

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



Université Mouloud MAMMARI de Tizi-Ouzou
Faculté de Génie électrique et Informatique
Département Informatique



Mémoire

De fin d'études

En vue de l'obtention de Master en Informatique

Thème

Navigation GPS sous Android

ANDROID

Proposé et dirigé par :
M^F DAOULM

Réalisé par :
M^F GUECHOU SABAR

2011/2012

Remerciements

Je remercie le Bon dieu de m'avoir donné le courage et la volonté pour la réalisation de mon projet.

Je tiens à remercier Mon promoteur Mr DAOUJ.M pour sa disponibilité, son aide, ses conseils précieux, ses critiques constructives, ses explications et ses suggestions pertinentes.

Je tiens à exprimer toute ma gratitude aux membres du jury pour avoir accepté d'évaluer et de juger mon travail.

Je tiens aussi à remercier Mr Maïdi.H pour son soutien, son aide, et ses conseils.

Et enfin je remercie tous ceux qui ont contribué de loin ou de près pour la réalisation de mon projet.

DEDICACES

*Je dédie ce modeste travail A la mémoire de mes grands parents,
que dieu bénisse leurs âmes.*

*A ceux qui ont éclairé ma vie, ma très chère mère (Aïcha) et mon
cher père (Rabah) pour leur aide et leur soutien tout au long de
mes études, et qui ont fait de moi ce que je suis aujourd'hui et
j'espère qu'un jour je serai capable de leurs donner au moins le
minimum car quoiqu'on face on arrivera jamais à leurs rendre
tout.*

A mes chère sœurs Katia, Yasmine, et Lahna.

*A tous mes ami(e)s, et particulièrement Anis, Takfarines, Reda,
Samy, Moh, Ghilas, alimoh, Amazigh, Brahim, Amel, doudou,
Nassim, Slimane, Nounou, Nassim, ma sœur linda(lyly), samira,
kahina, Hayet, nacima, nacira, Djouhar, Lynda, karima, Lydia,
dehia, dehia, Ghania, ouisa, Chabha, tina, mon ami d'enfance
Fahim, brahim et hakim.*

*A youyou et sa femme Lucette et à leurs famille particulièrement
(Na zehour et Na thaklithe). A baby.*

A mon cousin Salim et sa femme louisa.

A la famille de mon frère Anis (Da moh, na Farou et malik).

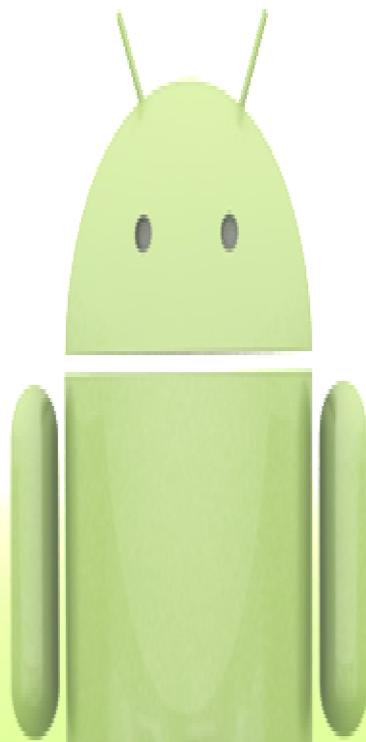
*A tous mes oncles : Brahim, Slimane, Hamouche, Abdelkader,
Nacer, et said.*

*A tous mes cousins et mes cousines, et particulièrement Idir,
Hamza, Fawzi, Moumouh, et Yacine.*

Sarah, Dounia, Lila, Agnes, Lydia et yamina.

Sans oublier mes camarades de la promotion 2012.

Sommaire:



android

Sommaire

INTRODUCTION GENERALE :	1
--------------------------------	----------

Chapitre I : Etat de l'art d'android.

I.1. INTRODUCTION :	2
I.2. DESCRIPTION :	3
I.3. HISTORIQUE D'ANDROID :	3
I.4. FONCTIONNALITES D'ANDROID :	4
I.5. ARCHITECTURE ANDROID :	6
I.5.1. Applications :	7
I.5.2. Framework de développement :	7
I.5.3. Bibliothèques :	8
I.5.4. Android Runtime :	10
I.5.5. Linux Kernel :	10
I.6. LES COMPOSANTS PRINCIPAUX D'UNE APPLICATION :	11
I.6.1. Activités:	11
I.6.2. Services:	11
I.6.3. Broadcast receivers:	11
I.6.4. Content providers :	11
I.7. CYCLE DE VIE D'UNE ACTIVITE :	12
I.8. DEVELOPPEMENT D'UNE APPLICATION ANDROID :	13
I.9. PRISE EN MAIN DE L'ENVIRONNEMENT ANDROID :	15
I.9.1. Présentation du SDK :	15
I.9.2. Les SDK Android :	16
I.9.3. ADT pour Eclipse :	17
I.9.4. Emulateur :	18
I.10. ENVIRONNEMENT DU DEVELOPPEMENT :	19
I.10.1. IDE Eclipse :	19
I.11. Conclusion :	20

Sommaire

Chapitre II : Géolocalisation.

II.1. INTRODUCTION :	21
II.2. TECHNIQUES DE GEOLOCALISATION:	22
II.2.1. Géolocalisation par satellite :	22
II.2.2. Géolocalisation par GSM :	22
II.2.3. Géolocalisation par Wifi :	23
II.2.4. Utilisation des méthodes mixtes de géolocalisation :	23
II.2.5. Géolocalisation indoor :	24
II.3. TECHNOLOGIES DE POSITIONNEMENT:	25
II.3.1. Global Positioning System (GPS):	25
II.3.1.1. Histoire (brève) du GPS :	26
II.3.1.2. Composition du GPS :	26
II.3.1.2.1. Segment spatial :	26
II.3.1.2.2. Le segment de contrôle :	27
II.3.1.2.3. Segment utilisateur :	28
II.3.1.3. Principe de Fonctionnement du GPS :	28
II.3.1.4. Les erreurs de positionnement :	31
II.3.1.5. Le système de coordonnées locales :	34
II.3.1.6. Les projections :	34
II.3.2. A-GPS (Assisted GPS):	37
Figure II.22 : A-GPS.	38
II.3.3. E-OTD (Enhanced Observed Time Difference):	38
II.3.4. TOA (Time Of Arrival):	39
II.3.5. Cell-ID:	39
II.4. AUTRES GNSS (SYSTEME DE NAVIGATION GLOBALE PAR SATELLITE):	40
II.4.1. Galileo :	40
II.4.2. GLONASS :	40
II.4.3. Beidou :	40
II.5. CONCLUSION:	41

Sommaire

Chapitre III : Analyse et conception.

III.1. INTRODUCTION :	42
III.2. PRESENTATION DE L'UML : [4]	43
III.2.1. Modélisation avec L'UML : [5]	43
III.2.2. La démarche de modélisation avec L'UML :	43
III.3. ANALYSE :	44
III.3.1. Identification des besoins :	44
III.3.2. Identification des acteurs : [6]	45
III.3.3. Spécification des tâches :	45
III.3.4. Spécification des scénarios : [6]	46
III.3.5. Les cas d'utilisation : [6]	47
III.3.5.1. Description des cas d'utilisations :	47
III.4. CONCEPTION :	52
III.4.1. Le niveau applicatif :	52
III.4.1.1. Le diagramme des cas d'utilisation: [7].....	52
III.4.1.2. Les diagrammes de séquences :	54
III.4.1.3. Les diagrammes d'activités : [8].....	58
III.4.1.4. Les diagrammes de classe : [8]	62
III.4.2. Le niveau données:	65
III.4.2.1. Schéma de la base de données :	65
III.4.2.2. Les tables :	65
III.4.3. Conception des maquettes préliminaires :	67
III.4.3.1. L'écran d'accueil :	67
III.4.3.2. Partie Géolocalisation / Itinéraire : La map :	68
III.4.3.3. Partie Géolocalisation / Itinéraire : La pop-up d'itinéraire et de navigation : ...	69
III.4.3.4. Partie Géolocalisation / Itinéraire : L'affichage des informations sur la map ...	70
III.4.3.5. Partie Recherche de bâtiment : les listes de catégories et de bâtiments :.....	71
III.4.3.6. Partie Recherche de bâtiment : la fiche détaillée d'un bâtiment :.....	72
III.5. CONCLUSION :	73

Sommaire

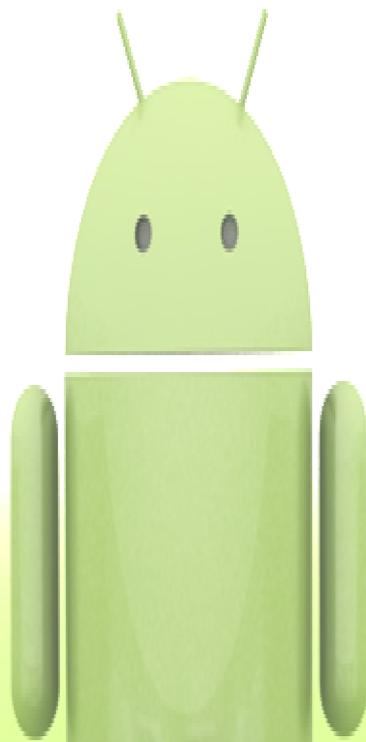
Chapitre IV : Réalisation et mise en œuvre.

IV.1. INTRODUCTION :	74
IV.2. ENVIRONNEMENT DE TRAVAIL :	75
IV.2.1. Environnement matériel :	75
IV.2.2. Environnement cible :	75
IV.2.3. Environnement logiciel :	75
IV.2.3.1. Système d'exploitation :	75
IV.2.3.2. Langage de programmation :	76
IV.2.3.3. Eclipse :	76
IV.2.3.4. IDE Eclipse :	76
IV.2.3.5. Le plugin ADT :	77
IV.2.3.6. Software Development Kit (SDK) :	77
IV.2.3.7. Android :	77
IV.2.3.8. Autres outils :	78
IV.2.3.8.1. Balsamiq :	78
IV.2.3.8.2. SQLite :	78
IV.3. FONCTIONNEMENT DE MON APPLICATION :	78
IV.3.1. L'écran d'accueil :	79
IV.3.2. L'écran de géolocalisation :	80
IV.3.3. L'écran de Pop-up itinéraire :	81
IV.3.4. L'écran de la liste des catégories :	82
IV.3.5. L'écran de La liste des bâtiments correspondants à une catégorie :	83
IV.3.6. L'écran de Tous les départements universitaires de Bastos :	84
IV.3.7. L'écran de la partie informations générales de la fiche détaillée d'un bâtiment :	85
IV.3.8. L'écran de la partie Localisation/Maps de la fiche détaillée d'un bâtiment :	86
IV.3.9. L'écran de la partie Complément d'information de la fiche détaillée d'un bâtiment :	87
IV.4. CONCLUSION :	88
 CONCLUSION GENERALE :	89
TABLE DES ILLUSTRATIONS :	91

Sommaire

ANNEXE A : MANUEL D'INSTALLATION ET DE MISE EN MARCHÉ.	94
ANNEXE B : OBTENTION D'UNE CLE POUR UTILISER GOOGLE MAPS.	95
GLOSSAIRE :	97
BIBLIOGRAPHIE :	98
NETOGRAPHIE :	99

Résumé:



ANDROID

Résumé

Résumé :

Le présent travail s'inscrit dans le cadre du projet de fin d'étude au département d'informatique de l'université Mouloud Mammeri, en vue d'obtention du diplôme de Master en informatique.

J'ai l'occasion de réaliser le projet au sein de l'université Mouloud Mammeri (Bastos). Ma mission était, donc, de développer une application mobile de navigation GPS au sein du campus.

La liste des fonctionnalités présentes sur l'application est la suivante :

- ✓ Se localiser
- ✓ Afficher les bâtiments d'une catégorie donnée
- ✓ Recherche d'un bâtiment sur le campus
- ✓ Itinéraire piéton ou en voiture vers un bâtiment du campus
- ✓ Navigation piéton ou en voiture via le GPS vers un bâtiment du campus
- ✓ Consulter la fiche détaillée de chaque bâtiment (Adresse, horaires, description)
- ✓ Appeler le bâtiment / Consulter son site internet.
- ✓ Consulter le menu du RU.

Pour réaliser mon application j'ai utilisé le langage java comme un outil de développement, et le logiciel utilisé: Eclipse.

Abstract:

This work is part of the final project study computer science department of the University Mouloud Mammeri in order to obtain the Master's degree in computer science.

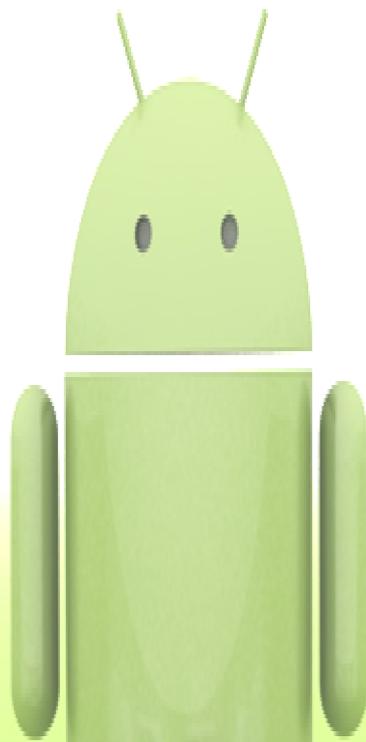
I have the opportunity to complete the project within the University Mouloud Mammeri (Bastos). My mission was therefore to develop a mobile application for GPS navigation on campus.

The list of the features present on the application is as follows:

- ✓ To locate.
- ✓ Show buildings in a given category.
- ✓ Search building on campus.
- ✓ Route pedestrian or car to a campus building.
- ✓ pedestrian or car navigation via GPS to a campus building.
- ✓ Consult the detailed each building (address, opening hours, description).
- ✓ Call Building / See its website.
- ✓ To consult the menu of RU.

To realize my application I used the Java language as a tool for development, and software used: Eclipse.

Introduction Générale



ANDROID

Introduction générale :

De plus en plus des personnes accèdent désormais aux services de Géolocalisation via des moyens "non traditionnels" comme les terminaux mobiles, et ce nombre ne cesse d'augmenter.

Par ailleurs, un Smartphone est basé sur un système d'exploitation ouvert tel que iPhone OS, Android, Blackberry OS, Windows Phone 7 ou encore Palm WebOS.

L'Android est une nouvelle plateforme en code source ouverte. De plus, selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sûre et ouverte. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Les utilisateurs peuvent adapter les applications à leur besoin. C'est pour cela j'ai choisi la plateforme Android pour faire une étude approfondie et pour développer mon application.

Le but de mon travail est de développer une application de navigation gps sous android pour les Smartphones.

L'objectif principal de l'application est de permettre à l'utilisateur de se repérer dans le campus universitaire, il est parfois un peu compliqué de trouver le lieu où l'on doit se rendre, surtout si l'on est un nouvel étudiant grâce a mon application, L'utilisateur est capable de se localiser sur le campus et de naviguer vers les endroits voulus (administration, départements, restaurants, et bibliothèques).

Pour cela j'ai structuré le travail en quatre chapitres :

- Le premier chapitre s'intitule « état de l'art d'android », Ce chapitre est une introduction à la plate-forme, aux outils et à la configuration de notre environnement de travail.
- Le deuxième chapitre s'intitule « Géolocalisation », dans ce chapitre j'ai opté pour une description de cette technologie qui fait parler d'elle de plus en plus. Cette dernière sert à déterminer la position géographique précise d'un individu dans un environnement bien déterminé, et de naviguer vers des emplacements précis.
- Le troisième chapitre sous le nom « analyse et conception », est consacré à l'analyse et la conception de mon application, afin de réaliser l'application mobile de navigation gps sous android, pour cela j'ai opté pour le langage UML étant le mieux adapté pour les applications mobiles.
- Le quatrième et le dernier chapitre « Réalisation et implémentation », comporte quand à lui la présentation de l'environnement dont lequel mon application a été réalisée, les outils utilisés et quelques interfaces de mon application.

Chapitre I :

Etat de l'art d'android



I.1. Introduction :

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

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

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

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

I.2. Description :

Android est un système d'exploitation open-source pour smartphones, PDA et autres terminaux mobiles, conçu par Android, une start-up rachetée par Google en juillet 2005. Il existe d'autres types d'appareils possédant ce système d'exploitation tels que les téléviseurs et les tablettes.

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 (OHA) créée officiellement le 5 novembre 2007. Toutes ces entreprises interviennent, plus ou moins directement, dans le marché de la téléphonie mobile.

Le but de cette alliance est de mettre en place des normes ouvertes dans le domaine de la téléphonie mobile. Ce qui veut dire que les développeurs d'application Android pourront accéder aux fonctionnalités du cœur de téléphone via une API très fournie.

Android aura comme principaux concurrents Apple avec l'iPhone, Microsoft et son Windows Mobile et Nokia avec Symbian mais également des solutions libres telles que LIMO ou OpenMoko.

I.3. Historique d'Android :



Figure I.1 : évolution des versions d'Android.

En juillet 2005, Google a acquis 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'avait des données sûres à propos des marchés dans lesquels ils allaient se positionner.

Après ce rachat fait par Google, une équipe dirigée par Andy Rubin, un ancien d'Android Inc, 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'elle développait un nouveau téléphone mobile, etc.

Chapitre I : ETAT DE L'ART D'ANDROID

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

En octobre 2008, apparait la première version d'Android qui n'avait pas reçu de nom. Cette version s'est avérée être la β du système.

La version 1.5 **Cupcake** corrigea le manque d'API et rendit le système plus utilisable.

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

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

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

On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google. La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont sûrement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.

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

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

I.4. Fonctionnalités d'Android :

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

Voici quelques fonctionnalités proposées par Android classées par version :

✓ Android version 1.5 (Cupcake) :

- Enregistrement et lecture des vidéos.
- Mise en ligne directe des vidéos sur YouTube.
- Mise en ligne directe des photos Picasa.
- Prise en charge du Bluetooth A2DP.
- Dossiers dynamiques et widgets pour le home.
- Copier/coller étendu aux pages web.
- Nouvelle version du clavier virtuel.

✓ Android version 1.6 (Donut) :

- L'application Galerie permet d'effacer plusieurs photos à la fois.
- Amélioration de l'Android Market.
- Amélioration de la vitesse de la recherche vocale et intégration étendue à plus d'applications natives.
- Prise en charge sur une seule application de la prise de photo et de l'enregistrement vidéo.
- Possibilité de rechercher simultanément dans les favoris, les historiques, les contacts et sur Google depuis le home via le widget recherche.
- Moteur Text-to-speech.
- Prise en charge de plusieurs résolutions d'écran.

✓ Android version 2.0/2.1 (Éclair) :

- Interface utilisateur revue (lock screen et lanceur d'application).
- Fonds d'écran animés.
- New browser interface avec prise en charge du HTML5.
- Prise en charge du protocole Microsoft Exchange.
- New contact lists.
- Prise en charge du Bluetooth 2.1.
- Amélioration du clavier virtuel.
- Prise en charge en natif du flash et du zoom numérique pour des appareils photos.
- Amélioration du ratio blanc/noir sur les fonds.
- Gestion multi-comptes Gmail et ajout de la synchronisation avec Facebook.

✓ Android version 2.2 :

- Augmentation de la performance et de la vitesse.
- Fonctionnalité de Hot spot Wifi.
- Partage de contact sur bluetooth.
- Mise à jour automatique des applications.

I.5. Architecture Android :

Le diagramme suivant illustre les composants principaux du système d'exploitation Android. Chaque section sera décrite dans ce qui suit :

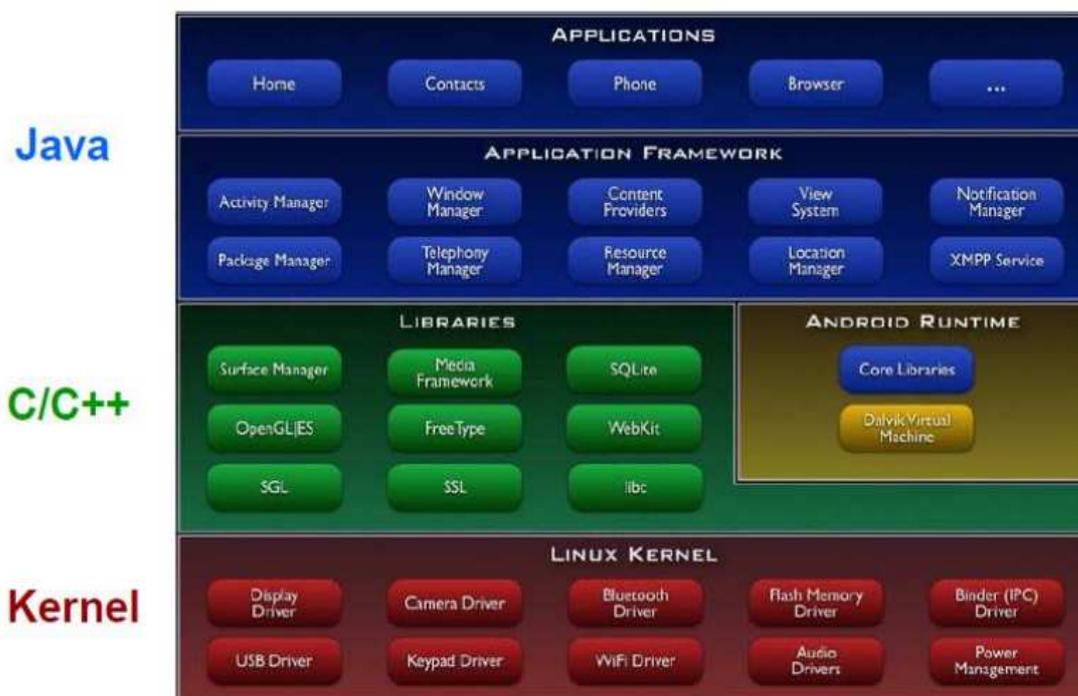


Figure I.2 : Architecture de l'Android¹

Android est basé sur un kernel linux 2.6.xx.

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

¹ <http://developer.android.com/guide/basics/what-is-android.html>.

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

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

I.5.1. Applications :

Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur... toutes écrites en JAVA.

I.5.2. Framework de développement :

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériel périphérique et informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Les développeurs ont un accès complet au même framework API utilisé par les applications de base. L'architecture d'application est conçue pour simplifier la réutilisation des composants; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités (soumis à des contraintes de sécurité appliquées par le framework). Ce même mécanisme permet aux composants d'être remplacés par l'utilisateur.

Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données.
- Un gestionnaire de ressources.
- Un gestionnaire de notification qui permet à toutes les demandes d'afficher des alertes personnalisées dans la barre d'état.

- Un gestionnaire d'activité qui gère le cycle de vie des applications et propose une navigation commune.

I.5.3. Bibliothèques :

Android dispose d'un ensemble de bibliothèques C / C++ utilisées par les différents composants du système Android. Elles sont offertes aux développeurs à travers le framework Android. En voici quelques unes:

Système de bibliothèque C – une mise en œuvre dérivée de BSD de la bibliothèque C standard du système (libc), destinés aux systèmes embarqués basés sur Linux.

Comme cela a été dit précédemment, Android ne supporte pas la glibc, donc les ingénieurs d'Android ont développé une bibliothèque C (libc) nommé Bionic libc . Elle est optimisée pour les appareils mobiles et a été développée spécialement pour Android.

Les ingénieurs d'Android ont décidé de développer une libc propre à la plateforme Android car ils avaient besoin d'une libc légère (la libc sera chargée dans chaque processus) et rapide (les appareils mobiles ne disposent pas de CPU puissant).

La Bionic libc a été écrit pour supporter les CPU ARM, bien que le support x86 soit présent. Il n'y pas de support pour les autres architectures CPU telles que PowerPC ou MIPS. Néanmoins, pour le marché des appareils mobiles, seulement l'architecture ARM est importante.

Cette libc est sous licence BSD. Elle reprend une grande partie du code des glibc issue d'OpenBSD, FreeBSD et NetBSD.

Ces caractéristiques importantes :

Elle pèse environ 200Ko, soit la moitié de la glibc L'implémentation des pthreads (POSIX thread) a été complètement réécrite pour supporter les threads de la machine virtuelle Dalvik. De ce fait, la Bionic libc ne supporte pas les threads POSIX .

Les exceptions C++ et les "wide char" ne sont pas supportés

Médiathèques – basée sur PacketVideo de OpenCore; les bibliothèques permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fichiers image, y compris MPEG4, H.264, MP3, AAC, AMR, JPG et PNG.

