

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



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

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de projet informatique**

Présenté par :

Ismail SLIMANI

Toufik HABIRECHE

Thème

Conception et réalisation d'un système de géolocalisation et de suivi Cas NAFTAL

Mémoire soutenu publiquement le 22 / 09 / 2016 devant le jury composé de :

Président : Mme djemah

Encadreur : Mme Seddoud

Co-Encadreur : Mr Abdelwahab AMMI

Examineur : Mr djemah

Examineur : Mme bousnina

REMERCIEMENTS

Remerciement

Nous remercions d'abord notre Dieu qui nous a aidé, donné le courage et la volonté pour réaliser ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à notre promotrice, Madame SEDOU pour son aide, ses précieux conseils, de nous avoir suivi et orienté tout au long de ce travail et pour la qualité de son encadrement.

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir accepté d'honorer par leur jugement notre travail.

Nos remerciements vont aussi aux personnels du Groupe Informatique de la Branche Carburants, en particulier à Monsieur Ammi Abdelwahab.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à

Ce que ce modeste travail voit le jour.

DEDICACE

Dédicaces

Je dédie ce modeste travail à :

- *Mes chers parents qui m'ont toujours soutenu*
- *Mes frères*
- *Mes sœurs*
- *Et à toute ma famille*
- *Mes amis(es)*

Toufik... 

Dédicaces

Je dédie ce modeste travail à :

- *Mes très chers parents*
- *A mon frère et à toute ma famille*
- *A tous mes amis*

Ismail... 

SOMMAIRE

Sommaire

❖ **Introduction générale**

❖ **Chapitre 1 : Etat de l'art**

Partie I : Android

Partie II : Géolocalisation

❖ **Chapitre 2 : Etude de l'existant**

❖ **Chapitre 3 : Analyse et conception**

❖ **Chapitre 4 : Réalisation**

❖ **Conclusion générale**

TABLE DES MATIERES

Table des matières

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

Chapitre 1 : Etat de l'art

Section I : Android

Introduction	3
I. Le système d'exploitation mobile.....	3
I.1. Définition	3
I.2. Caractéristique d'un système d'exploitation mobile.....	4
I.3. Les principaux systèmes d'exploitation mobile	4
I.4. Dispositifs mobile	5
II. Etude de la plateforme Android.....	6
II.1. Le système Android	6
II.2. Les caractéristiques du système Android.....	6
II.3. L'historique et les versions d'Android	7
II.4. Architecture Android	10
II.4.1. Application.....	10
II.4.2. Le Framework.....	11
II.4.3. Les bibliothèques (librairies)	12
II.4.4. Le moteur d'exécution Android.....	13
II.4.5. Le noyau linux (linux kernel)	14
III. Développement d'application Android	14
III.1. L'environnement de développement sous Android.....	14
III.1.1. Le JDK.....	14
III.1.2. Le SDK.....	15

III.1.3. L'ide Eclipse.....	16
III.1.4. Le plugin ADT pour Eclipse	16
III.1.5. L'émulateur de téléphone	16
III.2. Composent d'Android	17
III.2.1. Les activités	17
III.2.2. Les services	17
III.2.3. Les broadcast receivers.....	17
III.2.4. Les contents providers.....	17
III.2.5. Les Intents	18
III.3. Cycle de vie d'une application Android.....	18
IV. Les outils et technique de géolocalisation et d'orientation sous Android.....	19
IV.1. Google Maps API.....	19
IV.2. Le GPS	20
IV.3. Android radar	20
IV.4. Android Compass	20
Section II : Géolocalisation	
Introduction	21
I. La géolocalisation	21
I.1. Définition	21
II. Technique de géolocalisation	22
II.1. Géolocalisation par satellite.....	22
II.2. Géolocalisation par GSM.....	22
II.3. Géolocalisation par Wi-Fi.....	24
II.4. Géolocalisation par l'adresse IP.....	25
II.5. Géolocalisation par RFID	25
II.6. Combinaison de différente technique	27

III. Architecture temps réel	28
IV. Les plateformes de géolocalisation	28
IV.1. Définition	28
IV.2. Les composants de la plateforme de géolocalisation	28
IV.2.1. Le terminal communicant.....	28
IV.2.2. Le système informatique	30
IV.2.3. Le module cartographique.....	30
IV.3. Le fonctionnement de la plateforme de géolocalisation.....	30
Conclusion.....	31

Chapitre 2 : Etude de l'existant

Introduction	32
I. Présentation de la société NAFTAL	32
II. Historique	32
III. Organisation	33
IV. Présentation de la branche carburante.....	33
IV.1. Organisation de la branche carburante	34
IV.2. Missions principales de la branche carburante.....	34
IV.3. Produits commercialisés.....	35
IV.3.1. Par l'activité aviation	35
IV.3.2. Par l'activité marine	35
IV.3.3. Par l'activité CBRT.....	36
IV.4. Moyens humains et matériels	36
IV.4.1. Moyens humains	36
IV.4.2. Moyens matériels	36
IV.4.3. Infrastructures de la branche carburante	37

V. Circuit de distribution de NAFTAL	38
V.1. L'activité de distribution	38
V.1.1. Le stockage	39
V.1.2. Le transport.....	39
VI. Problématique	40
VII. Hypothèses.....	40
Conclusion	41

Chapitre 3 : Analyse et conception

Introduction	42
I. Analyse.....	42
I.1. Spécification des besoins	42
I.1.1. Spécification des besoins fonctionnels.....	42
I.1.2. Spécification des besoins non fonctionnels	42
I.2. Méthodologie et approche adoptée	43
I.2.1. Présentation de l'UML.....	43
I.2.2. Définition	43
I.2.3. Modélisation avec l'UML	44
I.2.4. Les avantages d'UML.....	44
I.3. Identification des acteurs.....	44
I.3.1. Définition d'un acteur	44
I.3.2. Les acteurs de notre système	45
I.3.3. Spécification des rôles.....	45
I.4. Les diagrammes de cas d'utilisation	45
I.4.1. Définition	45
I.4.2. Description des cas d'utilisations	46

I.4.3. Les diagrammes de cas d'utilisations pour chaque acteur.....	48
II. Conception	50
II.1. Le diagramme de déploiement.....	50
II.2. Les diagrammes de séquences	51
II.3. Les diagrammes d'activités.....	55
II.4. Les diagrammes de classe	57
II.5. Modale relationnel	59
Conclusion.....	60

Chapitre 4 : Réalisation

Introduction	61
I. Environnement de travail	61
I.1. L'architecture de l'application	61
I.2. Environnement matériel	62
I.3. Environnement logiciel	62
I.3.1. IDE Eclipse	63
I.3.2. Le plugin ADT	63
I.3.3. Le SDK	63
I.3.4. Xampp.....	63
I.3.5. Adobe Dreamweaver	64
II. Langages de programmation	65
II.1. Java	65
II.2. Html	65
II.3. PHP	65
III. Fonctionnement de notre application.....	66
III.1. Cote application mobile.....	66

III.2. Cote application web	66
III.2.1. La page de l'espace administrateur	66
III.2.2. La page de l'espace opérateur	68
III.2.3. Afficher la localisation des véhicules	69
Conclusion.....	70
Conclusion générale	71
Références bibliographie	72
Annexe	73

LISTE DES FIGURES

Liste des figures

Figure 1 : Des systèmes d'exploitation mobiles	4
Figure 2 : Tableau Caractéristique du système Android	7
Figure 3 : Evolution des versions d'Android.....	8
Figure 4 : Tableau synthèse des versions d'Android.....	9
Figure 5 : L'Architecture de la plateforme Android.....	10
Figure 6 : Les étapes d'exécution d'un programma Android	13
Figure 7 : Le SDK Android	15
Figure 8 : L'émulateur.....	16
Figure 9 : Le cycle de vie d'application Android	18
Figure 10 : Géolocalisation par satellite	22
Figure 11 : Positionnement par TAO.....	23
Figure 12 : Positionnement par Cell ID.....	24
Figure 13 : Géolocalisation par wifi	24
Figure 14 : Principe du Système RFID.....	26
Figure 15 : Géolocalisation GPS/GSM.....	27
Figure 16 : Les composants d'une plateforme de géolocalisation.....	28
Figure 17 : Organigramme de la société NAFTAL	33
Figure 18 : Organigramme de la Branche Carburant	34
Figure 19 : Tableau de moyen matériel	36
Figure 20 : Infrastructures de la branche carburante	37
Figure 21 : Circuit de distribution de NAFTAL.....	38
Figure 22 : Tableau des acteurs de notre système	45
Figure 23 : Cas d'utilisation « s'authentifier »	46
Figure 24 : Cas d'utilisation « Afficher la localisation des véhicules »	46
Figure 25 : Cas d'utilisation «Ajouter un véhicule».....	47
Figure 26 : Cas d'utilisation «Afficher la liste des appareils».....	47
Figure 27 : Cas d'utilisation « afficher les parcoures ».....	47
Figure 28 : Diagramme de cas d'utilisation « Acteur : Conducteur »	48
Figure 29 : Diagramme de cas d'utilisation « Acteur : Administrateur ».....	48
Figure 30 : Diagramme de cas d'utilisation : « Acteur : opérateur ».....	49

Figure 31 : Diagramme de déploiement	50
Figure 32 : Diagramme de séquence « Afficher la localisation des véhicules ».....	52
Figure 33 : Diagramme de séquence « Ajoute les appareils »	53
Figure 34 : Diagramme de séquence « Afficher l'historique des parcours »	54
Figure 35 : Diagramme d'activité « S'authentifier »	55
Figure 36 : Diagramme d'activité « Modification mot de passe »	56
Figure 37 : Diagramme d'activité « Afficher la localisation des véhicules».....	56
Figure 38 : Diagramme d'activité « Gérer les compte operateur »	57
Figure 39 : Diagramme de classes de la BDD locales.....	58
Figure 40 : Architecture de l'application.....	61
Figure 41 : L'interface principale de Dreamweaver	64
Figure 42 : Interface d'application mobile	66
Figure 43 : Interface de la page de l'espace administrateur	67
Figure 44 : Interface de la page d'accueil de l'espace administrateur	67
Figure 45 : Interface de la page de l'espace operateur	68
Figure 46 : Interface de la page localisation des véhicules	69

INTRODUCTION

GENERALE

Introduction générale

"Les mobiles sont aussi différents de l'internet que la télé l'a été de la radio", a annoncé Tomi Ahonen, meilleur auteur technologique et médiatique à succès, consultant de stratégie et orateur motivationnel.

Les téléphones portables sont devenus les premiers médias de masse dans le monde (on compte 4 milliards d'abonnés au téléphone mobile). Il n'est pas un PC plus bête, mais un téléphone portable est considéré comme un autre support à générer des formes médiatiques.

L'invention du premier PDA au monde, Le Pen Pad conçu par Apple, était dans le but de pouvoir prendre des notes, gérer son agenda, ses adresses, effectuer des calculs, etc., sans avoir à s'encombrer d'un ordinateur portable ou d'un bloc-notes.

Aujourd'hui ces périphériques ont atteint une puissance de calcul, une taille mémoire ainsi qu'un débit nécessaire pour faire tourner des applications aussi diverses que variées qui vont de l'Outlook mobile jusqu'aux applications de navigation GPS.

Les plates-formes de distribution de ces applications sont en plein essor, Windows Phone 7 à son magasin d'applications, l'iPhone à l'App Store, Android à son Market...etc. Ce qui ne cesse d'inciter beaucoup de développeurs à l'élaboration des petits logiciels très prisés en profitant des multiples apports des plateformes présentes sur le marché et des diverses innovations technologiques (Wifi, GPS, GPRS, etc.).

En effet, la géolocalisation du GPS des « téléphones intelligents » est très utile aux applications comme les annuaires, portails et autres outils permettant de trouver ce que l'on cherche autour d'un lieu, répondant par la fin aux besoins quotidiens de la communauté.

NAFTAL est une société par actions filiale à 100% de SONATRACH, son activité principale est la commercialisation et la distribution des produits pétroliers, organisée en trois branches principales : Carburants, Commercialisation et GPL.

La Branche carburants est chargée des activités d'approvisionnement, de stockage et de livraison des carburants Aviation, Marine et Terre ainsi que les lubrifiants et graisses

Les Moyens de transport de carburant sont par : Pipes, Cabotages, Rail et Camions.

Le moyen le plus important et le plus élevé est le moyen de transport par camion, il est de 58% du total transporté, représente plus de 700 Véhicules circulant à travers tout le territoire national, ce qui engendre une difficulté pour la société de connaître l'emplacement de ses véhicules à tout moment et de poursuivre le déplacement à temps réel de ses véhicules

C'est dans ce cadre que s'inscrit notre projet intitulé « Conception et réalisation d'une application de géolocalisation et de suivi Cas NAFTAL » dont l'objectif est de concevoir une application dédiée aux appareils mobiles, doté de la plateforme Android, permettant à son propriétaire la géolocalisation de son véhicule en cas d'oubli ou de perte, et d'une interface permettant à l'organisme le suivie de sa flotte, grâce aux outils et techniques de géolocalisation et d'orientation.

Pour mener à terme notre travail, nous l'avons réparti de la manière suivante :

- Le premier chapitre s'intitule « ETAT DE L'ART » comprend des généralités sur l'Android et les méthodes de Géolocalisation.
- Le deuxième chapitre concerne « L'ETUDE DE L'EXISTANT » nous faisons une présentation détaillée de l'organisme d'accueil.
- Le troisième chapitre dévoile « L'ANALYSE ET CONCEPTION » nous faisons une analyse détaillée et complète des cas d'utilisation repérés à travers le chapitre précédent, et nous élaborons une conception détaillée des cas d'utilisation, les diagrammes de séquence, ainsi que le diagramme de classe complet.
- Le quatrième chapitre s'intitule « REALISATION », nous présentons l'environnement matériel et logiciel, le passage vers le schéma relationnel et quelques composantes applicatives réalisées.

Enfin, nous clôturons ce mémoire par une conclusion dans laquelle nous résumons notre solution et exposant quelques perspectives futures.

CHAPITRE 1 : *ETAT DE L'ART*

PARTIE 1 : ANDROID

INTRODUCTION

En initiant le débat sur l'avenir de l'industrie du mobile, il apparaît que Google souhaite élargir autant que possible l'accès des utilisateurs à une grande variété de services comme les cartes géographiques, les réseaux sociaux et le partage de vidéo.

Android est un système d'exploitation pour Smartphone qu'est un projet né d'un consortium de 34 entreprises, initié par la société Google le 5 novembre 2007 et appelé l'Open Handset Alliance ou OHA. Son but consistait à trouver une solution fiable pour concurrencer Apple, l'iPhone OS, Microsoft avec Windows Mobile, Nokia avec Symbian, et Research In Motion avec Blackberry.

Android est distribué en open source sous licence Apache. La licence autorise les constructeurs qui intègrent Android dans leurs appareils à y apporter des modifications leur permettant de se distinguer de leurs concurrents et il a été adopté par de nombreux constructeurs de produits concurrents de l'iPhone.

Cette section a pour objectif la présentation de quelques notions sur la plateforme Android en premier lieu. Notamment sa définition, ses caractéristiques, ses versions et son architecture. Puis donnera un aperçu sur le développement d'application Android en particulier les composants d'une application Android, les outils de développement utilisés pour la mise en œuvre de ces applications. Enfin quelques exemples d'application Android sont présentés.

I. LE SYSTEME D'EXPLOITATION MOBILE

I.1. Définition

Un système d'exploitation mobile est un ensemble de programme responsable de la gestion des opérations, du contrôle, de la coordination, de l'utilisation du matériel et du partage des ressources d'un dispositif entre divers programmes tournant sur ce dispositif. Un système d'exploitation mobile est une plateforme logicielle sur laquelle les autres programmes appelés «programmes d'applications» peuvent s'exécuter sur des appareils mobiles tels que les PDA (personnal digital assistant), les téléphones cellulaires, Smartphone,....etc.

I.2. Caractéristique de d'un système d'exploitation mobile

Un système d'exploitation mobile regroupe un ensemble des fonctionnalités ; dont :

- La gestion de la mémoire ;
- La gestion des microprocesseurs et l'ordonnancement
- La gestion de système de fichiers
- La gestion des I/O
- La gestion de sécurité
- La gestion de fonctionnalités multimédia
-

I.3. Les principaux systèmes d'exploitation mobile

Les principaux systèmes d'exploitation mobiles sont : Android, Apple, Bada, Black Berry OS, Open Mako, Palm OS, HP web OS, Sybian OS, Windows CE, Windows mobile, Windows Phone 7...

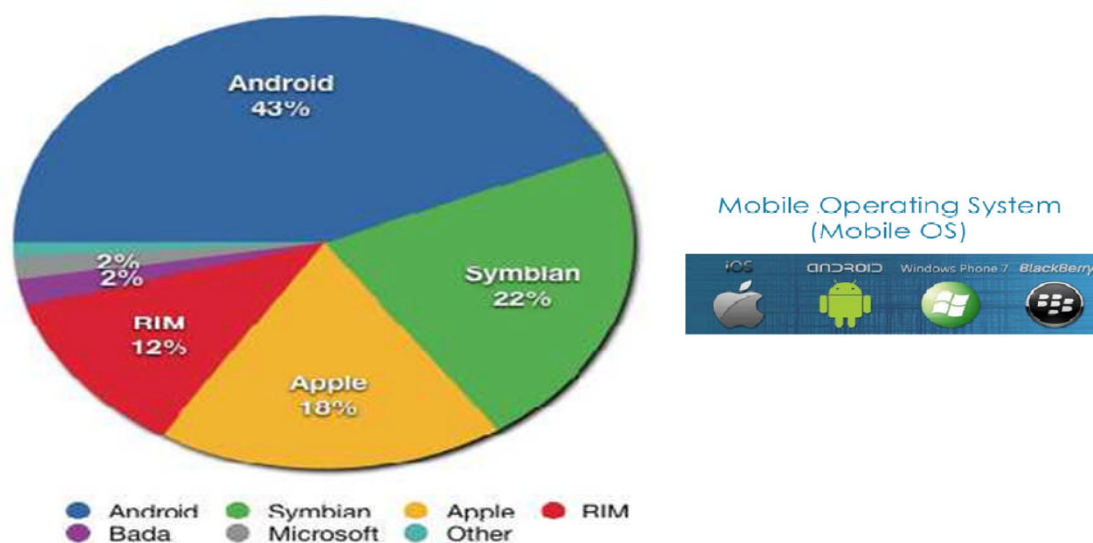


Figure 1 : Des systèmes d'exploitation mobiles

A partir de cette figure nous pouvons remarquer que le système Android domine largement les autres systèmes pour Smartphones. La section suivante est consacrée à la présentation de ce système.

I.4. Dispositifs mobile

Un dispositif mobile est un appareil informatique portable, qui a souvent un écran et une interface d'entrée/sortie, avec des dispositifs d'interaction nécessaires ou accessoires. Les dispositifs mobiles populaires peuvent être classés selon leurs caractéristiques comme ci-dessous :

- **LapTop** : Sont des ordinateurs portables de petite taille comme, UMPC (*ultra mobile personal computers*), laptop, net book, etc. ces dispositifs sont équipés de connexion réseau sans fil comme Wi-Fi, Bluetooth, 3G, etc. leur puissance de processeur, la résolution d'écran, la mémoire du système sont suffisantes pour utiliser la plupart des contenus multimédia. Le seul inconvénient est que leur mobile est moins importante que celle d'un PDA et d'un téléphone mobile, mais en même temps, plus que l'ordinateur portable.
- **Tablet pc** : Tablet pc est connu maintenant comme un ordinateur mobile en forme d'ardoise dépourvue de clavier et de souris, dont la principale interface est écran tactile
- **PDA (Personal digital assistant)** : Le PDA a une petite taille mais une puissance de processeur significative. Il peut reconnaître l'écriture manuscrite et peut réaliser beaucoup de tâches quotidiennes. Normalement, il a un écran plus grand qu'un Smartphone.
- **Téléphone portable** : Les téléphones portables peuvent être utilisés pour la communication vocale et l'envoi de messages textuels (SMS). Leur puissance informatique et le débit de transfert sont faibles. Mais avec la commercialisation de la 3G, les téléphones mobiles ont la possibilité d'accéder à internet via les technologies WAP, GPRS ou 3G, etc.
- **Smartphone** : Les Smartphones combinent les capacités du PDA et des téléphones portables. Ils des techniques d'interactions les plus modernes comme l'écran tactile multipoint de type iphone. La puissance de processeur et la mémoire sont aussi en croissance constante. Plusieurs types de systèmes d'exploitation sont utilisés comme Windows mobile, Windows ce, Symbian, linux mobile, Palm, Android, etc.
- **Autres dispositifs** : D'autres dispositifs comme les **baladeurs multimédia personnels** (mp3, mp4,...) ou les **consoles de jeux portables** (Sony PSP, Nintendo DS)

sont aussi considérés comme dispositifs mobiles, car ils ont une capacité de lecture de fichiers multimédia et une mémoire suffisant.

