

Résumé

De nos jours, l'informatique a atteint une extraordinaire évolution technologique dans différents domaines (réseaux informatiques, base de données, web, etc.) en particulier l'informatique mobile. Cette évolution est essentielle pour remédier aux problèmes exposés dans la vie quotidienne.

Dans ce travail, nous avons réalisé un système d'information géographique destiné aux conducteurs dans le but de leur offrir un choix d'emprunter le chemin le plus fluide. Pour la développer, nous avons utilisé plusieurs technologies parmi les quelles, nous citons : la plateforme Android en utilisant son outil de développement SDK Android, le format léger d'échange de données "JSON", le langage UML pour réaliser l'étude conceptuelle du système, apache comme serveur web, MySQL comme serveur de base de données, JAVA et PHP comme langages de programmation.

Mots clés : MySQL , UML, Java, trafic routier, Android, SIG, fluide.

Abstract

Nowadays, data processing reached extraordinary technological development in various fields (networks data processing, bases data, the Web, etc.)in particular mobile data processing.This evolution is essential to cure the problems exposed in the current life.

TrafRout is an application of decision-making aid, in other words, a geographical information system intended for the drivers to offer a choice to borrow to them the most fluid section.

To develop it, we used several technologies among them, we quote :the Android platform in using its development tool SDK Android, the light format to exchange data "JSON", language UML to realize the conceptual study of the system, apache like Web server, MySQL like Java and database server to manage them data and the accounts user.

Keywords : MySQL, UML, Java, road traffic, Adroid, GIS, fluid.

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

Université Abderrahmane MIRA de Béjaïa

Faculté des Sciences Exactes

Département Informatique



Mémoire de fin de cycle

En vue d'obtention d'un diplôme Master en Informatique

Option : Administration et sécurité des réseaux

Thème

**Conception et réalisation d'une application mobile à base d'un
Système d'Information Géographique pour la régulation du trafic
routier.**

Présenté par

M^{lle} BOUFARIK Wahiba

M^{lle} BOUHARIS Warda

Soutenu devant le jury composé de :

Président *M^r AISSANI Sofiane*

Promoteurs *M^{lle} BATTAT Nadia*

M^r ALLAM Khaled

Examineurs *M^r DJEBARI Nabil*

M^{lle} GHIDUCHE Kahina

Promotion 2012/2013

Remerciements

Avant toute chose, nous tenons à remercier le bon Dieu de nous avoir aidé, donné la force et la volonté à mener ce travail jusqu'à sa fin.

En premier lieu, nous tenons à transmettre l'expression de nos plus sincères gratitudee à notre promotrice et co-promoteur en l'occurrence Melle N. BATTAT et Mr K .ALLEM pour leurs conseils et avoir été très disponibles tout au long de la réalisation de ce mémoire, ainsi pour leurs inspirations, leurs aides et le temps qu'ils nous ont consacré et sans qui ce mémoire n'aurait jamais vu le jour.

Nous tenons également à remercier tout le personnel de cévital de Bejaia, surtout Mr AIT.IKHLEF Nourdine pour l'accueil chaleureux qu'il nous a réservé durant notre stage et son aide précieuse.

Nous remercions aussi Mr bouali Athmene pour son soutien, et son orientation dans le bon sens quant à l'élaboration de ce projet.

Nous tenons à remercier chaleureusement tous les membres de jury qui ont bien voulu juger ce travail et ce malgré leur plan de charge important en cette fin d'année universitaire.

Nous tenons aussi à remercier toute la promotion master 2 informatique recherche et professionnel 2012-2013. Ainsi que tous nos enseignants et les membres du département informatique de l'université ABDERRAHMANE MIRA.

Enfin, nos sincères sentiments vont à tous ceux qui nous ont aidé, de près ou de loin, pour la réalisation de ce modeste travail.

Dédicaces

Je dédie ce modeste travail à :

Ma mère qui a tant sacrifié pour moi ;

Vous ma promotrice et mon co-promoteur pour votre aide précieuse ;

Mes frères et sœurs ;

Mes oncles et tantes, cousins et cousines ;

Tous mes amis(es) ;

Toute notre promotion master 2 informatique recherche et professionnel 2012-2013 ;

Tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Wahiba

Dédicaces

Je dédie ce modeste travail à :

Mes chers parents et grands parents qui su m'aider avec leur profond amour, et qui ont toujours été a mes cotés pour me soutenir, me conseiller tout au long de mon parcours étudiantin, particulièrement pendant la réalisation de ce mémoire, je leur dis merci, que Dieu les garde et les protège ;

Vous ma promotrice et mon co-promoteur pour votre aide précieuse ;

Mes adorables sœurs ;

Ma grande famille : oncles, tantes, cousins et cousines sans exception ;

Tous mes amis(es) ;

Tous ceux qui ont participé de près ou de loin à la réalisation de notre travail ;

Tous ceux qui me connaissent.

Warda

Table des matières

Table des matières	i
Table des figures	vi
Liste des tableaux	vii
Liste des abréviations	vii
Introduction générale	1
1 Généralités sur le trafic routier	2
1.1 Introduction	3
1.2 Le trafic routier	3
1.2.1 Historique	3
1.2.2 Définition du trafic routier	3
1.2.3 Variables élémentaires du trafic routier	4
1.2.3.1 L'écart du temps inter-véhiculaire	4
1.2.3.2 Le débit	5
1.2.3.3 La concentration (ou la densité)	5
1.2.3.4 Le taux d'occupation	5
1.2.3.5 La vitesse instantannée des véhicules et les vitesses moyennes	5
1.2.3.6 Le diagramme fondamental	6
1.2.4 Réseau routier	7
1.2.5 Les différents modèles du trafic	7
1.2.5.1 Modèle macroscopique	8
1.2.5.2 Modèle microscopique	8
1.2.5.3 Modèle mésoscopique	8
1.2.6 La congestion	9
1.3 Conclusion	9
2 Les Systèmes d'Information Géographiques(SIG)	10
2.1 Introduction	11

2.2	Historique des SIG	11
2.3	Définition d'un SIG	11
2.4	Les avantages d'un SIG	12
2.5	Questions auxquelles peuvent répondre les systèmes d'information géographique	12
2.6	Objectifs visés par les SIG	12
2.7	Domaines d'application d'un SIG	14
2.7.1	La représentation géographique	14
2.7.2	Gestion géographique	14
2.7.3	Inventaire et gestion des éléments naturels	14
2.7.4	Environnement	14
2.7.5	Agriculture de précision	14
2.7.6	Géomarketing	14
2.8	Classification des SIG	14
2.8.1	Les SIG généralistes bureautiques	15
2.8.2	Les SIG généralistes de gestion	15
2.8.3	Les SIG généralistes de métiers	15
2.9	Facteurs liés aux SIG	15
2.9.1	Les couches	15
2.9.2	Les références géographiques	16
2.9.3	Les modèles géographiques utilisés par un SIG	18
2.9.3.1	Le mode raster (appelé aussi mode maillé)	18
2.9.3.2	Le mode vecteur	19
2.10	Fonctionnalités des SIG	19
2.10.1	Abstraction	19
2.10.2	Acquisition	19
2.10.3	Archivage	20
2.10.4	Analyse	20
2.10.5	Affichage	20
2.11	Composantes principales d'un SIG	20
2.11.1	Matériels	20
2.11.2	Logiciels	20
2.11.3	Données	21
2.11.4	Utilisateurs	21
2.11.5	Méthodes	21
2.12	SIG pour la régulation du trafic routier	22
2.13	Conclusion	22

3	Analyse et conception	23
3.1	Introduction	24
3.2	Présentation du langage UML	24
3.2.1	Définition	24
3.2.2	Historique	24
3.2.3	Intérêt d’UML	25
3.2.4	Présentation générale des diagrammes d’UML 2	25
3.2.4.1	Les diagrammes structurels	25
3.2.4.2	Les diagrammes de comportement	25
3.2.5	Processus associés à UML	26
3.2.6	Le processus unifié (UP)	26
3.2.6.1	Les phases d’UP	27
3.2.6.1.1	Inception (Lancement)	27
3.2.6.1.2	Élaboration	27
3.2.6.1.3	Construction	27
3.2.6.1.4	Transition	27
3.2.7	Modélisation du notre système TrafRout	27
3.2.7.1	Première phase (Inception)	28
3.2.7.1.1	Présentation des besoins fonctionnels	29
3.2.7.1.2	Présentation des besoins non fonctionnels	33
3.2.7.2	Deuxième phase (Élaboration)	33
3.2.7.2.1	Les diagrammes de séquences	34
3.2.7.2.2	Diagrammes d’activité	39
3.2.7.2.3	Diagramme de classes	41
3.3	Conclusion	45
4	Réalisation et test	46
4.1	Introduction	47
4.2	Les outils de développement	47
4.3	Environnement logiciel	47
4.3.1	Le système d’exploitation Android	47
4.3.1.1	Les terminaux mobiles	47
4.3.1.1.1	Définition des terminaux mobiles	47
4.3.1.1.2	Les différents types de terminaux mobiles	47
4.3.1.2	Définition d’Android	49
4.3.1.3	Architecture d’Android	49
4.3.2	Outils de développement	51
4.3.2.1	Le SDK Android	51

4.3.2.2	L'émulateur de téléphone	51
4.3.2.3	IDE Eclipse	51
4.3.2.4	IDE PHPEdit	52
4.3.2.5	EasyPHP	52
4.3.2.6	JSON	52
4.3.3	Les langages de programmation	53
4.3.3.1	Java	53
4.3.3.2	PHP	53
4.4	Environnement matériel	54
4.4.1	Architecture matérielle	54
4.5	Interfaces graphiques de TrafRout	54
4.5.1	Interfaces administrateur	55
4.5.1.1	Formulaire d'authentification	55
4.5.1.2	Ajout des lieux	55
4.5.1.3	Ajout des routes	57
4.5.2	Interfaces utilisateur	57
4.5.2.1	Page d'authentification	58
4.5.2.2	Interface d'accueil	59
4.5.2.3	Sélection des lieux	60
4.5.2.4	Trouver le chemin optimal	61
4.6	Test	62
4.7	Conclusion	65
	Conclusion et perspectives	66
	Bibliographie	67
	Glossaire	70

Table des figures

1.1	L'écart du temps inter-véhiculaire [22].	4
1.2	Le diagramme fondamental [22].	6
1.3	Représentation du réseau routier par un graphe orienté [8].	7
2.1	Les différentes couches des SIG [14].	16
2.2	Représentation des données spatiales par les trois types d'objets graphiques [17]	16
2.3	Exemple d'une image enregistrée.	17
2.4	Exemple d'un texte.	17
2.5	Exemple d'une carte.	18
2.6	Représentation des deux modes raster et vecteur[28]	19
2.7	Les composantes d'un SIG [18]	21
3.1	Le diagramme de contexte.	28
3.2	Relation d'inclusion.	29
3.3	Relation d'extension.	30
3.4	Diagramme de cas d'utilisation.	31
3.5	Diagramme de séquence du cas d'utilisation " s'authentifier ".	35
3.6	Diagramme de séquence du cas d'utilisation " Trouver le chemin optimal ".	36
3.7	Diagramme de séquence du cas d'utilisation " Ajout d'un lieu ".	37
3.8	Diagramme de séquence du cas d'utilisation " Suppression d'un lieu ".	38
3.9	Diagramme de séquence du cas d'utilisation " Ajout d'une route ".	39
3.10	Diagramme d'activité du cas d'utilisation " s'authentifier ".	40
3.11	Diagramme d'activité du cas d'utilisation " Ajout d'une route ".	41
3.12	Diagramme de classes	44
4.1	Les smartphones [38].	48
4.2	Les tablettes tactiles [39].	48
4.3	Les assistants personnels ou PDAs [40]	49
4.4	Architecture d'Android.	50
4.5	Interface de l'émulateur Android.	51
4.6	Logo de l'IDE Eclipse [33].	52

4.7	Architecture trois tiers de TrafRout.	54
4.8	Formulaire d'authentification.	55
4.9	Ajout d'un lieu.	56
4.10	Ajout d'une route.	57
4.11	Icone de TrafRout.	58
4.12	Page d'authentification.	59
4.13	Interface d'accueil.	60
4.14	La liste des lieux.	61
4.15	Représentation de la route sur la carte.	62
4.16	Test du cas d'utilisation " s'authentifier ".	63
4.17	Test d'ajout d'un lieu.	64
4.18	Test d'ajout d'une route.	65

Liste des tableaux

3.1	Tableau des messages échangés par l'utilisateur et le système.	28
3.2	Tableau des messages échangés par l'administrateur et le système.	29
3.3	Tableau des messages échangés par GoogleMaps et le système.	29
3.4	Description du cas d'utilisation "s'authentifier".	32
3.5	Description du cas d'utilisation "Trouver le chemin optimal".	33
3.6	Dictionnaire de données.	43

Liste des abréviations

- * **ADT** : **A**ndroid **D**evelopment **T**ools.
- * **API** : **A**pplication **P**rogramming **I**nterface.
- * **APK** : **A**ndroid **P**ackage **K**it.
- * **AVD** : **A**ndroid **V**irtual **D**evice.
- * **DVD** : **D**igital **V**ersatile **D**isc.
- * **GPRS** : **G**eneral **P**acket **R**adio **S**ervice.
- * **GPS** : **G**lobal **P**ositioning **S**ystem.
- * **HTML** : **H**yper**T**ext **M**arkup **L**anguage.
- * **IDE** : **I**ntegrated **D**evelopment **E**nvironment.
- * **JSON** : **J**ava **S**cript **O**bject **N**otation .
- * **MP3** : **M**edia **P**layer **3**.
- * **OMG** : **O**bject **M**anagement **G**roup.
- * **OMT** : **O**bject **M**odeling **T**echnique.
- * **OOSE** : the **O**lliptic **O**urve **S**ntegrated **E**ncryption.
- * **OS** : **O**perating **S**ystem.
- * **PC** : **P**ersonal **C**omputer.
- * **PCI** : **P**lan **C**adastral **I**nformatisé.
- * **PDA** : **P**ersonal **D**igital **A**ssistant.
- * **PHP** : **H**ypertext **P**reprocessor.
- * **PLU** : **P**lan de **L**ocal d'Urbanisme.
- * **RAM** : **R**andom **A**ccess **M**emory.
- * **RUP** : **R**ational **U**nified **P**rocess.
- * **SDK** : **S**oftware **D**evelopment kit.
- * **SIG** : **S**ystème d'**I**nformation **G**éographique.
- * **SQL** : **S**tructured **Q**uery **L**anguage.
- * **UML** : **U**nified **M**odeling **L**anguage.

- * **UP** : Unified **P**rocess.
- * **WAMP** : Windows **A**pache **M**ysql **P**HP.
- * **Wifi** : **W**ireless **F**idelity.
- * **XML** : e**X**tensible **M**arkup **L**anguage.
- * **XP** : e**X**treme **P**rogramming.

Introduction générale

Depuis quelques années, le trafic routier attire tout particulièrement l'attention de l'opinion publique. Avec la montée en puissance des préoccupations environnementales et devant l'urgence de la situation, la gestion routière est de moins en moins acceptée par les parties prenantes du phénomène et notamment par les usagers vu qu'ils perdent de nombreuses heures. Le phénomène de la congestion mène aussi à perdre la productivité dans les villes économiques comme la ville de Béjaia, à cause des retards des personnes à l'arrivée sur leurs lieux de travail et les livraisons tardives des approvisionnements et les services rendus qui peuvent être annulés. La solution conventionnelle revient à augmenter le nombre d'infrastructures routières mais cette solution souffre d'une indisponibilité d'espace.

Afin d'améliorer les conditions de trafic sans pour autant élargir les voies existantes, les gestionnaires de réseaux font désormais l'usage de dispositifs de régulation à l'image de la régulation d'accès ou de la régulation dynamique des vitesses. Cependant ces solutions n'ont pas été suffisantes pour réguler le trafic routier.

Afin de résoudre ce problème, nous avons opté pour le développement d'un système d'information géographique mobile sous Android qui peut aider à la régulation du trafic routier en proposant aux demandeurs, en particulier aux conducteurs, d'emprunter la route estimée la plus fluide à travers l'affichage cartographique de ce chemin meilleure présentation qu'une par rapport à un tableau rempli de valeurs numériques. Ce SIG mobile permet aussi d'afficher la plus courte route et d'autres routes possibles.

Dans ce but, nous allons opter pour le langage de modélisation UML en suivant le processus itératif UP et les langages de programmation Java et PHP.

Le mémoire est structuré de la façon suivante : les principaux concepts de bases sur la théorie du trafic routier font l'objet du premier chapitre.

Le deuxième chapitre intitulé " généralités sur les systèmes d'information géographiques " donne un panorama général des SIG, une brève introduction sur les objectifs généraux des SIG, nous rappellerons rapidement l'histoire du développement des systèmes d'information géographiques, ainsi que les fonctionnalités rencontrées dans les systèmes d'information.

Ensuite, le troisième chapitre va présenter la modélisation de notre système en suivant le processus itératif UP.

Notre SIG mobile sera testé et validé au travers de la réalisation du modèle proposé. Ce modèle est développé à l'aide de différents outils présentés dans le dernier chapitre.

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

Chapitre 1

Généralités sur le trafic routier

1.1 Introduction

La croissance permanente de la population dans les villes a augmenté le nombre des véhicules et aussi la nécessité d'une circulation fluide des personnes et des marchandises dans les réseaux urbains. Ainsi, le problème de la congestion du trafic est devenu l'un des principaux problèmes à résoudre par les systèmes de régulation du trafic d'aujourd'hui.

Particulièrement, au cours de la dernière décennie, la congestion du trafic a attiré une attention significative à cause de la crise énergétique mondiale et les préoccupations environnementales. Malgré qu'il existe de bons systèmes de gestion de la route et des infrastructures routières suffisantes pour le transport, nous faisons encore face à une congestion du trafic, en particulier, aux carrefours à feux [22].

En effet, les exploitants de réseaux de transport ont besoin d'outils d'observation du trafic pour améliorer le fonctionnement de leur réseau.

Dans ce chapitre, nous introduisons les variables élémentaires utilisées et les mesures usuelles pour l'analyse du trafic.

1.2 Le trafic routier

Le développement considérable de la circulation automobile et l'accroissement constant de la motorisation ont rendu nécessaire l'élaboration des théories du trafic, capables de décrire, d'expliquer et de prévoir les interactions entre les véhicules ainsi que les mouvements d'ensemble sur les infrastructures routières [22].

1.2.1 Historique

Historiquement, la théorie du trafic a vu le jour avec la description du trafic routier. Vers la fin des années quarante, plusieurs études ont été développées pour prendre en compte le comportement du conducteur. Les années cinquante marquent l'arrivée de nouvelles théories (hydrodynamique, mécanique des fluides, automates cellulaires, etc.) qui à ce jour sont encore utilisées et font l'objet de plusieurs études et améliorations de la théorie du trafic [23].

1.2.2 Définition du trafic routier

Un trafic routier est un ensemble de véhicules utilisant une infrastructure routière. L'infrastructure est un ensemble d'éléments structuraux interconnectés qui fournissent le cadre pour supporter la totalité de la structure du trafic. Par exemple, pour les réseaux routiers et autoroutiers, l'infrastructure contient les structures : ponts, pontceaux, signalisation et marquages, systèmes électriques (lampes d'éclairage public et de circulation) et les traitements de bord (bordures, trottoirs, aménagement paysager). Les mobiles sont les individus qui utilisent l'infrastructure, principalement dit, des véhicules. Les piétons ne sont pas considérés car leurs mouvements sont supposés dépendre de celui des mobiles. Notons que les véhicules à l'arrêt ou en stationnement, au sens défini par le code de la route, sont considérés comme faisant partie de l'infrastructure et ne sont généralement pas pris en compte. Autrement dit, seuls les véhicules en mouvement sur la route et en interaction avec la signalisation sont considérés [22].

1.2.3 Variables élémentaires du trafic routier

Dans une optique de compréhension de l'évolution du trafic routier, différents types d'information sont nécessaires tels que [8,22] :

1.2.3.1 L'écart du temps inter-véhiculaire

Le premier concept introduit est l'écart du temps inter-véhicule (en anglais, "the headway"). En un point donné de la route, c'est le temps qui s'écoule entre les instants de passage de deux véhicules successifs (voir Figure 1.1).

Cette variable se révèle particulièrement utile dans les études relatives à la sécurité routière et dans de nombreux problèmes de simulation du trafic. Sur autoroute, la "régulation d'intervalles" continue d'être une mesure d'exploitation de plus en plus répandue. Enfin, ce paramètre permet d'appréhender les effets de la composition du trafic sur les conditions d'écoulement de la circulation.