Le schéma ci-dessous décrit tous les éléments de l'architecture de ces médiathèques:

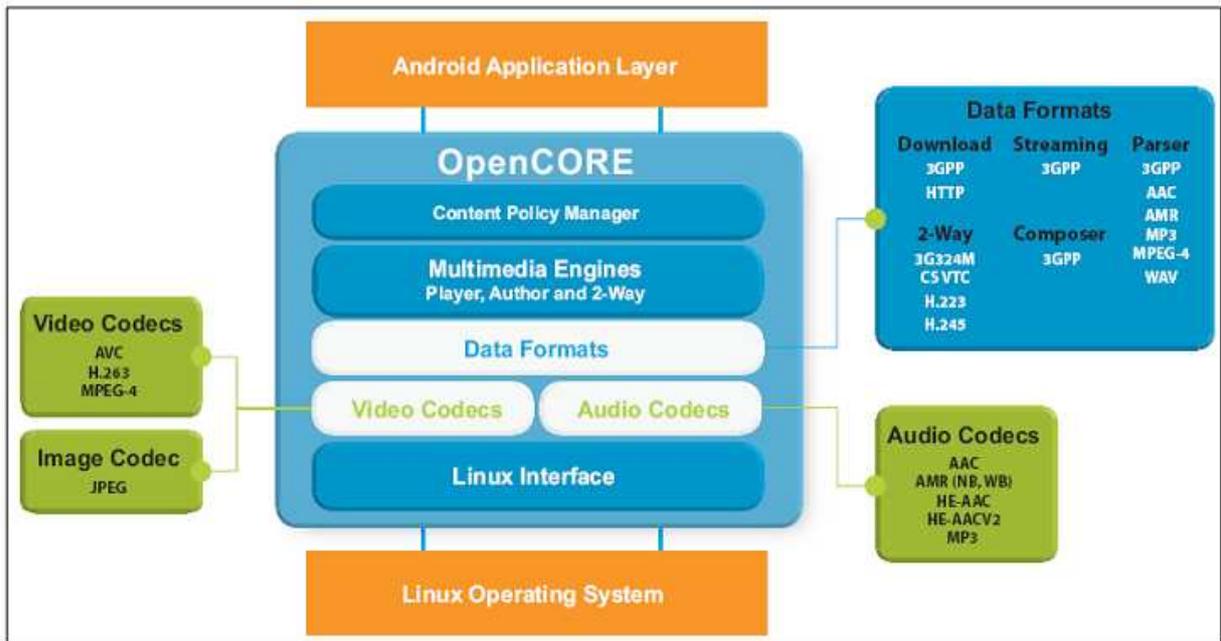


Figure I.3 : Architecture de ces médiathèques.

- Surface Manager – gère l'accès au sous-système d'affichage et de façon transparente.
- LibWebCore – Le navigateur web présent dans Android est basé sur le moteur de rendu sous licence BSD WebKit.
- WebKit est moteur de rendu, qui fournit une "fondation" sur laquelle on peut développer un navigateur web. Il a été originellement dérivé par Apple du moteur de rendu KHTML pour être utilisé par le navigateur web Safari et maintenant il est développé par KDE project, Apple, Nokia, Google et d'autres. WebKit est composé de deux bibliothèques : WebCore et JavascriptCore qui sont disponibles sous licence GPL.

WebKit supporte le CSS, Javascript, DOM, AJAX. La dernière version a obtenu 100% au test Acid 3. La version de WebKit présent dans Android à été légèrement modifiée pour s'adapter aux appareils mobiles. Ainsi, le moteur de rendu basé sur WebKit présent dans Android supporte l'affichage sur une colonne.

- SGL – le moteur graphique 2D.
- Bibliothèques 3D – une implémentation basée sur OpenGL ES 1.0 API; les bibliothèques utilisent l'accélération 3D matérielle (si disponible).
- FreeType – bitmap et vectoriel de rendu de police.
- SQLite – un moteur de base de données relationnelles puissant et léger, disponible pour toutes les applications.

I.5.4. Android Runtime :

Android inclut un ensemble de bibliothèques de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (.DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

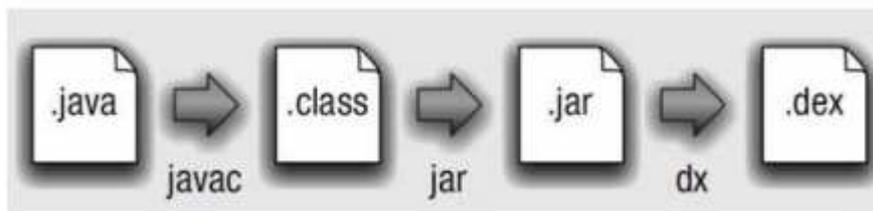


Figure I.4 : Conversation d'un fichier .java à un fichier .dex.

I.5.5. Linux Kernel :

Android est basé sur un kernel linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X window system). La glibc n'étant pas supportée, Android utilise une libc customisée appelée Bionic libc.

Enfin, Android utilise un kernel avec différents patches pour la gestion de l'alimentation, le partage mémoire, etc. permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles.

Android n'est pas linux mais il est basé sur un kernel linux. Pourquoi sur un kernel linux ?

Le kernel linux a un système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances.

Le model de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70

- Le kernel linux fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de bibliothèques entre différents processus, le chargement et le déchargement de modules à chaud.
- le kernel linux est entièrement open source et il y a une communauté de développeurs qui l'améliorent et rajoutent des drivers.

C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un kernel linux.

I.6. Les composants principaux d'une application :

Les applications Android sont composées de 4 types de composants :

I.6.1. Activités:

Une Activité représente un écran de l'application. Une application peut avoir une ou plusieurs activités (par exemple pour une application de messagerie on pourrait avoir une Activité pour la liste des contacts et une autre pour l'éditeur de texte). Chaque Activité est implémentée sous la forme d'une classe qui hérite de la classe Activity.

I.6.2. Services:

Les services n'ont pas d'interface graphique et tournent en tâche de fond. Il est possible de s'inscrire à un service et de communiquer avec celui-ci en utilisant l'API Android.

I.6.3. Broadcast receivers:

Il se contente d'écouter et de réagir aux annonces broadcast (par exemple changement de fuseau horaire, appel entrant...).

I.6.4. Content providers :

Il permet de partager une partie des données d'une application avec d'autres applications.

I.7. 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 présente ce que l'on appelle le cycle de vie d'une activité, c'est-à-dire qu'il indique les étapes que va traverser notre activité pendant sa vie, de sa naissance à sa mort.

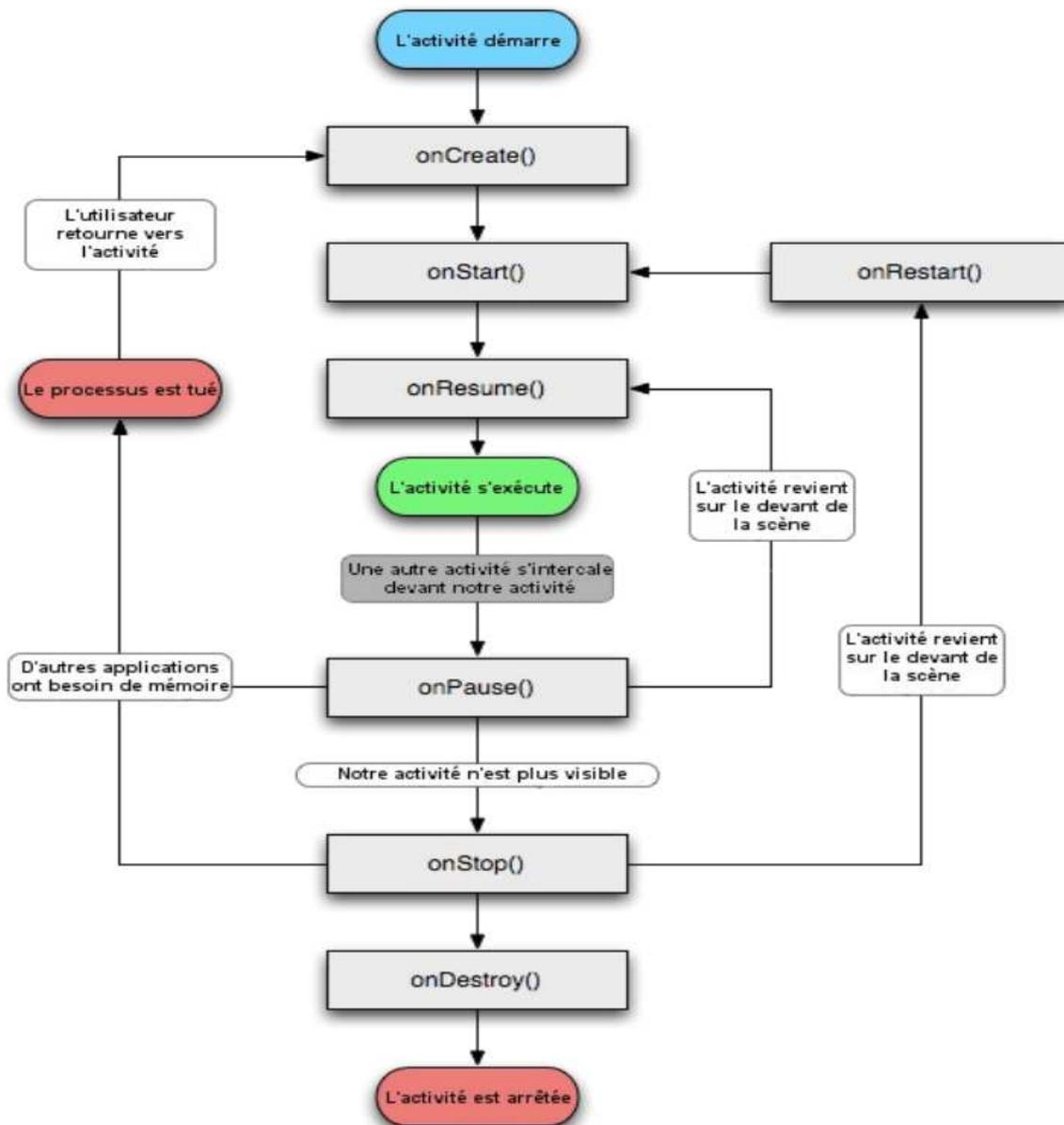


Figure I.5 : Cycle de vie d'une activité.

Chapitre I : ETAT DE L'ART D'ANDROID

- L'état **Active/courant** (Running) : C'est un état que l'activité marche en avant-plan. Dans ce cas, cette activité est active. On peut dire que cette activité ait la concentration (focus).
- L'état **Paused** (il est en pause) : Cette activité est visible mais il a perdu de se concentrer.
- L'état **Stopped** : Cette activité n'est pas visible. Si une activité est complètement masquée par une autre activité, il est arrêté. Il conserve tous les états membres et de l'information, cependant il n'est plus visible pour l'utilisateur que sa fenêtre est cachée et il sera souvent tués par le système lorsque la mémoire est nécessaire ailleurs.
- L'état **Dead** : Cette activité a terminé ou il n'a jamais été démarré. Si une activité est en pause ou arrêtée, le système peut chuter l'activité de la mémoire, soit par lui demandant de se terminer, ou tout simplement tuer le processus. Quand il est affiché de nouveau à l'utilisateur, il doit être redémarré et restauré à son état antérieur.

Il existe trois boucles principales:

- ✓ La durée de vie d'une activité se passe entre le premier appel à **onCreate()** par l'appel à **onDestroy()**. Une activité met en place tous les états globaux dans la méthode **onCreate()** et libère toutes les ressources restantes à **onDestroy()**.
- ✓ La durée de vie visible d'une activité se passe entre un appel à **onStart ()** jusqu'à ce qu'un appel correspondant à **onStop()**. Dans ce temps, l'utilisateur peut voir l'activité sur l'écran, même si elle n'est pas à l'avant et à l'interaction avec l'utilisateur. Entre ces deux méthodes, les ressources qui sont nécessaires pour montrer l'activité de l'utilisateur sont conservées.
- ✓ La durée de vie d'une activité en avant-plan se passe entre un appel à **onResume()** jusqu'à ce qu'un appel correspondant à **onPause()**. Dans ce temps, l'activité est en face de toutes les autres activités et d'interagir avec l'utilisateur. Une activité peut souvent changer son état entre l'état de reprise et l'état en pause.

I.8. Développement d'une application Android :

Voici quelques étapes principales dans le processus de développement d'une application sur Android :

- Faire la conception de la base de données.

Chapitre I : ETAT DE L'ART D'ANDROID

- Créer des classes pour représenter les données physiques (couche Mapping) et pour définir des actions comme : supprimer, ajouter, modifier des données.
- Dessiner des interfaces en les fichiers XML ou en codage :
 - ✓ Les vues (View) : Text, Edit, List, Image, Web, Map, etc.
 - ✓ Les arrangements (layout) : Frame, Linear, Relative, Table, Absolute.
- Choisir des arrangements (layout) : Les layouts sont les ressources qui indiquent les interfaces des activités. On utilise les fichiers XML pour exprimer les interfaces. Mais il existe d'autre technique pour dessiner l'interface. Dans cette technique, on programme directement les composants graphiques en utilisant le codage.
- Organiser des ressources: les constantes globales (string.xml), les icônes, les images, etc.
- Créer et mettre à jour le fichier de configuration : AndroidManifest.xml. AndroidManifest.xml (configuration de l'application) est utilisé pour stocker les dispositions (settings) globales comme les permissions de l'application, les activités, les filtres de l'intention.



Figure I.6 : Une application sur android.

- Créer des activités (Créer les classes pour exécuter les fonctions avec la base de données (ajouter, supprimer, modifier, mettre à jour, etc.) :

- ✓ Chaque activité peut correspondre avec un écran ou une fonction de cette application.
- ✓ Il existe quelques activités qui s'occupent des méthodes pour communiquer avec la base de données (Couche Mapping).
- ✓ Les Intents sont utilisées pour orienter des activités (CALL, ACTION_MAIN, ACTION_VIEW, etc).

I.9. Prise en main de l'environnement Android :

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

I.9.1. Présentation du SDK :

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

- **Le portail des développeurs**

La première chose à visiter est le portail des développeurs Android, mis en place par Google. [1].



Figure I.7 : Portail des développeurs Android.

Chapitre I : ETAT DE L'ART D'ANDROID

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

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

I.9.2. Les SDK Android :

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.). Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google tels que Maps etc.) ou de la documentation JavaDoc. Son fonctionnement est similaire aux gestionnaires de paquets de Linux. [2]

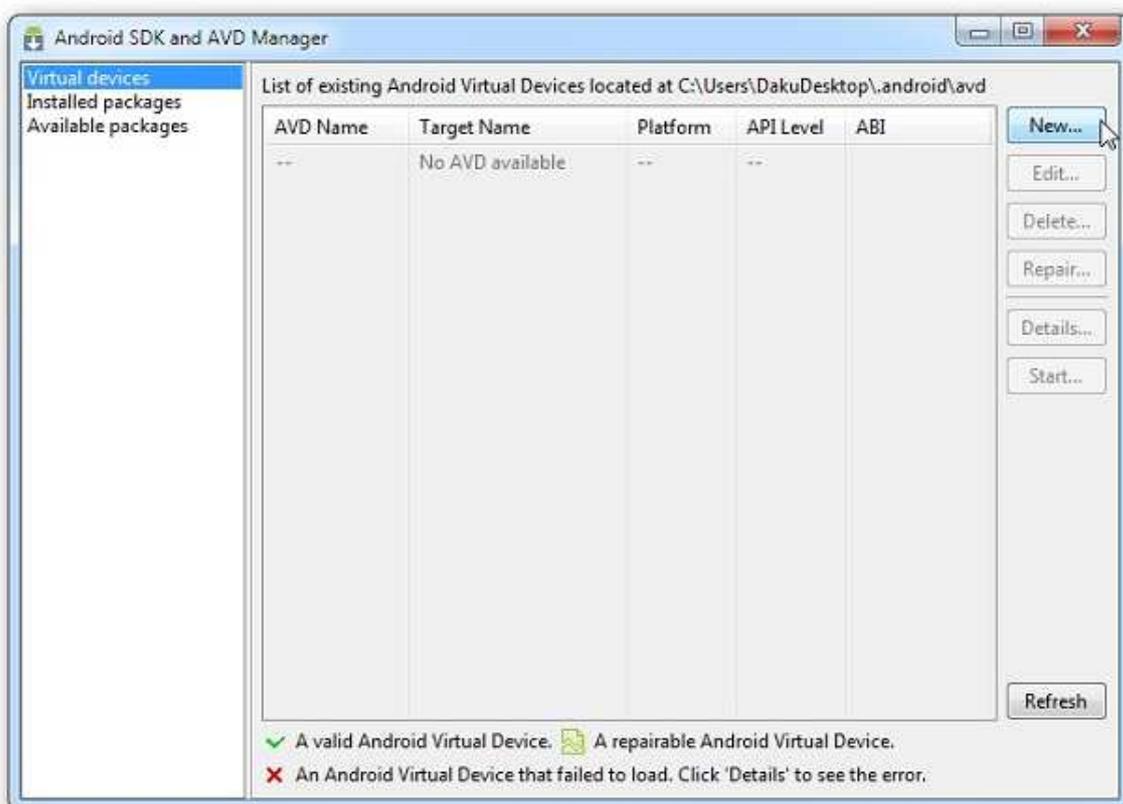


Figure I.8 : Interface d'installation du SDK Android.

I.9.3. ADT pour Eclipse :

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



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

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

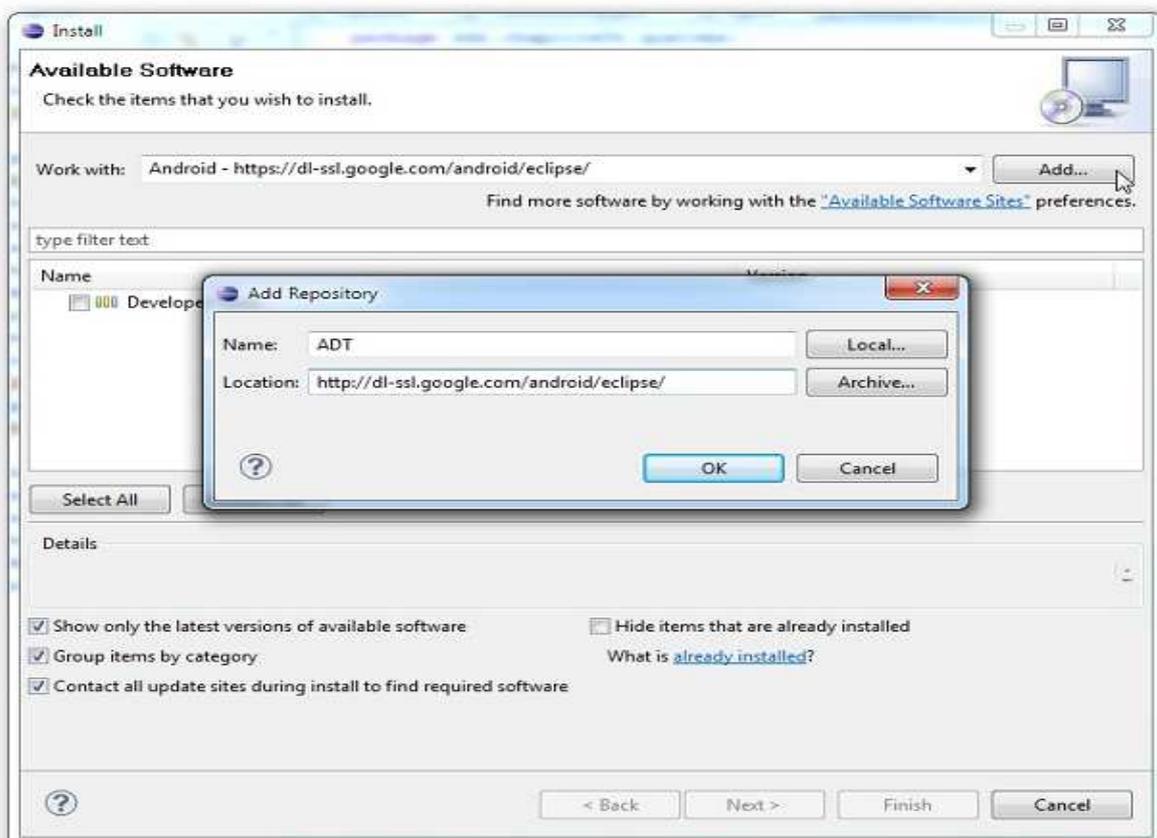


Figure I.9 : Interface d'installation ADT.

I.9.4. Emulateur :

Nous l'avons évoqué plus haut, 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.

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.



Figure I.10 : Interface du simulateur Android.

I.10. Environnement du développement :

I.10.1. IDE Eclipse :

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

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

Dans le cadre de notre projet, nous avons utilisé la version Eclipse **Indigo**, avec le plugin ADT de Google.

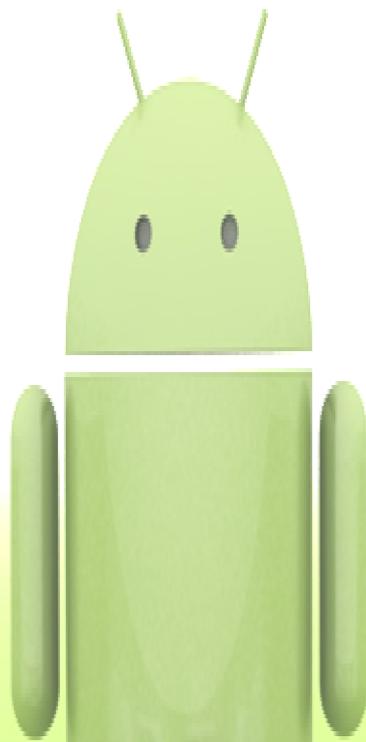
I.11. Conclusion :

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

En fin nous avons présenté l'environnement software et hardware utilisé pour la programmation Android.

Chapitre II :

Géolocalisation



ANDROID

II.1. Introduction :

La géolocalisation, un terme encore flou pour beaucoup d'entre nous, peu d'articles se consacrent à son aspect technique, nous avons donc choisi de donner une description de cette nouvelle technologie qui fait parler d'elle de plus en plus. Cette dernière sert à déterminer la position géographique précise d'un individu dans un environnement bien déterminé.

Les premiers services de géolocalisation commençaient à faire leur apparition sur le marché depuis quelques années dans les secteurs militaire et civil.

Étant un marché porteur, ces technologies sont un enjeu stratégique et commercial aussi bien pour les opérateurs de communications électroniques que pour les grands pays de ce monde.

II.2. Techniques de géolocalisation:

Pour géolocaliser l'utilisateur grâce à son Smartphone, il existe plusieurs techniques que l'on recoupe afin d'obtenir un résultat à la fois rapide et précis.

II.2.1. Géolocalisation par satellite :

La géolocalisation par satellite consiste à calculer, grâce aux signaux émis par une constellation de satellites prévue à cet effet, la position actuelle sur la face terrestre d'un terminal équipé d'une puce compatible.

Cette position est alors traduite en termes de latitude, longitude et parfois altitude et peut alors être représentée physiquement sur une carte. Le réseau satellite de positionnement le plus connu est le GPS (Global Positioning System). La géolocalisation par satellite est la méthode utilisée pour la navigation GPS en voiture.

Dans le cas du GPS, pour que le repérage spatial fonctionne, un réseau constitué d'une vingtaine de satellites tourne autour de la Terre à une altitude de 20 200 km, répartis sur 6 orbites (4 par orbite). Ces satellites constituent un maillage du ciel et servent de repères aux navigateurs GPS dans leur processus de calcul de position. Ce système de satellites est conçu de façon à ce qu'il y en ait toujours au moins quatre « visibles » par les navigateurs GPS, sans quoi la position ne peut pas être déterminée.

Pour qu'un terminal soit capable de se géolocaliser grâce au réseau GPS, celui-ci doit être équipé d'une puce électronique GPS, puce que l'on retrouve dans tous les Smartphones actuels.

Le GPS offre une précision allant de 15 à 100 mètres pour les applications civiles mais possède comme inconvénients une impossibilité d'utilisation à l'intérieur et un allumage assez long.

II.2.2. Géolocalisation par GSM :

Les téléphones mobiles sont connectés en permanence à des antennes GSM puisque c'est grâce à ces antennes que les appels et les envois de SMS peuvent être effectués. Grâce à la triangulation, c'est-à-dire avec trois antennes différentes, le Smartphone peut être repéré par sa position.

Ces antennes captent le signal émis par le téléphone et le calcul de la position peut être effectué par plusieurs méthodes :

- ✓ en fonction du temps que le signal met pour atteindre l'antenne.
- ✓ en fonction de l'angle d'arrivée et de la force du signal.

Chapitre II : GEOLOCALISATION.

- ✓ en fonction des identifiants des antennes GSM auxquels un terminal se connecte, cette méthode, appelée Cell ID, est couramment utilisée. Une base de données fait le lien entre les identifiants des antennes et leur position.

La précision de la géolocalisation par GSM se situe entre 200 mètres et plusieurs kilomètres, c'est pour cela que cette technique ne peut pas être utilisée seul pour géolocaliser un utilisateur et son Smartphone. De plus, la couverture géographique n'est pas complète en Algérie et dans le monde.

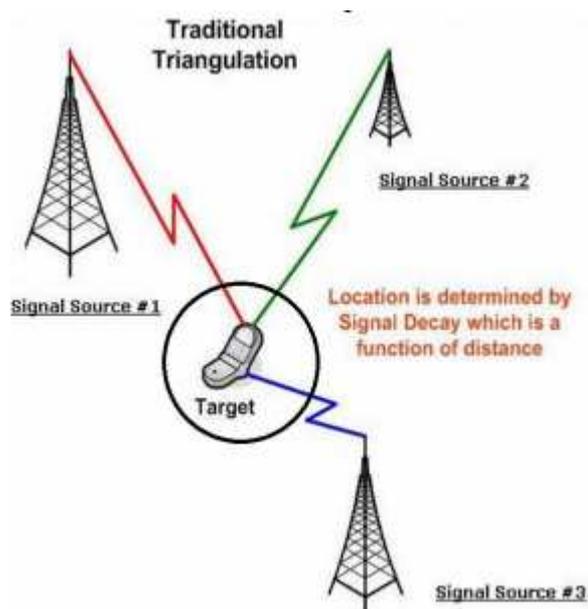


Figure II.1 : géolocalisation par GSM .

