

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



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

## **Mémoire de Fin d'Etudes de MASTER**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Ingénierie des Systèmes d'Information**

*Présenté par*

**DJAOUTI Kamel**

**AMENOUCHE Djaffar**

Thème

# **Conception et réalisation d'une application Anroid de géolocalisation des stations-service de NAFTAL**

*Mémoire soutenu le ...../...../20..... devant le jury composé de :*

**Président : Mr YACINE**

**Encadreur : Mr HAMMACHE Arezki**

**Examineur : Mr RADJA**

**Examineur : Mme ABERBOUR**

## INTRODUCTION GENERALE

Introduction générale.....1

### CHAPITRE I : Concept de base sur le développement Android

|  |                              |
|--|------------------------------|
| I.1 introduction .....                                 | Error! Bookmark not defined. |
| I.2 la plate formeAndroid .....                        | Error! Bookmark not defined. |
| I.2.1 Définition de la plate forme Android .....       | Error! Bookmark not defined. |
| I.2.2 Historique et version d'Android.....             | Error! Bookmark not defined. |
| I.2.3 Architecture d'Android.....                      | Error! Bookmark not defined. |
| I.2.3.1 Applications.....                              | Error! Bookmark not defined. |
| I.2.3.2 Le framework (Application Framework) :.....    | Error! Bookmark not defined. |
| I.2.3.3 Les bibliothèques (Libraries) : .....          | Error! Bookmark not defined. |
| I.2.3.4 Librairies .....                               | Error! Bookmark not defined. |
| I.2.3.5 Linux .....                                    | Error! Bookmark not defined. |
| I.2.3.6 Android Runtime .....                          | Error! Bookmark not defined. |
| I. 3 Développement d'application Android.....          | Error! Bookmark not defined. |
| I.3.1 Définition d'une application Android.....        | Error! Bookmark not defined. |
| I.3.2 Les composants d'une application Android .....   | Error! Bookmark not defined. |
| I.3.2.1 L'Intent.....                                  | Error! Bookmark not defined. |
| I.3.2.2 Les View.....                                  | Error! Bookmark not defined. |
| I.3.2.3 L'Activity .....                               | Error! Bookmark not defined. |
| I.3.2.4 Les Service .....                              | Error! Bookmark not defined. |
| I.3.2.5 Le ContentProvider .....                       | Error! Bookmark not defined. |
| I.3.2.6 Le BroadcastReceiver.....                      | Error! Bookmark not defined. |
| I.3.3 Les Outils de développement Android.....         | Error! Bookmark not defined. |
| I.3.3.1 Eclipse-SDK et Android studio.....             | Error! Bookmark not defined. |
| I.3.3.2 SDK .....                                      | Error! Bookmark not defined. |
| I.3.3.3 Emulateur .....                                | Error! Bookmark not defined. |
| I.3.3.4 Android studio .....                           | Error! Bookmark not defined. |
| I.3.3.5 Eclipse .....                                  | Error! Bookmark not defined. |
| I.3.3.6 MongoDB.....                                   | Error! Bookmark not defined. |
| I.4 Présentation de quelques applications Android..... | Error! Bookmark not defined. |
| I.4.1 La médecine.....                                 | Error! Bookmark not defined. |
| I.4.2 Dans le domaine des services .....               | Error! Bookmark not defined. |
| I.5 Conclusion .....                                   | Error! Bookmark not defined. |

### CHAPITRE II : Analyse et conception

|   |                              |
|---|------------------------------|
| II.1. Introduction .....                          | Error! Bookmark not defined. |
| II.1.1 Présentation de l'organisme d'accueil..... | 19                           |
| II.1.2 Problématique .....                        | 19                           |
| II.1.3 Solutions proposés .....                   | 20                           |
| II.2 l'UML .....                                  | Error! Bookmark not defined. |
| II.2.1 Définition.....                            | Error! Bookmark not defined. |
| II.2.2 Historique de langage UML .....            | Error! Bookmark not defined. |

|   |                              |
|---|------------------------------|
| II.2.3 Diagrammes d'UML .....   | Error! Bookmark not defined. |
| II.3 Démarche de modélisation utilisée .....                                      | Error! Bookmark not defined. |
| II.4. Analyse .....   | Error! Bookmark not defined. |
| II.4.1 Identification des besoins.....  | Error! Bookmark not defined. |
| II.4.1.1 Spécification des besoins fonctionnels .....                             | Error! Bookmark not defined. |
| II.4.1.2 Spécification des besoins non fonctionnels.....                          | Error! Bookmark not defined. |
| II.4.2 Identification des acteurs.....  | Error! Bookmark not defined. |
| II.5 Diagramme de cas d'utilisation.....  | Error! Bookmark not defined. |
| II.5.1 Les cas d'utilisation.....   | Error! Bookmark not defined. |
| II.5.1.1 Diagramme de cas d'utilisation globale .....                             | Error! Bookmark not defined. |
| II.5.1.2 Diagramme de cas d'utilisation consultation de la maps                   | Error! Bookmark not defined. |
| defined.  |                              |
| II.5.1.3 Diagramme de cas d'utilisation localisation des stations-services.....   | Error!                       |
| Bookmark not defined.   |                              |
| II.5.1.4 Diagramme de cas d'utilisation Trouver l'itinéraire le plus proche ..... | Error!                       |
| Bookmark not defined.   |                              |
| II.5.1.5 Diagramme de cas d'utilisation ajouté des stations service               | Error! Bookmark not defined. |
| defined.  |                              |
| II.5.1.6 Diagramme de cas d'utilisation ajouté des chauffeurs ...                 | Error! Bookmark not defined. |
| defined.  |                              |
| II.6 Conception .....   | Error! Bookmark not defined. |
| II.6.1 Le diagramme de séquence .....   | Error! Bookmark not defined. |
| II.6.2 Diagramme de séquence « s'authentifier » .....                             | Error! Bookmark not defined. |
| II.6.3 Le diagramme de séquence « Localisations des stations-services »           | Error! Bookmark not defined. |
| not defined.  |                              |
| II.6.4 Le diagramme de séquence « Ajouter une station-service ».                  | Error! Bookmark not defined. |
| defined.  |                              |
| II.6.5 Les diagrammes de classe .....   | Error! Bookmark not defined. |
| II.6.6 diagrammes de classe général .....   | Error! Bookmark not defined. |
| II.7 Les tables.....  | Error! Bookmark not defined. |
| II.7.1 Table utilisateur .....  | Error! Bookmark not defined. |
| II.7.2 Table chauffeur .....  | Error! Bookmark not defined. |
| II.7.3 Table administrateur .....   | Error! Bookmark not defined. |
| II.7.4 Table station service.....   | Error! Bookmark not defined. |
| II.7.5 Table Position .....   | Error! Bookmark not defined. |
| II.7.6 Table Situé.....   | 33                           |
| II.8 Conclusion.....  | Error! Bookmark not defined. |

## CHAPITRE III :Réalisation

|                                      |    |
|--------------------------------------|----|
| III.1 Introduction .....             | 39 |
| III.2 Architectura trois tiers ..... | 35 |
| III.3 Environnement de travail ..... | 39 |

|   |    |
|---|----|
| III.3.1 Environnement matériel .....                        | 39 |
| III.3.2 Environnement cible.....                            | 39 |
| III.3.3 Système d'exploitation.....                         | 40 |
| III.4 Environnement logiciel.....                           | 40 |
| III.5 Les langage de programmation .....                    | 40 |
| A. Le langage JAVA.....                                     | 40 |
| B. Le langage Python .....                                  | 40 |
| C. JavaScript.....  | 41 |
| D. Les CSS.....   | 40 |
| III.6 Les outils de programation utilisées.....             | 36 |
| A. Serveur d'Application. ....                              | 40 |
| B. SGBD. ....   | 40 |
| C. MongoDB. ....  | 41 |
| D Software Development Kit (SDK).....                       | 41 |
| E Android.....  | 42 |
| F Emulateur .....   | 42 |
| G Android studio.....                                       | 42 |
| III.7 Les différentes interfaces de notre application ..... | 43 |
| III.7.1 Lancement de l'application.....                     | 43 |
| III.7.2 Authentification de chauffeur.....                  | 44 |
| III.7.3 En cas d'erreur d'authentification.....             | 45 |
| III.7.4 Ce localiser .....                                  | 46 |
| III.7.5 Interface des stations service .....                | 47 |
| III.7.6 Interface recherche d'itinéraire .....              | 48 |
| III.7.7 Interface l'itinéraire en vu de satellite.....      | 49 |
| III.7.8 Interface l'itinéraire en mode hybrid .....         | 50 |
| III.7.9 Interface l'itinéraire en mode normal .....         | 51 |
| III.7.10 Interface authentification admin.....              | 52 |
| III.7.11 Interface espace admin.....                        | 52 |
| III.7.12 Interface menu admin .....                         | 53 |
| III.7.13 Interface gérer les stations .....                 | 54 |
| III.7.14 Interface gérer les chauffeurs.....                | 54 |
| III.7.15 Interface ajouter chauffeur.....                   | 55 |
| III.8 Conclusion .....                                      | 55 |

## CONCLUSION GENERALE

|                             |    |
|-----------------------------|----|
| La conclusion générale..... | 55 |
|-----------------------------|----|

# **TABLE DES FIGURES**

|  |    |
|--|----|
| Figure I.1 : Taux d'utilisation des system d'exploitation pour smartphone en 2014..... | 4  |
| Figure I.2 : Architecture d'android .....  | 7  |
| Figure I.3 Couche Application .....  | 7  |
| Figure I.4 Couche Framawork Application.....   | 8  |
| Figure I.5 Couche Libraries .....  | 8  |
| Figure I.6 Couche Linux Kernel .....   | 9  |
| Figure I.7 Couche Runtime .....  | 9  |
| Figure I.8 : les meilleurs environnements de développement pour Androi .....           | 12 |
| Figure I.9 SDK d'Android .....   | 12 |
| Figure I.10 Emulateur d'Android.....   | 13 |
| Figure I.11 Android studio .....   | 14 |
| Figure I.12 Eclipse .....  | 15 |
| Figure I.13 MongoDB.....   | 16 |
| Figure II.1: Le diagramme d'UML .....  | 20 |
| Figure II.2: Représentation graphique de la Démarche de modélisation utilisée.....     | 21 |
| Figure II.3 : Diagramme de cas d'utilisation globale.....                              | 24 |
| Figure II.4 : Diagramme « Consulter la maps ».....                                     | 24 |
| Figure II.5 : Diagramme de cas d'utilisation «Localiser les stations-services ».....   | 25 |
| Figure II.6 : Diagramme de cas d'utilisation « trouver les itinéraires ».....          | 25 |
| Figure II.7 : Diagramme de cas d'utilisation « Ajouter les stations service ».....     | 25 |
| Figure II.8 : Diagramme de cas d'utilisation « Ajouter des chauffeurs ».....           | 26 |
| Figure II.9 :Diagramme de séquence « Autehntification » .....                          | 27 |
| FigureII.10 :Diagramme de séquence « Localiser stations-services » .....               | 28 |
| Figure II.11 :Diagramme de séquence « Ajouter stations-services ».....                 | 29 |
| Figure II.12: Diagramme de class général.....  | 31 |
| Figure III.1 : architecture trois tiers.....   | 37 |
| Figure III.2 : Interface d'installation du SDK Android .....                           | 38 |
| Figure III.3 Emulateur d'Android .....   | 39 |
| Figure III.4 Android studio .....  | 40 |
| Figure III.5 : Interface de la Page d'accueil .....                                    | 41 |
| Figure III.6 : Interface d'authentification.....                                       | 41 |
| Figure III.7 : Interface erreur d'authentification .....                               | 42 |
| Figure III.8 : Interface géolocaliser.....   | 43 |
| Figure III.9 : Interface des stations service .....                                    | 44 |
| Figure III.10 : Interface recherche d'itinéraire .....                                 | 45 |
| Figure III.11 : Interface l'itinéraire en vu de satellite .....                        | 46 |
| Figure III.12 : Interface l'itinéraire en mode hybrid .....                            | 47 |
| Figure III.13 : Interface l'itinéraire en mode normal .....                            | 48 |
| Figure III.14 : Interface authentification admin .....                                 | 49 |
| Figure III.15: Interface espace admin .....  | 50 |
| Figure III.16 : Interface menu admin .....   | 50 |
| Figure III.17 : Interface gérer les stations .....                                     | 52 |
| Figure III.18 : Interface gérer les chauffeurs.....                                    | 53 |
| Figure III.19 : Interface ajouter chauffeurs .....                                     | 54 |

# **Introduction Générale**

## Introduction Générale

Durant les dernières années, les développements et les progrès technologiques dans le domaine des communications mobiles se sont succédés avec un rythme très rapide, Ces progrès technologiques, en combinaison avec d'autres facteurs constitueront des éléments déterminants qui influenceront le marché mobile dans les années à venir.

Depuis quelques années, les Smartphones sont dotés d'une puissance plus importante et d'espaces de stockage conséquents. Les téléphones tendent à devenir des objets artistiques, presque de reconnaissance sociale, et possèdent des fonctionnalités qu'aucun téléphone ne pouvait espérer auparavant : connexion haut débit, localisation GPS, boussole, accéléromètre, écran tactile souvent multipoint, marché d'applications en ligne, etc. Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité. La plateforme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes par son ouverture ; elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité.

De nos jours les applications Android touche divers domaines par exemple : médicale, touristique, météorologique, etc, et Google n'a pas l'intention de s'arrêter là, puisque il réussit à les intégrer aussi dans le domaine de l'horlogerie (Android wear) des montres intelligentes, et il a été également dans les téléviseurs (Android tv) .

Pour notre projet nous avons décidé de réaliser une application dans le domaine de la géolocalisation qui permet entre autre de localiser l'utilisateur (chauffeur) puis l'orienter et lui montrer le chemin le plus court vers une destination précise qui est dans notre cas les stations-services .

Pour réaliser ce projet nous avons fait une étude réaliste pour la difficulté que trouvent les chauffeurs pour rejoindre les différentes stations-services que possède la grande société NAFTAL et proposer une solution la plus satisfaisante possible pour faciliter l'approvisionnement en carburants dans de brefs délais.

Pour cela, nous avons organisé ce mémoire comme suit :

- Le premier chapitre présente le concept de base sur le développement Android.  
Dans ce chapitre nous avons défini les points suivants : une petite introduction, Définition de la plateforme Android, historique et version d'Android, architecture Android, définition d'application Android, les composants d'une application Android, les outils de développement Android et présentation, d'application Android.
- Le deuxième chapitre va illustrer l'analyse et la conception de notre application.  
Dans ce chapitre nous avons présenté l'organisme d'accueil et la problématique, proposé une solution adéquate, puis nous avons présenté l'UML avec une petite définition et son historique,  
La démarche de modélisation utilisée, après l'analyse on aura les spécifications des besoins fonctionnels et non fonctionnels, identification des acteurs, nous avons représenté les diagrammes suivants :
  - diagrammes de cas d'utilisation détaillé et globales,
  - diagrammes de séquences,
  - diagramme de classe générale.En fin nous avons présenté les tables de notre base de donnée
- Le troisième chapitre présente les outils utilisés pour la réalisation de notre application, ainsi que la présentation de l'application obtenue.  
Nous avons commencé par définir l'environnement de travail avec le matériel et logiciel utilisé (hardware et software), les langages de programmation, puis nous avons présenté quelques Interfaces qui constituent nos applications (mobile et desktop) et finalement une petite conclusion.



- Enfin une conclusion générale qui achèvera notre mémoire.

# **CHAPITRE I: Concepts de Base Sur le Devlopment Android**

## **I.1 introduction**

Android est un système d'exploitation pour Smartphones et tablettes tactiles conçu par Android. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radio-réveils ou des autoradios et même des voitures. Il a été développé par une petite startup qui fut achetée en 2007 par Google qui poursuit activement son développement avec l'Open Handset Alliance, Android est distribué sous licence open source depuis 2008. Ce système est assez nouveau auprès des programmeurs.

Selon Google qui est le majeur distributeur, Android est une plateforme puissante, moderne, sécurisée et ouverte. Elle est basée sur le kernel Linux 2.6 (noyau Linux 2.6) et utilisant la plateforme Java pour ses applications. Elle est entièrement gratuite et sa plateforme très flexible ce qui permet aux développeurs d'intégrer, d'agrandir et de remplacer les composants existants et d'adapter les applications aux besoins du client ou les remplacer entièrement, l'utilisateur peut donc personnaliser facilement son appareil.

De plus, il n'y a pas de distinction entre les applications natives et les applications qui sont développées par les développeurs, toutes sont disponibles sur l'**Android Market** (maintenant appelé **Google Play Store**).

En termes d'applications, Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps.<sup>[1]</sup>

Ce présent chapitre a pour objectif de présenter la plateforme Android et quelques versions d'Android, l'architecture d'Android, ainsi que quelques applications Android dans les domaines médical et service.

