobile. Le meilleur taux de performance réservé pour Android est confirmé par la grande part du marché qu'il réquisitionne illustrée dans l'histogramme de la figure 1.6

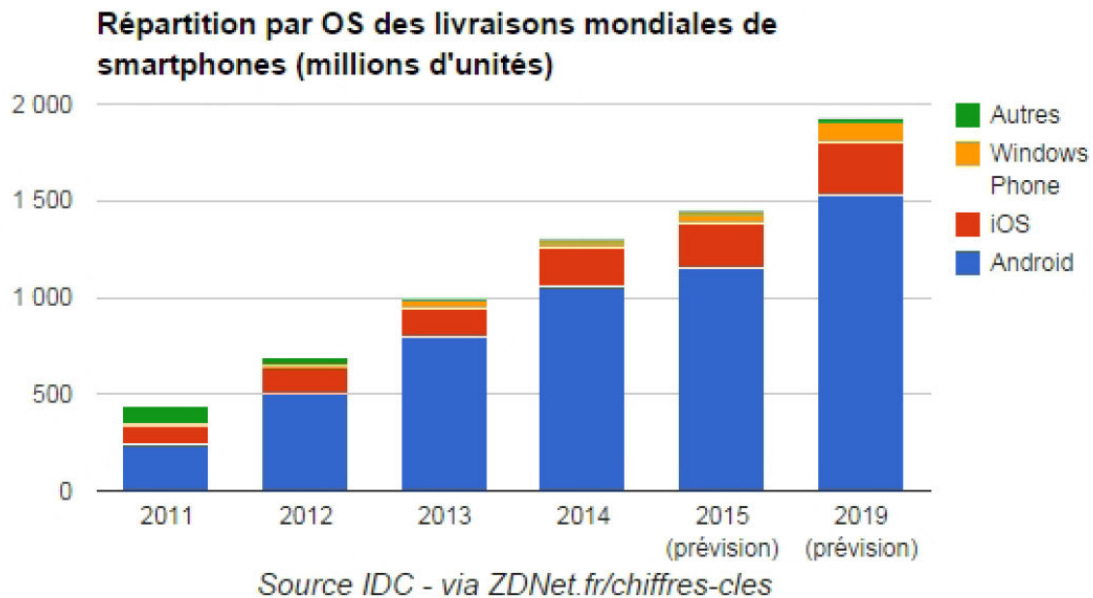


Fig. I.6. Parts de marché des Systèmes d'Exploitation mobile²

Selon l'histogramme sur les parts de marché des OS pour mobile, on remarque qu'Android est le plus commercialisé par rapport aux autres OS comme Windows Phone iOS et depuis 2011 il garde toujours sa première place dans le marché mondial.

Vue que nous allons développer notre application sous Android intéressons nous à ce système plus en profondeur.

III. Présentation de la plateforme Android³

Cette partie présente Android sous un autre angle que celui de l'utilisateur, qui est celui d'une vue de l'intérieur, en explorant les aspects techniques internes.

III. 1 .Architecture d'Android

La plate-forme Android est composée de différentes couches :

² Source IDC-via ZDNet.fr/chiffres-cles

³ Cours Andoid 2014 .Mr M. Daoui

III. 1.1- Application

Le système Android peut exécuter un grand nombre d'applications incluant les clients de messagerie, les programmes de gestion d'SMS, les gestionnaires de rendez-vous, les navigateurs, ...etc. Toutes ces applications sont écrites en Java.

III. 1.2- Framework Applicatif

C'est un ensemble de fonctions permettant aux applications d'exploiter au maximum les ressources de l'équipement, de communiquer avec le système et les autres applications (Accès à divers informations, exécution en tâche de fond, afficher des notifications dans la barre d'état, déclencher des alarmes, ...etc.). Ces fonctions sont les mêmes que celles utilisées par le système lui-même. Ce qui donne aux applications toute la puissance nécessaire.

III. 1.3- Les Bibliothèques

C'est un ensemble de bibliothèques développés en C/C++. Ces bibliothèques sont utilisées par divers composant d'Android dont le Framework applicatif.

Les composants majeurs de la plate-forme Android sont résumés sur le schéma suivant :

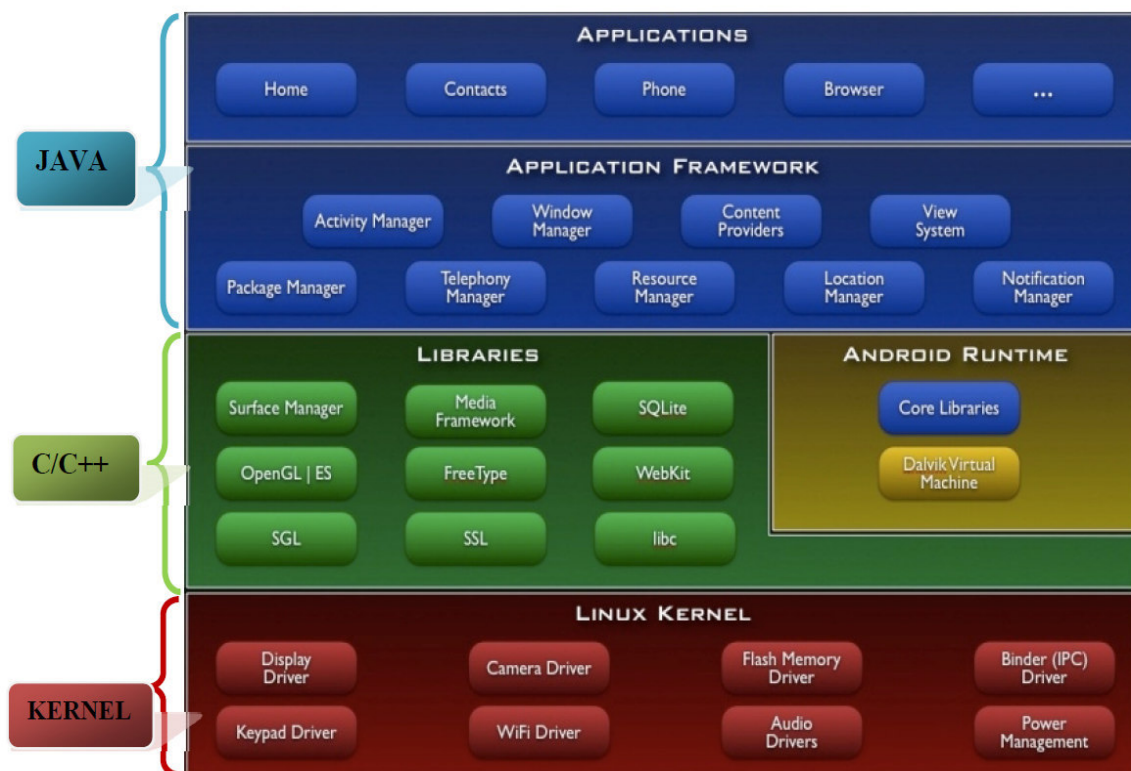


Fig. I.7. Architecture d'un système Android

III. 1.4- Environnement d'exécution Android

L'environnement d'exécution d'Android est la machine virtuelle Dalvik avec ses propres bibliothèques adaptées aux équipements mobiles. Dans la machine virtuelle Dalvik, les programmes sont écrits en Java et compilés avec les outils de java pour obtenir un byte code qu'on recompile encore avec un outil spécifique (dex) pour obtenir un code adapté à la machine Dalvik

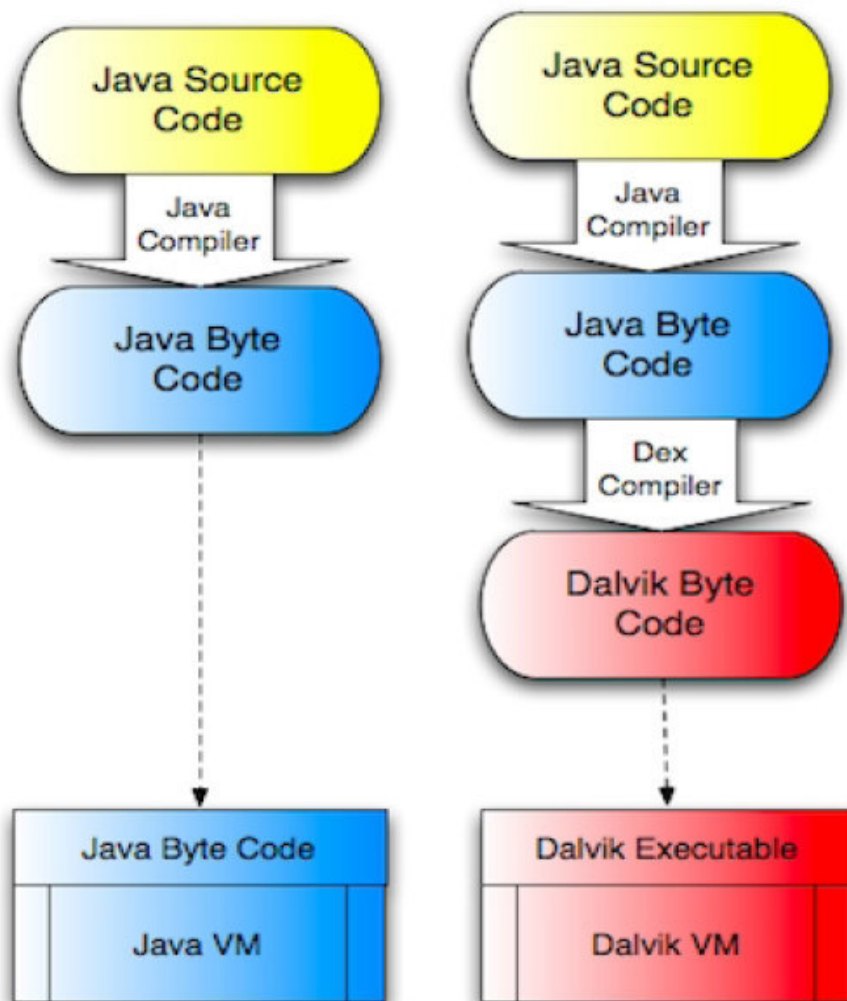


Fig. I.8. Différence entre la JVM et Dalvik

III. 1.5- Le noyau linux

Android repose sur le noyau Linux version 2.6 pour la gestion de la mémoire, des processus, et les pilotes des différents périphériques.

III. 2. Les concepts d'une application Android

Une application android est constituée d'un certain nombre de composants (Activity, service, Content Provider et Broadcast Receiver)

III. 2.1-L'activité (Activity)

Une activité est un composant fournissant un écran avec lequel les utilisateurs interagissent avec l'application. Quand on exécute n'importe quelle application Android on remarque qu'une première fenêtre s'ouvre à nous avec laquelle il est possible d'interagir (introduire ou extraire des données), elle possède donc une interface graphique. Dans certains cas, en appuyant sur un bouton par exemple ou un TextView cela nous mène vers une autre activité, et ainsi de suite. Cela est possible grâce au mécanisme d'Intent. Une application donc est un assemblage d'activités entre lesquelles il est possible de naviguer. Une activité remplit tout l'écran, une application ne peut en afficher qu'une seule à la fois. De plus, elle contient des informations sur l'état actuel de l'application. Ces informations s'appellent le Context. Toute activité peut passer à tout moment en arrière plan dans une pile d'attente lorsque son exécution est interrompu comme lors d'un appel entrant.

III. 2.1-1 .Etat d'une activité

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

- **Active (Resumed) :**

L'activité est visible en totalité. Elle est sur le dessus de la pile, c'est elle qui a le focus. C'est ce que l'utilisateur consulte et peut l'utiliser dans son intégralité et agir directement dessus.

- **Suspendue (Posed) :**

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

- **Arrêtée (Stopped) :**

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

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

onCreate () :

Est appelée au début de la création de l'activité et n'est appelée qu'une seule fois. Elle joue le rôle du constructeur en permettant d'initialiser des variables, affecter des listener...

onRestart () :

Appelée après un nouveau démarrage de la même activité (quand l'activité était arrêtée).

onStart () :

L'activité va devenir visible. Cette méthode sert à lancer les animations, ou généralement tout ce qui est liée à l'affichage graphique, car elle est également appelée lors d'un retour de focus sur l'activité (dans ce cas onRestart est appelé avant).

onResume () :

L'activité est maintenant visible, Cette méthode sera exécutée lorsque l'activité résume son exécution après la suspension (pause) et que l'activité commence à interagir avec l'utilisateur.

onPause () :

Méthode qui sert à arrêter une activité temporairement.

onStop () :

L'activité ne sera plus visible, cachée par une autre activité qui est en premier plan. Une activité stoppée est aussi en vie, elle est encore en mémoire mais elle n'est pas rattachée au gestionnaire des fenêtres du système Android. Elle peut être tuée par le système Android en cas de besoin en mémoires.

onDestroy() :

L'activité va être détruite. La destruction opère quand quelqu'un appelle cette méthode ou quand c'est le système qui décide de tuer l'activité pour économiser de l'espace.

