

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE.

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE



Mémoire

Pour l'obtention d'un diplôme de Master II en Systèmes Informatique



Conception et réalisation d'une application de géolocalisation

android

Dirigé par :

Mr H.Taleb

Réalisé par :

TATOULTE Hassen

LOMANI Jugurtha

2012/2013

Remerciements

Nous remercions le Bon dieu de nous avoir donné le courage et la volonté pour la réalisation de notre projet.

Nous tenons à remercier notre promoteur Mr TALEB.H pour sa disponibilité, son aide, ses conseils précieux, ses critiques constructives, ses explications et ses suggestions pertinentes.

Nous tenons à exprimer toute notre gratitude aux membres du jury pour avoir accepté d'évaluer et de juger notre travail.

Nous tenons aussi à remercier Mr SADOUS pour son soutien, son aide, et ses conseils.

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

DEDICACE

Je dédie ce modeste travail A ceux qui ont éclairé ma vie, ma mère et mon cher père pour leur aide et leur soutien tout au long de mes études, et qui ont fait de moi ce que je suis aujourd'hui et j'espère qu'un jour je serai capable de leurs donner au moins la moitié car quoi qu'on face on arrivera jamais à leurs rendre tout.

A mes chers frères, et ma chère sœur.

A tous mes amis (e)s, et particulièrement Kathia, jugurtha, Bouji, Sofiane, Samir, Ghiles, Tina, Ghania, Saloua, Ouiza, Chaveha

Sans oublier mes camarades de la promotion 2013.

Dédicace

« Louange à Dieu, le seul et unique »

Je dédie ce modeste travail A

Mes très chers parents

Témoignages d'affection et de grande reconnaissance

Mon très cher frère

Et a toute la famille

Tous mes amis (es).

Jugurtha

Sommaire



Sommaire

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

Chapitre I : La géolocalisation

I. Introduction.....	3
II. La géolocalisation.....	4
II.1. Définitions.....	4
II.2. Techniques de géolocalisation.....	4
II.2.1. Géolocalisation par satellites.....	4
II.2.2. Géolocalisation par GSM.....	5
II.2.3. Géolocalisation par WiFi.....	7
II.2.4. Combinaison des différentes techniques.....	7
II.2.5. Géolocalisation indoor.....	8
III. Plateformes logiciels du géolocalisation.....	9
III.1. Les composants.....	9
III.2. Architecture temps réel.....	10
IV. Types de terminaux existants.....	10
IV.1. Mode de fonctionnement.....	10
a. Data pullers.....	10
b. Data pushers.....	10
IV.2. Mode de communication.....	11
a. Terminaux GSM/GPS.....	11
b. Terminaux satellites.....	11
V. Exemples de services géolocalisés.	12
VI. Principes du GPS.....	13
VI.1. Définition.....	13
VI.2. Les composants d'un GPS.....	13
a. la partie spatiale.....	13
b. la partie de contrôle.....	14
c. la partie utilisateur.....	15
VI.3. Principes de fonctionnement d'un GPS.....	15
VI.4. Principes de positionnement d'un GPS.....	18

Sommaire

VII. Conclusion.....	21
-----------------------------	-----------

Chapitre II : Généralité sur le système ANDROID

I. Introduction.....	22
II. Le système d'exploitation ANDROID.....	23
II.1. Définition.....	23
II.2. Historique.....	23
II.3. Caractéristique.....	24
II.4. Architecture Android.....	25
a. Le noyau linux.....	26
b. La bibliothèque pour android.....	26
c. Le frameworks pour les applications.....	27
d. Les applications.....	27
II.5. Les outils.....	27
II.6. Application proposées.....	29
III. Composantes d'une application Android.....	29
a. Activité.....	30
b. Service.....	30
c. Broadcast Receive.....	30
d. Content providers.....	30
IV. Cycles de vie d'une activité.....	30
V. Développement d'une application Android.....	32
VI. Les outils et techniques de geolocalisation et d'orientation sous Android.....	33
a. Google Maps API.....	33
b. Le GPS.....	34
c. Android Radar.....	34
d. Android Compass.....	34
VII. Conclusion.....	35

Sommaire

Chapitre III : Analyse et conception

I. Introduction.....	36
II. Présentation de l'UML.....	36
II.1. Idée de base.....	36
II.2. Modélisation avec L'UML.....	36
II.3. La démarche de modélisation avec L'UML.....	37
III. Analyse.....	37
III.1. Identification des besoins.....	37
III.2. Identification des acteurs.....	38
III.3. Spécification des tâches.....	38
III.4. Spécification des scénarios.....	39
III.5. Les cas d'utilisation.....	39
IV. Conception.....	41
IV.1. Le niveau applicatif.....	42
IV.1.1. Le diagramme des cas d'utilisation.....	42
IV.1.2. Les diagrammes de séquences.....	43
IV.1.3. Les diagrammes d'activités.....	45
IV.1.4. Les diagrammes de classe.....	47
IV.2. Le niveau donné.....	50
IV.2.1. Schéma de la base de données.....	50
IV.2.2. Les table.....	51
IV.3. Conception des maquettes préliminaires.....	52
IV.3.1. L'écran d'accueil.....	52
IV.3.2. L'écran de l'espace personnel.....	53
IV.3.3. L'écran de la liste des véhicules.....	54
V. Conclusion.....	54

Sommaire

Chapitre IV : Réalisation et mise en œuvre.

I. Introduction.....	55
II. Environnement de travail.....	55
II.1. Environnement matériel.....	55
II.2. Environnement cible.....	55
II.3. Environnement logiciel.....	55
a. La plate forme Eclipse.....	55
b. La plate forme WampServer.....	56
II.4. Eclipse INDIGO.....	56
II.5. Le plugin ADT.....	57
II.6. Software Development Kit (SDK).....	57
II.7. Le SGBD MySQL.....	57
III. Langages de programmation.....	58
III.1. Le langage coté client.....	58
III.2. Le langage coté serveur.....	58
IV. Fonctionnement de notre application.....	59
IV.1. L'écran d'authentification.....	59
IV.2. L'écran d'accueil d'utilisateur.....	60
IV.3. L'écran de la liste des véhicules.....	61
IV.4. L'écran de géolocalisation d'un véhicule.....	62
IV.5. L'écran de modification de mot de passe	63
V. Conclusion.....	64
 Conclusion générale.....	 65
Annexe A : Manuel d'installation et de mise en marche.....	67
Annexe B : Obtention d'une Clé pour utiliser google Maps.....	68
Glossaire.....	70
Bibliographie.....	71

Table des illustrations



Table des illustrations

Les figures :

Chapitre I : La géolocalisation

Figure I.1 : Géolocalisation par satellite.....	5
Figure I.2 : Positionnement par TAO.....	6
Figure I.3 : Positionnement par Cell ID.....	7
Figure I.4 : Géolocalisation GPS/GSM.....	8
Figure I.5 : Géolocalisation indoor.....	9
Figure I.6 : La partie spatiale.....	14
Figure I.7 : La partie de contrôle	15
Figure I.8 : La partie utilisateur.....	15
Figure I.9 : Fonctionnement du GPS.....	16
Figure I.10 : Positionnement de l'observateur a l'intersection des 3 satellites.....	17
Figure I.11 : Positionnement de l'observateur dans la zone commune des 3 satellites.....	17

Chapitre II : Généralité sur les systèmes ANDROID

Figure II.1 : Architecture d'Android.....	26
Figure II.2 : Le concept de machine virtuelle java.....	28
Figure II.3 : l'émulateur.....	29
Figure II.4 : le cycle de vie d'une activité Android.....	31

Chapitre III : Analyse et conception

Figure III.1 : La démarche adoptée pour la modélisation.....	37
Figure III.3 : Cas d'utilisation « Afficher la liste des véhicules ».....	40
Figure III.4 : Cas d'utilisation « Accéder à la partie localisation ».....	40
Figure III.5 : Cas d'utilisation « Modifier le mot de passe».....	41
Figure III.6 : Diagramme de cas d'utilisation générale.....	42

Table des illustrations

Figure III.7: Diagramme de séquence de cas d'utilisation « localiser un véhicule».....	44
Figure III.8: Diagramme de séquence de cas d'utilisation « modifier le mot de passe».....	45
Figure III.9: Diagramme d'activité de cas d'utilisation « Localiser un véhicule ».....	46
Figure III.10 : Diagramme d'activité de cas d'utilisation «Changer le mot de passe ».....	47
Figure III.9 : Diagramme de classe.....	49
Figure III.10 : Le schéma de la base de données.....	50
Figure III.15 : Page d'accueil.....	53
Figure III.16 : Page espace personnel.....	53
Figure III.17 : Interface de la liste des véhicules.....	54

Chapitre IV : Réalisation et mise en œuvre

Figure IV.1 : Plate forme « Eclipse ».....	56
Figure IV.2 : Plate forme client/server.....	58
Figure IV.3 : Interface d'authentification	59
Figure IV.4 : Interface d'accueil.....	60
Figure IV.5 : Interface de la liste des véhicules.....	61
Figure IV.6 : Interface de géolocalisation d'un véhicule.....	62
Figure IV.7 : Interface de modification de mot de passe.....	63

Table des illustrations

Les tableaux :

Chapitre II : Généralité sur les systèmes ANDROID

Tableau II.1 : Les Caractéristiques du système d'exploitation Android.....	25
---	----

Chapitre III : Analyse et conception

Tableau III.1 : Spécification des tâches.....	38
Tableau III.2 : Spécification des scénarios.....	39
Tableau III.11 : Table Client.....	51
Tableau III.12 : Table véhicule.....	51
Tableau III.13 : Table Trame.....	52

Introduction Générale



Introduction :

L'utilisation étendue des réseaux téléphoniques mobiles a donné naissance naturellement à l'idée de fournir des services en fonction de la position de l'utilisateur: un besoin de localisation est donc apparu. L'information sur la position a été, dans un premier temps, utilisée pour la sécurité des usagers (par exemple, l'assistance aux personnes en danger). Elle a, ensuite, servi pour une utilisation optimale des services proposés sur le réseau tels que l'aide à la navigation, le tracking de véhicules. La qualité de ces services est étroitement liée à la précision de positionnement. Un intérêt considérable s'est alors développé au sein des communautés scientifiques et industrielles pour les techniques de localisation.

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

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

L'objectif de notre travail est de développer une application de géolocalisation de véhicules en temps réel sous android pour les Smartphones.

L'objectif principal de l'application est de permettre à l'utilisateur de localiser un de ses véhicules au choix, et d'avoir sa position sur la Map.

Pour cela on a structuré le travail en quatre chapitres :

- Le premier chapitre s'intitule « Géolocalisation », Ce chapitre est une introduction au service de géolocalisation et aux différentes méthodes utilisées.
- Le deuxième chapitre s'intitule « Généralité sur le système Android », dans ce chapitre on a définie les caractéristiques et l'historique de système android.

- Le troisième chapitre sous le nom « analyse et conception », est consacré à l'analyse et la conception de notre application, afin de réaliser l'application mobile. Pour cela on a fait recours au langage UML étant le mieux adapté pour les applications mobiles.
- Le quatrième est le dernier chapitre « Réalisation et implémentation », comporte la présentation de l'environnement dont lequel notre application a été réalisée, les outils utilisés et quelques interfaces de notre application.

Chapitre I

La géolocalisation



I. Introduction :

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

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

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

II. La géolocalisation :

II.1. Définition :

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. [1]

II.2. Technique de géolocalisation :

II.2.1. Géolocalisation par satellite :

La géolocalisation par satellite consiste à calculer, grâce aux signaux émis par une constellation de satellites prévue à cet effet, la position actuelle sur la face terrestre d'un terminal équipé d'une puce compatible. Cette position est alors traduite en termes de latitude, longitude et peut alors être représentée physiquement sur une carte. Le réseau satellite de positionnement le plus connu est le **GPS** (Global Positionning 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 géolocaliser 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. [2]



Figure I.1 :Géolocalisation par satellite.

II.2.2. Géolocalisation par GSM : [2]

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

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

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

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.

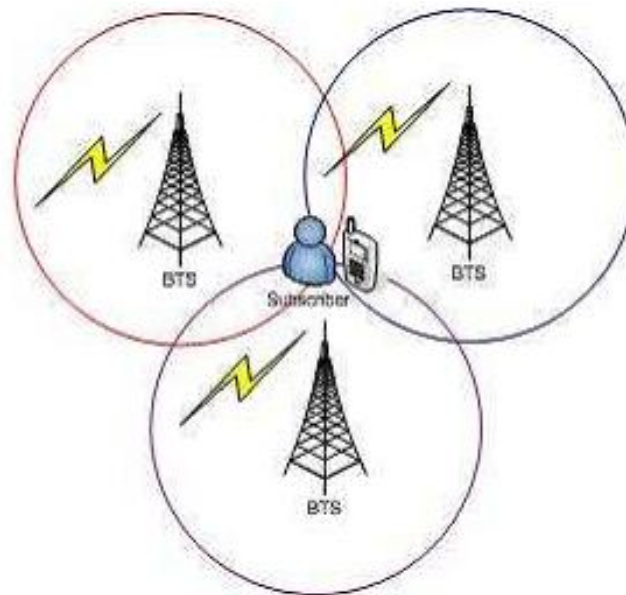


Figure I.2 : Positionnement par TAO

- **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 donnée. 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.

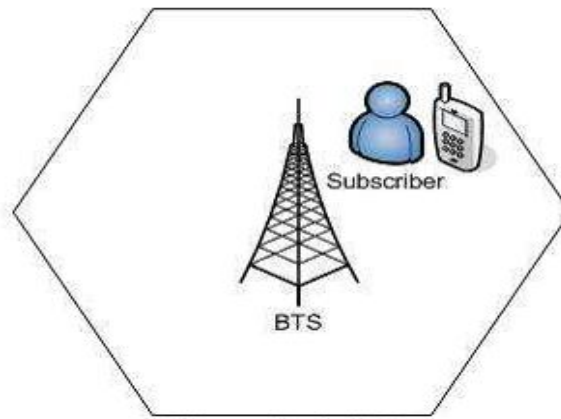


Figure I.3 : Positionnement par Cell ID

II.2.3. La géolocalisation par WiFi :

De la même façon qu'un terminal GSM peut se localiser par la méthode du Cell ID sur un réseau GSM, un terminal WiFi peut utiliser la même méthode en se basant sur les identifiants des bornes WiFi (adresses MAC) qu'il détecte. Il existe des bases de données recensant une multitude de bornes d'accès WiFi 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 WiFi et relié à un récepteur GPS, afin de recenser un maximum de points d'accès WiFi.

II.2.4. Combinaison de différentes techniques :

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 WiFi :** en zone rurale par exemple.

Les dispositifs qui combinent ces trois techniques sont capables à géolocaliser 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 géolocaliser 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 WiFi pour plus de précision.