## **I.2 La plateforme Android**

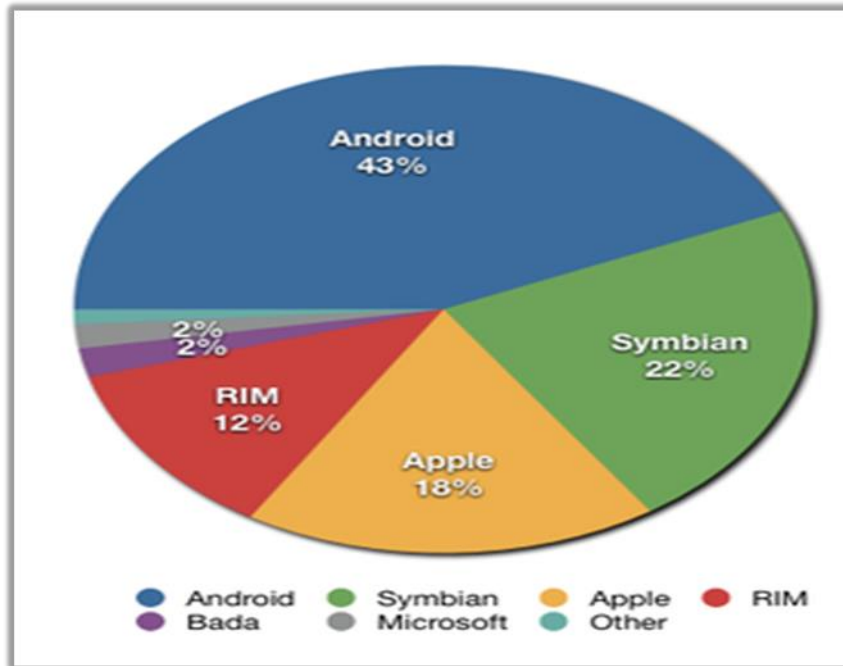
### **I.2.1 Définition de la plateforme Android :**

Android c'est une plateforme complète pour appareil mobile, lancée par Google et l'OHA (Open Handset Alliance) qui est composée d'opérateurs mobiles, de fabricants de téléphones et d'éditeurs logiciels. Elle concurrence des plateformes telles que l'iOS d'Apple, Windows Mobile de Microsoft, RIM (Research In Motion) intégré dans les BlackBerry, Symbian et MeeGo de Nokia.

La plateforme Android est un OS (Operating System) basée sur un kernel Linux entièrement gratuit, sous licence open source. Elle est composée d'un système d'exploitation, de bibliothèques, et d'un ensemble d'applications. La version 1.0 est annoncée en novembre 2007 et la version actuelle des sources est 5.0.0 sortie le novembre 2014.

Le succès d'une Plate-forme ne vient pas seulement par l'attraction des utilisateurs, opérateurs et constructeurs. Le succès est également porté par les développeurs tiers et l'économie qui se développe dans ce marché. Pour Android, la communauté de développeurs est très grande. Il y a aujourd'hui plus de 100 000 applications sur.

Il a pu dans un laps de temps concurrencer les légendes du mobile et à conquérir presque la moitié du marché comme on peut voir sur la figure suivante:



**Figure I.1** Taux d'utilisation des système d'exploitations pour smartphone en 2014[2]

### **I.2.2 Historique et version d'Android**

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008, puis il a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces noms de codes suivent une logique alphabétique.[3]

**Présentation de quelques versions :**

| <b>Les versions</b>                    | <b>Fonctionnalités</b>   |
|--|--|
| <b>Android 2.0(Eclair)</b>             | <p>Le 26 octobre 2009, le SDK pour Android 2.0 (Eclair), basé sur même noyau que Donut, est publié,La mise à jour comprend :</p> <p>Pour la version 2.0 :</p> <ul style="list-style-type: none"> <li>• Vitesse hardware optimisée</li> <li>• Support de plus de taille d'écran et résolutions</li> <li>• Réorganisation de l'UI</li> <li>• Nouvelle interface du navigateur et support de l'HTML5</li> <li>• Nouvelle liste de contact</li> <li>• Meilleur contraste pour les arrière-plan</li> <li>• Amélioration de Google Maps 3.1.2</li> <li>• Support de Microsoft Exchange Server par Exchange ActiveSync 2.5</li> <li>• Support de nouvelles fonctions pour l'appareil photo : flash, zoom digital, balance de blancs, effets de couleurs et focus macro</li> </ul> |
| <b>Android 3.0 (Honeycomb)</b>         | <p>Le 22 février 2011, le SDK pour Android 3.0 (Honeycomb), basé sur le noyau 2.6.36, est publié. Réserve aux tablettes tactiles, cette mise à jour comprend de nombreux changements dans l'interface, cette mise à jour comprend de nombreux changements dans l'interface :</p> <ul style="list-style-type: none"> <li>• Interface entièrement retravaillée et optimisée pour les tablettes tactiles</li> <li>• Ajout de la barre système en bas de l'écran qui regroupe l'heure, les notifications et les boutons de navigations</li> <li>• Ajout de la barre d'actions en haut de l'écran qui regroupe la barre de recherche, l'accès au menu paramètre et le menu d'applications</li> <li>• Bureau tridimensionnel avec widgets améliorés</li> </ul>                   |
| <b>Android 4.0 (IceCream Sandwich)</b> | <p>Le 19 octobre 2011, le SDK pour Android 4.0 (IceCream Sandwich), basé sur le noyau 3.0.1, est publié. Cette nouvelle version unifiée pour smartphones et tablettes tactiles</p>   |

|                                      |  |
|--------------------------------------|--|
|                                      | <p>apporte de nombreux changements. Pour la version 4.0.1 sortie le 19 octobre 2011 :</p> <ul style="list-style-type: none"> <li>• Boutons virtuels intégrés à l'interface, en remplacement des boutons physiques, ces derniers peuvent être maintenus selon les choix du constructeur</li> <li>• Ajouts des widgets depuis un nouveau menu similaires à celui des applications</li> <li>• Création plus facile des dossiers</li> <li>• Un nouveau launcher personnalisable</li> <li>• Amélioration de la messagerie visuelle</li> <li>• Ajout du pinch-to-zoom dans le calendrier</li> <li>• Capture d'écran native</li> </ul>  |
| <p><b>Android 5.0 (Lollipop)</b></p> | <p>Annoncée le 15 octobre 2014 et sortie le 3 novembre 2014, <i>Android 5.0 Lollipop</i> est une évolution majeure d'Android qui propose de nombreuses modifications et nouveautés, et qui étend sa disponibilité sur de nouveaux supports tels que la télévision, la voiture ou les montres connectées.</p> <p><i>Android Lollipop</i> est proposée dès sa sortie sur les appareils Nexus, notamment les Nexus 6, 9 et Player qui sont disponibles dès le 3 novembre 2014, et ceux qui sont certifiés Google Experience Edition.</p> <p>Voici la liste des principales nouveautés :</p> <ul style="list-style-type: none"> <li>• Refonte totale de l'interface graphique avec un nouveau design nommé Material Design</li> <li>• Nouveau moteur d'exécution ART (en) qui compile les applications dès leur installation plutôt que la compilation JIT de <u>Dalvik</u></li> </ul> |

**Tableau I.1 : Historique des versions d'Android**

## I.2.3 Architecture d'Android

Android est une pile applicative pour les appareils mobiles qui comprend un système d'exploitation, middleware et des applications clés. Le SDK Android fournit les outils et les API nécessaires pour commencer à développer des applications sur la plate-forme Android en utilisant le langage de programmation Java. [4]

Le schéma suivant illustre les principaux composants du système d'exploitation Android. Chaque section est décrite plus en détail ci-dessous.

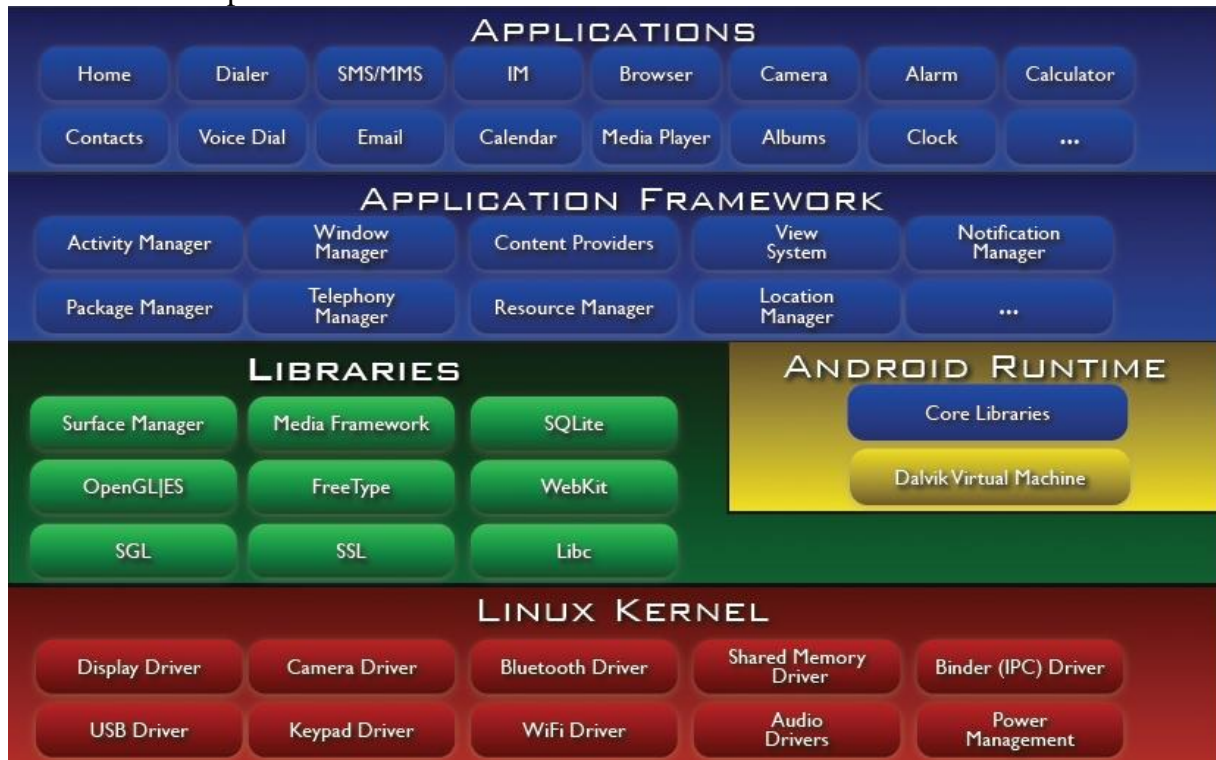


Figure I.2 Architecture d'android [4]

### I.2.3.1 Applications

Android est fourni avec un ensemble de programmes de base (également nommés applications natives) permettant d'accéder à des fonctionnalités comme les courriels, les SMS, le téléphone, le calendrier, les photos, les cartes géographiques, le Web, pour ne citer que quelques exemples. Ces applications sont développées à l'aide du langage de programmation Java. Pour l'utilisateur final, c'est la seule couche accessible et visible.

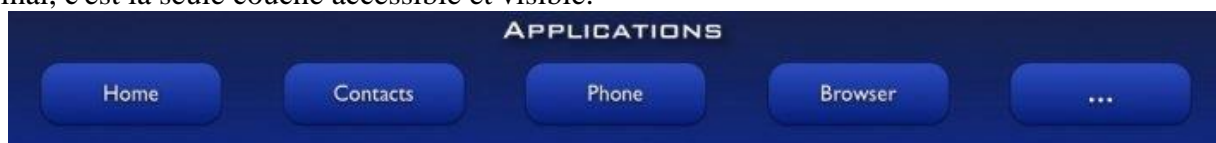


Figure I.3 Couche Application

### I.2.3.2 Le framework (Application Framework) :

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



Figure I.4 Couche Framework Application

### I.2.3.3 Les bibliothèques (Libraries) :

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du framework Android. En effet, le framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard. La technologie Java Native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++. La liste ci-dessous énumère quelques-unes des bibliothèques disponibles dans Android :

- **Bibliothèque système C.** Implémentation (dérivée de BSD) de la bibliothèque standard C (libc), optimisée pour les systèmes Linux embarqués.
- **Bibliothèques multimédias.** Basées sur StageFright, elles permettent le support de nombreux formats audio et vidéo, tels que MPEG4, H.264, MP3, AAC, AMR, JPG et PNG (la liste complète est disponible sur le site des développeurs Android : <http://d.android.com/guide/appendix/media-formats.html>).
- **SurfaceFlinger.** Permet l'accès au sous-système d'affichage.
- **LibWebCore.** Moteur de rendu de pages Internet basé sur Webkit. Cette bibliothèque est donc principalement utilisée dans le navigateur et dans les vues web embarquées (WebView).
- **Skia.** Moteur graphique 2D.
- **Bibliothèques 3D.** Implémentation basée sur OpenGL ES 1.0 API et plus récemment OpenGL ES 2.0.
- **FreeType.** Rendu des polices de caractères.
- **SQLite.** Base de données légère et puissante.

### I.2.3.4 Librairies

Au dessus du kernel, il y a les librairies natives. Ces librairies sont écrites en C/C++. Elles fournissent les fonctionnalités de bas niveau d'Android.



Figure I.5 Couche Libraries

### I.2.3.5 Linux

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



Figure I.6 Couche Linux Kernel



### I.2.3.6 Android Runtime

Cette couche se situe au dessus des libraires C/C++, elle se compose du "cœur" du Framework et de la machine virtuel dalvik.



Figure I.7 Couche Runtime

## I.3 Développement d'application Android

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

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

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

### I.3.1 Définition d'une application Android

Une application Android est une application mobile spécifiquement développée pour les Smartphones utilisant le système d'exploitation Android acheté et développé par Google. [5]

Comme les applications iPhone dont elles sont souvent des répliques, les applications Android sont de nature très variables, quelque application android sont présentées dans la section I.4.

### I.3.2 Les composants d'une application Android

Le développement sur Android s'appuie sur des classes importantes du framework. Ces classes sont, en quelque sorte, les "briques" élémentaires sur lesquelles les futures applications reposeront. Elles sont tellement indispensables, qu'il est tout simplement impossible de construire une application sans passer par au moins sur l'une d'elles. Cette partie tente de décrire de façon globale et succincte ces différentes "briques".

#### I.3.2.1 L'Intent

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. La raison d'être des intents provient du modèle de sécurité d'Android. Chaque application est en effet *sandboxée*. Cela veut dire qu'une application a ne peut accéder aux données d'une application b. Grâce aux intents, les applications ont la possibilité de fournir leurs services ou données si elles le souhaitent.

### **I.3.2.2 Les View**

Les views sont les composants de base de l'interface graphique. Elles permettent de construire l'interface utilisateur. Les *widgets* (nom donné à des composants graphiques "avancés" : une barre de progression, par exemple, est un widget), composants graphiques ou autres *layout* (composant permettant de placer les différents composants graphiques à l'écran) héritent en fait de cette classe élémentaire. Le rendu/dessin d'une View s'effectue par l'intermédiaire d'un Canvas (qu'on peut assimiler à une feuille transparente sur laquelle on dessine avec un crayon (Paint)). Pour finir, la vue est le principal composant qui s'occupe de gérer les actions utilisateurs (appui sur l'écran, sur le clavier, etc.).

Lorsqu'on crée une application à l'aide du framework Android, le développeur doit hériter d'au moins une des quatre classes brièvement décrites ci-dessous

### **I.3.2.3 L'Activity**

Le concept d'Activité repose sur la notion d'interaction utilisateur. Une Activité représente la fenêtre ou tout simplement l'écran qui sera affiché à l'utilisateur. Elle permet également de gérer des fonctionnalités telles que l'appui sur la touche [MENU] ou l'affichage de messages d'alerte (Toast). ne pas confondre la notion d'Activity et de View. Il est évident que faire la différence entre ces deux notions.

### **I.3.2.4 Les Service**

La notion de service est un avantage certain sur l'iPhone. En effet, un Service est en fait un programme tournant en tâche de fond et n'ayant pas d'interface graphique (ce qui n'est pas réalisable sur iPhone OS). L'exemple commun illustrant au mieux 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 travailler/s'exécuter librement.

### **I.3.2.5 Le ContentProvider**

Les ContentProvider sont, comme l'exprime leurs noms, des gestionnaires de données. Ils permettent de partager l'information entre applications. Imaginons une application qui permette de conserver les cartes de visite virtuelles d'un ensemble de personnes. Ces cartes de visite contiennent généralement le nom, le prénom et un moyen de contact de la personne. Un tel programme peut être créé sous forme de ContentProvider ce qui lui permettra de fournir à d'autres applications présentes sur le système les informations sur une personne. Une application tierce d'envoi de courriel peut par exemple interroger ce ContentProvider afin d'obtenir l'adresse courriel d'un contact.

### **I.3.2.6 Le BroadcastReceiver**

Pour finir, un BroadcastReceiver est une application qui est à "l'écoute" des autres applications. Ce type d'application tente de répondre au Intent qui lui sont adressés. Il ne fait donc rien d'autres que d'être à l'écoute des Intente envoyés par d'autres composants applicatifs.