II. ETUDE DE LA PLATEFORME ANDROID

II.1. Le Système Android [1]

Le système Android est défini comme un système d'exploitation mobile pour Smartphones, tablettes tactiles, PDA, Smartwatches et d'autres terminaux mobiles. C'est un système open source utilisant le noyau Linux. Il a été lancé par une start-up du même nom rachetée par Google en 2005. D'autres types d'appareils possédant ce système d'exploitation existent par exemple des téléviseurs, des radioréveils, des montres connectées, des autoradios et même des voitures.

II.2. Les Caractéristiques du système Android [2]

Le système Android a les caractéristiques suivantes :




-  **Open source** : Le contrat de licence pour Android respecte les principes de l'open source, c'est à-dire-que c'est possible de le télécharger à tout moment et le modifier selon nos besoins. Android utilise des bibliothèques open source puissantes par exemple SQLite pour les bases de données et OpenGL pour la gestion d'images 2D et 3D.
-  **Gratuit** : autant pour les utilisateurs que pour les constructeurs. La publication d'applications sur le Play Store (une plateforme de téléchargement d'applications Android), permettent de publier autant d'applications que l'on souhaite gratuitement.
-  **Disponibilité via une licence Apache version 2** : le système d'exploitation inclut tous les utilitaires requis par un constructeur ou par un opérateur pour mettre en œuvre un téléphone portable.

TABLEAU N°01 : Caractéristique du système Android [2]

Framework	Framework Java pour le développement d'application pour la plateforme Android
Machine virtuelle Dalvik	Machine virtuelle spécialement développée pour Android. Cette machine virtuelle permet d'exécuter les applications java développées avec le Framework.
Navigateur web	Navigateur web basé sur le moteur de rendu Webkit
Graphique	Librarie graphique 2D, librarie graphique 3D basé sur OpenGL ES 1.0. Accélération matériel possible.
Stockage	Base de données SQL : SQLite est utilisé pour le stockage des données
Média	Android supporte les formats audio/vidéo/image suivants : MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF
Connectivité	GSM, EDGE, 3G, Bluetooth, wifi
Support Matériel	Android est capable d'utiliser Camera, GPS, accéléromètre
environnement de développement	Android possède un environnement de développement complet contenant : un émulateur, un débogueur, un analyseur de mémoires et de performances et un plugin Eclipse.

Figure 2 : Tableau Caractéristique du système Android [2]

II.3. L'historique et les versions d'Android

La version d'Android a débuté avec la sortie de la version 1.0 en septembre 2008. Depuis Android a connu plusieurs mises à jour. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces noms de codes suivant une logique alphabétique.

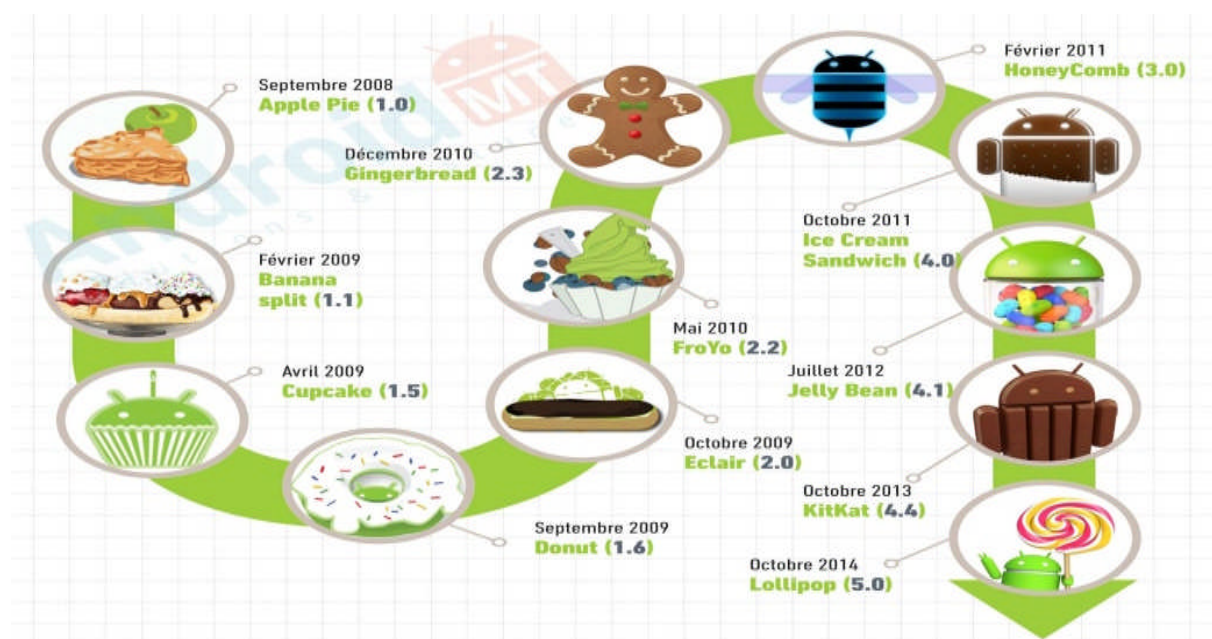


FIGURE 3 : Evolution des versions d'Android.

TABLEAU N°02 : synthèse des versions d'Android [3]

Numéro de version	Nom de la version	Date de sortie	caractéristiques majeurs ajoutées
Android 1.0	Apple pie	23-09-2008	Téléchargement et mise à jour des applications via Android Market.
Android 1.1	Banana bread	09-02-2009	Support pour sauvegarder les fichiers attachés aux MMS.
Android 1.5	Cupcake	30-04-2009	Envoi de vidéos vers You Tube et Picasa, rotation automatique.
Android 1.6	Donut	15-09-2009	Interface de l'Android Market améliorée. Amélioration de la rapidité dans la recherche Google Navigation.
Android 2.0	Eclair	26-10-2009	Supporte plus de taille d'écran et de résolutions. Réorganisation de l'interface utilisateur. Amélioration du clavier virtuel.

Android 2.2	Froyo	20-05-2010	Amélioration de la vitesse d'exécution. Partage de connexion USB. Supporte l'installation d'application sur support externe. L'envoi de fichier est supporté par le navigateur.
Android 2.3	Gingerbread	06-12-2010	Interface utilisateur est mise à jour. Supporte de grands écrans à résolution extra- large. Ajout d'un gestionnaire de téléchargement. Amélioration de la gestion de l'alimentation et du contrôle.
Android 3.2	Honeycomb	15-07-2011	Compatibilité pour les applications non-optimisées pour les résolutions d'écran des tablettes tactiles. Possibilité d'accéder aux fichiers de la carte SD.
Android 4.0	IceCream Sandwich	18-10-2011	Interface graphique plus ergonomique.
Android 4.1	Jelly Bean	09-07-2012	Extension de l'amélioration de la vitesse d'exécution et de gestion appareil photo. Accessibilité : mode gestuel, prise en charge clavier externe "Braille".
Android 4.4	KitKat	31-10-2013	Nouvelle interface translucide Enregistrement séquence vidéo de l'écran.
Android 5.0	Lollipop	17-10-2014	Nouvelle interface / design ("Material design"). Amélioration de la rapidité et la gestion de la batterie.
Android 6.0	Marshmallow	La fin mai 2015	Intégration finale du mode multifenêtre. Intégration de « adaptable storage » qui permet d'utiliser une carte microSD comme stockage

Figure 4 : Tableau synthèse des versions d'Android [3]

II.4. Architecture Android [4]

L'architecture de la plateforme Android se décline, selon une démarche bottom up, en quatre principaux niveaux que sont :

- Le noyau linux
- Les librairies et l'environnement d'exécution
- Le module de développement d'applications
- Les différentes applications

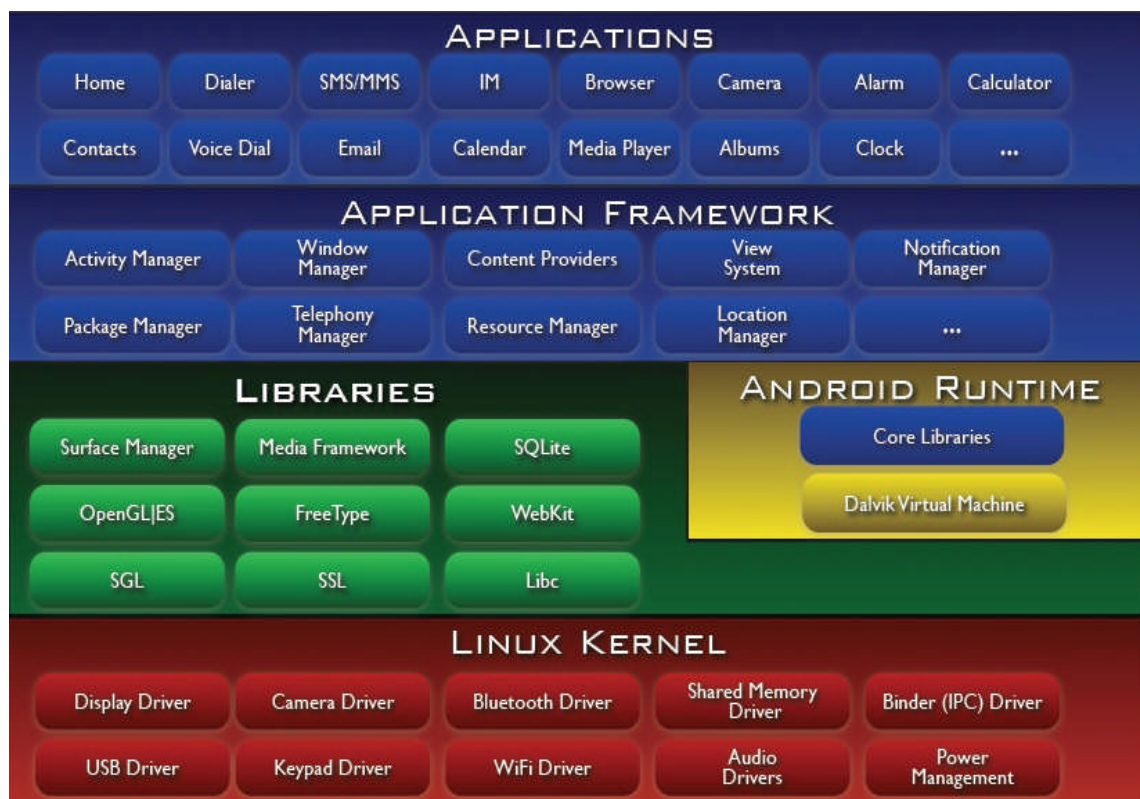


Figure 5 : L'Architecture de la plateforme Android [4]

II.4.1. Applications

Android est fourni avec un ensemble de programmes de base permettant d'accéder à des fonctionnalités comme les courriels, les SMS, le téléphone, le Web, etc. Ces applications sont développées à l'aide du langage de programmation Java. Pour l'utilisateur final, c'est la seule couche accessible et visible.

II.4.2. Le Framework (Application Framework)

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

A) CorePlatform Service

Les services cœurs de la plateforme (CorePlatform Services) fournissent des services essentiels au fonctionnement de la plateforme :

- ✓ **Activity Manager** : gère le cycle de vie des applications et maintient une "pile de navigation" (navigation backstack) permettant d'aller d'une application à une autre et de revenir à la précédente quand la dernière application ouverte est fermée.
- ✓ **Package Manager** : utilisé par l'ActivityManager pour charger les informations provenant des fichiers. "apk".
- ✓ **Window Manager** : juste au-dessus du "Surface Flinger", il gère les fenêtres des applications.
- ✓ **Resource Manager** : gère tous ce qui n'est pas du code, toutes les ressources images, fichier audio, etc.
- ✓ **Content Provider** : gère le partage de données entre applications, comme par exemple la base de données de contact, qui peut être consultée par d'autres applications que l'application "Contact ". Les données peuvent être partagées à travers une base de données (SQLite), des fichiers, le réseau, etc.
- ✓ **View System** : fournit tous les composants graphiques : listes, grille, texte box, buttons et même un navigateur web embarqué.

B) Hardwar Services :

Les services matériels (Hardware Services) fournissent un accès vers les API matérielles de bas niveau :

- ✓ **Telephony Service** : permet d'accéder aux interfaces "téléphonique".
- ✓ **Location Service** : permet d'accéder au GPS.
- ✓ **Bluetooth Service** : permet d'accéder à l'interface Bluetooth.
- ✓ **WiFi Service** : permet d'accéder à l'interface Wifi.
- ✓ **USB Service** : permet d'accéder aux interfaces USB.
- ✓ **Sensor Service** : permet d'accéder aux capteurs.

II.4.3. Les bibliothèques (Librairies)

Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants du système. Elles sont offertes aux développeurs par l'intermédiaire du "framework Android ". La liste ci-dessous énumère quelques-unes des bibliothèques disponibles dans Android :

- ✓ **Bibliothèque système C** : c'est une implémentation de la bibliothèque standard C, optimisée pour les systèmes Linux embarqués.
- ✓ **Bibliothèques multimédias** : basées sur StageFright, elles permettent le support de nombreux formats audio et vidéo.
- ✓ **SurfaceFlinger** : elle permet l'accès au sous-système d'affichage.
- ✓ **LibWebCore** : est un moteur de rendu de pages Internet basé sur Webkit. Cette bibliothèque est principalement utilisée dans le navigateur et dans les vues web embarquées (WebView).
- ✓ **Skia** : moteur graphique 2D.
- ✓ **Bibliothèques 3D** : est une implémentation basée sur OpenGL.
- ✓ **FreeType** : est un rendu des polices de caractères.
- ✓ **SQLite** : est une bibliothèque open source écrite en C permettant d'implémenter un moteur de base de données relationnelle.

II.4.4. Le moteur d'exécution Android (Android Runtime)

Android inclut un ensemble de bibliothèques qui fournit 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 machine virtuelle Dalvik.

Dalvik VM (Virtual machine) est une implémentation de machine virtuelle ayant été conçue pour optimiser l'exécution multiple de machines virtuelles. Elle exécute du bytecode qui lui est dédié : le bytecode dex. (Format qui est optimisé pour une empreinte mémoire minimale).

La figure suivant montre les étapes nécessaires à la compilation et à l'exécution d'un programme Android standard.

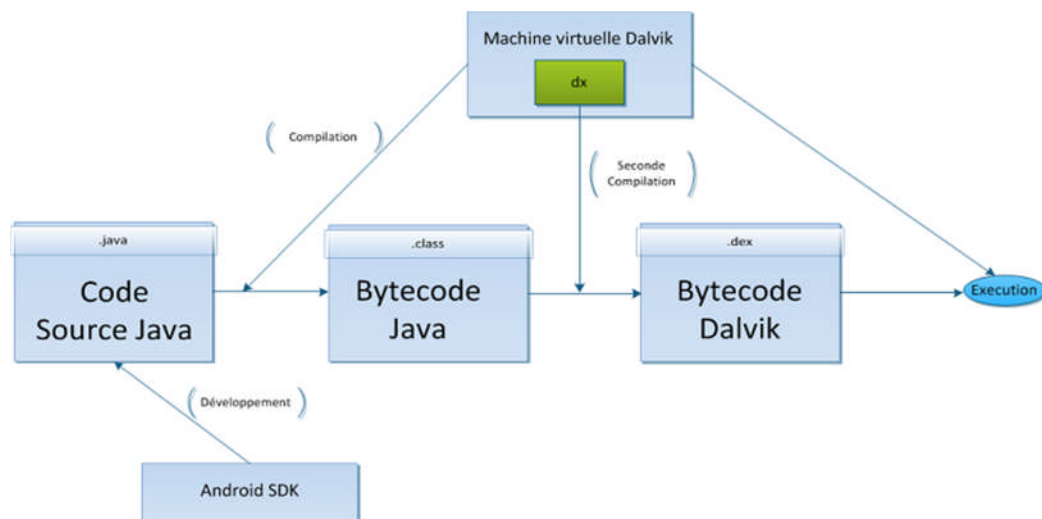


Figure 6 : Les étapes d'exécution d'un programma Android.

Un code source est une suite d'instructions que l'on trouve dans un fichier ".java" qui est traduit en une autre suite d'instructions dans un autre langage que l'on appelle le " bytecode ". Ce code est contenu dans un fichier ".class". Le bytecode est un langage spécial qu'une machine virtuelle Java peut comprendre et interpréter. Le bytecode Java ne peut être lu que par une machine virtuelle Java et Dalvik n'était pas une machine virtuelle Java. Il faut donc procéder à une autre conversion à l'aide d'un programme qui s'appelle "dx" qui s'occupe de traduire les applications de bytecode Java en bytecode Dalvik, qui est compréhensible par la machine virtuelle Dalvik, le résultat est mis dans un fichier ".dex".

II.4.5. Le noyau Linux (Linux Kernel)

Android s'appuie sur un noyau (kernel en anglais) linux 2.6. Pour être précis, le noyau est l'élément du système d'exploitation qui permet de faire le pont entre la partie matérielle et la partie logicielle. D'ailleurs, si vous regardez attentivement le schéma, vous remarquerez que cette couche est la seule qui gère le matériel. La version du noyau utilisée avec Android est une version conçue spécialement pour l'environnement mobile, avec une gestion avancée de la batterie et une gestion particulière de la mémoire.

III. DEVELOPPEMENT D'APPLICATION ANDROID :

III.1. L'ENVIRONNEMENT DE DEVELOPPEMENT SOUS ANDROID [5]

Afin de développer des applications sous Android, un ensemble d'outils est nécessaire. Vue que les procédures d'installation de ces outils sont assez longues et fastidieuses, alors les décrire, pas à pas, risquerai de prendre énormément de place et ainsi beaucoup de page. Alors, on se contentera juste d'évoquer les outils et leur intérêt.

III.1.1. Le JDK (Java Development Kit)

Les applications développées pour Android état essentiellement écrites en langage java ; un langage de programmation oriente objet qui a la particularité d'être très parable. Cela signifie qu'un programme java, fonctionnant sur Windows (par exemple), pourra facilement tourner sur MAC ou GNU/Linux.

Cette petit prouesse vient du fait que java s'appuie sur une machine virtuelle pour s'exécuter (appelée la JVM). Pour avoir une JVM sur votre ordinateur, il vous faut télécharger le JRE. Ce dernier contient, en plus de la JVM, des bibliothèques Java standards.

La JVM ne lit pas directement le code Java. Elle lit un code compile (le bytecode). Pour passer du code Java, que le développeur écrit, au code compile, lu par la JVM, des outils spéciaux sont nécessaires. Ces outils sont inclus dans le JDK. De plus, le JDK contient le JRE (et donc la machine virtuelle), ce qui est bien pratique.

Pour résumer, on dire que :

- Pour un simple utilisateur de Java : il doit avoir le JRE.
- Pour un développeur : il aura besoin des outils du JDK.

III.1.2. Le SDK (Software Development Kit) Android

Un SDK, c'est-à-dire un kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour Android et uniquement pour Android.

Au premier lancement du SDK, un écran semblable à la figure suivante s'affichera :

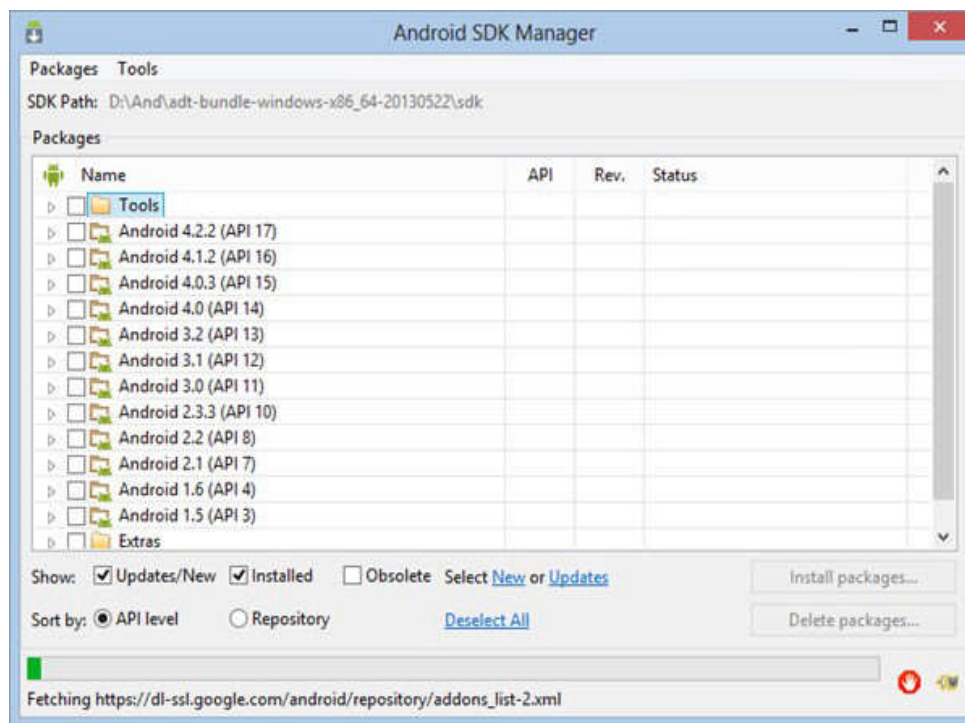


Figure 7 : Le SDK Android.

