

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



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

**Mémoire de Fin d'Etudes  
de MASTER ACADEMIQUE**  
Domaine : **Mathématiques et Informatique**  
Filière : **Informatique**  
Spécialité : **Système informatique**

*Présenté par*  
**BOUTERFA Sarah**  
**HADID Zahia**

**Thème**

***Conception et réalisation d'une application mobile  
multiplateformes pour le transport routier de  
personnes et de marchandises en temps réel.***

*Mémoire soutenu publiquement le 14/07/ 2016 devant le jury composé de :*

**Présidente : Mme DJAMAH**

**Encadreur : Mme AOUDJIT**

**Examinatrice : Mme HADAOUI**

**Examinatrice : Mme BALKADI**

***Promotion 2015/2016***

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



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

**Mémoire de Fin d'Etudes  
de MASTER ACADEMIQUE**  
Domaine : **Mathématiques et Informatique**  
Filière : **Informatique**  
Spécialité : **Système informatique**

*Présenté par*  
**BOUTERFA Sarah**  
**HADID Zahia**

**Thème**

***Conception et réalisation d'une application mobile  
multiplateformes pour le transport routier de  
personnes et de marchandises en temps réel.***

*Mémoire soutenu publiquement le 14/07/ 2016 devant le jury composé de :*

**Présidente : Mme DJAMAH**

**Encadreur : Mme AOUDJIT**

**Examinatrice : Mme HADAOUI**

**Examinatrice : Mme BALKADI**

***Promotion 2015/2016***

## ❧ Remerciements ❧

*Nous tenons à exprimer notre profonde gratitude à notre promotrice, madame AOUDJIT Rachida pour nous avoir encadrées durant cette année, ainsi que pour ses conseils judicieux.*

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

*Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que ce modeste travail voit le jour.*

## ❧ Dédicaces ❧

*Je dédie ce modeste travail, à tous ceux et toutes celles qui m'ont accompagné et soutenu durant cette année de formation, citer chacune d'elles m'est impossible, je tiens néanmoins à remercier particulièrement :*

*Madame R. Aoudjit, qui durant la période de travail nous a toujours encouragé et nous a apporté son aide précieuse.*

*Ma très chère Nadia Feghoul, sans qui je ne serais probablement pas là aujourd'hui, lui monter toute ma reconnaissance et ma gratitude m'est impossible.*

*A mon père et à ma mère pour leurs encouragements et leur soutien dans mes périodes les plus difficiles. Je remercie également mes sœurs et mon frère qui ont toujours été là pour moi quand j'en avais besoin. Ainsi qu'à ma cousine Dyna que j'aime plus que tout.*

*A ma binôme Zahia qui a su être patiente dans des périodes difficiles.*

*Enfin à mes camarades du département et à mes amis : Tassadit, Nacera, Amir, Kenza Et en particulier à Amar.*

*Sarah.*

## ❧ Dédicaces ❧

*Je dédie ce modeste travail à mes chers parents en témoignage de leurs sacrifices et leurs attentions durant toute ma vie ainsi leurs encouragements et soutiens tout au long de mes études.*

*A la personne qui m'a toujours encouragée et soutenue, ma source de bonheurs, à mon futur marri Nacim.*

*A mes sœurs chéries Nassima et Hassina.*

*A mes frères Meziane Mourad Nacer et Lyes.*

*A mes beaux parents et mes chères belles sœurs Ouiza et Aida.*

*A celle qui sera fière de moi ma grande famille.*

*A mes chères copines Amina et lynda.*

*A mon binôme Sarah et toute sa famille.*

*A tous ceux qui me connaissent.*

*A toute la promotion 2015/2016.*

**Zahia.**

---

# Sommaire

---

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

## **Chapitre I : Les smart cities et les transports intelligents**

I. Introduction.....	3
II. Définitions.....	3
II.1. Définition d'une ville intelligente.....	3
II.2. Définition d'un système de transport intelligent.....	3
III. Les composantes de la ville intelligente.....	4
- Gouvernance intelligente.....	5
- Citoyen intelligent.....	5
- Economie intelligente.....	5
- Mobilité intelligente.....	5
- Environnement intelligent.....	6
- Habitat intelligent.....	6
IV. Exemple d'applications pour la création d'une ville intelligente.....	6
- Mobilité et déplacements.....	7
- Gestion hydraulique.....	7
- Gestion de l'énergie et de l'entretien.....	8
- Capteurs environnementaux.....	8
- Accès sans fil.....	8
V. Les technologies STI.....	9
- Capteurs.....	9
- Diffusion de l'information.....	9
- Terminaux interactifs.....	9
- Diffusion passive (broadcast).....	10
- Localisation.....	10
- Les systèmes coopératifs.....	10
- Transmission.....	11
VI. Objectifs des politiques STI.....	11
VII. Avantages des STI.....	12

VIII. Champ des STI .....	12
- Sécurité routière .....	12
- Aide à la mobilité .....	12
- Sureté dans les transports .....	13
- Gestion des transports publics .....	13
Conclusion .....	14

## CHAPITRE II : Les applications mobiles

I. Introduction .....	15
II. Définitions .....	15
II.1. Les applications mobiles .....	15
II.1.1. Application iPhone .....	15
II.1.2. Application Android .....	16
III. Typologies .....	16
III.1. Application native .....	16
III.2. Application web ou web app .....	17
III.3. Application hybride .....	17
IV. Avantages et inconvénients des applications mobiles .....	18
IV.1. Avantages .....	18
IV.2. Inconvénients .....	19
V. Les plateformes mobiles .....	19
V.1. iOS .....	19
V.2. Windows Phone .....	19
V.3. Android .....	20
VI. Le marché mondial des applications mobiles .....	20
VII. Exemple d'applications destinées au transport .....	22
Conclusion .....	23

## CHAPITRE III : Analyse et Conception

I. Introduction .....	24
II. Objectif de notre application .....	24
III. Première étape : Analyse .....	25

III.1. Identification des acteurs .....	25
III.2. Les cas d'utilisation .....	25
III.2.1. Cas d'utilisation : création d'un compte utilisateur .....	26
III.2.2. Cas d'utilisation : création d'un compte transporteur .....	26
III.2.3. Cas d'utilisation : connexion au compte .....	27
III.2.4. Cas d'utilisation : demande d'un service de transport .....	27
III.2.5. Cas d'utilisation : réponse de l'utilisateur .....	27
III.3. Les diagrammes de séquence.....	29
III.3.1. Diagramme de séquence : création d'un compte .....	30
III.3.2. Diagramme de séquence : connexion au compte .....	30
III.3.3. Diagramme de séquence : demande d'un service de transport .....	31
III.3.4. Diagramme de séquence : réponse de l'utilisateur .....	31
III.4. Les diagrammes d'activité.....	31
III.4.1. Diagramme d'activité: création d'un compte .....	32
III.4.2. Diagramme d'activité : connexion au compte .....	33
III.4.3. Diagramme d'activité : demande d'un service de transport .....	33
III.4.4. Diagramme d'activité : réponse de l'utilisateur .....	34
IV. Deuxième étape : Conception .....	34
IV.1. Diagramme de classe .....	34
IV.2. Le modèle relationnel .....	35
IV.3. Codification .....	35
Conclusion .....	38

## CHAPITRE IV : Réalisation

I. Introduction .....	39
II. Environnement de développement.....	39
II.1. Environnement matériel.....	39
II.2. Environnement logiciel.....	39
III. Présentation du Framework ionic .....	39
III.1. Définition.....	39
III.2. Installation .....	39
III.3. Génération du squelette de l'application ionic .....	40

III.4. Déployer une application ionic .....	41
III.5. Tester une application ionic.....	42
III.6. Les outils utilisés par ionic .....	42
III.6.1. HTML (HyperText Markup Language).....	42
III.6.2. CSS (Cascading Style Sheets) .....	42
III.6.3. JavaScript.....	43
III.6.4. Angular JS.....	43
- Définition.....	43
- Concepts d'angular JS .....	44
- Architecture MVC .....	44
- Data binding bidirectionnel .....	45
- Injection de dépendance .....	45
- Routing (Routage).....	45
- Les tests .....	46
III.6.5. Cordova.....	46
- Définition.....	46
- Installation de Cordova.....	46
- Le fonctionnement de la technologie Apache cordova.....	46
1. Rédaction du code source.....	46
2. Compilation du code source.....	46
3. Génération d'une ou plusieurs applications mobiles.....	47
- Les plugins.....	48
IV. Présentation de Node.js .....	48
IV.1. Définition.....	48
IV.2. Fonctionnement de Node.js.....	49
IV.3. Pourquoi node.js est-il rapide ? .....	50
- Le moteur V8.....	50
- Le modèle non bloquant .....	50
IV.4. Installation de node.js.....	52
IV.5. Configurer un serveur avec node.js .....	53
IV.6. Node Package Manager (NPM).....	54
- Installer un module .....	54

- Mettre à jour les modules .....	55
- Les modules utilisés dans notre projet.....	55
➤ HTTP.....	55
➤ Express.....	55
➤ Cookie.....	56
➤ Socket.io.....	57
➤ Express-session.....	58
➤ body-Parser.....	59
➤ crypto.....	59
V. Présentation du langage de requête MYSQL .....	60
VI.1. Définition de MYSQL.....	60
1. Base de données.....	60
2. Système de gestion de base de données (SGBD) .....	60
3. Système de gestion de base de données relationnel (SGBDR) .....	61
IV.2. Installation de MYSQL.....	61
IV.3. Etablissement de la connexion avec la base de données MYSQL.....	61
VI. Présentation de quelques interfaces.....	62
VI.1. Page d'accueil.....	62
VI.2. Page d'authentification.....	63
VI.3. Page d'inscription de l'utilisateur.....	64
VI.4. Page d'inscription du transporteur.....	65
VI.5. Compte transporteur.....	66
VI.6. Compte utilisateur .....	67
Conclusion.....	68
Conclusion générale.....	69
Références bibliographiques.....	70

---

# Liste des figures

---

## **Chapitre I :**

Figure I.1 : Schéma des six leviers d'une ville intelligente.....	4
---	---

## **Chapitre II :**

Figure II.1 : Descriptif des principaux OS mobiles présents sur le marché .....	20
Figure II.2 : Les applications mobiles – Prévision annuelle du revenu 2015 vs 2020 par catégorie .....	21
Figure II.3 : Les applications mobiles – téléchargements annuelles 2015 vs 2020 par catégorie .....	21
Figure II.4 : Description des quelques applications destinées au transport .....	23

## **Chapitre III :**

Figure III.1 : Les démarches de modélisation de l'application .....	24
Figure III.2 : diagramme de cas d'utilisation « Utilisateur » .....	28
Figure III.3 : diagramme de cas d'utilisation « Transporteur » .....	29
Figure III.4 : diagramme de séquence : création d'un compte .....	30
Figure III.5 : diagramme de séquence : connexion au compte .....	30
Figure III.6 : diagramme de séquence : demande d'un service de transport .....	31
Figure III.7 : diagramme de séquence : réponse de l'utilisateur .....	31
Figure III.8 : diagramme d'activité : création d'un compte .....	32
Figure III.9 : diagramme d'activité : connexion au compte .....	33
Figure III.10 : diagramme d'activité : demande d'un service de transport .....	33
Figure III.11 : diagramme d'activité : Réponse de l'utilisateur .....	34
Figure III.12 : diagramme de classe global .....	35

## **Chapitre IV :**

Figure IV. 1 : Création d'un nouveau projet ionic .....	40
Figure IV.2 : Représentation du design pattern MVC.....	45

Figure IV.3 : Fonctionnement de la technologie Apache cordova .....	47
Figure IV.4 : Le système classique – PHP coté serveur, Javascript coté client .....	49
Figure IV.5 : Avec Node.js – JavaScript coté serveur et coté client .....	50
Figure IV.6 : Le modèle non bloquant en programmation .....	51
Figure IV.7 : La différence entre le modèle bloquant et le modèle non-bloquant .....	51
Figure IV.8: Installation de Node.js sous Windows .....	52
Figure IV.9 : Paramètre par défaut pour l’installation de Node.js sous Windows .....	53
Figure IV.10 : Création d’un serveur Node.js .....	53
Figure IV.11 : Page d’accueil du site de téléchargement des modules NPM.....	54
Figure IV.12 : Exemple d’utilisation de cookie et btoa .....	57
Figure IV.13 : Fonctionnement du web traditionnel .....	57
Figure IV.14 : fonctionnement du web temps réel .....	58
Figure IV.15 : Socket.io – script coté client .....	58
Figure IV.16 : Socket.io – script coté serveur .....	58
Figure IV.17 : Code permettant d’établir la connexion avec la base de données MYSQL ....	62
Figure IV.18 : Teste de la bonne connexion entre node.js et MYSQL .....	62
Figure IV.19 : Interface de la page d’accueil .....	63
Figure IV.20 : Interface d’authentification .....	64
Figure IV.21 : Interface d’inscription de l’utilisateur .....	65
Figure IV.22 : Interface d’inscription du transporteur .....	66
Figure IV.23 : Interface du compte transporteur .....	67
Figure IV.24 : Interface du compte utilisateur .....	68

*Introduction*  
*générale*

Il y a 10 ans, le téléphone mobile était juste un outil de communication pour émettre et recevoir des appels et des messages à partir d'autres mobiles.

Maintenant, avec l'arrivée des nouvelles générations (3ème, 4ème) le terminal mobile a retrouvé un nouvel aspect, il n'est plus ce téléphone simple qui existait avant.

A présent, on ne parle plus de téléphone mais de smartphone, ce qui est plus juste grâce à son intelligence artificielle et un certain degré d'autonomie dont il est doté. Les smartphones permettent l'exécution d'une panoplie d'applications et offrent beaucoup de services. Il est un véritable ordinateur de poche qui permet une connexion avec un ordinateur pour stocker des données, connexion internet, un appareil relativement pratique grâce à son écran tactile avec, bien sûr, toutes les fonctionnalités d'un téléphone portable : enregistrement photo, vidéos, sons, envoi et réception de MMS, baladeur MP3 et radio, appareil photo et caméra vidéo.

L'extension technologique apportée aux smartphones fait de ces derniers un outil indispensable dans notre vie quotidienne. Le développement d'applications (pour smartphone) et les navigateurs mobiles font de l'accès mobile l'un des éléments moteur de toute communication.

Notre application s'inscrit dans le cadre des transports intelligents qui sont des applications ou services avancés associant l'ingénierie des transports, les technologies de la communication, de l'information et du positionnement géographique.

Elle fournit d'une part, un moyen simplifié de déplacement pour les passagers, et d'autre part, un moyen utile pour exercer le métier de chauffeur.

Notre objectif est de proposer une application mobile de mise en relation de personnes désirant partager un véhicule avec d'autres usagers pour effectuer un trajet en commun. L'application propose de rechercher une offre grâce à plusieurs critères et permet de mettre directement les personnes en contact.

Elle est aussi multiplateformes, c'est-à-dire, une application qui fonctionnera sur tous les systèmes d'exploitation mobiles présents sur le marché, destinée au transport de personnes.

Pour vous mettre dans le cadre de réalisation de notre application, nous avons procédé ainsi :

D'abord, puisque notre application s'inscrit dans le domaine du transport intelligent, nous allons dans le premier chapitre présenter les deux concepts de « ville intelligente » et des « transports intelligents ».

Ensuite, le deuxième chapitre permet d'avoir une idée générale sur les applications mobiles, leurs multiples typologies ainsi que leurs avantages et inconvénients.

Le troisième chapitre sera consacré à l'analyse et la conception de notre application, pour cela nous utiliserons le langage de modélisation UML.

Enfin le dernier chapitre portera sur la réalisation proprement dite de notre application. On décrira les outils utilisés et nous présenterons quelques interfaces.

Notre travail s'achèvera par une conclusion générale et des perspectives.

CHAPITRE

*Les smart cities et les transports  
intelligents*

## **I. Introduction :**

Les villes d'aujourd'hui sont bouleversées en profondeur par le numérique, l'introduction des technologies de l'information et de la communication dans les espaces urbains ouvre la voie à de nouvelles fonctionnalités, de nouvelles manières de gérer, de gouverner, de se déplacer et de vivre dans une ville.

L'un des principaux composants qui permet de vivre dans une ville dite « intelligente » est l'adaptation des transports intelligents dans la ville.

Les systèmes de transport intelligents sont un élément essentiel du système et de la politique des transports. Ils couvrent l'ensemble des champs et des activités liés au transport (passagers, air, mer, rail, fluvial, route, transport public, nouvelles mobilités) et contribuent fondamentalement à son efficacité.

## **II. Définitions :**

### **II.1. Définition d'une ville intelligente :**

L'expression « ville intelligente » (en anglais « smart city »), désigne une ville utilisant les technologies de l'information et de la communication (TIC) pour améliorer la situation actuelle d'une ville dans différentes sphères et régler diverses problématiques urbaines [1].

Une ville intelligente est celle qui a su intégrer les TIC à grande échelle dans différents secteurs d'activité afin d'améliorer la vie quotidienne des usagers et des citoyens. De plus, les TIC permettent d'engager un changement de comportement chez les citoyens, mais aussi au sein de l'administration et dans les entreprises vers une croissance plus durable.

### **II.2. Définition d'un système de transport intelligent :**

Les systèmes de transport intelligents (STI) (en anglais *intelligent transportation systems (ITS)*) sont les applications des nouvelles technologies de l'information et de la communication au domaine des transports [2].

Ce sont, par exemple, les centres de contrôle de la circulation, les panneaux à messages variables, la gestion à distance des feux de circulation, les modes de paiement

électroniques (transport collectif, péage, stationnement), les applications mobiles pour téléphones intelligents, etc.

### III. Les composantes de la ville intelligente :

Différents modèles de ville intelligente sont présentés dans la littérature. Les modèles holistiques de **Giffinger** et de **Cohen** sont ceux qui sont le plus souvent utilisés pour démontrer les six composantes de la ville intelligente [3].

Le modèle de ville intelligente présenté ci-dessous, de Rudolf Giffinger, expert en recherche analytique du développement urbain et régional, présente les six leviers à considérer pour devenir une ville intelligente.

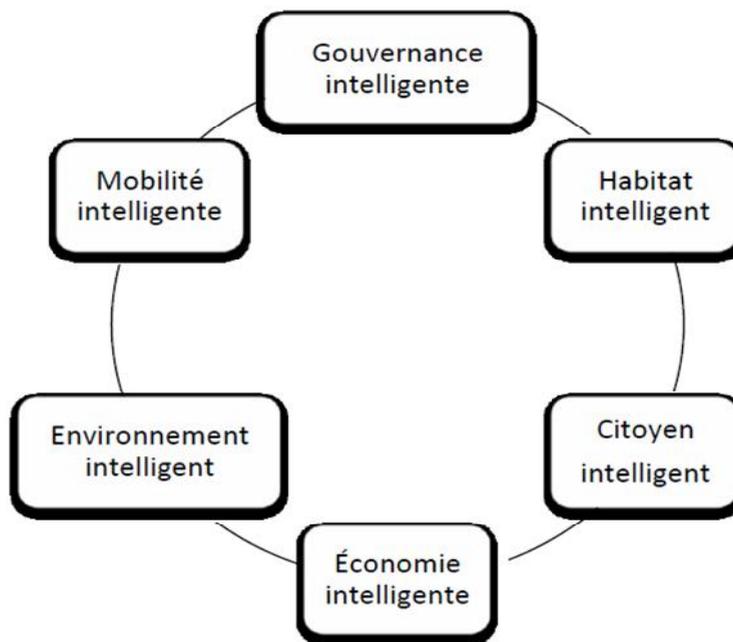


Figure I.1: Schéma des six leviers d'une ville intelligente [3].

Les six composantes de la ville intelligente présentées dans le schéma de ci-dessus sont illustrées de façon non hiérarchique. Cette hiérarchie est en fait une suite logique de processus ou étapes pour lesquels une ville deviendra intelligente. Avant le tout début d'une démarche, il est essentiel d'avoir l'appui de l'instance qui gouverne et par le fait d'avoir une bonne gouvernance qui est adaptée et spécifique à la démarche et aux objectifs visés.

Ensuite, dans une ville intelligente, le citoyen est une partie prenante qui participera de prêt ou de loin à la réussite de cette démarche.

Une fois que les deux dimensions précédentes sont ancrées et bien établies dans la démarche, les quatre autres dimensions sont d'égale importance. C'est-à-dire que l'on doit toutes les considérer si l'on veut devenir une ville intelligente.