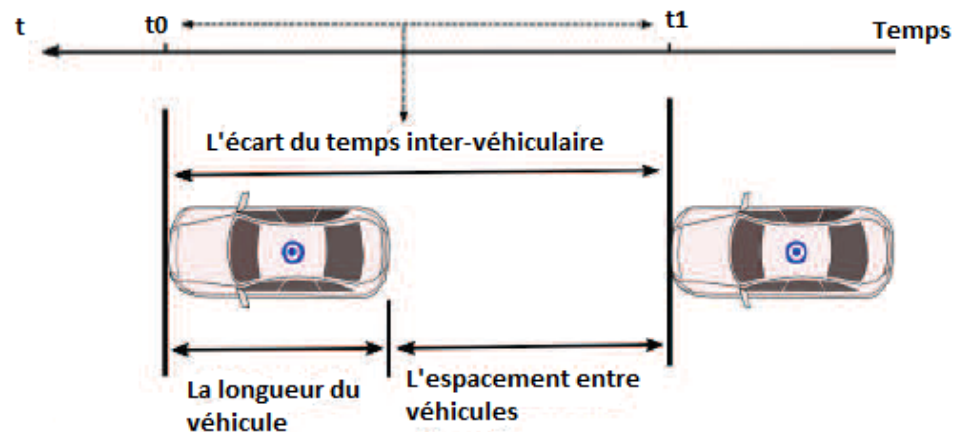


FIGURE 1.1 – L'écart du temps inter-véhiculaire [22].

1.2.3.2 Le débit

Il correspond à la répartition des véhicules dans le temps. On définit le débit moyen $q(t_1, t_2, x)$ au point d'abscisse x entre les instants t_1 et t_2 par le rapport :

$$q(t_1, t_2, x) = \frac{n(t_1, t_2, x)}{t_2 - t_1} \quad (1)$$

Dans lequel $n(t_1, t_2, x)$ désigne le nombre de véhicules passés en x entre les deux instants. Expérimentalement, le débit peut être déterminé par de simples comptages sur la voie.

1.2.3.3 La concentration (ou la densité)

Elle décrit la répartition des véhicules dans l'espace. La concentration moyenne $k(x_1, x_2, t)$ à l'instant t sur une section de route limitée par les points d'abscisses x_1 et x_2 correspond au rapport :

$$k(x_1, x_2, t) = \frac{n(x_1, x_2, t)}{x_2 - x_1} \quad (2)$$

Dans lequel $n(x_1, x_2, t)$ désigne le nombre des véhicules présents sur la section à l'instant t .

Une telle grandeur est par exemple directement observable par photographie aérienne ou encore par caméra vidéo.

1.2.3.4 Le taux d'occupation

Cette variable est couramment très exploitée. Le procédé de mesure du taux d'occupation le plus répandu fait appel à des capteurs (boucles magnétiques) enfouis dans la chaussée et sensibles aux variations du champ magnétique produites par le passage des masses métalliques des véhicules. Grandeur sans dimension, définie par la proportion de temps durant laquelle la boucle est occupée, le taux d'occupation est directement lié à la concentration.

1.2.3.5 La vitesse instantannée des véhicules et les vitesses moyennes

- **Vitesse instantannée d'un véhicule** : pour un même véhicule, l'enregistrement des vitesses instantanées permet de caractériser le profil temporel de la vitesse. Cet enregistrement est utile à l'obtention de divers paramètres. Il est révélateur de la qualité du trafic sur une voie et s'emploie fréquemment dans les calculs relatifs à la consommation d'énergie des véhicules.

- **Vitesse moyenne d'un véhicule** : sur un parcours de durée T , la vitesse moyenne v_{moy} d'un véhicule est définie par :

$$v_{moy} = \frac{\int_0^T v(t) dt}{T} \quad (3)$$

Où $v(t)$ désigne la vitesse instantanée du véhicule à l'instant t .

1.2.3.6 Le diagramme fondamental

Le trafic est supposé homogène et stationnaire. Autrement dit, le débit, la concentration et la vitesse varient peu autour de leurs moyennes. A faible concentration de véhicules, nous observons une vitesse moyenne de flot généralement élevée, appelée vitesse libre ou vitesse désirée u_{max} . A l'inverse, lorsque la concentration k augmente, les espaces entre les véhicules deviennent plus importants et la vitesse u diminue (voir la figure 1.2). Ce constat suggère que la vitesse est une fonction décroissante de la concentration k [22].

Compte tenu de la relation liant les trois variables, la loi d'écoulement du trafic sur une section de route peut donc être formulée par une fonction liant deux des trois variables : débit, concentration et vitesse. Le diagramme fondamental est sensible à de nombreux facteurs comme la géométrie de la route, la nature et la composition du trafic, les conditions météorologiques, les mesures d'exploitation, etc.

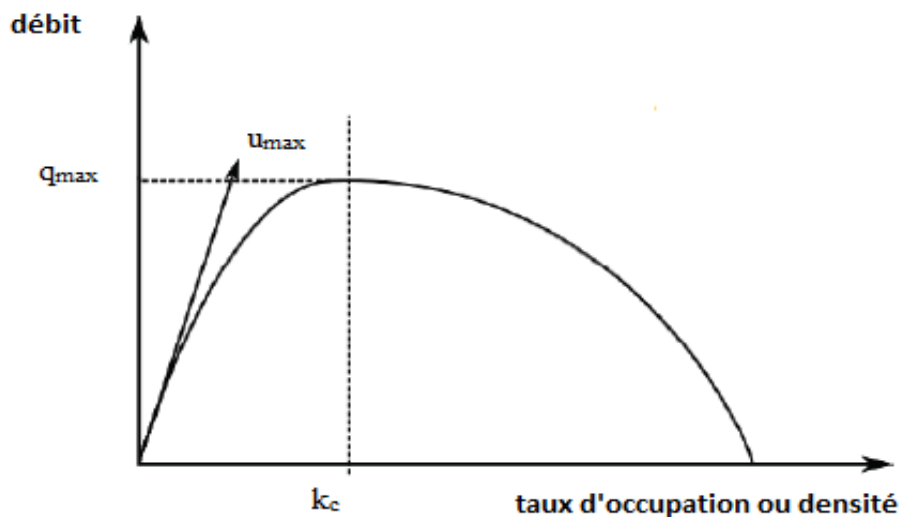


FIGURE 1.2 – Le diagramme fondamental [22].

Les valeurs des variables macroscopiques de trafic comme le débit, la vitesse, le taux d'occupation, la concentration et le temps de parcours qui sont recueillis à l'aide de capteurs déposés sur les axes routiers [8] :

- Les capteurs pneumatiques qui sont constitués d'un câble de caoutchouc relié à un manomètre. Le câble est tendu en travers de la voie ou de la chaussée sur laquelle on veut effectuer la mesure.
- Les boucles électromagnétiques, encore appelées inductives qui représentent le type de capteur le plus souvent utilisé. Chaque boucle électromagnétique est enfouie dans la chaussée sous une voie donnée. La boucle est activée en présence d'un véhicule, désactivée dans le cas contraire. Ce type de capteur permet donc de mesurer directement le taux d'occupation. Par déduction, on obtient le débit et le temps inter-véhiculaire très utilisé pour les applications liées à la sécurité.
- Les caméras vidéo couplées à un système de traitement d'images peuvent également être utilisées. Elles permettent d'obtenir des informations complémentaires comme la longueur des files d'attente par exemple. Il s'agit d'un capteur spatial, à la différence des boucles électromagnétiques et des capteurs pneumatiques.

1.2.4 Réseau routier

Le réseau routier est structuré en routes et autoroutes. Chaque axe de circulation est décomposé longitudinalement en sections, encore appelées tronçons. La chaussée est elle-même constituée de voies[8].

Le réseau routier peut être considéré comme un graphe de nœuds liés par des arcs. La topologie du réseau routier décrit la position des capteurs sur le réseau, comme la figure 1.3 l'illustre.

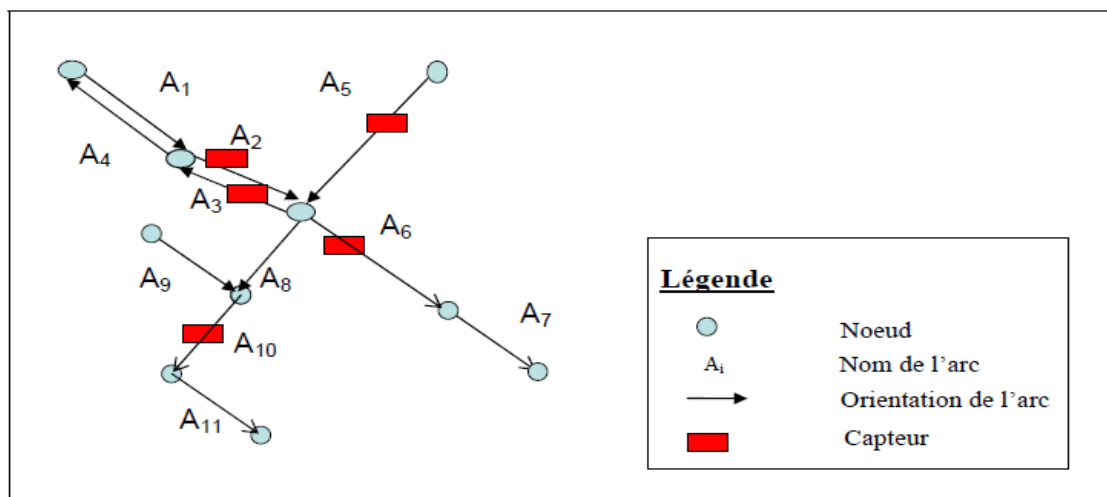


FIGURE 1.3 – Représentation du réseau routier par un graphe orienté [8].

1.2.5 Les différents modèles du trafic

Le trafic routier est un processus complexe où interviennent plusieurs comportements de natures diverses. Ces comportements peuvent être continus comme c'est le cas sur les axes autoroutiers, ou discrets au niveau des carrefours et des intersections avec des panneaux ou des feux de signalisation.

Pour décrire la dynamique de ces systèmes dans l'espace et dans le temps, il s'avère nécessaire de disposer d'outils adéquats pour prédire, évaluer et expliquer les phénomènes du trafic [21]. La théorie du trafic a été développée dans le

but de comprendre le comportement qui régit l'évolution du trafic à différents niveaux de détails. Devant la multitude des modèles du trafic plusieurs études ont porté sur leur classification ainsi que sur leur champ d'application. Dans cette optique la classification peut être effectuée en tenant compte du niveau de détail lié au nombre de véhicules. Ce critère a permis de classer les modèles du trafic en :

1.2.5.1 Modèle macroscopique

Les modèles macroscopiques assimilent l'écoulement du trafic routier à un flot continu sur chaque section de route. Le trafic est décrit à l'aide de variables macroscopiques (débit, concentration, taux d'occupation,...) temporelles mesurées par des capteurs [8].

1.2.5.2 Modèle microscopique

Les modèles microscopiques du trafic reposent sur une description du comportement individuel des véhicules. Cette description fine du trafic est réalisée en se basant sur des modèles représentant les interactions entre le véhicule individuel et son environnement (autoroute, par exemple). Dans le domaine de la modélisation microscopique, deux grandes classes de modèle peuvent être identifiées. La première est celle caractérisant les modèles "poursuites", la deuxième est basée sur les modèles de particules [21] :

- **Modèle de poursuite :** Les modèles de poursuite ou modèles "véhicule-suiveur" décrivent le comportement d'un véhicule par rapport au véhicule qui le précède. Ces modèles sont utilisés pour représenter et analyser les mécanismes qui régissent les interactions entre les véhicules ainsi que le mouvement d'un ensemble de véhicules sur les infrastructures routières. Dans ces modèles, le conducteur joue un rôle primordial en terme d'élément de commande actif et prévisible au sein de l'ensemble (véhicule, conducteur, route). En effet, les modèles de poursuite supposent l'existence d'une relation entre les véhicules et cherchent à formuler cette corrélation en tenant en compte de la perception, de la prise de décision et de la commande du conducteur. En conclusion, les modèles de poursuite considèrent que pour deux véhicules se déplaçant à la même, l'espace inter-véhicules est constant.
- **Modèle de particules :** Les modèles de particules sont issus principalement de la théorie des automates cellulaires (AC). Cette dernière théorie est très intéressante comme outil de modélisation capable de représenter d'une manière fine et rapide des comportements dynamiques complexes. Les modèles d'automate cellulaire représentent la route ou l'autoroute étudiée en termes d'un réseau de cellules identiques interconnectées. Généralement, la modélisation repose sur l'emploi de variables entières telles que l'entrée, l'état et la sortie. Le comportement des véhicules fournit une image globale des situations du trafic dans le réseau.

1.2.5.3 Modèle mésoscopique

Les modèles mésoscopiques sont situés au niveau intermédiaire entre le modèle microscopique et le modèle macroscopique. En effet, au lieu de s'intéresser au véhicule individuel, l'approche mésoscopique se focalise sur des paquets de véhicules caractérisés par les mêmes propriétés (i.e. même origine et même destination). Il existe plusieurs types de modèles mésoscopiques tels que le modèle de distribution et les modèles basés sur la cinématique des gaz. Ces modèles nécessitent relativement moins d'effort de calcul que les modèles microscopiques [21].

1.2.6 La congestion

Pour comprendre la congestion, il faut garder présent à l'esprit que c'est un phénomène qui survient lorsque la demande (le nombre de véhicules qui cherchent à utiliser une infrastructure donnée) est supérieure à la capacité de cette infrastructure. Si la demande excède la capacité, alors des véhicules seront ralentis à l'entrée de l'infrastructure, formant ainsi un bouchon. Ces véhicules excédentaires seront à chaque instant plus nombreux qu'à l'instant précédent. Comme chaque véhicule occupe une certaine longueur de voie, la longueur de la file d'attente ne fera que croître en proportion du nombre de véhicules présents dans cette file d'attente [9].

Or, en raison du caractère maillé de l'infrastructure, cet allongement de la file d'attente peut la conduire à atteindre des points de choix d'itinéraire, et ainsi congestionner des parties du réseau où circulent des véhicules qui n'utiliseront pas l'infrastructure à l'origine de la congestion.

D'autres événements peuvent augmenter la congestion tels que :

- les événements affectant le réseau tels comme les pannes de feu de circulation, les travaux, les accidents, etc.
- les mesures météorologiques et environnementales (neige, verglas, pollution, etc.).

Une des solutions qui essaie de minimiser les effets de la congestion, mis à part les solutions naturelles qui consiste à augmenter le nombre des infrastructures routières, est le système d'information géographique (SIG) qui fait l'objet de notre mémoire.

1.3 Conclusion

Dans ce chapitre, nous avons défini le trafic routier, ses variables élémentaires et ses trois classes principales de modèles : la première ignore le véhicule individuel et s'intéresse au mouvement du flux considéré d'un point de vue macroscopique, la deuxième classe repose sur des considérations microscopiques et s'intéresse au véhicule individuel et décrit son comportement et ses interactions avec son environnement. Enfin la troisième classe, regroupe les modèles mésoscopiques. Ces modèles décomposent le trafic en pelletons de véhicules caractérisés par des propriétés identiques.

Chapitre 2

Les Systèmes d'Information Géographiques(SIG)

2.1 Introduction

Auparavant, les territoires et leur environnement étaient appréhendés soit par une représentation géographique traditionnelle à l'aide de plans et de cartes, soit par une approche comptable en constituant et en exploitant des bases de données alphanumériques (automatisées ou non). L'apport des Systèmes d'Information Géographique(SIG) est de permettre une vision globale de ces territoires en proposant, par la mise en œuvre d'outils qui associent chiffres et cartes, de lier les deux domaines, de les enrichir et de les synthétiser [13].

Dans ce chapitre nous allons expliquer ce qui est réellement un SIG et répondre concrètement à la question suivante : " comment construire un système d'information géographique complet et opérationnel, en suivant les principes théoriques de la gestion de données et en les adaptant aux données géographiques ?"

2.2 Historique des SIG

L'historique du développement des SIG peut être divisé en quatre périodes [24] :

- **Les années 60-70** : représentant les débuts et les premières réalisations. Cette période est caractérisée par la montée en puissance des ordinateurs qui ont permis le développement des systèmes de cartographie numérique.
- **Les années 80** : la consolidation et l'apparition des premiers logiciels commerciaux. C'est la période de consolidation des SIG. En effet, la diffusion des outils de cartographie automatique dans les organismes d'Etat (armée, cadastre, service topographique, etc.) a permis d'apporter des améliorations aux SIG surtout du côté de la représentation (vecteur, raster, etc.).
- **Les années 90** : pour la diffusion générale des outils et de la technologie SIG. C'est la période où la technologie des SIG a été industrialisée et diffusée. Cette période était définie surtout par la poussée informatique qui a contribué au développement du matériel graphique et a permis aux SIG d'intégrer des données de différentes ressources.
- **Les années 2000** : un important travail sur les documents d'urbanisme a été effectué au cours de l'année 2007. En 2010, chaque commune disposant d'un Plan Local d'Urbanisme (PLU) ou d'une carte communale peut consulter son document en format numérique superposable au Plan Cadastral Informatisé (PCI) [37].

2.3 Définition d'un SIG

Avant d'aborder la notion de système d'information géographique, on doit d'abord définir un système d'information. Un système d'information est un outil informatique permettant, à partir de diverses sources de rassembler, d'organiser, de gérer, d'analyser, de combiner, d'élaborer et de présenter des informations afin de répondre le mieux possible aux attentes des utilisateurs [10].

Plusieurs définitions des SIG existent, parmi les plus connues, on cite : un système d'information géographique est avant tout un système d'information constitué de matériels et de logiciels utilisés pour la récupération, le stockage, l'analyse et la représentation graphique de données géographiques [15].

2.4 Les avantages d'un SIG

Les avantages d'un système d'information géographique sont [13] :

- Un SIG permet de réduire les coûts de production, des cartes, et des plans. Dans de nombreuses administrations, les cartes et les plans sont établis à la main, avec des délais et des coûts de correction, de mise à jour, de dessin, etc. Le SIG permet de les établir plus rapidement et à moindre frais.
- Un SIG permet aussi d'établir des cartes et des plans que l'on ne pouvait pas réaliser à la main. Grâce à l'informatique, il est possible de réaliser des produits qu'il était impossible de réaliser à la main.
- Un SIG évite d'avoir à refaire plusieurs fois les mêmes levés. Il évite que des services différents procèdent à des levés topographiques sur la même zone et évite les pertes d'information avec le temps en accumulant l'information recueillie sur le terrain.
- Lorsque le SIG est en place, installer une nouvelle application nécessite un investissement modeste et le retour sur investissement est rapide.
- Un SIG facilite la réalisation d'étude pour tous les projets ayant une composante géographique. Il permet de multiplier les représentations visuelles et facilite ainsi la prise de décision tout en diminuant les risques d'erreurs.
- Le SIG améliore le service rendu à l'utilisateur en permettant de lui fournir avec rapidité et fiabilité une information de qualité dont il a besoin. Par exemple, tous les renseignements délivrés par le service urbanisme seront, en principe, à jour et complets.
- Le SIG permet des calculs utiles à la prise de décision et cela va du calcul simple, la superposition cartographique, au calcul complexe d'analyse spatiale intégrant un grand nombre de paramètres.

2.5 Questions auxquelles peuvent répondre les systèmes d'information géographique

Un système d'information géographiques doit répondre à cinq questions, quel que soit le domaine d'application [16] :

Où ? : recherche spatiale d'objets par rapport à leurs caractéristiques,

Quoi ? : recherche de caractéristiques d'objets par rapport à leur positionnement,

Comment ? : recherche de relations qui existent entre différents objets, création d'une nouvelle information par croisement d'informations,

Quand ? : recherche de changements intervenus sur les données,

Et si ? : définir, en fonction de certaines hypothèses, l'évolution du terrain, et l'étude d'impact.

2.6 Objectifs visés par les SIG

Avant de présenter les fonctionnalités du SIG, nous dressons un panorama général des principaux objectifs des systèmes d'information géographiques [12] :

- **Saisie et stockage numérique de plans et de cartes**

Le premier et principal objectif des SIG reste le stockage numérique de données géographiques. Mais il y a beaucoup de différences entre un système qui va conserver des objets, avec une description aussi bien graphique que descriptive, et un système qui va seulement conserver un dessin sans contenu sémantique.

- **Structuration de l'information**

Comme tout système de gestion de bases de données, un SIG qui gère une base de données demande une modélisation du monde réel et une structuration de l'information. Cette structuration est souvent plus complexe, car elle touche des objets qui peuvent avoir de multiples représentations, aussi bien graphiques que descriptives, essentiellement en fonction de l'utilisation qui en sera faite.

- **Calculs métriques (distances, surfaces), calculs techniques (visibilité, volumes), positionnement et projections géographiques**

Les SIG permettent de calculer facilement surfaces, distances et volumes à partir des données de localisation des objets. Les calculs et les changements de projections géographiques sont facilement accessibles. La recherche opérationnelle (essentiellement calculs de chemins dans des graphes) trouve dans les SIG toutes les données dont elle a besoin.

- **Gestion et traitement des collections d'objets**

C'est l'un des objectifs principaux des SIG. Une fois l'information structurée, elle doit être saisie et gérée par le système. Souvent, les SIG laissent la gestion des données descriptives à des SGBD (Système de Gestion de Base de Données) relationnels classiques (comme ACCESS, ORACLE, SQL Server, DBase, etc.), et ne gèrent eux-mêmes que la localisation des objets et les liens entre graphique et description. Comme tout système de gestion de base de données, le SIG doit assurer la bonne gestion des flux d'informations, des modifications, des mises à jour, et notamment pour la partie graphique des objets.

