

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



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

Université de BEJAIA  
Faculté des Sciences Exactes  
Département Informatique

## Mémoire de Master Recherche

Option

INTELLIGENCE ARTIFICIELLE

et

RESEAUX ET SYSTEMES DISTRIBUES

Thème

Guidage routier dans les villes intelligentes. Cas  
d'étude : touristes de la ville de Béjaïa

Presenté par : Mlle REKKAM Meriem

Mlle SILAL Aldjia

Soutenu le : 03 Juillet 2022 Devant le jury composé de :

Présidente	Dr BACHIRI Lina	Maître de conf. A	U. Béjaïa.
Encadreur	Pr BOUALLOUCHE Louiza	Professeur	U. Béjaïa.
Co-encadreur	Dr YESSAD Samira	Maître de conf. B	U. Béjaïa.
Examineur	Dr. MOKTEFI Mohand	Maître de conf. B	U. Béjaïa.

Béjaïa, Juillet 2022.

## ※ *Remerciements* ※

Nous remercions avant tout Dieu le tout Puissant de nous avoir donné santé, courage, volonté et foi afin de réaliser ce modeste travail.

Nous adressons nos plus sincères remerciements particulièrement à M Yanis OULHACI et M Abdelmadjid HENNOUNI, ainsi qu'à nos familles, et tous nos proches et amis respectives qui nous ont accompagnés, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire. Notre profonde gratitude va à notre Co-promotrice Mme S. YESSAD et notre encadrante, Mme L. BOUALLOUCHE pour l'honneur qu'elles nous ont fait en nous encadrant, pour leurs précieux conseils, orientations et la confiance placée en nous, sans eux ce travail n'aurait jamais vu le jour.

Nos sincères remerciements vont aussi à : Mme L.BACHIRI pour l'honneur qu'elle nous a fait d'avoir accepté de présider le jury de notre soutenance.

M M. MOKETFI pour l'honneur qu'il nous a accordé en examinant ce modeste travail.

*REKKAM Meriem & SILAL Aldjia*

## ※ *Dédicaces* ※

Quoi que je fasse ou que je dise, je ne saurai point vous remercier comme il se doit. Tous les mots ne sauraient exprimer ma gratitude et ma reconnaissance. Je vous dédie ce travail comme fruit de vos efforts et l'expression de mon profond amour.

### *\*A mon très cher Père\**

Ce travail est dédié à mon père, la source de mes efforts, celui qui m'a indiqué la bonne voie en me rappelant que la volonté fait toujours les grandes femmes. Cet humble geste est une preuve de mon affection et ma gratitude.

### *\*Ma mère bien-aimée\**

À celle qui a attendu avec patience les fruits de sa bonne éducation, à la lumière de mes jours, maman mon exemple éternel, celle qui s'est toujours sacrifiée. Cela est en témoignage de ma reconnaissance envers sa confiance et sa tendresse.

Aux plus chères et plus proches personnes dans le monde, à mon soutien moral, mes sœurs « Mélissa » « Lydia » et « Amel ».

À mes grands-parents et ma tante « Mima », que je remercie profondément pour leurs aide, soutient et douaaas qu'ils m'ont toujours accordés.

À ma chère binôme « Djidji », pour sa patience, son dévouement, sa compréhension et ses efforts tout au long de ce projet, et à toute sa famille

À mes chers amis, « Salah », « Messaoud », « Fayçal » et « Sabrina » au nom de l'amitié qui nous réunit.

Et spécialement à la personne la plus importante pour moi, à mon soutien moral et source de joie, à mon mari Yanis OULHACI, que je remercie de tout mon cœur pour son aide précieuse et sa présence, sans lui, je ne serai pas arrivée là où je suis.

*REKKAM Meriem*

## ※ *Dédicaces* ※

Quoi que je fasse ou que je dise, je ne saurai point vous remercier comme il se doit. Tous les mots ne sauraient exprimer ma gratitude et ma reconnaissance. Je vous dédie ce travail comme fruit de vos efforts et l'expression de mon profond amour.

### *\*En hommage à mon défunt grand père\**

À la mémoire de mon grand-père décédé, « Ali », a qui je souhaite de tout cœur rendre hommage en lui dédiant ce travail. Par ce geste, j'exprime ma gratitude et reconnaissance, je souhaite le rendre fier de moi.

### *\*A mon très cher Père\**

Ce travail est dédié à mon père, la source de mes efforts, celui qui m'a indiqué la bonne voie en me rappelant que la volonté fait toujours les grandes femmes. Cet humble geste est une preuve de mon affection et ma gratitude.

### *\*Ma mère bien-aimée\**

À celle qui a attendu avec patience les fruits de sa bonne éducation, a la lumière de mes jours, maman mon exemple éternel, celle qui s'est toujours sacrifiée. Cela est en témoignage de ma reconnaissance envers sa confiance et sa tendresse.

Aux plus chères et plus proches personnes dans le monde, à mon soutien moral, mes deux frères « Karim » et « Nassim » et ma sœur « Lydia ».

À mes grands-parents, « Ima WAWA », « Ima YAYA » et « Jedi Meqrane » que je remercie profondément pour leurs aide, soutien et douaas qu'ils m'ont toujours accordés.

À ma chère binôme « Meriem », pour sa patience, son dévouement, sa compréhension et ses efforts tout au long de ce projet, et à toute sa famille.

Et spécialement à la personne la plus importante pour moi, à mon soutien moral et source de joie, à mon cher mari Lamine BOUDJELAL, que je remercie de tout mon cœur pour sa compréhension et sa présence, sans lui, je ne serai pas arrivée là où je suis.

*SILAL Aldjia*

# Table des matières

Table des matières	i
Table des figures	iv
Liste des abréviations	vi
Résumé	ix
Introduction générale	1
<b>1 Généralités et contexte du projet</b>	<b>3</b>
1.1 Introduction	3
1.2 Problématique	3
1.3 Villes intelligentes	4
1.3.1 Quelques définitions des villes intelligentes	4
1.3.2 Composantes de la ville intelligente	4
1.3.3 Quatre risques importants pour les villes intelligentes	8
1.3.4 Top 5 des villes intelligentes et durables dans le monde	9
1.3.5 Application sur les villes intelligentes	10
1.3.5.1 Parking intelligent	10
1.3.5.2 Eclairage intelligent	10
1.3.5.3 Détection des feux de forêt	10
1.3.5.4 Pollution de l'air	11
1.3.5.5 Route intelligente	11
1.4 Internet des objets ( <i>IdO</i> )	11
1.4.1 Définition d'objet connecté ( <i>OC</i> )	11
1.4.2 Définition d'Internet des Objets ( <i>IdO</i> )	11
1.4.3 Domaines d'application	12
1.4.4 Evolution de l'écosystème de l'Internet des Objets	13
1.4.5 Architecture de l'Internet des Objets	13
1.4.6 Technologies clés génériques	14
1.5 Systèmes de transports intelligents (STI)	16
1.5.1 Application des STI à l'exploitation des réseaux routiers	16
1.5.2 Contexte et applications au milieu urbain	16
1.6 Réseau véhiculaire <i>VANETs</i>	17
1.7 Internet des véhicules ( <i>IdV</i> )	18

1.8	Avantages d’IdO aux villes intelligentes . . . . .	18
1.9	Conclusion . . . . .	18
<b>2</b>	<b>Etat de l’art sur la gestion du trafic</b>	<b>20</b>
2.1	Introduction . . . . .	20
2.2	Définition de gestion du trafic . . . . .	20
2.3	Terminologie de gestion du trafic . . . . .	21
2.4	Pourquoi évaluer la technologie du trafic intelligent ? . . . . .	21
2.5	Nouvelles technologies derrière la gestion intelligente du trafic . . . . .	21
2.6	Systèmes existants de la Gestion du trafic . . . . .	22
2.6.1	SCATS . . . . .	22
2.6.2	SCOOT . . . . .	22
2.7	Guidage routier . . . . .	23
2.7.1	Définition . . . . .	24
2.7.2	Applications . . . . .	24
2.7.3	Critères d’optimisation . . . . .	24
2.8	Congestion urbaine . . . . .	25
2.8.1	Définition de la congestion . . . . .	25
2.8.2	Nature et causes de la congestion routière dans les zones urbaines . . . . .	25
2.8.2.1	Congestion récurrente . . . . .	25
2.8.2.2	Congestion incidente . . . . .	26
2.8.3	Évaluation de la congestion . . . . .	26
2.9	Algorithmes de plus court chemin dans le domaine du trafic . . . . .	26
2.10	Travaux antérieurs en relation avec le guidage routier . . . . .	27
2.10.1	Architecture SD-IoV . . . . .	27
2.10.1.1	Critique . . . . .	28
2.10.2	Algorithme de routage comparaison . . . . .	28
2.10.2.1	Critique . . . . .	29
2.10.3	Prédiction d’itinéraire . . . . .	30
2.10.3.1	Critique . . . . .	31
2.10.4	IdT/IoT guidage routier . . . . .	31
2.10.4.1	Critique . . . . .	32
2.10.5	Guidage sécurisé sur Manchester . . . . .	33
2.10.6	Comparaison entre les articles . . . . .	34
2.11	Conclusion . . . . .	35
<b>3</b>	<b>Nouvelle Solution pour le Guidage Routier</b>	<b>36</b>
3.1	Introduction . . . . .	36
3.2	Motivation . . . . .	36
3.3	Approche Proposée . . . . .	37
3.3.1	Démarche à suivre . . . . .	37
3.4	Modélisation du trafic routier urbain . . . . .	38
3.5	Relation demande vitesse-débit . . . . .	39
3.6	Impact d’encombrement sur le temps . . . . .	40

---

3.7	Géolocalisation . . . . .	40
3.7.1	Définition . . . . .	40
3.7.2	Techniques de géolocalisation . . . . .	41
3.7.2.1	Géolocalisation par GSM (Global System for Mobile Commu- nications) . . . . .	41
3.7.2.2	Géolocalisation par WIFI . . . . .	42
3.7.2.3	Géolocalisation par adresse IP (sur internet) . . . . .	42
3.7.2.4	Géolocalisation par RFID (Radio Frequency Identification) . . . . .	43
3.7.2.5	Géolocalisation par GPS . . . . .	43
3.8	Outils et Langage de programmation . . . . .	44
3.8.1	Flutter . . . . .	44
3.8.2	MongoDB . . . . .	44
3.8.3	Vscode . . . . .	45
3.8.4	MapBox . . . . .	46
3.8.5	ImgBB . . . . .	47
3.8.6	Pixabay . . . . .	47
3.8.7	Rydmike . . . . .	47
3.8.8	GPS . . . . .	48
3.8.9	Figma . . . . .	48
3.8.10	Git . . . . .	48
3.8.11	Github . . . . .	48
3.8.12	Emulateur . . . . .	49
3.8.13	Material Design Icons . . . . .	49
3.8.14	Json . . . . .	49
3.8.15	Dart . . . . .	50
3.9	Conclusion . . . . .	50
<b>4</b>	<b>Conception et modélisation de GuidTour</b> . . . . .	<b>51</b>
4.1	Introduction . . . . .	51
4.2	Objectifs de l'application . . . . .	51
4.3	Fonctionnalités de l'application . . . . .	52
4.3.1	Affichage de la carte interactive . . . . .	52
4.3.2	Affichage des informations sur les lieux touristiques . . . . .	52
4.3.3	Recherche et filtrage . . . . .	52
4.3.4	Stockage des lieux . . . . .	52
4.3.5	Compatibilité avec les appareils et les systèmes d'exploitation . . . . .	52
4.3.6	Sécurité . . . . .	53
4.3.7	Facilité d'utilisation . . . . .	53
4.4	Exigences techniques . . . . .	53
4.4.1	Systèmes d'exploitation compatibles . . . . .	53
4.4.2	Appareils compatibles . . . . .	53
4.4.3	Prérequis logiciels . . . . .	53
4.4.4	Exigences en matière de performance et de mémoire . . . . .	53
4.4.5	Stockage des données . . . . .	53

4.4.6	Sécurité . . . . .	53
4.5	Contraintes et hypothèses . . . . .	54
4.5.1	Temps limités . . . . .	54
4.5.2	Données des lieux touristiques et des plages . . . . .	54
4.5.3	Connexion Internet . . . . .	54
4.5.4	Utilisation sur mobile . . . . .	54
4.5.5	Langues supportées . . . . .	54
4.5.6	Compatibilité avec les systèmes d'exploitation et les appareils . . . . .	54
4.6	Modélisation . . . . .	55
4.6.1	Diagramme de Classe . . . . .	55
4.6.2	Diagramme de cas d'utilisation . . . . .	56
4.7	Conclusion . . . . .	56
<b>5</b>	<b>Réalisation et Proposition d'amélioration</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Description . . . . .	57
5.3	Architecture de fonctionnement . . . . .	58
5.4	Réalisation et Développement . . . . .	58
5.4.1	Logo . . . . .	58
5.4.2	Organigramme de l'application . . . . .	60
5.4.3	Accueil . . . . .	60
5.4.4	Authentification et Base de données . . . . .	61
5.4.5	Données et Trafic avec MapBox . . . . .	63
5.4.6	Mettre en œuvre la capture d'emplacement . . . . .	64
5.4.7	Utiliser le SDK Mapbox Maps . . . . .	65
5.4.8	Utiliser l'API Mapbox Directions . . . . .	65
5.4.9	Empilez un Carousel Slider sur la carte . . . . .	66
5.4.10	Ajouter des symboles et des couches de lignes . . . . .	66
5.4.11	SDK de navigation étape par étape . . . . .	67
5.5	Problèmes rencontrés . . . . .	68
5.6	Tests et Résultats . . . . .	70
5.6.1	Inscription . . . . .	70
5.6.2	Connexion . . . . .	71
5.6.3	Mot de passe Oublié . . . . .	71
5.6.4	Itinéraire et guidage . . . . .	72
5.6.5	Recherche . . . . .	75
5.6.6	Profil et paramètre . . . . .	75
5.7	Contribution à l'amélioration de GuidTour . . . . .	76
5.7.1	Contribution 1 . . . . .	76
5.7.2	Contribution 2 . . . . .	76
5.8	Comparaison entre notre contribution et les propositions . . . . .	77
5.9	Conclusion . . . . .	77
	<b>Bibliographie</b>	<b>81</b>

# Table des figures

1.1	Schéma des six leviers d'une ville intelligente . . . . .	5
1.2	Smart City wheel . . . . .	6
1.3	Domaines d'application de l'IdO . . . . .	13
1.4	Evolution d'IdO entre 2015 jusqu'a 2025. . . . .	14
1.5	Architecture d'Internet des Objets . . . . .	15
2.1	Architecture SCATS . . . . .	23
3.1	Intersection à quatre branches avec mouvement tourne à gauche, à priorité à droite. . . . .	39
3.2	Temps en fonction de taux d'encombrement et vitesse. . . . .	40
3.3	Illustration de la géolocalisation. . . . .	41
3.4	Principe de la géolocalisation par GPS. . . . .	44
3.5	Flutter Illustration. . . . .	45
3.6	mongoDB Compass. . . . .	45
3.7	VsCode. . . . .	46
3.8	MapBox. . . . .	47
3.9	ImgBB. . . . .	47
3.10	Pixabay. . . . .	48
3.11	Figma illustration . . . . .	49
4.1	Diagramme de Classe . . . . .	55
4.2	Diagramme de cas d'utilisation . . . . .	56
5.1	Architecture du système. . . . .	58
5.2	Logo GuidTour . . . . .	59
5.3	Organigramme de GuidTour . . . . .	60
5.4	Accueil . . . . .	61
5.5	Page de connexion . . . . .	63
5.6	Récupérer le Trafic avec MapBox . . . . .	64
5.7	La table des places touristique . . . . .	66
5.8	Afficher l'itinéraire et le carousel . . . . .	67
5.9	Turn by turn navigation . . . . .	68
5.10	Inscription . . . . .	70
5.11	Les utilisateur dans la BDD . . . . .	70
5.12	Connexion . . . . .	71

---

5.13	Confirmation email . . . . .	71
5.14	Mot de passe incorrecte . . . . .	71
5.15	Récupération de mot de passe . . . . .	71
5.16	Code de récupération . . . . .	72
5.17	Nouveau mot de passe . . . . .	72
5.18	Lieu et localisation . . . . .	73
5.19	Itinéraire à suivre . . . . .	73
5.20	navigation étape par étape . . . . .	73
5.21	Rapport et signalisation . . . . .	74
5.22	Recherche . . . . .	75
5.23	Profil et paramètre . . . . .	75
5.24	Architecture réseau sans fil . . . . .	77

# Abréviation

<b>API</b>	Application Programming Interface
<b>ATIS</b>	Advanced Traveler Information System
<b>BDD</b>	Base De Données
<b>BSSID</b>	Basic Service Set Identification
<b>DLRD</b>	Distributed-Learning-based vehicle Routing Decision Algorithm
<b>DoD</b>	US Department of Defense
<b>EOTD</b>	Enhanced-Observed Timed Difference
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communications
<b>IA</b>	Intelligence Artificielle
<b>IAB</b>	Internet Architecture Board
<b>IANA</b>	Internet Assigned Numbers Authority
<b>IBSG</b>	Internet Business Solutions Group
<b>IDE</b>	Integrated Development Environment
<b>IdO</b>	Internet des Objet
<b>IdV</b>	Internet des véhicules
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IEEE-SA</b>	Institute of Electrical and Electronics Engineers Standards Association
<b>IOS</b>	iPhone Operating System
<b>IoT</b>	Internet of Things
<b>IP</b>	Internet Protocol
<b>ITS</b>	Intelligent Transport Systems
<b>I2V</b>	Infrastructure to Vehicle
<b>Json</b>	JavaScript Object Notation
<b>JWT</b>	Json Web Token
<b>KNN</b>	K-nearest-neighbor
<b>LEED</b>	Leadership in Energy and Environmental Design
<b>MANETs</b>	Mobile Ad Hoc Networks
<b>MAP</b>	Maximum-A-Posterior
<b>M2M</b>	Machine to Machine
<b>NOSQL</b>	Not Only SQL database
<b>NRR</b>	Next Road Rerouting
<b>OBUs</b>	On Board Units
<b>OC</b>	Objet Connecté

**OCE** Objet Connecté Enrichi  
**OTP** OpenTripPlanner  
**PMV** Panneaux à Messages Variables  
**RCSF** Réseaux de capteurs Sans Fil  
**ReFocus+** RE-routing with FOg-CloUd System  
**RFC** Requests For Comments  
**RFID** Radio Frequency Identification  
**RGS** Routing Guidance System  
**RMI** Remote Method Invocation  
**RSUs** Road Side Units  
**SBS** SURF Backend System  
**SCATS** Sydney Coordinated Adaptive Traffic System  
**SCOOT** Split Cycle Offset Optimisation Technique  
**SDIoV** Software Defined Internet of Vehicles  
**SDK** Software Development Kit  
**SFS** SURF Frontend System  
**SIAD** Systèmes d'Information d'Aide au Déplacement  
**SIG** Système d'information géographique  
**SURF** Smart Urban Routing for FIESTA-IoT  
**SUS** System Usability Scale  
**STI** Systèmes/Services de Transport Intelligents  
**STRF** Spatio-Temporal Random Field  
**SUMO** Simulation Urban Mobility  
**TIC** Technologies d'Information et de Communication  
**TRL** Trafic Research Laboratory  
**UI** User Interface  
**UMTS** Universal Mobile Télécommunications System  
**UX** User Expreience  
**VANETs** Vehicular Ad Hoc Networks  
**VNS** Vehicule Navigation Systems  
**V2C** Véhicule au Cloud  
**V2I** Vehicle to Infrastructure  
**V2P** Véhicule à piéton  
**V2V** Vehicle to Vehicle  
**WSN** Wireless Sensor Network

# Résumé

## Résumé

Le guidage routier est devenu incontournable lors de la réalisation d'application de navigation pour assurer la sécurité et le confort des citoyens. L'objectif initial de notre projet était d'étudier le guidage routier dans les villes intelligentes et proposer une amélioration pour mieux assurer la sécurité routière et le confort pour les citoyens. Puis, suite à notre étude bibliographique sur les travaux réalisés dans ce domaine, nous avons proposé de développer une application de guidage routier dédié pour les touristes de la ville de Bejaia. Pour ce faire, nous avons commencé par la récolte des données et des coordonnées géographiques des lieux afin de pouvoir les représenter sur la carte de la ville de Bejaia fourni par Mapbox. Par la suite, la réalisation de l'application est mise en place. Cette dernière vise à proposer des lieux touristiques à visiter à un usager après la détection de sa localisation à l'aide du GPS, puis à lui proposer le meilleur chemin vers la destination choisie basé sur le temps le plus court.

**Mots clés :** Trafic, Guidage routier, villes intelligentes, Bejaia, lieux touristiques

## Abstract

Road guidance has become an essential part of navigation applications to ensure the safety and comfort of citizens. The initial objective of our project was to study the road guidance in smart cities and propose an improvement to ensure a better road safety and comfort for citizens. Then, following our bibliographic study on the works done in this field, we proposed to develop a dedicated road guidance application for tourists in the city of Bejaia. To do this, we started by collecting data and geographical coordinates of places in order to represent them on the map of the city of Bejaia provided by Mapbox. Thereafter, the realization of the application is set up. The latter aims at proposing tourist places to visit to a user after the detection of his location using GPS, then propose the best path to the chosen destination based on the shortest time. Key words : Traffic, Road guidance, smart cities, Bejaia, Tourist places.

**Keywords :** Traffic, Road guidance, smart cities, Bejaia, tourist places

# Introduction générale

Les villes intelligentes sont un domaine vaste et émergent, il englobe énormément de sujet et de domaines de recherches dans les différents secteurs, tel que la sécurité, les hôpitaux intelligents, les parkings, etc. Les villes intelligentes ont été conçues afin d'améliorer la vie des citoyens en général et ont vu le jour suite au développement conséquent des différentes technologies comme les capteurs, les actionneurs et l'internet des objets.

La gestion du trafic intelligent est un secteur qui y figure, qui a pour but d'améliorer le flux de trafic et de gérer les différents changements qui y surviennent. La gestion du trafic propose plusieurs et différentes fonctionnalités, tels que : le contrôle adaptatif, la détection de l'existence de travaux qui impacte le flux de circulation, la détection des embouteillages, la gestion des feux de routes, etc. Une application de gestion du trafic intelligent qui à la fois prédit le changement de flux dynamique et permet à l'utilisateur d'imposer ses propres conditions est indispensables aux citoyens et au gouvernement. Généralement, cette dernière est souvent accompagnée de guidage routier, ce qui permet de suivre l'utilisateur tout au long de son trajet.

L'objectif du guidage est de trouver une solution qui peut mettre fin aux divers problèmes routiers et d'aider les conducteurs ainsi que les passagers à trouver le meilleur chemin à emprunter et de l'optimiser suivant leurs besoins en temps réel.

Guidage routier du terme anglais « routing decision » est un sujet de recherche émergent ces dernières années. En général, il représente des applications qui ont pour but de mettre fin à la congestion et aide à la gestion du trafic routier, il consiste exactement à diriger ou à guider l'utilisateur vers la destination qu'il souhaite en lui choisissant un chemin qui répond à ses divers besoins. Les applications les plus connues sur ce domaine de recherche sont : Google Maps, Waze, HERE WeGo et Mappy, etc. Ces applications englobent des critères qui peuvent être ajoutés suivant le public visé. Par exemple : les lieux touristiques, la distance, le temps, les chemins qui contiennent des aires de repos, la température, paysage panoramique, etc.

Dans ce travail, nous avons fait une étude des travaux proposés dans la littérature pour l'amélioration du guidage routier en général, puis motivées par un travail personnalisé pour la ville de Manchester, nous avons pensé à développer une application pour notre ville Béjaïa. Vu le problème de l'encombrement que nous vivons dans la ville de Béjaïa quotidiennement et sa complication encore plus en été à cause du flux important des touristes, nous nous sommes penchées sur la problématique : « Comment créer une application de guidage routier dédiée pour les touristes, qui à la fois propose des lieux touristiques à l'utilisateur et calcule le meilleur chemin à emprunter vers le lieu de son choix : cas de la ville de Béjaïa ? » et nous avons essayé d'y prodiguer une solution, avec un plan de travail bien suivi. Nous avons commencé par développer une application de navigation avec cartographie et de guidage routier dédiée pour les touristes et visiteurs de la ville de Béjaïa qui a pour appellation « GuidTour ». Pour ce faire, nous avons

choisi plusieurs outils tels que : la plate-forme Flutter, la base de données NoSQL MongoDB, GPS et le langage de programmation Dart. Puis, nous avons proposé deux idées d'amélioration pour cette application. La première consiste à utiliser les algorithmes de machine learning et le deep learning pour l'analyse de l'historique de recherche et les suggestions des utilisateurs récoltés dans l'application. Cette analyse va permettre d'améliorer le service offert par l'application. La deuxième proposition consiste en une architecture basée sur les réseaux de capteurs installés à l'entrée des plages pour récolter des informations sur l'occupation de places de parking et sur la plage et l'envoyer à travers les réseaux véhiculaires vers le cloud. Ces informations seront récupérées et traitées par l'application pour proposer à l'utilisateur qui aime beaucoup plus la tranquillité les plages les moins peuplé en temps réel ou simplement éviter à l'utilisateur de choisir une plage qui est saturée et il ne trouvera même pas une place de parking. Nous allons penser à d'autres améliorations une fois que nous aurons récolté le ressenti des testeurs auxquels nous allons soumettre notre application à travers un questionnaire d'évaluation. Faute de temps, nous n'avons pas pu détailler ces propositions et évaluer l'architecture proposée par simulation.

Notre travail est structuré sur quatre chapitres. Dans le premier chapitre, nous avons parlé de la problématique dans sa globalité et des différentes applications existantes sur les villes intelligentes ainsi que plusieurs autres généralités.

Dans le deuxième chapitre, nous avons traité certaines solutions proposées qui sont reliées à notre problématique et ajouter certaines définitions qui nous sont indispensables.

Le troisième chapitre concerne l'approche à laquelle nous avons apporté des modifications et la démarche que nous allons suivre, une légère présentation du fonctionnement, une modélisation ainsi que les outils employés.

Dans la partie conception, nous avons développé une application de guidage routier pour les lieux touristiques de la ville qui propose le meilleur chemin. Nous avons expliqué dans cette partie les étapes de réalisation, l'architecture, les problèmes et résultats des tests.

# Chapitre 1

## Généralités et contexte du projet

### 1.1 Introduction

La qualité de vie d'un peuple est d'une importance primordiale. Cette dernière est en régression alarmante ces dernières années, et avoir recours aux technologies afin de l'améliorer semble être la solution optimale.

"Comment procéder à ce changement ?", "Comment utiliser ces nouvelles technologies ?", sont des questions qui reviennent à chaque fois, dont la réponse est plutôt simple. La transition vers des villes intelligentes. Ceci implique plusieurs changements sur les différents domaines de vie, dont la difficulté repose sur l'implémentation de la technologie dans la vie quotidienne et l'adaptation à ce changement. Les villes intelligentes ont été développées afin d'améliorer les conditions de vie et de mettre fin aux différents problèmes urbains. Ces dernières ont plusieurs applications et domaines reliés. Cela relèvera peut-être plus de questions telles que : qu'est-ce qu'une ville intelligente ? quelles sont ses applications ? Et quels sont les domaines auxquels elles sont reliées ?

Dans ce chapitre, nous allons essayer de répondre à ces questions, en commençant par présenter quelques domaines qui sont en relation avec les villes intelligentes, à savoir les systèmes de transport intelligents, les réseaux véhiculaires, l'internet des objets (IdO) et l'internet des véhicules (IoV). Puis, nous présentons les villes intelligentes, ses composantes et le top 5 de villes intelligentes et durables dans le monde. Enfin, nous donnons quelques usages des technologies IdO pour la ville intelligente et nous résumons les avantages d'IdO dans les villes intelligentes.