➤ **Gouvernance intelligente :**

La gouvernance à l'ère du numérique est collaborative, plus connectée et plus transparente grâce aux outils technologiques. C'est-à-dire que les nouvelles technologies de l'information et des communications servent de levier entre les décideurs, les acteurs publics ainsi que les citoyens. Pensons notamment à des tableaux électroniques dans des lieux publics qui peuvent afficher de l'information à l'intention des citoyens ou encore à une diffusion web simultanée des rencontres du conseil pour permettre à un plus grand nombre de personnes d'y assister.

➤ **Citoyen intelligent**

Le citoyen est une importante partie prenante dans la ville intelligente. En effet, sa participation est requise, que ce soit dans la phase de consultation en amont ou pendant la phase de mise en œuvre, comme acteur pour la protection de l'environnement, en matière d'économie ou dans le volet social au sein de sa communauté. Ensuite, le citoyen intelligent est celui qui utilisera les nouveaux outils technologiques, notamment pour participer aux débats publics et à la vie de quartier.

➤ **Economie 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, 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 :**

L'accès aux données de transport en temps réel via des écrans électroniques dans les stations, dans les wagons de métro ou dans les autobus ou encore via les téléphones intelligents personnels permettrait aux usagers de connaître une foule d'informations, c'est-à-

dire, l'état de la circulation sur le réseau routier, le temps d'attente aux arrêts et stations de transport en commun, les pannes et en somme une meilleure gestion des flux urbains.

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**

Dans une ville intelligente, les divers outils technologiques permettent notamment une protection et une préservation de nos ressources naturelles et des milieux naturels, comme par exemple, des capteurs pour détecter les fuites dans des réseaux, des senseurs pour suivre le transport des matières résiduelles ou des capteurs pour mesurer le niveau de pollution de l'air. Il s'agit là de nouvelles technologies qui permettent de fournir une panoplie d'informations en temps réel.

En matière d'énergie, les « smart grids », une technologie informatique des réseaux de distribution d'électricité intelligents, peut optimiser la production et la distribution d'électricité tout en s'ajustant à la demande. Economiser 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, notamment la domotique et la télémétrie comme l'évolution des techniques, matériaux de construction respectant les normes parasismiques, conservation de la chaleur ou de la fraîcheur, mais aussi l'interconnexion avec les réseaux de sécurité, la télésanté, la télémédecine et l'hospitalisation à domicile.

## **IV. Exemples d'application pour la création d'une ville intelligente :**

Il existe de nombreuses applications pour rendre une ville intelligente, nous présentons ici quelques exemples :

➤ **Mobilité et déplacements :**

Compte tenu des efforts marqués par les administrations pour améliorer les déplacements sur leur territoire, plusieurs capteurs permettent de planifier le trafic routier, les horaires d'autobus, les congestions potentielles selon les activités journalières ainsi que de prévoir les différents impacts qu'une activité, qu'une zone de travaux ou autre auront sur le flux de circulation.

Les nouvelles applications sont en mesure de guider les usagers en temps réel sur la meilleure façon de se mouvoir pour rendre leurs déplacements plus fluides et donc réduire les embouteillages.

De même, les endroits et le nombre de places de stationnement libres pourront être facilement identifiables sur le territoire.

✓ **Technologie :**

Capteurs d'analyse de circulation, capteurs de gestion des places de stationnements, capteurs de déplacement du transport en commun, capteurs et gestion de présence de véhicules aux feux de circulation, panneaux indicateurs interactifs, etc.

➤ **Gestion hydraulique :**

La gestion de l'eau est aussi une source de préoccupation pour plusieurs villes. Les coûts associés au traitement et à la distribution de l'eau sont substantiels. D'où l'importance d'améliorer la performance de la gestion du réseau d'eau potable et d'eaux usées. Le concept est de pouvoir surveiller en temps réel les fuites et les flux d'eau sur l'ensemble du réseau.

✓ **Technologie :**

Capteurs de détection de fuite (comparaison des données, surveillance et alertes en temps réel), compteurs d'eau, capteurs de débordement (surveillance et alertes), surveillance des eaux usées (capteurs ultrasoniques, alertes en temps réel), surveillance de qualité (capteurs de qualité), surveillance de pression (optimisation selon des points de référence, surveillance et alertes en temps réel).

➤ **Gestion de l'énergie et de l'entretien :**

L'énergie consommée par les infrastructures urbaines telles que les luminaires de rue peut représenter jusqu'à 40 % de l'énergie totale consommée par la ville. Les dépenses liées à la planification de l'entretien et à la gestion de ses infrastructures sont également élevées.

Le contrôle de l'éclairage urbain permet à ce jour une multitude de fonctionnalités. Notamment, l'abaissement de puissance des luminaires à des heures précises de la nuit ou en temps réel, le monitoring de la consommation, la détection des pannes en temps réel, le calcul de la durée de vie des luminaires. Une série de capteurs peuvent se greffer au système. Par exemple, des capteurs de luminosité ou environnementaux qui permettent un ajustement automatique des flux lumineux.

✓ **Technologie :**

Compteurs intelligents, systèmes de contrôle pour luminaires urbains (compteurs, détection des pannes, gradation et contrôle de l'intensité lumineuse planifiés ou en temps réel, durée de vie, surveillance et alertes, etc).

➤ **Capteurs environnementaux :**

La qualité de l'air est un sujet d'actualité récurrent. Nous savons que la pollution de l'air a un impact significatif sur la santé. Avec les avancées technologiques, des capteurs simples, mais très efficaces, installés à plusieurs endroits stratégiques peuvent s'intégrer au réseau communicant et fournir plusieurs informations pertinentes aux usagers. Il est ainsi possible de connaître en temps réel les taux de concentration de différentes matières nuisibles et d'ajuster les interventions.

✓ **Technologie :**

Capteurs environnementaux de mesure de la température, capteurs de polluants dans l'air, capteurs de son.

➤ **Accès sans-fil :**

Puisqu'un réseau de communication est déjà déployé sur l'ensemble de la ville pour les besoins identifiés ci-haut, il devient d'autant plus facile de permettre, au choix de la Ville, l'accès Internet gratuit à plusieurs endroits identifiés, comme les parcs et autres endroits publics.

## V. Les technologies STI:

Les technologies relatives aux systèmes de transport intelligents comprennent les technologies électroniques de communication, de traitement et d'analyse des données qui rehaussent la mobilité et la sécurité des transports, véhicules et réseaux de communication.

### ➤ **Capteurs :**

Les technologies de pointe dans le domaine des capteurs ont permis de développer les systèmes de transport intelligent sur la base de données fiables, fréquentes et en grande quantité. Suivant les capteurs, des mesures peuvent être effectuées sur les caractéristiques des véhicules (longueur, poids...), sur la circulation des véhicules (débit, taux d'occupation, vitesse...) ou sur des événements (incidents, files d'attente, franchissement de feux rouges...). Certaines informations sont très utiles dans une optique d'amélioration de la sécurité routière [4].

### ➤ **Diffusion de l'information :**

La diffusion est le pendant du recueil de données. Il s'agit de présenter l'information aux différents types d'utilisateur : grand public (usager, voyageur, internaute, etc.), mais aussi utilisateurs professionnels (conducteurs, agents d'exploitation, etc.).

Cette rubrique renvoie vers les différents produits et systèmes supports de diffusion, quel que soit le canal de diffusion de l'information [5].

La rubrique est structurée par type de support :

#### ✓ **Terminaux interactifs :**

##### • **Informatique fixe :**

L'accès aux applications STI se fait de plus en plus via des terminaux fixes banalisés (PC via le web), néanmoins les bornes interactives restent un support incontournable pour la vente sur le terrain.

##### • **Terminaux mobiles :**

C'est certainement le type de diffusion qui s'est le plus développé ces dernières années, en particulier grâce au succès du téléphone portable et de l'internet mobile.

✓ **Diffusion ‘passive’ (broadcast) :**

• **Voix :**

L’information vocale reste demandée, bien sûr à cause des services d’information téléphoniques, mais aussi des systèmes d’annonces sonores embarqués.

• **Afficheurs :** panneaux à messages variables, afficheurs en station, afficheurs embarqués.

Un des objectifs de bon nombre d’applications STI est de pouvoir diffuser l’information sur de multiples supports, ce qui suppose une certaine normalisation quant à la manière de diffuser cette information vers ces supports.

➤ **Localisation :**

La localisation des voyageurs, des marchandises et des véhicules sont des informations cruciales pour permettre le développement d’une mobilité plus intelligente, mieux optimisée et plus respectueuse de l’environnement.

Les services et technologies permettant de connaître et d’exploiter ces informations se développent très rapidement, avec comme ambition de faciliter l’accès aux positions géographiques en tous lieux, à tout moment, ce que les spécialistes nomment le **géopositionnement** ou la **géolocalisation** [6].

➤ **Les systèmes coopératifs :**

La coopération véhicule-véhicule et véhicule-infrastructure contribuent à ces objectifs au-delà des améliorations réalisables avec des systèmes autonomes. Autrement dit, dans les systèmes coopératifs l’intelligence est partagée entre le véhicule et l’infrastructure. Les véhicules communiquent les uns avec les autres et avec l’infrastructure, offrant ainsi le potentiel d’accroître considérablement la qualité et la fiabilité des informations disponibles sur les véhicules, leur emplacement et de l’environnement routier. Ces nouvelles informations permettent non seulement d’améliorer les services existants mais également de développer des nouveaux services pour les usagers de la route [7].

➤ **Transmission :**

Le développement des services de transport intelligents génère des besoins d'échanges de données de plus en plus importants entre les différents types de capteurs existants, les usagers des transports et les centres de données chargés du recueil, de l'analyse et du traitement des données puis de leur diffusion vers les utilisateurs des services de transport. Par ailleurs, les besoins d'accès à des informations pertinentes en temps réel et les besoins d'interactivité des usagers et des véhicules avec leur environnement immédiat se font de plus en plus pressants, notamment pour des raisons de sécurité des transports. Ces échanges sont rendus possibles aujourd'hui par la disponibilité de réseaux et systèmes de communication à haut débit accessibles presque partout et à tout moment [8].

## VI. Objectifs des politiques STI:

Les systèmes de transport intelligents doivent apporter leur concours au déploiement des politiques publiques de transport [9]:

- donner aux exploitants des réseaux d'infrastructures et de services de transport des outils leur permettant de proposer à leurs clients des solutions de bout en bout **optimisant l'usage des capacités existantes des différents modes** tout en respectant les règles de la concurrence.
- **améliorer la sécurité**, en particulier la sécurité routière, par la **connaissance** et la **compréhension des règles** à appliquer et faire évoluer les comportements en automatisant les contrôles.
- **améliorer la qualité de la vie** dans les villes et les **services de mobilité** qu'elles offrent à leurs habitants et à leurs visiteurs, en facilitant l'usage des transports publics et des modes doux, notamment par l'information multimodale, la billettique et des organisations de logistique urbaine performantes.
- **réduire les inégalités** en agissant sur l'accessibilité, à la fois pour les personnes à mobilité réduite et pour les territoires les moins bien desservis.

## VII. Avantage des STI:

Les avantages des STI résident dans leur capacité à optimiser l'utilisation des infrastructures existantes, limitant ainsi les coûts, et à répondre aux besoins croissants de mobilité des citoyens. Ils jouent également un rôle crucial dans l'atténuation des ruptures qui existent dans les chaînes des déplacements, spécialement les déplacements intermodaux. L'utilisation des technologies et l'automatisation des processus permettent également:

- d'améliorer la capacité et la rapidité d'intervention dans la gestion quotidienne des déplacements;
- de compiler une foule de données sur les flux des déplacements;
- de mieux diagnostiquer les situations problématiques;
- de mieux planifier les nouvelles infrastructures en fonction des constats observés.

## VIII. Champs des STI :

Les champs d'utilisation des STI sont multiples :

### ➤ **Sécurité routière :**

La sécurité routière est un thème transversal dans le domaine des transports, elle concerne l'ensemble des connaissances, dispositifs et mesures visant à éviter les accidents de la route ou à en atténuer les conséquences. Les STI peuvent, de fait, contribuer directement ou indirectement à la sécurité routière. Ainsi se développent les  **systèmes de surveillance et d'alerte "bord de voie"** qui permettent d'alerter les usagers d'une difficulté de conduite.

Il existe également **des systèmes embarqués d'aide à la conduite** qui sont en capacité de pallier les défaillances du conducteur et de l'aider dans sa tâche de conduite en l'informant ou en réalisant, à sa place, un certain nombre d'opérations visant à une gestion plus optimale du véhicule [10].

### ➤ **Aides à la mobilité :**

L'aide à la mobilité consiste à accompagner tous les usagers y compris les personnes handicapées dans leurs déplacements. Cette aide couvre le déplacement du

départ à l'arrivée. Les technologies de l'information et de communication ont révolutionné les moyens d'aide à la mobilité [11]:

- **L'information avant le déplacement :**

L'information avant le déplacement recouvre les fonctionnalités permettant à un usager de préparer à l'avance un trajet particulier, quels que soit les modes de transport qu'il envisage d'utiliser, mais aussi plus largement des services donnant des informations de conseil en mobilité.

- **L'information pendant le déplacement :**

Cet article concerne les informations qui peuvent être obtenues (via divers médias) ou demandées (en général depuis un mobile) en temps réel par l'utilisateur au cours de son déplacement. Cet article inclut aussi les fonctionnalités de retour d'information en temps réel par les usagers.

- **Les modes de paiement lors d'un déplacement :**

La billetterie désigne l'ensemble des opérations correspondant à la délivrance de titres de transport sur un support papier (ticket, billet,...).

La billettique désigne l'outil automatisé de gestion des titres de transport sur des supports aux technologies avancées (carte magnétique, carte à puce, téléphone portable, smart phone,...).

- **Sûreté dans les transports :**

La sûreté des transports est l'ensemble des actions entreprises pour assurer la protection du patrimoine des entreprises, qu'elles assurent le transport par voie routière, par voie fluviale ou ferroviaire, ainsi que la protection de leur personnel et des marchandises transportées contre les actes illicites que sont les violences, les vols, les déprédations ainsi que l'utilisation abusive de la chaîne logistique pour transporter frauduleusement des matières, des équipements ou des êtres humains [12].

- **Gestion des transports publics :**

Les transports publics offrent des services dit « réguliers » ou « à la demande » et sont organisés par des autorités organisatrices de transport (AOT), ils peuvent être

urbains ou interurbains et l'un de leurs atouts majeurs est d'offrir au client un réseau interconnecté au niveau de pôles d'échanges qui favorisent l'intermodalité [13].

## **Conclusion :**

Dans le but de comprendre l'intérêt et de classer notre application, nous avons étudié dans ce chapitre les notions des systèmes de transport intelligents et celle des villes intelligentes.

Notre application s'inscrit donc dans le domaine des systèmes de transport intelligents et plus exactement dans l'aide à la mobilité et à la gestion des transports publics, elle doit dans cette optique répondre aux objectifs de ses systèmes en générale.

*CHAPITRE*

*II*  
*Les applications mobiles*

## **I. Introduction :**

Le téléphone mobile est en train d'évoluer à grande vitesse. Ainsi, il est à la fois un téléphone et télétexte, baladeur, appareil photo, console de jeu, réveil et agenda, terminal Internet, balise GPS, téléviseur, etc.

Ce chapitre présente un bref récapitulatif des applications mobiles fonctionnant sous des téléphones artificiellement intelligents à savoir des « smartphones ».

Dans ce qui suivra seront abordées: la notion des applications mobiles, leurs typologies, leurs avantages et inconvénients, les principaux systèmes d'exploitation, l'évolution du marché, et enfin, quelques exemples d'applications destinées au transport.

## **II. Définitions :**

### **II.1. Les applications mobiles :**

Une application mobile est un programme autonome conçu pour s'exécuter sur un terminal mobile, comme un smartphone ou une tablette tactile. Elle a donc accès à toutes les fonctions « natives » de ces appareils : GPS, appareil photo, etc. Comme tout programme, elle doit donc être téléchargée au préalable pour être installée [14].

#### **II.1.1. Application iPhone :**

Une application iPhone est un type d'applications mobiles spécialement développée pour l'iPhone d'Apple.

Le développement d'une application iPhone permet de profiter du succès commercial de mobiles et de ses qualités ergonomiques d'usage. L'usage des fonctionnalités tactiles et de navigation de l'iPhone permet un accès en mobilité à des services proposés par un site de manière plus confortable et plus efficace qu'en passant par un accès web, même s'il est optimisé pour les accès mobiles [15].

Une application iPhone peut être proposée gratuitement à des fins marketings ou de création de commerce ou être payante et constituer en elle-même un « business model » comme pour le cas des jeux.

Les applications iPhone s'obtiennent sur l'« **Apple Store** ».

### **II.1.2. Application Android :**

Une application Android est une application mobile spécifiquement développée pour smartphones utilisant le système d'exploitation Android [16].

Comme les applications iPhone dont elles sont souvent des répliques, les applications Android sont de nature très variables :

- Jeux,
- Mobile commerce,
- Utilitaires,
- Services d'informations, etc.

Les applications Android s'obtiennent sur l'«**GooglePlay**».

## **III. Typologies :**

Il existe trois classes importantes pour les applications mobiles :

### **III.1. Application native :**

Une application native est une application développée spécifiquement pour une seule plateforme, grâce aux outils conçus pour celle-ci. Elle est développée avec un langage spécifique à son système d'exploitation et est distribuée uniquement par l'intermédiaire d'un app store propre à sa plateforme [17].

Le développement natif, permet essentiellement d'utiliser la mémoire du smartphone ainsi que toutes les fonctionnalités liées au système d'exploitation visé (GPS, caméra ...). Il permet également d'offrir une ergonomie adaptée à chaque modèle de smartphone et à chaque système d'exploitation. Les applications sont donc plus riches en termes de fonctionnalité, de qualité, de performance et d'ergonomie que les web apps ou les applications hybrides [17].

Cependant, pour développer une application native pour différent OS (Android et iOS par exemple), il convient de développer deux applications distinctes. Entraînant une perte de temps et une augmentation de coût. De plus, avec la sortie de nouveaux systèmes d'exploitation chaque année, des problèmes de rétrocompatibilité peuvent se poser. L'application doit être maintenue, ce qui augmente les coûts de développement.

### **III.2. Application web ou web app :**

Il s'agit de sites web optimisés pour mobiles, souvent conçus pour ressembler à de vraies applications. Elle peut être aussi appelée « apps navigateur mobile » car elles ne sont pas installées sur l'appareil. Celles-ci peuvent être accédées en employant le navigateur mobile grâce à l'URL du Web. Dans ce cas, la capacité de la mémoire de l'appareil n'est pas aussi importante étant donné que ni la forme ni les données de l'application ne sont pas stockés dedans. Il est complètement dépendant de la qualité du navigateur. Chaque donnée vient du serveur et rendu dans le navigateur quand vous accédez grâce à l'URL. Elle est développée avec les outils de développement web actuel: HTML5, CSS3 et JavaScript [17].

### **III.3. Application hybride :**

Une application mobile hybride est développée à partir de langages web (HTML, CSS, JavaScript). Cependant, elle s'appuie sur des technologies natives mobiles pour utiliser certaines fonctionnalités du smartphone [18].

Bien que développée avec du web, il s'agit bien d'une « application » dans le sens où elle sera téléchargée depuis les magasins d'applications et installée sur le mobile, contrairement au web app qui n'est consultable que depuis un navigateur [18].

#### **Une application mobile hybride à plusieurs avantages :**

- C'est une technologie multiplateforme, ce qui est un gain de temps considérable si le projet n'est pas trop complexe et ne nécessite pas le développement de plugins spécifiques.
- Les compétences web sont plus répandues que les compétences mobiles, les coûts de développement sont donc généralement moins importants.
- Conçue pour être téléchargeable, l'application mobile hybride peut être monétisée et bénéficie des classements dans les stores.

#### **En revanche, certains inconvénients sont à prendre en compte :**

- Le code source est mélangé et la compilation se fait en de multiples étapes, ce qui peut fragiliser le fonctionnement de l'application mobile et rendre la maintenance longue et complexe.

- Des contenus trop lourds peuvent avoir de plus forts impacts sur les performances qu'avec une application native.
- Le mode hors-ligne est plus limité et plus délicat à concevoir.
- Tout n'est pas réalisable en termes de fonctionnalités avec cette technologie, il faut s'assurer en amont de la compatibilité de votre projet et de ses évolutions futures avec les technologies hybride.