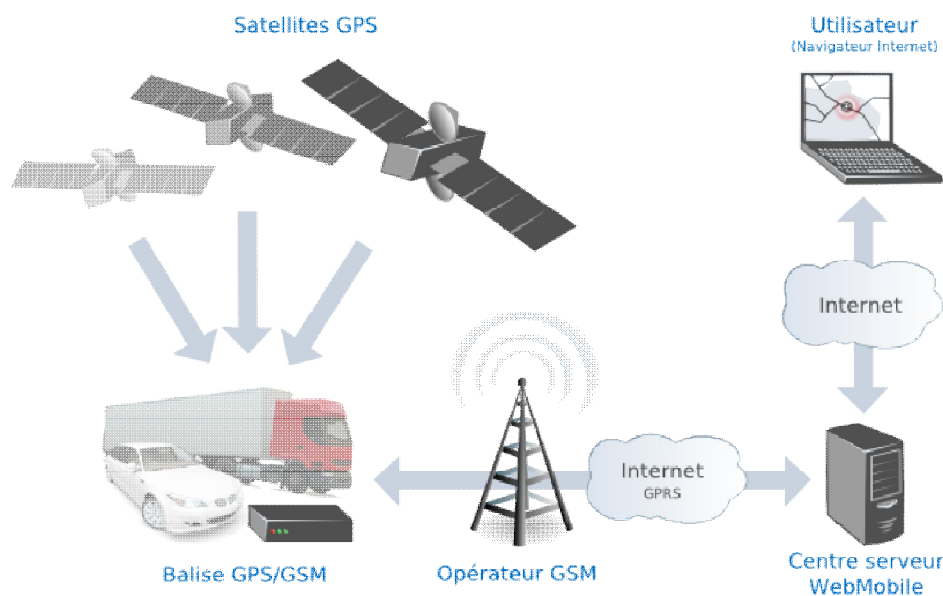


Figure I.4 : Géolocalisation GPS/GSM

II.2.5. Géolocalisation indoor : [3]

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



Figure I.5 : Géolocalisation indoor.

III. Plateformes logicielles de géolocalisation :

III.1. Les composants :

Les composants essentiels d'une plateforme de géolocalisation sont les suivants :

- **Terminal communicant** : C'est le terminal qui reçoit ses coordonnées géographiques d'un satellite et qui les envoie via un réseau de télécommunications à la plateforme.
- **Système informatique capable de recevoir, stocker et traiter les informations** : il s'agit des 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).
- **Module cartographique** : c'est le module intégré au système informatique qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté (Ordinateur, Smartphone...). 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 terrain (vitesses autorisées, la direction...).

III.2. Architecture temps réel :

La position géographique d'un terminal géolocalisé 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. Types de terminaux existants :

IV.1. Modes de fonctionnement : [4]

Les terminaux de géolocalisation existants peuvent être en 2 catégories :

- **Data pullers** : Ces terminaux envoient leur position à la demande;
- **Data pushers** : Ces terminaux effectuent des envois fréquents de leur position.

a. 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...

b. Data pushers

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

- Les terminaux mobiles de type PDA ou smartphones 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).
- Les dispositifs de poche destinés aux personnes et embarquant un récepteur GPS et un modem GSM/GPRS ou satellite.

Ces dispositifs peuvent nécessiter le branchement à une source électrique ou être autonomes grâce à une batterie interne. En fonction de l'utilisation du terminal, celui-ci pourra être connecté à la batterie d'un véhicule ou disposer d'une grande autonomie.

IV.2. Modes de communication :

Voici les différents types de terminaux disponibles selon leur mode de transmission des données.

a. Terminaux GSM/GPRS

Ce moyen de transmission nécessite un terminal disposant d'un modem GSM/GPRS ou 3G/UMTS, ainsi que d'une carte SIM de n'importe quel opérateur. Le terminal nécessite d'être sous couverture GSM/GPRS pour pouvoir envoyer les données vers la plateforme de traitement. Ce type de terminal est utilisé lorsque l'objet ou la personne à géolocaliser reste dans une zone bien couverte par les réseaux GSM/GPRS.

b. Terminaux satellite :

Ce type de terminaux envoie les données à travers un réseau de satellites de télécommunications, même si ces types de canaux sont plus restrictifs au niveau de la quantité de données envoyées, ils peuvent offrir par ailleurs une couverture mondiale en fonction des satellites et protocoles utilisés. Cela en fait des terminaux idéaux pour la géolocalisation de conteneurs, navires ou véhicules circulant en plein désert.

Pour des parcours longs distance ou des remontées d'information peu fréquentes, les solutions satellites sont plus économiques que les solutions GSM/GPRS.

V. Exemples de services géolocalisés :

- Le tracking de marchandises, de flottes de camions, de taxis, . . . est généralement considéré comme une des applications majeures des services géolocalisés car ils ont une utilité industrielle.

En effet, ils peuvent améliorer les rendements en fournissant des informations comme l'avertissement de retards de livraison. Ils permettent aussi d'optimiser l'usage d'une flotte de taxis. Ce genre de services est déjà mis en œuvre dans de nombreuses sociétés comme les Taxis Verts à Bruxelles.

- On peut également faire du tracking de personnes. Certains parents pourraient vouloir connaître à tout moment où se trouve leur enfant ou être avertis quand il quitte l'enceinte de l'école.
- Les informations générales par rapport au lieu où l'on se trouve sont des services ayant une forte valeur ajoutée. Dans cette catégorie on retrouve par exemple un service permettant de rechercher l'ensemble des restaurants selon des critères (prix, style, . . .) dans un rayon donné autour de l'utilisateur.
- Les services liés à la sécurité sont également une application possible pour les services géolocalisés. Certaines voitures pourraient être équipées d'un système de positionnement capable d'appeler une ambulance et de donner la position précise du véhicule en cas de problème. Mais cette catégorie regroupe également des informations d'intérêt général, comme l'avertissement de personnes présentes dans une certaine zone qu'ils doivent évacuer après un feu ou une explosion.
- Les services de communication sont une autre application des services géolocalisés. Dans la lignée des services de messageries instantanés, on localise et communique avec des connaissances situées dans une même zone géographique.
- Les divertissements sont une des possibilités pour l'adoption massive des services géolocalisés par le grand public. Des jeux grandeur nature comme des chasses au trésor peuvent être mis en œuvre à l'aide de services géolocalisés.
- Finalement, la publicité est vue comme une des industries majeures pouvant tirer parti des services géolocalisés. On pourrait demander la liste de magasins environnants qui font des promotions ou être avertis des produits en promotion lorsque l'on rentre dans un magasin.
- Une utilité des services géolocalisés est la capacité d'offrir des services différents en fonction de la position de l'utilisateur ou de lui fournir des données différentes en fonction de

cette position. Dans une usine d'assemblage de voitures, la personne contrôlant la qualité peut, grâce à de tels services géolocalisés, obtenir automatiquement des statistiques sur les différentes étapes de la chaîne de production. Il ne doit pas préciser quelle étape il est en train d'inspecter puisque sa position dans l'usine permet de conclure ce qu'il inspecte. Dans un musée, un visiteur pourrait automatiquement recevoir des informations sur l'œuvre d'art qu'il a devant soi.

VI. Principes du GPS :

VI.1. Définition du GPS :

Global Positioning System est un système de géolocalisation par satellite. C'est un réseau de 24 satellites actuellement en fonctionnement, développé par l'armée américaine, est mis à disposition des civils. Il permet de déterminer les coordonnées géographiques de n'importe quel point situé à la surface du globe. Sa précision peut atteindre 1 mètre. Le GPS s'utilise en association avec une carte pour se repérer et se positionner.

VI.2. Composition du GPS :

Dans le langage courant, on utilise le mot « GPS » pour désigner le récepteur ou « GPS » désigne le système de positionnement en entier, et ce système se compose de trois éléments : **la partie spatial, la partie contrôle et la partie utilisateur.** [5]

a. La partie spatiale :

Les satellites GPS décrivent des orbites circulaires d'une durée de 12 heures, à 17 440 km d'altitude. Les orbites sont inclinées de 55° par rapport à l'équateur pour assurer une couverture des régions polaires. Les satellites s'orientent continuellement pour pointer les panneaux solaires qui les alimentent vers le Soleil et les antennes vers la Terre.



Figure I.6 : La partie spatiale.

b. La partie de contrôle :

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

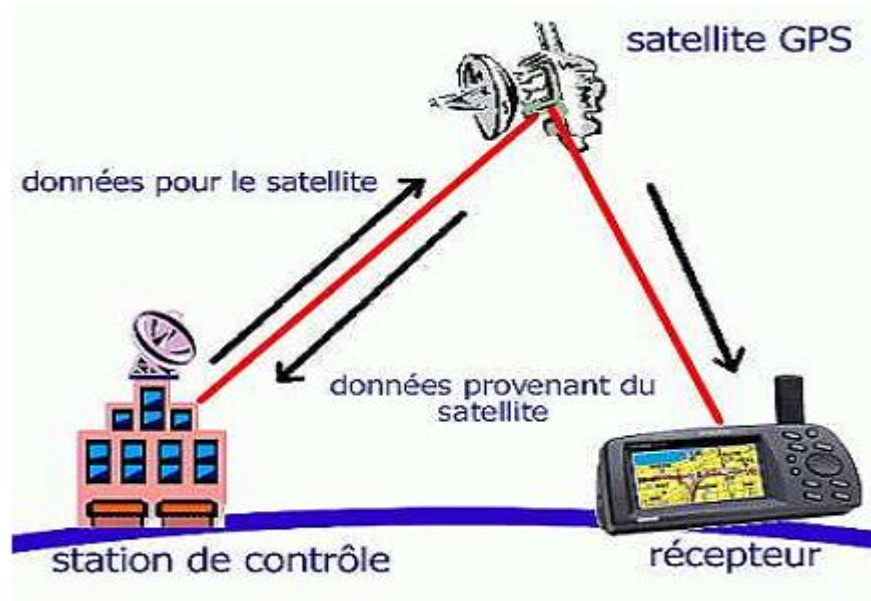


Figure I.7 : La partie de contrôle

c. La partie utilisateur :

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



Figure I.8 : La partie utilisateur

VI.3. Principes de fonctionnement d'un GPS : [6]

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

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

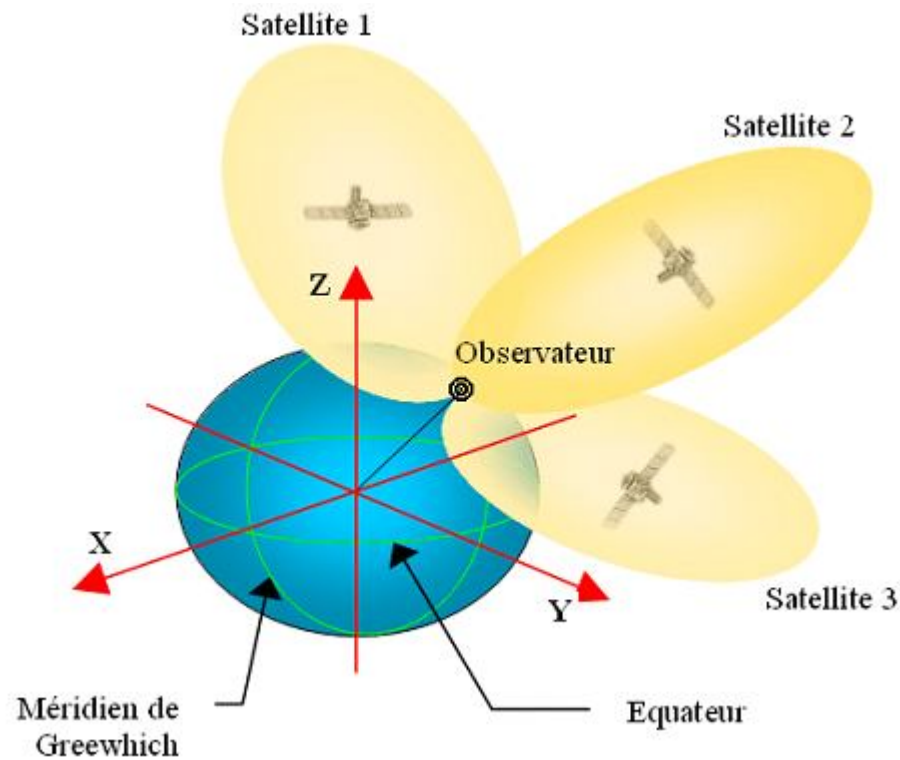


Figure I.9 : Fonctionnement du GPS

Les informations envoyées par chaque satellite sont entre autre :

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

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

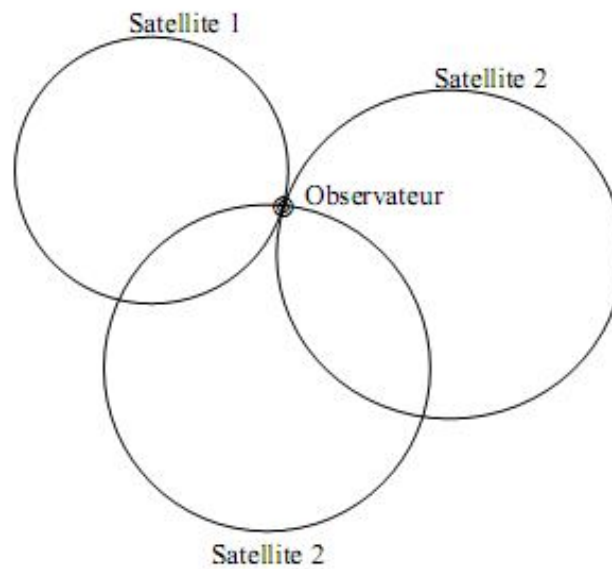


Figure I.10 : Positionnement de l'observateur à l'intersection des 3 satellites.