### 1.2 Problématique

Les problèmes qui affectent nos rues et nos autoroutes sont bien connus. Les ralentissements de la circulation peuvent provoquer des embouteillages débilissants et aggraver la pollution atmosphérique urbaine. Les entreprises souffrent de retards de livraison et d'une perte de productivité. Les véhicules d'urgence sont ralentis par les goulets d'étranglement, mettant potentiellement des vies en danger. Et tout cela nuit à la qualité de vie globale de la ville.

Pendant ce temps, les villes et les gouvernements régionaux demandent continuellement à leurs équipes de gestion du trafic, à leurs ingénieurs civils et à leurs équipes d'entretien des routes de faire plus avec moins. Face à ces défis, des villes innovantes - les "villes intelligentes" - utilisent un ensemble coordonné de solutions matérielles, logicielles et en nuage pour fluidifier le

trafic et améliorer la sécurité. Les systèmes intelligents de gestion du trafic, qui sont inclus dans le parapluie des "systèmes de transport intelligents" et parfois appelés "gestion intelligente du trafic", sont des systèmes automatisés qui intègrent les dernières avancées de la technologie de l'Internet des objets (IoT).

Ces systèmes peuvent optimiser la circulation et renforcer la sécurité en utilisant des capteurs, des caméras, des routeurs et des technologies cellulaires pour ajuster de manière dynamique les mécanismes de contrôle tels que les feux de signalisation, les compteurs des bretelles d'accès aux autoroutes, les voies de transit rapides par bus, les panneaux d'affichage des autoroutes et même les limitations de vitesse.

Aujourd'hui, les systèmes intelligents de gestion du trafic permettent d'augmenter la capacité des rues des villes sans réellement ajouter de nouvelles routes. Avec l'avènement de la technologie des véhicules connectés, ces systèmes seront également en mesure de contrôler directement les véhicules en cas de besoin - en les freinant dans les carrefours, par exemple, pour éviter les accidents avec les piétons ou les autres véhicules. Les villes intelligentes déploient ces systèmes dès maintenant pour être prêtes lorsque la technologie des véhicules sera entièrement testée et déployée.

Le trafic dans la ville de Bejaia ces derniers années est devenu un vrai problème . De plus les lieux touristiques ne sont pas valorisés et pas vraiment connu du grand public. L'utilisation de cette évolution numérique et technique ne peut que être bénéfique, non seulement pour les citoyens mais aussi toute personne souhaitant la visité.

## 1.3 Villes intelligentes

Avec des populations de plus en plus nombreuses en zones urbaines, les villes sont aujourd'hui contraintes de s'adapter à cette croissance de manière réfléchiée en adoptant des politiques prévues sur le long terme. C'est de cette obligation qu'est né le statut de "ville intelligente", appelée aussi Smart City en anglais [28].

### 1.3.1 Quelques définitions des villes intelligentes

**Définition 1 :** une ville intelligente (ou Smart city en anglais) est une ville qui privilégie les technologies de l'information et de la communication pour favoriser une meilleure interaction et collaboration avec ses citoyens, et améliorer la qualité de vie de ses habitants [28].

**Définition 2 :** les villes intelligentes (ou Smart Cities), sont des communautés qui exploitent la technologie pour transformer les systèmes et services physiques de manière à améliorer la vie de ses résidents et de ses entreprises tout en rendant l'administration plus efficace. Il ne s'agit pas uniquement d'une simple automatisation des processus, mais d'un lien entre systèmes et réseaux pour rassembler et analyser des données qui sont alors utilisées pour transformer des systèmes entiers [32].

### 1.3.2 Composantes de la ville intelligente

Différents modèles de ville intelligente sont présentés dans la littérature. Les modèles holistiques de Giffinger [30] et de Cohen [31] sont ceux qui sont le plus souvent utilisés pour

démontrer les six composantes de la ville intelligente. Le modèle de ville intelligente présenté dans la Figure 1.1 reprise de [5], de Rudolf Giffinger [30], expert en recherche analytique du développement urbain et régional de l'université technologique de Vienne, présente les six leviers à considérer pour devenir une ville intelligente.

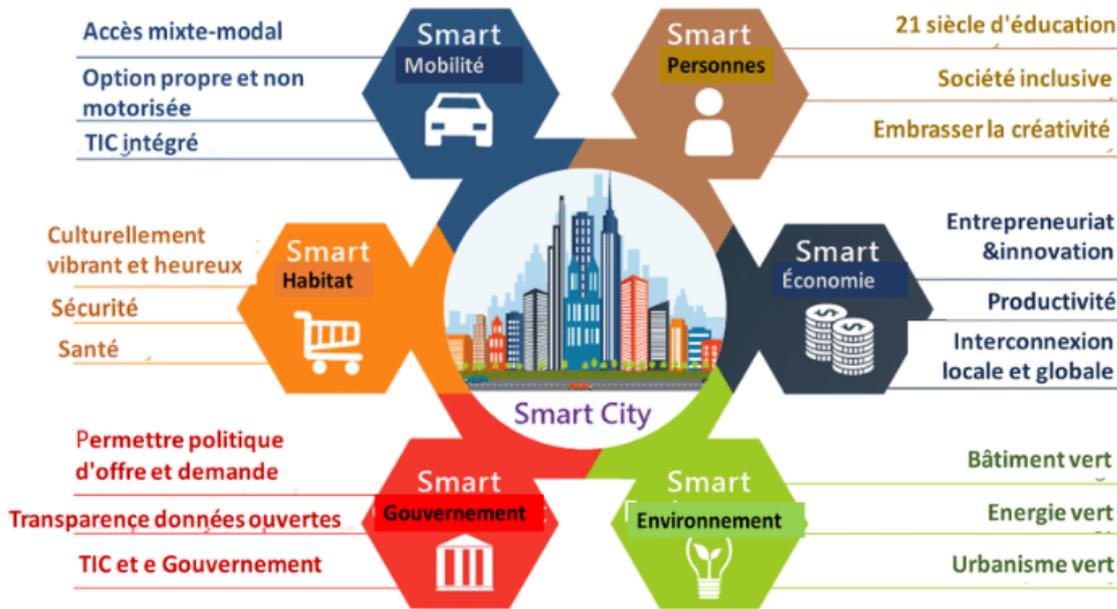


FIGURE 1.1 – Schéma des six leviers d'une ville intelligente

La seconde figure (Figure 1.2 présentée dans [6]) a été élaborée par Boyd Cohen. Cette figure, nommée la Smart City wheel, présente les six dimensions pour devenir une ville intelligente. Dans la deuxième roue de ce diagramme circulaire, Boyd Cohen suggère les domaines dans lesquels les différentes dimensions s'appliquent. Finalement, il propose également divers indicateurs pour mesurer la performance des six dimensions.

Le modèle de ville intelligente de Giffinger et celui de Cohen se ressemblent sensiblement. Les deux modèles intègrent six dimensions pour devenir une ville intelligente. Pour bien comprendre la pertinence de ces six dimensions en relation avec les problématiques auxquelles elles répondent, le tableau 1.1 constitue une synthèse présentant les dimensions ou leviers pour devenir une ville intelligente, les défis urbains actuels, les outils de mise en œuvre ainsi que les objectifs visés.

Les six composantes de la ville intelligente présentées dans les schémas de Giffinger et de Cohen sont illustrées de façon non-hiérarchique. Dans les paragraphes qui suivent, chacune des composantes est définie selon une hiérarchie proposée. Cette hiérarchie est en fait une suite logique de processus ou étapes par lesquels une ville deviendra intelligente [37] :

- **Gouvernance intelligente** : la gouvernance intelligente englobe l'ensemble des usages des technologies pour améliorer la gouvernance urbaine. Les autres dimensions relevées de la gouvernance intelligente sont l'interopérabilité entre les différents services de la ville, la transparence dans la génération des données, la protection des informations et des libertés individuelles des citoyens.

- **Citoyen intelligent** : le citoyen est une importante partie prenante dans la ville intelligente. Sa participation est requise, le citoyen intelligent est celui qui utilisera les nouveaux outils technologiques, notamment pour participer aux débats publics et à la vie de quartier.

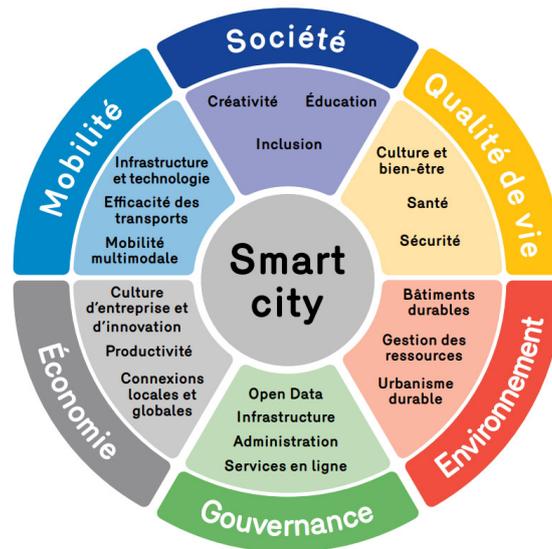


FIGURE 1.2 – Smart City wheel

- **Économie intelligente** : une économie intelligente, c'est un pilier économique dont on se sert comme vecteur pour l'innovation et la création d'emplois durables pour la ville. Selon Giffinger [30], une économie intelligente est basée sur un esprit d'innovation et d'entrepreneuriat, sur la productivité et la flexibilité du marché. Elle possède aussi une aptitude à se transformer et à enchâsser le marché international.

- **Mobilité intelligente** : le transport a été l'un des premiers secteurs à intégrer des dispositifs numériques pour mieux gérer les flux dans la ville. L'accès aux données de transport en temps réel via des écrans électroniques dans les stations, dans les autobus ou encore via les téléphones intelligents personnels permettrait aux usagers de connaître une foule d'informations. Une mobilité intelligente qui serait possible grâce aux divers centres de gestion des données, aux capteurs d'informations et aux caméras. Ainsi, les utilisateurs des transports deviennent des producteurs de données. Une mobilité intelligente passe aussi par le développement et l'accès aux applications qui permettront aux usagers de vivre l'expérience d'une mobilité intelligente.

- **Environnement intelligent** : la ville intelligente prend des mesures pour réduire les émissions de polluants et d'irritants, mais également assure une meilleure gestion des ressources de l'environnement fondé sur les principes du développement durable. Les smart grids ; une technologie informatique des réseaux de distribution d'électricité intelligents ; peuvent optimiser la production et la distribution d'électricité tout en s'ajustant à la demande. Économiser de l'énergie via de nouvelles technologies, c'est aussi ça un environnement intelligent.

- **Habitat intelligent** : l'habitat intelligent peut être applicable à différentes échelles. À l'échelle du milieu de vie, il peut s'agir d'un milieu de vie sécuritaire. De plus, il peut s'agir de développer des quartiers verts ou des éco-quartiers. A l'échelle de l'habitat, il peut s'agir d'habitations écologiques, voire des habitations qui sont certifiées selon le Leadership in Energy and Environmental Design (LEED) ou encore des habitations qui comprennent certaines composantes écologiques [46].

Dimensions de la ville intelligente	Défis urbain actuels	Outils de mise en œuvre	Objectifs et résultats visés
<b>Gouvernance intelligente</b>	-Gouvernement fermée et peu transparente	<ul style="list-style-type: none"> <li>— Panneaux informatifs électroniques</li> <li>— Web diffusion</li> <li>— logiciel pour le suivi des requêtes en ligne</li> </ul>	-Gouvernance intégrée transparente, ouverte, et favorisant la coopération
<b>Citoyen intelligent</b>	-La qualité des services rendus aux citoyens. -La participation citoyenne.	-Plateforme en ligne pour favoriser la participation du public	-Favoriser la créativité et la flexibilité des citoyens et la participation à la communauté dans le but d'améliorer son milieu de vie.
<b>Economie intelligente</b>	-Difficulté d'attirer de nouvelles entreprises au centre de la ville	-Transactions en lignes  -Echanges de données informatisées.	-Attirer des entreprises et de la main d'oeuvre. -Créer des emplois durables créant de la valeur.
<b>Mobilité intelligente</b>	-Infrastructures routières vieillissantes, congestion du réseau routier, demande croissante en transport en commun.	-Systèmes de géolocalisation -Bornes de recharge électriques  -Ecrans électroniques. -Applications cellulaire. -Caméras.	-Optimiser les différents systèmes de transport en commun et transport actif, le tout coordonné efficacement par des systèmes technologiques.

<p><b>Environnement intelligent</b></p>	<p>-Consommation des ressources naturelles, gestion de l'eau, gestion des matières résiduelles</p>	<p>-Capteurs -Sensors -Smarts gids</p>	<p>-Utiliser les ressources naturelles de façon durable et protéger l'environnement - naturel. -Favoriser la planification et l'aménagement durable du territoire. -Créer un environnement sain qui favorise une vie du quartier</p>
<p><b>Habitat intelligent</b></p>	<p>-Gestion du développement urbain, qualité et accessibilité aux services offerts.</p>	<p>-Plate-forme d'échange web pour les quartiers. -Nouvelles technologies pour les habitations écologiques.</p>	<p>- crée un milieu de vie sécuritaire qui rassemble des composantes culturelles, des services de santé et d'éducation, qui offre des bâtiments de qualité et ou il y a une cohésion sociale</p>

TABLE 1.1 – Synthèse des six dimensions pour devenir une ville intelligente

### 1.3.3 Quatre risques importants pour les villes intelligentes

Malgré l'intérêt des villes intelligentes, le chercheur néerlandais Jorrit de Jong [4]; directeur académique du programme "innovation et gouvernance" à l'université de Harvard ; perçois toutefois des risques potentiels importants :

- Le premier risque est celui du piratage informatique. Plus les systèmes deviennent interconnectés, plus les risques sont élevés que des informaticiens de mauvaise volonté piratent ces systèmes, tels que l'éclairage public, les centrales énergétiques et certaines autres infrastructures. Les autorités publiques de petite taille n'ont souvent pas suffisamment d'expertise pour se

protéger contre ce genre d'événements.

- Les problèmes éthiques liés au développement des algorithmes. Ces algorithmes permettent de découvrir par exemple une structure sous-jacente dans certains phénomènes, tels que la petite criminalité. L'application d'algorithmes peut mener à une discrimination de personnes issues de certaines classes sociales ou de certaines origines ethniques.

- Les entreprises gagnent trop d'influence. Les solutions des villes intelligentes sont souvent développées en collaboration avec les entreprises informatiques. Ceci leur permet de rassembler d'énormes bases de données, qui peuvent constituer une atteinte à la vie privée.

- Les autorités publiques et les grands projets de TIC (Technologies d'Information et de Communication) ne font pas toujours bon ménage. Il est important que tous les départements concernés partagent leurs bases de données et qu'il existe suffisamment d'expertise technique au sein des organisations.

### 1.3.4 Top 5 des villes intelligentes et durables dans le monde

1. **Singapour** « *la reine des Smart Cities* » : dans une ville où 85% des habitants possèdent un Smartphone, Singapour a lancé un programme Smart nation pour répondre aux enjeux de mobilité, de gestion de l'énergie et d'innovation verte.

La ville accueille ainsi le CleanTechPark, qui regroupe des industries vertes et des bâtiments écologiques.

En termes de mobilité, l'objectif est de réduire au maximum l'usage de la voiture : gratuité des transports sur certaines tranches horaires, modulation des tarifs des péages en fonction de la circulation, véhicules autonomes et partage de véhicules électriques.

Pour sensibiliser les citoyens aux économies d'énergie, les factures d'électricité et de gaz mentionnent les consommations moyennes du quartier afin de permettre de se situer par rapport aux autres.

2. **Barcelone** « *un modèle de Smart City* » : Barcelone est une cité intelligente et avant-gardiste.

La ville a créé l'UrbanLab, un laboratoire d'expérimentation à destination des entreprises, leur permettant de tester un projet sur le terrain avec l'appui des services de la ville.

Très connectée, la ville s'est dotée de capteurs pour gérer les éclairages, les espaces verts et les feux de signalisation. Outre la collecte de déchets intelligente afin d'optimiser les déplacements, la mobilité et la circulation occupe le devant de la scène des préoccupations.

La ville a été sollicitée par la Banque mondiale pour développer un standard de "Smart City" s'inspirant de son modèle déjà bien avancé.

3. **Londres** « *Smart City en CO-construction* » : avec l'augmentation croissante de sa population, la ville de Londres est impliquée depuis plusieurs années dans une démarche "intelligente" et durable.

Pour impliquer les citoyens, la ville a mis en place une plate-forme Talk London qui regroupe toutes les informations utiles pour vivre, travailler et circuler à Londres. Le site permet aussi de recueillir les avis et les opinions des habitants grâce à des enquêtes et des sondages. Concernant la mobilité, Londres a été précurseur en instaurant dès 2003 un péage pour les automobiles.

La ville a également développé un système d'incitation à la marche : les piétons sont accompagnés tout au long de leur parcours grâce à des bornes interactives.

4. **San Fransisco** « *entre open data et grid renouvelable* » : dès 2009, la mairie a mis à disposition des citoyens et des acteurs privés des données municipales en libre-accès. Cette politique de transparence a permis la création de pas moins de 60 applications, conçues pour faciliter la vie des habitants et les informer dans des domaines tels que la santé, la mobilité et la biodiversité. Déterminée à être une des villes les plus smart du monde, San Francisco s'est doté en 2012 d'un Chief Innovation Officer pour favoriser l'émergence de solutions connectées.

Très engagée dans le domaine de développement durable, et notamment le recyclage des déchets, la ville s'est également engagée à ce que la totalité des services municipaux passent à l'énergie 100% renouvelable.

5. **Oslo** « *la capitale Norvégienne connectée* » : à Oslo, l'accent est mis sur l'éclairage intelligent : 10 000 lampadaires ont été équipés de capteurs permettant d'ajuster la luminosité en fonction des saisons et des besoins en éclairage. L'objectif étant de réduire les consommations d'électricité de 70%.

L'exemple norvégien a inspiré d'autres villes d'Europe qui ont adopté le projet e-street : 11 pays s'engagent pour réduire leurs consommations électriques grâce aux lampadaires connectés.

### 1.3.5 Application sur les villes intelligentes

Les applications des smart cities sont des applications qui permettent la connexion entre les habitants et la ville dans laquelle ils résident, afin de leur faciliter et d'améliorer les conditions de vie. Ci-après quelques exemples d'application existants :

#### 1.3.5.1 Parking intelligent

Est une application qui permet en général de gérer les places d'un parking de façon optimal. Les utilisateurs savent le nombre de places disponibles sans accéder à ce dernier et sans perte de temps.

#### 1.3.5.2 Eclairage intelligent

Est de plus en plus utilisé que ce soit à la maison ou à l'extérieur, il permet d'économiser de l'énergie et faciliter l'installation. Il fonctionne de plusieurs façons différentes soit par la détection de présence grâce aux capteurs, soit grâce à une application mobile.

#### 1.3.5.3 Détection des feux de forêt

L'Algérie a vécu cet été une horrible période de feux de forêt, cette application permettra d'anticiper leur déclenchements. Et cela grâce à la surveillance constante des zones en danger et lancera des alertes dès le déclenchement de quelconque incendie.

#### 1.3.5.4 Pollution de l'air

Est une application qui permet d'analyser et de détecter la pollution d'air aux niveaux des villes urbaines. De base, cela se fait grâce à des stations de détection connectées mais avec l'évolution de la technologie, ils utilisent des réseaux de capteurs malgré leur imprécision par rapport aux stations.

#### 1.3.5.5 Route intelligente

Des autoroutes intelligentes avec des messages d'avertissement et des déviations en fonction des conditions climatiques et des événements inattendus comme les accidents ou les embouteillages.

## 1.4 Internet des objets (*IdO*)

### 1.4.1 Définition d'objet connecté (*OC*)

Avant de définir les concepts d'IdO, il est important de définir l'objet connecté. C'est un dispositif dont la finalité première n'est pas d'être un système informatique ni une interface d'accès au web. Par exemple, un objet tel qu'une machine à café ou une serrure était conçue sans intégration de systèmes informatiques ni connexion à Internet. L'intégration d'une connexion Internet à un objet connecté (OC) permet de l'enrichir en terme de fonctionnalités, d'interaction avec son environnement, il devient un OC Enrichi (OCE). Par exemple, l'intégration d'une connexion Internet à la machine à café la rendant accessible à distance. Un OC peut interagir avec le monde physique de manière indépendante sans intervention humaine [44]. Il possède plusieurs contraintes telles que la mémoire, la bande passante ou la consommation d'énergie, etc. Il doit être adopté à un usage, il a une certaine forme d'intelligence, une capacité de recevoir, de transmettre des données avec des logiciels grâce aux capteurs embarqués. Un objet connecté a une valeur lorsqu'il est connecté à d'autres objets et briques logicielles. Par exemple : une montre connectée n'a d'intérêt qu'au sein d'un écosystème orienté santé bien-être, qui va bien au-delà de connaître l'heure. Un OC a trois éléments clés :

1. Les données produites ou reçues, stockées ou transmises.
2. Les algorithmes pour traiter ces données.
3. L'écosystème dans lequel il va réagir et s'intégrer.

### 1.4.2 Définition d'Internet des Objets (*IdO*)

L'Internet des Objets ou IdO (the Internet of Things ou IoT en anglais) est l'extension de la connectivité vers les objets physiques comme (serveur, capteur, etc.) à travers l'Internet. Ceci permet à ces objets de communiquer et même permet de surveiller et commander ces objets à travers l'Internet. L'IdO est considéré comme la troisième évolution de l'Internet. L'IdO a plusieurs définitions. L'une de ces définitions est celle donnée par l'UIT (Union Internationale de Télécommunication) où elle décrit l'IdO comme infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des

objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution [8].

### 1.4.3 Domaines d'application

L'IdO couvrira un large éventail d'applications (comme illustré dans la Figure 1.3 de [33]) et touchera quasiment à tous les domaines que nous affrontons au quotidien. Ceci permettra l'émergence d'espaces intelligents autour d'une informatique omniprésente. Parmi ces espaces intelligents, on peut citer [2] :

— **L'énergie** : la gestion des grilles électriques se verra améliorée grâce à la télémétrie, permettant une gestion en temps réel de l'infrastructure de distribution de l'énergie. Cette interconnexion à large échelle facilitera la maintenance et le contrôle de la consommation et la détection des fraudes.

— **Le transport** : dans ce domaine l'IdO appuiera les efforts actuels autour des véhicules intelligents au service de la sécurité routière et l'aide à la conduite. Cela portera sur la communication inter-véhicule et entre véhicules et infrastructure routière. L'IdO constituera ainsi un prolongement naturel des "systèmes de transport intelligents" et leurs apports en termes de sécurité routière, confort, efficacité de la gestion du trafic et économie du temps et de l'énergie.

— **Les villes** : l'IdO permettra une meilleure gestion des réseaux divers qui alimentent nos villes (eaux, électricité, gaz, etc.) en permettant un contrôle continu en temps réel et précis. Des capteurs peuvent être utilisés pour améliorer la gestion des parkings et du trafic urbain et diminuer les embouteillages et les émissions en CO<sub>2</sub>.

— **L'agriculture** : dans ce domaine, des réseaux de capteurs inter-connectés à l'IdO peuvent être utilisés pour la supervision de l'environnement de culture. Ceci permettra une meilleure aide à la décision en agriculture, notamment pour optimiser l'eau d'irrigation, l'usage des intrants, et la planification de travaux agricoles. Ces réseaux peuvent être aussi utilisés pour lutter contre la pollution de l'air, du sol et des eaux et améliorer la qualité de l'environnement en général.

— **La santé** : la normalisation de l'Internet des Objets dans le domaine de la Santé va permettre de créer de nouveaux modèles de fonctionnement qui augmenteront la productivité des employés, mais aussi la collaboration entre soignants ainsi que la communication avec les patients.

Les objets connectés sont utilisés au quotidien pour :

- la surveillance au sein des établissements médicaux et la maintenance
- les opérations chirurgicales et le contrôle à distance,
- les services de géo-localisation.

— **L'industrie connectée** : dans le cadre des problématiques rencontrées dans le domaine industriel, l'usage des objets connectés est très spécifique et répond à des besoins :

- d'optimisation (chaîne logistique)
- de transformation des processus d'entreprise
- d'amélioration de l'efficacité et de la productivité,
- de traçabilité et de sécurité.

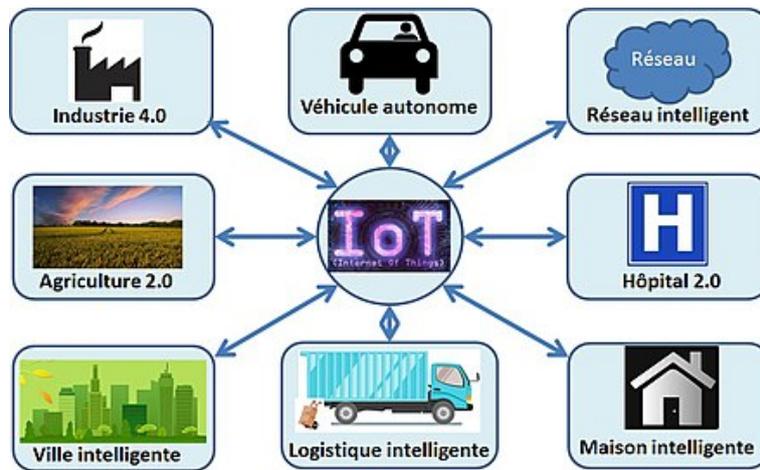


FIGURE 1.3 – Domaines d'application de l'IdO

#### 1.4.4 Evolution de l'écosystème de l'Internet des Objets

Les premiers objets connectés n'apparaissent que dans les années 1990. Il s'agit de grille-pain, machines à café ou autres objets du quotidien. En 2000, le fabricant coréen LG est le premier industriel à parler sérieusement d'un appareil électroménager relié à Internet. Les années 2000 verront les premières expérimentations d'appareils connectés à Internet. Ils l'utilisent notamment pour consulter des informations de manière automatique [3].

En 2003, la population mondiale s'élevait à environ 6,3 milliards d'individus et 500 millions d'appareils étaient connectés à internet. Le résultat de la division du nombre d'appareils par la population mondiale (0,08) montre qu'il y avait moins d'appareil connecté par personne. Selon la définition de Cisco IBSG, l'IdO n'existait pas encore en 2003 car le nombre d'objets connectés était faible.

En raison de l'explosion des Smartphones et des tablettes, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 (1,84 pour être exact) pour la première fois de l'histoire. En affinant ces chiffres, Cisco IBSG a situé l'apparition de l'IdO entre 2008 et 2009 (voir la Figure 1.4). En ce qui concerne l'avenir, Cisco IBSG estime que 50 milliards d'appareils seront connectés à Internet d'ici 2020. Il est important de noter que ces estimations ne tiennent pas compte des progrès rapides d'Internet ni des avancées technologiques, mais reposent sur les faits avérés à l'heure actuelle [40]

#### 1.4.5 Architecture de l'Internet des Objets

Vu le développement rapide de l'IdO, il est devenu nécessaire d'avoir une architecture de référence qui permettrait d'uniformiser la conception des systèmes et favoriserait l'interopérabilité et la communication entre les différents écosystèmes de l'IdO. La figure 1.5 présente la chaîne de valeur IdO/M2M. Par exemple, un objet de marque X devra pouvoir envoyer des informations à une plate-forme Y via le réseau Z. L'interopérabilité peut être vue sous deux angles, soit "fermée" au sein de grands écosystèmes qui partagent les mêmes standards, soit "native" basée sur des standards plus globaux [44].