II.2.3. Géolocalisation par Wifi :

Cette technique reprend le même principe que la localisation GSM qui utilise la méthode Cell ID. La géolocalisation se fait en utilisant l'identifiant des bornes d'accès Wifi que le Smartphone détecte et en recoupant les données. Cependant, cette technique a un défaut important car la présence de borne Wifi en zone rurale est faible.

II.2.4. Utilisation des méthodes mixtes de géolocalisation :

Pour faire face aux inconvénients de chaque technique de géolocalisation, il y a actuellement une technique qui combine les trois précédentes et qui est capable de géolocaliser le Smartphone dans n'importe quelle situation possible. La précision de ce positionnement va varier en fonction des technologies disponibles, mais le temps de réponse à l'allumage et l'adaptabilité s'en verront améliorées.

Chapitre II : GEOLOCALISATION.

Grâce a cette utilisation multiple qui permet de géolocaliser une personne à l'extérieur en utilisant le GPS, et de le suivre à l'intérieur des bâtiments ou des tunnels en utilisant la technologie GSM couplée au Wifi pour plus de précision.



Figure II.2 : géolocalisation mixte .

II.2.5. Géolocalisation indoor :

Comme vu précédemment, plusieurs technologies peuvent être employées pour géolocaliser un terminal. Et cette couverture fonctionne très bien en "outdoor" (à l'extérieur). En revanche, de par la nature des signaux émis, ces systèmes sont inefficaces en "indoor" (à l'intérieur). Il existe cependant un moyen de se géolocaliser à l'intérieur même d'un bâtiment. Il s'agit de placer des bornes Wi-Fi dans le bâtiment en question pour émettre des signaux Wi-Fi permettant ainsi la géolocalisation des Smartphones. C'est ce qui a été mis en place au sein du centre commercial de 4 étages "Les quatre temps"

Chapitre II : GEOLOCALISATION.

à Paris. Une application Smartphone permet de se localiser précisément dans le centre et nous permet de trouver facilement le chemin à emprunter pour se rendre à la boutique désirée.

La géolocalisation indoor va très certainement être amenée à se développer dans l'avenir car les possibilités d'exploitations sont très importantes.



Figure II.3 : géolocalisation Indoor.

II.3. Technologies de positionnement:

Il y a plusieurs méthodes pour localiser le MS¹ (Mobile station). Ces méthodes sont basées sur la transmission de certains signaux et leur réception à l'autre bout.

La technologie de positionnement utilisée est choisie selon les besoins (par exemple le temps de réponse) des applications. Parmi les différentes technologies de positionnement existantes, nous avons [3] :

II.3.1. Global Positioning System (GPS):

¹ MS est un terminal mobile : le téléphone portable, un PDA...

Chapitre II : GEOLOCALISATION.

Le GPS (système de positionnement global) permet de déterminer des positions en 3 dimensions en tout lieu du globe et à toute heure (sauf exceptions). Il est actuellement le principal système mondial de positionnement par satellite, et le seul à être entièrement opérationnel (contrairement à GLONASS [russe], Beidou [chinois] et Galileo [européen]).

II.3.1.1. Histoire (brève) du GPS :

Le GPS a été conçu au début des années 70 par le département de la Défense des Etats-Unis, mais la constellation de 24 satellites opérationnels a été complétée en 1993. À l'origine, le GPS était destiné à répondre aux besoins des militaires, mais la Presidential Decision Directive du 28 mars 1996 s'achevait par la déclaration suivante :

« Nous continuerons de fournir en permanence le service de positionnement standard GPS à des fins pacifiques d'utilisation civile, commerciale et scientifique à l'échelle mondiale et sans imposer de redevances directes aux utilisateurs ». [Traduction]

Le GPS offrait toutefois deux niveaux de service : un service de positionnement standard (SPS) accessible à tout utilisateur, et un système de positionnement précis (PPS) principalement réservé aux militaires américains. Le 1er mai 2000, le président Bill Clinton annonça qu'il mettait fin à cette dégradation volontaire du service. Aujourd'hui, le GPS est utilisable sans frais, par tous, et n'importe où dans le monde. Environ 80% des utilisateurs sont désormais des civils et profitent du système avec une précision de l'ordre de 20 mètres dans le plan horizontal, 95% du temps.

II.3.1.2. Composition du GPS :

Dans le langage courant, on utilise le mot « GPS » pour désigner le récepteur. Or « GPS » désigne le système de positionnement en entier, et ce système se compose de trois éléments : le **segment spatial**, le **segment contrôle** et le **segment utilisateur**.

II.3.1.2.1. Segment spatial :

Le segment spatial est composé d'un réseau de 28 satellites en orbite quasi-circulaire autour de la terre, à une hauteur à peu près de 20000 km, dont la période de révolution est de 12 heures sidérales. Ces satellites sont répartis sur 6 plans orbitaux inclinés 55° par rapport à l'équateur. Cette répartition spatiale garantit la visibilité en permanence d'au moins six satellites en tout point du globe. La durée de vie minimale du système est de 7 ans.

Il existe trois catégories de satellites GPS:

Chapitre II : GEOLOCALISATION.

Bloc I: Satellites lancés entre 1978 et 1985. Tous les satellites du Bloc I sont maintenant hors-service excepté un seul, qui est activé de manière périodique. Leur durée de vie est de 4,5 ans. La principale différence entre ces satellites et les générations suivantes est l'impossibilité de dégrader volontairement le signal transmis.

Bloc II : Seconde génération de satellites GPS mis sur orbite à partir de 1985. Ils ont la capacité de dégrader le signal émis et leur durée de vie est de 7,5 ans.

Bloc III: ils ont été construits pour avoir une durée de vie de 10 ans. Ils sont capables de communiquer entre eux et ont été mis sur orbite depuis 1996 afin de maintenir une constellation complète.



Figure II.4 : Segment spatial.

II.3.1.2.2. Le segment de contrôle :

Le segment de contrôle comprend cinq stations de poursuite situées à Colorado Springs, Hawaii, Ascension Island, Diego Garcia et Kwajalein. Ces stations sont les yeux et les oreilles du système GPS, elles vérifient l'état des satellites lors de leur passage au-dessus d'elles. Ces stations transmettent ensuite leurs données à la station principale de Colorado Springs. C'est là que les paramètres décrivant l'orbite des satellites et la qualité des horloges embarquées sont estimés, la vérification de la santé des satellites et la détermination d'un repositionnement éventuel sont également contrôlés. Cette information est alors renvoyée à trois stations de chargement qui transmettent l'information aux satellites. Grâce à la répartition uniforme des stations de contrôle, tous les satellites GPS sont captés à 92% du temps.

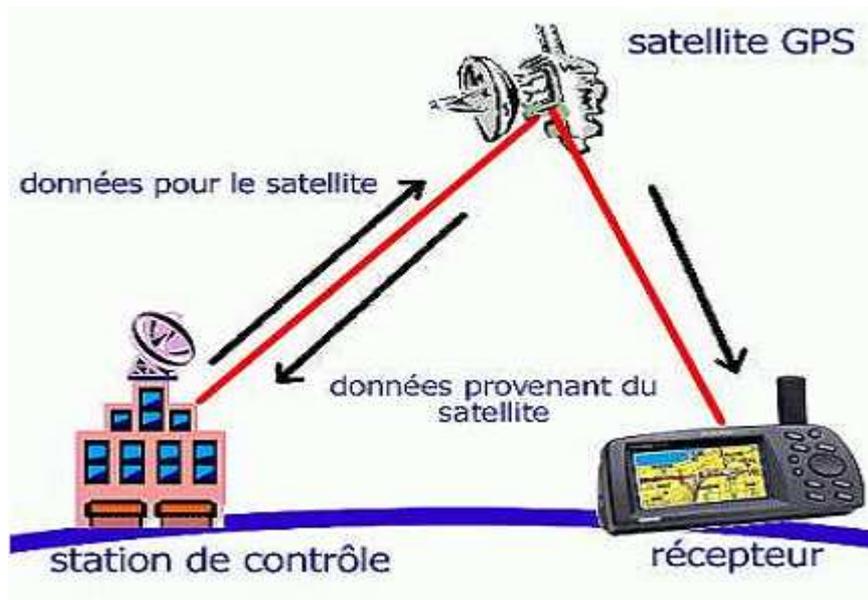


Figure II.5 : Le Segment de contrôle.

II. 3.1.2.3. Segment utilisateur :

Le segment utilisateur est constitué de récepteurs qui ont été conçus afin de décoder le signal transmis par les satellites pour déterminer la position, la vitesse et le temps de l'utilisateur.



Figure II.6 : Segment utilisateur.

II.3.1.3. Principe de Fonctionnement du GPS :

Notre observateur en possession d'un récepteur GPS reçoit les informations de tous les satellites de la constellation qui sont en visibilité. Les informations reçues permettent

Chapitre II : GEOLOCALISATION.

au récepteur GPS de donner une position par rapport à un référentiel appelé ECEF (Earth Centered, Earth Fixed).

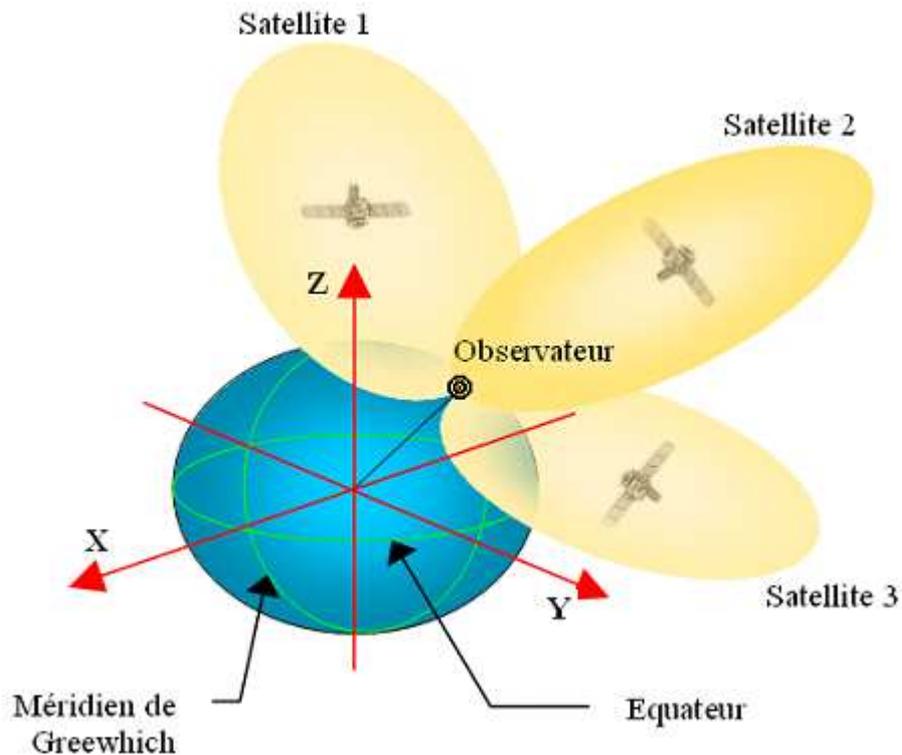


Figure II.7 : Fonctionnement du GPS.

Les informations envoyées par chaque satellite sont entre autre :

- La position exacte du satellite dans le système ECEF
- Le moment exact où le signal a été envoyé

Le récepteur lui analyse l'écart temporel entre l'émission du signal et la réception puis, grâce à un savant calcul il détermine sa distance par rapport aux différents satellites. Sa position sera par conséquent à l'intersection de toutes les sphères dont leur rayon sera égal à la distance satellite récepteur.

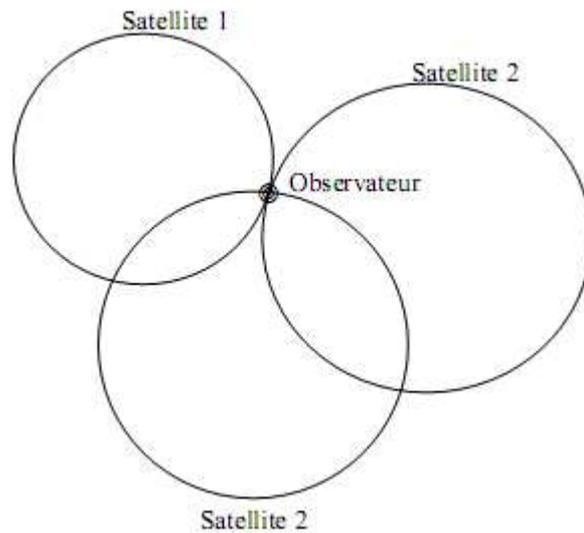


Figure II.8 : positionnement de l'observateur à l'intersection des 3 satellites.

Mais les choses n'étant pas parfaites la réalité sera tout autre. Notre observateur sera en fait quelque part dans la zone commune aux trois satellites.

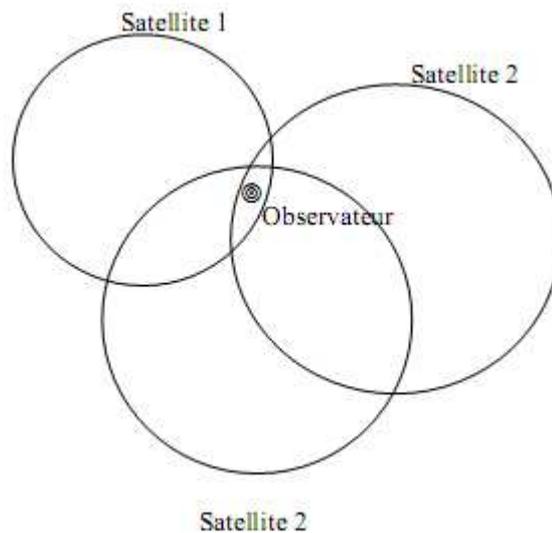


Figure II.9 : positionnement de l'observateur dans la zone commune des 3 satellites.

Quelques chiffres clef :

- La constellation GPS est constituée de 24 satellites

- Orbite elliptique presque circulaire de 26000 Km de rayon
- Le tour complet de l'orbite est effectué en 12 heures environ.

II.3.1.4. Les erreurs de positionnement :

Bien que la position soit donnée avec une grande précision, il faut tenir compte des erreurs du système GPS. Sont principalement à l'origine de ces erreurs :

- Freinage des ondes électromagnétiques dans l'ionosphère (5 à 100 Km d'altitude)
- Freinage des ondes électromagnétiques dans la troposphère (0 à 50 Km d'altitude)
- Erreur de synchronisation des horloges des satellites et du récepteur GPS
- Plus diverses raisons comme l'effet relativiste, réflexion des ondes etc...
- Le nombre de satellites en visibilité et leurs répartitions dans le ciel ont une influence sur la précision de la position.

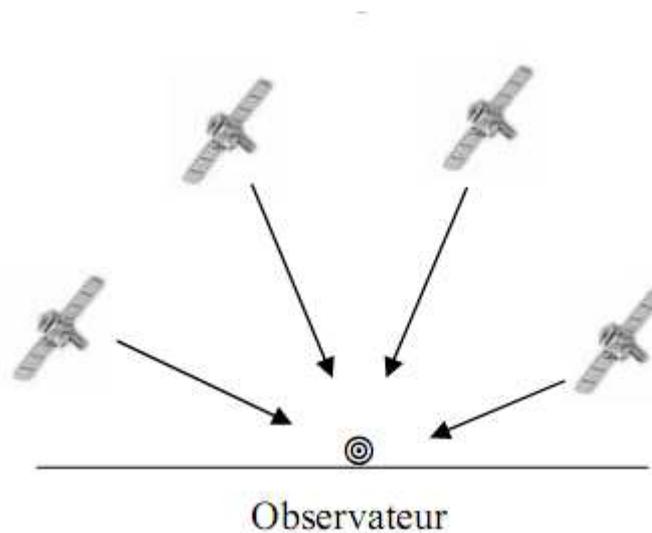


Figure II.10 : Bonne répartition.

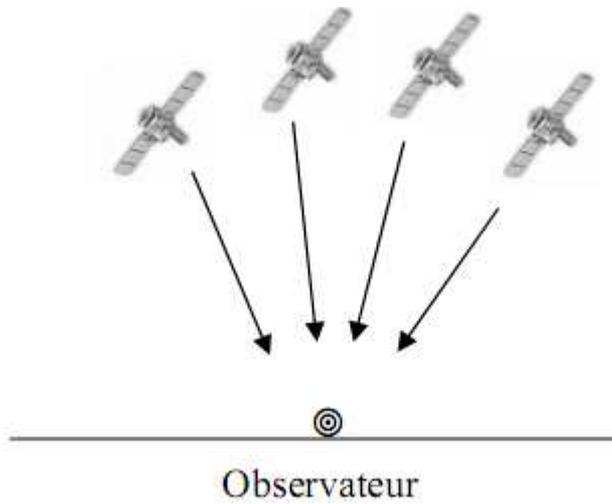


Figure II.11 : Mauvaise répartition.

- ❖ Le relief ainsi que la végétation perturbent la réception du signal.

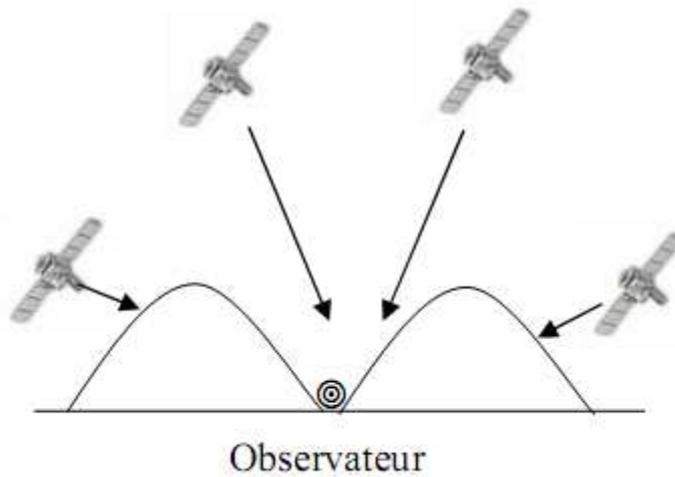


Figure II.12 : Dans une vallée, Mauvaise réception.

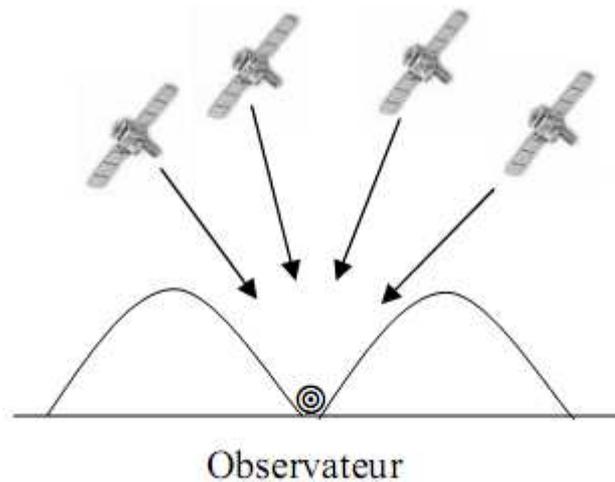


Figure II.13 : Dans une vallée, Meilleure réception.

Le récepteur GPS fournit en général la synthèse de ces informations en affichant la précision de la position. Une page particulière permet de visualiser la distribution des satellites ainsi que la qualité du signal.

Il est possible sur certains récepteurs d'utiliser le système WAAS (Wide Area Augmentation System). Si le satellite WAAS est en vue, la précision du récepteur sera améliorée.

Ce système DGPS permet de corriger les erreurs des satellites par comparaison de ces erreurs par rapport à une base référentielle terrestre.

Ordre de grandeur des erreurs :

- | | |
|---|----|
| ✓ Données des éphémérides | 2m |
| ✓ Traversée de l'ionosphère | 4m |
| ✓ Traversée de la troposphère | 1m |
| ✓ Réflexion du signal sur les obstacles | 2m |
| ✓ Horloges satellites | 2m |

II.3.1.5. Le système de coordonnées locales :

Le système de coordonnées utilisé par le système GPS (ECEF) est sans réel intérêt pour l'utilisateur. Une conversion dans un système plus pratique est nécessaire.

Le système le plus approprié est le système E, N, U (East, North, Up) qui s'exprime en général par :

- ✓ La latitude Φ
- ✓ La longitude λ
- ✓ L'altitude h
- ✓ N est la distance mesurée le long de la droite localement perpendiculaire à l'ellipsoïde entre l'observateur et l'intersection avec l'axe polaire

Il faut également tenir compte que la terre n'est pas une sphère mais un ellipsoïde dont les demi grands axes s'expriment par les valeurs a et b .

Le système ENU le plus couramment utilisé est le système WGS84 (World Geodetic Survey 1984).

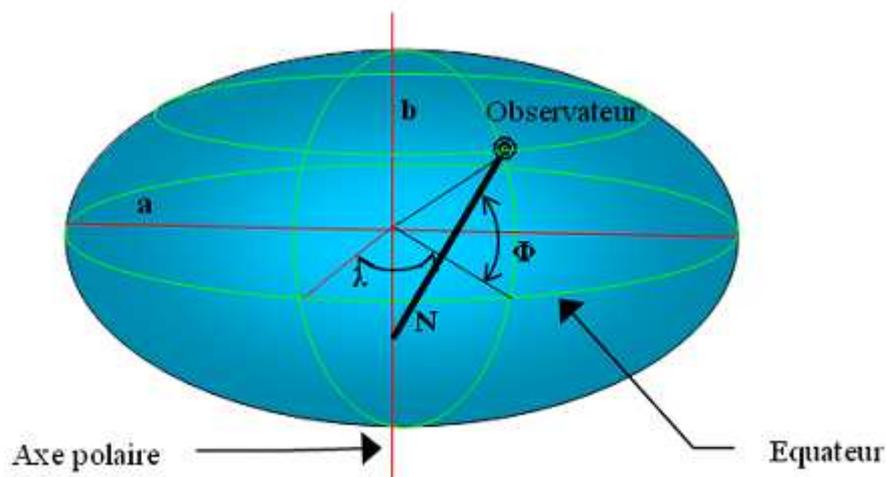


Figure II.14 : Système de coordonnées locales.

II.3.1.6. Les projections :

La terre est un ellipsoïde qui est par définition non développable. Il est nécessaire de trouver des artifices afin de pouvoir représenter la surface terrestre sur une feuille de papier qui elle est plane. Le principe de base est de projeter la surface de la terre sur une surface qui elle est développable (cône ou cylindre). Il y a deux types de représentations, elles sont caractérisées par la façon de représenter un cercle.

Chapitre II : GEOLocalISATION.

- ✓ La représentation conforme : l'image d'un cercle reste un cercle, les angles sont conservés.
- ✓ La représentation équivalente : la surface d'un cercle est représentée par une ellipse de même aire, les surfaces sont conservées.

Vous trouverez ci-après deux exemples de projections, les coniques et les cylindriques. Ce ne sont pas les seuls types de projections, il en existe une grande quantité mais ce sont les plus utilisés.

❖ Les représentations coniques.

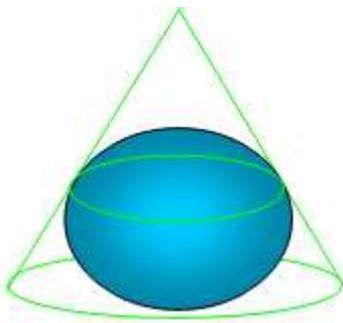


Figure II.15 : Directe tangente.

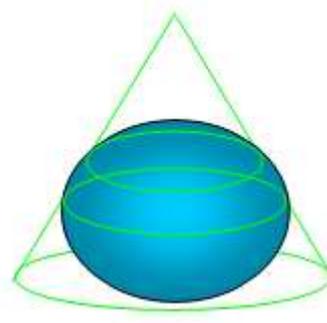


Figure II.16 : Directe sécante.

Exemple de représentation conique :

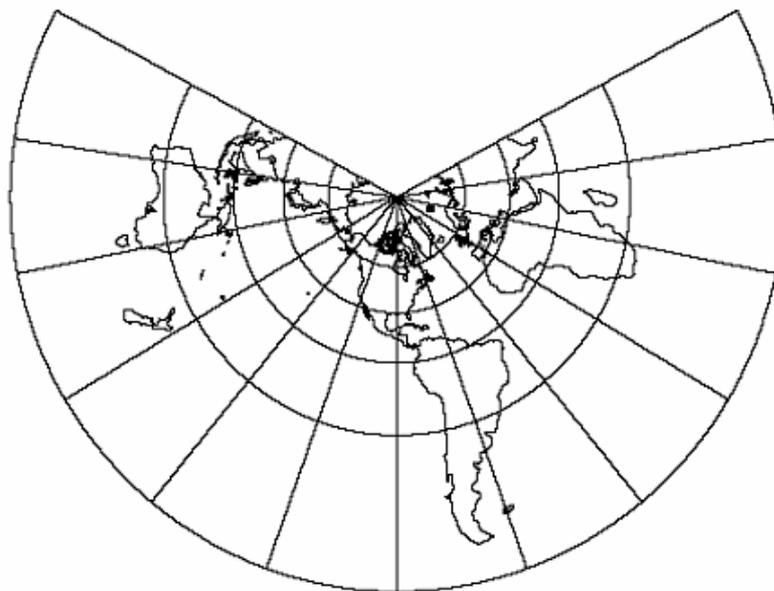


Figure II.17 : représentation conique.