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

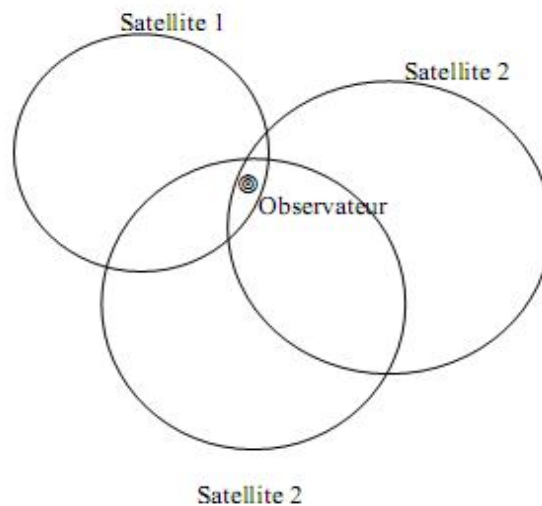


Figure I.11 : Positionnement de l'observateur dans la zone commune des 3 satellites

Quelques chiffres clef :

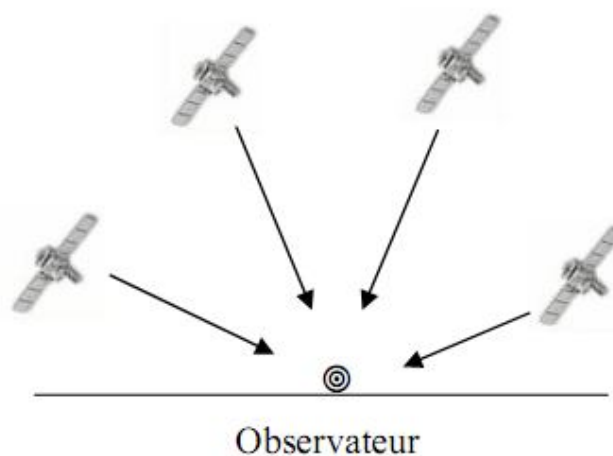
- La constellation GPS est constituée de 24 satellites

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

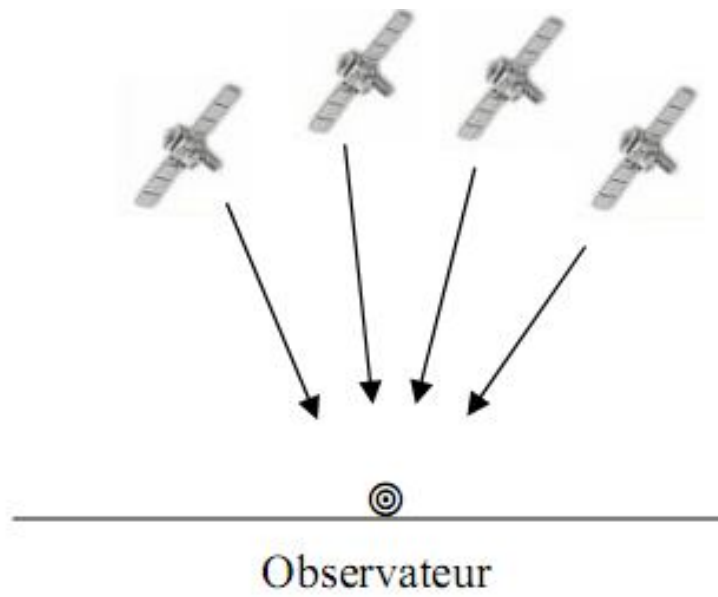
VI.4. Principes de positionnement d'un GPS : [6]

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

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

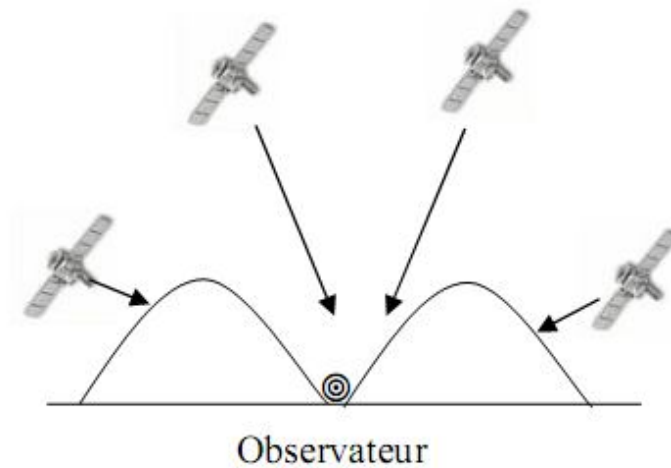


Bonne répartition.

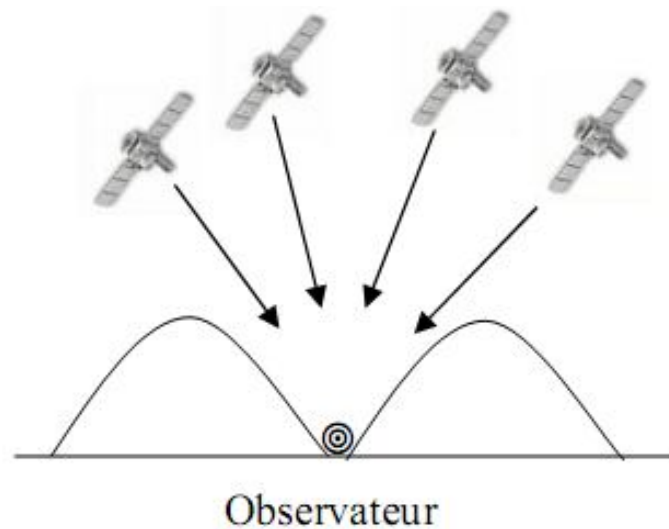


Mauvaise répartition

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



Dans une vallée, mauvaise réception

**Dans une vallée, bonne réception**

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

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

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

VII.Conclusion :

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

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

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

Chapitre II

Généralité sur ANDROID



I. Introduction :

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

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

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

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

II. Le système d'exploitation Android :

II.1. Définition :

Android est une couche logicielle qui inclut un système d'exploitation, un middleware et des applications clés. Google est le principal acteur, qui vient à l'esprit pour cette plateforme, mais d'autres acteurs comme les membres de l'**Open Handset Alliance** collaborent au développement. Le système d'exploitation est basé sur le noyau Linux.

II.2. Historiques:

En juillet 2005, Google a acquis Android, Inc., une petite startup qui développait des applications pour téléphones mobiles. C'est à ce moment là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'avait des données sûres à propos des marchés dans lesquels ils allaient se positionner.

Après ce rachat fait par Google, une équipe dirigée par Andy Rubin, un ancien d'Android Inc, a commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA soit crée officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que Google développait des applications mobiles de son moteur de recherche, qu'elle développait un nouveau téléphone mobile, etc.

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

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

En 2009, La version 1.5 **Cupcake** corrigea le manque d'API et rendit le système plus utilisable.

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

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

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

On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont surement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.

Android 4.0 Ice Cream Sandwich (Octobre 2011) il a rajouté encore plus de fonctionnalités aux terminaux. Pour le développement, ces nouvelles versions d'Android propose de nouveaux composants permettant de réaliser des applications avec une ergonomie plus adaptée aux tablettes tactiles.

Android 4.1 et 4.2 Jelly Bean (novembre 2012) il a apporté une nouvelle interface de l'appareil à photo, d'un système multi-compte uniquement sur tablette et d'amélioration de Google Now ajoutant la possibilité de suivre un colis ou d'embarquer à bord d'un avion.

II.3. Caractéristique [7] :

Le système d'exploitation Android est doté d'un ensemble de caractéristique faisant de lui un terrain d'application lucratif. Les principales caractéristiques sont résumées dans le tableau :

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/video/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.

Tableau II.1 : Les Caractéristiques du système d'exploitation Android.

II.4. Architecture Android [8] :

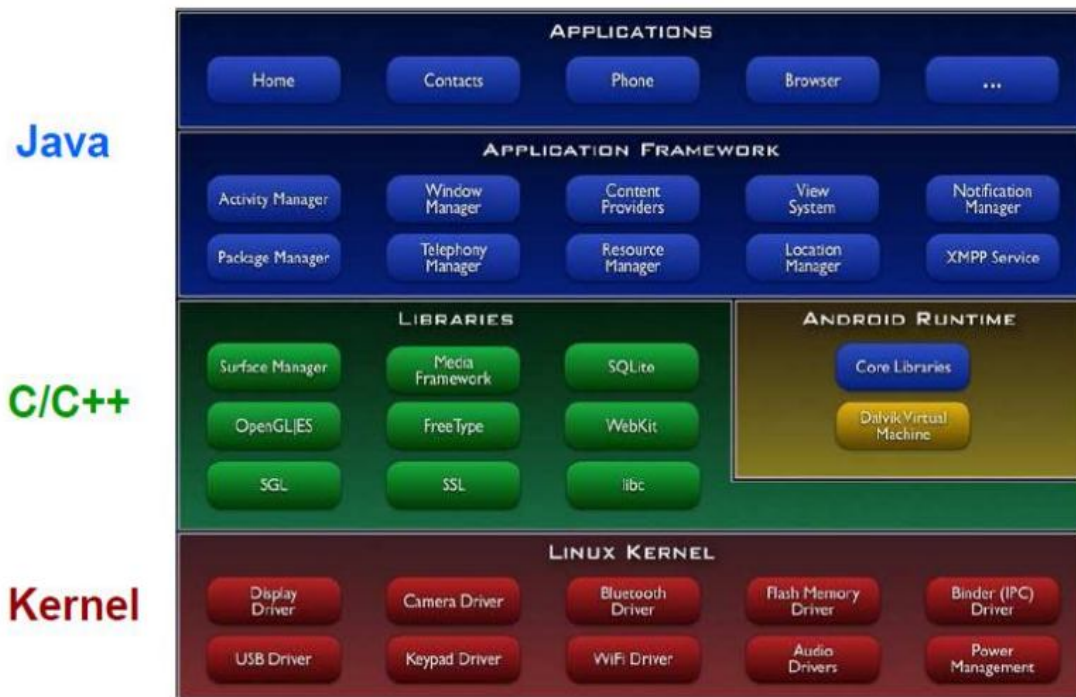


Figure II.1 : Architecture d'Android

a. Le noyau linux :

Le système d'exploitation d'Android se basait sur Linux, c'est le noyau («kernel » en anglais) de Linux qui est utilisé. Le noyau est l'élément du système d'exploitation qui permet de faire le pont entre le matériel et le logiciel. Par exemple les pilotes Wifi permettent de contrôler la connexion Wifi.

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. C'est cette couche qui fait en sorte qu'Android soit compatible avec tant de supports différents.

b. La bibliothèque pour android :

Ces bibliothèques proviennent de beaucoup de projets open-sources, écrits en C/C++ pour la plupart, comme SQLite pour les bases de données, WebKit pour la navigation web ou encore OpenGL afin de produire des graphismes en 2D ou en 3D.

c. Le frameworks pour les applications :

Un framework peut se traduire littéralement par un cadre de travail. Il s'agit d'un ensemble de composants qui définissent les fondations ainsi que les grandes lignes directrices de l'organisation d'un code, en d'autres termes on peut parler de son architecture ou de son squelette. Un framework prodigue aussi quelques fonctionnalités de base (accès à la base de données par exemple). Cet outil fournit ainsi une démarcation radicale entre plusieurs aspects d'un programme et permet de mieux diviser les tâches. En fait ce sont ces frameworks là qui sont disponibles quand on programme en Java, et qui s'occupent d'interagir avec la bibliothèque Android.

d. Les applications :

Il s'agit tout simplement d'un ensemble d'applications que l'on peut trouver sur Android, par exemple les fonctionnalités de base inclues un client pour recevoir/envoyer des emails, un programme pour envoyer/recevoir des SMS, un calendrier, un répertoire, etc...

II.5. Les outils :

Afin de programmer pour Android, plusieurs outils sont nécessaires, certains sont même fournis par Google.

Comme nous l'avons dit, les applications pour Android sont développées en Java, il s'agit d'un langage orienté objet avec une syntaxe relativement proche du C++. La particularité de ce langage est qu'il est indépendant de la plateforme. Le même code pourra être exécuté sur un Windows ou un Linux. Il s'agit du principe « Compile Once, Run Everywhere ».

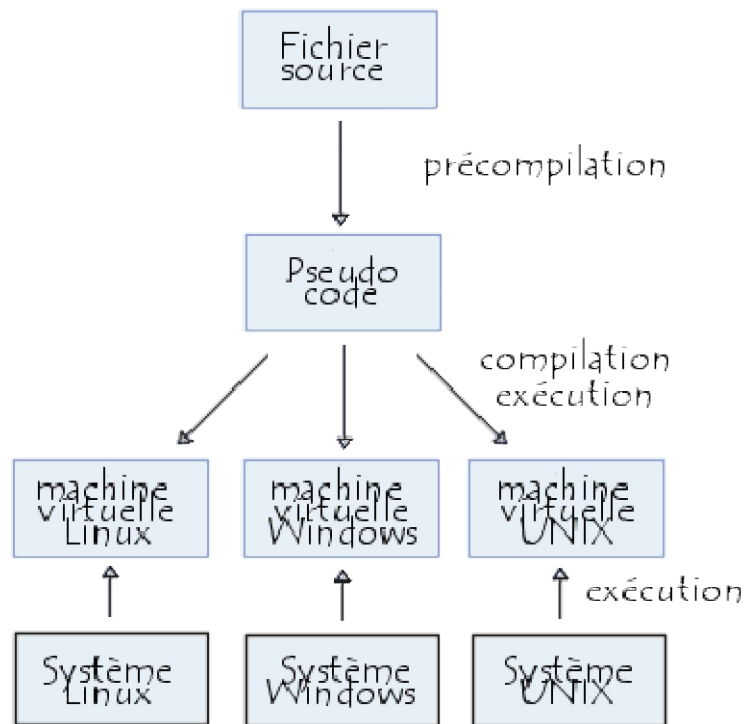


Figure II.2: Le concept de machine virtuelle java [9]

Google n'a pas utilisé la JVM pour Android, mais a plutôt réécrit sa propre machine virtuelle : « Dalvik ». Il s'agit donc d'une machine virtuelle pour des programmes Java sur appareils mobiles équipés d'Android. Dalvik exécute des applications converties au format Dalvik Exécutable (.dex).

Puisque l'environnement n'est pas exactement une JVM classique, nous ne pouvons pas utiliser directement les outils Java habituels afin de développer. Cependant puisque une des volontés de Google est que les développeurs créent un maximum d'applications pour Android, il est possible d'obtenir un grand nombre d'outils.

Le plus important est la distribution d'un kit de développement (**Standard Development Kit – SDK**). Il s'agit de la couche « Application Framework » présentée à la figure.1 On y retrouve l'ensemble des interfaces de développement (Application Programming Interface – API) que Google met à disposition. Nous travaillons alors avec les mêmes outils que les développeurs Android. Ce SDK contient une partie du noyau Java et il est étoffé avec les concepts Android.

Puisque Android est un système pour téléphones, afin de tester nos applications il faudrait en posséder un. Google a donc décidé de développer un émulateur afin que tout le monde puisse tester son travail.



Figure II.3 : l'émulateur

II.6. Application proposées :

Etant conçu par Google, l'intégration des applications existantes de Google dans la plateforme Android était la première cible, comme Gmail, Google Maps, Google Calendar, Google Talk, etc. En addition, Android accueille les applications développées par d'autres développeurs que Google, ce qui offre à ce dernier un apport commercial important. Il s'agit des applications innovantes dans des divers domaines tel que : le multimédia, la navigation GPS, la météo, la finance, etc. Ces applications peuvent être gratuites comme payantes. Elles sont accessibles depuis les téléphones Android. Les propriétaires de ces téléphones profitent d'une boutique de téléchargement d'application en ligne appelée Android Market.