III. 2.1-2 .Cycle de vie d'une activité

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

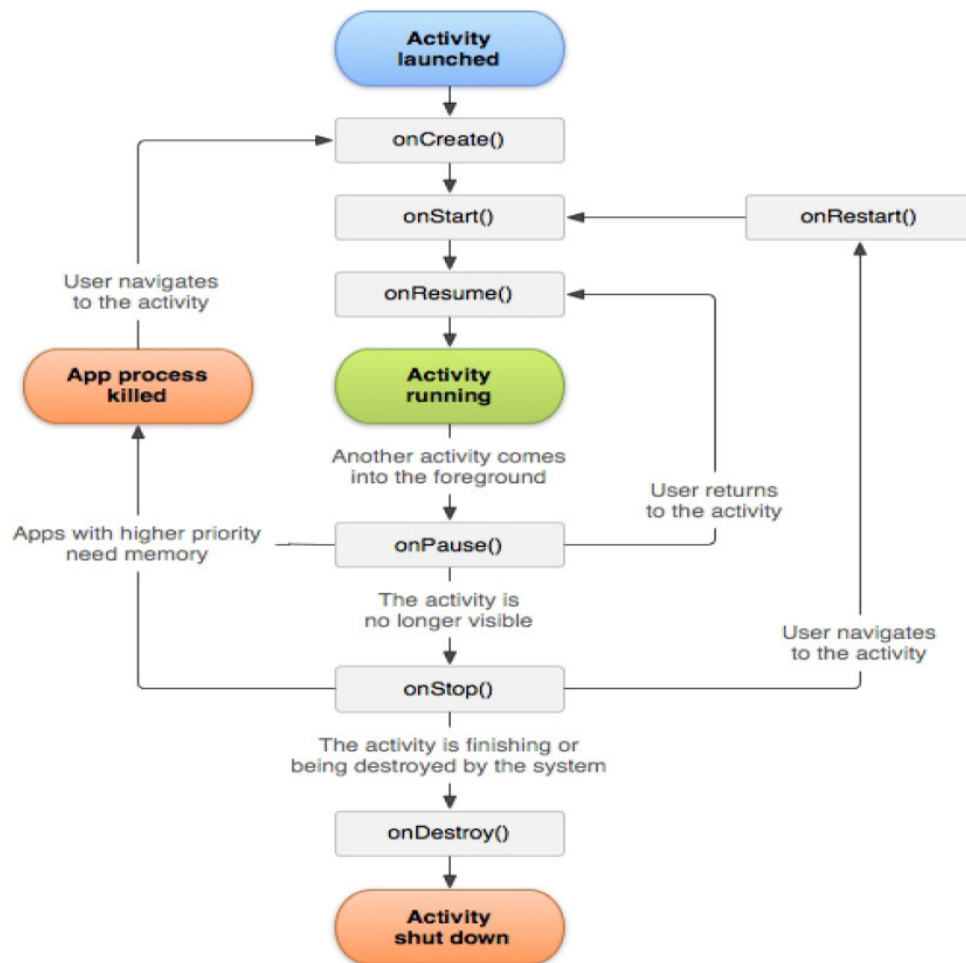


Fig. II.9.Cycle de vie d'une activité

III. 2.2 .Les Services

Les services sont des tâches qui peuvent être lancées avec ou sans intervention de l'utilisateur.

Elles s'exécutent en background de l'application et peuvent se terminer soit après la finalisation de la tâche, soit à travers une intervention externe. Les services représentent également une fonctionnalité d'une application exposée à d'autres applications. Il est important de mentionner que le service ne fournit pas d'interface graphique (User Interface).

Notre Player audio (lecteur de musique), par exemple, permet d'écouter la musique tout en consultant nos emails, etc.... Cette fonctionnalité n'est possible qu'à l'aide des Services.

III. 2.3 .Content Provider

Les fournisseurs de contenu offrent un moyen de partager des données entre les applications qui peuvent être stockées sur le système de fichier local, sur une base de données SQLite ou le réseau (FTP, Web, etc.). Les autres applications peuvent accéder à ces données (ou les modifier si l'application le permet) en utilisant un content provider.

Un content provider est une classe qui implémente un ensemble de méthodes standards permettant à d'autres applications de consulter et modifier les données prises en charge par ce content provider .Par exemple, le système Android fournit un content provider fournissant la liste des contacts téléphoniques qu'une autre application peut utiliser.

III. 2.4.Broadcast Receiver

Les broadcasters sont les diffuseurs d'évènements/messages via des intentions (intentions dites broadcast pour les différencier des intentions plus classiques). Les messages ainsi diffusés pourront être réceptionnés par plusieurs applications, les applications qui se seront abonnées à ces broadcasts (diffusions).

III. 2.5. Intent

Trois composants d'application sont lancés par les Intents : Les activités, Les services, Les receveurs de diffusion. Les Intents sont l'une des pierres angulaires de la plateforme Android. Nous pouvons les comparer à des actions ou même à des intentions, ils permettent de dialoguer à travers le système à partir de canaux qui leur sont dédiés. Quand le mobile reçoit un appel, la plateforme lance un Intent signalant l'arrivée d'un appel, de même pour un SMS. Nous les utiliserons dans notre application pour dialoguer à l'intérieur de celle-ci pour :

- Naviguer entre les activités.
- Surveiller les clicks sur l'AppWidget
- Lancer le service de téléchargement des données météorologiques

IV. Historique et versions d'Android

IV. 1.Open Handset Alliance

Afin de promouvoir ce nouveau système d'exploitation ouvert, Google a su fédérer autour de lui un consortium d'une trentaine d'entreprises. L'Open Handset Alliance (abrégé OHA) est un consortium Regroupant de grands constructeurs et développeurs de logiciels dans le but de développer des normes ouvertes pour les appareils de téléphonie mobile.

Ce consortium a été créé le 5 novembre 2007. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6.

IV. 2. Bugdroïde

Pour représenter Android la famille Google a utilisé un personnage nommé Bugdroid qui est un petit robot vert. 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.



Fig. I.10. Personnage Bugdroïde

IV. 3. Versions d'Android

Android a débuté avec la sortie de la version 1.0 en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces nom de codes suivent une logique alphabétique, en voici quelques-unes :

- **La version 2.3 (2.3.7) :**

Gingerbread (Pain d'épice), sortie le 6 décembre 2010, dernière version dédiée uniquement aux Smartphones. Cette version est parfois utilisée sur de petites tablettes.

Caractéristiques :

- Interface utilisateur mise à jour
- Support des grands écrans à résolutions extra-larges (WXGA et plus)
- Support de la VoIP et SIP

- Support des formats vidéo WebM/VP8, et l'encodage audio AAC
- Nouveaux effets audio tels que la réverbération, l'égalisation, la virtualisation du casque audio et accentuation des graves

- **La version 4.0.3 :**

(Aussi appelée Ice Cream Sandwich) unifie le développement des interfaces Smartphone, tablette, télévision connectée et système embarqué. Avant la version 4.0, le développement d'une application pour une tablette n'était pas compatible à un Smartphone à cause de la taille d'écran différente.

Il offre désormais la possibilité d'avoir les touches : retour, maison et applications récentes sur l'écran, éliminant ainsi le besoins de touches physiques.

Caractéristiques :

- Nouvelles APIs pour les développeurs (notification des mises à jour des contacts par les fournisseurs de services.
- Amélioration de l'accessibilité pour les lecteurs d'écrans
- Diverses modifications de l'interface de l'appareil photo
- Amélioration du calendrier (API diffuseurs de services)

- **La version 4.2:**

(Aussi appelée Jelly Bean) introduit photo sphère permettant une prise des photos à 360° type Street View, un système multi-compte sur tablette uniquement.

Caractéristiques :

- Corrige le bug qui ne faisait pas apparaître le mois de décembre dans la gestion des contacts
- Ajout du support pour les gamepads et joysticks Bluetooth

- **La version 4.4 : (Aussi appelée Android KitKat) :**

C'est le nom d'une version du système d'exploitation Android qui a comme numéro 4.4. Son nom vient de la « tradition » de nommer les différentes versions d'Android par un nom de dessert ou pâtisserie et d'un partenariat avec Nestlé qui possède la marque de biscuits chocolatés Kit Kat. Pendant longtemps il avait été pensé que la version d'Android devant succéder à la 4.3 serait la 5.0 Key Lime Pie

Caractéristiques :

- Nouvelle interface translucide
- Enregistrement séquence vidéo de l'écran
- Amélioration du système de notification
- Gestion système des sous-titres
- Amélioration des performances

- **La version 5.0 :**

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

Caractéristiques :

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

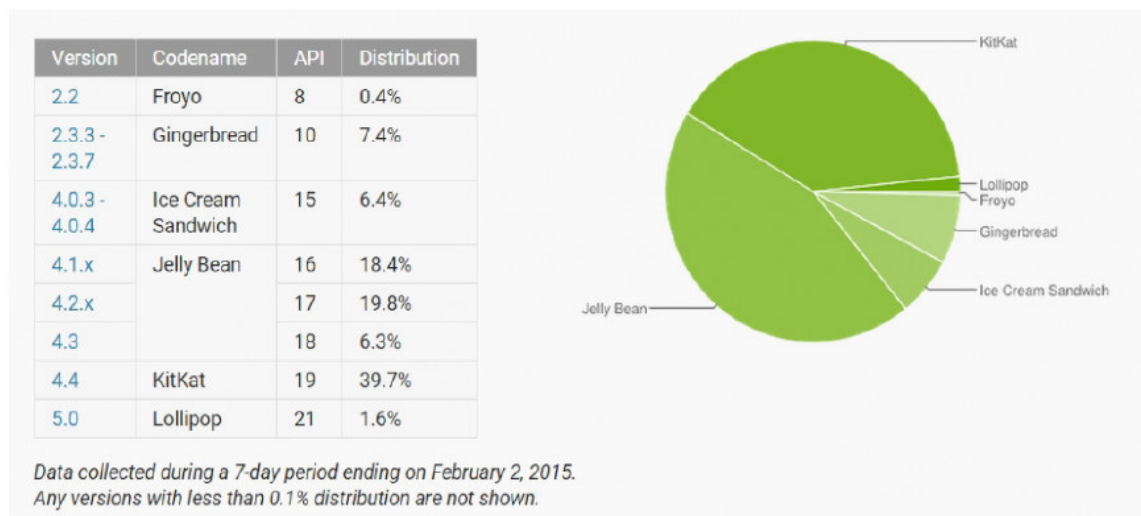


Fig. I.11. Répartition des versions d'Android

V. Différents équipements utilisant Android

Le succès d'Android est indéniable, les fournisseurs l'ont bien compris, d'où l'apparition sur le marché de plus en plus d'appareils utilisant ce système d'exploitation :

V. 1. Les Smartphones :

Est un téléphone intelligent, disposant d'un écran tactile, fonctionnant sur des réseaux haut débit et offrant de nombreuses fonctionnalités, dont la consultation du courrier électronique et la localisation par GPS. Le premier mobile commercialisé sous Android est le HTC Dream/G1 produit par la firme Taïwanaise HTC, lancé aux États-Unis le 22 octobre 2008. Depuis le lancement en 2007 de l'iPhone d'Apple, la progression de leurs ventes est spectaculaire et elles ont dépassé en 2012 celles des téléphones mobiles classiques.

V. 2. Les Tablettes

En septembre 2010, plusieurs marques comme Samsung ont utilisé le système Android pour les tablettes (version 2.2 Fro Yo)

Android a été ainsi porté sur d'autres appareils comme la HP TouchPad, le portage a été réalisé début 2012 basé sur la version Ice Cream Sandwich, et fin 2011 avec la version 7 de CyanogenMod basé sur Gingerbread.

V. 3. Les télévisions :

Le 5 avril 2010, la première télévision sous Android est dévoilée. Celle-ci est développée par l'entreprise suédoise People of Lava et se nomme Scandinavia. Elle possède les différentes applications connectant aux réseaux sociaux (facebook.....), elle possède un navigateur Web ainsi qu'un client de messagerie électronique. (ENIE)

V. 4. Montre :

Google a créé une version spécifique Android Wear, une version modifiée d'android dédiée au Smartphones, LG commercialise sa G Watch et Samsung une Gear Live, Motorola a aussi dévoilé sa montre sous le système Android.

V. 5. Consoles de jeux vidéo :

Une console de jeux vidéo portable sous Android portant le nom GamePad est commercialisée depuis septembre 2012.

Conclusion

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

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

Dans le chapitre suivant, nous parlerons des agrégateurs de contenu ainsi leurs fonctionnements

Les agrégateurs de contenus

Introduction

En technologies web, un logiciel d'agrégateur tresse plusieurs fils de syndication provenant de flux RSS différents en même temps. Il prévient de la mise à jour d'un site web ou des actualités qu'il publie (par notification sonore, visuelle, etc.). Également, il importe le nouveau contenu en question et il le fait pour un ensemble de sites.

Le terme est aussi utilisé de manière plus générale pour une activité tendant à regrouper plusieurs grandeurs (ou flux) en un seul, dans le but de mieux gérer les ressources comme les ressources énergétiques mais cela est en dehors du cadre de notre travail.

I. Description d'un agrégateur

L'agrégateur est une sorte de « facteur » qui va chercher le courrier à l'extérieur, puis le dépose chez l'utilisateur, dispensant ce dernier d'aller régulièrement aux nouvelles en visitant de nombreux sites internet. Il fonctionne un peu comme une messagerie électronique (quasiment en temps réel) mais (contrairement à un client de messagerie), l'utilisateur d'un agrégateur est souvent limité à la lecture passive des messages reçus (le « fil » de syndication). Il ne peut pas « répondre » aux éléments reçus. Il existe quelques exceptions dans le cas de billets blogs, où certains agrégateurs permettent de poster des commentaires. Un agrégateur ne peut traiter qu'une information spécialement structurée, par une technologie particulière.