Chapitre II : GEOLOCALISATION.

❖ Les représentations cylindriques.

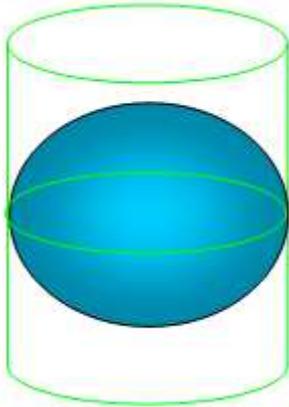


Figure II.18 : Directe.

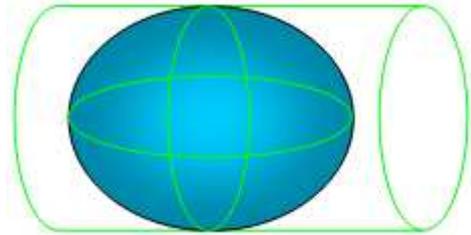


Figure II.19 : Transverse.

Exemple de représentation cylindrique :

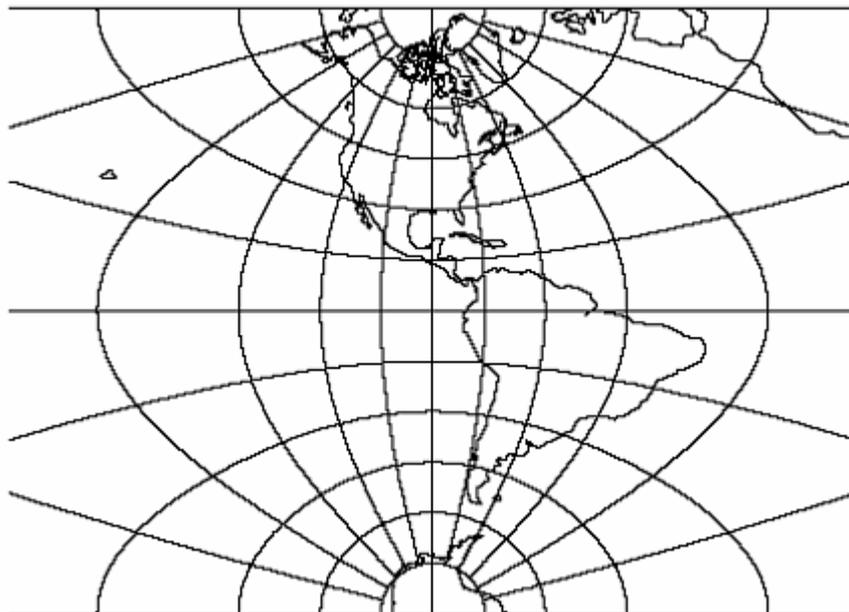


Figure II.20 : représentation cylindrique .

❖ La projection Universal Transverse Mercator (UTM).

La projection cylindrique UTM couvre le monde entier et est constituée de 60 fuseaux de 6 degrés d'amplitude en longitude.

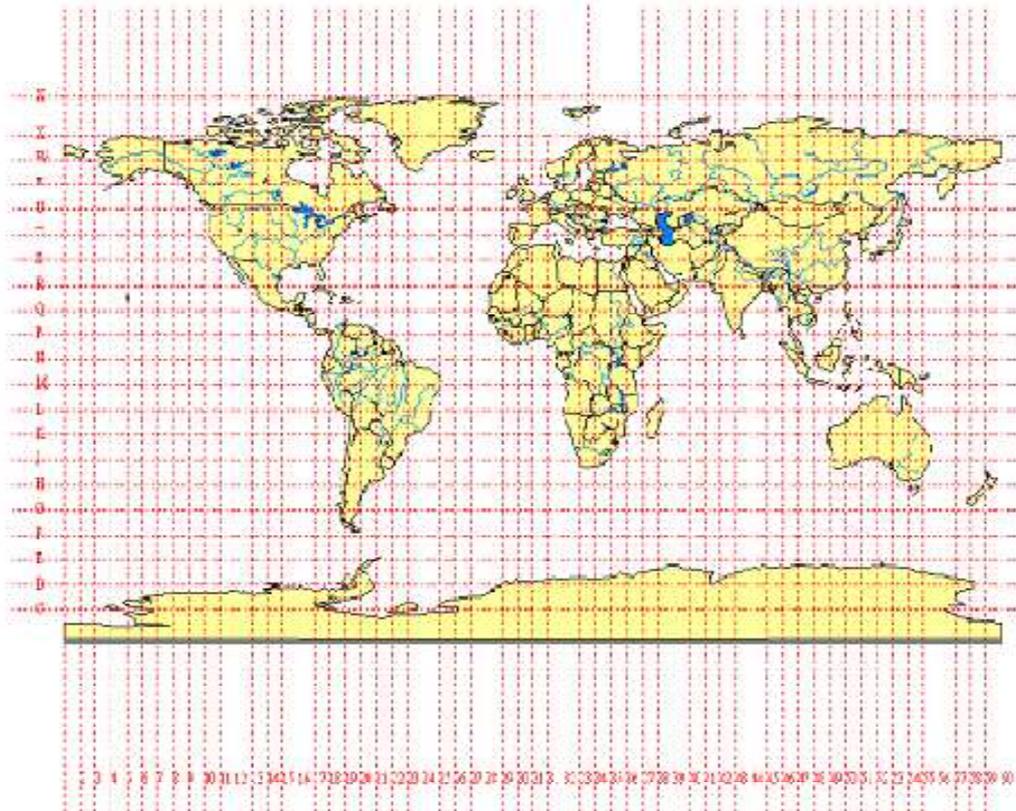


Figure II.21 : La projection UTM .

II.3.2. A-GPS (Assisted GPS):

L'Assisted GPS (A-GPS) a été conçu par Giovanni Vannucci, chercheur au centre de recherche de Lucent Technologies, Bell Labs. Au milieu des années 90, il entreprend de localiser les téléphones portables à l'aide de liaisons satellites afin de pouvoir secourir les gens en cas d'urgence. De sa collaboration avec Bob Richton, du Wireless Technology Laboratory, naît la méthode A-GPS qui réduit le temps nécessaire pour établir la première connexion satellite (TTFF pour Time To First Fix). Le délai passe de plusieurs minutes avec un système GPS traditionnel à quelques secondes avec l'A-GPS.

Contrairement au GPS, qui nécessite un récepteur et une antenne, l'A-GPS travaille en relation avec un serveur A-GPS hébergé chez l'opérateur. Le terminal mobile, muni d'un récepteur GPS miniaturisé, envoie une requête au serveur par le réseau IP. Celui-ci, qui connaît en temps réel le positionnement des satellites, sert d'aiguilleur et indique au terminal les signaux GPS à suivre. Avec cette méthode, le récepteur A-GPS du terminal mobile peut, à la différence des récepteurs GPS traditionnels, détecter des signaux de très faible amplitude.

Chapitre II : GEOLOCALISATION.

Si le mobile parvient à rester connecté au réseau de l'opérateur dans des endroits souterrains (parking...), il peut encore ?" théoriquement ?" fournir des données sur sa position. En revanche, dès que le mobile ne reçoit plus le réseau de l'opérateur et perd le signal GPS, les données ne sont plus rafraîchies. Le niveau de précision est élevé puisqu'il est annoncé entre 5 et 10 mètres contre 3 à 50 mètres avec le GPS traditionnel. De plus, le terminal mobile, à l'image du dernier iPAQ hw6515 de HP, n'a pas besoin d'antenne et de récepteur externe.

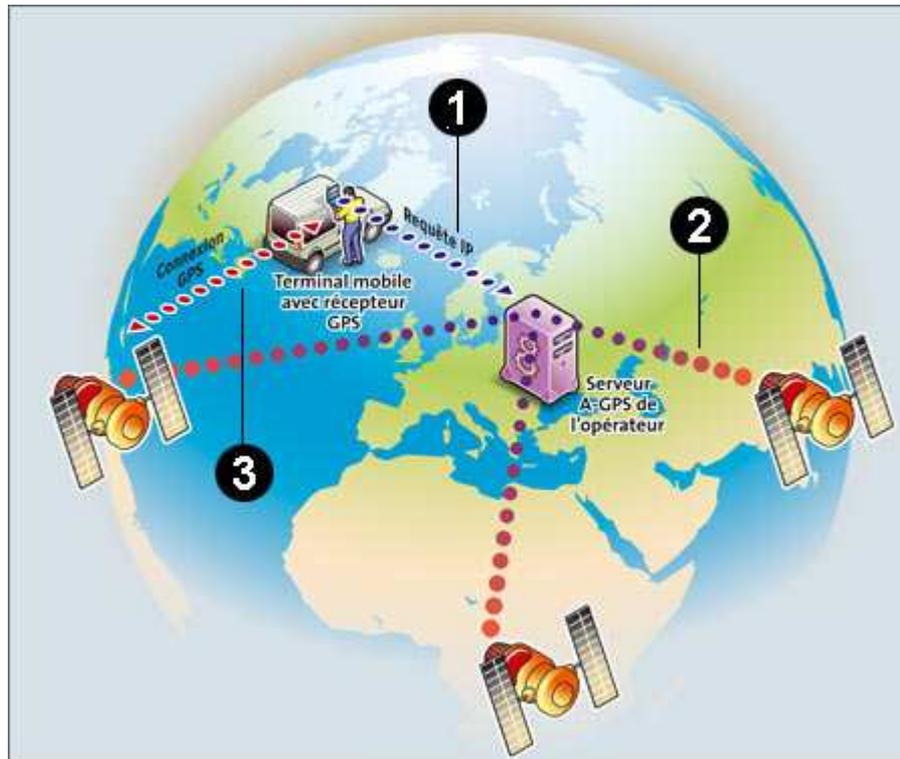


Figure II.22 : A-GPS.

1. Pour établir la liaison avec le satellite, le terminal mobile envoie une requête au serveur A-GPS hébergé chez l'opérateur.
2. Le serveur, qui suit en temps réel le positionnement des satellites, indique au terminal le signal GPS qu'il faut suivre.
3. Le terminal mobile fonctionne alors de manière autonome. En cas de perte de signal, il relance une requête auprès du serveur.

II.3.3. E-OTD (Enhanced Observed Time Difference):

Une station doit recevoir un signal synchrone de la part du MS ; la différence de temps de transmission entre le MS et deux BTS² décrit une hyperbole. Avec trois stations on peut estimer la position du MS grâce à l'intersection des hyperboles.

² BTS équipement radio émetteur/récepteur qui communique avec le MS.

L'exactitude de la position est de 125m, mais à la différence du GPS cette méthode ne dépend pas de la clarté du ciel.

II.3.4. TOA (Time Of Arrival):

TOA calcule le temps de transmission entre la station mobile et le BTS et vice versa. Considérant le fait que le temps de propagation d'une onde radio est connu, il est alors possible d'estimer la distance qui sépare la station mobile du BTS.

Cette méthode permet de localiser l'utilisateur dans un cercle qui a pour rayon la distance qui sépare le BTS de la MS et qui a pour centre le BTS.

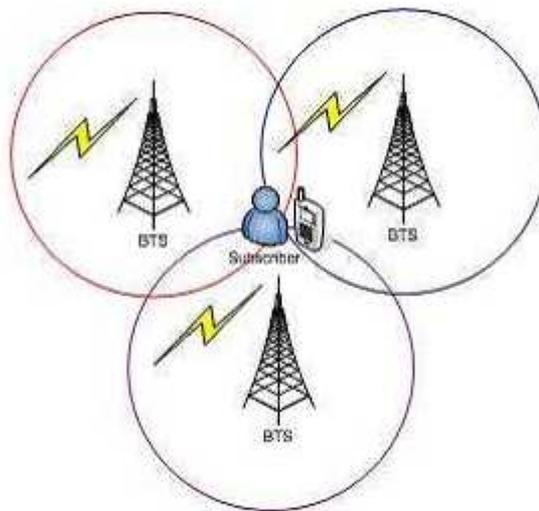


Figure II.23 : TOA.

II.3.5. Cell-ID:

Cette méthode est basée sur le réseau GSM et plus particulièrement sur l'identification de la cellule. Elle utilise l'identifiant de la BTS dans lequel la station mobile est actuellement enregistrée. La topologie du réseau GSM est ensuite utilisée afin d'estimer la position du mobile. La précision de cette méthode dépend principalement de la taille de la cellule ainsi que sur l'environnement avoisinant (c'est-à-dire rural ou urbain).

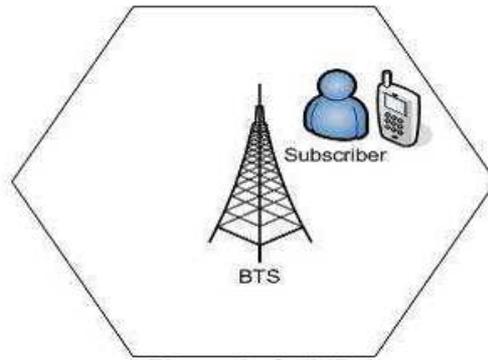


Figure II.24 : Cell-ID.

II.4. Autres GNSS (Système de navigation globale par satellite):

Il existe bien d'autres systèmes de navigation tels que :

II.4.1. Galileo :

Est le système lancé par l'Europe dans le but de ne plus être dépendant du système américain. Il sera opérationnel dès 2013. Le système Galileo aura quasiment la même structure que le NAVSTAR GPS et les récepteurs pourront fonctionner avec NAVSTAR GPS et Galileo. Ce système sera compatible avec EGNOS qui est un réseau de stations au sol en Europe, ce qui permettra d'être très précis.

II.4.2. GLONASS :

Est le système russe, fonctionnel depuis des années. Il est en partie compatible avec le système américain. Il est composé d'une flotte de 24 satellites.

II.4.3. Beidou :

Est le système chinois. Actuellement, il n'est composé que de cinq satellites géostationnaires au dessus de la Chine, couvrant ainsi l'intégralité de son territoire. Une deuxième phase de Beidou va être d'envoyer 30 satellites non géostationnaires afin de couvrir en entier la Terre. L'unique raison d'être de ce système est de s'affranchir de la dépendance des Etats-Unis. Le Japon et l'Inde sont en train de développer des systèmes locaux, composés de satellites géostationnaires.

Tous les principes de fonctionnements, calculs et problèmes physiques propres au GPS le sont également pour tous les GNSS.

II.5. Conclusion:

Le GPS est la méthode de localisation la plus utilisée, elle est composée de trois segments : spatial, contrôle et utilisateur. Cette technologie fonctionne avec les satellites qui émettent à des fréquences de 1575,42 MHz et 1227,6 MHz.

Cependant de nouvelles technologies concurrentes ont vu le jour avec la géolocalisation par satellite, par wifi, par GSM .Le positionnement par les systèmes sans fil est une aubaine pour les opérateurs de communication mobile qui sont toujours à la recherche de nouveaux services à offrir à leur clientèle.

Toutes ces techniques de géolocalisation ont pour but d'améliorer le quotidien des êtres humains.

Chapitre III :

Analyse et Conception



ANDROID

III.1. Introduction :

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

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

III.2. Présentation de l'UML : [4]

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

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

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

III.2.1. Modélisation avec L'UML : [5]

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

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

III.2.2. La démarche de modélisation avec L'UML :

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

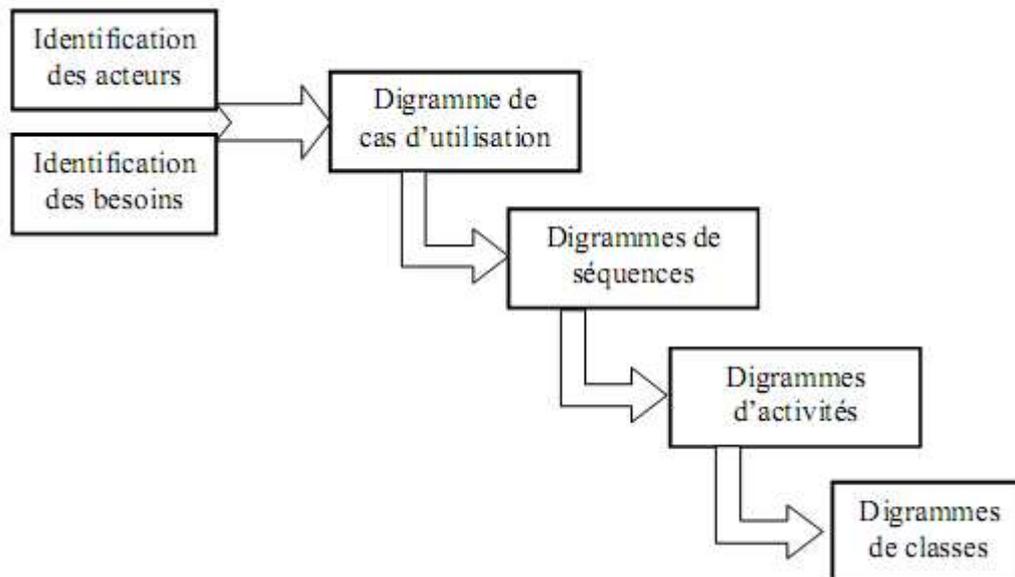


Figure III.1 : La démarche adoptée pour la modélisation.

III.3. Analyse :

Cette partie comprend l'identification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système ainsi que les cas d'utilisation.

III.3.1. Identification des besoins :

L'objectif principal de l'application est de permettre à l'utilisateur de se repérer dans le campus de Bastos. Il est parfois un peu compliqué de trouver le lieu où l'on doit se rendre, surtout si l'on est un nouvel étudiant. L'utilisateur est capable de se localiser sur le campus.

La liste des fonctionnalités présentes sur l'application est la suivante :

- ✓ Se localiser.
- ✓ Afficher les bâtiments d'une catégorie donnée
- ✓ Recherche d'un bâtiment sur le campus
- ✓ Itinéraire piéton ou en voiture vers un bâtiment du campus.
- ✓ Navigation piéton ou en voiture via le GPS vers un bâtiment du campus.
- ✓ Consulter la fiche détaillée de chaque bâtiment (Adresse, horaires, description).
- ✓ Appeler le bâtiment / Consulter son site internet.
- ✓ Consulter le menu du RU.

III.3.2. Identification des acteurs : [6]

Définition: Un acteur représente un ensemble de rôles joués par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Dans le cas de notre application, nous avons un acteur qui interagit avec le système qui est :

- ✓ **User :** Cet acteur doit lancer l'application, choisir l'action à faire et selon son choix, il peut accéder à l'interface « map » ou il se localisera au sein du campus. Il peut aussi accéder à l'interface « rechercher un lieu » afin de localiser des bâtiments recherchés dans le campus pour s'y rendre via un itinéraire.

III.3.3. Spécification des tâches :

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

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

Acteurs	Tâches
Utilisateur (user)	T1 : Lancer l'application. T2 : Accéder a la map. T3 : Rechercher un lieu. T4 : Déterminer un itinéraire ou une navigation. T5 : Se localiser. T6 : Réinitialiser la map. T7 : Consulter l'itinéraire ou la navigation. T8 : Recherche un lieu. T9 : Accéder aux catégories. T10 : Accéder aux bâtiments. T11 : Accéder a la fiche détaillée. T12 : Accéder a la partie informations générales. T13 : Accéder a la partie autres. T14 : Téléphoner au bâtiment. T15 : Consulter le site internet du bâtiment. T16 : Accéder a la partie itinéraire / navigation. T17 : Demander l'itinéraire. T18 : Localiser le bâtiment.

Tableau III.1 : Spécification des tâches.

III.3.4. Spécification des scénarios : [6]

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

Note :

Durant notre étude les scénarios seront symboliser par **si** (**i** : représente le numéro scénario). Le tableau ci-dessous récapitule les différents scénarios créés par les cas d'utilisation cités.

Acteurs	Taches	Scénarios
Utilisateur (user)	T1 : Lancer l'application.	S1 : Cliquer sur l'icône de l'application.
	T2 : Accéder a la map.	S2 : Cliquer sur le bouton « maps » dans l'interface d'accueil.
	T3 : Rechercher un lieu.	S3 : Cliquer sur l'icône qui représente « rechercher un lieu » dans l'interface maps.
	T4 : Déterminer un itinéraire ou une navigation.	S4 : Cliquer sur l'icône qui représente « déterminer un itinéraire » dans l'interface maps.
	T5 : Se localiser.	S5 : Cliquer sur l'icône qui représente « Se localiser » dans l'interface maps.
	T6 : Réinitialiser la map.	S6 : Cliquer sur l'icône qui représente « Réinitialiser la map » dans l'interface maps.
	T7 : Consulter l'itinéraire ou la navigation.	S7 : Afficher l'itinéraire ou la navigation.
	T8 : Recherche un lieu.	S8 : Cliquer sur le bouton « rechercher un lieu » dans l'interface d'accueil.
	T9 : Accéder aux catégories.	S9 : Cliquer sur l'une des catégories proposées.
	T10 : Accéder aux bâtiments.	S10 : Cliquer sur l'un des bâtiments proposés.
	T11 : Accéder a la fiche détaillée.	S11 : Afficher la fiche détaillée d'un bâtiment.
	T12 : Accéder a la partie informations générales.	S12 : Cliquer sur l'ongle « partie informations générales » dans la fiche détaillée d'un bâtiment.
	T13 : Accéder a la partie autres.	S13 : Cliquer sur l'ongle « partie autres » dans la fiche détaillée d'un bâtiment.
	T14 : Téléphoner au bâtiment.	S14 : Cliquer sur l'icône « téléphone » dans l'ongle « partie autres ».
	T15 : Consulter le site internet du bâtiment.	S15 : Cliquer sur l'icône « site internet » dans l'ongle « partie autres ».

Chapitre III : Analyse et Conception

T16 : Accéder a la partie itinéraire / navigation.	S16 : Cliquer sur l'ongle « partie itinéraire » dans la fiche détaillée d'un bâtiment.
T17 : Demander l'itinéraire.	S17 : Cliquer sur l'icone « itinéraire » dans l'ongle « partie itinéraire ».
T18 : Localiser le bâtiment.	S18 : Cliquer sur l'icone « localiser » dans l'ongle « partie itinéraire »

Tableau III.2 : Spécification des scénarios.

III.3.5. Les cas d'utilisation : [6]

Définition : Un cas d'utilisation (**user case**) représente un ensemble de séquence d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système.

III.3.5.1. Description des cas d'utilisations :

Nous procéderons aux descriptions des cas d'utilisation dans notre système :

Cas d'utilisation : Accéder à la partie Informations Générales.

Scénarios1 : S1,S2,S3,S9,S10,S11,S12 (via le bouton "map")

Scénarios2 : S1,S8,S9,S10,S11,S12 (via le bouton "rechercher un lieu")

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliquer sur l'icône.
- 2- Le système affiche l'ecran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « rechercher un lieu » situé en haut de l'interface.
- 6- Le système construit et affiche l'interface « catégorie ».
- 7- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 8- Le système construit et affiche l'interface « bâtiment ».
- 9- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 10- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 11- L'utilisateur clique ensuite sur l'ongle « informations générales » du bâtiment.

Description2 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche la page d'accueil.
- 3- L'utilisateur clique sur le bouton « rechercher un lieu ».
- 4- Le système construit et affiche l'interface « catégorie ».

- 5- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 6- Le système construit et affiche l'interface « bâtiment ».
- 7- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 8- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 9- L'utilisateur clique ensuite sur l'ongle « informations générale » du bâtiment.

Figure III.2 : Cas d'utilisation « Accéder a la partie Informations Générales ».

Cas d'utilisation : Téléphoner au bâtiment.

Scénarios1 : S1,S2,S3,S9,S10,S11,S13,S14 (via le bouton “map”)

Scénarios2 : S1,S8,S9,S10,S11,S13,S14 (via le bouton “rechercher un lieu”)

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « rechercher un lieu » situé en haut de l'interface.
- 6- Le système construit et affiche l'interface « catégorie ».
- 7- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 8- Le système construit et affiche l'interface « bâtiment ».
- 9- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 10- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 11- L'utilisateur clique ensuite sur l'ongle « autres » du bâtiment.
- 12- L'utilisateur clique sur l'icone téléphone pour appeler le bâtiment choisi.

Description2 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « rechercher un lieu ».
- 4- Le système construit et affiche l'interface « catégorie ».
- 5- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 6- Le système construit et affiche l'interface « bâtiment ».
- 7- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 8- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 9- L'utilisateur clique ensuite sur l'ongle « autres » du bâtiment.
- 10- L'utilisateur clique sur l'icone téléphone pour appeler le bâtiment choisi.

Figure III.3 : Cas d'utilisation « Téléphoner au bâtiment ».