- **Simulation et modélisation**

L'objectif d'un SIG peut également être l'utilisation d'un modèle pour la simulation d'un processus. Le SIG doit alors faciliter l'interface entre le programme de modélisation ou de simulation et la base de données géographiques, et doit prendre en charge l'ensemble de l'accès à l'information spatiale dont a besoin le programme d'application.

- **Téledétection, géo-référencement et traitement d'image**

Les SIG ont vocation à gérer tout type d'objet géographique. La téledétection aérienne ou spatiale offre une source privilégiée de données géographiques. Les SIG doivent donc également gérer et traiter des types de données, souvent volumineuses. Ils doivent en assurer le bon géo-référencement, permettre l'accès aux traitements propres à ce type de données, et permettre leur mise en relation avec l'ensemble des autres données localisées gérées par le système.

- **Dessin et édition cartographique, cartographie automatique et 3D**

Comme tout système de gestion de données, les SIG ont pour objectif l'édition des données résultats d'une requête. Cette édition est souvent graphique puisque l'on traite des données localisées. Les modules de cartographie automatique à partir des données gérées par le système sont donc fondamentaux pour l'utilisateur. De plus en plus, les systèmes intègrent la troisième dimension, et permettent l'édition de données en perspective.

Mais la saisie et la maintenance de la troisième dimension sont plus complexe.

- **Internet et accessibilité distante**

L'Internet offre depuis plusieurs années de nouvelles perspectives d'accès distant aux données. Les SIG doivent donc permettre cet accès, en gérant la complexité de structure de l'information localisée, de manière à fournir aux utilisateurs des méthodes simples de consultation et de cartographie via Internet.

2.7 Domaines d'application d'un SIG

Les SIG sont utilisés dans différents domaines, notamment [15] :

2.7.1 La représentation géographique

- Produire des cartes thématiques pour différents thèmes et domaines.

2.7.2 Gestion géographique

- Intègre l'étude et l'aménagement d'un espace ou d'un paysage.

2.7.3 Inventaire et gestion des éléments naturels

- Contrôle de la qualité de l'eau et des travaux de gestion de l'eau : systèmes d'irrigation et de drainage.
- Gestion des forêts.
- Gestion des parcs naturels.
- Inventaire et gestion des monuments nationaux et des sites archéologiques.
- Inventaire et cartographie des sols et des conditions géologiques.

2.7.4 Environnement

- Études d'impacts et développement de phénomènes (extension du désert, etc.).
- Prévision des risques naturels et prévention des calamités.

2.7.5 Agriculture de précision

Gestion des récoltes, estimations des productions et des dosages d'engrais.

2.7.6 Géomarketing

Gestion des ventes, des stocks, et de la logistique des transports.

2.8 Classification des SIG

Les SIG peuvent être classés en trois grandes familles [24] :

2.8.1 Les SIG généralistes bureautiques

Ce sont des SIG qui importent des données externes et les analysent pour donner des cartes à insérer dans des rapports ou présentations. Ils permettent de modifier des données géométriques ou descriptives. Ils disposent d'outils de développement pour s'adapter à tout type d'application, par contre, ils ne disposent pas d'outils d'assurance qualité perfectionnés pour la saisie de base de données complète (exemple : MapInfo, ArcView).

2.8.2 Les SIG généralistes de gestion

Ils disposent d'outils de modélisation très puissants, ce qui permet une assurance sur la qualité des données et cela en intégrant des contraintes à la saisie (exemple : APIC, GéoConcept).

Ces SIG possèdent aussi des capacités client/serveur qui permettent à plusieurs personnes de travailler sur une même base de données à partir de postes informatiques distants comme ils disposent d'outils de développement pour s'adapter à tout type d'application.

2.8.3 Les SIG généralistes de métiers

Ils sont très spécialisés, destinés à des particuliers. Ces SIG ont un champ d'application réduit, ils sont souvent les seuls ou les meilleurs dans leurs domaines.

2.9 Facteurs liés aux SIG

Plusieurs facteurs contribuent au cœur de fonctionnement des SIG, tels que[14] :

2.9.1 Les couches

Un SIG stocke les informations concernant le monde sous la forme de couches thématiques pouvant être reliées les unes aux autres par leurs coordonnées géographiques. On peut faire une analogie avec le travail "manuel" du cartographe qui superpose ses différents calques (calque des routes, calque des villes, etc.) afin de représenter l'information géographique. Chaque couche va donc contenir un ensemble unique de données (couches des bâtiments, couches des rues, couches des forêts, etc.) et c'est en rassemblant ces différentes couches que l'on va obtenir la carte finale comportant toutes les données géographiques voulues comme le montre l'image ci-dessous.

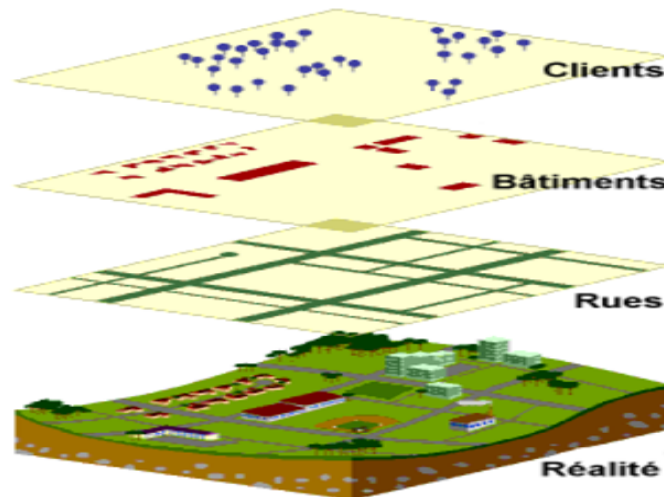


FIGURE 2.1 – Les différentes couches des SIG [14].

2.9.2 Les références géographiques

Lorsqu'on veut étudier une carte géographique, deux types de données sont à prendre en compte :

- ✓ **Les données attributaires** : description qualitative des objets géographiques (entité classique)
- ✓ **Les données spatiales** : qui décrivent l'emplacement absolu et relatif des objets géographiques, ainsi que leur étendue. Les données spatiales peuvent être représentées par les trois types d'objets graphiques suivant (voir la figure 2.2) :
 - Le point (ville, bâtiments, etc.).
 - Les données linéaires (routes, fleuves, réseaux, etc.).
 - Les polygones (wilaya, commune, surface d'un terrain, etc.).

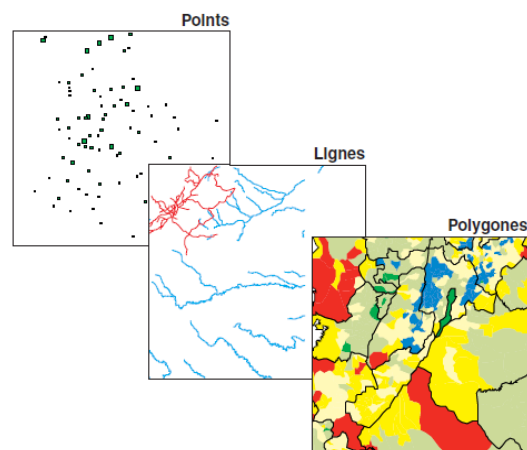


FIGURE 2.2 – Représentation des données spatiales par les trois types d'objets graphiques [17]

Parmi les données spatiales, géographiques, on distingue les références géographiques explicites (latitude et longitude ou grille de coordonnées nationales) et les références géographiques implicites (adresse, code postal, nom de route, etc.).

Les " données géographiques de référence " d'un SIG sont celles qui permettent de localiser les informations, non pas directement par leurs coordonnées géographiques, mais indirectement par une adresse, un numéro de parcelle ou d'ilot, dont les coordonnées sont déjà connues. Il est ainsi possible de représenter des données statistiques ou des données de gestion sur des cartes et d'analyser leur répartition spatiale.

L'information géographique peut se présenter par [29] :

- une image enregistrée de la surface terrestre où l'on peut voir une multitude d'objets mais sans connaître directement leurs attributs (on ne voit pas le nom de la route).



FIGURE 2.3 – Exemple d'une image enregistrée.

- un texte ou un fichier de données littérales où l'information géographique est représentée par des données numériques et par une adresse postale (exemple : fichier des abonnés au téléphone : nom, prénom, numéro de téléphone, adresse postale).

Nom commune	Pommeret
Nombre habitants	1 857
Nb propriétés bâties	456
Nb propriétés non bât	324
Nb propriétaires	566
Montant taxe foncière	2 324 554
Surface (hectares)	1 368,6
Surface agricole utile	845,7
Nombre exploitations	16
Surface forêts	28,4
Emprise routes	3,6
Emprise SNCF	0,9
Existence d'un POS	oui

FIGURE 2.4 – Exemple d'un texte.

- une carte : l'information géographique se prête particulièrement bien à la représentation sur une carte, où l'on situe les objets et les phénomènes dans un repère général et homogène et où l'on a une vue d'ensemble sur leur implantation sur le terrain.



FIGURE 2.5 – Exemple d'une carte.

Ces trois formes de représentation sont distinctes mais complémentaires :

- L'image comporte surtout des données géométriques (forme, dimensions, localisation).
- Le texte ou le fichier littéral comporte surtout des données sémantiques (attributs).
- La carte comporte des données à la fois sémantiques et géométriques.

Notons que les données sémantiques de la carte s'expriment principalement par des symboles (points, lignes, surfaces), dont les attributs sont expliqués par la légende de la carte).

La carte apparaît ainsi comme une forme intermédiaire (et optimale) de représentation de l'information géographique, avec un dosage particulier entre données sémantiques (on identifie moins d'attributs que dans un fichier) et géométriques (on voit moins d'objets que sur une image mais ils sont tous identifiés) [29].

2.9.3 Les modèles géographiques utilisés par un SIG

Les SIG exploitent deux différents types de modèles géographiques. Ces deux modèles sont utilisés pour numériser et stocker les données géographiques dans la base de données, le mode " raster " et le mode " vecteur " (voir la figure 2.3) [14] :

2.9.3.1 Le mode raster (appelé aussi mode maillé)

Les " couvertures " du type raster sont exclusivement constituées de fichiers numériques contenant une collection de pixels. Ces pixels ou cellules traduisent une vision continue des phénomènes sans délimiter d'objets géographiques à proprement parler. On les connaît mieux sous leur nom de format : bitmap, jpg, tif, etc. En effet, la carte géographique à traiter est découpée sous la forme d'une matrice de cellules (appelées pixels). L'espace géographique est alors décrit point par point, chaque cellule ou pixel va contenir une valeur représentant une portion de l'objet géographique sous-jacent, cela peut être la couleur du terrain, la température, l'altitude, etc.

2.9.3.2 Le mode vecteur

Les bases numériques vecteurs sont des outils de travail précieux et adaptables à de nombreuses applications. Elles sont le plus souvent issues de la digitalisation et du traitement de cartes papier ou encore directement créées à partir de photos aériennes ou d'images satellitaires. Elles se composent d'objets géométriques de type ponctuel, linéaire, surfacique et/ou textuel auxquels des informations peuvent être rattachées. Dans le modèle vecteur, les informations sont regroupées sous la forme de coordonnées (x, y) . Les objets ponctuels sont, dans ce cas, représentés par un simple point. Les objets linéaires (routes, fleuves, etc.) sont eux représentés par une succession de coordonnées (x,y) qui vont former des lignes. Les objets surfaciques, polygonaux (territoire géographique, parcelle, etc.) sont, quant à eux, représentés par une succession de coordonnées délimitant une surface fermée.

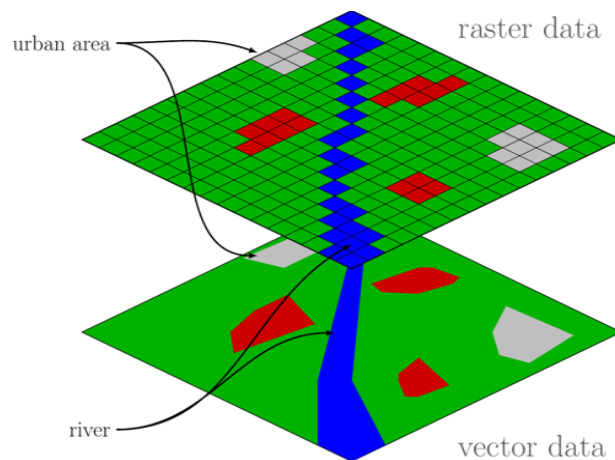


FIGURE 2.6 – Représentation des deux modes raster et vecteur[28]

2.10 Fonctionnalités des SIG

Les fonctionnalités techniques d'un SIG sont communément synthétisées selon le modèle des 5 A [25] :

2.10.1 Abstraction

Abstraire revient à dissocier les thèmes sous formes de couches. Chacun des thèmes peut être traité à part entière puisqu'il regroupe des objets géographiques homogènes.

Exemples : Dissocier les infrastructures réseau en couches : routes, eau, électricité, etc.

2.10.2 Acquisition

Acquérir revient à alimenter le SIG en données. Les fonctions d'acquisition consistent à entrer d'une part, la forme géométrique des objets et d'autre part, leurs attributs et relations.

Exemples : Alimentation de la base de données géographique par des données décrivant le réseau routier, les tronçons de route, etc.

2.10.3 Archivage

Archiver consiste à sauvegarder les données manipulées d'une manière organisée, au sein d'une base de données.

Exemples : Archiver un nombre d'habitants d'une commune à une année ultérieure.

2.10.4 Analyse

Permet de répondre aux questions que l'on se pose. Le SIG devient alors un tableau de bord cartographique.

Exemples : quelle est la route la plus fluide ?

2.10.5 Affichage

Afficher pour percevoir les relations spatiales entre objets et pour visualiser les données tout en faisant lien avec leur localisation spatiale.

Exemples : proposition d'une réponse à la question par l'affichage de la route.

2.11 Composantes principales d'un SIG

Un SIG est un outil particulier puisque pour assurer une opération correcte, il fait appel à de nombreux concepts techniques et pratiques et nécessite des connaissances dans plusieurs domaines.

Lorsqu'on parle de SIG, cinq aspects (cinq éléments clefs) doivent être pris en considération [15] :

2.11.1 Matériels

Un SIG est basé sur un ordinateur pour le stockage et le traitement des données. La taille du système informatique va dépendre du type et de la nature du SIG. Un SIG à échelle réduite, aura seulement besoin d'un petit ordinateur personnel pour fonctionner, alors qu'un grand projet nécessite un système plus conséquent avec des ordinateurs plus puissants et une foule de terminaux pour supporter de multiples utilisateurs.

En plus des ordinateurs, on peut utiliser une grande variété d'autres dispositifs pour capturer et introduire les données dans le système. Les scanners et les tables à digitaliser sont utilisés pour reproduire les cartes graphiques et schémas support-papier dans le système. Des récepteurs GPS sont utilisés pour cartographier des éléments sur le terrain et transmettre la position courante de véhicules en déplacement .

2.11.2 Logiciels

Au cœur de tout SIG se trouve le logiciel SIG lui-même qui procure les fonctionnalités pour stocker, gérer, lier, interroger et analyser les informations géographiques. En plus du noyau de logiciel SIG, d'autres composants logiciels variés peuvent être ajoutés pour permettre l'accès à des sources additionnelles de données et d'autres types de fonctions. Ces composants peuvent être :

- un système de gestion de base de données,
- des outils pour l'entrée et la manipulation de l'information géographique,

- des outils qui permettent l'interrogation, l'analyse et la visualisation géographiques,
- une interface utilisateur graphique pour un accès simple aux outils.

2.11.3 Données

Le concept de données de SIG présente deux aspects : les données objets et les données informations.

Toutes les données ont ou représentent, en fonction des cas, des attributs ou des caractéristiques et peuvent ainsi être référencées. Mais la majorité de l'information géographique (données objet) peut être aussi géo-référencée (c'est à dire définie dans un espace connu) soit par leur localisation ou par leur prolongation ou leur trajectoire, lorsque c'est le cas d'objets en mouvement.

Le géo-référencement donne aux objets leur possibilité d'être représentés dans un espace géographique et c'est la principale caractéristique des SIG.

2.11.4 Utilisateurs

La technologie des SIG est d'un intérêt limité sans les personnes qui contrôlent le système et qui développent des projets pour les utiliser. La population des utilisateurs de SIG s'étend des techniciens spécialisés qui conçoivent et améliorent le système à ceux qui utilisent les SIG comme outil dans leurs travaux quotidiens. L'objectif principal des SIG est de soutenir ses utilisateurs avec les données et les outils d'aide à la décision appropriés. Ainsi, on doit soigneusement prendre en compte les besoins particuliers de ces utilisateurs qui doivent être connus dès le stade de la conception du système afin que chaque groupe d'utilisateur bénéficie d'un accès adapté aux données et fonctions du système.

2.11.5 Méthodes

Les SIG sont conçus et développés pour faciliter les gestions de données et les processus d'aide à la décision d'une organisation. L'activité de toute organisation est basée sur un ensemble de pratiques et de logiques de fonctionnement spécifiques à chaque organisation, tel que des banques traitant des applications de prêt, des détaillants contrôlant la livraison des marchandises et des approvisionnements, services de réparation et d'entretien, organismes gouvernementaux contrôlant les autorisations et la régularité des processus de conformité, et beaucoup d'autres .



FIGURE 2.7 – Les composantes d'un SIG [18]

2.12 SIG pour la régulation du trafic routier

Notre solution consiste à concevoir un système d'information géographique mobile :

- géographique car il possède des données spatiales pour apporter un affichage cartographique du terrain (aide à la prise de décision) et offrir une pré-vision de l'état d'un réseau routier basée sur des constats et estimations (généralement, le plus court chemin est toujours le plus encombré),

- sa mobilité permet de répondre à une enquête à n'importe quel endroit, être à la portée de tout le monde et contribuer à l'évolution de l'informatique dans le domaine mobile.

En effet, il va fournir des informations aux utilisateurs des infrastructures routières sur le degré de congestion afin d'éviter son augmentation. Ceci nécessite beaucoup de précision et d'attention dans la collecte d'informations, sa modélisation, sa conception et sa présentation.

2.13 Conclusion

Nous avons appuyé dans ce chapitre sur la présentation des concepts de base relatifs aux systèmes d'information géographiques tels que ses outils et son fonctionnement et de souligner ses composantes principales, nous avons montré aussi comment l'information géographique est représentée dans les SIG et nous avons justifié l'apport des SIG dans le domaine de régulation du trafic routier qui est un problème de gestion.

Le prochain chapitre sera consacré à la présentation et la modélisation du notre système géographique mobile pour la régulation du trafic routier.

Chapitre 3

Analyse et conception

3.1 Introduction

La motivation fondamentale de la modélisation est de fournir une démarche antérieure afin de réduire la complexité du système étudié lors de la conception et d'organiser la réalisation du projet en définissant les modules et les étapes de la réalisation.

Nous adoptons dans ce mémoire, pour modéliser notre système nommé TrafRout (pour **T**rafic **R**outier), une approche objet basée sur le langage de modélisation UML.

Dans ce chapitre, nous allons présenter un aperçu sur UML en citant ses différents diagrammes. Notre système sera modélisé en suivant la démarche UP (unified process)

3.2 Présentation du langage UML

3.2.1 Définition

UML (Unified Modeling Language)[1,2] est le résultat d'une opération d'unification d'un ensemble de concepts pris à partir des méthodes orientées objets, OMT (Object Modeling Technique), Booch (Nom tiré de son inventeur) et OOSE (Object Oriented Software Engineering) dans le but de modéliser d'une manière claire et précise la structure et le comportement d'un système, indépendamment de toute méthode et tout langage de programmation.

UML se définit alors comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

3.2.2 Historique

UML est une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (Object Management Group), instance de normalisation internationale du domaine de l'objet. En quelques années, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet. A la fin de la première décennie des années 2000, nous avons déjà une dizaine d'années de recul sur l'enseignement et la pratique d'UML en entreprise [4].

L'évolution d'UML peut se résumer comme suit :

1994-1996 : rapprochement des méthodes OMT, BOOCH et OOSE et naissance de la première version d'UML.

23 novembre 1997 : version 1.1 d'UML adoptée par l'OMG.

1998-1999 : sortie des versions 1.2 à 1.3 d'UML.

2000-2001 : sortie des versions suivantes 1.x.

2002-2003 : préparation de la version 2.

10 octobre 2004 : sortie de la version 2.1.

5 février 2007 : sortie de la version 2.1.1.

2009 : sortie de la version 2.2.

3.2.3 Intérêt d'UML

Un des premiers avantages d'UML est de faire se rencontrer et communiquer les utilisateurs et les informaticiens, ce qui n'est pas le moindre des bénéfices du langage. UML permet également de documenter très clairement les besoins exprimés par ces derniers, dans le cadre d'une gestion de projet de développement qui va de l'analyse et la conception jusqu'au déploiement de l'application [1].