En mars 2015, le comité Internet Architecture Board (IAB) qui a pour objectif de veiller

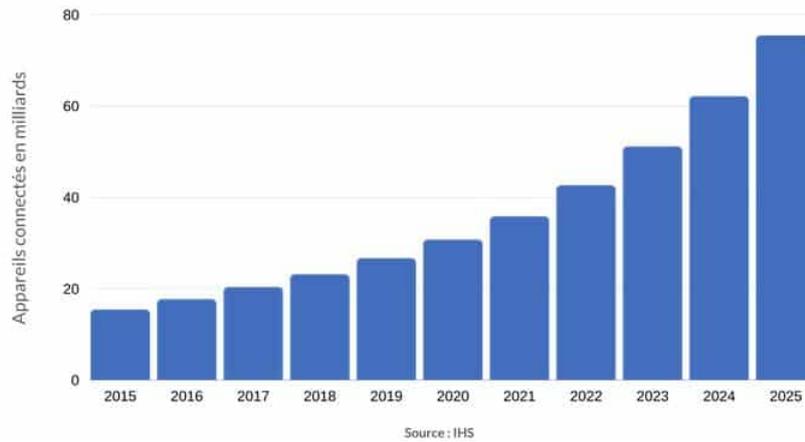


FIGURE 1.4 – Evolution d’IdO entre 2015 jusqu’à 2025.

au développement d’Internet, édite la RFC 7452 (Requests For Comments). Cette dernière est une série numérotée de documents officiels décrivant les aspects techniques d’Internet, ou de différents matériels informatiques. Il propose quatre modèles communs d’interactions entre des acteurs de l’IdO (voir [44]) :

- La communication entre objets : ce modèle est basé sur une communication sans fil entre deux objets. Les informations sont transmises grâce à l’intégration d’une technologie de communication sans-fil comme Bluetooth.

- La communication des objets vers le Cloud : dans ce modèle, les données collectées par les capteurs sont envoyées à des plate-formes de services via un réseau.

- La communication des objets vers une passerelle : ce modèle est basé sur un intermédiaire qui fait le lien entre les capteurs et les applications dans le Cloud.

- Des objets au partage des données en back-end : l’objectif de ce modèle permet le partage des données entre les fournisseurs de services. Il est basé sur le concept web programmable. Les fabricants mettent en place une API (Application Programming Interface) permettant l’exploitation des données agrégées par d’autres fabricants [45].

D’autres organismes proposent d’autres types d’architectures pour l’IdO qui privilégient les contextes des applications. L’organisme de standardisation IEEE Standards Association (IEEE-SA) a créé le groupe de travail IEEE P2413 qui prend en compte les variétés des contextes des domaines d’applications de l’IdO. IEEE s’est fixé les objectifs suivants :

- Proposer un modèle de référence qui prend en compte les relations, les interactions et les éléments d’architectures communes pour divers domaines.

- Développer une architecture de référence qui soit compatible et prendre en compte tous les domaines des applications [45].

### 1.4.6 Technologies clés génériques

L’IdO fonctionne avec le support de plusieurs technologies tels que les réseaux de capteurs sans fil, le Cloud Computing, les analyses Big Data, les protocoles de communication, les services web, etc.

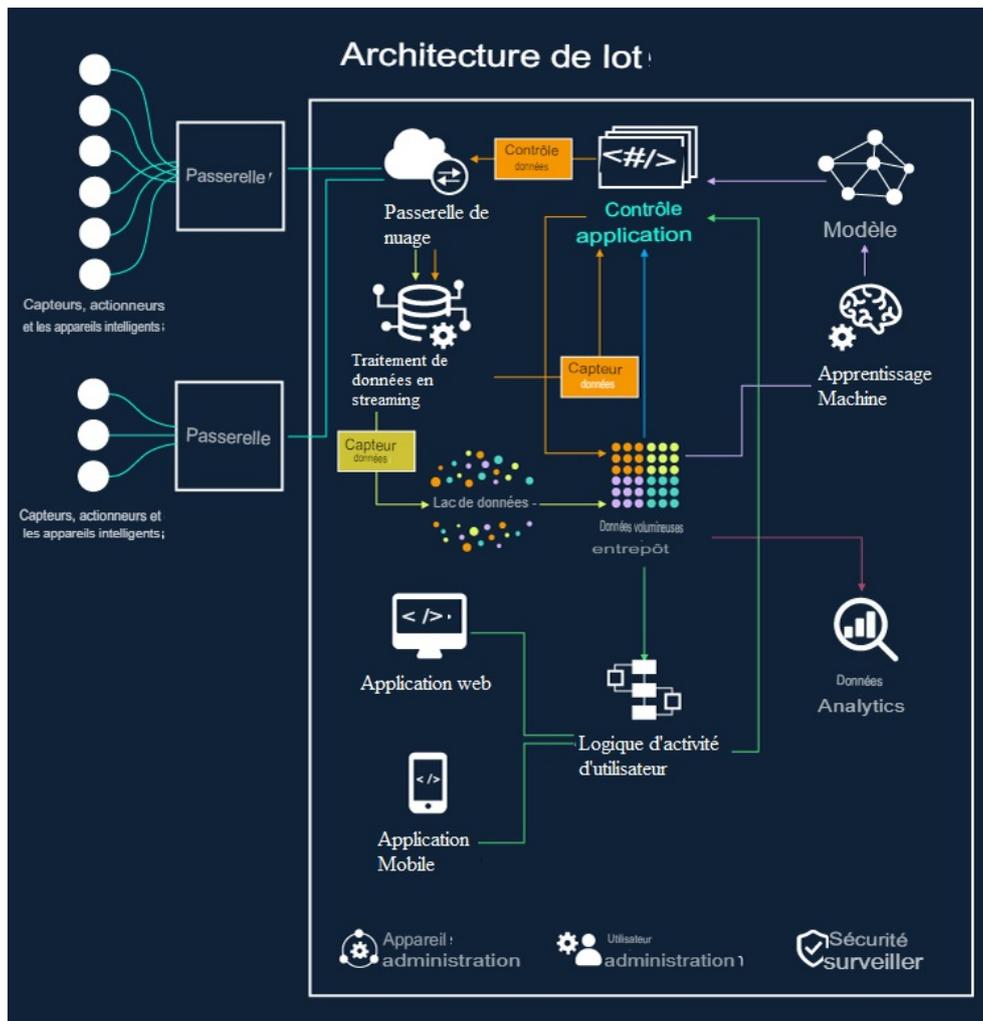


FIGURE 1.5 – Architecture d’Internet des Objets

- **Les réseaux de capteurs sans fil RCSF** : (*Wireless Sensor Network, WSN*) Un RCSF se compose d’un nombre de Noeuds Capteurs qui ont des fonctionnalités de capturer et traiter/transmettre les données.
- **Cloud Computing** : fournit un espace de stockage de données IdO et offre des services de visualisation, analyse et archivage des données.
- **Big Data** : offre des outils d’analyse avancées pour les données massives collectées par les objets IdO selon leurs caractéristiques : volume, vitesse, variabilité (forme de données : texte, audio, vidéo, image).
- **Les protocoles de communication** : sont indispensables pour assurer la connectivité entre objets et applications. Les protocoles de communication définissent le format des données, la taille des paquets, adressage, routage, etc.
- **Les systèmes embarqués** : Les objets connectés sont formés essentiellement des cartes à microcontrôleur intégrant un microprocesseur, une mémoire et des ports d’E/S pour la connexion des capteurs.

## 1.5 Systèmes de transports intelligents (STI)

Le terme de "Systèmes (ou Services) de Transports Intelligents" (STI ou ITS, pour "Intelligent Transport Systems") désigne les applications des Technologies de l'Information et de la Communication (TIC) aux transports, voire à la mobilité en général. Le terme système est vague et se décline en un ensemble de moyens mis en place pour gérer au mieux les contraintes liées au trafic routier, telles que les embouteillages, la sécurité ou même la pollution. En particulier, ces systèmes offrent un caractère réactif à une infrastructure pourtant fixe, mais dont la population peut grandement varier.

Les STI collectent, stockent, traitent et distribuent de l'information relative à l'état d'infrastructures, à la progression de véhicules et au mouvement des personnes et des marchandises.

### 1.5.1 Application des STI à l'exploitation des réseaux routiers

Les principales fonctions d'exploitation du réseau liées aux STI sont les suivantes :

- la surveillance du réseau, est une des fonctions clés de la collecte d'informations courante sur le réseau routier et du soutien des autres activités d'exploitation du réseau. Par conséquent, même si la surveillance fait partie intégrante de tous les services de STI, elle ne constitue habituellement pas un service en soi.

- afin d'améliorer la viabilité et la sécurité routières, des mesures proactives et réactives peuvent être déployées. Les mesures proactives mettent la priorité sur la prévention des incidents et de la congestion, et les mesures réactives mettent la priorité sur la détection et la vérification des incidents et des conditions routières dangereuses, les interventions et le dégagement, et sur le rétablissement des conditions normales. En effet, la sécurité routière ne peut pas être assurée uniquement au moyen de mesures préventives. Il est important de surveiller constamment l'état de la route et de la circulation, et de réagir aux situations.

- l'information aux voyageurs est une mesure de prévention secondaire. La diffusion d'avertissements en temps opportun, des conditions de circulation dangereuses et de la congestion réduit le nombre d'accidents.

### 1.5.2 Contexte et applications au milieu urbain

Le champ d'application des STI en milieu urbain est très large :

- **les intersections** : les STIs agissent sur les intersections en se chargeant d'appliquer une stratégie de changement des feux de circulation. Cette gestion des feux va représenter un aspect essentiel de la fluidité du trafic routier dans une ville.

- **le stationnement** : la gestion du stationnement est également primordiale et a une incidence directe sur la fluidité du trafic. Il paraît logique de dire que l'utilisation de la voiture en milieu urbain repose en partie sur le fait de savoir s'il y a une place disponible ou non sur le lieu d'arrivée.

Il faut savoir qu'en moyenne, en France par exemple, 10% des véhicules en circulation à un instant donné cherchent une place où se garer. Les STI vont aider à prendre des décisions, mais également informer les utilisateurs ou encore contrôler les véhicules.

Par exemple, des détecteurs peuvent être utilisés afin de détecter la présence d'un véhicule

sur une place, et calculer sa durée de stationnement. Ceci a été constaté dans la ville d'Amiens via des stationnements "minute" : une borne est associée à une place, et dès lors qu'un véhicule s'y gare, un compte à rebours se déclenche pour une durée déterminée. Si cette durée est dépassée, les agents de la voie publique sont automatiquement prévenus. Nous pouvons également citer l'utilisation de panneaux à messages variables (PMV) pour les parkings, systèmes très répandus dans les grandes métropoles qui indiquent le nombre de places disponibles (ceci n'utilisant pas nécessairement des détecteurs, mais étant généralement calculé en fonction des entrées / sorties dans le parking en lui-même).

- **la sécurité routière** : concernant la sécurité routière, les STI ont une grosse carte à jouer, principalement autour de deux catégories. D'une part, les systèmes hors-véhicules tels que les PMV vont permettre d'avertir l'utilisateur en cas de danger : vitesse d'un utilisateur trop élevée, conditions météo inadaptées, et travaux. Le but ici est d'influencer l'utilisateur. D'autres part, les systèmes sur-véhicule sont également nombreux et très développés : détection de piétons ou obstacles, capots intelligents capables de se soulever en cas d'impact avec un piéton, systèmes embarqués (exemple : appel des secours automatique en cas de choc), régulateur de vitesse, vision nocturne (à l'aide de caméras), etc.

- **les ronds-points** : certains travaux abordent la gestion des ronds-points, en se basant sur les priorités existantes. Dans de nombreux pays, il est, par exemple, commun de donner la priorité aux usagers étant déjà à l'intérieur d'un rond-point. Ce qui en ressort apparaît comme une évidence : soit deux flots, chacun étant sur une voie respectivement au cœur du rond-point et sur les côtés, la priorité doit être donnée au flot de la plus grande importance afin d'obtenir une gestion optimale du trafic. Les feux quant à eux améliorent grandement la fluidité du trafic, mais uniquement en cas de débit suffisamment élevé.

## 1.6 Réseau véhiculaire *VANETS*

Un réseau véhiculaire mobile s'agit d'un réseau de type ad-hoc. Il existe deux types de réseau Ad Hoc : les MANETs (Mobile Ad Hoc Networks) et les VANETs (Vehicular Ad Hoc Networks) [43].

Les VANETs [?] est une nouvelle technologie émergente. C'est des réseaux dans lesquels les nœuds mobiles sont des véhicules intelligents, équipés de matériels à très hautes technologies (Calculateurs, radars, systèmes de géolocalisation (GPS), différents types de capteurs et périphériques réseau). Les réseaux VANETs permettent divers types de communications : des communications inter-véhiculaires (V2V) et véhicules à infrastructure (V2I) ou Infrastructure à véhicule (I2V). Ainsi les réseaux VANETS ne sont pas des réseaux purement Ad Hoc suite à la communication avec des infrastructures qui peuvent donner accès à internet. Les infrastructures employées dans ces réseaux sont des «Unités de Bords de Routes», de l'anglais RoadsideUnits (RSU). Les multiples nœuds échangent tous types d'alertes ou informations utiles pour améliorer non seulement la sécurité de la circulation routière, mais aussi l'état du trafic. En plus de cela ils ont la capacité d'échanger des données tels que : musique, vidéo, publicités...

Les réseaux VANETs se basent sur deux types d'applications. Les applications qui constituent le noyau d'un système de transport intelligent ITS (Intelligent system), pour assurer l'amélioration de la sécurité routière, et des Transport applications déployées pour le confort des passagers.

## 1.7 Internet des véhicules (*IdV*)

L'internet des véhicules (IdV) [34] figure parmi les technologies exploitées dans les villes intelligentes, celle-ci est une sous-section de l'internet des objets. Les VANETs sont en cours de changement vers IdV, ceci suite au fait que ces réseaux ne couvrent pas globalement tous les besoins des utilisateurs.

IdV est un réseau distribué qui supporte l'utilisation des données créées par les voitures connectées et les réseaux ad hoc de véhicules (VANET). L'objectif le plus important de l'IdV est de permettre aux véhicules de communiquer en temps réel avec leurs conducteurs, les piétons, les autres véhicules, les infrastructures routières et les systèmes de gestion de flotte. L'IdV prend en charge cinq types de communication en réseau :

- **Systemes intra-véhicule**, qui s'en charge de la surveillance de l'état interne des véhicules grâce aux unités embarquées (OBU).

- **Systemes de véhicule à véhicule (V2V)** , qui prennent en charge l'échange sans fil d'informations sur la vitesse et la position des véhicules environnants.

- **Systemes de véhicule à infrastructure (V2I)**, qui prennent en charge l'échange sans fil d'informations entre un véhicule et des unités de soutien en bord de route (RSU).

- **Systemes "véhicule au Cloud" (V2C)**, qui permettent au véhicule d'accéder à des informations supplémentaires sur Internet par le biais d'interfaces de programmes d'application (API).

- **Systemes "véhicule-piéton" (V2P)**, qui permettent de sensibiliser les usagers vulnérables de la route (VRU), tels que les piétons et les cyclistes.

Lorsqu'ils sont mentionnés dans des systèmes de transport intelligents (STI), les cinq types de réseaux ci-dessus sont parfois appelés communication véhicule à tout (V2X).

## 1.8 Avantages d'IdO aux villes intelligentes

L'Internet des Objets offre de nombreux avantages et applications aux villes intelligentes, notamment [29] :

- Amélioration des services publics traditionnels tels que les transports, la circulation et le stationnement.

- Contrôle et maintenance des espaces publics.

- Suivre la validité des bâtiments et des établissements de travail.

- Réduire le temps perdu dans les transactions administratives dans la ville.

- Fournir la consommation d'énergie de la ville.

- Eclairage intelligent pour la ville.

## 1.9 Conclusion

Dans ce chapitre, nous avons survolé d'une façon générale des STI, les VANETs, la technologie Internet des Objets, internet des véhicules et les villes intelligentes. Nous avons présenté les STI et leur applications et défini les VANETs. Ensuite, nous avons défini c'est quoi l'IdO, son domaine d'application et son évolution. Après, nous avons présenté son architecture et ses

composants. Par la suite nous avons défini les villes intelligentes, donné quelques composantes et donné top 5 villes intelligentes. . Enfin, nous avons terminé par quelques usages clés des technologies IdO pour la ville intelligente et les avantages d'IdO aux villes intelligentes. Dans le deuxième chapitre, nous allons présenter un état de l'art de la gestion du trafic.

# Chapitre 2

## Etat de l'art sur la gestion du trafic

### 2.1 Introduction

Ces dernières années le trafic urbain rencontre plusieurs difficultés et des problèmes innombrables journaliers, ce qui requiert une solution qui peut les contrer ou plus au moins les diminuer. L'augmentation constante non seulement des conducteurs mais aussi des véhicules dans les villes est lui-même un obstacle et n'arrange pas la situation mais rajoute d'autres complications à l'équation. Par exemple : la pollution, les accidents et les embouteillages. Parmi les simples situations journalières que rencontre toute personne dans les grandes villes et même parfois dans les autoroutes, nous citons une personne qui souhaite rejoindre son domicile à une certaine heure peut se retrouver au milieu d'embouteillage pendant des heures, ou en cas d'urgence une personne qui ne connaît pas les routes ne pourra pas s'en sortir. Les gestionnaires de trafic sont surchargés et n'arrivent plus à régler la situation avec de simple panneau de régulation même les policiers ne savent plus comment la gérer. C'est ainsi que les chercheurs ont intervenu en proposant des solutions de gestion de trafic intelligent en ajoutant le guidage routier à leurs études.

### 2.2 Définition de gestion du trafic

La gestion du trafic en général est une manière de régulation utilisée dans le but de mettre fin aux différents problèmes que la circulation routière rencontre. Plusieurs et diverses méthodes ont été employées, mais ces dernières années avec le développement majeur dans les divers domaines, celles-ci ne sont plus à jours et nécessitent de nouvelles résolutions. C'est ainsi que le trafic intelligent est introduit et est considérée comme « LA » solution aux problèmes.

La gestion de trafic intelligent se résume à des systèmes de navigation qui incluent l'utilisation des GPS, contiennent plusieurs fonctionnalités et contribuent à l'amélioration de la qualité de vie. En d'autres termes, ces systèmes permettent aux usagers d'avoir le meilleur chemin avec un guidage routier tout au long, réduisent les accidents en exposant les différents changements et grâce aux fonctionnalités des réponses aux besoins spécifiques de l'utilisateur. Donc, en général, c'est le contrôle de trafic en exploitant les smartphones et tous autres outils numériques.

## 2.3 Terminologie de gestion du trafic

- **Un flux de véhicules** : est l'ensemble des véhicules entrant par une voie donnée et ressortant par une autre. Le trafic dans une intersection est constitué d'un ensemble de flux de véhicules, chacun provenant d'une source. Deux flux sont cohérents s'ils peuvent évacuer l'intersection simultanément.

- **Un cycle d'un feu** : représente la durée qui sépare deux phases identiques de l'intersection. Il est défini par une séquence de phases [48].

- **Une phase d'un feu** : est une période durant laquelle un ou plusieurs flux cohérents sont admis dans le carrefour [48].

- **Un carrefour** : est défini comme étant une intersection de plusieurs rues où différents flux de véhicules doivent circuler de manière ordonnée. Un ensemble de carrefours constitue un réseau, chaque carrefour possède un cycle [35].

## 2.4 Pourquoi évaluer la technologie du trafic intelligent ?

Les budgets des infrastructures publiques sont toujours serrés, et la construction de routes et de ponts est toujours coûteuse. Les systèmes intelligents de gestion du trafic aident les services de transport municipaux et régionaux à faire face à la situation rapidement et de manière rentable. L'intégration d'une technologie intelligente de gestion du trafic leur permet d'obtenir, à un coût raisonnable, de meilleures performances de leurs infrastructures existantes.

## 2.5 Nouvelles technologies derrière la gestion intelligente du trafic

La connectivité des terminaux nécessite du matériel et des logiciels sophistiqués. Ainsi plusieurs entreprises offrent une connectivité sécurisée, évolutive et à haut débit pour répondre à l'ensemble des exigences de gestion du trafic. Conçus pour fonctionner de manière optimale, indépendamment de l'humidité, des vibrations et des températures extrêmes, les routeurs industriels et de transit sont installés dans les villes intelligentes du monde entier.

Voici quelques-unes des principales fonctionnalités que les villes obtiennent avec ces systèmes :

- **Détection des embouteillages** : Grâce aux caméras et aux capteurs qui surveillent en permanence les intersections, les techniciens peuvent surveiller toute la ville depuis le centre de gestion du trafic de la ville.

- **Contrôle adaptatif** : La détection des encombrements permet également un contrôle adaptatif, qui entraîne des ajustements dynamiques des systèmes, notamment des feux de circulation, de la signalisation des bretelles d'accès et des voies de transit rapide par bus.

- **Véhicule connecté** : Cette technologie en plein essor permet aux véhicules de communiquer directement avec les intersections. La gestion intelligente du trafic peut inclure une unité de bord de route de véhicule connecté à cette fin.

- **Acheminement des secours** : Une application essentielle du système de gestion intelligente du trafic est la possibilité d'accorder un accès prioritaire aux services de police,

d'incendie et d'ambulance.

Toutes ces fonctions nécessitent une surveillance, une détection, un transfert de données fiable à haut débit et une automatisation. Des routeurs cellulaires spécialement conçus peuvent fournir l'épine dorsale de communication pour gérer tous ces systèmes.

## 2.6 Systèmes existants de la Gestion du trafic

Certain modèles de gestion de trafic ont été commercialisés. Ceux-ci utilisent soit des capteurs qui déterminent la position de véhicules en temps réel, soit une méthode qui permet de simuler le déplacement des véhicules ou bien les deux.

Les avancées ont été permises grâce à l'introduction de plusieurs solutions novatrices en termes de gestion du trafic routier : au total, ce sont plus d'une vingtaine de projets qui sont nés durant ces trente dernières années. Dans cette partie, nous allons essentiellement parler de deux systèmes qui ne sont pas nécessairement les plus performants, mais représentent les deux principaux systèmes de gestion du trafic routier utilisés dans le monde :

### 2.6.1 SCATS

SCATS (Sydney Coordinated Adaptive Traffic System) [1], qui a été à l'origine développé pour Sydney et d'autres villes Australiennes. Il est pour sa part entièrement adaptatif et utilise une notion d'hierarchie (ce qui forme une certaine distribution sur le réseau). Entre le recueil des données sur le terrain et le centre de contrôle, des contrôleurs intermédiaires sont insérés, permettant d'alléger la charge globale du système et d'avoir un contrôle découpé en plusieurs zones, l'ensemble des acteurs utilisant des communications synchronisées. De manière similaire à SCOOT, ce système ajuste le temps des cycles et autres paramètres en fonction des données recueillies afin de diminuer le délai et les arrêts, mais n'utilise pas la même stratégie. Les valeurs recueillies permettent la sélection de plans de feux parmi une large librairie, sur lesquels le système va se baser pour proposer des plans adaptés. De plus, contrairement à SCOOT, les détecteurs sont uniquement placés au niveau des feux de circulation. Cette stratégie s'appuie sur des bibliothèques séparées de durées de cycle, de décalages et de durées de vert et sur un algorithme temps réel de reconstitution du plan de feux. Le plan de feu est ainsi reconstitué et non stocké tel quel dans une bibliothèque (Figure 2.1). Ce système de régulation ne comprend pas de module d'écoulement du trafic : son fonctionnement ne repose que sur la disponibilité de données explicites décrivant le trafic. L'objectif général est de minimiser les retards et les arrêts par choix des paramètres de base du système de régulation du trafic, tels que la durée de verts, les décalages et la durée de cycle. La régulation se décompose en deux niveaux ; une régulation stratégique sur des ensembles de carrefours et une régulation tactique au niveau de chaque carrefour

### 2.6.2 SCOOT

SCOOT (Split Cycle Offset Optimisation Technique)[39], un système de contrôle à la fois réactif et adaptatif et entièrement centralisé, développé par le TRL (Trafic research laboratory, un centre de recherche anglais sur les transports). Dans ce système, l'ensemble des informations

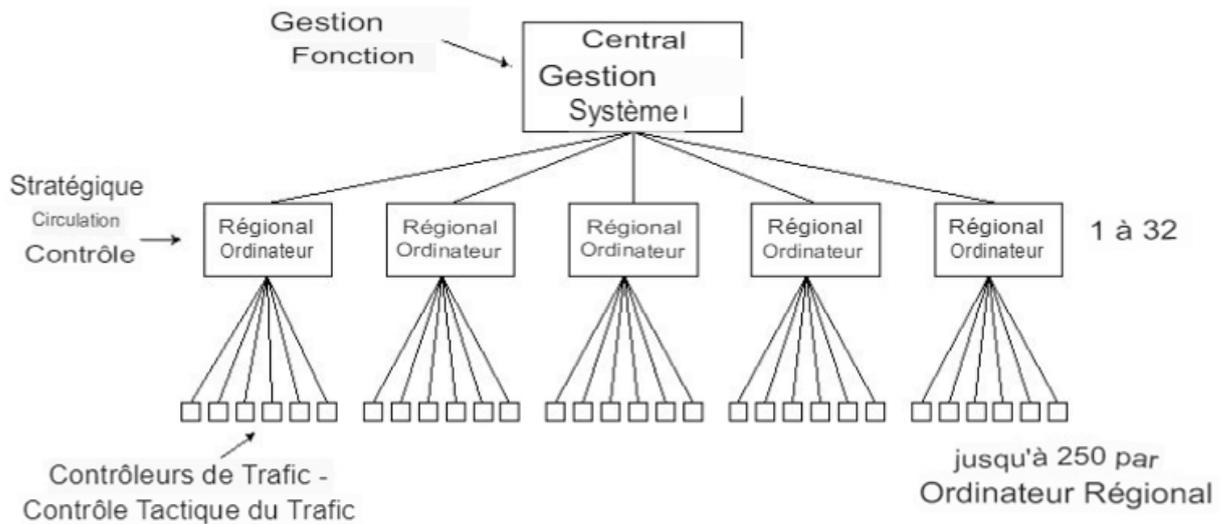


FIGURE 2.1 – Architecture SCATS

recueillies sur le terrain vont à un centre de gestion, qui s'occupe de traiter les informations et renvoyer des indications directement aux intersections. Les véhicules sont détectés par des dispositifs pouvant être placés à divers endroits sur les voies : au niveau des feux ou à une certaine distance afin de pouvoir mesurer le débit du trafic. SCOOT mesure en permanence le volume de véhicules de chaque côté de l'intersection et change la durée des phases en fonction d'un indice de performance, calculé par rapport au délai d'attente moyen, de la longueur des files d'attente et des arrêts sur le réseau. Ainsi, de la même manière que TRANSYT, système sur lequel il est basé, SCOOT génère des plans de feux en fonction de la demande des utilisateurs (côté adaptatif). En plus de cela, le système utilise des informations en ligne issues de bases de données (historique ou autres, côté réactif).

## 2.7 Guidage routier

Ces quinze dernières années, les villes urbaines subissent l'augmentation des nombres de véhicules sur leurs routes ainsi que le nombre des conducteurs causant de la sorte d'énormes problèmes de circulations, et plusieurs accidents. Pour essayer de mettre fin aux différents problèmes, les chercheurs ont opté pour l'utilisation de la technologie et l'internet. Créant ainsi un nouveau domaine de recherche connu sous le nom de « Guidage routier ». Dans ce nouveau domaine de recherche, l'objectif est de trouver une solution qui peut mettre fin aux divers problèmes routiers et d'aider les conducteurs ainsi que toute autre usagers à calculer le meilleur chemin à emprunter, en rajoutant la possibilité d'optimiser cette dernière suivant leurs besoins. Tout ceci avec des résultats dynamique et avec des changements à temps réels.

### 2.7.1 Définition

De nos jours, l'utilisation de la technologie est inévitable, suite à cela, les applications de guidage routier sont de plus en plus sollicitées par les conducteurs et même toute personne ayant envie de se déplacer.

Guidage routier du terme anglais « routing decision » est un sujet de recherche sur lequel énormément de chercheurs se sont focalisés ces dernières années. Le guidage routier en général représente des applications qui ont pour but de mettre fin à la congestion et aide à la gestion du trafic routier, il consiste exactement à diriger ou guider l'utilisateur vers la destination qu'il souhaite en lui choisissant un chemin qui répond à ses divers besoins.