III. Composantes d'une application Android [10]:

Les applications Android sont composées de 4 types de composants :

a. Activités :

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

b. Services :

Un service à la différence d'une activité, ne possède pas d'interface graphique mais permet de dérouler un algorithme sur un temps indéfini et il s'arrêtera lorsque sa tâche est finie ou que son exécution est arrêtée. Il peut être lancé au démarrage du téléphone (arrivée d'un appel, d'un SMS, mail, etc) ou de votre application, soit au cours d'une action particulière dans votre application via broadcast receivers.

c. Broadcast Receiver :

Un Broadcast Receiver permet d'écouter ce qui se passe sur le système ou sur votre application et éventuellement de déclencher une action que vous aurez définie. C'est souvent par ce mécanisme que les services sont lancés.

d. Content providers :

Les Content providers servent à accéder à un ensemble de données depuis votre application. Vous pouvez par exemple accéder aux contacts, à l'agenda, aux photos et d'autres données depuis votre application grâce aux content providers. Vous pouvez également définir vos propres content providers pour accéder à vos objets ou pour que d'autres applications utilisent vos données.

IV. Cycle de vie d'une activité :

Une activité n'a pas de contrôle direct sur son propre état, il s'agit plutôt d'un cycle rythmé par les interactions avec le système et d'autres applications. Voici un schéma qui présente ce que

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

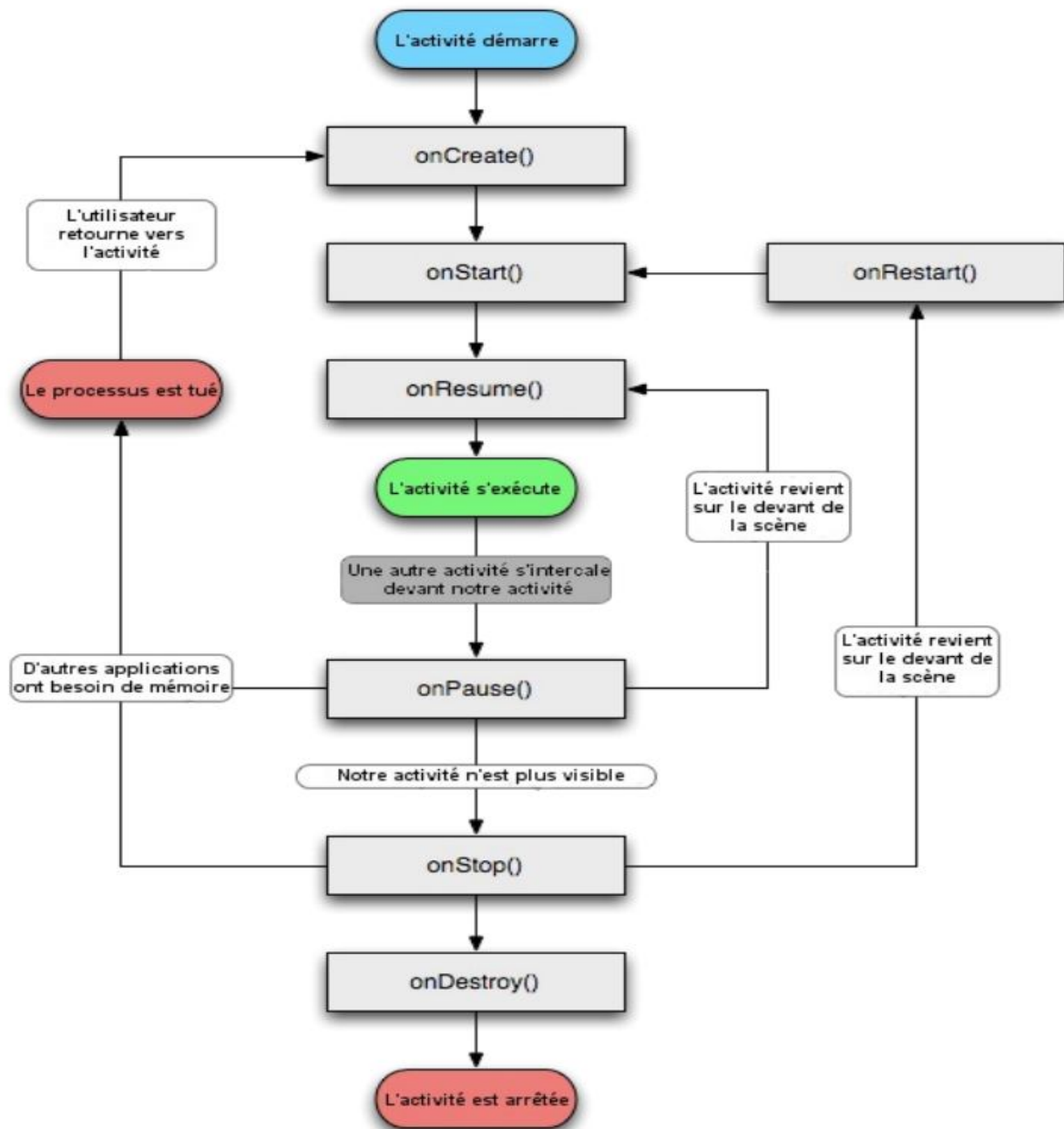


Figure II.4 : le cycle de vie d'une activité Android

- **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 votre 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.

V. Développement d'une application Android :

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

- Faire la conception de la base de données.
- Créer des classes pour représenter les données physiques (couche Mapping) et pour définir des actions comme : supprimer, ajouter, modifier des données.
- Dessiner des interfaces en les fichiers XML ou en codage :
 - ✓ Les vues (View) : Text, Edit, List, Image, Web, Map, etc.
 - ✓ Les arrangements (layout) : Frame, Linear, Relative, Table, Absolute.
- Choisir des arrangements (layout) : Les layouts sont les ressources qui indiquent les interfaces des activités. On utilise les fichiers XML pour exprimer les interfaces. Mais il existe d'autre technique pour dessiner l'interface. Dans cette technique, on programme directement les composants graphiques en utilisant le codage.

- Organiser des ressources: les constantes globales (string.xml), les icônes, les images, etc.
- Créer et mettre à jour le fichier de configuration : AndroidManifest.xml. AndroidManifest.xml (configuration de l'application) est utilisé pour stocker les dispositions (settings) globales comme les permissions de l'application, les activités, les filtres de l'intention.
- Créer des activités (Créer les classes pour exécuter les fonctions avec la base de données (ajouter, supprimer, modifier, mettre à jour, etc.) :
 - ✓ Chaque activité peut correspondre avec un écran ou une fonction de cette application.
 - ✓ Il existe quelques activités qui s'occupent des méthodes pour communiquer avec la base de données (Couche Mapping).
 - ✓ Les Intents sont utilisées pour orienter des activités (CALL, ACTION_MAIN, ACTION_VIEW, etc).

VI. Les outils et technique de géo-localisation et d'orientation sous Android :

a. 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 géoportail 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.

b. 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.

c. 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. [11]

d. 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. [11]

VII. Conclusion :

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

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

Chapitre III

Analyse et Conception



I. Introduction :

Notre travail s'est porté dans les chapitres précédents sur la présentation de l'état de l'art lié au domaine de la géolocalisation. Une attention particulière a été consacrée à la géolocalisation par GPS, une technique prédominante lorsqu'il s'agit de la géolocalisation à l'extérieur (Outdoor). L'application développée est une plateforme mobile sous un système Android qui sert à géolocaliser sur une carte (map) des véhicules en utilisant les données stoker dans une base de données à distance.

Le chapitre suivant présente les étapes d'analyse et de conception de l'application en faisant appel aux diagrammes UML.

II. Présentation de l'UML :

II.1. Idée de base : [12]

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

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

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

II.2. Modélisation avec L'UML : [13]

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, analyser par étapes, favoriser le prototypage, réduire et maîtriser l'inconnu.
- ✓ **Une démarche guidée par les besoins des utilisateurs** : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre

à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.

- ✓ **Une démarche centrée sur l'architecture logicielle** : c'est la clé de voute de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

II.3. La démarche de modélisation avec L'UML :

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

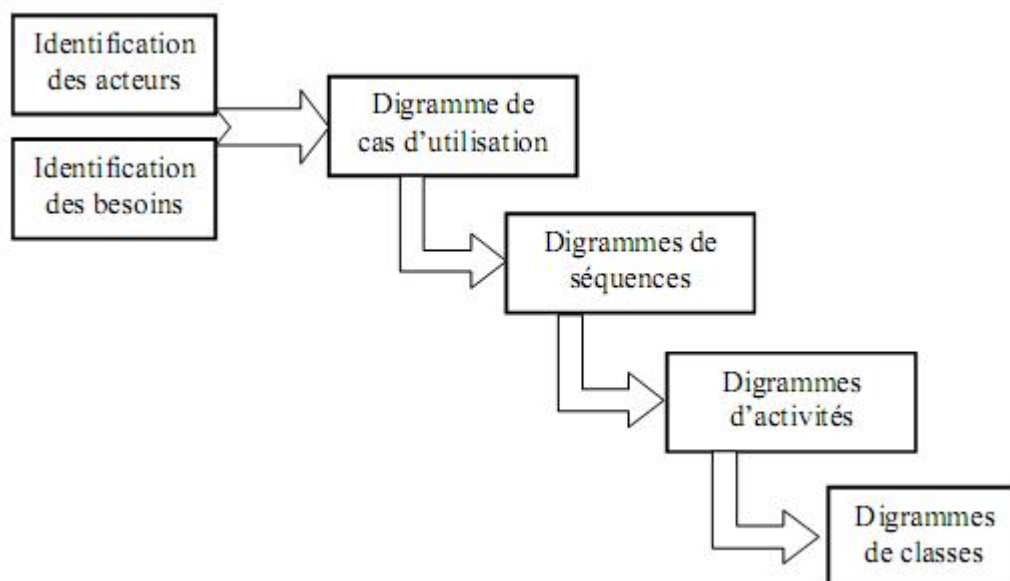


Figure III.1 : La démarche adoptée pour la modélisation.

Elle fait appel à certains diagrammes d'UML afin de spécifier et de représenter des aspects liés au processus de développement.

III. Analyse:

Cette partie comprend l'identification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système avant de conclure sur les cas d'utilisation.

III.1. Identification des besoins :

L'objectif principal de l'application développée est de permettre à l'utilisateur de surveiller sa flotte de transport en temps réel afin de mieux gérer ses livraisons et de localiser ses véhicules en cas de vole notamment.

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

- ✓ Affichez les véhicules correspondant au client.
- ✓ Modifier le mot de passe.
- ✓ Localiser en temps réel les véhicules du client.

III.2. Identification des acteurs : [14]

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

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

- ✓ **Utilisateur :** Cet acteur a pour responsabilité de lancer l'application, et s'authentifier puis par la suite il pourra accéder au menu des tâches et le cœur de notre application et delà, entre autre, localiser tous ses véhicules en temps réel.

III.3. Spécification des tâches :

✓ **Définition :** Une tâche est l'ensemble des différentes fonctions accessibles pas un acteur bien spécifié.

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

Acteurs	Tâches
Utilisateur (user)	T1 : Lancer l'application. T2 : S'authentifier. T3 : Accéder au menu des tâches. T4 : Modifier le mot de passe. T5 : Afficher la liste des véhicules. T6 : Afficher la position d'un véhicule sur la map et le suivre à la trace en temps réel. T7 : Déconnection.

Tableau III.1 : Spécification des tâches.

III.4. Spécification des scénarios :

✓ **Définition** : C'est une instance de cas d'utilisation, chaque fois qu'une instance d'un acteur déclenche un cas d'utilisation, un scénario est créé. Ce scénario suivra un chemin particulier dans le cas d'utilisation. [14]

Note :

Durant notre étude les scénarios seront symboliser par **si** (**i** : représente le numéro scénario).

Le tableau ci-dessous récapitule les différents scénarios créés par les cas d'utilisation cités.

Acteurs	Tâches	Scénarios
Utilisateur	T1 : Lancer l'application.	S1 : Cliquer sur l'icône de l'application.
	T2 : S'authentifier.	S2 : Saisie le Login et Password et Cliquer sur le bouton « Connexion ».
	T3 : Accéder à la liste des véhicules	S3 : Cliquer sur l'icône « Liste des véhicules » qui permet d'afficher la liste de tous les véhicules.
	T4 : Modifier le mot de passe.	S4 : Cliquer sur l'icône « modifier mot de passe » qui représente la modification de mot de passe. On saisit l'ancien et nouveau mot de passe puis valider.
	T5 : Localiser les véhicules en temps réels.	S5 : Cliquer sur l'icône « Localiser » qui va permettre de localiser le véhicule sur la map en temps réel.
	T6 : Déconnexion.	S6 : Cliquer sur l'icône « Déconnexion » pour se déconnecter

Tableau III.2 : Spécification des scénarios.

III.5. Les cas d'utilisation :

✓ **Définition** : Un cas d'utilisation 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. [14]

✓ **Description des cas d'utilisations :**

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

Cas d'utilisation : Affichage de la liste des véhicules.

Scénarios1 : Afficher la liste des véhicules

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliquer sur l'icône.
- 2- Le système affiche une page « Authentification ».
- 3- L'utilisateur saisie le login et Password et clique sur le bouton « connexion ».
- 4- Le système affiche l'interface « menu»
- 5- L'utilisateur clique sur le bouton « liste des véhicules ».
- 6- Le système construit et affiche les véhicules du client.
- 7- L'utilisateur clique sur le bouton « déconnexion ».

Figure III.3 : Cas d'utilisation « Afficher la liste des véhicules ».

Cas d'utilisation : Accéder à la partie localisation.

Scénarios1 : Localiser un véhicule

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliquer sur l'icône.
- 2- Le système affiche une page « Authentification ».
- 3- L'utilisateur saisie le login et Password et clique sur le bouton « connexion ».
- 4- Le système affiche l'interface « menu»
- 5- L'utilisateur clique sur le bouton « liste des véhicules ».
- 6- Le système construit et affiche les véhicules du client.
- 7- L'utilisateur choisi un véhicule parmi la liste proposées.
- 8- Le système construit et affiche l'interface « information ».
- 9- L'utilisateur clique sur le bouton « localiser ».
- 10- Le système affiche l'interface « map » et la position du véhicule en temps réel.
- 11- L'utilisateur clique sur le bouton « déconnexion ».

Figure III.4 : Cas d'utilisation « Accéder à la partie localisation ».

Cas d'utilisation : Modification de mot de passe

Scénarios1 : Modifier le mot de passe

Rôle : utilisateur

Description1 :

- 1- L'utilisateur lance l'application après avoir cliquer sur l'icône.
- 2- Le système affiche une page « Authentification ».
- 3- L'utilisateur saisie le login et Password et clique sur le bouton « connexion ».
- 4- Le système affiche l'interface « menu»
- 5- L'utilisateur clique sur le bouton « Modifier le mot de passe ».
- 6- Le système construit et affiche le formulaire de modification de mot de passe.
- 7- L'utilisateur saisie le nouveau mot de passe et valide.
- 8- L'utilisateur clique sur le bouton « déconnexion ».

Figure III.5 : Cas d'utilisation « Modifier le mot de passe».

IV. 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 digramme d'activité.
- ✓ Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

Le niveau 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.