Cas d'utilisation : Consulter le site internet du bâtiment.**Scénarios1 :** S1,S2,S3,S9,S10,S11,S13,S15

(via le bouton "map")

Scénarios2 : S1,S8,S9,S10,S11,S13,S15

(via le bouton "rechercher un lieu")

Rôle : utilisateur**Description1 :**

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « rechercher un lieu » situé en haut de l'interface.
- 6- Le système construit et affiche l'interface « catégorie ».
- 7- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 8- Le système construit et affiche l'interface « bâtiment ».
- 9- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 10- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 11- L'utilisateur clique ensuite sur l'ongle « autres » du bâtiment.
- 12- L'utilisateur clique sur le bouton pour accéder au site internet du bâtiment.

Description2 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « rechercher un lieu ».
- 4- Le système construit et affiche l'interface « catégorie ».
- 5- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 6- Le système construit et affiche l'interface « bâtiment ».
- 7- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 8- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 9- L'utilisateur clique ensuite sur l'ongle « autres » du bâtiment.
- 10- L'utilisateur clique sur le bouton pour accéder au site internet du bâtiment.

Figure III.4 : Cas d'utilisation « Consulter le site internet du bâtiment ».

Cas d'utilisation : Localiser le bâtiment.**Scénarios1 :** S1,S2,S3,S9,S10,S11,S16,S18

(via le bouton "map")

Scénarios2 : S1,S8,S9,S10,S11,S16,S18

(via le bouton "rechercher un lieu")

Rôle : utilisateur**Description1 :**

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « rechercher un lieu » situé en haut de

l'interface.

- 6- Le système construit et affiche l'interface « catégorie ».
- 7- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 8- Le système construit et affiche l'interface « bâtiment ».
- 9- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 10- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 11- L'utilisateur clique ensuite sur l'onglet « itinéraire / navigation » du bâtiment.
- 12- L'utilisateur clique sur l'icone « localiser » afin de localiser le bâtiment.

Description2 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « rechercher un lieu ».
- 4- Le système construit et affiche l'interface « catégorie ».
- 5- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.
- 6- Le système construit et affiche l'interface « bâtiment ».
- 7- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 8- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 9- L'utilisateur clique ensuite sur l'onglet « itinéraire / navigation » du bâtiment.
- 10- L'utilisateur clique sur l'icone « localiser » afin de localiser le bâtiment.

Figure III.5 : Cas d'utilisation « Localiser le bâtiment ».

Cas d'utilisation : Consulter Itinéraire / Navigation.

Scénarios1 : S1,S2,S4,S7

(via le bouton “map”)

Scénarios2 : S1,S8,S9,S10,S11,S16,S17

(via le bouton “rechercher un lieu”)

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icone.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « déterminer un itinéraire » situer en haut de l'interface.
- 6- Le système construit et affiche l'interface « itinéraire ».

Description2 :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « rechercher un lieu ».
- 4- Le système construit et affiche l'interface « catégorie ».
- 5- L'utilisateur choisi une catégorie parmi la liste des catégories proposées.

- 6- Le système construit et affiche l'interface « bâtiment ».
- 7- L'utilisateur choisi un bâtiment parmi la liste des bâtiments proposés.
- 8- Le système affiche l'interface « la fiche détaillée » du bâtiment.
- 9- L'utilisateur clique ensuite sur l'ongle « itinéraire / navigation » du bâtiment.
- 10- L'utilisateur clique sur l'icone « itinéraire ».
- 11- Le système affiche et construit l'itinéraire.

Figure III.6 : Cas d'utilisation « Consulter l'itinéraire / la navigation ».

Cas d'utilisation : Se Localiser.

Scénarios : S1,S2,S5 (via le bouton “map”)

Rôle : utilisateur

Description :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icone.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- Le système construit et affiche l'interface « géo localisation ».

Figure III.7 : Cas d'utilisation « Se localiser ».

Cas d'utilisation : Réinitialiser la map.

Scénarios : S1,S2,S6 (via le bouton “map”)

Rôle : utilisateur

Description :

- 1- L'utilisateur lance l'application après avoir cliqué sur l'icone.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur clique sur le bouton « map ».
- 4- Le système affiche l'interface « map »
- 5- L'utilisateur clique sur le bouton « réinitialisation » situer en haut de l'interface.
- 6- Le système réinitialise la map et affiche la position de l'utilisateur.

Figure III.8 : Cas d'utilisation « Réinitialiser la map ».

III.4. Conception :

Le processus de conception de mon projet se caractérise par deux niveaux : le niveau applicatif et le niveau données.

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

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

Le niveau données concerne l'organisation conceptuelle, logique et physique des données manipulées. Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application et à travers la conception du niveau applicatif nous allons dégager les classes significatives, dès lors on peut élaborer la conception de la base de données.

III.4.1. Le niveau applicatif :

III.4.1.1. Le diagramme des cas d'utilisation: [7]

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

Définition : Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

- ✓ **La relation d'inclusion (include) :** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.

Chapitre III : Analyse et Conception

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

En bleu sont représentés les états auxquels nous voulions arrivés dans nos scénarios.

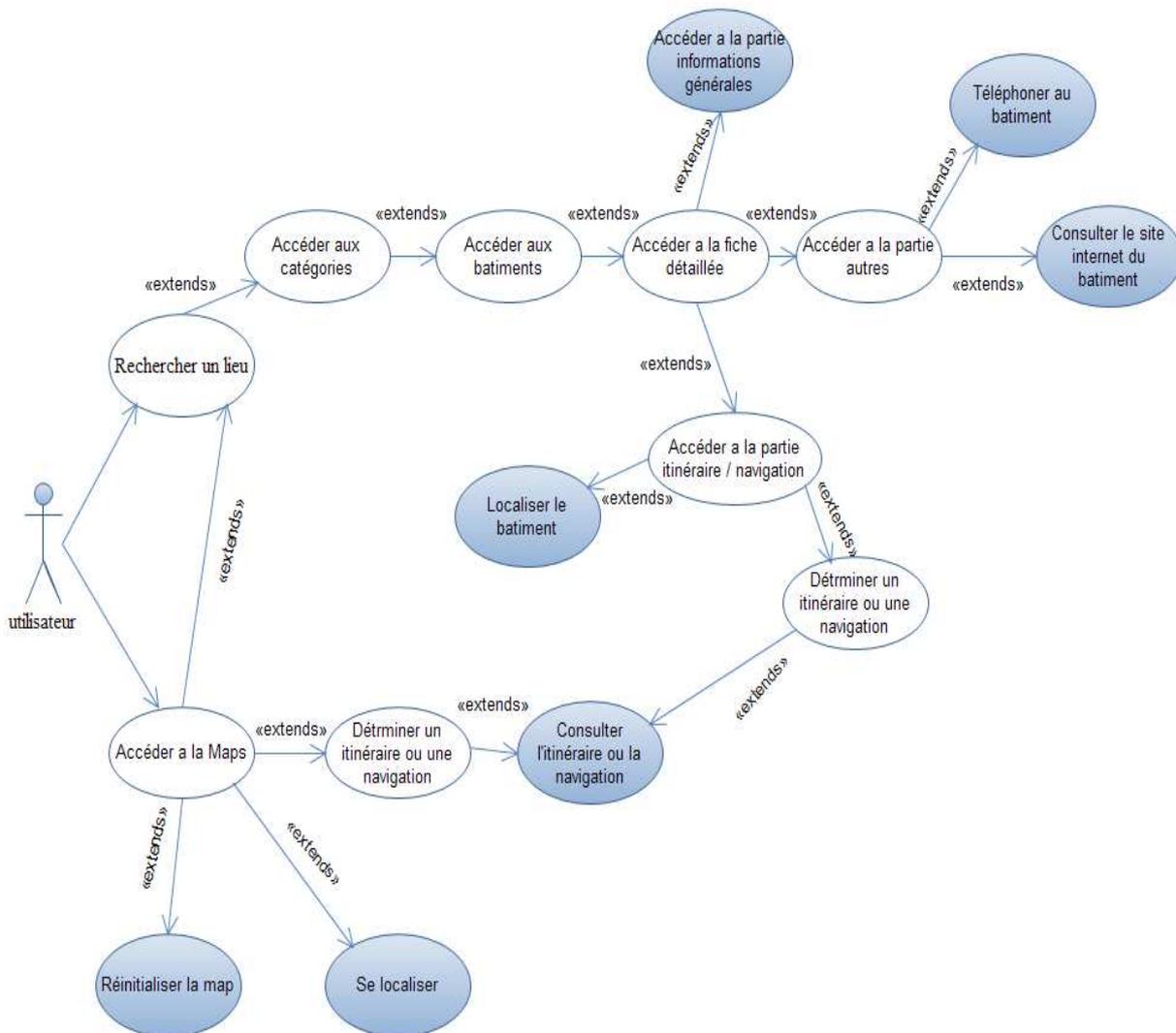


Figure III.9 : Diagramme de cas d'utilisation générale.

III.4.1.2. Les diagrammes de séquences :

Un diagramme de séquence permet de spécifier les interactions qui existent entre un groupe d'objet selon un point de vue temporel, on y met l'accent sur la chronologie des envois de message. Le diagramme de séquence est toujours de haut vers le bas, il illustre l'ordre dans lequel les messages sont envoyés entre les objets. il peut servir à illustrer un cas d'utilisation.

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

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

Les messages : ils sont représentés par des flèches directionnelles. au-dessus des flèches directionnelle figurent un texte nous informant du message envoyé entre les objets.



Figure III.10 : Diagramme de séquence de cas d'utilisation « se localiser ».

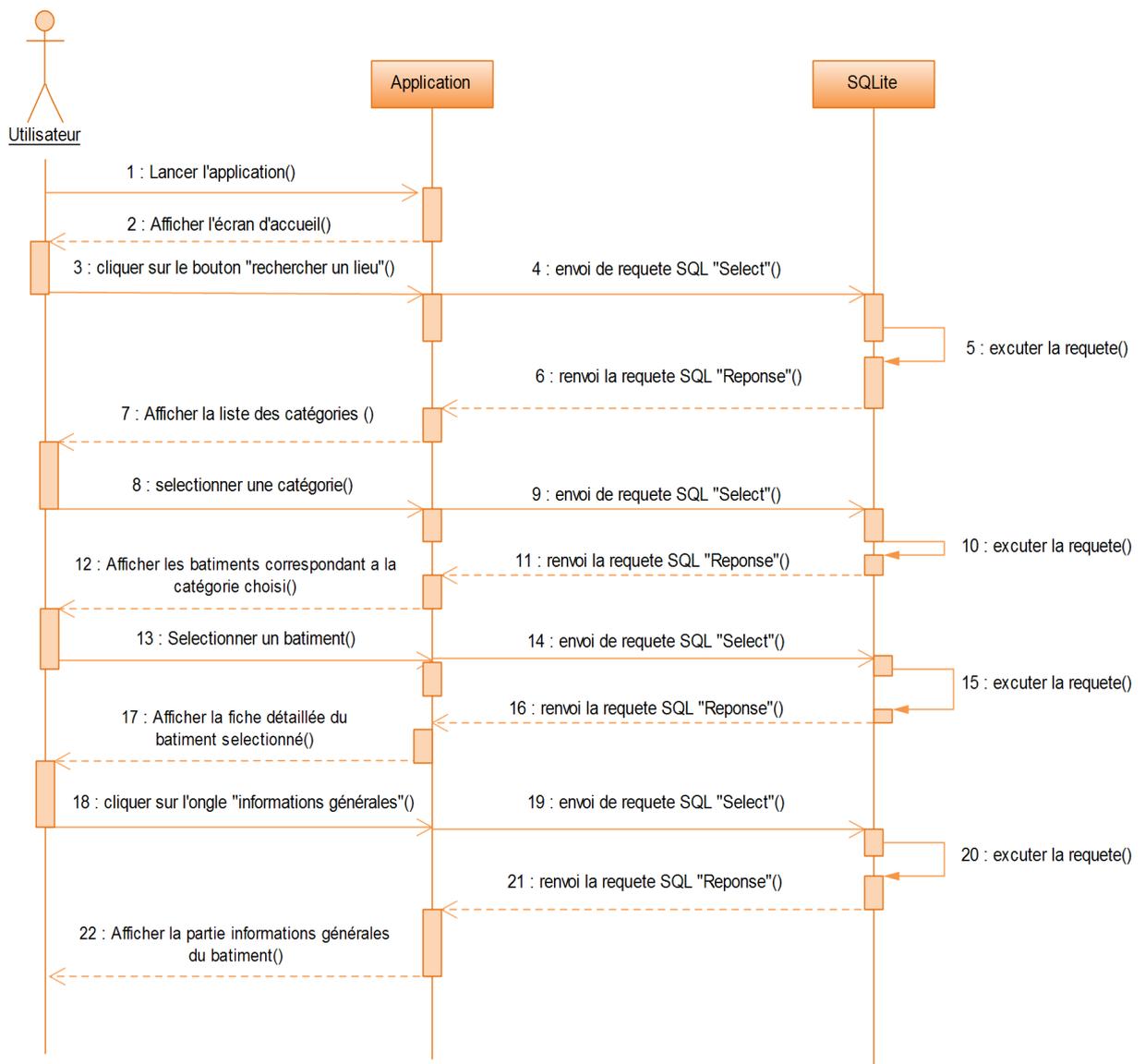


Figure III.11 : Diagramme de séquence de cas d'utilisation « Accéder au informations générales ».

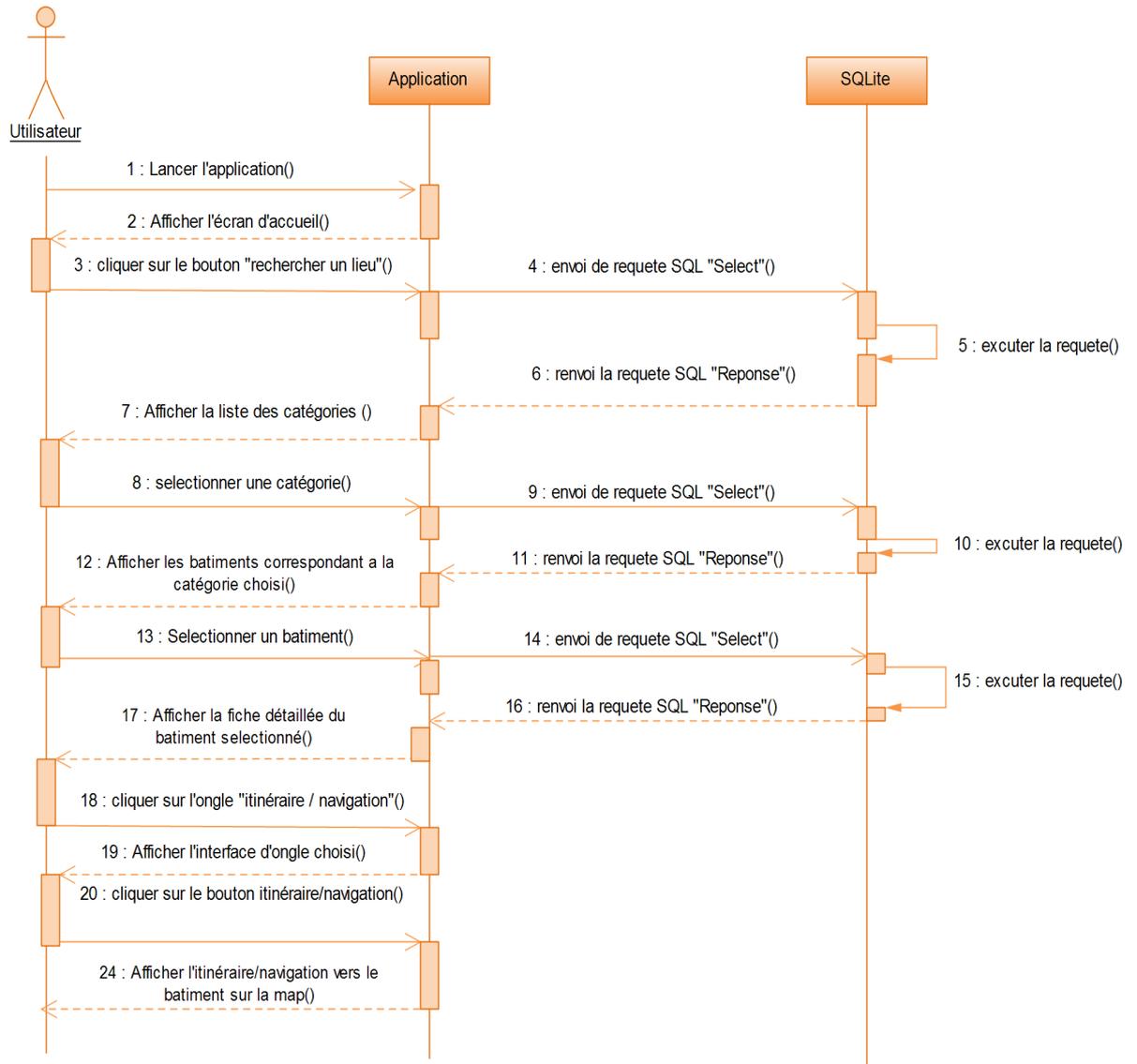


Figure III.12 : Diagramme de séquence de cas d'utilisation « Consulter l'itinéraire / la navigation ».

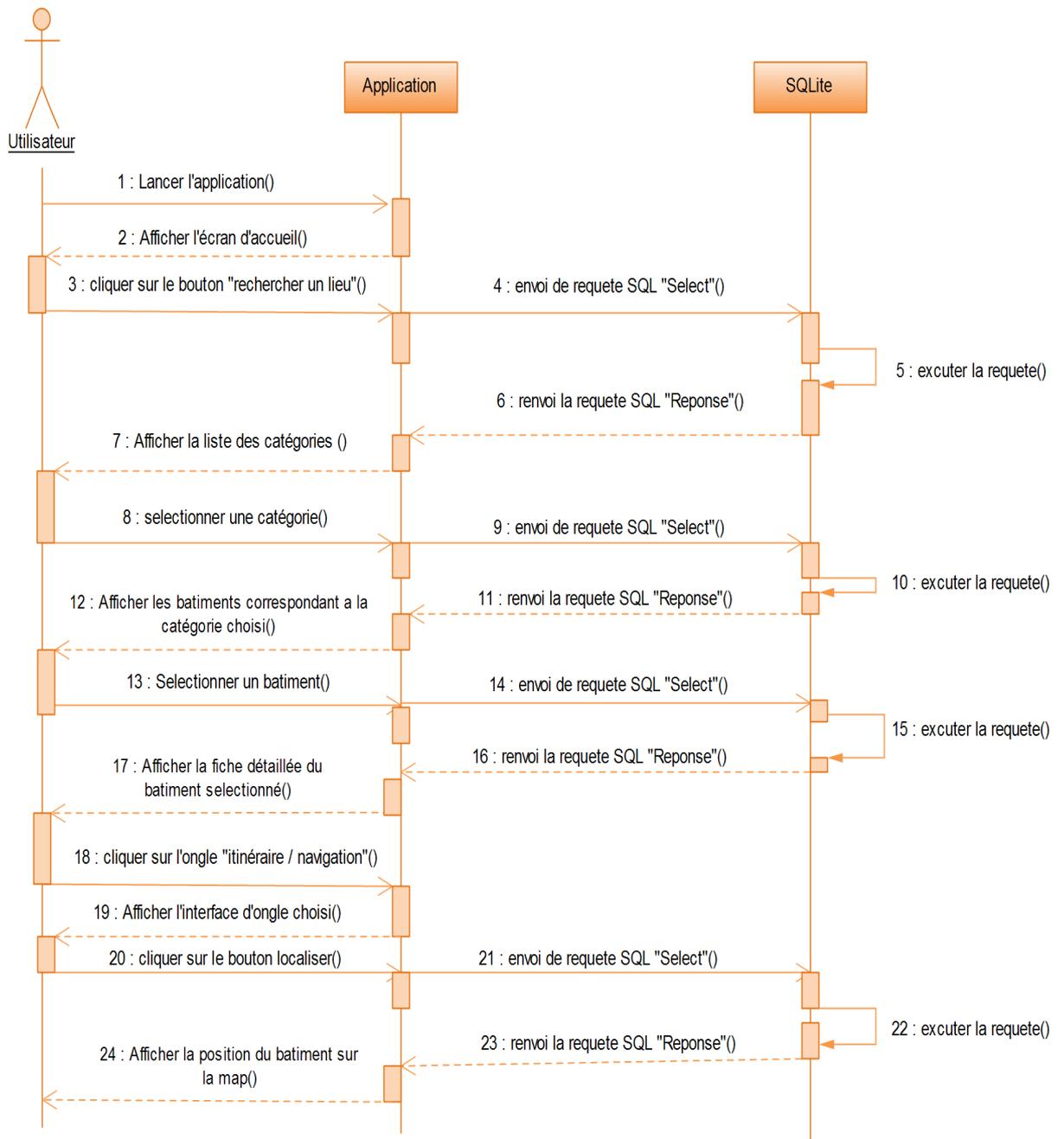


Figure III.13 : Diagramme de séquence de cas d'utilisation « Localiser un bâtiment ».

III.4.1.3. Les diagrammes d'activités : [8]

Un diagramme d'activités : apporte un point de vue complémentaire à l'aspect dynamique de la modélisation. Il offre un pouvoir d'expression très proche des langages de programmation objets. Il est donc bien adapté à la spécification détaillée des traitements en phase de réalisation. Un diagramme d'activités se concentre plutôt sur les activités entre les objets, c'est-à-dire, il met en évidence l'activité qui a lieu dans le temps, donc les opérations transmises entre les objets.

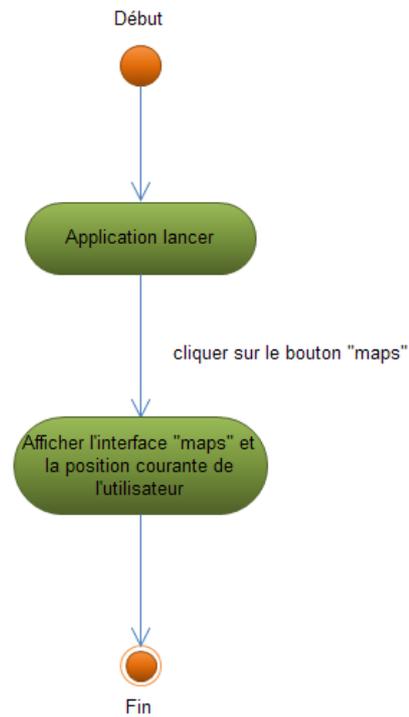


Figure III.14 : Diagramme d'activité de cas d'utilisation « Se localiser ».

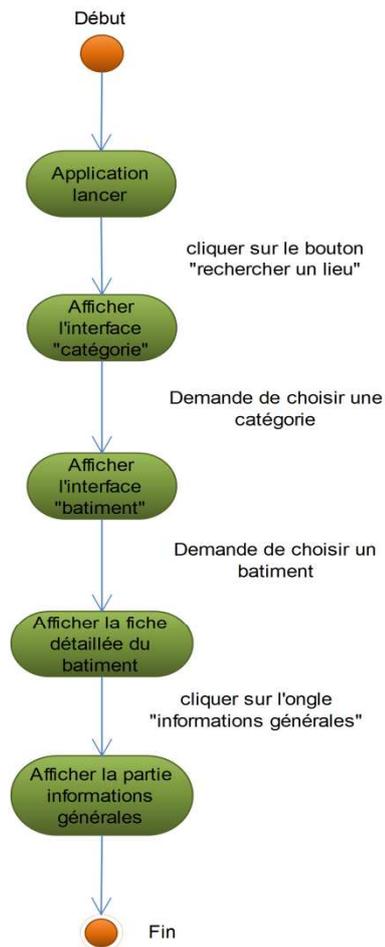


Figure III.15 : Diagramme d'activité de cas d'utilisation « Accéder au informations générales ».

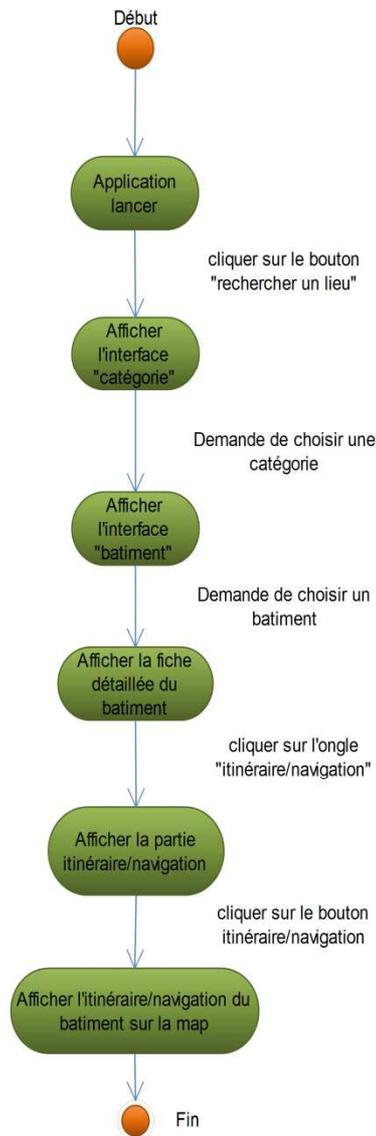


Figure III.16 : Diagramme d'activité de cas d'utilisation « Consulter l'itinéraire / la navigation ».