En regardant bien le nom des paquets, vous remarquerez qu'ils suivent tous un même motif. Il est écrit à chaque fois : Android [un nombre] (API [un autre nombre]). La présence de ces nombres s'explique par le fait qu'il existe plusieurs versions de plateforme en circulation. Le premier nombre correspond à la version d'Android et le second à la version de l'API Android associée. Quand on développe une application, il faut prendre en compte ces numéros, puisqu'une application développée pour une version précise d'Android ne fonctionnera pas pour les versions antérieures.

III.1.3. L'IDE Eclipse

Eclipse est un environnement de développement intégré. C'est un logiciel qui permet d'écrire un programme beaucoup plus facilement qu'avec le simple Bloc-notes. Outre la coloration du code, il permet d'apporter des outils très pratique pour compiler vos programmes, les déboguer, etc. Il peut être utilise pour programmer avec n'importe quel type de langage, mais nous l'utiliserons pour faire du Java. De plus, Eclipse est conçu pour pouvoir être complète avec des plugins (extension). Ainsi, il existe un plugin pour développer des applications Android que nous verrons dans la partie suivante.

III.1.4. Le Plugin ADT pour Eclipse

Google fournit un plugin pour Eclipse, nomme ADT (Android Development Tools). La fonction principale de ce plugin est de créer un pont entre Eclipse et le SDK Android.

III.1.5. L'émulateur de téléphone (Android Virtual Device) [6]:

L'Android Virtual Device, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui fait croire à votre ordinateur qu'il est un appareil sous Android. C'est la raison pour laquelle vous n'avez pas besoin d'un périphérique sous Android pour développer et tester la plupart de vos applications. En effet, une application qui affiche un calendrier par exemple peut très bien se tester dans un émulateur, mais une application qui exploite le GPS doit être éprouvée sur le terrain pour que l'on soit certain de son comportement.

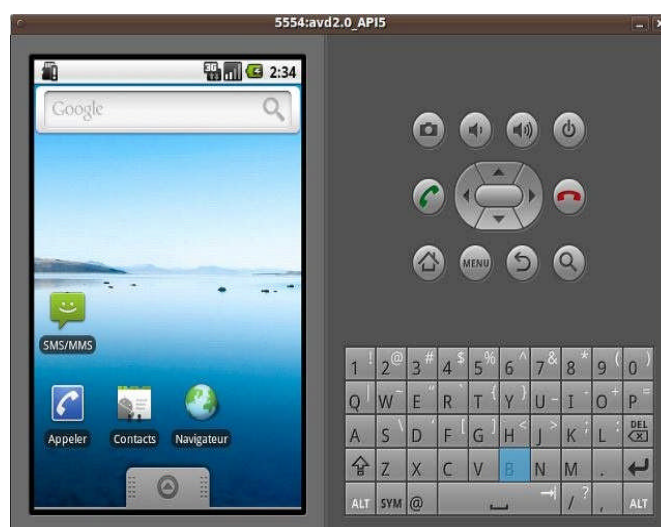


Figure 8 : L'émulateur [6]

III.2. COMPOSENT D'ANDROID [7]

Une application Android se compose de plusieurs éléments. Dans ce qui suit, nous allons essayer de découvrir les plus importants :

III.2.1. Les Activités

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

Plusieurs choix se proposent pour mettre en place l'interface visuelle :

- Utiliser un fichier XML pour décrire l'interface ;
- Créer les éléments de l'interface à l'intérieur du code java ;

III.2.2. Les services

Un service est, en fait, un programme tournant en tâche de fond et n'ayant pas d'interface graphique. L'exemple commun illustrant cette notion, est celui du lecteur mp3.

Un lecteur mp3 ne nécessite pas, pour la plupart du temps, d'interface graphique et doit tourner en tâche de fond, laissant la possibilité aux autres applications de s'exécuter librement.

Un service peut être lancé à différents moments :

- Au démarrage du téléphone ;
- Au moment d'un événement (arrivée d'un appel, SMS, mail, etc.) ;
- Lancement de l'application ;
- Action particulière dans l'application ;

III.2.3. Les Broadcast Receivers

Un Broadcast Receiver, comme son nom l'indique, permet d'écouter ce qui se passe sur le système ou sur votre application et de déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés.

III.2.4. Les Content Providers

Les Content Provider sont, comme l'exprime leur nom, des gestionnaires de données. Ils permettent de partager l'information entre applications. Vous pouvez accéder :

- Aux contacts stockés dans le téléphone ;
- A l'agenda ;
- Aux photos ;
- Ainsi que d'autres données depuis votre application grâce aux content providers ;

III.2.5. Les Intents

Les Intents sont des objets permettant de faire passer des messages contenant de l'information entre composants principaux. La notion d'Intent peut être vue comme une demande de démarrage d'un autre composant, d'une action à effectuer. Que de l'accéléromètre pour offrir une Street View de la carte.

III.3. CYCLE DE VIE D'UNE APPLICATION ANDROID [8]

Le diagramme d'état suivant présente les principaux états du cycle de vie d'une activité Android, il est suivi d'une description des principales méthodes événementielle du cycle de vie d'une activité

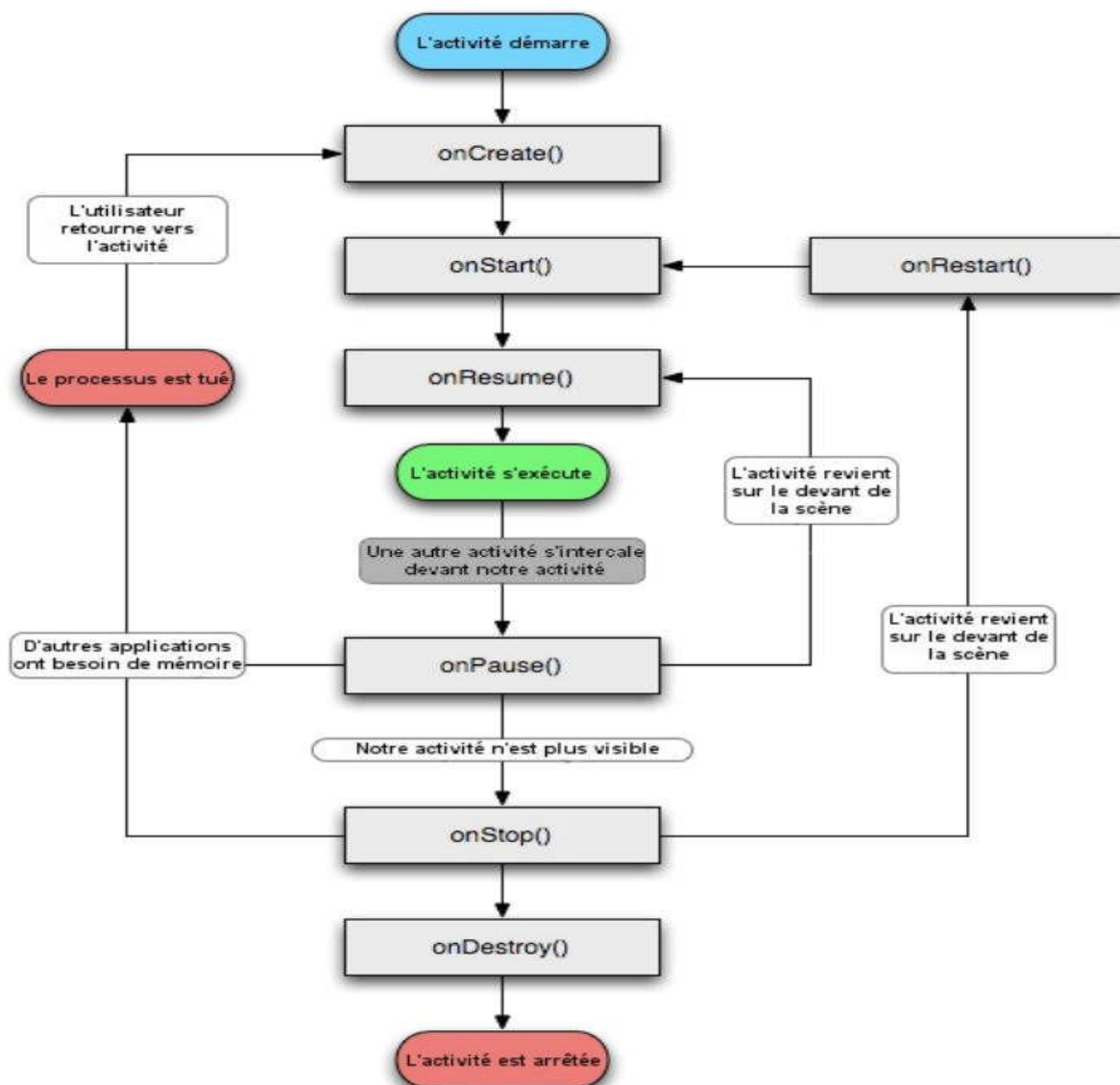


Figure 9 : Le cycle de vie d'application Android [8]

Le changement d'état d'une activité est provoqué par l'exécution d'une méthode particulière. Ci-dessous nous présentons les différentes méthodes :

- **onCreate** : Appelée quand votre première activité est créée. Elle sert à initialiser votre activité. Quand **onCreate** est appelée, on lui passe un Bundle qui contient l'état de sauvegarde précédent de l'activité s'il existe.
- **onStart** : Appelée juste avant que votre activité passe en avant plan (ne soit visible à l'écran). Si votre activité ne peut pas aller en avant plan pour n'importe quelle raison, l'activité sera transférée à **onStop**
- **onResume** : Appelée après **onStart** si votre activité est en avant plan, à ce stade de l'exécution votre activité est en cours d'exécution et l'interaction avec l'utilisateur est possible. **onResume** est aussi appelées quand votre activité passe en arrière-plan à cause d'une autre application.
- **onPause** : appelée juste avant qu'une activité que la vôtre passe en **onResume**. A ce stade, votre activité n'a plus accès à l'écran.
- **onStop** : appelée quand votre activité n'est plus visible, soit parce qu'une autre application a pris l'avant plan ou parce que votre application est terminée.
- **onDestroy** : appelée juste avant la fermeture de l'activité.

IV. Les outils et technique de géolocalisation et d'orientation sous Android :

IV.1. Google Maps API :

L'API Google Maps est l'une des applications de cartographie les plus utilisées au monde. C'est une application de service de géolocalisation gratuite en ligne. Il s'agit d'un géo portail lancé il y a quelques années aux Etats-Unis puis à l'Europe. Elle offre une vue de carte sur quatre plans à savoir un plan classique, un plan en image satellite, un plan mixte et un plan relief de la région.

Pour intégrer ces cartes interactives Google Maps à son propre application et bénéficier des données associées, l'utilisateur doit disposer d'une clé (Google Map API Key) propre à son domaine d'utilisation.

IV.2. Le GPS :

C'est un système de positionnement par satellite connu sous le nom du « Global Positioning System », conçu à l'origine par le département de la Défense des Etats-Unis. Il repose sur 24 satellites qui transmettent les positions en orbite ainsi que des signaux horaires. Couplé avec un logiciel de navigation, le GPS permet à l'utilisateur de connaître sa position, planifier son itinéraire, estimer son temps du trajet, connaître la localisation des radars, etc. Il renvoie même la vitesse de son déplacement à chaque instant. En outre, le positionnement par satellite est entièrement gratuit, mais pour s'en servir il faut disposer d'un équipement GPS qui est à acheter. Profitant de cette technologie, Google a créé Google Maps Navigation. Il s'agit d'un service de guidage par GPS ou encore un système de navigation GPS, gratuit disponible sur les téléphones Android.

IV.3. Android Radar :

En exploitant le Framework ouvert offert par Android, plusieurs applications sont disponibles chez Android Market, dont le radar d'Android. Ce logiciel de géolocalisation se sert du GPS pour déterminer selon la position actuelle de son utilisateur celle de sa cible en plus de la distance qui les sépare.

IV.4. Android Compass :

Pour s'orienter dans l'espace, certains téléphones Android embarquent une boussole numérique avec l'accéléromètre (pour calculer la vitesse) et les capteurs de position. Parmi les applications qui tirent avantage de cette boussole on cite Google Maps qui s'en sert ainsi que de l'accéléromètre pour offrir une Street View de la carte.

PARTIE 2 :
GEOLOCALISATION

INTRODUCTION

Contrairement à ce que l'on pourrait penser, les technologies de géolocalisations ne sont pas récentes. En effet, depuis la préhistoire, se situer dans l'espace et savoir prendre des points de repère pour trouver son chemin et le communiquer à d'autres fait partie de l'apprentissage de base de l'Homme. Ce sont les vecteurs par lesquels la localisation s'effectue qui ont évolué au fil du temps en fonction des progrès technologiques qui ont graduellement été débloqués.

La géolocalisation fait usage des plusieurs technologies différentes, touchant aux domaines d'activités complémentaires. Dans ce chapitre, nous présentons les concepts de base de la géolocalisation, principalement, nous définissons cette notion. Ensuite nous arborons les différentes techniques de géolocalisation, où nous détaillons la technique de géolocalisation par GPS. Puis nous traitons la plateforme de géolocalisation. Enfin, nous présentons les différents terminaux existants pour la géolocalisation.

I. LA GEOLOCALISATION

I.1. Définition [9]

La géolocalisation est un procédé permettant de positionner un objet sur un plan ou une carte à l'aide de ses coordonnées géographiques. Cette opération est réalisée à l'aide d'un terminal capable d'être localisé et de publier ses coordonnées géographiques (latitude/longitude). Les positions enregistrées peuvent être stockées au sein du terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation.

La transmission temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM, GPRS, UMTS, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet.

II. TECHNIQUE DE GEOLOCALISATION :

II.1. Géolocalisation par satellite [11]

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 peut alors être représentée physiquement sur une carte. Le réseau satellite de positionnement le plus connu est le **GPS** (Global Positioning System). Ces satellites constituent un maillage du ciel et servent de repères aux navigateurs GPS dans leur processus de calcul de position. Pour qu'un terminal soit capable de se géo-localiser grâce au réseau GPS, celui-ci doit être équipé d'une puce électronique GPS. Le GPS offre une précision allant de 15 à 100 mètres pour les applications civiles.



Figure 10 : Géolocalisation par satellite.

II.2. Géolocalisation par GSM (Global System Mobile) [Neviere, 2001]

Cette technique permet le positionnement d'un terminal GSM en se basant sur certaines informations relatives aux antennes GSM auxquelles le terminal est connecté. La précision du positionnement par GSM peut aller de 200 mètres à plusieurs kilomètres, selon que le terminal se trouve en milieu urbain, ou en milieu rural. Plusieurs techniques existent :

- **Le différentiel temps (E-OTD Enhanced Observed Time Difference) :**

Une station doit recevoir un signal synchrone de la part du MS¹, 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. L'exactitude de la position est de 125m, mais à la différence du GPS cette méthode ne dépend pas de la clarté du ciel.

- **L'angle d'arrivée (TAO 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.

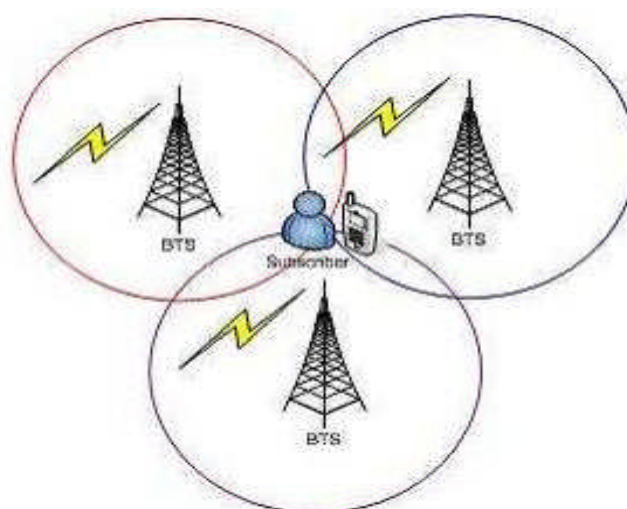


Figure 11 : Positionnement par TAO

- **Le système d'identification de cellule « Cell ID » :**

Aujourd'hui, la méthode GSM la plus utilisée est celle du Cell ID (identification de la cellule radio), consiste à récupérer les identifiants des BTS auxquelles le terminal est connecté à un instant donné. Par la suite, grâce à une base de données faisant le lien entre les identifiants des cellules et les positions géographiques des BTS, la station de base est capable de déterminer la position du terminal et d'émettre une estimation. Ces bases de données peuvent être mises à disposition par les opérateurs pour leurs abonnés, ou par des sociétés privées qui recensent les antennes GSM ou ayant des partenariats avec les opérateurs. Étant donné que les bases de données Cell ID ne sont pas stockées localement dans le terminal, une connexion internet de type GPRS/EDGE ou 3G peut être nécessaire afin d'émettre une requête pour obtenir la correspondance Cell ID / longitude latitude.

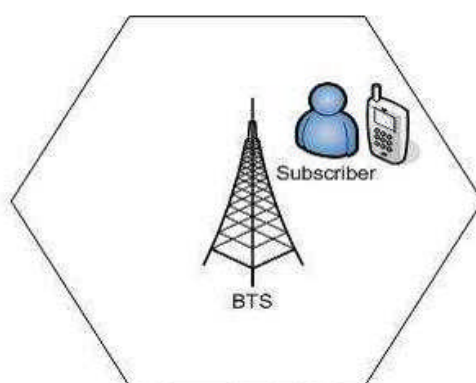


Figure 12 : Positionnement par Cell ID

II.3. La géolocalisation par Wi-Fi [11]

De la même façon qu'un terminal GSM peut se localiser par la méthode du Cell ID sur un réseau GSM, un terminal Wi-Fi peut utiliser la même méthode en se basant sur les identifiants des bornes Wi-Fi (adresses MAC) qu'il détecte. Il existe des bases de données recensant une multitude de bornes d'accès Wi-Fi ainsi que leur position géographique.

Ces bases peuvent appartenir à des entreprises privées ou à des communautés qui les publient gratuitement. Ces bases de données sont construites en utilisant la méthode appelée War Driving, qui consiste à parcourir les rues des villes en voiture avec un ordinateur portable équipé de Wi-Fi et relié à un récepteur GPS, afin de recenser un maximum de points d'accès Wi-Fi.

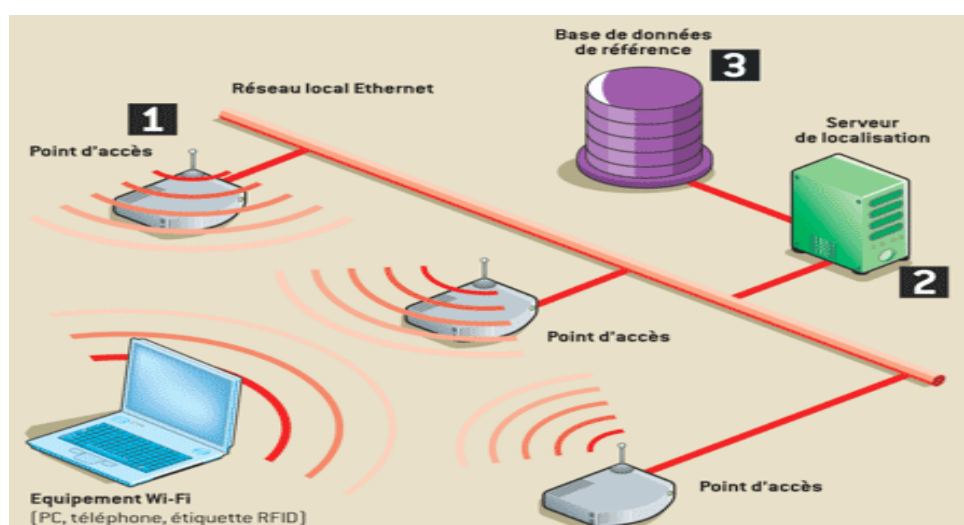


Figure 13 : Géolocalisation par wifi [10]

II.4. La géolocalisation par l'adresse IP [12]

Cette méthode permet de déterminer la position géographique d'un ordinateur ou de n'importe quel terminal connecté à internet en se basant sur son adresse IP (Internet Protocol). L'adresse IP (Internet Protocol) est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque appareil connecté à un réseau informatique utilisant l'Internet Protocol. L'adresse IP est à la base du système d'acheminement (le routage) des messages sur Internet.

Les adresses IP sont gérées par l'IANA (Internet Assigned Numbers Authority), une organisation qui s'occupe de découper les blocs d'adresses IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent. Toutes ces attributions étant très bien documentées, il est possible de savoir dans quel pays se trouve un terminal connecté à internet grâce à son adresse IP. On peut même obtenir un niveau de précision de l'ordre de la ville en se basant sur la distribution des adresses IP faite par les fournisseurs d'accès à internet.