IV.1. Le niveau applicatif :

IV.1.1. Le diagramme des cas d'utilisation:

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

✓ **Définition :** Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système. [15]

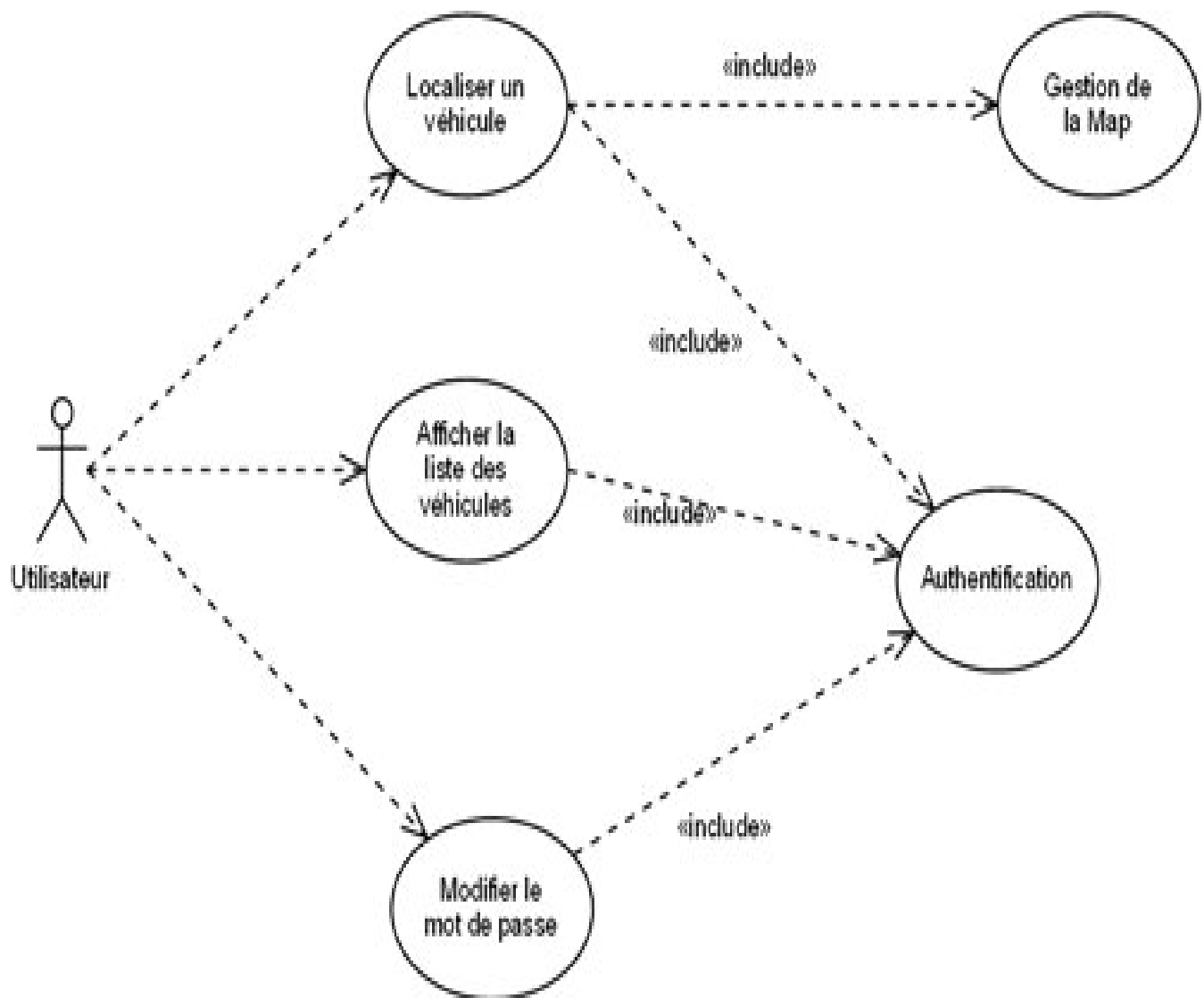


Figure III.6 : Diagramme de cas d'utilisation générale.

VI.1.2. Les diagrammes de séquences :

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

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

- ✓ **Les objets** : Ils apparaissent dans la partie supérieure, ce qui facilite l'indentification des classes qui participent à l'interaction.
- ✓ **Les messages** : Ils sont représentés par des flèches directionnelles annotées d'un texte informatif décrivant le message envoyé entre les objets.

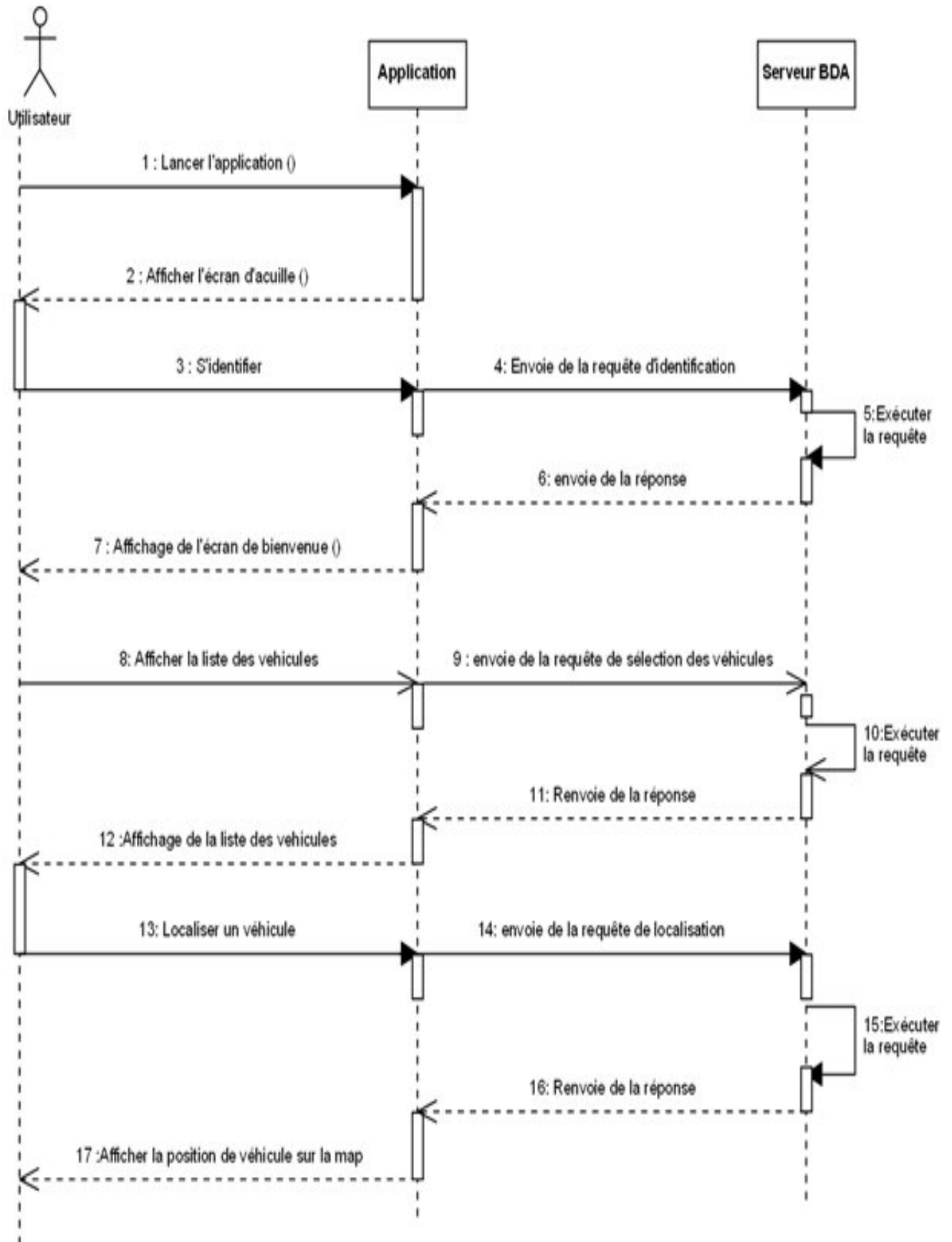


Figure III.7: Diagramme de séquence de cas d'utilisation « localiser un véhicule ».

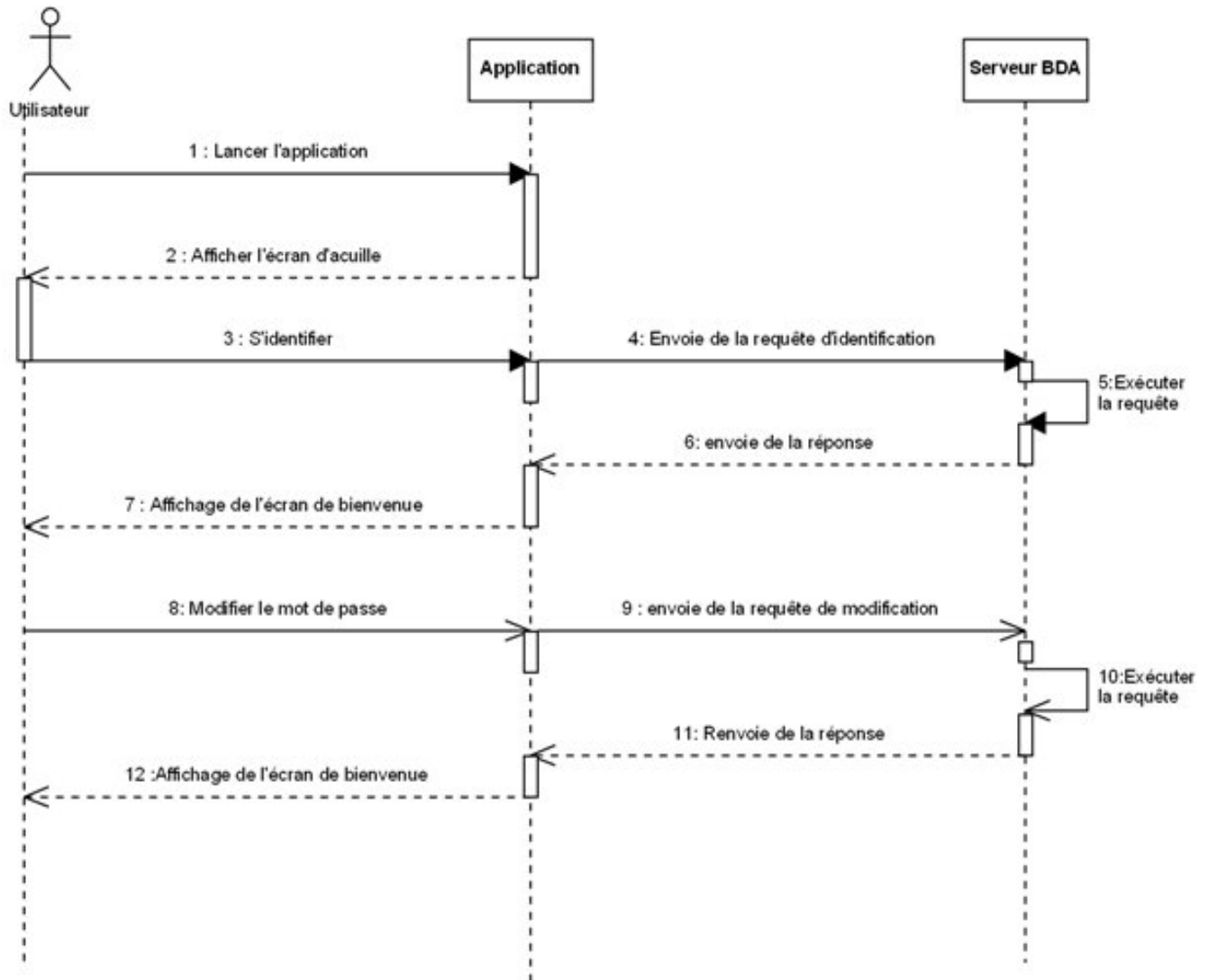


Figure III.8: Diagramme de séquence de cas d'utilisation « modifier le mot de passe ».

VI.1.3. Les diagrammes d'activités :

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

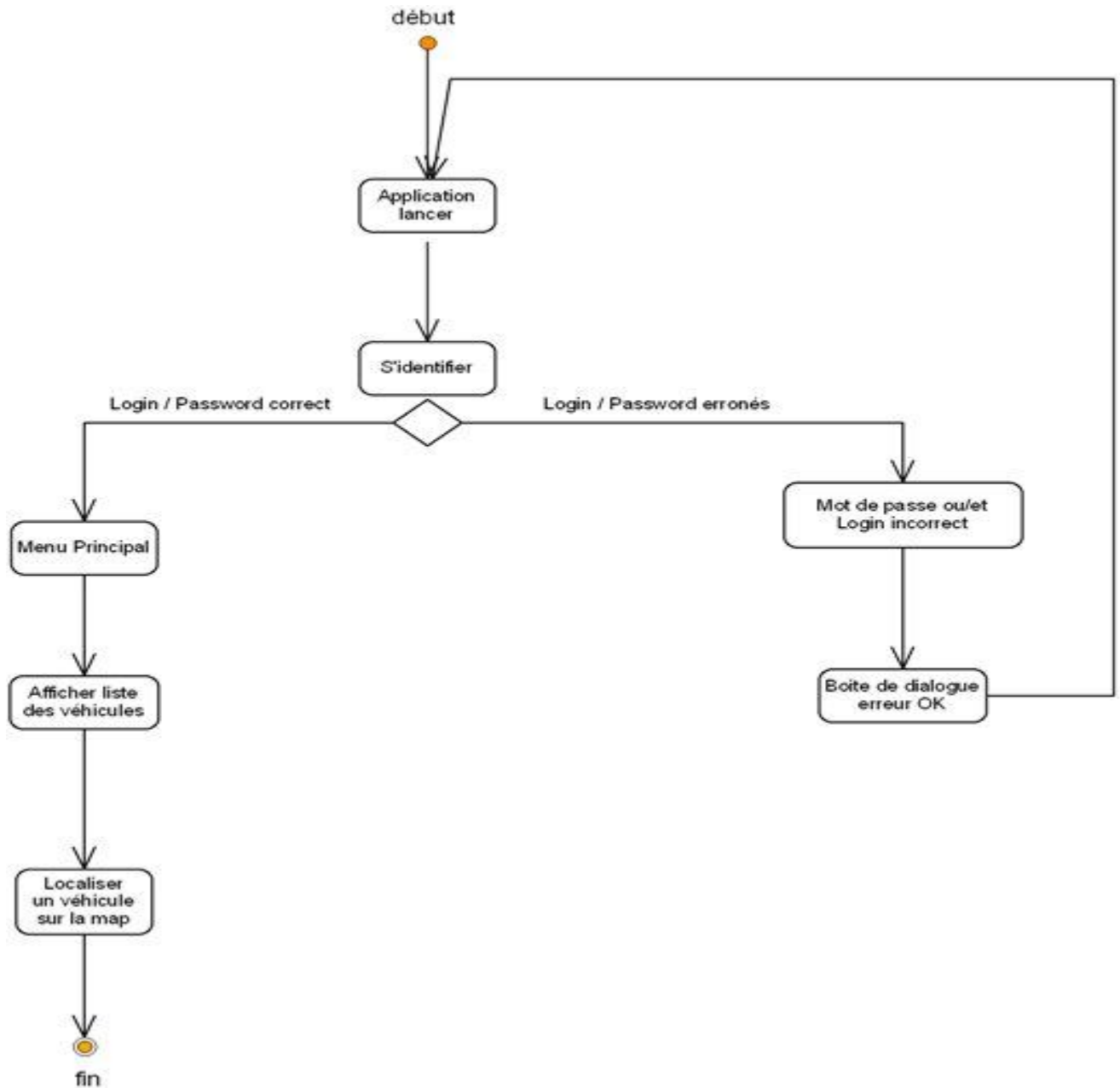


Figure III.9: Diagramme d'activité de cas d'utilisation « Localiser un véhicule ».