## **IV. Avantages et inconvénients des applications mobiles :**

Les applications mobiles s'étalent dans des différents domaines, et touchent toutes les générations grâce à leurs avantages et facilités d'utilisation, malgré quelques inconvénients qu'on leurs trouve.

### **IV.1. Avantages :**

Parmi les objectifs des applications mobiles on trouve:

- Contenus adaptés aux spécificités de chaque smartphone.
- Contenus disponibles hors ligne, sera toujours accessible (plus besoin d'attendre l'ouverture de la page).
- Possibilité de sélectionner des contenus à afficher selon la cible et l'objectif de communication.
- Etablissement d'un lien direct avec le consommateur.
- Rapidité d'exécution (l'application mobile exploite au mieux les capacités du téléphone).
- Facilité d'installation et d'accès.
- Possibilité de solliciter un mobinaute (Push).
- interface plus riche et des meilleures performances.
- Facilité de Monétisation (parce qu'elles sont directement disponible sur les plateformes de téléchargement).
- L'application a accès aux dispositifs du terminal (exemple appareil photo, GPS).
- Plus besoin de taper l'URL de votre site dans le petit navigateur de votre téléphone.

## IV .2. Inconvénients :

Pour développer une application mobile il faut savoir qu'elles ont aussi quelques inconvénients citons:

### ➤ **Le respect des règles des plateformes mobiles :**

L'Apple Store, le Play Store et le Windows Store imposent un certain nombre de règles pour les développeurs. Parfois contraignantes, elles sont un passage obligé si l'on souhaite pouvoir distribuer une application smartphone de manière optimale.

### ➤ **Un coût de développement élevé :**

Le coût de création d'une application mobile est relativement élevé et peut décupler les dépenses e-marketing, surtout si l'on souhaite que l'application soit disponible sur tous les systèmes d'exploitation pour mobiles.

### ➤ **Contrainte de la mise à jour pour l'utilisateur :**

En cas de mise à jour de l'application, l'utilisateur d'application mobile doit se rendre une nouvelle fois sur les plateformes de téléchargement, Le site mobile peut quant à lui évoluer sans contraintes pour le visiteur.

## V. Les plateformes mobiles :

Une application mobile est conçue pour fonctionner avec un système d'exploitation spécifique. Les trois principaux systèmes d'exploitation du marché sont :

### V.1. iOS :

iOS est le système d'exploitation d'Apple installé sur iPhone, iPod Touch, iPad et Apple Watch. Apple a été le premier à lancer un smartphone grand public avec le succès qu'on lui connaît : l'iPhone. Ce smartphone intègre un écran multi-touch, un accéléromètre et la localisation par triangulation GSM [19].

### V.2. Windows Phone :

Windows Phone est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, sa précédente plateforme logicielle qui a été renommée pour l'occasion en Windows Phone Classique [19].

### V.3. Android :

Android est un système d'exploitation pour Smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 15 novembre 2007. C'est un système open source basé sur un Kernel Linux qui peut être installé sur n'importe quel appareil [19].

Android n'a pas été conçu uniquement pour les Smartphones mais pour tous les appareils embarqués tels que des tablettes, des montres, des baladeurs, des télévisions, des voitures, etc.

Le tableau suivant résume les principaux systèmes d'exploitation mobiles présents sur le marché :

Société	Système d'exploitation	Langage de développement	Plateforme
Apple	iOS	Objective-C	AppStore
Microsoft	Windows Phone	C#	MarketPlace
Google	Android	Java	GooglePlay

Figure II.1: descriptif des principaux OS mobiles présents sur le marché [17].

## VI. Le marché mondial des applications mobiles :

Depuis plusieurs années l'utilisation des smartphones et des applications mobiles progresse régulièrement.

Le marché des applications mobiles pèsera 100 milliards de dollars en 2020 estime App Annie, soit le double de ce qu'il rapporte aujourd'hui : 50,9 milliards de dollars prévu pour 2016, et 41,1 milliards de dollars en 2015. Cela est dû à La pénétration des smartphones qui est désormais très forte sur l'ensemble des marchés mondiaux et l'e-commerce mobile qui se développent fortement [20].

App Annie mentionne qu'Apple possède 55% du chiffre d'affaires total du marché des magasins d'application en ligne. Une tendance déjà détectée en 2014. En 2020 cette part de marché devrait tomber à 44%, bousculée par le positionnement de Google qui se développe rapidement en Chine via ses partenaires. Les revenus de Google Play & Co devraient dépasser ceux d'Apple en 2017 [20].

App Annie est une société de business intelligent et un cabinet d'analystes dont le siège social se trouve à San Francisco en CALIFORNIE. Elle produit des outils de business intelligents et des rapports de marché sur les applications et les entreprises de production d'application.

La figure suivante présente la part des revenus des applications mobiles :

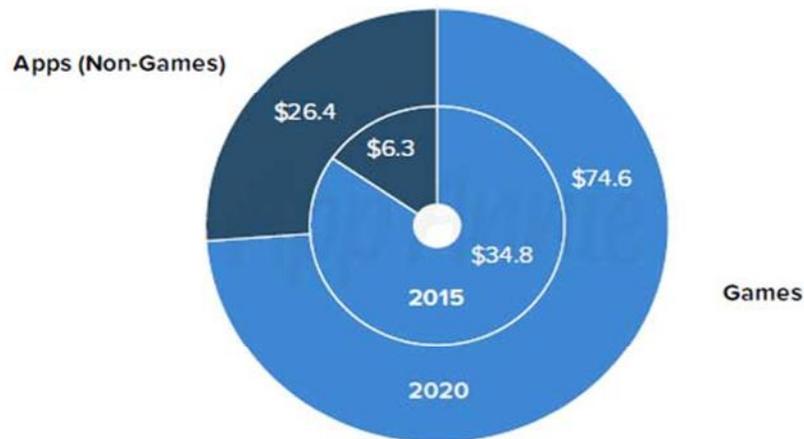


Figure II.2 : Les applications mobiles – Prévion annuelle du revenu 2015 vs 2020 par catégorie [20].

La part des téléchargements des applications mobiles est présentée dans la figure suivante :

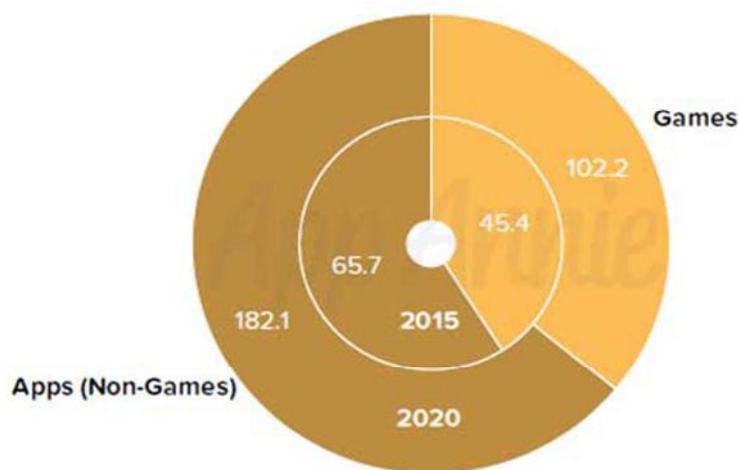


Figure II.3 : Les applications mobiles – téléchargements annuelles 2015 vs 2020 par catégorie [20].

## VII. Exemple d'applications destinées au transport :

Il existe de nombreuses applications qui sont destinées au transporteur, le tableau ci-dessous présente quelques exemples d'application :

<b>Nom de l'Application</b>	<b>Description</b>
<p data-bbox="263 585 391 620" style="text-align: center;"><b><i>Moovit</i></b></p>  <p data-bbox="197 947 455 1006" style="text-align: center;"><i>Slogan de l'application Moovit</i></p>	<p data-bbox="520 563 1410 875">Moovit est une application gratuite qui souhaite révolutionner l'expérience des transports en commun. Elle combine les données des transporteurs avec les rapports en temps réel des utilisateurs, Moovit donne un aperçu complet en temps réel de ce que sera le trajet, afin que les utilisateurs gagnent du temps et voyagent plus confortablement [21].</p>
<p data-bbox="232 1105 428 1140" style="text-align: center;"><b><i>STRASMAP</i></b></p>  <p data-bbox="201 1465 458 1541" style="text-align: center;"><i>Slogan de l'application STRASMAP</i></p>	<p data-bbox="520 1072 1410 1384">STRASMAP est une application de déplacement multimodal avec la société Thalès, téléchargeable sur les téléphones mobiles Android et iPhone. Elle facilite la vie des usagers dans leurs déplacements sur toute l'agglomération. Elle regroupe le transport en commun, piéton, vélo, voiture, auto partage, occupation des parkings et vélos disponibles [21].</p> <p data-bbox="617 1399 956 1434">Parmi ses services offerts :</p> <ul data-bbox="671 1458 1410 1734" style="list-style-type: none"><li>• Notification des alertes infos trafic.</li><li>• Consultation en temps réel de l'état du trafic.</li><li>• Consultation en temps réel des places de parking disponibles.</li><li>• Calcul d'itinéraire en temps réel.</li></ul>

<p><b>URBAN PULSE</b></p>  <p><i>Slogan de l'application</i> <b>URBAN PULSE</b></p>	<p>Ce logiciel est développé par Veolia Transdev, il trouve les trajets ou itinéraires en prenant en compte tous les modes de transport: Bus, Métro, RER, Tram, Taxis, Parkings, Bateau-bus, et même les stations-services.</p> <p>Urban Pulse comprend à la fois des fonctionnalités transport, réseau social, loisirs et bons plans. Disponibilité de voiture auto-partage, vélo libre service, indique les bornes de taxi, les horaires d'ouverture et tarifs des parkings [21].</p>
<p><b>UBER</b></p>  <p><i>Slogan de l'application</i> <b>UBER</b></p>	<p>Uber est une entreprise technologique qui développe et opère des applications mobiles de mise en contact d'utilisateurs avec des conducteurs réalisant des services de transport.</p> <p>Le service permet de localiser, via un téléphone intelligent, une tablette ou un ordinateur, le véhicule le plus proche de soi pour ensuite le réserver. Les utilisateurs peuvent suivre l'approche de la voiture qu'ils ont réservée sur l'application. Le paiement est réalisé automatiquement à la fin de chaque course par prélèvement bancaire. Il n'y a donc pas d'échange d'espèces entre conducteur et passager. Les services proposés par Uber sont particulièrement utilisés dans les périodes de moindre disponibilité des moyens de transport traditionnels, notamment la nuit et le week-end [21].</p>

**Figure II.4: Description de quelques applications destinées au transport.**

## **Conclusion :**

Dans ce chapitre nous avons vu les applications mobiles en général, et leurs caractéristiques ainsi que la part qu'elles détiennent dans le marché. Nous avons donné aussi des exemples d'application destinées au transport.

Dans ce qui va suivre nous allons entamer la partie analyse et conception de notre application.

CHAPITRE

*III*  
*Analyse et Conception*

## I. Introduction :

Afin d'aboutir à une meilleure organisation et une bonne maîtrise du travail et donc arriver à déployer de meilleures applications, il est nécessaire de suivre une démarche méthodologique rigoureuse. Pour cela nous avons opté pour le langage de modélisation UML pour la représentation qu'il offre à travers ses différents diagrammes.

Cette démarche peut être présentée graphiquement comme suit :

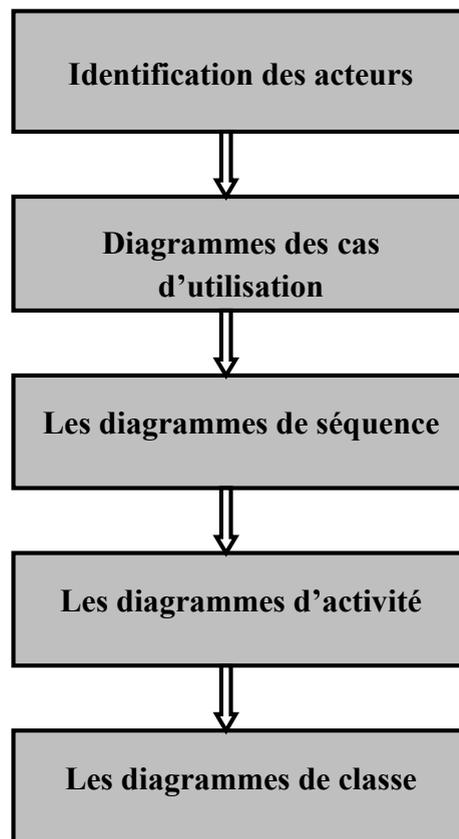


Figure III.1 : les démarches de modélisation de l'application.

## II. Objectif de notre application :

Le but principal de notre future application est la mise en contact des utilisateurs demandant un service de transport, et des transporteurs offrant ce service.

Le système permettra de localiser le point de départ ainsi que le point d'arrivée afin de déterminer le transporteur le plus proche pour ensuite le réserver.

### III. Première étape: Analyse

Cette étude commence par l'identification des acteurs afin de pouvoir passer à l'étude des cas d'utilisation et de leurs descriptions textuelles.

#### III.1. Identification des acteurs :

##### ➤ Un acteur :

Un acteur représente un ensemble de rôles joués par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié [22].

Dans notre application, nous avons trois acteurs qui interagissent avec le système :

- **L'utilisateur** : c'est la personne qui se connecte à l'application pour demander un service de transport.
- **Le transporteur** : c'est la personne offrant le service de transport.
- **Le visiteur** : c'est la personne qui visite l'application dans le but de connaître les services qu'elle propose.

#### III.2. Les cas d'utilisation :

Un cas d'utilisation représente un ensemble de séquences d'action qui sont réalisés par le système dont la finalité est de rendre un service à un acteur [22].

Les cas d'utilisation relatifs à notre application :

### III.2.1. Cas d'utilisation : création d'un compte utilisateur :

**Cas d'utilisation :** créer un compte utilisateur

**Acteur :** utilisateur.

**Description :**

1. Après ouverture de l'application, le système affiche deux boutons, « UTILISATEUR » et « TRANSPORTEUR ».
2. L'utilisateur clique sur le bouton « UTILISATEUR ».
3. Le système affiche le formulaire d'inscription des utilisateurs.
4. L'utilisateur remplit le formulaire et clique sur le bouton « s'inscrire ».
5. Le système affiche le profil de l'utilisateur.

### III.2.2. Cas d'utilisation : création d'un compte transporteur :

**Cas d'utilisation:** créer un compte transporteur

**Acteur:** transporteur.

**Description:**

1. Après ouverture de l'application, le système affiche deux boutons, « UTILISATEUR » et « TRANSPORTEUR ».
2. Le transporteur clique sur le bouton « TRANSPORTEUR »
3. Le système affiche le formulaire d'inscription des transporteurs.
4. Le transporteur renseigne ses informations personnelles ainsi que les informations relatives à un véhicule puis il clique sur le bouton « s'inscrire ».
5. Le système affiche le profil du transporteur.
6. Le transporteur est automatiquement localisé.

### III.2.3. Cas d'utilisation : connexion au compte :

**Cas d'utilisation:** se connecter

**Acteurs:** utilisateur ou transporteur

**Description:**

1. L'utilisateur ou le transporteur clique sur le bouton « connexion ».
2. Le système affiche l'écran d'authentification.
3. L'utilisateur ou le transporteur s'authentifie en saisissant son adresse e-mail et son mot de passe et clique sur le bouton « connexion ».
4. Le système affiche le profil de l'utilisateur ou du transporteur.

### III.2.4. Cas d'utilisation : Demande d'un service de transport :

**Cas d'utilisation:** Demander un service du transport

**Acteur:** utilisateur

**Description :**

1. L'utilisateur saisit son adresse de départ et d'arrivée. Elles sont situées sur la carte.
2. Il saisit aussi le nombre de places et clique sur le bouton « commander ».
3. Le système transmet les coordonnées du transporteur sélectionné à l'utilisateur.
4. Le système transmet les coordonnées de l'utilisateur au transporteur.

### III.2.5. Cas d'utilisation : réponse de l'utilisateur:

**Cas d'utilisation:** Confirmer ou annuler la commande

**Acteur:** utilisateur

**Description:**

1. L'utilisateur confirme la commande et attend le transporteur, ou il annule la commande.
2. En cas de validation, le transporteur reçoit une notification de confirmation et se rend sur le lieu de rendez-vous. En cas d'annulation, il reçoit une notification d'annulation.

Voici les diagrammes de cas d'utilisation qui résument les cas vus plus haut :

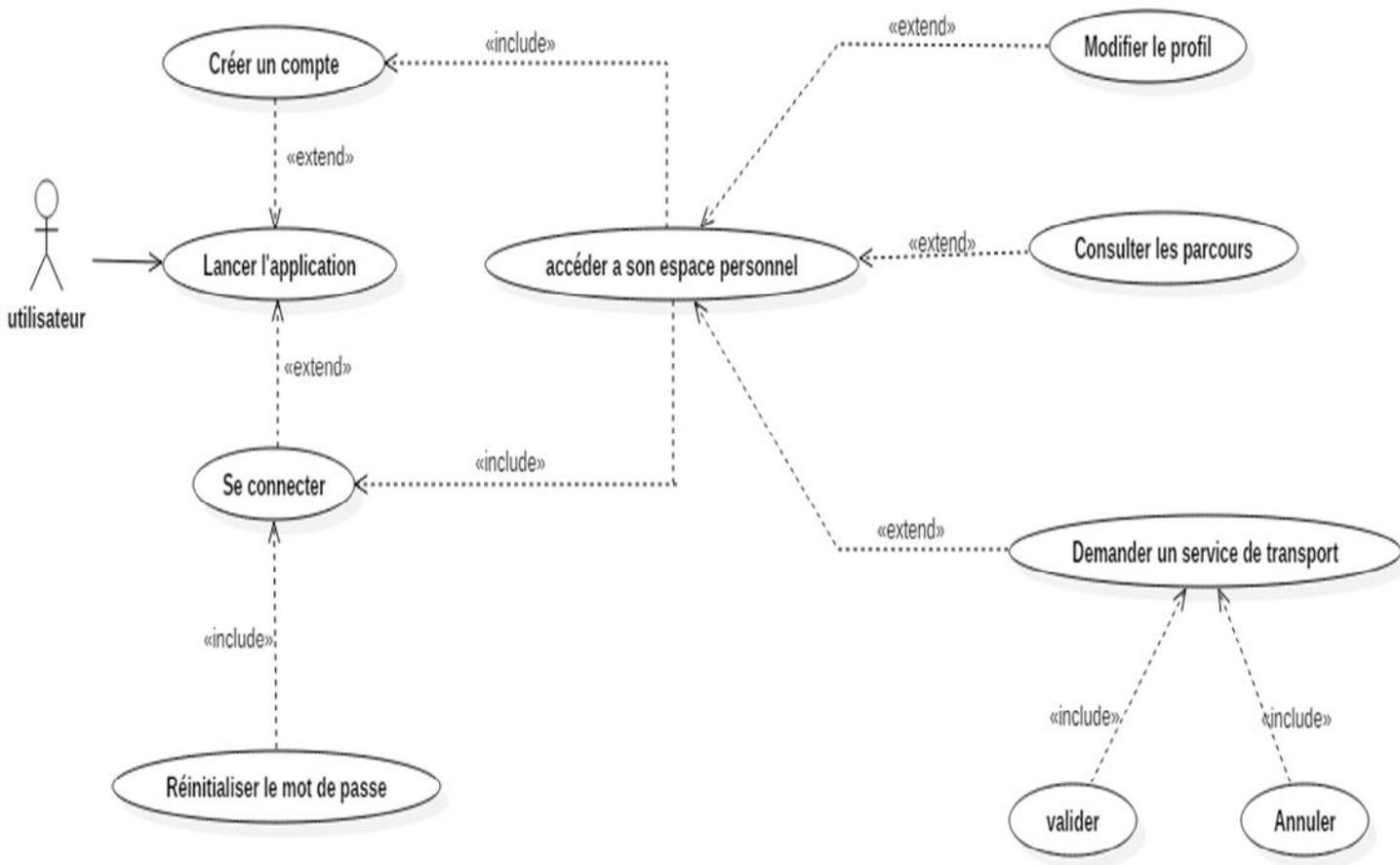


Figure III.2 : diagramme de cas d'utilisation « utilisateur ».