Les applications les plus connues sur ce domaine de recherche sont : Google Maps, Waze, HERE WeGo et Mappy, etc. Ces applications englobent des critères qui peuvent être ajoutés suivant le public visé. Par exemple : Les lieux touristiques, la distance, le temps, les chemins qui contiennent des aires de repos, la température, paysage panoramique, etc. En général, l'application ou le service web peut détecter la localisation et c'est à l'utilisateur de choisir la destination ainsi que l'optimisation qu'il souhaite apporter, c'est à ce moment-là que le meilleur chemin s'affiche à l'écran et le guidage routier commence.

### 2.7.2 Applications

Il existe plusieurs applications reliées au guidage routier, et elles ne sont pas seulement reliées aux moyens de transport (voiture, camion, bus, train et moto) mais aussi piétons et cyclistes. C'est pour cela, nous allons en citer quelques unes des plus utilisées :

**a) Google Maps :** Est un service mobile et web à la fois mis en œuvre par l'entreprise Google lancé en 2004 qui fonctionne grâce au GPS, avec plusieurs fonctionnalités. La principale s'agit de permettre à l'utilisateur de saisir la destination qu'il veut atteindre de sa localisation ou d'une source quelconque, tout en exposant les changements dynamiques qui se produisent au cours du trajet.

**b) Waze :** Est une application pour les smartphones et tout comme Google Maps fonctionne avec l'utilisation du GPS. Elle propose un service de navigation aux usagers et affiche les changements à temps réel qui s'y produisent.

**c) HERE WeGO :** Il s'agit d'une application de navigation, qui peut être utilisée même en étant hors ligne. Cette dernière est même installée comme GPS sur certaines voitures. Elle aussi a plusieurs paramètres intéressants tels que les changements à temps réels, partage de la localisation et les données de transport publics.

**d) Mappy :** Représente une application mobile qui permet aux utilisateurs d'avoir des informations à propos du trafic à temps réels. Elle contient des fonctionnalités bien précises mais comme toutes autres applications de guidage son objectif est d'apporter de l'aide vis-à-vis de la navigation.

### 2.7.3 Critères d'optimisation

Les critères d'optimisation sont des fonctionnalités ajoutées aux applications de guidage routier qui permettent à l'utilisateur de régler l'affichage du meilleur chemin qui correspond à son besoin spécifique. Ceci est possible suite à la fonction d'optimisation qui prend comme

entrée des variables déterminées. Ces dernières varient d'une application à une autre et cela est en relation avec le public visé. Par exemple, si le public visé sont des piétons et des cyclistes une solution considérant les chemins les moins pollués peut être proposée.

## 2.8 Congestion urbaine

La congestion est citée comme un mal majeur et grandissant dans les milieux urbains. De nombreux conducteurs se retrouvent chaque jour dans la congestion et les embouteillages, bien que son importance et sa fréquence varient d'un endroit à l'autre.

Les autorités urbaines et régionales de transport s'appliquent à gérer cette problématique au moyen d'une vaste gamme de mesures. Individuellement, les conducteurs n'interprètent pas tous de la même façon le sens de la congestion et n'ont pas la même perception. Pour certains, il s'agit d'un véritable embouteillage avec des arrêts complets et de longs retards. Pour d'autres, il s'agit plutôt d'une situation où l'on avance lentement, ou plus généralement, où on se déplace à une vitesse inférieure à la limite permise.

### 2.8.1 Définition de la congestion

La définition la plus simple et la plus générale consiste à dire que la congestion existe lorsque la demande de circulation excède la capacité routière. Il n'y a pas d'indicateur ni d'approche standard permettant de déterminer l'étendue de la congestion. Dans la zone métropolitaine, la situation est définie comme congestion lorsque la vitesse moyenne est inférieure de plus de 50% à la vitesse en situation d'écoulement libre.

### 2.8.2 Nature et causes de la congestion routière dans les zones urbaines

Lorsque l'on traite de la congestion, il est important de faire la distinction entre les deux principaux types de congestion : la congestion récurrente et la congestion incidente.

#### 2.8.2.1 Congestion récurrente

La congestion récurrente se produit principalement lorsqu'un trop grand nombre de véhicules utilisent le réseau routier en même temps. Typiquement, la congestion récurrente a lieu les jours de la semaine, le matin et l'après-midi, lorsque la vaste majorité des gens se rendent au travail ou à l'école et reviennent à la maison aux mêmes heures. Dans les grandes zones urbaines, la période de pointe s'observe généralement entre 6h et 9h30 le matin, et entre 15h30 et 19H en après-midi. Dans les zones urbaines plus petites, la période de pointe s'étend normalement sur une période plus courte (une heure ou deux).

La hausse de la congestion récurrente qui a lieu pendant les périodes hors pointe est une question qui suscite de l'intérêt. Cette congestion est attribuable à une croissance rapide de la demande de déplacements pendant les heures hors pointe.

### 2.8.2.2 Congestion incidente

La congestion incidente est associée à des conditions aléatoires ou spéciales et à des événements particuliers. Les quatre causes principales de la congestion incidente sont :

- Les incidents de la circulation (les pannes, les accidents graves).
- Les chantiers de construction.
- Les conditions météorologiques.
- Les événements spéciaux.

En raison du caractère aléatoire de ce type de congestion, elle est plus difficile à prévoir, à quantifier et à gérer. Les répercussions de la congestion incidente sont significatives. En effet, la fiabilité et la prévisibilité des temps de déplacement revêtent une très grande importance pour le public, les industries qui ont recours au transport de marchandises qui doivent respecter les délais de livraisons. Le manque de données sur les incidents et sur leur impact sur les temps de déplacement a mené certains organismes à recourir à des estimations provisoires de la congestion incidente.

### 2.8.3 Évaluation de la congestion

La congestion a de nombreuses conséquences, à la fois pour les usagers des installations de transport et pour le public. On peut distinguer quatre catégories de conséquences :

- Conséquences économiques.
- Conséquences environnementales.
- Conséquences sur la sécurité routière.
- Conséquences sociales et structurelles.

La congestion de circulation peut avoir à la fois des conséquences positives et négatives sur la sécurité routière. D'une part, les vitesses réduites en conditions de congestion peuvent réduire le nombre d'accidents dans la file. D'autre part, la circulation stop-and-go risque d'augmenter le nombre d'accidents d'embouteillage arrière dans la file. En outre, l'extrémité de la file est un risque potentiel pour les véhicules en approche, en particulier sur les autoroutes où les vitesses sont élevées.

## 2.9 Algorithmes de plus court chemin dans le domaine du trafic

Les algorithmes de plus court chemin sont énormément utilisés dans le domaine du transport spécialement pour deux applications du transport public :

-Analyse des performances des réseaux de transport. Une phase de modélisation du réseau et d'analyse de performances doit être établie avant toute conception de réseau de transport. Cette analyse permet de faire un choix sur l'emplacement des lignes de transport, des arrêts et des stations.

-Aide au déplacement et planification des itinéraires dans les systèmes d'information avancés ATIS (Advanced Traveler Information System). Ceci revient à implémenter des algorithmes de plus court chemin sur les systèmes d'information d'aide au déplacement SIAD qui permettent de répondre à des requêtes locales d'itinéraires provenant de différents utilisateurs. Le problème

de plus court chemin consiste à trouver le chemin qui minimise le coût global et ce en partant d'un nœud de départ  $x$  vers un nœud d'arrivée  $y$  dans un graphe

$$G = (N, E); x, y \in N \quad (2.1)$$

tel que les poids des arcs représentent le coût. Dans la littérature, cette problématique a fait l'objet de nombreuses recherches. Dans ce cadre, plusieurs algorithmes ont été mis en place pour traiter les spécificités des différents types de graphes (graphes cycliques, graphes avec valeurs négatives, ...). Ces algorithmes sont souvent classés en deux catégories : les algorithmes de correction d'étiquettes et les algorithmes de fixation d'étiquettes. Il existe plusieurs algorithmes pour trouver le plus court et meilleur chemin, tel que : algorithme de Bellman-Ford-Moore, algorithme de Prim, algorithme de Kruskal et algorithme de Dijkstra,...

## 2.10 Travaux antérieurs en relation avec le guidage routier

Nous avons collecté un ensemble d'articles qui traitent le guidage routier et proposent des solutions pour assurer la sécurité et le confort des conducteurs en leur proposant des routes optimales d'une source vers une destination de leurs choix. Dans ce qui suit, nous présentons quelques uns de ces travaux avec une analyse critique pour chacun.

### 2.10.1 Architecture SD-IoV

Dans cet article [41], les auteurs proposent une architecture SDIoV (Software Defined Internet of Vehicles) en exploitant l'edge intelligent computing. Cette architecture prend en charge l'apprentissage multi-agent pour mettre en œuvre intelligent edge décision et le contrôle du flux de données dans le réseau. Les auteurs ont adopté l'apprentissage en profondeur asynchrone multi-agents parallèle pour la conception d'un algorithme nommé DLRD (pour Distributed-Learning-based vehicle Routing Decision) dans le but d'adapter la route proposée dynamiquement en ligne.

L'objectif de cette étude est de trouver un itinéraire optimal pour les véhicules en tenant compte de son impact sur le trafic dans sa globalité, d'explorer Intelligent edge computing beaucoup plus pour l'apprentissage distribué profond et la prise de décision parallèle asynchrone, et de concevoir et déployer le modèle d'apprentissage par renforcement multi-agent distribué pour réaliser une décision de routage de véhicule [41].

Le processus de prise de décision se concentre sur les conditions de circulation réelles, dans le but de réduire les embouteillages, qui est obtenu en ajustant le choix de l'itinéraire du véhicule [41].

En prenant en considération la surcharge de ressources des tâches sous l'architecture edge intelligence, une méthode d'optimisation de l'organisation collaborative des équipements définis par logiciel est conçue. Elle se concentre sur la sélection d'ensembles d'équipements et l'adoption de modes interactifs pour compléter la prise de décision d'itinéraire basée sur l'apprentissage par renforcement. Dans cette méthode, l'ensemble initial d'équipements collaboratifs et le mode d'interaction utilisé sont formés en fonction de la situation de surcharge de ressources de l'exécution de la tâche. Puis, le réseau de neurones à mémoire à long et à court terme est utilisé

pour prédire la future surcharge de ressources de la tâche, et optimiser davantage pour réduire le délai de décision de route [41].

L'objectif ultime de la décision d'itinéraire de véhicule est d'améliorer l'efficacité du trafic en réduisant la congestion nécessitant une collaboration entre les véhicules. Chaque tâche multi-agents planifie un itinéraire pour les véhicules ayant le même point de départ et le même point d'arrivée, et les véhicules qui initient la demande forment un ensemble d'agents de tâche.

La simulation de ce dernier sur VISSIM a démontré qu'il peut réussir à prendre des décisions de route en temps réel pour les véhicules et alléger les embouteillages avec les changements dynamiques de l'environnement routier [41].

### 2.10.1.1 Critique

La proposition de cette œuvre scientifique repose sur la distribution de données qui est considéré comme son point fort et faible car la vitesse de communication entre les véhicules doit être améliorée c'est ainsi que les réponses vont être plus fluides. Mais aussi, Le but ultime de cette dernière est de réduire la congestion du trafic et les auteurs n'ont pas prêté d'attention aux envies de l'utilisateur et n'ont pas admis de périphérique d'optimisation à leur solution.

### 2.10.2 Algorithme de routage comparaison

L'application des algorithmes de choix d'itinéraires tels qu'ils sont sur les réseaux routiers peut rencontrer quelques difficultés suites aux différents paramètres dynamiques non adaptés[49].

C'est pour cette raison la sélection d'un algorithme nécessite une étude approfondie. Dans cette étude les chercheurs ont fait et discuté des propositions dont : Une nouvelle approche d'amélioration des algorithmes avec de nouveaux critères d'évolution pour répondre aux besoins des divers conducteurs et leurs points de vue sur le meilleur itinéraire [49].

Quelques informations influentes dans le choix du meilleur itinéraire peuvent être considérées comme des entrées dans les algorithmes de routes tels que les informations de routes, de destination et de véhicule en plus des caractéristiques du véhicule et les prédictions du trafic. Le meilleur choix doit être mis à jour pendant le trajet suite aux différents changements [49].

Il existe des critères de comparaison entre les routes disponibles à fin de choisir le meilleur chemin correspondant à la demande du conducteur comme : la distance , le temps ,la facilité de conduite ( sécurité et confort ) , les dépenses , combiner entre deux critères , la complexité , évolutive et la qualité du meilleur chemin [49].

L'algorithme choisit doit réussir à s'adapter aux changements dynamiques de l'environnement suivant des étapes précises. Les étapes en questions sont :

1. Calculer initialement le meilleur chemin du point de localisation à la destination.
2. Recalculer le meilleur itinéraire suite aux mises à jours des conditions du trafic, et à chaque intersection les conditions du trafic sont vérifié dans une mise à jour.
3. Si la destination est atteinte fin, sinon la 2e étape en boucle jusqu'à l'atteinte de la destination.

SUMO et TRACI ont été utilisés pour évaluer et comparer entre les algorithmes. Les premiers résultats démontrent la possibilité de l'utilisation de DIJKSTRA

### **2.10.2.1 Critique**

Cet article est seulement une étude comparative entre les algorithmes qui peuvent être utilisés pour proposer le chemin le plus court.

### 2.10.3 Prédiction d'itinéraire

Cette recherche est une approche de planification des trajets en fonction de la situation qui intègre des informations en temps réel obtenue à partir des capteurs dans une ville intelligente, et combine ces données avec un modèle pour estimer les situations de trafic futures pour le calcul d'itinéraire. Le système proposé fournit trois composants : [42]

1. Une interface utilisateur web basé sur le projet populaire OpenTripPlanner
2. Un moteur backend en temps réel, basé sur le Framework des flux
3. Un modèle de trafic dynamique sophistiqué qui est intégré dans le moteur des flux de trafic à des emplacements non observés à des moments futurs

Cette recherche propose une combinaison des trois composants qui est un planificateur de trajet qui intègre les dernières informations du trafic [42]

**Thomas Liebig et al**, ont proposé un système qui réponds aux problèmes de véracité, vitesse et parcimonie de la gestion du trafic urbain, le moteur principal de ce dernier est un graphe des flux qui est défini à l'aide du langage de configuration XML des flux et propose l'intégration de composants personnalisés directement dans le graphe de flux de données [42]. Le système se compose de :

- **Modèle du trafic (Traffic Model)** : Composant clé du système, il combine deux méthodes d'apprentissage automatique d'une manière novatrice afin d'obtenir des prévisions de flux presque arbitraires [42]

1. **Champ aléatoire spatio-temporel pour la prédiction de flux** : utilisée pour modéliser la dynamique temporelle du flux de trafic mesuré par les capteurs SCATS
2. **Modèle de processus gaussien pour l'imputation de flux** : c'est une modélisation des valeurs des flux de trafic basé sur les jonctions dans un cadre de régression de processus gaussien

- **OpenTripPlanner (OTP)** : une initiative open source pour le calcul d'itinéraire

1. Le réseau du trafic est généré à partir des données d'OpenStreetMap et des horaires des transports publics, permet le calcul d'itinéraire pour plusieurs modes de transport, y compris la marche, le vélo, le transport en commun ou ses combinaisons.
2. L'algorithme de routage par défaut d'OTP utilise une heuristique de coûts pour élaguer la recherche de Dijkstra
3. OTP se compose de deux composants : une API et une Application WEB qui interface l'API à l'aide de service complet RESET
4. L'API charge le graphique du réseau de trafic, et calcule les itinéraires. L'application Web fournit une interface utilisateur interactive basée sur un navigateur avec une vue cartographique.

- **Le cadre des flux (The streams Framework)** : Le besoin de capacités en temps réel dans le traitement des données d'aujourd'hui et la diminution constante de la latence entre l'acquisition des données et l'extraction des connaissances ou l'utilisation des informations à partir de ces données ont conduit à une demande croissante d'environnements de traitement en continu à usage général. Ils ont basé leur décisions pour framework streams sur ses applications

récentes qui mettent en avant ses capacités de haut débit, et les opérateurs d'exploration de données intégrés [42]

1. **Traitements de données SCATS** : une source de donnée est représentée par un ensemble de paires clé-valeur dans le cadre des flux
  - Le flux de données graphique est formulé dans un langage basé sur XML
2. **API au niveau du service** : Le runtime des flux fournit une API de service simple basée sur RMI invocation pour les composants de flux qui fournissent des services à distance
  - Le TrafficEstimator définit une interface distante qui permet aux méthodes de service de cet estimateur d'être appelées de manière asynchrone depuis l'extérieur des données

Dans cette étude, les chercheurs ont utilisé les capteurs de SCATS dans leur proposition « Planificateur de voyage dans un cas d'utilisation à Dublin », ces derniers transmettent des informations sur la fluidité du trafic toutes les 6 minutes. Le réseau de leur expérimentation est généré sur la base de données OpenStreetMap3, dans l'étape de prétraitement le réseau est limité à une fenêtre, ensuite chaque rue est divisé à n'importe quel carrefour afin de récupérer des segments de rue, pour au final obtenir un graphique qui représente le réseau de trafic [42].

Les lectures de capteurs sont agrégées dans des intervalles de 30 minutes de temps fixe, Le graphe spatial pour le STRF est structuré comme KNN (K-nearest-neighbor) du SCATS emplacement des capteurs [42].

Les auteurs ont utilisé un graphe 7NN, car un k plus petit induit des graphes avec des grandes composantes déconnectées, et un k plus grand donne des résultats plus complexes [42].

Les données d'évaluation sont diffusées sous forme de nœuds observés dans le STRF qui calcule une nouvelle prédiction MAP conditionnée, les chercheurs ont utilisé une structure de données KDTree avec la méthode de plus proche voisin pour éviter les problèmes de précisions lors de l'appariement entre les composants [42].

Ils ont appliqué leur planificateur pour calculer les itinéraires à différents horodatages en fonction de la situation du trafic [42].

Cette recherche a été basée sur des lectures de capteurs en temps réel et une interface utilisateur. C'est une combinaison du modèle de trafic en temps réel et le calcul de trajet avec une colonne vertébrale en continu [42].

### 2.10.3.1 Critique

La proposition est basée sur la prédiction des valeurs futures des capteurs rendant ainsi ces valeurs pas très fiables. Dans ce cas, il faut proposer une amélioration à la méthode d'estimation et de comparaison à la précision, afin d'avoir de meilleurs résultats.

### 2.10.4 IdT/IoT guidage routier

L'IoT est de plus en plus employé cela suite à ses bénéfices. Le transport urbain est en train de subir un changement radical, mais la navigation est toujours basée sur les anciens systèmes qui négligent plusieurs autres paramètres [36].

Ce travail propose une solution qui augmente les capacités des systèmes urbains basée sur l'IoT. Tout en exploitant les FIESTA-IoT, SURF est un système qui calcule le meilleur

itinéraire entre un couple source/destination selon une fonction objective définie par l'utilisateur. La fonction objective englobe plusieurs mesures, y compris le temps de trajet standard, la concentration de monoxyde de carbone (CO) et d'ozone (O<sub>3</sub>), le niveau de pression acoustique, l'humidité relative et la température de l'air ou l'exposition solaire le long du trajet. En d'autres termes cette solution combine entre les informations géographiques disponibles sur les services de navigation et les données récoltées par les capteurs déployés dans les différentes régions par FIESTA-IoT [36].

Dans cet article les auteurs ont traité des travaux sur le smart routing decision et ils les ont classés dans des catégories en se basant sur le but de leurs solutions. Ils ont ensuite comparé et expliqué comment leur travail est différent de ces derniers. SURF est un service WEB qui calcule le meilleur itinéraire entre un point de départ et d'arrivée. L'architecture de ce dernier est basée sur SFS et SBS [36].

SFS « SURF Frontend System » s'agit de l'interface ou plus exactement d'une page WEB pour l'interaction entre l'utilisateur et le système. Elle contient des options pour l'utilisateur qui sont : « maintenant » qui lui permet d'avoir le résultat à l'instant, « choix de l'horaire » qui lui permet de choisir le moment où il veut obtenir le résultat et enfin l'optimisation qui lui convient. Lors de la validation de la localisation (source / destination), le SFS transmet ses données au SBS qui répond en affichant une liste de divers types de capteurs disponibles sur la carte, ainsi permettant à l'utilisateur d'ajuster les mesures en choisissant le type de capteurs à utiliser. Après validation le système renvoie non seulement le meilleur itinéraire mais aussi le second meilleur en ajoutant des informations pour chaque. Dans le cas où aucun capteur n'est disponible entre le couple source / destination, un message est affiché par SFS ainsi que le calcul du meilleur itinéraire en se basant sur le plus court chemin. Le SBS est le composant central du système SURF et contrôle toutes les interactions avec les utilisateurs ainsi qu'avec la plate-forme FIESTA-IoT. Ses principales tâches sont : (i) la découverte de ressources à partir de FIESTA-IoT en fonction de l'entrée de l'utilisateur, (ii) le traitement des données pour le calcul des meilleurs itinéraires et (iii) la construction de modèles de prédiction pour le calcul des itinéraires dans le futur. Le SBS est implémenté sur un environnement virtuel Ubuntu 16.04 avec un processeur Intel Xeon E5-2676 v3 à 2,40 GHz et 1 Go de RAM [36].

Dans cette étude les auteurs ont évalué leur travail sur différents plans qui sont : Disponibilité des données, interpolation de données spatiales, prévision de données temporelles, impact et subjectivité.

En conclusion, SURF est un système de routing urbain intelligent basé sur la plate-forme FIESTA-IoT. Il est capable de faire face au manque de données spatiales et temporelles. L'étude objective et subjective démontre l'impact de ce type de système sur le public. Les auteurs ont ajouté les futures directions de recherche envisagées.

#### 2.10.4.1 Critique

La proposition de ce travail repose principalement sur la récolte de données sur la plate-forme FIESTA-IoT, il s'agit d'une page web seulement et prend en considération l'optimisation de l'utilisateur basé des périphériques bien précis. L'amélioration peut se faire sur la page web en créant une application afin d'être plus disponible et sur les périphériques d'optimisation. La solution propose deux meilleurs chemins sans ajouter les changements dynamiques qui peuvent se

produire. De plus cette application vise seulement les piétons et les cyclistes.

### 2.10.5 Guidage sécurisé sur Manchester

Vu l'augmentation de la congestion urbaine au fil du temps, les chercheurs ont développé plusieurs solutions pour faire face. Nous citons comme exemple, les systèmes de contrôle dynamique des feux de signalisation et les systèmes de navigation en temps réel (VNS) [47].

Dans cet article [47], les chercheurs ont proposé, implémenté et testé une nouvelle application de routage personnalisé et compatible avec Android et IOS qui peut fournir un choix d'itinéraire entre deux emplacements, en fonction des préférences de chaque conducteur sur la distance de déplacement, le temps de trajet et le niveau de sécurité. Cette dernière métrique est très importante car la survenance d'un incident aléatoire sur une route entraîne des embouteillages non récurrents difficiles à éviter. Ils ont également présenté les principaux résultats de validation de ce système basé sur les cartes de la ville de Manchester en Angleterre.

Dijkstra et A\* sont les deux algorithmes qu'ils ont utilisés dans leur système pour planifier des itinéraires. Le but de Dijkstra est de trouver le plus court chemin dans un réseau, tandis que A\* vise à trouver la solution la plus optimale. Si on compare entre les deux algorithmes, Dijkstra est meilleur pour le calcul des itinéraires si tous les nœuds doivent être visités dans la route, sinon A\* est le meilleur dans le cas contraire.

Les deux systèmes de routage avancé (NRR) et ( ReFocus+) ont été simulé par un simulateur de trafic SUMO (Simulation Urban Mobility) et leurs performances ont été comparées à d'autres solutions de pointe. Dans les deux cas, ils étaient plus efficaces que les systèmes existants auxquels ils étaient comparés en termes de temps de trajet [47].

Les critères d'itinéraire courants incluent la distance de déplacement et le temps de trajet estimé. Pour calculer un itinéraire, la carte d'un réseau routier est convertie en un graphe avec des nœuds et des arêtes pondérées. Les nœuds représentent les carrefours routiers tandis que les bords représentent les routes. En termes de congestion, les poids doivent tenir compte de la longueur de la route et de la densité du trafic [47]. Le système de routage personnalisé que les chercheurs ont mis en place est décrit de point de vue de développeur et de l'utilisateur.

Le système comprend une interface utilisateur, un algorithme de routage et une base de données, l'interface utilisateur accepte les entrées des utilisateurs et les transmet à l'algorithme de routage. Sur la base de cette entrée, l'algorithme récupère la carte correspondante, les données de trafic et les informations sur les événements de la base de données, calcule l'itinéraire le mieux adapté et renvoie les résultats à l'interface utilisateur pour les afficher aux utilisateurs finaux.

L'interface utilisateur, ainsi que l'algorithme de routage personnalisé décrit plus loin, sont tous développés à l'aide de framework Flutter et langage de programmation Dart basés sur ce framework. Flutter est choisi parce que l'application développée à l'aide de celui-ci peut être bien compatible avec de nombreuses plates-formes couramment utilisées.

Les chercheurs ont personnalisé l'algorithme A\* pour permettre un routage multicritères, dont les préférences sont personnalisées par différents utilisateurs. Ils ont considéré trois facteurs d'une route dans leur algorithme de routage : la distance de déplacement  $R_d$ , le temps de trajet estimé  $R_t$  et le taux d'accidents  $R_a$ . Ils ont agrégé ces trois facteurs sous forme de combinaison linéaire pour obtenir une valeur de pondération unique pour chaque route  $W$  :

$$W = W_d \times R_d + W_t \times R_t + W_a \times R_a [47]$$

Ou  $W_d$ ,  $W_t$ ,  $W_a$  sont les valeurs de poids pour chacun des trois facteurs mentionné ci-dessus. Les données cartographiques de la zone sélectionnée dans la ville de Manchester sont exportées à partir de la source de données ouverte largement utilisée : OpenStreetMap [47].

Ils ont choisi MongoDB comme base de données de leur système car il est principalement basé sur la technologie NoSQL. La première étape que l'utilisateur doit faire est de saisir les emplacements d'origine et de destination à partir de la carte donnée. La deuxième étape consiste à ajuster les préférences d'itinéraire. Plus précisément, les utilisateurs peuvent ajuster l'importance qu'ils accordent à chacun des trois critères de routage pour l'itinéraire souhaité. Une fois que l'utilisateur a cliqué sur le bouton Rechercher un itinéraire, le système calcule l'itinéraire en conséquence et affiche les résultats sous forme de formes d'itinéraire.

Les auteurs se sont intéressés à évaluer l'efficacité et la convivialité de leur système. Pour la première, ils ont testé l'application sous trois paires origine/destination (O/D) dans la ville de Manchester, au Royaume-Uni. Les tests ont confirmé que les itinéraires calculés étaient affichés sur la carte comme prévu et changeraient si un itinéraire différent avait été trouvé en fonction des poids des métriques donnés [47]. Concernant la convivialité de leur système de routage personnalisé vu son importance, cinq testeurs (d'âge et de sexe très variés) ont été sélectionnés pour tester le système et répondre à un questionnaire sur leur expérience. D'après les scores SUS (System Usability Scale) présentés cela indiquerait que la convivialité de l'application pourrait être améliorée.