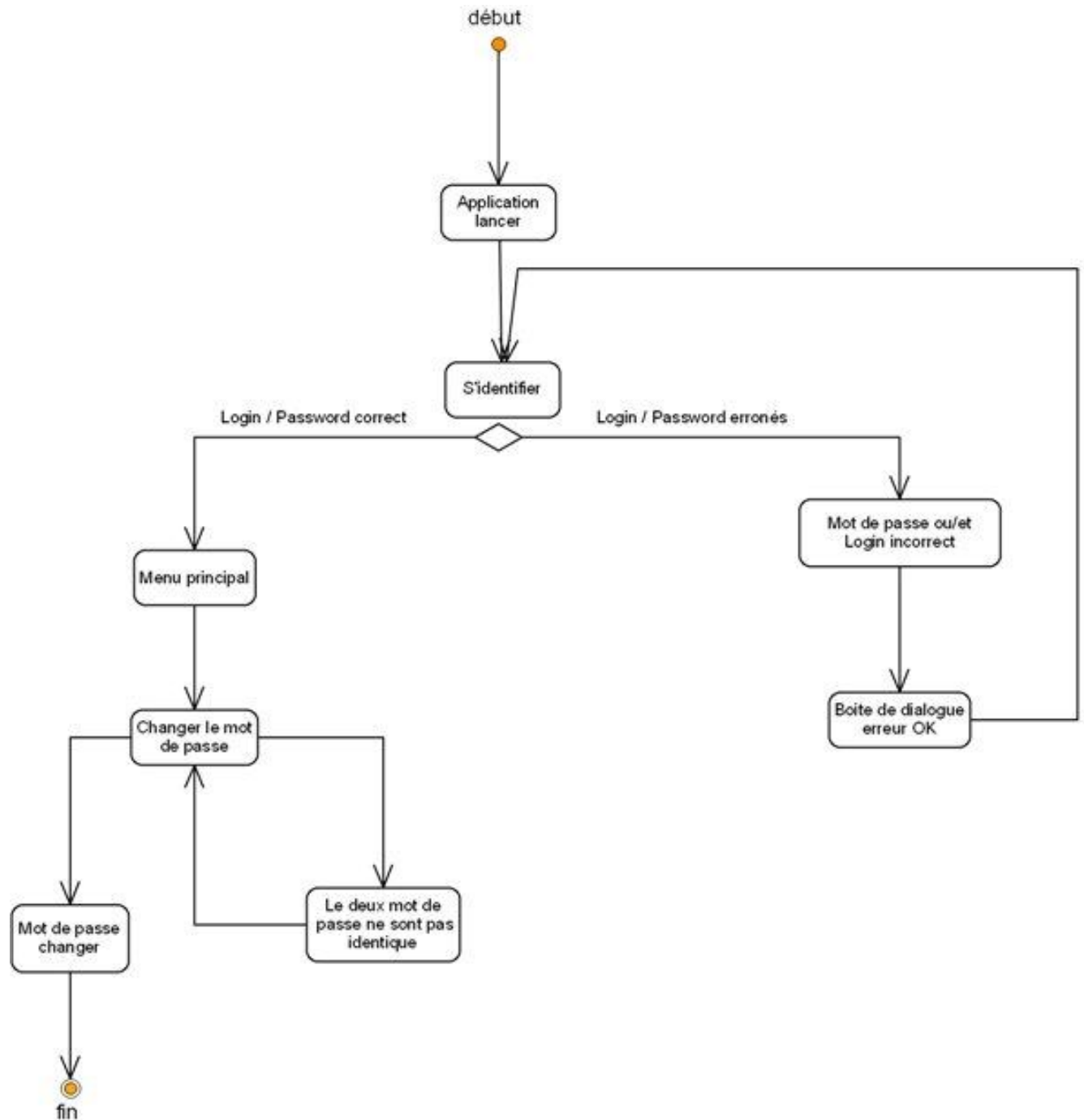


Figure III.10 : Diagramme d'activité de cas d'utilisation «Changer le mot de passe ».

VI.1.4. 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 diagramme 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 diagramme de classe en montre la structure interne. Il contient principalement des classes reliées par des associations et chaque classe contient des attributs et des opérations. [16]

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

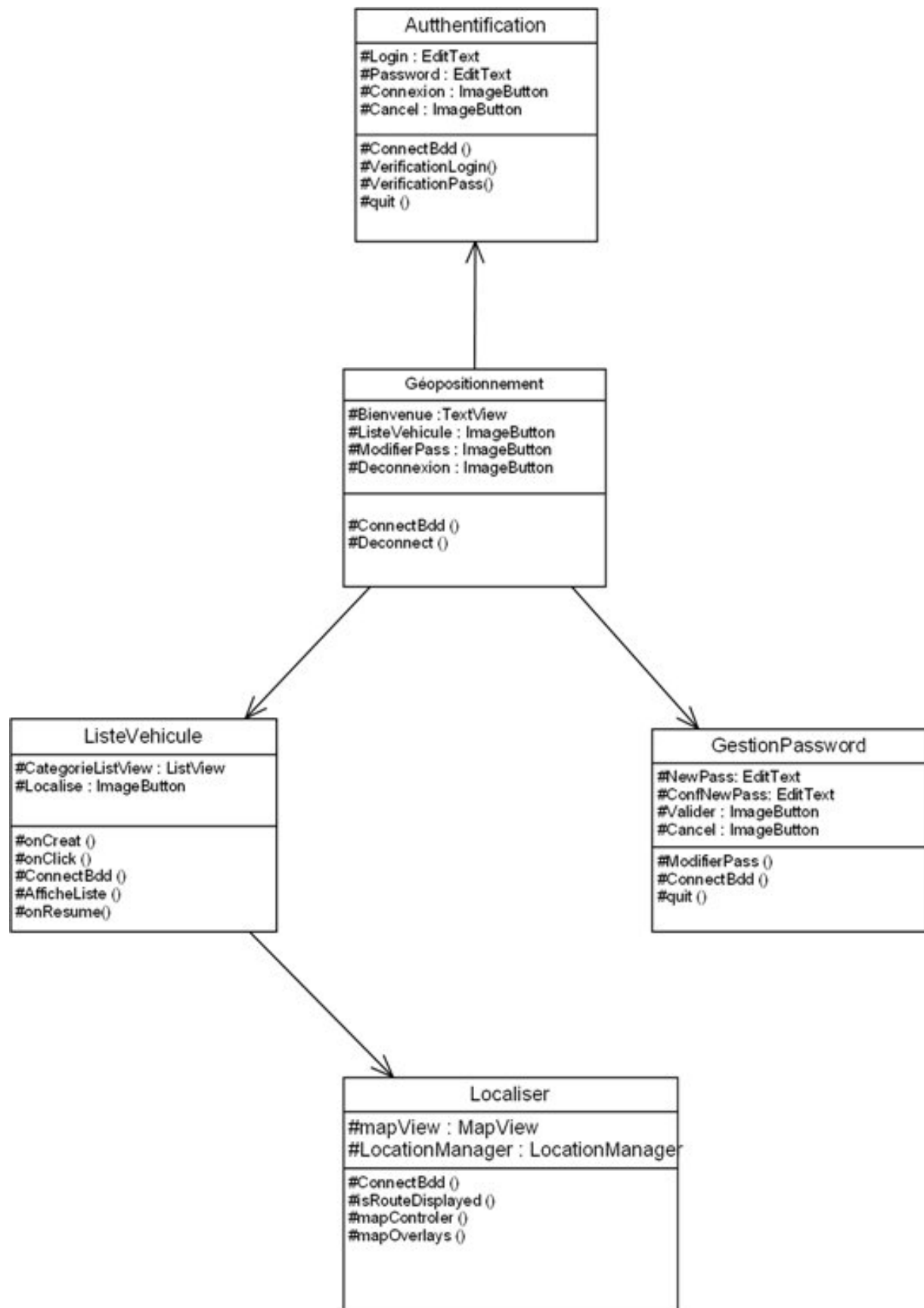


Figure III.9 : Diagramme de classe.

IV.2. Le niveau donnée:

IV.2.1. Schéma de la base de données :

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

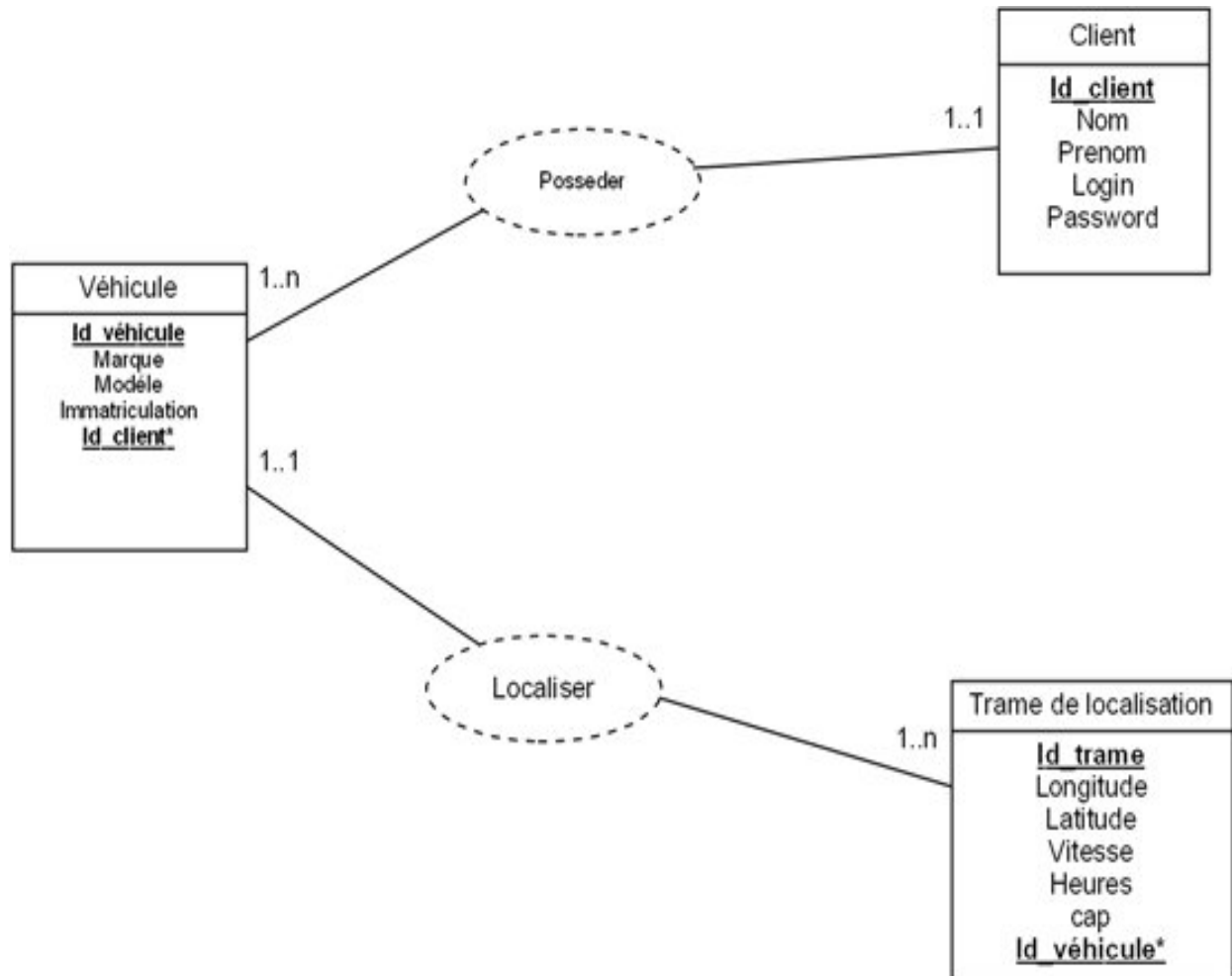


Figure III.10 : Le schéma de la base de données.

Notre base de donnée est constitué de 3 tables que nous allons détailler par la suite. Ces trois tables son liée entre elle avec les relations suivant :

Un seul et un seul client peut posséder un à plusieurs véhicules.

Un seul et un seul véhicule peut être localisé par une seul trame à la fois.

IV.2.2. Les tables :

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

La Table Client :

Champ	Signification	Type	Clé
ID_Client	l'identifiant unique pour chaque client auto-incrémenté.	INT	Primaire (PK)
Nom_Client	le nom du client	VARCHAR	
Prénom_Client	le prénom du client	VARCHAR	
Date_inscription	Date inscription de client dans la société de surveillance car il s'agit d'un abonnement qui doit payer.	VARCHAR	
Date_fin_abonnement	Date fin abonnement il s'agit de la date d'expiration de contrat d'abonnement.	VARCHAR	
Login_client	Login client il s'agit de nom d'utilisation pour la connexion à l'espace personnel du client	VARCHAR	
Password_Client	Password il s'agit du mot de passe associé au login avec la combinaison des deux en permet au client d'accéder à son propre espace d'où il peut surveiller ses véhicules.	VARCHAR	

Tableau III.11 : Table Client.

La table Véhicule :

Champ	Signification	Type	Clé
ID_véhicule	l'identifiant unique pour chaque véhicule auto-incrémentée	INT	Primaire (PK)
Nom_véhicule	le nom de véhicule.	VARCHAR	
Marque_véhicule	La marque a qui appartient le véhicule.	VARCHAR	
Matriculation_véhicule	Le numéro de circulation de véhicule.	VARCHAR	
ID_Client	l'identifiant de client à qui appartient le véhicule.	INT	Etrangère (FK)

Tableau III.12 : Table véhicule

La table Trame :

Champ	Signification	Type	Clé
ID_trame	l'identifiant unique pour chaque trame auto-incrémentée (une trame regroupe les données de localisation d'un véhicule particulier.)	INT	Primaire (PK)
Longitude	Indique la longitude d'un véhicule particulier.	VARCHAR	
Latitude	Indique la latitude d'un véhicule particulier.	VARCHAR	
Heure_date	Indique l'heure et la date d'acquisition des données.	VARCHAR	
ID_véhicule	l'identifiant de véhicule à qui appartient la trame.	INT	Etrangère (FK)

Tableau III.13 : Table Trame.