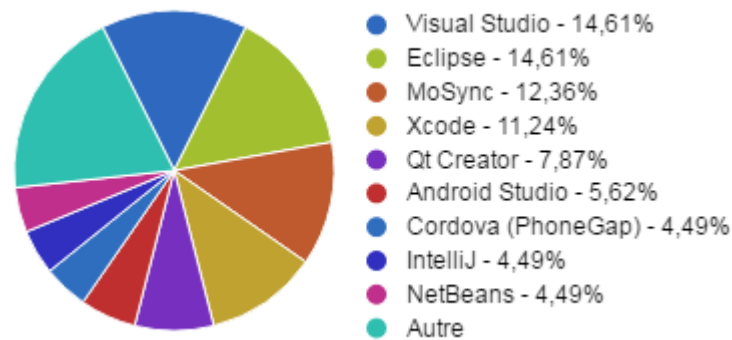
## **I.3.3 Les Outils de développement Android**

Deux framwerk de développent Android sont principalement utilisés :

### **I.3.3.1 Eclipse-SDK et Android studio.**

Le développement d'applications pour Android se fait entièrement en Java. Java est un puissant langage orienté objet, utilisé très largement dans le monde du développement, j'ai fait des recherches Sur les IDE: eclipse et NetBeans.

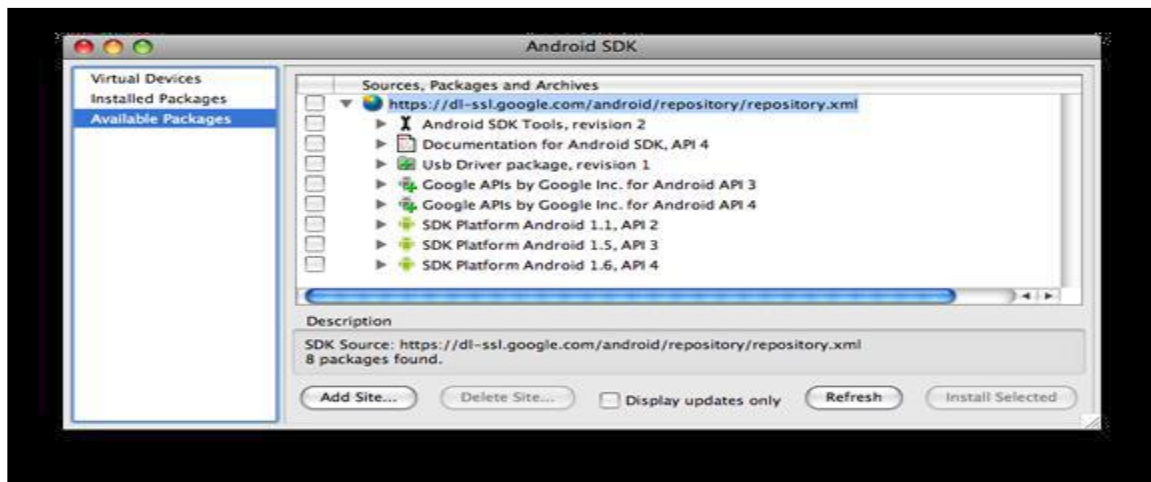
Les deux programmes étaient très connus mais Eclipse est le plus largement utilisé pour la programmation Java; il est très performant, de plus gratuit et open source. [6]



**Figure I.8** les meilleurs environnements de développement pour Android [16]

### I.3.3.2 SDK

Le kit de développement (SDK) d'Android est un ensemble complet d'outils de développement. Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux. Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows 7 ou version ultérieure. L'IDE officiellement supporté était Eclipse combiné au plugin d'outils de développement d'Android (ADT), mais depuis 2015, Google officialise Android Studio qui devient alors l'IDE officiel pour le SDK Android. Les développeurs peuvent utiliser n'importe quel éditeur de texte pour modifier les fichiers Java et XML, puis utiliser les outils en ligne de commande ([Java Development Kit](#) et [Apache Ant](#) sont obligatoires) pour créer, construire et déboguer des applications Android ainsi que contrôler des périphériques Android (pour déclencher un redémarrage, installer un logiciel à distance ou autre) [6]

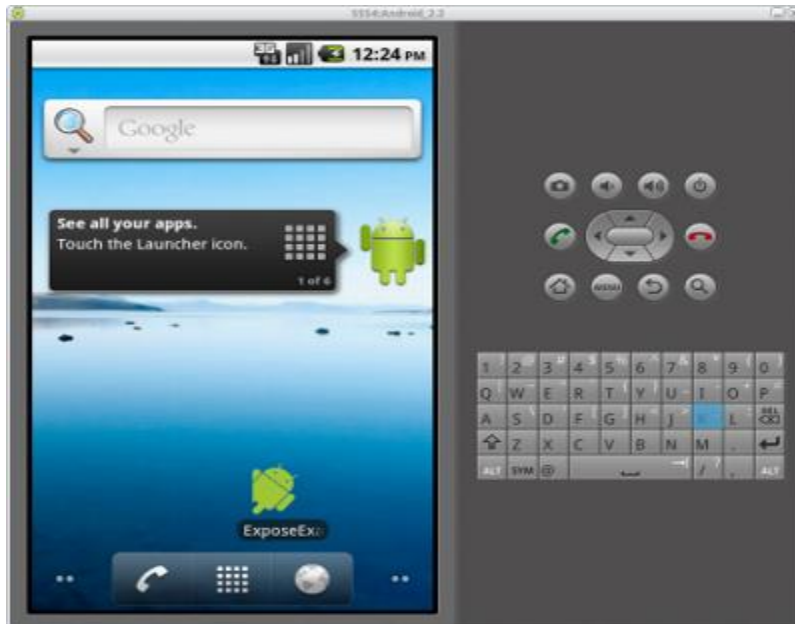


**Figure I.9** SDK d'Android [6]

### I.3.3.3 Emulateur

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

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth. [6]



**Figure I.10** Emulateur d'Android

#### **I.3.3.4 Android studio**

Android Studio est un environnement de développement d'applications Android. Il est basé sur IntelliJ IDEA.

Il permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément

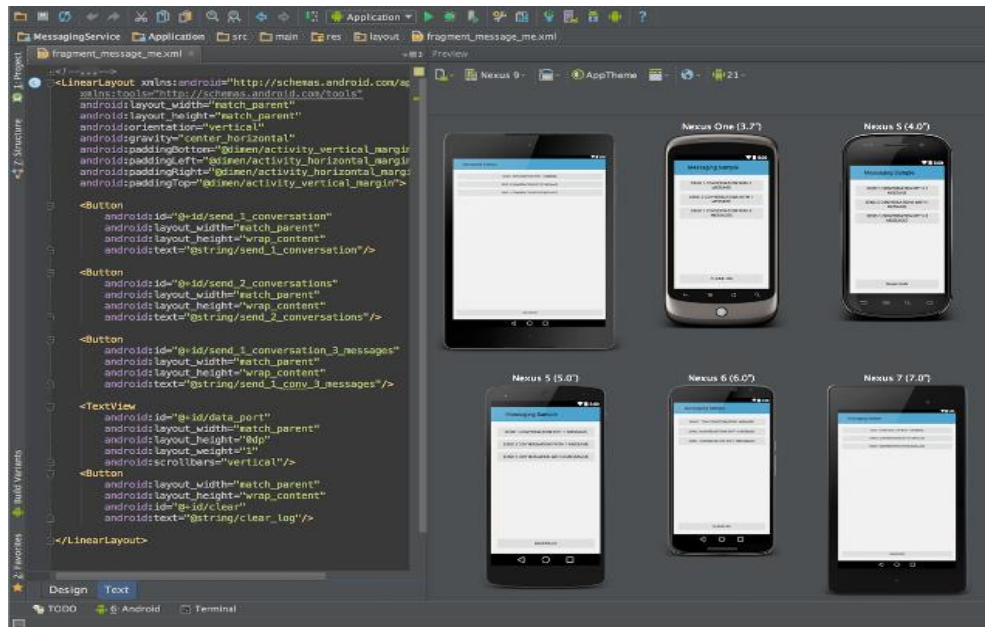


Figure I.11 Android studio

### I.3.3.5 Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

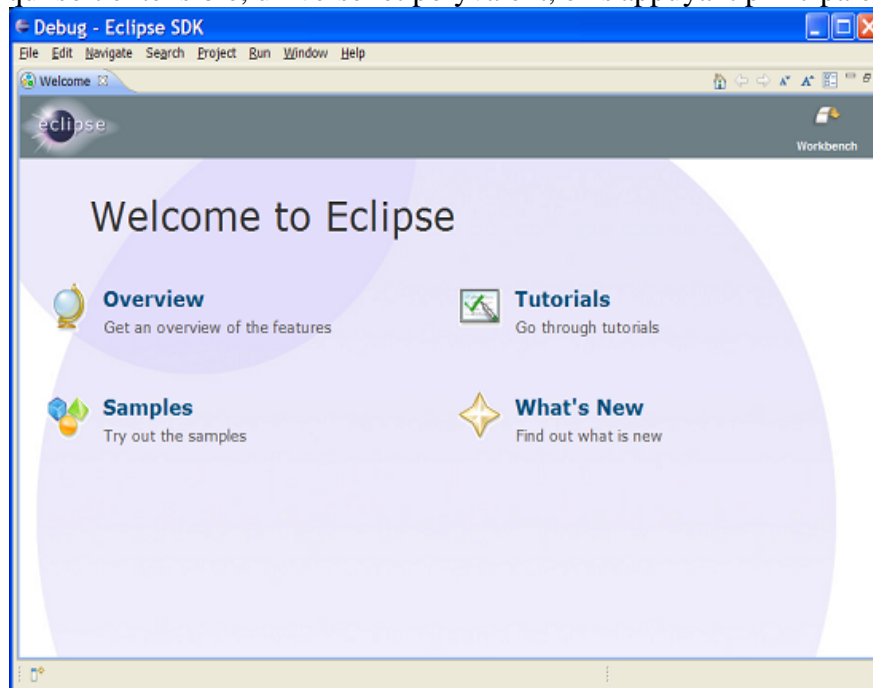


Figure I.12 Eclipse

## I.4 Présentation de quelques applications Android

Les applications Android sont largement utilisées dans divers domaines :

### **I.4.1 La médecine**

A la fin de l'année 2014, dans le monde entier, plus de 100.000 applis santé médecine étaient disponibles, dans un marché de plus de 5 milliards de dollars qui devrait atteindre 26 milliards de dollars en 2017, essentiellement grâce aux services fournis. Ces observations du rapport annuel de Recherche Guidance, analyste des marchés d'applications mobiles, annoncent l'arrivée d'une révolution des pratiques médicales. La m-santé, santé mobile, se développe par la demande des patients, qui entraîneront inévitablement les médecins avec eux. Elle ouvre de nouveaux horizons positifs pour la médecine prédictive, pour la gestion des maladies chroniques ou pour la prise en charge des patients dépendants. Elle pose également de nombreuses questions relatives aux données qu'elle collecte, à leur pertinence, à leurs usages et à leur confidentialité.

Ci-dessous, nous présentons une application du domaine de la santé.

#### **Virtual Consultation Spine**

L'application Virtual Consultation Spine se présente comme un logiciel s'adressant aux patients, permettant une consultation médicale informatisée sur les pathologies de la colonne vertébrale.

Cette application a pour objectif de permettre au patient de collecter et analyser toutes les informations concernant sa pathologie du dos. L'analyse de ces données apportée par le logiciel permet une information approfondie sur les diagnostics possibles et les options thérapeutiques.

En répondant aux mêmes questions qui sont posées lors d'une consultation médicale, l'application génère une interprétation personnalisée :

- Diagnostiques les plus probables
- Explication détaillée des pathologies
- Examens nécessaires
- Options de traitement
- Chances de succès et les risques

Virtual Consultation Spine est basé sur les données médicales les plus récentes et sur la vaste expérience clinique des chirurgiens spécialisés du rachis qui composent le comité scientifique. Elle ne remplace pas une consultation médicale mais peut être utile lors d'une consultation avec un spécialiste.



**Figure I.14 Virtual Consultation Spine**

## I.4.2 Dans le domaine des services

Qui dit voyage, dit découverte, donc la découverte de nouveaux endroits où les meilleures offres, Pour cela on voit de plus en plus d'applications Android du domaine des services dans les smartphones pour faciliter et aider les utilisateurs dans leur voyage et découverte de l'inconnu, Ce guide touristique informe sur tout ce qu'il y a à faire ou à voir à la destination de notre choix : restaurants, hôtels, boutiques, services et activités, comparer les prix.

Ci-dessous, nous présentons une application du domaine des services

### Tripadvisor Mobile:

Est une application mobile internationale d'origine américaine qui offre des avis et des conseils [touristiques](#) émanant de [consommateurs](#) sur des services (hôtels, restaurants, villes et régions, lieux de loisirs, etc) et qui fournit également des outils de réservation de logements et de billets d'avion en comparant des centaines de sites web afin de trouver les meilleurs prix.

L'application mobile TripAdvisor est reconnue par l'ensemble du secteur pour ses fonctionnalités et ses outils innovants. Elle a été nommée « Meilleure application mobile » dans le cadre de la remise des prix Innovation Awards 2014 décernés par la société Eye for Travel. Elle offre un grand nombre de fonctionnalités et une grande quantité de contenu afin d'aider les voyageurs à planifier leur voyage et à en profiter autant que possible. L'application est gratuite, disponible en 21 langues, et peut être téléchargée sur l'App Store iTunes ou sur Google Play.

Tripadvisor a atteint le nombre record de 100 millions de téléchargements de son application gratuite pour iPhone et terminaux Android en 2014 .

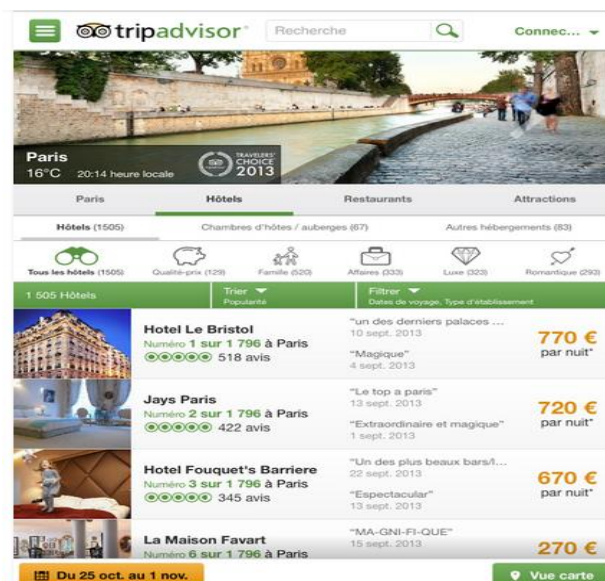


Figure I.15 application Android TRIPADVISOR

## I.5 Conclusion

Dans ce chapitre, nous avons présenté les concepts clés sur Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture d'Android, à savoir les principaux composants du système.

En fin nous avons présenté l'environnement logiciel utilisé pour la programmation Android et quelques applications dans divers domaines.

# **CHAPITRE II: Analyse et Conception**



## **II.1. Introduction**

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

Pour ce faire nous avons adopté la conception avec l'UML (UnifiedModelling Language), qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

La continuité du service de localisation des station service n'étant toujours pas assurée chez nous en Algérie, ce qu'il nous a poussé à réaliser cette application qu'elle permet au utilisateur (chauffeur) de ce localiser, localiser les stations service et de suivre le plus court chemin.

### **II.1.1 Présentation de l'organisme d'accueil**

Pour notre projet on propose de faire une application mobile (Android) pour la société nationale NAFTAL.

NAFTAL est une entreprise algérienne, filiale de SONATRACH, elle est chargée de distribution et la commercialisation des produits pétroliers sur le marché national, elle intervient dans les domaines :

- De la distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitumes, pneumatiques, GPL/carburant, produits spéciaux ;
- Du transport des produits pétroliers.

Pour assurer la disponibilité des produits sur tout le territoire, Naftal met à contribution plusieurs modes de transport :

- Le rail pour le ravitaillement des dépôts à partir des entrepôts ;
- La route pour livraison des clients et le ravitaillement des dépôts non desservis par le rail.

### **II.1.2 Problématique**

Etant obligé d'assurer la distribution des produits sur tous le territoire national

Avec les centaines Stations-services éparpillés un peu partout sur tout le territoire,

Les chauffeurs ont de plus en plus de mal à se retrouver et à trouver leur destinations,

Pour cela la géolocalisation est devenu indispensable pour le bon fonctionnement des services de distributions chez la société NAFTAL .

### **II.1.3 Solutions proposés**

Pour faciliter aux chauffeurs la livraison et la distribution des produits dans toutes les stations-services dans les plus brefs délais,

Nous avons proposés une application Android facile à utiliser pour les chauffeurs qui leur permet :

- Se localiser n'importe où sur le territoire ;
- Voir et choisir la station-service de destination ;
- Trouver le plus court chemin et voir la durée estimée pour son trajet ;
- De plus nous proposons aux responsable de la distribution aux niveaux de NAFTAL une application demis à jour de la base de données ajouter es stations service et es chauffeurs.