Les sources de contenu (des sites web en général) offrent l'adresse d'un fil de syndication mis à jour plus ou moins régulièrement. Cette première phase, dite « syndication de contenu » structure les données pour l'agrégateur. L'agrégation consiste à s'abonner à un ou plusieurs de ces fils de syndication. L'agrégateur détecte leurs mises à jour et avertit aussitôt l'utilisateur, sans qu'il ait à visiter périodiquement les sites internet diffusant les fils de syndication auxquels il s'est abonné. Chaque fil est associé à un dossier dans l'agrégateur, dossier qui contient les différentes entrées du fil, le plus souvent par ordre chronologique inverse (les plus récentes entrées en premier). La détection de nouveaux éléments dans un fil est périodique, ou réalisée à la demande de l'utilisateur qui peut quand il le souhaite mettre à jour tout ou partie de ses abonnements.

En général, un agrégateur permet de visualiser une liste des fils enregistrés, classés alphabétiquement ou par thématique. Pour chaque fil, les n-derniers éléments sont listés (n choisi par l'utilisateur ou fixé). Pour chaque élément (billet, article...) l'agrégateur affiche soit son résumé, soit son contenu complet. De ce fait, l'utilisateur peut être amené à quitter son agrégateur pour lire le contenu sur le site d'où il a été tiré, ou bien en faire l'entière lecture dans son logiciel.

II. Interface d'accès aux services

Le stockage des fils se fait de deux façons ; on peut utiliser des logiciels fonctionnant en local, à installer ou bien en s'inscrivant sur des sites de gestion et de partage de fils. Dans ce cas l'agrégateur est une application web, non un logiciel classique. Cela évite d'avoir à utiliser un logiciel de création de flux spécifique à chaque système d'exploitation (Feed Editor, RSS Wizard, Feed Mix sous Windows) qui nécessitent tous que le fichier XML soit déposé en ligne par le protocole FTP pour être accessible aux internautes par le protocole HTTP. Ils s'inscrivent dans le concept de « travail à distance » initié par WebDAV puis les systèmes de gestion de contenu utilisant fortement le concept de clients légers en s'appuyant sur la technologie AJAX(JavaScript).

III. RSS



II.1.logo des flux rss

RSS désigne une famille de formats XML utilisé pour la syndication de contenu Web. Ce standard est habituellement utilisé pour obtenir les mises à jour d'information dont la nature change fréquemment, typiquement cela peut être des listes de tâches dans un projet, des prix, des alertes de toute nature, des nouveaux emplois proposés, les sites d'information ou les blogs. Les Podcasts et videocasts sont conçus sur ce même standard en utilisant la balise 'Enclosure'. Pour les recevoir, l'utilisateur doit s'abonner aux fils RSS, ce qui lui permet de consulter rapidement les dernières mises à jour, à l'aide d'un agrégateur, sans avoir à se rendre

sur le site. Trois formats peuvent être désignés par ces initiales :

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.90 et 1.0)
- Really Simple Syndication (RSS 2.0)
- On parle aussi souvent de RSS pour désigner également le format Atom

III. 1. Domaines d'utilisation

Les flux RSS sont utilisés dans la diffusion d'alertes, de nouvelles ou de listes (au sens large). On trouve de nombreuses applications professionnelles. Le standard RSS est notamment utilisé pour la diffusion d'actualités sur Internet par les blogs professionnels ou

semi-professionnels. Des annuaires répertorient ainsi, un grand nombre d'actualités. Ceux-ci peuvent généralement être lus grâce à des lecteurs en ligne. Plusieurs navigateurs peuvent également lire les flux RSS, notamment Maxthon, Mozilla Firefox (d'origine, ou avec les extensions Witz RSS News Reader, infos ou Sage), ou encore Opéra et, depuis sa version 7, Internet Explorer, ainsi que Mac OS X, Safari et Camino.

III. 2. Type d'un fichier RSS

Un fichier RSS se présente sous la forme de la figure ..