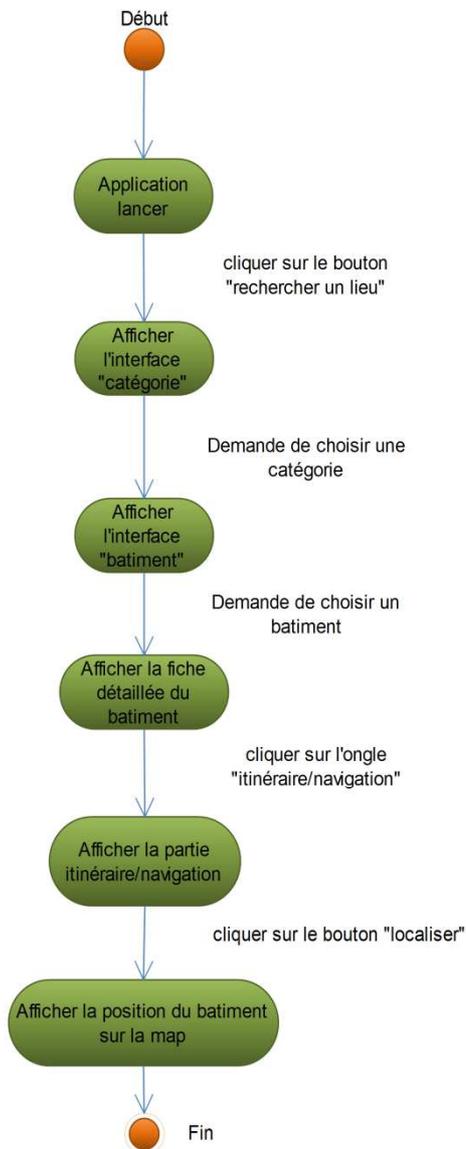


Figure III.17 : Diagramme d'activité de cas d'utilisation « Localiser un bâtiment ».

III.4.1.4. Les diagrammes de classe : [8]

Dans cette partie, nous allons passer à la modélisation de l'aspect statique de notre application, c'est-à-dire nous allons modéliser l'intérieur de notre système. Pour ce faire, nous allons utiliser les diagrammes de classes.

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Alors que les diagrammes précédents (diagrammes de séquences et diagrammes d'activités) montrent le système du point de vue dynamique, le diagramme de classe en montre la structure interne. Il contient principalement des classes reliées par des associations et chaque classe contient des attributs et des opérations.

Après avoir élaboré les diagrammes de séquences et les diagrammes d'activités pour les cas d'utilisation déjà décrit, nous allons élaborer des diagrammes de classes détaillés pour chacun d'eux.

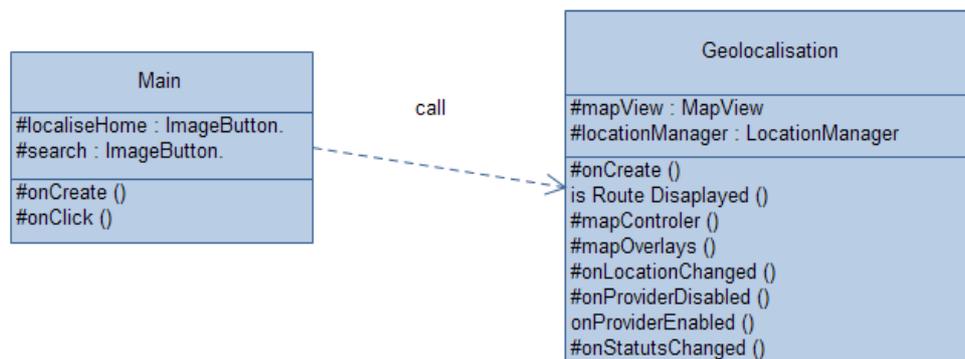


Figure III.18 : Diagramme de classe de cas d'utilisation « Se localiser ».

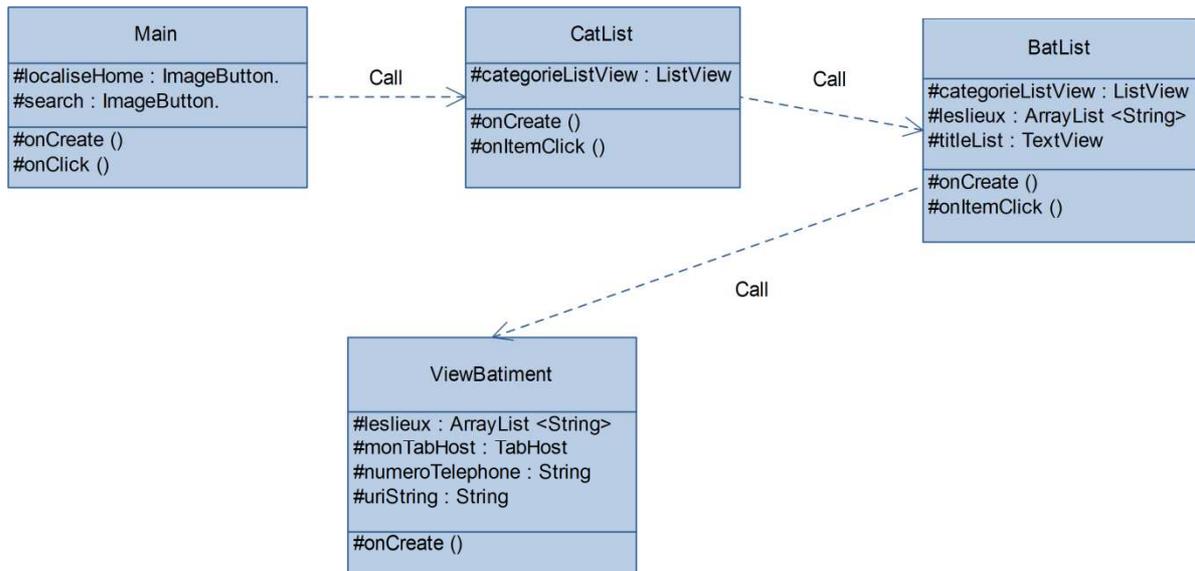


Figure III.19 : Diagramme de classe de cas d'utilisation « Accéder au informations générales ».

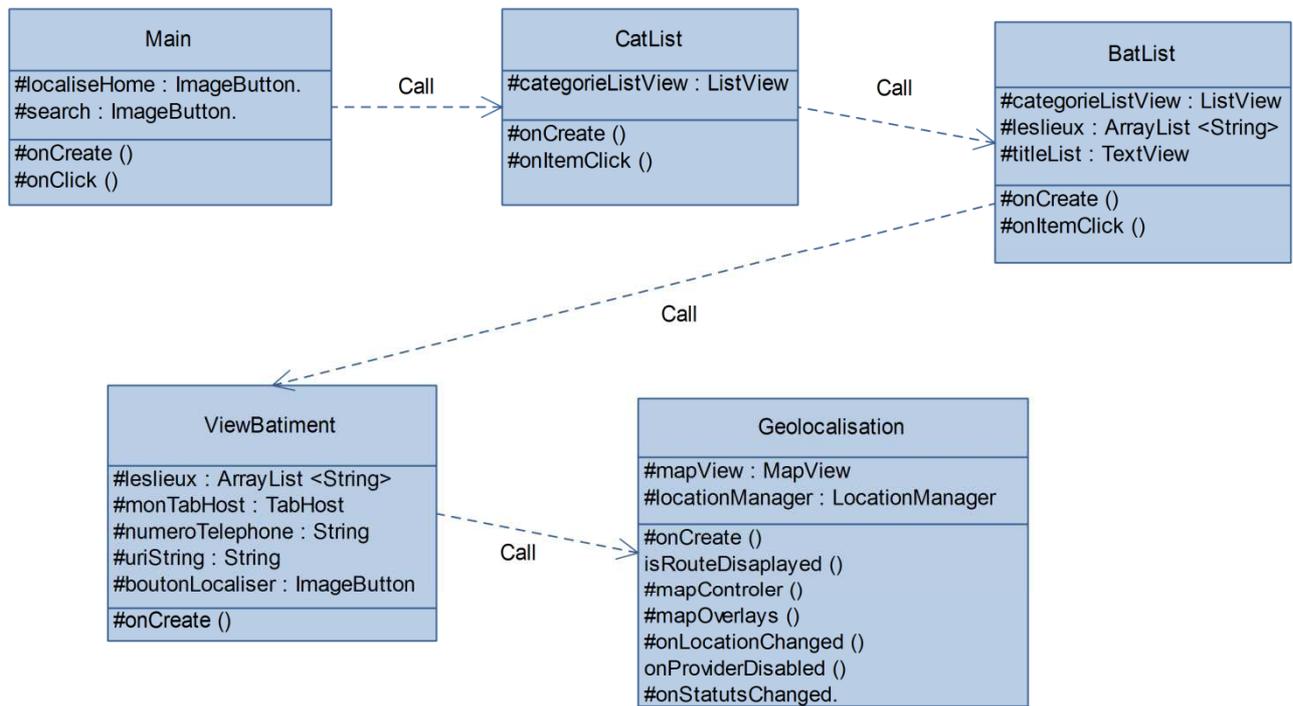


Figure III.20 : Diagramme de classe de cas d'utilisation « Localiser un bâtiment ».

III.4.2. Le niveau données:

III.4.2.1. Schéma de la base de données :

Le schéma conceptuel de la base de données est représenté par le diagramme suivant :

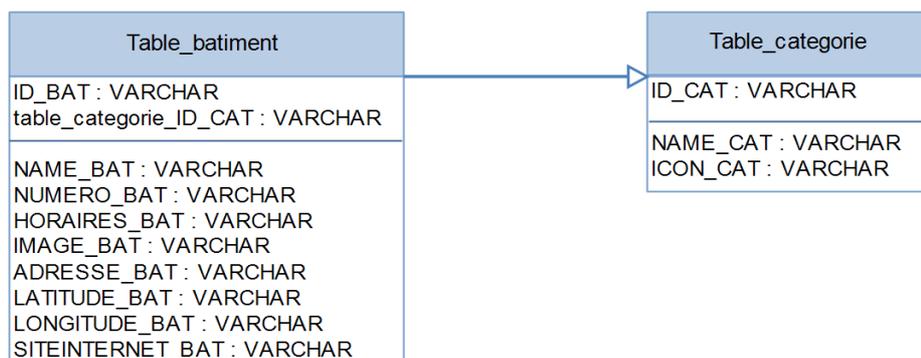


Figure III.21 : Le schéma de la base de données.

La base est séparée en deux parties (voir figure III.21) car nous n'avons besoin que de 2 objets à représenter à savoir d'un côté les bâtiments présents sur le Campus et de l'autre la catégorie de ce bâtiment afin de permettre un certain classement ou ordre dans la liste des bâtiments.

III.4.2.2. Les tables :

L'application dans tous ces cas d'utilisation manipule les données stockées dans la Base De Données, et elle contient les tables suivantes :

Chapitre III : Analyse et Conception

La Table bâtiment :

Champ	Signification	Type	Clé
ID_BAT	l'identifiant unique pour chaque bâtiment auto-incrémenté (ici il s'agit d'un String mais nous avons fait en sorte que l'auto-incrémentation se fasse quand même).	VARCHAR	Primaire (PK)
NAME_BAT	le nom du bâtiment	VARCHAR	
NUMERO_BAT	le numéro de téléphone du bâtiment	VARCHAR	
HORAIRES_BAT	Les horaires d'ouverture du bâtiment	VARCHAR	
IMAGE_BAT	le chemin vers l'image à afficher dans la fiche de description du bâtiment	VARCHAR	
LATITUDE_BAT	la latitude du bâtiment (utilisée pour la géolocalisation)	VARCHAR	
LONGITUDE_BAT	la longitude du bâtiment (utilisée pour la géolocalisation)	VARCHAR	
SITEINTERNET_BAT	le lien vers le site internet du bâtiment pour avoir plus d'informations	VARCHAR	
ID_CAT	l'identifiant de la catégorie à laquelle le bâtiment appartient	VARCHAR	Etrangère (FK)

Tableau III.22 : Table bâtiment.

La table catégorie :

Champ	Signification	Type	Clé
ID_CAT	l'identifiant unique pour chaque catégorie auto-incrémentée	VARCHAR	Primaire (PK)
NAME_CAT	le nom de la catégorie	VARCHAR	
ICON_CAT	le chemin vers l'icône représentant la catégorie	VARCHAR	

Tableau III.23 : Table catégorie.

III.4.3. Conception des maquettes préliminaires :

De cette phase d'analyse, a découlé la conception de mon application. Après avoir déterminé les scénarios d'usages, pour faire la maquette de mon application afin de voir si toutes les possibilités décrites dans mes scénarios seraient bien réalisables dans mon application finale. Ainsi, à l'aide l'outil Balsmiq, j'ai réalisé la maquette de mon application.

III.4.3.1. L'écran d'accueil :

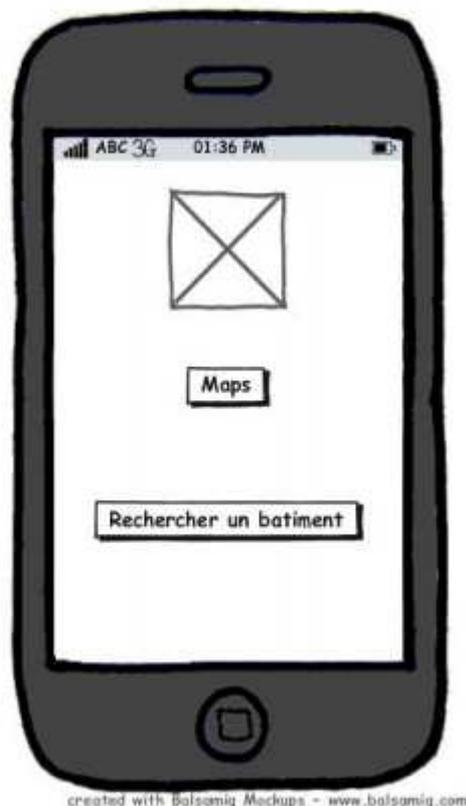


Figure III.24 : page d'accueil.

J'ai décidé de faire un écran d'accueil assez simple en termes de contenu d'information (voir figure III.24). Ainsi pour que l'utilisateur ne se perde pas, il n'est possible de réaliser que deux actions possibles via les deux boutons. J'ai aussi ajouté le logo de mon application sur cette page.

III.4.3.2. Partie Géolocalisation / Itinéraire : La map :

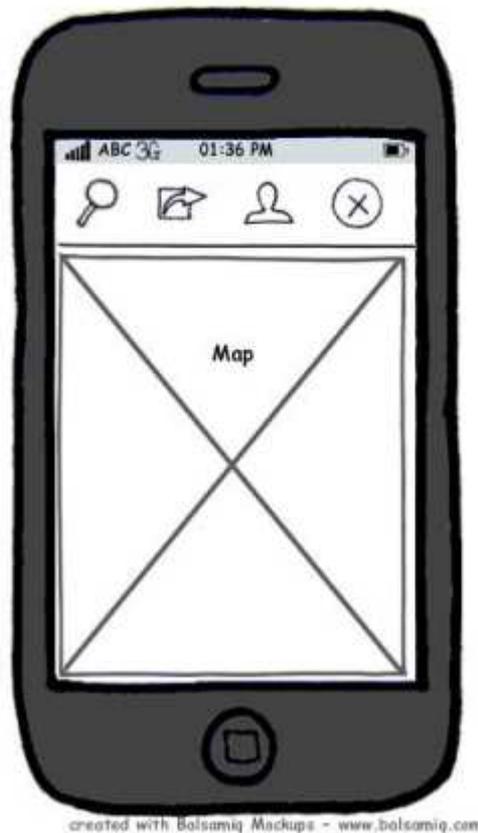


Figure III.25 : Partie Géolocalisation / Itinéraire.

Pour l'aspect ergonomique de la page Maps (voir figure III.25), Je me suis inspirés de l'application mobile Google Maps car je pense qu'elle était claire, ludique, pratique et intuitive. J'ai essayé de garder et de mettre en œuvre ces caractéristiques.

Cette page est composée de 2 parties :

- ✓ **Partie haute** : Une barre d'onglet contenant 4 boutons : *Afficher les catégories de lieu*
– *Définir un itinéraire* – *Se géolocaliser* – *Réinitialiser la carte*.
- ✓ **Partie centrale** : La carte Google Maps permettant de géo localiser l'utilisateur, d'afficher les différents lieux et de le guider.

III.4.3.3. Partie Géolocalisation / Itinéraire : La pop-up d'itinéraire et de navigation :

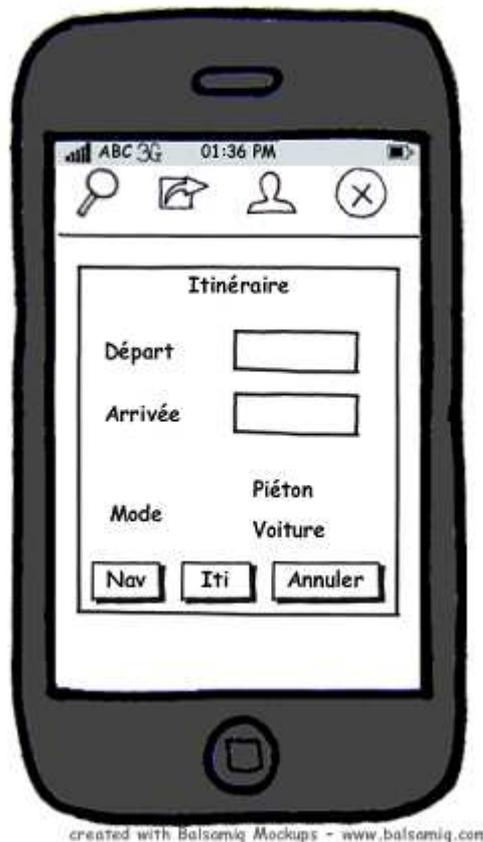


Figure III.26 : Partie itinéraire / navigation.

Cette pop-up (voir figure III.26) permet de renseigner les données d'un itinéraire :

- ✓ Départ : Position de l'utilisateur
- ✓ Arrivée : Destination de l'itinéraire
- ✓ Mode : Mode déplacement (Piéton ou Voiture)

2 boutons permettent de choisir le type d'itinéraire :

- ✓ Navigation : Itinéraire via le GPS
- ✓ Itinéraire : Itinéraire statique (trait bleu sur la map)

III.4.3.4. Partie Géolocalisation / Itinéraire : L'affichage des informations sur la map

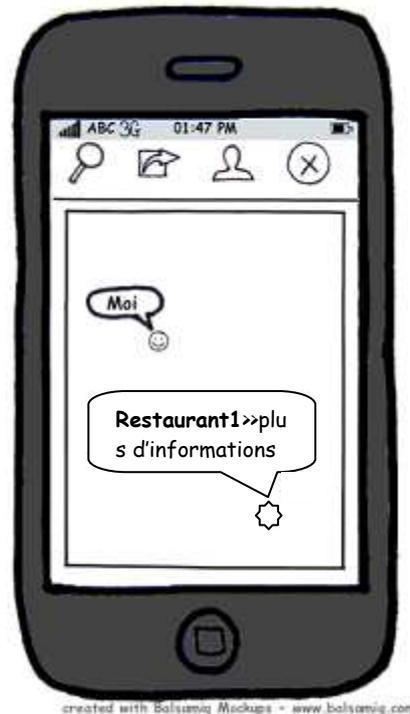


Figure III.27 : Partie affichage des informations sur la map.

Sur cette page (voir figure III.27), on retrouve la position de l'utilisateur, le ou les bâtiments que l'utilisateur souhaite afficher et l'itinéraire entre l'utilisateur et un bâtiment de l'université. Pour représenter l'utilisateur j'ai utilisé l'icône suivante :



Les bâtiments sont représentés par une icône symbolisant la catégorie de ce bâtiment. Chaque icône doit être le plus explicite et reconnaissable possible.

Je souhaitais qu'à partir de l'icône du bâtiment avoir un accès rapide à la fiche détaillée du bâtiment.

Afin de représenter l'itinéraire piéton ou en voiture de l'utilisateur vers un bâtiment, j'ai souhaité mettre à disposition de l'utilisateur 2 outils :

- ✓ La navigation GPS.
- ✓ L'itinéraire statique représenté par un trait bleu.

III.4.3.5. Partie Recherche de bâtiment : les listes de catégories et de bâtiments :

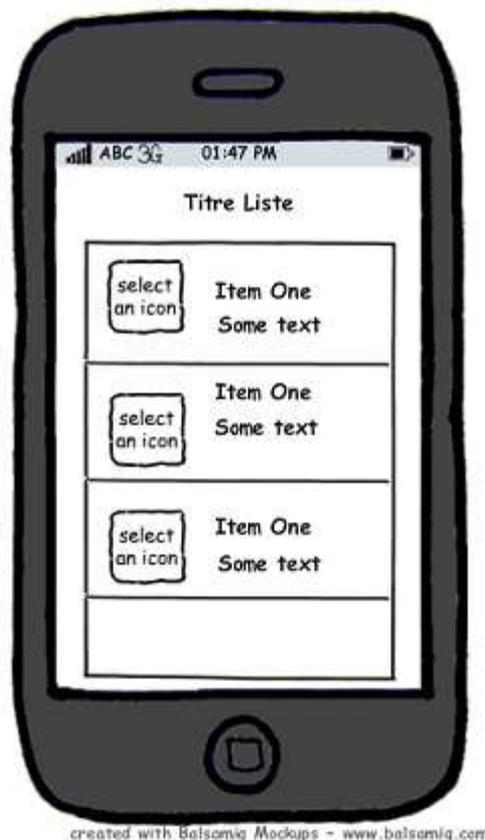


Figure III.28 : Partie recherche de bâtiment.

Afin d'avoir une navigation simple et agréable, j'ai décidé de présenter les catégories et les bâtiments sous forme de liste (voir figure III.28). Je me suis inspiré du tutoriel « Personnaliser une ListView » proposé sur le site <http://www.tutomobile.fr>. La découverte de ce tutoriel a été très bénéfique dans l'avancement de l'aspect graphique de mon application. Pour ne pas que l'utilisateur se perde dans la recherche de bâtiment, le titre de la catégorie est toujours présent sur chaque liste de bâtiments. De plus j'ai décidé de mettre des icônes assez explicites pour chaque catégorie (un cuisinier pour la catégorie Restaurants Universitaires par exemple).

III.4.3.6. Partie Recherche de bâtiment : la fiche détaillée d'un bâtiment :

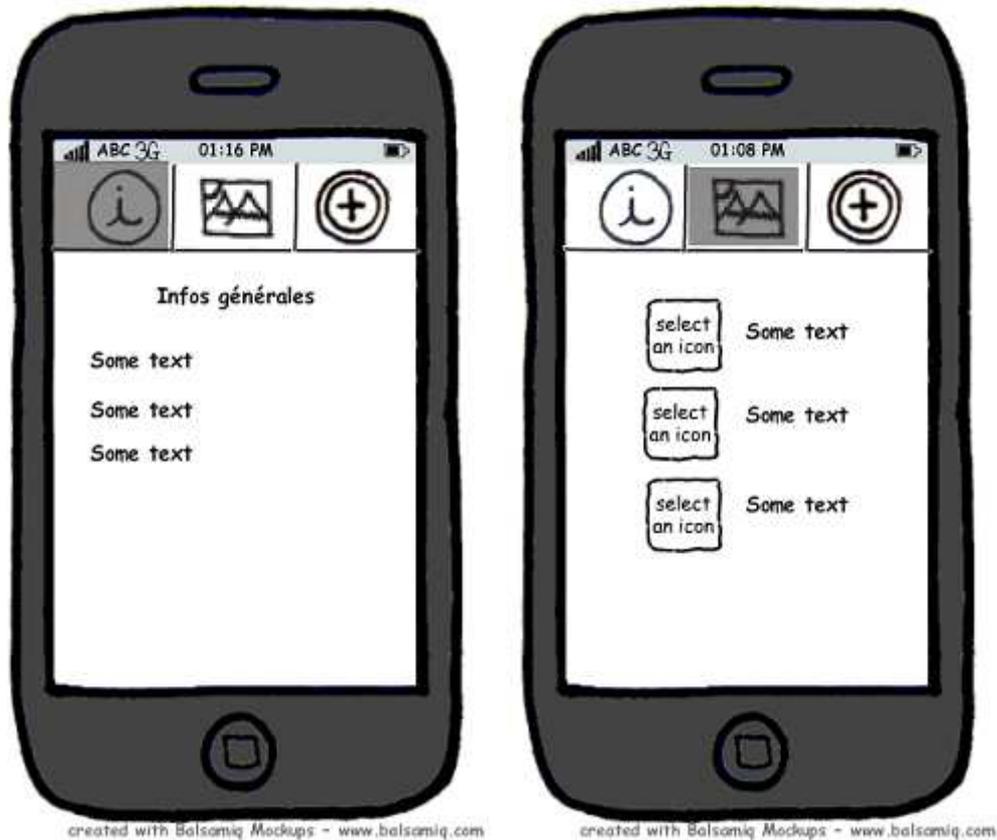


Figure III.29 : Partie fiche détaillée du bâtiment recherché.