3.2.4 Présentation générale des diagrammes d'UML 2

UML 2 s'articule autour de treize diagrammes différents, dont quatre nouveaux diagrammes introduits par UML 2. Chacun d'eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure du système pris " au repos ", l'autre concerne sa dynamique de fonctionnement. Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles [16] :

3.2.4.1 Les diagrammes structurels

Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système (classes, objets, composants, etc.).

- ◇ **Diagramme de classes** : ce diagramme représente la description statique en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements.
- ◇ **Diagramme d'objets** : le diagramme d'objet permet la représentation d'instances des classes et des liens entre instances.
- ◇ **Diagramme de composant** : ce diagramme représente les différents composants constituant du logiciel au niveau de l'implémentation d'un système.
- ◇ **Diagramme de déploiement** : ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.
- ◇ **Diagramme de paquetage** : ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants, etc.).
- ◇ **Diagramme de structure composite** : ce diagramme permet de décrire la structure interne d'un ensemble complexe composé par exemple de classes ou d'objets et de composants techniques. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent.

3.2.4.2 Les diagrammes de comportement

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML :

- ◇ **Diagramme des cas d'utilisation** : ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.

- ◇ **Diagramme d'état-transition** : ce diagramme montre les différents états des objets en réaction aux évènements.
- ◇ **Diagramme d'activités** : ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données.
- ◇ **Diagramme de séquence** : ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.
- ◇ **Diagramme de communication** : ce diagramme est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.
- ◇ **Diagramme global d'interaction** : ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
- ◇ **Diagramme de temps** : ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

3.2.5 Processus associés à UML

UML n'est qu'un langage de modélisation. Nous n'avons pas aujourd'hui dans la norme, de démarche unifiée pour construire les modèles et conduire un projet mettant en œuvre UML. C'est pourquoi des processus de développement ont été définis et ont lui a été associés.

Un processus de développement est un ensemble d'étapes partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant [3]. L'objectif est de produire des logiciels :

- ✓ de qualité (qui répondent aux besoins de leurs utilisateurs) ;
- ✓ dans des temps et des coûts prévisibles.

Et à chaque étape, on produit :

- ✓ des modèles ;
- ✓ de la documentation ou du code.

Il en existe plusieurs processus pour UML, on peut citer UP (*UnifiedProcess*), XP (*eXtremeProgramming*) et RUP (*RationalUnifiedProcess*) qui n'est d'autre qu'une dérivée d'UP.

Dans notre travail, nous allons opter pour le processus UP.

3.2.6 Le processus unifié (UP)

Le Processus Unifié (UP pour Unified Process) est un processus de développement logiciel " itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques " [1] :

- **Itératif et incrémental** : le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global.
- **Centré sur l'architecture** : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées.
- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

- **Conduit par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

3.2.6.1 Les phases d'UP

La gestion d'un tel processus est organisée suivant les quatre phases suivantes [4] :

3.2.6.1.1 Inception (Lancement) : cette phase correspond à l'initialisation du projet où l'on mène une étude d'opportunité et de faisabilité du système à construire. Une évaluation des risques est aussi réalisée dès cette phase.

En outre, une identification des principaux cas d'utilisation accompagnée d'une description générale est modélisée dans un diagramme de cas d'utilisation afin de définir le périmètre du projet. Il est possible, à ce stade, de faire réaliser des maquettes sur un sous-ensemble des cas d'utilisation identifiés.

Ce n'est qu'à l'issue de cette première phase que l'on peut considérer le projet véritablement lancé.

3.2.6.1.2 Élaboration : cette phase reprend les résultats de la phase d'inception et élargit l'appréciation de la faisabilité sur la quasi-totalité des cas d'utilisation. Ces cas d'utilisation se retrouvent dans le diagramme des cas d'utilisation qui est ainsi complété.

Cette phase a aussi pour but d'analyser le domaine technique du système à développer afin d'aboutir à une architecture stable. Ainsi, toutes les exigences non recensées dans les cas d'utilisation, comme par exemple les exigences de performances du système, seront prises en compte dans la conception et l'élaboration de l'architecture.

L'évaluation des risques et l'étude de la rentabilité du projet sont aussi précisées. Un planning est réalisé pour les phases suivantes du projet en indiquant le nombre d'itérations à réaliser pour les phases de construction.

3.2.6.1.3 Construction : cette phase correspond à la production d'une première version du produit. Elle est donc fortement centrée sur les activités de conception, d'implémentation et de test.

En effet, les composants et fonctionnalités non implémentés dans la phase précédente le sont ici.

Au cours de cette phase, la gestion et le contrôle des ressources ainsi que l'optimisation des coûts représentent les activités essentielles pour aboutir à la réalisation du produit. En parallèle est rédigé le manuel utilisateur de l'application.

3.2.6.1.4 Transition : après les opérations de test menées dans la phase précédente, il s'agit dans cette phase de livrer le produit pour une exploitation réelle. C'est ainsi que toutes les actions liées au déploiement sont traitées dans cette phase.

3.2.7 Modélisation du notre système TrafRout

On associe à chaque phase du processus UP une description associée à notre système qui montrera les diagrammes dont jouit UML qui pourront être utilisés dedans. Malgré que le processus UP est itératif dans ses stades de modélisation mais nous les présenterons d'une manière séquentielle afin d'éviter la reprise de ses stades.

3.2.7.1 Première phase (Inception)

Dans cette phase, nous effectuerons une étude préliminaire des principaux acteurs du système et définirons le rôle de chacun.

1. Les acteurs

Un acteur est un rôle joué par l'utilisateur du système logiciel. Les acteurs peuvent être des personnes physiques comme des systèmes automatisés. Ils se trouvent obligatoirement à l'extérieur du système [7].

Il existe deux acteurs principaux qui interagissent directement avec le système de notre application et un acteur secondaire. Nous appelons acteur principal celui pour qui le cas d'utilisation produit un résultat observable. Par opposition, nous qualifions d'acteurs secondaires les autres participants du cas d'utilisation. Les acteurs secondaires sont souvent sollicités pour des informations complémentaires [6] :

- ◊ **Utilisateur (user)** : il s'agit de toute personne qui bénéficie des fonctionnalités du SIG.
- ◊ **Administrateur (admin)** : comme son nom l'indique, il a pour rôle de maintenir le système et encore jouer le rôle d'un utilisateur.
- ◊ **GoogleMaps** : notre système fait appel à GoogleMaps qui lui retourne la carte géographisue d'une commune donnée.

2. Diagramme de contexte

Dans la figure ci-dessous, nous illustrerons les différents acteurs qui interagissent avec le système TrafRout que nous allons mettre en place :

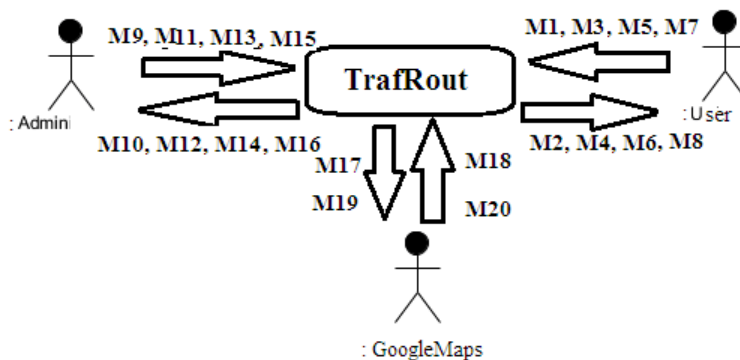


FIGURE 3.1 – Le diagramme de contexte.

Messages envoyés user vers le système	Messages réponses système vers user.
M1 : S'authentifier.	M2 : Interface d'authentification.
M3 : Trouver le route optimal.	M4 : Affichage du chemin sur la carte géographique.
M5 : Visualiser les autres chemins.	M6 : Affichage du chemin sur la carte géographique ou plusieurs.
M7 : S'inscrire.	M8 : Interface d'accueil.

TABLE 3.1 – Tableau des messages échangés par l'utilisateur et le système.

Messages envoyés user vers le système	Messages réponses système vers user.
M9 : S'authentifier.	M10 : Interface d'authentification.
M11 : Mettre à jour les lieux.	M12 : Interface de mise à jour des lieux.
M13 : Mettre à jour les routes.	M14 : Interface de mise à jour des routes.
M15 : Mettre à jour les communes.	M16 : Interface de mise à jour des communes.

TABLE 3.2 – Tableau des messages échangés par l'administrateur et le système.

Messages envoyés le système vers GoogleMaps	Messages réponses GoogleMaps vers système.
M17 : Demande d'une carte géographique.	M18 : Interface de la carte.
M19 : Tracer le chemin.	M20 : Affichage du chemin sur la carte.

TABLE 3.3 – Tableau des messages échangés par GoogleMaps et le système.

3.2.7.1.1 Présentation des besoins fonctionnels : dans ce qui suit nous décrivons les besoins fonctionnels en utilisant le diagramme de cas d'utilisation.

1. Diagramme des cas d'utilisation

Un cas d'utilisation spécifie une fonction offerte par l'application à son environnement. Un cas d'utilisation est spécifié uniquement par un intitulé. La façon dont sera réalisé concrètement un cas d'utilisation n'apparaît pas dans la définition du cas d'utilisation [11].

(a) Relations entre cas d'utilisation

- " Relation d'inclusion " include "

Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B [13].



FIGURE 3.2 – Relation d'inclusion.

- ” Relation d’extension ” extend ”

Une relation d’extension d’un cas d’utilisation A par un cas d’utilisation B signifie qu’une instance de A peut être étendue par le comportement décrit dans B [13].

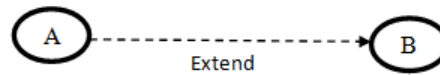


FIGURE 3.3 – Relation d’extension.

(b) **Les cas de notre système**

Dans notre projet, nous avons recensé les cas d’utilisation suivants :

- ◇ *S’authentifier*

Pour que l’utilisateur ou l’administrateur accède à l’application, il doit d’abord passer par l’authentification, cette dernière nécessite l’introduction du login et mot de passe associé.

- ◇ *S’inscrire*

Si un utilisateur ne possède pas un compte utilisateur, il doit s’inscrire pour qu’il puisse utiliser l’application.

- ◇ *Trouver le chemin optimal*

Le lancement d’une requête de recherche d’une route optimale (la plus fluide) nécessite la précision des paramètres suivants : lieu de départ, lieu d’arrivée et la commune dont ils se trouvent.

- ◇ *Visualiser les autres chemins*

Le système propose d’autres chemins à apercevoir.

- ◇ *Mettre à jour les lieux*

Ce cas est réservé à l’administrateur pour modifier, ajouter et supprimer les lieux et cela ne peut se faire qu’après s’être identifié.

- ◇ *Mettre à jour les routes*

Ce cas est réservé à l’administrateur pour modifier, ajouter et supprimer les routes et cela ne peut se faire qu’après s’être identifié.

- ◇ *Mettre à jour les communes*

Ce cas est réservé à l’administrateur pour modifier, ajouter et supprimer les communes et cela ne peut se faire qu’après s’être identifié.

- ◇ *Quitter*

Pour fermer et quitter l’application.

(c) **Diagramme de notre système**

Dans ce qui suit, nous allons présenter le diagramme des cas d’utilisation du notre système :

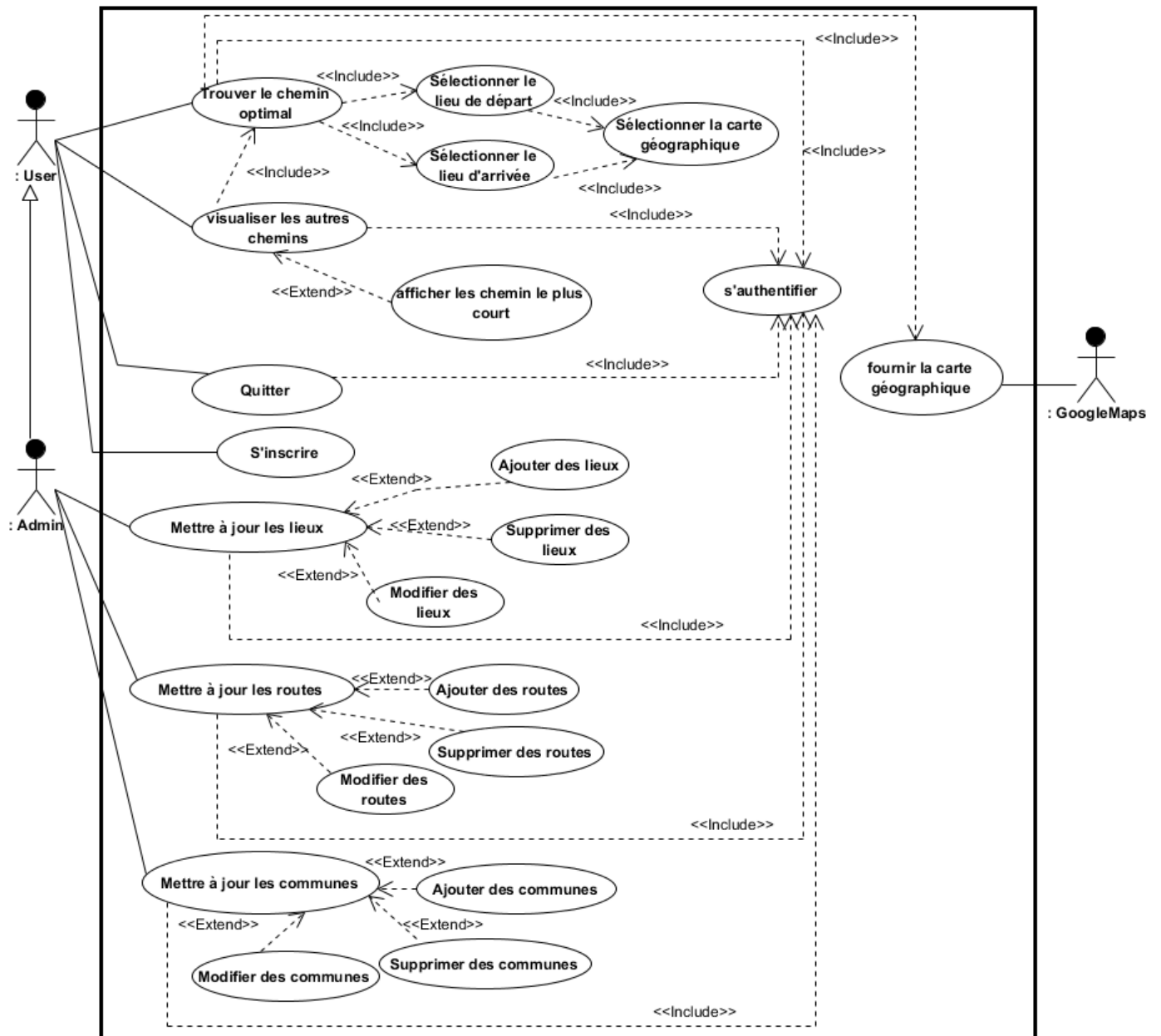


FIGURE 3.4 – Diagramme de cas d'utilisation.

✓ Tableau descriptif du cas d'utilisation "s'authentifier"

<i>Description</i>	
Titre :	S'authentifier.
But :	Authentification et autorisation d'accès.
Acteur :	Utilisateur ou administrateur .
Résumé :	L'utilisateur ou l'administrateur introduit son login et mot de passe pour accéder au système.
Pré conditions :	L'utilisateur/administrateur doit accéder au système.
Scénario nominal :	<ol style="list-style-type: none"> 1. L'utilisateur ou l'administrateur demande l'accès au système, 2. Le système affiche le formulaire d'authentification, 3. L'utilisateur/administrateur saisit son login et mot de passe, 4. Le système vérifie l'existence de l'utilisateur ou l'administrateur, 5. Si l'utilisateur est identifié, le système affiche l'interface d'accueil de l'application.
Post condition :	L'utilisateur accède au système.
Enchaînement alternatif :	<p>L'enchaînement démarre à partir du point 4 du scénario nominal.</p> <p>E1 : Mot de passe non valide ou vide</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur après la première tentative, 2. Le scénario reprend de 2. <p>E2 : Mot de passe et login valides</p> <ol style="list-style-type: none"> 3. afficher la page d'accueil.

TABLE 3.4 – Description du cas d'utilisation “s'authentifier”.

✓ Tableau descriptif du cas d'utilisation ”Trouver le chemin optimal”

<i>Description</i>	
Titre :	Trouver le chemin optimal.
But :	Afficher le chemin sur la carte.
Acteur :	Utilisateur.
Résumé :	L'utilisateur choisit le lieu de départ, le lieu d'arrivée et la commune. Le système affiche le chemin optimal et autres chemins disponibles qui correspondent à sa requête s'il y a lieu.
Pré conditions :	l'utilisateur s'est authentifié.
Scénario nominal :	<ol style="list-style-type: none"> 1. L'utilisateur s'authentifie, 2. L'utilisateur sélectionne la commune, le lieu de départ et d'arrivée. 3. Le système envoie une requête vers GoogleMaps, 4. GoogleMaps répond en affichant la carte géographique, 5. Le système trace sur la carte géographique le chemin le plus fluide, 6. Le système affiche la liste des autres chemins disponibles, 7. L'utilisateur sélectionne un chemin. 8. Le système affiche chemin sélectionné.
Post condition :	l'utilisateur voit le chemin le plus fluide et d'autres chemins possibles à emprunter.
Enchaînement alternatif :	<p>E1 : Connexion à GoogleMaps a échouée</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur " erreur de connexion ", 2. Le scénario reprend de 3. <p>E2 : Connexion à GoogleMaps a réussie</p> <ol style="list-style-type: none"> 3. Le système affiche le chemin sur la carte ",

TABLE 3.5 – Description du cas d'utilisation "Trouver le chemin optimal".

La description des cas d'utilisation se fait par l'identification des scénarios qui peuvent être représentés par un diagramme de séquence pour chacun d'eux.

3.2.7.1.2 Présentation des besoins non fonctionnels

- ◇ Existence d'un réseau sans fil.
- ◇ Un terminal mobile fonctionnant avec le système d'exploitation Android.

3.2.7.2 Deuxième phase (Élaboration)

Cette phase a pour objectif de décrire, de manière précise et compréhensible, les besoins issus des cas d'utilisations et les structures sous une forme qui facilite la compréhension (scénarios).

3.2.7.2.1 Les diagrammes de séquences : ces diagrammes permettent de modéliser la logique des flux à l'intérieur du système. Ils montrent les interactions entre objets selon un axe temporel pour chaque cas d'utilisation [19].

1. Les fragments d'interaction

Un fragment d'interaction se représente globalement comme un diagramme de séquence dans un rectangle avec indication dans le coin à gauche du nom du fragment [16]. Treize opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref.

Nous décrivons dans ce qui suit les opérateurs que nous allons utiliser pour l'analyse de notre projet : alt, loop, opt et ref.

• Opérateur alt

L'opérateur **alt** correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon. L'opérateur alt se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

• Opérateur loop

L'opérateur **loop** correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite. Il est possible aussi d'utiliser une condition portant sur un nombre minimum et maximum d'exécution de la boucle en écrivant : loop min, max. Dans ce cas, la boucle s'exécutera au minimum min fois et au maximum max fois. Il est aussi possible de combiner l'option min/max avec la condition associée à la boucle.

• Opérateur opt

L'opérateur **opt** (optional) correspond à une instruction de test sans alternative (sinon).

L'opérateur **opt** se représente dans un fragment possédant une seule partie.

• Opérateur ref

L'opérateur **ref** permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

Dans ce qui suit, nous représentons les diagrammes de séquences pour quelques cas :

1. Diagramme de séquence du cas d'utilisation " s'authentifier "

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un utilisateur ou un administrateur (généralement un login et un mot de passe). Ces informations sont préétablies dans une base de données.

Lors de l'authentification d'un utilisateur ou d'un administrateur, deux cas peuvent se présenter : informations correctes ou informations incorrectes, ce qui explique l'utilisation de l'opérateur " alt ". Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur ou l'administrateur saisit des informations incorrectes ou il laisse un champ vide, le système génère un message d'erreur et réaffiche la page d'authentification.

Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, c'est pourquoi nous avons utilisé l'opérateur " loop ".