IV.3. Conception des maquettes préliminaires :

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

IV.3.1. L'écran d'accueil :

C'est une page d'accueil constitué de deux champs de saisie qui sont Login et Password, qui permet d'identifier chaque utilisateur et de deux boutons.

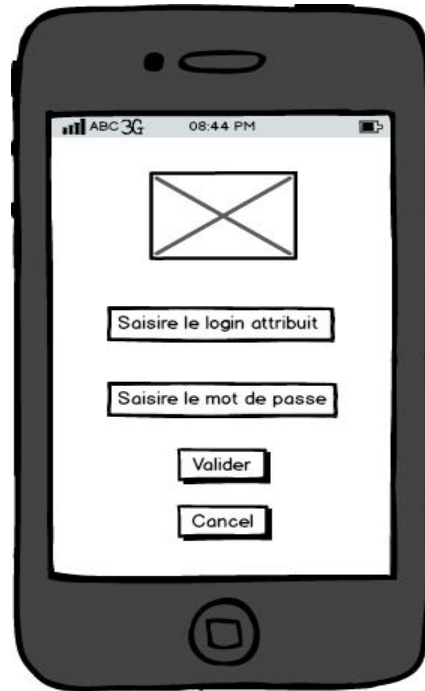


Figure III.15 : Page d'accueil.

IV.3.2. L'écran de l'espace personnel :

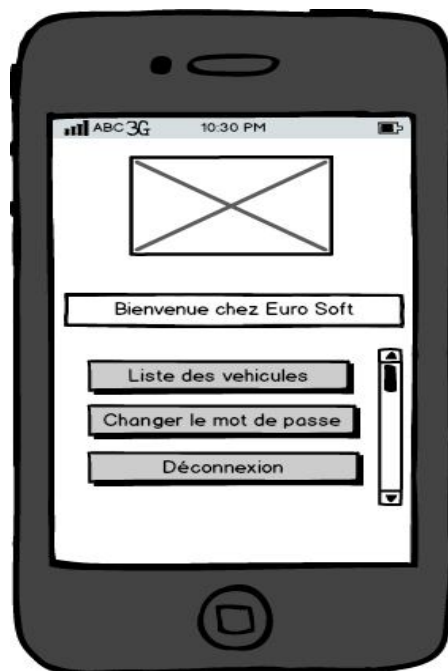


Figure III.16 : Page espace personnel.

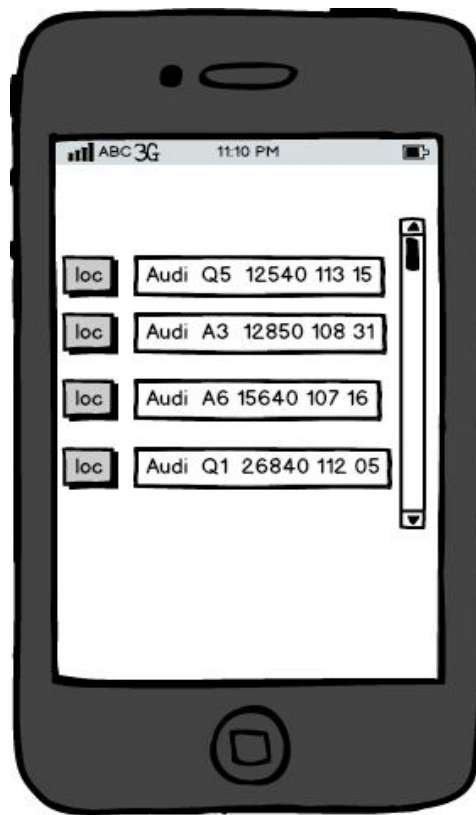
IV.3.3. L'écran de la liste des véhicules :

Figure III.17 : Interface de la liste des véhicules.

V. Conclusion :

A l'issue de ce chapitre, nous avons introduit les objectifs de l'application développée, tout en donnant une description complète de ces diagrammes, ce qui permet une meilleure compréhension du système.

Le chapitre suivant sera consacré à la réalisation et l'implémentation de notre application qui devra satisfaire les objectifs exprimés dans ce chapitre.

Chapitre IV

Réalisation et mise en œuvre



I. 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.

II. Environnement de travail :

II.1. Environnement matériel :

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

- PC Portable : Intel Core i3, 2.13 GHz, 4 GO de mémoire vive, Windows 7.
- PC Portable : Intel Core i5, 2.53 GHz, 4 GO de mémoire vive, Windows XP.
- Samsung galaxy SII : Modèle GT-I9100P, Version Android 4.0.3.

II.2. Environnement cible :

Notre application est développée sous un système **Android Ice Cream Sandwich 4.0**, elle est destinée pour tous les Smartphones équipé d'une version Android 4.0 et pour les versions ultérieures, toutefois l'application développée est testé sur la version 4.0.1.

II.3. Environnement logiciel :

Dans cette section nous allons décrire l'environnement utilisé pour le développement de notre application mobile sous un système Android connectée à une base de données MySQL, ainsi que le serveur Web utilisé.

a. La plate forme Eclipse :

Dans notre cas, on a choisi de faire notre développement mobile sur la plateforme Eclipse. Pour la préparer, On a effectué différentes tâches décrites dans ce qui suit :

- ✓ Installation Eclipse

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

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

b. La plate forme WampServer :

WampServer est une plateforme de développement Web de type WAMP (Windows-Apache-MySQL-PHP). WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

II.4. Eclipse INDIGO :

Eclipse INDIGO est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en java.

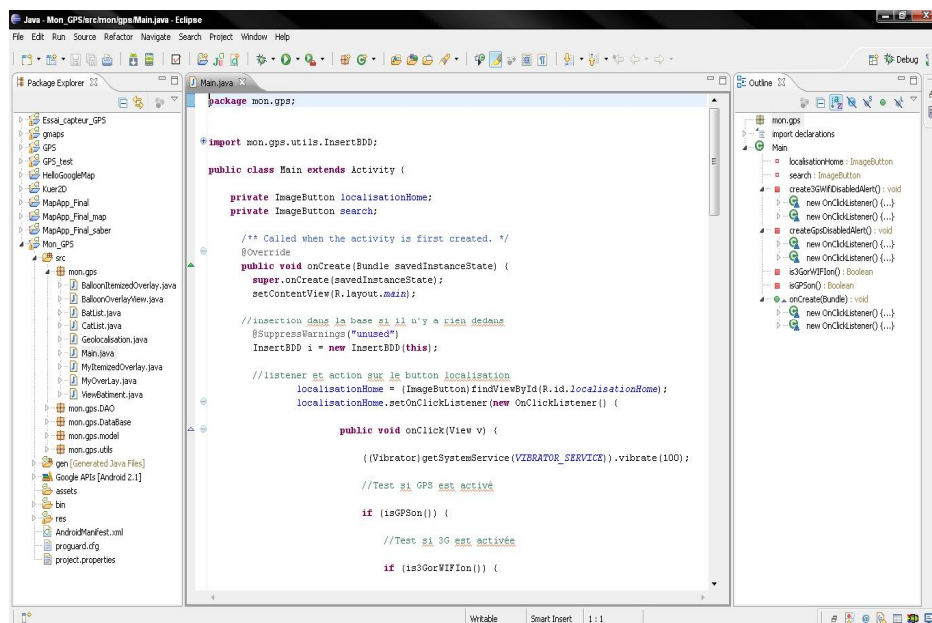


Figure IV.1 : Plate forme « Eclipse »

II.5. Le plugin ADT :

Le plugin ADT pour Eclipse permet de créer et de déboguer facilement et rapidement les applications :

- Il fournit un nouveau gestionnaire de projet, qui aide à initialiser tous les fichiers de base dont vous avez besoin pour une nouvelle application Android.
- Il fournit un éditeur Android qui aide à écrire des fichiers XML valide pour Android.
- Il permet aussi d'exporter votre projet dans un fichier APK signé. Il est important de noter que pour être installé sur un système Android, tout package APK doit être préalablement signé.

II.6. Software Development Kit (SDK):

Un SDK est un kit de développement, propre à chaque système d'exploitation. Il est, entre autre, utilisé pour développer des applications sous Android, il est gratuitement mis à la disposition des développeurs par Google. Le développement pour la plate-forme Android nécessite une connaissance optimale en programmation **Java** et se fait via d'autres logiciels tels que **Eclipse IDE**.

II.7. Le SGBD MySQL :

Le SGBD MySQL est un système de gestion de base de données relationnelles SQL développé dans un souci de performances élevées, il est multi-thread, multi-utilisateurs. C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui est faite dans un produit libre (open-source) ou dans un produit propriétaire.

Le couple PHP/MySQL est largement utilisé pour la création de sites web dynamiques et proposé par la majorité des hébergeurs.

Le SGBD MySQL est fréquemment utilisé avec une interface de gestion appelée phpmyadmin qui permet à un non-initié de générer des requêtes SQL sur une base de données.



Figure IV.2 : Plate forme client/server

III. Langages de programmation:

III.1. Le langage coté client :

✓ Le langage JAVA :

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

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

III.2. Le langage coté serveur :**✓ Le langage PHP :**

PHP est un langage de scripts cote serveur qui a été conçu spécifiquement pour le Web. Dans une page html, on peut inclure du PHP qui sera exécuté chaque fois qu'un visiteur affichera la page. Ce code PHP est interprété au niveau du serveur Web et génère du code html ou toute autre donnée affichable dans le navigateur de l'utilisateur.

IV. Fonctionnement de notre application :

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.

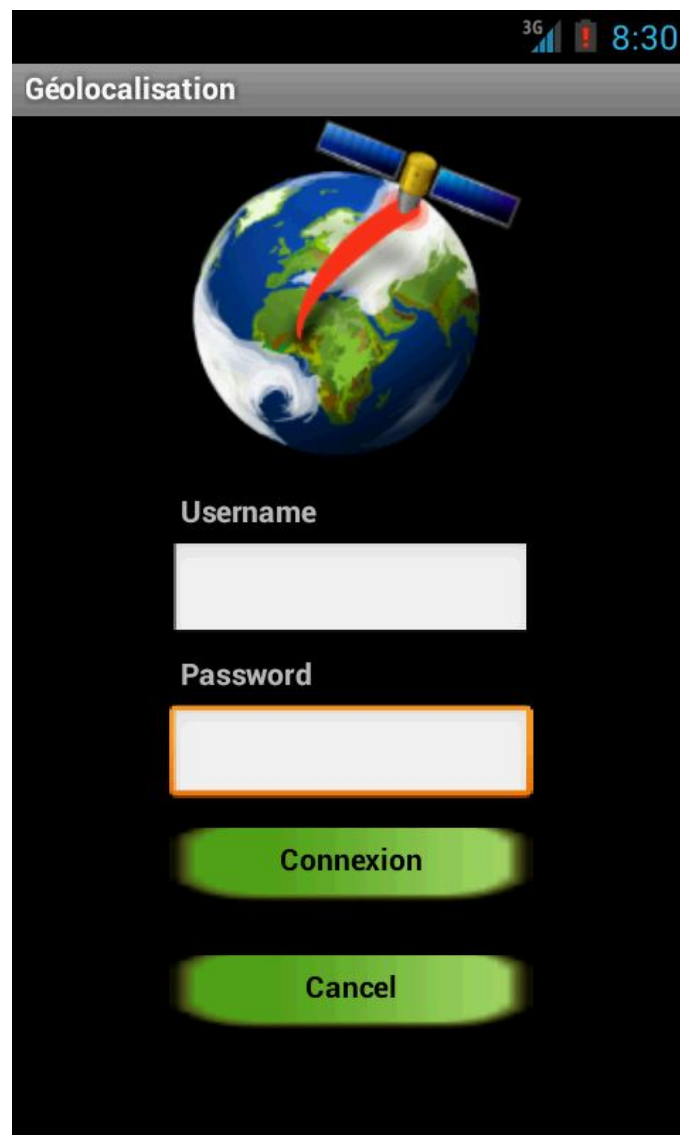
IV.1. L'écran d'authentification :

Figure IV.3 : Interface d'authentification

- 1 : Zone texte pour saisir l'identifiant.
- 2 : Zone texte pour saisir le mot de passe.
- 3 : Bouton de connexion au compte de l'utilisateur.
- 4 : Bouton pour retourner à l'écran d'accueil de terminal.

IV.2. L'écran d'accueil d'utilisateur :



Figure IV.4 : Interface d'accueil

Lors d'un clic sur le bouton « Connexion » de la page d'authentification, l'utilisateur se connecte à son propre compte.

- 1 : Nom et Prénom de l'utilisateur

- 2 : Bouton qui affiche la liste des véhicules.
- 3 : Bouton pour géolocaliser la position de l'utilisateur
- 4 : Bouton pour modifier le mot de passe
- 5 : Bouton de déconnexion.

IV.3. L'écran de la liste des véhicules



Figure IV.5 : Interface de la liste des véhicules

Lors d'un clic sur le bouton «liste des véhicules» de la page d'accueil d'utilisateur, la liste des véhicules de l'utilisateur s'affiche.

1 : Sélectionner le véhicule à géolocaliser.

2 : Bouton pour afficher la position du véhicule sur la map.

IV.4. L'écran de géolocalisation d'un véhicule :



Figure IV.6 : Interface de géolocalisation d'un véhicule

Après la sélection d'un véhicule dans la page liste des véhicule on click sur le bouton « Localiser », la position du véhicule s'affiche sur une map.

1 : la position de véhicule.