Pour une adresse IP associée à une date et une heure données, les forces de police peuvent obtenir auprès du fournisseur d'accès à internet la position géographique exacte, le nom et l'adresse communiquées par le titulaire. Mais ces données ne sont conservées qu'un temps limité, dont la durée varie selon les pays. De même pour un espace commercial d'accès à Internet ("cybercafé"), selon les pays le commerçant peut être tenu de garder les dates, heures et documents d'identité scannés pendant une certaine période.

II.5. Géolocalisation par RFID (Radio Frequency Identifier) [CiscoMag, 2006]

La **radio-identification**, le plus souvent désignée par le sigle **RFID** est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes ». (« RFID tag » ou « RFID transponder » en anglais).

- ✓ **Système RFID** : Comme la figure ci-dessus l'indique, RFID est un système qui se compose de transpondeurs (aussi nommés étiquettes, marqueurs, tags, identifiants...) et d'un ou plusieurs interrogateurs (aussi nommés coupleurs, lecteur antenne...).

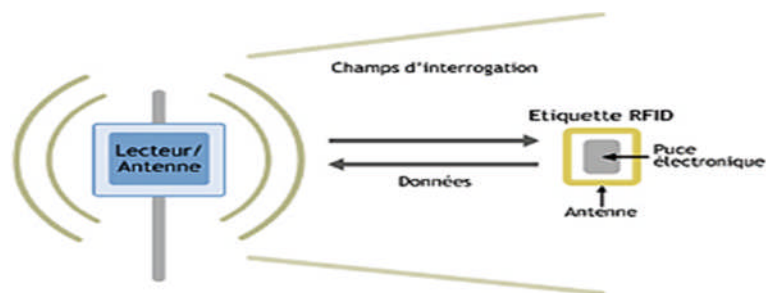


Figure 14 : Principe du Système RFID

- ✓ **Interrogateurs RFID** : Ce sont des dispositifs actifs, émetteurs de radiofréquences qui vont activer les tags qui passent devant eux en leur fournissant l'énergie dont ils ont besoin pour fonctionner. Outre de l'énergie pour l'étiquette, l'interrogateur envoie des commandes particulières auxquelles répond le tag. L'une des réponses les plus simples possibles est le renvoi d'une identification numérique.
- ✓ **Étiquettes RFID** : C'est un dispositif récepteur, que l'on place sur les éléments à tracer (objet, animal...). Ils sont munis d'une puce contenant les informations et d'une antenne pour permettre les échanges d'informations.

Par ailleurs, il existe deux grandes familles d'étiquettes RFID :

- ✓ **Les étiquettes actives** : reliées à une source d'énergie embarquée (pile, batterie, etc.). Les étiquettes actives possèdent une meilleure portée mais leur coût est plus élevé et leur durée de vie restreinte.
- ✓ **Les étiquettes passives** : utilisant l'énergie propagée à courte distance par le signal radio de l'émetteur. Ces étiquettes à moindre coût sont généralement plus petites et possèdent une durée de vie quasi-illimitée. En contrepartie, elles nécessitent une quantité d'énergie non négligeable de la part du lecteur pour fonctionner.

Avantage

- ✓ Bonne précision de localisation.
- ✓ Lire rapidement plusieurs étiquettes RFID en même temps.
- ✓ Protection de l'étiquette pour résister aux contraintes (chocs, traitements thermiques, lavage).

Inconvénient

- ✓ Cette technologie nécessite la mise en place d'un réseau spécifique, sensible à l'environnement (liquide, métal...)
- ✓ Coût très élevé des tags.
- ✓ Sensibilité aux ondes électromagnétique parasites.

Exemples d'utilisation

Cette technologie de localisation peut être utilisée pour identifier :

- ✓ Inventaire rapide en magasin.
- ✓ Suivi des vêtements de travail loués.
- ✓ Identification des animaux : remplace le tatouage.
- ✓ Gestion des livres d'une bibliothèque.

II.6. Combinaison de différente technique :

Il existe plusieurs inconvénients à l'utilisation d'une seule technique de géolocalisation :

- **La dépendance au réseau GPS :** L'incapacité de l'utiliser en intérieur et le temps de réponse à l'allumage.
 - **La dépendance au réseau GSM :** Sa couverture géographique, l'accès au réseau GPRS pour exploiter l'information.
 - **La dépendance à la présence de bornes d'accès Wi-Fi :** En zone rurale par exemple
- Les dispositifs qui combinent ces trois techniques sont capables à géo-localiser le terminal dans n'importe quelle situation. La précision de ce positionnement varie en fonction des technologies disponibles, mais le temps de réponse à l'allumage et l'adaptabilité s'en verront améliorées. Ceci permet par exemple de géo-localiser une personne à l'extérieur en utilisant le GPS, et de garder sa trace à l'intérieur des bâtiments ou des tunnels en utilisant la technologie GSM couplée au Wi-Fi pour plus de précision.

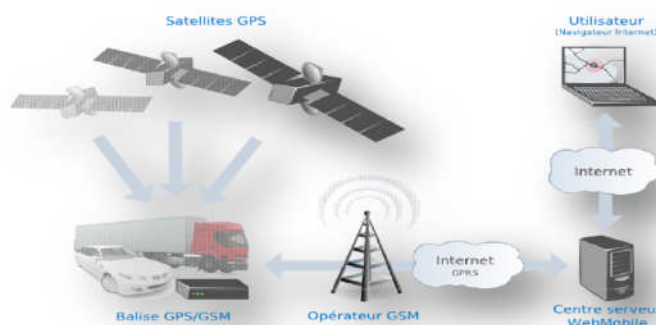


Figure 15 : Géolocalisation GPS/GSM.

III. ARCHITECTURE TEMPS REEL

La position géographique d'un terminal géo-localisé reste néanmoins une information brute qui peut être exploitée et couplée à d'autres données afin de créer une vaste quantité de services à forte valeur ajoutée. Afin d'exploiter cette information, la donnée (position) générée par un terminal qui se trouve sur le terrain doit être transmise à une plateforme logicielle qui va la traiter, la présenter graphiquement à l'utilisateur et l'associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux.

IV. LES PLATEFORMES DE GEOLOCALISATION

IV.1. Définition

Une plateforme de géolocalisation est l'ensemble des environnements manipulés pour localiser des matériels, personnes et informations. La figure ci-dessous nous montre ses composants :

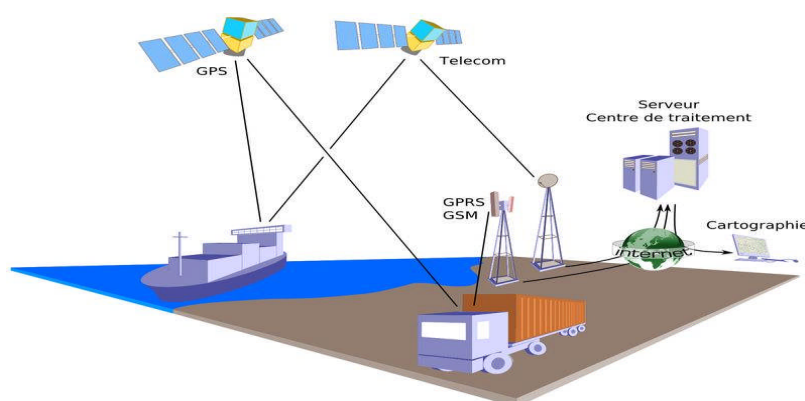


Figure 16 : Les composants d'une plateforme de géolocalisation.

IV.2. Les composants de la plateforme de géolocalisation

Les composants essentiels d'une plateforme de géolocalisation sont :

IV.2.1. Le terminal communicant

C'est le terminal qui reçoit ses coordonnées géographiques (via GPS ou tout autre moyen de géolocalisation) et qui les envoie via un réseau de télécommunications à la plateforme. Typiquement, les terminaux de géolocalisation existants peuvent être classés dans une de ces trois (03) catégories, même si certains peuvent être configurés pour fonctionner dans un mode au choix : les data loggers, les data pullers et les data pushers.

a) Les Data loggers : Ces terminaux sont généralement équipés de récepteurs GPS et se limitent à stocker dans leur mémoire interne leur position à des intervalles réguliers. Certains loggers GPS disposent de slots pour carte mémoire et/ou d'une mémoire interne ainsi qu'un port USB (le terminal est vu comme une clé USB). Ceci permet, postérieurement, de télécharger les données sur un ordinateur pour les traiter. Ce genre de terminaux est surtout utilisé par des sportifs (joggers, amateurs de VTT, etc.) qui téléchargent ensuite les données sur leur ordinateur personnel afin de calculer la durée du parcours ou d'afficher les points sur une carte à l'aide d'un logiciel SIG. Pour des parcours longs et non contrôlés, ce genre de dispositif peut permettre de déterminer le gagnant d'une épreuve et de déterminer si celui-ci est bien passé par des points de contrôle. Ces terminaux peuvent aussi être combinés à des appareils numériques afin de géo-tagger (insérer de coordonnées GPS dans les photos) les photos en se basant sur l'heure de la prise. Ces terminaux peuvent aussi être utilisés dans le cadre de surveillances ou de suivi de véhicules où l'émission de données serait impossible.

b) Les Data pushers : Ce sont les terminaux les plus utilisés pour des applications professionnelles. Ces terminaux envoient leur position à des intervalles réguliers et programmables vers une plateforme de géolocalisation qui traite la donnée en temps réel. Parmi les terminaux capables d'être géo-localisés et de transmettre l'information en temps réel, nous distinguons :

- Les terminaux mobiles de type PDA ou Smartphone équipés d'un GPS et/ou d'un modem GSM/GPRS.
- Les boîtiers de géolocalisation dédiés embarquant un récepteur GPS, un dispositif de télécommunications (GSM/GPRS ou satellite) et avec une capacité optionnelle de télé relèvement (consiste à récupérer à distance une série d'informations issues de capteurs ou de systèmes informatiques).
- Les dispositifs de poche destinés aux personnes et embarquant un récepteur GPS et un modem GSM/GPRS ou satellite. C'est une source électrique ou être autonomes grâce à une batterie interne. Les dispositifs peuvent nécessiter le branchement à une batterie externe. En fonction de l'utilisation du terminal, celui-ci pourra être connecté à la batterie d'un véhicule ou disposer d'une grande autonomie (exemples : suivi d'objets sans sources électriques comme des conteneurs, colis, animaux, remorques, etc...).

c) **Les Data pullers** : Contrairement aux dispositifs de type "push", les data pullers se limitent à envoyer l'information uniquement sur demande. Ces dispositifs sont suffisants dans les cas où la position de l'objet ou la personne n'a pas besoin d'être connue en continu. Par exemple, la position du véhicule ne sera nécessaire que si celui-ci est volé. De plus, cette méthode permet aux opérateurs de télécommunications de commercialiser un service de géolocalisation à travers des terminaux mobiles dépourvus de récepteurs GPS et de données mobiles. Il suffit d'avoir le numéro de téléphone du mobile et d'y être autorisé pour envoyer une requête de demande de position. L'opérateur localise alors la position du terminal grâce à la technique d'identification cellulaire et vous envoie la position de celui-ci. La facturation se fera typiquement à la position.

IV.2.2. Le système informatique

Il s'agit de serveurs informatiques qui hébergent l'infrastructure et qui reçoivent et traitent les données envoyées par les terminaux. Ce sont les mêmes serveurs qui mettront à disposition l'information aux utilisateurs (via une interface web par exemple).

IV.2.3. Le module cartographique

C'est le module intégré au système informatique qui permet d'afficher la position des terminaux sur un fond cartographique adapté. Ce module prend en charge les calculs de distances, d'itinéraires, détecte l'interaction avec les zones, et permet d'avoir accès à des informations sur terrain. Par exemple, le Google Maps, qui est une application intégrée dans le système Android qui permet d'explorer le monde plus rapidement et plus facilement.

IV.3. Le fonctionnement de la plateforme de géolocalisation

La plateforme de géolocalisation utilise une architecture en temps réel. La position géographique d'un terminal géo-localisé reste néanmoins une information brute qui peut être exploitée et couplée à d'autres données afin de créer une vaste quantité de services à forte valeur ajoutée.

Afin d'exploiter cette information, la donnée (position) générée par un terminal qui se trouve sur le terrain doit être transmise à une plateforme logicielle qui va la traiter, la présenter graphiquement à l'utilisateur et l'associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux. Les étapes de la chaîne

de traitement de l'architecture en temps réel se présentent comme suit :

1. Le terminal détermine sa position géographique grâce à une des techniques de géolocalisation citées précédemment (de préférence GPS, GSM et/ou Wifi).
2. Il envoie ces données vers une plateforme logicielle soit par le réseau GSM/GPRS soit par un réseau satellitaire de type Inmarsat (C'est une Organisation Maritime Satellitaire Internationale).
3. La plateforme logicielle de géolocalisation traite la donnée et positionne le terminal géographiquement sur une carte moyennant la précision offerte par la technique de géolocalisation utilisée.
4. Cette carte ainsi que tous les traitements effectués sont mis à disposition de l'utilisateur à travers un portail web hébergé sur un serveur accessible depuis internet, ou à travers d'une application métier installée sur le poste de travail.

CONCLUSION

Tout au long de ce chapitre, nous avons introduit l'informatique mobile dans son contexte ainsi que les définitions précises du support et de la plateforme adoptés.

Nous avons ensuite énuméré les technologies utilisées pour réaliser ce projet, pour garantir une meilleure compréhension du sujet et des fonctionnalités du système.

CHAPITRE 2 :
ETUDES DE L'EXISTANT

INTRODUCTION

Le pétrole et le gaz sont des produits de base stratégiques pour l'économie algérienne. Dans le but d'assurer le rôle de ces produits, et juste après la nationalisation du gaz et du pétrole Algérien en 1971, la mission de la gestion du secteur hydrocarbure fut attribuée à la société SONATRACH, qui ensuite s'est restructurée en filiales spécifiques pour chaque tâche telles que NAFTEC qui s'est chargée de l'opération du raffinages des produits pétroliers et NAFTAL qui s'est chargée de la distribution et la commercialisation des hydrocarbures.

Dans ce chapitre, nous allons donner une vue générale sur la société NAFTAL et une vue plus détaillée sur la branche Carburants.

I. PRESENTATION DE LA SOCIETE NAFTAL

NAFTAL est une société par actions filiale à 100% de SONATRACH. Elle a comme mission principale la commercialisation et la distribution des produits pétroliers. Elle compte plus de 30000 employés et son capital est estimé à 15 650 000 000 000 DA.

Son siège est sis à rue des Dunes BP 73, CHERAGA, ALGER.

II. HISTORIQUE

Issue de SONATRACH, l'entreprise ERDP (Entreprise de Raffinage et de Distribution des Produits Pétroliers) a été créée par le décret N° 80/101 du 06 avril 1981. Entrée en activité le 01 janvier 1982, elle a été chargée de l'industrie du raffinage des hydrocarbures liquides, de la commercialisation et de la distribution des produits pétroliers sur le territoire national sous le sigle NAFTAL. L'appellation « NAFTAL » est tirée de deux termes principaux : « NAFT » : qui désigne le pétrole et « AL » en référence à Algérie.

En 1987 l'activité « raffinage » est séparée de l'activité « distribution ». Suite à cette séparation la raison sociale de la société change, NAFTAL est désormais chargée uniquement de la commercialisation et la distribution des produits pétroliers et de leurs dérivés sur le territoire national. A partir de 1998, NAFTAL change de statut et devient société par actions, filiale à 100% de SONATRACH.

III. ORGANISATION

NAFTAL est organisée en trois branches principales :

- ✓ Branche Carburants.
- ✓ Branche Commercialisation.
- ✓ Branche GPL.

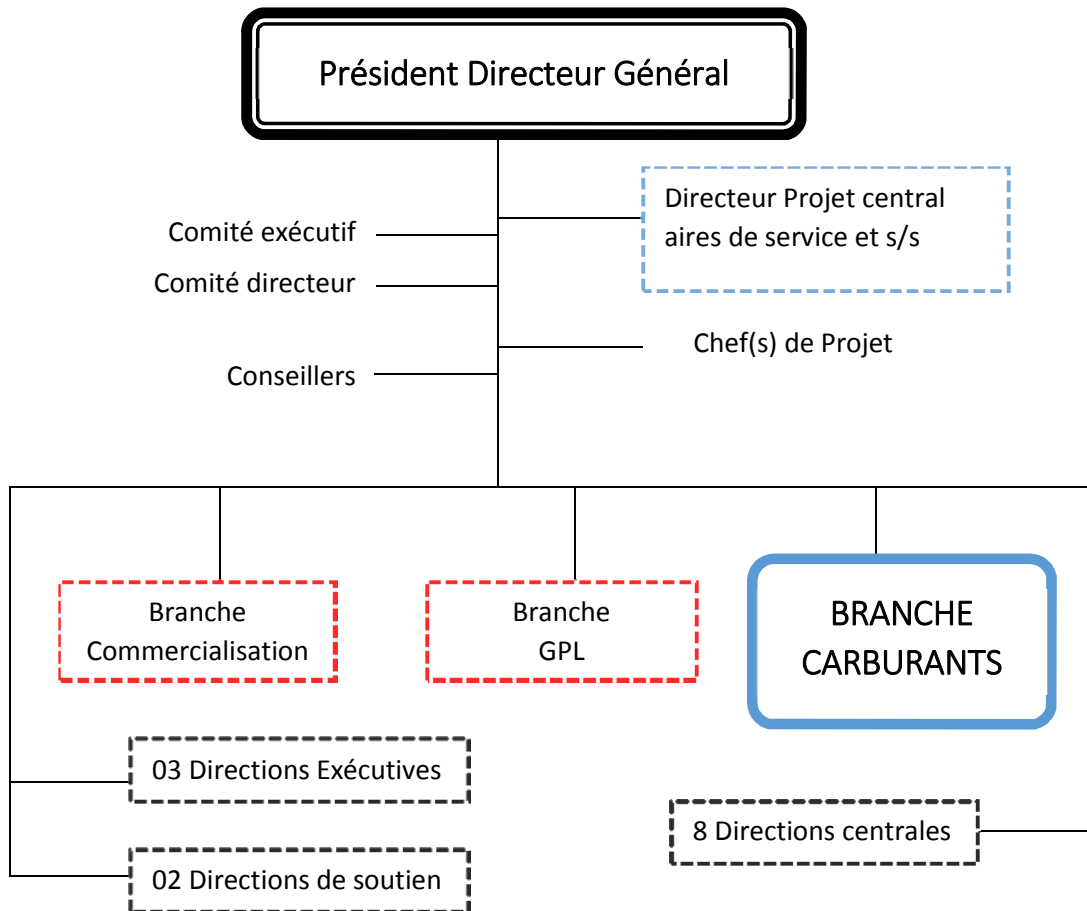


Figure 17 : Organigramme de la société NAFTAL

IV. PRESENTATION DE LA BRANCHE CARBURANTE

La Branche carburants est l'une des trois branches de NAFTAL. Elle est chargée des activités d'approvisionnement, de stockage et de livraison des carburants Aviation (Jet-A1 et Methmix), Marine (Gas-oil et fuel-oils) et Terre (Essences Super, normal et sans plomb, Gas-oil, A72, white spirit) ainsi que les lubrifiants et graisses aviation et marine.