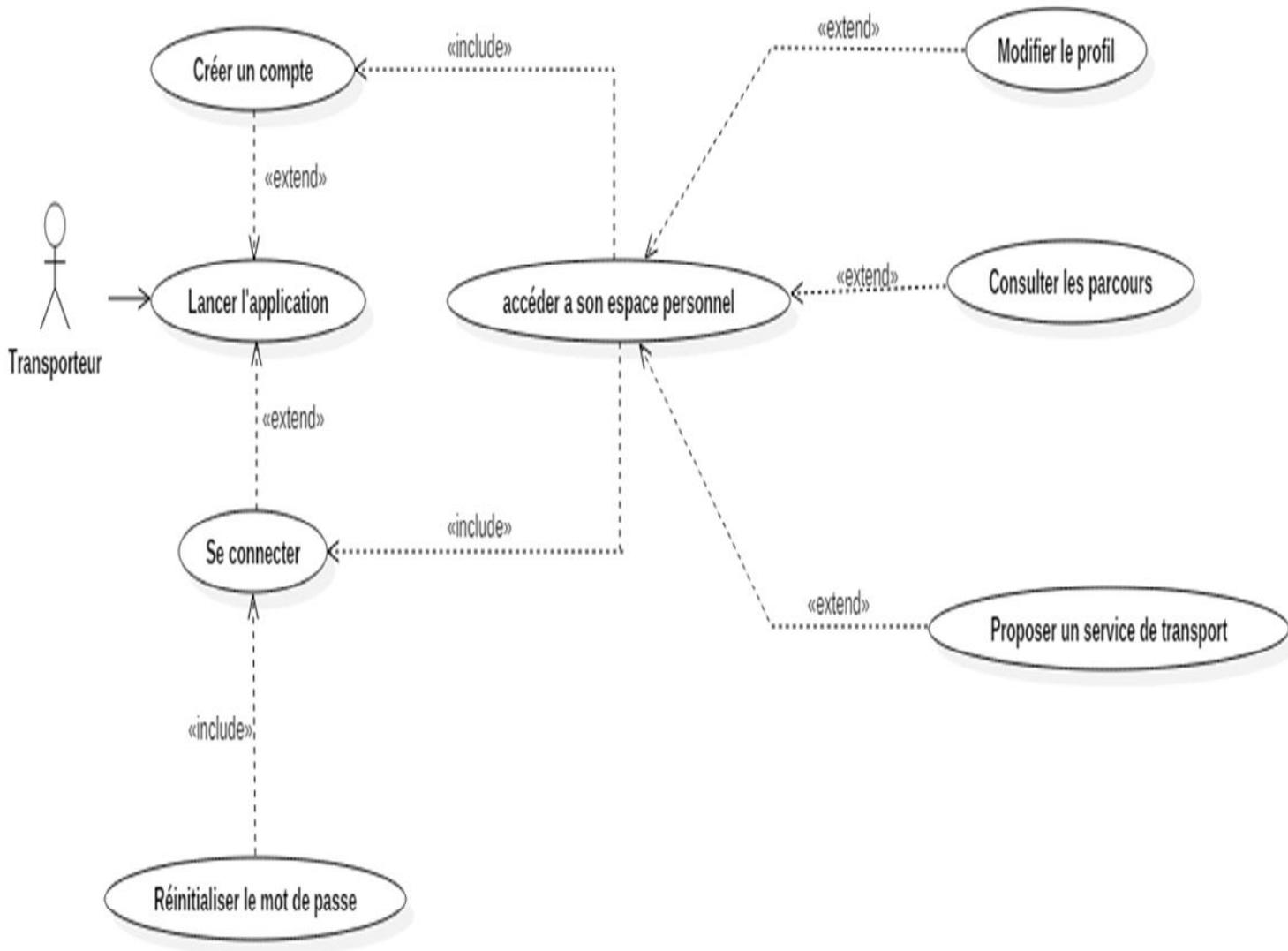


Figure III.3: diagramme du cas d'utilisation « transporteur »

### III.3. Les diagrammes de séquence :

Les diagrammes de séquence montrent des interactions entre objets selon un point de vue temporel. Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique [22].

III.3.1. Diagramme de séquence : création d'un compte :

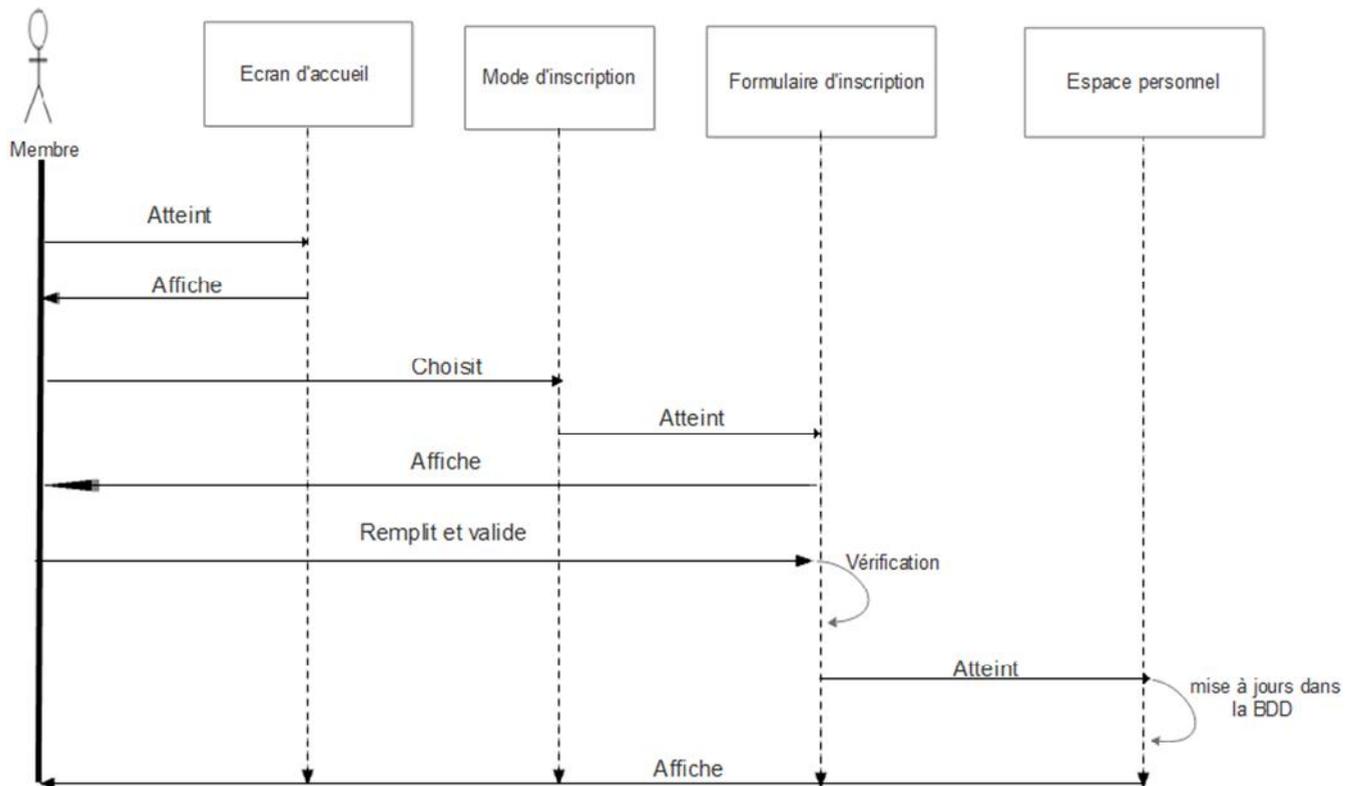


Figure III.4: diagramme de séquence « création d'un compte ».

III.3.2. Diagramme de séquence : connexion au compte:

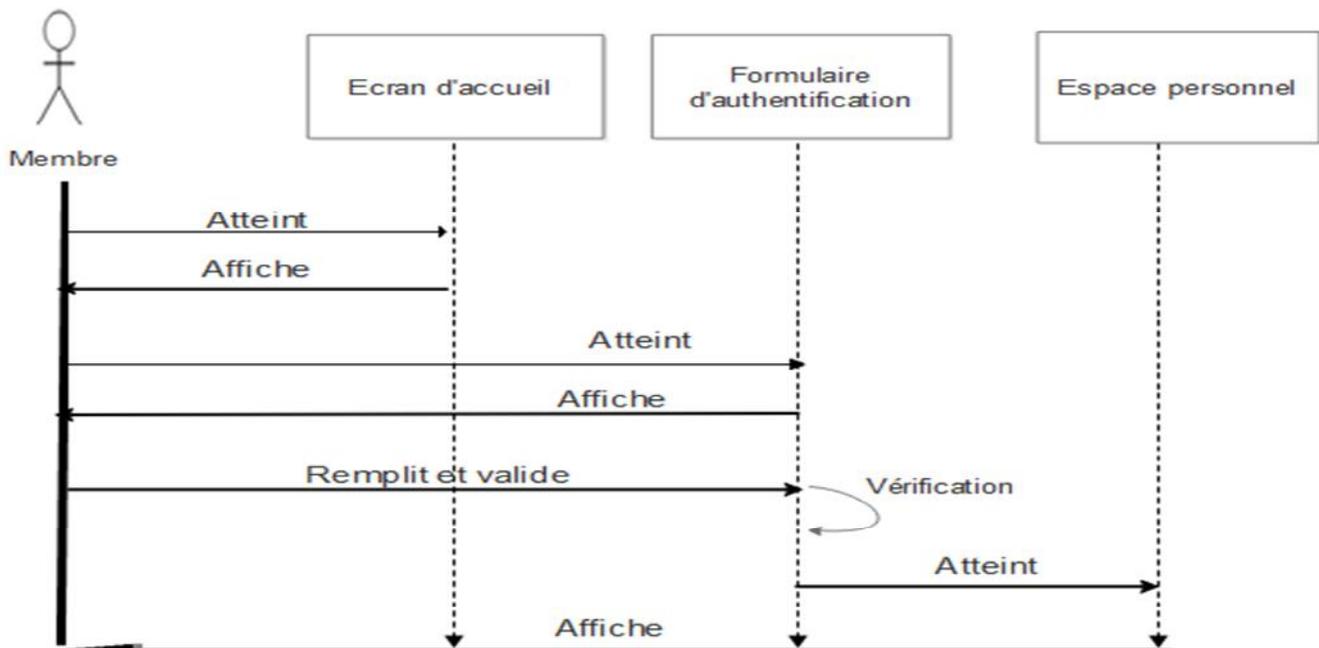


Figure III.5 : diagramme de séquence « connexion au compte ».

### III.3.3. Diagramme de séquence : demande d'un service de transport:

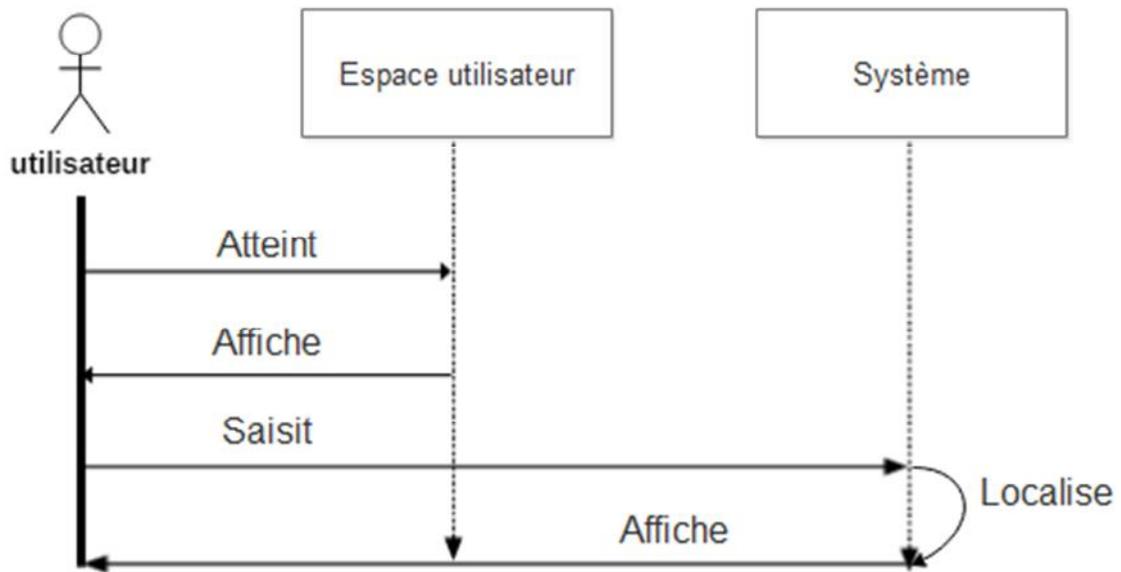


Figure III.6 : diagramme de séquence « demande d'un service de transport ».

### III.3.4. Diagramme de séquence : réponse de l'utilisateur :

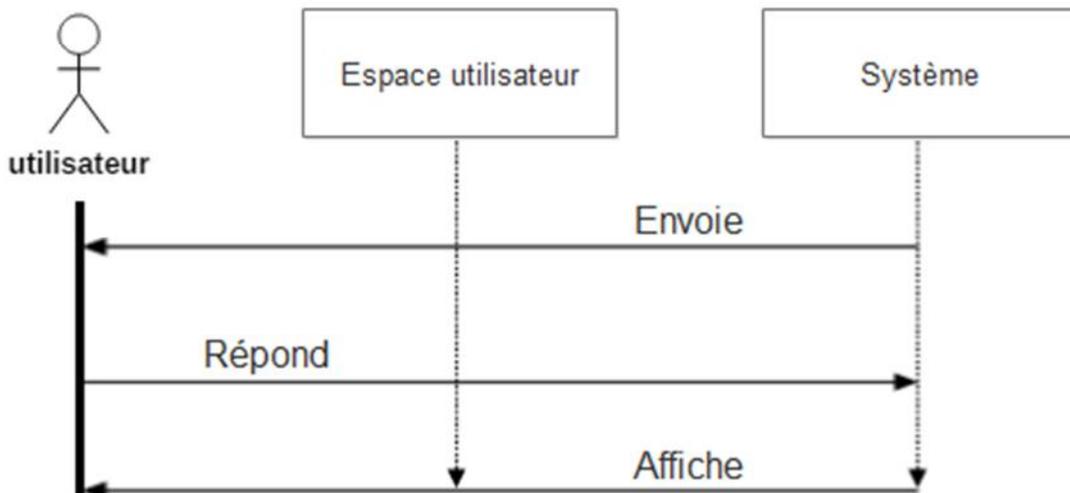


Figure III.7 : diagramme de séquence « Réponse de l'utilisateur »

## III.4. Les diagrammes d'activité :

Le diagramme d'activités est utilisé pour la modélisation des aspects dynamiques des systèmes, principalement destiné à représenter le comportement interne d'une méthode (la réalisation d'une opération) ou d'un cas d'utilisation [22].

III.4.1. Diagramme d'activité : création d'un compte :

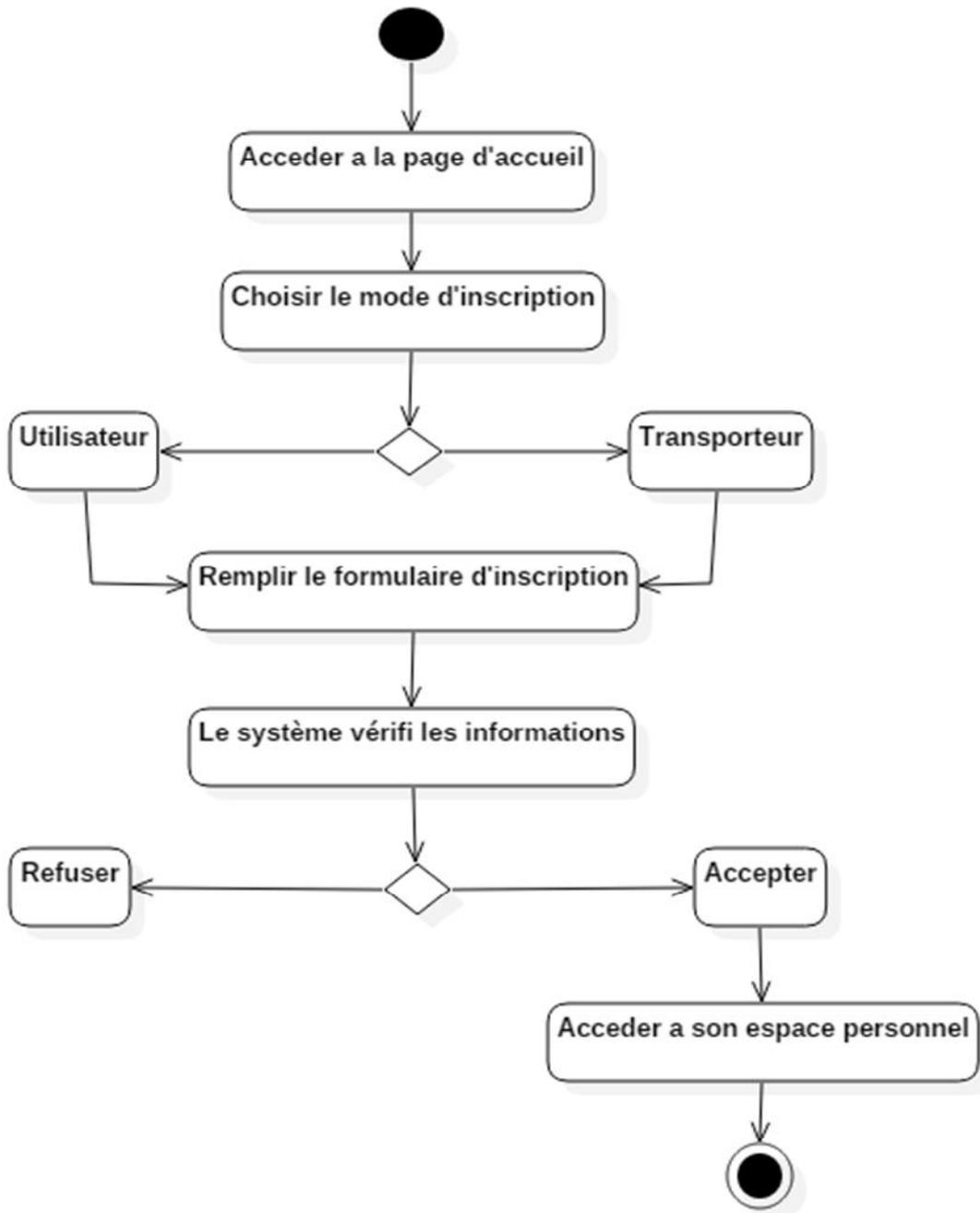


Figure III.8: diagramme d'activité « création d'un compte ».

III.4.2. Diagramme d'activité: connexion au compte :

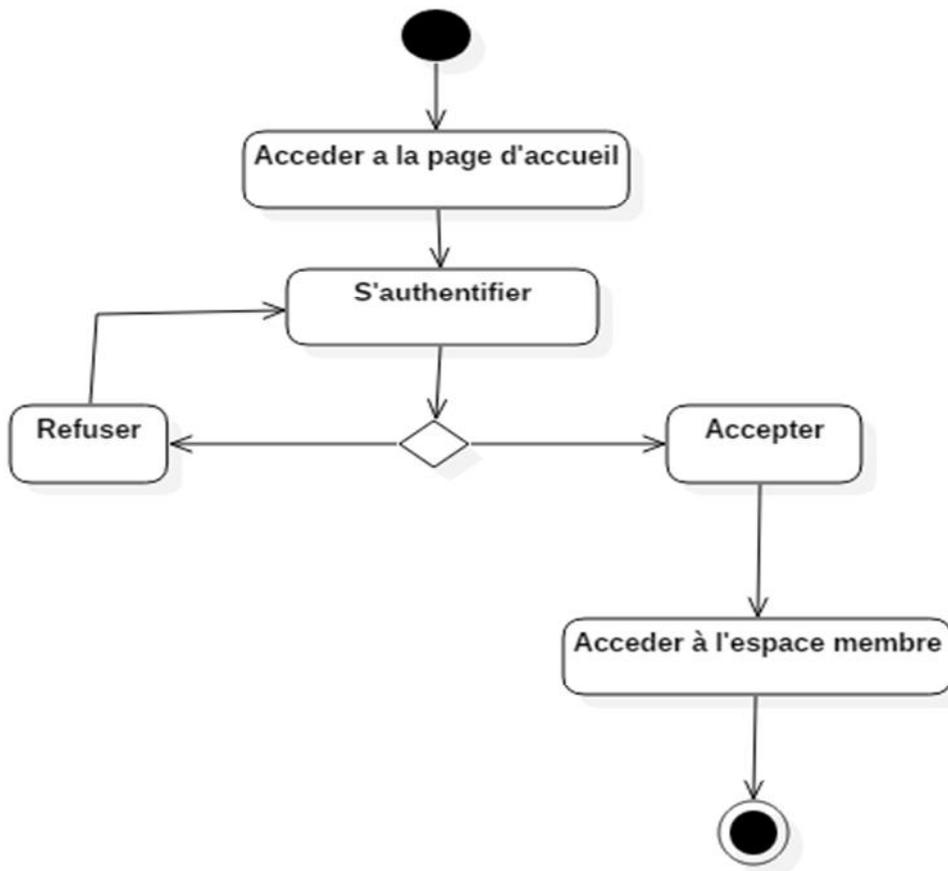


Figure III.9: diagramme d'activité « connexion au compte ».