## **II.2'UML**

### **II.2.1 Définition**

UML traduit en « langage de modélisation objet unifié » est un langage de description orienté objet qui permet de modéliser une application selon une vision objet. Un objet est décrit par les attributs qui le compose et les traitements, appelé méthodes, qui peuvent lui être appliqués. [7]

## II.2.2 Historique de langage UML

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

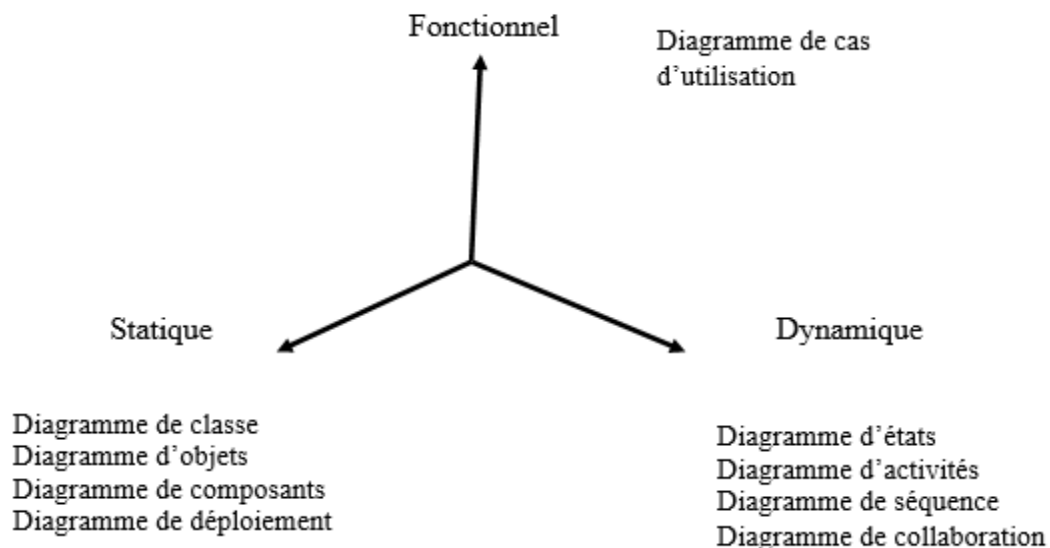
Lorsque la programmation orientée objet prend de l'importance au début des années 90, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995, mais uniquement trois d'entre elles se sont détachées nettement après quelque années : OMT,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

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

Pour la modélisation de notre application, nous allons utiliser le langage de modélisation unifié UML. [17]

## II.2.3 Diagrammes d'UML

UML propose 9 diagrammes de modélisation, réparties sur trois axes du niveau conceptuel : [8]



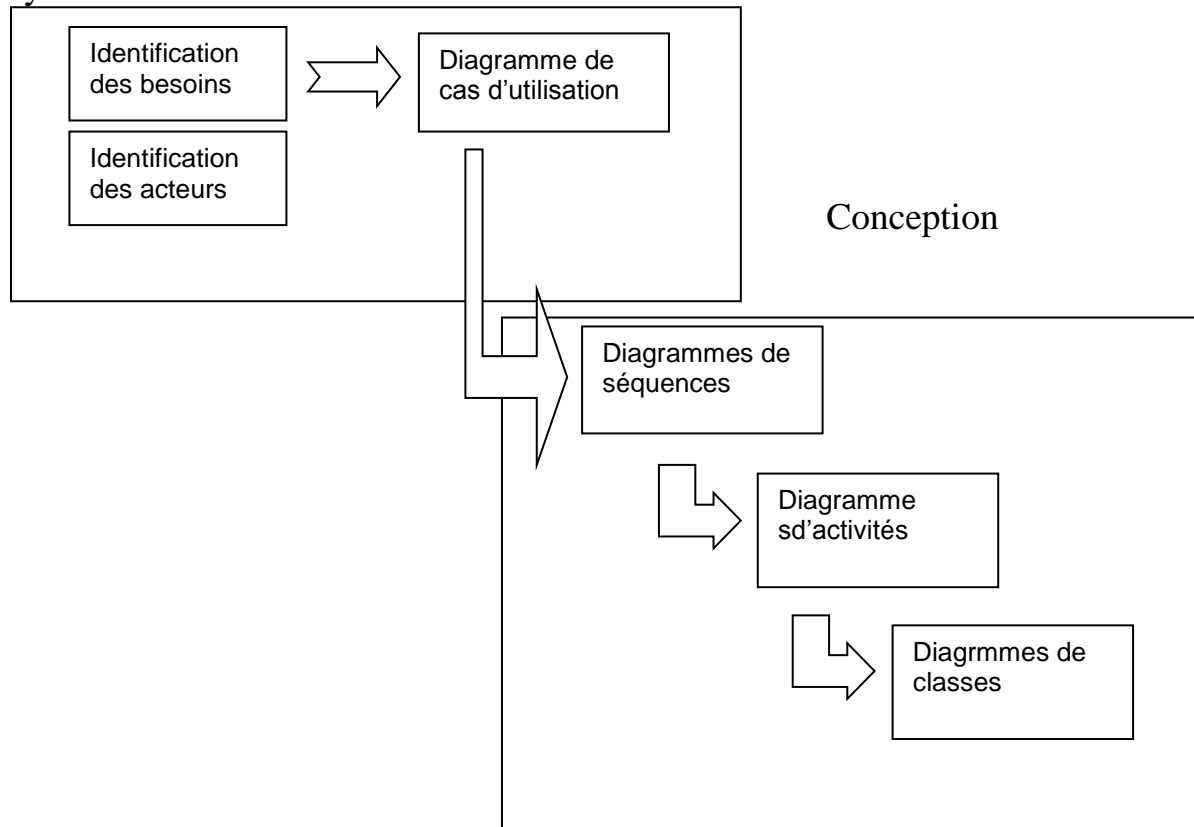
**Figure II.1** Le diagramme d'UML

La modélisation proposée par le langage UML se réalise principalement sous forme graphique, en utilisant divers types de diagrammes spécifiques, répartis en trois groupes, comme l'illustre la **figure II.1**.

## II.3 Démarche de modélisation utilisée

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

## Analyse



**Figure II.2** Représentation graphique de la démarche de modélisation utilisée

## II.4. Analyse

Il s'agit d'une étape cruciale dans la réalisation d'une application donnée. Le futur d'une application dépend beaucoup de cette phase, elle nous permet d'éviter le développement d'une application non satisfaisante. Pour cela le client et le développeur doivent être en étroites relations, voire avoir un intermédiaire entre eux.

### II.4.1 Identification des besoins

#### II.4.1.1 Spécification des besoins fonctionnels

L'objectif principal de notre application est de permettre à l'utilisateur de se localiser dans n'importe quel endroit. Il suffit juste que son appareil soit muni de la fonctionnalité GPS et qu'il ait accès à internet.

- L'authentification de chauffeur.
- Consultation de la maps.
- Localisation des stations-services.
- Trouver l'itinéraire le plus proche .

#### II.4.1.2 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. En ce qui concerne notre application, nous avons dégagé les besoins suivants :

- L'accès à internet : l'application doit être connectée à internet pour être localiser et se mettre à jour .

### II.4.2 Identification des acteurs

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

### **Les acteurs de notre système :**

**chauffeur** : c'est toute personne qui utilise une machine (Téléphone, Tablette, etc) qui marche sous le système Android pour localiser les station services.

**Administrateur** : Personne chargée d'administrer l'application, notamment la mise à jour de la base de données.

## **II.5 Diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux Utilisateurs du système. C'est le premier diagramme du modèle UML au niveau duquel s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.[9]

### **II.5.1 Les cas d'utilisation**

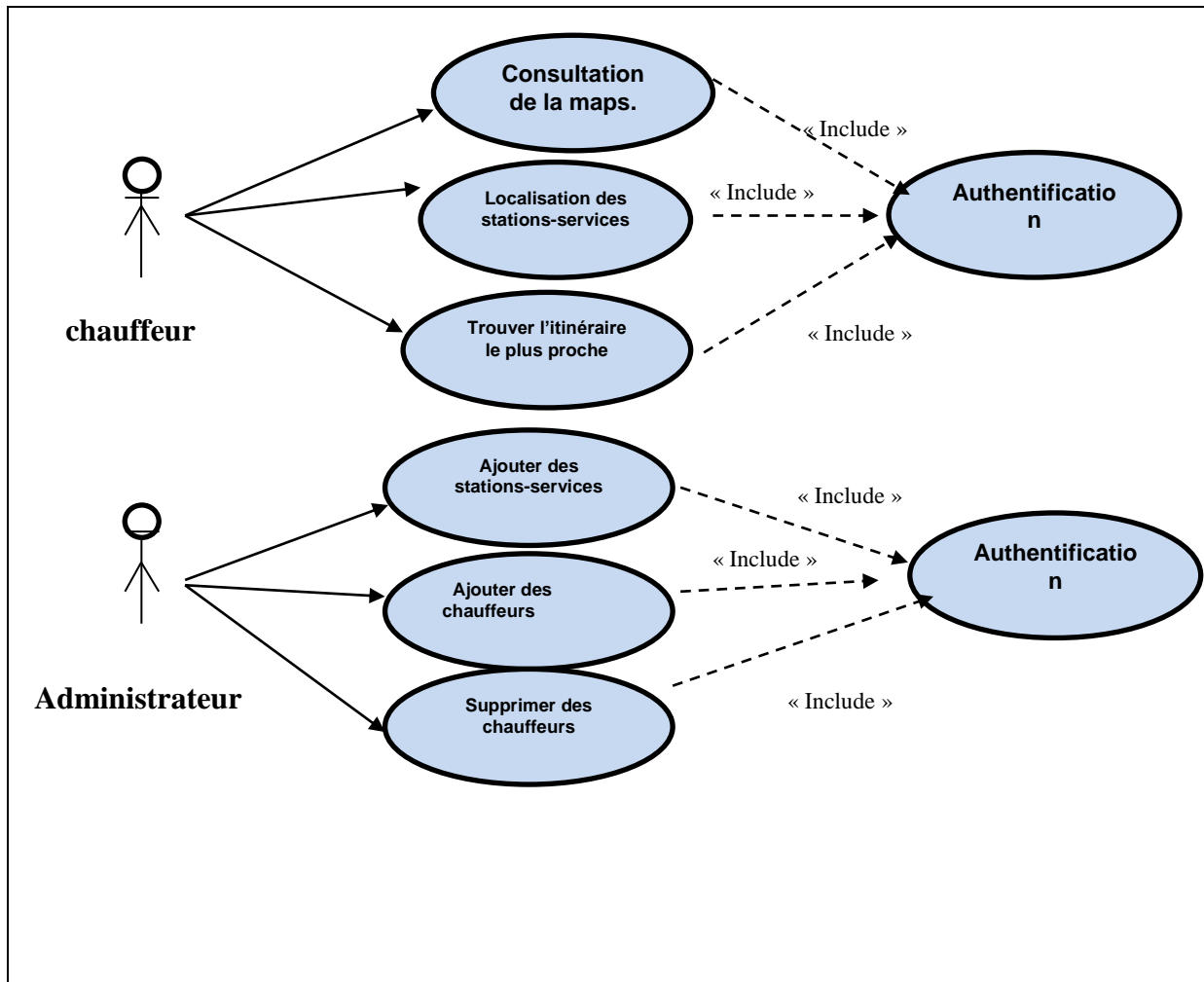
**Définition** : Un cas d'utilisation (**user case**) représente un ensemble de séquence d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système.

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

**La relation d'extension (Extend)** : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions. Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation suivant :

#### **II.5.1.1 Diagramme de cas d'utilisation globale**

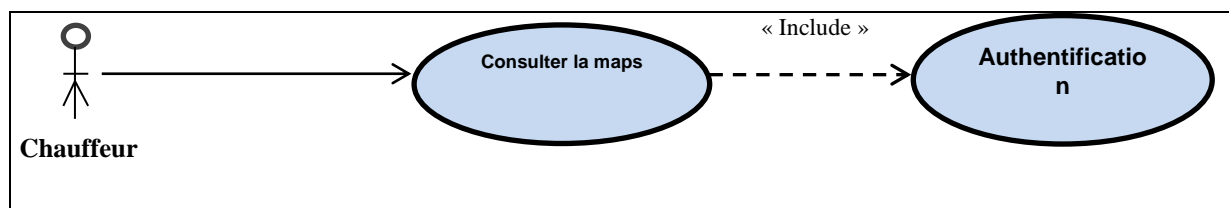
Ci-dessous, nous présentons le diagramme de cas d'utilisation pour la compréhension du fonctionnement du système.



**Figure II.3** Diagramme de cas d'utilisation globale

### II.5.1.2 Diagramme de cas d'utilisation consultation de la maps

Description du cas d'utilisation « consulter la maps »



**Figure II.4** Diagramme « Consulter la maps »

### II.5.1.3 Diagramme de cas d'utilisation localisation des stations-services

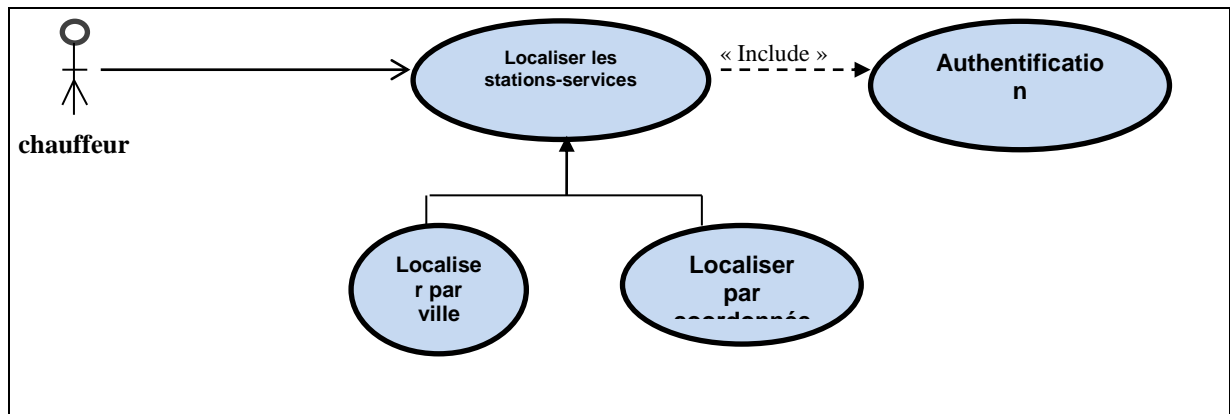


Figure II.5 Diagramme de cas d'utilisation «Localiser les stations-services »

### II.5.1.4 Diagramme de cas d'utilisation Trouver l'itinéraire le plus proche

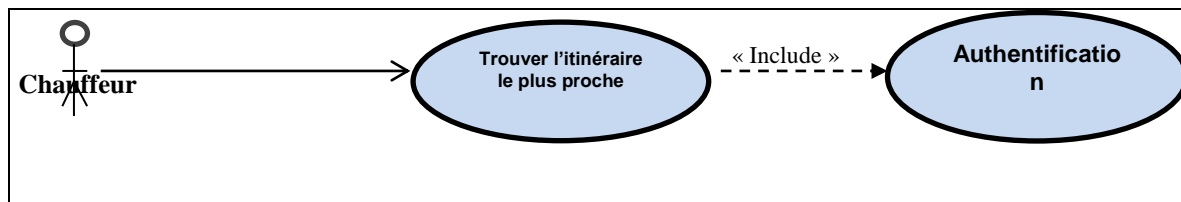


Figure II.6 Diagramme de cas d'utilisation« trouver les itinéraires »

### II.5.1.5 Diagramme de cas d'utilisation ajouté des stations-services

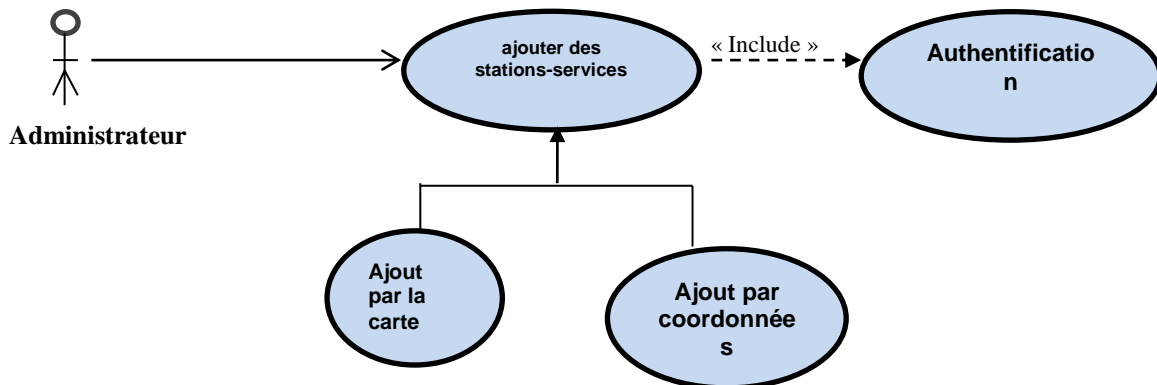


Figure II.7 Diagramme de cas d'utilisation « Ajouter les stations service »

### II.5.1.6 Diagramme de cas d'utilisation ajouté des chauffeurs

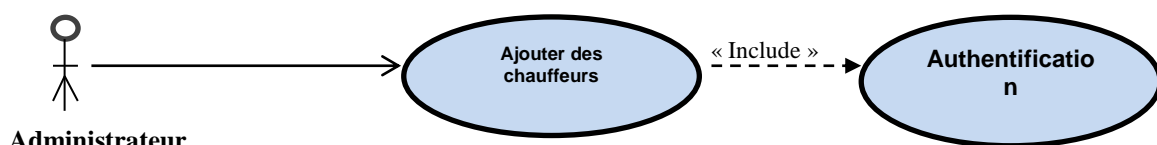


Figure II.8 Diagramme de cas d'utilisation « Ajouter des chauffeurs »

## II.6 Conception

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

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

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

### II.6.1 Le diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement ; il met en évidence les objets manipulés ainsi que les opérations qui font passer d'un objet à l'autre. Dans notre cas on s'intéresse seulement à effectuer la représentation du diagramme de séquence pour les cas d'utilisation déjà présentés auparavant.