V.1. ORGANISATION DE LA BRANCHE CARBURANTE :

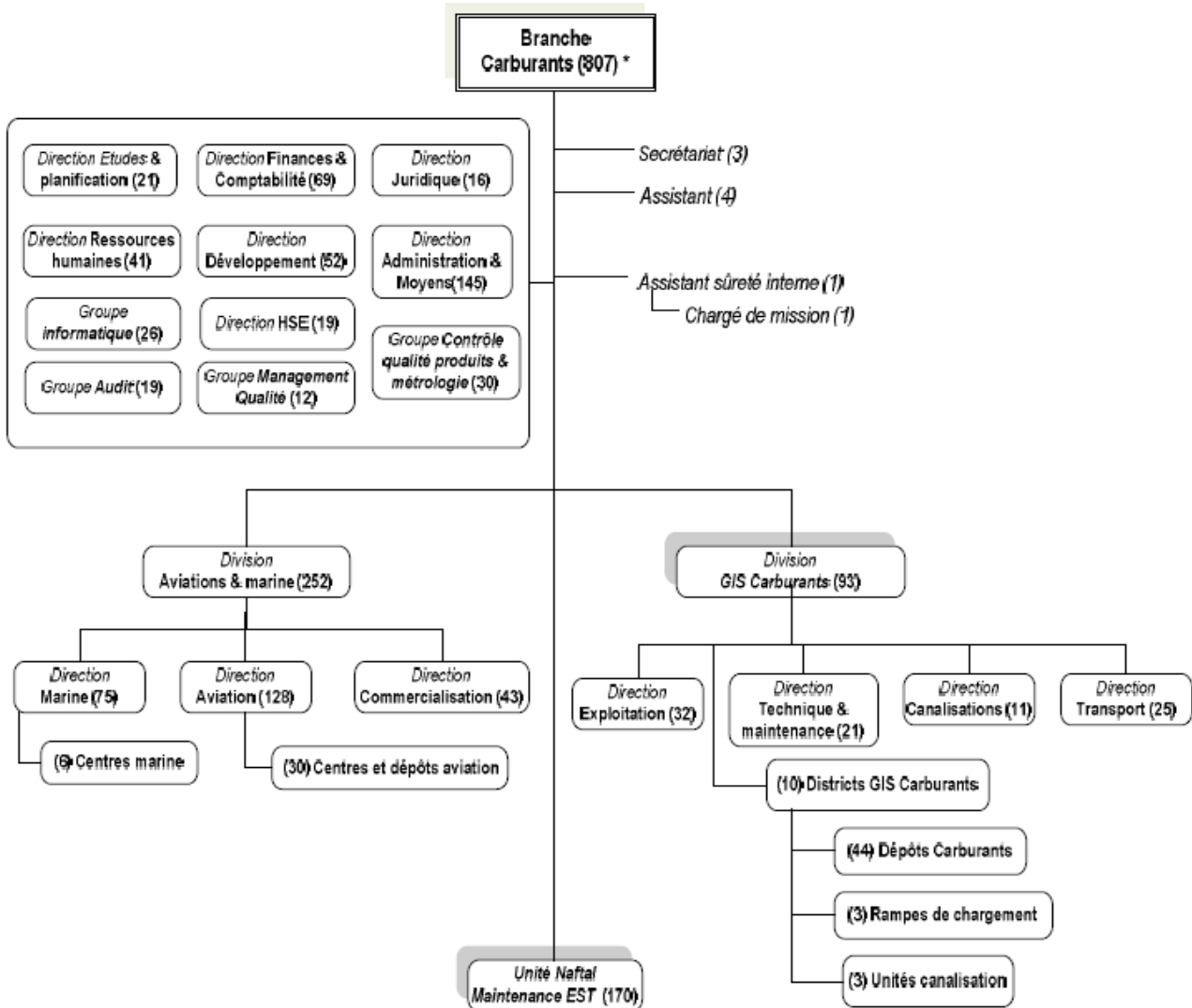


Figure 18 : Organigramme de la Branche Carburant

IV.2. MISSIONS PRINCIPALES DE LA BRANCHE CARBURANTS :

IV.2.1. Les missions des Directions Opérationnelles de la Branche Carburants dans ses activités Aviation et Marine Sont :

- ✓ Superviser, coordonner et contrôler les activités approvisionnement, stockage, ravitaillement, livraison et transport des carburants Aviation et Marine, sur les aéroports et ports où la BC/AVM est présente.
- ✓ Assurer la préservation de la conformité du produit depuis son approvisionnement jusqu'à la livraison de celui-ci et ce, conformément aux exigences du client.

- ✓ Assurer la maintenance des installations fixes et des moyens de distribution.
- ✓ Assurer une qualité de service répondant aux attentes de la clientèle et veiller au maintien de l'image de marque de l'Entreprise.

IV.2.2. Les missions des Directions opérationnelles de la Branche Carburants dans ses activités Carburants terre (CBRT) sont :

- ✓ Superviser, coordonner et contrôler les activités approvisionnement, stockage, ravitaillement et livraison des carburants terre au niveau des dépôts primaires et secondaires.
- ✓ Assurer la préservation de la conformité du produit dès son approvisionnement jusqu'à sa livraison ou son ravitaillement, conformément aux exigences des parties intéressées.
- ✓ Assurer la maintenance des installations de stockage et des moyens de distribution.
- ✓ Assurer une qualité de service répondant aux attentes de la Branche. Commercialisation et veiller au maintien de l'image de marque de l'Entreprise.

IV.3. PRODUITS COMMERCIALISES

IV.3.1. Par l'activité Aviation :

- ✓ Le Carburéacteur Jet-A1 destiné aux avions turboréacteurs.
- ✓ L'essence Avgas-100 LL, destinée aux moteurs d'avions à pistons (à hélices).
- ✓ Les Lubrifiants et graisses Aviation.
- ✓ Les Produits spéciaux (Méthmix).

IV.3.2. Par l'activité Marine :

- ✓ Les Fuel-oil Bunker C de haute viscosité (HS & LS), issus des importations et destinés aux navires de gros tonnages (moteurs lents).
- ✓ Le Fuel-oil BTS (Basse Teneur en Soufre), issu du pétrole Algérien et provenant des Raffineries Algériennes, destiné aux moteurs semi rapides.
- ✓ Les Inter Fuel-oils de différentes viscosités, issus de mélanges de Fuel-oil (Bunker C et/ou BTS) et de Gas-Oil.
- ✓ Le Gas-oil.
- ✓ Les lubrifiants et produits spéciaux Marine.

IV.3.3. Par l'activité CBRT :

- ✓ Le Gas-oil, utilisé dans les véhicules, engins et machines à moteur diesel ainsi que les besoins domestiques (éclairage, chauffage, etc.).
- ✓ Les essences utilisées dans les véhicules et engins à moteurs thermiques ou les moteurs à combustion interne à allumage commandé.

IV.4. MOYENS HUMAINS ET MATERIELS

IV.4.1. Moyens humains :

La Branche Carburants dispose d'un effectif de 7433 agents, opérant dans les activités prévues dans son organisation toutes catégories confondues.

IV.4.2. Moyens matériels :

	AVIATION	MARINE	TERRE
Capacités de stockage	55895 M3	112 000 M3	633 676M3
Moyens matériels	-Tracteurs routiers -Semi-remorques citernes -Camions avitailleurs. -Système Hydrant (livraisons par pipe via des oléo serveurs) -Oléo serveurs, -Serviteurs. -Attelages avitailleurs.	-Barges de capacité de 2000 tonnes -Barges de capacité de 1000 tonnes -Camions citernes -Bras de chargement -Camions pour la livraison des lubrifiants. -Vedettes	-Pipeline -Camions citernes.

Figure 19 : Tableau de moyen matériel

IV.4.3. Infrastructures de la branche carburante :

Activités Aviation : 28 Centres et dépôts Aviation opérationnels ;

Activités Marine : 06 Centres Marine opérationnels ;

Activités CBRT : 24 Dépôts carburants terre ;

IMPLANTATION GEOGRAPHIQUE :

Centres Aviation et Marine et Dépôts Carburants Terre :

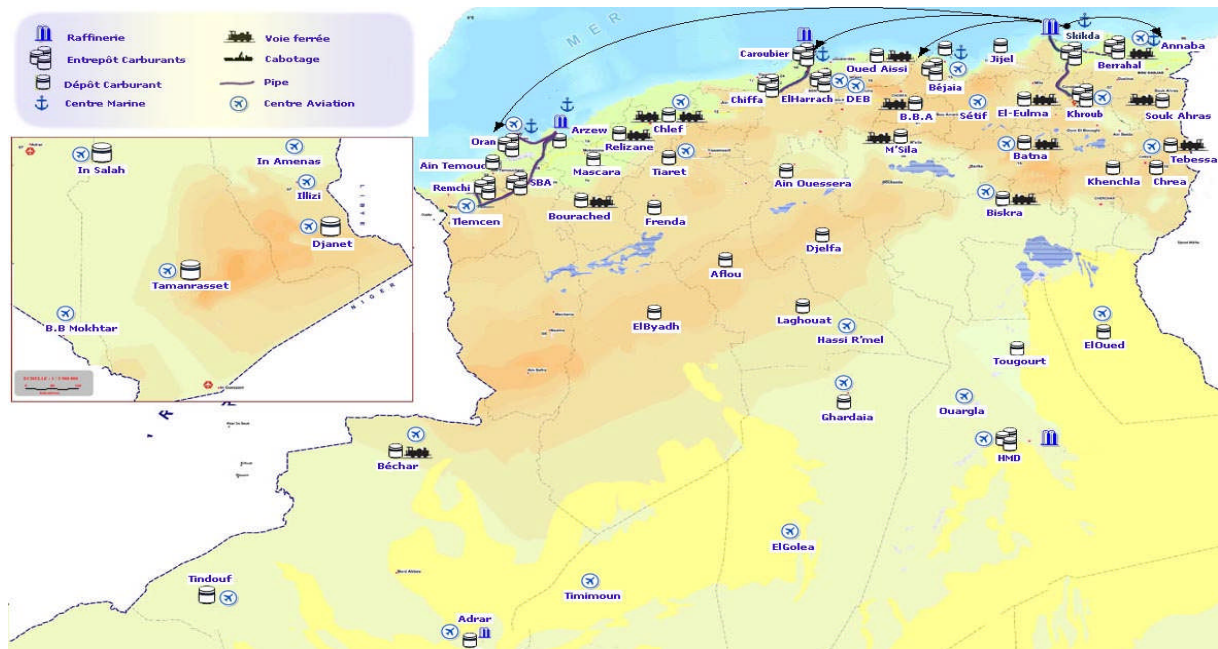


Figure 20 : Infrastructures de la branche carburante

INFRASTRUCTURE DU GROUPE INFORMATIQUE :

Le Groupe Informatique de la Branche Carburant est composé de trois départements qui sont :

- ✓ Département Développement
- ✓ Département Réseaux
- ✓ Département Maintenance

En plus de plusieurs départements ING (Informatique de Gestions) sur des sites distants. Il dispose aussi d'une infrastructure modernes dotée d'un data center virtualisée en mode IAAS (Infrastructure As a Service)

V. CIRCUIT DE DISTRIBUTION DE NAFTAL

- Un circuit de distribution est une chaine d'intermédiaires qui permettent la circulation du produit depuis sa sortie de la production jusqu'à son acquisition par le consommateur final.
- L'entreprise NAFTAL utilise un circuit de distribution composé de trois relais : les entrepôts, les dépôts, et les stations de services.

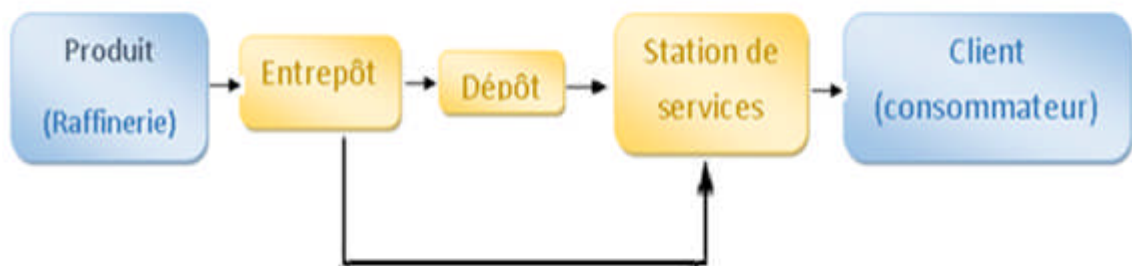


Figure 21 : Circuit de distribution de NAFTAL

V.1. L'ACTIVITE DE DISTRIBUTION

Le mot distribution s'inscrit dans la trilogie production-distribution-consommation qui correspond au cycle économique de tout produit.

L'activité de distribution se décompose en deux temps :

- Un arrêt intermédiaire (stockage).
- Un mouvement de produit (transport).

V.1.1. Le stockage :

Le stockage est l'opération de mise en réserve d'un produit ou d'un bien en vue d'une utilisation future. On stocke une quantité d'un produit qu'on en n'a pas besoin actuellement et on espère utiliser ultérieurement. On stocke aussi pour des raisons de précaution ou de crainte d'une éventuelle rupture.

Le mot stockage dans l'opération de distribution signifie autre chose.

En fait, le stockage dans ce cas devient une activité permanente, c'est-à-dire il y a toujours un flux entrant et un flux sortant. Ce qui nécessite une régulation entre les deux flux du fait que l'espace réservé pour le stock est limité. Cette régulation s'appelle la gestion du stock.

V.1.2. Le transport :

Le transport constitue le nerf du circuit de distribution des produits pétroliers, Pour assurer l'approvisionnement de tous les centres de distribution, il faudrait déployer d'énormes moyens de transport (navires, camion-citerne, wagons, gazoduc.... etc.), surtout lorsque le réseau de distribution s'étale sur une grande zone géographique. L'utilisation d'un tel arsenal de moyens de transport induirait des coûts et des charges (amortissement de l'actif, charges de maintenance, consommation du carburant, pneumatique, lubrifiant, paie du employésetc.) qu'on espère couvrir après la vente du produit. Cependant la bonne organisation en matière de transport peut procurer à l'entreprise une réduction des coûts de transport et par conséquent une augmentation des gains espérés.

MOYENS DE TRANSPORT NAFTAL :

Représentent les moyens mobiles dont NAFTAL s'en serve pour transporter les produits entre les différents centres de stockage jusqu'aux points de vente. Pour assurer cet équilibre entre l'offre et la demande, NAFTAL met à contribution plusieurs modes de transport :

- **Les Pipes :** représente **22%** du total transporté pour l'approvisionnement des entrepôts à partir des raffineries.
- **Les cabotages :** représente **14%** du total transporté.
- **Le rail :** représente **6%** du total transporté.
- **Les camions :** représente **58%** du total transporté, pour la livraison des clients et le ravitaillement des dépôts sont desservis par Rail.

VI. PROBLEMATIQUE :

La Branche Carburant dispose d'une flotte de plus de 700 véhicules (lourd et léger) circulant à travers tout le territoire national, ce qui engendre une difficulté pour la société de connaître l'emplacement de ses véhicules à tout moment et de poursuivre le déplacement à temps réel de ses véhicules. L'intégration de nouvelles technologies de l'information et de la communication semble indispensable.

Et vu le rôle majeur que cette flotte joue dans l'approvisionnement des produit carburants, NAFTAL ne dispose pas d'un outil de localisation de ses véhicules.

C'est dans ce but que nous avons développé notre application afin de mettre à la disposition de l'utilisateur une solution gratuite qui sera valable et fonctionnelle en permanence.

VII. HYPOTHESES :

Pour assurer l'approvisionnement et le ravitaillement des différents dépôts et afin d'optimiser le schéma de distribution, l'entreprise doit mettre en œuvre un system informatique qui lui permettra de suivre et analyser en temps réel ce schéma.

Le projet consiste à élaborer une solution de géo localisation qui permet l'identification, la localisation et l'analyse des chemins et distances parcourues.

La solution doit comporter une partie client (tablettes, smart phone, boitier GPRS...) et une partie web pour la consultation et le suivie de tous les mouvements de la flotte.

Une synchronisation automatique (planifier) entre les terminaux et le server doit être assuré.

De ce fait, nous proposons la conception et la réalisation d'une application de géolocalisation et de suivi de la flotte pour répondre à la problématique citée précédemment.

CONCLUSION :

Dans ce chapitre nous avons présenté la société NAFTAL et plus précisément la branche carburant ou nous avons exposé l'importance majeure de la flotte d'exploitation dans l'activité de distribution. Dans le chapitre suivant, nous exposerons le déroulement du développement de notre application.

CHAPITRE 3 :

ANALYSE ET CONCEPTION

INTRODUCTION

Dans les deux chapitres précédents, nous avons défini le domaine de notre application et les technologies qui participent à son informatisation. Ce chapitre sera consacré à l'analyse et la conception de notre application.

Pour analyser les besoins du système, nous nous sommes posé cette question :

- Quels sont les objectifs afin de savoir quelles seront les informations Véhiculées

Pour modéliser le système, on a opté pour l'UML qui est un langage unifié de modélisation.

I. ANALYSE

I.1. SPECIFICATION DES BESOINS

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

I.1.1. Spécification des besoins fonctionnels

Après une étude détaillée de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation.

L'application offre à ses utilisateurs finaux les fonctionnalités suivantes :

- Récupération de la location des véhicules automatiquement sur l'appareil mobile.
- Authentification des Cadres de NAFTAL sur l'interface web.
- Afficher aux cadres la localisation des véhicules sur l'interface.
- Consultation des véhicules et appareils.
- Afficher l'historique des parcours sur les appareils mobiles.

I.1.2. Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé le besoins suivants :

- **La disponibilité** : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.

- *La sécurité de l'accès aux informations critiques* : nous devons prendre en considération la confidentialité des données surtout au niveau de l'authentification. Pour cela nous devons restreindre l'accès à ces informations à l'administrateur.
- *La fiabilité* : les données fournies par l'application doivent être fiables.
- *La convivialité de l'interface graphique* : le système doit fournir des interfaces conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec le système et par le biais de celle-ci on découvrira ses fonctionnalités.
- *Une solution ouverte et évoluée* : la solution peut être améliorée par l'ajout d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.
- La possibilité de retourner au menu principal de l'application à partir de n'importe quelle fenêtre de celle-ci.

I.2. METHODOLOGIE ET APPROCHE ADOPTEE

Avant de programmer l'application et se lancer dans l'écriture du code : il faut tout d'abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. Cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit est un module.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Dans le cadre de notre projet on a utilisé la méthodologie UML pour la modélisation des différents diagrammes.

I.2.1. Présentation de l'UML :

En regardant les objectifs fixés pour la réalisation du projet, nous remarquons que nous sommes face à une application modulaire et qui devra rester ouverte pour les améliorations futures. De ce fait, il est très important d'utiliser un langage universel pour la modélisation afin de clarifier la conception et de faciliter les échanges. Notre choix est porté sur le langage UML puisqu'il convient pour toutes les méthodes objet et se prête bien à la représentation de l'architecture du système.

I.2.2. Définition :

UML : UML (Unified Modeling Language) est un langage de modélisation unifié permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orienté objet. UML permet de couvrir le cycle de vie d'un logiciel depuis la spécification des besoins jusqu'au codage en offrant plusieurs moyens de description et de modélisation des acteurs et

l'utilisation système, du comportement des objets, du flot de contrôle internes aux opérations, des composants d'implémentation et leurs relations, de la structure matérielle et de la distribution des objets et des composants indépendamment des techniques d'implémentation et peut être mis à jour selon les besoins.

I.2.3. Modélisation avec L'UML :

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

- *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.

I.2.4. Les avantage d'UML :

- Universel.
- Adopté par les grandes entreprises.
- Notation unifiée
- Facile à comprendre.
- Adopté par plusieurs processus de développement
- Limite les risques d'erreur.
- N'est pas limité au domaine informatique.

I.3. IDENTIFICATION DES ACTEURS

I.3.1. Définition d'un acteur

Un acteur représente un rôle joué par une entité externe (utilisateur, dispositif matériel ou autre système) qui interagit directement avec le système étudié, il peut consulter et/ou modifier directement l'état du système, en émettant et/ou recevant des messages susceptible d'être porteur des données.

I.3.2. Les acteurs de notre système

Notre système comprend trois acteurs :

- **Conducteur** : c'est tous chauffeur appartenant à notre société
- **L'opérateur** : s'occupe du suivie et analyse les déplacements des différents véhicules
- **L'administrateur** : (espace privé), il s'occupe de la gestion des comptes pour les opérateurs.

I.3.3. Spécification des rôles :

L'acteur défini précédemment effectue un certain nombre des rôles, ces rôles sont résumés dans le tableau ci-dessous :

Acteurs	Rôle
Conducteur	<ul style="list-style-type: none"> ▪ Afficher l'historiques des parcoures
Administrateur	<ul style="list-style-type: none"> ▪ S'authentifier. ▪ Gérer « Ajout-Modification-suppression » des comptes operateur. ▪ Afficher la localisation des véhicules. ▪ Modifier le mot de passe
Opérateur	<ul style="list-style-type: none"> ▪ S'authentifier. ▪ Gérer « Ajout-Modification-suppression » des véhicules. ▪ Gérer « Ajout-Modification-suppression » des appareils. ▪ Afficher la localisation des véhicules. ▪ Correspondre chaque appareil a son véhicule. ▪ Modifier le mot de passe

Figure 22 : Tableau des acteurs de notre système.

II.4. LES DIAGRAMMES DE CAS D'UTILISATION

II.4.1. Définition :

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces de future application. C'est le premier diagramme UML constitué d'un ensemble d'acteurs qui

agit sur des cas d'utilisation et qui décrit sous la forme d'actions et des réactions, le comportement d'un système du point de vue utilisateur.

- **Acteur** : un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne, système ou une entreprise.
- **Système** : cet élément fixe les limites du système en relation avec les acteurs qui l'utilisent (en dehors de système) et les fonctions qu'il doit fournir (à l'intérieur du système).
- **Cas d'utilisation** : un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système.