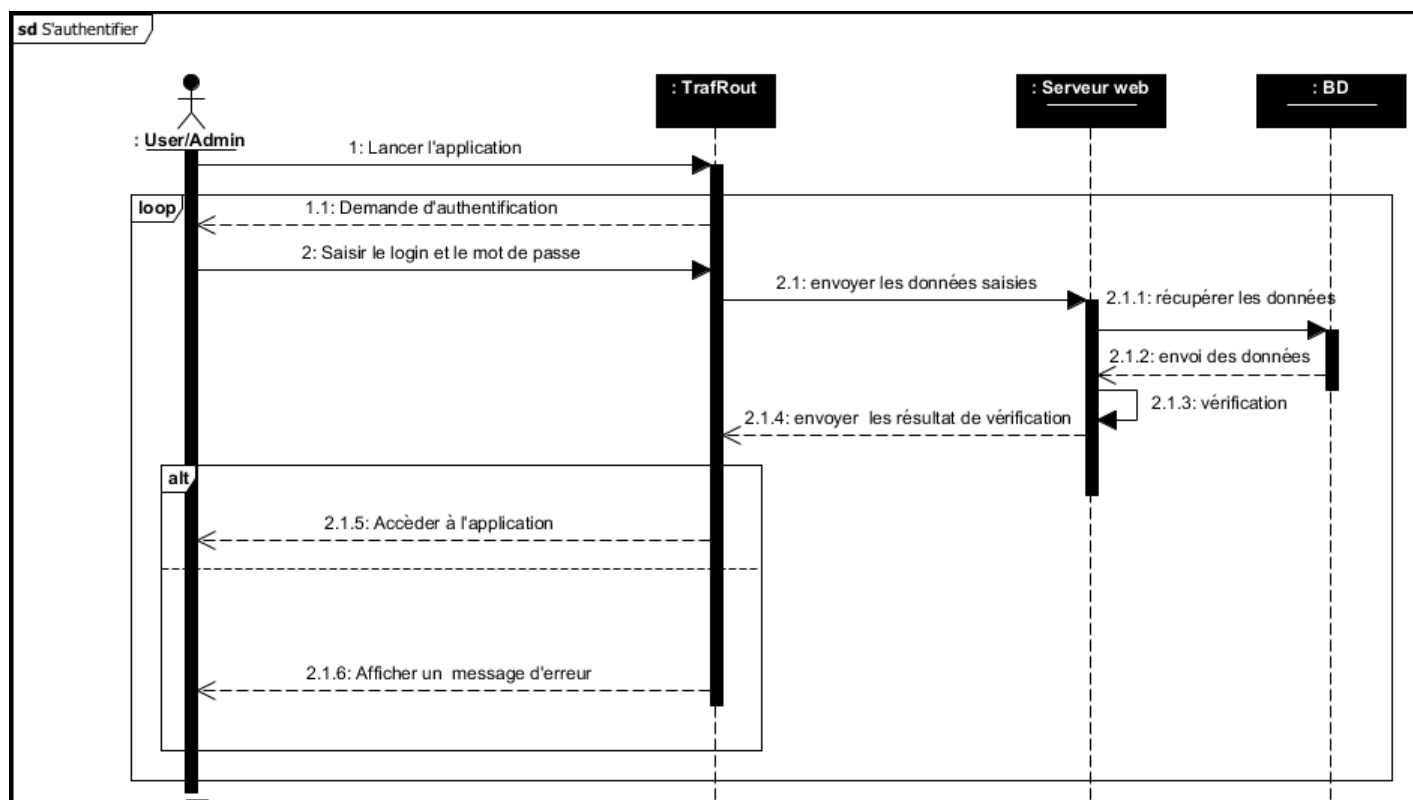


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation ” s'authentifier ”.

L'identification du fragment ” S'authentifier ” permet d'alléger les diagrammes de séquence présentés ci-dessous en utilisant le cadre ” ref ”

2. Diagramme de séquence du cas d'utilisation ” Trouver le chemin optimal ”

La recherche d'un chemin consiste à sélectionner ses lieux de départ et d'arrivée et la commune. Le système fait appel à GoogleMaps qui lui fournit la carte géographique sur laquelle notre système trace le chemin optimal. Le système offre aussi la possibilité de visualiser d'autres chemins s'ils existent.

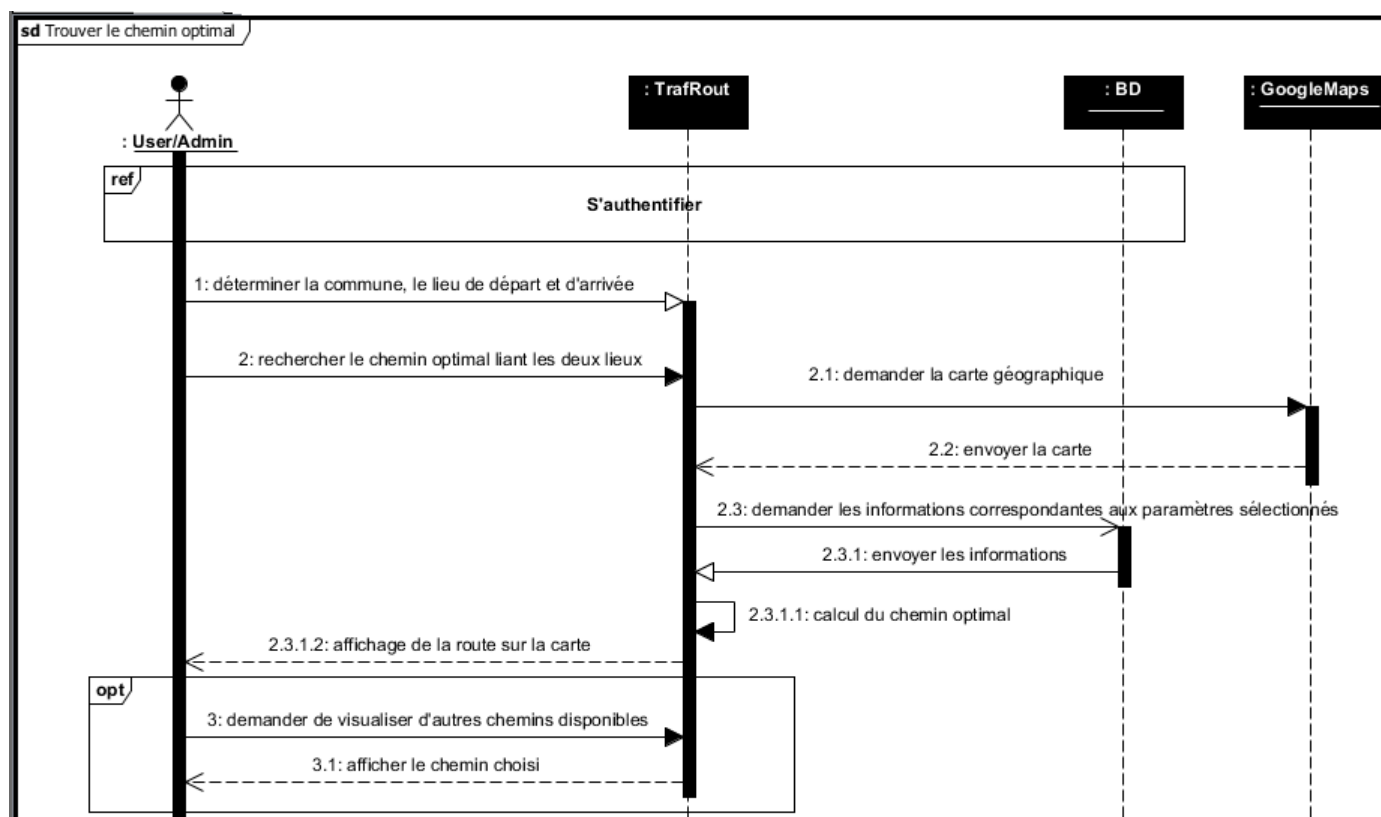


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation " Trouver le chemin optimal ".

3. Diagramme de séquence du cas d'utilisation " Ajout d'un lieu "

Après authentification, l'administrateur effectue une demande d'ajout d'un lieu, le système affiche le formulaire correspondant, l'administrateur introduit les informations qui définissent le lieu à ajouter (nom, latitude, longitude) et valide l'opération.

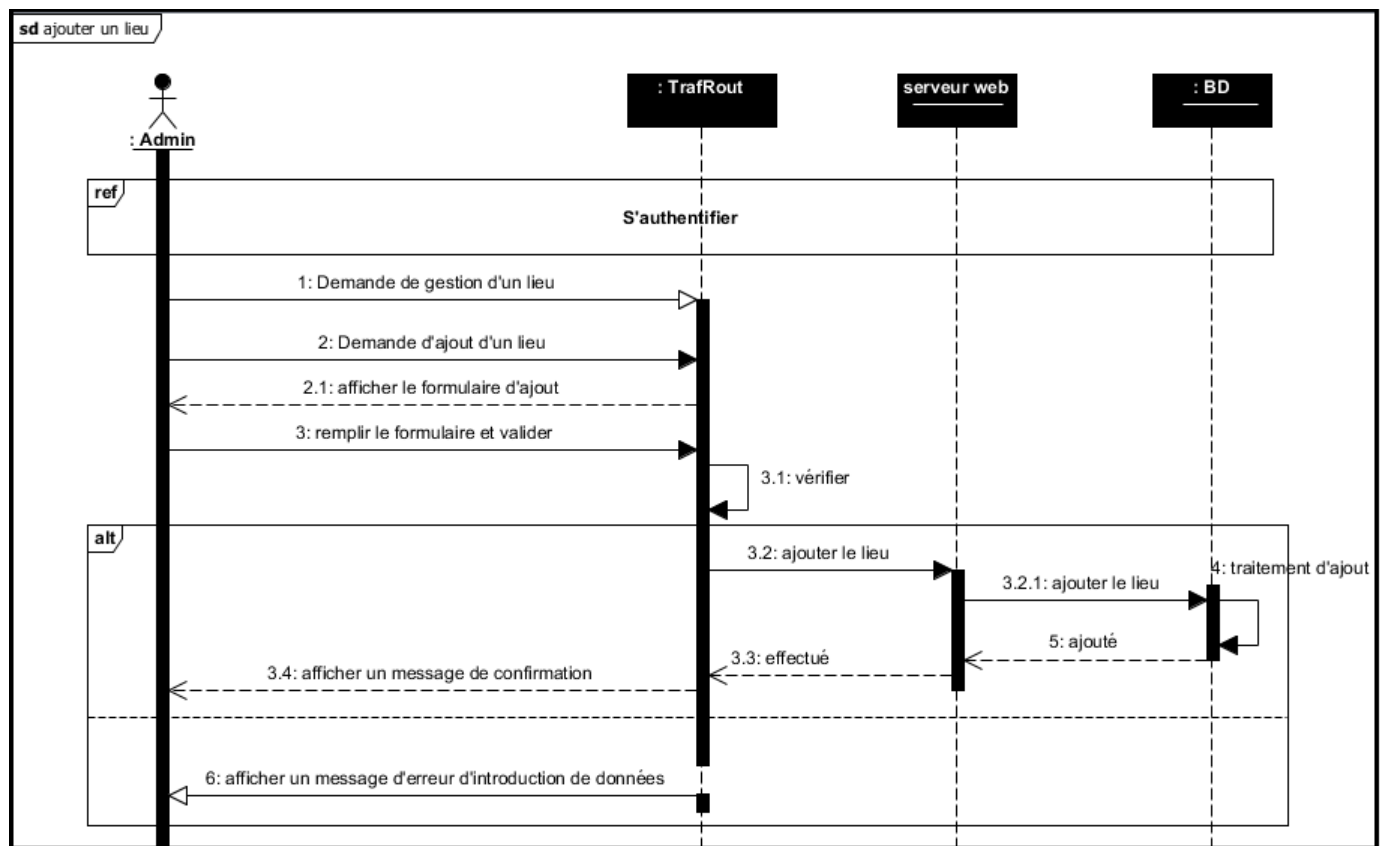


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation ” Ajout d’un lieu ”.

4. Diagramme de séquence du cas d'utilisation ” Suppression d’un lieu ”

Après la procédure d’authentification et de demande de suppression d’un lieu donné, le système supprime le lieu de la liste.

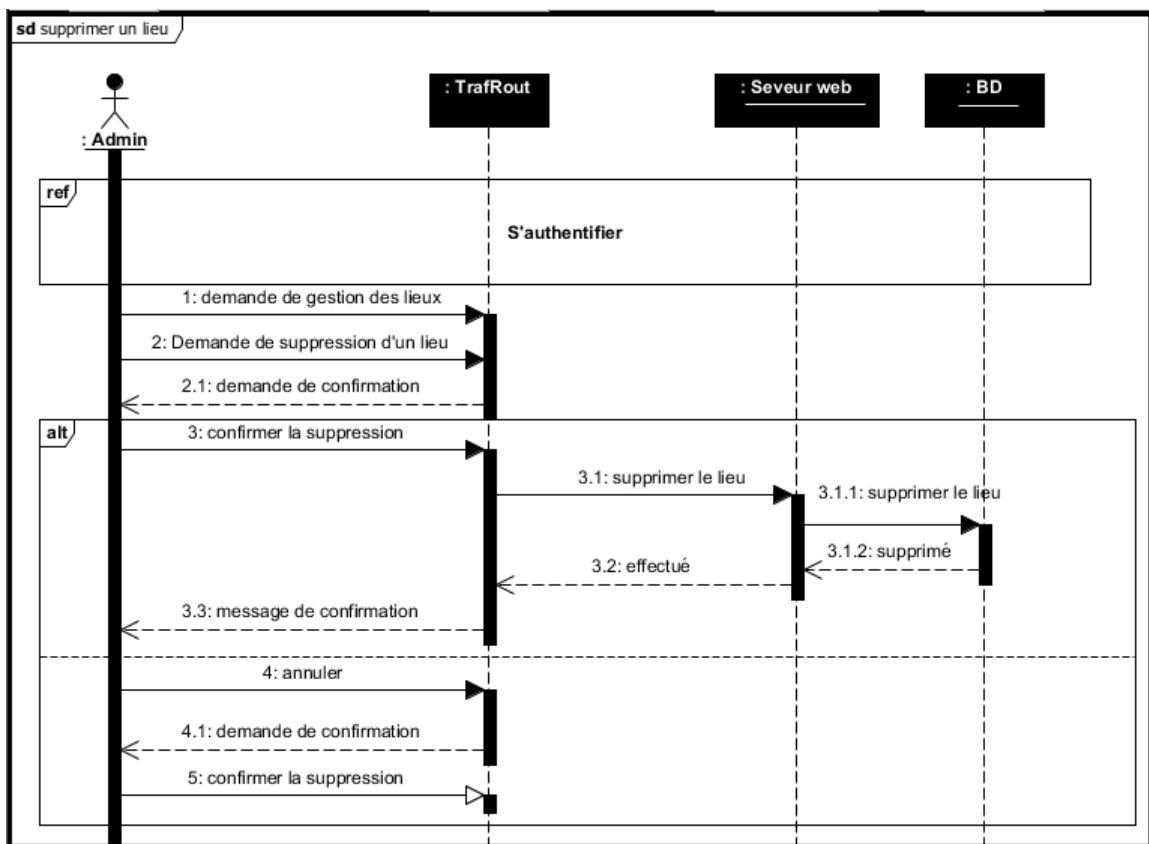


FIGURE 3.8 – Diagramme de séquence du cas d'utilisation " Suppression d'un lieu ".

5. Diagramme de séquence du cas d'utilisation " Ajout d'une route "

Pour ajouter une route, l'administrateur doit s'authentifier pour qu'il ait accès puis remplit le formulaire d'ajout ensuite sélectionne les lieux dont cette route traverse afin de la tracer.

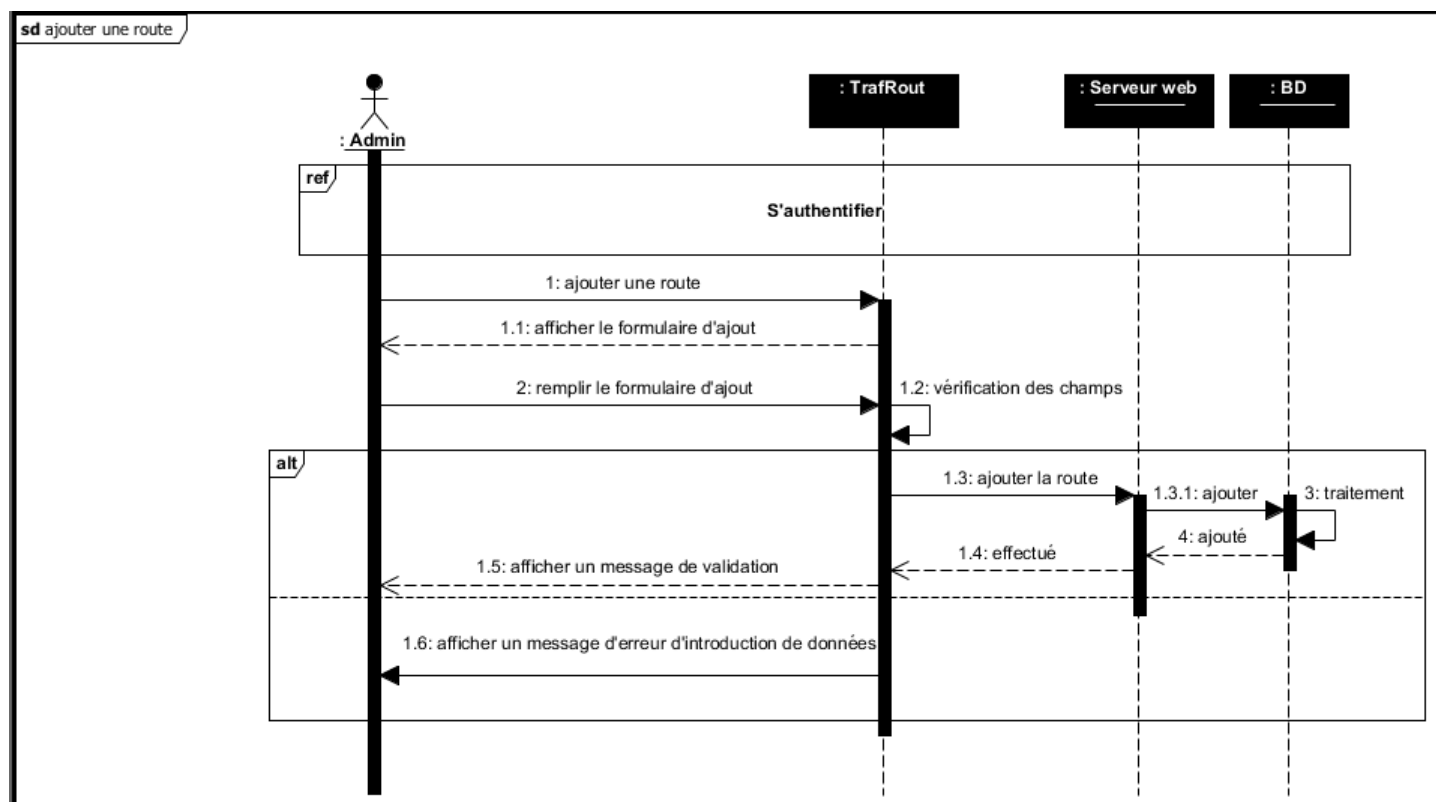


FIGURE 3.9 – Diagramme de séquence du cas d’utilisation ” Ajout d’une route ”.

3.2.7.2.2 Diagrammes d’activité : UML permet de représenter graphiquement le déroulement d’un cas d’utilisation en se servant de cercle définissant une action et qui provient soit d’un utilisateur soit d’un système, et de la flèche orientée définissant l’interaction entre ces actions [20].

◇ Diagramme d'activité du cas d'utilisation " s'authentifier "

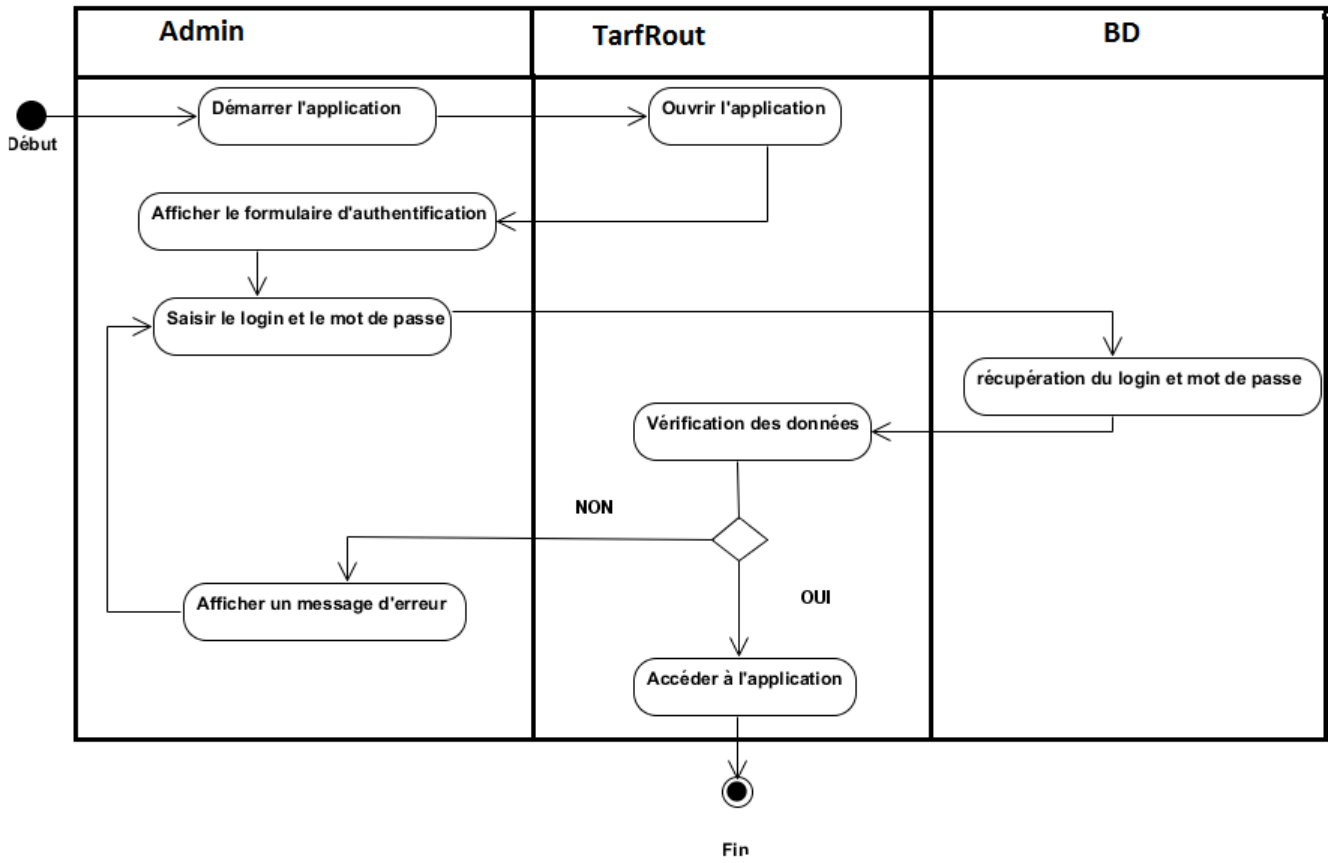


FIGURE 3.10 – Diagramme d'activité du cas d'utilisation " s'authentifier ".