## 2.10.6 Comparaison entre les articles

Dans notre étude nous avons traité cinq articles différents, chacun apporte une analyse et solution propre à lui. Dans le premier article « Distributed Learning for Vehicle Routing Decision in SoftwareDefined Internet of Vehicles » [41] les auteurs ont employé le multi-agent en plus d'avoir créé une nouvelle architecture SDIoV tout ceci afin de concevoir un algorithme DLRD et clôturé l'étude par une simulation pour analyser leur travail. Dans le « A Comparative Study of Vehicles Routing Algorithms for RoutePlanning in Smart Cities » [49] il s'agit d'une simple comparaison entre des algorithmes déjà existants afin de voir le meilleur a exploité dans le guidage routier bien entendu avec tout ceci avec une analyse des résultats. L'article « Predictive Trip Planning – Smart Routing in Smart Cities » [42] quand à lui c'est une estimation de flux du trafic dans les zones a faible couverture de capteurs a l'aide d'une régression de processus gaussien, les auteurs ont démonté et testé leur hypothèses du modèle avec un cas d'utilisation réel a Dublin, en Irlande. Cependant l'article « Walk this way An IoT-based urban routing system for smartcities » [36] lui repose sur l'exploitation des données sur FIESTA-IoT et il s'agit d'un site web qui concerne seulement les piétons et les cyclistes nommé SURF et propose les deux meilleures itinéraires avec des critères bien précis. Pour le dernier article « Important Personalized Road Networks Routing with RoadSafety Consideration A Case Study in Manchester » [47] c'est celui auquel nous avons accordé le plus d'attention car il permet aux utilisateurs d'ajuster de manière flexible leurs préférences d'itinéraire en fonction de la distance de déplacement, du temps de trajet estimé et du niveau de sécurité.

En conclusion tous les articles tournent sur le même sujet qui est le guidage routier et ont le but commun entre tous ces articles est de mettre fin aux problèmes de congestion et

d'apporter une aide de façon à ce que le système apporte des résultats rentables et optimales en utilisant les technologies les plus récentes.

## 2.11 Conclusion

Dans ce chapitre, nous avons défini la gestion de trafic intelligent en général ainsi que les systèmes existants, le guidage routier et les applications qui y sont relié, nous avons également éclairci la signification de la congestion et sa nature. En plus de tout cela, nous avons cité les travaux connexes traités et nous nous sommes penchées particulièrement sur le cas d'étude de Manchester auquel nous avons apporté des modifications et certaines améliorations dans le but de l'adapter pour le développement de notre application dédiée au guidage routier des touristes dans la wilaya de Bejaia en Algérie. La présentation de cette dernière est l'objectif du chapitre suivant.

# Chapitre 3

## Nouvelle Solution pour le Guidage Routier

### 3.1 Introduction

Bejaia une ville côtière qui regorge de multiples ressources culturelles et naturelles, la rendant ainsi un centre d'attraction pour les touristes des quatre coins du pays et parfois même du monde. L'été est considéré comme la saison pendant laquelle la ville est surpeuplée par des visiteurs en plus de sa propre population qui est en croissance impliquant l'augmentation des véhicules sur ses routes et conduisant ainsi à un trafic mal géré en plus des embouteillages impitoyables. Toute personne non familière avec l'architecture de la ville et ne connaissant pas ses recoins pourra souffrir du trafic et se perdre facilement, et s'il ne connaît pas les lieux populaires à visiter, le déplacement pour lui sera moins intéressant et une perte de temps énorme. C'est pour cela l'emploi du guidage routier semblera la solution à adopter afin d'améliorer le trafic en générale et apporter une satisfaction aux étrangers vis-à-vis des lieux à visiter et de cette manière gagner du temps.

Dans ce chapitre, nous allons nous intéresser à l'approche envisagée pour notre projet, puis nous avons une présentation générale et fonctionnement et par la suite parler des outils que nous allons employer. Nous allons aussi en parler de la démarche que nous allons suivre pour notre travail ainsi que la modélisation et l'architecture. Et clôturer ce chapitre avec une simple conclusion.

### 3.2 Motivation

Suite aux différents problèmes de circulation et d'encombrement que rencontre le trafic de la ville à cause du surplus de véhicules, les conséquences sont énormes, comme par exemple, le nombre de visiteurs qui diminuera. L'idée d'exploiter le guidage routier permettra à l'utilisateur d'être dirigé de A vers B comme expliqué auparavant mais aussi d'avoir des informations sur le trafic permettant ainsi d'éviter la perte de temps et savoir les lieux à visiter sans trop tourner en rond. Nous pouvons démontrer plusieurs exemples d'encombrement désastreux tel que : l'entrée de la ville « Les Quatre Chemins » ou le retour de plage au-delà de 18h sur la route de « TALA OURIANE » un encombrement qui peut prendre des heures, etc.

Afin de mettre fin à ces problèmes, nous proposons une application de navigation cartographique et de guidage routier. Cette application contiendra des critères propres à elle tel que : les lieux touristiques, les plages, café/restaurants et hôtels, en plus du temps et de la distance. Elle permettra de guider et diriger l'utilisateur de la source A qui est sa localisation vers la destination B de son choix. Le but principal de notre travail est subvenir aux besoins des utilisateurs.

Grâce à cette application, les usagers auront le calcul du meilleur chemin avec la plus courte distance et le moins de temps à perdre, ce dernier s'affichera en quelques secondes sur l'écran. Ainsi, l'utilisateur devra seulement sélectionner les lieux qu'il souhaitera visiter par exemple : plages, cafés/ restaurants, hôtels, brise de mer et bien sûr gare routière, ferroviaire, maritime et aéroport qui se trouve dans la ville de Bejaia. Cette application fonctionnera sur les smartphones de façons très simples avec l'utilisation du GPS.

### 3.3 Approche Proposée

Lors de notre recherche, nous nous sommes intéressées à plusieurs travaux reliés à notre thème. Une étude a su attiré notre attention suite à la méthode employée et l'amélioration personnalisable que nous pouvons lui apporter, il s'agit de «Personalized Road Networks Routing With Road Safety Consideration : A Case Study In Manchester » [47]. Leur solution repose sur la ville de Manchester dans le but de sécuriser les routes en plus de critère de temps et de distance. Leur application est développée avec le langage DART et la plate-forme FLUTTER, c'est une application de navigation cartographique et de guidage routier. Suite aux problèmes de trafic que rencontre notre ville et surtout pour les personnes étrangères, la navigation peut sembler problématique et causer une perte de temps, c'est pour cela que nous avons envisagé d'apporter des améliorations et de personnaliser cette approche.

Notre approche consiste en une application de navigation et de guidage routier conçu pour les touristes de la ville de Bejaia spécifiquement ou à toute personne ayant envie de se relaxer, visiter ou profiter des différents luxes qu'offre cette merveilleuse ville. Cette dernière permettra à toute personne de se déplacer plus facilement tout en évitant diverses embouteillages, mais aussi de guider l'utilisateur de la source A vers la destination B avec les critères de temps et de distance. Notre but ultime est de faciliter le déplacement aux touristes, accroître la popularité des lieux à visiter, gagner du temps et d'alléger le pourcentage des véhicules sur les routes.

#### 3.3.1 Démarche à suivre

Pour ce projet, nous allons réaliser une application de navigation cartographie dédiée aux touristes de la ville de Bejaia, qui est basée sur le guidage routier. Afin de procéder au développement de cette dernière, nous avons sélectionné plusieurs outils tel que : Flutter, MongoDB et GPS ainsi que Dart comme langage de programmation. Pour commencer, nous allons diviser les tâches et travailler en collaboration vis-à-vis du code de notre application. Ensuite, nous allons nous inspirer des applications existantes et chercher des projets open source qui sont reliés au notre. Puis, nous allons procéder à la récupération des données dont nous aurons besoin et se familiariser avec le langage choisit. Puis, enfin entamer le développement. Cette application contiendra la localisation des divers lieux touristiques ainsi que les plages, taxi, les gares

(routière, ferroviaire, maritime), aéroport, hôtels et cafés/ restaurants qui sont répartis aux quatre coins de la ville. Ceci va permettre à un utilisateur de faire une recherche et choisir une destination. Pour le diriger et lui donner une route optimale, nous commençons par récupérer sa localisation avec une technique de géolocalisation qui sont expliquées dans la section 3.7. La route proposée doit être optimale en fonction du trafic. Pour ce faire, nous récupérons le trafic en utilisant MapBox. Les sections 3.4, 3.5 et 3.6 expliquent tous ce qui est en relation avec ce dernier.

### 3.4 Modélisation du trafic routier urbain

Un réseau routier est conçu afin de permettre à ses usagers de se déplacer d'un point à un autre. Dans ce contexte, il est composé d'un ensemble de routes avec plus ou moins de voies, en fonction des zones d'activités et des lieux d'habitation. Les croisements entre les routes étant inévitables en milieu urbain, de par leur concentration, des intersections permettent de gérer les flux de véhicules entrant en conflit, pour leur sécurité et afin d'éviter les inter-blocages et de permettre aux automobilistes de changer de route. Ces intersections, ou dans certains cas des carrefours giratoires, règlent le trafic et appliquent généralement des règles de priorité ou sont équipés de feux de circulation pour gérer les situations particulièrement dangereuses. Le réseau routier urbain peut donc être vu comme étant une collection d'intersections, liées les unes aux autres par des routes, auxquelles peuvent venir se greffer des éléments opérationnels (panneaux, feux de circulation, etc.).

**-La voie :** c'est le chemin, la route, par laquelle on se rend d'un lieu à un autre. C'est un espace aménagé pour se déplacer en ville (voie urbaine) entre les localités ou en milieu rural (routes). Les voies de communication sont par conséquent les moyens de liaison à travers l'espace. Elle permet de circuler dans les deux sens [38].

**-La route :** itinéraire à suivre pour aller d'un endroit à un autre. C'est une voie carrossable (où peuvent circuler des voitures). Elle permet à l'homme de se déplacer de chez lui pour pénétrer d'autres régions plus ou moins lointaines. La route est caractérisée par sa longueur, son nombre de voies, ainsi que par les sens de circulation.

**-Le carrefour giratoire (rond-point) :** lieu où se croisent plusieurs voies. Les carrefours réduisent la capacité des voies, sauf s'ils sont aménagés à niveaux séparés [38].

**-L'intersection :** une intersection (voir 3.1 ) est le lieu de croisement d'au moins deux routes [38] . Un mouvement est défini par son origine et sa destination. Un courant de mobile est l'ensemble des mouvements de véhicules qui proviennent de la même branche d'accès, à savoir la même origine. Ils sont admis simultanément sans conflit. Certaines voies peuvent être dédiées à un mouvement particulier du courant, comme par exemple les mouvements de tourne-à-gauche. Une intersection peut revêtir plusieurs schémas relatifs à la géométrie de l'infrastructure et aux règles de franchissement.

Cependant, dans chaque intersection, nous pouvons identifier trois zones fonctionnelles :

- **Zone de conflit :** il s'agit de l'espace de croisement des routes. Autrement dit, c'est la ressource critique partagée par tous les véhicules qui traversent l'intersection.

- **Zone de stockage :** En amont de la zone de conflit, elle est l'entrée empruntée par les véhicules.

- **Zone de sortie** : En aval de la zone de conflit, elle permet le soulagement de cette zone.

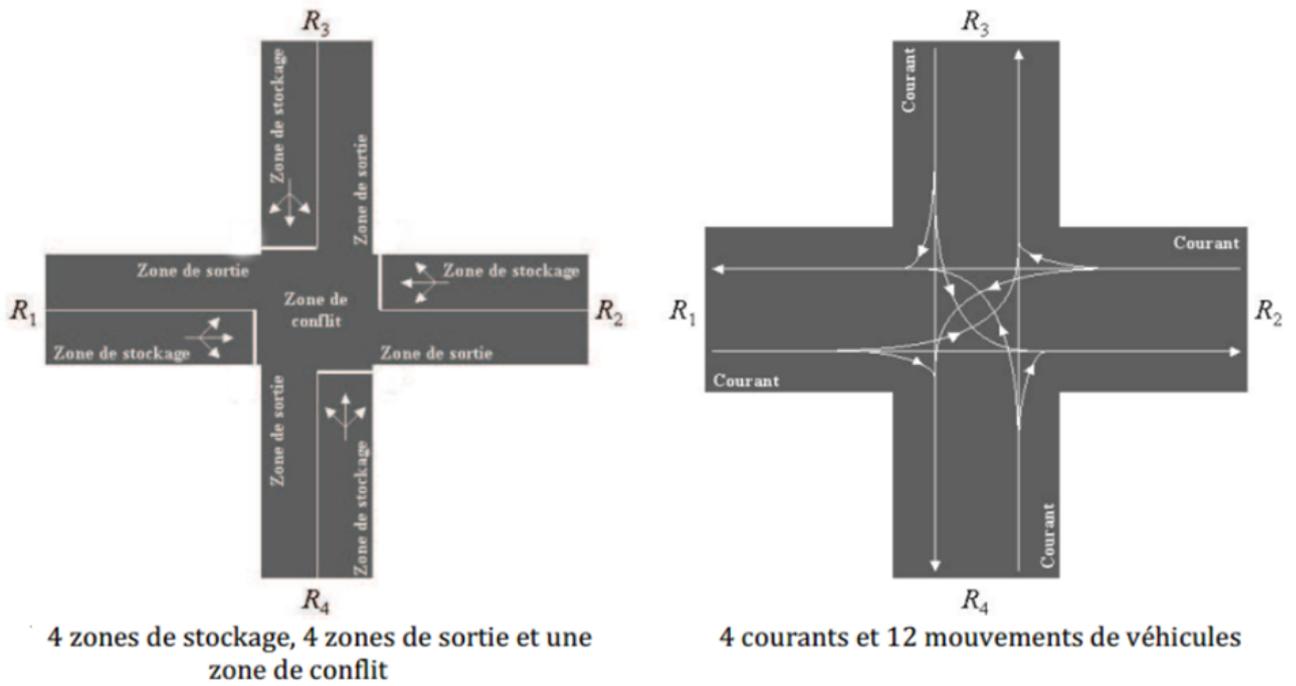


FIGURE 3.1 – Intersection à quatre branches avec mouvement tourne à gauche, à priorité à droite.

### 3.5 Relation demande vitesse-débit

La notion de débit de trafic est une fonction importante qui est liée à la vitesse et la densité de circulation des véhicules [38]. Elle se calcule à partir de l'équation suivante :

$$\text{débits} = \frac{\text{vitesse}}{\text{densité}} \tag{3.1}$$

Le débit varie en fonction de la vitesse et de la densité des véhicules. Cependant, plus la densité est élevée, plus il y a d'interactions entre les voitures, plus la congestion augmente et plus le débit diminue. Cela met en relation la capacité de la route, la demande des usagers, la vitesse de ceux-ci et le débit résultant. Dans une situation de trafic fluide, lorsque les véhicules sont peu nombreux, ils peuvent se déplacer à la vitesse désirée sans problème. Cette vitesse varie en fonction des performances de chaque véhicule, des caractéristiques de la route, des conditions atmosphériques et est limitée par la vitesse légale. Bien entendu, la vitesse diminue à mesure que la circulation s'intensifie. Ainsi, une augmentation de la demande entraînera d'abord une légère diminution puis, progressivement, une diminution plus forte de la vitesse instantanée.

## 3.6 Impact d'encombrement sur le temps

Si on rajoute à une circulation un petit accident ou bien un véhicule qui tombe en panne, les conducteurs qui à haute vitesse sont obligés de réduire leur vitesse et ralentir, Ils entraînent ainsi une réaction en chaîne provoquant un encombrement.

En effet, ce dernier contribue à augmenter le temps des automobilistes passés dans les files d'attente. La figure 3.2 montre l'évolution du temps de déplacement et le taux d'encombrement en fonction de la vitesse, ainsi, nous remarquons que le pourcentage de taux d'encombrement augmente avec la diminution de la vitesse (par exemple, il arrive à un pourcentage de 95% quand la vitesse est 20 km/h), pour le temps nous touchons aussi une augmentation avec l'augmentation du taux et la diminution de la vitesse [38]

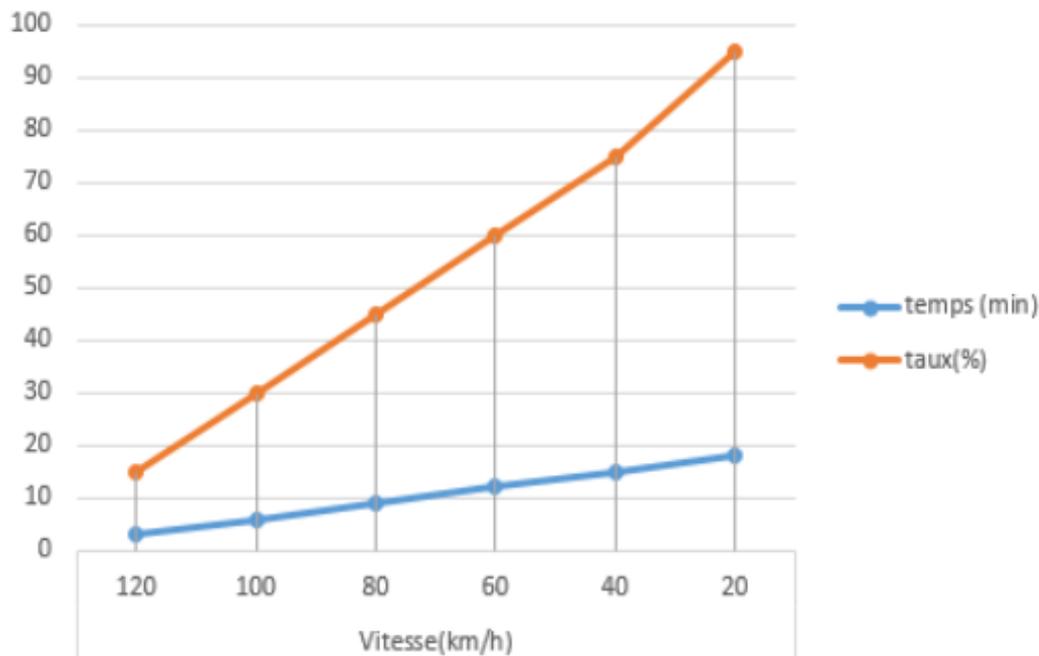


FIGURE 3.2 – Temps en fonction de taux d'encombrement et vitesse.

## 3.7 Géolocalisation

### 3.7.1 Définition

La géolocalisation [22] (géo référencement) est une technique ou pratique qui permet de positionner un objet ou une personne sur un plan ou une carte à l'aide de ses coordonnées géographiques qui sont : longitude, altitude et latitude. Cette opération est réalisée grâce à un terminal qui est capable d'être localisé (par exemple Un récepteur GPS ou système de positionnement par satellite ou d'autres techniques) et de publier à temps réel ou non ses coordonnées géographiques. Les positions enregistrées (voir 3.3) [23] peuvent être stockées au sein du terminal et être extraite postérieurement, ou être transmises en temps réel vers une plate-forme logicielle de géolocalisation. La transmission Temps réel nécessite que le terminal soit équipé d'un moyen de télécommunication de type GSM, GPRS, UMTS (Universal Mobile

Télécommunications System), radio ou satellite afin de lui permettre d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plate-forme de géolocalisation le plus souvent accessible depuis internet.



FIGURE 3.3 – Illustration de la géolocalisation.

### 3.7.2 Techniques de géolocalisation

Il existe plusieurs techniques de géolocalisation, et cela suivant le niveau de précision, le coût, la couverture réseau ou le droit d'accès. Ci-dessous, nous allons vous présenter quelques une des plus pertinentes :

#### 3.7.2.1 Géolocalisation par GSM (Global System for Mobile Communications)

Cette technique permet de connaître la position d'un terminal GSM en se basant sur quelques informations relatives aux antennes GSM aux quelles le terminal est connecté. La précision du positionnement par GSM peut aller de 200 mètres à plusieurs kilomètres, et cela selon si le terminal se trouve en milieu urbain (car dans ce cas-là la densité d'antennes est supérieure), ou en milieu rural.

Plusieurs techniques existent :

- Différence de temps observée ou EOTD (Enhanced-Observed Timed Difference) : le terminal calcule le temps écoulé entre l'émission et la réception de la requête envoyée à l'antenne, il peut alors calculer sa distance par rapport à celle-ci.
- Temps d'arrivée (Time of Arrival)
- Angle d'arrivée (Angle of Arrival)
- Cell ID (Identifiant de cellule)

Aujourd'hui, la méthode GSM la plus utilisée est celle du Cell ID, d'autant plus c'est la moins coûteuse car elle ne nécessite pas l'emploi d'un matériel. Cette technique consiste à récupérer les identifiants des antennes GSM aux quelles le terminal est connecté. Ensuite, grâce à une base de données faisant le lien entre les identifiants des cellules et les positions géographiques des antennes, le terminal est capable de déterminer sa position et d'émettre une estimation. En d'autre terme, tant que le portable se trouve dans une zone couverte par le réseau, il se connecte à une antenne relais GSM. Puis celui-ci est localisé à l'aide d'une base de données faisant le lien entre les identifiants des cellules et les positions géographiques des antennes. Cette localisation est rapide vis-à-vis du temps mais très peu précise car elle dépend du nombre d'antennes relais et

leur rapprochement (plus l'antenne est isolée, plus la zone de couverture est vaste et moins la localisation est précise, la précision est de l'ordre de 50 à 500 Mètres dans un milieu urbain alors qu'elle peut varier de 5 à 15Km dans les régions isolées).

Les bases de données peuvent être mises à disposition par les opérateurs pour leurs abonnés, ou par des sociétés privées qui recensent les antennes GSM ou ayant des partenariats avec les opérateurs. Des bases de données communautaires existent et sont le plus souvent alimentées par les utilisateurs eux-mêmes.

### 3.7.2.2 Géolocalisation par WIFI

Les émissions signaux permanent des points d'accès Wifi, qui se trouvent généralement dans la plupart des "box Internet", permettent à des ordinateurs ou à des téléphones mobiles de les reconnaître et de s'y connecter. Ces signaux contiennent notamment un numéro d'identification unique propre à chaque point d'accès (appelé "BSSID"), celui-ci est utilisé par les Smartphones afin de géo localiser une personne.

#### **Fonctionnement de la géolocalisation à partir des points d'accès wifi :**

Lorsqu'une personne lance une application de géolocalisation sur son Smartphone, ce dernier liste les points d'accès Wifi qui se trouve à sa portée, et interroge une base de données spécifique qui permet d'associer ce point d'accès à une position géographique. Le Smartphone va donc être capable de géolocaliser précisément son propriétaire. Plusieurs sociétés telles que Google, Skyhook Wireless, Microsoft et Apple ont donc constitué des bases cartographiques recensant ces points d'accès Wifi dans le but de fournir leurs services de géolocalisation. Ces bases peuvent être créées de deux manières :

- À partir des données transmises par les téléphones mobiles eux-mêmes lorsqu'ils demandent à être géo localisés.
- À partir des données Wifi collectées par des véhicules se déplaçant dans les rues des villes et sur les principaux axes routiers.

### 3.7.2.3 Géolocalisation par adresse IP (sur internet)

Cette méthode permet de désigner la position géographique de n'importe quel terminal connecté à internet en se basant sur son adresse IP. Les adresses IP sont gérées par l'IANA (Internet Assigned Numbers Authority), qui s'agit d'une organisation qui s'occupe de découper les blocs d'adresses IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent. Toutes ces attributions étant très bien documentées, il est possible de détecter dans quel pays se trouve un terminal connecté à internet et cela à l'aide de son adresse IP. Il est aussi possible d'obtenir un niveau de précision de l'ordre de la ville et cela simplement en se basant sur la distribution des adresses IP faite par les fournisseurs d'accès à internet. Quant à la précision de la géolocalisation par identification de l'adresse IP utilisée, cette dernière est cependant sans commune mesure avec celle permise par un GPS ou par triangulation des téléphones mobiles. On considère généralement que les meilleurs référentiels d'adresses IP sont globalement fiables à l'échelle des villes.

### 3.7.2.4 Géolocalisation par RFID (Radio Frequency Identification)

Les technologies RFID sont conçues afin d'identifier quelque chose à l'aide d'étiquette passive qui répond aux signaux de radiofréquence. Lors de cette requête, l'étiquette RFID fournira des informations sur l'objet. Elle peut également être en mesure d'enregistrer des informations. Le RFID peut ainsi déterminer quel objet ou produit est passé par un point de passage, où se trouvent les capteurs RFID, et peut également indiquer dans l'étiquette de l'objet qu'il est passé par tel point de passage. Les avantages sont évidents, en particulier pour toutes les chaînes logistiques. Nous disons parfois RFID passive pour montrer que les étiquettes RFID ne peuvent que réagir, pas émettre par elles-mêmes.

### 3.7.2.5 Géolocalisation par GPS

Le Global Positioning System (GPS) est un système de navigation par satellite qui a été conçu par l'US Department of Defence (DoD) au début des années 1970. Initialement, le GPS a été développé comme un système militaire pour répondre aux besoins de l'armée américaine. Toutefois, il a fini par être mis à la disposition des civils.

#### 1. Définition

Le Global Positioning System est un système de positionnement et de navigation par satellite, qui fonctionne grâce au réseau GPS qui consiste en 24 satellites qui tourne autour du globe. Le principe de fonctionnement est basé sur la réception de signaux du moins 4 satellites, ce qui permet de localiser le récepteur GPS dans l'espace sur les trois dimensions longitude, altitude et latitude. Le récepteur GPS permet la décharge des points enregistrés et les visualiser dans un SIG, celles-ci sont soit de nouvelles données ou elles servent à modifier et actualiser celles qui sont déjà existantes dans ce dernier (voir 3.4) [24].

#### 2. Fonctionnement

Le GPS comprend au moins 24 satellites tournant à une altitude de 20 200 km. Ces derniers émettent en permanence un ou plusieurs codes pseudo-aléatoires précisément datés grâce à leurs horloges atomiques et messages de navigation. Ce message est transmis à 50 bits/s, Et comprend en particulier les éphémérides qui permettent le calcul de la position des satellites, ainsi que des informations sur leur horloge interne. Les codes sont un code C/A (acronyme de « Coarse acquisition », en français : « acquisition brute ») avec un débit 1,023 Mbit/s et de période 1 ms, et un code P (pour « Précis ») de débit 10,23 Mbit/s et de période 1 semaine. Le premier est libre d'accès, le second quant à lui est réservé aux utilisateurs autorisés car il est le plus souvent chiffré : c'est ce qu'on appelle un Y-code.

C'est ainsi, qu'un récepteur GPS qui reçoit des signaux d'au moins quatre satellites équipés de plusieurs horloges atomiques peut connaître sa distance par rapport à ceux-ci, cela en calculant les temps de propagation de ces signaux entre les satellites et lui-même, et par trilatération, situer précisément en trois dimensions n'importe quel point placé en visibilité des satellites GPS, avec une précision de 3 à 50 mètres pour le système standard. Dans certains cas, seuls trois satellites peuvent suffire. La localisation en altitude (axe Z) n'est pas d'emblée correcte alors que la longitude et la latitude (axes X et Y) sont encore

bonnes. On peut donc se contenter de trois satellites lorsque l'on évolue au-dessus d'une surface « plane » .

Et pour le développement de notre application c'est la technique que nous avons choisie de suivre.

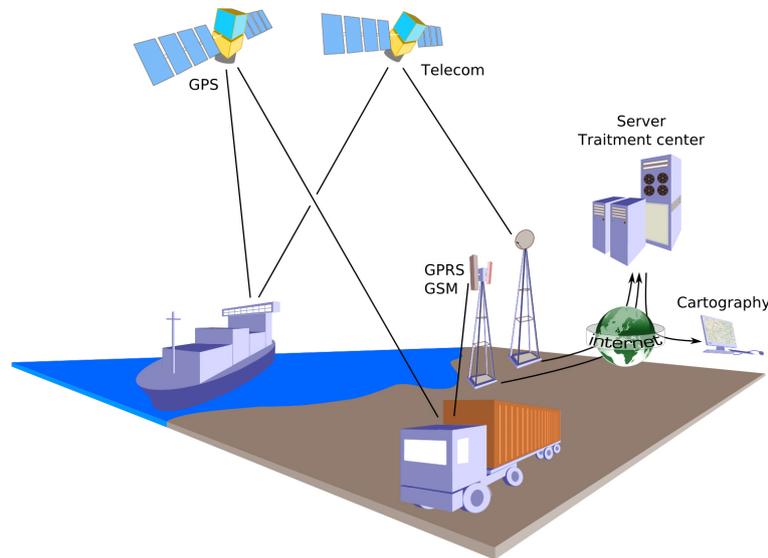


FIGURE 3.4 – Principe de la géolocalisation par GPS.