III.4.3. Diagramme d'activité: demande d'un service de transport :

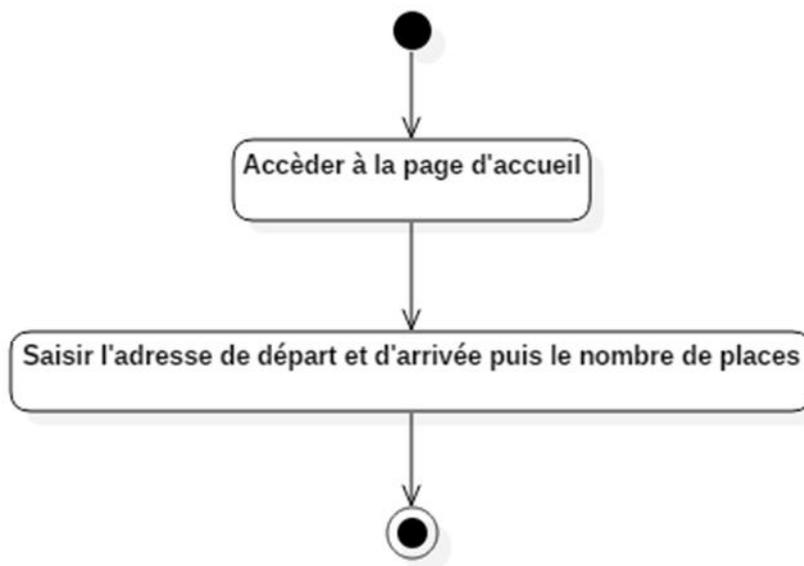


Figure III.10: diagramme d'activité « Demande d'un service de transport ».

#### III.4.4. Diagramme d'activité: réponse de l'utilisateur :

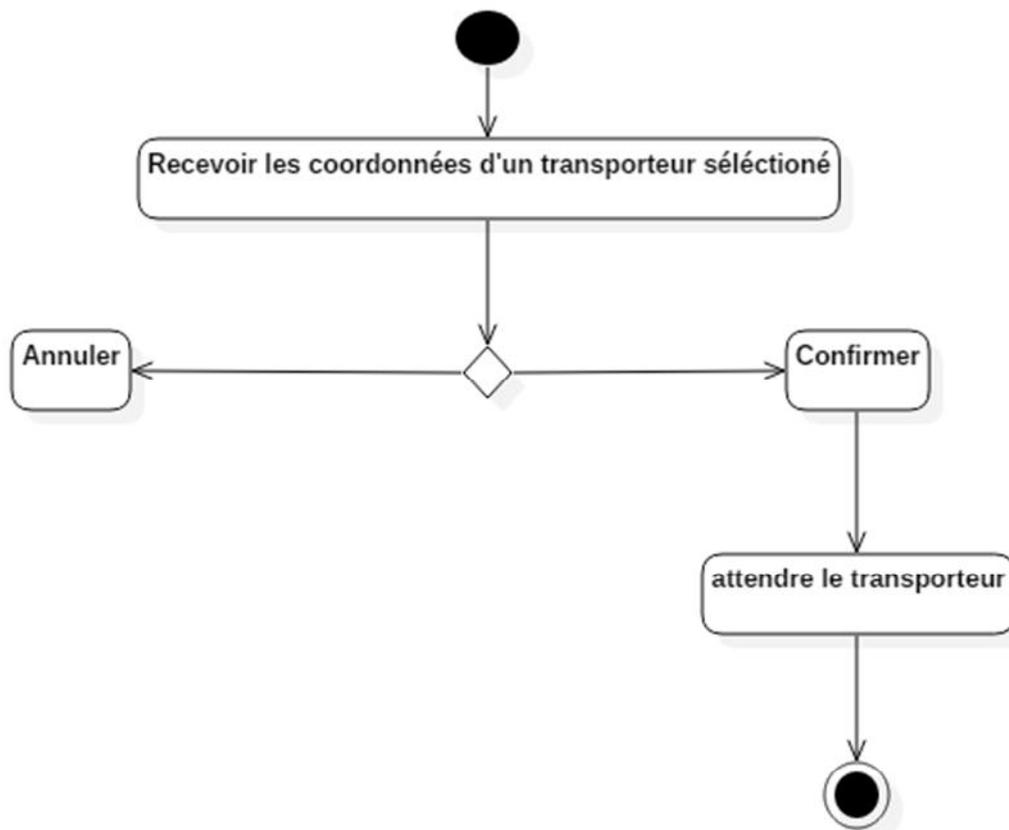


Figure III.11: diagramme d'activité « Réponse de l'utilisateur ».

## IV. Deuxième étape: Conception

La conception est la dernière étape avant la réalisation technique de la plateforme de l'application, elle consiste donc à réaliser le modèle qui va être implémenté.

Tout comme l'étape d'analyse, UML utilise différents diagrammes pour la conception, dans notre cas on va élaborer le diagramme de classe.

### IV.1. Diagramme de classe :

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Il est le seul obligatoire lors d'une modélisation [23].

Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation [23].

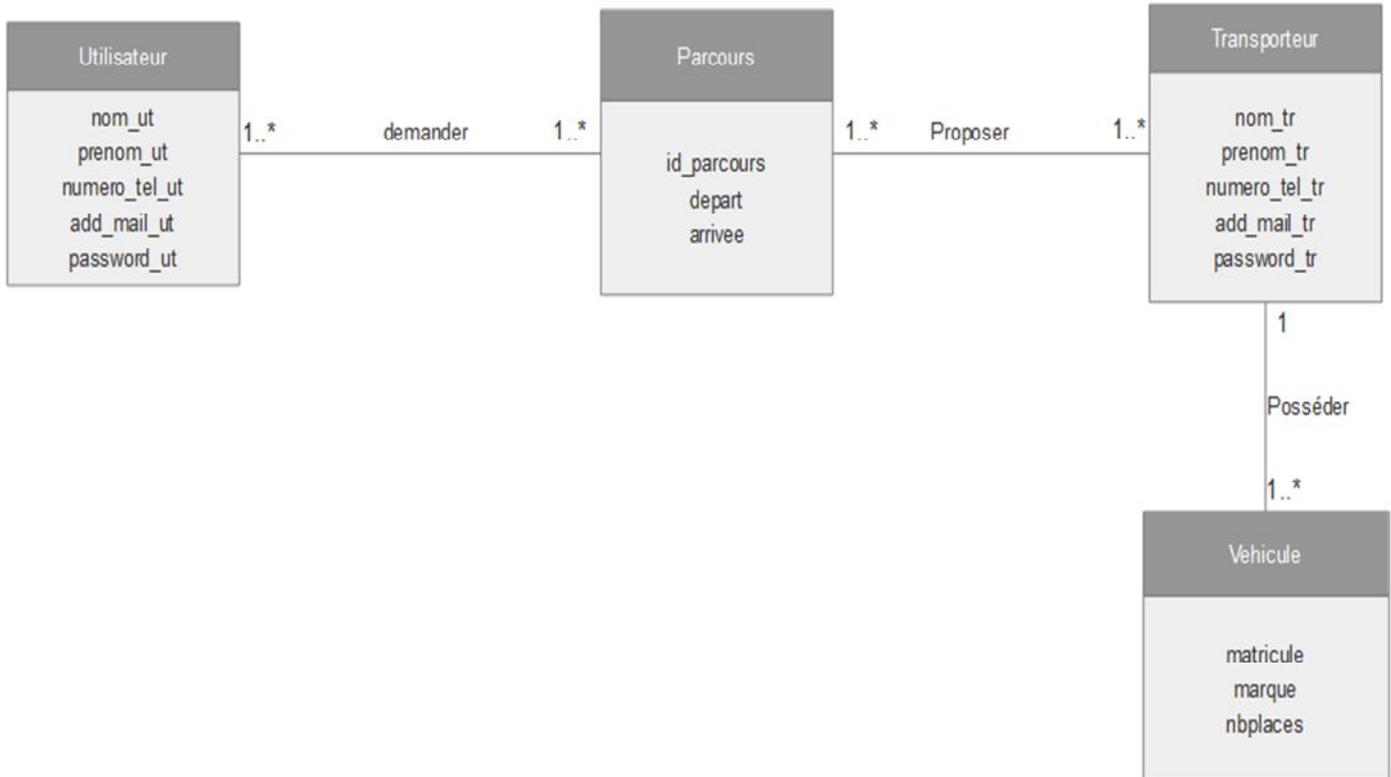


Figure III.12: diagramme de classe global.

## IV.2. Le modèle relationnel :

Le modèle relationnel représente l'information dans une collection de relations. Dans ce qui suit, on va déterminer l'ensemble des informations manipulées par notre application.

**Utilisateur** (nom\_ut, prenom\_ut, numero\_tel\_ut, **add\_mail\_ut**, password\_ut).

**Transporteur** (nom\_tr, prenom\_tr, numero\_tel\_tr, **add\_mail\_tr**, password\_tr).

**Vehicule**( **matricule**, add\_mail\_tr, marque, nbplaces).

**Parcours** (**id\_parcours**, add\_mail\_tr, add\_mail\_ut, depart, arrivee).

## IV.3. Codification :

Toutes les informations manipulées dans notre application doivent être stockées dans différentes tables d'une base de données, les noms attribués à ces tables et leurs différents champs doivent respecter certaines règles.

• Table « Utilisateur »

Nom	Type de données	Description	Clef	Observation
nom_ut	Chaine de caractères	Nom de l'utilisateur		
prenom_ut	Chaine de caractères	Prénom de l'utilisateur		
numero_tel_ut	Entier	Numéro de téléphone de l'utilisateur		Unique
Add_mail_ut	Chaine de caractères	Email de l'utilisateur	Primaire	Unique
Password_ut	Chaine de caractères	Mot de passe de l'utilisateur		Crypté

• Table « Transporteur »

Nom	Type de données	Description	Clef	Observation
nom_tr	Chaine de caractères	Nom du transporteur		
prenom_tr	Chaine de caractères	Prénom du transporteur		
numero_tel_tr	Numéro de téléphone du transporteur	Numéro de téléphone du transporteur		unique
add_mail_tr	Chaine de caractères	Email du transporteur	primaire	unique
Password_tr	Chaine de caractères	Mot de passe du transporteur		Crypté

- Table « Véhicule »

Nom	Type de données	Description	Clef	Observation
matricule	Entier	Matricule du véhicule	Primaire	unique
Add_mail_tr	Chaine de caractères	Identifiant du transporteur	étrangère	
marque	Chaine de caractères	La marque du véhicule		
nbplaces	Entier	Le nombre de places dans le véhicule		

- Table parcours

Nom	Type de données	Description	Clef	Observation
Id_parcours	entier	identifiant	Primaire	Auto increment
Add_mail_tr	Chaine de caractères	Identifiant du transporteur	étrangère	
Add_mail_ut	Chaine de caractères	Identifiant de l'utilisateur	Etrangère	
depart	Chaine de caractères	L'adresse de départ		
Arrive	Chaine de caractères	L'adresse d'arrivée		

## **Conclusion :**

Dans ce chapitre nous avons présenté une démarche de modélisation pour développer notre application, cela nous a permis d'identifier les différents objets qui nous aident à certifier les fonctionnalités souhaitées.

Dans le prochain chapitre, nous définirons les différents outils utilisés et nous utiliserons notre analyse et conception faite dans ce chapitre afin de nous aider à réaliser l'application demandée.

CHAPITRE

# IV

*Réalisation*

## **I. Introduction :**

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, et nous finissons par la présentation de ses fonctionnalités à travers ses différentes interfaces.

## **II. Environnement de développement :**

### **II.1. Environnement matériel :**

Pour développer cette application nous avons utilisé le matériel suivant :

- Pc portable HP avec une carte réseau.
- Pc portable ASUS avec une carte réseau.

### **II.2. Environnement logiciel :**

Pour la mise en œuvre de notre application, nous avons opté pour les outils logiciels suivants :

- ✓ Le Framework ionic pour développer des applications mobiles hybrides.
- ✓ Le langage de requête MYSQL pour interroger la base de données.
- ✓ Node.js qui est une technologie permettant d'écrire du JavaScript coté serveur.

## **III. Présentation du Framework Ionic :**

### **III.1. Définition :**

Le framework ionic est un mélange d'outils et de technologies pour développer des applications mobiles hybrides rapidement et facilement. Il s'appuie sur **HTML5**, **CCS3**, **AngularJS** et sur **Cordova**. Ce Framework open source permet de développer une application déployable sur plusieurs environnements tels qu'un site web ou une application mobile pour des systèmes tels qu'Android ou iOS ou Windows Phone [24].

### **III.2. Installation :**

Le framework ionic s'appuie sur la plateforme **NodeJS** et plus précisément **NPM** (Node Package Manager) pour installer les nouveaux modules développés par la

communauté et gérer les dépendances entre les modules, donc il faudra les installer. Ensuite, il suffit d'exécuter la commande suivante (en ligne de commande) :

```
$ npm install -g ionic
```

### III.3. Génération du squelette de l'application :

Pour créer un nouveau projet sous Ionic, on saisit la commande suivante :

```
$ ionic start nom_application blank
```

Voici le résultat de l'exécution de la commande :

```
D:\travail>ionic start Application blank
Creating Ionic app in folder D:\travail\Application based on blank project
Downloading: https://github.com/driftyco/ionic-app-base/archive/master.zip
[=====] 100% 0.0s
Downloading: https://github.com/driftyco/ionic-starter-blank/archive/master.zip
[=====] 100% 0.0s
Updated the hooks directory to have execute permissions
Update Config.xml
Initializing cordova project

Your Ionic project is ready to go! Some quick tips:

* cd into your project: $ cd Application
* Setup this project to use Sass: ionic setup sass
* Develop in the browser with live reload: ionic serve
* Add a platform (ios or Android): ionic platform add ios [android]
  Note: iOS development requires OS X currently
  See the Android Platform Guide for full Android installation instructions:
  https://cordova.apache.org/docs/en/edge/guide_platforms_android_index.md.html
* Build your app: ionic build <PLATFORM>
* Simulate your app: ionic emulate <PLATFORM>
* Run your app on a device: ionic run <PLATFORM>
* Package an app using Ionic package service: ionic package <MODE> <PLATFORM>

For more help use ionic --help or ionic docs
Visit the Ionic docs: http://ionicframework.com/docs

Create an ionic.io account to send Push Notifications and use the Ionic View app
?
<Y/n>: n
C:\Users\NEWGEN\AppData\Roaming\npm\node_modules\ionic\lib\cli.js:474
  process.stdout.write('Unable to fetch', err, res.statusCode);
                                     ^
TypeError: Cannot read property 'statusCode' of undefined
    at Request._callback (C:\Users\NEWGEN\AppData\Roaming\npm\node_modules\ionic\lib\cli.js:474:55)
    at self.callback (C:\Users\NEWGEN\AppData\Roaming\npm\node_modules\ionic\node_modules\request\request.js:373:22)
    at emitOne (events.js:77:13)
    at Request.emit (events.js:169:7)
    at Request.onRequestError (C:\Users\NEWGEN\AppData\Roaming\npm\node_modules\ionic\node_modules\request\request.js:971:8)
    at emitOne (events.js:77:13)
    at ClientRequest.emit (events.js:169:7)
    at Socket.socketErrorListener (_http_client.js:258:9)
    at emitOne (events.js:77:13)
```

Figure IV.1 : Création d'un nouveau projet Ionic.

À partir du répertoire de l'application, il est possible de lancer un serveur local de test à l'aide de la commande :

```
$ ionic serve
```

### III.4. Déployer une application Ionic :

Avant de pouvoir émuler ou compiler l'application, Il faut installer les outils de développement spécifiques à chacune des plateformes (Android et iOS) :

- ✓ **Pour installer les outils iOS**, il faut être dans la sphère Apple et donc posséder une machine avec OSX et XCode [25].
- ✓ **Pour installer les outils Android**, c'est un peu plus ouvert car basé sur java, nous avons donc la possibilité d'utiliser au choix, Windows, OSX ou une distribution Linux [25].

Une fois l'environnement prêt, nous pouvons générer les fichiers nécessaires à la plateforme désirée :

```
$ ionic platform add android  
$ ionic platform add ios
```

Nous pouvons alors packager l'application pour la tester soit sur un émulateur, soit sur un périphérique réel.

- **Pour packager :**

```
$ ionic build android  
$ ionic build ios
```

- **Pour émuler :**

```
$ ionic emulate android  
$ ionic emulate ios
```

Pour déployer sur un périphérique de test, cela diffère selon la plateforme :

- **Pour Android :**

```
$ ionic run android
```

▪ **Pour iOS :**

Il existe deux solutions :

- ✓ Ouvrir le fichier XCode généré dans le dossier `platforms/ios` avec XCode lui-même, puis sélectionner le périphérique souhaité via l'interface et cliquer sur le bouton Play.
- ✓ Installer le packet « `npm ios-deploy` » en global afin de pouvoir lancer directement l'application sur le périphérique via la commande :

```
ionic run ios --device
```

### III.5. Tester une application Ionic :

Il est important de noter que l'APK généré par le packaging Android peut être envoyé à n'importe qui pour être testé sur son propre appareil mobile. Cet APK se trouve dans `platforms/android/build/outputs/apk/android-debug.apk`. La version iOS quand à elle ne peut être partagée facilement. Il est nécessaire de préparer une version beta via **iTunes Connect** et d'y inscrire des comptes Apple enregistrés auprès des machines de tests [25].

### III.6. Les outils utilisés par Ionic :

#### III.6.1. HTML (Hyper Text Markup Language):

Un langage dit de « marquage » (de « structuration » ou de « balisage »). Ces balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. Cela veut dire que l'on va gérer la façon dont un texte va s'afficher au sein du navigateur [26].

HTML n'est pas un langage de programmation au sens classique du terme, mais est essentiellement un ensemble de règles qui indiquent à un navigateur comment afficher une page du web. Il est souvent utilisé conjointement avec des langages de programmation telle que JavaScript et des formats de présentation (feuilles de style en cascade) [26].

#### III.6.2. CSS (Cascading Style Sheets) ou (feuilles de style en cascade) :

C'est un langage qui permet de gérer la présentation d'une page web. Les feuilles de style permettent de définir des règles appliquées à un ou plusieurs documents

HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc [27].

Le but de CSS est séparer la structure d'un document HTML et sa présentation. En effet, avec HTML, on peut définir à la fois la structure (le contenu de la hiérarchie entre les différentes parties d'un document) et la présentation. Mais cela pose quelques problèmes. Avec le couple HTML/CSS, on peut créer des pages web où la structure du document se trouve dans le fichier HTML tandis que la présentation se situe dans un fichier CSS [27].

### III.6.3. JavaScript :

JavaScript est un langage de script incorporé aux balises HTML, permettant d'améliorer la présentation et l'interactivité des pages web. La particularité du JavaScript consiste à créer des petits scripts sur une page HTML dans le but d'ajouter une petite animation ou un effet particulier sur la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur [28].

Les balises annonçant un code JavaScript sont les suivantes :

```
<SCRIPT langage= JavaScript >  
Mettez ici votre code  
</SCRIPT>
```

### III.6.4. AngularJs :

#### ❖ Définition :

AngularJS est un Framework JavaScript, développé en 2009. C'est une bibliothèque écrite en JavaScript et est distribué comme un fichier JavaScript, et peut être ajouté à une page Web avec une balise de script :

```
<script src= "https://ajax.googleapis.com/ajax/libs/angularjs/1.2.16/angular.min.js"></script>
```

AngularJS est un outil puissant qui permet de simplifier l'utilisation de JavaScript, d'ajouter de nouvelles fonctionnalités au langage mais aussi de structurer plus « proprement » les applications. Ses capacités en termes de simplification de la syntaxe JavaScript lui permettent ainsi de réduire considérablement les lignes de code nécessaires à la création d'une application [29].