◇ Diagramme d'activité du cas d'utilisation " Ajout d'une route "

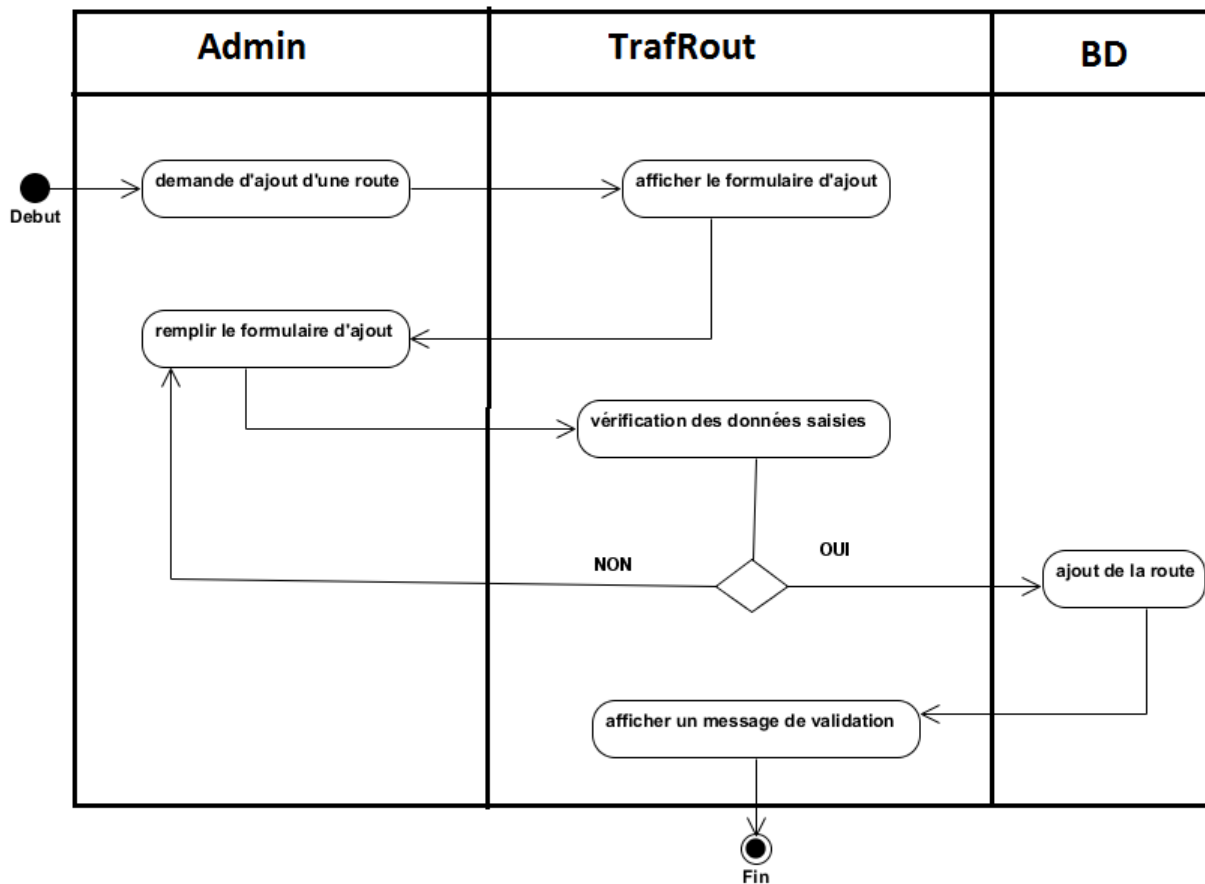


FIGURE 3.11 – Diagramme d'activité du cas d'utilisation " Ajout d'une route ".

Après avoir réalisé les diagrammes de séquence et d'activité, on passe au diagramme de classes.

3.2.7.2.3 Diagramme de classes : un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes [18].

- **Classe** : c'est la classe qui sert à regrouper les objets partageant la même structure de données et le même comportement. On dit qu'un objet est une instance d'une classe. Autrement dit, une classe est un modèle d'instanciation d'objet cela signifie que les objets de même type de classe disposeront des même attributs et méthodes définis dans la classe. La classe est séparée en deux parties [14] :
 - ◇ **Les attributs** : les attributs représentent la description des données propre à chaque classe d'objets.
 - ◇ **Les méthodes** : les méthodes représentent l'ensemble des actions, procédures, fonctions ou opérations que l'on peut associer à une classe. L'ensemble des méthodes de la classe définit le comportement.
- **Relation** : une relation exprime la sémantique bidirectionnelle entre deux classes [14].
- **L'agrégation** : l'agrégation est une relation qui permet de représenter un lien de type " ensemble " comprenant des " éléments ". Il s'agit d'une relation entre une classe représentant le niveau " ensemble " et 1 à n classes de niveau " éléments ". L'agrégation représente un lien structurel entre une classe et une ou plusieurs autres

classes [14].

- **Composition** : la composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe "composant" et la ou les classes "composé". Autrement dit la suppression de la classe "composé" implique la suppression de la ou des classes "composant" [14].
- **L'héritage** : l'héritage permet à une sous-classe de disposer des attributs et opérations de la classe dont elle dépend [14].

1. Identification des classes

Dans cette étape, on va identifier les classes qui représentent les entités réelles et/ou abstraites constituant le domaine de notre étude, nous avons pu extraire les classes suivantes :

- (a) **Commune** : cette classe représente la carte géographique dans la quelle sont situés les paramètres (les lieux) de recherche d'une route.
- (b) **Route** : c'est la classe qui représente les informations concernant le chemin liant deux lieux.
- (c) **Lieu** : cette classe représente les places existantes et situées sur une carte.
- (d) **Utilisateur** : cette classe représente les personnes qui auront l'accès à utiliser l'application.

2. Identification des principales relations entre classes

- Une commune se compose d'un ou plusieurs lieux.
- Un utilisateur trouve une ou plusieurs routes.
- Une route traverse plusieurs lieux.

3. Dictionnaire de données

Le dictionnaire des données est à la fois le support de travail et le résultat de la recherche et d'analyse des données, il se présente sous la forme d'un tableau.

Attributs	Signification de l'attribut	Type	Longueur
login	Le login d'utilisateur/administrateur	varchar	50
password	Le mot de passe	Varchar	50
id_commune	L'identifiant de la commune	Entier	—
nom	Le nom d'une commune	Varchar	50
wilaya	La ville où elle appartient la commune	Varchar	50
id_lieu	L'identifiant du lieu	Entier	—
nom	Le nom du lieu	Varchar	50
latitude	La latitude du lieu	Varchar	50
longitude	La longitude du lieu	Varchar	50
id_route	L'identifiant de la route	Entier	—
distance	La distance qui sépare deux lieux donnés	Entier	—
lieu_dep	Le point de départ d'une route	Varchar	50
lieu_arri	Le point d'arrivée d'une route	Varchar	50
description	Décrit l'état de la route	Varchar	50

TABLE 3.6 – Dictionnaire de données.

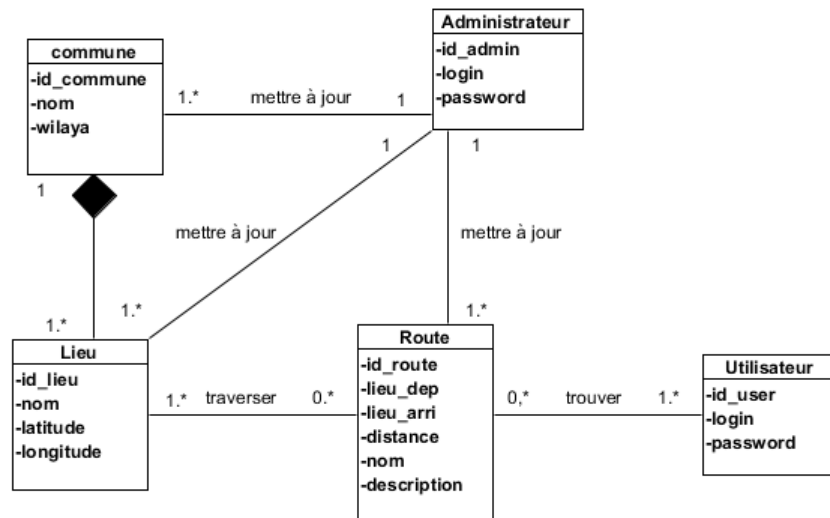


FIGURE 3.12 – Diagramme de classes

Après avoir réalisé le diagramme de classes, il convient d'aborder sa traduction en modèle relationnel.

4. Passage au modèle relationnel

Le modèle relationnel est une manière de modéliser les informations contenues dans une base de données, qui repose sur des principes mathématiques inventés par E.F.CODD en 1970, il est souvent considéré comme le plus simple et le plus élégant des modèles [15].

La traduction du diagramme de classes au modèle relationnel consiste à appliquer les règles de passage suivantes :

- **La traduction des classes en table :**

- ◊ Une classe est traduite en une relation (table).

- **La traduction des relations en tables :**

- ◊ Chaque relation " plusieurs à plusieurs "est traduite par une table distincte.

- ◊ Chaque relation " un à plusieurs " disparaît, la clé primaire du père va migrer vers le fils.

- ◊ Les relations " n-aire "sont traduites par des tables.

Après le passage du diagramme de classes au modèle relationnel, nous présentons ci-dessous les principales relations obtenues :

User(id_user, password, login).

Route(id_route, distance, description, lieu_dep, lieu_arr, nom, id_commune*).

Lieu (id_lieu, nom, latitude, longitude, id_commune*).

traverser(id_lieu*, id_route*).

trouver(id_user*, id_route*).

commune(id_commune, nom, wilaya).

3.3 Conclusion

À l'issue de cette étape, nous avons pu exprimer clairement les objectifs attendus du futur système à concevoir, ainsi que l'analyse associée à chaque cas d'utilisation et après avoir identifié les règles de gestion du système et les classes associées, nous avons procédé à l'extraction du dictionnaire des données afin d'élaborer un diagramme de classe détaillé. À base de ce dernier, nous avons abouti au modèle logique des données en faisant appel aux règles de passages. Il faut noter que l'étape d'analyse et conception sont des activités utiles qui vont nous permettre d'introduire la prochaine étape du Processus Unifié intitulé " réalisation ", que nous allons la détailler dans le chapitre suivant.

Chapitre 4

Réalisation et test

4.1 Introduction

A ce stade du processus UP, les cas d'utilisation sont terminés, le problème a été analysé en profondeur ; nous avons défini une conception que nous espérons la mieux appropriée. Nous pouvons alors entreprendre la troisième phase du Processus Unifié composée de deux parties (implémentation et test), avec comme objectif d'aboutir un produit final et exploitable par les utilisateurs.

Dans ce chapitre, nous allons présenter l'environnement logiciel, les technologies et les langages de programmation que nous avons utilisé ainsi que l'architecture matérielle mise en place suivi de quelques interfaces graphiques de l'application mobile ainsi que celle de la partie administration implémentée avec le langage Java accompagnées par des tests.

4.2 Les outils de développement

Dans ce qui suit nous présentons les outils de développement :

4.3 Environnement logiciel

4.3.1 Le système d'exploitation Android

Avant de présenter le système d'exploitation Android, nous allons définir les terminaux mobiles.

4.3.1.1 Les terminaux mobiles

4.3.1.1.1 Définition des terminaux mobiles : les terminaux mobiles se sont grandement améliorés au cours des dernières années ; leurs caractéristiques les distinguent néanmoins des ordinateurs portables classiques par un grand nombre de spécificités uniques malgré leur diversité, on peut citer les plus importantes : la capacité de stockage est limitée, leur persistance est limitée aux données, le processeur est faible, les systèmes d'exploitations sont variés (Android, iOS, Symbian, etc.) et les modes de communication sans fil sont multiples (GPRS(General Packet Radio Service), GPS (Global Positioning System), WiFi(Wireless Fidelity))[19].

Bien qu'ils entrent pleinement dans les champs de l'informatique mobile, les ordinateurs portables possèdent un certain nombre de caractéristiques spécifiques qui les rendent fondamentalement différents des autres types de terminaux mobiles. On peut citer par exemple : leur poids qui est élevé, leur grande capacité de stockage et la puissance du processeur qui est élevée, etc.

4.3.1.1.2 Les différents types de terminaux mobiles : il est possible de regrouper les terminaux mobiles en fonction de leur ergonomie et de leur usage. Nous en avons choisis les plus utilisés qu'on détaille ci-dessous [35] :

- **Smartphones**

Smartphone ou " téléphone intelligent " est un téléphone mobile qui possède des fonctions proches de celles d'un assistant numérique personnel grâce à un système d'exploitation, possédant des fonctionnalités précises

comme l'accès à Internet, la bureautique, la gestion de carnet d'adresses, le multimédia (MP3, vidéo, photos, etc), les jeux et sont mêmes équipés d'un système global de positionnement (GPS)(voir la figure 4.1).



FIGURE 4.1 – Les smartphones [38].

- **Tablettes**

Tablette (en anglais tablet, plaque) est le nom donné à une famille d'ordinateurs portables dépourvus de clavier à touches et munis d'un écran tactile, de la même dimension qu'une feuille A4 ou plus petit (voir la figure 4.2). L'écran tactile est toujours multipoints, donc capable de détecter plusieurs touches simultanées [29].

- ✓ **Utilisation des tablettes**

Ces ordinateurs sont essentiellement tournés vers l'utilisation d'Internet : consultation de pages Web, lecture de journaux en ligne ou de livres électroniques ou messageries. Leurs dimensions excluent l'intégration d'une mémoire de stockage mécanique, comme un lecteur-graveur de DVD ou un disque dur [36].



FIGURE 4.2 – Les tablettes tactiles [39].

- **PDA**s

Assistant personnel ou PDA (Personal Digital Assistant) ou ordinateur de poche en français est un équipement électronique bureautique de poche utilisé principalement pour ses fonctions d'agenda, de répertoire téléphonique et de bloc-note, mais les avancées technologiques ont permis de lui adjoindre des fonctionnalités multimédia, telles que le dictaphone, le lecteur de MP3 et de vidéos auxquels s'ajoutent des programmes qui le transforment en outil de navigation associé à un GPS (Global Positioning System), par exemple, il s'utilise

avec un stylet (voir la figure 4.3) [20].



FIGURE 4.3 – Les assistants personnels ou PDAs [40] .

4.3.1.2 Définition d'Android

Android est un système d'exploitation Open Source pour les terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 15 novembre 2007. Afin de promouvoir ce système d'exploitation ouvert, Google a su fédérer autour de lui une trentaine de partenaires réunis au sein de l'Open Handset Alliance. Ils ont comme principaux concurrents Apple avec iPhone OS qui équipe l'iPhone, Research In Motion avec BlackBerry OS, Microsoft et son Windows Mobile, sorti en 2008 [30].

Android a une très grande communauté de développeurs qui produisent des applications diverses et variées pour étendre les fonctionnalités du système d'exploitation. Il y a plus de 100 000 applications sur l'Android Market, ce qui en fait le deuxième environnement de développement le plus populaire, derrière iOS. Les applications sont écrites pour la plupart en Java, et peuvent utiliser le hardware et les fonctionnalités du système via des bibliothèques Java développées par Google.

Le fait que le code soit complètement ouvert, permet à de nombreux développeurs de le modifier pour y ajouter des fonctionnalités ou corriger des bugs, et ainsi de proposer à la communauté des builds personnalisés, souvent plus avancés que les versions officielles proposées par les constructeurs [26].

4.3.1.3 Architecture d'Android

Pour bien comprendre la plateforme Android, nous détaillons par la suite l'architecture du système Android (voir la figure 4.4) [31].

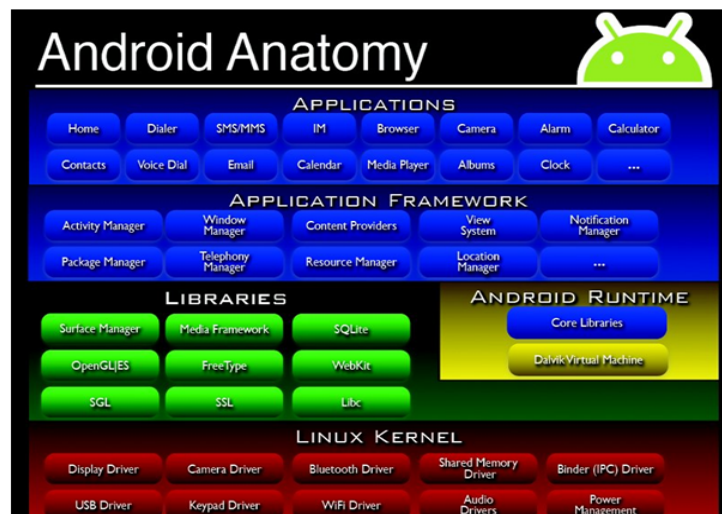


FIGURE 4.4 – Architecture d'Android.

- **Linux Kernel** : Android s'appuie sur le noyau Linux 2.6 pour les services système de base tels que la sécurité, la gestion de la mémoire et des processus, le réseau et la gestion des drivers. Le noyau sert de couche d'abstraction entre le matériel et le reste de la pile logicielle.
- **Android Runtime** : Android inclut un ensemble de bibliothèques fournissant la plupart des fonctionnalités des bibliothèques standard de Java. Chaque application Android s'exécute dans un processus, avec sa propre instance de la machine virtuelle Java, appelée Dalvik. Dalvik a été écrit pour optimiser l'exécution d'une multitude d'instances de la machine virtuelle, avec une empreinte mémoire réduite. Dalvik s'appuie sur le noyau Linux pour les fonctionnalités bas-niveau tels que les threads ou la gestion de la mémoire.
- **Librairies** : Android fournit un ensemble de bibliothèques C/C++ utilisées par différents composants du système. Ces fonctionnalités sont rendues disponibles aux développeurs au travers du framework d'application d'Android. On trouve parmi ces bibliothèques : bibliothèque C standard, moteurs d'affichage 2D et 3D, SQLite, etc.
- **Application Framework** : le framework d'application est la couche qui nous intéresse tout particulièrement. C'est elle qui fait le lien grâce à un ensemble d'APIs (Application Programming Interface) Java, entre le système et l'application. Étant un système ouvert, Android permet aux développeurs de concevoir des applications très riches et de tirer partie d'un maximum de fonctionnalités. Les développeurs ont donc accès aux mêmes fonctionnalités que celles utilisées par les applications fournies avec Android.

Toute application Android repose sur un ensemble de services et systèmes parmi lesquels :

- ◇ Un ensemble de **"Views"** permettant de construire l'interface graphique de l'application : listes, grilles, champs textes, images, et même intégration d'un navigateur web ou d'une vue Google Maps,
- ◇ Des **"Content Providers"** qui permettent aux applications d'accéder à des données des autres applications ou de partager ses propres données,
- ◇ **"Resource Manager"** pour accéder à des éléments autres que du code : données textuelles traduites, images, descriptions XML d'interfaces graphiques, etc.
- ◇ **"Activity Manager"** pour gérer le cycle de vie de l'application.

Ce rapide survol de l'architecture du système permet de mieux comprendre comment fonctionne une application Android. Confinée dans la couche la plus haute, elle accède au système uniquement via les API's Java exposées par la couche Application Framework. Ainsi, si une fonctionnalité est présente dans le noyau Linux (couche rouge sur le schéma) ou dans les bibliothèques système (couche verte), mais qu'elle n'est pas reliée au framework d'application, elle ne sera pas utilisable directement dans une application Android.