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Mon site</title>
    <description>Ceci est un exemple de flux RSS 2.0</description>
    <lastBuildDate>Sat, 07 Sep 2002 00:00:01 GMT</lastBuildDate>
    <link>http://www.example.org</link>
    <item>
      <title>Actualité N°1</title>
      <description>Ceci est ma première actualité</description>
      <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
      <link>http://www.example.org/actu1</link>
    </item>
    <item>
      <title>Actualité N°2</title>
      <description>Ceci est ma seconde actualité</description>
      <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
      <link>http://www.example.org/actu2</link>
    </item>
  </channel>
</rss>
```

Figure II.2 : La démarche adoptée pour la modélisation.

Le contenu se situe toujours entre les balises <rss> et </rss>. Elles possèdent obligatoirement un attribut version qui spécifie la version à laquelle le document RSS est conforme.

Au niveau suivant se trouve une unique balise <channel> qui contiendra les métadonnées du flux RSS, obligatoires ou non, ainsi que la liste des contenus.

III. 4. Métadonnées

En ce qui concerne les métadonnées, trois éléments sont obligatoires dans un channel:

- `<title>` : Définit le titre du flux ;
- `<description>` : Décrit succinctement le flux ;
- `<link>` : Définit l'URL du site correspondant au flux.

D'autres éléments optionnels existent comme :

- `<pubDate>` : Définit la date de publication du flux
- `<lastBuildDate>` : Définit la dernière date de modification du flux
- `<image>` : Permet d'insérer une image dans le flux
- `<language>` : Définit la langue du flux.
- `<enclosure>` : Permet d'insérer un média (vidéo,...) dans le flux

III. 5. Description de chaque article

Pour chaque article, une balise `<item>` est ajoutée dans notre document.

Dans cette balise se trouvent les données correspondant à l'actualité sous forme de balise. Les balises les plus courantes sont :

- `<title>` : Définit le titre de l'actualité ;
- `<link>` : Définit l'URL du flux correspondant à l'actualité ;
- `<pubDate>` : Définit la date de l'actualité ;
- `<description>` : Définit une description succincte de l'actualité ;
- `<guid>` : Définit de manière unique l'actualité.

Selon la DTD RSS 2.0, il doit y avoir au moins un `<title>` ou une `<description>` dans un `item` et le reste des balises est optionnel.

D'autres balises existent comme :

- `<author>` : Définit l'adresse électronique (mail) de l'auteur ;
- `<category>` : Associe l'item à une catégorie ;
- `<comments>` : Définit l'URL d'une page de commentaire en rapport avec l'item ;

Il peut également arriver que l'on trouve d'autres balises dans un flux RSS, grâce aux espaces de noms (*namespaces*). On peut par exemple inclure de nouvelles fonctionnalités comme cela a été fait pour iTunes. Voir concernant ce sujet la spécification RSS 2

III. 6. Modules

L'objectif premier de tout module RSS est d'étendre le schéma XML de base pour une souscription au contenu plus robuste du contenu. La modulation permet d'ébaucher des transactions plus diverses, mais standardisées, sans modifier les spécificités de base du RSS.

Pour accomplir cette extension syntaxique, un vocabulaire étroitement contrôlé (en termes RSS, un "module"; en termes XML, un "schéma") est déclaré en amont pour donner des noms aux concepts et établir des relations entre ces concepts.

IV. Applications mobiles pour lire les flux RSS

Les flux RSS divisent le monde des utilisateurs Android en deux groupes : ceux qui les connaissent, et ceux qui ne savent pas de quoi il s'agit. C'est pourtant une des innovations les plus importantes en termes d'information sur internet, c'est un moyen de se tenir informé qui gagne à être connu. La lecture de contenus d'actualité sur votre téléphone peut s'avérer être un vrai casse tête lorsque vous vous intéressez à différents sites de news. Même si certains d'entre eux proposent des applications permettant de parcourir leurs actualités (lemonde.fr ; slate.fr, etc.), la lecture n'en est pas pour autant agréable, et c'est sans parler des sites qui n'ont pas d'application dédiée. Les flux RSS constituent donc la meilleure façon d'agréger l'information intéressante et de la consulter simplement et rapidement

Conclusion

Les agrégateurs de contenu permettent de faciliter la vie aux utilisateurs en téléchargeant automatiquement du contenu sélectionné.

Avec la mobilité des utilisateurs, il devient, de plus en plus opératif d'avoir ces agrégateurs sur les équipements mobiles, d'où notre projet de création d'un agrégateur sur un client Android

Pour que le client Android se connecte à une base de données MySQL, la solution la plus réponde est d'écrire et d'exécuter des scripts PHP à l'aide d'un Protocole http. Ensuite coder les données dans le format JSON, afin de récupérer les données par le client Android. Le client Android, le serveur web et le serveur de base de données forment une architecture 3 tiers.