❖ **Concepts d'AngularJS :**

Les concepts caractérisant AngularJS sont :

⇒ **Architecture MVC :**

Le patron de conception MVC (design pattern MVC) permet aux développeurs de diviser leurs codes sources en trois couches distinctes :

▪ **Modèle (Model) :**

Cette couche représente le cœur de l'application. Dans une application mobile, elle consiste souvent à récupérer les données à partir d'une base de données ou d'un service web et de les traiter en fonction de l'affichage souhaité par les utilisateurs [19].

▪ **Vue (View):**

Dans une application mobile, les vues permettent d'afficher les données récupérées à partir du modèle à l'utilisateur. Elles permettent aussi de capter les interactions utilisateurs sur les éléments graphiques de l'interface. Après une interaction utilisateur, les vues transmettent les événements détectés aux contrôleurs [19].

▪ **Contrôleur (Controller):**

Les contrôleurs s'occupent de faire le lien entre les vues et le modèle. Ils récupèrent les événements utilisateurs et en fonction de l'état dans lequel se trouve l'application, effectuent les actions souhaitées (changement de vue ...). Si ces actions nécessitent un changement de la couche modèle, le contrôleur s'occupe de demander ces changements. Ensuite, il notifie la vue concernée pour qu'elle mette à jour son état et affiche les nouvelles données [19].

La figure suivante présente les différentes interactions possibles entre ces trois couches :

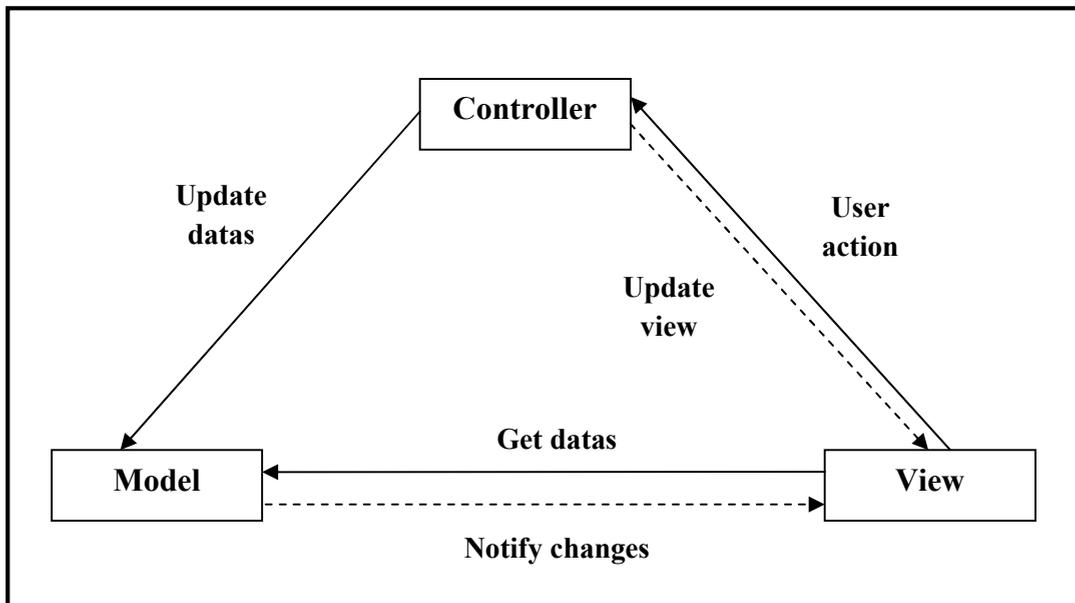


Figure IV.2 : Représentation du design pattern MVC [19].

⇒ **Data-binding bidirectionnel (liaison de données bidirectionnelles) :**

Le data-binding bidirectionnel est rendu possible grâce au scope. Le scope est le « liant » d'une application AngularJS, c'est lui qui contient les variables et fonctions qui font la liaison entre vues et contrôleurs. Il permet donc aux données de pouvoir être mise à jour par les vues et par le modèle [30].

⇒ **Injection de dépendance :**

L'injection de dépendance permet de charger certaines parties de l'application seulement quand c'est nécessaire [30].

⇒ **Routing (Routage) :**

AngularJS fournit un système de routage qui permet d'appeler certains composants pour une route donnée. Pour l'utiliser dans une application, il faut ajouter le script suivant [30] :

```
<script src= "https://code.angularjs.org/ 1.2.16/angular-route.min.js"></script>
```

### ⇒ Les tests:

Grâce aux tests, il est possible de s'assurer qu'un développement répond à la spécification initiale. Il permet ainsi de l'améliorer significativement. « **Un code qui n'est pas testable est un code qui n'est pas maintenable** » [30].

### III.6.5. Cordova :

#### ❖ Définition :

Maintenus par Adobe et la fondation Apache, Cordova et sa distribution PhoneGap sont des outils open source qui permettent de créer facilement et simultanément des applications mobiles pour plusieurs plateformes du marché, comme iOS, Android, Windows Phone, et bien d'autres encore. En effet, il suffit aujourd'hui d'un seul développement pour assurer un déploiement multiple vers les différents magasins d'applications (App Store, Google Play Store, etc.)[31].

#### ❖ Installation de Cordova:

L'installation de Cordova se fait depuis un package Node.js.

```
npm install -g cordova
```

#### ❖ Fonctionnement de la technologie Apache Cordova :

Le développement s'effectue toujours en trois étapes :

**RÉDACTION => COMPILATION => GÉNÉRATION**

##### 1. Rédaction du code source

La particularité de l'application mobile hybride est qu'elle est développée avec des langages web. Le contenu web est alors « encapsulé » avec une interface mobile. Dans le code source, le développeur fait d'abord appel au SDK de la technologie qu'il utilise (en l'occurrence Apache Cordova). Ce dernier va ensuite appeler les SDK des OS mobiles souhaités pour adapter le code source selon le système d'exploitation [32].

##### 2. Compilation du code source

Ce code source est ensuite compilé par Cordova. Selon les parties du code source et les SDK appelés en amont, le logiciel va générer des applications spécifiques aux systèmes d'exploitations visés [32].

### 3. Génération d'une ou plusieurs application(s) mobile(s) hybride(s)

Comme pour les applications mobiles natives générées, on obtient différents fichiers d'exécution de l'application mobile après compilation. Mais le contenu est très différent. Si on y regarde de plus près, on s'aperçoit que le code généré dans ces fichiers est mélangé [32]:

- une partie du code est toujours en langages web ;
- certaines parties restent en JavaScript Cordova ;
- enfin, seule une petite partie du code est du code binaire généré (nécessaire pour utiliser les fonctionnalités du téléphone).

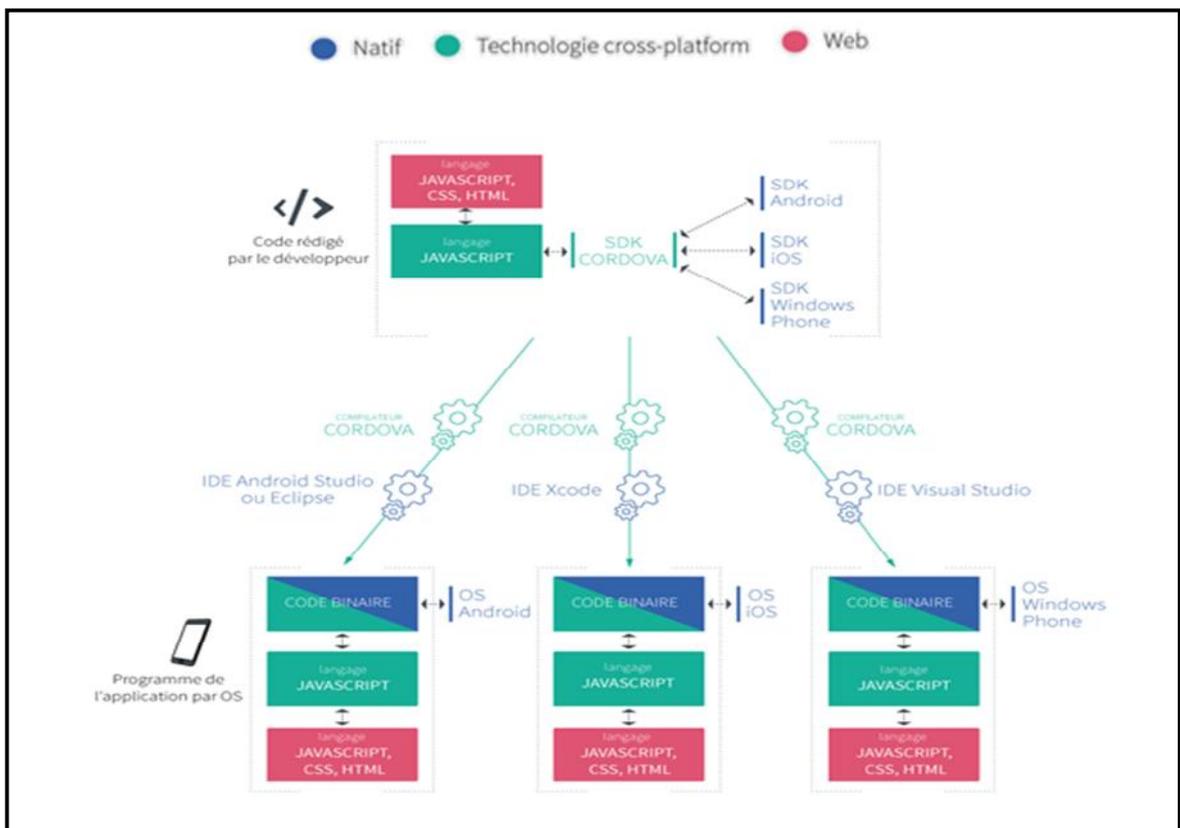


Figure IV.3 : Fonctionnement de la technologie Apache Cordova [32].

❖ **Les plugins :**

Les plugins permettent une interaction entre le smartphone et l'application web. Celle-ci tourne dans une webview et les plugins seront le lien entre cette webview et les API des différents outils disponibles dans le SDK du smartphone des utilisateurs.

- Pour découvrir la liste des plugins déjà installés, il suffit d'exécuter la commande suivante :

```
$ cordova plugin
```

- Pour ajouter un plugin à l'application, on tape la commande :

```
$ cordova plugin add nom_du_plugin
```

Voici quelques exemples de commandes pour ajouter des fonctionnalités à une application :

- **Camera :**

```
$ cordova plugin add cordova-plugin-camera
```

- **Géolocalisation :**

```
$ cordova plugin add cordova-plugin-geolocation
```

L'utilisation de **ngCordova** est fortement conseillée. Il s'agit d'un ensemble de plugins choisis et maintenus par l'équipe d'ionic, et donc toujours compatibles avec le Framework.

## IV. Présentation de node.js :

### IV.1. Définition:

Node.js est une plateforme de développement JavaScript. Ce n'est pas un serveur, ce n'est pas un Framework, c'est juste le langage JavaScript avec des bibliothèques permettant de

réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier.

## IV.2. Fonctionnement de Node.js :

Avant l'arrivée de Node.js, JavaScript avait toujours été utilisé du côté client, donc du côté de l'utilisateur. La figure suivante représente le fonctionnement classique d'une communication client-serveur :

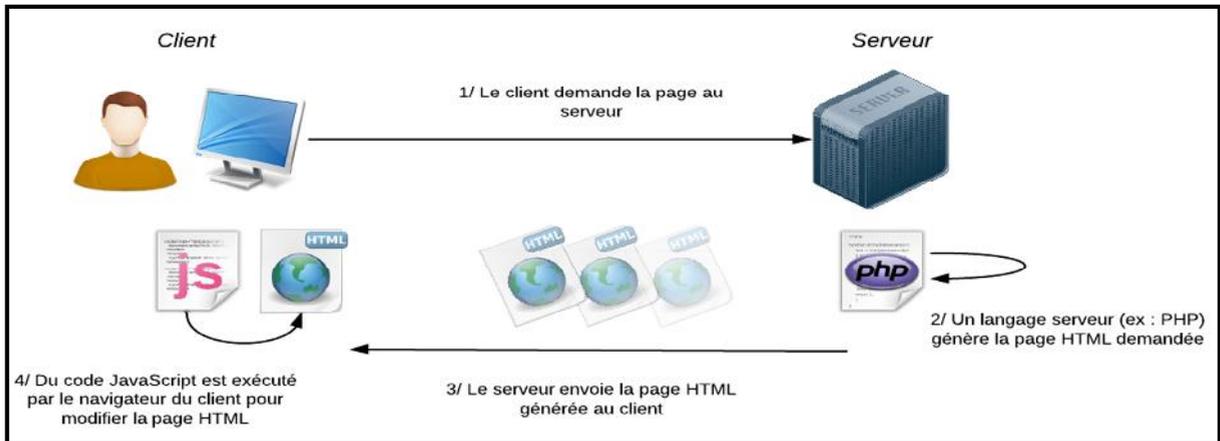


Figure IV.4: Le schéma classique - PHP coté le serveur, JavaScript coté client [33].

Depuis l'apparition de node.js, on peut bien évidemment toujours utilisé du JavaScript coté utilisateur. Par contre, node.js offre un environnement coté serveur qui nous permet aussi d'utiliser le langage JavaScript pour générer des pages web. En quelque sorte, node.js vient en remplacement de langages serveurs comme PHP, JAVA EE ...

Node.js est différent des autres langages serveurs, parce qu'il utilise JavaScript qui est basé sur les événements, et donc node.js est aussi basé sur les événements et c'est de la qu'il tire toute sa puissance et sa rapidité [33].

La figure suivante représente le fonctionnement d'une communication client-serveur avec node.js coté serveur :

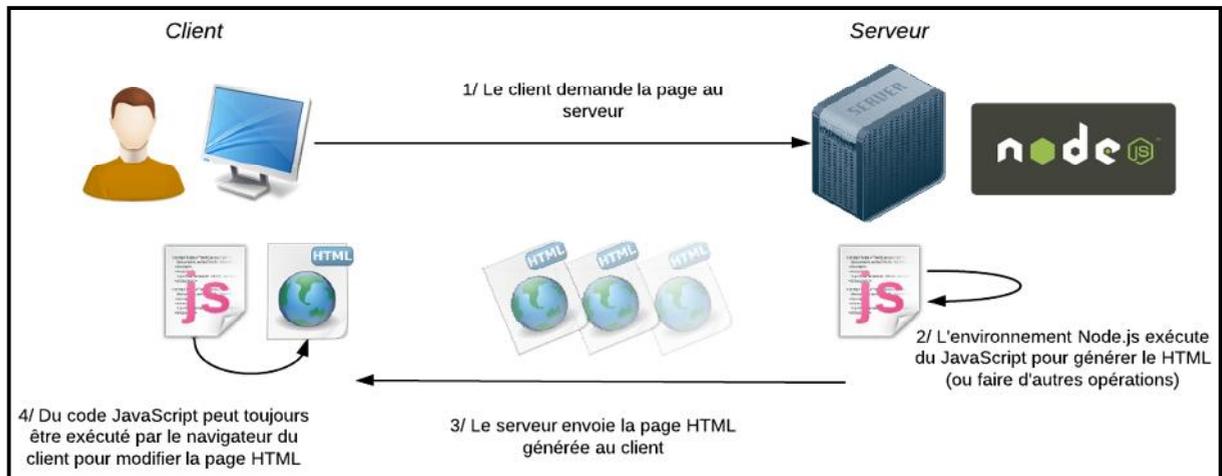


Figure IV.5 : Avec Node.js - JavaScript coté serveur et coté client [33].

### IV .3. Pourquoi node.js est-il rapide?

Si Node.js est rapide, cela tient principalement à deux choses : le moteur V8 et son fonctionnement non bloquant [33].

#### ❖ Le moteur V8 :

node.js utilise le moteur d'exécution ultra rapide V8 de Google Chrome. C'est un outil Open Source créé par Google, qui analyse et exécute du code JavaScript très rapidement [33].

#### ❖ Le modèle non bloquant :

Comme JavaScript est un langage conçu autour de la notion d'événement, node.js a pu mettre en place une architecture complètement non bloquante.

Dans les applications web, il est courant d'avoir des opérations longues et bloquantes (les appels aux bases de données, les appels à des services web). Node.js est conçu de façon à éviter de perdre du temps en nous permettant de faire autre chose en attendant que les actions longues soient terminées.

La figure suivante représente un exemple de fonctionnement du modèle non bloquant :

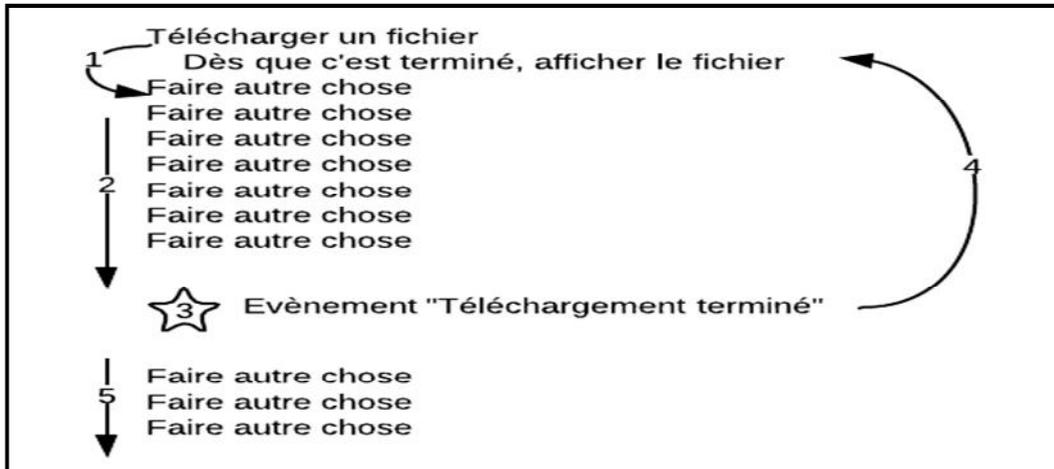


Figure IV.6 : Le modèle non bloquant en programmation [33].

❖ **Explication :**

Pour bien expliquer cela, on propose l'exemple suivant qui est une comparaison entre le modèle bloquant et le non-bloquant :

Si le modèle avait été bloquant, le programme aurait :

1. Lancé le téléchargement du fichier 1, et attendu qu'il se termine...
2. puis lancé le téléchargement du fichier 2, et attendu qu'il se termine.

Or, avec Node.js, les deux téléchargements sont lancés en même temps. Le programme n'attend pas la fin du premier téléchargement pour passer à l'instruction suivante.

Du coup, le téléchargement des 2 fichiers au total va beaucoup plus vite puisque le programme fait les 2 à la fois :

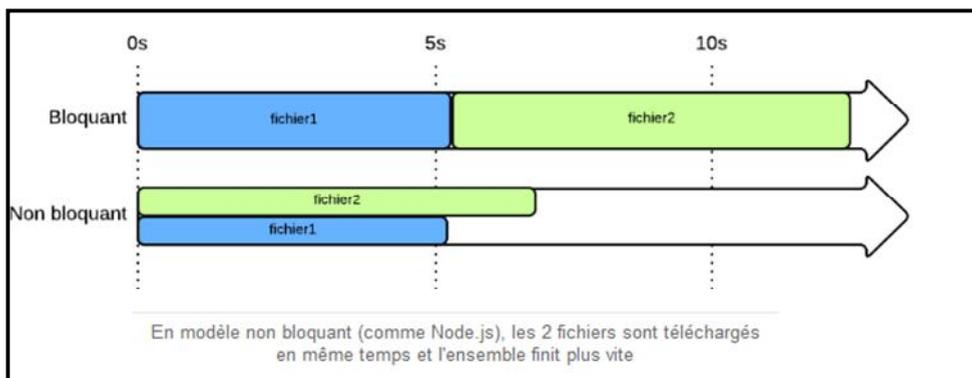


Figure IV.7 : La différence entre le modèle bloquant et le modèle non bloquant [33].

## IV.4. Installation de Node.js

Pour installer Node.js sous Windows, il suffit de télécharger l'installeur qui est proposé sur le site de Node.js. Il suffit de cliquer sur le lien **Install**.