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

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

**Les messages :** ils sont représentés par des flèches directionnelles. Au-dessus des flèches directionnelles figurent un texte nous informant du message envoyé entre les objets. [10]

Les figures suivantes reprennent quelque représentation graphique des diagrammes utilisés dans la réalisation de cette application :

## II.6 .2 Diagramme de séquence « s'authentifier »

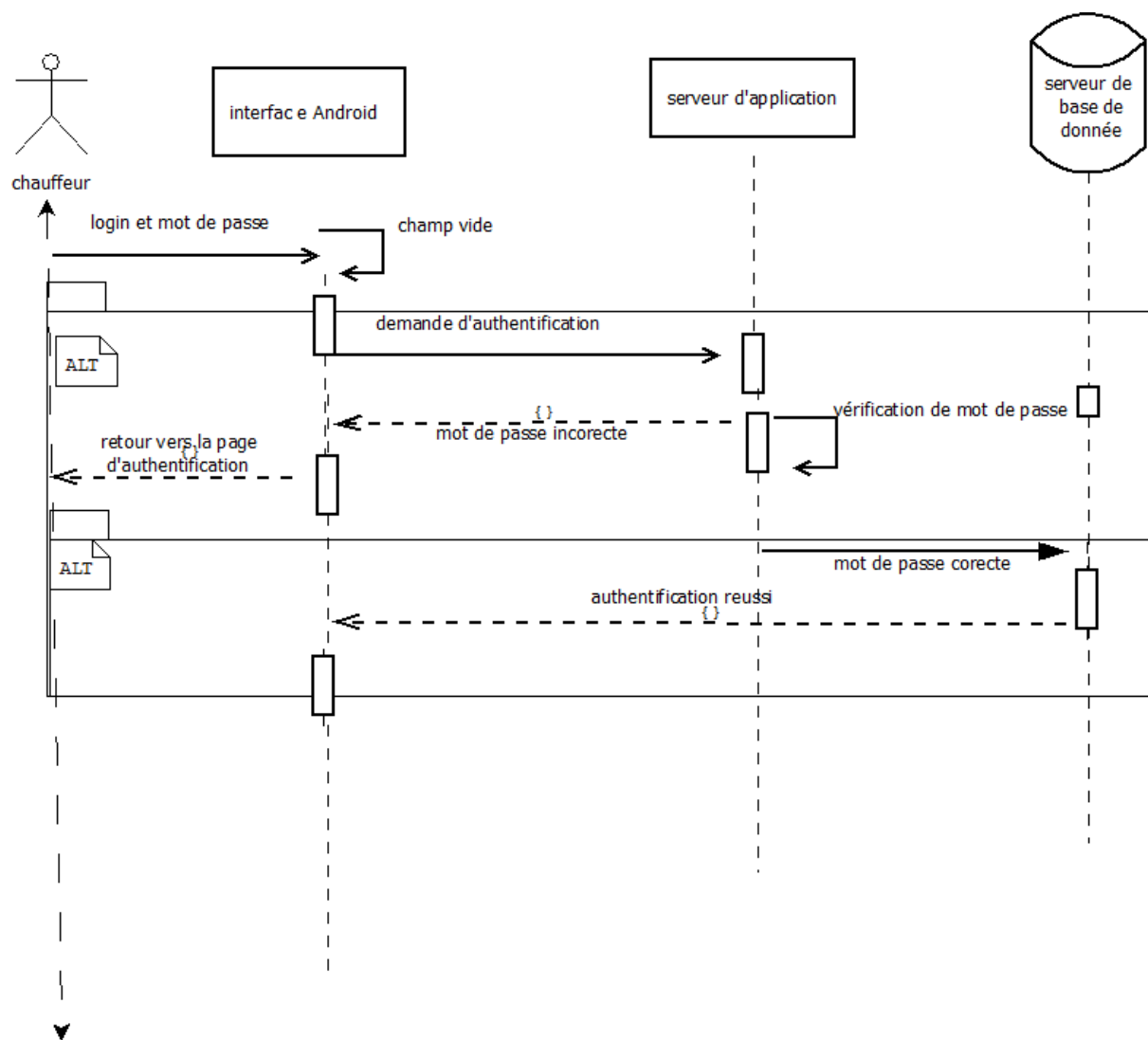
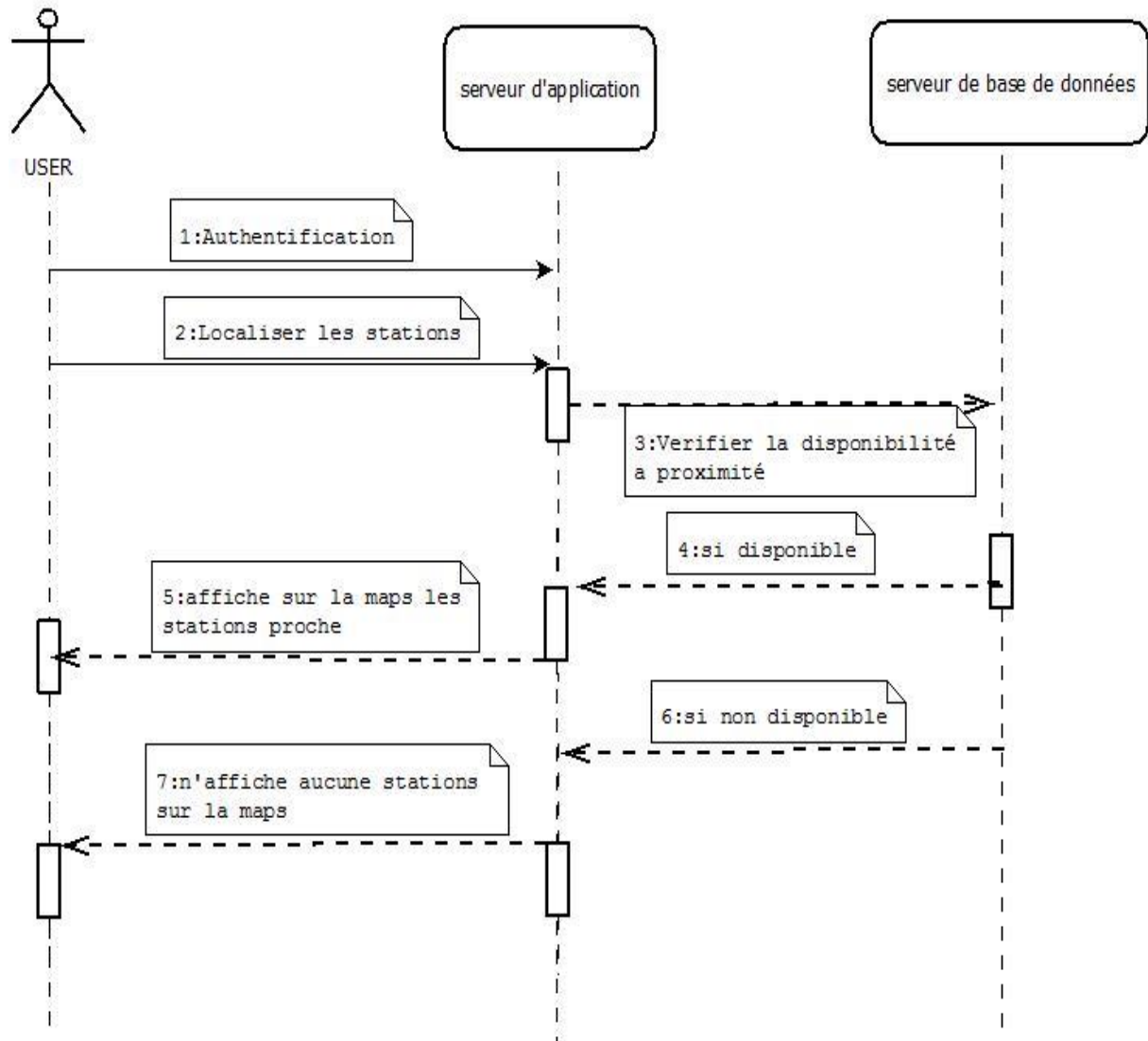


Figure II.9 Diagramme de séquence « s'authentifier »

## II.6.3 Le diagramme de séquence « Localisations des stations-services »

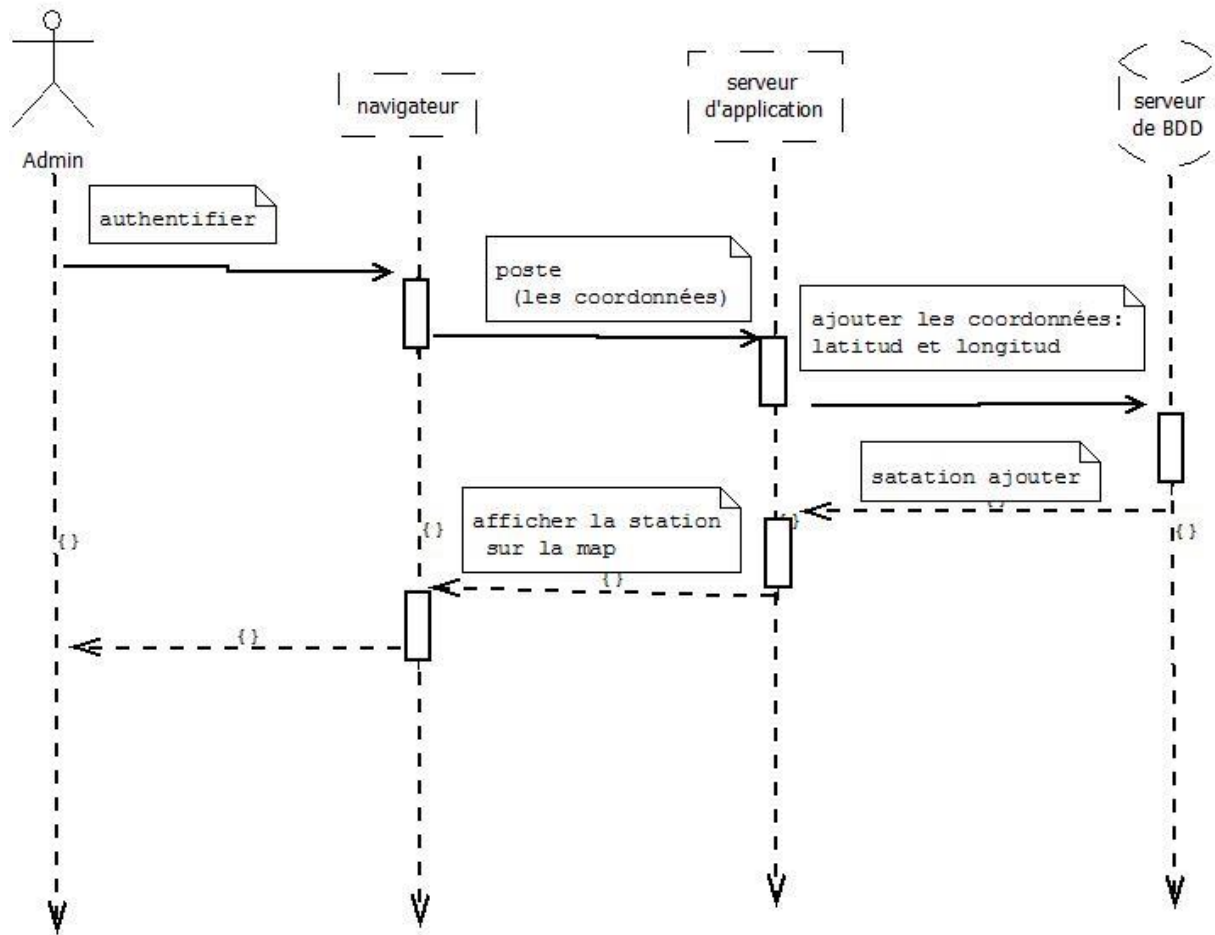




**Figure II.10** Diagramme de séquence détaillé « Localisation des stations-services

»

#### II.6.4 Le diagramme de séquence « Ajouter une station-service »



**Figure II.11** Diagramme de séquence «Ajouter une station-service».

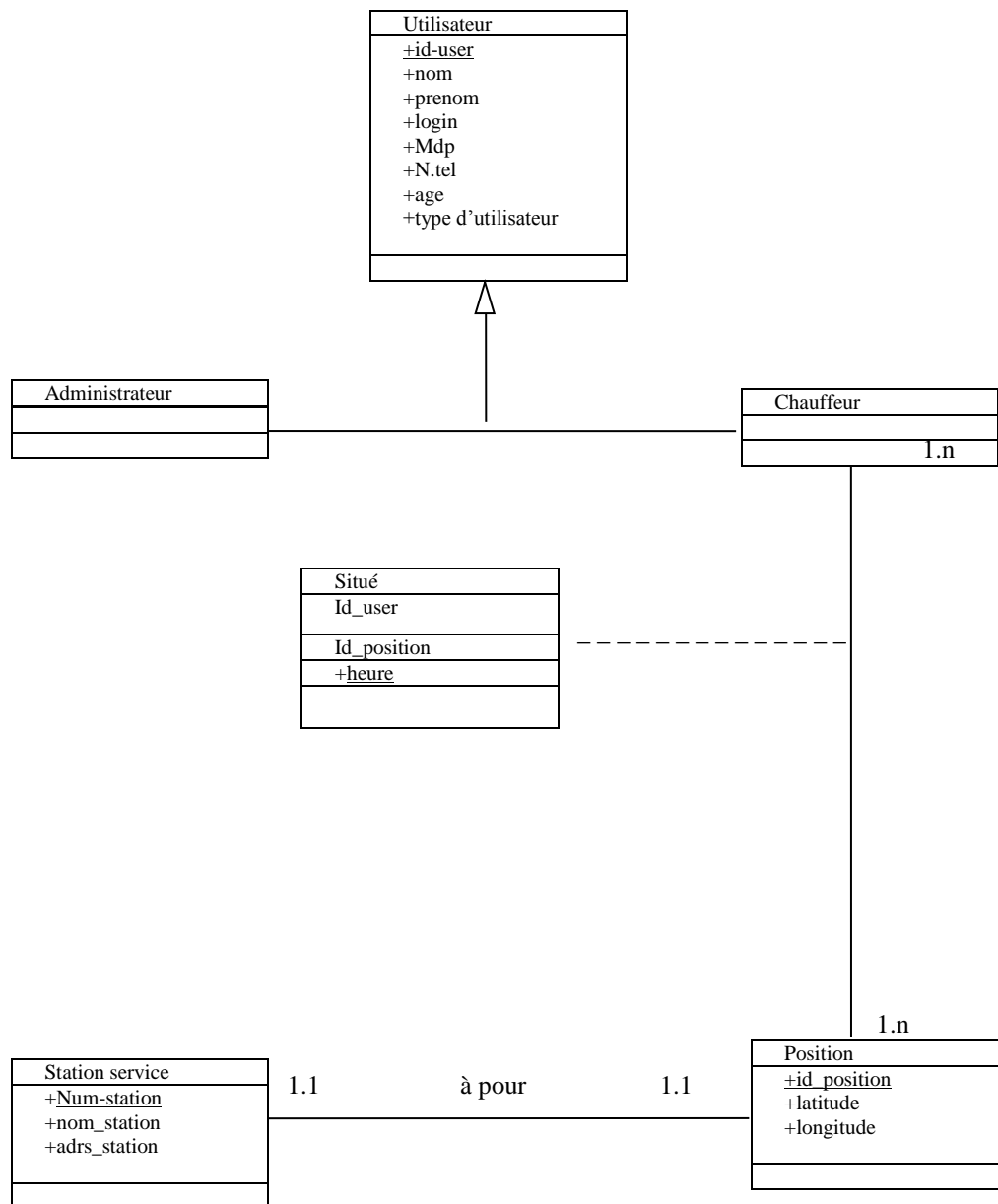
## II.6.5 Les diagrammes de classe

Le diagramme des classes représente la structure statique d'un système. Il contient principalement des classes ainsi que leurs associations. [11]

L'intérêt majeur du diagramme de classe est de modéliser les entités du système. Vu le grand nombre des cas utilisation présentés précédemment, nous allons juste présenter le diagramme de classe générale.