Dans le chapitre suivant, nous présenterons une analyse pour les cas d'utilisation de notre système.

Analyse et Conception

Partie 1 : Analyse

Introduction

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

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

I. Les Méthodes de conception

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

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

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

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

II. Présentation de l'UML

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

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

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

II. 1. Modélisation avec l'UML

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

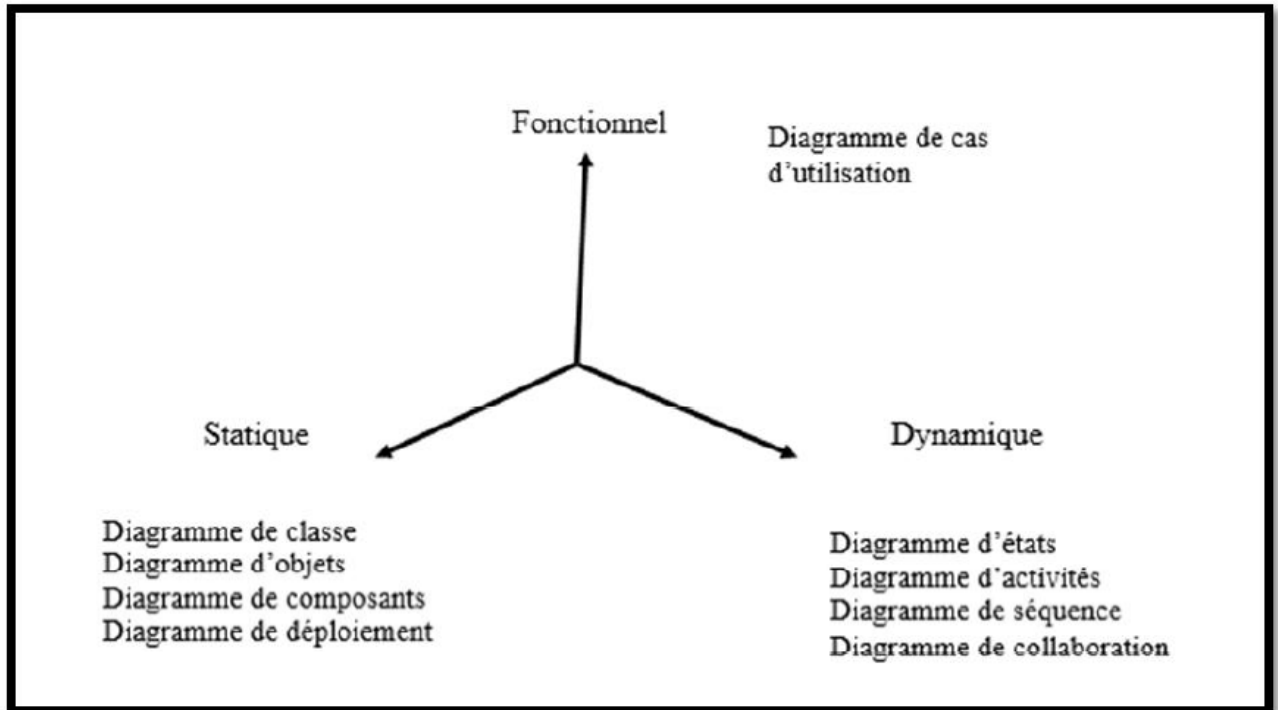


Figure III.1 : Les diagrammes de modélisation

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

- ❖ Une démarche itératif et incrémentale : Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maîtriser l'inconnu.
- ❖ Une démarche guidée par les besoins des utilisateurs : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.
- ❖ Une démarche centrée sur l'architecture logicielle : c'est la clé de voute de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

II. 2. La démarche de modélisation avec l'UML

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

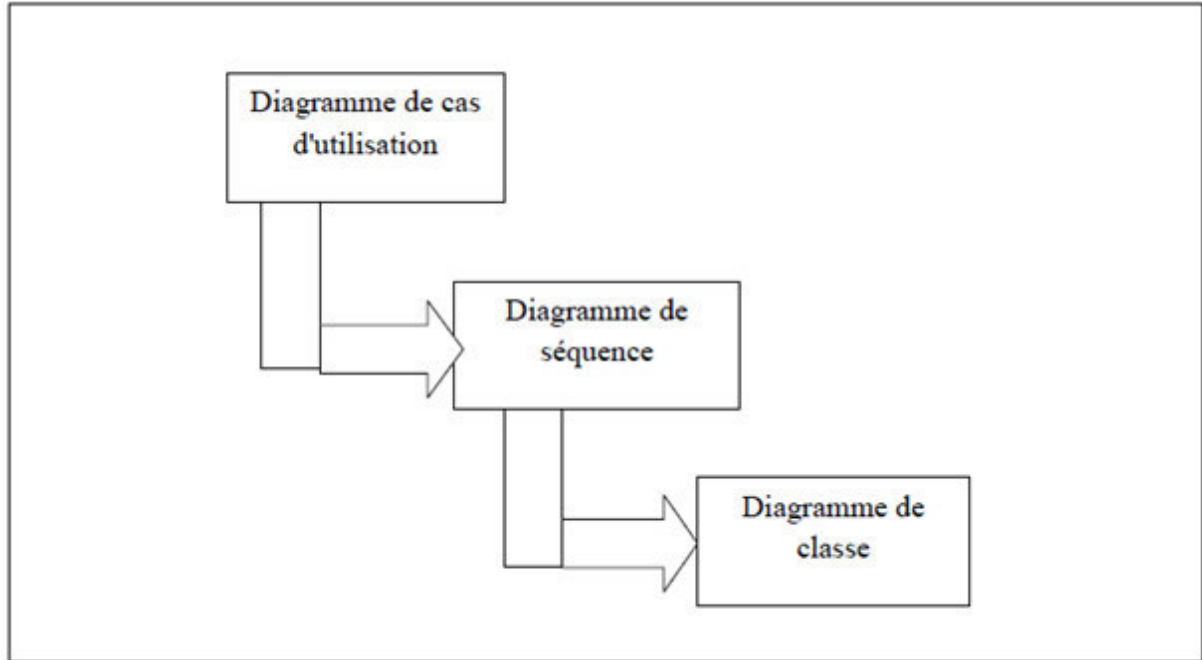


Figure III.2 : La démarche adoptée pour la modélisation.

III. 1. Identification des acteurs

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

Les acteurs de notre système sont :

- 👤 User : toute personne qui utilise une machine (Téléphone, Tablette, etc..) qui marche sous le système Android 2.1 ou une version ultérieure.
- 👤 Administrateur : Celui qui gère les drivers .

III. 2. Identification des besoins

L'objectif principal de notre application est lire les flux RSS des journaux pour que le client accède à l'actualité.

Les fonctionnalités de notre application sont :

- Affiche l'actualité selon votre localisation (votre ville, votre wilaya, votre pays).
- Rechercher rapide par mots clés dans toute l'actualité.
- Recherche avancée par titre, catégorie, nom du journal, auteur et mots clés.

III. 4. Spécification des tâches

III. 4.1-Définition

Une tâche est l'ensemble des différentes fonctions qui peuvent être accédés pas un acteur bien spécifiée.

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

Acteur	Tache
Administrateur	T0 : Authentification T1 : Activer les drivers T2 : Désactiver les drivers
User	T3 : Lancer l'application T4 : Voir les articles T5 : Voir les articles selon la localisation (GPS) T6 : Faire une recherche simple T7 : Faire une recherche avancée

Tableau III.1 : Spécification des taches

III. 2. Spécification des scénarios

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

Acteur	Tache	Scénario
Administrateur	T0 : Authentification T1 : Activer les drivers T2 : Désactiver les drivers	S0 : enter le pseudo et le mot de passe S1 : se connecter S2 : cliqué sur l'icône activer S3 : cliqué sur l'icône désactiver
User	T3 : Lancer l'application T4 : Voir les articles T5 : Voir les articles selon la localisation (GPS) T6 : Faire une recherche simple T7 : Faire une recherche avancée	S4 : cliqué sur l'icône Algérie Actu S5 : Accéder a tout les articles S6 : cliqué sur l'icône GPS S7 : Introduire les mots clés, recherchés S8 : Introduire un ou plusieurs éléments de la recherche, recherchée

Tableau III.2 : Spécification des scénarios

IV. Les cas d'utilisation

Une représentation d'un ensemble spécifique d'actions qui sont réalisées par le système et qui produit un résultat observable par un acteur en particulier. Elle permet de représenter l'ensemble des besoins et des exigences que notre application doit respecter.

Dans notre cas nous distinguons beaucoup de cas d'utilisation en voici certains :

IV. 1. Cas d'utilisation s'identifier à l'espace Administrateur

S'identifier a l'espace Administrateur

Scenario : S0, S1

Acteur : Administrateur

Résumé : cette fonctionnalité permet d'accéder a son propre compte (Compte administrateur)

Description

- 1- Lancer le site
- 2- Le système affiche la page principal
- 3- Saisir le login et mot de passe
- 4- Cliquer sur Ok pour valider le formulaire
- 5- Le système affiche l'interface administrateur ou un message d'erreur

Fig. III.3.Cas d'utilisation « s'identifier à l'espace Administrateur»

IV. 2 .Cas d'utilisation User

Scenario : S4, S5

Acteur : User

Résumé : cette fonctionnalité permet d'accéder à tous les articles

Description

- 6- Lancer l'application
- 7- Le système affiche la liste de tous les articles