Lancez ensuite l'installeur :

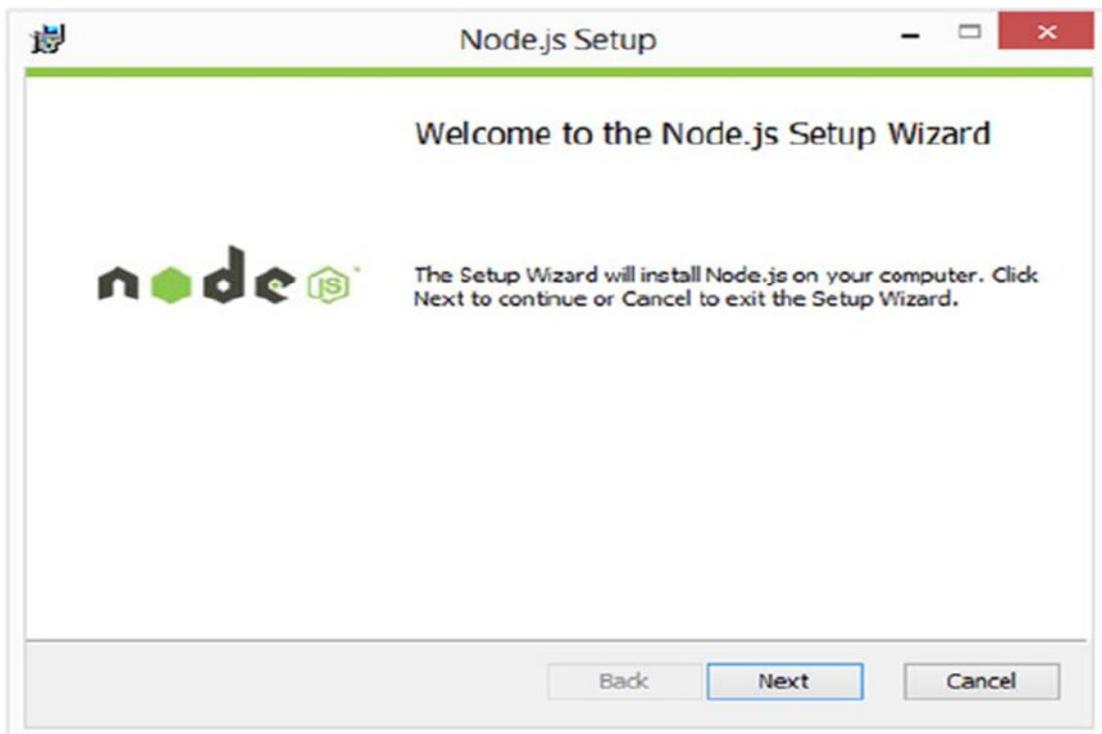


Figure IV.8 : Installation de node.js sous Windows.

Il est recommandé d'installer node.js dans le répertoire par défaut avec les paramètres par défaut, il est d'autant plus important de laisser l'installateur l'ajouter au PATH.

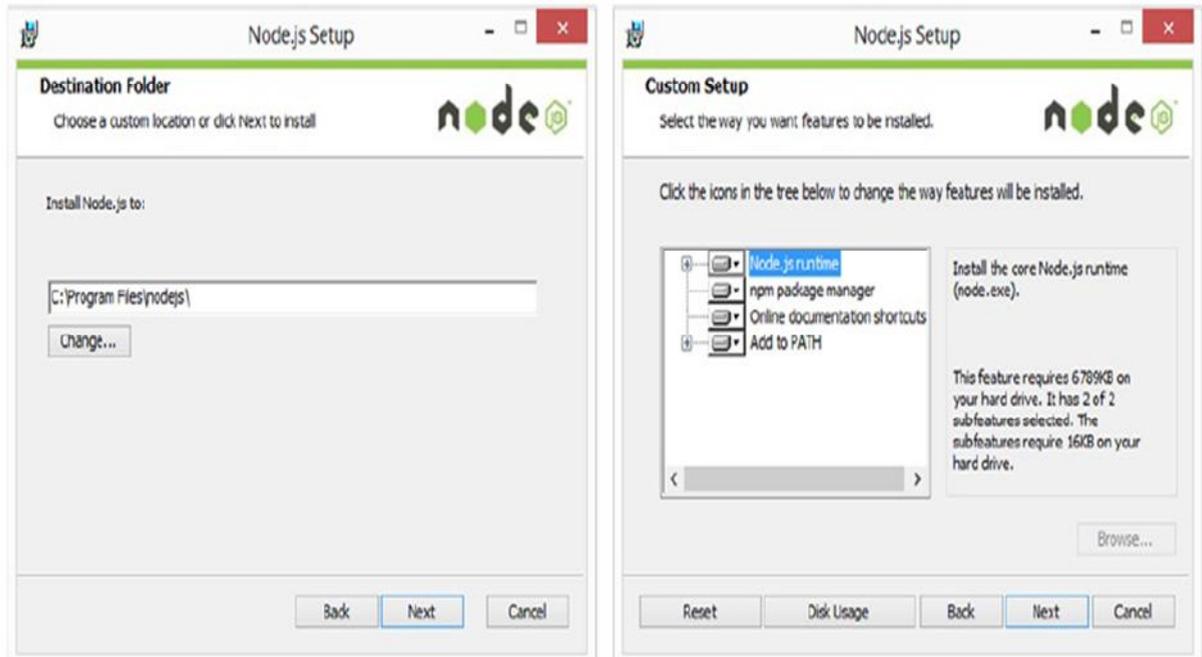


Figure IV.9 : Les paramètres par défaut pour l'installation de node.js sous Windows.

### IV.5. Configurer un serveur avec node.js :

On avait dit que node.js offrait un moyen de développer côté serveur, il est important de savoir qu'il ne met pas à disposition un serveur tout prêt, mais c'est au développeur de le configurer en précisant le numéro de port sur lequel il va écouter.

Pour cela, on se rend dans le dossier de l'application, on crée un fichier server.js et on ajoute le code ci-dessous :

```
const PORT=8081;
app.listen(PORT, function() {

  console.log("server listening on http://localhost:%s",PORT);
});
```

Figure IV.10: Création d'un serveur avec Node.js.

## IV.6. Node Package Manager (NPM) :

Il faut savoir que de base, le noyau de node.js est tout petit, il ne fait en fait pas grand chose. Pourtant, il est très riche grâce à son extensibilité. Ces extensions de Node.js sont appelées **modules**.

Il existe des milliers de modules qui offrent des fonctionnalités variées : de la gestion des fichiers uploadés à la connexion aux bases de données MySQL ou à MongoDB, en passant par des Framework, des systèmes de templates et la gestion de la communication temps réel avec le visiteur.

L'outil indispensable qui permet de télécharger facilement tous les modules de la communauté Node.js est **NPM**.

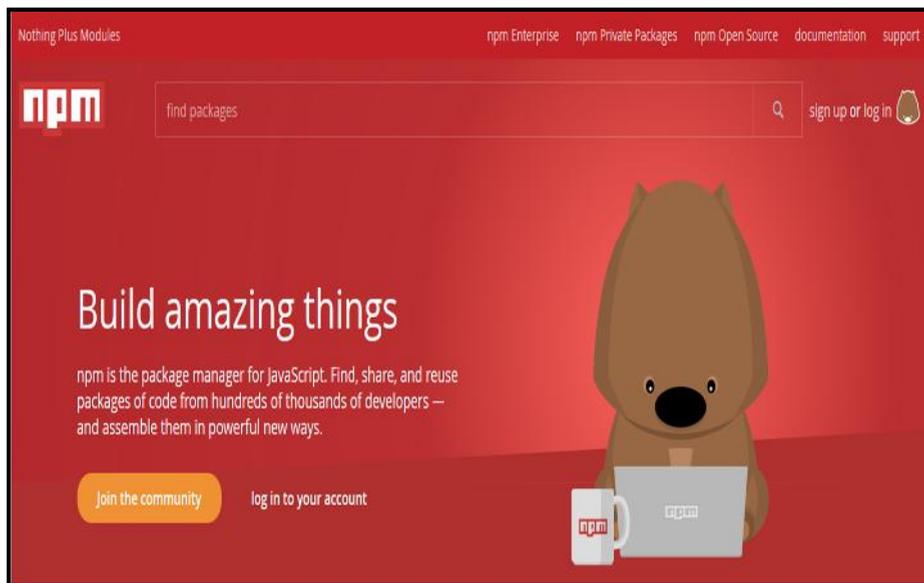


Figure IV.11 : Page d'accueil du site de téléchargement des modules NPM.

### ❖ Installer un module :

Pour installer un module, on se place dans le répertoire de l'application et on tape la commande :

```
npm install nom_du_module
```

Le module sera installé localement, spécialement pour le projet. Si on a un autre projet, il faudra donc relancer la commande pour l'installer à nouveau. Cela nous permet d'utiliser des versions différentes d'un même module en fonction de nos projets.

### ❖ **Mettre à jour les modules :**

Pour mettre à jour un module déjà installé, il suffit d'exécuter la commande suivante :

**npm update**

NPM va chercher sur les serveurs s'il y a de nouvelles versions des modules, puis mettre à jour les modules déjà installés (en veillant à ne pas casser la compatibilité) et il supprimera les anciennes versions.

### ❖ **Les modules utilisés dans notre projet :**

Pour notre projet nous avons installé les modules:

#### ➤ **HTTP :**

HyperText Transfer Protocol (Protocole de transfert hypertexte) est un protocole de la couche application du modèle TCP/IP, il définit comment le client et le serveur communiquent entre eux. Il peut fonctionner sur n'importe quelle connexion fiable. Les clients HTTP les plus connus sont les navigateurs web permettant à un utilisateur d'accéder à un serveur contenant les données [34].

Node.js utilise le module HTTP qui permet « le support de plusieurs fonctionnalités du protocole qui sont habituellement difficiles à utiliser ». En d'autres termes : il permet de répondre aux demandes des navigateurs (demande d'une page Internet par exemple). Pour que cela soit encore plus trivial, nous allons utiliser un module communautaire : le Framework Express qui permet de réaliser aisément des applications web.

#### ➤ **Express:**

Bien que node.js soit très utile, il reste bas niveau donc difficile d'aller vite, même si cela peut s'avérer utile dans certain cas, il peut être utile aussi d'avoir un outil pour pouvoir aller plus vite.

Si on cherche bien sur le site de NPM, on peut trouver **Express.js**, c'est un module très plébiscité. Il s'agit en fait d'un micro-Framework pour Node.js. Il nous fournit des outils de base pour aller plus vite dans la création d'applications Node.js.

La commande permettant d'installer express est :

```
npm install express
```

➤ **Cookie :**

Un cookie est un fichier texte généré par le serveur puis envoyé au client comme entête de message transmit via HTTP, puis enregistré dans le disque du client, le but du cookie est d'enregistrer des informations sur l'utilisateur pour les récupérer quand c'est nécessaire. Ces informations sont enregistrées en paire de nom-valeur (name-value pairs) [35].

Les cookies peuvent conserver une grande variété d'informations, incluant des informations d'identification personnelle (telles que votre nom, votre adresse postale, votre adresse électronique ou votre numéro de téléphone). Toutefois, ces informations ne peuvent être conservées que si vous les fournissez. Les sites web ne peuvent obtenir l'accès aux informations que vous ne leur avez pas fournies et ne peuvent accéder à d'autres fichiers sur votre appareil [36].

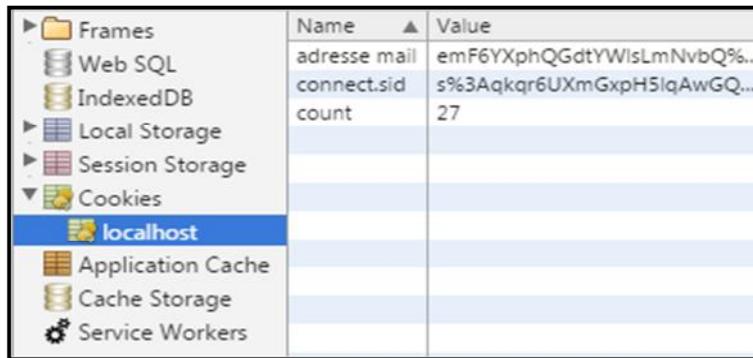
Comme pour tous les autres modules, pour l'installer il suffit de taper la commande :

```
npm install cookie-parser
```

🚩 **Remarque :**

Les cookies sont des fichiers textes ouverts pour la lecture, donc il y a lieux de penser à les crypter. Pour cela, on a utilisé le module « **btoa** » qui permet de crypter les informations de l'utilisateur.

Voici le résultat de l'utilisation de **cookie** et **btoa** après connexion d'un utilisateur :



Name	Value
adresse mail	emF6YXphQGdtYWIsLmNvbQ%...
connect.sid	s%3Aqkqr6UXmGxpH5IqAwGQ...
count	27

Figure IV.12 : Exemple d'utilisation de cookie et btoa.

➤ **Socket.io :**

Socket.io est une librairie JavaScript qui permet d'effectuer des communications asynchrones bidirectionnelles entre client et serveur.

Le web traditionnel fonctionne de la manière suivante : le client (le navigateur internet) envoie une requête au serveur. Le serveur, après avoir traité cette requête, retourne la réponse au client. On parle ici du mode PULL puisque c'est le client qui va chercher l'information [37].

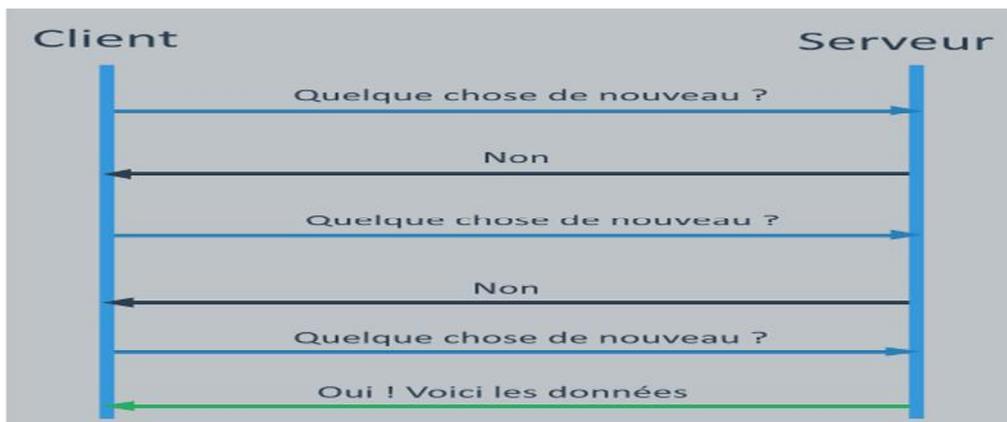


Figure IV.13 : Fonctionnement du web traditionnel [37].

Le web temps réel utilise le mode PUSH. Une fois que le client a accédé à une page web (de manière traditionnelle en PULL), le serveur peut par la suite en fonction d'événements envoyer des informations au client et vice et versa sans avoir à recharger la page.

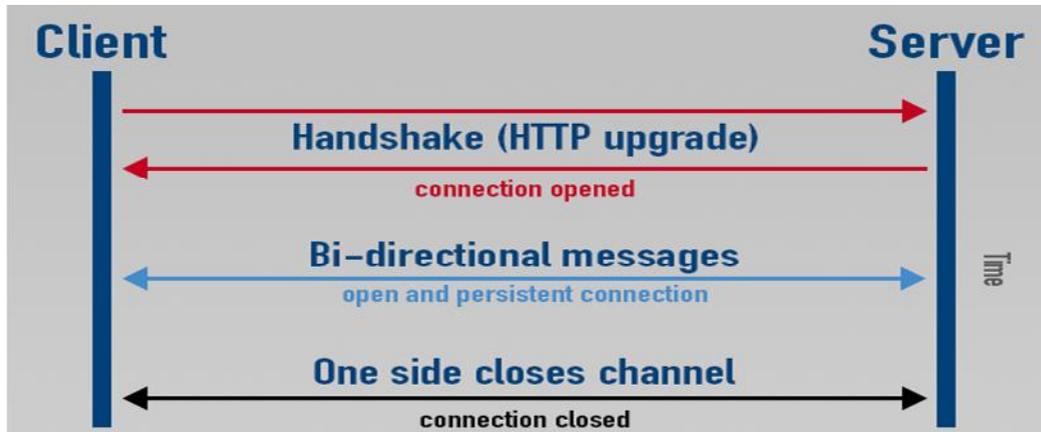


Figure IV.14 : Fonctionnement du web temps réel [37].

C'est le cas de Node.js qui est la solution la plus populaire depuis quelques années. Dans ce cas il suffit d'utiliser un module qui a fait ses preuves, tel que **Socket.io** ou **SockJs**. Ces bibliothèques contiennent deux parties, un package serveur s'exécutant via Node.js et un script JavaScript client permettant la communication avec le serveur.

```
<script src="/socket.io/socket.io.js"></script>
<script>
var socket = io.connect('http://localhost:8081');
socket.on('message', function (message){
  alert('message : ' + 'Bienvenue');
})
</script>
```

Figure IV.15 : Socket.io - script coté client.

```
var server = http.createServer(app);
var io = require('socket.io').listen(server);
const PORT=8081;
io.on('connection', function (socket){
  io.emit('message');
});
```

Figure IV.16 : Socket.io -script coté serveur.

✓ **Installer socket.io :**

La commande permettant d'installer socket.io est la suivante :

**npm install socket.io**

➤ **Express-session**

Une session est une période durant laquelle un utilisateur est connecté à une application, elle permet de conserver des informations sur ce dernier tout au long de cette période.

Express-session est un module NPM qui permet de gérer les sessions.

Pour l'installer, on exécute la commande suivante :

### npm install express-session

Pour l'utiliser dans l'application on ajoute la ligne :

```
var expressSession = require('express-session');
```

#### ➤ Body-parser

body-parser est un module qui permet d'extraire les données présente dans un flux de requete entrante et l'expose sur **req.body**, il utilise le format **JSON** pour l'affichage et la manipulation de ces données sous forme d'objet.

Pour l'installer, on exécute la commande suivante :

### npm install body-parser

Pour l'utiliser dans l'application on ajoute le code suivant:

```
var http = require('http');
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: false }));
```

#### ➤ Crypto

Le module Crypto offre un moyen d'encapsuler des informations d'identification sécurisée pour être utilisés dans une connexion http.

Pour accéder à ce module, il suffit d'utiliser cette ligne de code [38]:

```
var crypto = require('crypto');
```

Crypto met à disposition plusieurs méthodes de cryptage, parmi elles, celle utilisée dans notre travail. Elle s'ajoute comme suit :

```
var hash = crypto.createHash('sha256').update("phrase").digest('base64');
```

Voici le résultat de hachage de « phrase » :

```
D:\travail\Appli transport>node server.js  
wABi9+vcq0Lnerx1sEYEK2QFQpmibLoXvBBntNN1UqI=  
server listening on http://localhost:8081
```

## V. Présentation du langage de requête MYSQL:

### V.1. Définition de MYSQL :

MySQL est un système de gestion de bases de données relationnelles (SGBDR) permettant de manipuler des instructions adressées à la base de données sous forme de requêtes SQL.

C'est un langage de manipulation de base de données mis au point dans les années 70 par IBM. SQL ou (Standard Query Language) est un langage permettant d'interroger les bases de données de manière simple. Il est doté d'une syntaxe particulière que l'on doit respecter pour que la communication avec la base se passe au mieux.

A part le fait d'envoyer directement les requêtes SQL telles quelles au SGBD, le but ultime de l'utilisation de SQL sera aussi d'inclure ces requêtes SQL dans un programme écrit dans un autre langage. Ceci permet de coupler le SGBD à un langage informatique, donc à un programme.

Différents concepts tournent autour de cette définition citant : Base de données, SGBD et SGBDR.

#### 1. Base de données :

Une base de données informatique est un **ensemble de données** qui ont été stockées sur un support informatique, **organisées et structurées** de manière à pouvoir facilement consulter et modifier le contenu de la base.

Une base de données seule ne suffit donc pas, il est nécessaire d'avoir également:

- un **système permettant de gérer cette base.**
- un **langage pour transmettre des instructions** à la base de données.

#### 2. SGBD :

Un Système de Gestion de base de données (SGBD) est un **logiciel** (ou un ensemble de logiciels) permettant de **manipuler les données d'une base de données**. C'est-à-dire

sélectionner et afficher des informations tirées de cette base, modifier des données, en ajouter ou en supprimer (ce groupe de quatre opérations étant souvent appelé le **CRUD**, pour **Create, Read, Update, Delete**). MySQL est un système de gestion de bases de données.

### 3. SGBDR :

Un système de gestion de base de données relationnel (SGBDR), est un ensemble de programmes qui manipulent les données de base de données relationnelles avec des opérateurs issu de la théorie des ensembles.