4.3.2 Outils de développement

4.3.2.1 Le SDK Android

Le développement d'applications pour la plateforme Android nécessite l'utilisation de kit de développement spécifique à cette plateforme. Un SDK (Software Development kit Android.), c'est-à-dire un kit de développement, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet donc de développer des applications pour Android [32].

4.3.2.2 L'émulateur de téléphone

Le SDK propose un émulateur Android AVD (*Android Virtual Device*). Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. C'est bien évidemment un outil indispensable pour le développement mobile (voir la figure 4.5) [26].



FIGURE 4.5 – Interface de l'émulateur Android.

4.3.2.3 IDE Eclipse

Un IDE (Integrated Development Environment) est un logiciel dont l'objectif est de faciliter le développement, généralement pour un ensemble restreint de langages. Il contient un certain nombre d'outils, dont au moins un éditeur de texte (souvent étendu pour avoir des fonctionnalités avancées telles que l'auto-complétion ou la génération automatique de code) des outils de compilation et un débogueur. Dans le cas du développement Android, un IDE est très pratique pour éviter d'utiliser les lignes de commande [32].

Le langage privilégié pour le développement d'applications Android est justement Java. Google a donc tout naturellement conçu un plugin (un plugin est un module qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités) pour Eclipse car l'Eclipse seul ne permet pas le développement pour Android, Pour pouvoir l'utiliser, il est indispensable d'avoir le plugin ADT (Android Development Tools) avec les fichiers de base pour pouvoir tester, déboguer et exporter les projets au format APK (Android Package Kit), pour pouvoir les publier.

Remarque : ADT (Android Development Tools.) n'est pas le seul add-on qui permette de paramétrer Eclipse pour le développement Android, par exemple : le MOTODEV Studio For Android est aussi très évolué [32,26].



FIGURE 4.6 – Logo de l'IDE Eclipse [33].

4.3.2.4 IDE PHPEdit

Pour écrire les scripts PHP (Hypertext Preprocessor) nous avons choisi la version 12.1 de PHPEdit qui est un environnement de développement intégré spécialisé pour PHP [26].

4.3.2.5 EasyPHP

EasyPHP fut le premier package WAMP (Windows Apache Mysql PHP) à voir le jour (1999). Il s'agit d'une plateforme de développement web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script PHP, ainsi qu'une administration SQL et phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP. Par défaut, le serveur Apache crée un nom de domaine virtuel (en local) 127.0.0.1 ou localhost, ainsi que quand on choisit " Web local " dans le menu d'EasyPHP.

4.3.2.6 JSON

JSON (JavaScript Object Notation) est une syntaxe pour stocker et échanger des informations de texte [34].

Un document JSON ne comprend que deux éléments structurels :

- des ensembles de paires nom/valeur ;
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent trois types de données :

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

Exemple : {

```
"employees" :[
{ "firstName" : "Warda" , "lastName" : "BOUHARIS" },
{ "firstName" : "Wahiba" , "lastName" : "BOUFARIK" },
{ "firstName" : "Akil" , "lastName" : "RAMDHANI" }
]
}
```

Le principal avantage de l'utilisation de JSON, dans notre application, est qu'il est simple à mettre en œuvre et léger, ce qui le rend bien adapté aux terminaux mobiles [26]. JSON est utilisé de la manière suivante :

Lorsque l'application s'exécute, elle se connectera au script PHP. Le script PHP va récupérer les données depuis la base de données MySQL. Ensuite les données seront encodées au format JSON et envoyées au système Android. Ensuite, l'application Android va obtenir ces données codées. Elle les analysera et les affichera sur un terminal mobile.

4.3.3 Les langages de programmation

4.3.3.1 Java

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh et Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur Internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur [41].

4.3.3.2 PHP

C'est un langage de programmation web côté serveur, ce qui veut dire que c'est le serveur qui va interpréter le code PHP (langage de scripts) et générer du code HTML (Hypertext Markup Language) qui pourra être interprété par votre navigateur. Le PHP permet d'ajouter des fonctionnalités de plus en plus complexe, d'avoir des sites dynamiques, de pouvoir gérer une administration de boutique en ligne, de modifier un blog, de créer des réseaux sociaux, etc [42].

4.4 Environnement matériel

4.4.1 Architecture matérielle

TrafRout est une application qui se connecte à un serveur de base de données distant, via Internet, afin de récupérer les données. Ce qui nécessite l'intégration d'un serveur web entre l'application client et le serveur de base de données.

Si nous parlons de l'architecture 3-tiers de point de vue technologique, le client est la plateforme Android, le serveur web est apache et le serveur de bases de données est le MySQL (voir la figure 4.7) :

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

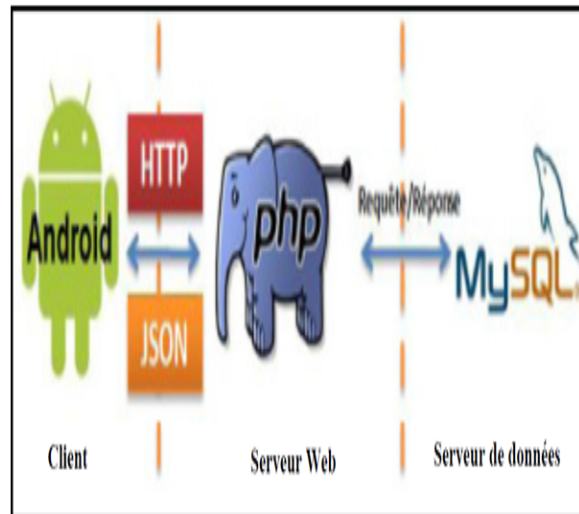


FIGURE 4.7 – Architecture trois tiers de TrafRout.

4.5 Interfaces graphiques de TrafRout

Notre application contient deux catégories d'utilisateurs : l'administrateur du système qui s'occupe de la mise à jour des données et les autres utilisateurs qui sont les clients exploitant l'application, donc, il y a lieu de deux types d'interfaces, administrateur : développée avec Java et utilisateur : développée sous Android.

4.5.1 Interfaces administrateur

La partie administrateur de l'application permet de mettre à jour les informations relatives aux communes, lieux et routes. L'accès à cet espace de gestion nécessite de passer par une authentification pour pouvoir accéder à la forme principale.

4.5.1.1 Formulaire d'authentification

L'accès à l'espace administration oblige une authentification de la personne qui devra entrer son login et son mot de passe. Si les champs introduits sont vides ou erronés, une boîte de dialogue s'ouvre signalant l'erreur.

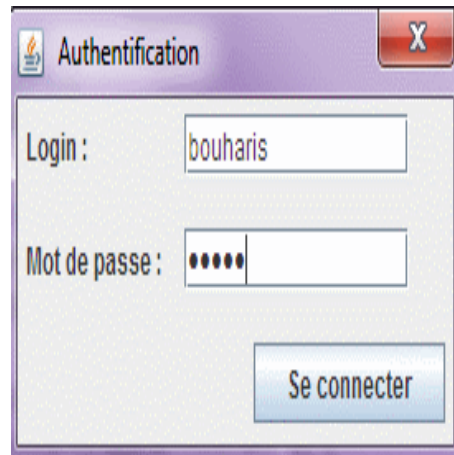
The image shows a Windows-style dialog box titled "Authentification". It has a standard title bar with a close button (X) in the top right corner. The dialog contains two text input fields. The first field is labeled "Login :" and contains the text "bouharis". The second field is labeled "Mot de passe :" and contains five black dots, indicating a password. Below the input fields is a single button labeled "Se connecter".

FIGURE 4.8 – Formulaire d'authentification.

4.5.1.2 Ajout des lieux

L'insertion des lieux se fait par le remplissage d'un formulaire comprenant des informations sur le lieu sans oublier de mentionner la commune dont il appartient pour ensuite l'afficher sur la liste. Une fois l'ajout a été effectué afin que l'ajout d'une route se servira de cette liste.

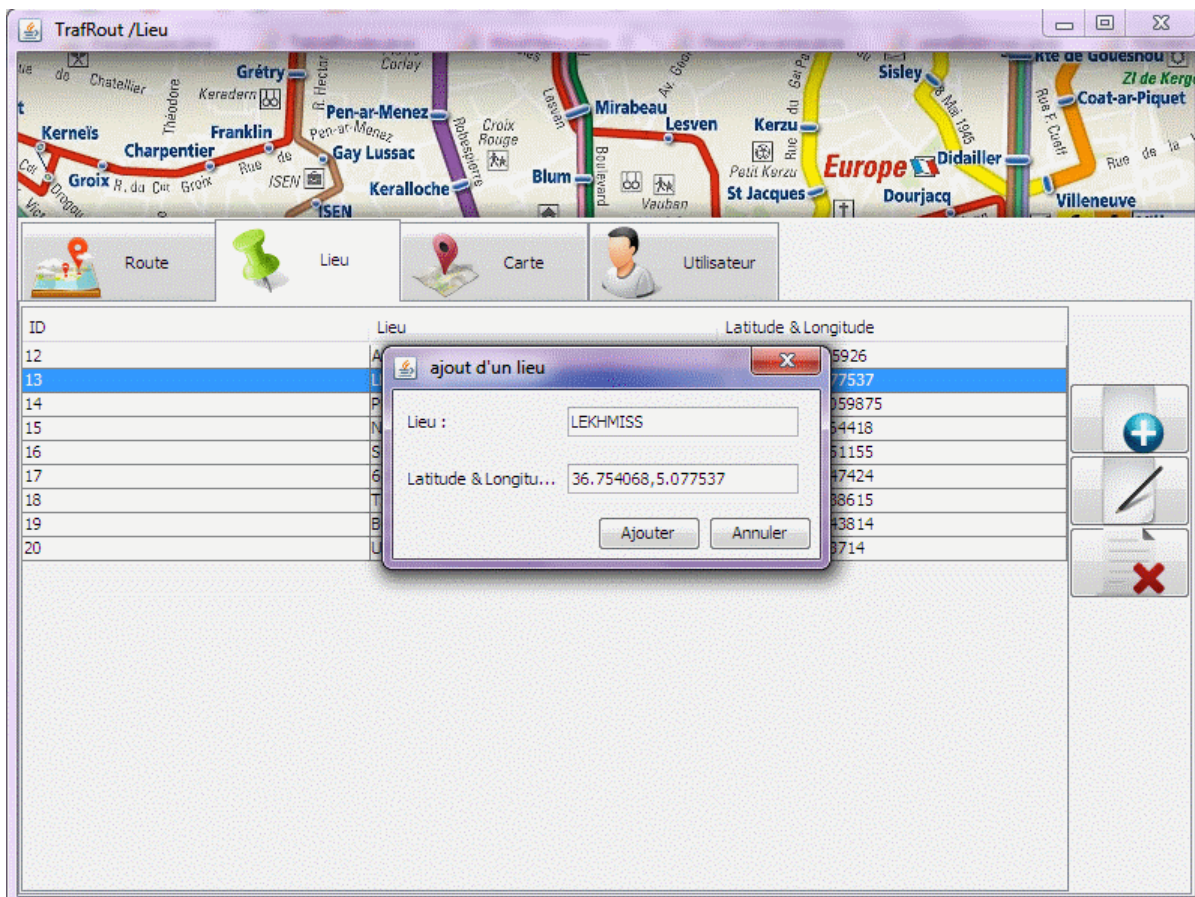


FIGURE 4.9 – Ajout d'un lieu.

4.5.1.3 Ajout des routes

L'insertion des routes se fait par le remplissage d'un formulaire comprenant des informations sur la route sans oublier de mentionner la commune dont il appartient pour ensuite l'afficher sur la liste une fois l'ajout a été effectué.

Ajout d'une route

ID :

Depart : Lat & Long :

Arrivé : Lat & Long :

Distance :

Description :

La route :

La commune :

ID	Lieu	Latitude & Longitude
12	AAMRIW	36.751543, 5.055926
13	LEKHMIS	36.754068, 5.077537
14	PEPINIERE	36.7449588, 5.059875
15	NACERIA	36.751176, 5.064418
16	SOUK ALFALLAH	36.747049, 5.051155
17	600 LGTS	36.736698, 5.047424
18	TAZBOUJT	36.747695, 5.038615
19	BOUKHIAMA	36.750829, 5.043814
20	UNIVERSITE	36.751388, 5.03714

ID	Lieu	Latitude & Longitude
15	NACERIA	36.751176, 5.064418
12	AAMRIW	36.751543, 5.055926

FIGURE 4.10 – Ajout d'une route.

4.5.2 Interfaces utilisateur

Sur cette interface apparait une icône de l'application qui permet d'accéder à ses différentes fonctionnalités (voir la figure 4.11).

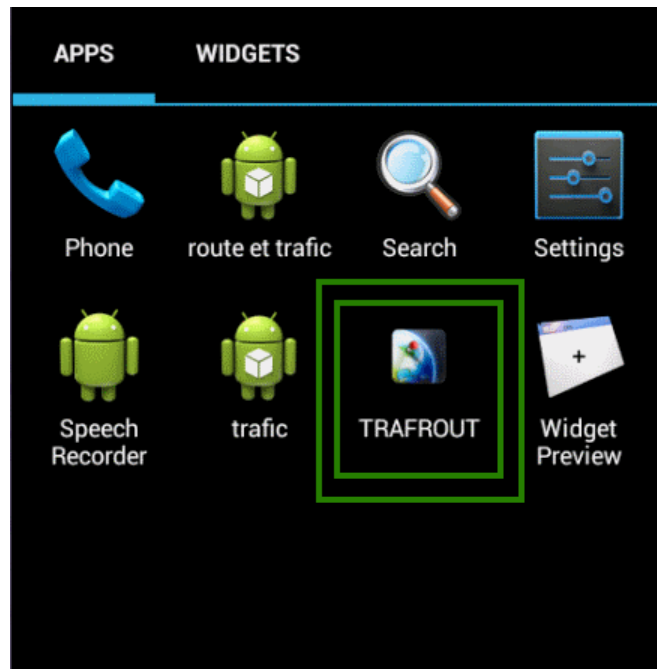


FIGURE 4.11 – Icône de TrafRout.

4.5.2.1 Page d'authentification

Dans la page d'authentification, l'utilisateur doit passer par un système d'authentification sécurisé en saisissant son login ainsi que son mot de passe. Le système affiche la page convenue si l'authentification aurait réussi, sinon un message d'erreur sera affiché.



FIGURE 4.12 – Page d'authentification.

4.5.2.2 Interface d'accueil

Quand l'authentification se fait avec succès, l'interface principale de notre application se lance et affiche la figure suivante :



FIGURE 4.13 – Interface d'accueil.

4.5.2.3 Sélection des lieux

Afin que l'utilisateur sélectionne les noms des lieux de départ et d'arrivée, une liste des noms existants apparaît, il suffit de cocher le lieu voulu, cela a pour but d'éviter la saisie manuelle.

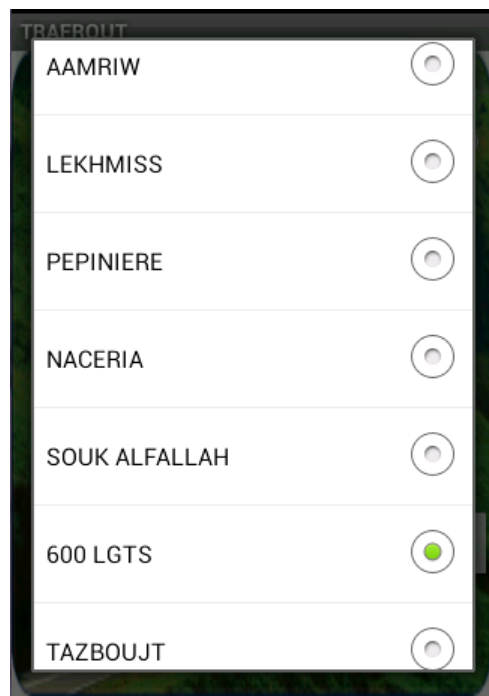


FIGURE 4.14 – La liste des lieux.

4.5.2.4 Trouver le chemin optimal

Afin de montrer le chemin liant le lieu de départ et le lieu d'arrivée après les avoir choisis (à partir des composants 1 et 2 de l'interface d'accueil) et lancé la recherche en cliquant sur le bouton 3, la route optimale apparaît sur la carte, l'utilisateur peut visualiser aussi les autres routes disponibles.

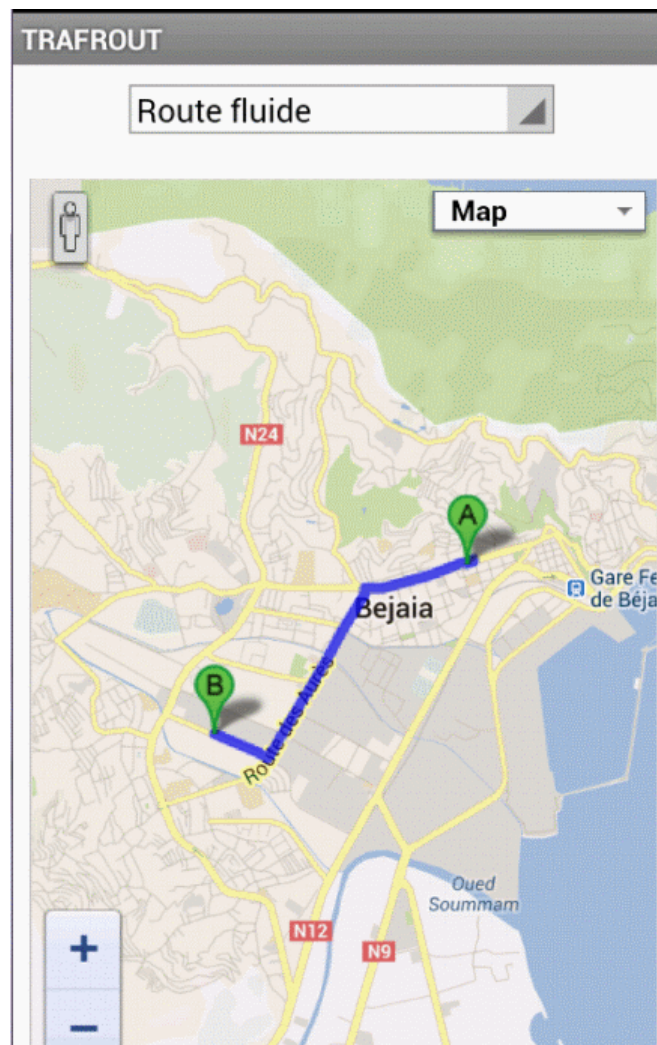


FIGURE 4.15 – Représentation de la route sur la carte.

4.6 Test

Cette activité consiste à tester les résultats de l'implémentation pour s'assurer du bon déroulement des fonctionnalités du système. Lors de l'évaluation des tests effectués, si nous détectons une anomalie quelconque, nous devrions la corriger.

◇ Test du cas d'utilisation "s'authentifier"

Pour tester le cas d'utilisation "s'authentifier", nous avons rempli les champs spécifiques pour le login et le mot de passe avec des données inexistantes dans la base de données et après la validation, un message d'erreur est affiché.



FIGURE 4.16 – Test du cas d'utilisation ” s’authentifier ”.

◇ **Test du cas d'utilisation ” ajout d’un lieu ”**

Nous avons saisi les informations concernant un lieu puis nous avons validé l’enregistrement, le résultat consiste à avoir le lieu ajouté dans la liste des lieux.