IV.4. L'écran de modification de mot de passe :

Lors d'un clic sur le bouton « Modifier le mot de passe » de la page d'accueil d'utilisateur :

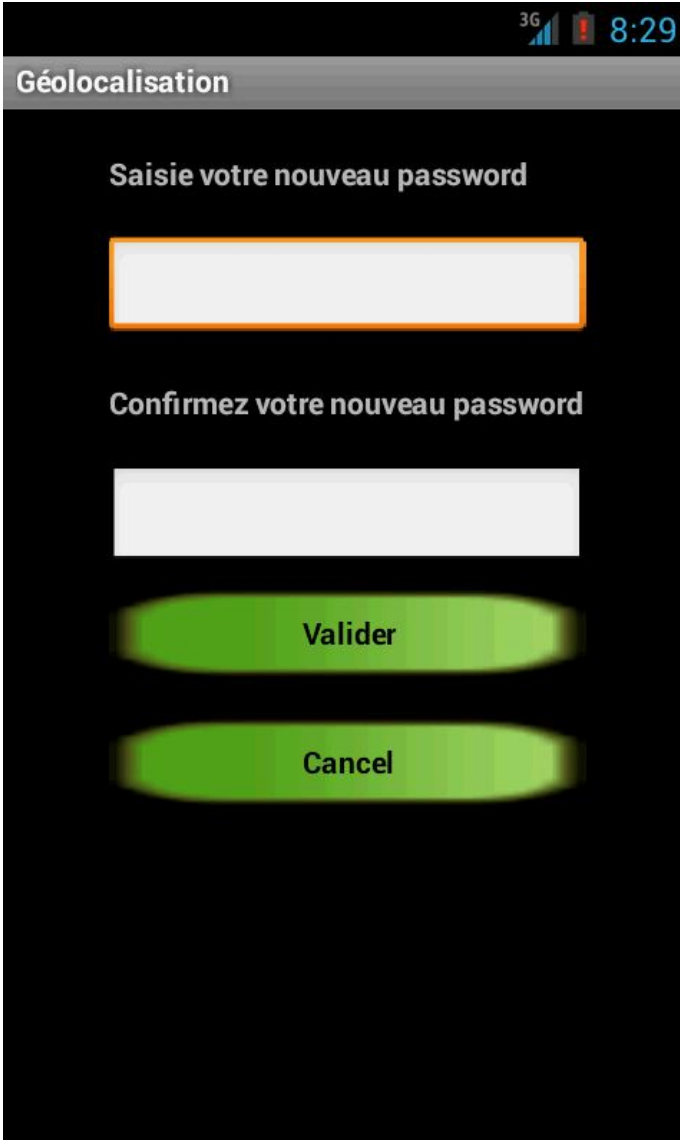
The image shows a mobile application interface for password modification. At the top, there is a status bar with '3G' signal, a battery icon, and the time '8:29'. Below this is a header bar labeled 'Géolocalisation'. The main content area has a black background. It contains two white text input fields. The first field is preceded by the text 'Saisie votre nouveau password'. The second field is preceded by the text 'Confirmez votre nouveau password'. Below the input fields are two green, pill-shaped buttons. The top button is labeled 'Valider' and the bottom button is labeled 'Cancel'.

Figure IV.7 : Interface de modification de mot de passe

- 1 : Zone texte pour saisir le nouveau mot de passe.
- 2 : Zone texte pour confirmer le nouveau mot de passe.
- 3 : Bouton pour enregistre le nouveau mot de passe dans la base de donnée.
- 4 : Bouton pour retourner à la page d'accueil de l'utilisateur.

V. Conclusion :

Au niveau de ce chapitre nous avons présenté l'environnement de développement et l'implémentation de notre application. Nous avons aussi décrit quelques interfaces d'utilisation.

Conclusion Générale



Conclusion Générale

L'élaboration de notre travail était dans le but de concevoir une application dédiée aux terminaux mobiles disposants de la plateforme Android. Cette application permet au propriétaire du téléphone la localisation en temps réel de tous ses véhicules équipés d'un boîtier de localisation. Pour ce faire, on a eu recours à différentes technologies et outils de localisation et de communication avec la base de données distante, jugés nécessaires pour aboutir à l'objectif de l'application développée.

Notre application ainsi réalisée permet de :

- ✓ Afficher les véhicules correspondants au client.
- ✓ Modifier le mot de passe.
- ✓ Localiser en temps réel les véhicules du client.

Ce projet nous a permis d'acquérir des nouvelles connaissances à propos d'UML, Eclipse, SQL, PHP, et de maîtriser les langages de programmation Java qui seront certainement utiles dans nos futures vies professionnelles.

Enfin, on ne peut sans doute pas affirmer que notre travail est complet car plusieurs améliorations peuvent être apportées. Mais nous espérons au moins qu'on a réussi à réaliser une simple application mobile Android fonctionnelle qui satisferait les besoins des futurs utilisateurs et qui conviendrait à leurs attentes.

Les perspectives :

En guise de perspectives on envisage d'enrichir l'application développée avec de nouvelles fonctionnalités comme :

1. Service livraison

Intégration d'une fonctionnalité qui permettrait à l'utilisateur de l'application de géolocaliser un véhicule, et de tracer un itinéraire entre sa position et le future point de livraison. Ce qui va lui permettre de choisir le véhicule le plus proche et ainsi avoir un gain de temps pour les livraisons.

2. Service dépannage

Intégration d'une fonctionnalité qui permettrait de positionner l'utilisateur de l'application. De cette façon si un véhicule tombe en panne, l'utilisateur peut tracer un itinéraire entre sa position et la position du véhicule en panne.

3. Service historique

Intégration d'une fonctionnalité qui fournirait le service historique, ce qui permettra à l'utilisateur de réaliser un traçage complet de l'utilisation du véhicule pendant une période définie par l'utilisateur de l'application.

Annexe A



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

Installation de l'Eclipse [N1].

2. Installation du SDK Android [N2].

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

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

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

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

Annexe B

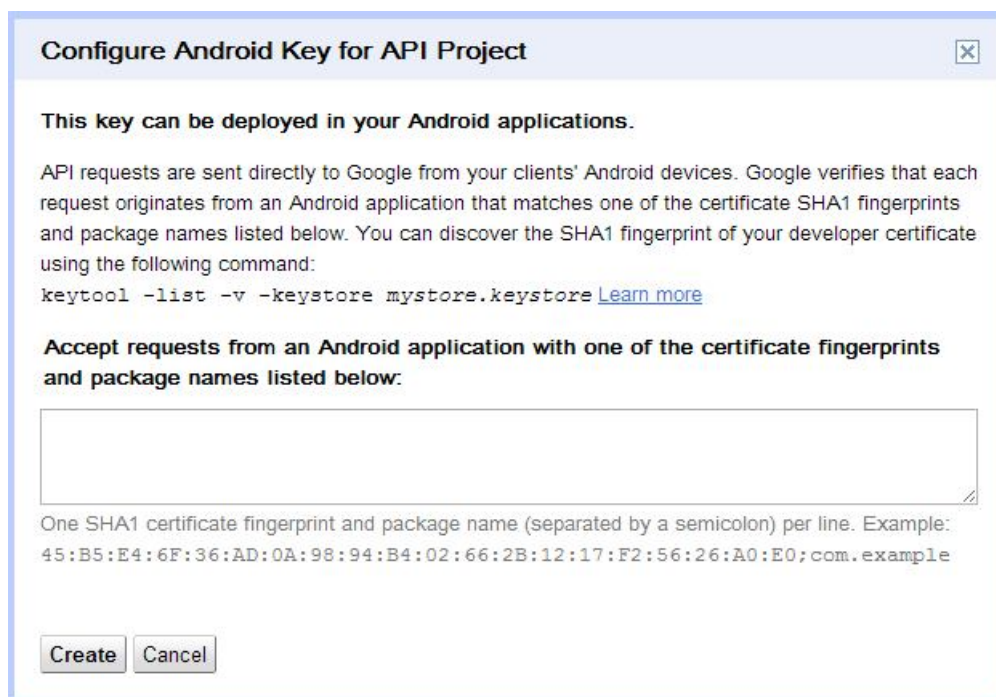


Annexe B : Obtention d'une clé pour utiliser Google Maps.

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

Pour cela, la première étape consiste à se rendre sur le site proposé par Google (**Console Google**). Il me faudra également un compte Google pour compléter l'opération. Enfin, pour obtenir cette clé, l'interface du site nous demande une empreinte SHA1 d'un certificat:

<https://code.google.com/apis/console/#project:128392069783:access>



Configure Android Key for API Project

This key can be deployed in your Android applications.

API requests are sent directly to Google from your clients' Android devices. Google verifies that each request originates from an Android application that matches one of the certificate SHA1 fingerprints and package names listed below. You can discover the SHA1 fingerprint of your developer certificate using the following command:

```
keytool -list -v -keystore mystore.keystore Learn more
```

Accept requests from an Android application with one of the certificate fingerprints and package names listed below:

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

Create Cancel

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

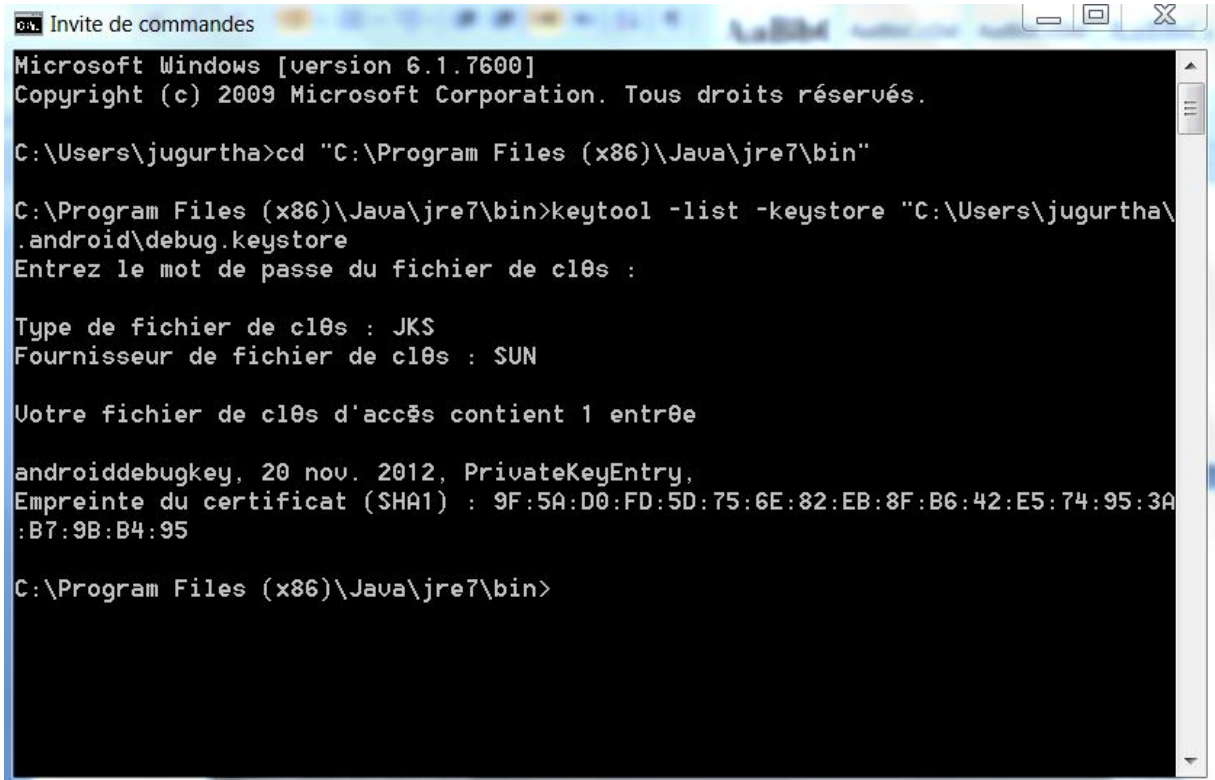
Eclipse crée le fichier **debug. Keystore** lors de la compilation du projet, ce fichier étant stocké dans le répertoire **.android** dont le chemin varie en fonction du système d'exploitation. Pour un système sous Windows 7 par exemple le chemin est :

✓ C:\Users\Utilisateurs\Nomd'Utilisateurs\.android.

La clé générée pour utiliser l'API Google Maps est basée sur ce certificat de débogage. Pour en obtenir l'empreinte (et en se basant sur un système 7 logué en administrateur), il faut utiliser la commande suivante :

Annexe B

- ✓ `keytool -list -keystore "C:\Users\jugurtha\.android\debug.keystore"`



```
Invite de commandes
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\jugurtha>cd "C:\Program Files (x86)\Java\jre7\bin"

C:\Program Files (x86)\Java\jre7\bin>keytool -list -keystore "C:\Users\jugurtha\.android\debug.keystore"
Entrez le mot de passe du fichier de clés :

Type de fichier de clés : JKS
Fournisseur de fichier de clés : SUN

Votre fichier de clés d'accès contient 1 entrée

androiddebugkey, 20 nov. 2012, PrivateKeyEntry,
Empreinte du certificat (SHA1) : 9F:5A:D0:FD:5D:75:6E:82:EB:8F:B6:42:E5:74:95:3A:B7:9B:B4:95

C:\Program Files (x86)\Java\jre7\bin>
```

De retour dans la **console Google**, vous pouvez maintenant saisir l’empreinte SHA1 trouvé, y ajouter un « ; » et saisir le nom du package de votre application afin de créer la clé.

Key for Android apps (with certificates)

API key:	AIzaSyD21yYcjcWaNFWJgA5UXKObSfaFz9LRoo
Android apps:	9F:5A:D0:FD:5D:75:6E:82:EB:8F:B6:42:E5:74:95:3A:B7:9B:B4:95;com.exemple.com.gps
Activated on:	Jun 9, 2013 5:33 PM
Activated by:	lomani.jugurtha@gmail.com – you

Glossaire



Glossaire :

ADT : Abstract Data Type , Plugin Android pour Eclipse .

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

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

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

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

PDA : Personal Digital Assistant , Appareil numérique portable .

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

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

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

Bibliographie



La bibliographie :

- [1] : <http://fr.wikipedia.org/wiki/G%C3%A9olocalisation>
- [2] : <http://anr-prodige.com/index.php?n=Technologies.Geolocalisation>
- [3] : Kevin VUILLARD, Laurent DUPLAT, Laurène HARDOUIN, Cyril RESSY, Jimmy SOURDIN. Livre Blanc « Géolocalisation des Smartphones dans le domaine culturel et patrimonial ».
- [4] : <http://www.companeo.com/logiciel-de-geolocalisation.htm#0>
- [5] : Jean-Baptiste HENRY, Ingénieur Géomètre-Topographe «Service régional de traitement d'Image et de télédétection ».
- [6] : Luc AEBI, « Découvrir L'univers du GPS », premier semestre 2007
- [7] : <http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/index.html#cara>
- [8] : http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_comp.html
- [9] : <http://www.commentcamarche.net/>
- [10] : <http://www.mti.epita.fr/blogs/2010/08/03/introduction-a-la-programmation-sous-android/>
- [11] : <http://www.memoireonline.com/03/12/5548/Rapport-de-stage-sur-le-projet-Locate-my-car-google-map-android.html>
- [12] : <http://www.pointgphone.com/android/emulateur/>
- [13] : <http://code.google.com/intl/fr-FR/android/maps-api-signup.html>.
- [14] : [http:// www.mti.epita.fr/blogs/2010/08/03/introduction-a-la-programmation-sous-android/](http://www.mti.epita.fr/blogs/2010/08/03/introduction-a-la-programmation-sous-android/).
- [15] : <http://developer.android.com/guide/topics/location/index.html>.
- [16] : <http://developer.android.com/guide/location/index.html>.
- [N1] : [http:// www.eclipse.org/downloads/](http://www.eclipse.org/downloads/).
- [N2] : [http:// developer.android.com/index.html](http://developer.android.com/index.html).
- [N3] : [https:// dl-ssl.google.com/android/eclipse/](https://dl-ssl.google.com/android/eclipse/).