II.4.2. Description des cas d'utilisations :

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

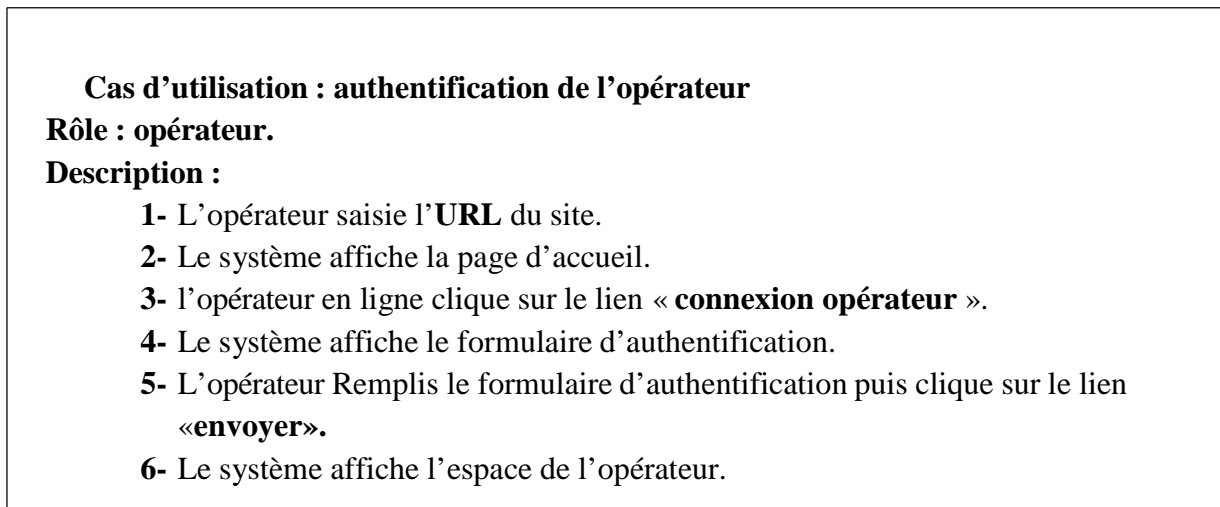


Figure 23 : Cas d'utilisation « s'authentifier ».

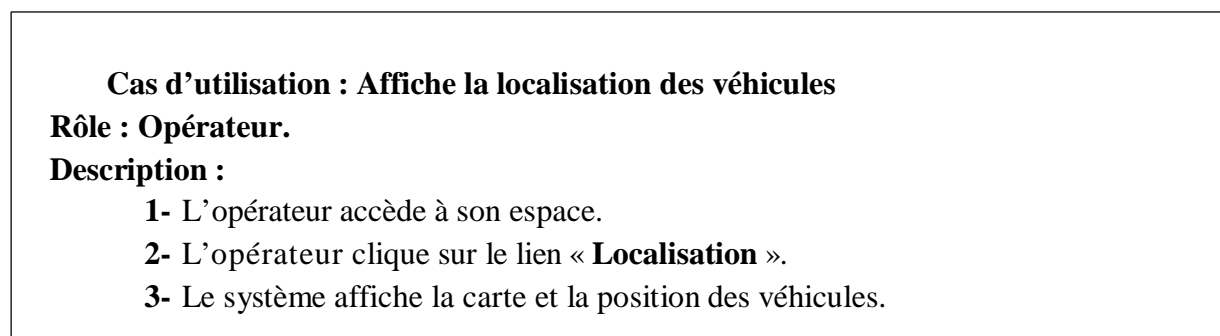


Figure 24 : Cas d'utilisation « Afficher la localisation des véhicules ».

Cas d'utilisation : Ajouter un véhicule**Rôle : Opérateur.****Description :**

- 1- L'opérateur accède à son espace.
- 2- L'opérateur clique sur le lien « **Ajouter véhicule** ».
- 3- Le système affiche le formulaire d'ajout des véhicules.
- 4- L'opérateur Remplit le formulaire puis clique sur le lien « **valider** ».
- 5- Le système affiche la page de confirmation.

Figure 25 : Cas d'utilisation « Ajouter un véhicule ».

Cas d'utilisation : Afficher la liste des appareils**Rôle : opérateur.****Description :**

- 1- L'opérateur accède à son espace.
- 2- L'opérateur clique sur le lien « **Liste appareil** ».
- 3- Le système affiche l'état descriptif des appareils.

Figure 26 : Cas d'utilisation « Afficher la liste des appareils ».

Cas d'utilisation : Afficher les parcoures**Rôle : Conducteur.****Description :**

- 1- Conducteur accède à l'application.
- 2- Le système affiche l'interface de l'application.
- 3- Conducteur clique sur bouton « **afficher l'historique des parcoures** ».
- 4- Le système affiche le formulaire de recherche.
- 5- Le conducteur Remplit le formulaire puis clique sur le lien « **envoyer** ».
- 6- Le système affiche l'historique

Figure 27 : Cas d'utilisation « afficher les parcoures ».

II.4.3. Les diagrammes de cas d'utilisations pour chaque acteur :

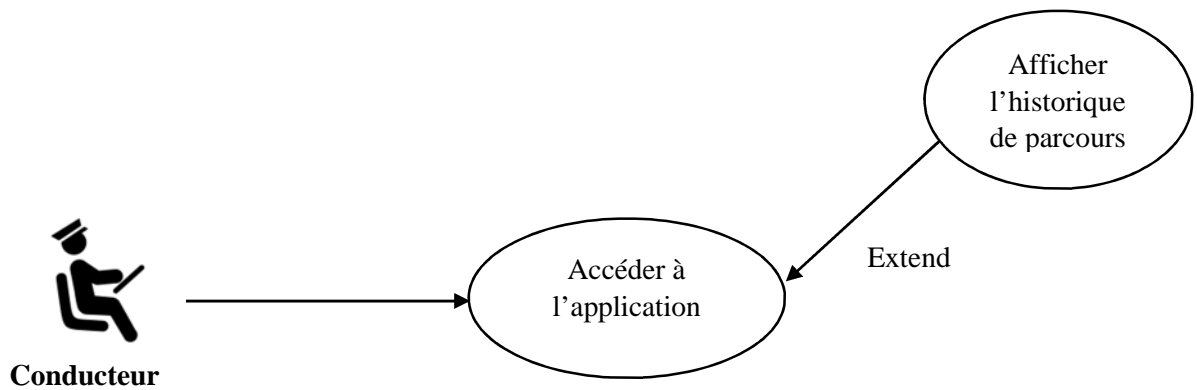


Figure 28 : Diagramme de cas d'utilisation « Acteur : Conducteur ».

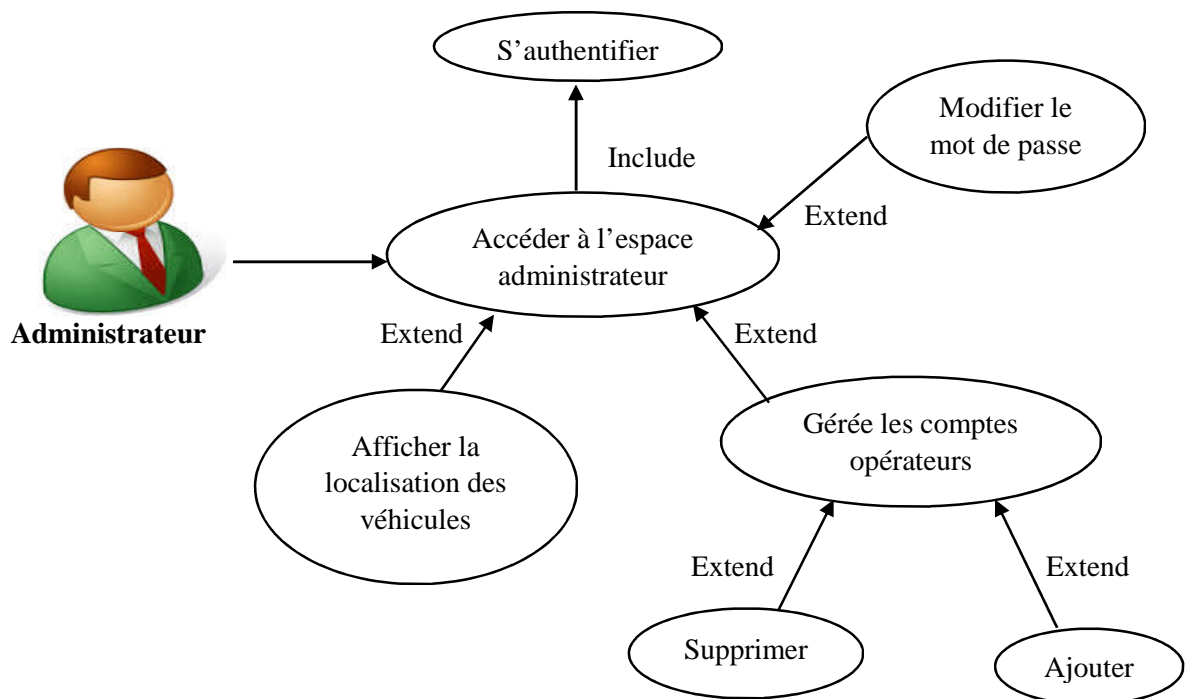


Figure 29 : Diagramme de cas d'utilisation « Acteur : Administrateur ».

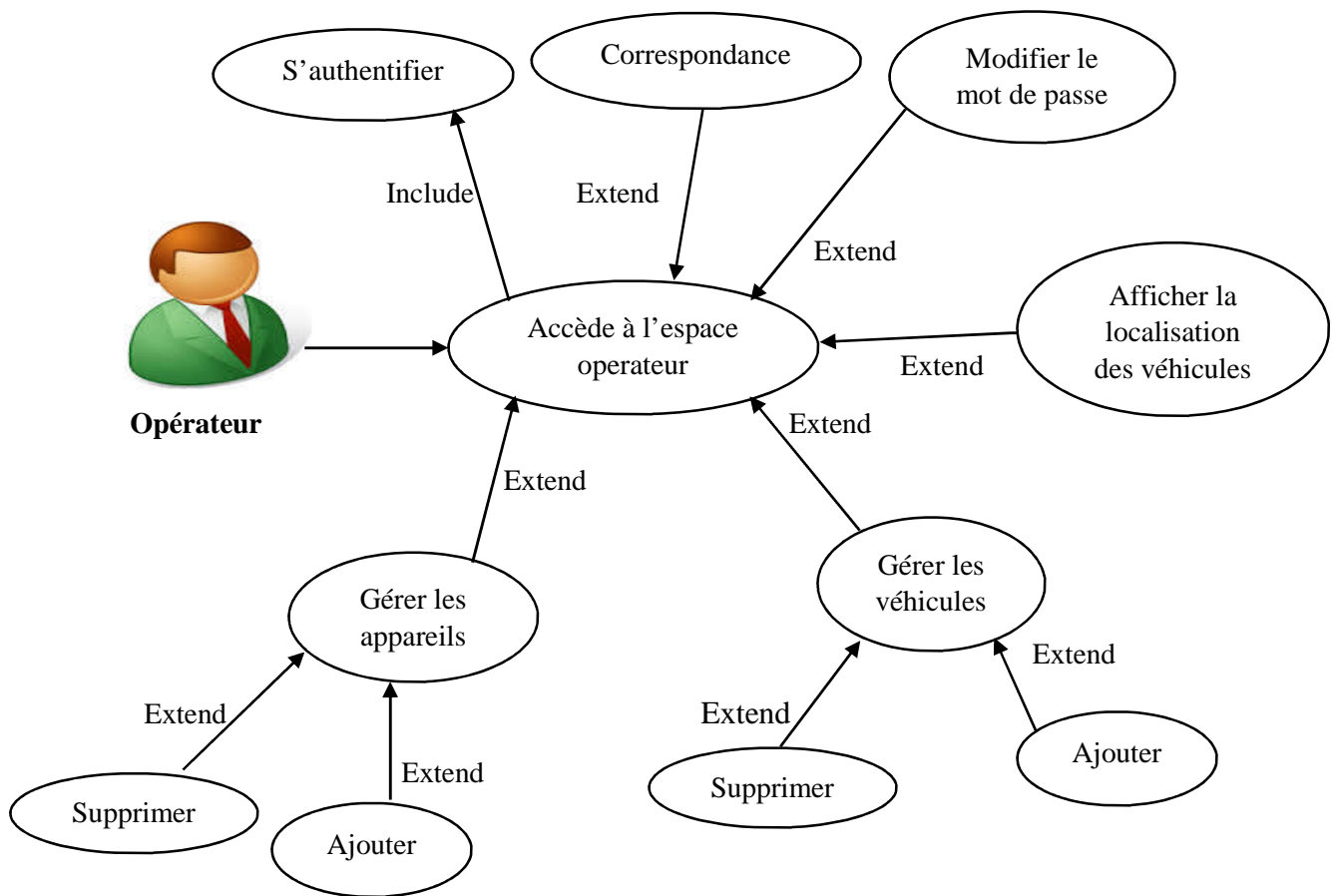


Figure 30 : Diagramme de cas d'utilisation : « Acteur : opérateur ».

II. CONCEPTION

Le processus de conception de notre projet se caractérise par deux niveaux : le niveau applicatif et le niveau donné. Le niveau applicatif s'appuie essentiellement sur quelques diagrammes 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 diagramme d'activité.
- Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

Le niveau donné 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.

II.1. Le diagramme de déploiement

Le diagramme de déploiement définit l'architecture matérielle de l'application. Il présente les périphériques utilisés et la répartition du système sur ces différents éléments. Il montre aussi les liens de communication entre ces diverses entités.

Le diagramme de déploiement de notre application est représenté par le diagramme ci-après :

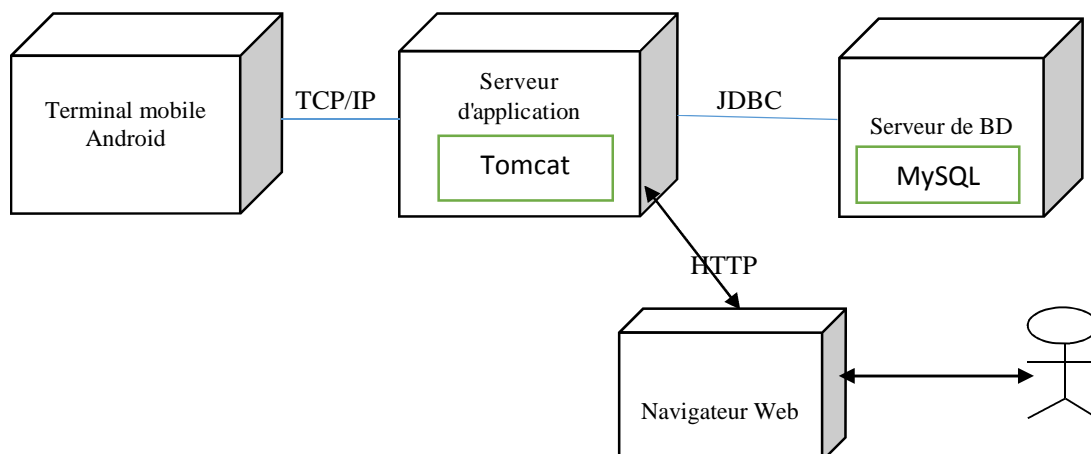


Figure 31 : diagramme de déploiement


II.2. LES DIAGRAMMES DE SEQUENCES :

Avec les diagrammes de séquences, l'UML fournit un moyen graphique pour représenter les interactions entre objets à travers le temps. Ces diagrammes montrent typiquement un utilisateur ou un acteur et les objets et composants avec lesquels ils interagissent au cours de l'exécution du cas d'utilisation.

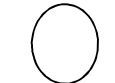
Un diagramme de séquence représente en général un seul 'scénario' de cas d'utilisation ou flux d'événements.

Les classes d'objets utilisées dans la représentation du diagramme de séquence peuvent être réparties dans les trois catégories suivantes :


Les objets d'interface : ils représentent l'interface entre l'acteur et le système. Des exemples classiques en sont les écrans de saisie et de contrôles spéciaux d'interface utilisateur, ou dans l'application Web, des pages Web complètes

L'icone : 

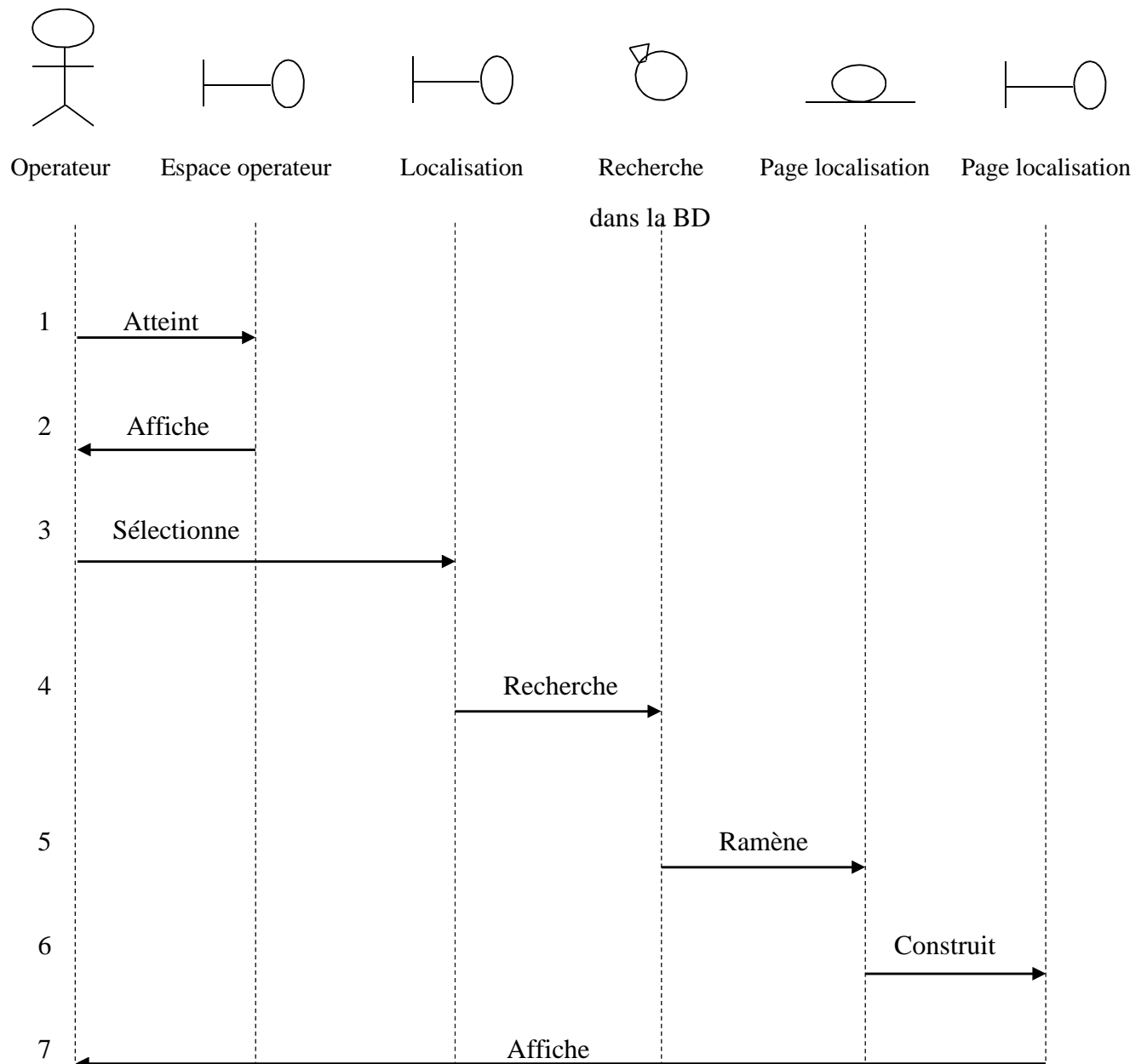
Les objets entité : ce sont des objets décrits dans le cas d'utilisation. Les commandes les produits sont des objets entités dont les instances peuvent apparaître dans de nombreux cas d'utilisation.