## 3.8 Outils et Langage de programmation

Pour le développement de notre application nous avons opté pour les outils et langage de programmation défini dans ce qui suit :

### 3.8.1 Flutter

Conçu par Google en 2011 [18] , c'est un Framework open source (voir 3.5) dédié pour le développement d'applications multiplateforme (Android et IOS), avec le langage Dart. Ce dernier est disponible sur tous les systèmes d'exploitation Windows, linux et mac ainsi que sur les machines virtuelles tel-que Visual code Studio et Android Studio. Le but est d'arriver à développer des applications plus rapidement et pour ceci il contient :

- Divers widget pour Android et iOS .
- Un moteur de rendu 2D optimisé avec un support de texte .
- Des API afin d'exécuter les tests unitaire et d'intégration et d'assurer l'interopérabilité avec des systèmes et SDK tiers .
- Des headless test runner dans le but d'exécuter des tests sur Windows, mac et linux .
- Des outils type command-line pour créer/tester/compiler des applications. [19]

### 3.8.2 MongoDB

MongoDB [9] (voir 3.6) est une base de données NoSQL open source orienté document. Souvent utilisée pour le stockage de volumes massifs de données, celles-ci sont sauvegardées



FIGURE 3.5 – Flutter Illustration.

sous forme de document JSON et cela contrairement aux bases de données relationnel SQL traditionnel ou les données sont stockées sous forme de tableaux et des colonnes. Parmi les avantages de cette BDD la simplicité, la flexibilité de ses environnements et les documents qui n'ont pas de schéma prédéfini. Ses fonctionnalités principales :

- SIG ou Système d'information géographique qui permet la manipulation des positions sur un plan ou sur le globe terrestre.
- La réplication qui permet de doubler les serveurs de base de données pour une meilleure tolérance de pannes.
- Le Sharding permet de répartir les données sur plusieurs serveurs soit pour simplement augmenter les performances ou pour répartir les données géographiquement.
- GridFS permet de stocker simplement des fichiers en base de données.

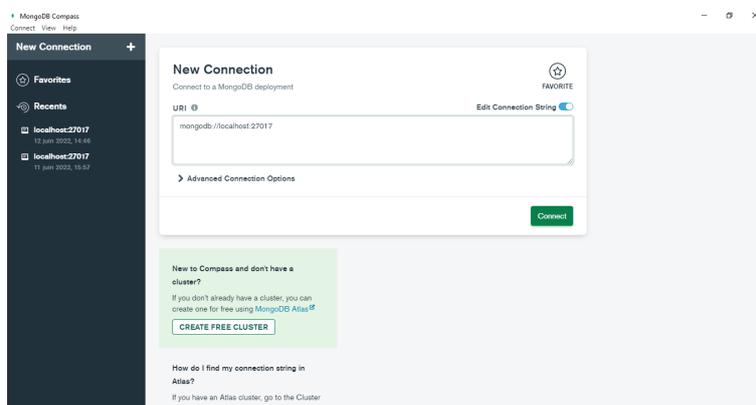


FIGURE 3.6 – mongoDB Compass.

### 3.8.3 Vscode

Visual studio code [10] (voir 3.7) est un éditeur de code open source, gratuit et multiplateforme (Windows, Mac et Linux) développé par Microsoft, à ne pas confondre avec l'IDE Visual Studio propriétaire de Microsoft. Vscode est développé à l'aide d'Electron et tire parti des capacités d'édition avancées du projet Monaco Editor. Conçu principalement pour le développement d'applications avec JavaScript, Type Script et Node.js, de plus il peut être adapté à d'autres

types de langages grâce à un système d’extension complet. Les caractéristiques principales du Vscode :

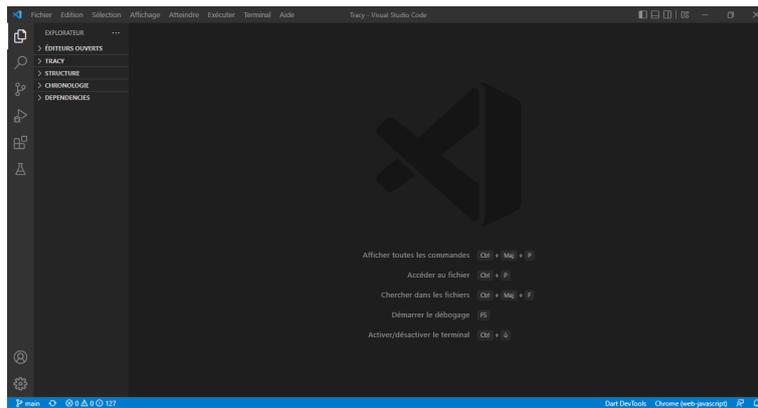


FIGURE 3.7 – VsCode.

- **IntelliSense** : une technologie avancée qui propose, outre à la mise en évidence de la syntaxe et la complétion automatique du code, un système d’inférence articulé et basé directement sur la logique du code source.

- **Intégration native avec Git** : le logiciel implémente le système de gestion de versions Git directement dans l’interface de l’éditeur, ce qui représente un avantage pour pouvoir effectuer les opérations de versioning plus aisément.

- **Ligne de commande intégrée** : toujours dans l’interface de l’éditeur, il est possible de lancer la ligne de commande et exécuter toutes les commandes disponibles sur le système d’exploitation.

- **Ecosystème d’extensions** : les extensions sont au cœur du projet et il existe même un système assez simple pour développer/publier ses propres extensions

- **Debugging intégré** : pour les développeurs plus avancés, il existe également des fonctionnalités de debugging directement à l’intérieur de l’éditeur.

### 3.8.4 MapBox

Spécialisé dans la cartographie en ligne. Mapbox [11] développe un ensemble de technologies et d’outils cartographiques, dont la bibliothèque Mapbox GL.js. Open source, elle permet d’intégrer des cartes customisables et interactives au diverses projets, en plus de cela elle rend modifiable le style dynamique suite à l’utilisation des vector tiles. Cette dernière repose sur les données d’OpenStreetMap, et elle fournit des moteurs de routage puissants, des temps de parcours précis basés sur le trafic et des indications intuitives de type turn by turn, en plus de tout cela elle contient des API, SDK et des données cartographiques mises à jour en direct offrant ainsi aux développeurs des outils pour créer de meilleures expériences de cartographie, de navigation et de recherche sur toutes les plateformes.

**Remarque** *Un vector tile est un format de donnée léger pour stocker nos données géo spatiales.*



FIGURE 3.8 – MapBox.  
[21]

### 3.8.5 ImgBB

C'est un site d'hébergement d'images [16], il permet aux utilisateurs de déposer des images qui ne doivent pas dépasser les 32 MB et d'obtenir en retour des types de lien (lien simple, intégration html ...).



FIGURE 3.9 – ImgBB.

### 3.8.6 Pixabay

Il s'agit d'un site [17] sur lequel sont partagées plusieurs images, illustrations, images vectorielles et clips vidéo. Ces derniers sont libres de droits, ce qui permet de les utiliser comme illustration lors du développement des applications.

### 3.8.7 Rydmike

Rydmike est un site web qui propose plusieurs thèmes que les développeurs peuvent exploiter dans leurs applications, et cela gratuitement et sans aucune licence. C'est un outil qui peut venir en aide à plus d'un.

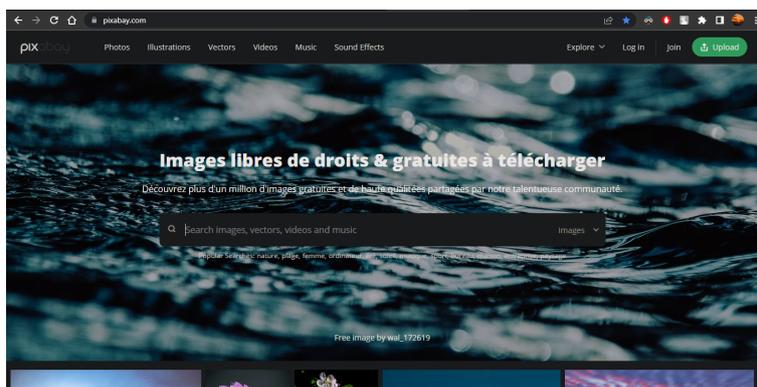


FIGURE 3.10 – Pixabay.

### 3.8.8 GPS

Développé par les Etats-Unis, le Global Positioning System est un système de positionnement et de navigation par satellite, qui fonctionne grâce au réseau GPS qui consiste en 24 satellites qui tourne autour du globe. Le principe de fonctionnement est basé sur la réception de signaux du moins 4 satellites, ce qui permet de localiser le récepteur GPS dans l'espace sur les trois dimensions longitude, altitude et latitude. Le récepteur GPS permet la décharge des points enregistrés et les visualiser dans un SIG, celles-ci sont soit de nouvelles données ou elles servent à modifier et actualiser celles déjà existantes dans ce dernier.

### 3.8.9 Figma

C'est un logiciel [25] de de prototypage et design qui combine leurs fonctions sur différents degrés d'abstraction (high and lowfidelity) ainsi qu'à différents niveaux d'interactivité (dynamique ou statique). Il est énormément exploité dans la création d'UI dans les domaines des UX. Celui-ci, non seulement propose des fonctionnalités de création de prototype mais en plus la capacité de collaborer le même projet avec d'autres infographe. De plus, le logiciel propose des éléments qui aide à la gestion d'un projet UX, comme par exemple : Un tableau blanc interactif, appelé FigJam, avec des Templates, pour l'organisation des projets. Les fichiers et extensions/plugins contribués par la communauté sont également disponibles (voir 3.11 [26]).

### 3.8.10 Git

C'est un logiciel de gestion de versions (Version Control System) [12], celui-ci suit l'évolution des fichiers sources et garde des traces de chacune. Cela permet de retrouver les différentes versions d'un fichier quelconque ou d'un lot de fichiers connexes et d'éviter ainsi différents problèmes. Il est généralement utilisé dans les programmes informatiques et leur développement.

### 3.8.11 Github

Github [13] est un site Web qui fournit un service cloud aux développeurs pour stocker et gérer leur code. Il offre la possibilité de suivi et de contrôle de la modification apportée à ce dernier. Il est considéré comme le moyen courant d'hébergement de projets open sources ainsi

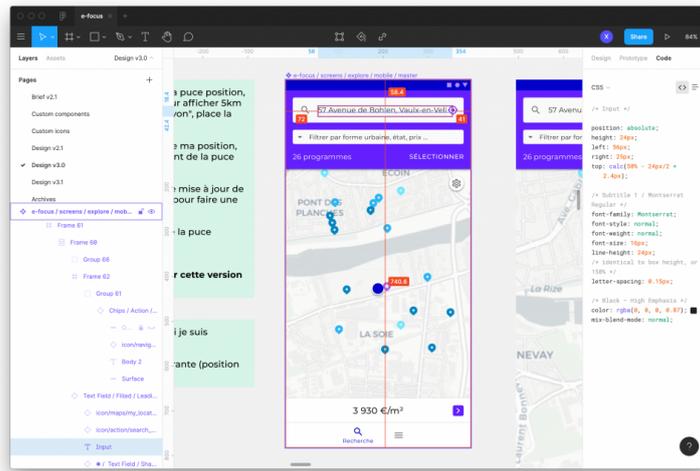


FIGURE 3.11 – Figma illustration

que le partage de contenu. Il permet d'importer n'importe quel code source sur une machine locale. Github accorde la capacité de collaboration entre développeurs.

### 3.8.12 Emulateur

Un émulateur [14] est un appareil qui représente un périphérique particulier (Android ou iOS). Il permet de tester des applications en cours de développement et de les exécuter. Il est considéré comme lent et n'est pas très conseillé à utiliser. Dans notre cas, nous allons utiliser nos téléphones Android respectives.

### 3.8.13 Material Design Icons

Il s'agit d'un site sur lequel sont partagées plusieurs icônes material design de Google. Ces dernières ont des licences libres de droit, ce qui permet de les utiliser lors de la création de logo ou différents projet de design.

### 3.8.14 Json

A côté de tout cela, nous allons manipuler Json [20] indirectement grâce à un module Dart et la syntaxe Dart. Ceci pour la communication entre notre base de données et le serveur ainsi que la communication entre le serveur et client soit l'application.

#### 1. Définition :

Il s'agit de l'acronyme de JavaScript Object Notation. C'est un format de fichier open-standard pour stocker des données dans un format organisé et lisible avec un accès facile. Les données sont affichées sous format texte, composé de couples " key / value ". Ce format est étroitement lié à JavaScript, et peut être généré et lu par la plupart des langages de programmation. Cette universalité la rendu très populaire pour stocker, organiser, lire et partager des données dans des applications et des services Web.

### 3.8.15 Dart

Dart [15] est un langage de programmation moderne qui peut se compiler également sur JavaScript, il a été développé par Google. Il peut être utilisé pour la programmation des applications côté serveur, mais aussi pour la création des applications mobiles avec flutter Framework. Quant à la syntaxe de ce dernier, elle ressemble au langage de programmation C.

## 3.9 Conclusion

Dans ce chapitre nous avons expliqué l'approche ainsi que la démarche que nous allons suivre. Puis nous avons fait une présentation et fonctionnement générale de notre application. Puis par la suite nous avons introduit la géo-localisation et sa technique ainsi que les outils et langage choisis pour le développement de notre application. Le chapitre suivant va se porter sur la réalisation, l'implémentations et les tests ainsi que leurs résultats.

# Chapitre 4

## Conception et modélisation de GuidTour

### 4.1 Introduction

Pour le développement et la concrétisation de notre approche, ainsi que la conception de notre application. L'étude théorique des besoins et de la problématique est une étape plus que nécessaire, celle-ci est même incontournable. Dans notre cas nous devons étudier les besoins des utilisateurs et leurs trouver une solution, comment leur faciliter la manipulation de l'application et surtout comment tout mettre en place pour avoir un rendu harmonieux et que celle-ci répond à la norme imposé (Ergonomique).

Cette étude visera principalement a contourner le problème de point à point, afin d'éviter tout problème lors de la conception, et surtout nous aidera a tracer un plan suivre lors du commencement de cette dernière.

Dans ce chapitre, nous allons tracer l'objectif de notre application ensuite nous allons étudier les fonctionnalités et les exigences techniques auxquels notre solution doit répondre. Puis nous parlerons des contraintes et exigences auxquels nous ferons fasse et enfin nous allons faire la modélisation de notre application. Afin de clôturer notre chapitre nous allons ajouter une conclusion.

### 4.2 Objectifs de l'application

Dans les chapitres précédant nous avons étudié le problème, l'approche à suivre et surtout les raisons de mettre en place notre application ainsi que son public visé. Nos lieux touristiques que ne sont pas mis suffisamment en valeur, les encombrements et la perte de temps etc.

L'objectif principal de cette application est de fournir aux utilisateurs une carte interactive des points d'intérêt touristiques de la ville de Bejaia. L'application doit permettre à l'utilisateur de naviguer sur la carte, de zoomer et de panoramiqer pour voir les détails de chaque point d'intérêt. En outre, l'application doit offrir une description et des informations supplémentaires sur chaque point d'intérêt lorsque l'utilisateur clique sur un marqueur sur la carte. Cela permettra aux utilisateurs de mieux comprendre l'histoire et l'importance de chaque point d'intérêt et de planifier leur voyage, sortie ou trajet en conséquence.

Enfin, l'application doit permettre à l'utilisateur de rechercher et de filtrer les points d'intérêt en fonction de leur nom, de leur emplacement ou de leur type (par exemple, plage, parc, monument etc.). Cela permettra aux utilisateurs de trouver rapidement les points d'intérêt

qui les intéressent et de planifier leur itinéraire de manière plus efficace. Surtout celle-ci leurs permettra de prendre le moins de temps possible sur le trajet et de suivre le chemin le plus rapide afin de plus profité de l'endroit.

En résumé, l'objectif de cette application est de fournir aux utilisateurs une source pratique et facile à utiliser pour découvrir et planifier leur voyage, trajet ou sortie en explorant les points d'intérêt touristiques de notre ville.

## 4.3 Fonctionnalités de l'application

Afin que l'application réponde aux besoins des utilisateurs et de les satisfaire, certaines fonctionnalité sont indispensable lors de la réalisation et conceptions de celle-ci, ci-dessous quelques fonctionnalités que nous souhaitons intégré à notre application :

### 4.3.1 Affichage de la carte interactive

L'application doit afficher une carte interactive de notre ville qui peut être zoomée et déplacée par l'utilisateur. La carte doit également afficher des marqueurs indiquant l'emplacement de chaque point d'intérêt.

### 4.3.2 Affichage des informations sur les lieux touristiques

Lorsque l'utilisateur clique sur un marqueur sur la carte, l'application doit afficher une fenêtre contextuelle avec une description et des informations supplémentaires sur le point d'intérêt sélectionné. Ces informations peuvent inclure le nom du point d'intérêt, son emplacement, sa catégorie (par exemple, monument, parc, etc.), sa description et toute autre information pertinente.

### 4.3.3 Recherche et filtrage

L'application doit permettre à l'utilisateur de rechercher et de filtrer les points d'intérêt en fonction de leur nom, de leur emplacement ou de leur type. L'utilisateur doit pouvoir entrer des mots-clés dans un champ de recherche ou sélectionner des filtres de catégorie pour afficher uniquement les points d'intérêt qui correspondent à ses critères de recherche.

### 4.3.4 Stockage des lieux

L'application doit stocker les points d'intérêt dans une base de données ou un fichier de données afin qu'ils puissent être chargés et affichés lorsque l'application est lancée.

### 4.3.5 Compatibilité avec les appareils et les systèmes d'exploitation

L'application doit être compatible avec les différents appareils et systèmes d'exploitation (iOS et Android) et doit être optimisée pour une utilisation sur mobile.

### 4.3.6 Sécurité

L'application doit être sécurisée et protéger les données de l'utilisateur contre les accès non autorisés.

### 4.3.7 Facilité d'utilisation

L'application doit être facile à utiliser et à naviguer pour les utilisateurs

## 4.4 Exigences techniques

Lors de la réalisation et développement de notre application, nous souhaitons répondre à certaines exigences techniques. Réalisant ainsi une application ergonomique, qui répond aux divers besoins des utilisateurs. De plus nous voulons mettre fin aux différents problèmes de navigation dans notre ville et mettre de la lumière sur les lieux touristiques existants :

### 4.4.1 Systèmes d'exploitation compatibles

L'application doit être compatible avec les systèmes d'exploitation iOS et Android.

### 4.4.2 Appareils compatibles

L'application doit être optimisée pour une utilisation sur les appareils mobiles, tels que les smartphones et les tablettes.

### 4.4.3 Prérequis logiciels

L'application nécessitera l'installation de Flutter et de toutes les dépendances requises pour développer et exécuter une application Flutter.

### 4.4.4 Exigences en matière de performance et de mémoire

L'application doit être performante et ne pas surcharger la mémoire de l'appareil lors de son exécution. Des tests de performance et de mémoire doivent être effectués pour s'assurer que l'application fonctionne de manière efficace sur les différents appareils et systèmes d'exploitation cibles.

### 4.4.5 Stockage des données

L'application doit stocker les points d'intérêt dans une base de données ou un fichier de données afin qu'ils puissent être chargés et affichés lorsque l'application est lancée.

### 4.4.6 Sécurité

L'application doit être sécurisée et protéger les données de l'utilisateur contre les accès non autorisés.

## 4.5 Contraintes et hypothèses

Pour la conception d'une application, il existe toujours des contraintes ou situations qui peuvent freiner l'avancement de notre projet. C'est pour cela qu'il faut en premier lieu étudier celles-ci afin de trouver une méthode ou une solution qui pourra nous aider lors du développement et de l'avancement :

### 4.5.1 Temps limités

Le développement de l'application sera limité par le temps alloué. En conséquence, certaines fonctionnalités ou améliorations pourraient ne pas être incluses dans la version initiale de l'application.

### 4.5.2 Données des lieux touristiques et des plages

Les données des lieux et des plages qui seront incluses dans l'application sont difficiles à trouver surtout par manque de moyens. De plus nous supposons que ces données sont exactes et à jour, mais nous n'avons pas la possibilité de vérifier l'exactitude de ces données.

### 4.5.3 Connexion Internet

L'application nécessitera une connexion Internet pour charger les données des lieux et afficher la carte interactive. Nous supposons que l'utilisateur dispose d'une connexion Internet stable et suffisamment rapide pour utiliser l'application de manière satisfaisante.

### 4.5.4 Utilisation sur mobile

L'application a été conçue pour être utilisée sur des appareils mobiles, tels que les smartphones et les tablettes. Nous supposons que l'utilisateur utilisera l'application sur l'un de ces appareils et que l'interface utilisateur sera adaptée à cet usage.

### 4.5.5 Langues supportées

L'application sera disponible en anglais et en français. Nous supposons que les utilisateurs parleront l'une de ces deux langues et que les descriptions seront traduites dans ces langues. Mais en premier lieu nous allons tout d'abord implémenter l'application uniquement en langue française.

### 4.5.6 Compatibilité avec les systèmes d'exploitation et les appareils

L'application a été conçue pour être compatible avec les systèmes d'exploitation iOS et Android et pour être optimisée pour une utilisation sur les appareils mobiles. Nous supposons que l'application fonctionnera de manière satisfaisante sur les différents appareils et systèmes d'exploitation cibles.

## 4.6 Modélisation

### 4.6.1 Diagramme de Classe

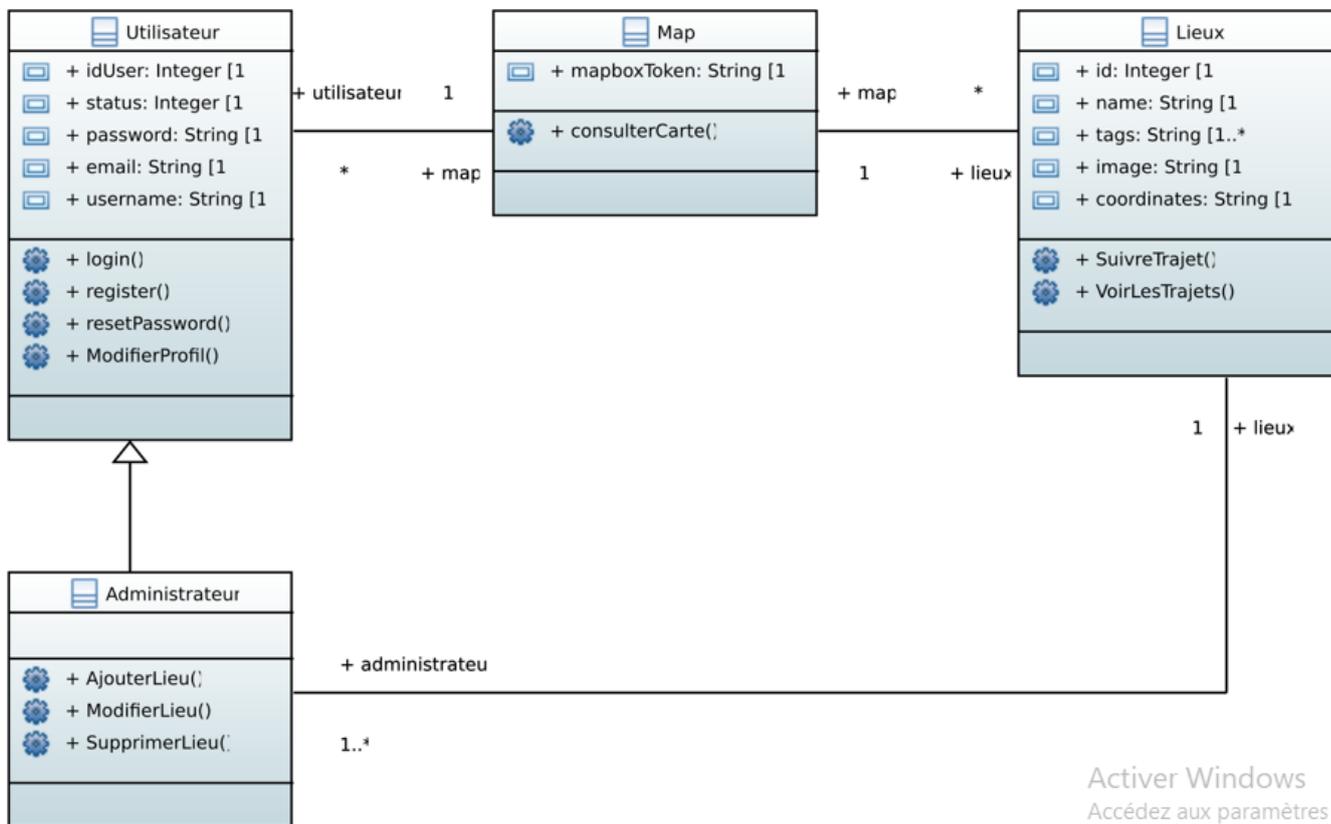


FIGURE 4.1 – Diagramme de Classe

## 4.6.2 Diagramme de cas d'utilisation

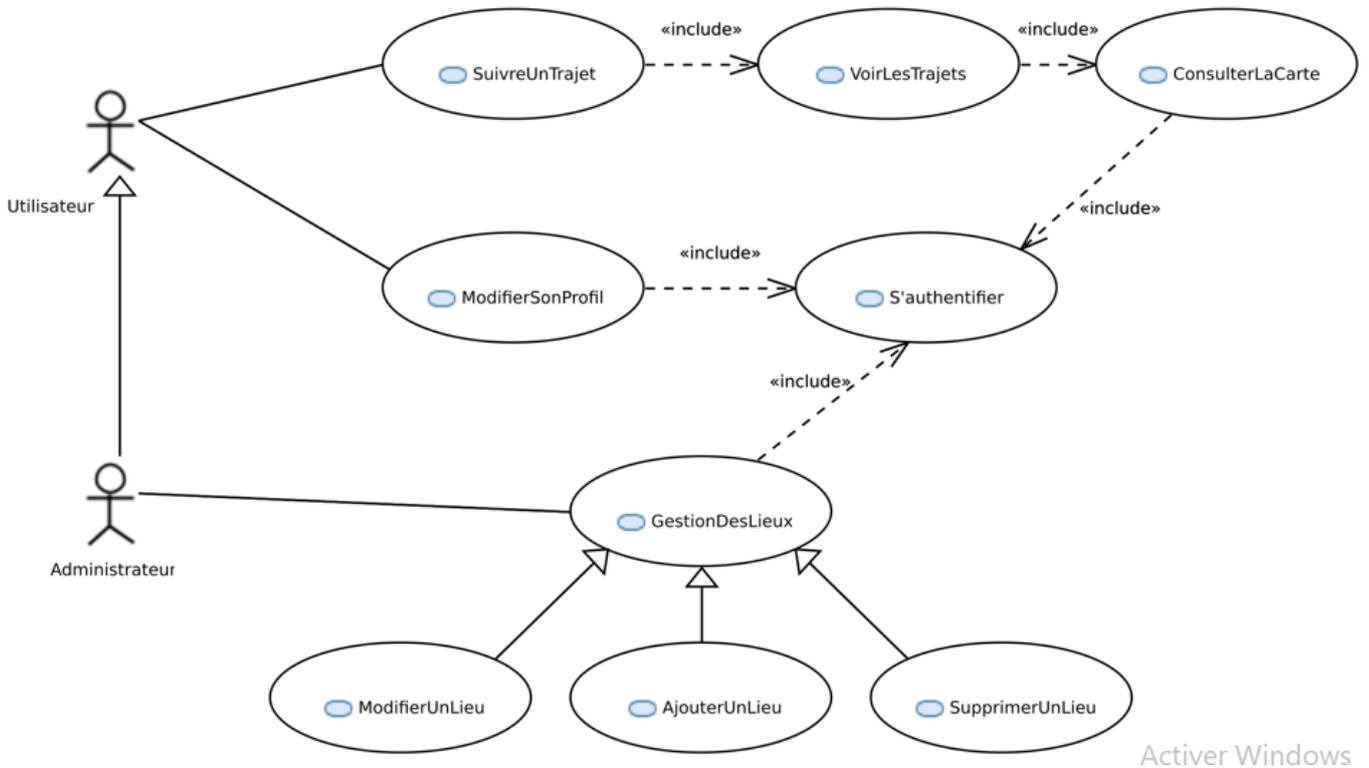


FIGURE 4.2 – Diagramme de cas d'utilisation

## 4.7 Conclusion

En conclusion, l'application de navigation qui répertorie tous les endroits touristiques à visiter vise à fournir aux utilisateurs une solution pratique et simple pour découvrir / redécouvrir de nouvelles destinations et planifier leurs voyages et sorties de manière plus organisée. Elle offrira une interface intuitive et des fonctionnalités utiles, telles que la recherche des lieux et plages par localisation, catégorie ou nom, et la possibilité de créer des listes de favoris à visiter plus tard.