Fig. III.4.Cas d'utilisation «Interface principal des utilisateurs».

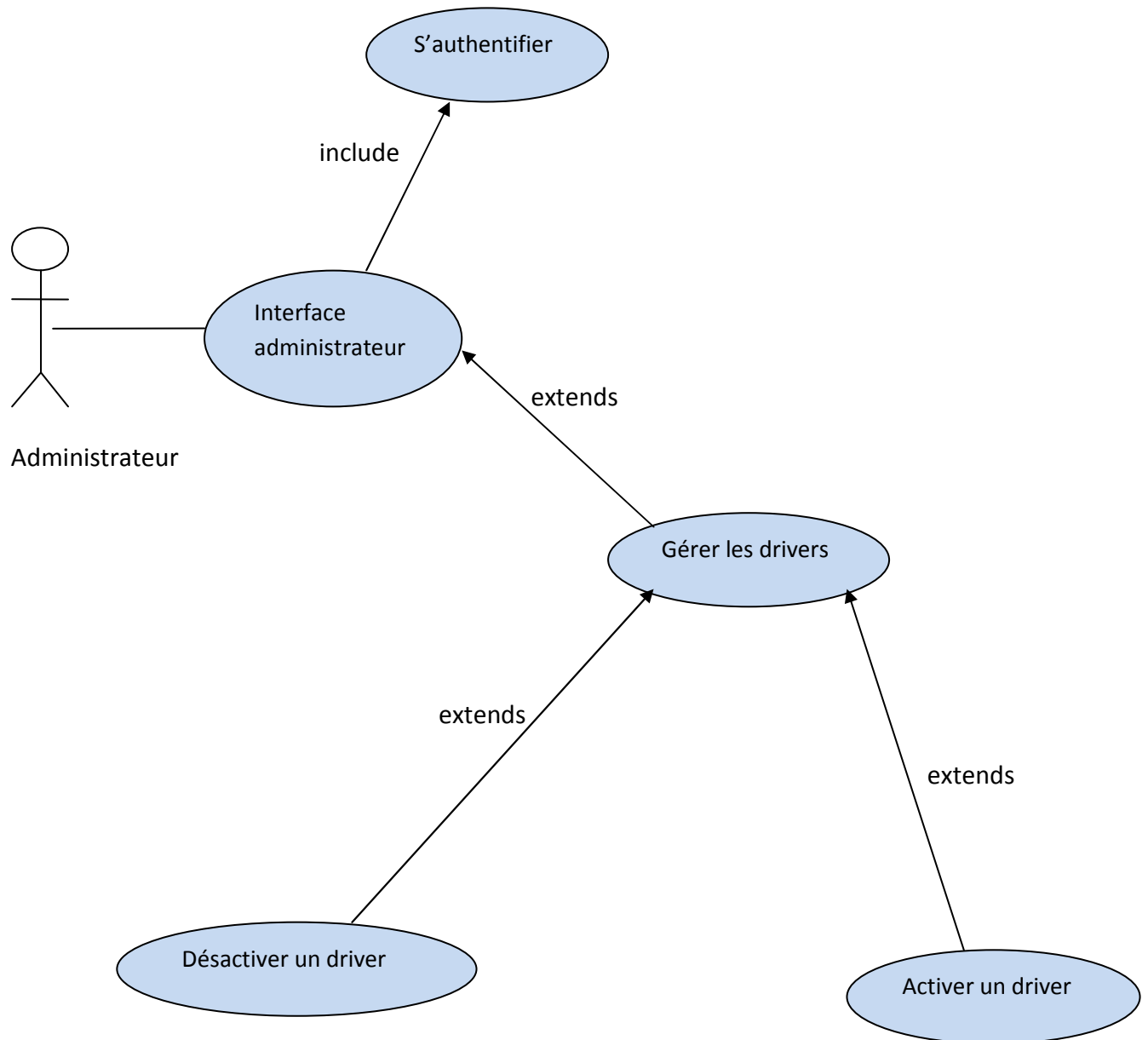
IV. 3. Diagramme de cas d'utilisation

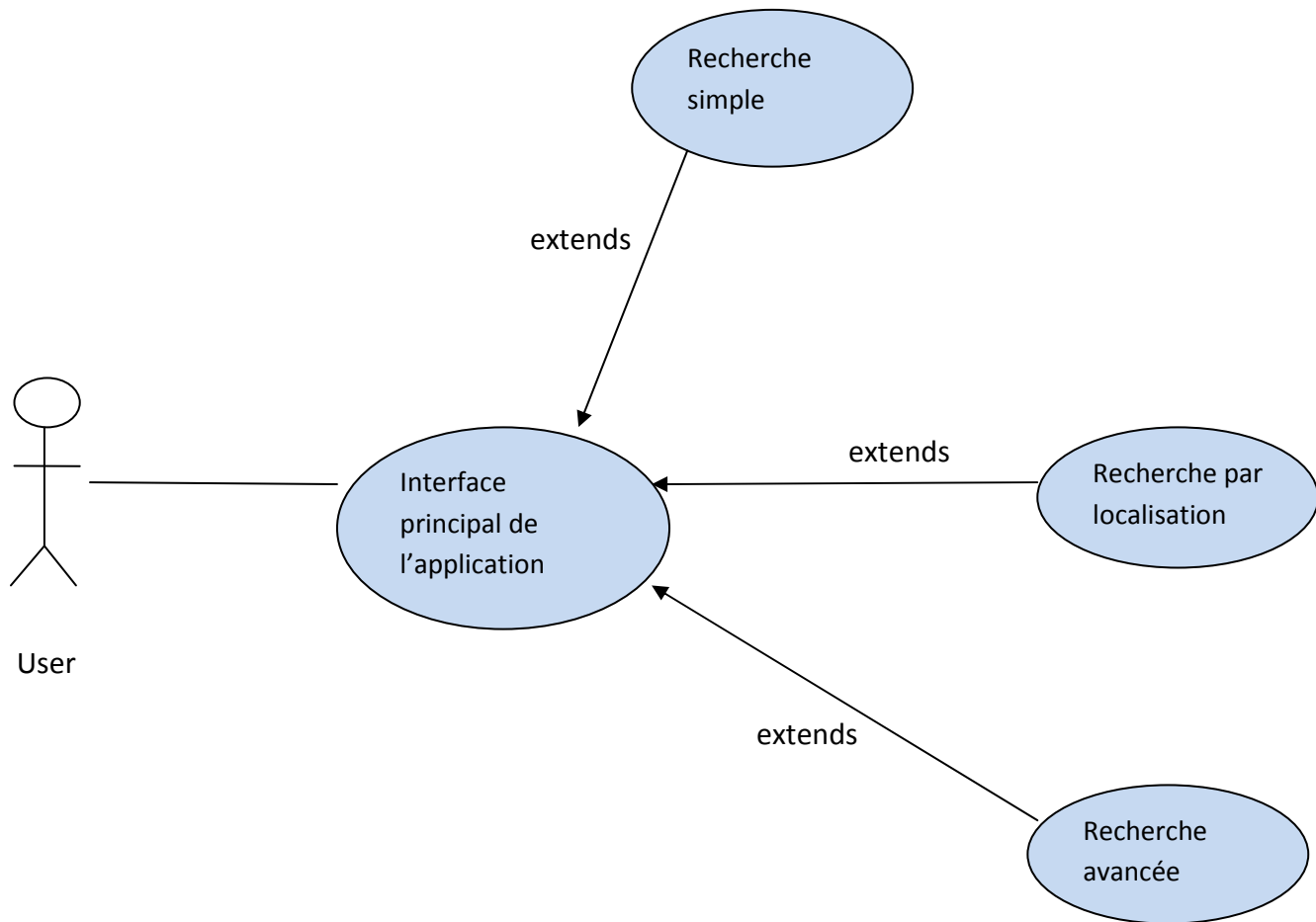
UML définit une notation graphique pour représenter les cas d'utilisations. Cette notation est appelée diagramme de cas d'utilisation qui fait intervenir les acteurs ainsi que les relations qui existent entre eux et les cas d'utilisations eux-mêmes.

Les diagrammes de cas d'utilisations décrivent sous forme d'actions et de réactions le comportement fonctionnel de l'application modélisée.

Il existe trois types de relations standards entre cas d'utilisations qui sont proposés par UML:

- « include » le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié.
- « extend » le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.
- Généralisation : les cas d'utilisations descendant héritent des propriétés de leurs parents.

IV. 3.1-Diagramme de cas d'utilisation « Administrateur» (Coté serveur)**Fig. III .5« Diagramme de cas d'utilisation pour l'administrateur ».**

IV. 3.2-Diagramme de cas d'utilisation «User» (Coté mobile)**Fig. III .6« Diagramme de cas d'utilisation pour un utilisateur».**

A la fin de cette partie nous avons pu couvrir les différents besoins fonctionnels et non fonctionnels des acteurs de notre système.

Nous avons fourni une analyse plus détaillée de ces besoins grâce à un diagramme de cas d'utilisation relatif aux acteurs réagissant avec le système, qui sont les propriétaires du téléphone Android.

Dans la partie qui suit, nous présenterons la conception de l'architecture de notre système.

Partie 2 : Conception

Introduction

A travers cette partie, nous décrivons la conception effectuée pour réaliser convenablement le travail demandé.

Pour ce faire, nous choisissons de donner en premier lieu une idée sur les composants d'une application Android afin de comprendre sa structuration et par la suite nous détaillons la conception de notre application au moyen des diagrammes de classes et de séquences.

Après avoir parlé de la conception générale, nous choisissons dans un second lieu le modèle d'architecture adéquat pour la réalisation.

I. 1. Conception générale

I. 1.1-Les composants d'une application Android

Une application Android consiste en un assemblage de composants liés via un fichier de configuration, qui présentent en quelque sorte les briques sur lesquelles se repose l'application. Ces concepts fondamentaux à préciser sont :

❖ Les vues

Les Views sont les composants basiques de l'interface graphique. Ce sont les éléments de l'interface que l'utilisateur voit et sur lesquels il agit. C'est de la classe View qu'héritent les widgets (exemple de composants graphiques tel que les boutons), les layouts (le plan sur lequel on organise et on place les composants graphiques) et tous les composants graphiques servant à la création d'une interface graphique interactive.

❖ Les contrôles

C'est bien la classe des composants graphiques cités plus haut. Les contrôles sont tels que les boutons, les champs de saisie de texte, les cases à cocher, etc.

❖ Les activités

Une Activity représente la fenêtre qui sera affichée à l'utilisateur. C'est un ensemble de vues et de contrôles composant une interface logique. Elle permet également de gérer des fonctionnalités telles que l'appui sur une touche, l'affichage de messages, etc. Ce concept repose essentiellement sur l'interaction de l'utilisateur avec l'écran.

❖ Les ressources

Chaque application Android a ses propres fichiers ressources. C'est dans ces fichiers que seront puisés les textes, les images, les couleurs, etc.

❖ Le fichier de configuration

C'est fichier auto généré par l'application, qui lui est indispensable. C'est un fichier XML appelé « AndroidManifest » qui décrit le point d'entrée de l'application (le code à exécuter), les composants du projet ainsi que les permissions nécessaires pour l'exécution du programme.

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

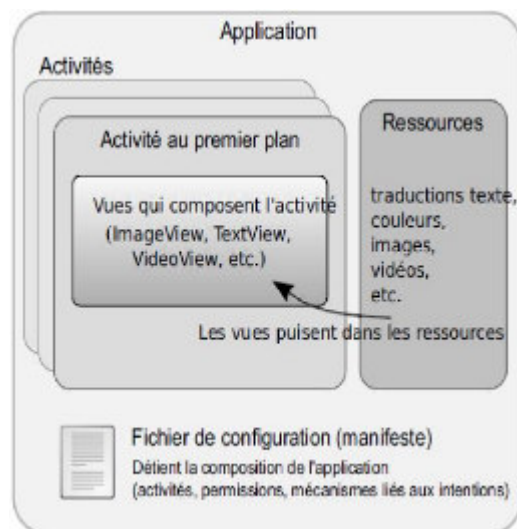
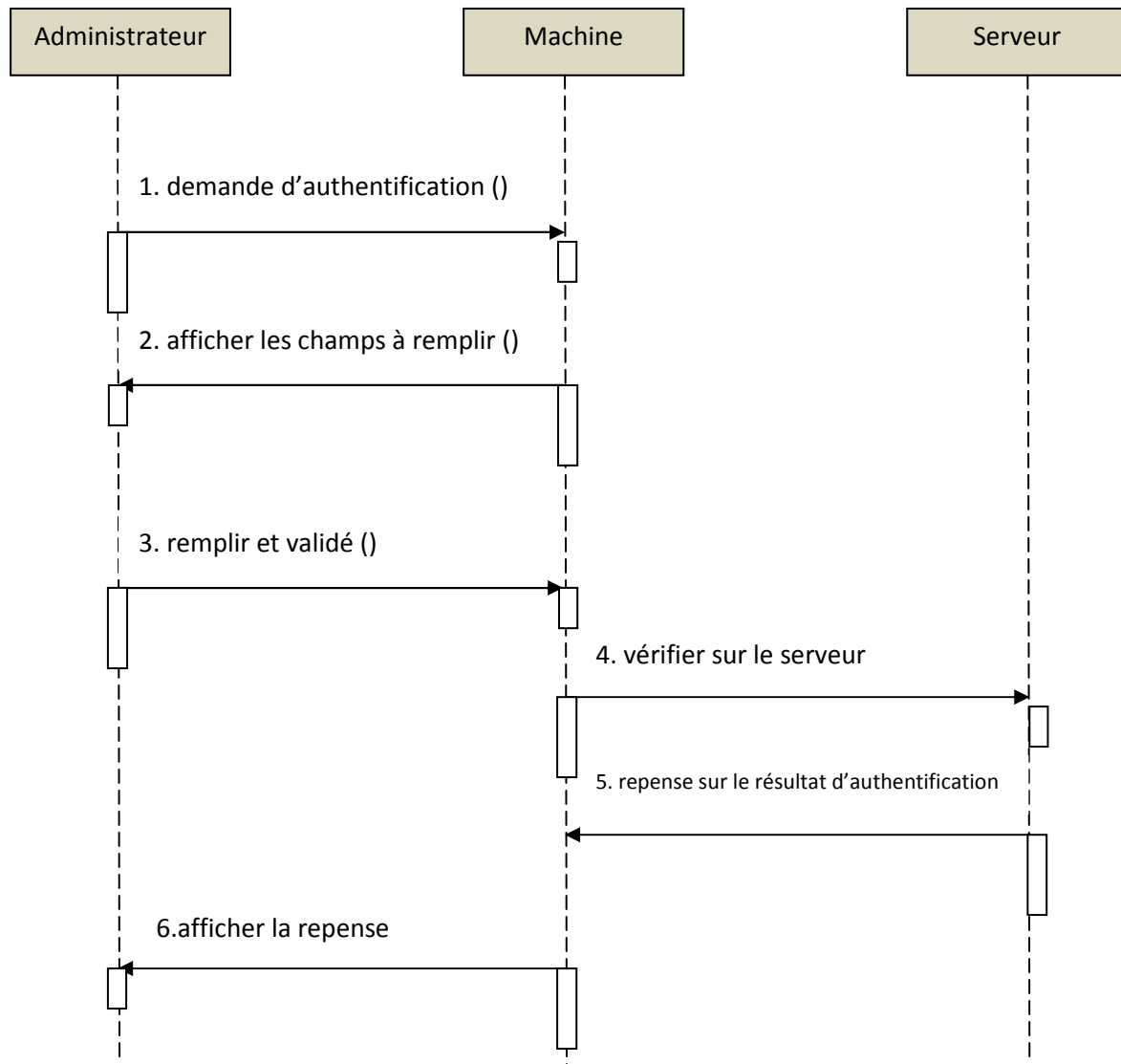


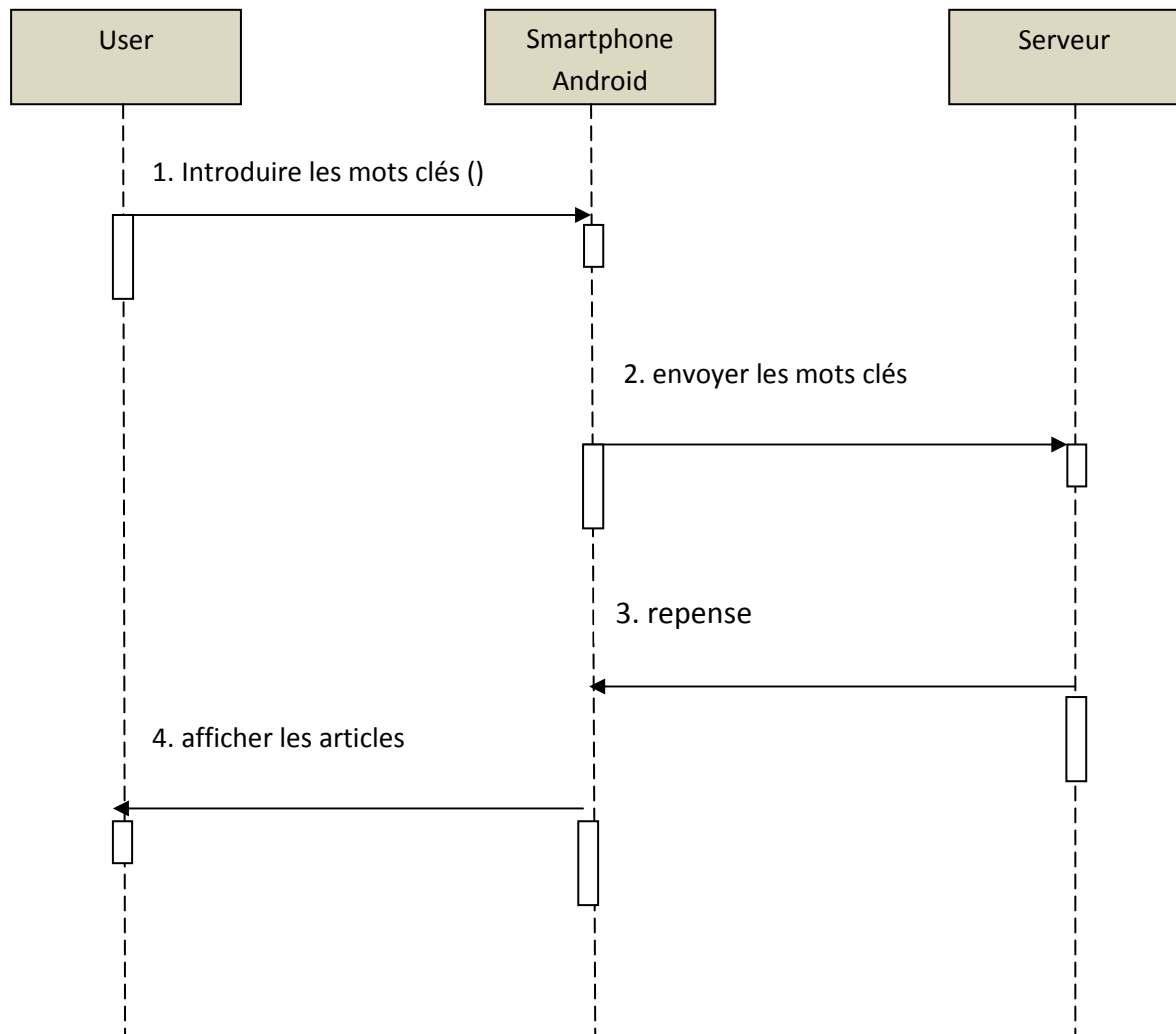
Fig. III .7 Composition d'une application Android.

II. Conception détaillée

II. 1. Diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par les acteurs. Il montre les interactions entre les objets, en montrant les messages qu'ils échangent entre eux, ordonnés dans le temps.

II. 1.1-Authentification (Coté serveur)**Fig. III.8: Diagramme de séquence du cas d'utilisation «Authentification»**

II. 1.2-Recherche simple (Coté mobile)**Fig. III.9: Diagramme de séquence du cas d'utilisation «Recherche simple»**

II. 1.3-Recherche avancée (Coté mobile)

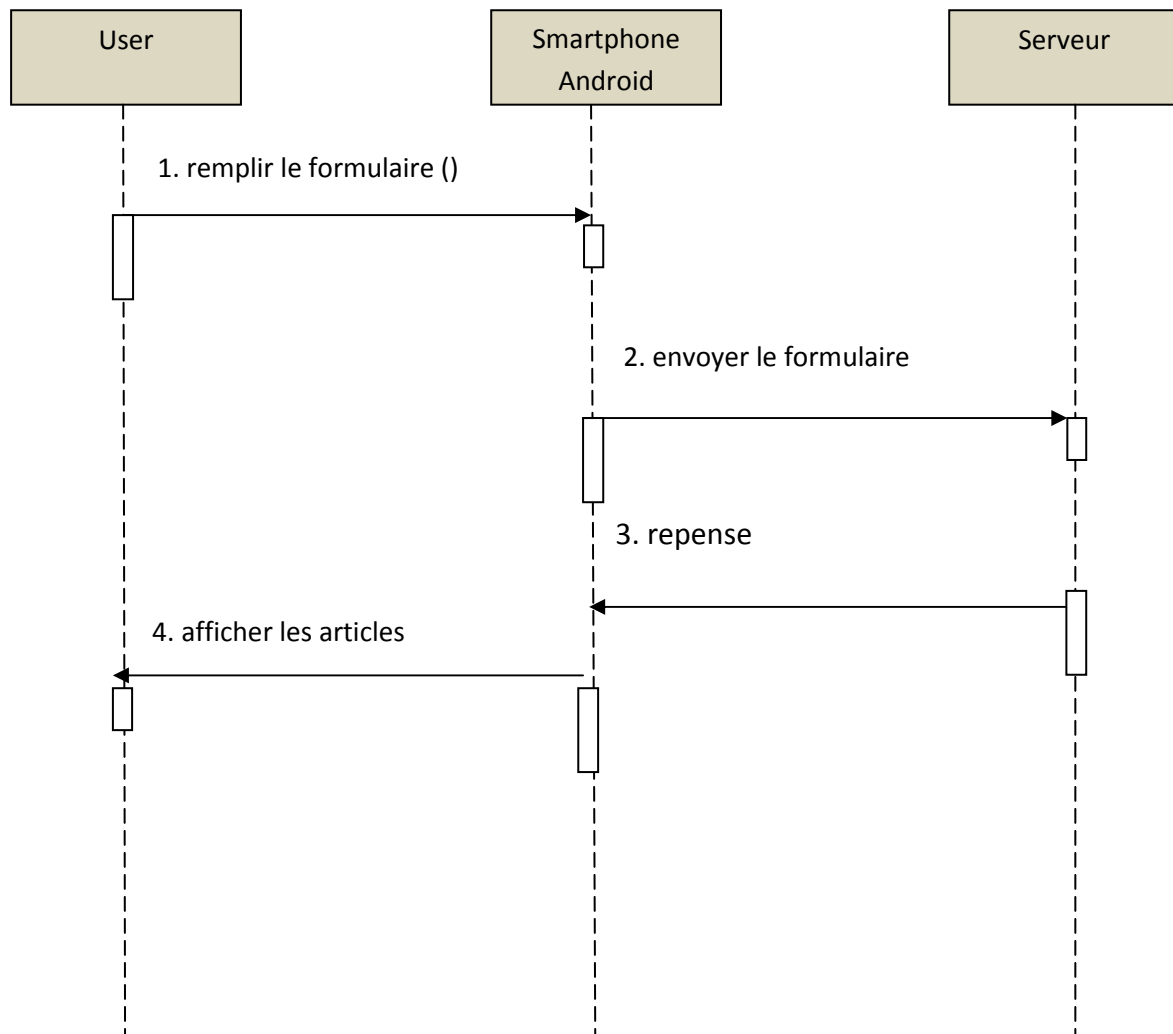


Fig. III.10: Diagramme de séquence du cas d'utilisation «Recherche avancées»