L'icone : 

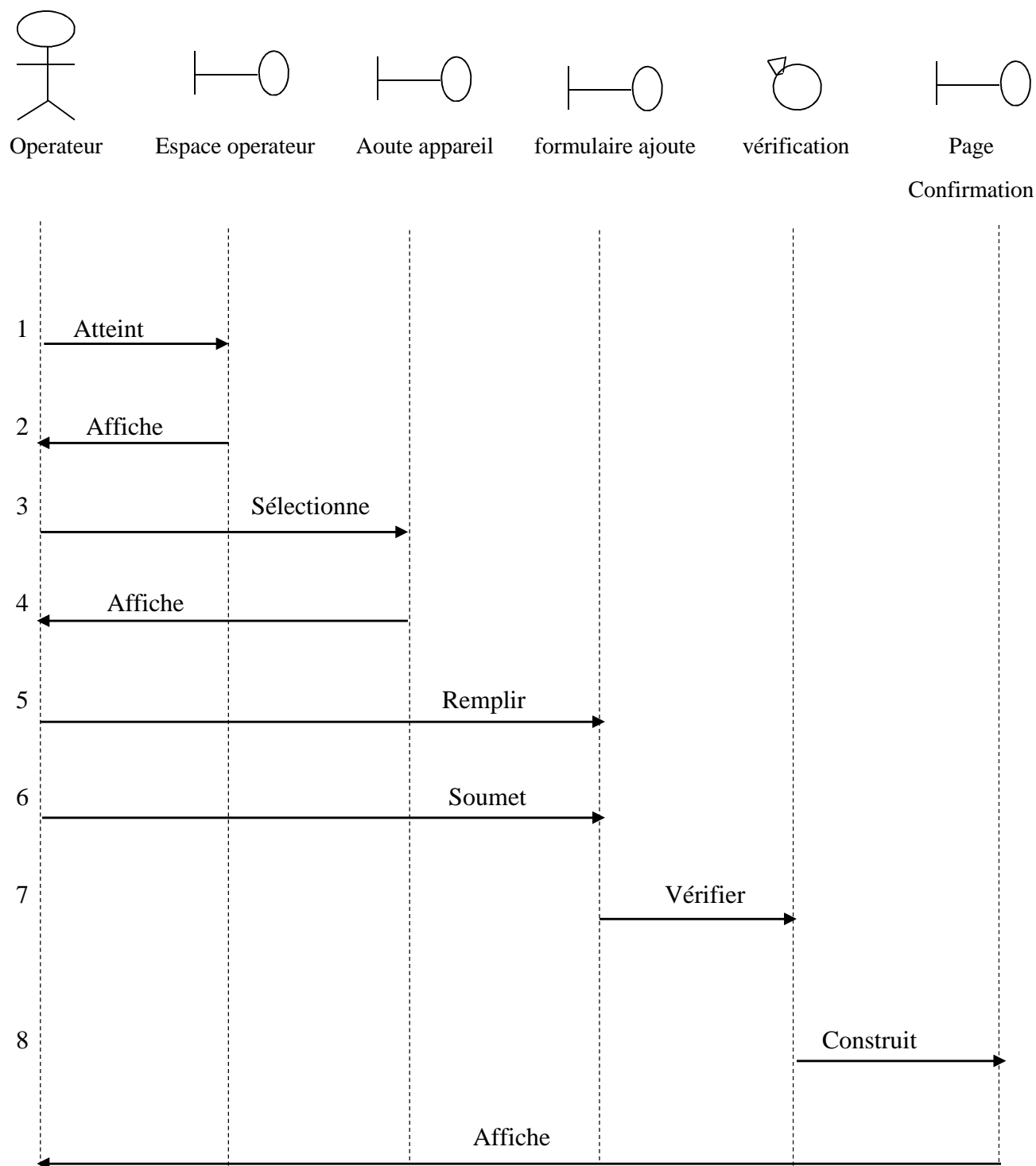
Les objets contrôle : ils représentent les processus, c'est -à-dire les activités système assez significatives pour être nommées. Les objets contrôle dirigent les activités des objets entités et d'interface.

L'icone : 

Dans cette partie, nous allons décrire les scénarios les plus importants ainsi que leurs représentations par les diagrammes de séquence.



**Figure 32 : Diagramme de séquence de cas d'utilisation
« afficher la localisation des véhicules ».**



**Figure 33 : Diagramme de séquence de cas d'utilisation
« Ajoute les appareils ».**

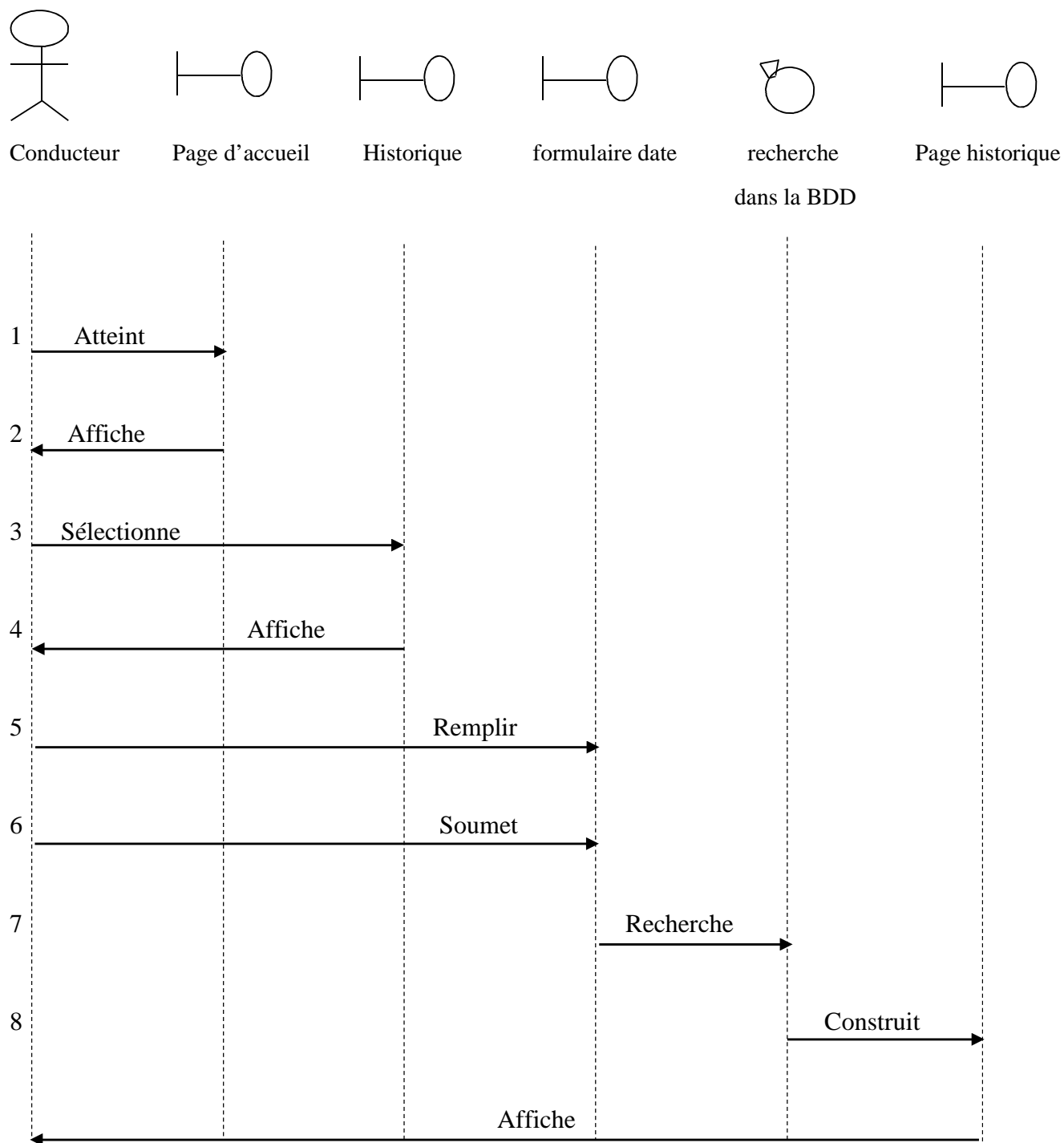


Figure 34 : Diagramme de séquence de cas d'utilisation « afficher l'historique des parcours ».

II.3. LES DIAGRAMMES D'ACTIVITES :

Le diagramme d'activité permet de représenter le déclenchement d'évènements en fonction des états du système et de modéliser des comportements parallélisés. Il donne une vision des activités propres à une opération ou à un cas d'utilisation.

Une activité est une opération d'une certaine durée qui peut être interrompue.

Dans ce cas, on va représenter ci-après des diagrammes d'activités qui décrivent :
L'authentification d'un utilisateur

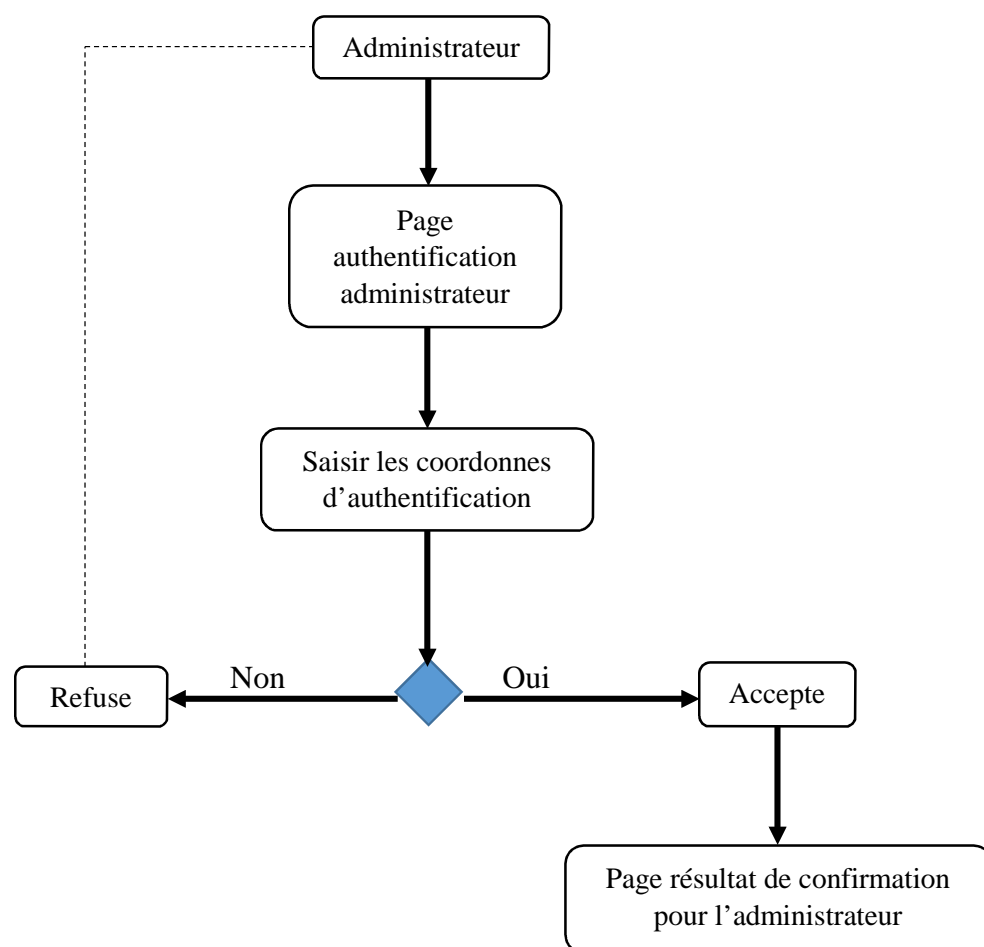


Figure 35 : Diagramme d'activité pour Le cas d'utilisation « s'authentifier ».

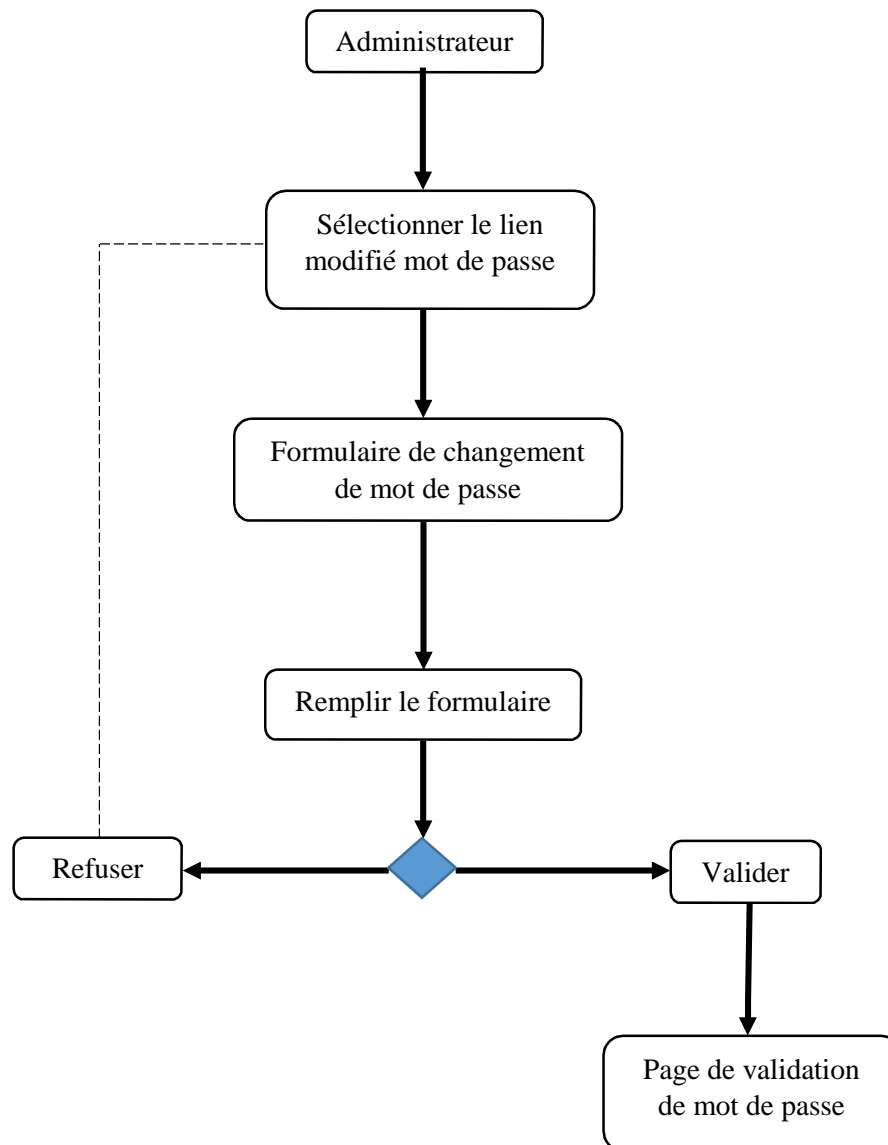


Figure 36 : Diagramme d'activité pour Le cas d'utilisation « modification mot de passe »

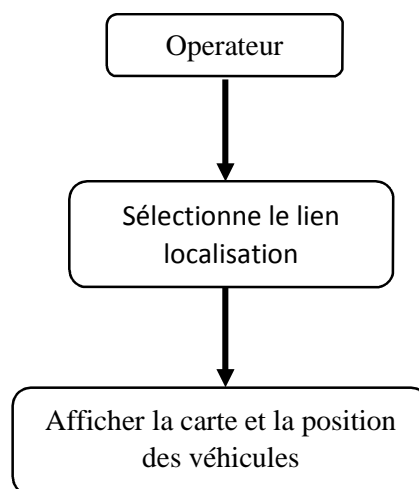


Figure 37 : Diagramme d'activité pour Le cas d'utilisation « Afficher la localisation des véhicules ».

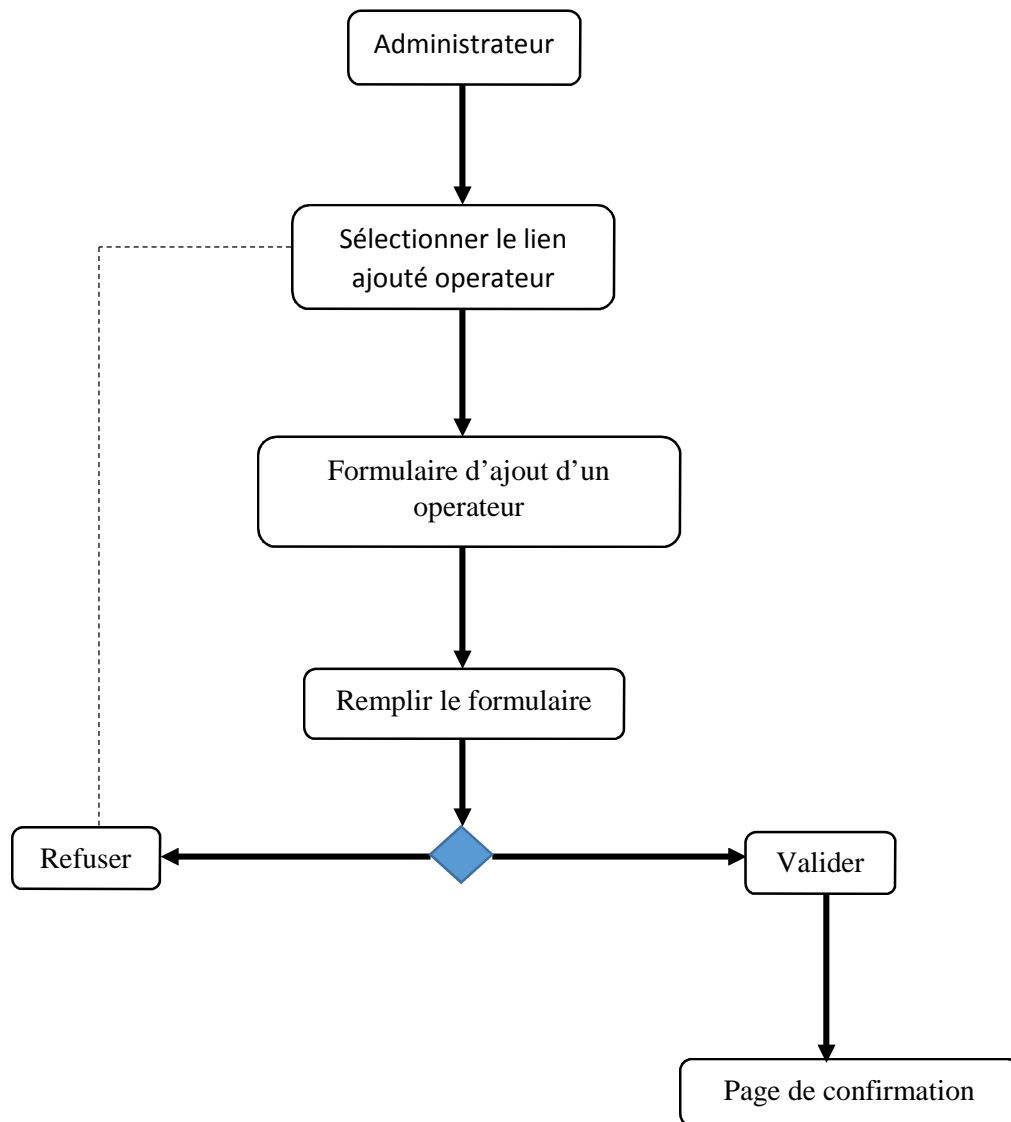


Figure 38 : Diagramme d'activité pour Le cas d'utilisation « gérer les compte operateur ».

II.4. LES DIAGRAMMES DE CLASSE :

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 digramme de classe est considéré comme le pivot 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 digramme 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 digrammes d'activités pour les cas d'utilisation déjà décrit, nous allons élaborer des diagrammes de classes détaillés pour chacun d'eux.

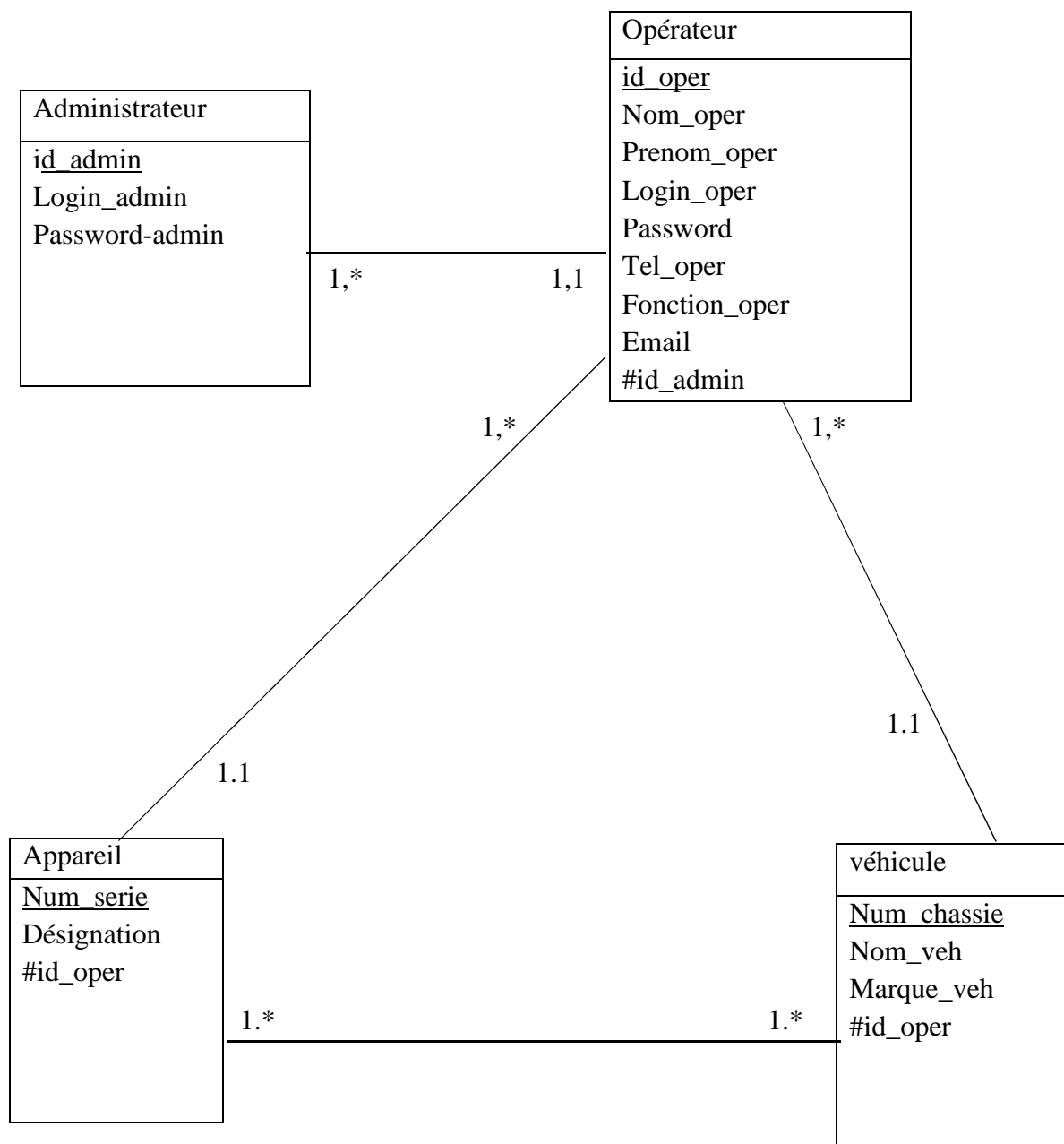


Figure 39 : Diagramme de classes de la BDD locales

II.5. MODALE RELATIONNEL :

ADMINISTRATEUR (Id_admin, Login_admin, Password_admin)

OPERATEUR (Id_oper, Nom_oper, Prenom_oper, Fonction_oper, Tel_oper, Email, Login_oper, Password_oper, Id_admin) où id_admin est une clé étrangère qui fait référence à la table administrateur.