La page représentant la fiche détaillée d'un bâtiment est découpée en trois parties (voir figure III.29) à savoir :

- ✓ La partie informations générales du bâtiment contenant l'adresse, le numéro de téléphone, ...
- ✓ Le partie navigation et itinéraire
- ✓ La partie autre dans laquelle on peut trouver la possibilité d'appeler le bâtiment par exemple j'ai décidé d'utilisé un système d'onglet pour permettre de mieux structurer les parties. Ainsi la lisibilité est plus claire. De plus, le fait d'utiliser des icônes dans les onglets permet un gain de place assez conséquent compte tenu du nombre d'information à afficher dans la page. Ainsi, j'ai réussi à faire en sorte que l'utilisateur n'ait pas à scroller dans la fiche du bâtiment.

III.5. Conclusion :

La phase de conception sert à identifier les différents objets qui nous aident à certifier les fonctionnalités souhaitées. Tout en donnant une description complète de ces diagrammes, ce qui permet une meilleure compréhension du système.

Après ce chapitre, j'ai passé au dernier chapitre dont j'ai décrit le travail réalisé ainsi expliqué l'environnement logiciel.

Chapitre IV :

Réalisation et mise en oeuvre



ANDROID

IV.1. Introduction :

La tâche de réalisation ou d'implémentation est la phase finale de l'élaboration d'un système. Elle permet aux matériels et logiciels d'entrer en fonction pour passer de l'expression d'un besoin informatique à un système fonctionnel fiable. Cette phase consiste à traduire la conception exprimée à l'aide d'un formalisme en un code source écrit dans un langage donné. Pour ce faire je vais montrer des aperçus d'écran qui décrivent les diverses fonctionnalités mise en place dans l'application.

IV.2. Environnement de travail :

IV.2.1. Environnement matériel :

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

- PC Portable : Intel Core 2 Duo, 2.20 GHZ, 2 GO de mémoire vive, Windows XP.

IV.2.2. Environnement cible :

Mon application est destinée pour tous les Smartphones équipé d'un système android 2.1 dite éclair, et toutefois, mon application, est utilisable sur la version 2.2 et sur les versions antérieures d'Android.

IV.2.3. Environnement logiciel :

Avant de pouvoir faire du développement, il faut nécessairement préparer l'environnement de travail. Dans mon cas, j'ai choisi de faire mon développement mobile sur la plateforme Eclipse. Pour la préparer, j'ai effectué différentes tâches que je décrirai dans ce qui suit.

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

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

IV.2.3.1. Système d'exploitation :

[Mon application a été développée sous le système d'exploitation Windows XP service pack2, en langage JAVA]. Elle peut être ainsi intégrée dans n'importe quel autre système d'exploitation supportant la machine virtuelle java.

IV.2.3.2. Langage de programmation :

Le langage java est un langage de programmation orienté objet mis au point par Sun Microsystems. Sa caractéristique principale est qu'il est indépendant de toute plate forme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une « Java Virtual Machine » (JVM). Ce concept est à la base du slogan de Sun pour java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java Development Kit).

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

IV.2.3.3. Eclipse :

Eclipse est un IDE qui permet de programmer dans différents langages grâce à ses nombreux plug-ins et notamment le plug-in d'Android. Une interface spécifique permet de gérer des fichiers java et de compiler ses programmes. Les fichiers sont organisés selon une arborescence qui correspond aux paquetages java définis. L'analyse syntaxique permet de mettre en valeur les mots clés dans les fichiers java. Eclipse dispose aussi d'un système d'auto complétion des fonctions, de détection des erreurs syntaxiques en temps réel sans oublier un système de débogage permettant d'exécuter ses programmes pas à pas.

IV.2.3.4. IDE Eclipse :

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

Pour le développement de mon application, j'ai choisi Eclipse INDIGO (Son interface principale est donnée dans la figure IV.1).

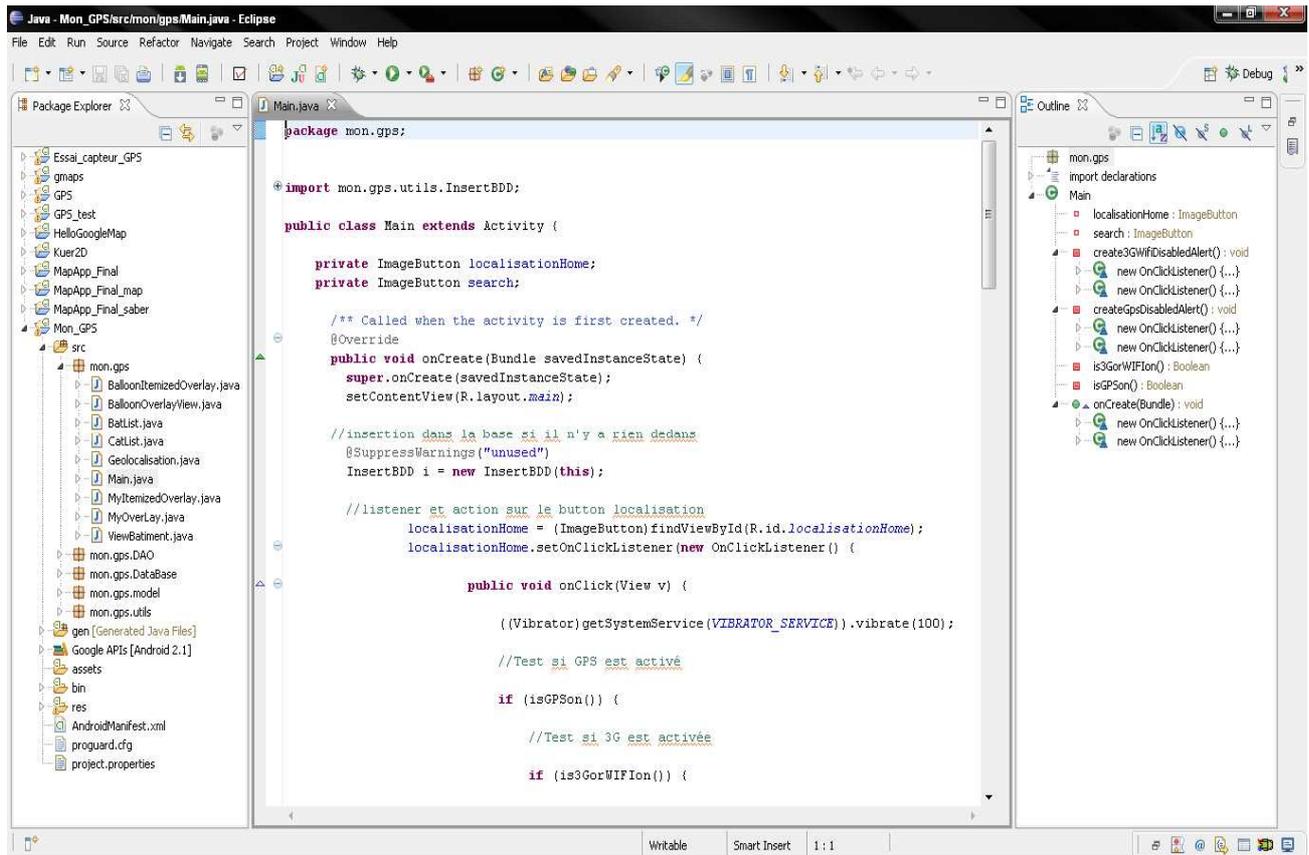


Figure IV.1 : plate forme d'éclipse.

IV.2.3.5. Le plugin ADT :

Pour développer Android, j'ai installé le plugin Android qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android.

IV.2.3.6. Software Development Kit (SDK) :

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

IV.2.3.7. Android :

Android est un système d'exploitation open source pour smart phones, PDA et terminaux mobiles. Il s'agit également d'un langage de programmation basé sur le JAVA et sur le XML. Le JAVA permet d'interagir avec l'utilisateur en faisant la liaison entre

l'interface graphique et la base de données. Quant au XML, il permet notamment de décrire les interfaces graphiques.

Mon Application a été développée pour la version 2.1 d'Android baptisé Eclair. Toutefois, mon application, est utilisable sur la version 2.2 et sur les versions antérieures d'Android.

IV.2.3.8. Autres outils :

IV.2.3.8.1. Balsamiq :

Balsamiq est un outil qui permet de faire du mock-up, c'est à dire de maquetter un site ou une application mobile, assez simplement. J'ai utilisé la version gratuite online car elle présentait les principaux éléments qui nous intéressaient pour maquetter notre application comme par exemple des boutons, des onglets, ...

Balsamiq est un outil assez puissant car il permet en quelques clics d'avoir une représentation très concrète de l'application finale. De plus le rendu final de la maquette au design crayonné est assez beau.

IV.2.3.8.2. SQLite :

SQLite est une bibliothèque écrite en C qui propose une base de données accessible par le langage SQL. Il est intégré de base à l'OS Android.

IV.3. Fonctionnement de mon application :

Je vais vous présenter les écrans de mon application Android ainsi que leurs fonctionnalités sous forme de zones de texte présentant les actions réalisées via cet élément.

IV.3.1. L'écran d'accueil :



Figure IV.2 : Ecran d'accueil.

1 : Ouvre les fonctionnalités de géolocalisation.

2 : Ouvre la fonctionnalité de recherche de lieu.

IV.3.2. L'écran de géolocalisation :

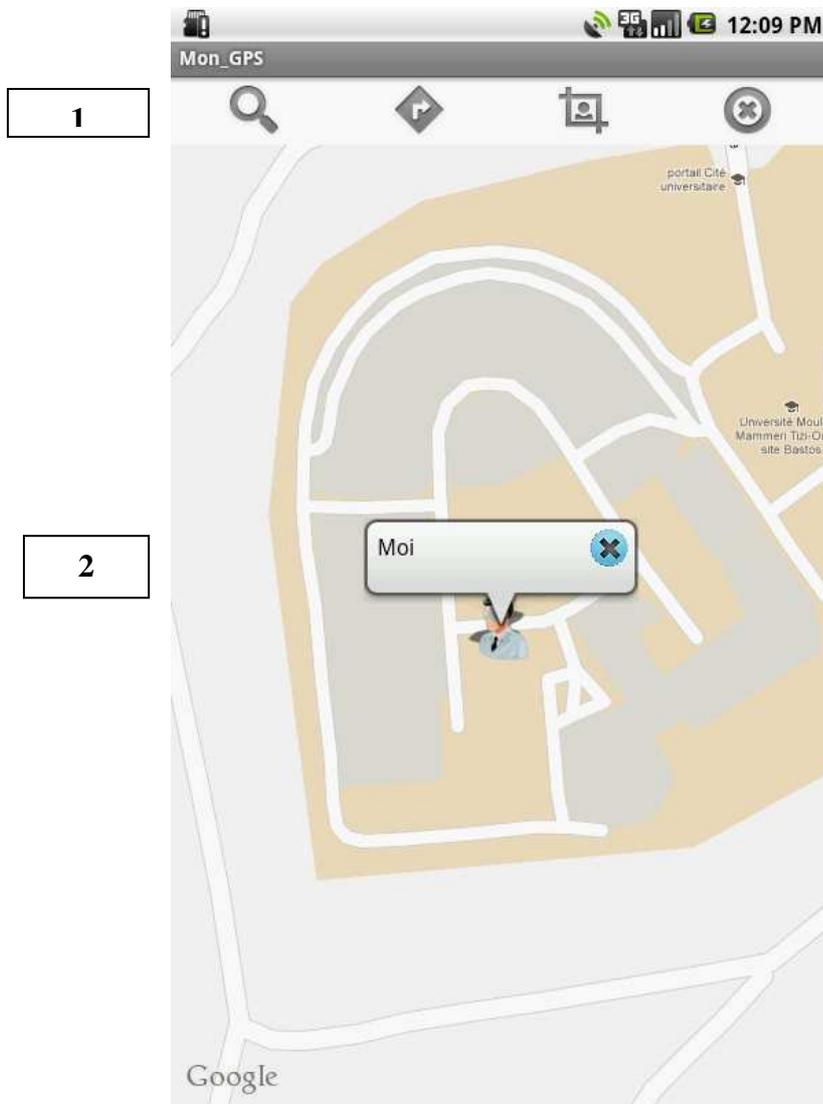


Figure IV.3 : Ecran de géolocalisation.

Lors d'un clic sur le bouton Maps de la page d'accueil, l'utilisateur est géolocaliser sur la carte.

- ✓ 1 : Barre d'onglets permettant de rechercher un lieu, un itinéraire, de se géolocaliser et de réinitialiser la carte
- ✓ 2 : Position de l'utilisateur.

IV.3.3. L'écran de Pop-up itinéraire :

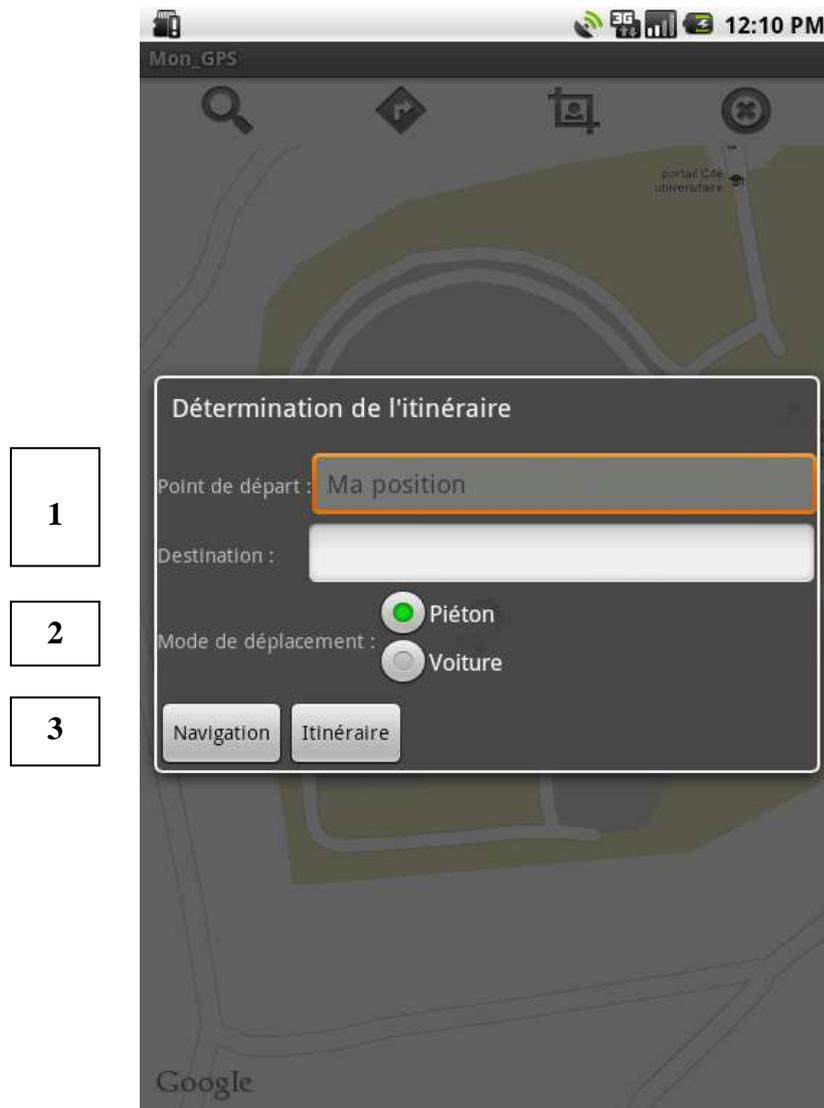


Figure IV.4 : Ecran de pop-up itinéraire.

Cette pop-up s'ouvre lors d'un click sur le deuxième élément de la barre d'onglets.

- ✓ 1 : Destination de l'itinéraire : Auto complétion à partir d'une lettre saisie par l'utilisateur
- ✓ 2 : Mode de déplacement Piéton ou Voiture
- ✓ 3 : Choix du mode de l'itinéraire : Via le GPS (Navigation) ou statique (Itinéraire).

IV.3.4. L'écran de la liste des catégories :

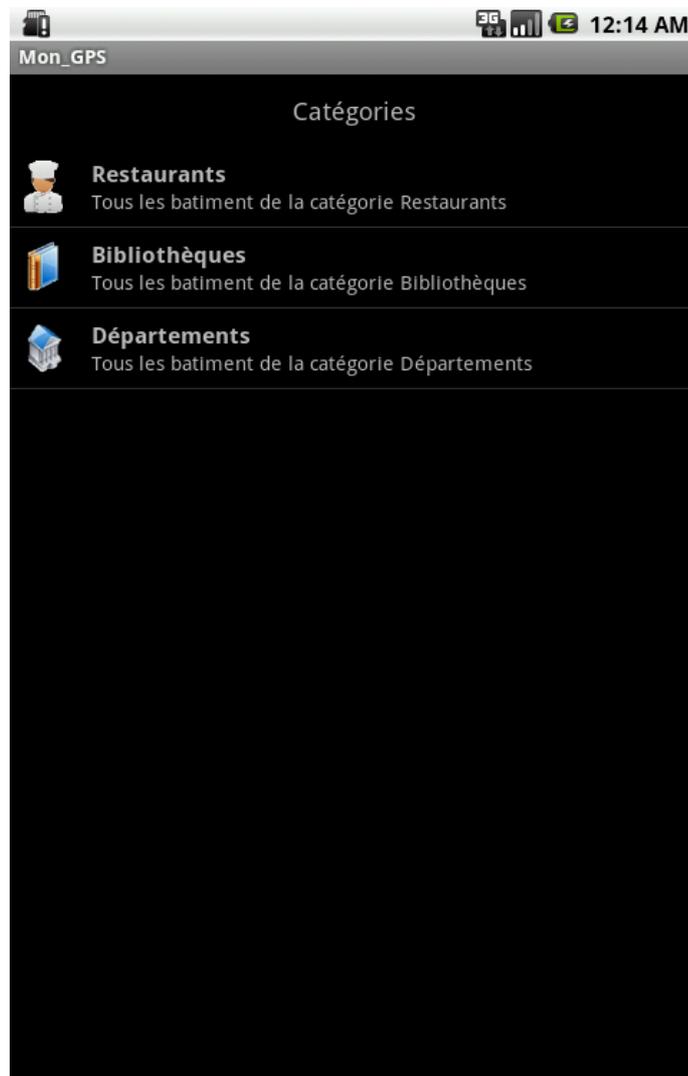


Figure IV.5 : La liste des catégories.

Lors d'un clic sur le bouton Rechercher un lieu de la page d'accueil, la liste des catégories de lieux est proposé à l'utilisateur.

- ✓ 1 : Ouvre la liste des bâtiments contenus dans la catégorie choisie.

IV.3.5. L'écran de La liste des bâtiments correspondants à une catégorie :

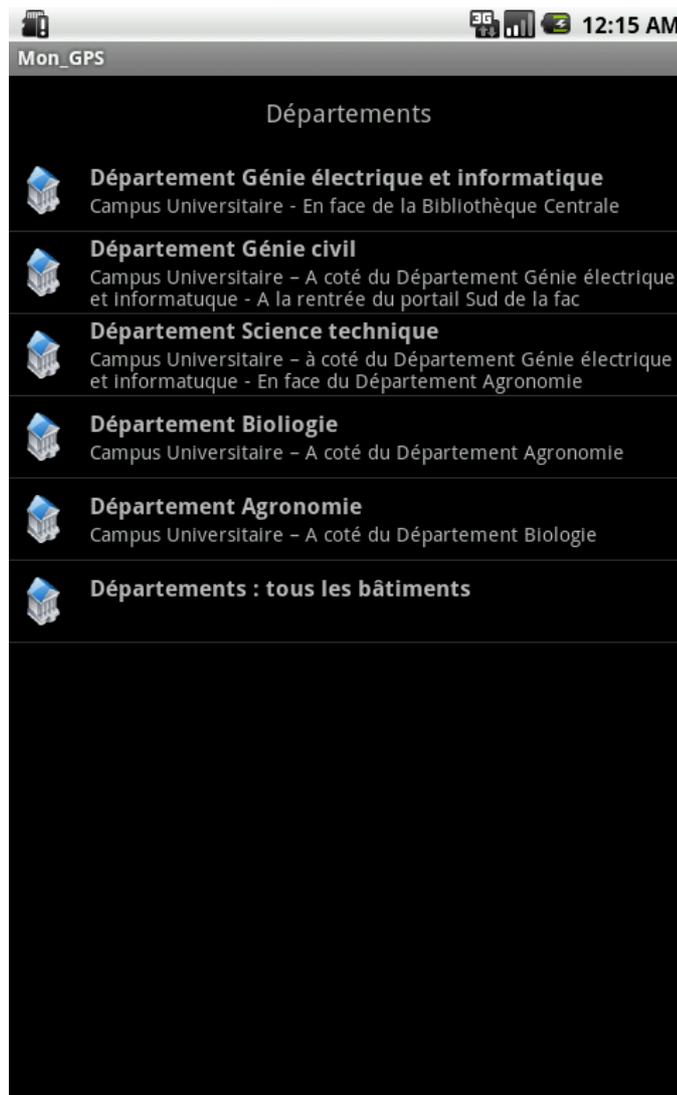


Figure IV.6 : La liste des bâtiments correspondants à une catégorie.

Lorsque l'utilisateur choisit une catégorie de lieux, les bâtiments de cette catégorie s'affichent.

- ✓ 1 : Ouvre la fiche détaillée du bâtiment sélectionné
- ✓ 2 : Affiche sur la map tous les bâtiments de la catégorie choisie

IV.3.6. L'écran de Tous les départements universitaires de Bastos :

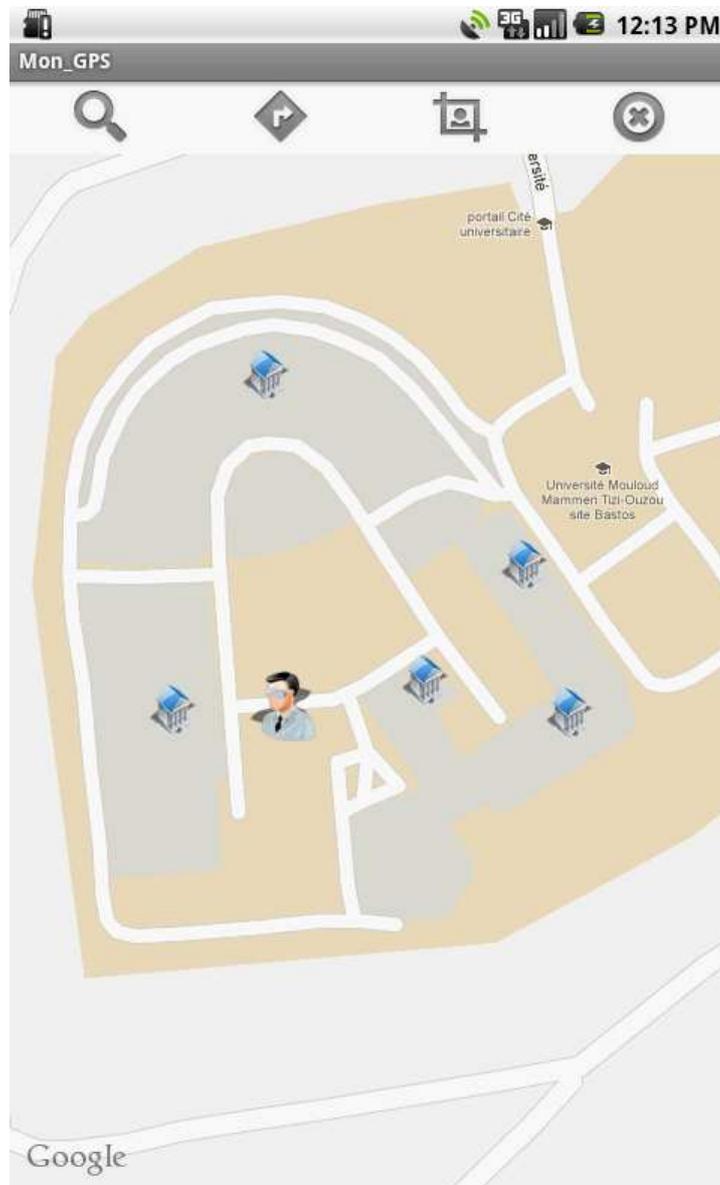


Figure IV.7 : Tous les départements universitaires de Bastos.

Si l'utilisateur a cliqué sur l'écran précédent sur tous les bâtiments, la carte contient tous les bâtiments de cette catégorie. Ces bâtiments sont représentés par l'icône de la catégorie.

IV.3.7.L'écran de la partie informations générales de la fiche détaillée d'un bâtiment :



Figure IV.8 : La partie informations générales de la fiche détaillée d'un bâtiment.

Si l'utilisateur a cliqué sur l'écran précédent sur un bâtiment, il accède à sa fiche détaillée.