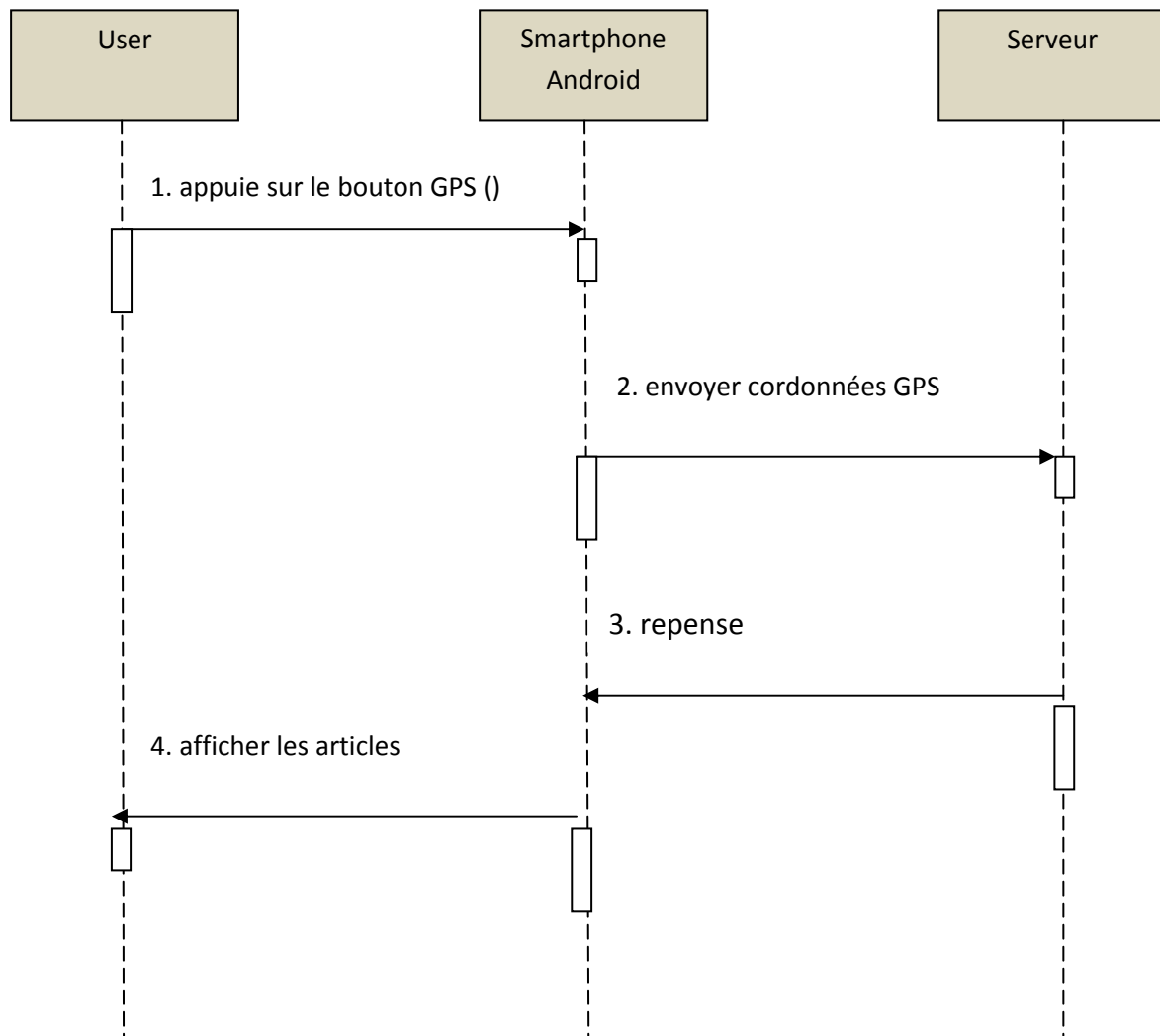
II. 1.4-Recherche par localisation (Coté mobile)

Fig. III.11: Diagramme de séquence du cas d'utilisation «Recherche par localisation»

III. Diagramme de classe général

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

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

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

Ce qui suit, nous présentons le diagramme de classe général de notre système

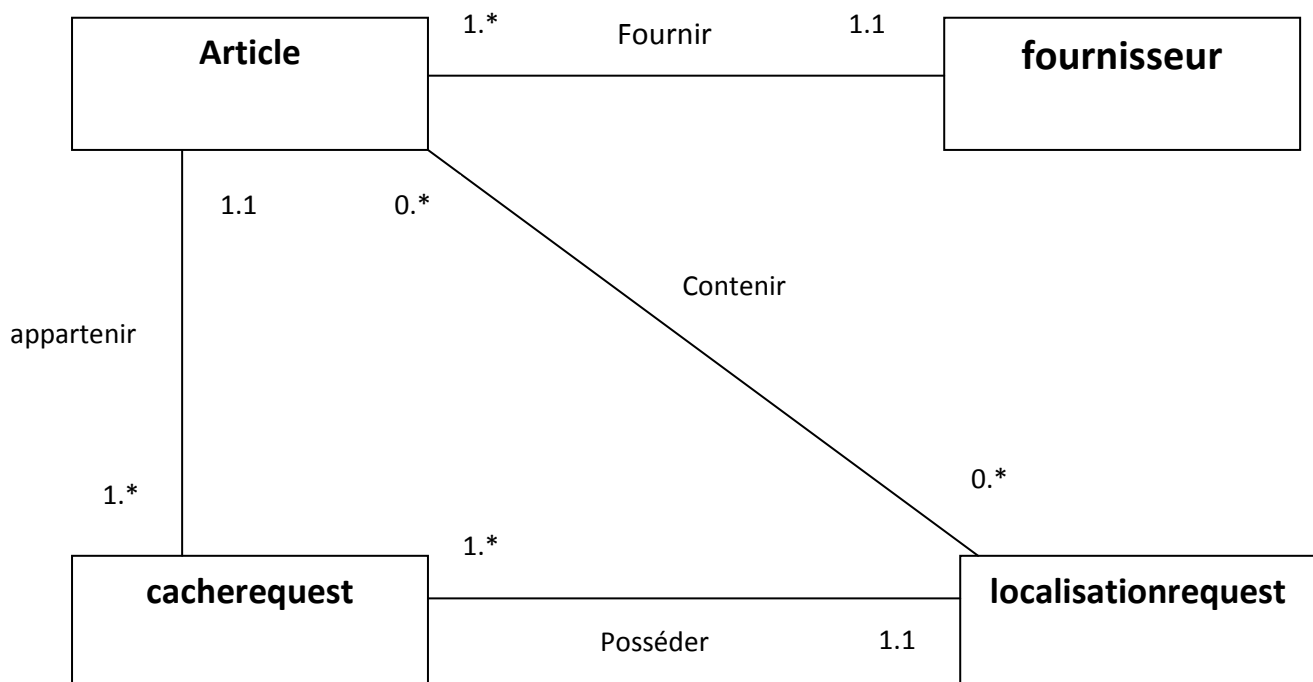


Fig. III.12. Diagramme de classe général

Conclusion

Dans ce chapitre, nous avons introduit les fonctionnalités de notre application. Pour les atteindre, nous avons proposé une solution fondée sur une analyse et une conception modélisées à l'aide du langage de modélisation unifié UML. Pour cela, nous avons défini les acteurs de notre application, les tâches qu'ils assurent et les scénarios associés à chaque tâche.

Nous avons élaboré les diagrammes de cas d'utilisation global, élaboré les diagrammes de séquence et enfin le diagramme de classe générale

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

Réalisation

Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. Nous commençons par présenter l'environnement matériel et logiciel, ensuite nous présenterons les différentes interfaces de l'application.

I. Environnement matériel et logiciel

I. 1. Environnement matériel

I. 1.1-Architecture matérielle

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

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

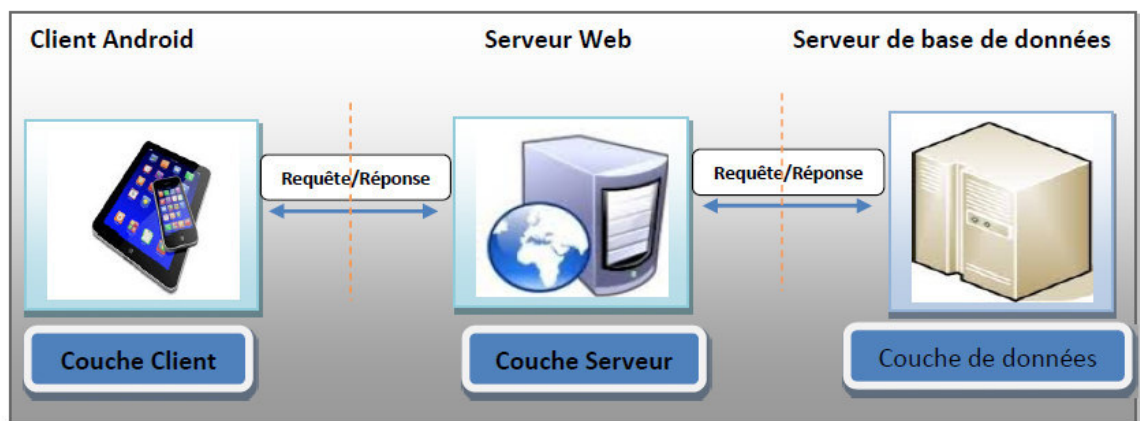


Fig. IV.1 : Architecture matérielle du système

I. 1.2-Matériels utilisés

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

- HP Intel core i3 2.20 GHz.
- 8 Go de mémoire vive.
 - Windows 7 - 64 bits.
- Un Smartphone Condor pour réaliser les tests.

I. 1.3-Technologie

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




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

Tableau IV.1 : Les différentes technologies utilisées dans l'application

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

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

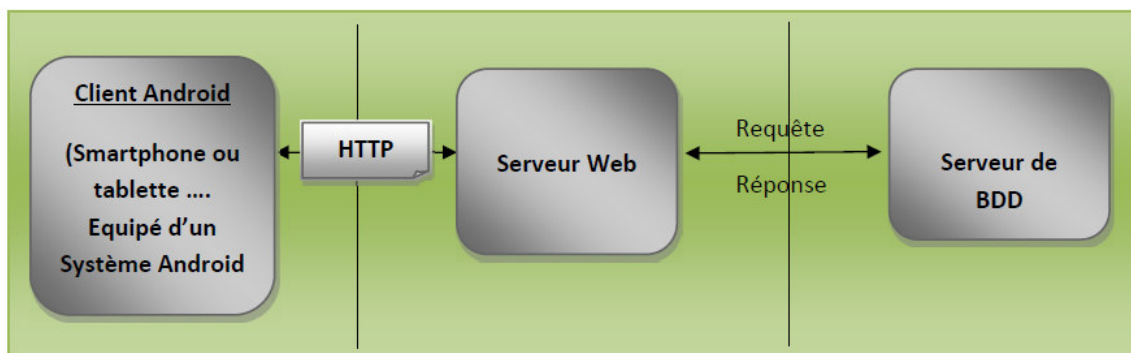


Fig. IV.2 : Architecture 3-tiers du point de vue technologique

I. 2. Environnement logiciel

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

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

I. 2.1-PHP

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

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

I. 2.2-Yii Framework

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



Fig. IV.3 : Logo framework yii

II. 2.3-PhpStorm

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

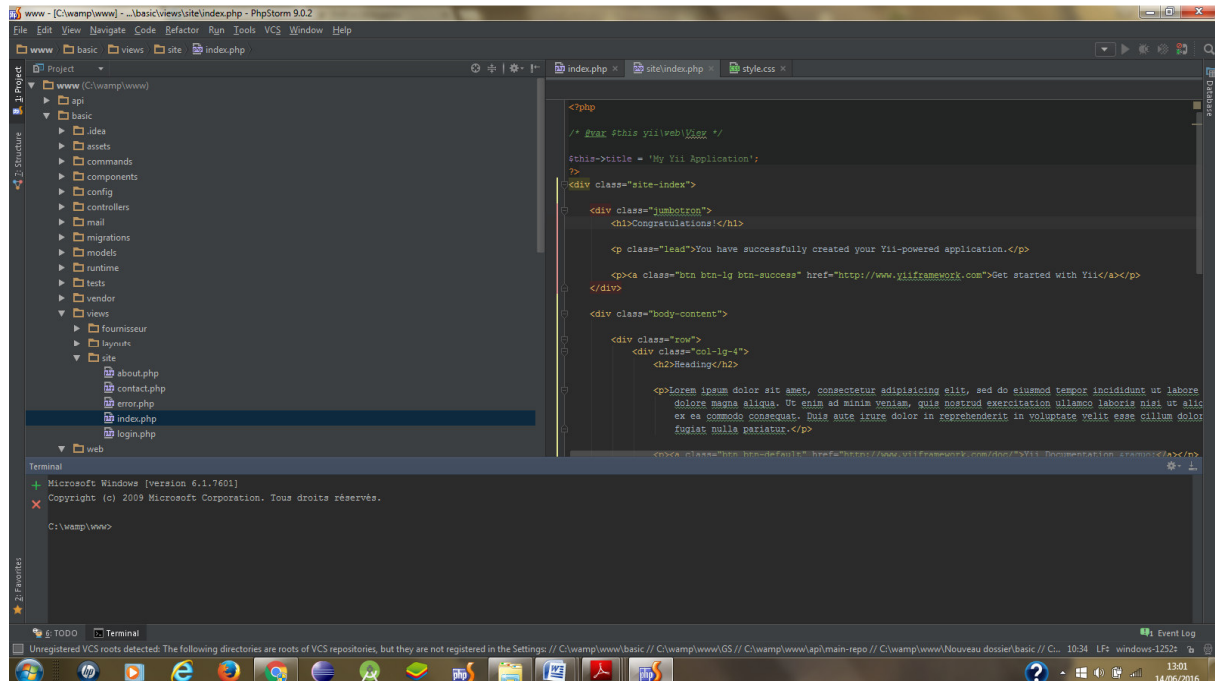


Fig. IV.4 : Interface de l'environnement PhpStorm

I. 2.3-JAVA

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

I. 2.2-XML

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

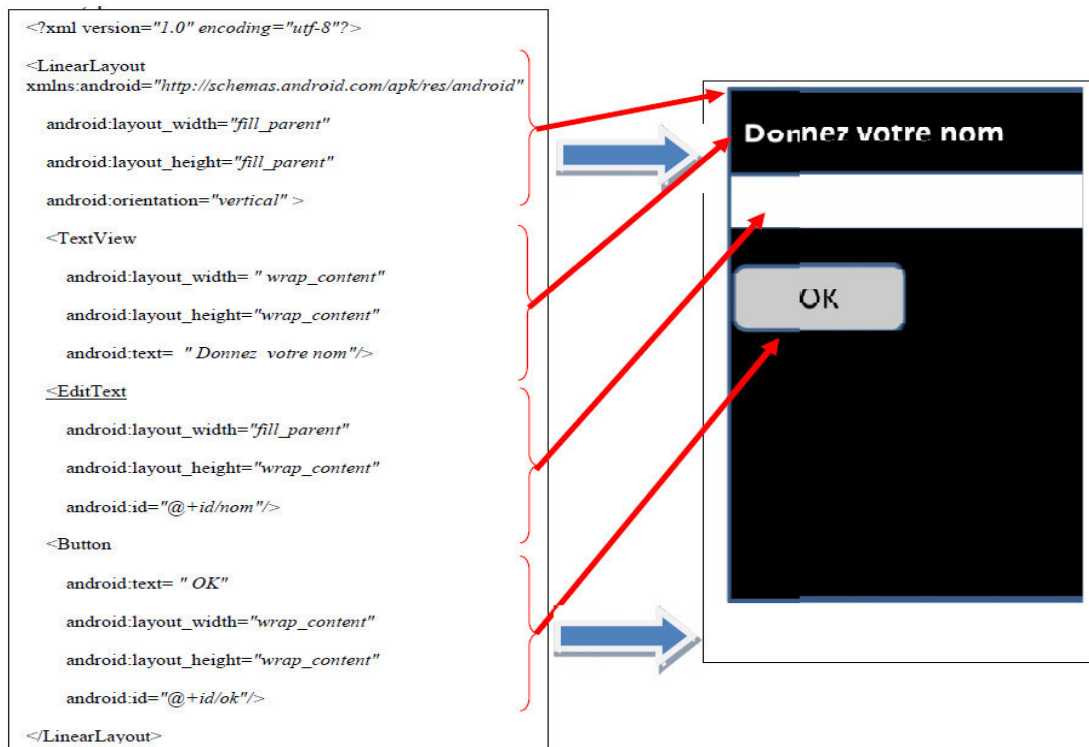


Fig. IV.5.Exemple d'un fichier XML

I. 2.3-Android Studio

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

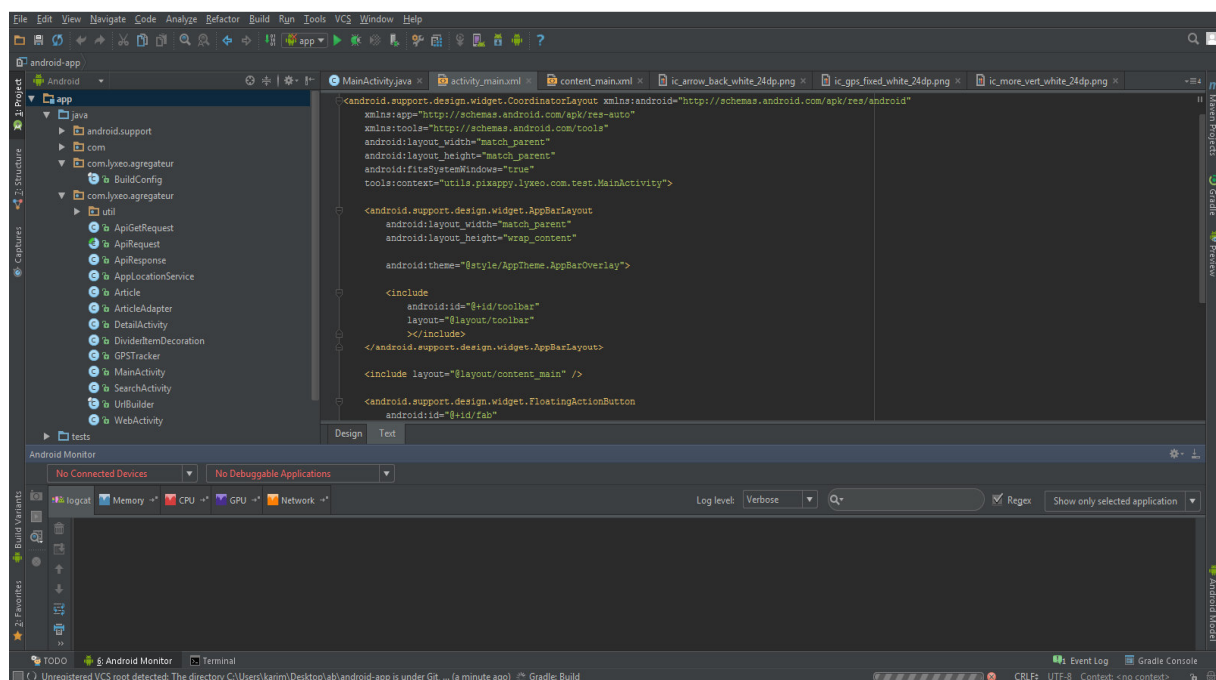


Fig. IV.6 : Interface de l'environnement Android Studio 1.1.0

I. 2.4-SDK

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

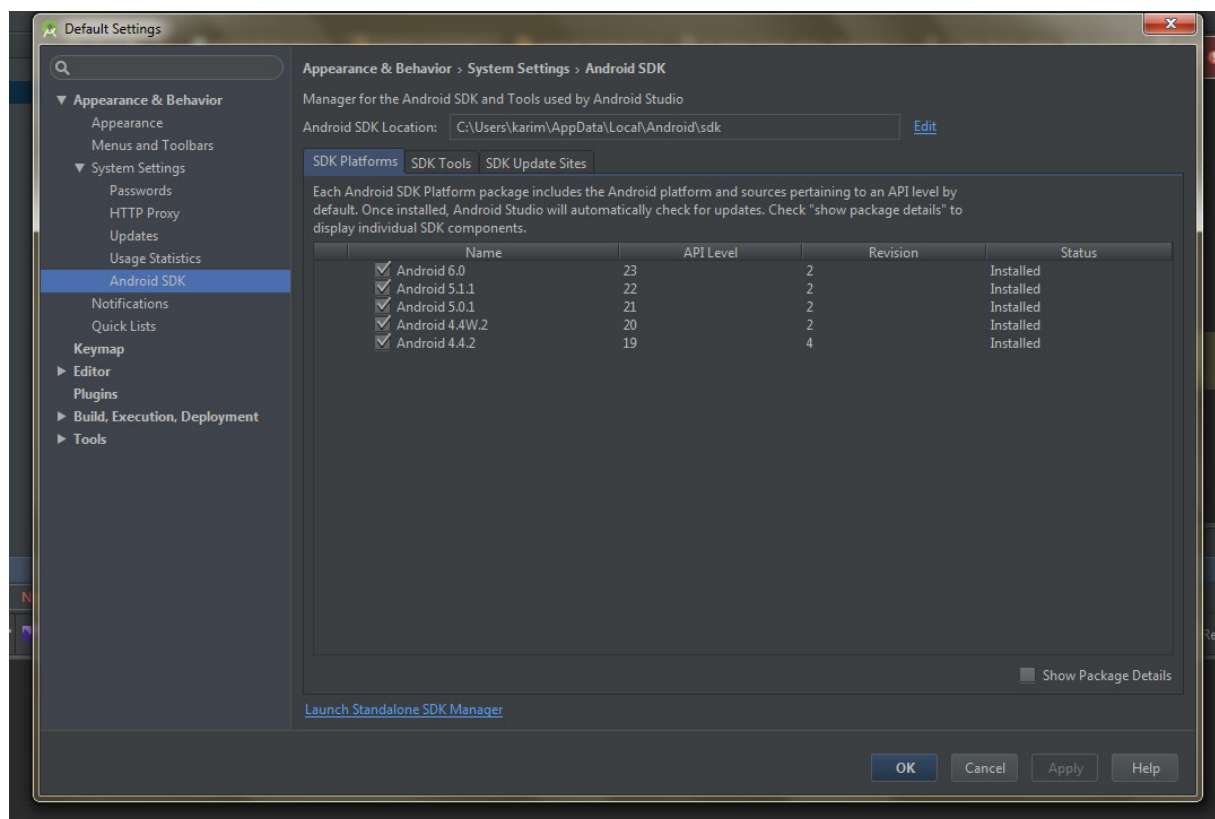


Fig. IV.7 : Interface de l'Android SDK Manager

I. 2.5-Git

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

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

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

Quelques commandes de Git :

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

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

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

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

II. Protocole et format de données

II. 1. Protocole de communication

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

II. 2. Format de données communiquées

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

Un document JSON ne comprend que deux éléments structurels :

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

Ces mêmes éléments représentent 3 types de données :

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

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

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

III. Interface Homme/Machine de l'application:

Dans ce qui suit nous présenterons les différentes interfaces de l'application en citant les détails de chaque capture d'écran.

III. 1. Interface « serveur Web » gestion des fournisseurs

The screenshot displays a web application interface for managing suppliers. The page title is 'Fournisseurs'. There is a 'Create Fournisseur' button. Below the button, it says 'Showing 1-12 of 12 items.' The table has three columns: '#', 'Nom', and 'Nomdriver'. The table contains 12 rows of data. Each row has a number in the first column, a name in the second column, a driver name in the third column, and three action icons (edit, delete, and a power icon) in the fourth column.