- Le caractère + est utilisé pour préciser la visibilité public.
- Le caractère - est utilisé pour préciser la visibilité protégé

## II.6.6 diagrammes de classe général



**Figure II.12** Diagramme de classe général

## II.7 Les tables

Dans cette partie nous avons présenté les table de notre application en suivant les règles suivantes :

1. Une classe devient une table;
2. Un identifiant d'une classe devient la clé de la table ;
3. Une classe d'association devient une table.

### II.7.1 Table utilisateur

| Codification     | Désignation                  | Type | Taille | Nature |
|------------------|------------------------------|------|--------|--------|
| Id-user          | Identifiant de l'utilisateur | AN   | 20     |        |
| nom_user         | Nom de l'utilisateur         | A    | 20     |        |
| Prenom           | Prénom de l'utilisateur      | A    | 20     |        |
| Login            | Login de l'utilisateur       | A    | 10     |        |
| Age              | Age de l'utilisateur         | N    | 2      |        |
| N° tel           | Numéro de téléphone          | N    | 15     |        |
| Mdp              | Mot de passe                 | AN   | 15     |        |
| Type_utilisateur | Type de l'utilisateur        | AN   | 20     |        |

**Tableau II.1 : Table utilisateur**

### II.7.2 Table chauffeur

| Codification | Désignation                  | Type | Taille | Nature |
|--------------|------------------------------|------|--------|--------|
| Id-user      | Identifiant de l'utilisateur | AN   | 20     |        |

**Tableau II.2 : Table chauffeur**

### II.7.3 Table administrateur

| Codification | Désignation                  | Type | Taille | Nature |
|--------------|------------------------------|------|--------|--------|
| Id-user      | Identifiant de l'utilisateur | AN   | 20     |        |

**Tableau II.3 : Table administrateur**

### II.7.4 Table station service

| Codification | Désignation                     | Type | Taille | Nature |
|--------------|---------------------------------|------|--------|--------|
| num_station  | Le numéro de la station service | N    | 4      |        |
| nom_station  | Nom de la station service       | A    | 20     |        |
| Adrs_station | Adresse de la station service   | AN   | 100    |        |
| Id_position  | Identifiant de la position      | AN   | 20     |        |

**Tableau II.4 : Table station service**

## II.7.5 Table Position

| Codification | Désignation                | Type | Taille | Nature |
|--------------|----------------------------|------|--------|--------|
| Id_position  | Identifiant de la position | AN   | 20     |        |
| Latitude     | Latitude de la station     | N    | 20     |        |
| Longitude    | Longitude de la station    | N    | 20     |        |

**Tableau II.5 : Table position**

## II.7.6 Table Situé

| Codification | Désignation                  | Type | Taille | Nature |
|--------------|------------------------------|------|--------|--------|
| Id_user      | Identifiant de l'utilisateur | AN   | 20     |        |
| Id_position  | Latitude de la position      | AN   | 20     |        |
| Heur         | Heur de la position          | N    | 20     |        |

**Tableau II.6 : Table Situé**

## II.8 Conclusion

Dans ce chapitre, nous avons introduit les fonctionnalités de notre application. Pour les atteindre, nous avons proposé une solution fondée sur une analyse et une conception modélisées à l'aide du langage de modélisation unifié UML. Pour cela, nous avons défini les acteurs de notre application, les tâches qu'ils assurent et les scénarios associés à chaque tâche. Nous avons élaboré les diagrammes de cas d'utilisation de chaque acteur, élaboré les diagrammes de séquence et le diagramme de classe générale. 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.

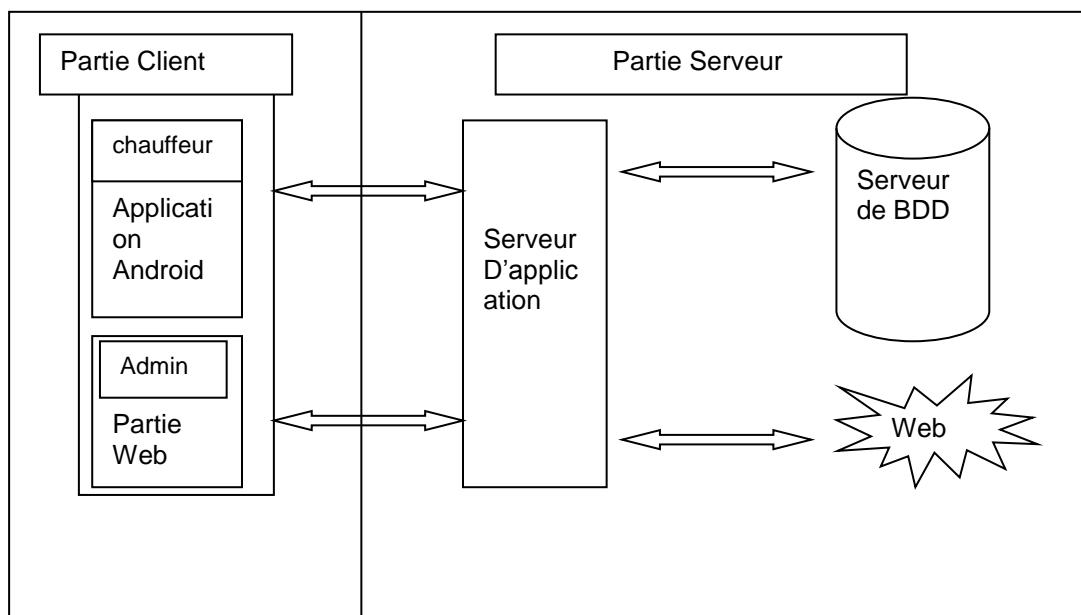
# **CHAPITRE III: REALISATION**

### III.1 Introduction

Après la présentation de l'analyse et de la conception de notre solution nous allons présenter dans ce chapitre l'environnement de développement de cette dernière ainsi que les différents outils utilisés pour sa réalisation puis expliquer son fonctionnement en présentant quelques interfaces illustratives.

### III.2 Architecture trois tiers

L'architecture de notre application est une architecture trois tiers comme le montre la figure suivante :



**Figure III.1 : architecture trois tiers**

Nous avons la partie client qui contient deux sous-partie destinées respectivement pour :

- Chauffeur : qui est l'utilisateur de l'application Android pour notamment localiser les stations service ;
- L'administrateur : qui utilise l'application web (desktop) qui s'occupe essentiellement de la gestion de la base de données.

La partie serveur contient :

- Un serveur d'application : qui gère la logique applicative ;
- Un serveur de base de données : qui s'occupe de la gestion des données de l'application.

### III.3 Environnement de travail

#### III.3.1 Environnement matériel

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

- Ordinateur Portable : Intel Core i5, 2.40 GHZ, 4,00 GO de mémoire vive, Windows 7.

#### III.3.2 Environnement cible

Notre application est destinée pour tous les Smartphones équipé d'un système Android 2.2 dite Froyo et plus.

### III.3.3 Système d'exploitation

Notre application a été développée sous le système d'exploitation Windows 8 service pack1, en langage JAVA. Elle peut être ainsi intégrée dans n'importe quel autre système d'exploitation supportant la machine virtuelle java.

### III.4 Environnement logiciel

Avant de pouvoir faire du développement, il faut nécessairement préparer l'environnement de travail. Dans notre cas, on a choisi de faire notre développement mobile sur la plateforme Android studio. Pour la préparer, on a effectué différentes tâches qui sont :

- Installation d'Android studio.
- Installation SDK Android.
- Intégration SDK Android pour pouvoir créer des projets Android.
- Installation eclipse avec extension PY\_DEV.

Après la préparation de logiciel de travail android studio et pour pouvoir développer une application dans de bonnes conditions, il faut bien savoir choisir son environnement de développement selon les besoins, de ce fait, on a développé une application pour mobile en JAVA, qui va être compilé en fichier DEX par le Dalvik VM ensuite exécuter avec l'Emulateur d'Android.

### III.5 Les langage de programmation utilisés

Nous avons utiliser les langages de programmation suivant :

#### A.Le langage JAVA

Est un langage de programmation orienté objet mis au point par Sun Microsystems (Oracle). Sa caractéristique principale est qu'il est indépendant de toute plate-forme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une «Java Virtual Machine » (JVM).

Ce concept est à la base du slogan de Sun pour java : WORA (Write Once, RunAnywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java Développement Kit).

Java est caractérisée aussi par la réutilisabilité de son code ainsi que la simplicité de sa mise en œuvre[12].

#### B. Le langage Python

Est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl[13].

#### C. JavaScript

Est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs.

#### D.Les CSS

Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.).[14]

### III.6 Les outils de programmation utilisés

#### A. Serveur d'Application.

Le serveur HTTP utilise est un serveur/Framework open source développe par FriendFeed (racheté par Facebook), c'est un serveur asynchrone non-bloquant, il est parmi les serveurs qui ont dépassé le problème des 10kb, il n'est pas officiellement compatible avec Windows, mais il est fonctionnelle.

#### B. SGBD.

Python étant un langage dynamiquement type, et la formation universitaire acquise



contenant la logique du SQL, et surtout, avoir une solution qui soit compatible avec le système d'exploitation sans avoir à rajouter d'autres couches pour la compatibilité; la plupart des SGBD ont été initialement développés pour Linux (performance et open source), donc pour l'adapter sur Windows, il faut installer *CyGwin* ce qui est déconseillé pour la production et conseillé uniquement pour le développement et les phases de petits tests, donc il fallait choisir une solution NoSql qui soit adaptable à ces contraintes, MongoDB est la solution parfaite pour ce problème.

### C. MongoDB.

MongoDB est un SGBD orienté document, le nom vient de *huMONGOus* qui veut dire énorme.

MongoDB utilise BSON (Binary-Son) pour la sauvegarde des données, BSON est une version améliorée de JSON vu qu'il contient d'autres formats de données : type Date et type BinType (Binary Type).

MongoDB est considéré le SGBD qui se rapproche plus du monde du SQL.

Le document MongoDB a la taille maximale de 16 Méga pour la version actuelle (2.4.3) et peut varier dans le futur, mais vaut mieux utiliser GridFS pour la sauvegarde des fichiers binaires (images, son, ...)[15].

### D. Software Development Kit (SDK)

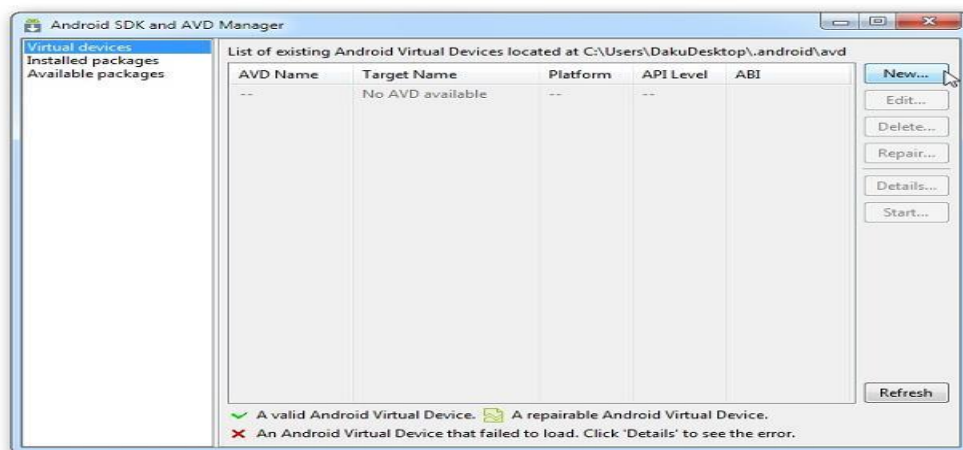
L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.).

Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google tels que Maps etc.) ou de la documentation JavaDoc. Son fonctionnement est similaire aux gestionnaires de paquets de Linux.

Le SDK Android est composé de plusieurs éléments pour aider les développeurs à créer et à maintenir des applications :

- Des API (interfaces de programmation) ;
- Des exemples de code ;
- De la documentation ;
- Des outils – parmi lesquels un émulateur – permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application .

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



**Figure III.2:** Interface d'installation du SDK Android

## E. Android

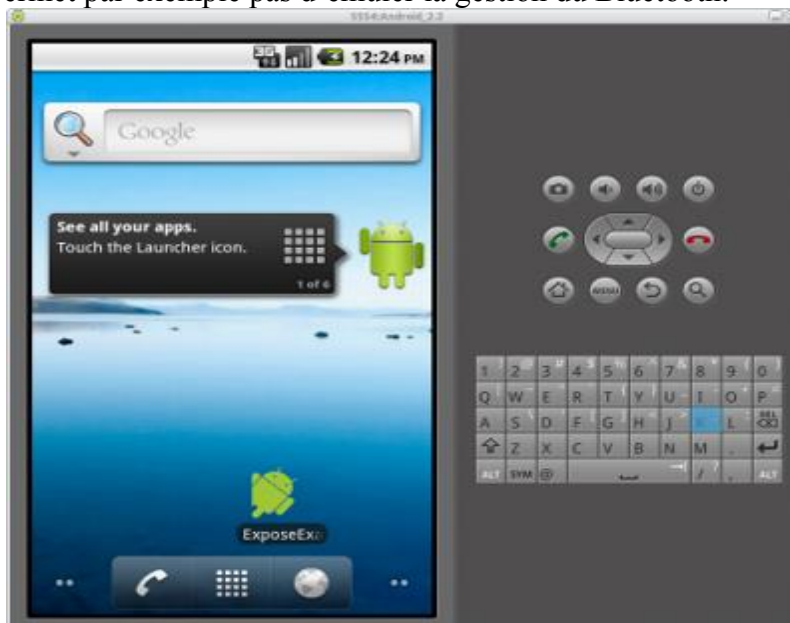
Android est un système d'exploitation open source pour Smartphones, PDA et terminaux mobiles. Il s'agit également d'un langage de programmation basé sur le JAVA et sur le XML. Le JAVA permet d'interagir avec l'utilisateur en faisant la liaison entre l'interface graphique et la base de données. Quant au XML, il permet notamment de décrire les interfaces graphiques.

## F. Emulateur

Evoqué plus haut, le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android.

C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel.

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



**Figure III. 3**Emulateur d'Android

## G. Android studio

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA.

Il permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément

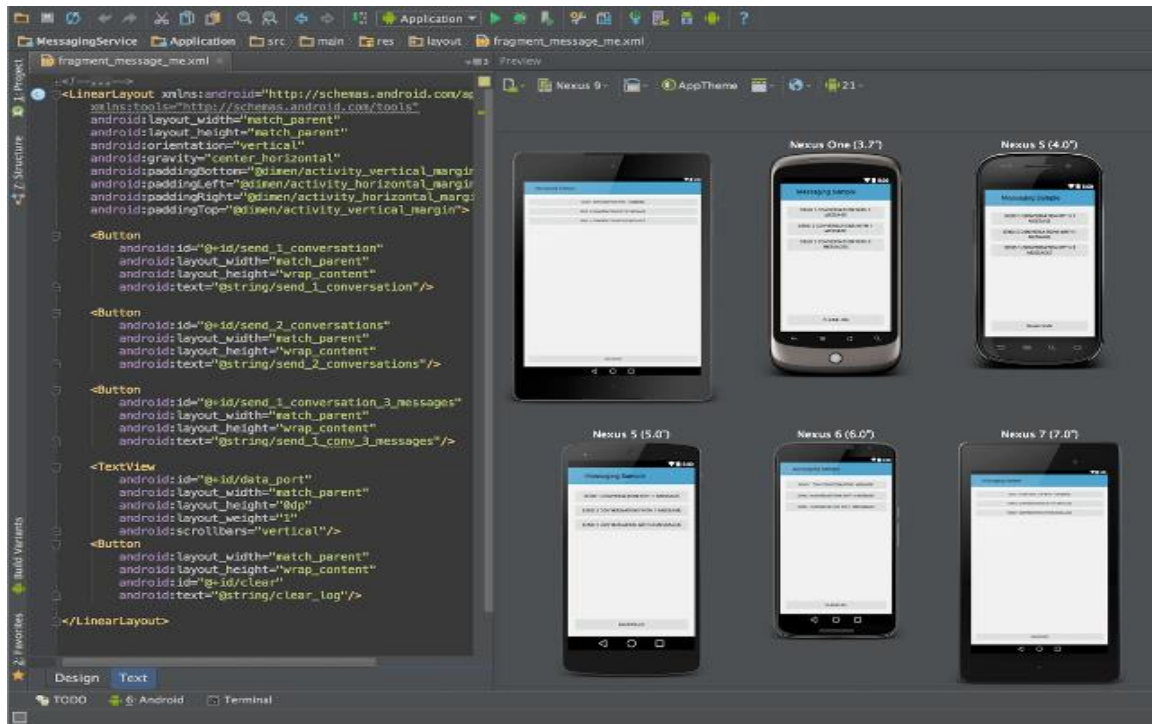


Figure III.4 Android studio

### III.7 Présentation de quelques interfaces de notre application

Dans cette partie de ce chapitre nous avons présenté quelques interfaces de notre application

#### III.7.1 Lancement de l'application

En premier lieu le chauffeur lance l'application en appuyant sur l'icône lui correspondant dans le menu du Smartphone.