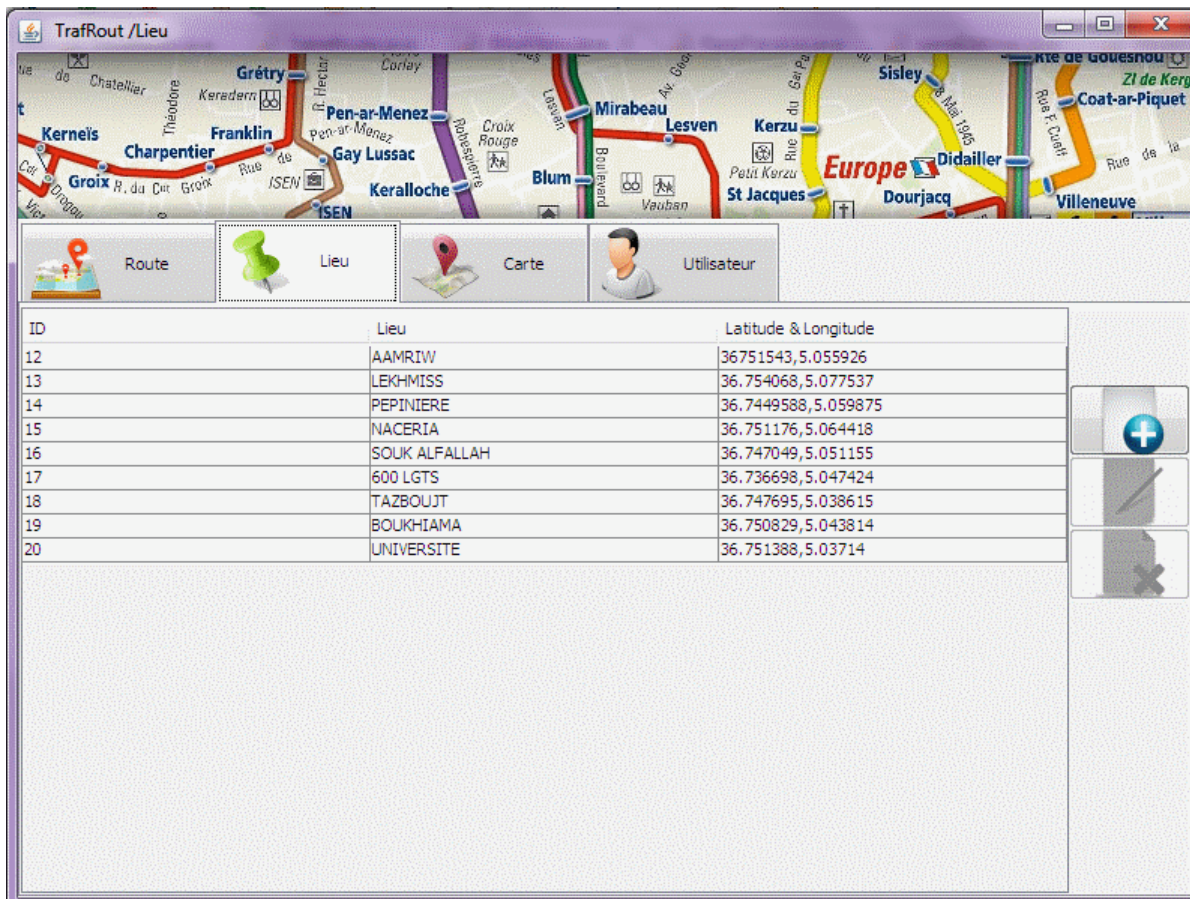


FIGURE 4.17 – Test d’ajout d’un lieu.

◇ **Test du cas d’utilisation ” ajout d’une route ”**

Pour ajouter une route, nous avons introduit les informations concernant la route dans leurs champs appropriés, l’ajout s’effectue et la route ajoutée s’affiche dans la liste des routes.

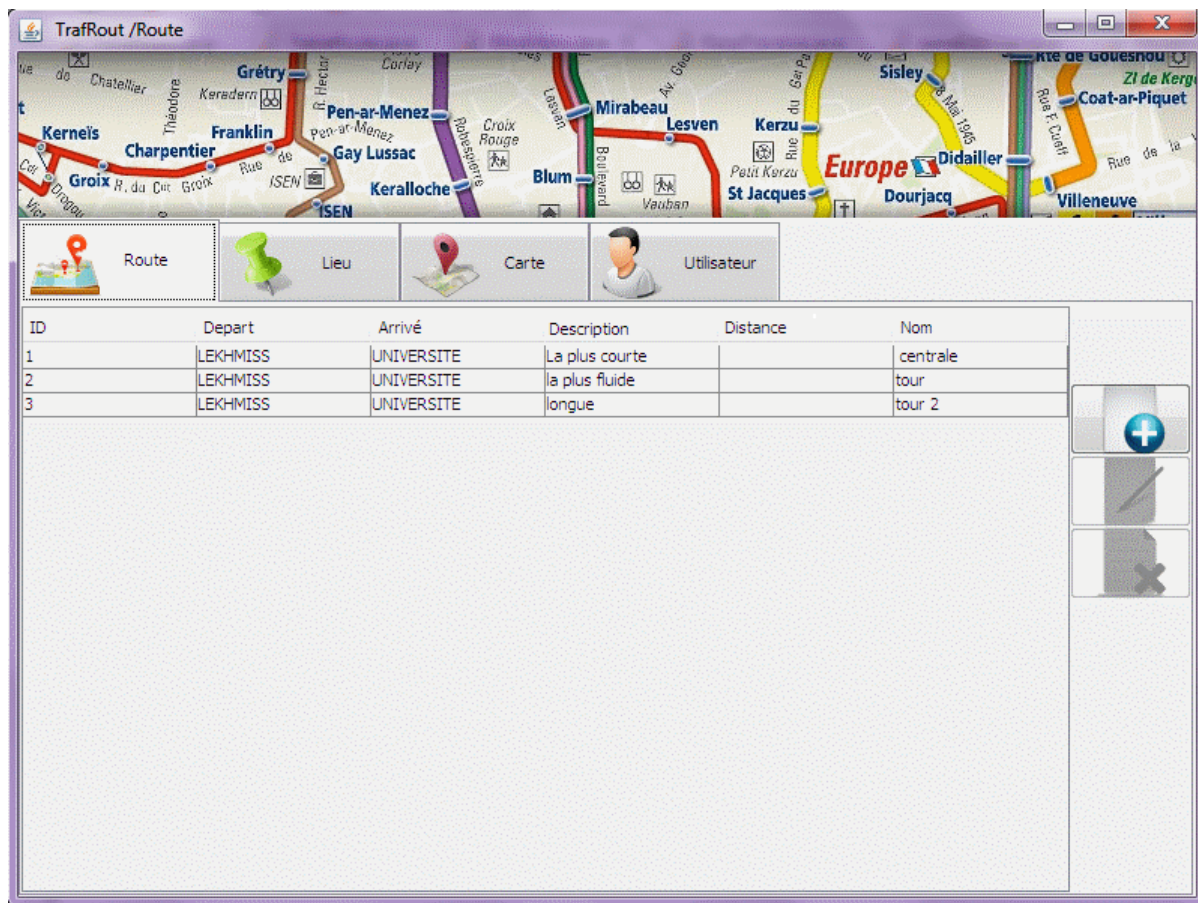


FIGURE 4.18 – Test d’ajout d’une route.

4.7 Conclusion

Dans ce chapitre, nous avons décrit brièvement l’environnement logiciel en spécifiant l’environnement de développement, l’architecture matérielle du système, l’implémentation de quelques exemples de la base de données, puis donné des exemples des interfaces graphiques des l’application mobile TrafRout ainsi que celle de la partie administration (l’application java qui permet d’administrer tels que l’ajout, la suppression et la modification de la base de données) et nous achevé le chapitre par trois exemples de testes importants.

Conclusion et perspectives

Ce mémoire consiste à concevoir et réaliser une application mobile sous Android donnant une aide aux conducteurs au niveau de la ville de Béjaïa vu qu'elle offre une démonstration graphique claire dans le cadre des infrastructures routières afin de faire face à la congestion en trouvant la route la plus fluide basée sur des observations et estimations.

A cet effet, Nous avons présenté l'application avec ses fonctionnalités modélisées suivant le langage UML que nous avons appris durant ce travail, notre choix de l'UML comme langage représentatif des méthodes à objet a été fait pour son utilisation très large par les concepteurs d'application informatique et vu qu'il bénéficie d'un large consensus international.

En effet, la solution adoptée consiste à exploiter les services offerts par GoogleMaps.

Comme perspectives, Notre outil obtenu atteindra sa maturité au fil de son utilisation, au fur et à mesure de l'intégration de nouvelles données dans les bases, de la validation de celles qui s'y trouvent déjà, de leur adéquation avec les axes de travail et les besoins des utilisateurs, l'intégration d'un moyen de calcul consistant à déduire les régions congestionnées à travers le nombre d'utilisateurs connectés en détectant leur position sera plus avantageux, à travers aussi des équipements de détection (exemple : capteurs) de calculs pour rendre son utilité en temps réel. Ainsi de fournir une information sur la disponibilité d'une route qui peut être bloquée.

Bibliographie

- [1] P.ROQUES. *UML 2 Modéliser une application Web*. 4ème édition. EYROLLES. 2007.
- [2] P.ROQUES. *UML 2 Modéliser une application web (les Cahiers du Programmeur)*. 3ème édition. EYROLLES. 2007.
- [3] P.ROQUES. *UML 2 en action*. 3ème édition. EYROLLES. 2006.
- [4] J.GABAY, D.GABAY. *UML 2 analyse et conception*. DUNOD. 2008.
- [5] H.BERSINI. *L'orienté objet cours et exercices*. EYROLLES. 2007.
- [6] P.ROQUES. *UML par la pratique*. EYROLLES. 2006.
- [7] H.BALZERT. *UML 2 Compact*. 2 ème édition. EYROLLES. 2006.
- [8] O.CARLES, G.OMIER, M.MANOUVRIER, Y.NAIJA, G.SCEMAMA. *Entrepôt de données complexes pour le trafic routier*. LAMSADE. 2006.
- [9] C.BUISSON, JB.LESORT. *Comprendre le trafic routier Méthodes et calculs*. CERTU. 2010.
- [10] P.MARMONIER. *Comprendre les SIG*. EYROLLES. 2003.
- [11] X.BLANC, I.MOUNIER. *UML2 pour les développeurs, Cours avec exercices corrigés*. EYROLLES.
- [12] M.SOURIS. *Les principes des systèmes d'information géographiques. Principes, algorithmes et architecture du système*. SAVANE. 2010.
- [13] *Système d'Information Géographique. Les enjeux de la société de l'information pour le développement des territoires rhônalpins*. LA CITOYENNE. 2009.
- [14] S.CLAIS. *Etude comparative des systèmes de gestion de bases de données spatiales*. Rapport de stage LIRMM. 2003.
- [15] *Principes fondamentaux de l'information géographique* : http://sigea.educagri.fr/fileadmin/user_upload/public/Ressources_1/Informations_generales/Informations_generales.SIGEA_Ref.pdf. Mise à jour 2010.
- [16] C.SATRA. *Les Systèmes d'Information Géographiques, pour l'intégration des systèmes d'information dans les systèmes de formation*. IFREMER. : http://icp.ge.ch/sem/f30405/IMG/pdf/SIG_presentation.pdf.
- [17] E.HABERT. *Cartographie. Laboratoire de cartographie appliquée-IRD* (institut de recherche pour le développement de france). 2000.
- [18] D.SANGA, B.DOSSO. *L'utilisation des systèmes d'information géographiques dans les Instituts*. Bureaux nationaux de statistique africains, journal statistique africain Volume 5 PP 165. 2007.

- [19] G.CARPENIER, T.COUSTENOBLE, B.CROMBE. *Solutions Mobiles avec les logiciels IBM Lotus, DB2, WebSphere, trivoli et Rational*. DUNOD. 2003.
- [20] C.BATTARD, S.BRIVET, F.GACOGNE, M.NOUCHY. *Informatique nomade*, Central formation. 2004 : <http://frantzgacogne.free.fr/ftp/informatiquenomade.pdf>
- [21] V.IORDANOVA. *Contribution à la modélisation et la commande du trafic routier : Approches par Bond Graph et commande par platitude*. Thèse de doctorat, l'université d'artois, 2006.
- [22] F.YAN. *Contribution à la modélisation et à la régulation du trafic aux intersections : Intégration des communications Véhicule-Infrastructure*. Thèse de doctorat, université de Technologie de Belfort-Montbéliard Ecole Doctorale Science Physiques pour l'Ingénieur et Microtechniques, 2012.
- [23] M.DJOUMER, A.FERHANI. *Conception et réalisation d'un SIG pour la régulation du trafic routier urbain*. Mémoire d'Ingénieur, université de Béjaia, 2012.
- [24] L.MEZHOUD, I.HAMIDECHÉ. *Conception et réalisation d'un SIG pour l'évaluation et l'analyse des risques naturels pour la Wilaya de Bejaia*. Mémoire de master 2, université de Béjaia. 2011.
- [25] L.BERARDOBARA. *Apport des systèmes d'information géographiques*. Thèse de doctorat, l'école Polytechnique Fédérale de Lausanne. 2003.
- [26] M.DAHDEH. *Conception, développement et intégration d'une application embarquée de téléchargement des applications Android FTAB STORE*. Mémoire master professionnel en nouvelles Technologies de Télécommunications et Réseaux, université virtuelle de TUNIS. 2011.
- [27] Google map, <http://whatis.techtarget.com/definition/Google-Maps>. Mise à jour 2013
- [28] http://commons.wikimedia.org/wiki/File:Raster_vector_tikz.png?uselang=fr. Mise à jour 2013
- [29] L'information géographique, <http://seig.ensg.ign.fr/fiche.php>. Mise à jour 2008.
- [30] <http://android-france.fr/android-cest-quoi/>
- [31] http://gdgdakar.blogspot.com/2011/02/saison-1-la-decouverte-dandroid-episode_13.html. Mise à jour 2011
- [32] www.siteduzero.com. Mise à jour 2012.
- [33] <http://www.webtutoriaux.com/tutoriel-318-java-installation-et-decouverte-de-l-ide-eclipse.html>
- [34] www.json.org
- [35] http://cellphones.about.com/od/smartphonebasics/a/what_is_smart.htm. Mise à jour 2013
- [36] G.PEPIN, http://www.futura-sciences.com/fr/definition/t/technologie-2/d/tablette_12437/. Mise à jour 2012
- [37] www.deltacc.com/delta/fr/_p_37.php. Mise à jour 2010
- [38] <http://www.riskmanagementmonitor.com/the-applesamsung-smartphone-patent-war-continues/>. Mise à jour 2012
- [39] <http://www.rueducommerce.fr/Accessoires-ConsommablesAccessoire-iPad/Stylet-pour-iPad/WACOM/4856350-Stylet-Wacom-pour-tablette-tactile-BAMBOO-STYLUS.htm>
- [40] M.REES, <http://www.pcinpact.com/news/45524-eten-m810-glofish-test-PDAphone.htm>. Mise à jour 2008

[41] http://www.futura-sciences.com/fr/definition/t/internet-2/d/java_485/

[42] <http://www.webmasterfrance.fr/definition-php.html>

- * **Android** : est un système d'exploitation Open Source pour smartphones, PDA et terminaux mobiles.
- * **Authentication** : signature électronique et vérification de l'identité d'un utilisateur. Cette technologie permet de garantir l'authenticité de la source d'une transmission électronique.
- * **AVD** : **A**ndroid **V**irtual **D**evice qui est un émulateur ou en d'autre terme, c'est un terminal mobile virtuel.
- * **Une applet** : est un programme Java dont le code est téléchargé sur le poste client depuis le serveur web et s'exécute ensuite dans le navigateur en utilisant un interpréteur Java (la machine virtuelle, ou JVM, qui est intégrée dans la plupart des navigateurs). Le modèle de sécurité des applets est très strict mais évolutif depuis Java 2 (utilisation des certificats). Autant le HTML pur et les techniques de script peuvent être utilisés par des non-informaticiens (des graphistes par exemple), autant l'utilisation de Java (pour être efficace) nécessite des connaissances préalables en conception et programmation objet.
- * **BD** : **D**ata **B**ase est Ensemble de données organisé en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes.
- * **Blog** : Le terme "Blog" est une abréviation de weblog, qui peut se traduire par " journal sur Internet ". Défini souvent comme un site personnel, il s'agit d'un espace individuel d'expression, créé pour donner la parole à tous les internautes (particuliers, entreprises, artistes, hommes politiques, associations...).

Un blog vous permet d'échanger avec les différents internautes, mais aussi de délivrer des informations sur ce qu'ils souhaitent. Ainsi de nombreux blogueurs parlent de leurs passions, de l'actualité, de ce qui les touchent ou les intéressent. Un blog est un réel espace de communication.
- * **Browser/ Navigateur** : Logiciel (navigateur) permettant de surfer et de télécharger sur leWeb.Ex : Netscape, Explorer, etc.
- * **Cartographie** : désigne la réalisation et l'étude des cartes. Elle mobilise un ensemble de techniques servant à la production des cartes. La cartographie constitue un des moyens privilégiés pour l'analyse et la communication en géographie. Elle sert à mieux comprendre l'espace, les territoires et les paysages. Elle est aussi utilisée dans des sciences connexes, démographie, économie dans le but de proposer une lecture spatialisée des phénomènes.
- * **Client** : Logiciel permettant de procéder à des requêtes auprès d'un serveur.
- * **code** : instructions d'un programme rédigées par un développeur en langage de programmation.
- * **Connexion** : En informatique une connexion Internet est un lien qui relie une machine au reste du réseau mondial informatique.
- * **EDI** : **I**ntegrated **D**evelopment **E**nvironment est un ensemble d'outils destinés à programmer dans un langage donné, qui sont distribués ensemble. il permet de manipuler les outils de programmation depuis une interface graphique simplifiée.
- * **La géomatique** : est la combinaison syntaxique de deux mots : Géographie et Informatique. La géomatique est un domaine qui fait appel aux sciences, aux technologies de mesure de la terre ainsi qu'aux technologies de l'information pour faciliter l'acquisition, le traitement et la diffusion des données sur le territoire (aussi appelées "données spatiales ", "données géospatiales" ou " données géographiques").

- * **GPS** : Global Positioning System est un dispositif de satellites géostationnaires ayant pour but de déterminer la localisation géographique de tout point à la surface de la terre à l'aide de récepteurs électroniques portables.
- * **Google Maps** : est un service basé sur le Web qui fournit des informations détaillées sur les régions géographiques et les sites à travers le monde. En plus des cartes routières classiques, Google Cartes offre des vues aériennes et satellitaires de nombreux endroits. Dans certaines villes, Google Maps offre une vue des rues contenant des photographies prises par les véhicules.
- * **HTML** : HyperText Markup Language est un langage hypertexte à balises (ou marqueurs). Cela veut dire que l'on va gérer la façon dont un texte va s'afficher au sein du navigateur. Ainsi, on peut voir dans tout code source HTML en comparaison d'un éditeur de texte (comme Word pour le plus connu), des différences marquantes, comme des balises (ou marqueurs) dans le texte d'une page web.
- * **Internet** : Internet est un réseau informatique mondial constitué d'un ensemble de réseaux nationaux, régionaux et privés. L'ensemble utilise un même protocole de communication : TCP/IP, (Transmission Control Protocol / Internet Protocol).
- * **Java** : Langage de programmation ressemble à celui du C++, parfait pour créer des programmes à intégrer sur le Web.
- * **JavaScript** : est un langage léger, mais relativement complexe et puissant, qui apporte des fonctions dynamiques à HTML dans les navigateurs. Malgré son nom, il est très différent du langage Java et pas vraiment orienté-objet (pas d'héritage, etc.).
ou Langage de script dont le code s'exécute côté client et qui s'intègre parfaitement aux pages HTML pour créer de petites animations ou interagir avec l'utilisateur. Ne pas confondre avec le Java, langage de programmation sans aucun rapport.
- * **JRE** : ou Java Runtime Environment permet de lire les programmes qui ont été codés en Java. Eclipse est codé en Java donc, pour utiliser Eclipse, il est indispensable d'avoir un JRE.
- * **Logiciel** : traduction du terme anglais Software, le logiciel constitue l'ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique. Dans la famille des logiciels, on trouve par exemple des logiciels d'application qui sont spécifiques à la résolution des problèmes de l'utilisateur (progiciel, tableur, traitement de texte, grapheur, etc.), mais aussi des logiciels d'enseignement ou didacticiels, des logiciels de jeu ou ludiciel, etc.
- * **Mappe** : Représentation de la localisation d'ensembles de données en vue d'en faciliter l'accès.
- * **Raster** : mode de représentation et de stockage des données spatiales par une matrice régulière de cellules ou pixels couvrant un espace déterminé.
- * **Serveur** : a) Logiciel permettant à un ordinateur d'offrir des services à d'autres ordinateurs connectés (clients).
b) Ordinateur sur lequel tourne le logiciel serveur.
- * **SGBD** : Système de Gestion de Base de Données permet de stocker et d'organiser des quantités significatives de données structurées. Les systèmes d'organisation les plus courants sont Relationnel (SGBDR) et Objet (SGBDO).

- * **startup** : est une jeune entreprise en construction qui n'est pas encore lancées sur le marché commercial (ou seulement à titre expérimental). Généralement créée par des étudiants récemment diplômés.
- * **Système embarqué** : peut être défini comme un système électronique et informatique, qui est dédié à une tâche précise. Ses ressources disponibles sont généralement limitées. Cette limitation est généralement d'ordre spatial (taille limitée) et énergétique (consommation restreinte). Les systèmes embarqués font très souvent appel à l'informatique, et notamment aux systèmes temps réel (la prise en compte de contraintes temporelles).
- * **Topographie** : technique de représentation sur un plan des formes du terrain, avec les détails des éléments naturels ou artificiels qu'il porte.
- * **Plugin** : Programme téléchargé et installé dans un navigateur.
Lorsque le navigateur détecte un format de fichier (texte, multimédia, etc.) qu'il reconnaît, il fait appel au "plug-in" concerné pour le traiter.
- * **Vecteur** : mode de représentation et de stockage des données spatiales par des points, des lignes, des polygones.
- * **XML** : Le XML, acronyme de eXtensible Markup Language (qui signifie : langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet. Contrairement à l'HTML qui présente un nombre finit de balises, le XML donne la possibilité de créer de nouvelles balises à volonté.