VEHICULE (Num_chassie, Nom_vehicule, Marq_veh, Id_oper) où id_oper est une clé étrangère qui fait référence à la table operateur.

APPAREIL (Num_serie, Désignation, Id_oper) où id_oper est une clé étrangère qui fait référence à la table operateur.

LOCALISATION (Id_loc, lat, lng, Date_L, Num_chassie, Num_serie) où Num_chassie, et Num_serie sont des clés étrangères qui font respectivement référence aux schémas de relation véhicule et appareil.

CORRESPONDANCE (Id_C, Date_C, Num_chassie, Num_serie, id_loc) où id_loc est une clé étrangère qui fait référence à la table localisation.

Conclusion :

Dans ce chapitre, nous avons introduit les fonctionnalités de notre application. Pour les atteindre, nous avons proposé une solution fondée sur une analyse et une conception modélisées à l'aide du langage de modélisation unifié UML. Pour cela, nous avons défini les acteurs de notre application, les tâches qu'ils assurent et les rôles associés à chaque tâche. Nous avons élaboré les diagrammes de cas d'utilisation de chaque acteur, élaboré les diagrammes de séquence, de classes de quelque cas d'utilisation. En fin, nous avons représenté les tables de la base de données de notre application.

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

CHAPITRE 4 : REALISATION

Introduction

Après avoir explicité dans le chapitre précédent la conception de notre système, nous allons présenter dans ce chapitre son implémentation. Cette phase finale consiste à traduire la conception exprimée à l'aide d'un formalisme en un code source écrit dans un langage donné. Nous allons tout d'abord commencer par la présentation de l'environnement et les outils de développements et d'implémentation de notre application, puis les différentes interfaces, et leurs fonctionnalités essentielles.

I. Environnement de travail

I.1. L'ARCHITECTURE DE L'APPLICATION

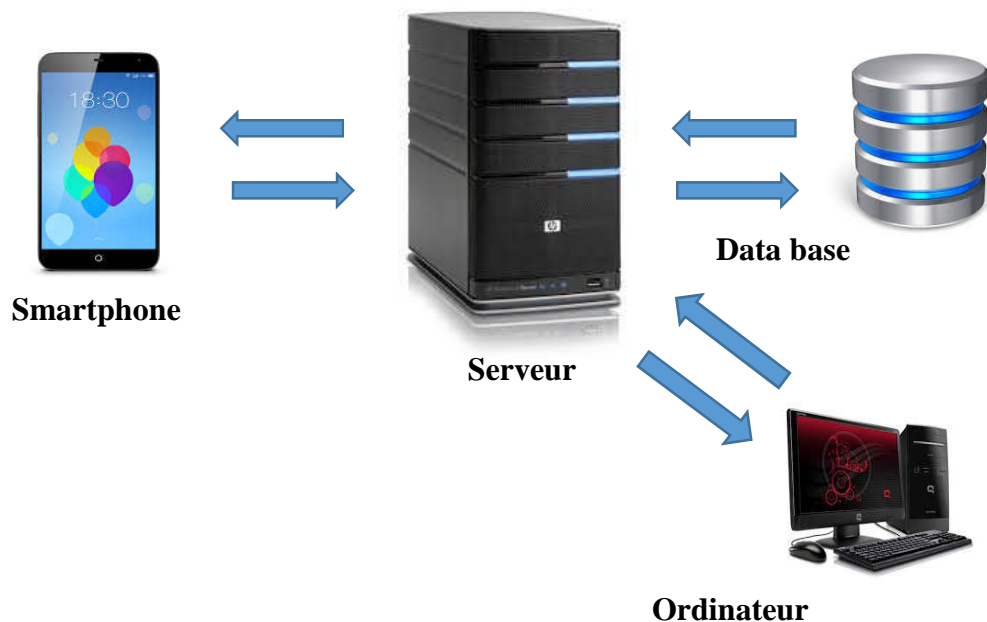


Figure 40 : Architecture de l'application.

Dans cette architecture, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/ serveur de base de données par exemple). Il permet :

- Une plus grande flexibilité/souplesse ;
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau ;
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

I.2. ENVIRONNEMENT MATERIEL

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

- PC Portable Toshiba : Intel(R) Core(TM) i5, 2.30 GHz, 6 GO de mémoire vive, système exploitation 64 bits Windows 8.1 pro.
- Disque dur externe : capacité 160 GO.
- Sur le pc portable on a installé une machine virtuelle Workstation Pro 12, Puis on a installé un système Windows 7(64 Bit) sur le disque dur externe.
- Smartphone Lenovo vibe x S960 doté d'un système Android (Version 5), sur lequel on a fait des tests.

I.3. ENVIRONNEMENT LOGICIEL

Dans cette section nous allons décrire l'environnement utilisé pour le développement de notre système embarqué composé d'un sous-système Android connectée à un server d'application, et une interface Web connecté au même server.

Pour pouvoir réaliser une application dans de bonnes conditions il faut en tout premier lieu bien choisir son environnement de travail.

Concernant notre Système, Nous avons opté pour la réalisation du sous-système Android de travailler sous Eclipse avec le langage JAVA et le SDK Android pour son développement, afin d'obtenir un fichier .APK qui sera par la suite installé sur les terminaux mobiles de types Smartphones fonctionnant sous système Android version 4.0.3 et plus.

Et pour la réalisation de l'interface web d'utiliser XAMPP et Dreamweaver.

Pour préparer l'environnement logiciels, nous avons effectué différentes tâches que nous décrivons dans ce qui suit :

- Installation d'éclipse (comme pour le développement d'applications java classique).
- Installation SDK (software development Kit) Android qui va contenir tous les « outils » nous permettant de « construire » des applications sous Android.
- Le plugin « Android pour Eclipse » ADT (Android Development Tools), qui adapte Eclipse au développement d'applications sous Android.
- Installation de XAMPP
- Installation de DREAMWEAVER

I.3.1. IDE 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.



I.3.2. Le plugin ADT

Pour développer sous Android, on a installé le plugin ADT qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android.



I.3.3. Le SDK

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

I.3.4. XAMPP

XAMPP est une distribution d'Apache populaire dans le milieu de la programmation PHP. Sa facilité d'installation et d'utilisation ainsi que sa gratuité permet à n'importe quel utilisateur de se mettre à la programmation PHP. Enfin, vous pourrez conserver votre configuration sur un support de stockage amovible pour une installation sur une autre station de travail.



❖ Pourquoi utiliser XAMPP ?

La plus part des utilisateurs sont habitués à avoir des interactions avec leurs sites préférés. Ceci requiert un site dynamique et l'html seul ne suffit pas. XAMPP contient la plus part des outils nécessaires pour reproduire le comportement d'un serveur web, à savoir :

- **Apache** : il s'agit du serveur web, c'est lui qui va réceptionner les requêtes HTTP et les étudier afin de présenter au visiteur la page demandée.
- **MySQL** : il s'agit du Système de Gestion de Bases de Données (le SGBD), il va permettre de sauvegarder les données de manière organisée sur le serveur.
- **PHP** : ce module d'Apache va lui permettre d'interpréter les pages PHP.



- **PhpMyAdmin** : phpmyadmin est une interface entre vous et vos données, il est fait pour simplifier l'administration de MySQL grâce à des pages web.

I.3.5. Adobe Dreamweaver

Adobe Dreamweaver CS5 est un outil convivial et très puissant destiné à la conception, au codage et au développement de sites, de pages et d'applications Web (éditeur HTML professionnel). Quel que soit l'environnement de travail utilisé (codage manuel HTML ou environnement d'édition visuel), Dreamweaver propose des outils qui aident à créer des applications Web.

Les fonctions d'édition visuelles de Dreamweaver permettent de créer rapidement des pages sans rédiger une seule ligne de code. On peut rationaliser les tâches de développement en créant et en modifiant des images dans Adobe Fireworks ou toute autre application graphique, puis en les important directement dans Dreamweaver, ou en ajoutant des objets Adobe Flash.

Dreamweaver propose également un environnement de codage complet comprenant des outils de modification du code (comme la coloration des codes et la création de balises) ainsi que des documents de référence sur les feuilles de style en cascade (CSS- Cascading Style Sheets), java scripts et Dreamweaver permettent d'importer des documents HTML codés manuellement sans en modifier le code pour pouvoir ensuite reformater ce dernier avec le style de formatage choisi par le programme. Dreamweaver permet aussi de créer des applications Web reposant sur des bases de données dynamique au moyen de technologies serveur comme ASP, JSP et PHP.



Figure 41 : L'interface principale de Dreamweaver.

II. Langages de programmation :

II.1. Java : Java est un langage de programmation orienté objet, développé par Sun Microsystems et destiné à fonctionner dans une machine virtuelle, il permet de créer des logiciels compatibles avec des nombreux systèmes d'exploitation.



Java est non seulement un langage de programmation puissant conçu pour être sûr, inter plateformes et international, mais aussi un environnement de développement qui est continuellement étendu pour fournir des nouvelles caractéristiques et des bibliothèques permettant de gérer de manière élégante des problèmes traditionnellement complexes dans les langages de programmation classiques, tels que le multithreading, les accès aux bases des données, la programmation réseau, l'informatique répartie.

En plus java est considéré comme un langage adaptable aux plusieurs domaines puisque une application web implémentée par celle-ci peut avoir des extensions ou des modifications dans le futur. De plus, java permet de réduire le temps de développement d'une application grâce à la réutilisation du code développé.

II.2. Html : L'html est un ensemble des règles qui indique à un navigateur comment afficher une page web. il est souvent utiliser avec des langages de programmation telle que JavaScript et PHP.



II.3. PHP : est un langage utilisé dans des applications web pour écrire des scripts html le but premier de ce langage est de permettre aux développeurs web de concevoir rapidement des sites aux pages dynamiques .c'est un langage exécuté cote serveur, à la différence de JavaScript et html, donc le code de l'utilisateur n'apparaît pas dans la source de sa page web, seul les informations qu'il souhaite afficher seront visible par les visiteurs.



III. FONCTIONNEMENT DE NOTRE APPLICATION :

III.1. Cote application mobile :

L'application est lancée à l'allumage de notre appareil mobile, au lancement l'application commence à récupérer la localisation automatiquement et enregistre dans la BDD locale qui est de type SQL lite. Puis les coordonnées sont envoyées à la BDD centrale qui reçoit les données de plusieurs BDD locales.

On présente les écrans de l'application développée ainsi que leurs fonctionnalités sous forme de zones de texte présentant les actions réalisées via cet élément.

III.1.1. L'écran de l'application mobile

Conducteur accède à l'application mobile. Dans cette interface d'accueil

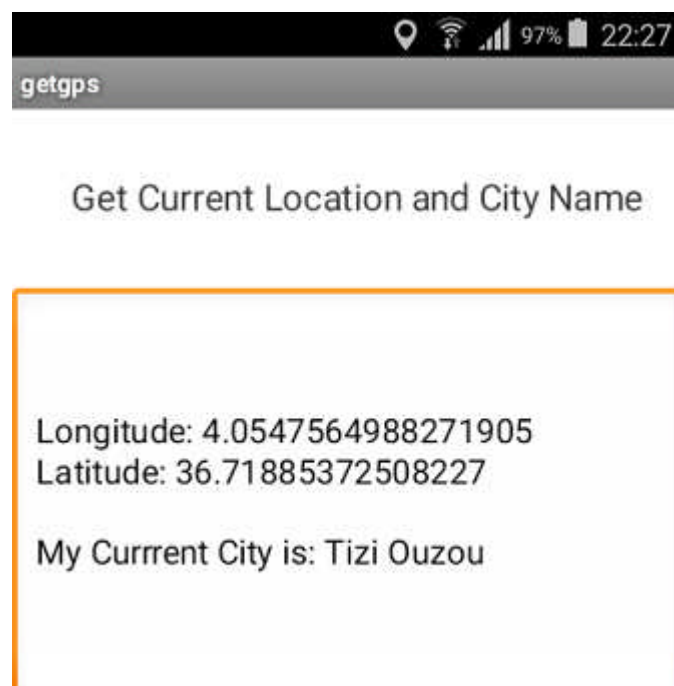


Figure 42 : Interface de l'application mobile.

III.2. Cote application web:

III.2.1. La page de l'espace administrateur

L'administrateur pour accéder à son espace, doit s'authentifier. Comme toute application, la sécurité d'accès est nécessaire. L'administrateur s'identifie. Il saisit son login et son mot de passe puis le serveur vérifie ces informations.

En cas d'erreur, un message d'erreur sera affiché dans le cas où les informations écrites ne sont pas présentes dans notre base de données.

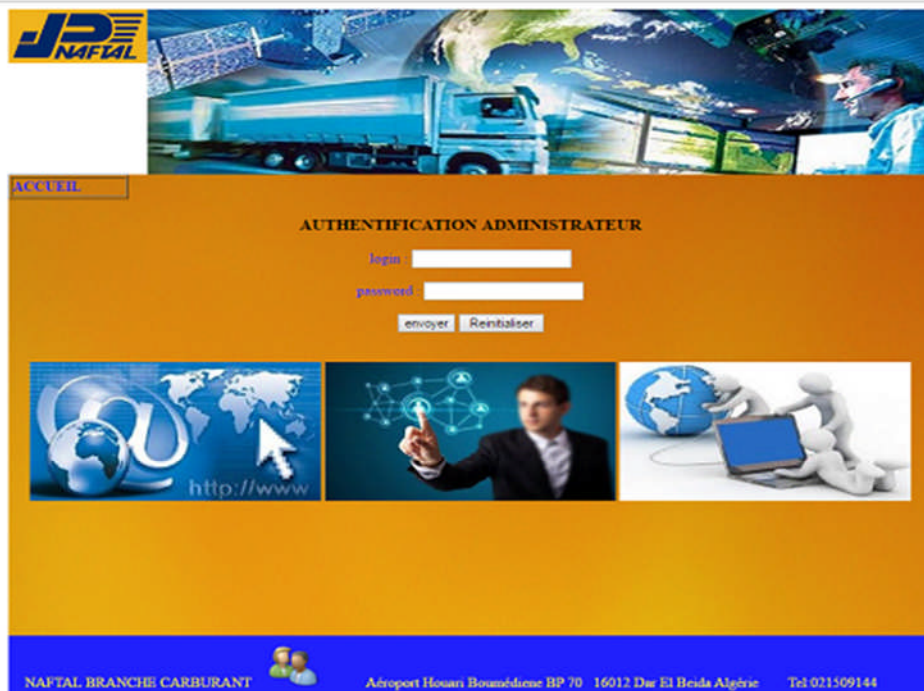


Figure 43 : Interface de la page de l'espace administrateur.

Une fois les données sont valides, l'administrateur accède à son espace. Dans cette interface d'accueil l'administrateur peut gérer toutes les fonctionnalités de l'application web.

- ✓ Ajouter un opérateur.
- ✓ Afficher la liste des opérateurs.
- ✓ Afficher la localisation des véhicules.
- ✓ Modification son mot de passe



Figure 44 : Interface de la page d'accueil de l'espace administrateur.

III.2.2. La page de l'espace opérateur

L'opérateur doit d'abord saisir son login et son password pour qu'il puisse accéder à son espace. Un message d'erreur est affiché si les coordonnées sont erronées.

Une fois les données sont valides, l'opérateur accède à son espace. Dans cette interface d'accueil l'opérateur peut gérer toutes les fonctionnalités de l'application web.

- ✓ Ajouter des véhicules
- ✓ Afficher la liste des véhicules
- ✓ Ajouter les appareils
- ✓ Afficher la liste des appareils
- ✓ Correspondre chaque appareil a son véhicule
- ✓ Afficher la localisation des véhicules
- ✓ Modification son mot de passe



Figure 45 : Interface de la page de l'espace operateur.

III.2.3. Afficher la localisation des véhicules :

Pour afficher la localisation des véhicules, l'opérateur sur son espace clique sur le lien « localisation »



Figure 46 : Interface de la page localisation des véhicules.

Conclusion :

Dans ce chapitre nous avons fait une description de notre application, ou nous avons cité l'environnement de développement et le différent outil de programmation qui nous a permet d'implémenter les différentes pages de notre application mobile et web, ensuite nous avons présenté quelques interfaces de notre application.

CONCLUSION

GENERALE

Conclusion générale

L'élaboration de notre travail était dans le but de concevoir une solution de géolocalisation pour la société NAFTAL, cette solution permet aux responsables de connaître à tout moment le positionnement de ses véhicules.

Notre travail ainsi réalisé permet de :

- Récupérer automatiquement les coordonnées GPS, le numéro de série de l'appareil plus la date et l'heure.
- Enregistrement de ces derniers sur la base de données mobiles.
- Envoie tous les données dans la base de données centrale.
- Afficher la localisation de tous les véhicules par les responsables de NAFTAL.

Ce projet s'est révélé profitable sur plusieurs points : il a été testé sur un appareil mobile et aussi sur un émulateur, il nous a permis de travailler sur une technologie pour terminaux mobiles et accroître nos connaissances dans le domaine de l'embarqué.

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

Comme d'autres solutions, notre projet peut être aisément amélioré. En effet, grâce à son aspect ouvert, Android offre l'opportunité de créer des logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à puiser dans leur imagination et à mobiliser toutes leurs compétences pour un meilleur de cette plateforme.

Les améliorations futures qui seront fait sont :

- Authentification des cadres de Nafta à l'interface web avec sa propre session Windows
- Amélioration des interfaces de localisation en ajoutant d'autres filtres de recherches par date et par véhicules et aussi ajouté des pop-ups sur les markers.
- Ajouter des interfaces sur les appareils mobiles pour permettre aux conducteurs de saisir les stocks transportés et vendu
- On peut même aller plus loin et ajouter des fonctionnalités qui permettent de générer et lire des QR Code ou code barre.

Références bibliographiques

Références Bibliographie

- [1] : Site Officiel Android <http://android.com>
- [2] : <http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/index.html#cara>
- [3] : <http://www.wikipedia.org/android/>
- [4] : http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_comp.html
- [5] : <http://developer.android.com/guide/topics/location/index.html>.
- [6] : <http://www.pointgphone.com/android/emulateur/>
- [7] : <http://www.mti.epita.fr/blogs/2010/08/03/introduction-a-la-programmation-sous-android/>
- [8] : Activité Android <http://developer.android.com/reference/android/app/Activity.html>
- [9] : Géolocalisation <http://blog.nordnet.com/loeil-sur-le-web/decryptages-loeil-sur-le-web/la-geolocalisation.html>
- [10] : <http://anr-prodige.com/index.php?n=Technologies.Geolocalisation>
- [Neviere, 2001], Philippe Nevier, Etude sur la co-localisation d'un réseau UMTS avec les réseaux GSM et DCS existants, 2001.
- [11] : Géolocalisation par wifi <http://www.cnil.fr/linstitution/actualite/article/article/la-geolocalisation-a-partir-des-points-dacces-wi-fi/>
- [12] : Géolocalisation par adresse IP <http://www.definitions-webmarketing.com/Definition-Geolocalisation-adresse-IP>
- [CiscoMag, 2006], CiscoMag, Géolocalisation WiFi & RFID décembre 2006.

ANNEXE

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

Installation de l'Eclipse.

2. Installation du SDK Android.
3. Installation du plugin ADT sous Eclipse.
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.