Parmi les services qu'il peut rendre, nous citerons :

- une administration puissante et simple avec des outils simples à manipuler par les utilisateurs.
- Un temps de réponse plus court.
- l'intégration du web pour des applications Internet.
- Permettre l'accès de plusieurs bases réparties quel que soit l'emplacement réel de l'information.

### V.2. Installation de MYSQL:

Comme n'importe quel autre module, MYSQL peut être installé dans l'application grâce à la commande suivante:

```
npm install mysql
```

### V.3. Etablissement de la connexion avec la base de données MySQL:

Voici le code permettant d'établir la connexion avec le node.js et le serveur de base de données MYSQL :

```
var mysql      = require('mysql');
var connection = mysql.createConnection({
  host      : 'localhost',
  user      : 'root',
  password  : 'password',
  database  : 'nom_de_la_base'
});

connection.connect(function(err){
  if(err){
    console.log('Error connecting to Db');
    return;
  }
  console.log('Connection established');
});

connection.end(function(err) {
});
```

Figure V.17 : Code permettant d'établir la connexion avec la base de données MYSQL.

Ce code doit être écrit dans un fichier JavaScript (exemple : insert.js).

Pour tester si la connexion a été établit avec succès, il suffit d'aller vers le chemin où se trouve le fichier JavaScript et taper la commande suivante :

**node insert.js**

Le résultat d'exécution de la commande est le suivant :

```
C:\Users\NEWGEN>d:
D:\>cd travail
D:\travail>cd Appli_transport
D:\travail\Appli_transport>cd sql_node
D:\travail\Appli_transport\sql_node>node insert.js
Connection established
D:\travail\Appli_transport\sql_node>
```

Figure V.18 : Teste de la bonne connexion entre le node.js et MYSQL.

## VI. Présentation de quelques interfaces :

### VI.1. Page d'accueil :

C'est la première page qui s'affiche au lancement de l'application, le visiteur aura le choix entre l'inscription autant qu'utilisateur ou transporteur, ou bien s'il est déjà inscrit il pourra se connecter.

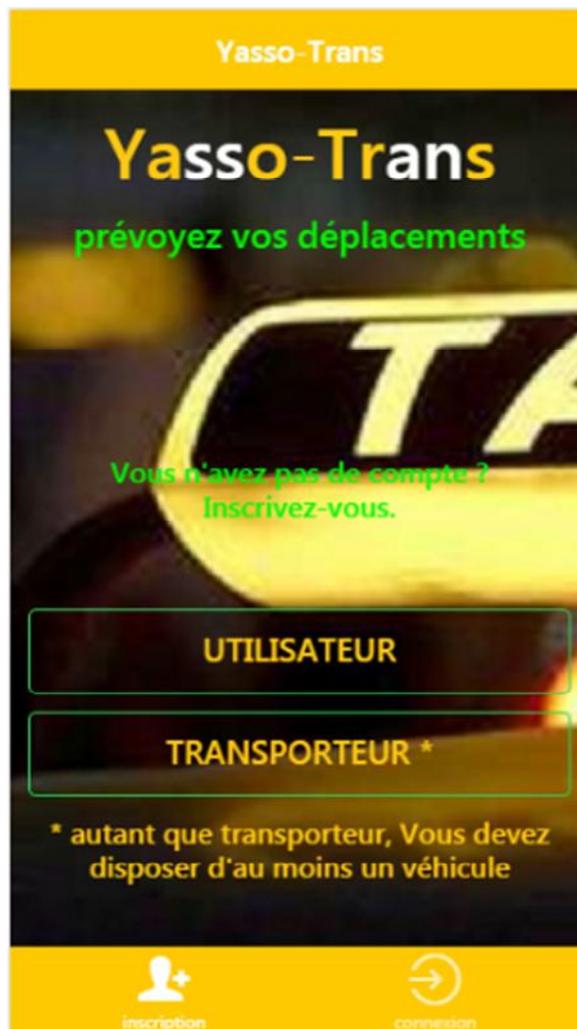


Figure IV.19 : interface de la page d'accueil.

## VI.2. Page d'authentification :

Sur cette page l'utilisateur ou le transporteur devra renseigner son adresse e-mail et son mot de passe afin d'accéder à son compte.

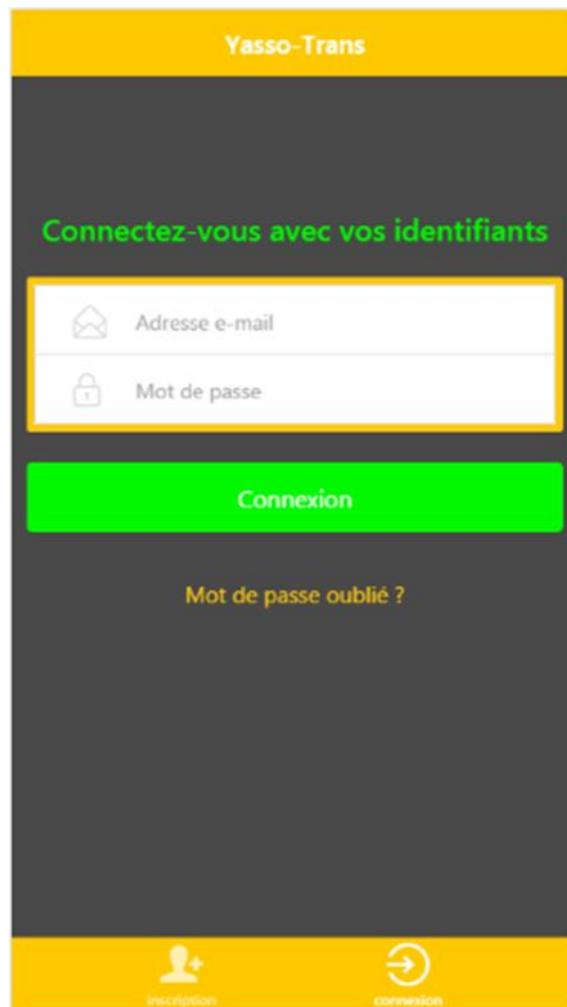


Figure IV.20 : interface d'authentification.

### VI.3. Page d'inscription de l'utilisateur:

Sur cette interface, l'utilisateur doit saisir ses informations personnelles plus un mot de passe et sa confirmation et clique sur le bouton « s'inscrire ». Si les informations sont correctes, il accède à son compte.

The screenshot shows a mobile application interface for user registration. At the top, there is a yellow header with a back arrow, the text 'Yesso-Trans', and the title 'UTILISATEUR'. Below the header, a green message reads 'Veuillez renseigner vos coordonnées'. The main content area is a white form with several input fields: 'Nom', 'Prénom', 'Téléphone mobile' (with an example 'ex: 0xxxxxxxx'), 'E-mail' (with an example 'ex: exemple@gmail.com'), 'Mot de passe', and 'Confirmer le mot de passe'. A large green button labeled 'S'inscrire' is positioned below the form. At the bottom, a yellow navigation bar contains two icons: a person with a plus sign labeled 'inscription' and a circular arrow labeled 'connexion'.

Figure IV.21 : Interface d’inscription de l’utilisateur.

#### VI.4. Page d’inscription du transporteur :

Sur cette interface, le transporteur doit saisir ses informations personnelles plus un mot de passe et sa confirmation puis il devra ajouter les informations relative à un véhicule et clique sur le bouton « s’inscrire ». Si les informations sont correctes, il accède à son compte.

The screenshot shows a mobile application interface for driver registration. At the top, there is a yellow header with a back arrow, the text 'Yasso-Trans', and the title 'TRANSPORTEUR'. Below the header is a white form with several input fields: 'Nom', 'Prénom', 'Téléphone mobile' (with an example 'ex: 0XXXXXXXXXX'), 'E-mail' (with an example 'ex: exemple@gmail.com'), 'Mot de passe', and 'Confirmer le mot de passe'. A green section header reads '" Informations concernant votre véhicule"'. Below this are three more input fields: 'Matricule' (with an example 'ex: 1122211315'), 'Nombre de places' (with an example 'ex: 5'), and 'Marque' (a dropdown menu). A large green button labeled 'S'inscrire' is positioned below the vehicle information fields. At the bottom of the screen is a yellow navigation bar with two icons: a person with a plus sign labeled 'inscription' and a circular arrow labeled 'connexion'.

Figure IV.22 : Interface d’inscription du transporteur.

### VI.5. Compte transporteur :

Après connexion ou inscription, le transporteur accède à son compte où il sera localisé automatiquement.

Le transporteur aura la possibilité d’afficher et modifier son profil, ajouter un ou plusieurs véhicules, modifier son mot de passe, consulter l’historique de ses courses ou retourner à sa position. Il pourra aussi consulter le guide de l’application.

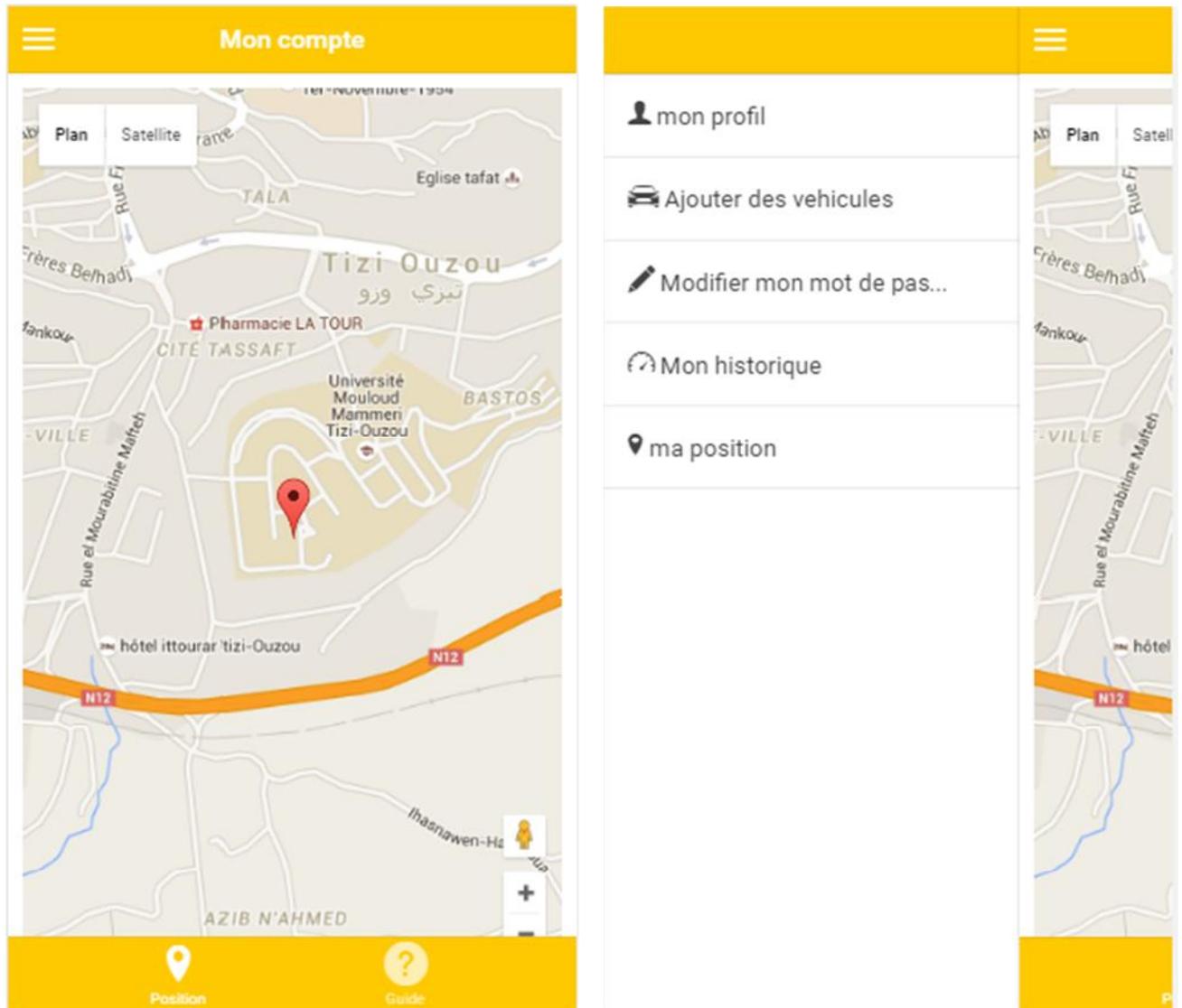


Figure IV.23 : Interface du compte transporteur.

## VI.6. Compte utilisateur :

Après connexion ou inscription, l'utilisateur accède à son compte où il pourra effectuer des commandes.

L'utilisateur aura la possibilité d'afficher et modifier son profil, modifier son mot de passe, consulter l'historique de ses courses. Il pourra aussi consulter le guide de l'application.

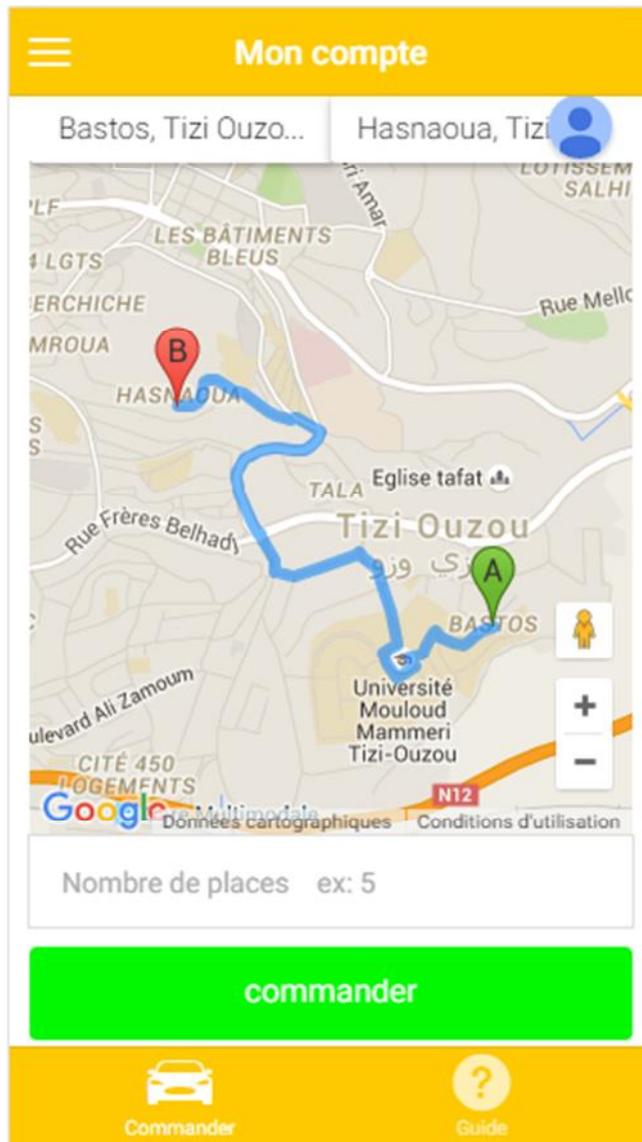


Figure IV.24 : Interface du compte utilisateur.

## Conclusion

Dans ce chapitre nous avons au premier lieu présenté l'environnement de développement et d'implémentation de notre application. Nous avons par la suite présenté quelques interfaces accompagnées d'une brève description.

***Conclusion générale  
et perspectives***

Le travail présenté s'inscrit dans le cadre des systèmes de transport intelligents.

Nous avons développé une application mobile multiplateforme permettant le transport routier de personnes en temps réel.

Nous avons commencé notre travail par la présentation des concepts des transports intelligents et des smart cities.

Ensuite, dans le second chapitre, nous avons décrit les applications mobiles et nous avons donné quelques exemples d'applications destinées au transport.

Enfin, dans les chapitres III et IV, nous avons conçu, analysé et réalisé notre application.

Le travail réalisé nous a permis d'acquérir des connaissances dans le domaine de la programmation, et de confronter nos connaissances en conception logicielle. Nous avons été confrontés à de nombreux problèmes et dans la plupart des cas nous avons pu trouver une solution alternative afin de les résoudre partiellement.

A l'état actuel, notre application est capable de mettre en contact un utilisateur demandant un service de transport, et un transporteur offrant ce service, tout en garantissant la notion de temps réel.

Dans un état futur, on pourrait imaginer comme perspective ajouter la fonctionnalité de transport de marchandises et celle du paiement électronique.

Nous souhaitons que notre travail soit un guide pour les nouvelles promotions.

— *Références*

*Bibliographiques* —

- [1] : [https://fr.wikipedia.org/wiki/Ville\\_intelligente](https://fr.wikipedia.org/wiki/Ville_intelligente)
- [2] : [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_transport\\_intelligent](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_transport_intelligent)
- [3] : « LA VILLE INTELLIGENTE COMME VECTEUR POUR LE DÉVELOPPEMENT DURABLE : LE CAS DE LA VILLE DE MONTRÉAL » Par Joëlle Simard, Juillet 2015.
- [4] : <http://dictionnaire.sensagent.com/syst%C3%A8me%20de%20transport%20intelligent/fr-fr/>
- [5] : <http://transport-intelligent.net/technologies/diffusion-de-l-information/>
- [6] : <http://transport-intelligent.net/technologies/localisation-78/>
- [7] : <http://transport-intelligent.net/technologies/systemes-cooperatifs/>
- [8] : <http://transport-intelligent.net/technologies/transmission/#technologies/>
- [9] : Brochure, Direction Générale des infrastructures, des transports et de la Mer, « Les systèmes de transport intelligents en France (STI) ».
- [10] : <http://www.transport-intelligent.net/champs-des-sti/securite-routiere/>
- [11] : <http://www.transport-intelligent.net/champs-des-sti/aides-a-la-mobilite/>
- [12] : <http://www.transport-intelligent.net/champs-des-sti/surete-dans-les-transports/>
- [13] : <http://www.transport-intelligent.net/champs-des-sti/gestion-des-transports-publics/>
- [14] : <http://www.definitions-webmarketing.com/>
- [15] : <http://www.definitions-marketing.com/definition/application-iphone/>
- [16] : <http://www.definitions-marketing.com/definition/application-android/>
- [17] : <http://www.appsolute.fr/natives-web-ou-hybrides-que-choisir/>
- [18] : <http://www.mobizel.com/2015/08/developpement-dune-application-mobile-hybride-33/>
- [19] : Joachim PERCHAT, Thèse de doctorat « Composants multiplateformes pour la prise en compte de l'hétérogénéité des terminaux mobiles », l'université de Valenciennes et du Hainaut-Cambrésis, 01/2015.
- [20] : <http://www.zdnet.fr/actualites/le-marche-mondial-des-apps-va-doubler-d-ici-a-2020-39832644.htm>

- [21] : « Gestion en temps réel du transport multimodal dans une application mobile ». Université d'Avignon et des pays de Vaucluse, Centre d'Enseignement et de recherche en Informatique. Année universitaire 2014/2015.
- [22] : UML 2 par la pratique, livre de Pascal Roques, édition EYROLLES
- [23] : <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes>
- [24] : <http://www.duchess-france.org/presentation-de-ionic-framework/>
- [25] : <http://makina-corpus.com/blog/metier/2015/bien-demarrer-avec-ionic>
- [26] : <http://glossaire.infowebmaster.fr/html/>
- [27] : <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/>
- [28] : <http://www.commentcamarche.net/contents/577-javascript-introduction-au-langage-javascript>
- [29] : <http://www.ib-formation.fr/catalogue/nbs-details/catref/universib-formations-informatiques-internet-et-mobilite-outils-et-langages-du-web/ref/li251/angularjs-developpement-dapplications-web>
- [30] : [http://www.synbioz.com/blog/introduction\\_a\\_angularjs](http://www.synbioz.com/blog/introduction_a_angularjs)
- [31] : <http://ebooks-ddl.com/ebooks/41490-applications-mobiles-avec-cordova-et-phonegap.html>
- [32] : <http://www.open-source-guide.com/Solutions/Developpement-et-couches-intermediaires/Frameworks-mobiles-cross-platform>
- [33] : <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert>
- [34] : [https://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- [35] : <http://computer.howstuffworks.com/cookie1.htm>
- [36] : <https://support.mozilla.org/fr/kb/cookies-informations-sites-enregistrent>
- [37] : <https://blog.bini.io/intro-websocket/>
- [38] : [https://nodejs.org/api/crypto.html#crypto\\_crypto\\_createhash\\_algorithm](https://nodejs.org/api/crypto.html#crypto_crypto_createhash_algorithm)