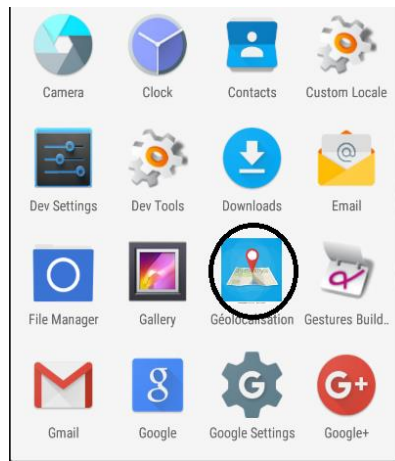
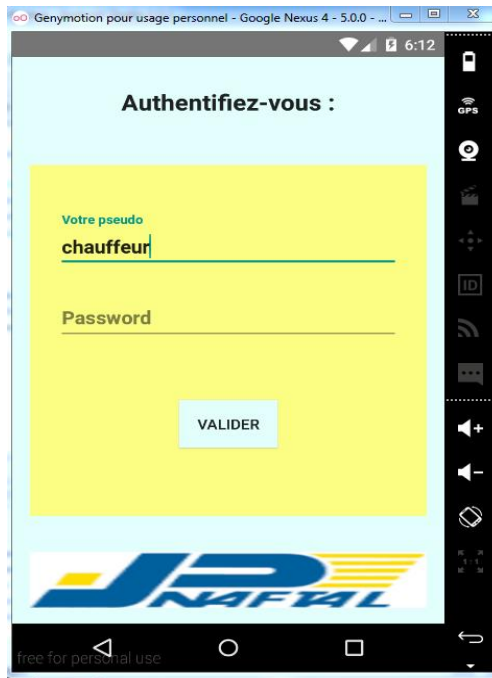


Figure III.5 Icône de l'application sous android

### III.7.2 Authentification de chauffeur

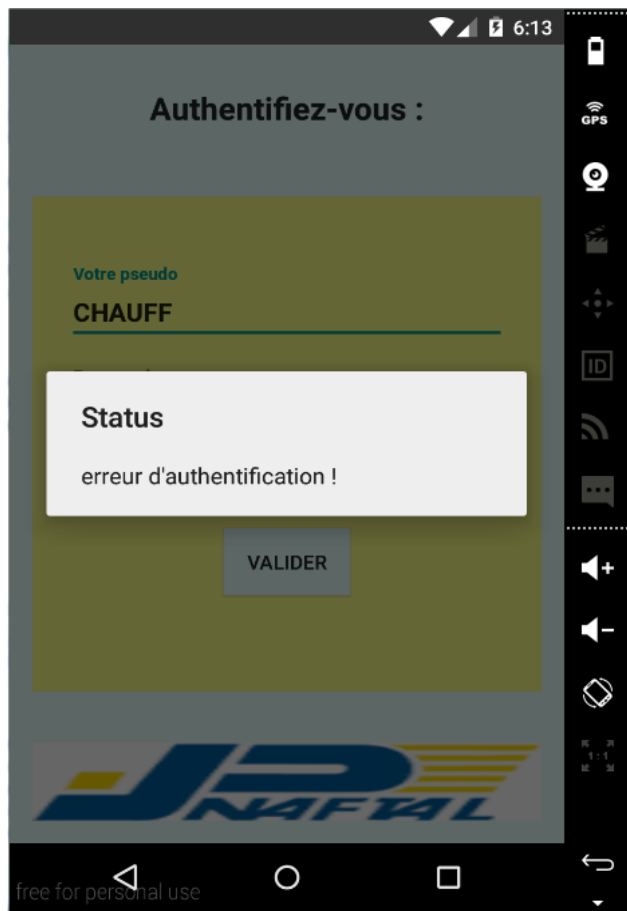
Le chauffeur doit s'authentifier pour accéder a son compte sur l'application ,  
Et pour cela il doit avoir son pseudo et son mot de passe attribué par l'administrateur  
Puis il clique sur le boton valider .



**Figure III.6 :** Interface d'authentification chauffeur

### III.7.3 Interface Erreur d'authentification

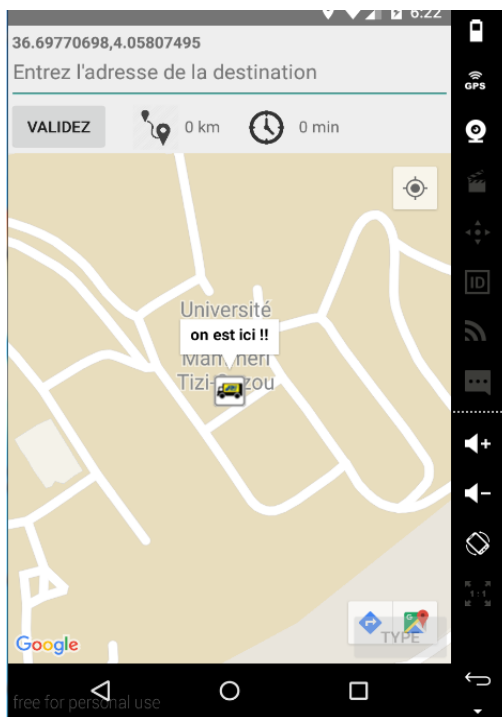
L'interface suivante montre que le chauffeur a mal saisi ses coordonnées, Donc un message d'information lui apparait et lui demande de vérifier son pseudo et mot de passe.



**Figure III.7 :** Interface erreur d'authentification

### III.7.4 Interface localisation chauffeur

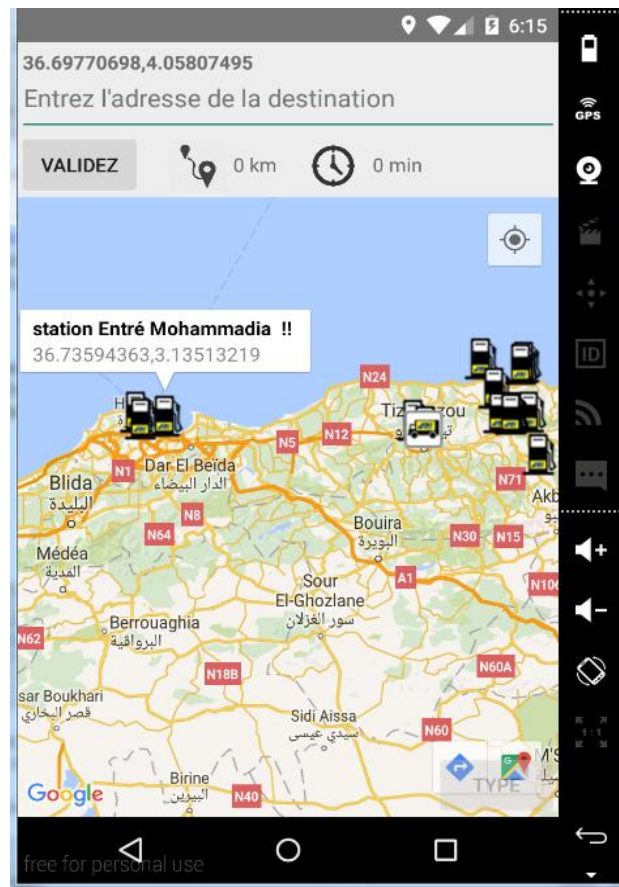
Cette interface permet de visualiser clairement la position initiale du chauffeur en temps réel avec ses coordonnées GPS après localisation par l'application .



**Figure III.8 :** Interface localisation chauffeur

### III.7.5 Interface des stations service

Dans cette interface l'application permet au chauffeur de voir toutes les stations services disponibles avec leurs noms et leurs coordonnées GPS, il peut ensuite choisir la station service avec un clic sur son icône, ou bien saisir directement l'adresse de destination pour la visualiser.



**Figure III.9 :** Interface des stations service

### III.7.6 Interface recherche d'itinéraire

Dans cette interface on peut voir apres validation de la destination (station service) que les coordonnées GPS sont visible dans la zone de saisie de texte (« entrer l'adresse de destination »), un message qui nous montre que l'application cherche un itinéraire ( le plus proche).

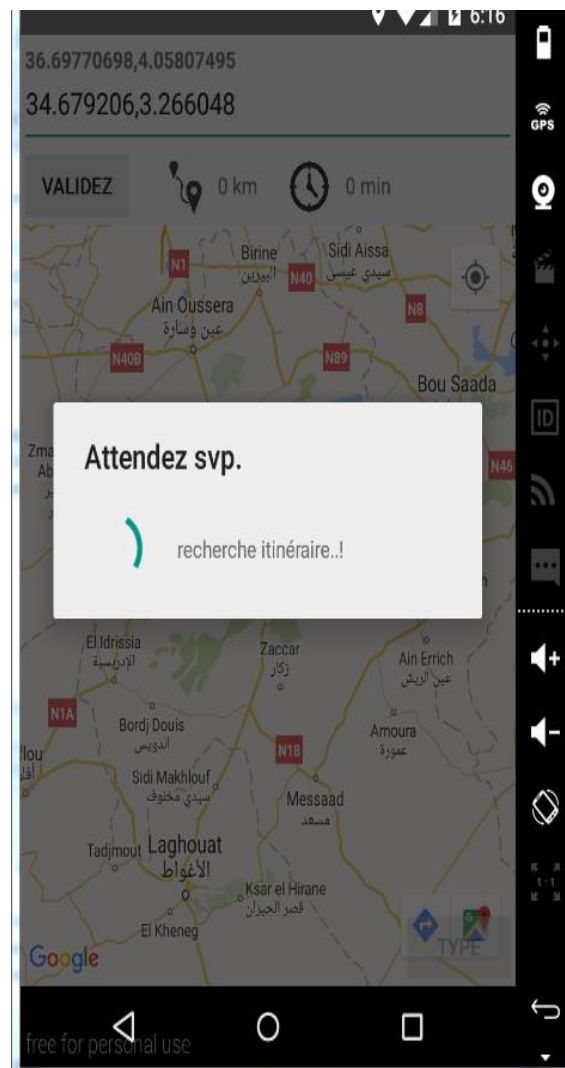
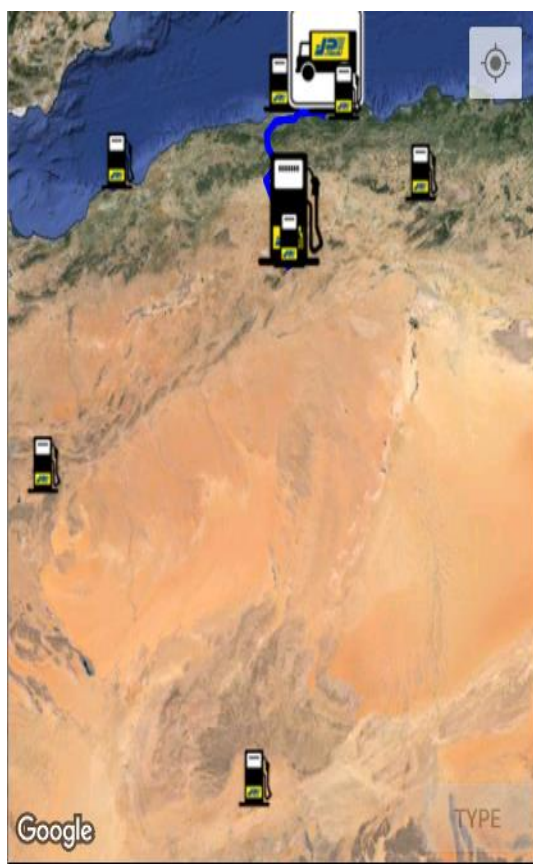


Figure III.10 : Interface recherche d'itinéraire



### III.7.7 Interface de visualisation d'itinéraire en mode satellite

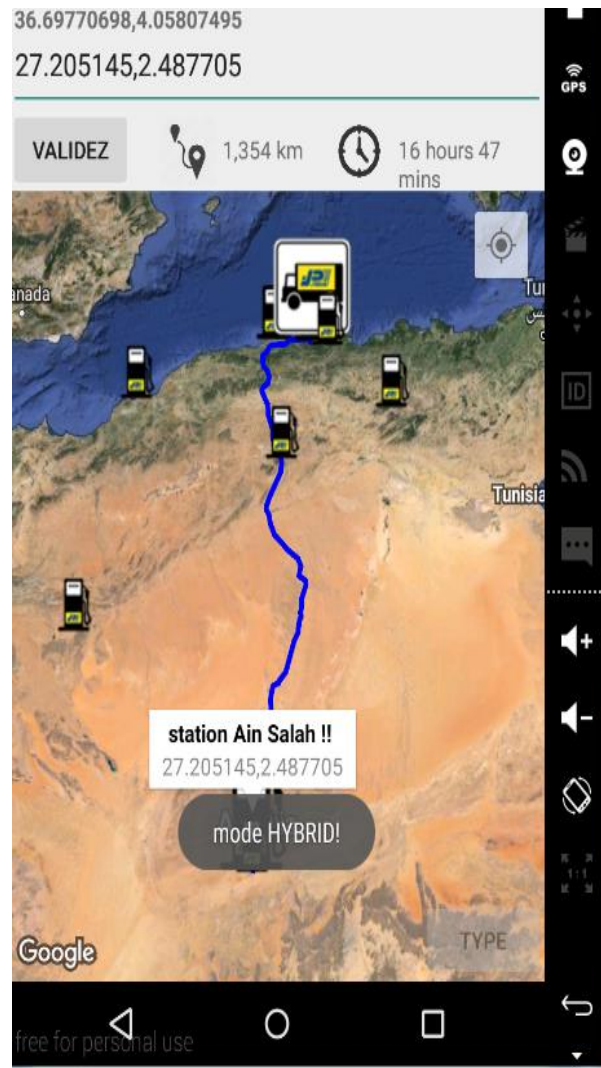
Dans cette interface on peut voir que apres la validation de son choix par le chauffeur L'itinéraire le plus proche aparait depuis le point de depart du chauffeur qui est representé par l'icône du camion et l'adresse de destination representée par une icône de stations service plus grande avec une map de type satellite .



**Figure III.11 :** Interface de visualisation d'itinéraire en mode satellite

### III.7.8 Interface de visualisation d'itinéraire en mode hybrid

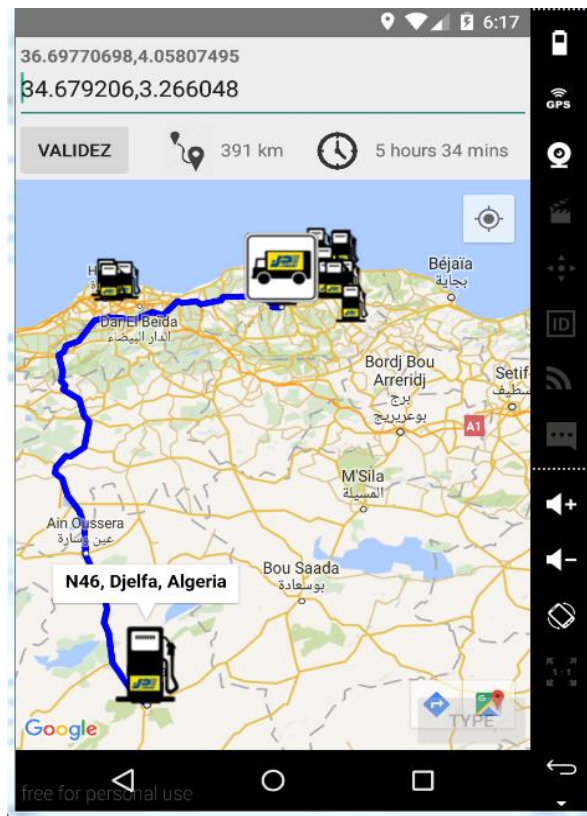
L'interface nous montre clairement le parcours à suivre depuis notre localisation jusqu'à la destination avec plusieurs informations supplémentaires comme la distance entre ces deux points et la durée estimée avec une carte de type hybride (mixture entre type normal et satellite).