- ✓ 1 : Ouvrir la partie géolocalisation de la fiche détaillée
- ✓ 2 : Ouvrir la partie Complément d'information de la fiche détaillée

IV.3.8. L'écran de la partie Localisation/Maps de la fiche détaillée d'un bâtiment :

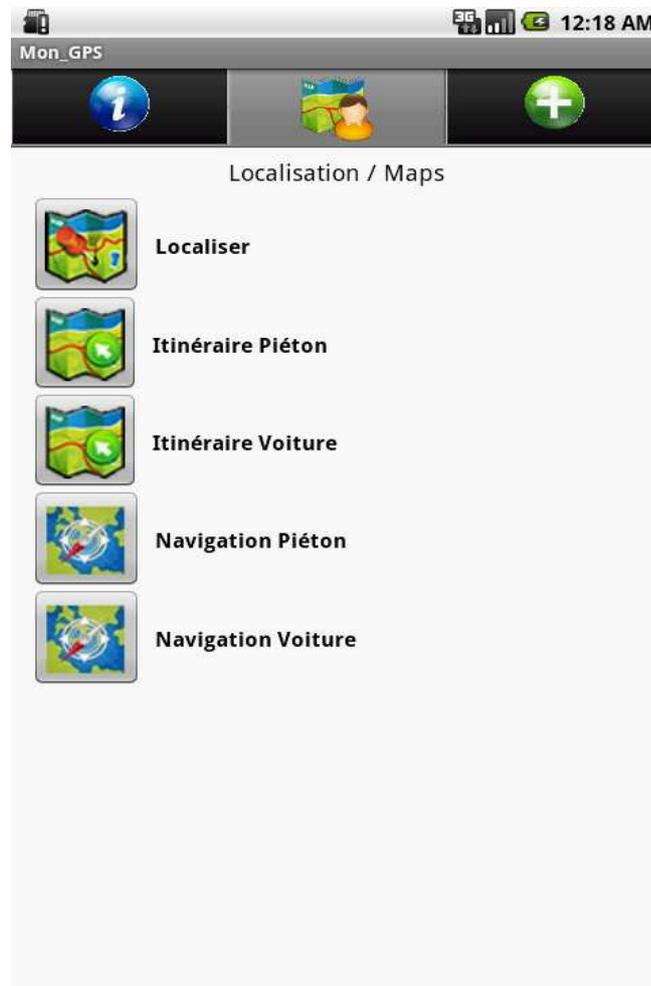


Figure IV.9 : La partie Localisation/Maps de la fiche détaillée d'un bâtiment.

Ouverture de l'onglet Localisation/Maps de la fiche détaillée.

- ✓ 1 : Affiche le bâtiment sur la map
- ✓ 2 : Affiche l'itinéraire à pied vers le bâtiment sur la map
- ✓ 3 : Affiche l'itinéraire en voiture vers le bâtiment sur la map
- ✓ 4 : Ouvre le service de Google Navigation en mode piéton vers le bâtiment
- ✓ 5 : Ouvre le service de Google Navigation en mode voiture vers le bâtiment

IV.3.9. L'écran de la partie Complément d'information de la fiche détaillée d'un bâtiment :



Figure IV.10 : La partie Complément d'information de la fiche détaillée d'un bâtiment.

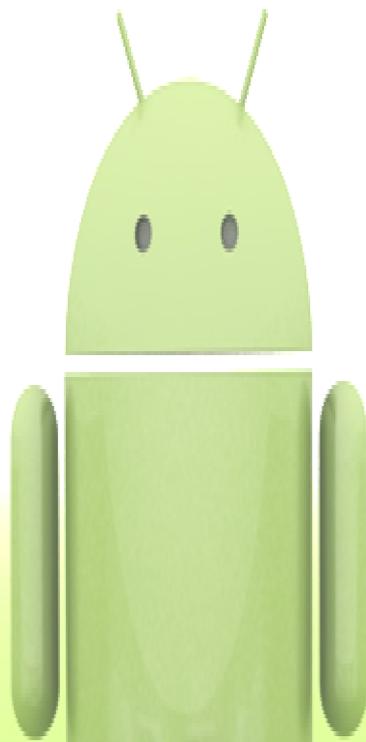
Ouverture de l'onglet Complément d'information de la fiche détaillée.

- ✓ 1 : Appel le bâtiment
- ✓ 2 : Affiche le menu (ou le site internet) du bâtiment dans un navigateur web

IV.4. Conclusion :

J'ai présenté dans ce chapitre une vue globale sur le système réalisé. La partie mise en œuvre traduit les besoins fonctionnels et techniques déjà définis par l'implémentation de différentes interfaces.

Conclusion Générale



ANDROID

Conclusion Générale :

L'élaboration de mon travail était dans le but de concevoir une application dédiée aux terminaux mobiles disposant de la plateforme Android. Cette application permet au propriétaire du téléphone la navigation simple et rapide au sein du campus de bastos. Pour ce faire, j'ai recouru à différentes technologies et outils de localisation et d'orientation jugés nécessaires pour aboutir à l'objectif de mon application.

Mon application ainsi réalisée permet de :

- Se localiser
- Afficher les bâtiments d'une catégorie donnée
- Recherche d'un bâtiment sur le campus
- Itinéraire piéton ou en voiture vers un bâtiment du campus
- Navigation piéton ou en voiture via le GPS vers un bâtiment du campus
- Consulter la fiche détaillée de chaque bâtiment (Adresse, horaires, description)
- Appeler le bâtiment / Consulter son site internet.
- Consulter le menu du RU

Ce projet m'avait donné de plus l'occasion d'acquérir des nouvelles connaissances à propos d'Uml, Eclipse, SQLiteBrowser et de maîtriser les langages de programmation java qui seront certes utiles dans ma future vies professionnelle.

Enfin, je ne peu pas sans doute affirmer que mon travail est complet d'où plusieurs améliorations peuvent être apportées. Mais, nous espérons au moins que j'ai réussi à réaliser une simple application mobile Android fonctionnelle qui satisfait aux besoins des futurs utilisateurs et qui convient à leurs attentes.

Les perspectives :

En guise de perspectives j'envisage d'enrichir mon application avec de nouvelles fonctionnalités comme :

1. Elargir les cibles potentielles :

Tout d'abord, il serait intéressant d'ajouter un module de reconnaissance vocale pour permettre à des personnes déficientes visuelles d'utiliser l'application. De plus, proposer à l'utilisateur le choix de la langue peut être très utile.

2. Ajout de nouvelles fonctionnalités :

Il est envisageable d'enrichir l'application par des services utiles à chaque étudiants. On peut imaginer intégrer une boite mail pour les mails de la Fac, un agenda de ses cours.

Une version widget de l'application, permettrait un accès plus rapide à une des fonctionnalités de l'application car il est accessible directement via le bureau de

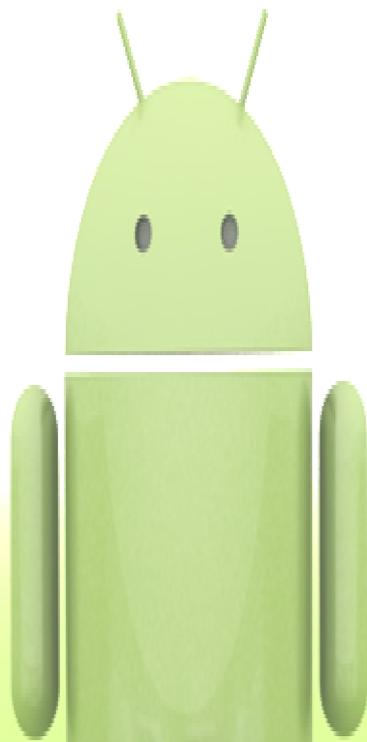
Conclusion Générale

son smart phone. Par exemple ce widget pourrait servir à récupérer le numéro de téléphone d'un bâtiment. Il s'agit en fait de déterminer quelle est la fonctionnalité la plus utilisée de l'application et d'en faire un widget.

3. Mode Offline :

Un deuxième mode sera envisageable, afin que l'utilisateur puisse utiliser l'application sans se soucier s'il est connecté sur internet ou pas.

Table des illustrations:



android

Table des illustrations

Les figures

Chapitre I : Etat de l'art d'android.

Figure I.1 : évolution des versions d'Android.....	3
Figure I.2 : Architecture de l'Android	6
Figure I.3 : Architecture de ces médiathèques.	9
Figure I.4 : Conversation d'un fichier .java à un fichier .dex.	10
Figure I.5 : Cycle de vie d'une activité.	12
Figure I.6 : Une application sur android.	14
Figure I.7 : Portail des développeurs Android.	15
Figure I.8 : Interface d'installation du SDK Android.	16
Figure I.9 : Interface d'installation ADT.	17
Figure I.10 : Interface du simulateur Android.....	18

Chapitre II : Géolocalisation.

Figure II.1 : géolocalisation par GSM	23
Figure II.2 : géolocalisation mixte	24
Figure II.3 : géolocalisation Indoor.....	25
Figure II.4 : Segment spatial.	27
Figure II.5 : Le Segment de contrôle.....	28
Figure II.6 : Segment utilisateur.....	28
Figure II.7 : Fonctionnement du GPS.	29
Figure II.8 : positionnement de l'observateur a l'intersection des 3 satellites.....	30
Figure II.9 : positionnement de l'observateur dans la zone commune des 3 satellites.	30
Figure II.10 : Bonne répartition.....	31
Figure II.11 : Mauvaise répartition.	32
Figure II.12 : Dans une vallée, Mauvaise réception.....	32
Figure II.13 : Dans une vallée, Meilleure réception.....	33
Figure II.14 : Système de coordonnées locales.....	34
Figure II.15 : Directe tangente.	35

Table des illustrations

Figure II.17 : représentation conique.	35
Figure II.18 : Directe.	36
Figure II.19 : Transverse.....	36
Figure II.20 : représentation cylindrique	36
Figure II.21 : La projection UTM	37
Figure II.22 : A-GPS.	38
Figure II.23 : TOA.	39
Figure II.24 : Cell-ID.....	40

Chapitre III : Analyse et conception.

Figure III.1 : La démarche adoptée pour la modélisation.	43
Figure III.2 : Cas d'utilisation « Accéder a la partie Informations Générales ».	47
Figure III.3 : Cas d'utilisation « Téléphoner au bâtiment ».	47
Figure III.4 : Cas d'utilisation « Consulter le site internet du bâtiment ».	48
Figure III.5 : Cas d'utilisation « Localiser le bâtiment ».	49
Figure III.6 : Cas d'utilisation « Consulter l'itinéraire / la navigation ».	50
Figure III.7 : Cas d'utilisation « Se localiser ».	50
Figure III.8 : Cas d'utilisation « Réinitialiser la map ».	50
Figure III.9 : Diagramme de cas d'utilisation générale.	52
Figure III.10 : Diagramme de séquence de cas d'utilisation « se localiser ».	53
Figure III.11 : Diagramme de séquence de cas d'utilisation « Accéder au informations générales ».	54
Figure III.12 : Diagramme de séquence de cas d'utilisation « Consulter l'itinéraire / la navigation ».	55
Figure III.13 : Diagramme de séquence de cas d'utilisation « Localiser un bâtiment ».	56
Figure III.14 : Diagramme d'activité de cas d'utilisation « Se localiser ».	57
Figure III.15 : Diagramme d'activité de cas d'utilisation « Accéder au informations générales ».	58
Figure III.16 : Diagramme d'activité de cas d'utilisation « Consulter l'itinéraire / la navigation ».	59
Figure III.17 : Diagramme d'activité de cas d'utilisation « Localiser un bâtiment ».	60
Figure III.18 : Diagramme de classe de cas d'utilisation « Se localiser ».	61
Figure III.19 : Diagramme de classe de cas d'utilisation « Accéder au informations générales ».	62

Table des illustrations

Figure III.20 : Diagramme de classe de cas d'utilisation « Localiser un bâtiment ».....	63
Figure III.21 : Le schéma de la base de données.....	64
Figure III.24 : page d'accueil.....	66
Figure III.25 : Partie Géolocalisation / Itinéraire.....	67
Figure III.26 : Partie itinéraire / navigation.....	68
Figure III.27 : Partie affichage des informations sur la map.....	69
Figure III.28 : Partie recherche de bâtiment.....	70
Figure III.29 : Partie fiche détaillée du bâtiment recherché.....	71

Chapitre IV : Réalisation et mise en œuvre.

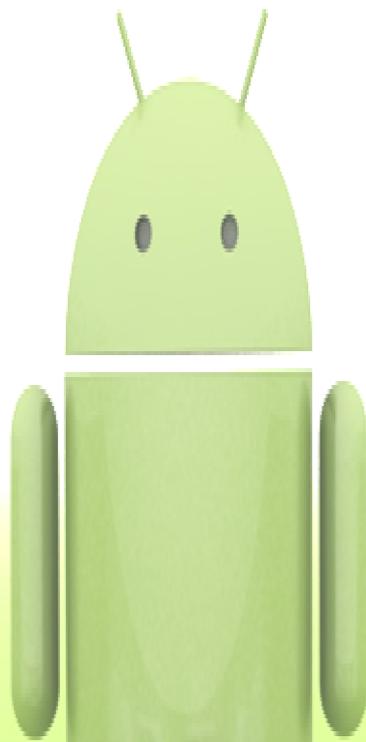
Figure IV.1 : plate forme d'éclipse.....	76
Figure IV.2 : Ecran d'accueil.....	78
Figure IV.3 : Ecran de géolocalisation.....	79
Figure IV.4 : Ecran de pop-up itinéraire.....	80
Figure IV.5 : La liste des catégories.....	81
Figure IV.6 : La liste des bâtiments correspondants à une catégorie.....	82
Figure IV.7 : Tous les départements universitaires de Bastos.....	83
Figure IV.8 : La partie informations générales de la fiche détaillée d'un bâtiment.....	84
Figure IV.9 : La partie Localisation/Maps de la fiche détaillée d'un bâtiment.....	85
Figure IV.10 : La partie Complément d'information de la fiche détaillée d'un bâtiment.....	86

Les Tableaux

Chapitre III : Analyse et conception.

Tableau III.1 : Spécification des tâches.....	45
Tableau III.2 : Spécification des scénarios.....	47
Tableau III.22 : Table bâtiment.....	66
Tableau III.23 : Table catégorie.....	66

Annexe A:



android

Annexe A : Manuel d'installation et de mise en marche.

Installation de l'Eclipse [N1].

2. Installation du SDK Android [N2].

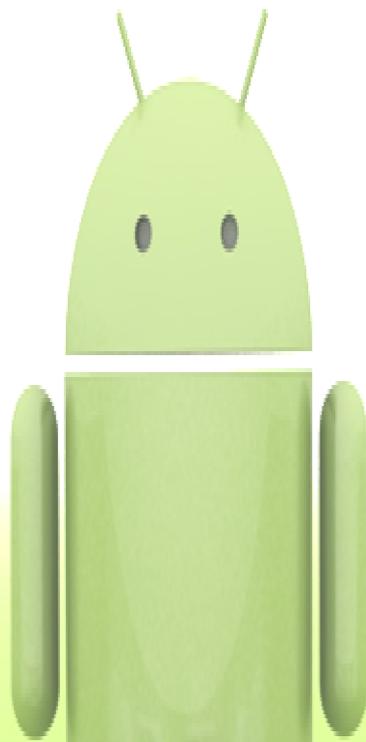
3. Installation du plugin ADT sous Eclipse [N3].

4. Création d'un Android Virtual Device(AVD) :

La Configuration d'un nouveau AVD se fait en remplissant les champs suivants lors de sa création:

- ✓ Name : Le nom à donner à votre émulateur (sans espace).
- ✓ Target : La version du SDK Android sur lequel l'émulateur doit fonctionner (Dans notre cas elle doit être de type Google APIs pour pouvoir faire fonctionner le programme).
- ✓ SD Card: Configuration de la SD Card (Taille, etc.). Ce champ est facultatif.
- ✓ Skins : Choix du thème de l'émulateur. Des émulateurs préconfigurés se trouvent dans Built-in.
- ✓ Hardware: Cette partie permet de rajouter le matériel et de le personnaliser.

Annexe B:



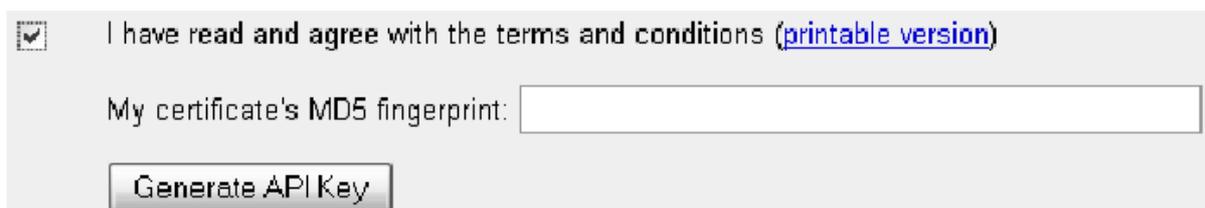
ANDROID

Annexe B : Obtention d'une clé pour utiliser Google Maps.

J'en ai besoin de cette configuration pour toute la partie concernant la manipulation de carte via l'API Google Maps. Une étape supplémentaire sera alors nécessaire : l'obtention d'une clé de licence.

Pour cela, la première étape consiste à se rendre sur le site proposé par Google. Il me faudra également un compte Google pour compléter l'opération. Enfin, pour obtenir cette clé, l'interface du site nous demande une empreinte MD5 d'un certificat :

<http://code.google.com/android/maps-api-signup.html>.

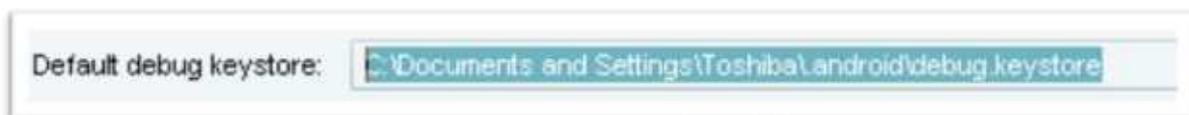


The screenshot shows a web form for generating an API key. It includes a checked checkbox for 'I have read and agree with the terms and conditions (printable version)', a text input field for 'My certificate's MD5 fingerprint:', and a 'Generate API Key' button.

Android n'autorise l'installation que des applications signées. Avant d'installer une application via l'émulateur, Eclipse signe notre application en utilisant un certificat de débogage qui est fourni avec le SDK d'Android.

Eclipse crée le fichier **debug.keystore** lors de la compilation du projet, ce fichier étant stocké dans le répertoire **.android** dont le chemin varie en fonction du système d'exploitation. Pour un système sous Windows XP par exemple le chemin est :

- ✓ C:\Documents and Settings\Nom d'Utilisateur\.android.



La clé générée pour utiliser l'API Google Maps est basée sur ce certificat de débogage. Pour en obtenir l'empreinte (et en se basant sur un système XP logué en administrateur), il faut utiliser la commande suivante :

- ✓ `keytool -list -keystore "C:\Documents and Settings\Administrateur\.android\debug.keystore"`.

Annexe B

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrateur>keytool -list -keystore "C:\Documents and Settings\Administrateur\.android\debug.keystore"
Tapez le mot de passe du Keystore :

***** A U E R T I S S E M E N T *****
* L'intégrité des informations enregistrées dans votre Keystore *
* n'a PAS été vérifiée ! Pour cela, *
* vous devez spécifier le mot de passe de votre Keystore. *
***** A U E R T I S S E M E N T *****

Type Keystore : JKS
Fournisseur Keystore : SUN

Votre Keystore contient 1 entrée(s)
androiddebugkey, 3 mai 2009, PrivateKeyEntry,
Empreinte du certificat (MD5) : 2B:88:E5:1F:59:A7:5A:59:E7:65:0E:96:E3:8B:33:BC
C:\Documents and Settings\Administrateur>_
```

Une fois l'empreinte saisie dans le formulaire et envoyée au site, une clé est générée ainsi qu'un morceau de code XML montrant un exemple d'utilisations.

Voici votre clé :

```
OVpxagd1X0sCrqGUvTZI1DYU3SkQDJ6Vvr1mkGg
```

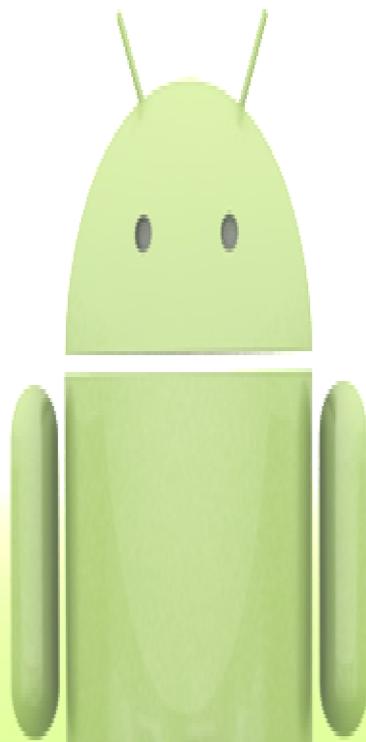
Cette clé fonctionne avec les applications signées avec votre certificat ; l'empreinte de celui-ci est :

```
C6:28:72:6F:9C:E3:1E:A1:45:F1:8E:76:66:47:34:A6
```

Voici un exemple de code xml qui vous aidera à commencer à exploiter efficacement nos outils cartographiques :

```
<com.google.android.maps.MapView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:apiKey="OVpxagd1X0sCrqGUvTZI1DYU3SkQDJ6Vvr1mkGg"
/>
```

Glossaire:



android

Glossaire :

ADT : Abstract Data Type , Plugin Android pour Eclipse .

API : Application Programming Interface : Interface de programmation , Bibliothèque.

AVD : Android Virtual Device , terminale Android Virtuel et configurable.

GPS : Global Positioning Système, Système de positionnement pas satellite.

OHA : Open Handset Alliance, Consortium d'entreprise soutenant le projet android.

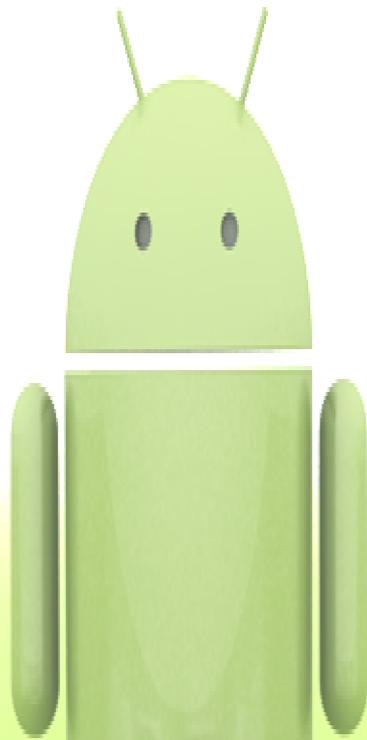
PDA : Personal Digital Assistant , Appareil numérique portable .

SDK : Software Development Kit , Kit de développement, Permettant la création des applications de type défini .

UML : Unified Modeling Language ,Langage de programmation graphique dans le monde de génie logiciel .

XML : Extensible Markup Language , Langage informatique de balisage générique.

Bibliographie :



android

Bibliographie :

[1] : Portail des développeurs Android [En ligne].

-<http://developer.android.com/>.

[2] : Référence du SDK Android[En ligne].

-http://developer.android.com/sdk/ndk/1.5_r1/index.htm.

[3] : <http://www.wirelessdevnet.com/channels/lbs/features/mobilepositioning.html>.

[4] : Alain Le Guennec, « génie logiciel et méthode formelle avec UML spécification et généralisation de testes »juin 2001.

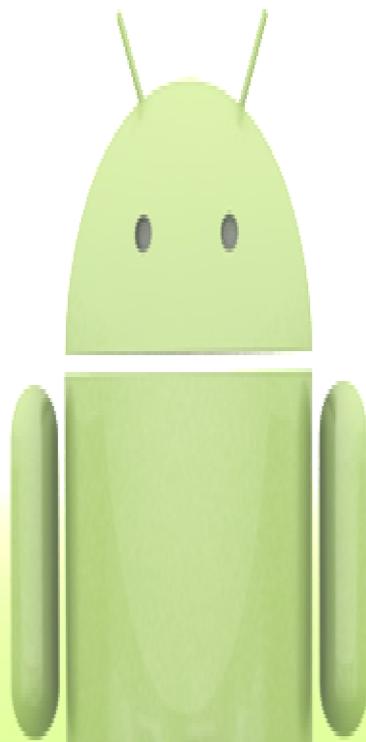
[5] : Grady Booch, James Rambough et Ivar Jacobso « Le guide de l'utilisateur UML », Edition Eyrolles, 2001.

[6] : Jim Conallen, « Concevoir des application web avec l'UML » Edition Eyrolles, Octobre 2000.

[7] : " Analyse et Conception Orientées Objet avec UML et réalisation en C++ " Guide étudiant, Edition : Sun 1999, pdf.

[8] : Grady Booch et al, " Le Guide de l'utilisateur UML ", Edition : Eyrolles, 2003.

Netographie:



android

Netographie :

[N1] : <http://www.eclipse.org/downloads/>.

[N2] : <http://developer.android.com/index.html>.

[N3] : <https://dl-ssl.google.com/android/eclipse/>.

[N4] : <http://www.pointgphone.com/android/emulateur>.

[N5] : <http://code.google.com/intl/fr-FR/android/maps-api-signup.html>.

[N6] : <http://www.mti.epita.fr/blogs/2010/08/03/introduction-a-la-programmation-sous-android/>.

[N7] : <http://developer.android.com/guide/topics/location/index.html>.

[N8] : <http://www.anddev.org/>.