Le développement de l'application suivra un processus rigoureux de conception, de développement et de test pour répondre aux questions et aux problèmes des utilisateurs. Nous sommes convaincus que cette application répondra aux besoins et aux attentes des utilisateurs et leur offrira une expérience de navigation agréable et enrichissante.

Dans le chapitre suivant, nous allons démontrer et expliquer le processus de la réalisation de l'application et tous ces contenus.

# Chapitre 5

## Réalisation et Proposition d'amélioration

### 5.1 Introduction

Les applications de guidage routier sont de plus en plus employées et chacune apporte une nouveauté spécifique. Le développement d'une application de navigation spécialement conçu pour notre ville ne peut être que bénéfique, surtout que celle-ci est considérée parmi les villes les plus visitées en Algérie, grâce aux plusieurs merveilles et lieux à explorer, mais elle souffre du grand problème de l'encombrement.

Cette application viendra en aide pour tous les touristes et facilitera la recherche des divers lieux à visiter et tout cela en quelques clics.

Dans ce chapitre, nous allons faire une description générale de l'application, puis l'architecture de son fonctionnement ainsi que la récolte de données. Nous allons ensuite décrire les étapes de réalisation et de développement de cette dernière afin d'en parler par la suite des problèmes rencontrés. Nous allons finir avec les tests et résultats ainsi qu'une conclusion afin de clôturer ce chapitre.

### 5.2 Description

L'application qui a pour appellation « GuidTour », Guid venant du mot guidage et Tour du mot touriste, donnant une abréviation à «Guidage de Touriste» . GuidTour, comme présenté auparavant, s'agit d'une application de navigation qui aide, dirige et guide son utilisateur d'une source vers une destination. C'est une application basée sur GPS lui permettant ainsi de détecter la localisation et de connaître la situation du trafic.

L'utilisateur doit soit s'authentifier ou créer son compte pour que la carte puisse s'afficher, modifier son profil et sélectionner un lieu à visiter, en plus de cela, l'utilisateur devra accorder son autorisation d'accès à sa localisation. La carte se porte sur la ville de Bejaia et en premier temps l'application propose les lieux les plus visités et adaptés au tourisme sur la ville, en d'autre termes elle propose la localisation, la distance et le temps qui sépare la personne du lieu conseillé. Ensuite, quand l'usager sélectionne le lieu qui lui convient avant que l'interface affiche en quelques secondes sur la carte l'unique meilleur chemin à suivre et cela après le calcul de ce dernier suivant le temps minimal du trajet, il doit d'abord choisir s'il veut voir le lieu sur la carte ou de faire la navigation vers le lieu. Par la suite, l'utilisateur est suivi, guidé et dirigé

jusqu'à arriver à destination. GuidTour ne fait pas seulement que proposer les lieux, mais elle permet aussi aux usagers de faire leur propre recherche.

Les lieux et leurs localisations sont des données récoltés sur Google Maps et mis dans la base de données, et pour cela MongoDB a été sélectionnée car elle peut contenir des données de grande taille. Pour les données du trafic, cartographique, temps et distance ils étaient fournis grâce à MapBox. Quant à l'éditeur de code et émulateur pour coder et tester le développement de l'application, nous avons utilisé Vscode et nos téléphones respectifs.

## 5.3 Architecture de fonctionnement

Comme expliqué dans le chapitre précédent, notre projet est basé sur un travail déjà existant mais notre système de fonctionnement diffère légèrement du leur :

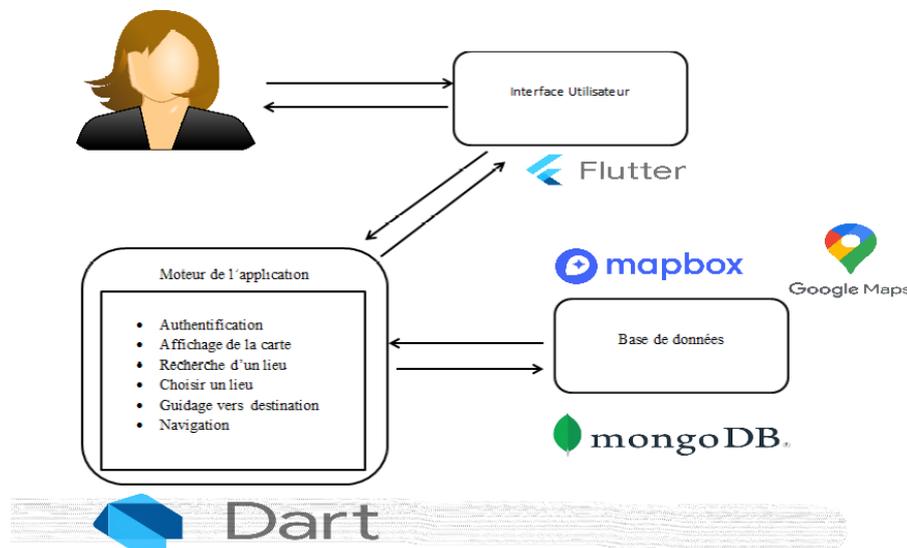


FIGURE 5.1 – Architecture du système.

Ainsi, nous avons utilisé Mapbox pour fournir les données cartographiques ainsi que celles du trafic, temps et distance et Google Maps afin de récupérer les coordonnées géographiques des lieux. Le schéma de la figure 5.1 résume l'architecture générale de cette application.

## 5.4 Réalisation et Développement

### 5.4.1 Logo

Le logo est une représentation graphique et symbolique d'une entreprise, marque ou une application, il représente son identité visuelle. Pour la conception du nôtre, nous avons opté pour une représentation basique et simple. Pour commencer, nous avons choisi figma pour la réalisation de celui-ci, c'est ainsi que nous avons écrit simplement les initiales de l'application avec la police d'écriture nommée Lobster. Ensuite, nous avons cherché le logo « place » dans les logos open source de Google qui est nommé « map-marker-outline », et cela, afin de faire la fusion entre l'écriture et le logo téléchargé. Par la suite, nous avons apporté une légère modification

telle que nous avons effacé l'espace entre les deux pour qu'ils ne soient pas collés et donneraient un mauvais rendu visuel. Finalement, nous avons décuplé pour réaliser la version blanche de celui-ci. La figure 5.2 montre le rendu visuel.



FIGURE 5.2 – Logo GuidTour

### 5.4.2 Organigramme de l'application

Dans la figure 5.3, nous représenterons le fonctionnement de GuidTour par un organigramme et elle sera suivi d'une explication détaillée de ce dernier.

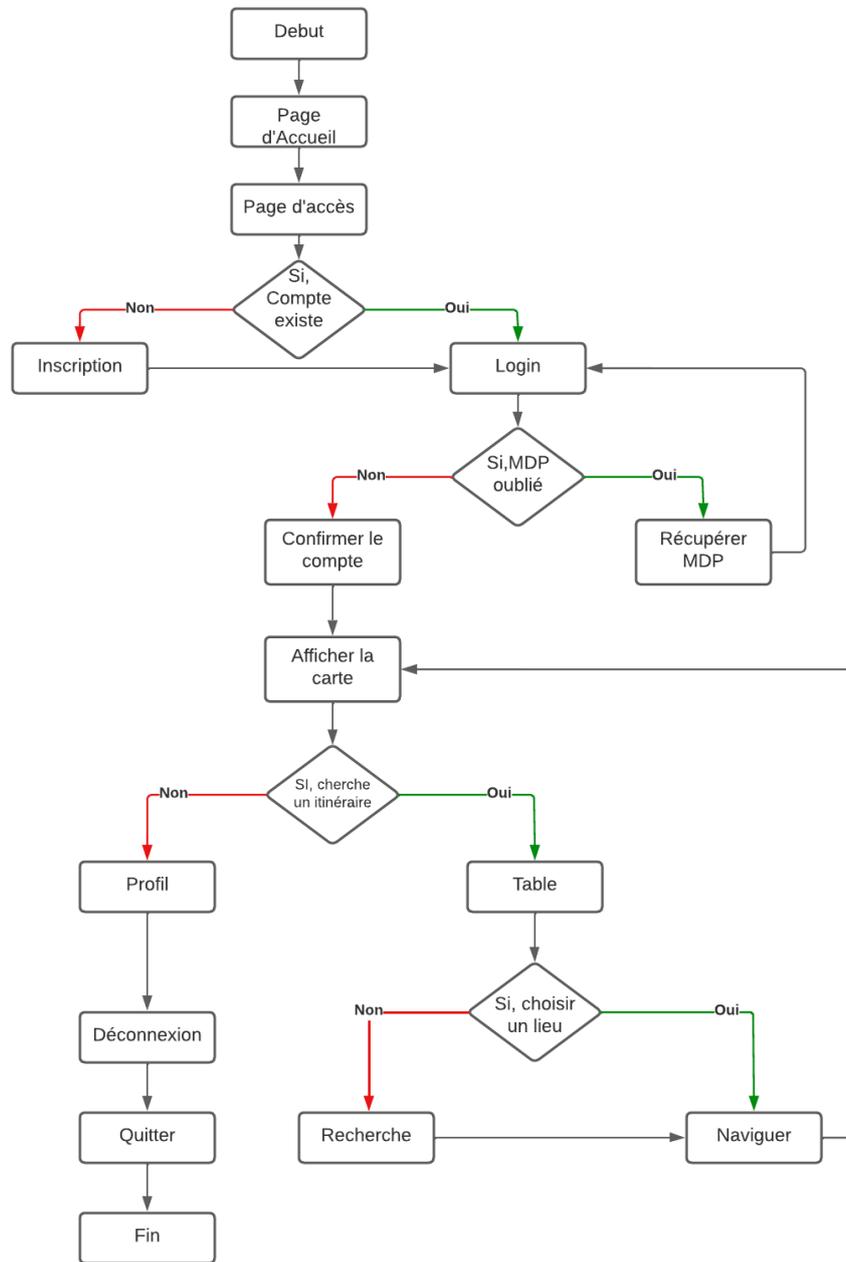


FIGURE 5.3 – Organigramme de GuidTour

### 5.4.3 Accueil

La page d'accueil est importante, étant la première page avec laquelle l'utilisateur interagi. La réalisation de cette dernière est simple et rapide. Vis-à-vis de la conception de notre page d'accueil, nous avons suivi un design facile et attirant. Nous avons tout d'abord commencé par importer toutes les images dont nous aurons besoin pour ce projet, ceci dans un dossier nommé images que nous avons déclaré dans les *assets* de la page *pubspec.yaml*, nous avons aussi ajouté

dans ce même fichier tous les packages à employer. Par la suite, nous sommes passées à la partie codage avec le langage Dart, pour cela, nous avons créé la classe `delayed_animation.dart` qui est une fonction que nous souhaitant utiliser pour donner un effet d'animation à nos divers outils, cette dernière contient deux variables locales : `delay` qui est une valeur que nous souhaitant accorder et `Child` qui contiendra le `widget` concerné. Ensuite, nous nous sommes engagés à la `welcome_page.dart`, dans celle-ci, nous avons mis en avant le logo de l'application, puis une image qui représente le but de cette dernière, par la suite, nous avons mis un petit texte d'accueil et fini avec un bouton qui permet de naviguer vers les pages suivantes et chaque `widget` a son propre délai d'animation qui ajoute un petit air de sophistiqué. GuidTour est reliée à la ville de Béjaïa et ses touristes, et l'été est la saison dans laquelle cette ville est le plus convoité, d'où le choix des couleurs pour le thème était simple et il s'est porté sur l'*aqua* ou connu comme le vert d'eau et du blanc (voir la figure 5.4).



FIGURE 5.4 – Accueil

#### 5.4.4 Authentification et Base de données

L'authentification est la partie dans laquelle l'utilisateur s'identifie en remplissant le formulaire mis en œuvre avec ses informations personnelles. Nous avons choisi d'ajouter une partie d'authentification dans notre projet afin d'identifier nos utilisateurs. Pour réaliser cette option, nous avons tout d'abord créé la `connexion_page.dart`, celle-ci, contient plusieurs `widgets` :

- une *App-barre* qui contient une flèche de retour en arrière ;
- une image toujours dans le thème de l'application et qui est bien sûr libre de droits ;
- un texte avec deux tailles de police différentes ;
- deux boutons : se connecter qui mène vers la `login_page.dart` et s'inscrire qui dirige vers la `signup_page.dart`.

La réalisation de la page de connexion a été faite dans l'ordre des widgets ci-dessus. Ensuite, nous sommes passées à la réalisation de la *login\_page.dart*, nous avons commencé par l'*App barre* qui contient une croix **x** de retour en arrière, puis écrit le texte dans deux tailles différentes. Par la suite, nous avons fait appel au *loginForm* déclarer afin de le positionner au milieu, puis, nous sommes passées au bouton de confirmation, ainsi que les boutons textes du mot de passe oublié, ce dernier mène vers la page *forgot\_page.dart*, le deuxième s'agit de créer un compte qui le guide vers la *signup\_page.dart* et le troisième s'agit du bouton Skip qui permet le retour à la *connexion\_page.dart*. Quant à la déclaration des deux *TextFormField*, ils se trouvent dans *loginForm* qui est une classe de type *StatefulWidget*.

La conception de la page d'inscription s'est produite de la même façon que celle de login, car elle contient du texte, des *TextFormField* et un bouton de connexion seulement. Quand il s'agit de la *forgot\_page.dart*, celle-ci, tout d'abord contient un texte et *TextFormField* ainsi qu'un bouton de confirmation. Après le remplissage du *TextFormField* et l'appuie sur le bouton de confirmation, un nouveau *TextFormField* s'affiche avec un autre bouton de confirmation, celui-ci, permet l'introduction du code reçu sur sa boîte mail.

Tous les *TextFormField* ont fonction que vérifie qu'ils ne sont pas vides, dans le cas contraire un message d'erreur s'affiche, et les *TextFormField* réservé aux adresses email contiennent une fonction qui vérifie que l'email a une certaine forme à suivre. Tout le travail au-dessus a été effectué au niveau du dossier *GuidTour-client*, et dans le passage suivant nous allons nous intéresser au côté serveur.

Pour l'authentification, avoir une base de données est essentiel afin de stocker et enregistrer les différents types d'informations des utilisateurs.

Dans notre projet, MongoDB a été sélectionnée afin d'être la BDD de notre système et cela suite à sa technologie NoSQL, c'est ainsi que tout d'abord nous avons créé une base nommée GuidTour avec la ligne de commande `use GuidTour` et puis une collection `users` dans cette dernière. Puis, nous avons initialisé le serveur avec la ligne de commande `dart create GuidTour-server`, puis, nous avons importé toutes les dépendances dont nous aurons besoin dans le fichier *pubspec.yaml*. Le lancement du serveur se fait grâce au `main.dart` à l'aide du package `shelf`, pendant que le package de `mongo_dart` permet de créer la connexion entre la base de données et le serveur. Ensuite, nous avons choisi le port sur lequel le serveur va se lancer, et celui-ci est lancé avec la méthode `serve` du `shelf` package.

Afin de préciser l'objectif des requêtes http reçu par notre serveur, nous avons ajouté une route qui les redirige suivant leurs URL vers les fonctions définies dans la classe du fichier *auth\_rest\_api.dart*. Dans celle-ci, se trouvent les contrôleurs de l'authentification : Inscription, connexion, récupération du compte, confirmation de l'email et confirmation du code lors de la restauration du mot de passe. Pour l'inscription, nous avons utilisé le package `dpcrypt` afin de hacher le mot de passe et de générer le code de 6 chiffres qui sera envoyé par email à l'aide du package `mailer`. Quant à la connexion, nous avons utilisé le package `jaguar_jwt` de Dart afin de générer un jwt qui est envoyé à l'utilisateur pour qu'il soit capable de s'identifier sans introduire ses identifiants à nouveau ou interroger la base de données. Dans le but de générer le jwt, nous avons besoin d'un mot secret (mot de passe) et l'id de l'utilisateur à chiffrer. Ainsi, toute action effectuée par l'utilisateur sera reconnue, car suite à chaque requête il est dans l'obligation d'envoyer le jeton qui lui correspond. Ce dernier sera analysé pour savoir s'il est correct et correspond à la personne, sinon dans le cas contraire, nous envoyons une erreur

403 (non autorisée) et tout cela grâce au Middleware authorize qui se trouve dans le fichier `auth_middleware.dart`. Ce mot secret est sauvegardé dans une variable d'environnement dans le fichier `chier.env` et récupéré via le package `dotenv`, car dans le cas où une personne arrive le récupérer elle sera capable de générer un token à n'importe quel utilisateur.

Puis, nous avons créé le fichier `mailer.dart` et il s'agit de la configuration du compte Gmail pour l'envoi d'email en cas de :

- Inscription pour confirmer l'email.
- Récupération de compte en cas d'oubli du mot de passe.

### Remarque

Chaque widget cité auparavant est accompagné d'un Delayed animation afin d'avoir le même effet que la page d'accueil.

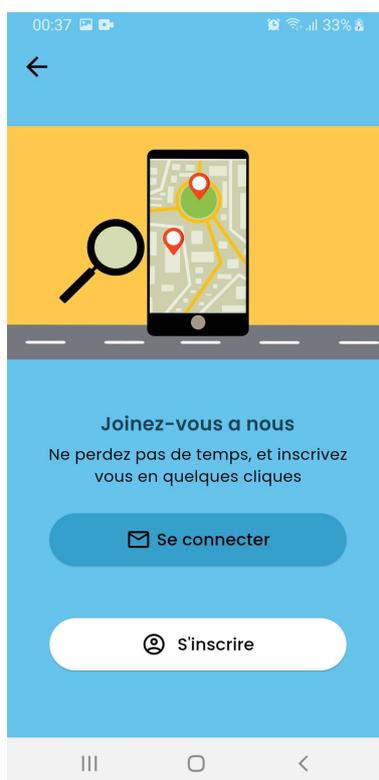


FIGURE 5.5 – Page de connexion

## 5.4.5 Données et Trafic avec MapBox

Avant de plonger dans Mapbox et de mettre en œuvre le projet, nous devons configurer notre application Flutter. En résumé, nous avons ajouté quelques jetons et effectué une configuration initiale avec Android et IOS séparément avant de pouvoir commencer.

Nous avons les données des places touristiques qu'on a récupéré manuellement depuis Google Maps dans le fichier `constants/plages.dart`. Chaque place a un identifiant, un nom, des items, une image et des coordonnées. Les autres données (temps d'attente, heure de fermeture, etc.) sont fictives. Permettez-moi de vous présenter rapidement les autres fichiers du dossier lib du projet.

- **Helpers** Le dossier contient 7 fonctions d'assistance qui gèrent des éléments tels que la gestion des données extraites des API, la gestion des données de préférences partagées et le

formatage ou la modification des données.

- **routes\_manager** dossier contient l'animation qu'on a utilisé dans l'application, et les *routes* entre les pages

- **Screens** contiennent les 13 écrans réels que nous voyons dans l'application - un pour Home, un pour les cartes et un pour une vue tabulaire des Places touristiques, l'autre pour la navigation étape par étape, les autres pour la connexion. Contient aussi 1 fichier — *splash.dart*, qui est notre écran de démarrage. Il a également une logique cruciale dans sa fonction *initState()*.

- **Widgets** contient 6 fichiers — *carousel\_card.dart*, qui sera ensuite utilisé pour créer un carrousel sur la carte, puis basculer entre les choix de places, — *search\_widget.dart* et *build\_search.dart*, ils seront utilisé pour chercher des places dans le Tableau, *forgot\_password.dart* le fichier utilisé pour récupérer le mot de passe oublié et *delayed\_animation.dart*, le dernier fichier *build\_plage.dart* sera utilisé pour la mise en place des places touristiques dans le tableau.

- **Providers** contient 4 fichiers qui vont faciliter l'utilisation et la manipulation entre les classes de l'application

Le fichier *request.dart* contient la requête nécessaire pour récupérer le trafic depuis Map-Box et il est illustré dans la figure 5.6.

```
import 'package:dio/dio.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter_dotenv/flutter_dotenv.dart';
import 'package:mapbox_gl/mapbox_gl.dart';

import '../helpers/dio_exceptions.dart';

String baseUrl = 'https://api.mapbox.com/directions/v5/mapbox';
String accessToken = dotenv.env['MAPBOX_ACCESS_TOKEN']!;
String navType = 'driving-traffic';

Dio _dio = Dio();

Future getTrafficRouteUsingMapbox(LatLng source, LatLng
destination) async {
  String url =
    '$baseUrl/$navType/${source.longitude},${source.latitude}
;${destination.longitude},${destination.latitude}?alternatives=
true&continue_straight=true&geometries=geojson&language=en&over
view=full&steps=true&access_token=$accessToken';
  try {
    _dio.options.contentType = Headers.jsonContentType;
    final responseData = await _dio.get(url);
    return responseData.data;
  } catch (e) {
    final errorMessage = DioExceptions.fromDioError(e as
DioError).toString();
    debugPrint(errorMessage);
  }
}
```

FIGURE 5.6 – Récupérer le Trafic avec MapBox

#### 5.4.6 Mettre en œuvre la capture d'emplacement

Dans cette section, nous allons d'abord capturer l'emplacement de l'utilisateur, puis l'afficher sur l'écran Maps. Pour cela, nous allons commencer par nous assurer que toutes les

autorisations requises sont disponibles, puis capturer l'emplacement de l'utilisateur. Nous stockerons ensuite ces données dans les préférences partagées. Enfin, nous supprimerons la façon dont la navigation se fait avec un fichier `Future.delayed()`.

Maintenant que cet emplacement est stocké dans `sharedPreferences`, tout ce que nous avons à faire est de le capturer dans la classe `PlageMapProvider` et de montrer les données à l'utilisateur. Nous utilisons la fonction d'assistance `LatLng getLatLngFromSharedPrefs()` à cette fin.

### 5.4.7 Utiliser le SDK Mapbox Maps

Nous pouvons maintenant afficher une carte à l'écran et y afficher notre position. Pour cela, nous devons déclarer 2 variables `_initialCameraPosition` et `controller`

Nous pouvons ensuite initialiser l'`_initialCameraPosition` et la définir sur l'emplacement actuel de l'utilisateur. De même, le contrôleur est défini sur le contrôleur retourné à l'intérieur de la fonction `_onMapCreated()`.

Enfin, nous modifions le code flutter pour afficher la carte à l'écran à l'aide du `MapboxMap` widget, et voilà, nous avons l'emplacement de l'utilisateur sur la carte. Nous avons également ajouté un bouton d'action flottant en bas à droite, qui re-dirige l'utilisateur vers l'emplacement actuel.

### 5.4.8 Utiliser l'API Mapbox Directions

Dans cette section, regardons l'API Direction, sa réponse et comment nous l'utilisons dans notre application. Nous pouvons voir la réponse de l'API dans le Mapbox Playground. Dans la réponse, 3 choses sont importantes pour nous. Ils sont l'objet **geometry** qui représente les coordonnées à utiliser dans le calque de ligne, **la distance** et **la durée**. Nous en prenons note et ajoutons les lignes suivantes dans le fichier `splash.dart` à l'intérieur de la fonction `initState()`.

Enfin, à titre de vérification, nous redémarrons l'application et affichons la distance réelle obtenue à partir de l'API dans le fichier `plage Table`

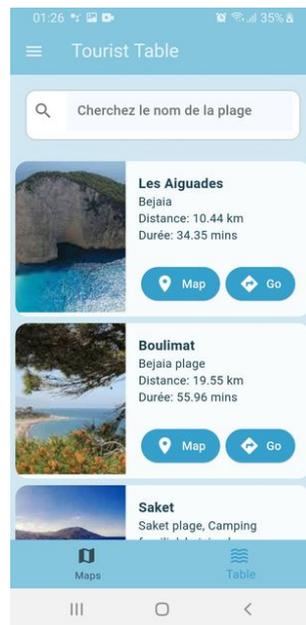


FIGURE 5.7 – La table des places touristique

#### 5.4.9 Empilez un Carousel Slider sur la carte

Un curseur de carrousel composé de cartes de places touristiques peut être empilé en haut de la carte. L'idée serait d'afficher un itinéraire particulier chaque fois qu'un élément spécifique est mis en sur-brillance. Nous trierions également les emplacements en fonction du temps mis pour les atteindre (le chemin le plus court selon le trafic et la durée).

#### 5.4.10 Ajouter des symboles et des couches de lignes

Nous pouvons enfin ajouter des symboles et des couches de lignes sur la carte pour représenter les itinéraires. Pour cela, nous utiliserons les méthodes `addSymbol` et `addLineLayer` sur l'objet contrôleur. Nous veillerons également à supprimer les couches de lignes précédentes afin de ne pas avoir plusieurs couches.

Maintenant, nous pouvons exécuter notre application et après un redémarrage à chaud, voici les derniers écrans que nous pouvons voir 5.8 :

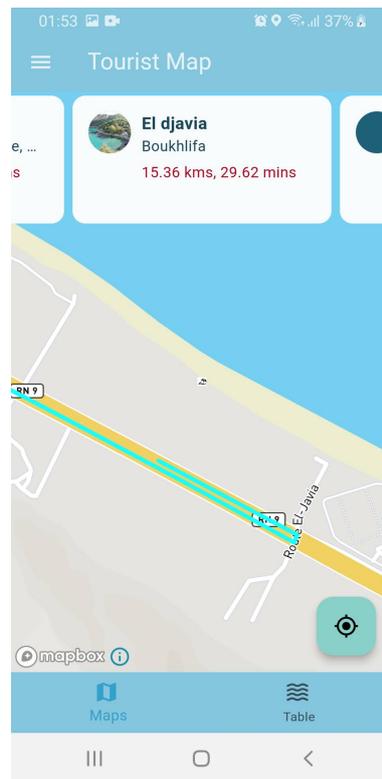


FIGURE 5.8 – Afficher l'itinéraire et le carrousel

#### 5.4.11 SDK de navigation étape par étape

La navigation étape par étape, également appelée guidage actif, est probablement ce qui nous vient à l'esprit lorsque nous pensons à une expérience de navigation typique : un utilisateur reçoit des instructions étape par étape au fur et à mesure qu'il progresse le long de l'itinéraire. Si une session de trajet est en cours et qu'un itinéraire est sélectionné, le SDK de navigation passe en état interne active guidance [27].

Le SDK de navigation nous permet de créer une expérience de navigation complète grâce à la puissance de l' API Mapbox Directions[27]. Nous allons créer une interface utilisateur de navigation étape par étape complète dans notre application à l'aide des composants prédéfinis ou utiliser directement la logique de base pour créer quelque chose de vraiment personnalisé.

Le SDK de navigation fournit un ensemble de fonctionnalités essentielles lors de la création de projets de navigation [27], notamment :

- Itinéraires de navigation.
- Mises à jour précises de l'emplacement de l'appareil.
- Annonces d'instructions vocales.
- Progression de l'utilisateur en temps réel vers sa destination
- Détection hors route et ré-acheminement.
- Réglages dynamiques de la caméra pendant la navigation pas à pas
- Composants d'interface utilisateur prédéfinis pour afficher les informations de navigation changeantes.
- Prise en charge de la navigation avec une connexion réseau limitée ou inexistante.