**Figure III.12 :** Interface de visualisation d'itinéraire en mode hybride

### III.7.9 Interface de visualisation d'itinéraire en mode normal

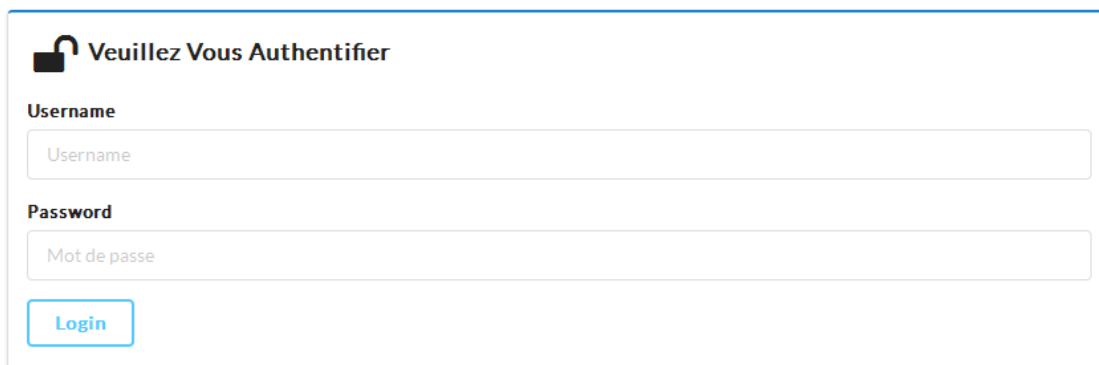
Toujours la même interface avec les informations précédentes, l'adresse de départ, l'adresse de destination, la distance entre les deux points, la durée estimer pour y parvenir, l'itinéraire le plus proche (illustrer en bleu ) avec une carte de type Normal (classique).



**Figure III.13** : Interface de visualisation d'itinéraire en mode normal

### III.7.10 Interface authentification administrateur

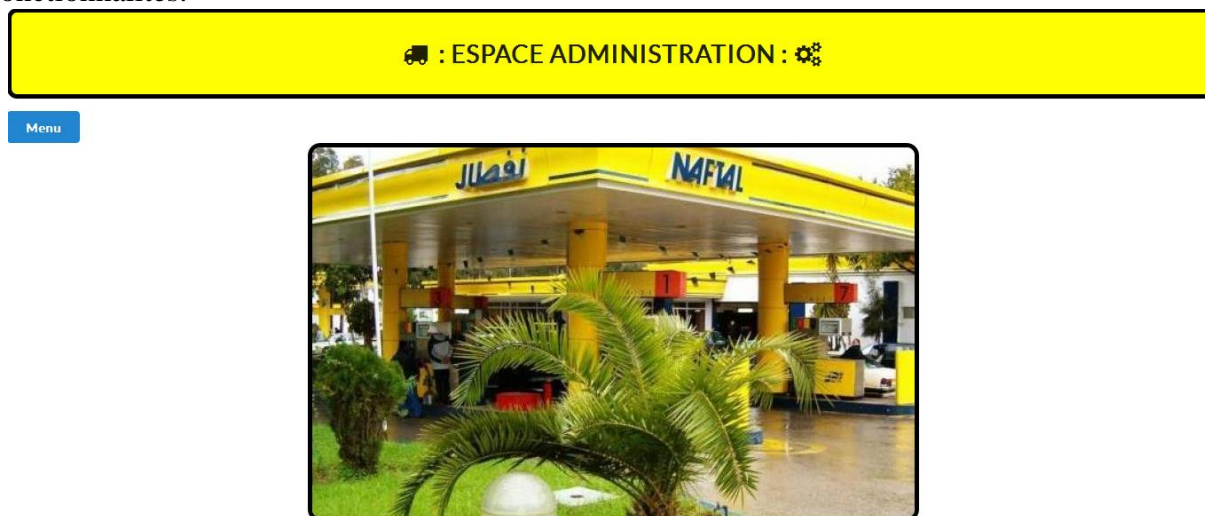
L'interface de l'application web (administrateur) qui permet aux administrateurs de saisir Leur pseudo (username) et mot de passe (password) pour s'authentifier et être dirigé vers leur espace personnel, l'interface suivante montre l'interface d'authentification



**Figure III.14 :** Interface authentification de l'administrateur

### III.7.11 Interface espace administrateur

L'interface suivante illustre l'espace administrateur qui lui permet d'accéder au menu de l'application avec le bouton (menu) en haut à gauche de l'interface ou il va trouver plusieurs fonctionnalités.



**Figure III.15:** Interface espace administrateur

### III.7.12 Interface menu administrateur

Après avoir cliquer sur le menu, l'administrateur peut utiliser les fonctionnalités a savoir :

- Gérer les chauffeurs (ajouter, etc) ;
- Gérer les stations-services (ajouter, supprimer, etc).

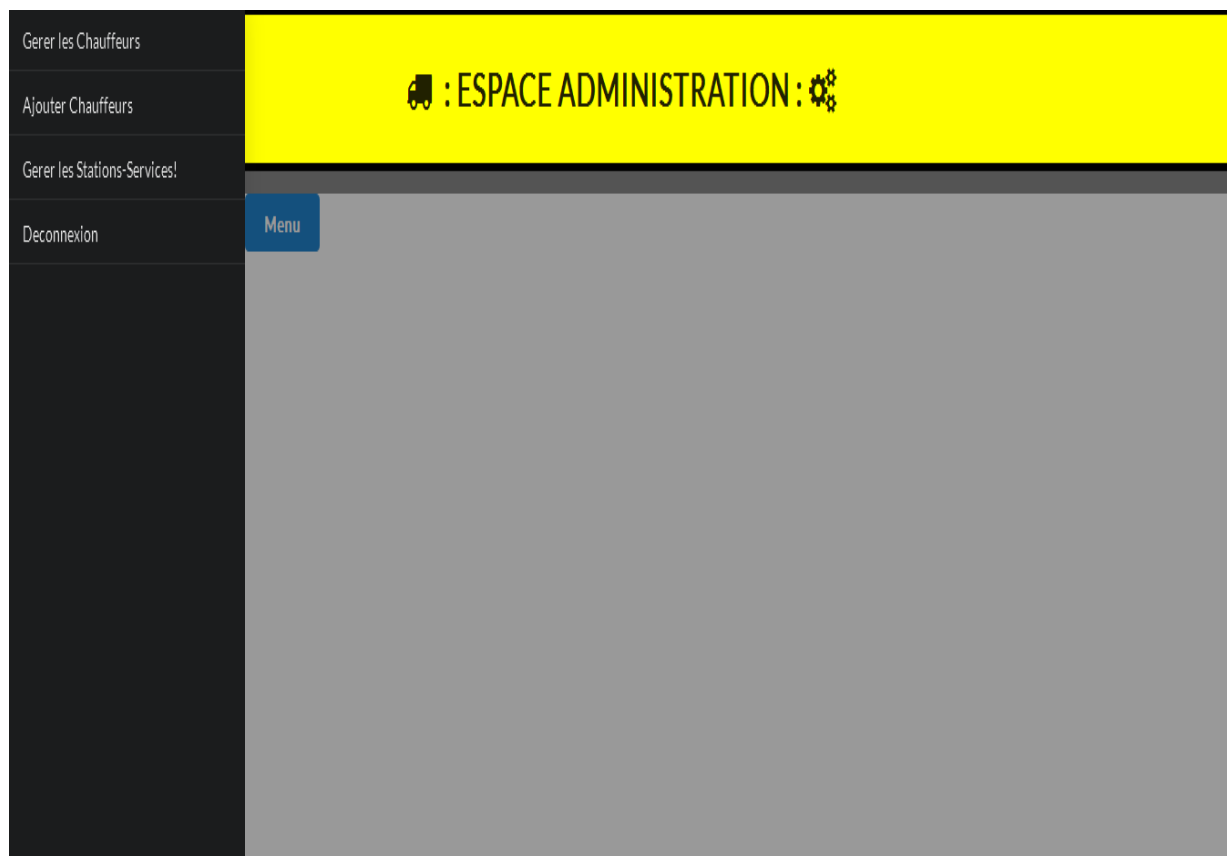


Figure III.16: Interface menu administrateur

### III.7.13 Interface gestion des stations service

Cette interface permet a l'administrateur d'ajouter les stations services avec un nom de la stations et une description ,et pour cela il a deux possibilités soit :

Il selectionne directement sur la carte un points précis ,

Il ajoute avec les coordonnées GPS (latitude et longitude) dans leur champs puis un clic sur « mettre a jour sur la carte » pour positionnée le marqueur sur la carte .

Puis un clic sur le bouton « ajouter station »

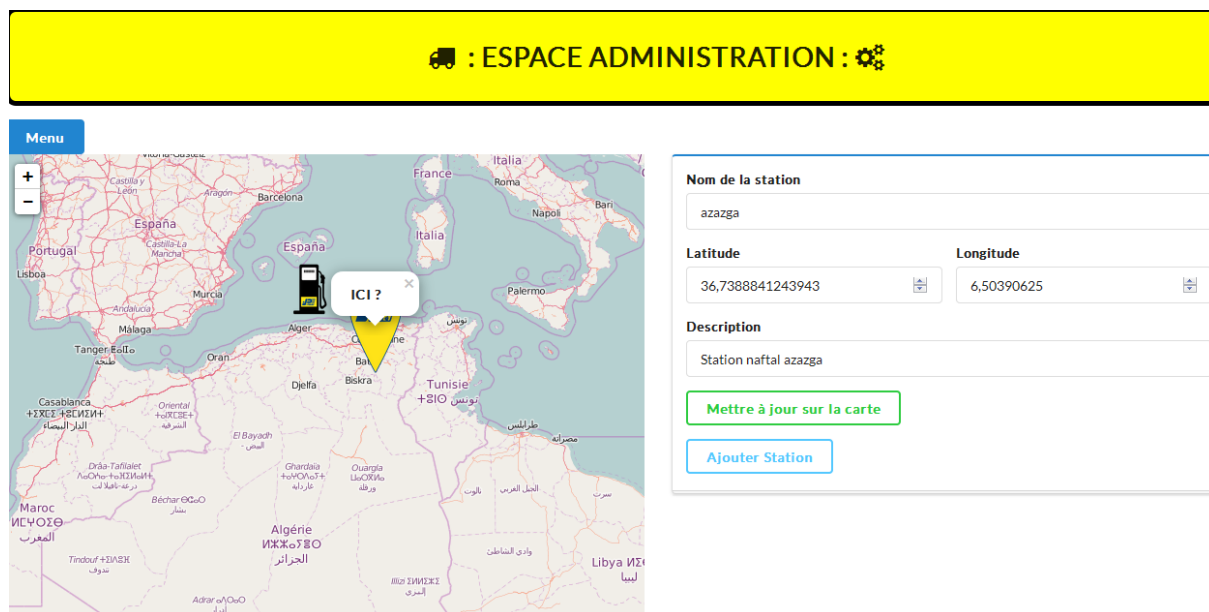


Figure III.17 : Interface gérer les stations

### III.7.14 Interface gérer les chauffeurs

Cette interface nous montre l'espace administrateur qui permet a l'administrateur :

De voir tous les chauffeurs ,

De supprimer un ou plusieurs chauffeur ,

Et lui donne la possibilités d'ajouter un chauffeur avec le bouton « ajouter chauffeur! » qui le redérige vers la page ajouter chauffeur .

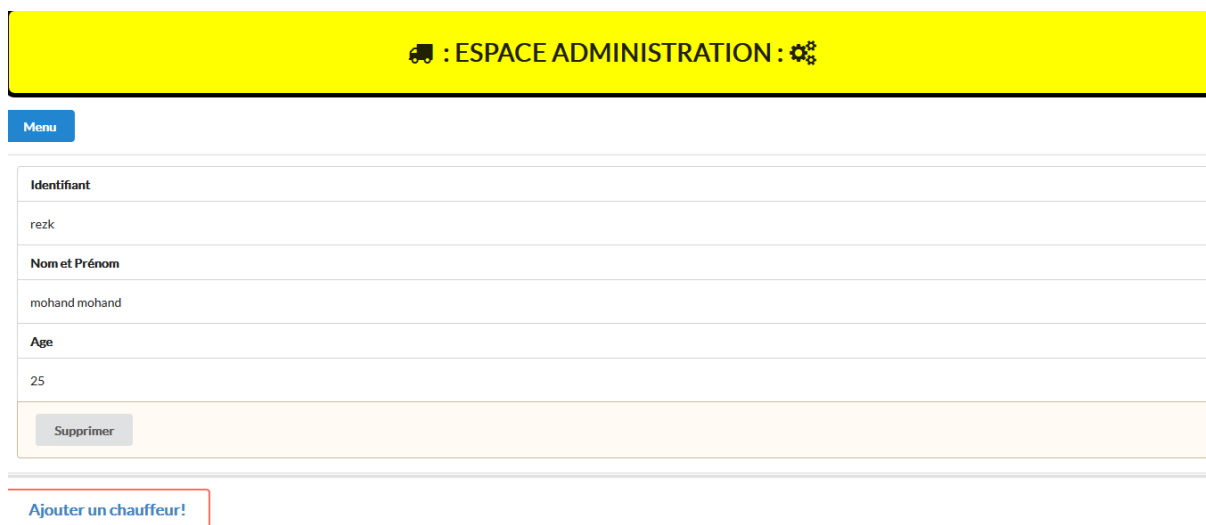
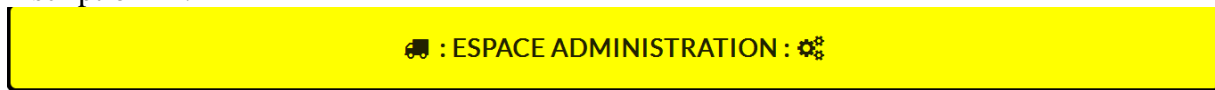


Figure III.18 : Interface gérer les chauffeurs

### III.7.15 Interface ajouter chauffeur

Dans cette interface on voit comment l'administrateur peut ajouter un chauffeur, Et ceci en remplissant tous les champs qui sont obligatoires puis cliquer sur «valider votre inscription» .



Menu

**Veillez Ajouter un chauffeur**

**Pseudo**  
Exemple: test, test1264, test-1264, test\_1264

**Mot de passe** **Confirmer le mot de passe**  
6 char min Confirmer

**Nom** **Prénom** **Age**  
Le nom du chauffeur Le prénom du chauffeur L'âge du chauffeur

Valider votre inscription

**Figure III.19 : Interface ajouter chauffeurs**

### III.8 Conclusion

Ce troisième chapitre a été consacré à la phase réalisation de notre application. On y a défini tout d'abord l'environnement de développement, puis illustré les différentes interfaces qui la compose.

# **Conclusion Générale**



## **Conclusion générale**

Nous avons, au cours de ce mémoire, conçu et réalisé une application Android pour la géolocalisation des stations service pour la société nationale NAFTAL .

Pour cela,nous avons étudié le système Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture d'Android, à savoir les principaux composants du système.

Après avoir entamé l'analyse et la conception de notre système en utilisant le langage de modélisation UML ainsi que la conception, nous sommes passés à la réalisation.

Au cours de la réalisation de ce projet nous avons enrichi notre savoir et développé nos connaissances informatiques, notamment dans le domaine de la programmation et de la mobilité. En effet, l'application a exigé des connaissances du langage JAVA,PYTHON, etc et des outils de développement indispensables à sa réalisation. La mise en œuvre de notre travail a exigé des connaissances très approfondies en la matière ainsi qu'une bonne maîtrise de la configuration d'Android studio et de l'environnement Android.

Et enfin, notre travail ainsi présenté reste un prototype sur lequel nous espérons apporter plus de connaissance pour le rendre meilleur et plus fonctionnel.

# **BIBLIOGRAPHIE**

## Bibliographie

[1]

[http://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/3265/3265-android-howto-1\\_0.pdf](http://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/3265/3265-android-howto-1_0.pdf)

[2] <https://cordova.apache.org/docs/fr/latest/guide/platforms/android/>

[3] [https://fr.wikipedia.org/wiki/Historique\\_des\\_versions\\_d%27Android](https://fr.wikipedia.org/wiki/Historique_des_versions_d%27Android)

[4] [http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi\\_comp.html](http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_comp.html)

[5] <http://www.definitions-marketing.com/definition/application-android/>

[6] <http://www.journaldunet.com/developpeur/outils/outils-de-developpement-android/>

[7] <http://uml.developpez.com/livres/index/?page=Les-livres-francais>

[8] [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

[9] [https://fr.wikipedia.org/wiki/Diagramme\\_des\\_cas\\_d%27utilisation](https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation)

[10] [https://fr.wikipedia.org/wiki/Diagramme\\_de\\_s%C3%A9quence](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence)

[11] [https://www/wiki.com/wiki/Diagramme\\_de\\_classes](https://www/wiki.com/wiki/Diagramme_de_classes)

[12] [https://fr.wikipedia.org/wiki/Java\\_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))

[13] [https://fr.wikipedia.org/wiki/Python\\_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage))

[14] <https://fr.wikipedia.org/wiki/JavaScript>

[15] <https://github.com/bitly/asyncmongo>

[16] <http://mobiles.developpez.com/>

[17] <http://www.volle.com/travaux/umlimp.htm>

<https://github.com/mongodb/motor>

<https://github.com/marcelnicolay/mongotor>

[http://www.isima.fr/~lacomme/DevPortable/tutorial\\_android.pdf](http://www.isima.fr/~lacomme/DevPortable/tutorial_android.pdf)

[http://www.univ-orleans.fr/lifo/Members/Jean-](http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pd)

[Francois.Lalande/enseignement/android/cours-android.pd](http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pd)

[http://www.inforef.be/swi/download/python\\_notes.pdf](http://www.inforef.be/swi/download/python_notes.pdf)