#	Nom	Nomdriver	
1	EiWatan	app\models\api\drivers\EiWatanDriver	
2	Liberté	app\models\api\drivers\LiberteDriver	
3	EnNahar	app\models\api\drivers\EnNaharDriver	
4	Echorouk	app\models\api\drivers\EchoroukDriver	
5	Aps	app\models\api\drivers\ApsDriver	
6	EiWatani	app\models\api\drivers\EiWataniDriver	
7	Ouarsenis	app\models\api\drivers\OuarsenisDriver	
8	Reflexion	app\models\api\drivers\ReflexionDriver	
9	La Depechede la Kabylie	app\models\api\drivers\DepechedeKabylieDriver	
10	JeuneIndependant	app\models\api\drivers\JeuneIndependantDriver	
11	SawtAlahrar	app\models\api\drivers\SawtAlahrarDriver	
12	WaktEldjazair	app\models\api\drivers\WaktEldjazairDriver	

Fig. IV.8 : Interface de gestion des fournisseurs

III. 2. Interface « Client Android» Accueil

Au premier lancement de l'application c'est l'interface principale qui sera visible, elle s'affiche lors du lancement de notre application, elle permet l'affichage de tous les articles

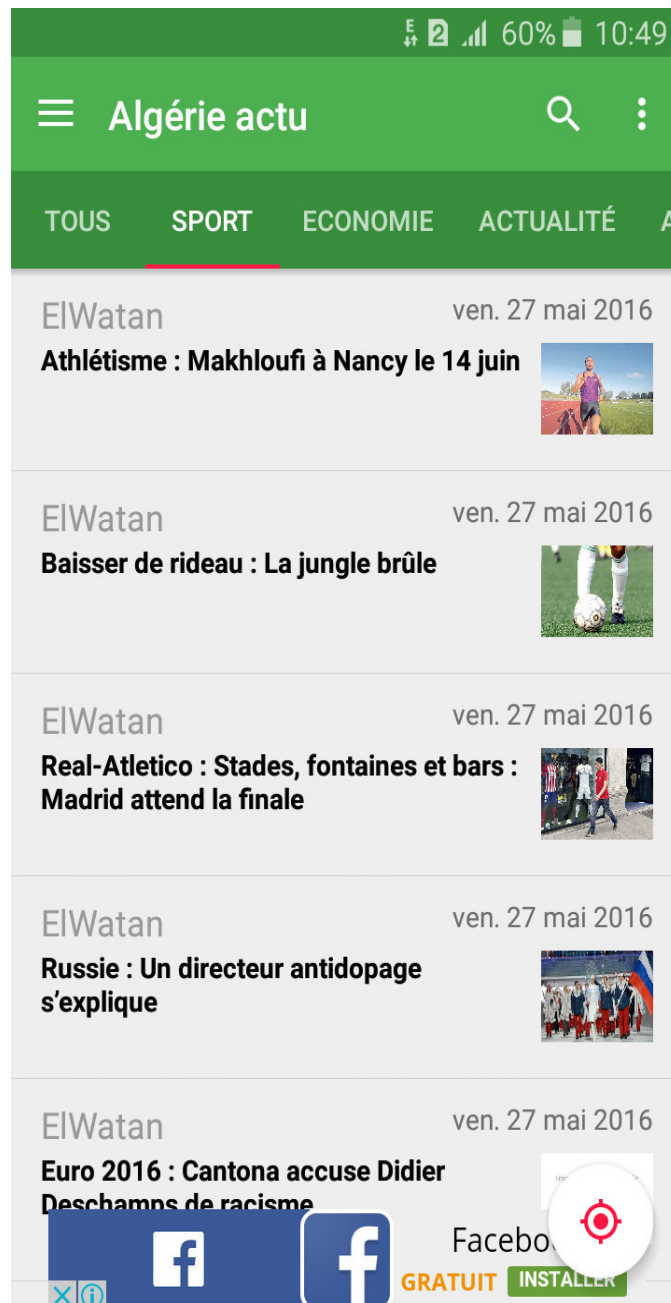


Fig. IV.9 : Interface accueil

III. 2. Interface recherche par mots clés

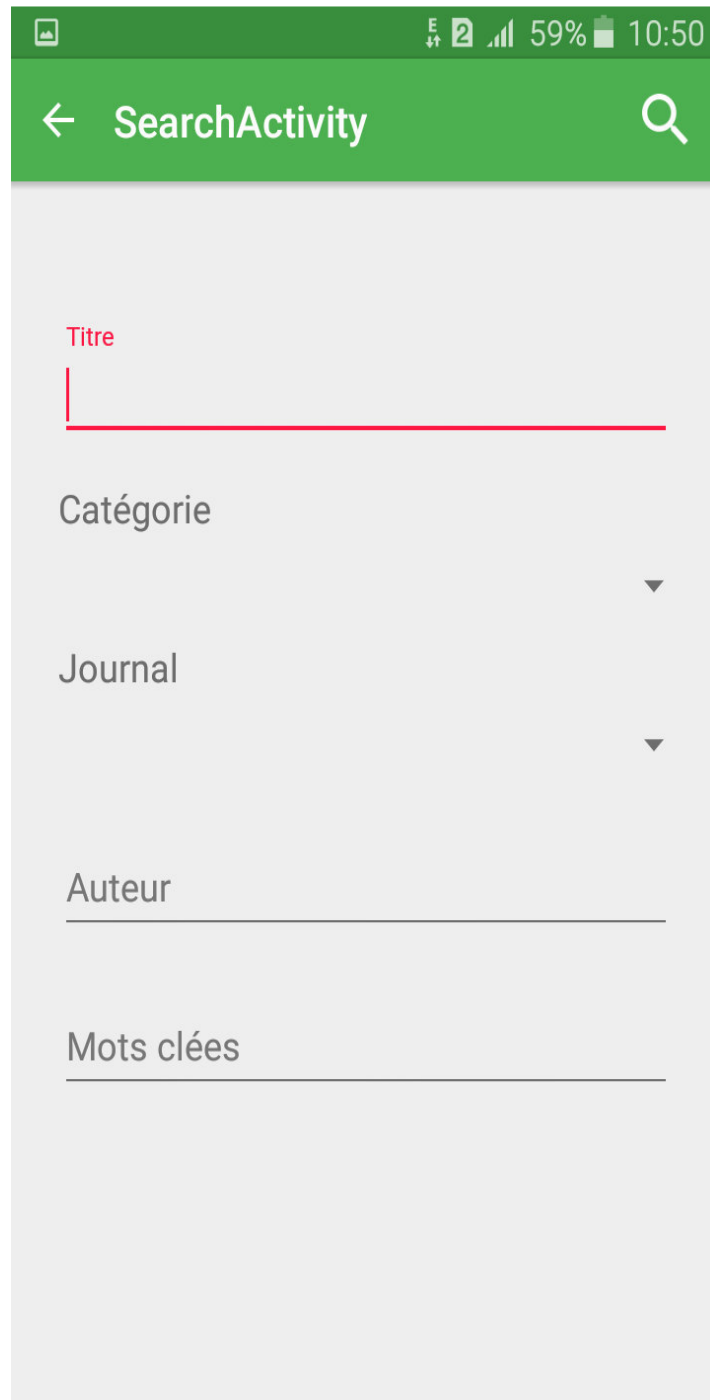
Quand un utilisateur clique sur l'icône recherche une zone de texte lui sera afficher, elle permet d'introduire les mots qu'il souhaite chercher dans les articles.



Fig. IV.10 : Interface recherche par mots clés

III. 3. Interface recherche avancée

Sur cette interface l'utilisateur pourra faire une recherche avancée par titre, catégorie, date, auteur, journal et mots clés



Titre

Catégorie ▼

Journal ▼

Auteur

Mots clés

Fig. IV.11 : Interface recherche avancée

III. 4. Recherche par localisation (GPS)

Une fois l'utilisateur a cliqué sur le bouton GPS il aura les articles de sa région en premier ainsi les régions administrative par suite



Fig. IV.12 : Recherche par localisation (GPS)

III. 5. Interface liste des fournisseurs (journaux)

Dans ce menu il a la liste des journaux qui s'affiche sur notre application

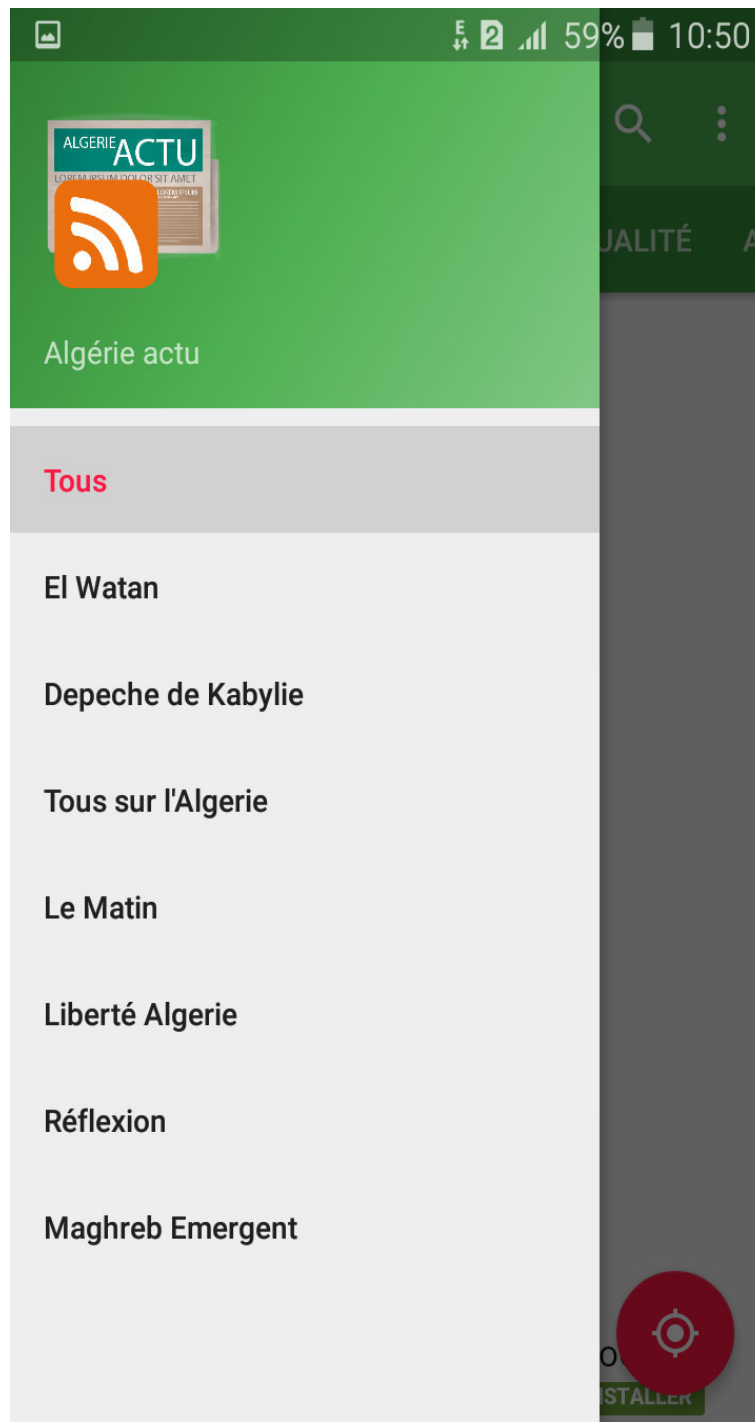


Fig. IV.13 : Interface liste des fournisseurs

Conclusion

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

Conclusion générale

L'élaboration de notre travail est dans le but de concevoir une application dédiée aux terminaux mobiles disposant de la plateforme Android. Cette application permet d'accéder à l'actualité directement à partir de son Smartphone.

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

Dans ce rapport, nous avons détaillé les différentes étapes suivies pour une réalisation réussie du travail demandé.

Dans un premier lieu, nous avons fait une étude théorique sur les notions de base d'Android et son positionnement par rapport aux autres OS mobiles.

La deuxième étape dans la réalisation de notre projet était de faire une étude sur les agrégateurs de contenus et d'expliquer leurs fonctionnements .

Nous avons par la suite entamé l'analyse et la spécification des besoins en citant les besoins fonctionnels et non fonctionnels de notre projet et les modéliser par le diagramme des cas d'utilisation en indiquant le langage de modélisation utilisé.

Partant de la spécification, nous avons commencé la conception de l'application à travers les différents diagrammes UML à savoir les diagrammes de classes et les diagrammes de séquences.

Enfin, nous avons décrit, dans le chapitre de réalisation notre application tout en présentant quelques imprimes écran de certaines interfaces.

Ce stage s'est révélé bénéfique sur plusieurs points : il nous a permis de travailler sur une technologie pour terminaux mobiles et accroître nos connaissances dans le domaine de l'embarqué en abordant plusieurs aspects techniques d'Android.

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

En effet, Android offre l'opportunité de créer des logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à puiser dans leur imagination et à mobiliser toutes leurs compétences pour exploiter au mieux les fonctionnalités de la plateforme.

Bibliographie

- ✚ Mémoire promotion 2014/2015 Réalisation d'une application mobile d'enrichissement de contenu HTML5 sur la plateforme Pixappy réalisé par KESRAOUI SMAIL et LYAZIDI REZKI dirigé par Mr: DAOUI MEHAMMED
- ✚ Mémoire promotion 2014/2015 *Conception et réalisation d'une application mobile sous android de Géo localisation indoor (cas : supermarché Priba de AZAZGA) réalisé par SAMAH Kahina et ZEGHBIB Yamina dirigé par Mme TAOURI Dalila*
- ✚ Cours Andoid 2014 .Mr M. Daoui

Webliographie

<https://fr.wikipedia.org/>

<http://stackoverflow.com/>

<https://github.com/>

<http://www.yiiframework.com/>

<https://material.google.com/>

<https://fr.wikipedia.org/>