FIGURE 5.9 – Turn by turn navigation

Avant de commencer à développer notre application avec le SDK de navigation, nous devons configurer nos informations d'identification et ajouter le SDK en tant que dépendance.

Après la configuration, nous devons déclarer une instance `MapBoxNavigation _directions` et l'initialiser à l'intérieur de la fonction `initState()` dans une Classe `TurnByTurn`, en ce qui suit nous allons définir les options d'itinéraire dans l'instance `MapBoxOptions _options`. La fonction `_onEmbeddedRouteEvent` nous permet d'écouter les événements de la Route ( c.à.d la progression de l'utilisateur en cours de la route). Enfin, nous devons déclarer les `WayPoint` pour pouvoir commencer à naviguer.

Cette dernière est une excellente fonctionnalité disponible dans Mapbox. Nous pouvons créer une gamme complète de produits, des solutions entièrement personnalisées à la navigation dans l'application avec des valeurs par défaut intelligentes en quelques minutes. Nous avons couvert la plupart des fonctionnalités de base disponibles dans Mapbox.

## 5.5 Problèmes rencontrés

Rencontrer des problèmes, lors de la réalisation de quelconque projet est inévitable. Quant à la gravité de ces derniers elle varie d'un problème à un autre. Il existe celui qui prend des heures, d'autres des jours voir des semaines et même d'autres des mois et des semaines afin d'être résolu. Mais, dans les projets qui ont des délais bien déterminés il faut bien trouver une solution intermédiaire le temps que la résolution de ce dernier soit possible pour ensuite faire la mise à jour. Pendant la conception de notre projet, nous avons fait face à certains problèmes, les plus pertinents sont les suivants :

- Avant le 30/05/2022, il était possible d'autorisé notre application à se connecter au

compte Gmail grâce du package mailer en lui donnant seulement l'email et le mot de passe, sauf que Google a désactivé cette option partir de cette date d'où nous étions obligées de trouvé une alternative à l'aide StackOverflow [7].

- Les coordonnées géographiques étaient compliquées à récupérer pour la ville de Bejaia, cela suite au manque de data set. Nous étions dans l'obligation de récupérer ces données manuellement.

- L'incapacité d'exploité Google Maps et ses données, car cette dernière est payante, d'où après documentation et recherche nous avons opté à utiliser Mapbox et ces avantages.

## 5.6 Tests et Résultats

### 5.6.1 Inscription

Lorsque l'utilisateur s'inscrit sur la page d'inscription 5.10, les informations sont stockées automatiquement dans la base de données 5.11 :

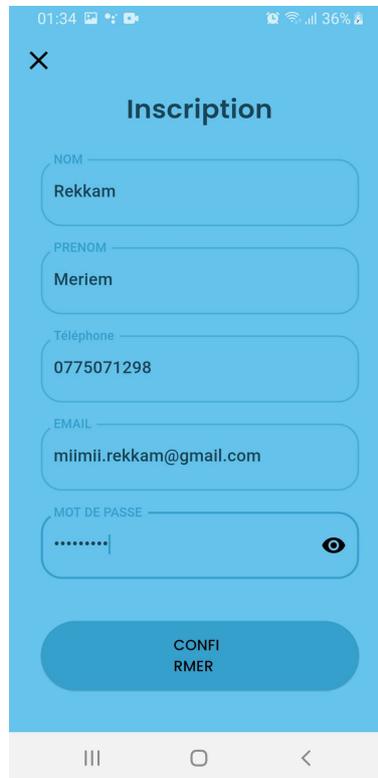


FIGURE 5.10 – Inscription

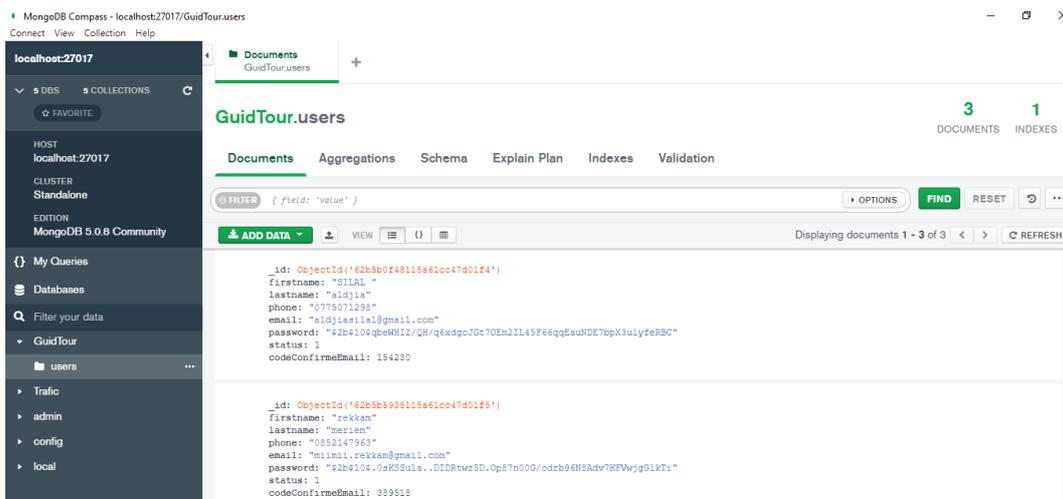


FIGURE 5.11 – Les utilisateur dans la BDD

### 5.6.2 Connexion

Pour que l'utilisateur puisse se connecter directement 5.12, il se retrouve dans l'obligation d'avoir un compte, puis par la suite confirmer son identité avec son code 5.13 pour enfin accéder à l'interface de l'application, qui s'agit de la carte et sa position sur cette dernière :



FIGURE 5.12 – Connexion



FIGURE 5.13 – Confirmation email

### 5.6.3 Mot de passe Oublié

Dans le cas où l'utilisateur oublie son mot de passe 5.14, il existe le code de récupération grâce à un courrier électronique envoyé par GuidTour 5.16 :



FIGURE 5.14 – Mot de passe incorrecte



FIGURE 5.15 – Récupération de mot de passe

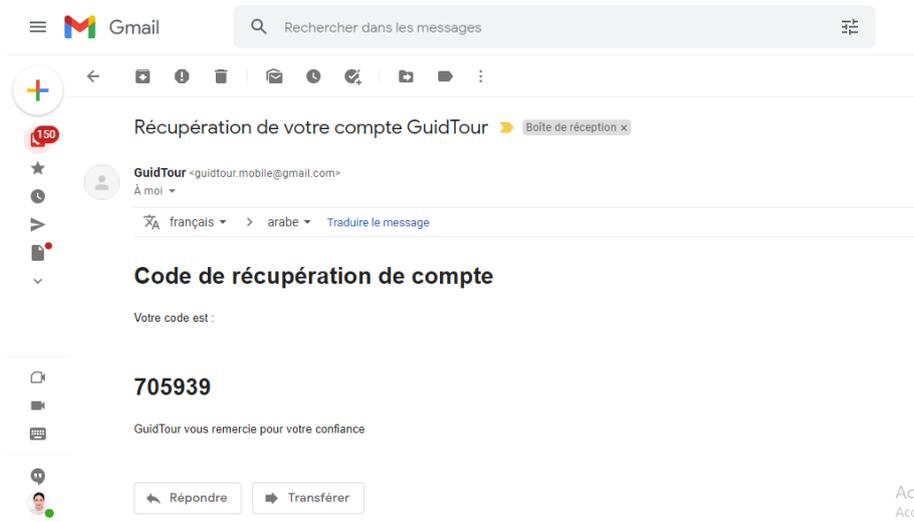


FIGURE 5.16 – Code de récupération



FIGURE 5.17 – Nouveau mot de passe

### 5.6.4 Itinéraire et guidage

Quand l'utilisateur sélectionne la destination en premier lieu, l'utilisateur peut appuyer sur la carte il affiche sa position et la localisation du lieu , ainsi que les données de temps et de distance et l'heure d'arrivé 5.20 :

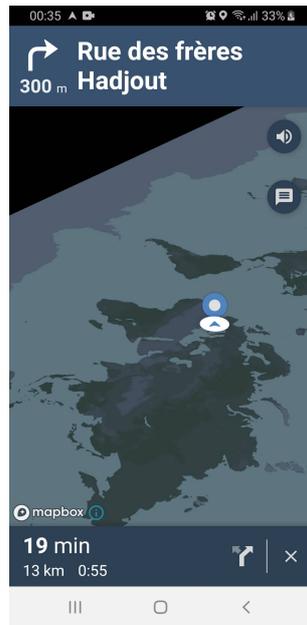


FIGURE 5.18 – Lieu et localisation

Comme il peut choisir en appuyant sur le bouton de navigation, le turn by turn :

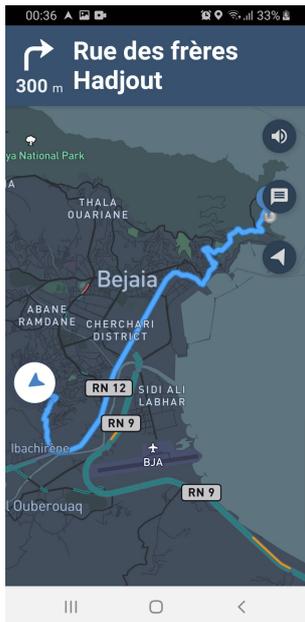


FIGURE 5.19 – Itinéraire à suivre



FIGURE 5.20 – navigation étape par étape

L'utilisateur a la capacité de signaler certains problèmes rencontrés sur le trajet 5.21, mais malheureusement ce n'est pas encore activé :

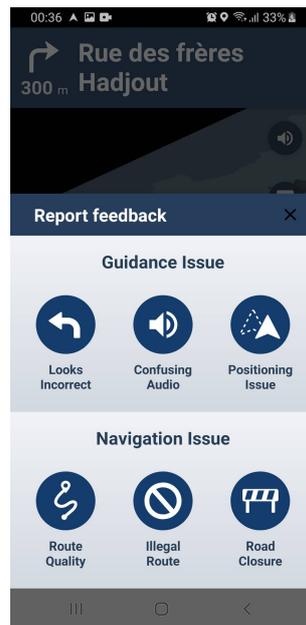


FIGURE 5.21 – Rapport et signalisation

### 5.6.5 Recherche

L'utilisateur a l'opportunité d'utilisé l'option recherche pour les lieux 5.22 :

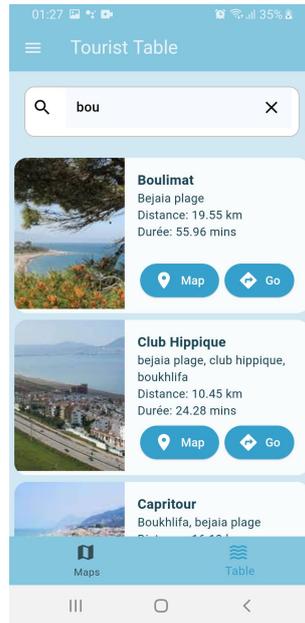


FIGURE 5.22 – Recherche

### 5.6.6 Profil et paramètre

L'utilisateur a accès à son profil, se déconnecter et un peu de paramètres de réglages 5.23 , cette partie est à mettre à jour :

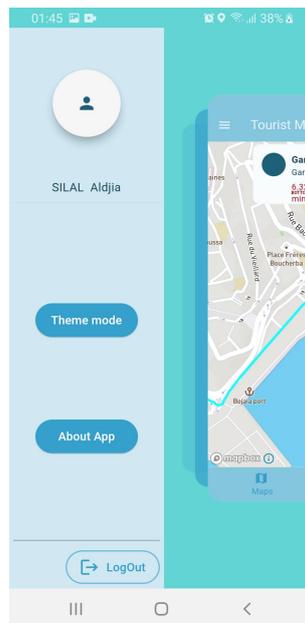


FIGURE 5.23 – Profil et paramètre

## 5.7 Contribution à l'amélioration de GuidTour

### 5.7.1 Contribution 1

Le manque de données sur notre ville nous a poussés à réfléchir à une nouvelle solution qui pourra être implémenté dans notre application dans l'objectif de mettre fin à ce problème avec l'aide de diverses techniques dont l'intelligence artificielle et le big data. Premièrement accorder à l'utilisateur la capacité de signaler et rapporté diverses situations sur son trajet, ainsi des données seront récoltées dans notre base, afin que ces dernières puisse être analysées, cela dans le but d'apporter des améliorations à l'application et à sa gestion. En second lieu, l'historique de recherche pourrai être d'une grande aide, afin d'avoir la possibilité de faire des suggestions ciblés vis à vis des envies des utilisateurs. De plus nous voulons permettre aux usagers de suggérer l'addition de lieu non existant dans la carte, ainsi avoir plus de données dans notre base de données. Et finalement nous voulons ajouter un questionnaire afin d'avoir le ressenti de nos clients. Toutes ces options seront faites grâce à l'emploi du machine learning et le deep learning avec un algorithme bien définit.

Le machine learning/ Apprentissage automatique est une méthode de l'IA, employé souvent pour la prédiction et l'analyse de données. Elle consiste à laisser des algorithmes découvrir des " patterns ", à savoir des motifs récurrents, dans les ensembles de données. Quant au deep learning ou l'apprentissage profond il s'agit d'une sous-catégorie de machine learning qui est autonome donc en d'autre terme la machine apprend d'elle-même.

Malheureusement, ces options non pas encore été ajouté à l'application faute de temps, mais nous travaillons afin de les avoir le plus tôt possible.

Comme nous avons exploité MapBox, celle-ci travaille avec les techniques et technologies d'IA et de Big data pour le calcul du meilleur chemin et la récolte des données du trafic. De plus grâce à cette dernière notre application propose un meilleur chemin régulièrement suivant la mise à jour de l'état du trafic.

### 5.7.2 Contribution 2

La majorité des gens aiment se retrouver dans des lieux très peu peuplé, pour réaliser cette option nous allons exploiter l'internet des objets.

Comme la figure 5.24 l'illustre, à chaque entrée de plage nous aurons un réseau de capteurs qui communique le nombre de places disponibles au sein du parking, cela en étant connecté à une station de base propre à lui, qui elle par la suite sera connectée au cloud. Nous aurons aussi tous les véhicules qui accèdent à la plage ou circulent sur la route principale seront eux aussi connectés aux stations de bases dans le cas où il n'est pas possible de joindre le cloud directement. Puis ensuite le serveur GuidTour qui est connecté au cloud afin de récupérer les données communiquées par les stations de bases et les véhicules. Par la suite, ces informations seront affichées sous forme de message «Les plages les moins peuplées sur Bejaia sont : » sur l'interface pour l'utilisateur.

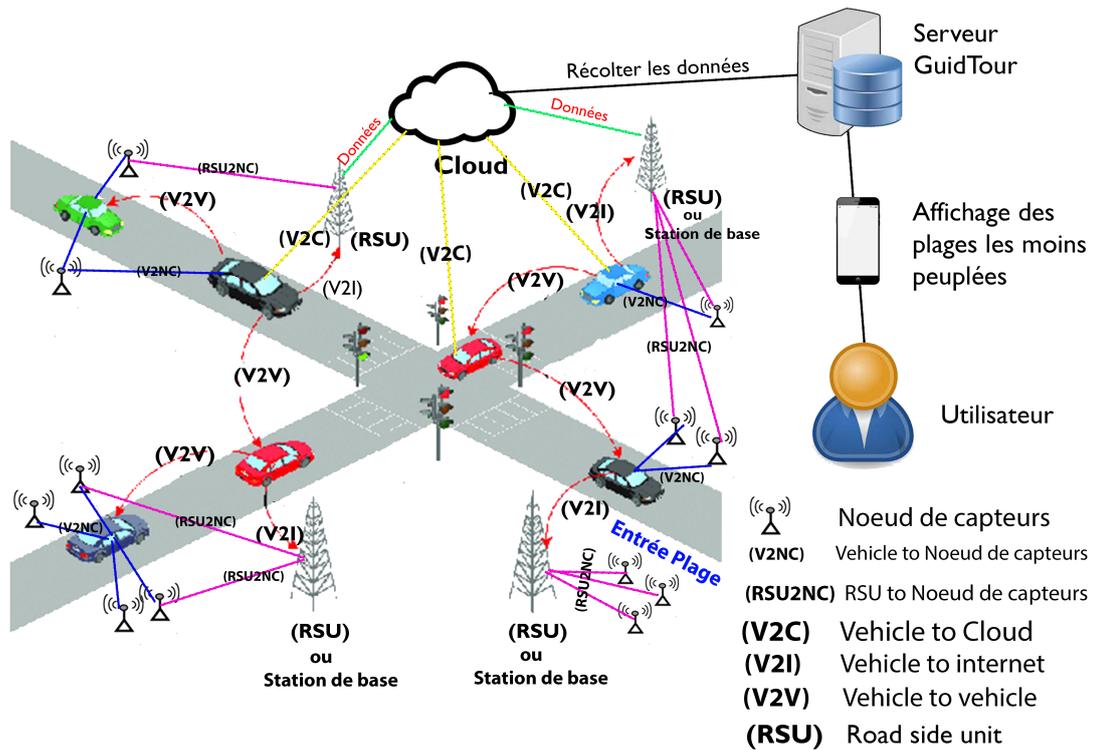


FIGURE 5.24 – Architecture réseau sans fil

## 5.8 Comparaison entre notre contribution et les propositions

Les auteurs de l'article ont présenté uniquement une application de guidage routier qui a pour but d'éviter les embouteillages en prenant en compte la préférence des conducteurs par rapport à la distance, le temps et la sécurité au niveau de la ville de Manchester, donc celle-ci signal un accident et propose un chemin moins encombrant a l'utilisateur. Notre contribution quant à elle est divisée en deux partie, une application de guidage routier pour la ville de Bejaia visant un public bien précis « les touristes ou toute personne souhaitant se divertir», celle-ci met en avant les lieux touristiques de notre ville et propose le plus court chemin qui prend le moins de temps possible. La deuxième partie s'agit de permettre à nos utilisateurs de se connecter entre eux donc, savoir le nombre de véhicules sur la route et le nombre de place de parking disponible etc.

Les deux propositions ont en commun l'utilisation fluteur comme interface, nous nous sommes inspirées de leurs propositions pour la personnalisé par la suite pour notre ville. Et pour la deuxième partie de notre contribution nous nous sommes inspirés d'un second article dont la solution s'agit d'un algorithme de meilleurs chemins et qui emploie l'IoV.

## 5.9 Conclusion

Dans ce chapitre, nous avons tout d'abord fait une description générale, présenter l'architecture de l'application, parler de la réalisation et développement de cette dernière, de la navigation et localisation, les problèmes rencontré et puis fait des tests et exposé les résultats

de ces derniers. Nous avons proposé des améliorations à adopter dans GuidTour pour mettre fin aux problèmes existants.

# Conclusion Générale

Le guidage routier est un domaine de recherche récent et très intéressant pour porter des solutions pour la gestion du trafic qui est en croissance constante dans le monde entier. De plus, avec les nouveaux matériels et outils mis à la disposition du développeur, la conception et le développement d'application en tout genre est devenue moins compliqué et très répondu.

Ce projet a eu pour objectif d'étudier la gestion du trafic dans les villes intelligentes et les solutions existantes dans la littérature pour le guidage routier. Malheureusement, en Algérie nous remarquons que très peu d'application de ce genre existe et plus particulièrement, nous avons noté que la ville de Béjaia souffre du problème du trafic surtout pendant l'été malgré que ses lieux touristiques ne sont pas mis en avant. Cette étude et ces remarques concernant notre ville nous a motivé à proposer une application de guidage routier dédiée pour les touristes de la ville de Béjaia en Algérie. L'objectif de cette dernière est de mettre en avant les lieux touristiques de notre ville ainsi que de proposer le chemin qui prend le moins de temps possible aux utilisateurs et cela en se basant sur l'état du trafic, tout cela, avec guidage de la source vers la destination.

Le développement de cette application nous a permis d'adapter les applications de guidage en général aux touristes de notre villes Béjaia et de penser à des solutions pour l'amélioration de ces dernières. Ainsi, à la fin de ce projet, nous donnons deux importantes contributions à l'amélioration du guidage routier dédiés pour les touristes. Nous espérons qu'elles peuvent être évaluées par d'autres étudiants qui viendront compléter ce travail.

Pour l'évolution de ce projet, il existe de nombreuses perspectives, et les plus essentielles sont :

- La possibilité d'utiliser sans internet
- Historique de recherche
- Avoir un coté admin pour la gestion
- Plus de paramètres
- Permettre à l'utilisateur de modifier certaines de ses informations
- Toucher d'autres régions du pays.
- Activer le rapport et signalisation
- Permettre à l'utilisateur de suggérer l'addition de lieux

Ce projet a été très enrichissant et cela par les divers connaissances qu'il nous a permis d'acquérir, d'une part, par la diversité des technologies auxquels nous nous sommes intéressées et d'autre part, nous avons découvert un nouveau domaine qui s'agit de la recherche ainsi que sa difficulté ce qui nous permis de développer de nouvelles compétences. De plus, nous nous

sommes rendues à l'évidence des compétences que nous avons acquises dans nos cinq années d'études acharnées.

# Bibliographie

- [1] Association mondiale de la route / world road association. *AIPCR Manuel sur les systèmes de transport intelligents (STI) (seconde édition).*, Anglaise : ISBN 2-84060- 174-5 Route 2 Market Ltd. Française : ISBN 2-84060-188-5 AIPCR Secrétariat. Paris,2003.
- [2] [://blog.lesjeudis.com/10-applications-de-l-internet-des-objets-quirevolutionnent-la](http://blog.lesjeudis.com/10-applications-de-l-internet-des-objets-quirevolutionnent-la) (consulté le 01/12/2021).
- [3] [://qualite829.wordpress.com/2017/10/23/internet-des-objets-internet-ofthings/](http://qualite829.wordpress.com/2017/10/23/internet-des-objets-internet-ofthings/), (consulté le 01/12/2021).
- [4] <http://www.wbi.be/fr/page/quatre-risques-importants-villes-intelligentes>, (consulté le 02/12/2021).
- [5] [https://www.researchgate.net/figure/Schema-des-six-leviers-dune-ville-intelligente-fig4\\_333903401](https://www.researchgate.net/figure/Schema-des-six-leviers-dune-ville-intelligente-fig4_333903401), (consulté le 11/05/2021).
- [6] <https://www.smartcity-bern.ch/ueber-uns/smart-city-wheel/>, (consulté le 11/05/2021).
- [7] <https://stackoverflow.com>, (consulté le 11/06/2022).
- [8] <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11559&lang=fr>, (consulté le 12/12/2021).
- [9] <https://www.next-decision.fr>,<https://datascientest.com>, (consulté le 14/06/2022).
- [10] <https://edutechwiki.unige.ch>, (consulté le 14/06/2022).
- [11] <https://blog.ippon.fr>,<https://www.mapbox.com>+<https://fr.wikipedia.org>, (consulté le 14/06/2022).
- [12] [www.nicelydev.com](http://www.nicelydev.com),<https://edutechwiki.unige.ch>, (consulté le 14/06/2022).
- [13] <https://www.lebigdata.fr>, (consulté le 14/06/2022).
- [14] <http://www.embarcadero.com>, (consulté le 14/06/2022).
- [15] <https://www.ionos.fr>,<https://www.numendo.com>, (consulté le 14/06/2022).
- [16] <https://www.codeur.com>, (consulté le 15/06/2022).
- [17] <https://www.encyclopedie.fr>, (consulté le 15/06/2022).
- [18] <https://www.ionos.fr>+<https://www.numendo.com>, (consulté le 16/06/2022).
- [19] <https://www.jesuisundev.com>, (consulté le 16/06/2022).
- [20] <https://www.lebigdata>., (consulté le 16/06/2022).
- [21] <https://hackersandslackers.com>, (consulté le 19/06/2022).

- [22] TITLE = Réalisation d'une application Android de navigation par GPS en utilisant des cartes OSM, <https://www.techno-science.net>, (consulté le 20/06/2022).
- [23] <https://www.algeriaentreprise.com>, (consulté le 20/06/2022).
- [24] <https://fr.wikipedia.org>, (consulté le 20/06/2022).
- [25] <https://edutechwiki.unige.ch>, (consulté le 20/06/2022).
- [26] <https://blog.prototypr.iopic>, (consulté le 20/06/2022).
- [27] <https://docs.mapbox.com/>, (consulté le 20/06/2022).
- [28] <https://modellium.com/nouvelles/definition-ville-intelligente/>, Consulté le (20/11/2021).
- [29] <https://steknol.com/internet-des-objets-iot-pour-les-ville-intelligentes/>, (consulté le 22/12/2021).
- [30] The smart city model in european smart cities. [.http://www.smartcities.eu/model.html](http://www.smartcities.eu/model.html), (consulté le 26/11/2021).
- [31] Smart city wheel in boyd cohen smart cities. urban and climate strategist boyd cohen. <http://www.boydcohen.com/smartcities.html>, (consulté le 26/11/2021).
- [32] Villes intelligentes. <https://gfi.world/uploads/media/Article/0001/04/fd355df86a8f1c0a56b3962303352847ab8e7fdf.pdf>, (consulté le 26/11/2021).
- [33] <https://www.domotique-tunisie.com/domotique-tunisie-les-objets-connectes-IoT.html>, (consulté le 26/12/2021).
- [34] <https://www.techtarget.com/whatis/definition/Internet-of-Vehicles>, (consulté le 27/06/2022).
- [35] Aidée carrière. *La gestion en temps réel d'un feu de circulation dans le contexte des systèmes de transport intelligents*, Novembre 2005, Mémoire de maîtrise ès sciences.
- [36] A. R. Andreo Pimpinella, Matteo Cesana. Walk this way! an iot-based urban routing system for smart cities. <https://www.researchgate.net/publication/334440815>, Juillet 2019.
- [37] S. B. et Jeremy Diaz. La ville intelligente origine, définitions, forces et limites d'une expression polysémique. <http://espace.inrs.ca/4917/1/Rapport-LaVilleIntelligente.pdf>, (consulté le 28/11/2021).
- [38] H. R. S. et Nouari Djahida. Feux tricolores intelligents pour les villes intelligentes. *Université Akli Mohand Oulhadj de Bouira*, 2018-2019.
- [39] D. R. et R.D. Bretherton. Optimizing networks of traffic signals in real time-the scoot method. *IEEE Transactions on Vehicular Technology* 40.1, février 1991.
- [40] D. Evans. *L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?*, Avril 2011.
- [41] Kai Lin, Chensi Li, Yihui Li, Claudio Savaglio and Giancarlo Fortino. Distributed learning for vehicle routing decision in software defined internet of vehicles. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, (2020).
- [42] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik. Predictive trip planning - smart routing in smart cities. In *Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference (EDBT/ICDT 2014), Athens, Greece, March 28, 2014*, volume 1133, pages 331–338. CEUR-WS.org, 2014.

- [43] O. RIVATION. Le routage de l'information dans les réseaux véhiculaires mobiles, year=2016,.
- [44] B. Roxin. Ecosystème de l'internet des objets, dans bouhai net saleh i, (dir) "*Internet des objets : Evaluations et Innovations*", ISTE Editions Londres, Mai 2017.
- [45] I. Saleh. *Internet des Objets (IdO) : Concepts, Enjeux, Défis et Perspectives*. //www.openscience.fr/IMG/pdf/isteido18v2n11.pdf, (Consulté le 20/12/2021).
- [46] J. Simard. La ville intelligente comme vecteur pour le développement durable. <https://www.usherbrooke.ca/environnement/fileadmin/sites/environnement/documents/Essais2015/SimardJoelleMEnv2015.pdf>, (consulté le 02/12/2021).
- [47] S. D. Sophie Hayes, Shen Wang. Personalized road networks routing with road safety consideration a case study in manchester. 2020.
- [48] M. Tlig. Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier. *Ecole doctorale IAEM Lorraine*, (26 mars 2015).
- [49] J. Vi.tran-ngoc nha, Soufiene.djahel. A comparative study of vehicles routing algorithms for route planning in smart cities. *LERO*, (2012).